



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

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

(11)



1-0026721

(51)⁷ H04N 19/70

(13) B

- (21) 1-2015-04133 (22) 04/04/2014
(86) PCT/JP2014/001967 04/04/2014 (87) WO 2014/167817 16/10/2014
(30) 13/858,076 07/04/2013 US; 61/844,272 09/07/2013 US; 61/845,309 11/07/2013 US;
61/856,575 19/07/2013 US
(45) 25/12/2020 393 (43) 25/01/2017 346A
(73) Dolby International AB (SE)
Dolby International AB, Apollo Building, 3E Herikerbergweg 1-35 Amsterdam
Zuidoost, Netherlands 1101 CN
(72) DESHPANDE, Sachin G. (US).
(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) PHƯƠNG PHÁP GIẢI MÃ DÒNG BIT VIDEO

(57) Sáng chế đề xuất phương pháp giải mã dòng bit video, phương pháp này bao gồm các bước: thu dòng bit và các dòng bit nâng cao cùng với bước thu tập tham số video và phần mở rộng của tập tham số video. Phương pháp này còn bao gồm bước thu thông báo thay đổi của tập hợp lớp kết xuất có chứa thông tin chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất.

output_layer_sets_change(payloadSize) {	Tên
active_vps_id	u(4)
num_changed_output_layer_sets	ue(v)
for(i = 0; i < num_changed_output_layer_sets; i++) {	
changed_output_layer_set_idx_entry[i]	ue(v)
clsIdx = output_layer_set_idx[changed_output_layer_set_idx_entry[i]]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[clsIdx][j])	
output_layer_flag[clsIdx][j]	u(1)
}	
num_addl_output_layer_sets	
for(i = num_output_layer_sets; i < num_output_layer_sets + num_addl_output_layer_sets; i++) {	
addlIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIdx][j])	
output_layer_flag[addlIdx][j]	u(1)
}	
}	

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến các thiết bị điện tử. Cụ thể hơn, sáng chế đề cập đến các thiết bị điện tử để truyền tín hiệu các tham số của bộ giải mã tham chiếu giả định dựa trên hình ảnh phụ, và đến các hệ thống và các phương pháp vận hành hỗn hợp dùng cho bộ đệm hình ảnh giải mã (DPB - decoded picture buffer).

Tình trạng kỹ thuật của sáng chế

Các thiết bị điện tử ngày càng có kích thước nhỏ hơn và mạnh mẽ hơn để đáp ứng nhu cầu người dùng và nâng cao tính di động và tiện lợi. Người dùng ngày càng phụ thuộc vào các thiết bị điện tử và trông chờ chức năng được nâng cao. Một số ví dụ về các thiết bị điện tử bao gồm máy tính để bàn, máy tính xách tay, điện thoại di động, điện thoại thông minh, máy chơi đa phương tiện, mạch tích hợp v.v..

Một số thiết bị điện tử được sử dụng để xử lý và hiển thị các phương tiện số. Ví dụ, các thiết bị điện tử xách tay ngày nay cho phép các phương tiện số được sử dụng hầu như ở bất kỳ nơi nào, ở đó có người sử dụng. Ngoài ra, một số các thiết bị điện tử có thể tải xuống hoặc truyền suốt nội dung của phương tiện số để người dùng sử dụng và thưởng thức.

Tính phổ cập ngày càng tăng của các phương tiện số đã đặt ra một số vấn đề. Ví dụ, các phương tiện số chất lượng cao đòi hỏi lưu trữ phải thể hiện có hiệu quả, việc truyền và phát lại nhanh chóng đặt ra một số thách thức. Có thể thấy từ sáng chế là, các hệ thống và các phương pháp đại diện một cách hiệu quả cho phương tiện số với tính năng nâng cao có thể là có lợi.

Các mục đích trên và các mục đích khác, các dấu hiệu và ưu điểm của sáng chế sẽ được hiểu rõ hơn khi xem xét phần mô tả chi tiết dưới đây của sáng chế, cùng với các hình vẽ kèm theo.

Bản chất kỹ thuật của sáng chế

Giải pháp kỹ thuật

Theo một khía cạnh, sáng chế đề xuất phương pháp giải mã dòng bit video bao gồm các bước: (a) thu dòng bit cơ sở đại diện cho chuỗi video mã hóa; (b) thu các dòng bit nâng cao đại diện cho chuỗi video mã hóa đã nêu; (c) thu tập hợp tham số video chứa các phần tử cú pháp để áp dụng cho dòng bit cơ sở đã nêu và các dòng bit nâng cao đã nêu, trong đó tập hợp tham số video đã nêu chứa phần tử cú pháp truyền tín hiệu đến phần mở rộng tập hợp

tham số video; (d) thu phần mở rộng tập hợp tham số video đã nêu chứa các phần tử cú pháp liên quan đến ít nhất một trong các dòng bit nâng cao đã nêu; (e) thu thông báo thay đổi của tập hợp lớp kết xuất bao gồm thông tin chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất.

Theo một khía cạnh, sáng chế đề xuất phương pháp giải mã dòng bit video bao gồm các bước: (a) thu dòng bit cơ sở đại diện cho chuỗi video mã hóa; (b) thu các dòng bit nâng cao đại diện cho chuỗi video mã hóa đã nêu; (c) thu tập hợp tham số video chứa các phần tử cú pháp áp dụng cho dòng bit cơ sở đã nêu và các dòng bit nâng cao đã nêu, trong đó tập hợp tham số video đã nêu chứa phần tử cú pháp truyền tín hiệu đến phần mở rộng tập hợp tham số video; (d) thu phần mở rộng của tập hợp tham số video đã nêu chứa các phần tử cú pháp bao gồm các tham số liên quan đến bộ đệm hình ảnh giải mã dùng cho bộ đệm hình ảnh giải mã cho ít nhất một trong các dòng bit nâng cao đã nêu.

Theo một khía cạnh, sáng chế đề xuất phương pháp mã hóa video, bao gồm các bước: bắt đầu phân tích cú pháp tiêu đề phiên thứ nhất của hình ảnh hiện thời; xác định các bước nào được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) sẽ dựa trên hình ảnh và các bước nào sẽ dựa vào đơn vị truy cập (AU – access unit); thực hiện việc gỡ bỏ hình ảnh khỏi DPB; thực hiện việc kết xuất hình ảnh từ DPB; thực hiện việc giải mã và lưu trữ hình ảnh giải mã hiện thời trong DPB; đánh dấu hình ảnh giải mã hiện thời trong DPB; và thực hiện việc kết xuất hình ảnh bổ sung từ DPB.

Theo một khía cạnh, sáng chế đề xuất thiết bị điện tử được tạo cấu hình để mã hóa video, bao gồm: bộ xử lý; bộ nhớ nối thông điện tử với bộ xử lý, trong đó các lệnh được lưu trữ trong bộ nhớ có thể thực hiện được để: bắt đầu phân tích cú pháp tiêu đề phiên thứ nhất của hình ảnh hiện thời; xác định các bước nào được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) sẽ dựa vào ảnh và các bước nào sẽ dựa vào đơn vị truy cập (AU); thực hiện việc gỡ bỏ hình ảnh khỏi DPB; thực hiện việc kết xuất hình ảnh từ DPB; thực hiện việc giải mã và lưu trữ hình ảnh giải mã hiện thời trong DPB; đánh dấu hình ảnh giải mã hiện thời trong DPB; và thực hiện việc kết xuất hình ảnh bổ sung từ DPB.

Mô tả vắn tắt các hình vẽ

Fig.1A là sơ đồ khối thể hiện ví dụ về một hoặc nhiều thiết bị điện tử, trong đó các hệ thống và các phương pháp gửi thông báo và bước đệm dòng bit có thể được thực hiện.

Fig.1B là sơ đồ khối khác minh họa ví dụ về một hoặc nhiều thiết bị điện tử trong đó các hệ thống và các phương pháp gửi thông báo và bước đệm dòng bit có thể được thực hiện.

Fig.2 là sơ đồ khối thể hiện một cấu hình của phương pháp gửi thông báo.

Fig.3 là lưu đồ khối thể hiện một cấu hình của phương pháp xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập.

Fig.4 là lưu đồ khối thể hiện một cấu hình của phương pháp đệm dòng bit;

Fig.5 là lưu đồ khối thể hiện một cấu hình của phương pháp xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập.

Fig.6A là sơ đồ khối thể hiện một cấu hình của bộ mã hóa 604 trên thiết bị điện tử.

Fig.6B là sơ đồ khối khác minh họa một cấu hình của bộ mã hóa 604 trên thiết bị điện tử.

Fig.7A là sơ đồ khối thể hiện một cấu hình của bộ giải mã trên thiết bị điện tử.

Fig.7B là sơ đồ khối khác minh họa một cấu hình của bộ giải mã trên thiết bị điện tử.

Fig.8 minh họa các thành phần khác nhau có thể được sử dụng trong thiết bị điện tử truyền.

Fig.9 là sơ đồ khối thể hiện các thành phần khác nhau có thể được sử dụng trong thiết bị điện tử thu.

Fig.10 là sơ đồ khối thể hiện một cấu hình của thiết bị điện tử trong đó các hệ thống và các phương pháp gửi thông báo có thể được thực hiện.

Fig.11 là sơ đồ khối thể hiện một cấu hình của thiết bị điện tử trong đó các hệ thống và các phương pháp đệm dòng bit có thể được thực hiện.

Fig.12 là sơ đồ khối thể hiện một cấu hình của phương pháp vận hành bộ đệm hình ảnh giải mã.

Fig.13A minh họa cú pháp của tiêu đề đơn vị NAL khác.

Fig.13B minh họa cú pháp của tiêu đề đơn vị NAL khác.

Fig.13C minh họa cú pháp của tiêu đề đơn vị NAL khác.

Fig.14 minh họa cú pháp của đơn vị NAL chung.

Fig.15 minh họa tập tham số video hiện tại.

Fig.16 minh họa các loại khả năng mở rộng hiện tại.

Fig.17 minh họa tập tham số video làm ví dụ.

Fig.18 minh họa cú pháp ánh xạ khả năng mở rộng làm ví dụ.

Fig.19 minh họa tập tham số video làm ví dụ.

Fig.20 minh họa tập tham số video hiện tại.

Fig.21 minh họa loại kích thước hiện thời, cú pháp nhận dạng kích thước.

Fig.22 minh họa tập tham số video làm ví dụ.

Fig.23 minh họa cú pháp ánh xạ khả năng mở rộng làm ví dụ.

Fig.24 minh họa tập tham số video làm ví dụ.

Fig.25 minh họa tập tham số video làm ví dụ.

Fig.26 minh họa tập tham số video làm ví dụ.

Fig.27 minh họa cú pháp che khả năng mở rộng làm ví dụ.

Fig.28 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.29 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.30 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.31 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.32 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.33 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.34 minh họa cú pháp của tập tham số video làm ví dụ.

Fig.35 minh họa cú pháp mở rộng tập tham số video làm ví dụ.

Fig.36 minh họa cú pháp thay đổi các tập hợp lớp kết xuất làm ví dụ.

Fig.37 minh họa cú pháp thay đổi các tập hợp lớp kết xuất làm ví dụ.

Fig.38A minh họa cú pháp mở rộng tham số video làm ví dụ.

Fig.38B minh họa cú pháp mở rộng tham số video làm ví dụ.

Fig.39A minh họa cú pháp `op_dpb_info_parameters(j)` làm ví dụ.

Fig.39B minh họa cú pháp `op_dpb_info_parameters(j)` làm ví dụ.

Fig.40 minh họa cú pháp mở rộng tham số video làm ví dụ khác.

Fig.41 minh họa cú pháp `oop_dpb_info_parameters(j)` làm ví dụ khác.

Fig.42 minh họa cú pháp `oop_dpb_info_parameters(j)` làm ví dụ khác.

Fig.43 minh họa cú pháp `num_dpb_info_parameters` làm ví dụ.

Fig.44 minh họa cú pháp `oop_dpb_info_parameters(j)` làm ví dụ khác.

Fig.45 minh họa cú pháp `num_dpb_info_parameters` làm ví dụ khác.

Fig.46 minh họa cú pháp `num_dpb_info_parameters` làm ví dụ khác.

Fig.47 minh họa cú pháp mở rộng tham số video làm ví dụ khác và cú pháp `layer_dpb_info(i)`.

Fig.48 minh họa cú pháp `oop_dpb_info_parameters andlayer_dpb_info(i)` làm ví dụ.

Fig.49A minh họa cú pháp `vps_extension()` làm ví dụ khác.

Fig.49B minh họa cú pháp `vps_extension()` làm ví dụ khác.

Fig.50 minh họa cú pháp `oop_dpb_maxbuffering_parameters(i)` làm ví dụ.

Fig.51 minh họa cú pháp `layer_dpb_info_parameters(i)` làm ví dụ.

Fig.52 minh họa cú pháp `vps_extension()` làm ví dụ khác.

Fig.53 minh họa cú pháp `vps_extension()` làm ví dụ khác.

Fig.54 minh họa cú pháp `oop_dpb_maxbuffering_parameters(i,k)` làm ví dụ.

Fig.55 minh họa cú pháp `oop_dpb_maxbuffering_parameters(i,k)` làm ví dụ.

Fig.56 minh họa cú pháp `vps_extension()` làm ví dụ khác.

Fig.57 minh họa cú pháp `oop_dpb_maxbuffring_parameters(i,k)` làm ví dụ.

Fig.58 minh họa cú pháp `oop_dpb_maxbuffring_parameters(i,k)` làm ví dụ.

Fig.59 minh họa cú pháp `oop_dpb_maxbuffring_parameters(i,k)` làm ví dụ.

Fig.60 minh họa cú pháp `oop_dpb_maxbuffring_parameters(i,k)` làm ví dụ.

Fig.61 minh họa cú pháp `oop_dpb_maxbuffering_parameters(i,k)` làm ví dụ.

Fig.62 minh họa cú pháp `seq_parameter_set_rbsp()` làm ví dụ.

Fig.63 là sơ đồ khối thể hiện việc mã hóa video giữa nhiều thiết bị điện tử.

Fig.64 là lưu đồ của phương pháp vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp.

Fig.65 là lưu đồ của phương pháp khác vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp.

Fig.66 là sơ đồ khối thể hiện một cấu hình của bộ giải mã;

Fig.67A là sơ đồ khối thể hiện việc sử dụng cả lớp nâng cao và lớp cơ sở để mã hóa video với các bộ đệm hình ảnh giải mã (DPB) riêng và các môđun vận hành của bộ đệm hình ảnh giải mã (DPB) hỗn hợp riêng cho lớp cơ sở và lớp nâng cao.

Fig.67B là sơ đồ khối thể hiện việc sử dụng bộ đệm hình ảnh giải mã (DPB) dùng chung và các môđun vận hành của bộ đệm hình ảnh giải mã (DPB) hỗn hợp dùng chung cho lớp cơ sở và lớp nâng cao.

Fig.68 thể hiện sơ đồ định thời mô tả hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp.

Fig.69 là sơ đồ khối thể hiện kết cấu và định thời cho các đơn vị của lớp trừu tượng hóa mạng (NAL) có các lớp dùng cho các hình ảnh mã hóa và các đơn vị truy cập (AU) khi lớp nâng cao thứ hai (EL2) có thời gian ảnh thấp hơn lớp cơ sở (BL) và lớp nâng cao thứ nhất (EL1).

Fig.70 là sơ đồ khối thể hiện kết cấu và định thời dùng cho các đơn vị lớp trừu tượng hoá mạng (NAL) của các lớp đối với các hình ảnh mã hóa và các đơn vị truy cập (AU) khi lớp cơ sở (BL) có thời gian ảnh thấp hơn lớp nâng cao thứ nhất (EL1 – enhancement layer 1) và lớp nâng cao thứ hai (EL2 - enhancement layer 2).

Mô tả chi tiết sáng chế

Ví dụ 1

Thiết bị điện tử để gửi thông báo được mô tả dưới đây. Thiết bị điện tử này bao gồm bộ xử lý và các lệnh được lưu trữ trong bộ nhớ mà nó nối thông điện tử với bộ xử lý. Thiết bị điện tử này xác định, khi bộ đệm hình ảnh mã hoá (CPB) hỗ trợ hoạt động ở mức hình ảnh phụ, liệu có phải bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung trong thông báo thông tin nâng cao bổ sung (SEI - Supplemental Enhancement Information) định thời hình ảnh hay không. Thiết bị điện tử này còn tạo ra, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung sẽ có trong thông báo SEI định thời hình ảnh (hoặc thông

báo SEI nào đó hoặc tập tham số khác nào đó ví dụ tập tham số hình ảnh hoặc tập tham số chuỗi hoặc tập tham số video hoặc tập tham số tương thích), tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung, trong đó tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập từ CPB. Thiết bị điện tử này còn tạo ra, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung sẽ không có trong thông báo SEI định thời hình ảnh, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng cho mỗi đơn vị giải mã trong đơn vị truy cập. Thiết bị điện tử này còn gửi thông báo SEI định thời hình ảnh với tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hoặc các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã.

Tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB của đơn vị giải mã đứng ngay trước trước khi gỡ bỏ khỏi CPB đơn vị giải mã hiện thời trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Ngoài ra, khi đơn vị giải mã là đơn vị giải mã trong đơn vị truy cập, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB đơn vị giải mã sau cùng trong đơn vị truy cập được kết hợp với thông báo SEI về khoảng thời gian đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi CPB đơn vị giải mã thứ nhất trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Trái lại, khi đơn vị giải mã là đơn vị giải mã không phải là thứ nhất trong đơn vị truy cập, thì tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB đơn vị giải mã trước đó trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh trước khi gỡ bỏ khỏi CPB đơn vị giải mã hiện thời trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB đơn vị giải mã sau cùng trước khi gỡ bỏ khỏi CPB đơn vị giải mã thứ i trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Thiết bị điện tử này có thể tính toán các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã theo phần dư của môđun bộ đếm $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$, trong đó $\text{cpb_removal_delay_length_minus1} + 1$ là độ dài của tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung.

Thiết bị điện tử này cũng có thể tạo ra, khi CPB hỗ trợ hoạt động ở mức của đơn vị truy cập, thông báo SEI định thời hình ảnh bao gồm tham số trễ gỡ bỏ khỏi CPB xác định phải chờ bao nhiêu nhịp đồng hồ sau khi gỡ bỏ khỏi CPB của đơn vị truy cập được kết hợp với thông báo SEI với khoảng thời gian đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi CPB dữ liệu của đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Thiết bị điện tử này cũng có thể xác định liệu CPB có hỗ trợ hoạt động ở mức hình ảnh phụ hoặc mức của đơn vị truy cập hay không. Việc này có thể bao gồm việc xác định cờ định thời hình ảnh chỉ báo liệu bộ đệm hình ảnh mã hoá (CPB) có tạo ra các tham số hỗ trợ hoạt động ở mức hình ảnh phụ hay không dựa trên giá trị của cờ định thời hình ảnh. Cờ định thời hình ảnh này có thể có trong thông báo SEI định thời hình ảnh.

Việc xác định liệu có tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hay không có thể bao gồm việc thiết đặt cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung bằng 1 hay không, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có trong thông báo SEI định thời hình ảnh. Điều này cũng có thể bao gồm việc thiết đặt cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung là 0 khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung không có trong thông báo SEI định thời hình ảnh. Cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể có trong thông báo SEI định thời hình ảnh.

Thiết bị điện tử này cũng có thể tạo ra, khi CPB hỗ trợ hoạt động ở mức hình ảnh phụ, các tham số liên quan đến các đơn vị lớp trừu tượng hoá mạng (NAL) riêng chỉ báo lượng, được dịch đi một, các đơn vị NAL cho mỗi đơn vị giải mã trong đơn vị truy cập. Theo cách khác, hoặc ngoài, thiết bị điện tử có thể tạo ra tham số NAL chung chỉ báo lượng, được dịch đi một, các đơn vị NAL thông thường đối với mỗi đơn vị giải mã trong đơn vị truy cập.

Thiết bị điện tử dùng để đệm dòng bit cũng sẽ được mô tả. Thiết bị điện tử này bao gồm bộ xử lý và các lệnh được lưu trữ trong bộ nhớ nội thông điện tử với bộ xử lý. Thiết bị điện tử này xác định rằng CPB truyền tín hiệu các tham số ở mức hình ảnh phụ cho đơn vị truy cập. Thiết bị điện tử này cũng xác định, khi thông báo thông tin nâng cao bổ sung (SEI) định thời hình ảnh nhận được bao gồm cờ trễ gỡ bỏ khỏi bộ đệm hình ảnh mã hoá (CPB) của đơn vị giải mã chung, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập. Thiết bị điện tử này cũng xác định, khi thông báo SEI định thời hình ảnh không bao gồm cờ trễ gỡ bỏ khỏi CPB của đơn vị giải

mã chung, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng cho mỗi đơn vị giải mã trong đơn vị truy cập. Thiết bị điện tử này cũng gỡ bỏ các đơn vị giải mã khỏi CPB sử dụng tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hoặc các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng. Thiết bị điện tử này cũng giải mã các đơn vị giải mã trong đơn vị truy cập.

Theo một cấu hình, thiết bị điện tử này xác định rằng cờ định thời hình ảnh được đặt trong thông báo SEI định thời hình ảnh. Thiết bị điện tử này cũng có thể thiết đặt tham số trễ gỡ bỏ khỏi CPB, $cpb_removal_delay$, theo

Biểu thức 1

$$cpb_removal_delay = \frac{\left(\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] \right) * t_{c,sub}}{t_c}$$

trong đó $du_cpb_removal_delay[i]$ là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một và i là chỉ số.

Theo cách khác, thiết bị điện tử này có thể thiết đặt tham số trễ gỡ bỏ khỏi CPB, $cpb_removal_delay$ và $du_cpb_removal_delay[num_decoding_units_minus1]$ để thỏa mãn biểu thức:

Biểu thức 2

$$-1 \leq [cpb_removal_delay * t_c - \left(\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] \right) * t_{c,sub}] \leq 1$$

trong đó $du_cpb_removal_delay[i]$ là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một và i là chỉ số.

Theo cách khác, thiết bị điện tử có thể thiết đặt tham số trễ gỡ bỏ khỏi CPB, $cpb_removal_delay$ và $du_cpb_removal_delay[num_decoding_units_minus1]$ theo $cpb_removal_delay * t_c = du_cpb_removal_delay[num_decoding_units_minus1] * t_{c,sub}$ trong đó $du_cpb_removal_delay[num_decoding_units_minus1]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ $num_decoding_units_minus1$, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một.

Theo một cấu hình, thiết bị điện tử này xác định rằng cờ định thời hình ảnh được đặt trong thông báo SEI định thời hình ảnh. Thiết bị điện tử này cũng có thể thiết đặt các tham số trễ gỡ bỏ khỏi CPB, $cpb_removal_delay$ và $du_cpb_removal_delay[num_decoding_units_minus1]$ để thỏa mãn biểu thức: $-1 \leq (cpb_removal_delay * t_c - du_cpb_removal_delay[num_decoding_units_minus1] * t_{c,sub}) \leq 1$ trong đó $du_cpb_removal_delay[num_decoding_units_minus1]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ $num_decoding_units_minus1$, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một.

Biến số $ClockDiff$ có thể được xác định theo $ClockDiff = (num_units_in_tick - (num_units_in_sub_tick * (num_decoding_units_minus1 + 1)) / time_scale)$, trong đó $num_units_in_tick$ là số đơn vị thời gian của đồng hồ hoạt động ở tần số $time_scale$ Hz tương ứng với số gia lượng của bộ đếm nhịp đồng hồ, $num_units_in_sub_tick$ là số đơn vị thời gian của đồng hồ hoạt động ở tần số $time_scale$ Hz tương ứng với số gia lượng của số nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1 + 1$ là số lượng đơn vị giải mã trong đơn vị truy cập và $time_scale$ là số đơn vị thời gian đi qua trong một giây.

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, $low_delay_hrd_flag$) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1, CPB hoạt động ở mức hình ảnh phụ và $ClockDiff$ lớn hơn 0, thì thời gian gỡ bỏ đối với đơn vị giải mã m , $t_r(m)$ được xác định theo: $t_r(m) = t_{r,n}(m) + t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}) + ClockDiff$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, $low_delay_hrd_flag$) được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1, CPB hoạt động ở mức của đơn vị truy cập và $ClockDiff$ lớn hơn 0, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ được xác định theo: $t_r(n) = t_{r,n}(n) + t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c) - ClockDiff$, trong đó $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $Ceil()$ là biểu thức trần và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, $low_delay_hrd_flag$) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết

đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + \max((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/ t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/ t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + \max((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/ t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/ t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/ t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/ t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/ t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/ t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ dùng cho đơn vị giải mã m không phải là đơn vị giải mã sau cùng được thiết đặt tại $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ dùng cho đơn vị giải mã m là đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ dùng cho đơn vị giải mã m không phải là đơn vị giải mã sau cùng được thiết đặt tại $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ dùng cho đơn vị giải mã m là đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ dùng cho đơn vị giải mã m được thiết đặt tại $t_r(m) = t_{af}(m)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{af}(n)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Ngoài ra trong một số trường hợp cờ có thể được gửi một phần dòng bit để truyền tín hiệu trong số các biểu thức thay thế nêu trên được sử dụng để quyết định thời gian gỡ bỏ các

đơn vị giải mã và thời gian gỡ bỏ của đơn vị truy cập. Trong một trường hợp cờ này có thể được gọi là `du_au_cpb_alignment_mode_flag`. Nếu `du_au_cpb_alignment_mode_flag` bằng 1 thì các biểu thức nói trên sắp đặt hoạt động của CPB, CPB này hoạt động ở chế độ dựa vào hình ảnh phụ với CPB hoạt động ở chế độ của đơn vị truy cập được sử dụng. Nếu `du_au_cpb_alignment_mode_flag` là 0, thì các biểu thức nói trên không sắp đặt hoạt động của CPB hoạt động ở chế độ dựa vào hình ảnh phụ với CPB hoạt động ở chế độ của đơn vị truy cập được sử dụng.

Trong một trường hợp cờ `du_au_cpb_alignment_mode_flag` có thể được truyền tín hiệu bằng thông tin khả dụng video (VUI - video usability information). Trong trường hợp khác, cờ `du_au_cpb_alignment_mode_flag` có thể được gửi trong thông báo SEI định thời hình ảnh. Trong trường hợp khác, nữa cờ `du_au_cpb_alignment_mode_flag` có thể được gửi trong phần danh định khác của dòng bit. Một ví dụ về cú pháp và ngữ nghĩa cải biến theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện trong bảng (0) dưới đây.

Bảng 0

<code>pic_timing(payloadSize) {</code>
<code>if(CpbDpbDelaysPresentFlag) {</code>
<code>cpb removal delay</code>
<code>dpb output delay</code>
<code>if(sub pic cpb params present flag) {</code>
<code>num decoding units minus1</code>
<code>du_au_cpb_alignment_mode_flag</code>
<code>for(i = 0; i <= num decoding units minus1; i++) {</code>
<code>num nalus in du minus1[i]</code>
<code>du_cpb_removal_delay[i]</code>
<code>}</code>
<code>}</code>
<code>}</code>
<code>}</code>

Bảng (0)

Cần lưu ý rằng có thể sử dụng các ký hiệu (tên gọi) khác với những gì được sử dụng trên đây cho nhiều biến đổi. Ví dụ $t_r(n)$ về đơn vị truy cập n có thể được gọi là $CpbRemovalTime(n)$, $t_r(m)$ của đơn vị giải mã n có thể được gọi là $CpbRemovalTime(m)$, t_{c_sub} có thể được gọi là $ClockSubTick$, t_c có thể được gọi là $ClockTick$, $t_{af}(n)$ của đơn vị truy cập m có thể được gọi là $FinalArrivalTime(n)$ của đơn vị truy cập n , $t_{af}(m)$ của đơn vị giải mã m có thể được gọi là $FinalArrivalTime(m)$, $t_{r,n}(n)$ có thể được gọi là $NominalRemovalTime(n)$ của đơn vị truy cập n , $t_{r,n}(m)$ có thể được gọi là $NominalRemovalTime(m)$ của đơn vị giải mã m .

Phương pháp gửi thông báo bằng thiết bị điện tử này cũng sẽ được mô tả dưới đây. Phương pháp này bao gồm bước xác định, khi bộ đệm hình ảnh mã hoá (CPB) hỗ trợ hoạt động ở mức hình ảnh phụ, liệu có bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung vào thông báo thông tin nâng cao bổ sung (SEI) định thời hình ảnh hay không. Phương pháp này cũng bao gồm bước tạo ra, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có trong thông báo SEI định thời hình ảnh, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung, trong đó tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập từ CPB. Phương pháp này cũng bao gồm bước tạo ra, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung không có trong thông báo SEI định thời hình ảnh, thì tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng cho mỗi đơn vị giải mã trong đơn vị truy cập. Phương pháp này cũng bao gồm bước gửi thông báo SEI định thời hình ảnh cùng với tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hoặc các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã.

Phương pháp đệm dòng bit bằng thiết bị điện tử này cũng sẽ được mô tả dưới đây. Phương pháp này bao gồm bước xác định rằng CPB truyền tín hiệu các tham số ở mức hình ảnh phụ dùng cho đơn vị truy cập. Phương pháp này cũng bao gồm bước xác định, khi thông báo thông tin nâng cao bổ sung (SEI) định thời hình ảnh nhận được bao gồm cờ trễ gỡ bỏ khỏi bộ đệm hình ảnh mã hoá (CPB) của đơn vị giải mã chung, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập. Phương pháp này cũng bao gồm bước xác định, khi thông báo SEI định thời hình ảnh không bao gồm cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung, tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng cho mỗi đơn vị giải mã trong đơn vị truy cập. Phương pháp này cũng bao gồm bước gỡ bỏ các đơn vị giải mã khỏi CPB sử dụng tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hoặc các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng. Phương pháp này còn bao gồm bước giải mã các đơn vị giải mã trong đơn vị truy cập.

Các hệ thống và các phương pháp được bộc lộ trong sáng chế mô tả các thiết bị điện tử để gửi thông báo và đệm dòng bit. Ví dụ, các hệ thống và các phương pháp được bộc lộ trong sáng chế mô tả bước đệm dùng cho các dòng bit bắt đầu bằng các tham số hình ảnh phụ, trong một số cấu hình, các hệ thống và các phương pháp được bộc lộ trong sáng chế có thể mô tả việc truyền tín hiệu các tham số của bộ giải mã tham chiếu giả định (HRD) dựa trên hình ảnh phụ. Chẳng hạn, các hệ thống và các phương pháp được bộc lộ trong sáng chế mô tả cải biến đối với thông báo thông tin nâng cao bổ sung (SEI) định thời hình ảnh. Các

hệ thống và các phương pháp được bộc lộ trong sáng chế (ví dụ, sự cải biến HRD) có thể dẫn đến việc truyền tín hiệu gọn nhẹ hơn đối với các tham số khi mỗi hình ảnh phụ đi tới và được gỡ bỏ khỏi CPB theo các khoảng đều nhau.

Ngoài ra, khi các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ có mặt, thì bộ đệm hình ảnh mã hoá (CPB) có thể hoạt động ở mức của đơn vị truy cập hoặc mức hình ảnh phụ. Các hệ thống và các phương pháp hiện thời có thể còn áp đặt giới hạn dòng bit sao cho hoạt động của CPB dựa trên mức hình ảnh phụ và hoạt động của CPB của mức của đơn vị truy cập dẫn đến sự định thời giống nhau về việc gỡ bỏ đơn vị giải mã. Cụ thể là định thời gỡ bỏ đơn vị giải mã sau cùng trong đơn vị truy cập khi hoạt động ở chế độ hình ảnh phụ và định thời gỡ bỏ đơn vị truy cập khi hoạt động ở chế độ của đơn vị truy cập sẽ là giống nhau.

Cần lưu ý rằng mặc dù thuật ngữ “giả thiết” được sử dụng có liên quan đến HRD, HRD có thể được thực hiện theo phương pháp vật lý. Ví dụ, “HRD” có thể được sử dụng để mô tả việc thực hiện của bộ giải mã thực. Trong một số cấu hình, HRD có thể được thực hiện để xác định liệu dòng bit phù hợp với chuẩn mã hóa video hiệu suất cao (High Efficiency Video Coding - HEVC). Chẳng hạn, HRD có thể được sử dụng để xác định liệu các dòng bit Loại I và các dòng bit Loại II phù hợp với chuẩn HEVC hay không. Dòng bit Loại I có thể chỉ bao gồm các thiết bị của Lớp truy nhập mạng (NAL) của Lớp mã hóa video (VCL) và các đơn vị NAL của dữ liệu đột. Dòng bit Loại II có thể chứa các đơn vị NAL và các phần tử cú pháp bổ sung khác.

Tài liệu JCTVC-I0333 của Nhóm hợp tác chung về mã hóa video (Joint Collaborative Team on Video Coding - JCTVC) bao gồm HRD dựa trên hình ảnh phụ và hỗ trợ các thông báo SEI định thời hình ảnh. Chức năng này đã được kết hợp trong dự thảo của Hội đồng mã hóa video hiệu suất cao (High Efficiency Video Coding-HEVC) (JCTVC-I1003), toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn. B. Bros, W-J. Han, J-R. Ohm, G.J. Sullivan, Wang, and T-. Wiegand, “High efficiency video Coding (HEVC) text specification draft 10 (for DFIS & Last Call)”, JCTVC-J10003_v34, Geneva, January 2013 toàn bộ nội dung của nó được kết hợp trong bản mô tả này bằng cách viện dẫn. B. Bros, W-J. Han, J-R. Ohm, G.J. Sullivan, Wang, and T-. Wiegand, “High efficiency video Coding (HEVC) text specification draft 10”, JCTVC-L1003, Geneva, January 2013, toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn.

Một ví dụ về cú pháp và ngữ nghĩa cải biến theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện trong bảng (1) dưới đây.

Bảng 1

<code>pic_timing(payloadSize){</code>
<code> iff CpbDpbDelaysPresentFlag){</code>
<code> cpb_removal_delay</code>
<code> dpb_output_delay</code>
<code> iff(sub_pic_cpb_params_present_flag){</code>
<code> num_decoding_units_minus1</code>
<code> common_du_cpb_removal_delay_flag</code>
<code> iff(common_du_cpb_removal_delay_flag){</code>
<code> common_du_cpb_removal_delay</code>
<code> }</code>
<code> for(i=0; i<=num_decoding_units_minus1; i++){</code>
<code> num_nalus_in_du_minus1[i]</code>
<code> iff(common_du_cpb_removal_delay_flag)</code>
<code> du_cpb_removal_delay[i]</code>
<code> }</code>
<code> }</code>
<code> }</code>
<code>}</code>

Bảng (1)

Các ví dụ về ngữ nghĩa thông báo SEI về khoảng đệm theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện dưới đây. Cụ thể, các chi tiết bổ sung về ngữ nghĩa của các phần tử cú pháp cải biến được thể hiện dưới đây. Khi NALHrdBpPresentFlag hoặc VCLHrdBpPresentFlag đều bằng 1, thông báo SEI về khoảng đệm có thể được kết hợp với đơn vị truy cập bất kỳ trong dòng bit và thông báo SEI về khoảng đệm có thể được kết hợp với mỗi đơn vị truy cập IDR, với mỗi đơn vị truy cập CRA và với mỗi đơn vị truy cập được kết hợp với thông báo SEI điểm phục hồi. Đối với một số ứng dụng, sự có mặt thường xuyên của thông báo SEI về khoảng đệm có thể được mong muốn. Khoảng đệm được xác định là tập hợp của các đơn vị truy cập giữa hai thời điểm của thông báo SEI về khoảng đệm theo thứ tự giải mã.

'seq_parameter_set_id' xác định tập tham số chuỗi chứa các đặc tính HDR dãy. Giá trị seq_parameter_set_id có thể bằng giá trị seq_parameter_set_id trong tập tham số hình ảnh được tham chiếu bởi hình ảnh mã hoá sơ cấp được kết hợp với thông báo SEI về khoảng đệm. Giá trị seq_parameter_set_id có thể nằm trong khoảng từ 0 đến 31, bao gồm cả giá trị đầu cuối.

'initial_cpb_removal_delay'[SchedSelIdx] xác định khoảng trễ cho CPB thứ SchedSelIdx giữa thời gian đi đến CPB của bit thứ nhất của dữ liệu mã hoá được kết hợp với đơn vị truy cập được kết hợp với thông báo SEI về khoảng đệm và thời gian gỡ bỏ khỏi CPB của dữ liệu mã hoá được kết hợp với cùng một đơn vị truy cập, đối với khoảng đệm

thứ nhất sau khi khởi động HRD. Phần tử cú pháp có các độ dài theo bit được xác định là $\text{initial_cpb_removal_delay_length_minus1} + 1$. Nó được tính theo đơn vị xung nhịp 90 kHz. $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ có thể không bằng 0 và có thể không vượt quá $90000 * (\text{CpbSize}[\text{SchedSelIdx}] / \text{BitRate}[\text{SchedSelIdx}])$, đương lượng thời gian của kích thước CPB được tính theo các đơn vị thời gian 90 kHz.

' $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ ' được sử dụng cho SchedSelIdx-th CPB phối hợp với cpb_removal_delay để xác định thời gian cung cấp ban đầu của các đơn vị truy cập mã hoá tới CPB. $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ là theo đơn vị xung nhịp 90 kHz. Phần tử cú pháp $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ mã có độ dài cố định mà độ dài của nó theo bit được xác định là $\text{initial_cpb_removal_delay_length_minus1} + 1$. Phần tử cú pháp này không được sử dụng bởi các bộ giải mã và chỉ cần cho bộ lập lịch biểu (HSS) (ví dụ, như được xác định trong Phụ lục C của JCTVC-I1003).

Trên toàn bộ chuỗi video mã hóa, tổng của $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ và $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ có thể không đổi đối với mỗi giá trị của SchedSelIdx.

' $\text{initial_du_cpb_removal_delay}[\text{SchedSelIdx}]$ ' xác định khoảng trễ cho CPB thứ SchedSelIdx-th giữa thời gian đi đến CPB của bit thứ nhất của dữ liệu mã hoá được kết hợp với đơn vị giải mã thứ nhất trong đơn vị truy cập được kết hợp với thông báo SEI về khoảng đệm và thời gian gỡ bỏ khỏi CPB của dữ liệu mã hoá được kết hợp với cùng một đơn vị giải mã, đối với khoảng đệm thứ nhất sau khi khởi động HRD. Phần tử cú pháp có các độ dài theo bit được xác định là $\text{initial_cpb_removal_delay_length_minus1} + 1$. Nó được tính theo đơn vị xung nhịp 90 kHz. $\text{initial_du_cpb_removal_delay}[\text{SchedSelIdx}]$ có thể không bằng 0 và có thể không vượt quá $90000 * (\text{CPBSize}[\text{SchedSelIdx}] / \text{BitRate}[\text{SchedSelIdx}])$, đương lượng thời gian của kích thước CPB được tính theo các đơn vị thời gian 90 kHz.

' $\text{initial_du_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ ' được sử dụng cho SchedSelIdx-th CPB phối hợp với cpb_removal_delay để xác định thời gian cung cấp ban đầu của các đơn vị giải mã tới CPB.

$\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ được tính theo xung nhịp 90 kHz. Phần tử cú pháp $\text{initial_du_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ là mã có độ dài cố định mà độ dài của nó theo bit được xác định là $\text{initial_cpb_removal_delay_length_minus1} + 1$.

Phần tử cú pháp này không được sử dụng bởi các bộ giải mã và chỉ cần cho bộ lập lịch biểu (HSS) (ví dụ, như được xác định trong Phụ lục C của JCTVC-I1003).

Trên toàn bộ chuỗi video mã hóa, tổng của `initial_du_cpb_removal_delay[SchedSelIdx]` và `initial_du_cpb_removal_delay_offset[SchedSelIdx]` có thể không đổi đối với mỗi giá trị của `SchedSelIdx`.

Các ví dụ về ngữ nghĩa thông báo SEI định thời hình ảnh theo các hệ thống và các phương pháp được bộc lộ trong sáng chế sẽ được thể hiện dưới đây. Cụ thể, các chi tiết bổ sung về ngữ nghĩa của các phần tử cú pháp cải biến sẽ được thể hiện dưới đây.

Cú pháp của thông báo SEI định thời hình ảnh phụ thuộc vào nội dung của tập tham số chuỗi hoạt hoá đối với hình ảnh mã hoá được kết hợp với thông báo SEI định thời hình ảnh. Tuy nhiên, trừ phi thông báo SEI định thời hình ảnh của đơn vị truy cập làm mới giải mã tức thì (IDR - Instantaneous Decoding Refresh) được đề cập trước bởi thông báo SEI về khoảng đệm bên trong cùng một đơn vị truy cập, việc kích hoạt tập tham số chuỗi được liên kết (và, đối với các hình ảnh IDR không phải là hình ảnh thứ nhất trong dòng bit, việc xác định rằng hình ảnh mã hoá đó là hình ảnh IDR) không xuất hiện cho tới khi giải mã đơn vị lớp trừu tượng hoá mạng (NAL) của phiên mã hoá của hình ảnh mã hoá đó. Do đơn vị NAL của phiên mã hoá của hình ảnh mã hoá sẽ theo sau thông báo SEI định thời hình ảnh theo thứ tự đơn vị NAL, có thể có các trường hợp trong đó cần thiết đối với bộ giải mã lưu trữ chuỗi byte thô (RBSP) có chứa thông báo SEI định thời hình ảnh cho tới khi xác định các tham số của tham số chuỗi hoạt hoá đối với hình ảnh mã hoá và sau đó thực hiện việc phân giải thông báo SEI định thời hình ảnh.

Sự có mặt của thông báo SEI định thời hình ảnh trong dòng bit được xác định như sau. Nếu `CpbDpbDelaysPresentFlag` bằng 1, một thông báo SEI định thời hình ảnh có thể có mặt trong mỗi đơn vị truy cập của chuỗi video giải mã. Theo cách khác, thì (`CpbDpbDelaysPresentFlag` bằng 0), không có các thông báo SEI định thời hình ảnh có thể có mặt trong bất kỳ đơn vị truy cập nào của chuỗi video mã hóa.

'`cpb_removal_delay`' xác định phải chờ bao nhiêu nhịp đồng hồ (xem khoản E.2.1 của JCTVC-I1003) sau khi gỡ bỏ khỏi CPB của đơn vị truy cập được kết hợp với thông báo SEI về khoảng đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi bộ đệm dữ liệu của đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh. Giá trị này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu của đơn vị truy

cập vào CPB đối với HSS, như được xác định trong Phụ lục C của JCTVC-I1003. Phần tử cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định là $\text{cpb_removal_delay_length_minus1} + 1$. cpb_removal_delay là phần dư của bộ đếm $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ môđun.

Giá trị $\text{cpb_removal_delay_length_minus1}$ xác định độ dài (bằng bit) của phần tử cú pháp cpb_removal_delay là giá trị $\text{cpb_removal_delay_length_minus1}$ đã mã hoá trong tập tham số chuỗi hoạt hoá đối với hình ảnh mã hoá sơ cấp được kết hợp với thông báo SEI định thời hình ảnh, mặc dù cpb_removal_delay xác định nhịp đồng hồ liên quan đến thời gian gỡ bỏ của đơn vị truy cập đứng trước có chứa thông báo SEI về khoảng đệm, có thể là đơn vị truy cập của chuỗi video mã hóa khác.

' dpb_output_delay ' được sử dụng để tính toán thời gian đầu ra của bộ đệm hình ảnh giải mã (DPB) của hình ảnh. Nó xác định phải chờ bao nhiêu nhịp đồng hồ sau khi gỡ bỏ đơn vị giải mã sau cùng trong đơn vị truy cập từ CPB trước khi hình ảnh giải mã được kết xuất từ DPB (xem khoản mục C.2 của JCTVC-I1003).

Liên quan đến DPB, ảnh không được gỡ bỏ từ DPB tại thời điểm đầu ra của hình ảnh khi ảnh được đánh dấu là “sử dụng để tham chiếu ngắn hạn” hoặc “sử dụng để tham chiếu dài hạn”. Chỉ một dpb_output_delay được xác định đối với một ảnh được giải mã. Độ dài của phần tử cú pháp dpb_output_delay được tính theo bit bằng $\text{dpb_output_delay_length_minus1} + 1$. Khi $\text{max_dec_pic_buffering}[\text{max_temporal_layers_minus1}]$ bằng 0, thì dpb_output_delay có thể bằng 0.

Thời gian kết xuất bắt nguồn từ dpb_output_delay của hình ảnh bất kỳ được kết xuất từ bộ giải mã phù hợp định thời kết xuất như được xác định trong khoản mục C.2 của JCTVC-I1003 có thể trước thời gian kết xuất bắt nguồn từ dpb_output_delay của tất cả các hình ảnh trong bất kỳ chuỗi video mã hóa tiếp theo theo thứ tự giải mã. Thứ tự kết xuất hình ảnh được thiết lập bởi các giá trị của phần tử cú pháp này có thể giống như thứ tự như được thiết lập bởi các giá trị của $\text{PicOrderCnt}()$ như được xác định bởi khoản đó. Đối với các hình ảnh không được kết xuất bởi quá trình “đệm” của khoản đó vì chúng đứng trước, theo thứ tự giải mã, hình ảnh IDR cùng với $\text{no_output_of_prior_pics_flag}$ bằng 1 hoặc được suy ra bằng 1, thời gian kết xuất nhận được từ dpb_output_delay có thể tăng lên với việc tăng giá trị của $\text{PicOrderCnt}()$ liên quan đến tất cả các hình ảnh trong cùng chuỗi video mã hóa.

'num_decoding_units_minus1' + 1 xác định số lượng các đơn vị giải mã trong đơn vị truy cập mà thông báo SEI định thời hình ảnh liên quan. Giá trị num_decoding_units_minus1 có thể nằm trong khoảng từ 0 đến PicWidthInCtbs * PicHeightInCtbs - 1, bao gồm cả giá trị đầu cuối.

'common_du_cpb_removal_delay_flag' bằng 1 xác định rằng phần tử cú pháp common_du_cpb_removal_delay có mặt.

'common_du_cpb_removal_delay_flag' bằng 0 xác định rằng phần tử cú pháp common_du_cpb_removal_delay không có mặt.

'common_du_cpb_removal_delay' xác định thông tin như sau: Nếu đơn vị giải mã là đơn vị giải mã thứ nhất trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh thì common_du_cpb_removal_delay xác định phải chờ bao nhiêu nhịp đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) phải chờ sau khi gỡ bỏ khỏi CPB đơn vị giải mã sau cùng trong đơn vị truy cập được kết hợp với thông báo SEI về khoảng đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi CPB đơn vị giải mã thứ nhất trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh.

Theo cách khác, common_du_cpb_removal_delay xác định phải chờ bao nhiêu nhịp đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) sau khi gỡ bỏ khỏi CPB đơn vị giải mã trước đó trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh trước khi gỡ bỏ khỏi CPB đơn vị giải mã hiện thời trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh. Giá trị này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu của đơn vị giải mã vào CPB đối với HSS, như được xác định trong Phụ lục C. Phần tử cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định là cpb_removal_delay_length_minus1 + 1. common_du_cpb_removal_delay là phần dư của bộ đếm môđun $2^{(cpb_removal_delay_length_minus1 + 1)}$.

Cách thay thế để xác định 'common_du_cpb_removal_delay' như sau.

common_du_cpb_removal_delay xác định phải chờ bao nhiêu nhịp đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) để chờ sau khi gỡ bỏ khỏi CPB đơn vị giải mã sau cùng trước khi gỡ bỏ khỏi CPB đơn vị giải mã hiện thời trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh. Giá trị này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu của đơn vị giải mã vào CPB đối với HSS, như được xác định trong Phụ lục C. Phần tử cú pháp là mã có độ dài cố định mà độ dài của

nó theo bit được xác định là $\text{cpb_removal_delay_length_minus1} + 1$. $\text{common_du_cpb_removal_delay}$ là phần dư của bộ đếm $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ môđun.

Giá trị $\text{cpb_removal_delay_length_minus1}$ xác định độ dài (bằng bit) của phần tử cú pháp $\text{common_du_cpb_removal_delay}$ là giá trị $\text{cpb_removal_delay_length_minus1}$ đã mã hóa trong tập tham số chuỗi hoạt hoá đối với hình ảnh mã hoá được kết hợp với thông báo SEI định thời hình ảnh, mặc dù $\text{common_du_cpb_removal_delay}$ xác định nhịp đồng hồ của các hình ảnh phụ liên quan đến thời gian gỡ bỏ của đơn vị giải mã thứ nhất trong đơn vị truy cập đứng trước có chứa thông báo SEI về khoảng đệm, có thể là đơn vị truy cập của chuỗi video mã hóa khác.

' $\text{num_nalus_in_du_minus1}[i]$ ' + 1 xác định số lượng đơn vị NAL trong đơn vị giải mã thứ i của đơn vị truy cập mà thông báo SEI định thời hình ảnh liên quan. Giá trị $\text{num_nalus_in_du_minus1}[i]$ có thể nằm trong khoảng từ 0 đến $\text{PicWidthInCtbs} * \text{PicHeightInCtbs} - 1$, bao gồm cả giá trị đầu cuối.

Đơn vị giải mã thứ nhất của đơn vị truy cập gồm $\text{num_nalus_in_du_minus1}[0] + 1$ đơn vị NAL liên tiếp thứ nhất theo thứ tự giải mã trong đơn vị truy cập. Đơn vị giải mã thứ i (với i lớn hơn 0) của đơn vị truy cập gồm $\text{num_nalus_in_du_minus1}[i] + 1$ đơn vị NAL liên tiếp đầu tiên ngay sau đơn vị NAL cuối cùng trong đơn vị giải mã trước đó của đơn vị truy cập, theo thứ tự giải mã. Có thể có ít nhất một đơn vị NAL VCL trong mỗi đơn vị giải mã. Tất cả các đơn vị NAL không VCL được kết hợp với đơn vị NAL VCL có thể có trong cùng một đơn vị giải mã.

' $\text{du_cpb_removal_delay}[i]$ ' xác định các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) phải chờ bao nhiêu nhịp đồng hồ sau khi gỡ bỏ khỏi CPB đơn vị giải mã thứ nhất trong đơn vị truy cập được kết hợp với thông báo SEI về khoảng đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi CPB đơn vị giải mã thứ i trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh. Giá trị này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu của đơn vị giải mã vào CPB đối với HSS (ví dụ, như được xác định trong Phụ lục C của JCTVC-I1003). Phần tử cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định là $\text{cpb_removal_delay_length_minus1} + 1$. $\text{du_cpb_removal_delay}[i]$ là phần dư của bộ đếm môđun $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$.

Giá trị `cpb_removal_delay_length_minus1` xác định độ dài (bằng bit) của phần tử cú pháp `du_cpb_removal_delay[i]` là giá trị `cpb_removal_delay_length_minus1` đã mã hoá trong tập tham số chuỗi hoạt hoá đối với hình ảnh mã hoá được kết hợp với thông báo SEI định thời hình ảnh, mặc dù `du_cpb_removal_delay[i]` xác định nhịp đồng hồ của các hình ảnh phụ liên quan đến thời gian gỡ bỏ của đơn vị giải mã thứ nhất trong đơn vị truy cập đứng trước có chứa thông báo SEI về khoảng đệm, có thể là đơn vị truy cập của chuỗi video mã hóa khác.

Theo một cấu hình, bước định thời gỡ bỏ đơn vị giải mã và bước giải mã của các bộ giải mã có thể được thực hiện như sau.

Nếu `SubPicCpbFlag` bằng 0, biến `CpbRemovalDelay(m)` được thiết đặt bằng giá trị `cpb_removal_delay` trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập nghĩa là bộ giải mã m và biến T_c được thiết đặt bằng t_c . Theo cách khác, nếu `SubPicCpbFlag` bằng 1 và `common_du_cpb_removal_delay_flag` bằng 0 thì biến `CpbRemovalDelay(m)` được thiết đặt bằng giá trị `du_cpb_removal_delay[i]` cho đơn vị giải mã m (với m nằm trong khoảng từ 0 đến `num_decoding_units_minus1`) trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập có chứa bộ giải mã m và biến T_c được thiết đặt bằng t_{c_sub} .

Trong một số trường hợp, theo cách khác, nếu `SubPicCpbFlag` bằng 1 và `common_du_cpb_removal_delay_flag` bằng 0 thì biến `CpbRemovalDelay(m)` được thiết đặt bằng giá trị $(m+1)*du_cpb_removal_delay[i]$ cho bộ giải mã m (với m nằm trong khoảng từ 0 đến `num_decoding_units_minus1`) trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập có chứa bộ giải mã m và biến T_c được thiết đặt bằng t_{c_sub} .

Theo cách khác, nếu `SubPicCpbFlag` bằng 1 và `common_du_cpb_removal_delay_flag` bằng 1 thì biến `CpbRemovalDelay(m)` được thiết đặt bằng giá trị `common_du_cpb_removal_delay` cho đơn vị giải mã m trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập có chứa bộ giải mã m và biến T_c được thiết đặt bằng t_{c_sub} .

Khi đơn vị giải mã m là đơn vị giải mã với n bằng 0 (đơn vị giải mã thứ nhất của đơn vị truy cập khởi động HRD), thời gian gỡ bỏ danh định của đơn vị giải mã từ CPB được xác định là $t_{r,n}(0) = \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] / 90000$.

Khi đơn vị giải mã m là đơn vị giải mã thứ nhất của đơn vị truy cập thứ nhất của khoảng đệm không khởi động HRD, thời gian gỡ bỏ danh định của đơn vị giải mã từ CPB được xác định là $t_{r,n}(m) = t_{r,n}(m_b) + T_c * CPBRemovalDelay(m)$, trong đó $t_{r,n}(m_b)$ là thời gian gỡ bỏ danh định của đơn vị giải mã thứ nhất của khoảng đệm trước đó.

Khi đơn vị giải mã m là đơn vị giải mã thứ nhất của khoảng đệm, m_b được thiết đặt bằng m tại thời điểm gỡ bỏ $t_{r,n}(m)$ của đơn vị giải mã m . Thời gian gỡ bỏ danh định $t_{r,n}(m)$ của đơn vị giải mã m không phải là đơn vị giải mã thứ nhất của khoảng đệm được xác định là $t_{r,n}(m) = t_{r,n}(m_b) + T_c * CPBRemovalDelay(m)$, trong đó $t_{r,n}(m_b)$ là thời gian gỡ bỏ danh định của đơn vị giải mã thứ nhất của khoảng đệm hiện thời.

Thời gian gỡ bỏ đối với đơn vị giải mã m được xác định như sau. Nếu $low_delay_hrd_flag$ bằng 0 hoặc $t_{r,n}(m) \geq t_{af}(m)$, thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m)$. Theo cách khác, ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m) + T_c * Ceil((t_{af}(m) - t_{r,n}(m)) / T_c)$. Trường hợp sau ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$) chỉ báo rằng kích thước của đơn vị giải mã m , $b(m)$, lớn tới mức ngăn không cho gỡ bỏ tại thời điểm gỡ bỏ danh định.

Trong trường hợp khác, thời gian gỡ bỏ đối với đơn vị giải mã m được xác định như sau. If $low_delay_hrd_flag$ bằng 0 hoặc $t_{r,n}(m) \geq t_{af}(m)$, thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m)$. Theo cách khác, ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thời gian gỡ bỏ đối với đơn vị giải mã m không phải là đơn vị giải mã sau cùng trong đơn vị truy cập được xác định là $t_r(m) = t_{af}(m)$ và thời gian gỡ bỏ đối với đơn vị giải mã m là đơn vị giải mã sau cùng trong đơn vị truy cập $t_r(m) = t_{r,n}(m) + T_c * Ceil((t_{af}(m) - t_{r,n}(m)) / T_c)$. Trường hợp sau ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$) chỉ báo rằng kích thước của đơn vị giải mã m , $b(m)$, lớn tới mức ngăn không cho gỡ bỏ tại thời điểm gỡ bỏ danh định.

Trong trường hợp khác, thời gian gỡ bỏ đối với đơn vị giải mã m được xác định như sau. Nếu $low_delay_hrd_flag$ bằng 0 hoặc $t_{r,n}(m) \geq t_{af}(m)$, thì thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m)$. Theo cách khác, nếu ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian gỡ bỏ đối với đơn vị giải mã m không phải là đơn vị giải mã sau cùng trong đơn vị truy cập được xác định là $t_r(m) = t_{af}(m)$ và thời gian gỡ bỏ đối với đơn vị giải mã m là đơn vị giải mã sau cùng trong đơn vị truy cập $t_r(m) = t_{r,n}(m) + T_c * Ceil((t_{af}(m) - t_{r,n}(m)) / T_c)$. Trường hợp sau ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$)

chỉ báo rằng kích thước của đơn vị giải mã m , $b(m)$, lớn tới mức ngăn không cho gỡ bỏ tại thời điểm gỡ bỏ danh định.

Trong trường hợp khác, thời gian gỡ bỏ đối với đơn vị giải mã m được xác định như sau. Nếu $\text{low_delay_hrd_flag}$ bằng 0 hoặc $t_{r,n}(m) \geq t_{af}(m)$, thì thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m)$. Theo cách khác, ($\text{low_delay_hrd_flag}$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{af}(m)$. Trường hợp sau ($\text{low_delay_hrd_flag}$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$) chỉ báo rằng kích thước của đơn vị giải mã m , $b(m)$, lớn tới mức ngăn không cho gỡ bỏ tại thời điểm gỡ bỏ danh định.

Khi SubPicCpbFlag bằng 1, thì thời gian gỡ bỏ khỏi CPB danh định của đơn vị truy cập n $t_{r,n}(n)$ được thiết đặt bằng thời gian gỡ bỏ khỏi CPB danh định của đơn vị giải mã sau cùng trong đơn vị truy cập n , thời gian gỡ bỏ khỏi CPB của đơn vị truy cập n $t_r(n)$ được thiết đặt bằng thời gian gỡ bỏ khỏi CPB của đơn vị giải mã sau cùng trong đơn vị truy cập n .

Khi SubPicCpbFlag bằng 0, thì mỗi đơn vị giải mã là đơn vị truy cập, do vậy thời gian gỡ bỏ khỏi CPB danh định và thời gian gỡ bỏ khỏi CPB của đơn vị truy cập n là thời gian gỡ bỏ khỏi CPB danh định và thời gian gỡ bỏ khỏi CPB của đơn vị giải mã n .

Tại thời điểm gỡ bỏ khỏi CPB của đơn vị giải mã m , đơn vị giải mã sẽ được giải mã tức thì.

Ví dụ khác về cú pháp và ngữ nghĩa cải biến đối với thông báo SEI định thời hình ảnh theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện trong Bảng (2) dưới đây. Các cải biến theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện bằng chữ in đậm.

Bảng 2

pic_timing(payloadSize) {
if(CpbDpbDelaysPresentFlag) {
cpb_removal_delay
dpb_output_delay
if(sub_pic_cpb_params_present_flag) {
num_decoding_units_minus1
common_du_cpb_removal_delay_flag
if(common_du_cpb_removal_delay_flag) {
common_num_nalus_in_du_minus1
common_du_cpb_removal_delay
}
for(i = 0; i <= num_decoding_units_minus1; i++) {
num_nalus_in_du_minus1[i]
if(!common_du_cpb_removal_delay_flag)
du_cpb_removal_delay[i]
}
}
}
}

Bảng (2)

Ví dụ minh họa trong bảng (2) bao gồm phần tử cú pháp `common_num_nalus_in_du_minus1`, thành phần này nó có thể được sử dụng để xác định bao nhiêu dữ liệu cần được gỡ bỏ khỏi CPB khi gỡ bỏ đơn vị giải mã. '`common_num_nalus_in_du_minus1`' + 1 xác định số lượng đơn vị NAL trong mỗi đơn vị giải mã của đơn vị truy cập mà thông báo SEI định thời hình ảnh liên quan. Giá trị `common_num_nalus_in_du_minus1` có thể nằm trong khoảng từ 0 đến $\text{PicWidthInCtbs} * \text{PicHeightInCtbs} - 1$, bao gồm cả giá trị đầu cuối.

Đơn vị giải mã thứ nhất của đơn vị truy cập gồm `common_num_nalus_in_du_minus1` + 1 bộ NAL liên tiếp thứ nhất theo thứ tự giải mã trong đơn vị truy cập. Đơn vị giải mã thứ i (với i lớn hơn 0) của đơn vị truy cập gồm `common_num_nalus_in_du_minus1` + 1 bộ NAL liên tiếp thứ nhất ngay sau đơn vị NAL cuối cùng trong đơn vị giải mã trước đó của đơn vị truy cập, theo thứ tự giải mã. Có thể có ít nhất một đơn vị NAL VCL trong mỗi đơn vị giải mã. Tất cả các đơn vị NAL không VCL được kết hợp với đơn vị NAL VCL có thể có trong cùng một đơn vị giải mã.

Ví dụ khác về cú pháp và ngữ nghĩa cải biến đối với thông báo SEI định thời hình ảnh theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện trong Bảng (3) dưới đây. Các cải biến theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được thể hiện bằng nét in đậm.

Bảng 3

<code>pic_timing(payloadSize) {</code>
<code>if (CpbDpbDelaysPresentFlag) {</code>
<code> cpb_removal_delay</code>
<code> dpb_output_delay</code>
<code> if (sub_pic_cpb_params_present_flag) {</code>
<code> num_decoding_units_minus1</code>
<code> common_num_nalus_in_du_flag</code>
<code> if (common_num_nalus_in_du_flag) {</code>
<code> common_num_nalus_in_du_minus1</code>
<code> }</code>
<code> common_du_cpb_removal_delay_flag</code>
<code> if (common_du_cpb_removal_delay_flag) {</code>
<code> common_du_cpb_removal_delay</code>
<code> }</code>
<code> for (i = 0; i <= num_decoding_units_minus1; i++) {</code>
<code> if (common_num_nalus_in_du_flag)</code>
<code> num_nalus_in_du_minus1[i]</code>
<code> if (common_du_cpb_removal_delay_flag)</code>
<code> du_cpb_removal_delay[i]</code>
<code> }</code>
<code> }</code>
<code>}</code>

Bảng (3)

Ví dụ minh họa trong bảng (3) bao gồm phần tử cú pháp 'common_num_nalus_in_du_flag' mà, khi bằng 1, xác định rằng phần tử cú pháp 'common_num_nalus_in_du_minus1' có mặt. 'common_num_nalus_in_du_flag' bằng 0 xác định rằng phần tử cú pháp 'common_num_nalus_in_du_minus1' không có mặt.

Theo phương án khác nữa, các cờ common_du_cpb_removal_delay_flag common_num_nalus_in_du_minus1, có thể không được gửi. Thay vào đó các phần tử cú pháp common_num_nalus_in_du_minus1 và common_du_cpb_removal_delay có thể được gửi mọi lần. Trong trường hợp này giá trị 0 (hoặc giá trị khác nào đó) đối với các phần tử cú pháp này có thể được sử dụng để chỉ báo rằng các thành phần này không được truyền tín hiệu.

Ngoài các cải biến đối với các phần tử cú pháp và ngữ nghĩa của thông báo SEI định thời hình ảnh, các hệ thống và các phương pháp hiện thời cũng có thể thực hiện việc ràng buộc dòng bit sao cho hoạt động khỏi CPB dựa trên hình ảnh phụ và hoạt động khỏi CPB ở mức của đơn vị truy cập dẫn đến sự định thời giống nhau về việc gỡ bỏ đơn vị giải mã.

Khi sub_pic_cpb_params_present_flag là bằng 1 theo đó các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ có mặt thì CPB có thể hoạt động ở mức của đơn vị truy cập hoặc mức hình ảnh phụ, sub_pic_cpb_params_present_flag bằng 0 thì xác định rằng các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ không có mặt và CPB hoạt động ở mức của đơn vị truy cập. Khi sub_pic_cpb_params_present_flag không có mặt, thì giá trị của được suy ra bằng 0.

Để hỗ trợ cho hoạt động tại cả mức đơn vị truy cập hoặc mức hình ảnh phụ, các ràng buộc của dòng bit sau đây có thể được sử dụng: Nếu sub_pic_cpb_params_present_flag bằng 1 thì cần phải có tính phù hợp của dòng bit mà các ràng buộc sau đây phải được tuân theo khi truyền tín hiệu các giá trị cho cpb_removal_delay và du_cpb_removal_delay[i] cho mọi i:

Biểu thức 3

$$\text{cpb_removal_delay} = \frac{\left(\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i] \right) * t_{c,\text{sub}}}{t_c}$$

trong đó du_cpb_removal_delay[i] là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, thì t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi

một và i là chỉ số. Theo một số phương án, tham số dung sai có thể được bổ sung để thỏa mãn ràng buộc nêu trên.

Để hỗ trợ cho hoạt động tại cả mức đơn vị truy cập hoặc mức hình ảnh phụ, các ràng buộc của dòng bit dưới đây có thể được sử dụng: biến $T_{du}(k)$ được xác định như sau:

Biểu thức 4

$$T_{du}(k) = T_{du}(k-1) + t_{c,sub} * \sum_{i=0}^{num_decoding_units_minus1} (du_cpb_removal_delay_minus1[i] + 1)$$

trong đó $du_cpb_removal_delay_minus1[i]$ và $num_decoding_units_minus1_k$ là các tham số cho đơn vị giải mã thứ I của đơn vị truy cập thứ k (với $k=0$ cho đơn vị truy cập đã khởi động HRD và $T_{du}(k)=0$ cho $k<1$) và trong đó $du_cpb_removal_delay_minus1_k[i]+1=du_cpb_removal_delay_minus1_k[i]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã cho đơn vị giải mã thứ I của đơn vị truy cập thứ k và $num_decoding_units_minus1_k$ là số lượng đơn vị giải mã trong đơn vị truy cập thứ k , t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ và i và k là các chỉ số. Sau đó khi cờ định thời hình ảnh (ví dụ, `sub_pic_cpb_params_present_flag`) được thiết đặt bằng 1, thì ràng buộc sau phải là đúng ($du_cpb_removal_delay_minus1 + 1) * t_c == T_{du}(k)$, trong đó $(du_cpb_removal_delay_minus1 + 1)=cpb_removal_delay$, khoảng trễ gỡ bỏ CPB. Do đó trong trường hợp này, khoảng trễ gỡ bỏ CPB $(du_cpb_removal_delay_minus1 + 1)$ được đặt sao cho thao tác của thao tác CPB dựa trên hình ảnh phụ và thao tác CPB dựa trên đơn vị truy cập dẫn đến sự định thời giống nhau của việc gỡ bỏ đơn vị truy cập và đơn vị giải mã cuối cùng của việc gỡ bỏ đơn vị truy cập.

Để hỗ trợ cho hoạt động tại cả mức đơn vị truy cập hoặc mức hình ảnh phụ, các ràng buộc của dòng bit sau đây có thể được sử dụng: Nếu `sub_pic_cpb_params_present_flag` bằng 1 thì cần phải có tính phù hợp của dòng bit theo đó các ràng buộc sau đây phải được tuân theo khi truyền tín hiệu các giá trị cho `cpb_removal_delay` và `du_cpb_removal_delay[i]` cho mọi i :

Biểu thức 5

$$-1 \leq [cpb_removal_delay * t_c - (\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] * t_{c,sub})] \leq 1$$

trong đó $du_cpb_removal_delay[i]$ là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một và i là chỉ số.

Để hỗ trợ cho hoạt động tại cả mức đơn vị truy cập hoặc mức hình ảnh phụ, các ràng buộc của dòng bit sau đây có thể được sử dụng: Nếu `sub_pic_cpb_params_present_flag` bằng 1 thì cần phải có tính phù hợp của dòng bit theo đó các ràng buộc sau đây phải được tuân theo khi truyền tín hiệu các giá trị cho `cpb_removal_delay` và `du_cpb_removal_delay[num_decoding_units_minus1]`:
$$\text{cpb_removal_delay} * t_c = \text{du_cpb_removal_delay[num_decoding_units_minus1]} * t_{c,\text{sub}}$$
 trong đó `du_cpb_removal_delay[num_decoding_units_minus1]` là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ `num_decoding_units_minus1`, t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, `num_decoding_units_minus1` là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một. Theo một số phương án, tham số dung sai có thể được bổ sung để thỏa mãn ràng buộc nêu trên.

Để hỗ trợ cho hoạt động tại cả mức đơn vị truy cập hoặc mức hình ảnh phụ, các ràng buộc của dòng bit sau đây có thể được sử dụng: Nếu `sub_pic_cpb_params_present_flag` bằng 1 thì cần phải có tính phù hợp của dòng bit theo đó các ràng buộc sau đây phải được tuân theo khi truyền tín hiệu các giá trị cho `cpb_removal_delay` và `du_cpb_removal_delay[i]` cho mọi i :
$$-1 \leq (\text{cpb_removal_delay} * t_c - \text{du_cpb_removal_delay[num_decoding_units_minus1]} * t_{c,\text{sub}}) \leq 1$$
 trong đó `du_cpb_removal_delay[num_decoding_units_minus1]` là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ `num_decoding_units_minus1`, t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, `num_decoding_units_minus1` là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một.

Ngoài ra, các hệ thống và các phương pháp hiện thời có thể cải biến định thời về việc gỡ bỏ đơn vị giải mã. Khi các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ có mặt, thì thời gian gỡ bỏ đối với đơn vị giải mã đối với “các hình ảnh lớn” (khi `low_delay_hrd_flag` bằng 1 và $t_{r,n}(m) < t_{af}(m)$) có thể được thay đổi để bù trừ cho chênh lệch có thể xuất hiện do bộ đếm nhịp đồng hồ và bộ đếm nhịp đồng hồ hình ảnh phụ.

Khi `sub_pic_cpb_params_present_flag` là bằng 1, thì các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ có mặt và CPB có thể hoạt động ở mức của đơn vị truy cập hoặc mức hình ảnh phụ, `sub_pic_cpb_params_present_flag` bằng 0 xác định rằng các tham số trễ gỡ bỏ khỏi CPB ở mức hình ảnh phụ không có mặt và CPB hoạt động ở mức của đơn vị truy cập. Khi `sub_pic_cpb_params_present_flag` không có mặt, thì giá trị của được suy ra bằng 0.

Cụ thể là, một ví dụ về định thời việc gỡ bỏ đơn vị giải mã và bước giải mã đối với phương án thực hiện của của bộ giải mã như sau. Biến SubPicCpbPreferredFlag hoặc được xác định là các công cụ bên ngoài, hoặc khi không được xác định là các công cụ bên ngoài, thì được thiết đặt bằng 0. Biến SubPicCpbFlag được suy ra như sau: $\text{SubPicCpbFlag} = \text{SubPicCpbPreferredFlag} \ \&\& \ \text{sub_pic_cpb_params_present_flag}$. Nếu SubPicCpbFlag bằng 0, thì CPB hoạt động ở mức của đơn vị truy cập và mỗi đơn vị giải mã là đơn vị truy cập. Theo cách khác, CPB hoạt động ở mức hình ảnh phụ và mỗi đơn vị giải mã là tập hợp phụ của đơn vị truy cập.

Nếu SubPicCpbFlag bằng 0, biến CpbRemovalDelay(m) được thiết đặt bằng giá trị cpb_removal_delay trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập là đơn vị giải mã m và biến T_c được thiết đặt bằng t_c . Theo cách khác, biến CpbRemovalDelay(m) được thiết đặt bằng giá trị $\text{du_cpb_removal_delay}[i]$ cho đơn vị giải mã m trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập có chứa đơn vị giải mã m và biến T_c được thiết đặt bằng t_{c_sub} .

Khi đơn vị giải mã m là đơn vị giải mã với n bằng 0 (đơn vị giải mã thứ nhất của đơn vị truy cập khởi động HRD), thì thời gian gỡ bỏ danh định của đơn vị giải mã từ CPB được xác định là $t_{r,n}(0) = \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] / 90000$.

Khi đơn vị giải mã m là đơn vị giải mã thứ nhất của đơn vị truy cập thứ nhất của khoảng đệm không khởi động HRD, thì thời gian gỡ bỏ danh định của đơn vị giải mã từ CPB được xác định là $t_{r,n}(m) = t_{r,n}(m_b) + T_c * \text{CPBRemovalDelay}(m)$, trong đó $t_{r,n}(m_b)$ là thời gian gỡ bỏ danh định của đơn vị giải mã thứ nhất của khoảng đệm trước đó.

Khi đơn vị giải mã m là đơn vị giải mã thứ nhất của khoảng đệm, thì m_b được thiết đặt bằng m tại thời điểm gỡ bỏ $t_{r,n}(m)$ của đơn vị giải mã m.

Thời gian gỡ bỏ danh định $t_{r,n}(m)$ của đơn vị giải mã m không phải là đơn vị giải mã thứ nhất của khoảng đệm được xác định là $t_{r,n}(m) = t_{r,n}(m_b) + T_c * \text{CPBRemovalDelay}(m)$, trong đó $t_{r,n}(m_b)$ là thời gian gỡ bỏ danh định của đơn vị giải mã thứ nhất của khoảng đệm hiện thời.

Thời gian gỡ bỏ đối với đơn vị giải mã m được xác định như sau. Biến ClockDiff được xác định là $\text{ClockDiff} = (\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1))) / \text{time_scale}$. Trong một số trường hợp có thể cần có tính phù hợp của dòng bit nghĩa là các tham số num_units_in_tick , $\text{num_units_in_sub_tick}$,

num_decoding_units_minus1 được truyền tín hiệu sao cho thỏa mãn được biểu thức sau đây. $(\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1))) \geq 0$

Trong một số trường hợp khác, có thể cần có tính phù hợp của dòng bit mà các tham số num_units_in_tick, num_units_in_sub_tick, num_decoding_units_minus1 có thể được truyền tín hiệu sao cho thỏa mãn được biểu thức sau đây. $(\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1))) \leq 0$. Nếu low_delay_hrd_flag bằng 0 hoặc $t_{r,n}(m) \geq t_{af}(m)$, thì thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m)$.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và khi sub_pic_cpb_params_present_flag là bằng 1 và CPB hoạt động ở mức hình ảnh phụ và nếu ClockDiff lớn hơn 0, thì thời gian gỡ bỏ đối với đơn vị giải mã m khi đó là đơn vị giải mã sau cùng của đơn vị truy cập n được xác định là $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / T_c) + \text{ClockDiff}$.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và khi sub_pic_cpb_params_present_flag là bằng 1 và CPB hoạt động ở mức của đơn vị truy cập và nếu ClockDiff nhỏ hơn 0, thì thời gian gỡ bỏ của đơn vị truy cập n được xác định là $t_r(m) = t_{r,n}(m) + t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_c) - \text{ClockDiff}$.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian gỡ bỏ đối với đơn vị giải mã m được xác định là $t_r(m) = t_{r,n}(m) + T_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / T_c)$. Trường hợp (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) sau chỉ báo rằng kích thước của đơn vị giải mã m, b(m), lớn tới mức ngăn không cho gỡ bỏ tại thời điểm gỡ bỏ danh định.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và khi cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và khi cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n))/t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Theo cách khác, (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thời gian gỡ bỏ đối với đơn vị giải mã không phải là đơn vị giải mã sau cùng của đơn vị truy cập được thiết đặt bằng $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m. và thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ được thiết đặt theo: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m))/t_{c_sub}))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến

cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m trong đơn vị truy cập n .

Theo cách khác, ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thời gian gỡ bỏ đối với đơn vị giải mã không phải là đơn vị giải mã sau cùng của đơn vị truy cập được đặt là $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m , và thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ được thiết đặt theo: $t_r(m) = t_{r,n}(m) + (t_c * Ceil((t_{af}(m) - t_{r,n}(m)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m trong đơn vị truy cập n .

Theo cách khác, ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thời gian gỡ bỏ đối với đơn vị giải mã được thiết đặt bằng $t_r(m) = t_{af}(m)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m trong đơn vị truy cập n .

Theo cách khác, ($low_delay_hrd_flag$ bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{af}(n)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi $SubPicCpbFlag$ bằng 1, thì thời gian gỡ bỏ khỏi CPB danh định của đơn vị truy cập n $t_{r,n}(n)$ được thiết đặt bằng thời gian gỡ bỏ khỏi CPB danh định của đơn vị giải mã sau cùng trong đơn vị truy cập n , thời gian gỡ bỏ khỏi CPB của đơn vị truy cập n $t_r(n)$ được thiết đặt bằng thời gian gỡ bỏ khỏi CPB của đơn vị giải mã sau cùng trong đơn vị truy cập n .

Khi SubPicCpbFlag bằng 0, thì mỗi đơn vị giải mã là đơn vị truy cập, do vậy thời gian gỡ bỏ khỏi CPB danh định và thời gian gỡ bỏ khỏi CPB của đơn vị truy cập n là thời gian gỡ bỏ khỏi CPB danh định và thời gian gỡ bỏ khỏi CPB của đơn vị giải mã n . Tại thời điểm gỡ bỏ khỏi CPB của đơn vị giải mã m , đơn vị giải mã sẽ được giải mã tức thì.

Như được minh họa ở phần trên, các hệ thống và các phương pháp được bộc lộ trong sáng chế tạo ra cú pháp và ngữ nghĩa để cải biến các dòng bit thông báo SEI định thời hình ảnh mang các tham số dựa trên hình ảnh phụ. Trong một số cấu hình, các hệ thống và các phương pháp được bộc lộ trong sáng chế có thể được áp dụng cho các tiêu chuẩn HEVC.

Để tiện lợi, một số định nghĩa được thể hiện dưới đây, có thể được áp dụng cho các hệ thống và các phương pháp được bộc lộ trong sáng chế. Điểm truy nhập ngẫu nhiên có thể là điểm bất kỳ trong luồng dữ liệu (ví dụ, dòng bit), trong đó việc giải mã dòng bit không yêu cầu phải truy nhập bất kỳ điểm nào trong dòng bit đứng trước điểm truy nhập ngẫu nhiên để giải mã ảnh hiện thời và tất cả các hình ảnh tiếp theo hình ảnh hiện thời theo thứ tự kết xuất.

Khoảng đệm có thể được xác định dưới dạng tập hợp của các đơn vị truy cập giữa hai thời điểm của thông báo SEI về khoảng đệm theo thứ tự giải mã. Thông tin nâng cao bổ sung (SEI) có thể chứa thông tin không cần thiết để giải mã các mẫu của các hình ảnh mã hoá từ các đơn vị NAL VCL. Các thông báo SEI có thể giúp trong quy trình liên quan đến việc giải mã, hiển thị hoặc các mục đích khác. Việc làm phù hợp các bộ giải mã có thể không được yêu cầu để xử lý thông tin này đối với tính phù hợp thứ tự kết xuất thành chuẩn HEVC (ví dụ, Phụ lục C của chuẩn HEVC (JCTVC-I1003) bao gồm chuẩn cho tính phù hợp). Một số thông tin thông báo SEI có thể được sử dụng để kiểm tra tính phù hợp của dòng bit và đối với tính phù hợp của bộ giải mã định thời kết xuất.

Thông báo SEI về khoảng đệm có thể là thông báo SEI liên quan đến khoảng đệm. Thông báo SEI định thời hình ảnh có thể là thông báo SEI liên quan đến thời điểm gỡ bỏ khỏi CPB. Các thông báo này có thể xác định cú pháp và ngữ nghĩa mà chúng xác định định thời đến của dòng bit và định thời gỡ bỏ hình ảnh mã hoá.

Bộ đệm hình ảnh mã hoá (CPB) có thể là bộ đệm vào trước ra trước chứa các đơn vị truy cập theo thứ tự giải mã được xác định trong bộ giải mã tham chiếu giả định (HRD). Đơn vị truy cập có thể là tập hợp các bộ lớp truy nhập mạng (NAL), các bộ này là liên tiếp theo thứ tự giải mã và chứa chính xác một hình ảnh mã hoá. Ngoài đơn vị NAL của các phiên mã hoá của hình ảnh mã hoá, đơn vị truy cập cũng có thể chứa các đơn vị NAL khác

không chứa các phiên của hình ảnh mã hoá. Sự giải mã của đơn vị truy cập luôn dẫn đến ảnh giải mã. Đơn vị NAL có thể là cấu trúc cú pháp có chứa dấu hiệu của loại dữ liệu tiếp sau đó và các byte có chứa dữ liệu đó dưới dạng chuỗi byte thô nằm rải rác cần thiết với các byte ngăn ngừa mô phỏng.

Như được sử dụng trong bản mô tả này, thuật ngữ “chung” nói chung để chỉ phần tử cú pháp hoặc biến có thể áp dụng cho những thứ nhiều hơn một. Ví dụ, trong ngữ cảnh các phần tử cú pháp trong thông báo SEI định thời hình ảnh, thuật ngữ “chung” có thể có nghĩa rằng phần tử cú pháp đó (ví dụ, `common_du_cpb_removal_delay`) có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập được kết hợp với thông báo SEI định thời hình ảnh. Ngoài ra, các đơn vị dữ liệu được mô tả theo “n” và “m” nói chung để chỉ lần lượt các đơn vị truy cập và các đơn vị giải mã.

Một số cấu hình sẽ được mô tả cùng với các hình vẽ, trong đó các số chỉ dẫn giống nhau có thể để chỉ các thành phần tương tự về mặt chức năng. Các hệ thống và các phương pháp như được mô tả chung và minh họa trên các hình vẽ trong bản mô tả này có thể được trang bị và thiết kế theo nhiều cấu hình khác nhau. Do vậy, mô tả chi tiết sâu hơn sau đây về một số cấu hình, như được thể hiện trên các hình vẽ, không phải là giới hạn phạm vi, như được yêu cầu bảo hộ, mà chỉ mang tính đại diện đối với các hệ thống và các phương pháp.

Fig.1A là sơ đồ khối thể hiện ví dụ về một hoặc nhiều thiết bị điện tử 102, trong đó các hệ thống và các phương pháp gửi thông báo và đệm dòng bit có thể được thực hiện. Trong ví dụ này, thiết bị điện tử A 102a và thiết bị điện tử B 102b được minh họa. Tuy nhiên, cần lưu ý rằng, một hoặc nhiều dấu hiệu và chức năng được mô tả liên quan đến thiết bị điện tử A 102a và thiết bị điện tử B 102b có thể được kết hợp thành một thiết bị điện tử trong một số cấu hình.

Thiết bị điện tử A 102a bao gồm bộ mã hóa 104. Bộ mã hóa 104 bao gồm môđun tạo thông báo 108. Mỗi thành phần này được bao gồm bên trong thiết bị điện tử A 102a (ví dụ, bộ mã hóa 104 và môđun tạo thông báo 108) có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai.

Thiết bị điện tử A 102a có thể nhận một hoặc nhiều hình ảnh đầu vào 106. Trong một số cấu hình, (các) hình ảnh đầu vào 106 có thể được chụp trên thiết bị điện tử A 102a sử dụng cảm biến ảnh, có thể được truy hồi từ bộ nhớ và/hoặc có thể được thu từ thiết bị điện tử khác.

Bộ mã hóa 104 có thể mã hóa (các) hình ảnh đầu vào 106 để tạo ra dữ liệu mã hoá. Ví dụ, bộ mã hóa 104 có thể mã hóa một loạt hình ảnh đầu vào 106 (ví dụ, video). Theo một cấu hình, bộ mã hóa 104 có thể là bộ mã hóa HEVC. Dữ liệu mã hoá này có thể là dữ liệu số (ví dụ, một phần dòng bit 114). Bộ mã hóa 104 có thể tạo ra việc truyền tín hiệu bổ sung dựa trên tín hiệu vào.

Môđun tạo thông báo 108 có thể tạo ra một hoặc nhiều thông báo. Ví dụ, môđun tạo thông báo 108 có thể tạo ra một hoặc nhiều thông báo SEI hoặc các thông báo khác. Đối với CPB hỗ trợ hoạt động ở mức hình ảnh phụ, thiết bị điện tử 102 có thể gửi các tham số hình ảnh phụ, (ví dụ, tham số trễ gỡ bỏ khỏi CPB). Cụ thể là, thiết bị điện tử 102 (ví dụ, bộ mã hóa 104) có thể xác định liệu có bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung trong thông báo SEI định thời hình ảnh hay không. Ví dụ, thiết bị điện tử có thể thiết đặt cờ (ví dụ, `common_du_cpb_removal_delay_flag`) bằng 1 khi bộ mã hóa 104 bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) trong thông báo SEI định thời hình ảnh. Khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung được bao gồm, thì thiết bị điện tử có thể tạo ra tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung, mà có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập. Nói cách khác, thay vì bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã cho mỗi đơn vị giải mã trong đơn vị truy cập, tham số chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập mà cùng với nó thông báo SEI định thời hình ảnh liên quan.

Trái lại, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung không có trong thông báo SEI định thời hình ảnh, thì thiết bị điện tử 102 có thể tạo ra khoảng trễ riêng gỡ bỏ khỏi CPB của đơn vị giải mã cho mỗi đơn vị giải mã trong đơn vị truy cập mà cùng với nó thông báo SEI định thời hình ảnh liên quan. Môđun tạo thông báo 108 có thể thực hiện một hoặc nhiều quy trình được mô tả dưới đây có tham chiếu đến Fig.2 và Fig.3.

Trong một số cấu hình, thiết bị điện tử A 102a có thể gửi thông báo đến thiết bị điện tử B 102b là một phần của dòng bit 114. Trong một số cấu hình của thiết bị điện tử A 102a có thể gửi thông báo đến thiết bị điện tử B 102b bằng dòng truyền riêng biệt 110. Ví dụ, việc truyền riêng biệt này có thể không phải là một phần dòng bit 114. Chẳng hạn, thông báo SEI định thời hình ảnh hoặc thông báo khác có thể được gửi sử dụng thiết bị ngoài băng tần nào đó. Cần lưu ý rằng, trong một số cấu hình, thông báo khác có thể bao gồm một hoặc nhiều dấu hiệu của thông báo SEI định thời hình ảnh được mô tả trên đây. Ngoài ra, thông

báo khác, theo một hoặc nhiều khía cạnh, có thể được sử dụng tương tự như đối với thông báo SEI được mô tả trên đây.

Bộ mã hóa 104 (và, ví dụ, môđun tạo thông báo 108) có thể tạo ra dòng bit 114. Dòng bit 114 có thể bao gồm dữ liệu hình ảnh mã hoá dựa trên (các) hình ảnh đầu vào 106. Trong một số cấu hình, dòng bit 114 có thể bao gồm dữ liệu tiêu đề, như thông báo SEI định thời hình ảnh hoặc thông báo khác, (các) tiêu đề phiên, (các) PPS v.v.. Do các hình ảnh đầu vào 106 bổ sung được mã hoá, dòng bit 114 có thể bao gồm một hoặc nhiều hình ảnh mã hoá. Chẳng hạn, dòng bit 114 có thể bao gồm một hoặc nhiều hình ảnh mã hoá tương ứng với dữ liệu tiêu đề (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác).

Dòng bit 114 có thể được cung cấp cho bộ giải mã 112. Theo một ví dụ, dòng bit 114 có thể được truyền đến thiết bị điện tử B 102b sử dụng kết nối hữu tuyến hoặc vô tuyến. Trong một số trường hợp, việc này có thể được thực hiện qua mạng, như Internet hoặc mạng cục bộ (LAN). Như được thể hiện trên Fig.1A, bộ giải mã 112 có thể được thực hiện trên thiết bị điện tử B 102b riêng với bộ mã hóa 104 trên thiết bị điện tử A 102a. Tuy nhiên, cần lưu ý rằng bộ mã hóa 104 và bộ giải mã 112 có thể được thực hiện trên cùng thiết bị điện tử trong một số cấu hình. Theo cách thực hiện trong đó bộ mã hóa 104 và bộ giải mã 112 được thực hiện trên cùng thiết bị điện tử, chẳng hạn, dòng bit 114 có thể được cung cấp qua bus cho bộ giải mã 112 hoặc được lưu trữ trong bộ nhớ để truy hồi từ bộ giải mã 112.

Bộ giải mã 112 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Theo một cấu hình, bộ giải mã 112 có thể là bộ giải mã HEVC. Bộ giải mã 112 có thể thu (ví dụ, nhận được) dòng bit 114. Bộ giải mã 112 có thể tạo ra một hoặc nhiều hình ảnh giải mã 118 dựa trên dòng bit 114. (các) hình ảnh giải mã 118 có thể được hiển thị, được phát lại, được lưu trữ trong bộ nhớ và/hoặc được truyền đến thiết bị khác v.v..

Bộ giải mã 112 có thể bao gồm CPB 120. CPB 120 này có thể tạm thời lưu trữ các hình ảnh mã hoá. CPB 120 này có thể sử dụng các tham số được tìm thấy thông báo SEI định thời hình ảnh để xác định khi gỡ bỏ dữ liệu. Khi CPB 120 hỗ trợ hoạt động ở mức hình ảnh phụ, các đơn vị giải mã đơn lẻ có thể được gỡ bỏ ở thời điểm nào đó chứ không phải là toàn bộ các đơn vị truy cập. Bộ giải mã 112 có thể bao gồm bộ đệm hình ảnh giải mã (DPB) 122. Mỗi hình ảnh giải mã được đặt vào DPB 122 để được tham chiếu nhờ quy trình giải mã cũng như để kết xuất và cắt xén. Hình ảnh giải mã được gỡ bỏ khỏi DPB ở phần sau thời gian kết xuất DPB hoặc thời gian mà tại đó không còn cần dùng để tham chiếu dự báo liên kết.

Bộ giải mã 112 có thể thu thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác). Bộ giải mã 112 có thể xác định liệu thông báo thu được đó bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) hay không. Điều này có thể bao gồm việc xác định cờ (ví dụ, `common_du_cpb_removal_delay_flag`) theo đó nó được đặt khi tham số chung có mặt trong thông báo SEI định thời hình ảnh. Nếu tham số chung có mặt, bộ giải mã 112 có thể xác định tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập. Nếu tham số chung không có mặt, bộ giải mã 112 có thể xác định tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng cho mỗi đơn vị giải mã trong đơn vị truy cập. Bộ giải mã 112 cũng có thể gỡ bỏ các đơn vị giải mã khỏi CPB 120 sử dụng hoặc tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hoặc tham số trễ gỡ bỏ khỏi CPB các đơn vị giải mã riêng. CPB 120 có thể thực hiện một hoặc nhiều quy trình được mô tả dưới đây cùng với Fig.4 và Fig.5.

HRD được mô tả trên đây có thể là một ví dụ về bộ giải mã 112 được thể hiện trên Fig.1A. Do vậy, thiết bị điện tử 102 có thể hoạt động phù hợp với HRD, CPB 120 và DPB 122 được mô tả trên đây, trong một số cấu hình.

Cần lưu ý rằng một hoặc nhiều thành phần hoặc tiểu mục của nó có trong (các) thiết bị điện tử 102 có thể được thực hiện trong phần cứng. Ví dụ, một hoặc nhiều thành phần này hoặc tiểu mục của nó có thể được thực hiện như chip, mạch điện hoặc các bộ phận phần cứng v.v.. Cần lưu ý rằng một hoặc nhiều chức năng hoặc phương pháp được mô tả trong bản mô tả này có thể được thực hiện trong và/hoặc được thực hiện sử dụng phần cứng. Ví dụ, một hoặc nhiều phương pháp này được mô tả trong bản mô tả này có thể được thực hiện trong và/hoặc được thực hiện sử dụng bộ chip, mạch tích hợp chuyên dụng (ASIC - Application-Specific Integrated Circuit), mạch tích hợp cỡ lớn (LSI - Large-Scale Integrated) hoặc mạch tích hợp v.v..

Fig.1B là sơ đồ khối thể hiện ví dụ khác nữa về bộ mã hóa 1908 và bộ giải mã 1972. Trong ví dụ này, thiết bị điện tử A 1902 và thiết bị điện tử B 1970 được minh họa. Tuy nhiên, cần lưu ý rằng các dấu hiệu và chức năng được mô tả liên quan đến thiết bị điện tử A 1902 và thiết bị điện tử B 1970 này có thể được kết hợp thành một thiết bị điện tử trong một số cấu hình.

Thiết bị điện tử A 1902 bao gồm bộ mã hóa 1908. Bộ mã hóa 1908 có thể bao gồm bộ mã hóa lớp cơ sở 1910 và bộ mã hóa lớp nâng cao 1920. Bộ mã hóa video 1908 phù hợp

cho việc mã hóa video có thể mở rộng và mã hóa video đa cảnh, như sẽ được mô tả dưới đây. Bộ mã hóa 1908 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Theo một cấu hình, bộ mã hóa 1908 có thể là bộ mã hóa theo chuẩn mã hóa video hiệu suất cao (HEVC - high-efficiency video coding), bao gồm ảnh mở rộng được và/hoặc đa cảnh. Các bộ mã hóa khác cũng có thể được sử dụng. Thiết bị điện tử A 1902 có thể thu được nguồn 1906. Trong một số cấu hình, nguồn 1906 có thể được chụp trên thiết bị điện tử A 1902 sử dụng cảm biến ảnh, được truy hồi từ bộ nhớ hoặc thu được từ thiết bị điện tử khác.

Bộ mã hóa 1908 có thể mã hóa nguồn 1906 để tạo ra dòng bit lớp cơ sở 1934 và dòng bit lớp nâng cao 1936. Ví dụ, bộ mã hóa 1908 có thể mã hóa loạt hình ảnh (ví dụ, video) trong nguồn 1906. Cụ thể, khả năng mã hóa video mở rộng được theo SNR cũng được biết là khả năng mở rộng chất lượng của cùng nguồn 1906 có thể được cung cấp cho bộ mã hóa lớp cơ sở thứ nhất và bộ mã hóa lớp nâng cao. Cụ thể, để mã hóa video mở rộng được sử dụng theo khả năng điều chỉnh được về mặt không gian, nguồn được giảm lấy mẫu có thể được sử dụng cho bộ mã hóa lớp cơ sở. Cụ thể, để mã hóa đa cảnh, một nguồn ảnh khác có thể được sử dụng cho bộ mã hóa lớp cơ sở và bộ mã hóa lớp nâng cao. Bộ mã hóa 1908 có thể tương tự với bộ mã hóa 1782 được mô tả dưới đây cùng với Fig.6B.

Các dòng bit 1934, 1936 có thể bao gồm dữ liệu hình ảnh mã hoá dựa trên nguồn 1906. Trong một số cấu hình, các dòng bit 1934, 1936 có thể bao gồm dữ liệu tiêu đề, như thông tin tiêu đề phiên, thông tin PPS v.v.. Khi các hình ảnh bổ sung trong nguồn 1906 đã mã hoá, các dòng bit 1934, 1936 có thể bao gồm một hoặc nhiều hình ảnh mã hoá.

Các dòng bit 1934, 1936 có thể được cung cấp cho bộ giải mã 1972. Bộ giải mã 1972 có thể bao gồm bộ giải mã lớp cơ sở 1980 và bộ giải mã lớp nâng cao 1990. Bộ giải mã video 1972 là phù hợp với việc giải mã video mở rộng được và mã hóa video đa cảnh. Theo một ví dụ, các dòng bit 1934, 1936 có thể được truyền đến thiết bị điện tử B 1970 sử dụng kết nối hữu tuyến hoặc vô tuyến. Trong một số trường hợp, thao tác này có thể được thực hiện qua mạng, như Internet hoặc Mạng cục bộ (LAN - Local Area Network). Như được thể hiện trên Fig.1B, bộ giải mã 1972 có thể được thực hiện trên thiết bị điện tử B 1970 một cách riêng rẽ với bộ mã hóa 1908 trên thiết bị điện tử A 1902. Tuy nhiên, cần lưu ý rằng, bộ mã hóa 1908 và bộ giải mã 1972 có thể được thực hiện trên cùng thiết bị điện tử trong một số cấu hình. Theo phương án thực hiện, trong đó bộ mã hóa 1908 và bộ giải mã 1972 được thực hiện trên cùng thiết bị điện tử, chẳng hạn, các dòng bit 1934, 1936 có thể được cung

cấp qua bus cho bộ giải mã 1972 hoặc được lưu trữ trong bộ nhớ để truy hồi từ bộ giải mã 1972. Bộ giải mã 1972 có thể cung cấp lớp cơ sở được giải mã 1992 và (các) ảnh của lớp nâng cao đã giải mã 1994 làm đầu ra.

Bộ giải mã 1972 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Theo một cấu hình, bộ giải mã 1972 có thể là bộ giải mã chuẩn mã hóa video hiệu suất cao (HEVC), bao gồm mở rộng được và/hoặc đa cảnh. Các bộ giải mã khác cũng có thể được sử dụng. Bộ giải mã 1972 có thể tương tự với bộ giải mã 1812 được mô tả dưới đây cùng với Fig.7B. Ngoài ra, mỗi bộ mã hóa lớp cơ sở thứ nhất và/hoặc bộ mã hóa lớp nâng cao có thể bao gồm môđun tạo thông báo, như được mô tả cùng với Fig.1A. Ngoài ra, bộ giải mã lớp cơ sở thứ nhất và/hoặc bộ giải mã lớp nâng cao có thể bao gồm bộ đệm hình ảnh mã hoá và/hoặc bộ đệm hình ảnh giải mã, như được mô tả cùng với Fig.1A. Ngoài ra, các thiết bị điện tử trên Fig.1B có thể hoạt động phù hợp với các chức năng của các thiết bị điện tử trên Fig.1A, nếu có thể áp dụng.

Fig.2 là sơ đồ khối thể hiện một cấu hình của phương pháp 200 gửi thông báo. Phương pháp 200 này có thể được thực hiện nhờ bộ mã hóa 104 hoặc một trong các bộ phận phụ của nó (ví dụ, môđun tạo thông báo 108). Bộ mã hóa 104 này có thể tại bước 202 xác định cờ định thời hình ảnh (ví dụ, `sub_pic_cpb_params_present_flag`) chỉ báo liệu CPB 120 có hỗ trợ hoạt động ở mức hình ảnh phụ hay không. Ví dụ, khi cờ định thời hình ảnh được thiết đặt bằng 1, thì CPB 120 có thể hoạt động ở mức của đơn vị truy cập hoặc mức hình ảnh phụ. Cần lưu ý rằng, ngay cả khi khi cờ định thời hình ảnh được thiết đặt bằng 1, thì quyết định về việc liệu có hoạt động ở mức hình ảnh phụ hay không được để lại cho chính bộ giải mã 112.

Ở bước 204, bộ mã hóa 104 cũng có thể xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập. Ví dụ, bộ mã hóa 104 có thể xác định tham số trễ gỡ bỏ khỏi CPB đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập từ CPB 120. Theo cách khác, bộ mã hóa 104 có thể xác định khoảng trễ gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, `du_cpb_removal_delay[i]`) cho mỗi đơn vị giải mã trong đơn vị truy cập.

Ở bước 206, bộ mã hóa 104 cũng có thể xác định một hoặc nhiều tham số NAL chỉ báo lượng, được dịch đi một, các đơn vị NAL trong mỗi đơn vị giải mã ở điểm truy nhập. Ví dụ, bộ mã hóa 104 có thể xác định tham số NAL chung (ví dụ, `common_num_nalus_in_du_minus1`) có thể áp dụng được cho tất cả các đơn vị giải mã

trong đơn vị truy cập từ CPB 120. Theo cách khác, bộ mã hóa 104 có thể xác định khoảng trễ gỡ bỏ khỏi CPB của đơn vị giải mã riêng (ví dụ, `num_nalus_in_du_minus1[i]`) cho mỗi đơn vị giải mã trong đơn vị truy cập.

Ở bước 208, bộ mã hóa 104 cũng có thể gửi thông báo SEI định thời hình ảnh bao gồm cờ định thời hình ảnh, các khoảng trễ gỡ bỏ và các tham số NAL. Thông báo SEI định thời hình ảnh có thể bao gồm các tham số khác (ví dụ, `cpb_removal_delay`, `dpb_output_delay` v.v.). Ví dụ, thiết bị điện tử 102 có thể truyền thông báo qua một hoặc nhiều truyền vô tuyến, truyền hữu tuyến, bus thiết bị, mạng v.v.. Chẳng hạn, thiết bị điện tử A 102a có thể truyền thông báo đến thiết bị điện tử B 102b. Ví dụ, thông báo này có thể là một phần dòng bit 114. Trong một số cấu hình, ở bước 208, thiết bị điện tử A 102a có thể gửi thông báo đến thiết bị điện tử B 102b theo dòng truyền riêng biệt 110 (không phải là một phần dòng bit 114).

Chẳng hạn, thông báo đó có thể được gửi sử dụng thiết bị ngoài băng tần nào đó. Trong một số trường hợp, thông tin được chỉ báo trong bước 204, 206 có thể được gửi trong thông báo SEI khác với thông báo SEI định thời hình ảnh. Trong trường hợp khác, nữa thông tin được chỉ báo trong bước 204, 206 có thể được gửi trong tập tham số ví dụ tập tham số video và/hoặc tập tham số chuỗi và/hoặc tập tham số hình ảnh và/hoặc tập tham số tương thích và/hoặc tiêu đề phiên.

Fig.3 là sơ đồ khối thể hiện một cấu hình của phương pháp 300 để xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập. Nói cách khác, ở bước 204, phương pháp này 300 được thể hiện trên Fig.3 có thể tiếp tục minh họa trong phương pháp 200 được thể hiện trên Fig.2. Phương pháp 300 có thể được thực hiện bởi bộ mã hóa 104. ở bước 302, bộ mã hóa 104 có thể xác định liệu có bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) hay không.

Việc này có thể bao gồm việc xác định liệu cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay_flag`) được đặt hay không. Bộ mã hóa 104 có thể gửi tham số chung trong trường hợp các đơn vị giải mã được gỡ bỏ khỏi CPB ở các khoảng đều nhau. Đây có thể là trường hợp, ví dụ, khi mỗi đơn vị giải mã tương ứng với số lượng hàng nhất định của hình ảnh hoặc có kết cấu bình thường khác.

Ví dụ, cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể được thiết đặt bằng 1 khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có trong thông báo SEI định thời hình ảnh và 0 khi nó không được bao gồm. Nếu có (ví dụ, cờ được thiết đặt bằng 1), thì bộ

mã hóa 104 có thể tại bước 304 xác định tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập. Nếu sai (ví dụ, cờ được thiết đặt bằng 0), thì bộ mã hóa 104 có thể tại bước 306 xác định tham số trễ riêng gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, `du_cpb_removal_delay[i]`) cho mỗi đơn vị giải mã trong đơn vị truy cập.

Nếu tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có mặt trong thông báo SEI định thời hình ảnh, thì nó có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã đứng trước trước khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã hiện thời trong đơn vị truy cập kết hợp với thông báo SEI định thời hình ảnh.

Ví dụ, khi đơn vị giải mã là đơn vị giải mã thứ nhất trong đơn vị truy cập, thì tham số trễ gỡ bỏ khỏi CPB 120 của đơn vị giải mã chung có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã sau cùng trong đơn vị truy cập kết hợp với thông báo SEI về khoảng thời gian đệm gần đây nhất trong đơn vị truy cập đứng trước trước khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã thứ nhất trong đơn vị truy cập kết hợp với thông báo SEI định thời hình ảnh.

Khi đơn vị giải mã là đơn vị giải mã không phải là thứ nhất trong đơn vị truy cập, thì tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã trước đó trong đơn vị truy cập kết hợp với thông báo SEI định thời hình ảnh trước khi gỡ bỏ khỏi CPB đơn vị giải mã hiện thời trong đơn vị truy cập kết hợp với thông báo SEI định thời hình ảnh.

Trái lại, khi tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) không được gửi trong thông báo SEI định thời hình ảnh, thì tham số trễ riêng gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, `du_cpb_removal_delay[i]`) có thể có trong thông báo SEI định thời hình ảnh cho mỗi đơn vị giải mã trong đơn vị truy cập. Các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, `du_cpb_removal_delay[i]`) có thể xác định nhịp đồng hồ của hình ảnh phụ để chờ sau khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã sau cùng trước khi gỡ bỏ khỏi CPB 120 của đơn vị giải mã thứ i trong đơn vị truy cập kết hợp với thông báo SEI định thời hình ảnh. Các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã có thể được tính toán theo phần dư của bộ đếm $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ môđun trong đó `cpb_removal_delay_length_minus1 + 1` là độ dài của tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung.

Fig.4 là sơ đồ khối thể hiện một cấu hình của phương pháp 400 dùng để đếm dòng bit. Phương pháp 400 có thể được thực hiện nhờ bộ giải mã 112 trong thiết bị điện tử 102 (ví dụ, thiết bị điện tử B 102b), ở bước 402 có thể thu thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác). Ví dụ, thiết bị điện tử 102 ở bước 402 có thể thu thông báo qua một hoặc nhiều truyền vô tuyến, truyền hữu tuyến, bus thiết bị, mạng v.v.. Chẳng hạn, ở bước 402, thiết bị điện tử B 102b có thể thu thông báo từ thiết bị điện tử A 102a. Thông báo có thể là một phần dòng bit 114, ví dụ. Trong ví dụ khác, thiết bị điện tử B 102b có thể thu thông báo từ thiết bị điện tử A 102a theo dòng truyền riêng biệt 110 (ví dụ không phải là một phần dòng bit 114). Chẳng hạn, thông báo SEI định thời hình ảnh có thể được thu sử dụng thiết bị ngoài băng tần nào đó. Trong một số cấu hình, thông báo có thể bao gồm một hoặc nhiều cờ định thời hình ảnh, một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập và một hoặc nhiều tham số NAL. Do vậy, ở bước 402, việc thu thông báo có thể bao gồm thu một hoặc nhiều cờ định thời hình ảnh, một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập và một hoặc nhiều tham số NAL.

Ở bước 404, bộ giải mã 112 có thể xác định liệu CPB 120 có hoạt động ở mức của đơn vị truy cập hoặc mức hình ảnh phụ hay không. Ví dụ, bộ giải mã 112 có thể quyết định hoạt động trên cơ sở hình ảnh phụ nếu nó muốn đạt được thời gian chờ ngắn. Theo cách khác, quyết định có thể dựa trên việc liệu bộ giải mã 112 có đủ các nguồn để hỗ trợ cho hoạt động dựa trên hình ảnh phụ hay không. Nếu CPB 120 hoạt động ở mức hình ảnh phụ, ở bước 406, bộ giải mã có thể xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập. Ví dụ, bộ giải mã 112 có thể xác định tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập. Theo cách khác, bộ giải mã 112 có thể xác định khoảng trễ riêng gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, `du_cpb_removal_delay[i]`) cho mỗi đơn vị giải mã trong đơn vị truy cập. Nói cách khác, thông báo SEI định thời hình ảnh có thể bao gồm tham số chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập hoặc các tham số riêng cho từng đơn vị giải mã.

Ở bước 408, bộ giải mã 112 cũng có thể gỡ bỏ các đơn vị giải mã dựa trên các khoảng trễ gỡ bỏ cho các đơn vị giải mã, tức là, sử dụng hoặc tham số chung có thể áp dụng cho tất cả các đơn vị giải mã trong đơn vị truy cập hoặc các tham số riêng cho từng đơn vị giải mã. Ở bước 410, bộ giải mã 112 cũng có thể giải mã các đơn vị giải mã.

Bộ giải mã 112 có thể sử dụng ClockDiff biến đổi khi xác định thời gian gỡ bỏ để được xác định từ một số tham số đã được truyền tín hiệu. Cụ thể là, ClockDiff có thể được xác định theo
$$\text{ClockDiff} = (\text{num_units_in_tick} - (\text{num_units_in_sub_tick} * (\text{num_decoding_units_minus1} + 1)) / \text{time_scale}),$$
 trong đó num_units_in_tick là số đơn vị thời gian của đồng hồ hoạt động ở tần số time_scale Hz tương ứng với số gia lượng của bộ đếm nhịp đồng hồ, num_units_in_sub_tick là số đơn vị thời gian của đồng hồ hoạt động ở tần số time_scale Hz tương ứng với số gia lượng của số nhịp đồng hồ của hình ảnh phụ, num_decoding_units_minus1+1 là số lượng đơn vị giải mã trong đơn vị truy cập và time_scale là số đơn vị thời gian đi qua trong một giây.

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, low_delay_hrd_flag) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1, CPB hoạt động ở mức hình ảnh phụ và ClockDiff lớn hơn 0, thì thời gian gỡ bỏ đối với đơn vị giải mã m, $t_r(m)$ được xác định theo: $t_r(m) = t_{r,n}(m) + t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}) + \text{ClockDiff}$ trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, Ceil() là biểu thức trần và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m.

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, low_delay_hrd_flag) được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1, CPB hoạt động ở mức của đơn vị truy cập và ClockDiff lớn hơn 0, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n, $t_r(n)$ được xác định theo: $t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c) - \text{ClockDiff}$, trong đó $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ, Ceil() là biểu thức trần và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, low_delay_hrd_flag) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + \max((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m, t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, Ceil() là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m, $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n, t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n.

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + \max((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + \min((t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c)))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của

đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + (t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag` được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã không phải là đơn vị giải mã sau cùng của đơn vị truy cập được thiết đặt bằng $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m . và thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag` được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã không phải là đơn vị giải mã sau cùng của đơn vị truy cập được đặt là $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m . và thời gian gỡ bỏ đối với đơn vị giải mã sau cùng m của đơn vị truy cập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * \text{Ceil}((t_{af}(m) - t_{r,n}(m)) / t_c))$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $\text{Ceil}()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp (ví dụ, `low_delay_hrd_flag`) được thiết đặt bằng 1, $t_{r,n}(m) < t_{af}(m)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức hình ảnh phụ, thì thời gian gỡ bỏ đối với đơn vị giải mã được thiết đặt bằng $t_r(m) = t_{af}(m)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã m , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ, $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n và $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã trong đơn vị truy cập n .

Khi cờ của bộ giải mã tham chiếu giả định (HRD) có độ trễ thấp được thiết đặt bằng 1, $t_{r,n}(n) < t_{af}(n)$, cờ định thời hình ảnh được thiết đặt bằng 1 và CPB hoạt động ở mức của đơn vị truy cập, thì thời gian gỡ bỏ dùng cho đơn vị truy cập n , $t_r(n)$ theo: $t_r(n) = t_{af}(n)$, trong đó $t_{r,n}(m)$ là thời gian gỡ bỏ danh định của đơn vị giải mã sau cùng n , t_{c_sub} là nhịp đồng hồ của hình ảnh phụ, $Ceil()$ là biểu thức trần, $t_{af}(m)$ là thời gian đi đến cuối cùng của đơn vị giải mã sau cùng m , $t_{r,n}(n)$ là thời gian gỡ bỏ danh định của đơn vị truy cập n , t_c là nhịp đồng hồ và $t_{af}(n)$ là thời gian đi đến cuối cùng của đơn vị truy cập n .

Nếu CPB hoạt động ở mức của đơn vị truy cập, thì bộ giải mã 112 có thể tại bước 412 xác định tham số trễ gỡ bỏ khỏi CPB. Điều này có thể có trong thông báo SEI định thời hình ảnh thu được (ví dụ, `cpb_removal_delay`). Bộ giải mã 112 cũng có thể tại bước 414 gỡ bỏ đơn vị truy cập dựa trên tham số khoảng trễ gỡ bỏ CPB và tại bước 416 giải mã đơn vị truy cập. Nói cách khác, bộ giải mã 112 có thể một lúc giải mã toàn bộ các đơn vị truy cập, thay vì các bộ giải mã bên trong đơn vị truy cập.

Fig.5 là sơ đồ khối thể hiện một cấu hình của phương pháp 500 cho bước xác định một hoặc nhiều khoảng trễ gỡ bỏ cho các đơn vị giải mã trong đơn vị truy cập. Nói cách khác, phương pháp 500 được thể hiện trên Fig.5 có thể tiếp tục minh họa bước 406 trong phương pháp 400 được thể hiện trên Fig.4. Phương pháp 500 có thể được thực hiện bởi bộ giải mã 112. Ở bước 502, bộ giải mã 112 có thể xác định liệu thông báo SEI định thời hình ảnh thu được có bao gồm tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung hay không. Việc này có thể bao gồm việc xác định liệu cờ trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay_flag`) có được đặt hay không. Nếu có, thì ở bước 504, bộ giải mã 112 có thể xác định tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã chung (ví dụ, `common_du_cpb_removal_delay`) có thể áp dụng được cho tất cả các đơn vị giải mã trong đơn vị truy cập. Nếu không, thì ở bước 506, bộ giải mã 112 có thể xác định tham số

trễ riêng gỡ bỏ khỏi CPB của đơn vị giải mã (ví dụ, $du_cpb_removal_delay[i]$) cho mỗi đơn vị giải mã trong đơn vị truy cập.

Ngoài việc cải biến ngữ nghĩa của thông báo SEI định thời hình ảnh, các hệ thống và các phương pháp hiện thời cũng có thể áp đặt giới hạn dòng bit sao cho thao tác của thao tác CPB dựa trên hình ảnh phụ và thao tác CPB dựa trên đơn vị truy cập dẫn đến sự định thời giống nhau về việc gỡ bỏ đơn vị giải mã. Cụ thể là, khi cờ định thời hình ảnh (ví dụ, $sub_pic_cpb_params_present_flag$) được thiết đặt bằng 1, thì khoảng trễ gỡ bỏ CPB có thể được thiết đặt theo

Biểu thức 6

$$cpb_removal_delay = \frac{\left(\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i] \right) * t_{c,sub}}{t_c}$$

trong đó $du_cpb_removal_delay[i]$ là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ, $num_decoding_units_minus1$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một và i là chỉ số.

Theo cách khác, khoảng trễ gỡ bỏ CPB có thể được thiết đặt như mô tả tiếp sau đây: biến $T_{du}(k)$ được xác định theo:

Biểu thức 7

$$T_{du}(k) = T_{du}(k-1) + t_{c,sub} * \sum_{i=0}^{num_decoding_units_minus1} (du_cpb_removal_delay_minus1_k[i] + 1)$$

trong đó $du_cpb_removal_delay_minus1_k[i]$ và $num_decoding_units_minus1_k$ là các tham số cho đơn vị giải mã thứ I của đơn vị truy cập thứ k (với $k=0$ cho đơn vị truy cập đã khởi động HRD và $T_{du}(k)=0$ cho $k<1$) và trong đó $du_cpb_removal_delay_minus1_k[i]+1=du_cpb_removal_delay_minus1_k[i]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã cho đơn vị giải mã thứ I của đơn vị truy cập thứ k và $num_decoding_units_minus1_k$ là số lượng các đơn vị giải mã trong đơn vị truy cập thứ k , t_c là nhịp đồng hồ, $t_{c,sub}$ là nhịp đồng hồ của hình ảnh phụ và i và k là các chỉ số. Sau đó khi cờ định thời hình ảnh (ví dụ, $sub_pic_cpb_params_present_flag$) được thiết đặt bằng 1, thì điều kiện sau đây sẽ là đúng:

$(au_cpb_removal_delay_minus1 + 1) * t_c == T_{du}(k)$, trong đó $(au_cpb_removal_delay_minus1 + 1)=cpb_removal_delay$, khoảng trễ gỡ bỏ CPB. Do đó trong trường hợp này khoảng trễ gỡ bỏ CPB $(au_cpb_removal_delay_minus1 + 1)$ được đặt sao cho thao tác của thao tác CPB dựa trên hình ảnh phụ và thao tác CPB dựa trên đơn vị

truy cập dẫn đến sự định thời giống nhau của việc gỡ bỏ đơn vị truy cập và đơn vị giải mã cuối cùng của việc gỡ bỏ đơn vị truy cập.

Theo cách khác, khoảng trễ gỡ bỏ CPB có thể được thiết đặt theo

Biểu thức 8

$$-1 \leq [\text{cpb_removal_delay} * t_c - (\sum_{i=0}^{\text{num_decoding_units_minus1}} \text{du_cpb_removal_delay}[i]) * t_{c,\text{sub}}] \leq 1$$

trong đó $\text{du_cpb_removal_delay}[i]$ là các tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã, t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, $\text{num_decoding_units_minus1}$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một và i là chỉ số.

Theo cách khác, cpb_removal_delay và $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ có thể được thiết đặt theo: $\text{cpb_removal_delay} * t_c = \text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}] * t_{c,\text{sub}}$ trong đó $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ $\text{num_decoding_units_minus1}$, t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, $\text{num_decoding_units_minus1}$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một.

Ngoài việc cải biến ngữ nghĩa thông báo SEI định thời hình ảnh, các hệ thống và các phương pháp hiện thời cũng có thể áp đặt giới hạn dòng bit sao cho thao tác của thao tác CPB dựa trên hình ảnh phụ và thao tác CPB dựa trên đơn vị truy cập dẫn đến sự định thời giống nhau về việc gỡ bỏ đơn vị giải mã. Cụ thể là, khi cờ định thời hình ảnh (ví dụ, $\text{sub_pic_cpb_params_present_flag}$) được thiết đặt bằng 1, thì các giá trị cho cpb_removal_delay và $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ có thể được thiết đặt để thỏa mãn: $-1 \leq (\text{cpb_removal_delay} * t_c - \text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}] * t_{c,\text{sub}}) \leq 1$ trong đó $\text{du_cpb_removal_delay}[\text{num_decoding_units_minus1}]$ là tham số trễ gỡ bỏ khỏi CPB của đơn vị giải mã dùng cho đơn vị giải mã thứ $\text{num_decoding_units_minus1}$, t_c là nhịp đồng hồ, $t_{c,\text{sub}}$ là nhịp đồng hồ của hình ảnh phụ, $\text{num_decoding_units_minus1}$ là số lượng đơn vị giải mã trong đơn vị truy cập được dịch đi một.

Fig.6A là sơ đồ khối thể hiện một cấu hình của bộ mã hóa 604 trên thiết bị điện tử 602. Cần lưu ý rằng một hoặc nhiều thành phần được minh họa như được bao gồm bên trong thiết bị điện tử 602 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả

hai. Ví dụ, thiết bị điện tử 602 bao gồm bộ mã hóa 604, có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Chẳng hạn, bộ mã hóa 604 có thể được thực hiện như mạch điện, mạch tích hợp, mạch tích hợp chuyên dụng (ASIC), bộ xử lý nổi thông điện tử với bộ nhớ với các lệnh có thể thực hiện được, phần sụn, mảng cổng lập trình được dạng trường (FPGA - field-programmable gate array) v.v. hoặc kết hợp của chúng. Trong một số cấu hình, bộ mã hóa 604 có thể là bộ mã hóa HEVC.

Thiết bị điện tử 602 có thể bao gồm nguồn 622. Nguồn 622 có thể cung cấp ảnh hoặc dữ liệu ảnh (ví dụ, video) dưới dạng một hoặc nhiều hình ảnh đầu vào 606 cho bộ mã hóa 604. Các ví dụ về nguồn 622 có thể bao gồm các cảm biến ảnh, bộ nhớ, các giao diện truyền thông, các giao diện mạng, các bộ thu vô tuyến, các cổng v.v..

Một hoặc nhiều hình ảnh đầu vào 606 có thể được cung cấp cho môđun dự báo trong ảnh và bộ đệm tái tạo 624. Hình ảnh đầu vào 606 cũng có thể được cung cấp cho môđun đánh giá chuyển động và bù chuyển động 646 và cho môđun trừ 628.

Môđun dự báo trong ảnh và bộ đệm tái tạo 624 này có thể tạo ra thông tin chế độ trong ảnh 640 và tín hiệu trong ảnh 626 dựa trên một hoặc nhiều hình ảnh đầu vào 606 và dữ liệu tái tạo 660. Môđun đánh giá chuyển động và bù chuyển động 646 có thể tạo ra thông tin chế độ liên kết 648 và tín hiệu liên kết 644 dựa trên một hoặc nhiều hình ảnh đầu vào 606 và hình ảnh tham chiếu 678 từ bộ đệm hình ảnh giải mã 676. Trong một số cấu hình, bộ đệm hình ảnh giải mã 676 có thể bao gồm dữ liệu từ một hoặc nhiều hình ảnh tham chiếu trong bộ đệm hình ảnh giải mã 676.

Bộ mã hóa 604 có thể chọn giữa tín hiệu trong ảnh 626 và tín hiệu liên kết 644 phù hợp với chế độ. Tín hiệu trong ảnh 626 có thể được sử dụng để khai thác các đặc điểm không gian bên trong ảnh ở chế độ mã hóa trong ảnh. Tín hiệu liên kết 644 có thể được sử dụng để khai thác các đặc điểm tạm thời giữa các hình ảnh ở chế độ liên mã hóa. Mặc dù ở chế độ mã hóa trong ảnh, nhưng tín hiệu trong ảnh 626 có thể được cung cấp cho môđun trừ 628 và thông tin chế độ trong ảnh 640 có thể được cung cấp cho môđun mã hóa entropy 642. Mặc dù ở chế độ mã hóa liên kết, nhưng tín hiệu liên kết 644 có thể được cung cấp cho môđun trừ 628 và thông tin chế độ liên kết 648 có thể được cung cấp cho môđun mã hóa entropy 642.

Hoặc tín hiệu trong ảnh 626 hoặc tín hiệu liên kết 644 (tùy thuộc vào chế độ) được trừ ra khỏi hình ảnh đầu vào 606 tại môđun trừ 628 để tạo ra phần dư dự báo 630. Phần dư dự báo 630 được cung cấp cho môđun biến đổi 632. Môđun biến đổi 632 có thể nén phần dư

dự báo 630 để tạo ra tín hiệu biến đổi 634 được cung cấp cho môđun lượng tử hóa 636. Môđun lượng tử hóa 636 sẽ lượng tử hóa tín hiệu biến đổi 634 để tạo ra các hệ số biến đổi và lượng tử hóa (TQC - transformed and quantized coefficient) 638.

Các TQC 638 được cung cấp cho môđun mã hóa entropy 642 và môđun lượng tử hóa ngược 650. Môđun lượng tử hóa ngược 650 thực hiện lượng tử hóa ngược trên các TQC 638 để tạo ra tín hiệu lượng tử hóa ngược 652 mà được cung cấp cho môđun biến đổi ngược 654. Môđun biến đổi ngược 654 này sẽ giải nén tín hiệu lượng tử hóa ngược 652 để tạo ra tín hiệu giải nén 656 mà được cung cấp cho môđun tái tạo 658.

Môđun tái tạo 658 có thể tạo ra dữ liệu tái tạo 660 dựa trên tín hiệu giải nén 656. Ví dụ, môđun tái tạo 658 có thể tái tạo các hình ảnh (được biến đổi). Dữ liệu tái tạo 660 có thể được cung cấp cho bộ lọc giải khối 662 và cho môđun trong dự báo và bộ đệm tái tạo 624. Bộ lọc giải khối 662 đó có thể tạo ra tín hiệu lọc 664 dựa trên dữ liệu tái tạo 660.

Tín hiệu lọc 664 có thể được cung cấp cho môđun độ lệch thích ứng mẫu (SAO - sample adaptive offset) 666. Môđun SAO 666 có thể tạo ra thông tin SAO 668 được cung cấp cho môđun mã hóa entropy 642 và tín hiệu SAO 670 được tạo ra cho bộ lọc vòng lặp tương thích (ALF - adaptive loop filter) 672. ALF 672 tạo ra tín hiệu ALF 674 được cung cấp cho bộ đệm hình ảnh giải mã 676. Tín hiệu ALF 674 có thể bao gồm dữ liệu từ một hoặc nhiều hình ảnh mà chúng có thể được sử dụng làm các hình ảnh tham chiếu.

Môđun mã hóa entropy 642 có thể mã hóa các TQC 638 để tạo ra dòng bit A 614a (ví dụ, dữ liệu hình ảnh mã hoá). Ví dụ, môđun mã hóa entropy 642 có thể mã hóa các TQC 638 sử dụng kỹ thuật mã hoá độ dài biến đổi phù hợp với tình huống (CAVLC - Context-Adaptive Variable Length Coding) hoặc mã hóa thuật toán nhị phân phù hợp với tình huống (CABAC - Context-Adaptive Variable Length Coding). Cụ thể, môđun mã hóa entropy 642 có thể mã hóa các TQC 638 dựa trên một hoặc nhiều thông tin trong số thông tin chế độ trong ảnh 640, thông tin chế độ liên kết 648 và thông tin SAO 668. Dòng bit A 614a (ví dụ, dữ liệu hình ảnh mã hoá) có thể được cung cấp cho môđun tạo thông báo 608. Môđun tạo thông báo 608 có thể được tạo cấu hình tương tự với môđun tạo thông báo 108 được mô tả cùng với Fig.1. Ngoài ra hoặc Theo cách khác, môđun tạo thông báo 608 có thể thực hiện một hoặc nhiều quy trình như đã được mô tả cùng với Fig.2 và Fig.3.

Ví dụ, môđun tạo thông báo 608 có thể tạo ra thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác) bao gồm các tham số hình ảnh phụ. Các tham số hình ảnh phụ có thể bao gồm một hoặc nhiều khoảng trễ gỡ bỏ đối với các đơn vị giải mã (ví dụ,

`common_du_cpb_removal_delay` hoặc `du_cpb_removal_delay[i]`) và một hoặc nhiều tham số NAL (ví dụ, `common_num_nalus_in_du_minus1` hoặc `num_nalus_in_du_minus1[i]`). Trong một số cấu hình, thông báo này có thể được chèn vào trong dòng bit A 614a để tạo ra dòng bit B 614b. Do vậy, thông báo này có thể được tạo ra, ví dụ, sau khi dòng bit tổng thể A 614a được tạo ra (ví dụ, sau khi phần lớn dòng bit B 614b được tạo ra). Trong các cấu hình khác, thông báo này có thể không được chèn vào trong dòng bit A 614a (trong trường hợp đó dòng bit B 614b có thể là giống như dòng bit A 614a), nhưng có thể được tạo ra trong dòng truyền riêng biệt 610.

Trong một số cấu hình, thiết bị điện tử 602 sẽ gửi dòng bit 614 tới thiết bị điện tử khác. Ví dụ, dòng bit 614 có thể cung cấp cho giao diện truyền thông, giao diện mạng, máy truyền phát vô tuyến, cổng v.v.. Chẳng hạn, dòng bit 614 có thể được truyền đến thiết bị điện tử khác thông qua LAN, Internet, trạm cơ sở điện thoại di động v.v.. Dòng bit 614 có thể bằng cách bổ sung hoặc thay thế được lưu trữ trong bộ nhớ hoặc phần tử khác trên thiết bị điện tử 602.

Fig.6B là sơ đồ khối thể hiện một cấu hình của bộ mã hóa video 1782 trên thiết bị điện tử 1702. Bộ mã hóa video 1782 này có thể bao gồm bộ mã hóa lớp nâng cao 1706, bộ mã hóa lớp cơ sở 1709, khối nâng cao độ phân giải 1770 và giao diện đầu ra 1780. Bộ mã hóa video trên Fig.6B, ví dụ, phù hợp cho việc mã hóa video có thể mở rộng và mã hóa video đa cảnh, như được mô tả trong bản mô tả này.

Bộ mã hóa lớp nâng cao 1706 có thể bao gồm đầu vào video 1781 để thu hình ảnh đầu vào 1704. Đầu ra của đầu vào video 1781 có thể được cung cấp cho bộ cộng/bộ trừ 1783 để thu đầu ra của bộ chọn dự báo 1750. Đầu ra của bộ cộng/bộ trừ 1783 có thể được cung cấp cho khối biến đổi và lượng tử hóa 1752. Đầu ra của khối biến đổi và lượng tử hóa 1752 này có thể được cung cấp cho khối mã hóa entropy 1748 và khối định tỷ lệ và biến đổi ngược 1772. Sau khi mã hóa entropy 1748 được thực hiện, đầu ra của khối thực hiện mã hóa entropy 1748 có thể được cung cấp cho giao diện đầu ra 1780. Giao diện đầu ra 1780 này có thể kết xuất cả dòng bit video của lớp cơ sở giải mã 1707 và dòng bit video của lớp nâng cao mã hóa 1710.

Đầu ra của khối định tỷ lệ và biến đổi ngược 1772 có thể được cung cấp cho bộ cộng 1779. Bộ cộng 1779 cũng có thể thu đầu ra của bộ chọn dự báo 1750. Đầu ra của bộ cộng 1779 có thể được cung cấp cho khối giải khối 1751. Đầu ra của khối giải khối 1751 có thể được cung cấp cho bộ đệm tham chiếu 1794. Đầu ra của bộ đệm tham chiếu 1794 có thể

được cung cấp cho khối bù chuyển động 1754. Đầu ra của khối bù chuyển động 1754 có thể được cung cấp cho bộ chọn dự báo 1750. Đầu ra của bộ đệm tham chiếu 1794 cũng có thể được cung cấp cho bộ dự báo trong ảnh 1756. Đầu ra của bộ dự báo trong ảnh 1756 có thể được cung cấp cho bộ chọn dự báo 1750. Bộ chọn dự báo 1750 cũng có thể thu đầu ra của khối nâng cao độ phân giải 1770.

Bộ mã hóa lớp cơ sở thứ nhất 1709 có thể bao gồm đầu vào video 1762 thu hình ảnh đầu vào được giảm lấy mẫu, hoặc nội dung ảnh khác phù hợp để kết hợp với ảnh khác, hoặc hình ảnh đầu vào của hình vẽ thay thế hoặc cùng hình ảnh đầu vào 1703 (nghĩa là giống như hình ảnh đầu vào 1704 thu được bởi bộ mã hóa lớp nâng cao 1706). Đầu ra của đầu vào video 1762 có thể được cung cấp cho vòng lặp dự báo mã hóa 1764. Mã hóa entropy 1766 có thể được tạo ra ở đầu ra của vòng lặp dự báo mã hóa 1764. Đầu ra của vòng lặp dự báo mã hóa 1764 cũng có thể được cung cấp cho bộ đệm tham chiếu 1768. Bộ đệm tham chiếu 1768 có thể tạo ra hồi tiếp cho vòng lặp dự báo mã hóa 1764. Đầu ra của bộ đệm tham chiếu 1768 cũng có thể được cung cấp cho khối nâng cao độ phân giải 1770. Một khi bước mã hóa entropy 1766 được thực hiện, thì đầu ra có thể được cung cấp cho giao diện đầu ra 1780. Dòng bit video của lớp cơ sở mã hóa 1707 và/hoặc dòng bit video của lớp nâng cao mã hóa 1710 có thể được cung cấp cho một hoặc nhiều môđun tạo thông báo, nếu cần.

Fig.7A là sơ đồ khối thể hiện một cấu hình của bộ giải mã 712 trên thiết bị điện tử 702. Bộ giải mã 712 có thể có trong thiết bị điện tử 702. Ví dụ, bộ giải mã 712 có thể là bộ giải mã HEVC. Bộ giải mã 712 và một hoặc nhiều thành phần được minh họa như có trong bộ giải mã 712 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Bộ giải mã 712 có thể thu dòng bit 714 (ví dụ, một hoặc nhiều hình ảnh mã hoá và dữ liệu tiêu đề có trong dòng bit 714) để giải mã. Trong một số cấu hình, dòng bit thu được 714 có thể bao gồm dữ liệu tiêu đề thu được, như thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác), tiêu đề phiên, PPS v.v.. Trong một số cấu hình, bộ giải mã 712 còn có thể thu bổ sung dòng truyền riêng biệt 710. Dòng truyền riêng biệt 710 có thể bao gồm thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác). Ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác có thể được thu trong dòng truyền riêng biệt 710 thay vì trong dòng bit 714. Tuy nhiên, cần lưu ý rằng dòng truyền riêng biệt 710 có thể là tùy ý và có thể không được sử dụng trong một số cấu hình.

Bộ giải mã 712 bao gồm CPB 720. CPB 720 này có thể được tạo cấu hình tương tự với CPB 120 được mô tả cùng với Fig.1 nêu trên. Ngoài ra hoặc Theo cách khác, bộ giải mã

712 có thể thực hiện một hoặc nhiều quy trình được mô tả cùng với Fig.4 và Fig.5. Ví dụ, bộ giải mã 712 có thể thu thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác) với các tham số hình ảnh phụ và gỡ bỏ và giải mã các đơn vị giải mã trong đơn vị truy cập dựa trên các tham số hình ảnh phụ, Cần lưu ý rằng, một hoặc nhiều đơn vị truy cập có thể có trong dòng bit và có thể bao gồm một hoặc nhiều dữ liệu hình ảnh mã hoá và dữ liệu tiêu đề.

Bộ đệm hình ảnh mã hoá (CPB) 720 có thể cung cấp dữ liệu hình ảnh mã hoá cho môđun giải mã entropy 701. Dữ liệu hình ảnh mã hoá có thể đã giải mã entropy nhờ môđun giải mã entropy 701, nhờ đó tạo ra tín hiệu thông tin chuyển động 703 và các hệ số lượng tử hóa, định tỷ lệ và/hoặc biến đổi 705.

Tín hiệu thông tin chuyển động 703 có thể được kết hợp với một phần tín hiệu khung tham chiếu 798 từ bộ đệm hình ảnh giải mã 709 tại môđun bù chuyển động 780, có thể tạo ra tín hiệu dự báo liên khung 782. Các hệ số lượng tử hóa, định tỷ lệ và/hoặc biến đổi 705 có thể được lượng tử hóa ngược, được định tỷ lệ và được biến đổi ngược bởi môđun ngược 707, nhờ đó tạo ra tín hiệu dư giải mã 784. Tín hiệu dư giải mã 784 có thể có thể được bổ sung vào tín hiệu dự báo 792 để tạo ra tín hiệu kết hợp 786. Tín hiệu dự báo 792 này có thể là tín hiệu được chọn từ hoặc tín hiệu dự báo liên khung 782 được tạo ra bởi môđun bù chuyển động 780 hoặc tín hiệu dự báo trong khung 790 được tạo ra bởi môđun dự báo trong khung 788. Trong một số cấu hình, việc chọn tín hiệu này có thể dựa trên (ví dụ, được điều khiển bởi) dòng bit 714.

Tín hiệu dự báo trong khung 790 đó có thể được dự báo từ thông tin giải mã trước đây từ tín hiệu kết hợp 786 (ví dụ trong khung hiện thời). Tín hiệu kết hợp 786 cũng có thể được lọc bởi bộ lọc giải khối 794. Tín hiệu lọc 796 sinh ra có thể được ghi vào bộ đệm hình ảnh giải mã 709. Tín hiệu lọc 796 sinh ra này có thể bao gồm hình ảnh giải mã. Bộ đệm hình ảnh giải mã 709 có thể cung cấp hình ảnh giải mã mà nó có thể được kết xuất trong bước 718. Trong một số trường hợp bộ đệm hình ảnh giải mã 709 có thể được cân nhắc làm bộ nhớ khung.

Fig.7B là sơ đồ khối thể hiện một cấu hình của bộ giải mã video 1812 trên thiết bị điện tử 1802. Bộ giải mã video 1812 có thể bao gồm bộ giải mã lớp nâng cao 1815 và bộ giải mã lớp cơ sở 1813. Bộ giải mã video 812 có thể bao gồm giao diện 1889 và nâng cao độ phân giải 1870. Bộ giải mã video trên Fig.7B, ví dụ, là phù hợp với việc giải mã video mở rộng được và mã hóa video đa cảnh, như được mô tả trong bản mô tả này.

Giao diện 1889 có thể thu dòng video mã hóa 1885. Dòng video mã hóa 1885 này có thể gồm có dòng video mã hóa lớp cơ sở và dòng video mã hóa lớp nâng cao. Hai luồng này có thể được gửi một cách riêng hoặc cùng nhau. Giao diện 1889 có thể cung cấp một phần hoặc toàn bộ dòng video mã hóa 1885 đến khối giải mã entropy 1886 trong bộ giải mã lớp cơ sở thứ nhất 1813. Đầu ra của khối giải mã entropy 1886 có thể được cung cấp cho vòng lặp dự báo giải mã 1887. Đầu ra của vòng lặp dự báo giải mã 1887 có thể được cung cấp cho bộ đệm tham chiếu 1888. Bộ đệm tham chiếu có thể tạo ra hồi tiếp cho vòng lặp dự báo giải mã 1887. Bộ đệm tham chiếu 1888 cũng có thể kết xuất dòng video lớp cơ sở đã giải mã 1884.

Giao diện 1889 cũng có thể cung cấp một phần hoặc toàn bộ dòng video mã hóa 1885 cho khối giải mã entropy 1890 trong bộ giải mã lớp nâng cao 1815. Đầu ra của khối giải mã entropy 1890 có thể được cung cấp cho khối lượng tử hóa ngược 1891. Đầu ra của khối lượng tử hóa ngược 1891 có thể được cung cấp cho bộ cộng 1892. Bộ cộng 1892 này có thể bổ sung đầu ra của khối lượng tử hóa ngược 1891 và đầu ra của khối chọn dự báo 1895. Đầu ra của bộ cộng 1892 này có thể được cung cấp cho khối giải khối 1893. Đầu ra của khối giải khối 1893 này có thể được cung cấp cho bộ đệm tham chiếu 1894. Bộ đệm tham chiếu 1894 này có thể kết xuất dòng video lớp nâng cao đã giải mã 1882. Đầu ra của bộ đệm tham chiếu 1894 cũng có thể được cung cấp cho bộ dự báo trong ảnh 1897. Bộ giải mã lớp nâng cao 1815 có thể bao gồm bộ bù chuyển động 1896. Bộ bù chuyển động 1896 có thể được thực hiện sau khi nâng cao độ phân giải 1870. Khối chọn dự báo 1895 có thể thu đầu ra của bộ dự báo trong ảnh 1897 và đầu ra của bộ bù chuyển động 1896. Ngoài ra, bộ giải mã có thể bao gồm một hoặc nhiều bộ đệm hình ảnh mã hóa, nếu cần, như cùng với giao diện 1889.

Fig.8 minh họa các thành phần khác nhau có thể được sử dụng trong thiết bị điện tử truyền 802. Một hoặc nhiều thiết bị điện tử 102, 602, 702 được mô tả trong bản mô tả này có thể được thực hiện phù hợp với thiết bị điện tử truyền 802 được thể hiện trên Fig.8.

Thiết bị điện tử truyền 802 bao gồm bộ xử lý 817 điều khiển hoạt động của thiết bị điện tử 802. Bộ xử lý 817 này cũng có thể được gọi là CPU. Bộ nhớ 811, có thể bao gồm cả bộ nhớ chỉ đọc (ROM - read-only memory), bộ nhớ truy cập ngẫu nhiên (RAM - random access memory) hoặc loại thiết bị bất kỳ có thể lưu trữ thông tin, tạo ra các lệnh 813a (ví dụ, các lệnh thực thi được) và dữ liệu 815a cho bộ xử lý 817. Một phần bộ nhớ 811 có thể bao

gồm bộ nhớ truy cập ngẫu nhiên bất khả biến (NVRAM - non-volatile random access memory). Bộ nhớ 811 có thể được nối thông điện tử với bộ xử lý 817.

Các lệnh 813b và dữ liệu 815b cũng có thể nằm trong bộ xử lý 817. Các lệnh 813b và/hoặc dữ liệu 815b được tải vào bộ xử lý 817 có thể bao gồm các lệnh 813a và/hoặc dữ liệu 815a từ bộ nhớ 811 được tải về để thực hiện hoặc xử lý bởi bộ xử lý 817. Các lệnh 813b có thể được thực hiện bởi bộ xử lý 817 để thực hiện các hệ thống và các phương pháp được bộc lộ trong sáng chế. Ví dụ, các lệnh 813b có thể thực hiện được để thực hiện một hoặc nhiều phương pháp 200, 300, 400, 500 được mô tả trên đây.

Thiết bị điện tử truyền 802 có thể bao gồm một hoặc nhiều giao diện truyền thông 819 để truyền thông với các thiết bị điện tử khác (ví dụ, thiết bị điện tử thu). Các giao diện truyền thông 819 có thể dựa trên công nghệ truyền thông hữu tuyến, công nghệ truyền thông vô tuyến, hoặc cả hai. Các ví dụ về giao diện truyền thông 819 bao gồm cổng nối tiếp, cổng song song, bus nối tiếp đa năng (USB - Universal Serial Bus), bộ thích ứng Ethernet, bus giao diện IEEE 1394, giao diện bus của giao diện hệ thống máy tính nhỏ (SCSI - small computer system interface), cổng giao tiếp hồng ngoại (IR - infrared), bộ thích ứng truyền thông vô tuyến Bluetooth, bộ thu phát vô tuyến phù hợp với tiêu chuẩn Dự án đối tác thế hệ thứ 3 (3GPP - 3rd Generation Partnership Project) và tương tự.

Thiết bị điện tử truyền 802 có thể bao gồm một hoặc nhiều thiết bị kết xuất 823 và một hoặc nhiều thiết bị nhập liệu 821. Các ví dụ về các thiết bị kết xuất 823 bao gồm loa, máy in v.v.. Một loại thiết bị kết xuất có thể có trong thiết bị điện tử 802 là thiết bị hiển thị 825. Các thiết bị hiển thị 825 được sử dụng với các cấu hình được bộc lộ trong sáng chế có thể sử dụng công nghệ dự báo ảnh phù hợp bất kỳ, như đèn tia âm cực (CRT - light-emitting diode), màn hình tinh thể lỏng (LCD - light-emitting diode), điốt phát sáng (LED - light-emitting diode), plasma khí, điện phát quang hoặc loại tương tự. Bộ điều khiển màn hiển thị 827 có thể được tạo ra để biến đổi dữ liệu được lưu trữ trong bộ nhớ 811 thành văn bản, đồ thị và/hoặc các hình ảnh động (nếu thích hợp) được thể hiện trên màn hình 825. Các ví dụ về các thiết bị nhập liệu 821 bao gồm bàn phím, chuột, micrô, thiết bị điều khiển từ xa, nút bấm, cần điều khiển, bi xoay, bàn rê chuột cảm ứng, màn hình cảm ứng, bút quang v.v..

Các thành phần khác nhau của thiết bị điện tử truyền 802 được ghép nối với nhau nhờ hệ thống bus 829, có thể bao gồm bus công suất, bus tín hiệu điều khiển và bus tín hiệu trạng thái, ngoài bus dữ liệu. Tuy nhiên, để rõ ràng, các bus khác nhau được thể hiện trên

Fig.8 là hệ thống bus 829. Thiết bị điện tử truyền 802 được thể hiện trên Fig.8 là sơ đồ khối chức năng thay vì liệt kê các phần tử cụ thể.

Fig.9 là sơ đồ khối thể hiện các thành phần khác nhau có thể được sử dụng trong thiết bị điện tử thu 902. Một hoặc nhiều thiết bị điện tử 102, 602, 702 được mô tả trong bản mô tả này có thể được thực hiện phù hợp với thiết bị điện tử thu 902 được thể hiện trên Fig.9.

Thiết bị điện tử thu 902 bao gồm bộ xử lý 917 điều khiển hoạt động của thiết bị điện tử 902. Bộ xử lý 917 cũng có thể được gọi là CPU. Bộ nhớ 911, có thể bao gồm cả bộ nhớ chỉ đọc (ROM), bộ nhớ truy cập ngẫu nhiên (RAM) hoặc loại thiết bị bất kỳ có thể lưu trữ thông tin, tạo ra các lệnh 913a (ví dụ, các lệnh thực thi được) và dữ liệu 915a cho bộ xử lý 917. Một phần bộ nhớ 911 có thể bao gồm bộ nhớ truy cập ngẫu nhiên bất khả biến (NVRAM). Bộ nhớ 911 có thể được nối thông điện tử với bộ xử lý 917.

Các lệnh 913b và dữ liệu 915b cũng có thể nằm trong bộ xử lý 917. Các lệnh 913b và/hoặc dữ liệu 915b được tải vào bộ xử lý 917 có thể bao gồm các lệnh 913a và/hoặc dữ liệu 915a từ bộ nhớ 911 được tải về để thực hiện hoặc xử lý bởi bộ xử lý 917. Các lệnh 913b có thể được thực thi bởi bộ xử lý 917 để thực hiện các hệ thống và các phương pháp được bộc lộ trong sáng chế. Ví dụ, các lệnh 913b có thể thực hiện được để thực hiện một hoặc nhiều phương pháp 200, 300, 400, 500 được mô tả trên đây.

Thiết bị điện tử thu 902 có thể bao gồm một hoặc nhiều giao diện truyền thông 919 để truyền thông với các thiết bị điện tử khác (ví dụ, thiết bị điện tử truyền). Giao diện truyền thông 919 có thể dựa trên công nghệ truyền thông hữu tuyến, công nghệ truyền thông vô tuyến, hoặc cả hai. Các ví dụ về giao diện truyền thông 919 bao gồm cổng nối tiếp, cổng song song, bus nối tiếp đa năng (USB), bộ thích ứng Ethernet, giao diện bus IEEE 1394, giao diện bus của giao diện hệ thống máy tính nhỏ (SCSI), cổng giao tiếp hồng ngoại (IR), bộ thích ứng truyền thông không dây Bluetooth, bộ thu phát vô tuyến phù hợp với tiêu chuẩn Dự án đối tác thế hệ thứ 3 (3GPP) và tương tự.

Thiết bị điện tử thu 902 có thể bao gồm một hoặc nhiều thiết bị kết xuất 923 và một hoặc nhiều thiết bị nhập liệu 921. Các ví dụ về các thiết bị kết xuất 923 bao gồm loa, máy in v.v.. Một loại thiết bị kết xuất có thể có trong thiết bị điện tử 902 là thiết bị hiển thị 925. Các thiết bị hiển thị 925 được sử dụng với các cấu hình được bộc lộ trong sáng chế có thể sử dụng công nghệ dự báo ảnh phù hợp bất kỳ, như đèn tia catốt (CRT), màn hình tinh thể lỏng (LCD), diốt phát sáng (LED), plasma khí, điện phát quang hoặc loại tương tự. Bộ điều khiển màn hiển thị 927 có thể được tạo ra để biến đổi dữ liệu được lưu trữ trong bộ nhớ 911 thành

văn bản, đồ thị và/hoặc các hình ảnh động (nếu thích hợp) được thể hiện trên màn hình 925. Các ví dụ về các thiết bị nhập liệu 921 bao gồm bàn phím, chuột, micrô, thiết bị điều khiển từ xa, nút bấm, cần điều khiển, bi xoay, bàn rê chuột cảm ứng, màn hình cảm ứng, bút quang v.v..

Các thành phần khác nhau của thiết bị điện tử thu 902 được ghép nối với nhau nhờ hệ thống bus 929, có thể bao gồm bus công suất, bus tín hiệu điều khiển và bus tín hiệu trạng thái, ngoài bus dữ liệu. Tuy nhiên, để rõ ràng, các bus khác nhau được thể hiện trên Fig.9 là hệ thống bus 929. Thiết bị điện tử thu 902 được thể hiện trên Fig.9 là sơ đồ khối chức năng thay cho việc liệt kê các phần tử cụ thể.

Fig.10 là sơ đồ khối thể hiện một cấu hình của thiết bị điện tử 1002 trong đó các hệ thống và các phương pháp gửi thông báo có thể được thực hiện. Thiết bị điện tử 1002 bao gồm các phương tiện mã hóa 1031 và các phương tiện truyền 1033. Các các phương tiện mã hóa 1031 và các phương tiện truyền 1033 này có thể được tạo cấu hình để thực hiện một hoặc nhiều chức năng được mô tả liên quan đến một hoặc nhiều hình vẽ trong số Fig.1, Fig.2, Fig.3, Fig.6 và Fig.8 nêu trên. Ví dụ, các phương tiện mã hóa 1031 và các phương tiện truyền 1033 có thể tạo ra dòng bit 1014. Fig.8 nêu trên minh họa một ví dụ về kết cấu thiết bị cụ thể trên Fig.10. Các kết cấu khác có thể được thực hiện để thực hiện một hoặc nhiều chức năng trên Fig.1, Fig.2, Fig.3, Fig.6 và Fig.8. Ví dụ, DSP có thể được thực hiện bởi phần mềm.

Fig.11 là sơ đồ khối thể hiện một cấu hình của thiết bị điện tử 1102 trong đó các hệ thống và các phương pháp đệm dòng bit 1114 có thể được thực hiện. Thiết bị điện tử 1102 có thể bao gồm các phương tiện thu 1135 và các phương tiện giải mã 1137. Các phương tiện thu 1135 và các phương tiện giải mã 1137 này có thể được tạo cấu hình để thực hiện một hoặc nhiều chức năng được mô tả liên quan đến một hoặc nhiều hình vẽ trong số Fig.1, Fig.4, Fig.5, Fig.7 và Fig.9 nêu trên. Ví dụ, các phương tiện thu 1135 và các phương tiện giải mã 1137 có thể thu dòng bit 1114. Fig.9 nêu trên minh họa một ví dụ về kết cấu thiết bị cụ thể trên Fig.11. Các kết cấu khác có thể được thực hiện để thực hiện một hoặc nhiều chức năng trên Fig.1, Fig.4, Fig.5, Fig.7 và Fig.9. Ví dụ, DSP có thể được thực hiện bởi phần mềm.

Fig.12 là sơ đồ khối thể hiện một cấu hình của phương pháp 1200 để vận hành bộ đệm hình ảnh giải mã (DPB). Phương pháp 1200 này có thể được thực hiện bởi bộ mã hóa 104 hoặc một trong các bộ phận phụ của nó (ví dụ, môđun của bộ đệm hình ảnh giải mã 676). Phương pháp 1200 này có thể được thực hiện bởi bộ giải mã 112 trong thiết bị điện tử

102 (ví dụ, thiết bị điện tử B 102b). Ngoài ra hoặc Theo cách khác, phương pháp 1200 có thể được thực hiện bởi bộ giải mã 712 hoặc một trong các bộ phận phụ của nó (ví dụ, môđun của bộ đệm hình ảnh giải mã 709). Bộ giải mã này có thể phân giải cú pháp tiêu đề phiên thứ nhất của hình ảnh 1202. Việc kết xuất và gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thì khi đơn vị giải mã thứ nhất của đơn vị truy cập có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB và tiếp tục diễn ra như sau.

- Quy trình giải mã đối với tập hình ảnh tham chiếu (RPS) được hủy bỏ. Tập hình ảnh tham chiếu là tập hợp của các hình ảnh tham chiếu kết hợp với ảnh, gồm có tất cả các hình ảnh tham chiếu mà chúng đứng trước hình ảnh liên quan trong thứ tự giải mã, có thể được sử dụng để dự báo liên kết của hình ảnh liên quan hoặc ảnh bất kỳ đứng sau hình ảnh liên quan trong thứ tự giải mã.

- Dòng bit của video có thể bao gồm cấu trúc cú pháp được đặt vào các bó dữ liệu logic được gọi chung là các đơn vị lớp trừu tượng hoá mạng (NAL - Network Abstraction Layer). Mỗi đơn vị NAL bao gồm tiêu đề của đơn vị NAL, chẳng hạn như tiêu đề của đơn vị NAL hai byte (ví dụ, 16 bit), để xác định mục đích của phần dữ liệu chuyên liên quan. Ví dụ, mỗi phiên (và/hoặc hình ảnh) đã mã hóa có thể được mã hóa thành một hoặc nhiều phiên (và/hoặc ảnh) đơn vị NAL. Các đơn vị NAL khác có thể được bao gồm cho các hạng mục dữ liệu khác, như ví dụ, thông tin nâng cao bổ sung, phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời (TSA - temporal sub-layer access), phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời theo bước (STSA - step-wise temporal sub-layer access), phiên mã hoá không TSA, ảnh kế tiếp không STSA, phiên mã hoá của hình ảnh truy nhập liên kết gãy, phiên mã hoá của hình ảnh làm mới giải mã tức thì, phiên mã hoá của hình ảnh truy nhập ngẫu nhiên sạch, phiên mã hoá của hình ảnh chính có thể giải mã, phiên mã hoá của được dán nhãn đối với hình ảnh gỡ bỏ, tập tham số video, tập tham số chuỗi, tập tham số hình ảnh, dấu phân cách đơn vị truy cập, đuôi của chuỗi, đuôi của dòng bit, dữ liệu của bộ lọc và/hoặc thông báo thông tin nâng cao chuỗi. Bảng (4) minh họa một ví dụ về các mã của đơn vị NAL và các lớp của loại đơn vị NAL. Các loại đơn vị NAL khác có thể được bao gồm, nếu cần. Cần hiểu rằng, các giá trị của loại đơn vị NAL đối với các đơn vị NAL được thể hiện trong bảng (4) có thể được trộn lại và được ấn định lại. Ngoài ra các loại đơn vị NAL bổ sung cũng có thể được bổ sung. Ngoài ra, một số loại đơn vị NAL cũng có thể được gỡ bỏ.

Hình ảnh của điểm truy nhập ngẫu nhiên trong ảnh (IRAP - intra random access point) là hình ảnh mã hoá, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` nằm trong khoảng từ `BLA_W_LP` đến `RSV_IRAP_VCL23`, bao gồm cả giá trị đầu cuối như được thể hiện trong bảng (4). Hình ảnh IRAP chỉ chứa các phiên mã hóa trong ảnh (i). Hình ảnh làm mới giải mã tức thì (IDR) là hình ảnh IRAP, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` bằng `IDR_W_RADL` hoặc `IDR_N_LP` như được thể hiện trong bảng (4). Hình ảnh làm mới giải mã tức thì (IDR - instantaneous decoding referesh) chỉ chứa các phiên I và có thể là hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh IDR là hình ảnh thứ nhất của chuỗi video mã hóa (CVS - coded video sequence) theo thứ tự giải mã. Hình ảnh truy nhập liên kết gãy (BLA - broken link access) là hình ảnh IRAP, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` bằng `BLA_W_LP`, `BLA_W_RADL`, hoặc `BLA_N_LP` như được thể hiện trong bảng (4). Hình ảnh BLA chỉ chứa các phiên I và có thể là hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh BLA bắt đầu chuỗi video mã hóa mới và có cùng tác dụng trên quy trình giải mã giống như hình ảnh IDR. Tuy nhiên, hình ảnh BLA chứa các phần tử cú pháp để xác định tập hình ảnh tham chiếu không trống.

Bảng 4

<code>nal_unit_type</code>	Tên của <code>nal_unit_type</code>	Nội dung đơn vị NAL và cấu trúc cú pháp của phần dữ liệu chuyển chuỗi byte thô (RBSP)	Lớp của loại đơn vị NAL
0	<code>TRAIL_N</code>	Phân đoạn phiên mã hoá của hình ảnh kéo không TSA, không STSA	Lớp mã hóa video (VCL)
1	<code>TRAIL_R</code>	<code>slice_segment_layer_rbsp()</code>	
2	<code>TSA_N</code>	Phân đoạn phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời (TSA)	VCL
3	<code>TSA_R</code>	<code>slice_segment_layer_rbsp()</code>	
4	<code>STSA_N</code>	Phân đoạn phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời theo bước (STSA)	VCL
5	<code>STSA_R</code>	<code>slice_segment_layer_rbsp()</code>	
6	<code>RADL_N</code>	Phân đoạn phiên mã hoá của hình ảnh chính có thể giải mã truy nhập ngẫu nhiên (RADL)	VCL
7	<code>RADL_R</code>	<code>slice_segment_layer_rbsp()</code>	
8	<code>RASL_N</code>	Phân đoạn phiên mã hoá của hình ảnh dẫn đầu bỏ qua truy nhập ngẫu nhiên (RASL)	VCL
9	<code>RASL_R</code>	<code>slice_segment_layer_rbsp()</code>	

10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Các loại đơn vị NAL VCL không tham chiếu lớp phụ không IRAP dự phòng	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Các loại đơn vị NAL VCL tham chiếu lớp phụ không IRAP dự phòng	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Phân đoạn phiên mã hoá của hình ảnh truy nhập liên kết gãy (BLA) slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Phân đoạn phiên mã hoá của hình ảnh làm mới giải mã tức thì (IDR) slice_segment_layer_rbsp()	VCL
21	CRA_NUT	Phân đoạn phiên mã hoá của hình ảnh truy nhập ngẫu nhiên sạch (CRA) slice_segment_layer_rbsp()	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Các loại đơn vị NAL VCL IRAP dự phòng	VCL
24..31	RSV_VCL24.. RSV_VCL31	Các loại đơn vị NAL VCL không IRAP dự phòng	VCL
32	VPS_NUT	Tập tham số video Video_parameter_set_rbsp()	lớp mã hóa không video (không
33	SPS_NUT	Tập tham số dãy seq_parameter_set_rbsp()	không VCL
34	PPS_NUT	Tập tham số hình ảnh pic_parameter_set_rbsp()	không VCL
35	AUD_NUT	Dấu phân cách đơn vị truy cập access_unit_delimiter_rbsp()	không VCL
36	EOS_NUT	Đuôi của chuỗi end_of_seq_rbsp()	không VCL
37	EOB_NTUT	Đuôi của dòng bit end_of_bitstream_rbsp()	không VCL
38	FD_NUT	Dữ liệu của bộ lọc filler_data_rbsp()	không VCL
39 40	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin nâng cao bổ sung SEI_rbsp()	không VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Dự phòng	không VCL
48..63	UNSPEC48.. UNSPEC83	Không xác định	không VCL

Bảng (4)

Như được thể hiện trong bảng (5), cú pháp tiêu đề của đơn vị NAL có thể bao gồm hai byte dữ liệu, tức 16 bit. Bit thứ nhất là “forbidden_zero_bit” luôn được thiết đặt bằng 0 tại điểm bắt đầu của đơn vị NAL. Sáu bit tiếp theo là “nal_unit_type” xác định phần dữ liệu chuyển của chuỗi byte thô (“RBSP”) được chứa trong đơn vị NAL như được thể hiện trong bảng (4). 6 bit tiếp theo là “nuh_layer_id” xác định sự phân cách của lớp đó. Trong một số trường hợp, sáu bit này có thể được xác định để thay thế là “nuh_reserved_zero_6bits”. nuh_reserved_zero_6bits đó có thể bằng 0 theo tiêu chuẩn thông số kỹ thuật cơ sở. Trong các phần mở rộng mã hóa video mở rộng được và/hoặc các phần mở rộng cú pháp thì nuh_layer_id có thể xác định rằng đơn vị NAL cụ thể này thuộc về lớp được xác định bởi giá trị của 6 bit này. Phần tử cú pháp tiếp theo là “nuh_temporal_id_plus1”. nuh_temporal_id_plus1 - 1 đó có thể xác định nhận diện tạm thời cho đơn vị NAL đó. Nhận diện tạm thời khác nhau TemporalId có thể được xác định là TemporalId = nuh_temporal_id_plus1 - 1. Nhận diện tạm thời TemporalId được sử dụng để xác định lớp phụ tạm thời. Biến HighestTid xác định lớp phụ tạm thời cao nhất được giải mã.

Bảng 5

nal_unit_header() {	Tên
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

Bảng (5)

Bảng (6) trình bày cấu trúc cú pháp tập tham số chuỗi (SPS) làm ví dụ.

pic_width_in_luma_samples xác định rằng chiều rộng của mỗi hình ảnh giải mã trong các bộ luma samples. pic_width_in_luma_samples không bằng 0.

pic_height_in_luma_samples xác định rằng chiều cao mỗi hình ảnh giải mã trong các bộ luma samples. pic_height_in_luma_samples không bằng 0.

sps_max_sub_layers_minus1 + 1 xác định lượng tối đa của các lớp phụ tạm thời có thể có mặt trong mỗi CVS là SPS. Giá trị sps_max_sub_layers_minus1 nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu cuối.

Cờ sps_sub_layer_ordering_info_present_flag bằng 1 xác định rằng các phần tử cú pháp sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i] và sps_max_latency_increase_plus1[i] có mặt đối với các lớp phụ

`sps_max_sub_layers_minus1 + 1`. `sps_sub_layer_ordering_info_present_flag` bằng 0 xác định rằng các giá trị của `sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]`, `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]` và `sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]` áp dụng cho tất cả các lớp phụ.

`sps_max_dec_pic_buffering_minus1[i] + 1` xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS trong các bộ của các bộ đệm lưu trữ hình ảnh khi `HighestTid` bằng `i`. Giá trị `sps_max_dec_pic_buffering_minus1[i]` nằm trong khoảng từ 0 đến `MaxDpbSize - 1`, bao gồm cả giá trị đầu cuối trong đó `MaxDpbSize` xác định kích thước tối đa của bộ đệm hình ảnh giải mã trong các bộ của các bộ đệm lưu trữ hình ảnh. Khi `i` lớn hơn 0, thì `sps_max_dec_pic_buffering_minus1[i]` sẽ lớn hơn hoặc bằng `sps_max_dec_pic_buffering_minus1[i - 1]`. Khi `sps_max_dec_pic_buffering_minus1[i]` không có mặt với `i` nằm trong khoảng từ 0 đến `sps_max_sub_layers_minus1 - 1`, bao gồm cả giá trị đầu cuối, do `sps_sub_layer_ordering_info_present_flag` bằng 0, thì suy ra `sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1]`.

`sps_max_num_reorder_pics[i]` chỉ báo lượng tối đa cho phép của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS theo thứ tự giải mã và tiếp theo hình ảnh đó theo thứ tự kết xuất khi `HighestTid` bằng `i`. Giá trị `sps_max_num_reorder_pics[i]` nằm trong khoảng từ 0 đến `sps_max_dec_pic_buffering_minus1[i]`, bao gồm cả giá trị đầu cuối. Khi `i` lớn hơn 0, thì `sps_max_num_reorder_pics[i]` sẽ lớn hơn hoặc bằng `sps_max_num_reorder_pics[i - 1]`. Khi `sps_max_num_reorder_pics[i]` không có mặt với `i` nằm trong khoảng từ 0 đến `sps_max_sub_layers_minus1 - 1`, bao gồm cả giá trị đầu cuối, do `sps_sub_layer_ordering_info_present_flag` bằng 0, thì suy ra `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]`.

`sps_max_latency_increase_plus1[i]` không bằng 0 được sử dụng để tính toán giá trị `SpsMaxLatencyPictures[i]`, xác định lượng tối đa của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS theo thứ tự kết xuất và tiếp theo hình ảnh đó theo thứ tự giải mã khi `HighestTid` bằng `i`.

Khi `sps_max_latency_increase_plus1[i]` không bằng 0, thì giá trị `SpsMaxLatencyPictures[i]` được xác định như sau:

$\text{sps_MaxLatencyPictures}[i] = \text{sps_max_num_reorder_pics}[i] + \text{sps_max_latency_increase_plus1}[i] - 1$

Khi $\text{sps_max_latency_increase_plus1}[i]$ bằng 0, thì không có giới hạn tương ứng nào được thể hiện.

Giá trị $\text{sps_max_latency_increase_plus1}[i]$ nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu cuối. Khi $\text{sps_max_latency_increase_plus1}[i]$ không có mặt với i nằm trong khoảng từ 0 đến $\text{sps_max_sub_layers_minus1} - 1$, bao gồm cả giá trị đầu cuối, do $\text{sps_sub_layer_ordering_info_present_flag}$ bằng 0, thì suy ra $\text{sps_max_latency_increase_plus1}[\text{sps_max_sub_layers_minus1}]$.

Bảng 6

<code>seq_parameter_set_rbsp0 {</code>
<code>...</code>
<code>sps_max_sub_layers_minus1</code>
<code>...</code>
<code>pic_width_in_luma_samples</code>
<code>pic_height_in_luma_samples</code>
<code>...</code>
<code>for (i = (sps_sub_layer_ordering_info_present_flag ? 0 :</code>
<code> sps_max_sub_layers_minus1);</code>
<code> i <= sps_max_sub_layers_minus1; i++) {</code>
<code> sps_max_dec_pic_buffering_minus1[i]</code>
<code> sps_max_num_reorder_pics[i]</code>
<code> sps_max_latency_increase_plus1[i]</code>
<code> }</code>
<code>...</code>
<code>}</code>

Bảng (6)

Khi ảnh hiện thời là hình ảnh IRAP, quy định sau đây được áp dụng:

- Nếu hình ảnh hiện thời là hình ảnh IDR, hình ảnh BLA, hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã, hoặc hình ảnh thứ nhất tiếp theo đuôi đơn vị NAL của dãy theo thứ tự giải mã, thì biến `NoRaslOutputFlag` được thiết đặt bằng 1.

- Theo cách khác, nếu một số phương tiện bên ngoài là có thể sử dụng được để thiết đặt biến `HandleCraAsBlaFlag` ở giá trị đối với ảnh hiện thời, thì biến `HandleCraAsBlaFlag` được thiết đặt bằng giá trị được tạo ra bởi các phương tiện bên ngoài đó và biến `NoRaslOutputFlag` được thiết đặt bằng `HandleCraAsBlaFlag`.

- Theo cách khác, biến `HandleCraAsBlaFlag` được thiết đặt bằng 0 và biến `NoRaslOutputFlag` được thiết đặt bằng 0.

Nếu hình ảnh hiện thời là hình ảnh IRAP có `NoRaslOutputFlag` bằng 1 không phải là ảnh 0, thì các bước theo thứ tự dưới đây được áp dụng:

1. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm như sau:

- Nếu hình ảnh hiện thời là ảnh CRA, thì NoOutputOfPriorPicsFlag được thiết đặt bằng 1 (không quan tâm đến giá trị no_output_of_prior_pics_flag).

- Theo cách khác, nếu giá trị pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] được suy ra từ SPS hoạt động lần lượt khác với giá trị pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động đối với ảnh đứng trước, thì NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được thiết đặt bằng 1 bằng bộ giải mã trong thử nghiệm, không quan tâm đến giá trị no_output_of_prior_pics_flag.

- Theo cách khác, thì NoOutputOfPriorPicsFlag được thiết đặt bằng no_output_of_prior_pics_flag.

2. Giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HRD như sau:

- Nếu NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh trong DPB được làm rỗng mà không cần kết xuất các hình ảnh chúng chứa và mức đầy DPB được thiết đặt bằng 0.

- Theo cách khác, (NoOutputOfPriorPicsFlag bằng 0), thì tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm rỗng bởi yêu cầu lặp đi lặp lại quy trình “đệm” 1204 và mức đầy DPB được thiết đặt bằng 0.

- Theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1), tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB được giảm bớt một. Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình “đệm” 1204 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt mức đầy DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm rỗng, cho tới khi không có điều kiện nào sau đây là đúng:

1. Số lượng hình ảnh có giá trị nuh_layer_id cụ thể trong DPB được đánh dấu là “cần kết xuất” lớn hơn sps_max_num_reorder_pics[HighestTid] từ tập tham số chuỗi hoạt động (khi

giá trị `nuh_layer_id` cụ thể này bằng 0) hoặc từ tập tham số chuỗi của lớp hoạt động được thiết đặt cho giá trị `nuh_layer_id` cụ thể đó.

2. Nếu `sps_max_latency_increase_plus1[HighestTid]` từ tập tham số chuỗi hoạt động (khi giá trị `nuh_layer_id` cụ thể này bằng 0) hoặc từ tập tham số chuỗi của lớp hoạt động được thiết đặt cho giá trị `nuh_layer_id` cụ thể đó không bằng 0 và có ít nhất một ảnh có giá trị `nuh_layer_id` cụ thể trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan `PicLatencyCount` lớn hơn hoặc bằng `SpsMaxLatencyPictures[HighestTid]` đối với giá trị `nuh_layer_id` cụ thể đó.

3. Số lượng hình ảnh có giá trị `nuh_layer_id` cụ thể trong DPB lớn hơn hoặc bằng `sps_max_dec_pic_buffering[HighestTid] + 1` từ tập tham số chuỗi hoạt động (khi giá trị `nuh_layer_id` cụ thể này bằng 0) hoặc từ tập tham số chuỗi của lớp hoạt động được thiết đặt cho giá trị `nuh_layer_id` cụ thể đó.

Quy trình giải mã hình ảnh trong khối 1206 (giải mã và đánh dấu hình ảnh) xảy ra một cách tức thì khi đơn vị giải mã sau cùng của đơn vị truy cập có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB.

Đối với mỗi hình ảnh có giá trị `nuh_layer_id` bằng giá trị `nuh_layer_id` của hình ảnh hiện thời trong DPB được đánh dấu là “cần kết xuất”, thì biến liên quan `PicLatencyCount` được thiết đặt bằng `PicLatencyCount + 1`.

Hình ảnh hiện thời được coi là đã được giải mã sau khi đơn vị giải mã sau cùng của hình ảnh giải mã. Hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống trong DPB và quy định sau đây được áp dụng:

- Nếu hình ảnh giải mã hiện thời có `PicOutputFlag` bằng 1, thì nó được đánh dấu là “cần kết xuất” và biến liên quan `PicLatencyCount` của nó được thiết đặt bằng 0.

- Theo cách khác, (hình ảnh giải mã hiện thời có `PicOutputFlag` bằng 0), thì nó được đánh dấu là “không cần kết xuất”.

Hình ảnh giải mã hiện thời được đánh dấu là “sử dụng để tham chiếu ngắn hạn”.

Khi một hoặc nhiều điều kiện sau đây là đúng, thì quy trình “đệm” bổ sung 1208 được hủy bỏ lặp đi lặp lại cho tới khi không có điều kiện nào sau đây là đúng:

- Số lượng hình ảnh với giá trị `nuh_layer_id` bằng giá trị `nuh_layer_id` của hình ảnh hiện thời trong DPB mà chúng được đánh dấu là “cần kết xuất” lớn hơn

`sps_max_num_reorder_pics[HighestTid]` từ tập tham số chuỗi hoạt động (khi giá trị `nuh_layer_id` của hình ảnh hiện thời bằng 0) hoặc từ tập tham số chuỗi của lớp hoạt động được thiết lập cho giá trị `nuh_layer_id` của hình ảnh hiện thời.

- `sps_max_latency_increase_plus1[HighestTid]` từ tập tham số chuỗi hoạt động (khi giá trị `nuh_layer_id` của hình ảnh hiện thời bằng 0) hoặc từ tập tham số chuỗi của lớp hoạt động được thiết lập cho giá trị `nuh_layer_id` của hình ảnh hiện thời không bằng 0 và có ít nhất một ảnh có giá trị `nuh_layer_id` cụ thể trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan `PicLatencyCount` lớn hơn hoặc bằng `SpsMaxLatencyPictures[HighestTid]` đối với giá trị `nuh_layer_id` cụ thể đó.

Quy trình “đệm” 1204 và quy trình đệm bổ sung 1208 là giống nhau về các bước và gồm các bước theo thứ tự sau đây: Các hình ảnh mà để kết xuất trước tiên được chọn là những ảnh có giá trị nhỏ nhất của số đếm thứ tự hình ảnh (`PicOrderCntVal`) của tất cả các hình ảnh trong DPB được đánh dấu là “cần kết xuất”. Số đếm thứ tự hình ảnh là biến liên quan đến với mỗi hình ảnh, xác định một cách duy nhất hình ảnh liên quan trong số tất cả các hình ảnh trong CVS và, khi hình ảnh liên quan đó được kết xuất từ bộ đệm hình ảnh giải mã, thì nó chỉ báo vị trí của hình ảnh liên quan theo thứ tự kết xuất liên quan đến các vị trí thứ tự kết xuất của các hình ảnh khác trong cùng CVS mà chúng sẽ được kết xuất từ bộ đệm hình ảnh giải mã.

- Các hình ảnh này được cắt xén, sử dụng của sổ cắt xén phù hợp được xác định trong tập tham số chuỗi hoạt động đối với hình ảnh có `nuh_layer_id` bằng 0 hoặc trong tập tham số chuỗi của lớp hoạt động được thiết lập cho giá trị `nuh_layer_id` bằng giá trị `nuh_layer_id` của hình ảnh, các hình ảnh được cắt xén đó được kết xuất theo thứ tự tăng dần của `nuh_layer_id` và các hình ảnh này được đánh dấu là “không cần kết xuất”.

- Mỗi bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không sử dụng để tham chiếu” và theo đó bao gồm một trong các hình ảnh đã được cắt xén và đầu ra được làm rỗng.

Như được thể hiện trên Fig.13A, do cú pháp tiêu đề của đơn vị NAL đã mô tả trên đây có thể bao gồm hai byte dữ liệu, tức là, 16 bit. Bit thứ nhất là “`forbidden_zero_bit`” luôn được thiết lập bằng 0 tại điểm bắt đầu của đơn vị NAL. Sáu bit tiếp theo là “`nal_unit_type`” xác định phần dữ liệu chuyển của chuỗi byte thô (“RBSP”) chứa trong đơn vị NAL. 6 bit tiếp theo là “`nuh_reserved_zero_6bits`”, `nuh_reserved_zero_6bits` này có thể bằng 0 theo tiêu chuẩn thông số kỹ thuật cơ sở. Các giá trị khác của `nuh_reserved_zero_6bits` có thể được

xác định nếu cần. Các bộ giải mã có thể bỏ qua (nghĩa là, gỡ bỏ hình ảnh khỏi dòng bit và vớt bỏ) tất cả các đơn vị NAL có các giá trị `nuh_reserved_zero_6bits` không bằng 0 khi khi thao tác luồng dựa trên tiêu chuẩn thông số kỹ thuật cơ sở. Trong `nuh_reserved_zero_6bits` mở rộng được hoặc các phần mở rộng khác có thể xác định các giá trị khác, để truyền tín hiệu mã hóa video mở rộng được và/hoặc các phần mở rộng cú pháp. Trong một số trường hợp phần tử cú pháp `nuh_reserved_zero_6bits` có thể được gọi là `reserved_zero_6bits`. Trong một số trường hợp phần tử cú pháp `nuh_reserved_zero_6bits` có thể được gọi là `layer_id_plus1` hoặc `layer_id`, như được thể hiện trên Fig.13B và Fig.13C. Trong trường hợp này, thành phần `layer_id` sẽ là `layer_id_plus1 - 1`. Trong trường hợp này, nó có thể được sử dụng để truyền tín hiệu thông tin liên quan đến video mã hóa mở rộng được. Phần tử cú pháp tiếp theo là “`nuh_temporal_id_plus1`”. `nuh_temporal_id_plus1 - 1` có thể xác định nhận diện tạm thời cho đơn vị NAL đó. Nhận diện tạm thời khác nhau `TemporalId` có thể được xác định là `TemporalId = nuh_temporal_id_plus1 - 1`.

Như được thể hiện trên Fig.14, cấu trúc cú pháp của đơn vị NAL chung được minh họa. Cú pháp hai byte của tiêu đề của đơn vị NAL trên Fig.13 được bao gồm có tham khảo đến `NAL_unit_header()` trên Fig.14. Phần dư của cú pháp đơn vị NAL chủ yếu liên quan đến RBSP.

Một kỹ thuật hiện có để sử dụng “`nuh_reserved_zero_6bits`” là truyền thông tin tín hiệu mã hóa video mở rộng được bằng cách phân chia 6 bit của `nuh_reserved_zero_6bits` thành các trường bit riêng biệt, gọi là, một hoặc nhiều trong số ID phụ thuộc, ID chất lượng, ID cảnh và cờ độ sâu, mỗi loại chỉ sự nhận diện lớp khác nhau của video đã mã hóa mở rộng được. Do đó 6 bit đó chỉ báo đơn vị NAL cụ thể này thuộc lớp nào của kỹ thuật mã hóa mở rộng được. Sau đó phần dữ liệu chuyển, chẳng hạn như cú pháp mở rộng của tập hợp tham số video (“VPS”) (“`scalability_type`”) như được thể hiện trên Fig.15, thông tin về lớp được xác định. Cú pháp mở rộng VPS trên Fig.15 bao gồm 4 bit cho loại mở rộng được (phần tử cú pháp `scalability_type`) xác định các loại mở rộng được đang được sử dụng trong chuỗi video mã hóa và kích thước được truyền tín hiệu thông qua `layer_id_plus1` (hoặc `layer_id`) trong tiêu đề của đơn vị NAL. Khi loại có thể mở rộng bằng 0, chuỗi video mã hóa phù hợp với chuẩn cơ sở, do vậy `layer_id` của tất cả các đơn vị NAL bằng 0 và không có đơn vị NAL nào thuộc lớp hoặc cảnh nâng cao hoặc. Các giá trị cao hơn của loại mở rộng sẽ được biên dịch như được thể hiện trên Fig.16.

$layer_id_dim_len[i]$ xác định độ dài, theo bit, của ID kích thước có thể mở rộng thứ i . Tổng của các giá trị $layer_id_dim_len[i]$ cho mọi giá trị i nằm trong khoảng từ 0 đến 7 sẽ nhỏ hơn hoặc bằng 6. $vps_extension_byte_alignment_reserved_zero_bit$ là 0. $vps_layer_id[i]$ xác định giá trị của $layer_id$ của lớp thứ i mà thông tin phụ thuộc của lớp tiếp theo thích hợp với nó. $num_direct_ref_layers[i]$ xác định số lượng lớp mà lớp thứ i phụ thuộc trực tiếp vào chúng. $ref_layer_id[i][j]$ xác định lớp thứ j mà lớp thứ i phụ thuộc trực tiếp vào nó.

Theo cách này, kỹ thuật hiện có truyền tín hiệu đến các bộ nhận diện mở rộng được trong đơn vị NAL và trong tập tham số video để phân bổ các bit trong số các loại có khả năng mở rộng liệt kê trên Fig.16. Sau đó, đối với mỗi loại có khả năng mở rộng, Fig.16 xác định bao nhiêu kích thước được hỗ trợ. Ví dụ, loại có khả năng mở rộng 1 có 2 kích thước (đó là không gian và chất lượng). Đối với mỗi kích thước, $layer_id_dim_len[i]$ xác định số lượng bit được cấp phát cho mỗi trong hai kích thước này, trong đó tổng + tất cả các giá trị của $layer_id_dim_len[i]$ nhỏ hơn hoặc bằng 6, là số lượng bit trong $nuh_reserved_zero_6bits$ của tiêu đề của đơn vị NAL. Do vậy, để kết hợp kỹ thuật này nhận diện các loại mở rộng nào đang được sử dụng và 6 bit của tiêu đề của đơn vị NAL được cấp phát như thế nào trong số khả năng mở rộng.

Mặc dù kết hợp cố định như vậy của các kích thước có khả năng mở rộng khác nhau, như được thể hiện trên Fig.16, là phù hợp cho nhiều ứng dụng còn có các kết hợp được mong muốn mà chúng chưa được bao gồm. Trên Fig.17, cú pháp mở rộng tập tham số video được biến đổi xác định loại có khả năng mở rộng cho mỗi bit trong phần tử cú pháp $nuh_reserved_zero_6bits$, $vps_extension_byte_alignment_reserved_zero_bit$ được thiết đặt bằng 0. $Max_num_layers_minus1_bits$ chỉ báo tổng số lượng bit được sử dụng cho phần tử cú pháp trong hai byte đầu tiên của tiêu đề của đơn vị NAL trên Fig.13 được gọi là $layer_id_plus1$ hoặc $nuh_reserved_zero_6bits$. $scalability_map[i]$ xác định loại có khả năng mở rộng cho mỗi bit trong phần tử cú pháp $layer_id_plus1$. Trong một số trường hợp, phần tử cú pháp $layer_id_plus1$ có thể còn được gọi là phần tử cú pháp $nuh_reserved_zero_6bits$ hoặc $rreserved_zero_6bits$, Ánh xạ khả năng mở rộng đối với tất cả các bit của phần tử cú pháp $layer_id_plus1$ cùng xác định khả năng mở rộng đang được sử dụng trong chuỗi video mã hóa. Giá trị thực của nhận diện đối với mỗi loại khả năng mở rộng được truyền tín hiệu thông qua các bit tương ứng này trong trường $layer_id_plus1$ ($nuh_reserved_zero_6bits$) trong tiêu đề của đơn vị NAL. Khi $scalability_map[i]$ bằng 0 cho mọi giá trị i , thì chuỗi

video mã hóa phù hợp với chuẩn cơ sở, do vậy giá trị `layer_id_plus1` của các đơn vị NAL bằng 0 và không có đơn vị NAL nào thuộc lớp hoặc cảnh nâng cao. `vps_layer_id[i]` xác định giá trị `layer_id` của lớp thứ `i` mà thông tin phụ thuộc của lớp tiếp theo thích hợp với nó. `num_direct_ref_layers[i]` xác định số lượng lớp mà lớp thứ `i` phụ thuộc trực tiếp vào chúng. `ref_layer_id[i][j]` xác định lớp thứ `j` mà lớp thứ `i` phụ thuộc trực tiếp vào nó.

Các giá trị cao hơn của `scalability_map[i]` được biên dịch như được thể hiện trên Fig.18. Ánh xạ khả năng mở rộng `[i]` bao gồm kích thước có khả năng mở rộng (0) không có gì; (1) không gian; (2) chất lượng; (3) độ sâu; (4) đa cảnh; (5) không xác định; (6) dự phòng; và (7) dự phòng.

Do đó mỗi bit trong tiêu đề của đơn vị NAL được diễn giải dựa trên 3 bit trong tập tham số video về kích thước khả năng mở rộng là gì (ví dụ, không có gì, không gian, chất lượng, độ sâu, đa cảnh, không xác định, dự phòng). Ví dụ, để truyền tín hiệu rằng tất cả các bit trong `layer_id_plus1` tương ứng với khả năng mở rộng không gian, các giá trị `scalability_map` trong VPS có thể đã giải mã là 001 001 001 001 001 001 cho 6 bit của tiêu đề của đơn vị NAL. Ngoài ra ví dụ, để truyền tín hiệu rằng 3 bit trong `layer_id_plus1` tương ứng với khả năng mở rộng không gian và 3 bit tương ứng với khả năng mở rộng chất lượng, các giá trị `scalability_map` trong VPS có thể đã giải mã là 001 001 001 010 010 010 cho 6 bit của tiêu đề của đơn vị NAL.

Trên Fig.19, phương án khác bao gồm tập tham số video truyền tín hiệu lượng kích thước có khả năng mở rộng trong 6 bit của tiêu đề của đơn vị NAL sử dụng `num_scalability_dimensions_minus1`. `num_scalability_dimensions_minus1` cộng 1 chỉ báo lượng kích thước có khả năng mở rộng được truyền tín hiệu thông qua các phần tử cú pháp `layer_id_plus1`; `nuh_reserved_zero_6bits`; và/hoặc `reserved_zero_6bits`. `scalability_map[i]` có cùng ngữ nghĩa như được mô tả trên đây tham chiếu đến Fig.17. `num_bits_for_scalability_map[i]` xác định độ dài theo bit đối với kích thước có khả năng mở rộng thứ `i`. Tổng của tất cả `num_bits_for_scalability_map[i]` đối với $i=0$, `num_scalability_dimensions_minus1` bằng sáu (hoặc theo cách khác, bằng lượng bit được sử dụng đối với các phần tử cú pháp `layer_id_plus1`; `vps_reserved_zero_6bits`; `max_num_layers_minus1`; `reserved_zero_6bits`; `nuh_reserved_zero_6bits`).

Như được thể hiện trên Fig.17 và Fig.19, các thay đổi có thể được sử dụng, nếu muốn. Ví dụ theo một phương án, `scalability_map[i]` có thể được truyền tín hiệu với $u(4)$ (hoặc $u(n)$ với $n>3$ hoặc $n<3$). Trong trường hợp này các giá trị cao hơn của

scalability_map[i] có thể được xác định là dự phòng cho các dòng bit phù hợp với biên dạng cụ thể của kỹ thuật video. Ví dụ, ví dụ các giá trị ánh xạ khả năng mở rộng 6..15 có thể được xác định là “dự phòng” khi truyền tín hiệu scalability_map[i] với u(4). Ví dụ Theo cách khác, scalability_map[i] có thể được truyền tín hiệu với ue(v) hoặc sơ đồ mã hóa khác nào đó. Ví dụ Theo cách khác, ràng buộc có thể được xác định sao cho các giá trị scalability_map[i] được trang bị theo thứ tự đơn điệu không giảm (hoặc không tăng). Việc này tạo ra một số trường kích thước có khả năng mở rộng trong trường layer_id_plus1 trong tiêu đề của đơn vị NAL có liên quan.

Một kỹ thuật hiện có khác đối với truyền tín hiệu mã hóa video mở rộng được sử dụng phần tử cú pháp “layer_id_plus1” hoặc “nuh_reserved_zero_6bits” ánh xạ layer_id_plus1 trong tiêu đề của đơn vị NAL đến nhận diện lớp bằng cách truyền tín hiệu bằng dò tìm chung trong tập tham số video. Như được thể hiện trên Fig.20, kỹ thuật hiện có này bao gồm tập hợp tham số video để xác định số lượng loại kích thước và các nhận diện kích thước đối với lớp thứ i của bảng dò tìm đó. Cụ thể, vps_extension_byte_alignment_reserved_zero_bit là 0. num_dimensions_minus1[i] + 1 xác định số lượng loại kích thước (dimension_type[i][j]) và các nhận diện kích thước (dimension_id[i][j]) đối với lớp thứ i. dimension_type[i][j] xác định loại kích thước có khả năng mở rộng thứ j của lớp thứ i, có layer_id hoặc layer_id_plus1 bằng i, như được xác định trên Fig.31. Như được thể hiện trên Fig.21, các kích thước này được xác định bao gồm (0) idx thứ tự hình; (1) cỡ độ sâu; (2) ID phụ thuộc; (3) ID chất lượng; (4)-(15) dự phòng. dimension_id[i][j] xác định nhận diện loại kích thước có khả năng mở rộng thứ j của lớp thứ i, khi không có mặt thì suy ra 0. num_direct_ref_layers[i] xác định số lượng lớp mà lớp thứ i phụ thuộc trực tiếp vào chúng. ref_layer_id[i][j] xác định lớp thứ j mà lớp thứ i phụ thuộc trực tiếp vào nó. Tiếc là phương án dự kiến được thể hiện trên Fig.20 tạo ra bảng tìm kiếm lớn không nhìn thấy được.

Như được thể hiện trên Fig.22, một mở rộng tập tham số video được biến đổi bao gồm mặt nạ có khả năng mở rộng được sử dụng kết hợp với kích thước có khả năng mở rộng. scalability_mask truyền tín hiệu sơ đồ gồm 0 và 1 bit với mỗi bit tương ứng với một kích thước có khả năng mở rộng như được chỉ báo bởi cú pháp ánh xạ khả năng mở rộng trên Fig.23. Giá trị 1 đối với kích thước có khả năng mở rộng cụ thể chỉ báo rằng kích thước có khả năng mở rộng này có mặt trong lớp này (lớp thứ I). Giá trị 0 đối với kích thước có khả năng mở rộng cụ thể chỉ báo rằng kích thước có khả năng mở rộng này không có mặt

trong lớp này (lớp thứ I). Ví dụ, tập hợp các bit 00100000 liên quan đến khả năng mở rộng chất lượng. Giá trị nhận diện thực tế của kích thước có khả năng mở rộng cụ thể có mặt được chỉ báo bởi giá trị `scalability_id[j]` được truyền tín hiệu. Các giá trị `num_scalability_types[i]` bằng tổng số lượng bit trong `scalability_mask` có giá trị bằng 1. Do vậy

Biểu thức 9

$$\text{num_scalability_types}[i] = \sum_{k=0}^7 \text{scalability_mask}[i](k)$$

`scalability_id[j]` chỉ báo giá trị nhận diện của kích thước có khả năng mở rộng thứ j đối với loại của các giá trị của khả năng mở rộng mà chúng được truyền tín hiệu nhờ giá trị `scalability_mask`.

Như được thể hiện trên Fig.24, một cải biến của Fig.22, bao gồm mặt nạ có khả năng mở rộng được truyền tín hiệu ngoài vòng lặp. Việc này tạo ra một mặt nạ chung cho mỗi lần nhận diện lớp. Như được thể hiện trên Fig.25, trong cải biến này tập tham số video tương ứng làm ví dụ có thể bao gồm sự nhận diện mở rộng được với mặt nạ có khả năng mở rộng hiện không được bao gồm. Trong trường hợp này, phần tử cú pháp `scalable_id[j]` có cùng biểu hiện là phần tử cú pháp `scalability_id[j]` trên Fig.22.

Như được thể hiện trên Fig.26, một cải biến của Fig.22 bao gồm mặt nạ có khả năng mở rộng (`scalability_mask`) được truyền tín hiệu ngoài vòng lặp. Việc này tạo ra một mặt nạ chung cho mỗi lần nhận diện lớp. `scalability_mask` truyền tín hiệu sơ đồ gồm 0 và 1 bit với mỗi bit tương ứng với một kích thước có khả năng mở rộng như được chỉ ra bởi cú pháp ánh xạ khả năng mở rộng trên Fig.27. Giá trị 1 đối với kích thước có khả năng mở rộng cụ thể chỉ báo rằng kích thước có khả năng mở rộng này có mặt trong lớp này (lớp thứ I). Giá trị 0 đối với kích thước có khả năng mở rộng cụ thể chỉ báo rằng kích thước có khả năng mở rộng này không có mặt trong lớp này (lớp thứ I). Ví dụ, tập hợp các bit 00100000 liên quan đến khả năng mở rộng chất lượng. Giá trị nhận diện thực tế của kích thước có khả năng mở rộng cụ thể có mặt được chỉ báo bằng giá trị `scalability_id[j]` được truyền tín hiệu. Các giá trị của `num_scalability_types[i]` bằng tổng số lượng bit trong `scalability_mask` có giá trị bằng 1. Do vậy

Biểu thức 10

$$\text{NumScalabilityTypes}[i] = \sum_{k=0}^{15} \text{scalability_mask}(k)$$

Trong trường hợp này biến `scalability_id[j]` còn được gọi là biến `dimension_id[i][j]`. `dimension_id[i][j]` xác định nhận diện của khả năng mở rộng của kích thước có khả năng mở rộng thứ `j` của lớp thứ `i`. Sau đó biến `ScalabilityId[i][j]` được suy ra như sau.

Bảng 7

for(i = 1; i <= vps_max_layers_minus1; i++) {
 for(k=0, j=0; k<=15; k++) {
 if(scalability_mask(k)==1)
 ScalabilityId [i][k]=dimension_id[i][j++]
 else
 ScalabilityId [i][k]=0;
 }
}

Trong đó `ScalabilityId [i][k]` truyền tín hiệu ID kích thước cho loại có khả năng mở rộng tương ứng sau đây.

Bảng 8

k	ScalabilityId [i][k]
0	DependencyId[i][k]
1	QualityId[i][k]
2	depthFlag[i][k]
3	ViewId[i][k]
4-15	Dự trữ

Trong đó `DependencyId[i][1]` là ID phụ thuộc đối với kích thước khả năng mở rộng không gian cho lớp thứ `i`, `QualityId[i][2]` là ID chất lượng đối với kích thước khả năng mở rộng chất lượng đối với lớp thứ `i`, `depthFlag[i][3]` là cờ độ sâu/ ID độ sâu đối với kích thước khả năng mở rộng độ sâu cho lớp thứ `i` và `ViewId[i][4]` là ID cảnh đối với kích thước khả năng mở rộng đa cảnh cho lớp thứ `i`.

Cũng như được thể hiện trên Fig.26, `avc_base_codec_flag` bằng 1 xác định rằng lớp cơ sở thứ nhất phù hợp với Rec. ITU-T H.264 | ISO/IEC 14496-10 và `avc_base_codec_flag` bằng 1 xác định tới HEVC. `Vps_nuh_layer_id_presnet_flag` chỉ báo nếu biến `layer_id_in_nuh[i]` truyền tín hiệu giá trị `layer_id` trong tiêu đề của đơn vị NAL được truyền tín hiệu.

Theo cách khác, một hoặc nhiều phần tử cú pháp `scalability_mask[i]`, `scalability_mask`, `scalability_id[j]` có thể được truyền tín hiệu sử dụng số lượng bit khác với `u(8)`. Ví dụ chúng có thể được truyền tín hiệu với `u(16)` (hoặc `u(n)` với `n>8` hoặc `n<8`). Theo cách khác, một hoặc nhiều phần tử cú pháp có thể được truyền tín hiệu với `ue(v)`. Theo cách

khác, scalability_mask có thể được truyền tín hiệu trong tiêu đề của đơn vị NAL trong các phần tử cú pháp layer_id_plus1; vps_reserved_zero_6bits; max_num_layers_minus1; reserved_zero_6bits; và/hoặc nuh_reserved_zero_6bits, theo một số phương án, hệ thống này chỉ có thể thực hiện điều này cho các đơn vị NAL VPS, hoặc chỉ cho các đơn vị NAL không VPS, hoặc cho tất cả các đơn vị NAL. Theo phương án khác nữa, scalability_mask có thể được truyền tín hiệu mỗi ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ có thể được truyền tín hiệu ở phần đầu lát, tập tham số hình ảnh, tập tham số video, hoặc bất kỳ tập tham số khác hoặc bất kỳ phần danh định nào của dòng bit.

Cần lưu ý rằng các hình vẽ Fig.13, Fig.15, Fig.18, Fig.20, Fig.21, Fig.22, Fig.23 và các mô tả tương ứng có liên quan đến 6 bit do phần tử cú pháp nuh_reserved_zero_6bits hoặc layer_id_plus1 trong tiêu đề của đơn vị NAL trên Fig.13 có 6 bit. Tuy nhiên tất cả các mô tả nêu trên có thể được biến đổi phù hợp nếu phần tử cú pháp đó sử dụng lượng bit khác với 6 bit. Ví dụ nếu phần tử cú pháp đó (nuh_reserved_zero_6bits hoặc layer_id_plus1) sử dụng 9 bit, thì sau đó trên Fig.17, giá trị bit max_num_layer_minus1 sẽ là 9 và scalability_map[i] sẽ được truyền tín hiệu cho mỗi 9 bit thay vì 6 bit.

Như được thể hiện trên Fig.24 một cải biến của Fig.22 đưa ra cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Phần tử cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp độc lập. Giá trị 1 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp phụ thuộc phụ thuộc vào một hoặc nhiều lớp khác.

Các giá trị NumDepLayers bằng tổng số lượng bit trong layer_dependency_information_pattern có giá trị bằng 1. Do vậy

Biểu thức 11

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

Như được thể hiện trên Fig.29, một cải biến của Fig.26 đưa ra cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Phần tử cú pháp mới layer_dependency_flag[i] được xác định. layer_dependency_flag[i] truyền tín hiệu nếu một lớp phụ thuộc vào các lớp khác. Giá trị 0

đối với cờ chỉ báo rằng lớp có layer_id i là lớp độc lập. Giá trị 1 cho bit thứ i chỉ báo rằng lớp có layer_id i là lớp phụ thuộc.

Như được thể hiện trên Fig.30 một cải biến của Fig.26 đưa ra cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Phần tử cú pháp mới layer_dependency_map[i] được xác định. layer_dependency_map[i] truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i không phụ thuộc vào lớp có layer_id (k+1). Giá trị 1 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i phụ thuộc vào lớp có layer_id (k+1).

Như được thể hiện trên Fig.31, một cải biến của Fig.26 đưa ra cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Phần tử cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp độc lập. Giá trị 1 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp phụ thuộc phụ thuộc vào một hoặc nhiều lớp khác. Các giá trị của NumDepLayers bằng tổng số lượng bit trong layer_dependency_information_pattern có giá trị bằng 1. Do vậy:

Biểu thức 12

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

layer_dependency_map[i] truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i không phụ thuộc vào lớp có layer_id (k+1). Giá trị 1 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i phụ thuộc vào lớp có layer_id (k+1).

Như được thể hiện trên Fig.32, một cải biến của Fig.26 đưa ra cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Fig.28 là cú pháp biến đổi dựa trên cú pháp trên Fig.27. Phần tử cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp độc lập. Giá trị 1 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp phụ thuộc phụ thuộc vào một hoặc nhiều lớp khác.

Các giá trị của NumDepLayers bằng tổng số lượng bit trong layer_dependency_information_pattern có giá trị bằng 1. Do vậy

Biểu thức 13

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

Các phần tử cú pháp num_direct_ref_layers[i] và ref_layer_id[i][j] được truyền tín hiệu chỉ khi layer_dependency_information_pattern(i) có giá trị 1. Trong đó layer_dependency_information_pattern(i) là bit thứ i của phần tử cú pháp layer_dependency_pattern.

Như được thể hiện trên Fig.33, một cải biến của Fig.26 là cú pháp để truyền tín hiệu thông tin phụ thuộc lớp. Fig.29 là cú pháp biến đổi dựa trên cú pháp trên Fig.31. Phần tử cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp độc lập. Giá trị 1 cho bit thứ i chỉ báo rằng lớp có layer_id (i+1) là lớp phụ thuộc phụ thuộc vào một hoặc nhiều lớp khác.

Các giá trị của NumDepLayers bằng tổng số lượng bit trong layer_dependency_information_pattern có giá trị bằng 1. Do vậy

Biểu thức 14

$$\text{NumDepLayers} = \sum_{k=0}^{\text{vps_max_layer_minus1}-1} \text{layer_dependency_information_pattern}(k)$$

layer_dependency_map[i] truyền tín hiệu sơ đồ gồm 0 và 1 bit có độ dài là vps_max_layers_minus1. Giá trị 0 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i không phụ thuộc vào lớp có layer_id (k+1). Giá trị 1 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i phụ thuộc vào lớp có layer_id (k+1). Các phần tử cú pháp layer_dependency_map[i] được truyền tín hiệu chỉ khi layer_dependency_information_pattern(i) có giá trị 1. Trong đó layer_dependency_information_pattern(i) là bit thứ i của phần tử cú pháp layer_dependency_pattern.

Theo cách khác, phần tử cú pháp `layer_dependency_information_pattern` có thể được truyền tín hiệu dưới dạng tập hợp của các giá trị cờ 1 bit. Trong trường hợp này tổng các giá trị 1 bit `vps_max_layers_minus1` sẽ được truyền tín hiệu là:

Bảng 9

<code>for(i = 1; i <= vps_max_layers_minus1 ; i++)</code>
<code>{</code>
<code>layer_dependency_information_pattern_flags[i];</code>
<code>}</code>

Theo cách khác, `layer_dependency_map[i]` phần tử cú pháp có thể được truyền tín hiệu dưới dạng tập hợp của các giá trị cờ 1 bit. Trong trường hợp này, tổng các giá trị 1 bit `vps_max_layers_minus1` sẽ được truyền tín hiệu là:

Bảng 10

<code>for(j = 1; j <= vps_max_layers_minus1 ; j++)</code>
<code>{</code>
<code>layer_dependency_map_values[i][j];</code>
<code>}</code>

Theo cách khác, một hoặc nhiều phần tử cú pháp `layer_dependency_information_pattern`, `layer_dependency_map` có thể được truyền tín hiệu sử dụng số lượng bit cố định đã biết thay cho `u(v)`. Ví dụ chúng có thể được truyền tín hiệu sử dụng `u(64)`.

Theo cách khác, một hoặc nhiều trong số hoặc nhiều phần tử cú pháp `layer_dependency_information_pattern`, `layer_dependency_map` có thể được truyền tín hiệu với `ue(v)` hoặc một số sơ đồ mã hóa khác.

Theo cách khác, tên của các phần tử cú pháp khác nhau và ngữ nghĩa của chúng có thể được sửa đổi bằng cách bổ sung `plus1` hoặc `plus2` hoặc bằng cách trừ bớt `minus1` hoặc `minus2` so với cú pháp và ngữ nghĩa đã được mô tả.

Theo phương án khác nữa, các phần tử cú pháp khác như `layer_dependency_information_pattern`, `layer_dependency_map`, `layer_dependency_flag[i]` v.v. có thể được truyền tín hiệu mỗi ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ nó có thể được truyền tín hiệu ở phần đầu lát, `pps/ sps/ vps/ aps` hoặc tập tham số bất kỳ khác hoặc phần danh định khác của dòng bit.

Như đã mô tả ở phần trên, mã hóa video mở rộng được là kỹ thuật mã hóa dòng bit video cũng chứa một hoặc nhiều dòng bit của tập hợp phụ. Dòng bit video tập hợp phụ có thể được suy ra là cách đẹp bỏ các bộ từ video lớn để giảm bớt băng tần cần thiết đối với dòng bit tập hợp phụ. Dòng bit tập hợp phụ có thể đại diện cho độ phân giải không gian thấp (màn hình nhỏ), độ phân giải tạm thời thấp (tỷ lệ khung hình thấp), hoặc tín hiệu video chất lượng thấp. Ví dụ, dòng bit video có thể bao gồm 5 dòng bit của tập hợp phụ, trong đó mỗi dòng bit của tập hợp phụ bổ sung thêm nội dung cho dòng bit cơ sở. Hannuksela và các đồng tác giả, “Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)” JCTVC-L0453, Shanghai, October 2012, được kết hợp ở đây bằng cách tham chiếu toàn bộ trong bản mô tả này. Chen và các đồng tác giả, “SHVC Draft Text 1”, JCTVC-L1008, Geneva, March, 2013, toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn. Wang và các đồng tác giả, “AHG9: On VPS and SPS in HEVC 3DV and scalable extensions”, JCTVC-M0268, Incheon, April 2013 toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn.

Như đã mô tả ở phần trên, mã hóa video đa cảnh là kỹ thuật mã hóa dòng bit video cũng chứa một hoặc nhiều dòng bit khác đại diện cho các hình thay thế. Ví dụ, đa cảnh có thể là cặp hình đối với video ba chiều. Ví dụ, đa cảnh có thể đại diện cho đa cảnh của cùng một cảnh từ nhiều điểm quan sát khác nhau. Đa cảnh nhìn chung có chứa lượng lớn các phụ thuộc thống kê liên hình, do các hình ảnh là của cùng một cảnh từ nhiều điểm quan sát khác nhau. Do đó, việc sự báo kết hợp tạm thời và liên hình có thể đạt được sự mã hóa đa cảnh hiệu quả. Ví dụ, khung có thể được dự báo hiệu quả không chỉ từ các khung liên quan một cách tạm thời, mà còn từ các khung của các điểm quan sát ngoài. Hannuksela và các đồng tác giả, “Common specification text for scalable and multi-view extensions”, JCTVC-L0452, Geneva, January 2013, toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn. Tech, et. al. “MV-HEVC Draft Text 3 (ISO/IEC 23008-2:201x/PDAM2)”, JCT3V-C1004_d3, Geneva, January 2013, toàn bộ nội dung của tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn.

Chen và các đồng tác giả, “SHVC Draft Text 1”, JCTVC-L1008, Geneva, January 2013; Hannuksela và các đồng tác giả “Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)”, JCTVC-L0453-spec-text, Shanghai, October 2012; và Hannuksela, “Draft Text for Multiview Extension of High Efficiency Video Coding (HEVC)”, JCTVC-L0452-spec-text-r1, Shanghai, October 2012; toàn bộ nội dung của mỗi

tài liệu này được kết hợp trong bản mô tả này bằng cách viện dẫn. JCTVC-L0452 và JCTVC-L0453 mỗi tài liệu đều có bối cảnh về điểm hoạt động đầu ra. Bối cảnh này sau đó được đổi thành bối cảnh các tập hợp lớp kết xuất trong JCTVC-L1008. Đối với các phần tử cú pháp `num_output_operation_points`, `output_op_point_index[]` và `output_layer_flag[][]` được xác định đối với các điểm hoạt động đầu ra đối với JCTVC-L0453 và LCTVC-L0452 và được biến đổi thành `num_output_layer_sets`, `output_layer_set_idx[]` và `output_layer_flag[][]` trong JCTVC-L1008. Trong mỗi tài liệu đó, thông báo SEI thay đổi phụ thuộc lớp được xác định. Thông báo SEI này cho phép truyền tín hiệu các thay đổi về thông tin phụ thuộc lớp bắt đầu với đơn vị truy cập hiện thời có chứa thông báo SEI.

Chen và các đồng tác giả, “SHVC Draft Text 1”, JCTVC-L1008, Geneva, January 2013; Hannuksela và các đồng tác giả “Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)”, JCTVC-L0453-spec-text, Shanghai, October 2012; và Hannuksela, “Draft Text for Multiview Extension of High Efficiency Video Coding (HEVC)”, JCTVC-L0452-spec-text-r1, Shanghai, October 2012; toàn bộ nội dung của mỗi tài liệu được kết hợp trong bản mô tả này bằng cách viện dẫn, mỗi tài liệu đều bộ đệm hình ảnh giải mã (DPB) của lệnh kết xuất hoạt động dựa trên việc sử dụng các phần tử cú pháp `sps_max_num_reorder_pics[HighestTid]`, `sps_max_latency_increase_plus1[HighestTid]` và `sps_max_dec_pic_buffering[HighestTid]` cho kết xuất và gỡ bỏ các hình ảnh 0 từ DPB. Thông tin này được truyền tín hiệu trong tập tham số video cho lớp cơ sở, cung cấp tạo ra thông tin đệm cho nội dung video bao gồm các lớp nâng cao, nếu có.

Như được thể hiện trên Fig.34, ví dụ khác về tập hợp tham số video được minh họa. Cụ thể, tập tham số video trên Fig.34 bao gồm là phần mở rộng của tập tham số video được liên kết (`VPS_extension`). `vps_extension` có thể được truyền tín hiệu với `vps_extension_flag`, `vps_extension2_flag` và/hoặc `vps_extension_data_flag`.

Như được thể hiện trên Fig.35, ví dụ khác về phần mở rộng của tập tham số video được minh họa. Cụ thể, phần mở rộng của tập tham số video trên Fig.35 bao gồm là `num_output_layer_sets`, xác định những lớp của tập hợp các lớp của dòng bit có thể là đầu ra cho người xem nhờ bộ giải mã. `num_output_layer_sets` xác định số lượng tập hợp lớp, mà đối với nó, các lớp đầu ra được xác định có `output_layer_set_index[i]` và `output_layer_flag[lsIdx][j]`. Khi không có mặt, giá trị `num_output_layer_sets` được suy ra bằng 0. Tập hợp lớp mô tả các lớp đầu ra là tập hợp lớp đầu ra.

`output_layer_set_idx[i]` xác định chỉ số `lsIdx` của tập hợp lớp, mà đối với nó, `output_layer_flag[lsIdx][j]` là có mặt.

`output_layer_flag[lsIdx][j]` bằng 1 xác định rằng lớp có `nuh_layer_id` bằng `j` là lớp đầu ra đích của tập hợp lớp thứ `lsIdx`. Giá trị `output_layer_flag[lsIdx][j]` bằng 0 xác định rằng lớp có `nuh_layer_id` bằng `j` không phải là lớp đầu ra đích của tập hợp lớp thứ `lsIdx`.

Với `num_output_layer_sets` được xác định trong `vps_extension` theo cách thức đã được mô tả, `num_output_layer_sets` có thể được cập nhật bằng cách gửi tập tham số video mới đi kèm với phần mở rộng của tập tham số video tương ứng. Tiếp là, việc gửi tập tham số video mới đi kèm với phần mở rộng của tập tham số video tương ứng gây ra việc chuyển giảm mạnh về băng tần khả dụng. Ngoài ra tập tham số video mới có thể chỉ được kích hoạt tại một số loại ảnh nhất định, ví dụ tại các loại ảnh điểm truy nhập ngẫu nhiên trong.

Ví dụ, dòng bit có thể bao gồm lớp cơ sở 0 và bốn lớp nâng cao, gọi là lớp nâng cao 1, lớp nâng cao 2, lớp nâng cao 3 và lớp nâng cao 4. Tập hợp lớp thứ nhất có thể là lớp cơ sở 0 và lớp nâng cao 1. Tập hợp lớp thứ hai có thể là lớp cơ sở 0, lớp nâng cao 1 và lớp nâng cao 2. Tập hợp lớp thứ ba có thể là lớp cơ sở 0, lớp nâng cao 1 và lớp nâng cao 3. Tập hợp lớp thứ tư có thể là lớp cơ sở 0 và lớp nâng cao 4. Các tập hợp lớp kết xuất xác định các lớp cụ thể của tập hợp lớp có thể được tạo ra làm đầu ra. Ví dụ, tập hợp lớp kết xuất 1 có thể là lớp 0 là đầu ra và lớp 1 không phải là đầu ra. Ví dụ, tập hợp lớp kết xuất 2 có thể là lớp 0 là đầu ra, lớp 1 là đầu ra và lớp 2 không phải là đầu ra. Ví dụ, tập hợp lớp kết xuất 3 có thể là lớp 0 không phải là đầu ra, lớp 1 không phải là đầu ra và lớp 3 là đầu ra. Ví dụ, tập hợp lớp kết xuất 4 có thể là lớp 0 là đầu ra và lớp 4 không phải là đầu ra. Lượng giới hạn các lớp kết xuất có ích để hỗ trợ các khả năng giải mã giới hạn. Trong một số trường hợp, có mong muốn vô hiệu hóa một số sự phụ thuộc giữa các lớp khác nhau để hỗ trợ việc giải mã các lớp khác và/hoặc các khả năng của bộ giải mã. Khi một số phụ thuộc lớp thay đổi thì có thể sẽ mong muốn làm cho có thể, vô hiệu hóa và/hoặc bổ sung các lớp khác làm các lớp kết xuất. Ví dụ sự thay đổi phụ thuộc lớp có thể cần một số ít lớp phải được giải mã để giải mã lớp đích cụ thể (ví dụ hình trong MV-HEVC). Do vậy bộ giải mã có thể giải mã và và kết xuất lớp (ví dụ và hình bổ sung) sau khi thay đổi phụ thuộc lớp mà trước đây chưa thể do giới hạn phần cứng của bộ giải mã. Hiện nay vẫn chưa có thiết bị cho phép truyền tín hiệu thay đổi trong các tập hợp lớp kết xuất mà không phải gửi tập tham số video mới.

Như được thể hiện trên Fig.36, cú pháp thông báo SEI có thể được sử dụng để hỗ trợ các thay đổi của các tập hợp lớp kết xuất, gọi là, `output_layer_sets_change(payloadSize)`.

Thông báo SEI này chỉ báo rằng các thay đổi thông tin của các tập hợp lớp kết xuất bắt đầu với đơn vị truy cập hiện thời có chứa thông báo SEI và được diễn giải liên quan đến tập tham số video hoạt động. Hơn một tập tham số video có thể được bao gồm với dòng bit và tập tham số video hoạt động được thay đổi, nếu cần. Khi có mặt, thông báo SEI thay đổi của các tập hợp lớp kết xuất áp dụng cho đơn vị truy cập đích bao gồm đơn vị truy cập hiện thời và tất cả các đơn vị truy cập theo sau, theo thứ tự giải mã, cho tới khi có thông báo SEI thay đổi của các tập hợp lớp kết xuất tiếp theo hoặc kết thúc chuỗi video mã hóa, tùy việc nào xảy ra sớm hơn theo thứ tự giải mã. Tốt hơn nếu các thông báo SEI thay đổi của các tập hợp lớp kết xuất không có tác động cộng dồn.

'active_vps_id' nhận diện tập tham số video hoạt động có chứa thông tin các tập hợp lớp kết xuất mà vì đó thay đổi hoặc bổ sung được áp dụng. Giá trị active_vps_id bằng giá trị video_parameter_set_id của tập tham số video hoạt động đối với các đơn vị NAL VCL của đơn vị truy cập có chứa thông báo SEI.

'num_changed_output_layer_sets' xác định số lượng tập hợp lớp kết xuất thay đổi, mà các lớp kết xuất này được xác định trong đoạn mở rộng của tập tham số video hoạt động được xác định bởi active_vps_id. Giá trị num_changed_output_layer_sets nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả giá trị đầu cuối. Khi không có mặt, giá trị num_changed_output_layer_sets được suy ra bằng 0.

'changed_output_layer_sets_idx_entry'[i] xác định mục nhập chỉ số trong Danh mục của các mục nhập tập hợp lớp kết xuất trong phần mở rộng VPS được xác định bởi active_vps_id mà thay đổi output_layer_flag[clsIdx][j] áp dụng cho nó. Giá trị changed_output_layer_set_idx_entry[i] nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả giá trị đầu cuối. changed_output_layer_sets_idx_entry[i] không bằng changed_output_layer_sets_idx_entry[j] đối với j bất kỳ không bằng i.

'output_layer_flag'[clsIdx][j] bằng 1 xác định rằng lớp có nuh_layer_id bằng j là lớp đầu ra đích của tập hợp lớp thứ clsIdx. Giá trị output_layer_flag[clsIdx][j] bằng 0 xác định rằng lớp có nuh_layer_id bằng j không phải là lớp đầu ra đích của tập hợp lớp thứ clsIdx đó.

Khi output_layer_flag[clsIdx][j] không có mặt đối với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối, output_layer_flag[clsIdx][j] được suy ra bằng (j = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

'num_addl_output_layer_sets' xác định số lượng tập hợp lớp bổ sung, mà đối với nó, các lớp đầu ra được xác định có addl_output_layer_set_idx[i] và output_layer_flag[addllsIdx][j]. Khi không có mặt, giá trị num_addl_output_layer_sets được suy ra bằng 0. 'addl_output_layer_sets_idx'[i] xác định chỉ số addllsIdx của tập hợp lớp, mà đối với nó, output_layer_flag[addllsIdx][j] là có mặt.

'output_layer_flag'[addllsIdx][j] bằng 1 xác định rằng lớp có nuh_layer_id bằng j là lớp đầu ra đích của tập hợp lớp thứ addllsIdx. Giá trị output_layer_flag[addllsIdx][j] bằng 0 xác định rằng lớp có nuh_layer_id bằng j không phải là lớp đầu ra đích của tập hợp lớp thứ addllsIdx.

Khi output_layer_flag[addllsIdx][j] không có mặt đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối, thì output_layer_flag[addllsIdx][j] được suy ra bằng (j = = LayerSetLayerIdList[addllsIdx][NumLayersInIdList[addllsIdx] - 1]).

Khi output_layer_flag[addllsIdx][j] không có mặt đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối và khi output_layer_flag[clsIdx][j] không có mặt đối với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối, thì output_layer_flag[addllsIdx][j] được suy ra bằng (j = = LayerSetLayerIdList[addllsIdx][NumLayersInIdList[addllsIdx] - 1]).

Khi output_layer_flag[addllsIdx][j] không có mặt đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối và khi output_layer_flag[clsIdx][j] không có mặt đối với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả giá trị đầu cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả giá trị đầu cuối, thì output_layer_flag[clsIdx][j] được suy ra bằng (j = = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

Danh mục đích của các lớp ra có thể được suy ra, nếu muốn. Danh mục nhận diện của lớp kết xuất đích, targetOpLayerIdList, xác nhận diện sách củ giá trị nuh_layer_ids, theo thứ tự tăng dần lên của các giá trị nuh_layer_id mà chúng là các lớp kết xuất cho các tập hợp lớp kết xuất được chọn có thể được suy ra như sau:

Đối với tập hợp lớp kết xuất được chọn oLsIdx, danh mục nhận diện của lớp kết xuất TargetOpLayerIdList được suy ra là:

Bảng 11

for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList[numOutputLayers++] = layer_id_in_nuh[k]

Tập hợp lớp kết xuất đích có thể được xác định bởi danh mục nhận diện lớp kết xuất đích được liên kết TargetOpLayerIdList.

Tập hợp lớp kết xuất được chọn oLsIdx là ở giữa 0 và num_output_layer_sets nếu không có các thông báo SEI thay đổi của các tập hợp lớp kết xuất.

Tập hợp lớp kết xuất được chọn oLsIdx là ở giữa 0 và num_output_layer_sets + num_addl_output_layer_sets nếu có các thông báo SEI thay đổi của các tập hợp lớp kết xuất có mặt.

Danh mục đích của các lớp giải mã có thể được suy ra, nếu muốn. Danh mục nhận diện lớp giải mã đích, targetDLayerIdList, xác nhận diện sách của các giá trị nuh_layer_id, mà chúng cần được giải mã đối với các tập hợp lớp kết xuất được chọn có thể được suy ra như sau:

Đối với tập hợp lớp kết xuất được chọn oLsIdx, danh mục nhận diện của lớp kết xuất TargetOpLayerIdList được suy ra là:

Bảng 12

for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)
if(output_layer_flag[oLsIdx][k])
targetOpLayerIdList[numOutputLayers++] = layer_id_in_nuh[k]

Danh mục nhận diện lớp giải mã đích targetDLayerIdList có thể được suy ra là:

Bảng 13

for(m=0, numDecodedLayers=0; m< numOutputLayers; m++) {
for(n=0; n< NumDirectRefLayers[LayerIdInVps[targetOpLayerIdList[m]]]; n++) {
rLid=RefLayerId[LayerIdInVps[targetOpLayerIdList[m]]][n]
if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])
targetDLayerIdList[numDecodedLayers++]=rLid;
}
}

Trong một số trường hợp, targetDLayerIdList cuối cùng có thể thu được bằng cách phân loại targetDLayerIdList nêu trên theo thứ tự tăng dần lên của các giá trị nuh_layer_id.

Tập hợp nhận diện lớp giải mã đích được xác định bởi danh mục nhận diện lớp giải mã đích được liên kết targetDLayerIdList. Trong một số trường hợp, targetDLayerIdList này có thể giống như danh mục nhận diện lớp TargetDecLayerIdList từ JCTVC-L1008, xác nhận diện sách của các giá trị nuh_layer_id, theo thứ tự tăng dần lên của các giá trị nuh_layer_id, của các đơn vị NAL được được giải mã:

Tập hợp lớp kết xuất được chọn oLsIdx là ở giữa 0 và num_output_layer_sets nếu không có các thông báo SEI thay đổi của các tập hợp lớp kết xuất.

Tập hợp lớp kết xuất được chọn oLsIdx là ở giữa 0 và num_output_layer_sets + num_addl_output_layer_sets nếu có các thông báo SEI thay đổi của các tập hợp lớp kết xuất có mặt.

Như được thể hiện trên Fig.37, cú pháp thông báo SEI khác có thể được sử dụng để hỗ trợ cho sự thay đổi các tập hợp lớp kết xuất, gọi là, output_layer_sets_change(payloadSize).

'num_deleted_output_layer_sets' xác định số lượng tập hợp lớp kết xuất mà chúng được xóa và không còn có mặt. Giá trị num_deleted_output_layer_sets nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả giá trị đầu cuối. Khi không có mặt, thì giá trị num_deleted_output_layer_sets được suy ra bằng 0.

'deleted_output_layer_sets_idx_entry'[i] xác định mục nhập chỉ số trong Danh mục các mục nhập tập hợp lớp kết xuất được chỉ báo phải được xóa và không còn có mặt. Giá trị deleted_output_layer_set_idx_entry[i] nằm trong khoảng từ 0 đến num_output_layer_sets,

bao gồm cả giá trị đầu cuối. `deleted_output_layer_sets_idx_entry[i]` này không bằng `deleted_output_layer_sets_idx_entry[j]` cho j bất kỳ không bằng i .

'`num_changed_output_layer_sets`' xác định số lượng tập hợp lớp kết xuất thay đổi, mà đối với nó, các lớp kết xuất được xác định theo đoạn mở rộng vps của tập tham số video hoạt động được xác định bởi `active_vps_id`. Giá trị `num_changed_output_layer_sets` nằm trong khoảng từ 0 đến `num_output_layer_sets - num_deleted_output_layer_sets`, bao gồm cả giá trị đầu cuối. Khi không có mặt, giá trị `num_changed_output_layer_sets` được suy ra bằng 0.

Danh mục đích của các lớp ra có thể được suy ra, nếu muốn. Danh mục nhận diện của lớp kết xuất đích, `targetOpLayerIdList`, xác nhận diện sách của các giá trị `nuh_layer_id`, theo thứ tự tăng dần lên của các giá trị `nuh_layer_id` mà chúng là các lớp kết xuất cho các tập hợp lớp kết xuất được chọn có thể được suy ra như sau:

Đối với tập hợp lớp kết xuất được chọn `oLsIdx`, danh mục nhận diện của lớp kết xuất `TargetOpLayerIdList` được suy ra là:

Bảng 14

<code>for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)</code>
<code>if(output_layer_flag[oLsIdx][k])</code>
<code>targetOpLayerIdList [numOutputLayers++] = layer_id_in_nuh[k]</code>

Tập hợp lớp kết xuất đích được xác định bởi danh mục nhận diện lớp kết xuất đích được liên kết `TargetOpLayerIdList`.

Tập hợp lớp kết xuất được chọn `oLsIdx` là ở giữa 0 và `num_output_layer_sets` nếu không có các thông báo SEI thay đổi của các tập hợp lớp kết xuất.

Tập hợp lớp kết xuất được chọn `oLsIdx` là ở giữa 0 và `num_output_layer_sets - num_deleted_output_layer_sets + num_addl_output_layer_sets` nếu có các thông báo SEI thay đổi của các tập hợp lớp kết xuất có mặt.

Danh mục đích của các lớp giải mã có thể được suy ra, nếu muốn.

Danh mục nhận diện lớp giải mã đích, `targetDLayerIdList`, xác nhận diện sách của các giá trị `nuh_layer_id`, mà chúng cần được giải mã đối với các tập hợp lớp kết xuất được chọn có thể được suy ra như sau:

Đối với tập hợp lớp kết xuất được chọn `oLsIdx`, danh mục nhận diện của lớp kết xuất `TargetOpLayerIdList` có thể được suy ra là:

Bảng 15

<code>for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)</code>
<code>if(output_layer_flag[oLsIdx][k])</code>
<code>targetOpLayerIdList[numOutputLayers++] = layer_id_in_nuh[k]</code>

Sau đó danh mục nhận diện lớp giải mã đích `targetDLayerIdList` có thể được suy ra là:

Bảng 16

<code>for(m=0, numDecodedLayers=0; m< numOutputLayers; m++) {</code>
<code>for(n=0; n<NumDirectRefLayers[layer_id_in_vps[targetOpLayerIdList[m]]]; n++) {</code>
<code> rLid=RefLayerId[layer_id_in_vps[targetOpLayerIdList[m]]][n]</code>
<code> if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])</code>
<code> targetDLayerIdList[numDecodedLayers++] = rLid;</code>
<code> }</code>
<code>}</code>

Trong một số trường hợp, `targetDLayerIdList` cuối cùng có thể thu được bằng cách phân loại `targetDLayerIdList` nêu trên theo thứ tự tăng dần lên của các giá trị `nuh_layer_id`.

Tập hợp lớp giải mã đích được xác định bởi danh mục nhận diện lớp giải mã đích kết hợp `targetDLayerIdList`. Trong một số trường hợp, `targetDLayerIdList` này có thể giống như danh mục nhận diện lớp `TargetDecLayerIdList` từ JCTVC-L1008, xác nhận diện sách của các giá trị `nuh_layer_id`, theo thứ tự tăng dần lên của các giá trị `nuh_layer_id`, của các đơn vị NAL được giải mã:

Tập hợp lớp kết xuất được chọn `oLsIdx` là ở giữa 0 và `num_output_layer_sets` nếu không có các thông báo SEI thay đổi của các tập hợp lớp kết xuất.

Tập hợp lớp kết xuất được chọn `oLsIdx` là ở giữa 0 và `num_output_layer_sets - num_deleted_output_layer_sets + num_addl_output_layer_sets` nếu có các thông báo SEI thay đổi của các tập hợp lớp kết xuất có mặt.

Thông báo SEI thay đổi của các tập hợp lớp kết xuất được xác định sao cho tác động của nó không có tính cộng dồn. Theo cách khác, cú pháp thông báo này và/hoặc ngữ nghĩa

có thể được xác định sao cho chúng có tính cộng dồn. Trong trường hợp này bộ giải mã (và bộ mã hóa) có thể theo dõi các giá trị trước đây của `num_output_layer_sets` (từ phần mở rộng VPS) và/ hoặc `num_deleted_output_layer_sets` và/ hoặc `num_changed_output_layer_sets` và/ hoặc `num_addl_output_layer_sets` và có thể tích lũy các thay đổi mỗi lần khi thông báo SEI mới được truyền tín hiệu.

Theo cách khác, một hoặc nhiều phần tử cú pháp có thể được truyền tín hiệu sử dụng số lượng bit cố định đã biết thay cho $u(v)$ thay cho $ue(v)$. Ví dụ chúng có thể được truyền tín hiệu sử dụng $u(8)$ hoặc $u(16)$ hoặc $u(32)$ hoặc $u(64)$ v.v..

Theo cách khác, một hoặc nhiều phần tử cú pháp này có thể được truyền tín hiệu với $ue(v)$ hoặc sơ đồ mã hóa khác nào đó thay cho số lượng cố định các bit như mã hóa $u(v)$.

Theo cách khác, tên của các phần tử cú pháp khác nhau và ngữ nghĩa của chúng có thể được sửa đổi bằng cách bổ sung plus1 hoặc plus2 hoặc bằng cách trừ bớt minus1 hoặc minus2 so với cú pháp và ngữ nghĩa đã được mô tả.

Theo phương án khác nữa, các phần tử cú pháp khác có trong thông báo SEI của các tập hợp lớp kết xuất có thể được truyền tín hiệu mỗi ảnh hoặc theo tần suất khác tại bất kỳ chỗ nào trong dòng bit. Ví dụ chúng có thể được truyền tín hiệu trong tiêu đề phân đoạn phiên, tập tham số tương thích pps/ sps/ vps/ hoặc bất kỳ tập tham số khác hoặc phần danh định khác của dòng bit.

Theo phương án khác nữa, các phần tử cú pháp khác có thể được truyền tín hiệu mỗi ảnh hoặc theo tần suất khác tại bất kỳ chỗ nào trong dòng bit. Ví dụ, chúng có thể được truyền tín hiệu trong tiêu đề phân đoạn phiên, tập tham số tương thích pps/ sps/ vps/ hoặc bất kỳ tập tham số khác hoặc phần danh định khác của dòng bit.

Theo các phương án khác nữa tất cả các bối cảnh được xác định trong sáng chế liên quan đến các tập hợp lớp kết xuất có thể được áp dụng cho các điểm hoạt động kết xuất như được xác định trong JCTVC-L0452 và JCTVC-L0453 và/ hoặc cho các điểm hoạt động như được xác định trong JCTVC-L1003.

Ví dụ 2

Ví dụ khác của sáng chế được mô tả dưới đây. Lưu ý là, để tiện lợi, các bộ phận có các chức năng giống như chức năng của các bộ phận tương ứng được minh họa trong ví dụ thứ nhất được biểu thị bằng các số chỉ dẫn giống nhau và việc mô tả các bộ phận này được bỏ qua trong bản mô tả.

Đã xác định rằng việc truyền tín hiệu cho bộ đệm hình ảnh giải mã (DPB) của thứ tự kết xuất dựa trên việc sử dụng các phần tử cú pháp `sps_max_num_reorder_pics[HighestTid]`, `sps_max_latency_increase_plus1[HighestTid]` và `sps_max_dec_pic_buffering[HighestTid]` để kết xuất và gỡ bỏ các hình ảnh khỏi DPB không ghi chép các đặc điểm của bộ đệm mà có thể phát sinh từ việc mã hóa video mở rộng được, chẳng hạn như khi các lượng khác nhau của các lớp nâng cao được sử dụng mà chúng có xu hướng thay đổi sau khi nội dung đã mã hóa dựa vào các ưu tiên kiểm tra ảnh của người sử dụng và các lớp nâng cao đa cảnh mà chúng có xu hướng thay đổi sau khi nội dung đã mã hóa dựa vào các ưu tiên kiểm tra ảnh của người sử dụng. Cũng xác định rằng việc truyền tín hiệu cho bộ đệm hình ảnh giải mã (DPB) của thứ tự kết xuất dựa trên sử dụng các phần tử cú pháp `sps_max_num_reorder_pics[HighestTid]`, `sps_max_latency_increase_plus1[HighestTid]` và `sps_max_dec_pic_buffering[HighestTid]` để kết xuất và gỡ bỏ các hình ảnh khỏi DPB có thể không phải là tối ưu về mặt sử dụng bộ nhớ của DPB khi bộ giải mã hoạt động tại điểm hoạt động nhất định và/ hoặc đang kết xuất tập hợp lớp kết xuất được chọn. Để tạo điều kiện cho các khác biệt như vậy trong các ưu tiên kiểm tra ảnh, bộ đệm hình ảnh giải mã (DPB) của thứ tự kết xuất có thể còn và/hoặc Theo cách khác, phải dựa vào các phần tử cú pháp như vậy được bao gồm cùng với phần mở rộng của tập tham số video (phần mở rộng VPS) để tạo ra các phần tử cú pháp cho một hoặc nhiều lớp nâng cao. Theo cách này các phần tử cú pháp có thể được chọn để phù hợp đặc biệt với điểm hoạt động cụ thể hoặc tập hợp lớp kết xuất, có xu hướng tương ứng với các ưu tiên kiểm tra ảnh của người sử dụng.

Các tham số liên quan đến việc đệm của DPB, `vps_max_dec_pic_buffering_minus1`, `vps_max_num_reorder_pics`, `vps_max_latency_increase_plus1` có thể được truyền tín hiệu cho các lớp phụ đối với CVS cho một hoặc nhiều điểm hoạt động và/hoặc cho các tập hợp lớp kết xuất trong phần mở rộng VPS. Tương tự, hệ thống này có thể xác định quy trình hoạt động và đệm cho DPB của thứ tự kết xuất để sử dụng các tham số đệm DPB được truyền tín hiệu nêu trên từ phần mở rộng VPS nếu chúng được truyền tín hiệu cho điểm hoạt động trong thử nghiệm hoặc đối với các tập hợp lớp kết xuất được chọn. Theo cách khác, các tham số cấp SPS tương ứng từ SPS hoạt động (khi `currLayerId` tương ứng với `nuh_layer_id` của hình ảnh hiện thời bằng 0) hoặc từ SPS lớp hoạt động phụ thuộc vào `layer_id` của lớp hiện thời được sử dụng. Trong một số trường hợp các tham số `vps_max_dec_pic_buffering_minus1`, `vps_max_num_reorder_pics`,

`vps_max_latency_increase_plus1` có thể được gọi là `max_vps_dec_pic_buffering_minus1`, `max_vps_num_reorder_pics`, `max_vps_latency_increase_plus1`.

Trong trường hợp khác, tham số liên quan đến việc đệm của DPB `max_vps_dec_pic_buffering_minus1` được truyền tín hiệu cho các lớp phụ đối với CVS cho một hoặc nhiều tập hợp lớp kết xuất trong phần mở rộng VPS. Các tập hợp lớp kết xuất có thể tương ứng với các điểm hoạt động. Các tham số liên quan đến việc đệm của DPB `max_vps_num_reorder_pics`, `max_vps_latency_increase_plus1` có thể được truyền tín hiệu cho các lớp phụ đối với CVS cho mỗi lớp trong phần mở rộng VPS. Tương tự, hệ thống này có thể xác định quy trình hoạt động và đệm cho DPB của thứ tự kết xuất để sử dụng các tham số đệm DPB được truyền tín hiệu nêu trên từ phần mở rộng VPS nếu chúng được truyền tín hiệu cho điểm hoạt động trong thử nghiệm hoặc đối với các tập hợp lớp kết xuất được chọn. Theo cách khác, các tham số cấp SPS tương ứng từ SPS hoạt động (khi `currLayerId` tương ứng với `nuh_layer_id` của hình ảnh hiện thời bằng 0) hoặc từ SPS lớp hoạt động phụ thuộc vào `layer_id` của lớp hiện thời được sử dụng.

Như được thể hiện trên Fig.38A, `vps_extension` được biến đổi làm ví dụ được minh họa. Phần mở rộng VPS bao gồm cú pháp mới, gọi là, `num_op_dpb_info_parameters` và `operation_point_layer_set_idx[i]`. Phần mở rộng VPS được biến đổi này có thể được xác định về điểm hoạt động là dòng bit được tạo ra từ dòng bit khác nhờ hoạt động của quy trình trích xuất dòng bit phụ với dòng bit khác, `temporalId` cao nhất đích và danh mục nhận diện lớp đích dưới dạng các dữ liệu đầu vào.

`num_output_layer_sets` xác định số lượng tập hợp lớp, mà đối với nó, các lớp đầu ra được xác định có `output_layer_set_index[i]` và `output_layer_flag[lsIdx][j]`. Khi không có mặt, thì giá trị `num_output_layer_sets` được suy ra bằng 0. Tập hợp lớp mô tả các lớp đầu ra là tập hợp lớp đầu ra.

`output_layer_set_idx[i]` xác định chỉ số `lsIdx` của tập hợp lớp, mà đối với nó, `output_layer_flag[lsIdx][j]` là có mặt.

`output_layer_flag[lsIdx][j]` bằng 1 xác định rằng lớp có `nuh_layer_id` bằng `j` là lớp đầu ra đích của tập hợp lớp thứ `lsIdx`. Giá trị `output_layer_flag[lsIdx][j]` bằng 0 xác định rằng lớp có `nuh_layer_id` bằng `j` không phải là lớp đầu ra đích của tập hợp lớp thứ `lsIdx`.

`num_op_dpb_info_parameters` xác định số lượng cấu trúc cú pháp `op_dpb_parameters()` có mặt trong RBSP của phần mở rộng VPS, được xác định về điểm

hoạt động. `num_op_dpb_info_parameters` của các bộ giải mã là nằm trong khoảng từ 0 đến `vps_num_layer_sets_minus1`, bao gồm cả giá trị đầu cuối.

`operation_point_layer_set_idx[i]` xác định chỉ số, đến Danh mục của các tập hợp lớp được xác định bởi các điểm hoạt động mà cấu trúc cú pháp `op_dpb_info_parameters()` thứ *i* trong phần mở rộng VPS tác động đến. Giá trị `operation_point_layer_set_idx[i]` có thể nằm trong khoảng từ 0 đến `vps_num_layer_sets_minus1`, bao gồm cả giá trị đầu cuối. Để có được tính phù hợp của dòng bit `operation_point_layer_set_idx[i]` không bằng `operation_point_layer_set_idx[j]` cho *j* bất kỳ không bằng *i*.

Như được thể hiện trên Fig.39A, `op_dpb_info_parameters` xác định `vps_max_sub_layers_minus1[j]`, `vps_sub_layer_ordering_info_present_flag[j]`, `vps_max_dec_pic_buffering_minus1[j][k]`, `vps_max_num_reorder_pics[j][k]` và `vps_max_latency_increase_plus1[j][k]`.

`vps_max_sub_layers_minus1[j] + 1` chỉ báo có bao nhiêu lớp phụ được bao gồm. `vps_max_sub_layers_minus1[j] + 1` xác định lượng tối đa của các lớp phụ tạm thời có thể có mặt trong CVS cho lớp có `nuh_layer_id` bằng *j*. Giá trị `vps_max_sub_layers_minus1[j]` nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu cuối.

`vps_sub_layer_ordering_info_present_flag[j]` chỉ báo liệu cú pháp này là cho một tập hợp bao gồm tất cả các lớp hay cho mỗi lớp đơn lẻ. `vps_sub_layer_ordering_info_present_flag[j]` bằng 1 xác định rằng `vps_max_dec_pic_buffering_minus1[j][k]`, `vps_max_num_reorder_pics[j][k]` và `vps_max_latency_increase_plus1[j][k]` là có mặt cho lớp có `nuh_layer_id` bằng *j* cho các lớp phụ `vps_max_sub_layers_minus1[j] + 1`. `vps_sub_layer_ordering_info_present_flag[j]` bằng 0 xác định rằng các giá trị `vps_max_dec_pic_buffering_minus1[j][vps_max_sub_layers_minus1[j]]`, `vps_max_num_reorder_pics[j][vps_max_sub_layers_minus1[j]]` và `vps_max_latency_increase_plus1[j][vps_max_sub_layers_minus1[j]]` áp dụng cho tất cả các lớp phụ đối với lớp có `nuh_layer_id` bằng *j*.

`vps_max_dec_pic_buffering_minus1[j][k] + 1` xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS cho lớp có `nuh_layer_id` bằng *j* trong các bộ của các bộ đệm lưu trữ hình ảnh khi `HighestTid` bằng *k*. Giá trị `vps_max_dec_pic_buffering_minus1[j][k]` nằm trong khoảng từ 0 đến `MaxDpbSize - 1` (như được xác định trong khoản mục A.4), bao gồm cả giá trị đầu cuối. Khi *k* lớn hơn 0, thì

$vps_max_dec_pic_buffering_minus1[j][k]$ sẽ lớn hơn hoặc bằng $vps_max_dec_pic_buffering_minus1[j][k - 1]$. Khi $vps_max_dec_pic_buffering_minus1[j][k]$ không có mặt đối với k nằm trong khoảng từ 0 đến $vps_max_sub_layers_minus1[j] - 1$, bao gồm cả giá trị đầu cuối, do $vps_sub_layer_ordering_info_present_flag[j]$ bằng 0, thì suy ra $vps_max_dec_pic_buffering_minus1[j][vps_max_sub_layers_minus1[j]]$.

$vps_max_num_reorder_pics[j][k]$ chỉ báo lượng tối đa cho phép của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng j theo thứ tự giải mã và tiếp theo hình ảnh đó theo thứ tự kết xuất khi $HighestTid$ bằng k . Giá trị $vps_max_num_reorder_pics[j][k]$ nằm trong khoảng từ 0 đến $vps_max_dec_pic_buffering_minus1[j][k]$, bao gồm cả giá trị đầu cuối. Khi k lớn hơn 0, thì $vps_max_num_reorder_pics[j][k]$ lớn hơn hoặc bằng $vps_max_num_reorder_pics[j][k - 1]$. Khi $vps_max_num_reorder_pics[j][k]$ không có mặt đối với k nằm trong khoảng từ 0 đến $vps_max_sub_layers_minus1[j] - 1$, bao gồm cả giá trị đầu cuối, do $vps_sub_layer_ordering_info_present_flag[j]$ bằng 0, thì suy ra $vps_max_num_reorder_pics[j][vps_max_sub_layers_minus1[j]]$.

$vps_max_latency_increase_plus1[j][k]$ không bằng 0 được sử dụng để tính toán giá trị $VpsMaxLatencyPictures[j][k]$, xác định lượng tối đa của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng j theo thứ tự kết xuất và tiếp theo hình ảnh đó theo thứ tự giải mã khi $HighestTid$ bằng k .

Khi $vps_max_latency_increase_plus1[j][k]$ không bằng 0, giá trị $VpsMaxLatencyPictures[j][k]$ có thể được xác định như sau:

$$VpsMaxLatencyPictures[j][k] = vps_max_num_reorder_pics[j][k] + vps_max_latency_increase_plus1[j][k] - 1$$

Khi $vps_max_latency_increase_plus1[j][k]$ bằng 0, thì không có giới hạn tương ứng nào được thể hiện.

Giá trị $vps_max_latency_increase_plus1[j][k]$ nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu cuối. Khi $vps_max_latency_increase_plus1[j][k]$ không có mặt đối với k nằm trong khoảng từ 0 đến $vps_max_sub_layers_minus1[j] - 1$, bao gồm cả giá trị đầu cuối, do $vps_sub_layer_ordering_info_present_flag[j]$ bằng 0, thì suy ra $vps_max_latency_increase_plus1[j][vps_max_sub_layers_minus1[j]]$.

'Vps_max_sub_layers_minus1'[id][j] + 1 xác định lượng tối đa của các lớp phụ tạm thời có thể có mặt trong CVS cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id. Giá trị vps_max_sub_layers_minus1[id][j] nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu cuối.

vps_sub_layer_ordering_info_present_flag[id][j] bằng 1 xác định rằng vps_max_dec_pic_buffering_minus1[id][j][k], vps_max_num_reorder_pics[id][j][k] và vps_max_latency_increase_plus1[id][j][k] là có mặt cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id cho các lớp phụ vps_max_sub_layers_minus1[id][j] + 1.

vps_sub_layer_ordering_info_present_flag[id][j] bằng 0 xác định rằng các giá trị vps_max_dec_pic_buffering_minus1[id][j][vps_max_sub_layers_minus1[id][j]], vps_max_num_reorder_pics[id][j][vps_max_sub_layers_minus1[id][j]] và vps_max_latency_increase_plus1[id][j][vps_max_sub_layers_minus1[id][j]] áp dụng cho tất cả các lớp phụ cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id.

vps_max_dec_pic_buffering_minus1[id][j][k] + 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id trong các bộ của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng k. Giá trị vps_max_dec_pic_buffering_minus1[id][j][k] nằm trong khoảng từ 0 đến MaxDpbSize - 1 (như được xác định trong khoản mục A.4), bao gồm cả giá trị đầu cuối. Khi k lớn hơn 0, thì vps_max_dec_pic_buffering_minus1[id][j][k] sẽ lớn hơn hoặc bằng vps_max_dec_pic_buffering_minus1[id][j][k - 1]. Khi vps_max_dec_pic_buffering_minus1[id][j][k] không có mặt đối với k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[id][j] - 1, bao gồm cả giá trị đầu cuối, do vps_sub_layer_ordering_info_present_flag[id][j] bằng 0, thì suy ra vps_max_dec_pic_buffering_minus1[id][j][vps_max_sub_layers_minus1[id][j]].

vps_max_num_reorder_pics[id][j][k] chỉ báo lượng tối đa cho phép của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id theo thứ tự giải mã và tiếp theo hình ảnh đó theo thứ tự kết xuất khi HighestTid bằng k. Giá trị vps_max_num_reorder_pics[id][j][k] nằm trong khoảng từ 0 đến vps_max_dec_pic_buffering_minus1[id][j][k], bao gồm cả giá trị đầu cuối. Khi k lớn hơn 0, thì vps_max_num_reorder_pics[id][j][k] sẽ lớn hơn hoặc bằng vps_max_num_reorder_pics[id][j][k - 1]. Khi vps_max_num_reorder_pics[id][j][k] không

có mặt đối với k nằm trong khoảng từ 0 đến $\text{vps_max_sub_layers_minus1}[\text{id}][j] - 1$, bao gồm cả giá trị đầu cuối, do $\text{vps_sub_layer_ordering_info_present_flag}[\text{id}][j]$ bằng 0, thì suy ra $\text{vps_max_num_reorder_pics}[\text{id}][j][\text{vps_max_sub_layers_minus1}[\text{id}][j]]$.

$\text{vps_max_latency_increase_plus1}[\text{id}][j][k]$ không bằng 0 được sử dụng để tính toán giá trị $\text{VpsMaxLatencyPictures}[\text{id}][j][k]$, which xác định lượng tối đa của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id theo thứ tự kết xuất và tiếp theo hình ảnh đó theo thứ tự giải mã khi HighestTid bằng k .

Khi $\text{vps_max_latency_increase_plus1}[\text{id}][j][k]$ không bằng 0, thì giá trị $\text{VpsMaxLatencyPictures}[\text{id}][j][k]$ được xác định như sau:

$$\text{VpsMaxLatencyPictures}[\text{id}][j][k] = \text{vps_max_num_reorder_pics}[\text{id}][j][k] + \text{vps_max_latency_increase_plus1}[\text{id}][j][k] - 1$$

Khi $\text{vps_max_latency_increase_plus1}[\text{id}][j][k]$ bằng 0, thì không có giới hạn tương ứng nào được thể hiện.

Giá trị $\text{vps_max_latency_increase_plus1}[\text{id}][j][k]$ nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu cuối. Khi $\text{vps_max_latency_increase_plus1}[\text{id}][j][k]$ không có mặt đối với k nằm trong khoảng từ 0 đến $\text{vps_max_sub_layers_minus1}[\text{id}][j] - 1$, bao gồm cả giá trị đầu cuối, do $\text{vps_sub_layer_ordering_info_present_flag}[\text{id}][j]$ bằng 0, thì suy ra $\text{vps_max_latency_increase_plus1}[\text{id}][j][\text{vps_max_sub_layers_minus1}[\text{id}][j]]$.

Như được thể hiện trên Fig.39 B, $\text{op_dpb_info_parameters}$ có thể được biến đổi như được thể hiện là $\text{op_dpb_info_parameters}(\text{id}, j)$. Trong trường hợp này cú pháp của phần mở rộng VPS có thể được thể hiện trên Fig.38B. Bộ giải mã tham chiếu giả định (HRD) được sử dụng để kiểm tra tính phù hợp của dòng bit và bộ giải mã. Hai loại dòng bit hoặc tập hợp phụ của dòng bit phụ thuộc vào việc kiểm tra tính phù hợp HRD theo Nhóm hợp tác về mã hóa video (JCT-VC). Loại thứ nhất, gọi là dòng bit Loại I, là luồng đơn vị NAL chỉ chứa các đơn vị NAL VCL và các đơn vị NAL với nal_unit_type bằng FD_NUT (các đơn vị NAL dữ liệu của bộ lọc) cho tất cả các đơn vị truy cập trong dòng bit. Loại thứ hai, gọi là dòng bit Loại II, có chứa, ngoài các đơn vị NAL VCL và các đơn vị NAL dữ liệu của bộ lọc đó cho tất cả các đơn vị truy cập trong dòng bit, ít nhất một trong số (a) các đơn vị NAL không VCL bổ sung khác với các đơn vị NAL dữ liệu của bộ lọc và (b) tất cả các phần tử cú pháp

leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes và trailing_zero_8bits mà chúng tạo thành luồng byte từ luồng đơn vị NAL.

Các phần tử cú pháp của các đơn vị NAL không VCL (hoặc các giá trị mặc định đối với một số phần tử cú pháp), cần thiết đối với HRD, được xác định trong các khoản có ý nghĩa của điều 7, các Phụ lục D và E.

Hai loại tập tham số HRD (các tham số HRD NAL và các tham số HRD VCL) được sử dụng. Các tập tham số HRD được truyền tín hiệu thông qua cấu trúc cú pháp `hrd_parameters()`, có thể là tiêu mục của cấu trúc cú pháp SPS hoặc cấu trúc cú pháp VPS.

Có thể cần đến nhiều thử nghiệm để kiểm tra tính phù hợp của dòng bit, được gọi là dòng bit trong thử nghiệm. Đối với mỗi thử nghiệm, áp dụng các bước theo thứ tự liệt kê sau đây:

(1) Điểm hoạt động trong thử nghiệm, ký hiệu là `TargetOp`, được chọn. Danh mục nhận diện lớp `OpLayerIdList` của `TargetOp` gồm danh mục của các giá trị `nuh_layer_id`, theo thứ tự tăng dần lên của các giá trị `nuh_layer_id`, có mặt trong tập hợp phụ của dòng bit kết hợp với `TargetOp`, là tập hợp phụ của các giá trị `nuh_layer_id` có mặt trong dòng bit trong thử nghiệm. `OpTid` của `TargetOp` bằng `TemporalId` cao nhất có mặt trong tập hợp phụ của dòng bit được kết hợp với `TargetOp`.

(2) `TargetDecLayerIdList` được thiết đặt bằng `OpLayerIdList` của `TargetOp`, `HighestTid` được thiết đặt bằng `OpTid` của `TargetOp` và quy trình trích xuất dòng bit phụ như được xác định trong khoản 10 được hủy bỏ với dòng bit trong thử nghiệm, `HighestTid` và `TargetDecLayerIdList` dưới dạng các dữ liệu đầu vào và đầu ra được gán cho `BitstreamToDecode`.

(3) Cấu trúc cú pháp `hrd_parameters()` và cấu trúc cú pháp `sub_layer_hrd_parameters()` có thể áp dụng cho `TargetOp` được chọn. Nếu `TargetDecLayerIdList` chứa tất cả các giá trị `nuh_layer_id` có mặt trong dòng bit trong thử nghiệm, thì cấu trúc cú pháp `hrd_parameters()` trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được xác định trong bản mô tả này) đã chọn. Theo cách khác, cấu trúc cú pháp `hrd_parameters()` trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được xác định trong bản mô tả này) áp dụng cho `TargetOp` được chọn. Bên trong cấu trúc cú pháp `hrd_parameters()` đã chọn, nếu `BitstreamToDecode` là dòng bit Loại I, thì cấu trúc cú pháp `sub_layer_hrd_parameters(HighestTid)` theo ngay sau điều kiện “if(

vcl_hrd_parameters_present_flag)” được chọn và biến NalHrdModeFlag được thiết đặt bằng 0; theo cách khác, (BitstreamToDecode là dòng bit Loại II), thì cấu trúc cú pháp sub_layer_hrd_parameters(HighestTid) theo sau hoặc điều kiện “if(vcl_hrd_parameters_present_flag)” (trong trường hợp này biến NalHrdModeFlag được thiết đặt bằng 0) hoặc điều kiện “nếu (NAL_hrd_parameters_present_flag)” (trong trường hợp này biến NalHrdModeFlag được thiết đặt bằng 1) đã chọn. Khi BitstreamToDecode là dòng bit Loại II và NalHrdModeFlag bằng 0, thì tất cả các đơn vị NAL không VCL trừ các đơn vị NAL dữ liệu của bộ lọc và tất cả các phần tử cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes và trailing_zero_8bits mà chúng tạo thành luồng byte từ luồng đơn vị NAL (như được xác định trong Phụ lục B), khi có mặt, được loại ra khỏi BitstreamToDecode và dòng bit còn lại được gán cho BitstreamToDecode.

Trong trường hợp khác, có thể cần đến nhiều thử nghiệm để kiểm tra tính phù hợp của dòng bit, được gọi là dòng bit trong thử nghiệm. Đối với mỗi thử nghiệm, áp dụng các bước theo thứ tự liệt kê sau đây:

(1) Tập hợp lớp kết xuất trong thử nghiệm, ký hiệu là TargetOpLs được chọn. Điểm hoạt động được tham chiếu trong TargetOpLs nhờ output_layer_set_idx[] xác định điểm hoạt động trong thử nghiệm. Danh mục nhận diện lớp kết xuất OpLayerIdList của TargetOpLs gồm danh mục của các giá trị nuh_layer_id, theo thứ tự tăng dần lên của các giá trị nuh_layer_id, có mặt trong tập hợp phụ của dòng bit kết hợp với TargetOp và TargetOpLs, là tập hợp phụ của các giá trị nuh_layer_id có mặt trong dòng bit trong thử nghiệm. OpTid của TargetOp bằng TemporalId cao nhất có mặt trong tập hợp phụ của dòng bit kết hợp với TargetOp.

(2) TargetDecLayerIdList được thiết đặt bằng danh mục nhận diện lớp giải mã đích targetDLayerIdList đối với các tập hợp lớp kết xuất được chọn TargetOpLs, HighestTid được thiết đặt bằng OpTid của TargetOp và quy trình trích xuất dòng bit phụ như được xác định trong khoản 10 được hủy bỏ với dòng bit trong thử nghiệm, HighestTid và TargetDecLayerIdList dưới dạng các dữ liệu đầu vào và đầu ra được gán cho BitstreamToDecode.

(3) Cấu trúc cú pháp hrd_parameters() và cấu trúc cú pháp sub_layer_hrd_parameters() có thể áp dụng cho TargetOp được chọn. Nếu TargetDecLayerIdList chứa tất cả các giá trị nuh_layer_id có mặt trong dòng bit trong thử nghiệm, cấu trúc cú pháp hrd_parameters() trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được

xác định trong bản mô tả này) đã chọn. Theo cách khác, cấu trúc cú pháp `hrd_parameters()` trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được xác định trong bản mô tả này) áp dụng cho `TargetOp` được chọn. Bên trong cấu trúc cú pháp `hrd_parameters()` đã chọn, nếu `BitstreamToDecode` là dòng bit Loại I, thì cấu trúc cú pháp `sub_layer_hrd_parameters(HighestTid)` theo ngay sau điều kiện “if(`vcl_hrd_parameters_present_flag`)” được chọn và biến `NalHrdModeFlag` được thiết đặt bằng 0; theo cách khác, (`BitstreamToDecode` là dòng bit Loại II), thì cấu trúc cú pháp `sub_layer_hrd_parameters(HighestTid)` theo ngay sau hoặc điều kiện “if(`vcl_hrd_parameters_present_flag`)” (trong trường hợp này biến `NalHrdModeFlag` được thiết đặt bằng 0) hoặc điều kiện “if(`NAL_hrd_parameters_present_flag`)” (trong trường hợp này biến `NalHrdModeFlag` được thiết đặt bằng 1) đã chọn. khi `BitstreamToDecode` là dòng bit Loại II và `NalHrdModeFlag` bằng 0, tất cả các đơn vị NAL không VCL trừ các đơn vị NAL dữ liệu của bộ lọc và tất cả các phần tử cú pháp `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes` và `trailing_zero_8bits` mà chúng tạo thành luồng byte từ luồng đơn vị NAL (như được xác định trong Phụ lục B), khi có mặt, được vứt bỏ khỏi `BitstreamToDecode` và dòng bit còn lại được gán cho `BitstreamToDecode`.

Bộ giải mã phù hợp có thể đáp ứng tất cả các yêu cầu được xác định trong khoản này.

(1) Bộ giải mã đòi hỏi tính phù hợp với biên dạng, lớp và mức cụ thể phải có khả năng giải mã thành công tất cả các dòng bit phù hợp với các yêu cầu về phù hợp của dòng bit được xác định trong khoản mục C.4, theo cách thức được xác định trong Phụ lục A, với điều kiện là tất cả các VPS, SPS và PPS được đề cập trong các đơn vị NAL VCL và khoảng đệm thích hợp và các thông báo SEI định thời hình ảnh được chuyển đến bộ giải mã, một cách kịp thời, hoặc là theo dòng bit (bởi các đơn vị NAL không VCL), hoặc nhờ các phương tiện bên ngoài không được xác định trong bản mô tả này.

(2) Khi dòng bit chứa các phần tử cú pháp có các giá trị mà chúng được xác định là dự phòng và nó được xác định rằng các bộ giải mã cần bỏ qua các giá trị của các phần tử cú pháp hoặc Các đơn vị NAL chứa các phần tử cú pháp có các giá trị dự phòng và mặt khác dòng bit phù hợp với bản mô tả này, thì bộ giải mã phù hợp sẽ giải mã dòng bit theo cùng cách thức giống như khi nó giải mã dòng bit phù hợp và sẽ bỏ qua các phần tử cú pháp hoặc các đơn vị NAL chứa các phần tử cú pháp có các giá trị dự phòng như đã xác định.

Có hai loại phù hợp đối với bộ giải mã: phù hợp định thời kết xuất và phù hợp thứ tự kết xuất.

Để kiểm tra tính phù hợp của bộ giải mã, các dòng bit phù hợp với biên dạng, lớp và mức được đòi hỏi, như được xác định trong khoản mục C.4 được tạo ra bởi bộ lập lịch trình luồng giả định (HSS) cho cả HRD và bộ giải mã trong thử nghiệm (DUT). Tất cả hình ảnh giải mã được cắt xén được kết xuất bởi HRD cũng sẽ được kết xuất bởi DUT, mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT sẽ là ảnh với PicOutputFlag bằng 1 và, đối với mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT, các giá trị của tất cả các mẫu được kết xuất phải bằng các giá trị của các mẫu được tạo ra bởi quy trình giải mã được xác định.

Để phù hợp bộ giải mã định thời kết xuất, HSS hoạt động như được mô tả trên đây, với các lịch trình được chọn chỉ từ tập hợp phụ của các giá trị SchedSelIdx, mà đối với nó, tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A đối với các biên dạng, lớp và mức, hoặc với các lịch trình cung cấp “nội suy” như được xác định dưới đây, mà đối với nó, tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A. Lịch trình cung cấp giống nhau được sử dụng cho cả HRD và DUT.

Khi các tham số HRD và các thông báo SEI về khoảng đệm có mặt với $\text{cpb_cnt_minus1}[\text{HighestTid}]$ lớn hơn 0, bộ giải mã sẽ có khả năng giải mã dòng bit như được tạo ra từ sự vận hành HSS sử dụng lịch trình cung cấp “nội suy” được xác định là có tốc độ bit đỉnh r , kích thước CPB $c(r)$ và khoảng trễ gỡ bỏ CPB ban đầu

Biểu thức 15

$$(f(r) \div r)$$

là như sau:

Bảng 17

$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) + (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]),$	(C-22)
$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha) * \text{CpbSize}[\text{SchedSelIdx} - 1],$	(C-23)
$f(r) = \alpha * \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] + (1 - \alpha) * \text{InitCpbRemovalDelay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1]$	(C-24)

cho bất kỳ một $\text{SchedSelIdx} > 0$ và r sao cho $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$ sao cho r và $c(r)$ nằm trong các giới hạn như được xác định trong Phụ lục A đối với tốc độ bit tối đa và kích thước bộ đệm đối với các biên dạng, lớp và mức. $\text{InitCpbRemovalDelay}[\text{SchedSelIdx}]$ có thể khác biệt cho mỗi khoảng đệm và cần được tính toán lại.

Để phù hợp bộ giải mã định thời kết xuất, HRD như được mô tả trên đây được sử dụng và định thời (liên quan đến thời gian cung cấp bit thứ nhất) của kết xuất hình ảnh là giống nhau đối với cả HRD và DUT cho tới khoảng trễ cố định.

Để phù hợp bộ giải mã của thứ tự kết xuất, quy định sau đây được áp dụng:

(1) HSS cung cấp dòng bit BitstreamToDecode cho DUT “theo yêu cầu” từ DUT, nghĩa là HSS cung cấp các bit (theo thứ tự giải mã) chỉ khi DUT cần nhiều bit hơn để tiếp tục quá trình xử lý của nó. Điều này có nghĩa là đối với thử nghiệm này, bộ đệm hình ảnh giải mã của DUT có thể nhỏ bằng kích thước của đơn vị giải mã lớn nhất.

(2) HDR cải biến như được mô tả dưới đây được sử dụng và HSS cung cấp dòng bit đến HRD nhờ một trong các lịch trình được xác định trong dòng bit BitstreamToDecode sao cho tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A. Thứ tự kết xuất các hình ảnh là giống nhau đối với cả HRD và DUT.

(3) Kích thước CPB HRD được xác định là $CPBSize[SchedSelIdx]$ như được xác định trong khoản E.2.3, trong đó $SchedSelIdx$ và các tham số HRD được chọn như được xác định trong khoản mục C.1. Kích thước DPB được xác định là $sps_max_dec_pic_buffering_minus1[HighestTid] + 1$ từ SPS hoạt động (khi nuh_layer_id đối với hình ảnh giải mã hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho giá trị nuh_layer_id của hình ảnh giải mã hiện thời. Trong một số trường hợp, nếu các tham số thông tin DPB của điểm hoạt động $op_dpb_info_parameters()$ có mặt đối với các tập hợp lớp kết xuất được chọn, thì kích thước DPB được xác định là $vps_max_dec_pic_buffering_minus1[HighestTid]$ khi $currLayerId$ bằng 0 hoặc được thiết đặt bằng $vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid]$ đối với $currLayerId$ cho điểm hoạt động trong thử nghiệm khi $currLayerId$ lớn hơn 0, trong đó $currLayerId$ là nuh_layer_id của hình ảnh giải mã hiện thời. Theo cách khác, nếu điểm hoạt động DPB information parameters $op_dpb_info_parameters()$ không có mặt cho điểm hoạt động trong thử nghiệm, thì kích thước DPB được xác định là $sps_max_dec_pic_buffering_minus1[HighestTid] + 1$ từ SPS hoạt động (khi nuh_layer_id đối với hình ảnh giải mã hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho giá trị nuh_layer_id của hình ảnh giải mã hiện thời.

Trong một số trường hợp, nếu các tham số thông tin DPB của tập hợp lớp kết xuất $op_dpb_info_parameters()$ có mặt đối với các tập hợp lớp kết xuất được chọn, thì kích thước DPB được xác định là $vps_max_dec_pic_buffering_minus1[HighestTid]$ khi

`currLayerId` bằng 0 hoặc được thiết đặt bằng `vps_max_dec_pic_buffering_minus1[currLayerId][HighestTid]` cho `currLayerId` đối với các tập hợp lớp kết xuất được chọn, trong đó `currLayerId` là `nuh_layer_id` của hình ảnh giải mã hiện thời. Theo cách khác, nếu các tham số thông tin DPB của tập hợp lớp kết xuất `oop_dpb_info_parameters()` không có mặt đối với các tập hợp lớp kết xuất được chọn, thì kích thước DPB được xác định là `sps_max_dec_pic_buffering_minus1[HighestTid] + 1` từ SPS hoạt động (khi `nuh_layer_id` đối với hình ảnh giải mã hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `nuh_layer_id` của hình ảnh giải mã hiện thời.

Thời gian gỡ bỏ khỏi CPB đối với HRD là thời gian tới của bit cuối cùng và việc giải mã là ngay lập tức. Hoạt động của DPB của HRD này là như được mô tả trong các khoản từ C.5.2 đến C.5.2.3.

Bộ đệm hình ảnh giải mã chứa các bộ đệm lưu trữ hình ảnh. Số lượng bộ đệm lưu trữ hình ảnh cho `nuh_layer_id` bằng 0 bắt nguồn từ SPS hoạt động. Số lượng bộ đệm lưu trữ hình ảnh cho mỗi giá trị `nuh_layer_id` không phải là 0 bắt nguồn từ SPS lớp hoạt động cho giá trị `nuh_layer_id` không phải 0 đó. Mỗi trong số các bộ đệm lưu trữ hình ảnh chứa hình ảnh giải mã được đánh dấu là “được sử dụng để tham chiếu” hoặc được giữ lại để kết xuất trong tương lai. Quy trình kết xuất và gỡ bỏ các hình ảnh khỏi DPB như được xác định trong khoản mục F.13.5.2.2 được hủy bỏ, tiếp theo là yêu cầu của quy trình giải mã hình ảnh, đánh dấu, đệm bổ sung và lưu trữ như được xác định trong khoản mục F.13.5.2.3. Quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 và được hủy bỏ như được xác định trong các khoản mục F.13.5.2.2 và F.13.5.2.3.

Kết xuất và gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thì khi đơn vị giải mã thứ nhất của đơn vị truy cập có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB và tiếp tục diễn ra như sau.

Quy trình giải mã đối với RPS như được xác định trong khoản 8.3.2 được hủy bỏ.

(1) Nếu hình ảnh hiện thời là hình ảnh IRAP với `NoRaslOutputFlag` bằng 1 và với `nuh_layer_id` bằng 0 không phải là ảnh 0, các bước theo thứ tự như sau được áp dụng:

(A) Biến `NoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm theo cách sau:

(i) Nếu hình ảnh hiện thời là ảnh CRA, thì NoOutputOfPriorPicsFlag được thiết đặt bằng 1 (không quan tâm đến giá trị no_output_of_prior_pics_flag).

(ii) Theo cách khác, nếu giá trị pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] được suy ra từ SPS hoạt động lần lượt khác với giá trị pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động đối với ảnh đứng trước, NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được thiết đặt bằng 1 nhờ bộ giải mã trong thử nghiệm, không quan tâm đến giá trị no_output_of_prior_pics_flag. Mặc dù việc thiết đặt NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

(iii) Theo cách khác, NoOutputOfPriorPicsFlag được thiết đặt bằng no_output_of_prior_pics_flag.

(B) Giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HRD như sau:

(i) Nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh trong DPB được làm rỗng mà không cần kết xuất các hình ảnh chúng chứa và mức đầy DPB được thiết đặt bằng 0.

(ii) Theo cách khác, (NoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm rỗng bởi yêu cầu lặp đi lặp lại quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 và mức đầy DPB được thiết đặt bằng 0.

(iii) Theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB được giảm bớt một. Biến currLayerId được thiết đặt bằng nuh_layer_id của hình ảnh giải mã hiện thời.

Các biến MaxNumReorderPics[TargetOp][currLayerId][HighestTid],
MaxLatencyIncreasePlus1[TargetOp][currLayerId][HighestTid],

MaxLatencyPictures[TargetOp][currLayerId][HighestTid],
 MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid] được suy ra như sau dựa trên điểm hoạt động hiện thời trong thử nghiệm:

(1) Nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() có mặt cho điểm hoạt động trong thử nghiệm TargetOp, MaxNumReorderPics[TargetOp][currLayerId][HighestTid] được thiết đặt bằng vps_max_num_reorder_pics[HighestTid] khi currLayerId bằng 0 hoặc được thiết đặt bằng vps_max_num_reorder_pics[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động trong thử nghiệm khi currLayerId lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không có mặt cho điểm hoạt động trong thử nghiệm, thì MaxNumReorderPics[TargetOp][currLayerId][HighestTid] được thiết đặt bằng sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho giá trị currLayerId.

(2) Nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() có mặt cho điểm hoạt động đang thttttử nghiệm TargetOp, thì MaxLatencyIncreasePlus1[TargetOp][currLayerId][HighestTid] được thiết đặt bằng vps_max_latency_increase_plus1[HighestTid] khi currLayerId bằng 0 hoặc được thiết đặt bằng vps_max_latency_increase_plus1[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động trong thử nghiệm khi currLayerId lớn hơn 0. Nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() có mặt cho điểm hoạt động trong thử nghiệm, MaxLatencyPictures[TargetOp][currLayerId][HighestTid] được thiết đặt bằng VpsMaxLatencyPictures[HighestTid] khi currLayerId bằng 0 hoặc được thiết đặt bằng VpsMaxLatencyPictures[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động trong thử nghiệm khi currLayerId lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không có mặt cho điểm hoạt động trong thử nghiệm, thì MaxLatencyIncreasePlus1[TargetOp][currLayerId][HighestTid] được thiết đặt bằng sps_max_latency_increase_plus1[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS lớp hoạt động cho giá trị currLayerId và MaxLatencyPictures[TargetOp][currLayerId][HighestTid] được thiết đặt bằng SpsMaxLatencyPictures[HighestTid] được suy ra từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho giá trị currLayerId.

(3) Nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` có mặt đối với điểm hoạt động được chọn trong thử nghiệm `TargetOp`, thì `MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid]` được thiết đặt bằng `vps_max_dec_pic_buffering_minus1[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `vps_max_dec_pic_buffering_minus1[TargetOp][CurrLayerId][HighestTid]` cho `currLayerId` cho điểm hoạt động trong thử nghiệm khi `currLayerId` lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` không có mặt cho điểm hoạt động trong thử nghiệm, `MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid]` được thiết đặt bằng `sps_max_dec_pic_buffering_minus1[HighestTid]` từ SPS hoạt động (khi `currLayerId` bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `currLayerId`.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt mức đầy DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm rỗng, cho tới khi không có điều kiện nào sau đây là đúng:

- (1) Số lượng hình ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB được đánh dấu là “cần kết xuất” lớn hơn `MaxNumReorderPics[TargetOp][CurrLayerId][HighestTid]`.
- (2) Nếu `MaxLatencyIncreasePlus1[TargetOp][CurrLayerId][HighestTid]` không bằng 0 và có ít nhất một ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan `PicLatencyCount[currLayerId]` lớn hơn hoặc bằng `MaxLatencyPictures[TargetOp][CurrLayerId][HighestTid]`.
- (3) Số lượng hình ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB lớn hơn hoặc bằng `MaxDecPicBuffering[TargetOp][CurrLayerId][HighestTid]`.

Các quy trình được xác định theo khoản này xảy ra một cách tức thì khi đơn vị giải mã sau cùng của đơn vị truy cập n có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB.

Biến `currLayerId` được thiết đặt bằng `nuh_layer_id` của hình ảnh giải mã hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là “cần kết xuất” và có giá trị `nuh_layer_id` bằng `currLayerId`, thì biến liên quan `PicLatencyCount[currLayerId]` được thiết đặt bằng `PicLatencyCount[currLayerId] + 1`.

Hình ảnh hiện thời được coi là đã được giải mã sau khi đơn vị giải mã sau cùng của hình ảnh giải mã. Hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống trong DPB và quy định sau đây được áp dụng:

(A) Nếu hình ảnh giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu là “cần kết xuất” và Biến liên quan PicLatencyCount[currLayerId] được thiết đặt bằng 0.

(B) Theo cách khác, (hình ảnh giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu là “không cần kết xuất”.

Hình ảnh giải mã hiện thời được đánh dấu “sử dụng để tham chiếu ngắn hạn”.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình “đệm” được xác định theo khoản mục F.13.5.2.4 được hủy bỏ lặp đi lặp lại cho tới khi không có điều kiện nào sau đây là đúng.

(A) Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất” lớn hơn MaxNumReorderPics[TargetOp][CurrLayerId][HighestTid].

(B) MaxLatencyIncreasePlus1[TargetOp][CurrLayerId][HighestTid] không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan PicLatencyCount[currLayerId] lớn hơn hoặc bằng MaxLatencyPictures[TargetOp][CurrLayerId][HighestTid].

Trong trường hợp khác, các biến MaxNumReorderPics[currLayerId][HighestTid], MaxLatencyIncreasePlus1[currLayerId][HighestTid], MaxLatencyPictures[currLayerId][HighestTid], MaxDecPicBufferingMinus1[currLayerId][HighestTid] có thể được suy ra như sau:

(1) Nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() có mặt cho điểm hoạt động trong thử nghiệm, thì MaxNumReorderPics[currLayerId][HighestTid] được thiết đặt bằng vps_max_num_reorder_pics[HighestTid] khi currLayerId bằng 0 hoặc được thiết đặt bằng vps_max_num_reorder_pics[CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động trong thử nghiệm khi currLayerId lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không có mặt cho điểm hoạt động trong thử nghiệm thì MaxNumReorderPics[currLayerId][HighestTid] được thiết đặt bằng sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho giá trị currLayerId.

(2) Nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` có mặt cho điểm hoạt động trong thử nghiệm, thì `MaxLatencyIncreasePlus1[currLayerId][HighestTid]` được thiết đặt bằng `vps_max_latency_increase_plus1[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `vps_max_latency_increase_plus1[CurrLayerId][HighestTid]` đối với `currLayerId` cho điểm hoạt động trong thử nghiệm khi `currLayerId` lớn hơn 0. Nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` có mặt cho điểm hoạt động trong thử nghiệm, thì `MaxLatencyPictures[currLayerId][HighestTid]` được thiết đặt bằng `VpsMaxLatencyPictures [HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `VpsMaxLatencyPictures [CurrLayerId][HighestTid]` đối với `currLayerId` cho điểm hoạt động trong thử nghiệm khi `currLayerId` lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` không có mặt cho điểm hoạt động trong thử nghiệm, thì `MaxLatencyIncreasePlus1[currLayerId][HighestTid]` được thiết đặt bằng `sps_max_latency_increase_plus1[HighestTid]` của SPS hoạt động (khi `currLayerId` bằng 0) hoặc SPS lớp hoạt động cho giá trị `currLayerId` và `MaxLatencyPictures[currLayerId][HighestTid]` được thiết đặt bằng `SpsMaxLatencyPictures [HighestTid]` được suy ra từ SPS hoạt động (khi `currLayerId` bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `currLayerId`.

(3) Nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` có mặt đối với điểm hoạt động được chọn trong thử nghiệm, thì `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` được thiết đặt bằng `vps_max_dec_pic_buffering_minus1[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid]` đối với `currLayerId` cho điểm hoạt động trong thử nghiệm khi `currLayerId` lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động `op_dpb_info_parameters()` không có mặt cho điểm hoạt động trong thử nghiệm, `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` được thiết đặt bằng `sps_max_dec_pic_buffering_minus1[HighestTid]` từ SPS hoạt động (khi `currLayerId` bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `currLayerId`.

Khi một hoặc nhiều điều kiện sau đây là đúng, thì quy trình “đệm” được xác định theo khoản mục F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt

mức đầy DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm rỗng, cho tới khi không có điều kiện nào sau đây là đúng:

- (1) Số lượng hình ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB được đánh dấu là “cần kết xuất” lớn hơn `MaxNumReorderPics[CurrLayerId][HighestTid]`.
- (2) Nếu `MaxLatencyIncreasePlus1[CurrLayerId][HighestTid]` không bằng 0 và có ít nhất một ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan `PicLatencyCount[currLayerId]` lớn hơn hoặc bằng `MaxLatencyPictures[CurrLayerId][HighestTid]`.
- (3) Số lượng hình ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB lớn hơn hoặc bằng `MaxDecPicBuffering[CurrLayerId][HighestTid]`.

Các quy trình được xác định theo khoản này xảy ra một các tức thì khi đơn vị giải mã sau cùng của đơn vị truy cập n có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB.

Biến `currLayerId` được thiết đặt bằng `nuh_layer_id` của hình ảnh giải mã hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là “cần kết xuất” và có giá trị `nuh_layer_id` bằng `currLayerId`, thì biến liên quan `PicLatencyCount[currLayerId]` được thiết đặt bằng `PicLatencyCount[currLayerId] + 1`.

Hình ảnh hiện thời được coi là đã được giải mã sau khi đơn vị giải mã sau cùng của hình ảnh giải mã. Hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống trong DPB và quy định sau đây được áp dụng:

(A) Nếu hình ảnh giải mã hiện thời có `PicOutputFlag` bằng 1, thì nó được đánh dấu là “cần kết xuất” và biến liên quan `PicLatencyCount[currLayerId]` được thiết đặt bằng 0.

(B) Theo cách khác, (hình ảnh giải mã hiện thời có `PicOutputFlag` bằng 0), thì nó được đánh dấu là “không cần kết xuất”.

Hình ảnh giải mã hiện thời được đánh dấu là “sử dụng để tham chiếu ngắn hạn”.

Khi một hoặc nhiều điều kiện sau đây là đúng, thì quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 được hủy bỏ lặp đi lặp lại cho tới khi không có điều kiện nào sau đây là đúng.

(A) Số lượng hình ảnh với `nuh_layer_id` bằng `currLayerId` trong DPB được đánh dấu là “cần kết xuất” lớn hơn `MaxNumReorderPics[CurrLayerId][HighestTid]`.

(B) MaxLatencyIncreasePlus1[CurrLayerId][HighestTid] không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, biến liên quan PicLatencyCount[currLayerId] lớn hơn hoặc bằng MaxLatencyPictures[CurrLayerId][HighestTid].

Quy trình “đệm” gồm các bước theo thứ tự sau đây:

(A) Các hình ảnh mà để kết xuất trước tiên được chọn làm các hình ảnh có giá trị PicOrderCntVal nhỏ nhất trong số tất cả các hình ảnh trong DPB được đánh dấu là “cần kết xuất”.

(B) Các hình ảnh này được cắt xén, sử dụng của sổ cắt xén phù hợp được xác định trong SPS hoạt động đối với hình ảnh có nuh_layer_id bằng 0 hoặc trong SPS lớp hoạt động đối với giá trị nuh_layer_id bằng giá trị của hình ảnh, các hình ảnh được cắt xén được kết xuất theo thứ tự tăng dần của nuh_layer_id và các hình ảnh này được đánh dấu là “không cần kết xuất”.

(C) Mỗi bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không sử dụng để tham chiếu” và nó bao gồm một trong các hình ảnh đã được cắt xén và đầu ra được làm rỗng.

Phân mở rộng VPS có thể có các cải biến thêm, nếu muốn.

Như được thể hiện trên Fig.40, cải biến thêm có thể bao gồm các tham số DPB đang được gửi trong phân mở rộng VPS đối với các tập hợp lớp kết xuất thay vì đối với các điểm hoạt động, trong đó oops_dpb_info_parameters(j) được thể hiện trên Fig.41.

num_dpb_info_parameters xác định số lượng cấu trúc cú pháp op_dpb_parameters() có mặt trong RBSP của phân mở rộng VPS. Các bộ giải mã num_dpb_info_parameters nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả giá trị đầu cuối.

output_point_layer_set_idx[i] xác định chỉ số vào danh mục của các tập hợp lớp kết xuất đích mà cấu trúc cú pháp oops_dpb_info_parameters() thứ i trong phân mở rộng VPS tác động đến.

Giá trị output_point_layer_set_idx[i] nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả giá trị đầu cuối. Yêu cầu về tính phù hợp của dòng bit là output_point_layer_set_idx [i] không bằng output_point_layer_set_idx [j] cho j bất kỳ không bằng i.

Như được thể hiện trên Fig.42, `oop_dpb_info_paremters(c)` có thể còn được biến đổi, trong đó cú pháp trong phần mở rộng VPS có thể như được thể hiện trên Fig.43.

Như được thể hiện trên Fig.44, `oop_dpb_info_paremters(c)` có thể còn được biến đổi, trong đó cú pháp trong phần mở rộng VPS có thể như được thể hiện trên Fig.45 hoặc Fig.46.

Một phương án thay đổi làm ví dụ đối với cú pháp trong phần mở rộng VPS là

Bảng 18

<code>for(j = 0; j <= vps_max_layer_id; j++)</code>
<code>oop_dpb_info_parameters(j)</code>

có thể được thay đổi thành

Bảng 19

<code>for(j = 0; j <= vps_max_layers_minus1; j++)</code>
<code>oop_dpb_info_parameters(j)</code>

`vps_max_layer_id` xác định giá trị `nuh_layer_id` tối đa của tất cả các đơn vị NAL trong CVS. `vps_max_layers_minus1`, xác định lượng tối đa của các lớp mà chúng có thể có mặt trong CVS, trong đó một lớp có thể ví dụ là lớp có thể mở rộng không gian, lớp có thể mở rộng chất lượng, bố cục bề mặt hoặc độ sâu.

Phương án thay thế làm ví dụ khác đối với cú pháp trong phần mở rộng VPS là

Bảng 20

<code>for(j = 0; j <= vps_max_layer_id; j++)</code>
<code>oop_dpb_info_parameters(j)</code>

có thể được thay đổi thành

Bảng 21

<code>for(j = 0; j < numOutputLayers; j++)</code>
<code>oop_dpb_info_parameters(j)</code>

trong đó `numOutputLayers` đối với chỉ số tập hợp lớp kết xuất được chọn `oplsIdx` được suy ra là:

Bảng 22

<code>for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)</code>
<code>if(output_layer_flag[opLsIdx][k])</code>
<code>targetOpLayerIdList [numOutputLayers++] = layer_id_in_nuh[k]</code>

Phương án thay thế làm ví dụ khác đối với cú pháp trong phần mở rộng VPS là

Bảng 23

for(j = 0; j <= vps_max_layer_id; j++)
oop_dpb_info_parameters(j)

có thể được thay đổi thành

Bảng 24

for(j = 0; j < numDecodedLayers; j++)
oop_dpb_info_parameters(j)

trong đó numOutputLayers cho opIsIdx được chọn được suy ra là:

Bảng 25

for(k=0, numOutputLayers=0; k<=vps_max_layer_id; k++)
if(output_layer_flag[opIsIdx][k])
targetOpLayerIdList [numOutputLayers++] = layer_id_in_nuh[k]

Sau đó danh mục nhận diện lớp giải mã đích targetDLayerIdList và numDecodedLayers đối với opIsIdx được chọn được suy ra là:

Bảng 26

for(m=0, numDecodedLayers=0; m< numOutputLayers; m++) {
for(n=0; n< NumDirectRefLayers[LayerIdInVps[targetOpLayerIdList[m]]]; n++) {
rLid=RefLayerId[LayerIdInVps[targetOpLayerIdList[m]]][n]
if(rLid not included in targetDLayerIdList[0,..., numDecodedLayers])
targetDLayerIdList[numDecodedLayers++] = rLid;
}
}

Theo một phương án cờ bổ sung có thể được truyền tín hiệu để chỉ báo nếu oop_dpb_information_parameters được truyền tín hiệu cho lớp bổ sung cụ thể như sau:

Bảng 27

<code>for(j = 0; j <= vps_max_layer_id; j++) {</code>	
<code> vps_layer_info_present_flag[j]</code>	<code>u(1)</code>
<code> if(vps_layer_info_present_flag)</code>	
<code> oop_dpb_info_parameters(j)</code>	
<code>}</code>	

`vps_layer_info_present_flag[j]` bằng 1 xác định rằng `oop_dpb_info_parameters` có mặt đối với lớp thứ `j` cho tập hợp lớp kết xuất cụ thể. `vps_layer_info_present_flag[j]` bằng 0 xác định rằng `oop_dpb_info_parameters` không có mặt đối với lớp thứ `j` cho tập hợp lớp kết xuất cụ thể.

Theo phương án khác, `num_dpb_info_parameters` của các bộ giải mã nằm trong khoảng từ 0 đến 1024, bao gồm cả giá trị đầu cuối. Theo phương án khác nữa, một số cố định khác có thể được sử dụng thay cho 1024.

Theo phương án thay thế `output_point_layer_set_idx[i]` nằm trong khoảng từ 0 đến 1023, bao gồm cả giá trị đầu cuối.

Như được thể hiện trên Fig.47, phần mở rộng VPS cải biến khác và `layer_dpb_info(i)` có thể được sử dụng nếu cá tham số DPB được gửi trong phần mở rộng VPS một cách độc lập cho mỗi lớp của các tập hợp lớp kết xuất và các điểm hoạt động.

Như được thể hiện trên Fig.48, `layer_dpb_info(i)` cải biến có thể được sử dụng trong đó phần tử cú pháp `vps_max_sub_layer_minus1` được truyền tín hiệu từ VPS được sử dụng cho tất cả các lớp và không được truyền tín hiệu một cách riêng trong `oop_dpb_info_parameters(id)/ op_dpb_info_parameters( id)`.

Như được thể hiện trên Fig.49A và Fig.49B, `vps_extension` được biến đổi làm ví dụ được minh họa. Phần mở rộng VPS bao gồm cú pháp mới, gọi là, `max_sub_layers_vps_predict_flag[i]`, `max_sub_layers_vps_minus1[i]`, `num_dpb_info_parameters`, `output_point_layer_set_idx[i]`, `oop_dpb_maxbuffering_parameters(i)` và `layer_dpb_info_parameters(i)`. `num_output_layer_sets` xác định số lượng tập hợp lớp, mà đối với nó, các lớp đầu ra được xác định có `output_layer_set_index[i]` và `output_layer_flag[lsIdx][j]`. Khi không có mặt, thì giá trị `num_output_layer_sets` được suy ra bằng 0. Tập hợp lớp mô tả các lớp đầu ra là tập hợp lớp đầu ra.

'max_sub_layers_vps_predict_flag'[i] bằng 1 xác định rằng max_sub_layers_vps_minus1[i] được suy ra bằng max_sub_layers_vps_minus1[i - 1].

max_sub_layers_vps_predict_flag[i] bằng 0 xác định rằng max_sub_layers_vps_minus1[i] đã được truyền tín hiệu một cách rõ ràng. Giá trị max_sub_layers_vps_predict_flag[0] được suy ra bằng 0.

'max_sub_layers_vps_minus1'[i] + 1 xác định lượng tối đa của các lớp phụ tạm thời có thể có mặt trong CVS cho lớp có nuh_layer_id bằng i. Giá trị max_vps_sub_layers_vps_minus1[i] nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu cuối. Trong một số trường hợp max_sub_layers_vps_minus1[i] được sử dụng cho sự suy ra của phần tử cú pháp SPS sps_max_sub_layers_minus1. Khi max_sub_layers_vps_predict_flag[i] bằng 1, thì max_sub_layers_vps_minus1[i] được suy ra bằng max_sub_layers_vps_minus1[i - 1]. Giá trị max_sub_layers_vps_minus1[0] được suy ra bằng vps_max_sub_layers_minus1.

Biến MaxSubLayers[setId] đối với setId nằm trong khoảng từ 0 đến num_dpb_info_parameters- 1, bao gồm cả giá trị đầu cuối, được suy ra như sau:

Bảng 28

for(setId = 0; setId < num_dpb_info_parameters; setId++) {
lsIdx =
output_layer_set_idx_minus1[output_point_layer_set_idx[setId]] + 1
highestLayerId =
layerSetLayerIdList[lsIdx][numLayersInIdList[lsIdx]-1]
MaxSubLayers[setId] =
(max_sub_layers_vps_minus1[highestLayerId]+1)
}

'num_dpb_info_parameters' xác định số lượng cấu trúc cú pháp oop_dpb_maxbuffering_parameters(i) có mặt trong RBSP của phần mở rộng VPS. num_dpb_info_parameters của các bộ giải mã nằm trong khoảng từ 0 đến numOutputLayersets, bao gồm cả giá trị đầu cuối.

'output_point_layer_set_idx'[i] xác định chỉ số vào danh mục của các tập hợp lớp kết xuất mà cấu trúc cú pháp oop_dpb_maxbuffering_parameters(i) thứ i trong phần mở rộng VPS áp dụng cho nó.

Giá trị `output_point_layer_set_idx[i]` nằm trong khoảng từ 0 đến `numOutputLayersets`, bao gồm cả giá trị đầu cuối. Yêu cầu về tính phù hợp của dòng bit là `output_point_layer_set_idx[i]` không bằng `output_point_layer_set_idx[j]` cho `j` bất kỳ không bằng `i`.

Như được thể hiện trên Fig.50, `oop_dpb_maxbuffering_parameters` xác định `'sub_layer_vps_buf_info_present_flag'[i]`, `'max_vps_dec_pic_buffering_minus1'[i][j]`.

`'sub_layer_vps_buf_info_present_flag'[i]` bằng 1 xác định rằng `max_vps_dec_pic_buffering_minus1[i][j]` là có mặt đối với các lớp phụ `MaxSubLayers[i]`. `sub_layer_vps_buf_info_present_flag[i]` bằng 0 xác định rằng các giá trị của `max_vps_dec_pic_buffering_minus1[i][MaxSubLayers[i] - 1]` áp dụng cho tất cả các lớp phụ.

`'max_vps_dec_pic_buffering_minus1'[i][j] + 1` xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS cho lớp có `nuh_layer_id` bằng `highestLayerId` trong tập hợp lớp kết xuất kết hợp với chỉ số `i` trong các bộ của các bộ đệm lưu trữ hình ảnh khi `HighestTid` bằng `j`. Giá trị `max_vps_dec_pic_buffering_minus1[i][j]` nằm trong khoảng từ 0 đến `MaxDpbSize - 1` (như được xác định trong khoản mục A.4), bao gồm cả giá trị đầu cuối. Khi `j` lớn hơn 0, `max_vps_dec_pic_buffering_minus1[i][j]` sẽ lớn hơn hoặc bằng `max_vps_dec_pic_buffering_minus1[i][j] - 1`. Trong một số trường hợp `max_vps_dec_pic_buffering_minus1[i][j]` được sử dụng để suy ra của các giá trị của các phần tử cú pháp SPS `sps_max_dec_pic_buffering_minus1[j]`. Khi `max_vps_dec_pic_buffering_minus1[i][j]` không có mặt cho `i` nằm trong khoảng từ 0 đến `MaxSubLayers[i] - 2`, bao gồm cả giá trị đầu cuối, do `sub_layer_vps_buf_info_present_flag[i]` bằng 0, thì suy ra `max_vps_dec_pic_buffering_minus1[i][MaxSubLayers[i] - 1]`.

Giá trị `max_vps_dec_pic_buffering_minus1[0][j]` đối với mỗi giá trị của `j` được suy ra bằng `vps_max_dec_pic_buffering_minus1[j]`.

Như được thể hiện trên Fig.51, `layer_dpb_info_parameters` xác định `'sub_layer_vps_ordering_info_present_flag'[i]`, `max_vps_num_reorder_pics[i][j]`, `'max_vps_latency_increase_plus1'[i][j]`.

'sub_layer_vps_ordering_info_present_flag'[i] bằng 1 xác định rằng max_vps_num_reorder_pics[i][j] và max_vps_latency_increase_plus1[i][j] có mặt đối với các lớp phụ max_sub_layers_vps_minus1 + 1.

sub_layer_vps_ordering_info_present_flag[i] bằng 0 xác định rằng các giá trị của max_vps_num_reorder_pics[i][vps_max_sub_layers_minus1] và max_vps_latency_increase_plus1[i][max_sub_layers_vps_minus1] áp dụng cho tất cả các lớp phụ.

'max_vps_num_reorder_pics'[i][j] chỉ báo lượng tối đa cho phép của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng i theo thứ tự giải mã và tiếp theo hình ảnh đó theo thứ tự kết xuất khi HighestTid bằng j. Giá trị max_vps_num_reorder_pics[i][j] nằm trong khoảng từ 0 đến max_vps_dec_pic_buffering_minus1[i][j], bao gồm cả giá trị đầu cuối. Khi j lớn hơn 0, thì max_vps_num_reorder_pics[i][j] sẽ lớn hơn hoặc bằng max_vps_num_reorder_pics[i][j-1]. Trong một số trường hợp max_vps_num_reorder_pics[i][j] được sử dụng để suy ra của các giá trị của phần tử cú pháp SPS sps_max_num_reorder_pics[j]. Khi max_vps_num_reorder_pics[i][j] không có mặt cho i nằm trong khoảng từ 0 đến max_sub_layers_vps_minus1[i] - 1, bao gồm cả giá trị đầu cuối, do sub_layer_vps_ordering_info_present_flag[i] bằng 0, thì suy ra max_vps_num_reorder_pics[i][max_sub_layers_vps_minus1[i]].

max_vps_latency_increase_plus1[i][j] không bằng 0 được sử dụng để tính toán giá trị VpsMaxLatencyPictures[i][j], xác định lượng tối đa của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS cho lớp có nuh_layer_id bằng i theo thứ tự kết xuất và tiếp theo hình ảnh đó theo thứ tự giải mã khi HighestTid bằng j.

Khi max_vps_latency_increase_plus1[i][j] không bằng 0, thì giá trị VpsMaxLatencyPictures[i][j] được xác định như sau:

$$\text{VPSMaxLatencyPictures}[i][j] = \text{max_vps_num_reorder_pics}[i][j] + \text{max_vps_latency_increase_plus1}[i][j] - 1$$

Khi max_vps_latency_increase_plus1[i][j] bằng 0, thì không có giới hạn tương ứng nào được thể hiện.

Giá trị max_vps_latency_increase_plus1[j][k] nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu cuối. Trong một số trường hợp max_vps_latency_increase_plus1[i][j]

được sử dụng để suy ra của các giá trị của các phần tử cú pháp SPS $\text{sps_max_latency_increase_plus1}[j]$. Khi $\text{max_vps_latency_increase_plus1}[i][j]$ không có mặt cho i nằm trong khoảng từ 0 đến $\text{max_sub_layers_vps_minus1}[i] - 1$, bao gồm cả giá trị đầu cuối, do $\text{sub_layer_vps_ordering_info_present_flag}[i]$ bằng 0, thì suy ra $\text{max_vps_latency_increase_plus1}[i][\text{max_sub_layers_vps_minus1}[i]]$.

Như được thể hiện trên Fig.52, vps_extension được biến đổi làm ví dụ được minh họa. Phần mở rộng VPS bao gồm cải biến nữa về cú pháp trên Fig.49 với cú pháp mới, gọi là, $\text{sub_layer_vps_buf_info_predict_flag}[i]$, $\text{sub_layer_vps_ordering_info_predict_flag}[i]$ mà chúng được truyền tín hiệu một cách có điều kiện. $\text{sub_layer_vps_buf_info_predict_flag}[i]$ bằng 1 xác định rằng $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ được suy ra bằng $\text{max_vps_dec_pic_buffering_minus1}[i - 1][j]$ đối với mỗi giá trị của j .

$\text{sub_layer_vps_buf_info_predict_flag}[i]$ bằng 0 xác định rằng $\text{max_vps_dec_pic_buffering_minus1}[i][j]$ cho ít nhất một giá trị j đã được truyền tín hiệu một cách rõ ràng. $\text{sub_layer_vps_buf_info_predict_flag}[0]$ được suy ra bằng 0. Khi không có mặt, thì $\text{sub_layer_vps_buf_info_predict_flag}[i]$ được suy ra bằng 0.

$\text{sub_layer_vps_ordering_info_predict_flag}[i]$ bằng 1 xác định rằng các phần tử cú pháp $\text{sub_layer_vps_ordering_info_present_flag}[i]$, $\text{max_vps_num_reorder_pics}[i][j]$ và $\text{max_vps_latency_increase_plus1}[i][j]$ lần lượt được suy ra bằng $\text{sub_layer_vps_ordering_info_present_flag}[i - 1]$, $\text{max_vps_num_reorder_pics}[i - 1][j]$ và $\text{max_vps_latency_increase_plus1}[i - 1][j]$. $\text{sub_layer_vps_ordering_info_predict_flag}[i]$ bằng 0 chỉ báo rằng các phần tử cú pháp $\text{sub_layer_vps_ordering_info_present_flag}[i]$, $\text{max_vps_num_reorder_pics}[i][j]$ và $\text{max_vps_latency_increase_plus1}[i][j]$ đã được truyền tín hiệu một cách rõ ràng. Khi không có mặt, thì giá trị $\text{sub_layer_vps_ordering_info_predict_flag}[i]$ được thiết đặt bằng 0.

Các phần tử cú pháp khác và ngữ nghĩa trên Fig.52 là giống như ngữ nghĩa trên Fig.49.

Như được thể hiện trên Fig.53, vps_extension được biến đổi làm ví dụ được minh họa. Phần mở rộng VPS bao gồm cải biến nữa về cú pháp trên Fig.49. Trên Fig.53 $\text{oop_dpb_maxbuffering_parameters}(i,j)$ được truyền tín hiệu cho mỗi lớp j đối với tập hợp lớp kết xuất cụ thể i so với Fig.49 truyền tín hiệu tập hợp đơn lẻ các tham số $\text{oop_dpb_maxbuffering_parameters}(i)$ cho lớp cao nhất id trong tập hợp lớp kết xuất i .

Như được thể hiện trên Fig.54, `oop_dpb_maxbuffering_parameters` xác định `sub_layer_vps_buf_info_present_flag[i][k]`, `max_vps_dec_pic_buffering_minus1[i][k][j]`.

`sub_layer_vps_buf_info_present_flag[i][k]` bằng 1 xác định rằng `max_vps_dec_pic_buffering_minus1[i][k][j]` có mặt đối với các lớp phụ `max_sub_layers_vps_minus1[k]`. `sub_layer_vps_buf_info_present_flag[i]` bằng 0 xác định rằng các giá trị của `max_vps_dec_pic_buffering_minus1[i][k][max_sub_layers_vps_minus1[k]]` áp dụng cho tất cả các lớp phụ.

`max_vps_dec_pic_buffering_minus1[i][k][j] + 1` xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS cho lớp có `nuh_layer_id` bằng `k` cho tập hợp lớp kết xuất kết hợp với chỉ số `i` trong các bộ của các bộ đệm lưu trữ hình ảnh khi `HighestTid` bằng `j`. Giá trị `max_vps_dec_pic_buffering_minus1[i][k][j]` nằm trong khoảng từ 0 đến `MaxDpbSize - 1` (như được xác định trong khoản mục A.4), bao gồm cả giá trị đầu cuối.

Khi `j` lớn hơn 0, `max_vps_dec_pic_buffering_minus1[i][k][j]` sẽ lớn hơn hoặc bằng `max_vps_dec_pic_buffering_minus1[i][k][j - 1]` được sử dụng để suy ra của các giá trị của các phần tử cú pháp SPS `sps_max_dec_pic_buffering_minus1[j]`.

Khi `max_vps_dec_pic_buffering_minus1[i][j]` không có mặt cho `i` nằm trong khoảng từ 0 đến `max_sub_layers_vps_minus1[k] - 1`, bao gồm cả giá trị đầu cuối, do `sub_layer_vps_buf_info_present_flag[i][k]` bằng 0, thì suy ra `max_vps_dec_pic_buffering_minus1[i][k][max_sub_layers_vps_minus1[k]]`.

Giá trị `max_vps_dec_pic_buffering_minus1[i][0][j]` đối với mỗi giá trị của `i` và `j` được suy ra bằng `vps_max_dec_pic_buffering_minus1[j]`.

Như được thể hiện trên Fig.55 `oop_dpb_maxbuffering_parameters` xác định `sub_layer_vps_buf_info_present_flag[i][k]`, `max_vps_dec_pic_buffering_minus1[i][k][j]`. Fig.55 là cú pháp biến đổi cho `oop_dpb_maxbuffering_parameters` so với cú pháp trên Fig.54 cho `oop_dpb_maxbuffering_parameters`.

Biến `MaxSubLayers[setId][k]` đối với `setId` nằm trong khoảng từ 0 đến `num_dpb_info_parameters - 1`, bao gồm cả giá trị đầu cuối, được suy ra như sau:

Bảng 29

<code>for(setId = 0; setId < num_dpb_info_parameters; setId++) {</code>
<code> for(k = 1; k <= vps_max_layers_minus1; i++) {</code>
<code> MaxSubLayers[setId][k] =(max_sub_layers_vps_minus1[k]+1)</code>
<code> }</code>
<code>}</code>

Trong trường hợp này, các tham số (i, k) của oop_dpb_maxbuffering_parameters sẽ được xác định như trên Fig.55.

Như được thể hiện trên Fig.56, vps_extension được biến đổi làm ví dụ được minh họa. Phân mở rộng VPS bao gồm cải biến nữa về cú pháp trên Fig.52 với cú pháp mới. Trên Fig.53 theo biến đổi này oop_dpb_maxbuffering_parameters(i,j) được truyền tín hiệu cho mỗi lớp j đối với tập hợp lớp kết xuất cụ thể i so với Fig.52 truyền tín hiệu tập hợp đơn lẻ các tham số oop_dpb_maxbuffering_parameters(i) cho lớp cao nhất id trong tập hợp lớp kết xuất i.

oop_dpb_maxbuffring_parameters(I,k) được thể hiện trên Fig.57.

Như được thể hiện trên Fig.58 oop_dpb_maxbuffering_parameters xác định sub_layer_vps_buf_info_present_flag[i][k], max_vps_dec_pic_buffering_minus1[i][k][j]. Fig.58 là cú pháp biến đổi cho oop_dpb_maxbuffering_parameters so với cú pháp trên Fig.57 cho oop_dpb_maxbuffering_parameters.

Biến MaxSubLayers[setId][k] đối với setId nằm trong khoảng từ 0 đến num_dpb_info_parameters - 1, bao gồm cả giá trị đầu cuối, được suy ra như sau:

Bảng 30

<code>for(setId = 0; setId < num_dpb_info_parameters; setId++) {</code>
<code> for(k = 1; k <= vps_max_layers_minus1; i++) {</code>
<code> MaxSubLayers[setId][k] =(max_sub_layers_vps_minus1[k]+1)</code>
<code> }</code>
<code>}</code>

Trong trường hợp này các tham số (i, k) của oop_dpb_maxbuffering_parameters sẽ được xác định như trên Fig.58.

Như được thể hiện trên Fig.59, vps_extension được biến đổi làm ví dụ được minh họa. Phân mở rộng VPS bao gồm cải biến nữa của cú pháp trên Fig.56 với cú pháp mới, gọi

là, cờ bổ sung `layer_dpb_info_parameters_presence_flag` mà chúng được truyền tín hiệu một cách có điều kiện. `layer_dpb_info_parameters_presence_flag` thực hiện truyền tín hiệu của `layer_dpb_info_parameters(i)` trong phần mở rộng VPS tùy ý.

`layer_dpb_info_parameters_presence_flag[i]` bằng 1 xác định rằng các phần tử cú pháp `sub_layer_vps_ordering_info_predict_flag[i]` và `layer_dpb_info_parameters(i)` có mặt đối với các lớp `vps_max_num_layers_minus1`. `layer_dpb_info_parameters_presence_flag[i]` bằng 0 xác định rằng các phần tử cú pháp `sub_layer_vps_ordering_info_predict_flag[i]` và `layer_dpb_info_parameters(i)` không có mặt đối với các lớp `vps_max_num_layers_minus1`.

`oop_dpb_maxbuffering_parameters(i,k)` được thể hiện trên Fig.60.

Như được thể hiện trên Fig.61 `oop_dpb_maxbuffering_parameters` xác định `sub_layer_vps_buf_info_present_flag[i][k]`, `max_vps_dec_pic_buffering_minus1[i][k][j]`. Fig.61 là cú pháp biến đổi cho `oop_dpb_maxbuffering_parameters` so với cú pháp trên Fig.60 cho `oop_dpb_maxbuffering_parameters`.

Biến `MaxSubLayers[setId][k]` đối với `setId` nằm trong khoảng từ 0 đến `num_dpb_info_parameters - 1`, bao gồm cả giá trị đầu cuối, được suy ra như sau:

Bảng 31

<code>for(setId = 0; setId < num_dpb_info_parameters; setId++) {</code>
<code> for(k = 1; k <= vps_max_layers_minus1; k++) {</code>
<code> MaxSubLayers[setId][k] =(max_sub_layers_vps_minus1[k]+1)</code>
<code> }</code>
<code>}</code>

Trong trường hợp này các tham số (i, k) của `oop_dpb_maxbuffering_parameters` sẽ được xác định như trên Fig.61.

Một phương án thay đổi làm ví dụ đối với cú pháp trong phần mở rộng VPS là

Bảng 32

<code>for(k = 1; k <= vps_max_layers_minus1; k++)</code>
<code> oop_dpb_maxbuffering_parameters(k)</code>

có thể được thay đổi thành

Bảng 33

<code>for(k = 0; k <= vps_max_layers_minus1; k++)</code>
<code> oop_dpb_maxbuffering_parameters(k)</code>

Do vậy chỉ số k có thể bắt đầu bằng 0 thay cho bằng 1.

Một phương án thay đổi làm ví dụ đối với cú pháp trong phần mở rộng VPS là

Bảng 34

for(k = 0; k <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(k)

có thể được thay đổi thành

Bảng 35

for(k = 0; k <= vps_max_layer_id; k++)
oop_dpb_maxbuffering_parameters(k)

vps_max_layer_id xác định giá trị tối đa nuh_layer_id của tất cả các đơn vị NAL trong CVS. vps_max_layers_minus1, xác định lượng tối đa của các lớp mà chúng có thể có mặt trong CVS, trong đó một lớp có thể ví dụ be lớp có thể mở rộng không gian, lớp có thể mở rộng chất lượng, bố cục bề mặt hoặc độ sâu.

Phương án thay thế làm ví dụ khác đối với cú pháp trong phần mở rộng VPS là

Bảng 36

for(k = 1; k <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(I,k)

có thể được thay đổi thành

Bảng 37

for(k = 1; k < numOutputLayers; k++)
oop_dpb_maxbuffering_parameters(I,k)

trong đó numOutputLayers được suy ra là

Bảng 38

for(setId = 0; setId < num_dpb_info_parameters; setId++) {
lsIdx =
output_layer_set_idx_minus1[output_point_layer_set_idx[setId]] + 1
numOutputLayers = [numLayersInIdList[lsIdx] - 1]
}

hoặc nó có thể được thay đổi thành

Bảng 39

```

for( k = 1; k <=
numLayersInIdList[output_layer_set_idx_minus1][output_
point_layer_set_idx[i]]-1; i++) {
    oop_dpb_maxbuffering_parameters(i, k)
}

```

Theo một phương án cờ bổ sung có thể được truyền tín hiệu để chỉ báo nếu oop_dpb_information_parameters được truyền tín hiệu cho lớp bổ sung cụ thể như sau:

Bảng 40

for(k = 1; k <= vps_max_layers_minus1; i++) {	
vps_layer_info_present_flag[k]	u(1)
oop_dpb_maxbuffering_parameters(i, k)	
}	

vps_layer_info_present_flag[k] bằng 1 xác định rằng oop_dpb_maxbuffering_parameters có mặt đối với lớp thứ k cho tập hợp lớp kết xuất cụ thể. vps_layer_info_present_flag[k] bằng 0 xác định rằng oop_dpb_maxbuffering_parameters không có mặt đối với lớp thứ k cho tập hợp lớp kết xuất cụ thể.

Các phần tử cú pháp của các đơn vị NAL không VCL (hoặc các giá trị mặc định đối với một số các phần tử cú pháp), cần thiết đối với HRD, được xác định trong các khoản có ý nghĩa của điều 7, các Phụ lục D và E.

Hai loại tập tham số HRD (các tham số HRD NAL và các tham số HRD VCL) được sử dụng. Các tập tham số HRD được truyền tín hiệu thông qua cấu trúc cú pháp hrd_parameters(), có thể là tiêu mục của cấu trúc cú pháp SPS hoặc cấu trúc cú pháp VPS.

Có thể cần đến nhiều thử nghiệm để kiểm tra tính phù hợp của dòng bit, được gọi là dòng bit trong thử nghiệm. Đối với mỗi thử nghiệm, áp dụng các bước theo thứ tự liệt kê sau đây:

(1) Tập hợp lớp kết xuất trong thử nghiệm, ký hiệu là TargetOpLs được chọn. Điểm hoạt động được tham chiếu trong TargetOpLs by output_layer_set_idx[] xác định điểm hoạt động trong thử nghiệm. Danh mục nhận diện lớp kết xuất OpLayerIdList của TargetOpLs gồm danh mục của các giá trị nuh_layer_id, theo thứ tự tăng dần lên của các giá trị nuh_layer_id, có mặt trong tập hợp phụ của dòng bit kết hợp với TargetOp và TargetOpLs, là tập hợp phụ của các giá trị nuh_layer_id có mặt trong dòng bit trong thử nghiệm. OpTid

của TargetOp bằng TemporalId cao nhất có mặt trong tập hợp phụ của dòng bit kết hợp với TargetOp.

(2) TargetDecLayerIdList được thiết đặt bằng danh mục nhận diện lớp giải mã đích targetDLayerIdList đối với các tập hợp lớp kết xuất được chọn TargetOpLs, HighestTid được thiết đặt bằng OpTid của TargetOp và quy trình trích xuất dòng bit phụ như được xác định trong khoản 10 được hủy bỏ với dòng bit trong thử nghiệm, HighestTid và TargetDecLayerIdList dưới dạng các dữ liệu đầu vào và đầu ra được gán cho BitstreamToDecode.

(3) Cấu trúc cú pháp hrd_parameters() và cấu trúc cú pháp sub_layer_hrd_parameters() có thể áp dụng cho TargetOp được chọn. Nếu TargetDecLayerIdList chứa tất cả các giá trị nuh_layer_id có mặt trong dòng bit trong thử nghiệm, cấu trúc cú pháp hrd_parameters() trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được xác định trong bản mô tả này) đã chọn. Theo cách khác, cấu trúc cú pháp hrd_parameters() trong SPS hoạt động (hoặc được tạo ra thông qua các phương tiện bên ngoài không được xác định trong bản mô tả này) áp dụng cho TargetOp được chọn. Trong cấu trúc cú pháp hrd_parameters() đã chọn, nếu BitstreamToDecode là dòng bit Loại I, thì cấu trúc cú pháp sub_layer_hrd_parameters(HighestTid) theo ngay sau điều kiện “if(vcl_hrd_parameters_present_flag)” được chọn và biến NalHrdModeFlag được thiết đặt bằng 0; theo cách khác, (BitstreamToDecode là dòng bit Loại II), thì cấu trúc cú pháp sub_layer_hrd_parameters(HighestTid) theo ngay sau hoặc điều kiện “if(vcl_hrd_parameters_present_flag)” (trong trường hợp này biến NalHrdModeFlag được thiết đặt bằng 0) hoặc điều kiện “if(NAL_hrd_parameters_present_flag)” (trong trường hợp này biến NalHrdModeFlag được thiết đặt bằng 1) đã chọn. Khi BitstreamToDecode là dòng bit Loại II và NalHrdModeFlag bằng 0, tất cả các đơn vị NAL không VCL trừ các đơn vị NAL dữ liệu của bộ lọc và tất cả các phần tử cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes và trailing_zero_8bits mà chúng tạo thành luồng byte từ luồng đơn vị NAL (như được xác định trong Phụ lục B), khi có mặt, được loại ra khỏi BitstreamToDecode và dòng bit còn lại được gán cho BitstreamToDecode.

Bộ giải mã phù hợp có thể đáp ứng tất cả các yêu cầu được xác định trong khoản này.

(1) Bộ giải mã đòi hỏi tính phù hợp với biên dạng, lớp và mức cụ thể phải có khả năng giải mã thành công tất cả các dòng bit có phù hợp với các yêu cầu về tính phù hợp của dòng bit được xác định trong khoản mục C.4, theo cách thức được xác định trong Phụ lục A, với điều

kiện là tất cả các VPS, SPS và PPS được đề cập trong các đơn vị NAL VCL và khoảng đệm thích hợp và các thông báo SEI định thời hình ảnh được chuyển đến bộ giải mã, một cách kịp thời, hoặc là theo dòng bit (nhờ các đơn vị NAL không VCL), hoặc nhờ các phương tiện bên ngoài không được xác định trong bản mô tả này.

(2) Khi dòng bit chứa các phần tử cú pháp có các giá trị mà chúng được xác định là dự phòng và nó được xác định rằng các bộ giải mã cần bỏ qua các giá trị của các phần tử cú pháp hoặc các đơn vị NAL chứa các phần tử cú pháp có các giá trị dự phòng và dòng bit theo cách khác phù hợp với bản mô tả này, bộ giải mã phù hợp sẽ giải mã dòng bit theo cùng cách thức giống như khi nó giải mã dòng bit phù hợp và bỏ qua các phần tử cú pháp hoặc các đơn vị NAL chứa các phần tử cú pháp có các giá trị dự phòng như đã xác định.

Có hai loại phù hợp của bộ giải mã: phù hợp định thời kết xuất và phù hợp thứ tự kết xuất.

Để kiểm tra tính phù hợp của bộ giải mã, các dòng bit phù hợp với biên dạng, lớp và mức được đòi hỏi, như được xác định trong khoản mục C.4 được tạo ra bởi bộ lập lịch trình luồng giả định (HSS - hypothetical stream scheduler) cho cả HRD và bộ giải mã trong thử nghiệm (DUT). Tất cả hình ảnh giải mã được cắt xén được kết xuất bởi HRD cũng sẽ được kết xuất bởi DUT, mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT sẽ là ảnh với PicOutputFlag bằng 1 và, đối với mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT, các giá trị của tất cả các mẫu được kết xuất phải bằng các giá trị của các mẫu được tạo ra bởi quy trình giải mã được xác định.

Đối với tính phù hợp của bộ mã hóa định thời kết xuất, HSS hoạt động như được mô tả trên đây, với các lịch trình được chọn chỉ từ tập hợp phụ của các giá trị SchedSelIdx, mà đối với nó, tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A đối với biên dạng, lớp và mức được xác định, hoặc với các lịch trình cung cấp “nội suy” như đã xác định dưới đây, mà đối với nó, tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A. Lịch trình cung cấp giống nhau được sử dụng cho cả HRD và DUT.

Khi các tham số HRD và các thông báo SEI về khoảng đệm có mặt với $cpb_cnt_minus1[HighTid]$ lớn hơn 0, bộ giải mã sẽ có khả năng giải mã dòng bit như được tạo ra từ sự vận hành HSS sử dụng lịch trình cung cấp “nội suy” được xác định là có tốc độ bit đỉnh r , kích thước CPBc(r) và khoảng trễ gỡ bỏ CPB ban đầu

Biểu thức 16

$$(f(r) \div r)$$

là như sau:

Bảng 41

$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) + (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]),$	(C-22)
$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha) * \text{CpbSize}[\text{SchedSelIdx} - 1],$	(C-23)
$f(r) = \alpha * \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] +$ $(1 - \alpha) * \text{InitCpbRemovalDelay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1]$	(C-24)

cho bất kỳ $\text{SchedSelIdx} > 0$ và r sao cho $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$ sao cho r và $c(r)$ nằm trong các giới hạn như được xác định trong Phụ lục A đối với tốc độ bit tối đa và kích thước bộ đệm đối với biên dạng, lớp và mức được xác định. $\text{InitCpbRemovalDelay}[\text{SchedSelIdx}]$ có thể khác biệt cho mỗi khoảng đệm và cần được tính toán lại.

Đối với tính phù hợp của bộ mã hóa định thời kết xuất, HRD như được mô tả trên đây được sử dụng và định thời (liên quan đến thời gian cung cấp bit thứ nhất) của kết xuất hình ảnh là giống nhau đối với cả HRD và DUT cho tới khoảng trễ cố định.

Để phù hợp bộ giải mã của thứ tự kết xuất, các quy định sau đây được áp dụng:

(1) HSS cung cấp dòng bit BitstreamToDecode cho DUT “theo yêu cầu” từ DUT, nghĩa là HSS cung cấp các bit (theo thứ tự giải mã) chỉ khi DUT cần nhiều bit hơn để tiếp tục quá trình xử lý của nó. Điều này có nghĩa là đối với thử nghiệm này, bộ đệm hình ảnh giải mã của DUT có thể nhỏ bằng kích thước của đơn vị giải mã lớn nhất.

(2) HDR cải biến như được mô tả dưới đây được sử dụng và HSS cung cấp dòng bit đến HRD nhờ một trong các lịch trình được xác định trong dòng bit BitstreamToDecode sao cho tốc độ bit và kích thước CPB được ràng buộc như được xác định trong Phụ lục A. Thứ tự kết xuất các hình ảnh là giống nhau đối với cả HRD và DUT.

(3) Kích thước CPB HRD được xác định là $\text{CPBSize}[\text{SchedSelIdx}]$ như được xác định trong khoản E.2.3, trong đó SchedSelIdx và các tham số HRD được chọn như được xác định trong khoản mục C.1. Kích thước DPB được xác định là $\text{sps_max_dec_pic_buffering_minus1}[\text{HighestTid}] + 1$ từ SPS hoạt động (khi nuh_layer_id đối với hình ảnh giải mã hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho giá trị nuh_layer_id của hình ảnh giải mã hiện thời.

Trong một số trường hợp nếu các tham số thông tin DPB của tập hợp lớp kết xuất `oop_dpb_maxbuffering_parameters()` có mặt đối với các tập hợp lớp kết xuất được chọn, Kích thước DPB được xác định là `max_vps_dec_pic_buffering_minus1[CurrLayerId][HighestTid]` trong đó `currLayerId` là `nuh_layer_id` của hình ảnh giải mã hiện thời. Theo cách khác, nếu các tham số thông tin DPB của tập hợp lớp kết xuất `oop_dpb_maxbuffering_parameters()` không có mặt đối với các tập hợp lớp kết xuất được chọn, kích thước DPB được xác định là `sps_max_dec_pic_buffering_minus1[HighestTid] + 1` từ SPS hoạt động (khi `nuh_layer_id` đối với hình ảnh giải mã hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `nuh_layer_id` của hình ảnh giải mã hiện thời.)

Thời gian gỡ bỏ khỏi CPB đối với HRD là thời gian tới của bit cuối cùng và việc giải mã là ngay lập tức. Hoạt động của DPB của HRD này là như được mô tả theo các khoản từ C.5.2 đến C.5.2.3.

Bộ đệm hình ảnh giải mã chứa các bộ đệm lưu trữ hình ảnh. Số lượng bộ đệm lưu trữ hình ảnh cho `nuh_layer_id` bằng 0 bắt nguồn từ SPS hoạt động. Số lượng bộ đệm lưu trữ hình ảnh cho mỗi giá trị `nuh_layer_id` không phải là 0 bắt nguồn từ SPS lớp hoạt động cho giá trị `nuh_layer_id` không phải 0 đó. Mỗi trong số các bộ đệm lưu trữ hình ảnh chứa hình ảnh giải mã mà nó được đánh dấu là “được sử dụng để tham chiếu” hoặc được giữ lại để kết xuất trong tương lai. Quy trình kết xuất và gỡ bỏ các hình ảnh khỏi DPB như được xác định trong khoản mục F.13.5.2.2 được hủy bỏ, tiếp theo là yêu cầu của quy trình giải mã hình ảnh, đánh dấu, đệm bổ sung và lưu trữ như được xác định theo khoản mục F.13.5.2.3. Quy trình “đệm” được xác định theo khoản mục F.13.5.2.4 và được hủy bỏ như được xác định theo các khoản mục F.13.5.2.2 và F.13.5.2.3.

Kết xuất và gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã của hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thì khi đơn vị giải mã thứ nhất của đơn vị truy cập có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB và tiếp tục diễn ra như sau.

Quy trình giải mã đối với RPS như được xác định trong khoản 8.3.2 được hủy bỏ.

(1) Nếu hình ảnh hiện thời là hình ảnh IRAP với `NoRaslOutputFlag` bằng 1 và với `nuh_layer_id` bằng 0 không phải là ảnh 0, các bước theo thứ tự như sau được áp dụng:

(A) Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm như sau:

(i) Nếu hình ảnh hiện thời là ảnh CRA, thì NoOutputOfPriorPicsFlag được thiết đặt bằng 1 (không quan tâm đến giá trị no_output_of_prior_pics_flag).

(ii) Theo cách khác, nếu giá trị pic_width_in_luma_samples, thì pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] được suy ra từ SPS hoạt động lần lượt khác với giá trị pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động đối với ảnh đứng trước, NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được thiết đặt bằng 1 nhờ bộ giải mã trong thử nghiệm, không quan tâm đến giá trị no_output_of_prior_pics_flag. Mặc dù việc thiết đặt NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

(iii) Theo cách khác, NoOutputOfPriorPicsFlag được thiết đặt bằng no_output_of_prior_pics_flag.

(B) Giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HRD theo cách sau:

(i) Nếu NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh trong DPB được làm rỗng mà không cần kết xuất các hình ảnh chúng chứa và mức đầy DPB được thiết đặt bằng 0.

(ii) Theo cách khác, (NoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm rỗng bởi yêu cầu lặp đi lặp lại quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 và mức đầy DPB được thiết đặt bằng 0.

(iii) Theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB được giảm bớt một. Biến currLayerId được thiết đặt bằng nuh_layer_id của hình ảnh giải mã hiện thời.

Các biến `MaxNumReorderPics[currLayerId][HighestTid]`, `MaxLatencyIncreasePlus1[currLayerId][HighestTid]`, `MaxLatencyPictures[currLayerId][HighestTid]`, `MaxDecPicBufferingMinus1[currLayerId][HighestTid]`, được suy ra như sau:

Nếu các tham số thông tin DPB của lớp `layer_dpb_info_parameters()` có mặt trong VPS, thì `MaxNumReorderPics[currLayerId][HighestTid]` được thiết đặt bằng `vps_max_num_reorder_pics[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `max_vps_num_reorder_pics[CurrLayerId][HighestTid]` cho `currLayerId` khi `currLayerId` lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của lớp `layer_dpb_info_parameters()` không có mặt, thì `MaxNumReorderPics[currLayerId][HighestTid]` được thiết đặt bằng `sps_max_num_reorder_pics[HighestTid]` từ SPS hoạt động (khi `currLayerId` bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `currLayerId`.

Nếu các tham số thông tin DPB của lớp `layer_dpb_info_parameters()` có mặt trong VPS, thì `MaxLatencyIncreasePlus1[currLayerId][HighestTid]` được thiết đặt bằng `vps_max_latency_increase_plus1[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `max_vps_latency_increase_plus1[CurrLayerId][HighestTid]` cho `currLayerId` khi `currLayerId` lớn hơn 0. Nếu các tham số thông tin DPB của lớp `layer_dpb_info_parameters()` có mặt trong VPS, thì `MaxLatencyPictures[currLayerId][HighestTid]` được thiết đặt bằng `SpsMaxLatencyPictures[HighestTid]` khi `currLayerId` bằng 0 hoặc được thiết đặt bằng `VpsMaxLatencyPictures[CurrLayerId][HighestTid]` cho `currLayerId` khi `currLayerId` lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của lớp `layer_dpb_info_parameters()` không có mặt đối với điểm hoạt động trong thử nghiệm, `MaxLatencyIncreasePlus1[currLayerId][HighestTid]` được thiết đặt bằng `sps_max_latency_increase_plus1[HighestTid]` của SPS hoạt động (khi `currLayerId` bằng 0) hoặc SPS lớp hoạt động cho giá trị `currLayerId` và `MaxLatencyPictures[currLayerId][HighestTid]` được thiết đặt bằng `SpsMaxLatencyPictures[currLayerId][HighestTid]` được suy ra từ SPS hoạt động (khi `currLayerId` bằng 0) hoặc từ SPS lớp hoạt động cho giá trị `currLayerId`.

Nếu các tham số thông tin DPB của điểm hoạt động `oop_dpb_maxbuffering_parameters()` có mặt đối với các tập hợp lớp kết xuất được chọn, thì `MaxDecPicBufferingMinus1[currLayerId][HighestTid]` được thiết đặt bằng

$\text{vps_max_dec_pic_buffering_minus1}[\text{HighestTid}]$ khi currLayerId bằng 0 hoặc được thiết đặt bằng $\text{max_vps_dec_pic_buffering_minus1}[\text{CurrLayerId}][\text{HighestTid}]$ cho currLayerId cho điểm hoạt động trong thử nghiệm khi currLayerId lớn hơn 0. Theo cách khác, nếu các tham số thông tin DPB của điểm hoạt động $\text{oop_dpb_maxbuffering_parameters}()$ không có mặt cho điểm hoạt động trong thử nghiệm, thì $\text{MaxDecPicBufferingMinus1}[\text{currLayerId}][\text{HighestTid}]$ được thiết đặt bằng $\text{sps_max_dec_pic_buffering_minus1}[\text{HighestTid}]$ từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho giá trị currLayerId .

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt mức đầy DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm rỗng, cho tới khi không có điều kiện nào sau đây là đúng:

(1) Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất” lớn hơn $\text{MaxNumReorderPics}[\text{CurrLayerId}][\text{HighestTid}]$.

(2) Nếu $\text{MaxLatencyIncreasePlus1}[\text{CurrLayerId}][\text{HighestTid}]$ không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB mà được đánh dấu là “cần kết xuất” mà đối với nó biến liên quan $\text{PicLatencyCount}[\text{currLayerId}]$ lớn hơn hoặc bằng $\text{MaxLatencyPictures}[\text{CurrLayerId}][\text{HighestTid}]$.

(3) Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB lớn hơn hoặc bằng $\text{MaxDecPicBuffering}[\text{CurrLayerId}][\text{HighestTid}]$.

Các quy trình được xác định trong khoản mục này xảy ra một cách tức thì khi đơn vị giải mã sau cùng của đơn vị truy cập n có chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB.

Biến currLayerId được thiết đặt bằng nuh_layer_id của hình ảnh giải mã hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là “cần kết xuất” và có giá trị nuh_layer_id bằng currLayerId , thì biến liên quan $\text{PicLatencyCount}[\text{currLayerId}]$ được thiết đặt bằng $\text{PicLatencyCount}[\text{currLayerId}] + 1$.

Hình ảnh hiện thời được coi là đã được giải mã sau khi đơn vị giải mã sau cùng của hình ảnh giải mã. Hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống trong DPB và quy định sau đây được áp dụng:

(A) Nếu hình ảnh giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu là “cần kết xuất” và biến liên quan $\text{PicLatencyCount}[\text{currLayerId}]$ được thiết đặt bằng 0.

(B) Theo cách khác, (hình ảnh giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu là “không cần kết xuất”.

Hình ảnh giải mã hiện thời được đánh dấu là “sử dụng để tham chiếu ngắn hạn”.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình “đệm” được xác định trong khoản mục F.13.5.2.4 được hủy bỏ lặp đi lặp lại cho tới khi không có điều kiện nào sau đây là đúng.

(A) Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất” lớn hơn MaxNumReorderPics[CurrLayerId][HighestTid].

(B) MaxLatencyIncreasePlus1[CurrLayerId][HighestTid] không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là “cần kết xuất”, mà đối với nó, thì biến liên quan PicLatencyCount[currLayerId] lớn hơn hoặc bằng MaxLatencyPictures[CurrLayerId][HighestTid].

Quy trình “đệm” gồm các bước theo thứ tự sau đây:

(A) Các hình ảnh mà để kết xuất trước tiên được chọn làm các hình ảnh có giá trị nhỏ nhất PicOrderCntVal của tất cả các hình ảnh trong DPB được đánh dấu là “cần kết xuất”.

(B) Các hình ảnh này được cắt xén, sử dụng cửa sổ cắt xén phù hợp được xác định trong SPS hoạt động đối với hình ảnh có nuh_layer_id bằng 0 hoặc trong SPS lớp hoạt động cho giá trị nuh_layer_id bằng giá trị đó của hình ảnh, các hình ảnh được cắt xén đó được kết xuất theo thứ tự tăng dần của nuh_layer_id và các hình ảnh này được đánh dấu là “không cần kết xuất”.

(C) Mỗi bộ đệm lưu trữ hình ảnh có chứa ảnh được đánh dấu là “không sử dụng để tham chiếu” và nó bao gồm một trong các hình ảnh đã được cắt xén và đầu ra được làm rỗng.

Như được thể hiện trên Fig. 62, cú pháp của tập tham số chuỗi (sps - sequence parameter set) cải biến làm ví dụ seq_parameter_set_rbsp được minh họa. SPS cải biến bao gồm sps_dpb_params_present_flag. Theo một phương án dựa trên giá trị cờ này và dựa trên giá trị nuh_layer_id một số các phần tử cú pháp (ví dụ sps_sub_layer_ordering_info_present_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], sps_max_latency_incease_plus1[i]) có thể không được truyền tín hiệu.

sps_dpb_params_present_flag bằng 0 xác định rằng, đối với SPS với nuh_layer_id >0 các phần tử cú pháp sps_sub_layer_ordering_info_present_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], sps_max_latency_increase_plus1[i] không có mặt trong SPS này. Giá trị của các tham số này được thiết đặt bằng giá trị của các tham số sub_layer_vps_buf_info_present_flag[i], max_vps_num_reorder_pics[i][j] và max_vps_latency_increase_plus1[i][j] được truyền tín hiệu trong phần mở rộng VPS hoạt động. sps_dpb_params_present_flag bằng 1 xác định rằng, cho SPS với nuh_layer_id >0 thì các phần tử cú pháp sps_sub_layer_ordering_info_present_flag, sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], sps_max_latency_increase_plus1[i] có mặt trong SPS này.

Theo cách khác, một hoặc nhiều phần tử cú pháp có thể được truyền tín hiệu sử dụng số lượng bit cố định đã biết thay cho u(v) thay cho ue(v). Ví dụ chúng có thể được truyền tín hiệu sử dụng u(8) hoặc u(16) hoặc u(32) hoặc u(64) v.v..

Theo cách khác, một hoặc nhiều phần tử cú pháp có thể được truyền tín hiệu với ue(v) hoặc sơ đồ mã hóa khác nào đó thay cho số lượng cố định các bit như mã hóa u(v).

Theo cách khác, tên của các phần tử cú pháp khác nhau và ngữ nghĩa của chúng có thể được sửa đổi bằng cách bổ sung plus1 hoặc plus2 hoặc bằng cách trừ bớt minus1 hoặc minus2 so với cú pháp và ngữ nghĩa đã được mô tả.

Theo phương án khác nữa, các phần tử cú pháp khác có thể được truyền tín hiệu mỗi ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ chúng có thể được truyền tín hiệu trong tiêu đề phân đoạn phiên, pps/ sps/ vps/ hoặc bất kỳ tập tham số khác hoặc phần danh định khác của dòng bit.

Theo các phương án khác nữa, tất cả các bối cảnh được xác định trong sáng chế liên quan đến các tập hợp lớp kết xuất có thể được áp dụng cho các điểm hoạt động kết xuất [2,3] và/ hoặc cho các điểm hoạt động [1].

Ví dụ 3

Phương pháp mã hóa video được bộc lộ. Phương pháp này bao gồm bắt đầu phân giải cú pháp tiêu đề phiên thứ nhất của hình ảnh hiện thời. Đã xác định được những bước nào được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) sẽ dựa vào ảnh và các bước nào sẽ dựa vào đơn vị truy cập (AU). Gỡ bỏ hình ảnh khỏi DPB được thực hiện. Kết xuất hình ảnh từ DPB được thực hiện. Bước giải mã và lưu trữ hình ảnh giải mã hiện thời trong DPB được

thực hiện. Hình ảnh giải mã hiện thời trong DPB được đánh dấu. Việc kết xuất hình ảnh bổ sung từ DPB cũng được thực hiện.

Trong một số cấu hình, việc gỡ bỏ và kết xuất từ DPB có thể dựa trên ít nhất một cờ kết xuất AU, như cờ kết xuất AU, cờ kết xuất không rasl của AU và/hoặc cờ của các hình ảnh trước không kết xuất của AU. Cờ kết xuất AU có thể được suy ra dựa trên các phần tử cú pháp được truyền tín hiệu trong dòng bit và các điều kiện khác. Các cờ AU đại diện cho các cờ được suy ra và được áp dụng ở mức của đơn vị truy cập (AU). Trong một số trường hợp, các cờ đó sẽ là các cờ khác với các cờ tương tự mà chúng được truyền tín hiệu hoặc suy ra ở mức ảnh, như cờ kết xuất hình ảnh (ví dụ, `pic_output_flag`), cờ kết xuất không rasl của hình ảnh (ví dụ, `NoRaslOutputFlag`) và/hoặc cờ không kết xuất hình ảnh của các hình ảnh trước (ví dụ, `no_output_of_prior_pics` flag).

Theo một cấu hình, cờ kết xuất AU thứ nhất có thể được suy ra từ và được sử dụng bởi DPB và cờ kết xuất AU thứ hai nhận được từ và được sử dụng bởi DPB. Theo cấu hình khác, tất cả các giá trị phần tử cú pháp của cờ kết xuất hình ảnh có thể được ràng buộc ở cùng giá trị đối với tất cả hình ảnh mã hoá AU khi kết xuất hình ảnh flag có mặt trong tiêu đề phiên thứ nhất. Trong một số cấu hình, tất cả các giá trị phần tử cú pháp của cờ kết xuất hình ảnh có thể được ràng buộc ở cùng giá trị đối với tất cả hình ảnh mã hoá AU khi không có cờ kết xuất hình ảnh nào có mặt trong tiêu đề phiên thứ nhất.

Theo một cấu hình, việc gỡ bỏ có thể dựa trên hình ảnh, việc kết xuất hình ảnh có thể dựa trên AU (đơn vị truy cập), việc lưu trữ và giải mã có thể dựa trên hình ảnh, việc đánh dấu có thể dựa trên hình ảnh và việc kết xuất hình ảnh bổ sung có thể dựa trên AU. Việc gỡ bỏ hình ảnh khỏi SPB có thể gỡ bỏ một hoặc nhiều hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời.

Theo cấu hình khác, việc gỡ bỏ có thể dựa trên hình ảnh và việc kết xuất hình ảnh, việc giải mã và lưu trữ, việc đánh dấu và kết xuất hình ảnh bổ sung có thể có thể dựa trên AU. Theo cấu hình khác nữa, việc gỡ bỏ, việc kết xuất hình ảnh, việc giải mã và lưu trữ, việc đánh dấu và kết xuất hình ảnh bổ sung có thể có thể dựa trên AU. Theo cấu hình khác, việc gỡ bỏ, việc giải mã và lưu trữ và việc đánh dấu có thể dựa trên hình ảnh và việc kết xuất hình ảnh và việc kết xuất hình ảnh bổ sung có thể có thể dựa trên AU. Theo cấu hình khác nữa, việc gỡ bỏ, việc kết xuất hình ảnh, việc giải mã và lưu trữ, việc đánh dấu và kết xuất hình ảnh bổ sung có thể dựa trên hình ảnh.

Việc đánh dấu hình ảnh giải mã hiện thời trong DPB có thể bao gồm bước đánh dấu tham chiếu và bước đánh dấu kết xuất. bước đánh dấu tham chiếu có thể dựa trên hình ảnh và bước đánh dấu kết xuất có thể có thể dựa trên AU. Mức đầy DPB có thể được tăng thêm một khi ảnh giải mã được lưu trữ trong DPB trong bộ đệm lưu trữ trống. Mức đầy DPB có thể được giảm bớt một khi một ảnh được kết xuất từ DPB. Mức đầy DPB có thể được theo dõi theo mỗi lớp. Mức đầy DPB cũng có thể được theo dõi đối với tập hợp lớp kết xuất.

DPB có thể bao gồm các bộ đệm ảnh xác định và quản lý một cách riêng đối với các hình ảnh giải mã có một hoặc nhiều độ phân giải khác nhau, độ sâu khác nhau và màu sắc khác nhau. DPB có thể bao gồm một nhóm bộ đệm lưu trữ hình ảnh chung. Hình ảnh giải mã có thể được lưu trữ trong các bộ đệm lưu trữ hình ảnh dựa trên ít nhất một kích thước, độ phân dải và độ sâu bit. Theo một cấu hình, ảnh giải mã có thể được lưu trữ trong một khe của bộ đệm ảnh của các bộ đệm lưu trữ hình ảnh đó.

Phương pháp này có thể được thực hiện nhờ bộ giải mã bên trong thiết bị điện tử phù hợp với chuẩn mã hóa video hiệu suất cao có thể mở rộng (SHVC - scalable high efficiency video coding). Phương pháp này cũng có thể được thực hiện nhờ bộ giải mã bên trong thiết bị điện tử phù hợp với chuẩn mã hóa video hiệu suất cao đa cảnh (MV-HEVC - multi-view high efficiency video coding).

Thiết bị điện tử được tạo cấu hình để mã hóa video cũng được bộc lộ. Thiết bị điện tử bao gồm bộ xử lý và bộ nhớ nối thông điện tử với bộ xử lý. Các lệnh trong bộ nhớ có thể thực hiện được để bắt đầu phân giải cú pháp tiêu đề phiên thứ nhất của hình ảnh hiện thời. Các lệnh trong bộ nhớ cũng có thể tiến hành xác định các bước nào được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) sẽ dựa vào ảnh và các bước nào sẽ dựa vào đơn vị truy cập (AU). Các lệnh trong bộ nhớ còn có thể tiến hành thực hiện việc gỡ bỏ hình ảnh khỏi DPB. Các lệnh trong bộ nhớ cũng có thể tiến hành thực hiện việc kết xuất hình ảnh từ DPB. Các lệnh trong bộ nhớ còn có thể tiến hành thực hiện việc giải mã và lưu trữ hình ảnh giải mã hiện thời trong DPB. Các lệnh trong bộ nhớ cũng có thể tiến hành đánh dấu hình ảnh giải mã hiện thời trong DPB. Các lệnh trong bộ nhớ còn có thể tiến hành thực hiện việc kết xuất hình ảnh bổ sung từ DPB.

Một số cấu hình sẽ được mô tả cùng với các hình vẽ, trong đó các số chỉ dẫn giống nhau có thể để chỉ các thành phần tương tự về mặt chức năng. Các hệ thống và các phương pháp như được mô tả chung và minh họa trên các hình vẽ ở bản mô tả này có thể được trang bị và thiết kế theo nhiều cấu hình khác nhau. Do vậy, mô tả chi tiết sâu hơn sau đây về một

số cấu hình, như được thể hiện trên các hình vẽ, không phải là để giới hạn phạm vi, như được yêu cầu bảo hộ, mà chỉ mang tính đại diện đối với các hệ thống và các phương pháp đó.

Fig.63 là sơ đồ khối thể hiện việc mã hóa video giữa nhiều thiết bị điện tử 2102a-b. Thiết bị điện tử thứ nhất 2102a và thiết bị điện tử thứ hai 2102b được minh họa. Tuy nhiên, cần lưu ý rằng một hoặc nhiều dấu hiệu và chức năng được mô tả liên quan đến thiết bị điện tử thứ nhất 2102a và thiết bị điện tử thứ hai 2102b có thể được kết hợp thành một thiết bị điện tử 102 trong một số cấu hình. Mỗi thiết bị điện tử 102 có thể được tạo cấu hình để mã hóa video và/hoặc để giải mã video. Các thiết bị điện tử 102 có thể được tạo cấu hình để sử dụng tác vụ của bộ đệm hình ảnh giải mã (DPB) hỗn hợp. Tác vụ của bộ đệm hình ảnh giải mã (DPB) hỗn hợp này là các kịch bản trong đó nhiều bước của việc gỡ bỏ, kết xuất (đệm), lưu trữ, đánh dấu và kết xuất (đệm) bổ sung được thực hiện trên bộ đệm hình ảnh giải mã (DPB) 116 xảy ra hoặc trên cơ sở ảnh hoặc trên cơ sở đơn vị truy cập (AU). Mặc dù các kết hợp của các bước này được là như được thực hiện trên cơ sở ảnh hoặc trên cơ sở đơn vị truy cập (AU), tất cả các kết hợp có thể của việc thực hiện mỗi bước này được trợ giúp riêng lẻ hoặc trên cơ sở ảnh hoặc trên cơ sở đơn vị truy cập (AU).

Trong bản mô tả này, đơn vị truy cập (AU) là tập hợp các đơn vị lớp trừu tượng hoá mạng (NAL) mà chúng được kết hợp với nhau theo quy tắc phân loại xác định, mà chúng có tính liên tiếp theo thứ tự giải mã và chúng bao gồm các đơn vị NAL của lớp mã hóa video (VCL) của tất cả các hình ảnh mã hóa liên kết với cùng thời gian kết xuất và các đơn vị NAL không VCL liên kết của chúng. Lớp cơ sở thứ nhất là lớp trong đó tất cả các đơn vị NAL VCL có `nuh_layer_id` bằng 0. Hình ảnh mã hoá là sự trình bày đã mã hoá của hình ảnh bao gồm các đơn vị NAL VCL với giá trị `nuh_layer_id` và nó bao gồm tất cả các bộ dạng cây mã hóa của hình ảnh. Trong một số trường hợp, hình ảnh mã hoá có thể được gọi là thành phần lớp. Các chi tiết bổ sung về các bước dựa trên hình ảnh hoặc dựa trên đơn vị truy cập (AU) được thể hiện như sau cùng với các hình vẽ Fig.69 và 70.

Theo một cấu hình, mỗi trong số các thiết bị điện tử 102 có thể phù hợp với chuẩn mã hóa video hiệu suất cao đa cảnh (MV-HEVC), chuẩn mã hóa video hiệu suất cao có thể mở rộng (SHVC) hoặc chuẩn mã hóa video hiệu suất cao đa cảnh (MV-HEVC). Chuẩn HEVC là chuẩn nén video hoạt động như chuẩn kế tiếp của định dạng H.264/MPEG-4 AVC (Mã hóa video tiên tiến) và tạo ra chất lượng video cải tiến và các tỷ lệ nén dữ liệu nâng cao. Trong bản mô tả này, ảnh là một dãy các mẫu độ sáng theo định dạng đơn sắc hoặc một

dãy các mẫu độ sáng và hai dãy các mẫu màu sắc tương ứng theo định dạng màu sắc 4:2:0, 4:2:2 và 4:4:4 hoặc định dạng khác nào đó. Hoạt động của bộ giải mã tham chiếu giả định (HRD) và hoạt động của bộ đệm hình ảnh giải mã (DPB) của thứ tự kết xuất 116 được mô tả cho SHVC và MV-HEVC trong JCTVC-M1008, JCTVC-L1008, JCTVC-D1004, JCT3V-C1004, JCTVC-L0453 và JCTVC-L0452.

Thiết bị điện tử thứ nhất 2102a có thể bao gồm bộ mã hóa 2108 và môđun truyền tín hiệu thời gian tồn thêm 2112. Thiết bị điện tử thứ nhất 2102a có thể được suy ra hình ảnh đầu vào 2106. Trong một số cấu hình, hình ảnh đầu vào 2106 này có thể được chụp trên thiết bị điện tử thứ nhất 2102a sử dụng cảm biến ảnh, được truy hồi từ bộ nhớ và/hoặc thu được từ thiết bị điện tử khác 102. Bộ mã hóa 2108 có thể mã hóa hình ảnh đầu vào 2106 để tạo ra dữ liệu mã hoá 2110. Ví dụ, bộ mã hóa 2108 này có thể mã hóa một loạt hình ảnh đầu vào 2106 (ví dụ, video). Dữ liệu mã hoá 2110 có thể là dữ liệu số (ví dụ, dòng bit).

Môđun truyền tín hiệu thời gian tồn thêm 2112 có thể tạo ra việc truyền tín hiệu bổ sung dựa trên dữ liệu mã hoá 2110. Ví dụ, môđun truyền tín hiệu thời gian tồn thêm 2112 có thể bổ sung dữ liệu thời gian tồn thêm cho dữ liệu mã hoá 2110 dưới dạng thông tin tiêu đề phiên, thông tin tập tham số video (VPS - video parameter set), thông tin tập tham số chuỗi (SPS - sequence parameter set), thông tin tập tham số hình ảnh (PPS - picture parameter set), số đếm thứ tự ảnh (POC - picture order count), ký hiệu hình ảnh tham chiếu v.v.. Trong một số cấu hình, môđun truyền tín hiệu thời gian tồn thêm 2112 có thể tạo ra chỉ số bao bộ chỉ báo sự quá độ giữa hai tập hợp hình ảnh.

Bộ mã hóa 2108 (và môđun truyền tín hiệu thời gian tồn thêm 2112, ví dụ) có thể tạo ra dòng bit 2114. Dòng bit 2114 này có thể bao gồm dữ liệu hình ảnh mã hoá dựa trên hình ảnh đầu vào 2106. Trong một số cấu hình, dòng bit 2114 này có thể bao gồm dữ liệu tiêu đề, như thông tin tiêu đề phiên, thông tin VPS, thông tin SPS, thông tin PPS v.v.. Do các hình ảnh đầu vào bổ sung 2106 được mã hoá, dòng bit 2114 có thể bao gồm một hoặc nhiều hình ảnh mã hoá. Chẳng hạn, dòng bit 2114 có thể bao gồm một hoặc nhiều hình ảnh tham chiếu mã hoá và/hoặc các hình ảnh khác.

Dòng bit 2114 có thể được cung cấp cho bộ giải mã 2104. Theo một ví dụ, dòng bit 2114 có thể được truyền đến thiết bị điện tử thứ hai 2102b sử dụng kết nối hữu tuyến hoặc vô tuyến. Trong một số trường hợp, việc này có thể được thực hiện qua mạng, như Internet hoặc mạng cục bộ (LAN). Như được thể hiện trên Fig.63, bộ giải mã 2104 có thể được thực hiện trên thiết bị điện tử thứ hai 2102b riêng với bộ mã hóa 2108 trên thiết bị điện tử thứ

nhất 2102a. Tuy nhiên, cần lưu ý rằng bộ mã hóa 2108 và bộ giải mã 2104 có thể được thực hiện trên cùng thiết bị điện tử 102 trong một số cấu hình. khi bộ mã hóa 2108 và bộ giải mã 2104 được thực hiện trên cùng thiết bị điện tử, chẳng hạn, dòng bit 2114 có thể được cung cấp qua bus cho bộ giải mã 2104 hoặc được lưu trữ trong bộ nhớ để truy hồi từ bộ giải mã 2104.

Bộ giải mã 2104 có thể thu (ví dụ, thu được) dòng bit 2114. Bộ giải mã 2104 có thể tạo ra hình ảnh giải mã 2118 (ví dụ, một hoặc nhiều hình ảnh giải mã 2118) dựa trên dòng bit 2114. Hình ảnh giải mã 2118 có thể được hiển thị, được phát lại, được lưu trữ trong bộ nhớ và/hoặc được truyền đến thiết bị khác v.v..

Bộ giải mã 2104 có thể bao gồm bộ đệm hình ảnh giải mã (DPB) 116. Bộ đệm hình ảnh giải mã (DPB) 116 có thể là bộ đệm giữ lại các hình ảnh giải mã để tham chiếu, sắp xếp lại thứ tự kết xuất hoặc khoảng trễ kết xuất được xác định cho bộ giải mã tham chiếu giả định (HRD). Trên thiết bị điện tử 102, bộ đệm hình ảnh giải mã (DPB) 116 có thể được sử dụng để lưu trữ các hình ảnh được tái thiết (ví dụ, được giải mã) tại bộ giải mã 2104. Các hình ảnh được lưu trữ này sau đó có thể được sử dụng, ví dụ, theo cơ chế dự đoán liên kết. Khi các hình ảnh giải mã không đúng thứ tự, thì các hình ảnh này có thể được lưu trữ trong bộ đệm hình ảnh giải mã (DPB) 116 theo đó sau này chúng có thể được hiển thị theo thứ tự.

Theo chuẩn H.264 hoặc chuẩn mã hóa video tiên tiến (AVC), việc quản lý bộ đệm hình ảnh giải mã (DPB) 116 (ví dụ, xóa, bổ sung ảnh, sắp xếp lại thứ tự ảnh v.v..) được thực hiện sử dụng các hoạt động kiểm soát quản lý bộ nhớ (MMCO). Nhiều phương pháp quản lý bộ đệm hình ảnh giải mã (DPB) 116 khác nhau đang được xem xét.

Bộ giải mã 2104 có thể bao gồm môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 2120. Môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 2120 chú ý đến các phương pháp quản lý bộ đệm hình ảnh giải mã (DPB) 116 với các bước dựa trên cơ sở ảnh 2122 và/hoặc các phương pháp quản lý bộ đệm hình ảnh giải mã (DPB) 116 với các bước dựa trên đơn vị truy cập (AU) 2124. Ví dụ, một ưu điểm của việc sử dụng các bước dựa trên hình ảnh 2124 đối với việc gỡ bỏ, lưu trữ và đánh dấu tham chiếu là bộ nhớ tối ưu của bộ đệm hình ảnh giải mã (DPB) 116 sẽ được sử dụng bởi nhiều lớp khác nhau. Do vậy, tổng bộ nhớ cần thiết khi sử dụng các bước dựa trên hình ảnh có thể nhỏ hơn. Một ưu điểm của việc sử dụng các bước dựa trên đơn vị truy cập (AU) 2124 đối với việc kết xuất (bao gồm kết xuất, đánh dấu kết xuất và kết xuất bổ sung) là quy trình kết xuất có thể được đơn giản hóa.

Fig.64 thể hiện sơ đồ luồng của phương pháp 2200 để hoạt động bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116. Phương pháp 2200 này có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 dưới dạng một phần của bộ giải mã 2104 trên thiết bị điện tử 102. Theo một cấu hình, phương pháp 2200 này có thể được thực hiện nhờ môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 2120. Bộ đệm hình ảnh giải mã (DPB) 116 có thể bắt đầu bước phân giải 2202 tiêu đề phiên thứ nhất của hình ảnh hiện thời. Bộ đệm hình ảnh giải mã (DPB) 116 có thể xác định 2204 các bước hoạt động nào của bộ đệm hình ảnh giải mã (DPB) hỗn hợp sẽ dựa vào ảnh và các bước nào sẽ dựa vào đơn vị truy cập (AU).

Ở bước 2206, bộ đệm hình ảnh giải mã (DPB) 116 có thể thực hiện gỡ bỏ (không có kết xuất) từ bộ đệm hình ảnh giải mã (DPB) 116. Việc gỡ bỏ này có thể gỡ bỏ các hình ảnh khỏi bộ đệm hình ảnh giải mã (DPB) 116 trước khi giải mã hình ảnh hiện thời. Bộ đệm hình ảnh giải mã (DPB) 116 này có thể thực hiện 2208 việc kết xuất hình ảnh (đệm) từ bộ đệm hình ảnh giải mã (DPB) 116. Kết xuất hình ảnh (đệm) có thể chỉ việc kết xuất các hình ảnh từ bộ đệm hình ảnh giải mã (DPB) 116 vào thời gian gỡ bỏ bộ đệm hình ảnh mã hoá (CPB). Trong một số cấu hình, thuật ngữ đệm có thể được sử dụng để chỉ việc kết xuất một hoặc nhiều hình ảnh từ bộ đệm hình ảnh giải mã (DPB) 116. Do vậy, các thuật ngữ đệm và kết xuất có thể được sử dụng thay đổi cho nhau.

Ở bước 2210, bộ đệm hình ảnh giải mã (DPB) 116 có thể giải mã và lưu trữ hình ảnh hiện thời trong bộ đệm hình ảnh giải mã (DPB) 116. Ở bước 2212, bộ đệm hình ảnh giải mã (DPB) 116 có thể đánh dấu hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm hình ảnh giải mã (DPB) 116 đó. Ví dụ, ở bước 2212, bộ đệm hình ảnh giải mã (DPB) 116 này có thể đánh dấu hình ảnh giải mã hiện thời là “không sử dụng để tham chiếu”, “được sử dụng để tham chiếu” “cần kết xuất” hoặc “không cần kết xuất”. Ở bước 2214, bộ đệm hình ảnh giải mã (DPB) 116 này cũng có thể thực hiện việc kết xuất hình ảnh khác nữa (đệm bổ sung) từ bộ đệm hình ảnh giải mã (DPB) 116. Trong một số cấu hình, việc gỡ bỏ/đệm lặp lại của các hình ảnh từ bộ đệm hình ảnh giải mã (DPB) 116 có thể xuất hiện cho tới khi các điều kiện nhất định được thỏa mãn.

Fig.65 thể hiện sơ đồ luồng của phương pháp khác 2300 để hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116. Ví dụ, phương pháp này 2300 trên Fig.65 có thể là phương pháp được ưu tiên để hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116. Phương pháp này 2300 có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 là một phần của bộ giải mã 2104 trên thiết bị điện tử 102. Theo một cấu hình, phương pháp

này 2300 có thể được thực hiện nhờ môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 2120. Bộ đệm hình ảnh giải mã (DPB) 116 này có thể bắt đầu bước phân giải 2302 tiêu đề phiên thứ nhất của hình ảnh hiện thời. Thuật ngữ hỗn hợp chỉ thực tế là một số bước hoạt động của bộ đệm hình ảnh giải mã (DPB) 116 được thực hiện trên cơ sở ảnh và một số bước hoạt động của bộ đệm hình ảnh giải mã (DPB) 116 được thực hiện trên cơ sở đơn vị truy cập (AU). Ở bước 2314, bộ đệm hình ảnh giải mã (DPB) 116 này có thể thực hiện 2304 việc gỡ bỏ dựa trên hình ảnh (không có kết xuất) từ bộ đệm hình ảnh giải mã (DPB) 116. Bộ đệm hình ảnh giải mã (DPB) 116 có thể thực hiện 2306 việc kết xuất (đệm) ảnh dựa trên đơn vị truy cập (AU) từ bộ đệm hình ảnh giải mã (DPB) 116. Bộ đệm hình ảnh giải mã (DPB) 116 có thể thực hiện 2308 việc giải mã dựa trên hình ảnh và lưu trữ hình ảnh hiện thời trong bộ đệm hình ảnh giải mã (DPB).

Ở bước 310, bộ đệm hình ảnh giải mã (DPB) 116 có thể thực hiện đánh dấu dựa trên hình ảnh đối với hình ảnh giải mã hiện thời trong bộ đệm hình ảnh giải mã (DPB). Bước đánh dấu được thực hiện nhờ bộ đệm hình ảnh giải mã (DPB) 116 có thể được tiếp tục phân chia để bao gồm cả bước đánh dấu tham chiếu và bước đánh dấu kết xuất. Trong bản mô tả này, việc đánh dấu các hình ảnh là “không sử dụng để tham chiếu” hoặc “được sử dụng để tham chiếu” được gọi là bước đánh dấu tham chiếu. Hình ảnh mã hoá trong bộ đệm hình ảnh giải mã (DPB) 116 có thể chỉ được đánh dấu là một trong số “không sử dụng để tham chiếu”, “sử dụng để tham chiếu ngắn hạn”, hoặc “sử dụng để tham chiếu dài hạn” vào khoảng khắc được quy định bất kỳ trong khi vận hành quy trình giải mã. Việc ấn định một trong các dấu hiệu này đối với ảnh sẽ gỡ bỏ rõ ràng các dấu hiệu khác mà chúng được ấn định đối với ảnh. Khi ảnh được đánh dấu là “được sử dụng để tham chiếu” thì điều này để chỉ chung đối với ảnh được đánh dấu là hoặc “sử dụng để tham chiếu ngắn hạn” hoặc “sử dụng để tham chiếu dài hạn”, nhưng không bao giờ để chỉ cả hai trường hợp. Trong bản mô tả này, việc đánh dấu các hình ảnh là “cần kết xuất” hoặc “không cần kết xuất” được gọi là bước đánh dấu kết xuất.

Bộ đệm hình ảnh giải mã (DPB) 116 có thể hoạt động theo cách thức mà bước đánh dấu tham chiếu và bước đánh dấu kết xuất có thể xảy ra hoặc trên cơ sở ảnh hoặc trên cơ sở đơn vị truy cập (AU). Nói chung, tất cả các kết hợp có thể (có bốn kết hợp điển hình) của việc thực hiện hai bước đánh dấu này một cách riêng lẻ trên cơ sở ảnh hoặc trên cơ sở đơn vị truy cập (AU) được trợ giúp. Tuy nhiên, tốt hơn nếu bước đánh dấu tham chiếu dựa trên hình ảnh và bước đánh dấu kết xuất dựa trên đơn vị truy cập (AU).

Bộ đệm hình ảnh giải mã (DPB) 116 cũng có thể thực hiện 2312 việc đánh dấu kết xuất dựa trên đơn vị truy cập (AU) của hình ảnh giải mã hiện thời. Bộ đệm hình ảnh giải mã (DPB) 116 có thể thực hiện 2314 việc gỡ bỏ hình ảnh dựa trên đơn vị truy cập (AU) khác nữa (đệm bổ sung) từ bộ đệm hình ảnh giải mã (DPB) 116. Trong một số cấu hình, việc gỡ bỏ/đệm lặp lại của các hình ảnh từ bộ đệm hình ảnh giải mã (DPB) 116 có thể xuất hiện cho tới khi các điều kiện nhất định được thỏa mãn.

Theo một số phương pháp, có thể cần các ràng buộc về tính phù hợp của dòng bit đối với các cờ trong tiêu đề phân đoạn phiên. Trong một số trường hợp, các ràng buộc này có thể được áp dụng cho các cờ đối với các hình ảnh thuộc cùng một đơn vị truy cập (AU). Ví dụ, các cờ `pic_output_flag` và/hoặc `no_output_of_prior_pics_flag` có thể là cần thiết để tuân thủ các ràng buộc tính phù hợp của dòng bit. Chẳng hạn, JCTVC-L1003, JCTVC-M1008 và JCT3V-D1004 mỗi tài liệu đều mô tả việc truyền tín hiệu trong tiêu đề phân đoạn phiên sử dụng `pic_output_flag` và `no_output_of_prior_pics_flag`. Ngoài ra, các cờ như `PicOutputFlag`, `NoRaslOutputFlag` và `NoOutputOfPriorPicsFlag` có thể suy ra được dựa trên các phần tử cú pháp và các loại đơn vị NAL.

JCTVC-L1003, JCTVC-M1008 và JCT3V-D1004, cũng bao gồm mô tả về DPB cho HEVC, SHVC và MV-HEVC. JCTVC-M1008 SHVC Draft Text 1 cung cấp dự thảo văn bản cho phần mở rộng có thể mở rộng của HEVC. JCT3V-D1004 MV-HEVC Draft Text 4 mô tả dự thảo văn bản cho phần mở rộng đa cảnh của HEVC.

Dòng bit mã hoá video, theo JCTVC-L1003, JCTVC-M1008 và/hoặc JCT3V-D1004, có thể bao gồm cấu trúc cú pháp được đặt vào các bó dữ liệu logic được gọi chung là các đơn vị lớp trừu tượng hoá mạng (NAL). Mỗi đơn vị NAL bao gồm tiêu đề của đơn vị NAL, như tiêu đề của đơn vị NAL hai byte (ví dụ, 16 bit), để xác định mục đích của phần dữ liệu chuyển liên quan. Ví dụ, mỗi phiên mã hoá (và/hoặc ảnh) có thể là mã hóa thành một hoặc nhiều đơn vị NAL của phiên (và/hoặc ảnh). Các đơn vị NAL khác có thể được bao gồm dùng cho các hạng mục dữ liệu khác, như ví dụ, thông tin nâng cao bổ sung, phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời (TSA), phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời theo bước (STSA), phiên mã hoá không TSA, ảnh kế tiếp không STSA, phiên mã hoá của hình ảnh truy nhập liên kết gãy, phiên mã hoá của hình ảnh làm mới giải mã tức thì, phiên mã hoá của hình ảnh truy nhập ngẫu nhiên sạch, phiên mã hoá của hình ảnh chính có thể giải mã, phiên mã hoá của được dán nhãn đối với hình ảnh gỡ bỏ, tập tham số video, tập tham số chuỗi, tập tham số hình ảnh, dấu phân cách đơn vị truy cập, đuôi của chuỗi,

đuôi của dòng bit, dữ liệu của bộ lọc và/hoặc thông báo thông tin nâng cao chuỗi. Bảng (7) dưới đây minh họa một ví dụ về các mã của đơn vị NAL và các lớp của loại đơn vị NAL. Các loại đơn vị NAL khác cũng có thể được bao gồm, nếu cần.

Cần hiểu rằng các giá trị của loại đơn vị NAL đối với các đơn vị NAL được thể hiện trong bảng (7) có thể là được trang bị lại và được ấn định lại. Ngoài ra, các loại đơn vị NAL bổ sung có thể được bổ sung hoặc gỡ bỏ.

Hình ảnh của điểm truy nhập ngẫu nhiên trong ảnh (IRAP) là hình ảnh mã hoá, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` nằm trong khoảng từ `BLA_W_LP` đến `RSV_IRAP_VCL23`, bao gồm cả giá trị đầu cuối, như được thể hiện trong bảng (7) dưới đây. Hình ảnh IRAP chỉ bao gồm các phiên mã hóa trong ảnh (i).

Hình ảnh làm mới giải mã tức thì (IDR) là hình ảnh IRAP, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` bằng `IDR_W_RADL` hoặc `IDR_N_LP`, như được thể hiện trong bảng (7). Hình ảnh làm mới giải mã tức thì (IDR) chỉ bao gồm các phiên I và có thể là hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong dòng bit.

Mỗi hình ảnh IDR là hình ảnh thứ nhất của chuỗi video mã hóa (CVS) theo thứ tự giải mã. Hình ảnh truy nhập liên kết gãy (BLA) là hình ảnh IRAP, mà đối với nó, mỗi đơn vị NAL lớp mã hóa video có `nal_unit_type` bằng `BLA_W_LP`, `BLA_W_RADL`, hoặc `BLA_N_LP`, như được thể hiện trong bảng (7).

Hình ảnh BLA chỉ bao gồm các phiên I và có thể là hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh BLA bắt đầu chuỗi video mã hóa mới và có cùng tác dụng trên quy trình giải mã giống như hình ảnh IDR. Tuy nhiên, hình ảnh BLA bao gồm các phần tử cú pháp mà chúng tập hình ảnh tham chiếu không trống.

Bảng 42

<code>nal_unit_type</code>	Tên của <code>nal_unit_type</code>	Nội dung đơn vị NAL và cấu trúc cú pháp của phần dữ liệu chuyển chuỗi byte thô (RBSP)	Lớp của loại đơn vị NAL
0	<code>TRAIL_N</code>	Phân đoạn phiên mã hoá của hình ảnh kéo không TSA, không STSA	Lớp mã hóa video
1	<code>TRAIL_R</code>	<code>slice_segment_layer_rbsp()</code>	(VCL)

2 3	TSA_N TSA_R	Phân đoạn phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời (TSA) slice_segment_layer_rbsp()	VCL
4 5	STSA_N STSA_R	Phân đoạn phiên mã hoá của hình ảnh truy nhập lớp phụ tạm thời theo bước (STSA) slice_segment_layer_rbsp()	VCL
6 7	RADL_N RADL_R	Phân đoạn phiên mã hoá của hình ảnh chính có thể giải mã truy nhập ngẫu nhiên (RADL) slice_segment_layer_rbsp()	VCL
8 9	RASL_N RASL_R	Phân đoạn phiên mã hoá của hình ảnh dẫn đầu bỏ qua truy nhập ngẫu nhiên (RASL) slice_segment_layer_rbsp()	VCL
10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Các loại đơn vị NAL VCL không tham chiếu lớp phụ không IRAP dự phòng	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Các loại đơn vị NAL VCL tham chiếu lớp phụ không IRAP dự phòng	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Phân đoạn phiên mã hoá của hình ảnh truy nhập liên kết gãy (BLA) slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Phân đoạn phiên mã hoá của hình ảnh làm mới giải mã tức thì (IDR) slice. segment_layer_rbsp()	VCL
21	CRA_NUT	Phân đoạn phiên mã hoá hình ảnh truy nhập ngẫu nhiên sạch (CRA) slice_segment_layer_rbsp()	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Các loại đơn vị NAL VCL IRAP dự phòng	VCL
24..31	RSV_VCL24.. RSV_VCL31	Các loại đơn vị NAL VCL không IRAP dự phòng	VCL
32	VPS_NUT	Tập tham số video Video_parameter_set_rbsp()	lớp mã hóa không
33	SPS_NUT	Tập tham số dãy Seq_parameter_set_rbsp()	không VCL
34	PPS_NUT	Tập tham số hình ảnh pic_parameter_set_rbsp()	không VCL
35	AUD_NUT	Dấu phân cách đơn vị truy cập access_unit_delimiter_RBSP()	không VCL
36	EOS_NUT	Đuôi của chuỗi end_of_seq_rbsp()	không VCL
37	EOB_NTUT	Đuôi của dòng bit end_of_bitstream_rbsp()	không VCL
38	FD_NUT	Dữ liệu của bộ lọc filler_data_rbsp()	không VCL

39 40	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Thông tin nâng cao bổ sung sei_rbsp()	không VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Dự phòng	không VCL
48..63	UNSPEC48.. UNSPEC83	Không xác định	không VCL

Bảng (7)

Trên bảng (8) dưới đây, cú pháp tiêu đề của đơn vị NAL có thể bao gồm hai byte dữ liệu, gọi là, 16 bit. Bit thứ nhất có thể là “forbidden_zero_bit” luôn được thiết đặt bằng 0 tại điểm bắt đầu của đơn vị NAL. Sáu bit tiếp theo có thể là “nal_unit_type” xác định phần dữ liệu chuyển của chỗi byte thô (“RBSP”) có trong đơn vị NAL như được thể hiện trong bảng (7) nêu trên. 6 bit tiếp theo có thể là “nuh_layer_id” xác định sự phân cách của lớp đó. Trong một số trường hợp, sáu bit này có thể được xác định để thay thế là “nuh_reserved_zero_6bits”. nuh_reserved_zero_6bits có thể bằng 0 theo tiêu chuẩn thông số kỹ thuật cơ sở. Mã hóa trong ảnh video mở rộng được và/hoặc các phần mở rộng cú pháp, nuh_layer_id có thể xác định rằng đơn vị NAL cụ thể này thuộc về lớp được xác định bởi giá trị 6 bit này.

Phần tử cú pháp tiếp theo có thể là “nuh_temporal_id_plus1”. nuh_temporal_id_plus1 - 1 có thể xác định nhận diện tạm thời cho đơn vị NAL đó. Nhận diện tạm thời khác nhau được TemporalId có thể được xác định là TemporalId = nuh_temporal_id_plus1 - 1. Nhận diện tạm thời TemporalId được sử dụng để xác định lớp phụ tạm thời. Biến HighestTid xác định lớp phụ tạm thời cao nhất được giải mã.

Bảng 43

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

Bảng (8)

Bảng (9) dưới đây thể hiện cấu trúc cú pháp tập tham số chuỗi (SPS) làm ví dụ. pic_width_in_luma_samples xác định chiều rộng của mỗi hình ảnh giải mã trong các bộ luma samples. pic_width_in_luma_samples không bằng 0. pic_height_in_luma_samples xác định chiều cao mỗi hình ảnh giải mã trong các bộ luma samples. pic_height_in_luma_samples không bằng 0.

$\text{sps_max_sub_layers_minus1} + 1$ xác định lượng tối đa của các lớp phụ tạm thời có thể có mặt trong mỗi CVS là SPS. Giá trị $\text{sps_max_sub_layers_minus1}$ nằm trong khoảng từ 0 đến 6, bao gồm cả giá trị đầu cuối.

Cờ $\text{sps_sub_layer_ordering_info_present_flag}$ bằng 1 xác định rằng $\text{sps_max_dec_pic_buffering_minus1}[i]$, $\text{sps_max_num_reorder_pics}[i]$ và $\text{sps_max_latency_increase_plus1}[i]$ các phần tử cú pháp có mặt đối với các lớp phụ $\text{sps_max_sub_layers_minus1} + 1$. $\text{sps_sub_layer_ordering_info_present_flag}$ bằng 0 xác định rằng các giá trị của $\text{sps_max_dec_pic_buffering_minus1}[\text{sps_max_sub_layers_minus1}]$, $\text{sps_max_num_reorder_pics}[\text{sps_max_sub_layers_minus1}]$ và $\text{sps_max_latency_increase_plus1}[\text{sps_max_sub_layers_minus1}]$ áp dụng cho tất cả các lớp phụ.

$\text{sps_max_dec_pic_buffering_minus1}[i] + 1$ xác định kích thước cần thiết tối đa của bộ đệm hình ảnh giải mã đối với CVS trong các bộ của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng i. Giá trị $\text{sps_max_dec_pic_buffering_minus1}[i]$ nằm trong khoảng từ 0 đến $\text{MaxDpbSize} - 1$, bao gồm cả giá trị đầu cuối trong đó MaxDpbSize xác định kích thước tối đa của bộ đệm hình ảnh giải mã trong các bộ của các bộ đệm lưu trữ hình ảnh. Khi i lớn hơn 0, thì $\text{sps_max_dec_pic_buffering_minus1}[i]$ sẽ lớn hơn hoặc bằng $\text{sps_max_dec_pic_buffering_minus1}[i - 1]$. Khi $\text{sps_max_dec_pic_buffering_minus1}[i]$ không có mặt cho i nằm trong khoảng từ 0 đến $\text{sps_max_sub_layers_minus1} - 1$, bao gồm cả giá trị đầu cuối, do $\text{sps_sub_layer_ordering_info_present_flag}$ bằng 0, thì suy ra $\text{sps_max_dec_pic_buffering_minus1}[\text{sps_max_sub_layers_minus1}]$.

$\text{sps_max_num_reorder_pics}[i]$ chỉ báo lượng tối đa cho phép của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS theo thứ tự giải mã và tiếp theo hình ảnh đó theo thứ tự kết xuất khi HighestTid bằng i. Giá trị $\text{sps_max_num_reorder_pics}[i]$ nằm trong khoảng từ 0 đến $\text{sps_max_dec_pic_buffering_minus1}[i]$, bao gồm cả giá trị đầu cuối. Khi i lớn hơn 0, thì $\text{sps_max_num_reorder_pics}[i]$ có thể là lớn hơn hoặc bằng $\text{sps_max_num_reorder_pics}[i - 1]$. Khi $\text{sps_max_num_reorder_pics}[i]$ không có mặt cho i nằm trong khoảng từ 0 đến $\text{sps_max_sub_layers_minus1} - 1$, bao gồm cả giá trị đầu cuối, do $\text{sps_sub_layer_ordering_info_present_flag}$ bằng 0, thì suy ra $\text{sps_max_num_reorder_pics}[\text{sps_max_sub_layers_minus1}]$.

`sps_max_latency_increase_plus1[i]` không bằng 0 có thể được sử dụng để tính toán giá trị `SpsMaxLatencyPictures[i]`, giá trị này xác định lượng tối đa của các hình ảnh mà chúng đứng trước ảnh bất kỳ trong CVS theo thứ tự kết xuất và tiếp theo hình ảnh đó theo thứ tự giải mã khi `HighestTid` bằng `i`. Khi `sps_max_latency_increase_plus1[i]` không bằng 0, thì giá trị `SpsMaxLatencyPictures[i]` được xác định là `SpsMaxLatencyPictures[i] = sps_max_num_reorder_pics[i] + sps_max_latency_increase_plus1[i] - 1`. Khi `sps_max_latency_increase_plus1[i]` bằng 0, thì không có giới hạn tương ứng nào được thể hiện.

Giá trị `sps_max_latency_increase_plus1[i]` nằm trong khoảng từ 0 đến $2^{32} - 2$, bao gồm cả giá trị đầu cuối. Khi `sps_max_latency_increase_plus1[i]` không có mặt cho `i` nằm trong khoảng từ 0 đến `sps_max_sub_layers_minus1 - 1`, bao gồm cả giá trị đầu cuối, do `sps_sub_layer_ordering_info_present_flag` bằng 0, nên nó được suy ra bằng `sps_max_latency_increase_plus1[sps_max_sub_layers_minus1]`.

Bảng 44

<code>seq_parameter_set_rbsp() {</code>
<code>...</code>
<code>sps_max_sub_layers_minus1</code>
<code>...</code>
<code>pic_width_in_luma_samples</code>
<code>pic_height_in_luma_samples</code>
<code>...</code>
<code>for(i = (sps_sub_layer_ordering_info_present_flag ? 0 :</code>
<code>sps_max_sub_layers_minus1);</code>
<code> i <= sps_max_sub_layers_minus1; i++) {</code>
<code> sps_max_dec_pic_buffering_minus1[i]</code>
<code> sps_max_num_reorder_pics[i]</code>
<code> sps_max_latency_increase_plus1[i]</code>
<code> }</code>
<code>...</code>
<code>}</code>

Bảng (9)

Ngoài ra, JCTVC-L1003 mô tả chuẩn HEVC. Ví dụ, chi tiết liên quan đến `pic_out_flag` và `no_output_of_prior_pics_flag` được thể hiện trong bảng (10) dưới đây:

Bảng 45

CÚ PHÁP PHẦN ĐẦU ĐOẠN PHIÊN CHUNG
<pre> slice_segment_header() { first_slice_segment_in_pic_flag if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23) no_output_of_prior_pics_flag slice_pic_parameter_set_id if(!first_slice_segment_in_pic_flag) { if(dependent_slice_segments_enabled_flag) dependent_slice_segment_flag slice_segment_address } if(!dependent_slice_segment_flag) { for(i = 0; i < num_extra_slice_header_bits; i++) slice_reserved_flag[i] slice_type if(output_flag_present_flag) pic_output_flag } } </pre>

Bảng (10)

Trong bảng (10), `no_output_of_prior_pics_flag` tác động đến đầu ra của các hình ảnh giải mã trước đây trong bộ đệm hình ảnh giải mã (DPB) sau bước giải mã của IDR hoặc hình ảnh BLA không phải là hình ảnh thứ nhất trong dòng bit như được xác định trong Phụ lục C của JCTVC-L1003.

`output_flag_present_flag` bằng 1 chỉ báo rằng phần tử cú pháp `pic_output_flag` có mặt trong các tiêu đề phiên liên kết. `output_flag_present_flag` bằng 0 chỉ báo rằng phần tử cú pháp `pic_output_flag` không có mặt trong các tiêu đề phiên liên kết. `pic_output_flag` tác động tới các quy trình kết xuất hình ảnh giải mã và gỡ bỏ như được xác định trong Phụ lục C của JCTVC-L1003. Khi `pic_output_flag` không có mặt, thì suy ra 1.

Đối với quá trình giải mã chung (như được nêu tại 8.1 của JCTVC-L1003), `PicOutputFlag` được đặt như sau:

- Nếu hình ảnh hiện thời là ảnh RASL và `NoRaslOutputFlag` của hình ảnh IRAP liên kết bằng 1, thì `PicOutputFlag` có thể được thiết đặt bằng 0.

- Theo cách khác, `PicOutputFlag` có thể được thiết đặt là `pic_output_flag`.

Ngoài ra, trong quá trình giải mã chung để tạo ra các hình ảnh tham chiếu không có để sử dụng (như được nêu tại phần 8.3.3.1 của JCTVC-L1003), giá trị `PicOutputFlag` đối với ảnh tạo ra có thể được thiết đặt bằng 0 trong các điều kiện nhất định.

Khi ảnh hiện thời là hình ảnh IRAP, quy định sau đây được áp dụng:

- Nếu hình ảnh hiện thời là hình ảnh IDR, hình ảnh BLA, hình ảnh thứ nhất trong dòng bit theo thứ tự giải mã hoặc hình ảnh thứ nhất tiếp theo đuôi đơn vị NAL của dãy theo thứ tự giải mã, biến NoRaslOutputFlag có thể được thiết đặt bằng 1.

- Theo cách khác, nếu một số phương tiện bên ngoài chưa được xác định trong JCTVC-L1003 có thể sử dụng được để thiết đặt biến HandleCraAsBlaFlag tại giá trị đối với ảnh hiện thời, thì biến HandleCraAsBlaFlag có thể được thiết đặt là giá trị được tạo ra bởi các phương tiện bên ngoài và biến NoRaslOutputFlag có thể được thiết đặt là HandleCraAsBlaFlag.

- Theo cách khác, biến HandleCraAsBlaFlag có thể được thiết đặt bằng 0 và biến NoRaslOutputFlag có thể được thiết đặt bằng 0.

Như đã mô tả trên đây, đơn vị truy cập (AU) là tập hợp các đơn vị lớp trừu tượng hoá mạng (NAL) mà chúng được kết hợp với nhau theo quy tắc phân loại xác định, mà chúng có tính liên tiếp theo thứ tự giải mã và chúng bao gồm các đơn vị NAL của lớp mã hóa video (VCL) của tất cả hình ảnh mã hoá kết hợp với cùng thời gian kết xuất và các đơn vị NAL không VCL liên kết của chúng. Lớp cơ sở thứ nhất là lớp trong đó tất cả các đơn vị NAL VCL có nuh_layer_id bằng 0. Hình ảnh mã hoá là sự biểu diễn giải mã của hình ảnh bao gồm các đơn vị NAL VCL với giá trị nuh_layer_id và chúng bao gồm tất cả các bộ dạng cây mã hóa của hình ảnh. Trong một số trường hợp, hình ảnh mã hoá có thể được gọi là thành phần lớp. Các chi tiết bổ sung về các bước dựa trên hình ảnh hoặc dựa trên đơn vị truy cập (AU) được thể hiện liên quan đến các hình vẽ Fig.69 và Fig.70 dưới đây.

Trong một số cấu hình, các ràng buộc tính phù hợp của dòng bit đối với pic_output_flag và/hoặc no_output_of_prior_pics_flag có thể được sử dụng đối với các hình ảnh mã hoá trong đơn vị truy cập (AU). Ngoài ra, ba cờ kết xuất mới của đơn vị truy cập, cờ kết xuất AU (ví dụ, AuOutputFlag), cờ kết xuất không RASL AU (ví dụ, AuNoRaslOutputFlag) và không kết xuất AU của cờ các hình ảnh trước (ví dụ, AuNoOutputOfPriorPicsFlag), có thể được suy ra cho AU dựa trên giá trị các phân tử cú pháp khác và các loại đơn vị NAL đối với các hình ảnh mã hoá trong AU. Trong một số cấu hình, việc kết xuất và gỡ bỏ các hình ảnh có thể dựa trên ba cờ bày (ví dụ, AuOutputFlag, AuNoRaslOutputFlag và AuNoOutputOfPriorPicsFlag) cho SHVC và HEVC đa cảnh.

Ví dụ, các ràng buộc tính phù hợp của dòng bit đối với các phần mở rộng HEVC có thể tiếp theo như được mô tả trong các hệ thống và các phương pháp nêu trong bản mô tả này. Cụ thể, các ràng buộc tính phù hợp của dòng bit có thể được áp dụng cho dòng bit

SHVC. Ngoài ra, các ràng buộc tính phù hợp của dòng bit có thể được áp dụng cho dòng bit MV-HEVC.

Theo một cấu hình, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `pic_output_flag` có thể cần phải giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá trong đơn vị truy cập (AU). Theo cấu hình khác, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `pic_output_flag`, khi có mặt, có thể giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá trong đơn vị truy cập (AU) khi các hình ảnh mã hoá có cùng đơn vị NAL.

Theo một cấu hình, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `pic_output_flag` cho các phân đoạn phiên với `nuh_layer_id` bằng giá trị `nuh_layer_id` của lớp đích có thể giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá đó trong đơn vị truy cập (AU). Theo cấu hình khác, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `pic_output_flag` cho các phân đoạn phiên với `nuh_layer_id` không bằng giá trị `nuh_layer_id` của lớp đích có thể bằng 0 trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá đó trong đơn vị truy cập (AU).

Theo một cấu hình, lớp đích có thể là lớp thuộc một tập hợp lớp hoặc tập hợp lớp đích hoặc tập hợp lớp kết xuất như được xác định trong JCTVC-L1003, JCTVC-M1008, hoặc JCT3V-D1004. Theo cấu hình khác, lớp đích có thể là lớp dự định sẽ được giải mã. Theo cấu hình khác nữa, lớp đích có thể là lớp dự định sẽ được giải mã và kết xuất (được hiển thị hoặc theo cách khác, thì được gửi để kết xuất).

Theo một số cấu hình, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `no_output_of_prior_pics_flag` có thể giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá trong đơn vị truy cập (AU). Trong các cấu hình khác, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `no_output_of_prior_pics_flag` có thể giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá trong đơn vị truy cập (AU) khi các hình ảnh mã hoá có cùng đơn vị NAL. Theo một cấu hình, khi có mặt, giá trị các phần tử cú pháp của tiêu đề phân đoạn phiên `no_output_of_prior_pics_flag` cho các phân đoạn phiên với `nuh_layer_id` bằng giá trị `nuh_layer_id` của lớp đích có thể giống nhau trong tất cả các tiêu đề phân đoạn phiên của các hình ảnh mã hoá đó trong đơn vị truy cập (AU).

Theo cấu hình khác, các phần tử cú pháp `pic_output_flag` và/hoặc `no_output_of_prior_pics_flag` có thể không được truyền tín hiệu khi `nuh_layer_id > 0`.

Trong trường hợp này, các giá trị cho các lớp có `nuh_layer_id > 0` có thể được suy ra bằng các giá trị được truyền tín hiệu đối với `nuh_layer_id` bằng 0.

Theo một số cấu hình, các cờ bổ sung, như `AuOutputFlag` và `AuNoRaslOutputFlag` có thể được sử dụng. Các cờ `AuOutputFlag` và `AuNoRaslOutputFlag` có thể được suy ra theo nhiều phương pháp. Theo một phương pháp hoặc cấu hình, hai cờ, `AuOutputFlag` và `AuNoRaslOutputFlag` có thể được suy ra từ và được sử dụng cho hoạt động của DPB. `AuOutputFlag` có thể được thiết đặt bằng 1 nếu `PicOutputFlag` bằng 1 đối với tất cả các hình ảnh trong AU. Theo cách khác, `AuOutputFlag` có thể được thiết đặt bằng 0.

Theo phương pháp khác, `AuOutputFlag` có thể được thiết đặt bằng 1 nếu `PicOutputFlag` bằng 1 cho ít nhất một ảnh trong AU. Theo cách khác, `AuOutputFlag` có thể được thiết đặt bằng 0. Do vậy, trong trường hợp này, `AuOutputFlag` có thể được thiết đặt bằng 0 nếu `PicOutputFlag` bằng 0 đối với tất cả các hình ảnh trong AU.

Theo phương pháp khác nữa, `AuOutputFlag` có thể được thiết đặt bằng 1 nếu `PicOutputFlag` bằng 1 cho các hình ảnh thuộc tất cả các lớp kết xuất đích trong AU. Theo cách khác, thì `AuOutputFlag` có thể được thiết đặt bằng 0.

Theo phương pháp khác nữa, `AuOutputFlag` có thể được thiết đặt bằng 1 nếu `PicOutputFlag` bằng 1 đối với ảnh thuộc ít nhất một trong các lớp kết xuất đích trong AU. Theo cách khác, `AuOutputFlag` có thể được thiết đặt bằng 0.

Theo một phương pháp, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 1 nếu `NoRaslOutputFlag` bằng 1 đối với tất cả các hình ảnh trong AU. Theo cách khác, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 0.

Theo phương pháp khác, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 1 nếu `NoRaslOutputFlag` bằng 1 cho ít nhất một ảnh trong AU. Theo cách khác, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 0. Do vậy, trong trường hợp này, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 0 nếu `NoRaslOutputFlag` bằng 0 đối với tất cả các hình ảnh trong AU.

Theo phương pháp khác nữa, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 1 nếu `NoRaslOutputFlag` bằng 1 cho các hình ảnh thuộc tất cả các lớp kết xuất đích trong AU. Theo cách khác, `AuNoRaslOutputFlag` có thể được thiết đặt bằng 0.

Theo phương pháp khác nữa, AuNoRaslOutputFlag có thể được thiết đặt bằng 1 nếu NoRaslOutputFlag bằng 1 đối với ảnh thuộc ít nhất một các lớp kết xuất đích trong AU. Theo cách khác, AuNoRaslOutputFlag có thể được thiết đặt bằng 0.

Theo một số phương pháp và cấu hình nêu trên, hoạt động của DPB có thể sử dụng AuOutputFlag thay cho PicOutputFlag. Ngoài ra, hoạt động của DPB có thể sử dụng AuNoRaslOutputFlag thay cho NoRaslOutputFlag.

Các ví dụ thể hiện việc sử dụng các cờ kết xuất AU, như AuOutputFlag và AuNoRaslOutputFlag, theo các hệ thống và các phương pháp hiện thời, được đưa ra dưới đây trong Tiêu mục (1A) và Tiêu mục (2A) dưới đây. Ngoài ra, như được mô tả dưới đây trong Danh mục (1), Tiêu mục (1A), Danh mục (2) và Tiêu mục (2A), cờ kết xuất AU AuNoOutputOfPriorPicsFlag có thể được suy ra từ và sử dụng cho hoạt động của DPB.

Các hệ thống và các phương pháp hiện thời có thể được thực hiện nhờ các thay đổi về danh mục tiêu chuẩn. Danh mục (1) dưới đây tạo ra các phần JCTVC-L1003 có thể thay đổi được để bao gồm các hệ thống và các phương pháp hiện thời.

Danh mục 1

C.3 Hoạt động của bộ đệm hình ảnh giải mã (DPB)

C.3.1 Những vấn đề chung

Các đặc tính kỹ thuật theo tiêu mục này áp dụng một cách độc lập đối với mỗi tập hợp các tham số DPB được chọn như được chỉ ra ở tiêu mục C.1. DPB hoạt động riêng hoặc độc lập đối với mỗi lớp. Do vậy, các bước sau đây sẽ tiến hành riêng đối với mỗi hình ảnh được giải mã với giá trị cụ thể nuh_layer_id.

Bộ đệm hình ảnh giải mã chứa các bộ đệm lưu trữ hình ảnh. Mỗi lớp bao gồm tập hợp của các bộ đệm lưu trữ hình ảnh của lớp đó. Do vậy, các bộ đệm lưu trữ hình ảnh của mỗi lớp được kết hợp với giá trị nuh_layer_id của lớp đó. Mỗi bộ đệm lưu trữ hình ảnh có thể chứa ảnh giải mã là ảnh được đánh dấu là “sử dụng để tham chiếu” hoặc được giữ lại để kết xuất trong tương lai. Các quy trình được quy định theo các tiêu mục C.3.2, C.3.3 và C.3.4 được áp dụng theo trình tự cụ thể dưới đây.

C.3.2 Gỡ bỏ các hình ảnh khỏi DPB

Việc gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra tức thì

tại thời điểm gỡ bỏ CBP của đơn vị giải mã thứ nhất của hình ảnh hiện thời thuộc đơn vị truy cập n (chứa hình ảnh hiện thời) và sẽ tiến hành như sau:

- Quy trình giải mã đối với RPS như được xác định ở khoản mục 8.3.2 được viện dẫn.
- Khi ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1 không phải là ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm như sau:

- Nếu hình ảnh hiện thời là ảnh CRA, NoOutputOfPriorPicsFlag được thiết đặt bằng 1 (không quan tâm đến giá trị no_output_of_prior_pics_flag).

- Theo cách khác, nếu giá trị pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minus1[HighestTid] lần lượt được suy ra từ SPS hoạt động tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời lần lượt khác với giá trị pic_width_in_luma_samples, thì pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minus1[HighestTid] được suy ra từ SPS hoạt động đối với ảnh đứng trước với giá trị nuh_layer_id bằng giá trị nuh_layer_id của hình ảnh hiện thời, NoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được thiết đặt bằng 1 bởi bộ giải mã trong thử nghiệm, không quan tâm đến giá trị no_output_of_prior_pics_flag.

Lưu ý - mặc dù thiết đặt NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt NoOutputOfPriorPicsFlag là 1 trong trường hợp này.

- Theo cách khác, NoOutputOfPriorPicsFlag được thiết đặt bằng no_output_of_prior_pics_flag.

2. Giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HRD, sao cho khi giá trị NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB được làm rỗng mà không kết xuất các hình ảnh mà chúng chứa và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

Theo một phương án, giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR, sao cho khi giá trị NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả giá trị nuh_layer_id trong

DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB đối với tất cả các giá trị `nuh_layer_id` được thiết đặt bằng 0.

Theo một phương án, giá trị `NoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR, sao cho khi giá trị `NoOutputOfPriorPicsFlag` bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh `PSB[nuh_layer_id]` với giá trị `currLayerId` bằng `nuh_layer_id` của hình ảnh hiện thời trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB `DPBFullness[currLayerId]` cho giá trị `nuh_layer_id` `currLayerId` của hình ảnh hiện thời được thiết đặt bằng 0.

Theo một phương án, giá trị `NoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR, sao cho khi giá trị `NoOutputOfPriorPicsFlag` bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh `PSB[nuh_layer_id]` đối với tất cả giá trị `nuh_layer_id` trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB `DPBFullness[nuh_layer_id]` cho tất cả các giá trị `nuh_layer_id` được thiết đặt bằng 0.

- Khi cả hai điều kiện sau đây là đúng đối với các hình ảnh bất kỳ k trong bộ đệm lưu trữ hình ảnh tương ứng với giá trị `nuh_layer_id` của hình ảnh hiện thời trong DPB, tất cả các hình ảnh k đó trong DPB được gỡ bỏ từ DPB:

- ảnh k được đánh dấu “không sử dụng để tham chiếu”
- ảnh k có `PicOutputFlag` bằng 0 hoặc thời gian kết xuất DPB của nó nhỏ hơn hoặc bằng thời gian gỡ bỏ khỏi CPB của đơn vị giải mã thứ nhất (được ký hiệu là đơn vị giải mã m) của hình ảnh hiện thời n ; nghĩa là `DpbOutputTime[k]` nhỏ hơn hoặc bằng `CpbRemovalTime(m)`
- Đối với mỗi hình ảnh được gỡ bỏ từ DPB, mức đầy DPB bị giảm đi một.
- Theo một phương án, đối với mỗi hình ảnh k có giá trị `nuh_layer_id` `nuhLayerIdk` được gỡ bỏ từ DPB, mức đầy DPB `DPBfulness[nuhLayerIdk]` bị giảm đi một.

C.3.3 Kết xuất hình ảnh

Các quy trình được xác định trong tiểu mục này xảy ra tức thì tại thời điểm gỡ bỏ CBP của đơn vị truy cập n , `AuCpbRemovalTime[n]`.

Khi ảnh n có `PicOutputFlag` bằng 1, thì thời gian kết xuất DPB của nó `DpbOutputTime[n]` được suy ra như sau, trong đó biến `firstPicInBufferingPeriodFlag` bằng

1 nếu đơn vị truy cập n là đơn vị truy cập thứ nhất của khoảng đệm và theo cách khác, bằng 0:

```

if( !SubPicHrdFlag) {
    DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockTick * picDpbOutputDelay
(C-16) nếu( firstPicInBufferingPeriodFlag)
    DPBOutputTime[ n ] -= ClockTick * DPBDelayOffset
} else
    DpbOutputTime[ n ] = AuCpbRemovalTime[ n ] + ClockSubTick *
picSptDpbOutputDuDelay

```

trong đó picDpbOutputDelay là giá trị của pic_dpb_output_delay trong thông báo SEI định thời hình ảnh liên quan đến đơn vị truy cập n và picSptDpbOutputDuDelay là giá trị của picSptDpbOutputDuDelay, khi có mặt, trong các thông báo SEI thông tin về bộ giải mã được kết hợp với đơn vị truy cập n hoặc giá trị pic_dpb_output_du_delay trong thông báo SEI định thời hình ảnh liên kết với đơn vị truy cập n khi không có thông báo SEI thông tin về bộ giải mã được kết hợp với đơn vị truy cập n hoặc không có thông báo SEI thông tin về bộ giải mã được kết hợp với đơn vị truy cập n có pic_spt_dpb_output_du_delay có mặt.

Lưu ý - Khi phân tử cú pháp pic_spt_dpb_output_du_delay không có mặt trong thông báo SEI thông tin về bộ giải mã bất kỳ được kết hợp với đơn vị truy cập n, thì giá trị được suy ra là pic_dpb_output_du_delay trong thông báo SEI định thời hình ảnh được kết hợp với đơn vị truy cập n.

Việc kết xuất hình ảnh hiện thời nếu nuh_layer_id của nó thuộc lớp trong TargetDecLayerIdList được xác định cụ thể như sau:

- Nếu PicOutputFlag bằng 1 và DpbOutputTime[n] bằng AuCpbRemovalTime[n], thì ảnh hiện thời được kết xuất.
- Theo cách khác, nếu PicOutputFlag bằng 0, thì ảnh hiện thời không được kết xuất, mà sẽ được lưu trữ trong bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB như được xác định ở tiểu mục C.3.4.

Theo một phương án: Theo cách khác, nếu PicOutputFlag bằng 0, thì ảnh hiện thời không được kết xuất, mà sẽ được lưu trữ vào trong bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB như được xác định ở tiểu mục C.3.4.

- Theo cách khác, (PicOutputFlag bằng 1 và DpbOutputTime[n] lớn hơn AuCpbRemovalTime[n]), thì ảnh hiện thời được kết xuất trễ hơn và sẽ được lưu trữ vào trong bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB (như được xác định ở tiểu mục C.3.4) và được kết xuất tại thời điểm DpbOutputTime[n] trừ phi được chỉ báo là không được kết xuất bằng cách giải mã hoặc suy ra no_output_of_prior_pics_flag bằng 1 tại thời điểm trước DpbOutputTime[n].

Theo một phương án khác, các bước nêu trên được xác định đối với:

Việc kết xuất hình ảnh hiện thời nếu nuh_layer_id của nó thuộc lớp mà lớp này thuộc tập hợp lớp kết xuất tương ứng với điểm hoạt động (hiện thời).

Theo phương án khác nữa, các bước nêu trên được xác định đối với:

Việc kết xuất ảnh hiện thời (không cần kiểm tra liệu nó có thuộc TargetDecLayerIdList hoặc tập hợp lớp kết xuất đối với điểm hoạt động hiện thời hay không).

Khi kết xuất, thì ảnh bị cắt xén, sử dụng cửa sổ cắt xén phù hợp được xác định trong SPS hoạt động đối với ảnh đó.

Khi ảnh n là ảnh được kết xuất và không phải là ảnh cuối cùng của dòng bit được kết xuất, thì giá trị của biến DpbOutputInterval[n] được suy ra như sau:

$$\text{DpbOutputInterval}[n] = \text{DpbOutputTime}[\text{nextPicInOutputOrder}] - \text{DpbOutputTime}[n] \quad (\text{C-17})$$

trong đó nextPicInOutputOrder là ảnh tiếp theo hình ảnh n theo thứ tự kết xuất và có PicOutputFlag bằng 1.

C.3.4 Đánh dấu và lưu hình ảnh giải mã hiện thời

Quy trình được xác định trong tiểu mục này xảy ra tức thì tại thời điểm gỡ bỏ CBP của đơn vị truy cập n, CpbRemovalTime[n].

Hình ảnh giải mã hiện thời được lưu trữ trong DPB trong bộ đệm lưu trữ hình ảnh trống tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời, mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được tăng thêm một và ảnh hiện thời được đánh dấu “sử dụng để tham chiếu ngắn hạn”.

Theo một phương án:

Hình ảnh giải mã hiện thời với `nuh_layer_id` bằng `currLayerId` được lưu trữ trong DPB trong bộ đệm lưu trữ hình ảnh trống `PSB[currLayerId]` với `currLayerId` bằng giá trị `nuh_layer_id` của hình ảnh hiện thời trong DPB, mức đầy DPB cho giá trị `nuh_layer_id` của hình ảnh hiện thời `DPBFullness[currLayerId]` được tăng thêm một và ảnh hiện thời được đánh dấu “sử dụng để tham chiếu ngắn hạn”.

C.4 Tính phù hợp của dòng bit

Các đặc tính kỹ thuật ở tiểu mục C.4 áp dụng.

C.5 Tính phù hợp của bộ giải mã

F.8.1.1C.5.1 Những vấn đề chung

Các đặc tính kỹ thuật ở tiểu mục C.5.1 áp dụng.

Danh mục (1)

Như được sử dụng trong danh mục (1) nêu trên, PSB là để chỉ bộ đệm lưu trữ hình ảnh. `DPBFullness` là để chỉ biến được sử dụng để mô tả làm đầy của bộ đệm hình ảnh giải mã (DPB) 116.

Tiểu mục (1A) dưới đây đề xuất phương pháp thay thế cho Tiểu mục C.3.2 của Danh mục (1) để bao gồm các hệ thống và các phương pháp hiện thời. Trong một số cấu hình, Tiểu mục (1A) có thể chỉ trình bày các thay đổi cho Tiểu mục C.3.2 trong JCTVC-L1003. Tiểu mục (1A) có thể sử dụng các cờ `AuNoOutputOfPriorPicsFlag` và `AuNoRaslOutputFlag` được xác định như phần trên đây.

Tiểu mục (1A)

C.3.2 Gỡ bỏ các hình ảnh khỏi DPB

Việc gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra tức thì tại thời điểm gỡ bỏ CBP của đơn vị giải mã thứ nhất của hình ảnh hiện thời thuộc đơn vị truy cập n (chứa hình ảnh hiện thời) và tiến hành như sau:

- Quy trình giải mã đối với RPS như được xác định ở khoản mục 8.3.2 được viện dẫn.
- Khi ảnh hiện thời là hình ảnh IRAP với `AuNoRaslOutputFlag` bằng 1 không phải là ảnh 0, thì các bước theo thứ tự sau đây được áp dụng:
- Theo một cấu hình khác, khi ảnh hiện thời là hình ảnh IRAP với `NoRaslOutputFlag`

bằng 1 không phải là ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến `AuNoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm như sau:

- Nếu hình ảnh hiện thời là ảnh CRA, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 (không quan tâm đến giá trị `no_output_of_prior_pics_flag` đối với ảnh hiện thời hoặc các hình ảnh khác trong AU).

- Theo cách khác, nếu giá trị `pic_width_in_luma_samples`, `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering_minus1[HighestTid]` được suy ra từ SPS hoạt động lần lượt khác với giá trị `pic_width_in_luma_samples`, `pic_height_in_luma_samples` hoặc `sps_max_dec_pic_buffering_minus1[HighestTid]`, được suy ra từ SPS hoạt động đối với ảnh đứng trước, thì `AuNoOutputOfPriorPicsFlag` có thể (mà lẽ ra là không) được thiết đặt bằng 1 bởi bộ giải mã trong thử nghiệm, không quan tâm đến giá trị `no_output_of_prior_pics_flag`.

Lưu ý - mặc dù việc thiết đặt `AuNoOutputOfPriorPicsFlag` bằng `no_output_of_prior_pics_flag` được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt `AuNoOutputOfPriorPicsFlag` bằng 1 trong trường hợp này.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được đặt dựa trên giá trị `no_output_of_prior_pics_flag` đối với ảnh hiện thời và các hình ảnh khác trong AU như sau:

- `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 cho ít nhất một ảnh trong AU. Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0. Do đó trong trường hợp này `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0 nếu `no_output_of_prior_pics_flag` bằng 0 đối với tất cả các hình ảnh trong AU.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với ảnh hiện thời. Trường hợp khác, `AuNoOutputOfPriorPicsFlag` giữ nguyên không đổi.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với tất cả các hình ảnh trong AU. Trường hợp khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với tất cả các hình ảnh thuộc các lớp

kết xuất đích trong AU. Theo cách khác, AuNoOutputOfPriorPicsFlag được thiết đặt bằng 0.

- Theo cách khác, AuNoOutputOfPriorPicsFlag được thiết đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho ít nhất một ảnh thuộc các lớp kết xuất đích trong AU. Theo cách khác, AuNoOutputOfPriorPicsFlag được thiết đặt bằng 0.

2. Giá trị AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR, sao cho khi giá trị AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả các giá trị nuh_layer_id trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB đối với tất cả giá trị nuh_layer_id được thiết đặt bằng 0.

Theo cách khác, giá trị AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR, sao cho khi giá trị AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

Tiêu mục (1A)

Danh mục (2) dưới đây cung cấp các phần JCTVC-L1008 mà chúng có thể được thay đổi để điều chỉnh các hệ thống và các phương pháp hiện thời.

Danh mục 2

F.13 Bộ giải mã tham chiếu giả định

F.13.1 Những vấn đề chung

Các đặc tính kỹ thuật ở tiêu mục C.1 được áp dụng.

F.13.2 Hoạt động của bộ đệm hình ảnh giải mã (CPB)

Các đặc tính kỹ thuật ở mục C.2 được áp dụng.

F.13.3 Hoạt động của bộ đệm hình ảnh giải mã (DPB)

Các đặc tính kỹ thuật ở tiêu mục C.3 được áp dụng riêng cho mỗi tập hợp của các hình ảnh giải mã với giá trị cụ thể nuh_layer_id.

PicOutputFlag cho các hình ảnh không có trong lớp kết xuất đích được thiết đặt bằng

0.

Các hình ảnh giải mã với cùng thời gian kết xuất DPB và với PicOutputFlag bằng 1 được kết xuất theo thứ tự tăng dần lên của giá trị `nuh_layer_ids` của các hình ảnh giải mã này.

F.13.5 Tính phù hợp của bộ giải mã

F.13.5.1 Những vấn đề chung

Các đặc tính kỹ thuật ở tiểu mục C.5.1 được áp dụng.

F.13.5.2 Hoạt động của DPB theo thứ tự kết xuất

F.13.5.2.1 Những vấn đề chung

Bộ đệm hình ảnh giải mã chứa các bộ đệm lưu trữ hình ảnh. Mỗi lớp gồm tập hợp của các bộ đệm lưu trữ hình ảnh của lớp đó. Do vậy các bộ đệm lưu trữ hình ảnh của mỗi lớp được kết hợp với giá trị `nuh_layer_id` của lớp đó. Số lượng bộ đệm lưu trữ hình ảnh cho giá trị `nuh_layer_id` bằng 0 được suy ra từ SPS hoạt động của lớp có giá trị `nuh_layer_id` bằng 0. Số lượng bộ đệm lưu trữ hình ảnh cho mỗi giá trị `nuh_layer_id` khác 0 được suy ra từ SPS của lớp hoạt động cho giá trị `nuh_layer_id` khác 0. Mỗi bộ đệm lưu trữ hình ảnh chứa hình ảnh mã hoá được đánh dấu “sử dụng để tham chiếu” hoặc được giữ lại để kết xuất trong tương lai. Quy trình kết xuất và gỡ bỏ các hình ảnh khỏi DPB như được xác định ở khoản mục F.13.5.2.2 được viện dẫn, tiếp theo là sự viện dẫn quy trình giải mã hình ảnh, đánh dấu, đệm bổ sung và lưu trữ như được xác định ở khoản mục F.13.5.2.3. Quy trình “đệm” được xác định ở khoản mục F.13.5.2.4 và được viện dẫn như được xác định ở các tiểu mục F.13.5.2.2 và F.13.5.2.3.

F. 13.5.2.2 Kết xuất và gỡ bỏ các hình ảnh khỏi DPB

Việc kết xuất và gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra tức thì khi đơn vị giải mã thứ nhất của hình ảnh hiện thời thuộc đơn vị truy cập chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB và tiến hành như sau.

Quy trình giải mã cho RPS như được xác định ở khoản mục 8.3.2 được viện dẫn.

- Nếu hình ảnh hiện thời là hình ảnh IRAP với `NoRaslOutputFlag` bằng 1 và với giá trị `nuh_layer_id` bằng 0 không phải là ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến `NoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm như

sau:

- Nếu hình ảnh hiện thời là ảnh CRA, NoOutputOfPriorPicsFlag được thiết đặt bằng 1 (không quan tâm đến giá trị no_output_of_prior_pics_flag).
- Theo cách khác, khi giá trị pic_width_in_luma_samples, pic_height_in_luma_samples hoặc SPS max_dcc_pic_buffering_minus1[HighestTid] được suy ra từ SPS hoạt động tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời lần lượt khác với giá trị pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minus1[HighestTid], được suy ra từ SPS hoạt động đối với ảnh đứng trước với giá trị nuh_layer_id bằng giá trị nuh_layer_id của hình ảnh hiện thời, thì NoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được thiết đặt bằng 1 bởi bộ giải mã trong thử nghiệm, không quan tâm đến giá trị no_output_of_prior_pics_flag.

Lưu ý - mặc dù việc xác định NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

- Theo cách khác, NoOutputOfPriorPicsFlag được thiết đặt bằng no_output_of_prior_pics_flag.

2. Giá trị NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR như sau:

- Nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

Theo một phương án, nếu NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả các giá trị nuh_layer_id trong DPB được làm rỗng không kết xuất các hình ảnh chúng chứa và sự mức đầy DPB đối với tất cả giá trị nuh_layer_id được thiết đặt bằng 0.

Theo một phương án, nếu NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh PSB[currLayerId] tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy

DPB DPBFullness[nuh_layer_id] được thiết đặt bằng 0.

Theo một phương án, nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] cho tất cả các giá trị nuh_layer_id trong DPB được làm rỗng mà không kết xuất các hình ảnh chúng chứa và mức đầy DPB DPBFullness[nuh_layer_id] đối với tất cả các giá trị nuh_layer_id được thiết đặt bằng 0.

- Theo cách khác, (NoOutputOfPriorPicsFlag bằng 0), thì tất cả các bộ đệm lưu trữ hình ảnh chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình “đệm” được xác định ở khoản mục F. 13.5.2.4 và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

- Theo cách khác, khi NoOutputOfPriorPicsFlag bằng 0, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả các giá trị nul _ layer _id chứa hình ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống tương ứng với tất cả các giá trị nul _ layer _id trong DPB được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình “đệm” được xác định ở khoản mục F. 13.5.2.4 và mức đầy DPB đối với tất cả giá trị nuh_layer_id được thiết đặt bằng 0.

- Theo cách khác, theo cách khác, (NoOutputOfPriorPicsFlag bằng 0), thì tất cả các bộ đệm lưu trữ hình ảnh chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống PSB[nuh_layer_id] trong DPB tương ứng với giá trị nuh_layer_id currLayerId của hình ảnh hiện thời được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở khoản mục F.13.5.2.4 và mức đầy DPB DPBFullness[nuh_layer_id] cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

- Theo cách khác, khi NoOutputOfPriorPicsFlag bằng 0, tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] đối với tất cả giá trị nuh_layer_id trong DPB chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống PSB[nuh_layer_id] tương ứng với tất cả giá trị nuh_layer_ids trong DPB được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở khoản mục F.13.5.2.4 và mức đầy DPB DPBFullness[nuh_layer_id] đối với tất cả giá trị nuh_layer_id được thiết đặt bằng 0.

- Theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB tương ứng với giá trị nuh_layer_id của hình ảnh giải mã hiện thời bị giảm đi một. Biến currLayerId được thiết đặt bằng nuh_layer_id của hình ảnh giải mã hiện thời và khi một hoặc một số các điều kiện sau đây là đúng, thì quy trình đệm được xác định ở khoản mục F.13.5.2.4 được viện dẫn lặp đi lặp lại trong lúc mức đầy DPB bị giảm tiếp một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung tương ứng với giá trị nuh_layer_id của hình ảnh giải mã hiện thời được làm rỗng, cho đến khi không một điều kiện nào sau đây là đúng:

- Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB mà chúng được đánh dấu “không cần kết xuất” lớn hơn sps_max_num_reorderpics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho giá trị currLayerId.

- sps_max_latency_increase_plus1[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS của lớp hoạt động cho giá trị currLayerId là không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu “không cần kết xuất”, mà đối với nó, biến liên quan PicLatencyCount[currLayerId] lớn hơn hoặc bằng SpsMaxLatencyPictures[HighestTid] được suy ra từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho giá trị currLayerId.

- Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB lớn hơn hoặc bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho giá trị currLayerId.

- Theo cách khác, khác đi thì (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả giá trị nuh_layer_id chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB tương ứng với giá trị nuh_layer_id của hình ảnh bị giảm đi một.

- Theo cách khác, theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] trong DPB tương ứng với giá trị nuh_layer_id currLayerId của hình

ảnh hiện thời chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB DPBFullness[nuh_layer_id] tương ứng với giá trị nuh_layer_id của hình ảnh giải mã hiện thời bị giảm đi một.

Theo cách khác, theo cách khác, (hình ảnh hiện thời không phải là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với nuh_layer_id bằng 0), tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] đối với tất cả giá trị nuh_layer_id trong DPB chứa ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm rỗng, mức đầy DPB DPBFullness[nuh_layer_id] tương ứng với giá trị nuh_layer_id của hình ảnh trống bị giảm đi một.

F.13.5.2.3 Giải mã, đánh dấu, đệm bổ sung và lưu trữ hình ảnh

Các quy trình được xác định cụ thể trong tiêu mục này xảy ra tức thì khi đơn vị giải mã cuối cùng của đơn vị truy cập n chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB.

Biến currLayerId được thiết đặt bằng nuh_layer_id của hình ảnh giải mã hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu “không cần kết xuất” và có giá trị nuh_layer_id bằng currLayerId, biến liên quan PicLatencyCount[currLayerId] được thiết đặt bằng PicLatencyCount[currLayerId] + 1.

Ảnh hiện thời được xem là đã được giải mã sau khi đơn vị giải mã cuối cùng của hình ảnh được giải mã. Hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống tương ứng với currLayerId (giá trị nuh_layer_id của hình ảnh hiện thời) trong DPB và áp dụng theo quy tắc như sau:

- Nếu hình ảnh giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu “không cần kết xuất” và biến liên quan của nó PicLatencyCount[currLayerId] được thiết đặt bằng 0.
- Theo cách khác, (hình ảnh giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu “không cần kết xuất”.

Theo một phương án, hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống tương ứng với currLayerId (giá trị nuh_layer_id của hình ảnh hiện thời) trong DPB và áp dụng theo quy tắc như sau:

- Nếu hình ảnh giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu “không cần kết xuất” và biến liên quan của nó PicLatencyCount[currLayerId] được thiết đặt bằng 0. Tất cả các bộ đệm lưu trữ hình ảnh chứa ảnh có cùng giá trị đếm thứ tự ảnh (PicOrderCntVal) như là hình ảnh giải mã hiện thời được đánh dấu “không cần kết xuất”.
- Trong trường hợp khác, (hình ảnh giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu “không cần kết xuất”.
- Theo một phương án, hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống PSB[curreLayerId] tương ứng với currLayerId (giá trị nuh_layer_id của hình ảnh hiện thời) trong DPB và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời DPBFullness[nuh_layer_id] được tăng thêm một và áp dụng quy tắc như sau:

Hình ảnh giải mã hiện thời được đánh dấu “sử dụng để tham chiếu ngắn hạn”.

Khi một hoặc một số các điều kiện sau đây là đúng, quy trình đệm được xác định ở khoản mục F. 13.5.2.4 được viện dẫn lặp đi lặp lại cho đến khi không một điều kiện nào sau đây là đúng.

- Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu “không cần kết xuất” lớn hơn sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho giá trị currLayerId.
- sps_max_latency_increase_plus1[HighestTid] không bằng 0 và có ít nhất một ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu “không cần kết xuất” cho biến liên quan PicLatencyCount[currLayerId] lớn hơn hoặc bằng SpsMaxLatencyPicturesf[HighestTid] được suy ra từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho giá trị currLayerId.

F.13.5.2.4 Quy trình “đệm”

Quy trình đệm gồm các bước theo thứ tự như sau:

1. Các hình ảnh mà chúng được kết xuất trước tiên được chọn làm các hình ảnh có giá trị nhỏ nhất PicOrderCntVal trong số tất cả các hình ảnh trong DPB được đánh dấu “không cần kết xuất”.
2. Các hình ảnh này được cắt xén, sử dụng cửa sổ cắt xén phù hợp được xác định trong SPS hoạt động đối với hình ảnh với nuh_layer_id bằng 0 hoặc trong SPS của lớp hoạt động cho giá trị nuh_layer_id bằng giá trị của hình ảnh, các hình ảnh được cắt xén được kết xuất

theo thứ tự tăng dần lên của `nuh_layer_id` và các hình ảnh này được đánh dấu “không cần kết xuất”

3. Mỗi bộ đệm lưu trữ hình ảnh chứa ảnh được đánh dấu “không sử dụng để tham chiếu” và mỗi bộ đệm lưu trữ hình ảnh đó bao gồm một ảnh mà đã được cắt xén và được đầu ra được làm rỗng.

Danh mục (2)

Tiêu mục (2A) dưới đây đề xuất phương pháp thay thế cho phần F.13.5.2.2 của Danh mục (2) để điều chỉnh các hệ thống và các phương pháp hiện thời. Trong một số cấu hình, Tiêu mục (2A) có thể chỉ trình bày các thay đổi cho phần F.13.5.2.2 trong JCTVC-L1008. Tiêu mục (2A) có thể sử dụng các cờ `AuNoOutputOfPriorPicsFlag` và `AuNoRaslOutputFlag` được xác định như phân trên đây.

Tiêu mục 2A

F.13.5.2.2 Kết xuất và gỡ bỏ các hình ảnh khỏi DPB

Việc kết xuất và gỡ bỏ các hình ảnh khỏi DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân giải cú pháp tiêu đề phiên của phiên thứ nhất của hình ảnh hiện thời) xảy ra tức thì khi đơn vị giải mã thứ nhất của đơn vị truy cập chứa hình ảnh hiện thời được gỡ bỏ khỏi CPB và tiến hành như sau:

Quy trình giải mã đối với RPS như được xác định ở khoản mục 8.3.2 được viện dẫn.

- Nếu hình ảnh hiện thời là hình ảnh IRAP với `AuNoRaslOutputFlag` bằng 1 và với `nuh_layer_id` bằng 0 không phải là ảnh 0, các bước theo thứ tự sau đây được áp dụng:

- Theo cấu hình khác, nếu hình ảnh hiện thời là hình ảnh IRAP với `NoRaslOutputFlag` bằng 1 và với `nuh_layer_id` bằng 0 không phải là ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến `AuNoOutputOfPriorPicsFlag` được suy ra cho bộ giải mã trong thử nghiệm như sau:

- Nếu hình ảnh hiện thời có `cnuh_layer_id` bằng 0, thì `AuNoOutputOfPriorPicsFlag` được bắt đầu bằng 0. Sau đó:

- Nếu hình ảnh hiện thời là ảnh CRA, thì `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 (không quan tâm đến giá trị `no_output_of_prior_pics_flag` đối với

ảnh hiện thời hoặc các hình ảnh khác trong AU).

- Theo cách khác, nếu giá trị `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, hoặc `sps_max_dec_pic_buffering_minus1[HighestTid]` được suy ra từ SPS hoạt động lần lượt khác với giá trị `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, hoặc `sps_max_dec_pic_buffering_minus1[HighestTid]`, được suy ra từ SPS hoạt động đối với ảnh đứng trước, thì `AuNoOutputOfPriorPicsFlag` có thể (mà lẽ ra là không) được thiết đặt bằng 1 bởi bộ giải mã trong thử nghiệm, không quan tâm đến giá trị `no_output_of_prior_pics_flag`.

Lưu ý - mặc dù việc xác định `AuNoOutputOfPriorPicsFlag` bằng `no_output_of_prior_pics_flag` được ưu tiên trong các điều kiện này, bộ giải mã trong thử nghiệm được phép thiết đặt `AuNoOutputOfPriorPicsFlag` bằng 1 trong trường hợp này.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được đặt dựa trên giá trị `no_output_of_prior_pics_flag` đối với ảnh hiện thời và các hình ảnh khác trong AU như sau:

- `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 cho ít nhất một ảnh trong AU. Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0. Do đó trong trường hợp này, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0 nếu `no_output_of_prior_pics_flag` bằng 0 đối với tất cả các hình ảnh trong AU.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với ảnh hiện thời. Theo cách khác, `AuNoOutputOfPriorPicsFlag` giữ nguyên không đổi.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với tất cả các hình ảnh trong AU. Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu `no_output_of_prior_pics_flag` bằng 1 đối với tất cả các hình ảnh thuộc các lớp kết xuất đích trong AU. Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 0.

- Theo cách khác, `AuNoOutputOfPriorPicsFlag` được thiết đặt bằng 1 nếu

no_output_of_prior_pics_flag bằng 1 cho ít nhất một ảnh thuộc lớp kết xuất đích trong AU. Theo cách khác, AuNoOutputOfPriorPicsFlag được thiết đặt bằng 0.

2. Giá trị AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã trong thử nghiệm được áp dụng cho HDR như sau:

- Nếu AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh trong DPB tương ứng với tất cả giá trị nuh_layer_id được làm rỗng mà không kết xuất các hình ảnh mà chúng bao gồm và mức đầy DPB được thiết đặt bằng 0 đối với tất cả giá trị nuh_layer_id.

- Theo cách khác, nếu AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh trong DPB tương ứng với tất cả giá trị nuh_layer_id của hình ảnh hiện thời được làm rỗng mà không kết xuất các hình ảnh mà chúng bao gồm và mức đầy DPB được thiết đặt bằng 0 cho giá trị nuh_layer_id của hình ảnh hiện thời.

Theo cách khác, (AuNoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả giá trị nuh_layer_id bao gồm ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở khoản mục F.13.5.2.4 và mức đầy DPB đối với tất cả giá trị nuh_layer_id được thiết đặt bằng 0.

Theo cách khác, (AuNoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời bao gồm ảnh được đánh dấu “không cần kết xuất” và “không sử dụng để tham chiếu” được làm rỗng (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm rỗng bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở khoản mục F.13.5.2.4 và mức đầy DPB cho giá trị nuh_layer_id của hình ảnh hiện thời được thiết đặt bằng 0.

Tiểu mục (2A)

Theo một cấu hình, đối với văn bản dự kiến trong Danh mục (1), tiểu mục (1A), Danh mục (2) và tiểu mục (2A) nêu trên, tất cả những chỗ xuất hiện “ảnh hiện thời” có thể được thay thế bằng “hình ảnh giải mã hiện thời”.

Danh mục (3) dưới đây cung cấp phần bổ sung đối với việc giải mã để điều chỉnh các hệ thống và các phương pháp hiện thời. Danh mục (3) cung cấp các tiểu mục của JCTVC-

L1003 (nghĩa là tiêu chuẩn HEVC phiên bản 34) mà chúng có thể được thay đổi để bao gồm các hệ thống và các phương pháp hiện thời.

Danh mục 3

8.3.2. Quy trình giải mã dùng cho tập hình ảnh tham chiếu

Khi ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1, tất cả các hình ảnh tham chiếu hiện thời trong DPB tương ứng với giá trị nuh_layer_id của hình ảnh hiện thời (nếu có) được đánh dấu “không sử dụng để tham chiếu”.

Theo một phương án khác: Khi ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1, tất cả các hình ảnh tham chiếu hiện có trong DPB tương ứng với tất cả các giá trị nuh_layer_id (nếu có) được đánh dấu “không sử dụng để tham chiếu”.

Danh mục (3)

Nhiều biến đổi khác nhau có thể được sử dụng để xác định hoạt động của các bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116 với nhiều bước khác nhau dựa trên hình ảnh hoặc dựa trên đơn vị truy cập. Các bảng từ (11) đến (15) dưới đây liệt kê nhiều biến đổi của các bước đó và tương ứng với các thiết đặt dựa trên hình ảnh/dựa trên đơn vị truy cập (AU).

Bảng 46

Bước	Dựa trên hình ảnh/ Dựa trên AU
Gỡ bỏ	Dựa trên hình ảnh
Kết xuất	Dựa trên AU
Lưu trữ	Dựa trên hình ảnh
Đánh dấu	Dựa trên AU để đánh dấu kết xuất, Dựa trên hình ảnh để đánh dấu tham chiếu
Kết xuất bổ sung	Dựa trên AU

Bảng (11)

Bảng 47

Bước	Dựa trên hình ảnh/ Dựa trên AU
Gỡ bỏ	Dựa trên hình ảnh
Kết xuất	Dựa trên AU
Lưu trữ	Dựa trên AU
Đánh dấu	Dựa trên AU để đánh dấu kết xuất, Dựa trên AU để đánh dấu tham chiếu
Kết xuất bổ sung	Dựa trên AU

Bảng (12)

Bảng 48

Bước	Dựa trên hình ảnh/ Dựa trên AU
Gỡ bỏ	Dựa trên AU
Kết xuất	Dựa trên AU
Lưu trữ	Dựa trên AU
Đánh dấu	Dựa trên AU để đánh dấu kết xuất, Dựa trên AU để đánh dấu tham chiếu
Kết xuất bổ sung	Dựa trên AU

Bảng (13)

Bảng 49

Bước	Dựa trên hình ảnh/ Dựa trên AU
Gỡ bỏ	Dựa trên hình ảnh
Kết xuất	Dựa trên AU
Lưu trữ	Dựa trên hình ảnh
Đánh dấu	Dựa trên hình ảnh để đánh dấu kết xuất, Dựa trên hình ảnh để đánh dấu tham chiếu
Kết xuất bổ sung	Dựa trên AU

Bảng (14)

Bảng 50

Bước	Dựa trên hình ảnh/ Dựa trên AU
Gỡ bỏ	Dựa trên hình ảnh
Kết xuất	Dựa trên hình ảnh
Lưu trữ	Dựa trên hình ảnh
Đánh dấu	Dựa trên hình ảnh để đánh dấu kết xuất, Dựa trên hình ảnh để đánh dấu tham chiếu
Kết xuất bổ sung	Dựa trên hình ảnh

Bảng (15)

Fig.66 là sơ đồ khối thể hiện một cấu hình của bộ giải mã 2404. Bộ giải mã 2404 có thể có trong thiết bị điện tử 2402. Ví dụ, bộ giải mã 2404 có thể là bộ giải mã mã hóa video hiệu suất cao (HEVC). Bộ giải mã 2404 và/hoặc một hoặc nhiều thành phần được minh họa như có trong bộ giải mã 2404 có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Bộ giải mã 2404 có thể thu dòng bit 2414 (ví dụ, một hoặc nhiều hình ảnh mã hoá có trong dòng bit 2414) để giải mã. Trong một số cấu hình, dòng bit thu được 2414 có thể bao gồm thông tin bổ sung thu được, như tiêu đề phiên thu được, PPS thu được, thông

tin mô tả bộ đệm thu được v.v.. Các hình ảnh mã hoá có trong dòng bit 2414 có thể bao gồm một hoặc nhiều hình ảnh tham chiếu mã hoá và/hoặc một hoặc nhiều hình ảnh mã hoá khác.

Các biểu tượng thu được (trong một hoặc nhiều hình ảnh mã hoá có trong dòng bit 2414) có thể được giải mã entropy bằng môđun giải mã entropy 454, nhờ đó tạo ra tín hiệu thông tin chuyển động 456 và các hệ số lượng tử hóa, định tỷ lệ và/hoặc biến đổi 458.

Tín hiệu thông tin chuyển động 456 có thể được kết hợp với một phần tín hiệu khung tham chiếu 484 từ bộ nhớ khung 464 tại môđun bù chuyển động 460, tín hiệu thông tin chuyển động này có thể tạo ra tín hiệu dự báo liên khung 468. Các hệ số lượng tử hóa, định tỷ lệ và/hoặc biến đổi 458 có thể được lượng tử hóa ngược, được định tỷ lệ và được biến đổi ngược bởi môđun ngược 462, nhờ đó tạo ra tín hiệu dư giải mã 470. Tín hiệu dư giải mã 470 này có thể được bổ sung vào tín hiệu dự báo 478 để tạo ra tín hiệu kết hợp 472. Tín hiệu dự báo 478 có thể là tín hiệu được chọn từ hoặc tín hiệu dự báo liên khung 468 hoặc tín hiệu dự báo trong khung 476 được tạo ra bởi môđun dự báo trong khung 474. Trong một số cấu hình, việc chọn tín hiệu này có thể dựa trên (ví dụ, được điều khiển bởi) dòng bit 2414.

Tín hiệu dự báo trong khung 476 có thể được dự báo từ thông tin giải mã trước đây từ tín hiệu kết hợp 472 (trong khung hiện thời, ví dụ). Tín hiệu kết hợp 472 này cũng có thể được lọc bởi bộ lọc giải khối 480. Tín hiệu lọc sinh ra 482 có thể được ghi vào bộ nhớ khung 464. Tín hiệu lọc sinh ra 482 có thể bao gồm hình ảnh giải mã.

Bộ nhớ khung 464 này có thể bao gồm bộ đệm hình ảnh giải mã (DPB) 2416 như được mô tả trong bản mô tả này. Bộ đệm hình ảnh giải mã (DPB) 2416 có thể có khả năng thực hiện các hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116. Bộ đệm hình ảnh giải mã (DPB) 2416 có thể bao gồm một hoặc nhiều hình ảnh giải mã mà chúng có thể được duy trì làm các khung tham chiếu ngắn hoặc dài hạn. Bộ nhớ khung 464 có thể bao gồm thông tin bổ sung tương ứng với các hình ảnh giải mã. Ví dụ, bộ nhớ khung 464 có thể bao gồm các tiêu đề phiên, thông tin tập tham số video (VPS), thông tin tập tham số chuỗi (SPS), thông tin tập tham số hình ảnh (PPS), các tham số chu kỳ, thông tin mô tả bộ đệm v.v.. Một hoặc nhiều mảnh thông tin này có thể được truyền tín hiệu từ bộ mã hóa (ví dụ, bộ mã hóa 2108, môđun truyền tín hiệu thời gian tốn thêm 2112).

Fig.67A là sơ đồ khối thể hiện việc sử dụng cả lớp nâng cao và lớp cơ sở để mã hóa video với các bộ đệm hình ảnh giải mã (DPB) riêng 516a-b và các môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 520a-b tách biệt cho lớp cơ sở và lớp nâng cao. Thiết bị điện tử thứ nhất 502a và thiết bị điện tử thứ hai 502b được minh họa. Thiết bị điện tử thứ

nhất 502a có thể bao gồm bộ mã hóa video 508 bao gồm bộ mã hóa lớp nâng cao 526 và bộ mã hóa lớp cơ sở 528. Mỗi thành phần này được bao gồm bên trong thiết bị điện tử thứ nhất 502a (nghĩa là bộ mã hóa lớp nâng cao 526 và bộ mã hóa lớp cơ sở 528) có thể được thực hiện trong phần cứng, phần mềm hoặc kết hợp cả hai. Thiết bị điện tử thứ nhất 502a có thể được suy ra hình ảnh đầu vào 2506. Trong một số cấu hình, hình ảnh đầu vào 2506 có thể được chụp trên thiết bị điện tử thứ nhất 502a sử dụng cảm biến ảnh, được truy hồi từ bộ nhớ hoặc thu được từ thiết bị điện tử khác 502.

Bộ mã hóa lớp nâng cao 526 có thể mã hóa hình ảnh đầu vào 2506 để tạo ra dữ liệu mã hoá. Ví dụ, bộ mã hóa lớp nâng cao 526 có thể mã hóa một loạt hình ảnh đầu vào 2506 (ví dụ, video). Dữ liệu mã hoá có thể có trong dòng bit video của lớp nâng cao đã mã hóa 530. Bộ mã hóa lớp nâng cao 526 có thể tạo ra việc truyền tín hiệu bổ sung dựa trên hình ảnh đầu vào 2506.

Bộ giải mã video của lớp nâng cao 534 có thể bao gồm bộ đệm hình ảnh giải mã (DPB) 516a và môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 520a. Tương tự, bộ giải mã của lớp cơ sở 536 có thể bao gồm bộ đệm hình ảnh giải mã (DPB) 516b và môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 520b.

Bộ mã hóa lớp cơ sở 528 cũng có thể mã hóa hình ảnh đầu vào 2506. Theo một cấu hình, cùng hình ảnh đầu vào 2506 được sử dụng bởi bộ mã hóa lớp nâng cao 526 cũng có thể được sử dụng bởi bộ mã hóa lớp cơ sở 528. Theo cấu hình khác, hình ảnh đầu vào khác (nhưng tương tự) với hình ảnh đầu vào 2506 được sử dụng bởi bộ mã hóa lớp nâng cao 526 có thể được sử dụng bởi bộ mã hóa lớp cơ sở 528. Ví dụ, cho khả năng mở rộng tỷ lệ tín hiệu - nhiễu (còn được gọi là khả năng mở rộng chất lượng), cùng hình ảnh đầu vào 2506 có thể được sử dụng bởi cả bộ mã hóa lớp nâng cao 526 và bộ mã hóa lớp cơ sở 528. Theo ví dụ khác, theo khả năng điều chỉnh được về mặt không gian, ảnh được giảm lấy mẫu có thể được sử dụng bởi bộ mã hóa lớp cơ sở 528. Theo ví dụ khác nữa, cho khả năng mở rộng đa cảnh, ảnh quan sát khác có thể được sử dụng bởi bộ mã hóa lớp cơ sở 528. Bộ mã hóa lớp cơ sở 528 có thể tạo ra dữ liệu đã mã hoá có trong dòng bit video của lớp cơ sở đã mã hoá 532.

Mỗi dòng bit video của lớp nâng cao đã mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532 có thể bao gồm dữ liệu đã mã hoá dựa trên hình ảnh đầu vào 2506. Theo một ví dụ, dòng bit video của lớp nâng cao đã mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532 có thể bao gồm dữ liệu hình ảnh mã hoá. Trong một số cấu hình, dòng bit video của

lớp nâng cao đã mã hoá 530 và/hoặc dòng bit video của lớp cơ sở đã mã hoá 532 này có thể bao gồm dữ liệu tiêu đề, như thông tin tập tham số chuỗi (SPS), thông tin tập tham số hình ảnh (PPS), thông tin tập tham số video (VPS), thông tin tiêu đề phiên v.v..

Dòng bit video của lớp nâng cao đã mã hoá 530 có thể được cung cấp cho thiết bị điện tử thứ hai 502b. Tương tự, dòng bit video của lớp cơ sở đã mã hoá 532 có thể được cung cấp cho thiết bị điện tử thứ hai 502b. Thiết bị điện tử thứ hai 502b này có thể bao gồm bộ giải mã video 2504. Bộ giải mã video 2504 có thể bao gồm bộ giải mã lớp nâng cao 534 và bộ giải mã lớp cơ sở 536. Theo một cấu hình, dòng bit video của lớp cơ sở đã mã hoá 530 được giải mã bởi bộ giải mã của lớp cơ sở 536 trong khi dòng bit video của lớp nâng cao đã mã hoá 530 được giải mã bởi bộ giải mã lớp nâng cao 534.

Theo một ví dụ, dòng bit video của lớp nâng cao đã mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532 có thể được truyền đến thiết bị điện tử thứ hai 502b sử dụng kết nối hữu tuyến hoặc vô tuyến. Trong một số trường hợp, việc này có thể được thực hiện qua mạng, như Internet, mạng cục bộ (LAN) hoặc loại mạng khác để truyền thông giữa các thiết bị. Cần lưu ý rằng trong một số cấu hình, các bộ mã hóa (nghĩa là bộ mã hóa lớp nâng cao 526 và bộ mã hóa lớp cơ sở 528) và bộ giải mã 2504 (ví dụ, bộ giải mã lớp cơ sở 536 và bộ giải mã lớp nâng cao 534) có thể được thực hiện trên cùng thiết bị điện tử 502 (nghĩa là thiết bị điện tử thứ nhất 502a và thiết bị điện tử thứ hai 502b có thể là tiểu mục của thiết bị điện tử 502 đơn lẻ). Theo việc thực hiện trong đó các bộ mã hóa và các bộ giải mã được thực hiện trên cùng thiết bị điện tử 502, chẳng hạn, dòng bit video của lớp nâng cao mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532 có thể sử dụng được cho bộ giải mã video 2504 theo nhiều cách. Ví dụ, dòng bit video của lớp nâng cao đã mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532 có thể được cung cấp qua bus cho bộ giải mã video 2504 hoặc được lưu trữ trong bộ nhớ để truy hồi từ bộ giải mã video 2504.

Bộ giải mã video 2504 có thể tạo ra một hoặc nhiều hình ảnh mã hoá dựa trên dòng bit video của lớp nâng cao đã mã hoá 530 và dòng bit video của lớp cơ sở đã mã hoá 532. Hình ảnh giải mã 2118 (nó có thể bao gồm hình ảnh giải mã của lớp nâng cao 538 và hình ảnh giải mã của lớp cơ sở 540) có thể được hiển thị, được phát lại, được lưu trữ trong bộ nhớ và/hoặc được truyền đến thiết bị khác v.v..

Theo một ví dụ, hình ảnh giải mã 2118 có thể được truyền đến thiết bị khác hoặc quay trở về thiết bị điện tử thứ nhất 502a. Hình ảnh giải mã 2118 cũng có thể được lưu trữ hoặc theo cách khác, được duy trì trên thiết bị điện tử thứ hai 502b. Trong ví dụ khác, thiết

bị điện tử thứ hai 502b có thể hiển thị hình ảnh giải mã 2118. Trong các cấu hình khác, hình ảnh giải mã 2118 bao gồm các thành phần của hình ảnh đầu vào 2506 với các đặc tính khác nhau dựa trên việc mã hóa và các hoạt động khác được thực hiện trên dòng bit 2114. Trong một số cấu hình, hình ảnh giải mã 2118 có thể có trong chuỗi ảnh với độ phân giải, định dạng, đặc tính kỹ thuật khác nhau hoặc thuộc tính khác từ hình ảnh đầu vào 2506.

Fig.67B là sơ đồ khối thể hiện việc sử dụng bộ đệm hình ảnh giải mã (DPB) dùng chung 516c và môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 520c dùng chung cho lớp cơ sở và lớp nâng cao. Fig.67B bao gồm cùng các bộ phận như thể hiện trên Fig.67A ngoại trừ bộ giải mã video của lớp nâng cao 534 và bộ giải mã của lớp cơ sở 536 chia sẻ cả bộ đệm hình ảnh giải mã (DPB) 516c và môđun vận hành bộ đệm hình ảnh giải mã (DPB) hỗn hợp 520c.

Fig.68 thể hiện sơ đồ định thời mô tả hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116. Các hoạt động của bộ đệm hình ảnh giải mã (DPB) hỗn hợp 116 trên Fig.68 trình bày các bước của biến đổi được ưu tiên trong đó việc gỡ bỏ, lưu trữ và đánh dấu tham chiếu được dựa trên hình ảnh và việc kết xuất, đánh dấu kết xuất và kết xuất bổ sung được dựa trên đơn vị truy cập (AU). Bộ đệm hình ảnh giải mã (DPB) 116 lý tưởng và bộ giải mã tham chiếu giả định (HRD) có thể hoạt động theo cách thức mà nhiều bước đơn lẻ được thể hiện (ví dụ, bước gỡ bỏ 621a-b, bước kết xuất 623, bước lưu trữ 625a-c, bước đánh dấu 627a-c, 629 và bước kết xuất bổ sung 631) tất cả đều được thực hiện tức thì. Chuỗi của các bước này và các dịch vị định thời được minh họa giữa các bước đơn lẻ cho mục đích minh họa. Đối với tính phù hợp của bộ giải mã định thời kết xuất, định thời (liên quan đến thời gian cung cấp bit thứ nhất) của bước kết xuất hình ảnh là giống nhau đối với cả bộ giải mã tham chiếu giả định (HRD) và bộ giải mã trong thử nghiệm (DUT) cho tới khoảng trễ cố định. Do đó, dịch vị định thời được minh họa có thể xảy ra cho DUT tạo ra khoảng trễ cố định so với HRD.

Các bước cho bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b và bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644 được minh họa. Thời gian gỡ bỏ bộ đệm hình ảnh mã hoá (CPB) 2646 được minh họa. Sau thời gian gỡ bỏ 2646 của bộ đệm hình ảnh mã hoá (CPB), việc gỡ bỏ (không có kết xuất) dựa trên hình ảnh 621a có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b và việc gỡ bỏ (không có kết xuất) dựa trên hình ảnh 621b có thể được thực hiện bởi bộ đệm

hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644. Sau khi dịch vị định thời, việc kết xuất (đệm) ảnh dựa trên đơn vị truy cập (AU) 623 có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b.

Sau dịch vị định thời khác, quá trình 613 liên quan đến hình ảnh giải mã hiện thời được minh họa. bước lưu trữ dựa trên hình ảnh 625a-c có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b. bước lưu trữ có thể được thực hiện sau khi mỗi trong số các hình ảnh của lớp cơ sở (BL) 2644, lớp nâng cao thứ nhất (EL1) 2642a và lớp nâng cao thứ hai (EL2) 2642b được giải mã. bước lưu trữ 625 có thể được chia nhỏ hơn. Trong các bước lưu trữ 625, ảnh giải mã được lưu trữ trong bộ đệm hình ảnh giải mã (DPB) 116 trong bộ đệm lưu trữ trống và làm đầy của bộ đệm hình ảnh giải mã (DPB) được tăng thêm một. Ngoài ra, khi ảnh được gỡ bỏ (không có kết xuất) từ bộ đệm hình ảnh giải mã (DPB) 116, thì làm đầy của bộ đệm hình ảnh giải mã (DPB) được giảm bớt một. Tương tự, khi ảnh được đệm từ bộ đệm hình ảnh giải mã (DPB) 116 (trong hoặc bước đệm hoặc bước đệm bổ sung), làm đầy của bộ đệm hình ảnh giải mã (DPB) được giảm bớt một.

Bộ đệm hình ảnh giải mã (DPB) 116 có thể bao gồm các bộ đệm ảnh xác định và quản lý một cách riêng đối với các hình ảnh mã hoá có các đặc điểm khác nhau. Ví dụ, bộ đệm hình ảnh giải mã (DPB) 116 có thể bao gồm các bộ đệm ảnh xác định và quản lý một cách riêng đối với các hình ảnh mã hoá với các độ phân giải khác nhau, độ sâu khác nhau và/hoặc màu sắc khác nhau.

Hình ảnh giải mã có thể được lưu trữ trong một nhóm bộ đệm lưu trữ hình ảnh chung trong bộ đệm hình ảnh giải mã (DPB) 116. Ví dụ, hai trường hợp phụ bổ sung có thể được sử dụng để xác định các ràng buộc kích thước của bộ đệm hình ảnh giải mã (DPB) 116 tác động quá trình đệm/gỡ bỏ và xác định mức. Theo ràng buộc của bộ đệm hình ảnh giải mã (DPB) 116 dựa trên byte, thì hình ảnh giải mã có thể được lưu trữ với cân nhắc về kích thước dựa trên độ phân giải và/hoặc độ sâu bit. Các ràng buộc kích thước của bộ đệm hình ảnh giải mã (DPB) 116 có thể được xác định theo giới hạn byte cân nhắc độ phân giải và độ sâu bit của mỗi hình ảnh giải mã. Theo ràng buộc của bộ đệm hình ảnh giải mã (DPB) 116 dựa trên đơn vị ảnh, thì hình ảnh giải mã có thể được lưu trữ (và được cân nhắc chiếm một

khe của bộ đệm ảnh). Các ràng buộc kích thước của bộ đệm hình ảnh giải mã (DPB) 116 sau đó có thể được xác định theo giới hạn số lượng khe hình mà không cần cân nhắc đến độ phân giải và độ sâu bit của mỗi hình ảnh giải mã.

Theo một cấu hình, làm đầy của bộ đệm hình ảnh giải mã (DPB) có thể được theo dõi theo mỗi lớp. Ví dụ, các ràng buộc kích thước của bộ đệm hình ảnh giải mã (DPB) 116 có thể được truyền tín hiệu và bước đệm có thể được áp dụng, cho mỗi lớp. Trong đó mỗi lớp bao gồm các bộ đệm lưu trữ hình ảnh, biến `DPBFullness[nuh_layer_id]` có thể được sử dụng để theo dõi làm đầy của bộ đệm hình ảnh giải mã (DPB) của mỗi lớp. Khi ảnh được gỡ bỏ khỏi lớp có giá trị ID của lớp bằng `nuh_layer_id`, thì biến `DPBFullness[nuh_layer_id]` có thể được thiết đặt là `DPBFullness[nuh_layer_id]-1` (nghĩa là `DPBFullness[nuh_layer_id]` có thể được giảm bớt một). Trong trường hợp này, thì ảnh đó đã được gỡ bỏ khỏi bộ đệm lưu trữ hình ảnh PSB[`nuh_layer_id`].

Tương tự, khi hình ảnh giải mã hiện thời với giá trị ID của lớp bằng `nuh_layer_id` được lưu trữ trong bộ đệm hình ảnh giải mã (DPB) 116, thì biến `DPBFullness[nuh_layer_id]` được thiết đặt bằng `DPBFullness[nuh_layer_id]+1` (nghĩa là `DPBFullness[nuh_layer_id]` được tăng thêm một). Trong trường hợp này, ảnh đó đã được lưu trữ vào bộ đệm lưu trữ hình ảnh PSB[`nuh_layer_id`].

Mức đầy của bộ đệm hình ảnh giải mã (DPB) cũng có thể được theo dõi đối với tập hợp lớp kết xuất. Các ràng buộc kích thước của bộ đệm hình ảnh giải mã (DPB) 116 sau đó có thể được truyền tín hiệu và bước đệm có thể được áp dụng, dựa trên các ràng buộc được xác định cho tập hợp lớp kết xuất. Giá trị `DPBFullness` có thể được theo dõi đối với tập hợp lớp kết xuất được kết hợp với điểm hoạt động trong thử nghiệm. Do vậy, khi ảnh được gỡ bỏ khỏi lớp thuộc tập hợp lớp kết xuất, thì giá trị làm đầy của bộ đệm hình ảnh giải mã (DPB) có thể được giảm bớt một vì $DPBFullness = DPBFullness - 1$. Tương tự, khi hình ảnh giải mã hiện thời được lưu trữ trong bộ đệm hình ảnh giải mã (DPB) 116, thì làm đầy của bộ đệm hình ảnh giải mã (DPB) có thể được giảm bớt một vì $DPBFullness = DPBFullness + 1$.

Trong quá trình 613 liên quan đến hình ảnh giải mã hiện thời, quy trình đánh dấu 611 được minh họa. Quy trình đánh dấu 611 này có thể bao gồm bước đánh dấu tham chiếu 627a-c và bước đánh dấu kết xuất 629. Sau khi dịch vị định thời từ các bước lưu trữ 625a-c, bước đánh dấu tham chiếu dựa trên hình ảnh 627a-c có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao

thứ hai (EL2) 2642b. Sau dịch vị định thời khác, bước đánh dấu kết xuất dựa trên đơn vị truy cập (AU) 629 có thể được thực hiện bởi bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b. Khi quá trình 613 liên quan đến hình ảnh giải mã hiện thời được hoàn thành, bước 631 của việc kết xuất hình ảnh dựa trên đơn vị truy cập (AU) (đệm bổ sung) có thể được thực hiện nhờ bộ đệm hình ảnh giải mã (DPB) 116 của lớp cơ sở (BL) 2644, bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh giải mã (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b.

Fig.69 là sơ đồ khối thể hiện kết cấu và định thời cho các đơn vị lớp trừu tượng hoá mạng (NAL) của các lớp đối với các hình ảnh mã hóa và các đơn vị truy cập (AU) khi lớp nâng cao thứ hai (EL2) 942b có thời gian ảnh thấp hơn lớp cơ sở (BL) 944 và lớp nâng cao thứ nhất (EL1) 942a. Các đơn vị NAL của hình ảnh mã hoá EL1 953a được minh họa cùng với lớp nâng cao thứ nhất (EL1) 942a. Các đơn vị NAL của hình ảnh mã hoá EL2 953b được minh họa cùng với lớp nâng cao thứ hai (EL2) 942b. Các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 953c được minh họa cùng với lớp cơ sở (BL) 944.

Tại thời điểm t1, các đơn vị NAL của hình ảnh mã hoá EL1 953a, các đơn vị NAL của hình ảnh mã hoá EL2 953b và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 953c là tiểu mục của đơn vị truy cập (AU) 955a. Tại thời điểm t2, các đơn vị NAL của hình ảnh mã hoá EL1 953a và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 953c là tiểu mục của đơn vị truy cập (AU) 955b. Tại thời điểm t3, các đơn vị NAL của hình ảnh mã hoá EL1 953a, các đơn vị NAL của hình ảnh mã hoá EL2 953b và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 953c là tiểu mục của đơn vị truy cập (AU) 955c. Tại thời điểm t4, các đơn vị NAL của hình ảnh mã hoá EL1 953a và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 953c là tiểu mục của đơn vị truy cập (AU) 955d.

Fig.70 là sơ đồ khối thể hiện kết cấu và định thời cho các đơn vị lớp trừu tượng hoá mạng (NAL) của các lớp đối với các hình ảnh mã hóa và các đơn vị truy cập (AU) khi lớp cơ sở (BL) 1044 có thời gian ảnh thấp hơn lớp nâng cao thứ nhất (EL1) 1042a và lớp nâng cao thứ hai (EL2) 1042b. Các đơn vị NAL của hình ảnh mã hoá EL1 1053a được minh họa cùng với lớp nâng cao thứ nhất (EL1) 1042a. Các đơn vị NAL của hình ảnh mã hoá EL2 1053b được minh họa cùng với lớp nâng cao thứ hai (EL2) 1042b. Các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 1053c được minh họa cùng với lớp cơ sở (BL) 1044.

Tại thời điểm t1, các đơn vị NAL của hình ảnh mã hoá EL1 1053a, các đơn vị NAL của hình ảnh mã hoá EL2 1053b và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 1053c là tiểu mục của đơn vị truy cập (AU) 1055a. Tại thời điểm t2, các đơn vị NAL của hình ảnh mã hoá EL1 1053a và các đơn vị NAL của hình ảnh mã hoá EL2 1053b là tiểu mục của đơn vị truy cập (AU) 1055b. Tại thời điểm t3, các đơn vị NAL của hình ảnh mã hoá EL1 1053a, các đơn vị NAL của hình ảnh mã hoá EL2 1053b và các đơn vị NAL của hình ảnh mã hoá của lớp cơ sở 1053c là tiểu mục của đơn vị truy cập (AU) 1055c. Tại thời điểm t4, các đơn vị NAL của hình ảnh mã hoá EL1 1053a và các đơn vị NAL của hình ảnh mã hoá EL1 1053b là tiểu mục của đơn vị truy cập (AU) 1055d.

Thuật ngữ “vật ghi đọc được bằng máy tính” là vật ghi bất kỳ có thể sử dụng được có thể được truy nhập bởi máy tính hoặc bộ xử lý. Thuật ngữ “vật ghi đọc được bằng máy tính” trong bản mô tả này, có thể biểu thị vật ghi đọc được bằng máy tính và/hoặc vật ghi đọc được bằng bộ xử lý mà chúng có tính chất không chuyển tiếp và hữu hình. Bằng cách ví dụ và không ràng buộc, vật ghi đọc được bằng máy tính hoặc vật ghi đọc được bằng bộ xử lý có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc các thiết bị lưu trữ đĩa quang, các thiết bị lưu trữ đĩa từ hoặc các thiết bị lưu trữ từ khác, hoặc vật ghi bất kỳ khác mà chúng có thể được sử dụng để vận chuyển hoặc lưu trữ mã chương trình mong muốn lưu trữ mã chương trình mong muốn lưu trữ dưới dạng các lệnh hoặc các kết cấu dữ liệu và chúng có thể được truy nhập bởi máy tính hoặc bộ xử lý. Các đĩa, trong bản mô tả này, bao gồm đĩa compact (CD - compact disc), đĩa laze, đĩa quang, đĩa đa năng số (DVD - digital versatile disc), đĩa mềm và đĩa blu-ray (nhãn hiệu đã đăng ký), trong đó đĩa mềm thường sao chép lại dữ liệu bằng phương pháp từ, trong khi các đĩa kia sao chép dữ liệu theo phương pháp quang với tia laze.

Cần lưu ý rằng có thể thực hiện một hoặc nhiều phương pháp được mô tả trong bản mô tả này trong và/hoặc được thực hiện bằng cách sử dụng phần cứng. Ví dụ, một hoặc nhiều phương pháp hoặc cách tiếp cận được mô tả trong bản mô tả này có thể được thực hiện trong và/hoặc được thực hiện bằng cách sử dụng bộ chip, ASIC, LSI hoặc mạch tích hợp v.v..

Mỗi phương pháp đó được bộc lộ trong sáng chế bao gồm một hoặc nhiều bước hoặc các hoạt động để đạt được phương pháp đã mô tả. Các bước và/hoặc các hoạt động của phương pháp có thể được hoán đổi với nhau và/hoặc được kết hợp thành một bước mà không nằm ngoài phạm vi bảo hộ của sáng chế. Nói cách khác, trừ phi thứ tự cụ thể của các

bước hoặc các hoạt động là cần thiết đối với sự vận hành chính xác hoạt động của phương pháp này đang được mô tả, thứ tự và/hoặc việc sử dụng các bước và/hoặc các hoạt động cụ thể đó có thể được biến đổi mà không nằm ngoài phạm vi bảo hộ của sáng chế.

Cần hiểu rằng các điểm yêu cầu bảo hộ không bị giới hạn ở cấu hình và các bộ phận được minh họa ở phần trên. Có thể thực hiện một số cải biến, thay đổi và biến đổi theo cách bố trí, hoạt động và các chi tiết của các hệ thống, phương pháp và thiết bị đã được mô tả trong bản mô tả này mà không nằm ngoài phạm vi bảo hộ của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dòng bit video, phương pháp này bao gồm các bước:

- (a) thu dòng bit cơ sở đại diện cho chuỗi video mã hóa;
- (b) thu các dòng bit nâng cao đại diện cho chuỗi video mã hóa này;
- (c) thu tập tham số video chứa các phần tử cú pháp để áp dụng cho dòng bit cơ sở và các dòng bit nâng cao, trong đó tập tham số video này chứa phần tử cú pháp truyền tín hiệu phần mở rộng của tập tham số video;
- (d) thu phần mở rộng của tập tham số video chứa các phần tử cú pháp liên quan đến ít nhất một trong các dòng bit nâng cao; trong đó các phần tử cú pháp này bao gồm phần tử cú pháp thứ nhất chỉ báo nhiều tập hợp lớp kết xuất được xác định cho dòng bit video;
- (e) thu các phần tử cú pháp chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất, khác biệt ở chỗ:

các phần tử cú pháp này chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất bao gồm phần tử cú pháp thứ hai chỉ báo nhiều tập hợp lớp bổ sung mà cho chúng các lớp kết xuất được xác định bên cạnh nhiều tập hợp lớp được xác định cho dòng bit video được chỉ báo bởi phần tử cú pháp thứ nhất mà các lớp kết xuất được xác định cho chúng.

2. Phương pháp theo điểm 1, trong đó các phần tử cú pháp chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất nằm trong phần mở rộng của tập tham số video.

3. Phương pháp theo điểm 1, trong đó các phần tử cú pháp chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất nằm trong thông báo thay đổi của tập hợp lớp kết xuất.

4. Phương pháp theo điểm 3, trong đó thông báo thay đổi của tập hợp lớp kết xuất này là thông báo thông tin nâng cao bổ sung (SEI).

5. Phương pháp theo điểm 3, trong đó thông báo thay đổi của tập hợp lớp kết xuất đã nêu bao gồm cú pháp gỡ bỏ một trong số ít nhất một tập hợp lớp kết xuất.

6. Phương pháp theo điểm 3, trong đó thông báo thay đổi của tập hợp lớp kết xuất bao gồm cú pháp để bổ sung tập hợp lớp kết xuất vào ít nhất một tập hợp lớp kết xuất đã nêu.

7. Phương pháp theo điểm 3, trong đó thông báo thay đổi của tập hợp lớp kết xuất bao gồm cú pháp để thay đổi tập hợp lớp kết xuất trong số ít nhất một tập hợp lớp kết xuất đã nêu.

8. Phương pháp theo điểm 5, trong đó thông báo thay đổi của tập hợp lớp kết xuất đã nêu bao gồm cú pháp để bổ sung tập hợp lớp kết xuất vào ít nhất một tập hợp lớp kết xuất đã nêu.
9. Phương pháp theo điểm 8, trong đó thông báo thay đổi của tập hợp lớp kết xuất đã nêu bao gồm cú pháp để thay đổi tập hợp lớp kết xuất trong số ít nhất một tập hợp lớp kết xuất đã nêu.
10. Phương pháp theo điểm 6, trong đó thông báo thay đổi của tập hợp lớp kết xuất đã nêu bao gồm cú pháp để thay đổi tập hợp lớp kết xuất trong số ít nhất một tập hợp lớp kết xuất đã nêu.
11. Phương pháp theo điểm 7, trong đó thông báo thay đổi của tập hợp lớp kết xuất đã nêu bao gồm cú pháp để gỡ bỏ một trong số ít nhất một tập hợp lớp kết xuất đã nêu.
12. Phương pháp theo điểm 1, trong đó các phần tử cú pháp để chỉ báo thay đổi về ít nhất một tập hợp lớp kết xuất bao gồm phần tử cú pháp thứ ba để xác định các tập hợp lớp bổ sung và cờ để chỉ báo liệu lớp cụ thể có phải là lớp kết xuất đích hay không.

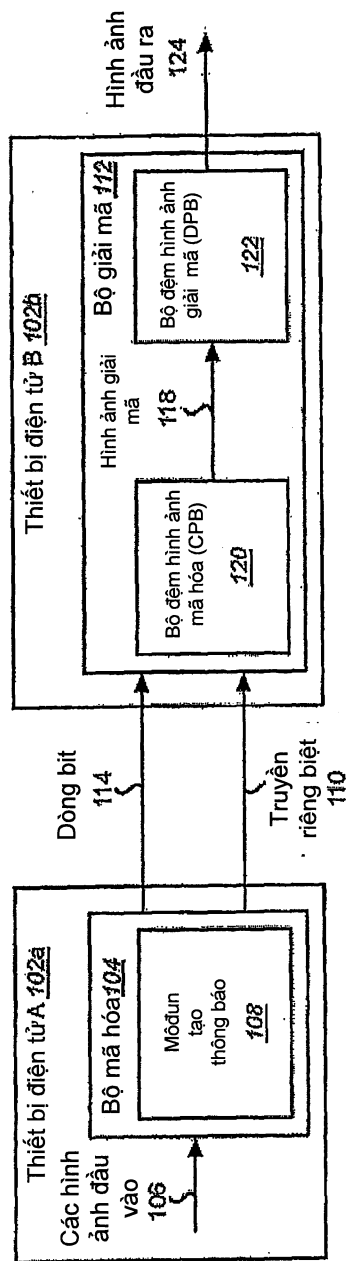


FIG. 1A

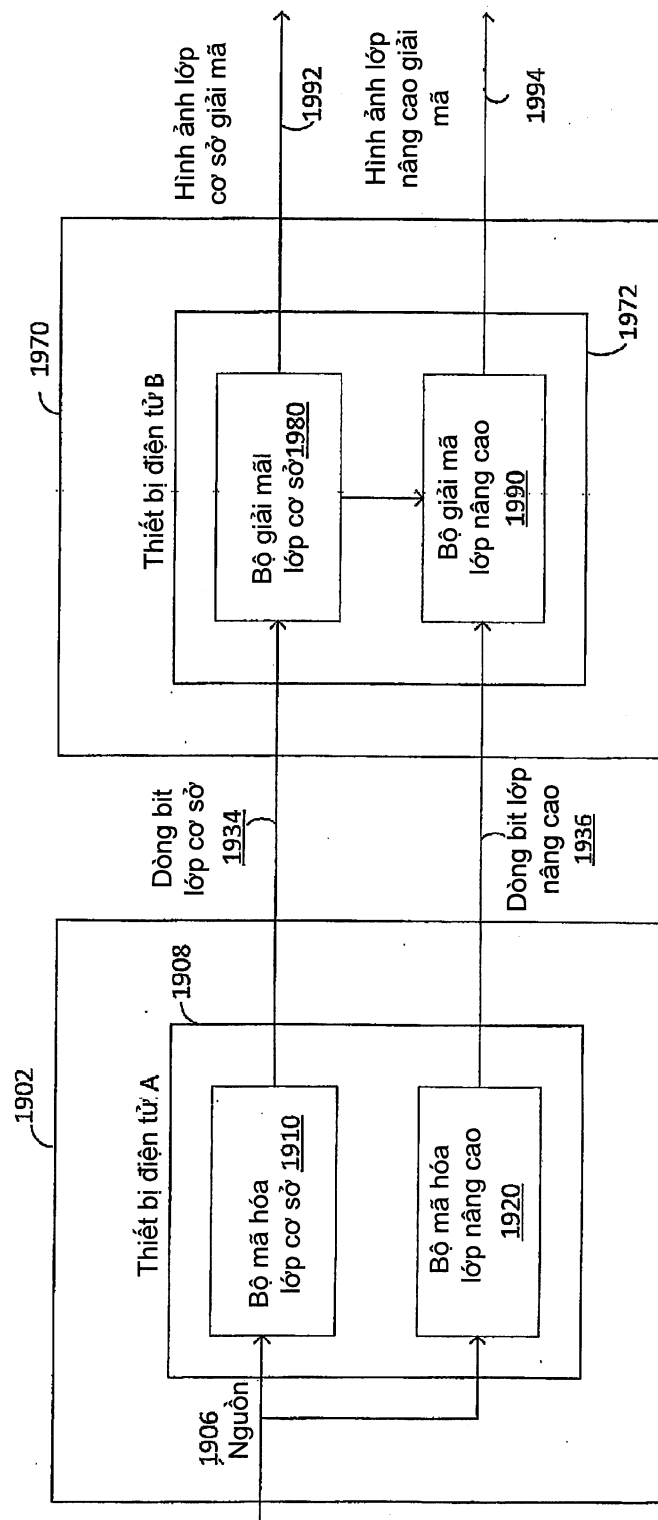


FIG. 1B

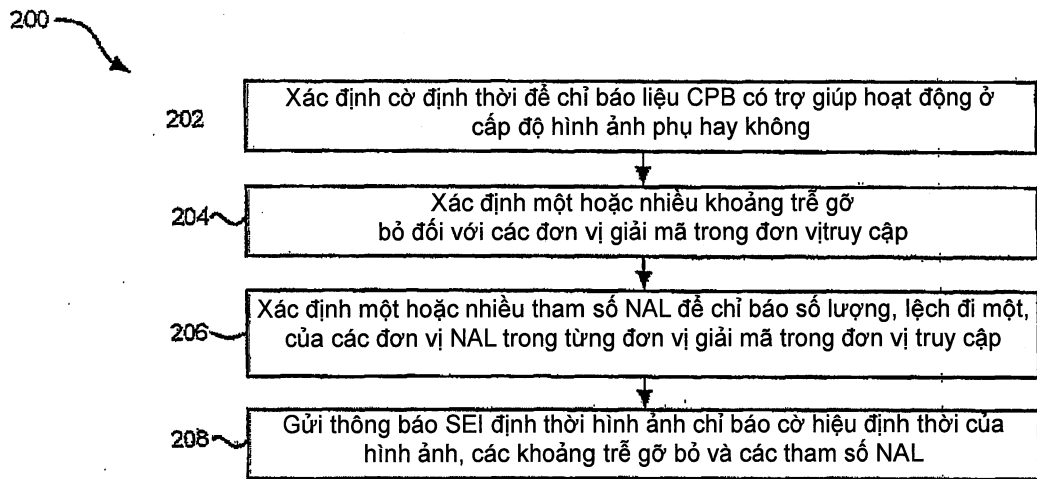


FIG. 2

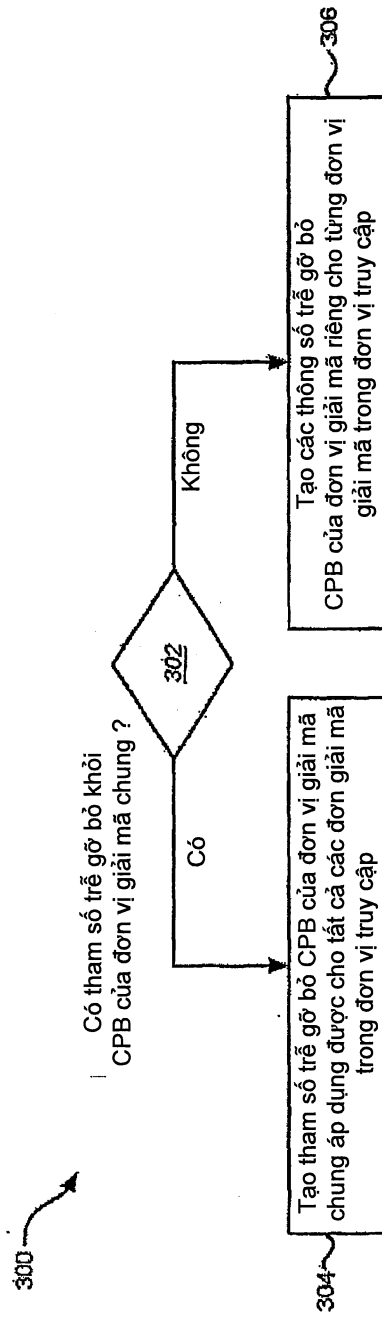


FIG. 3

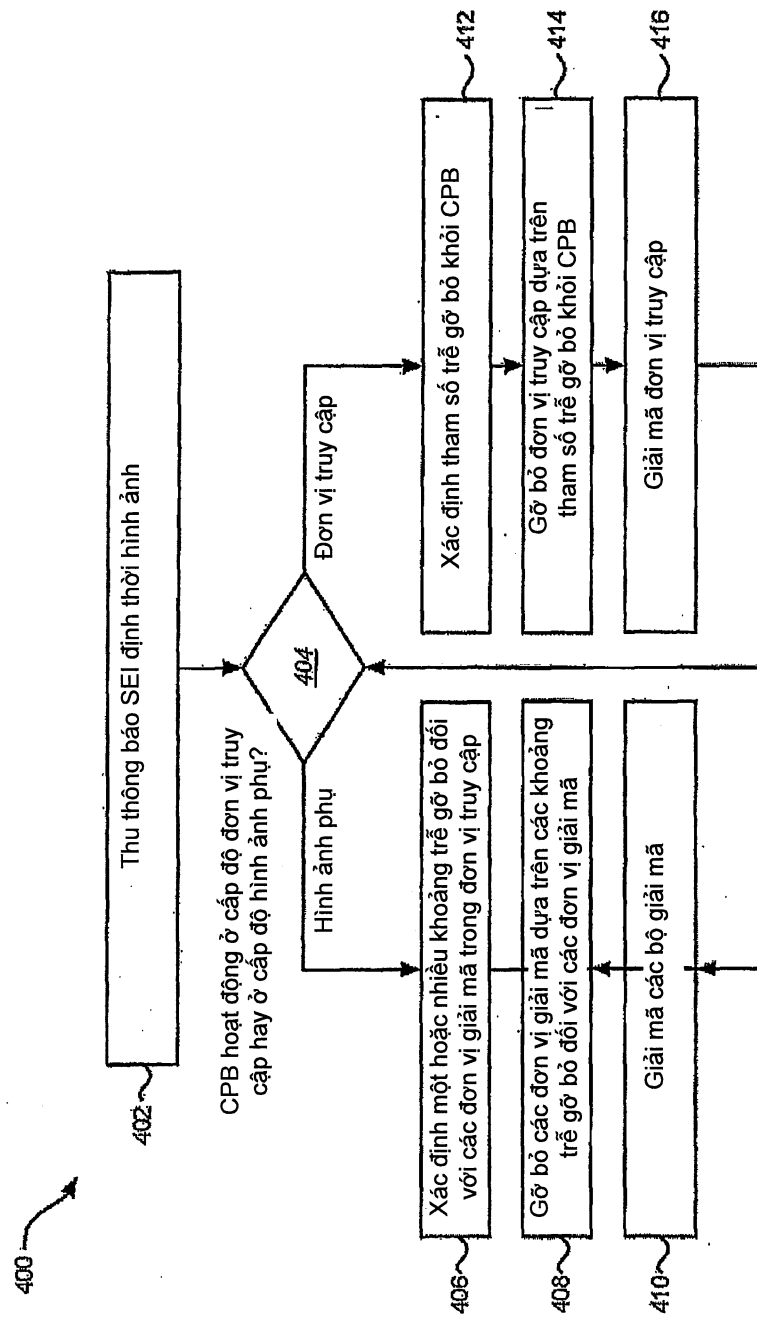


FIG. 4

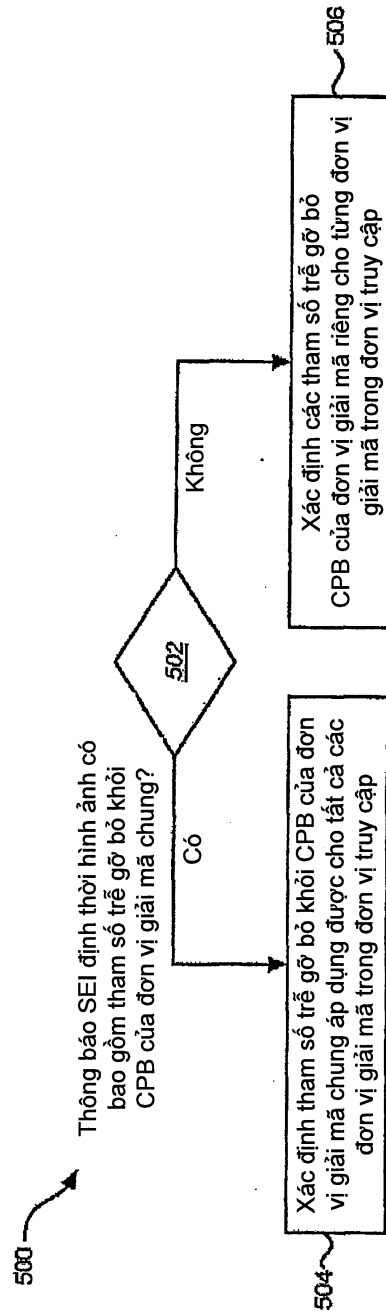


FIG. 5

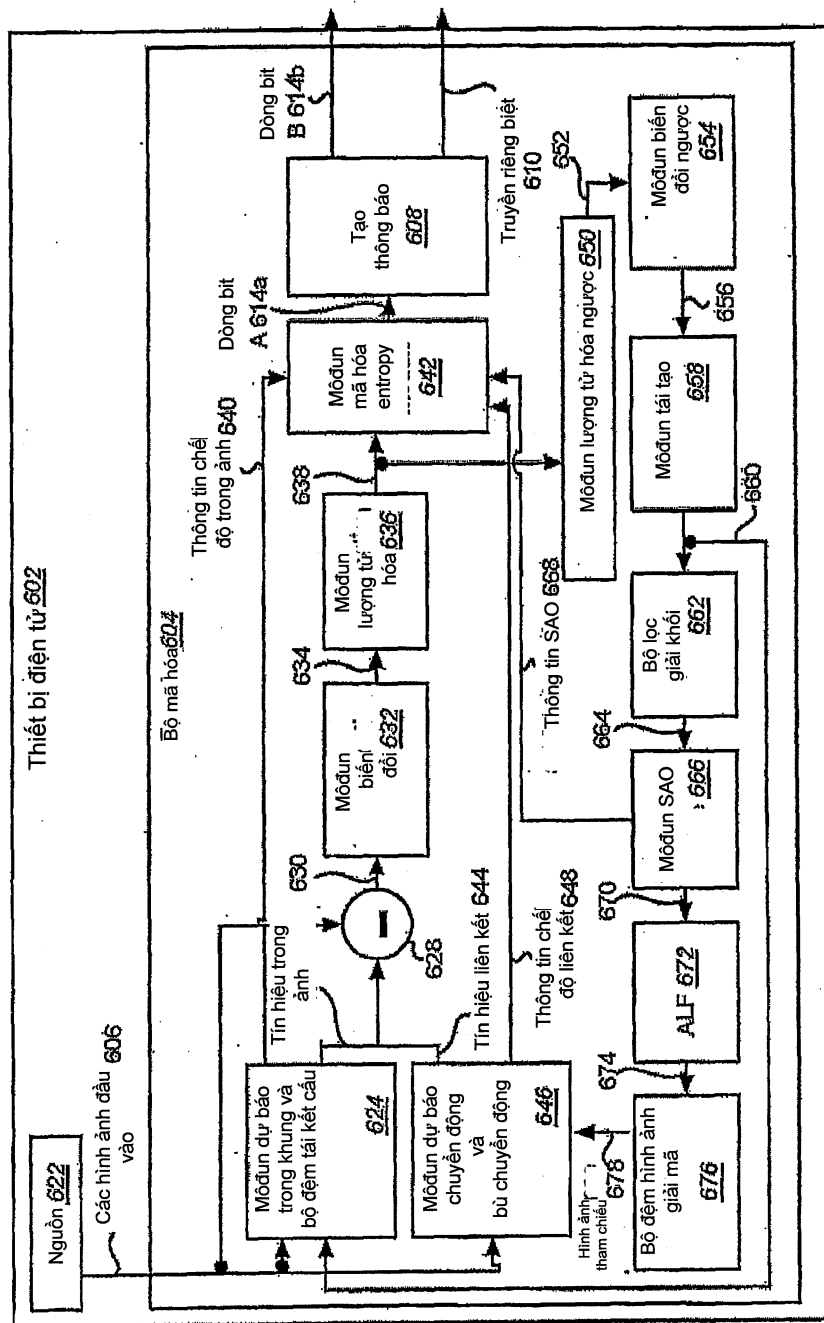
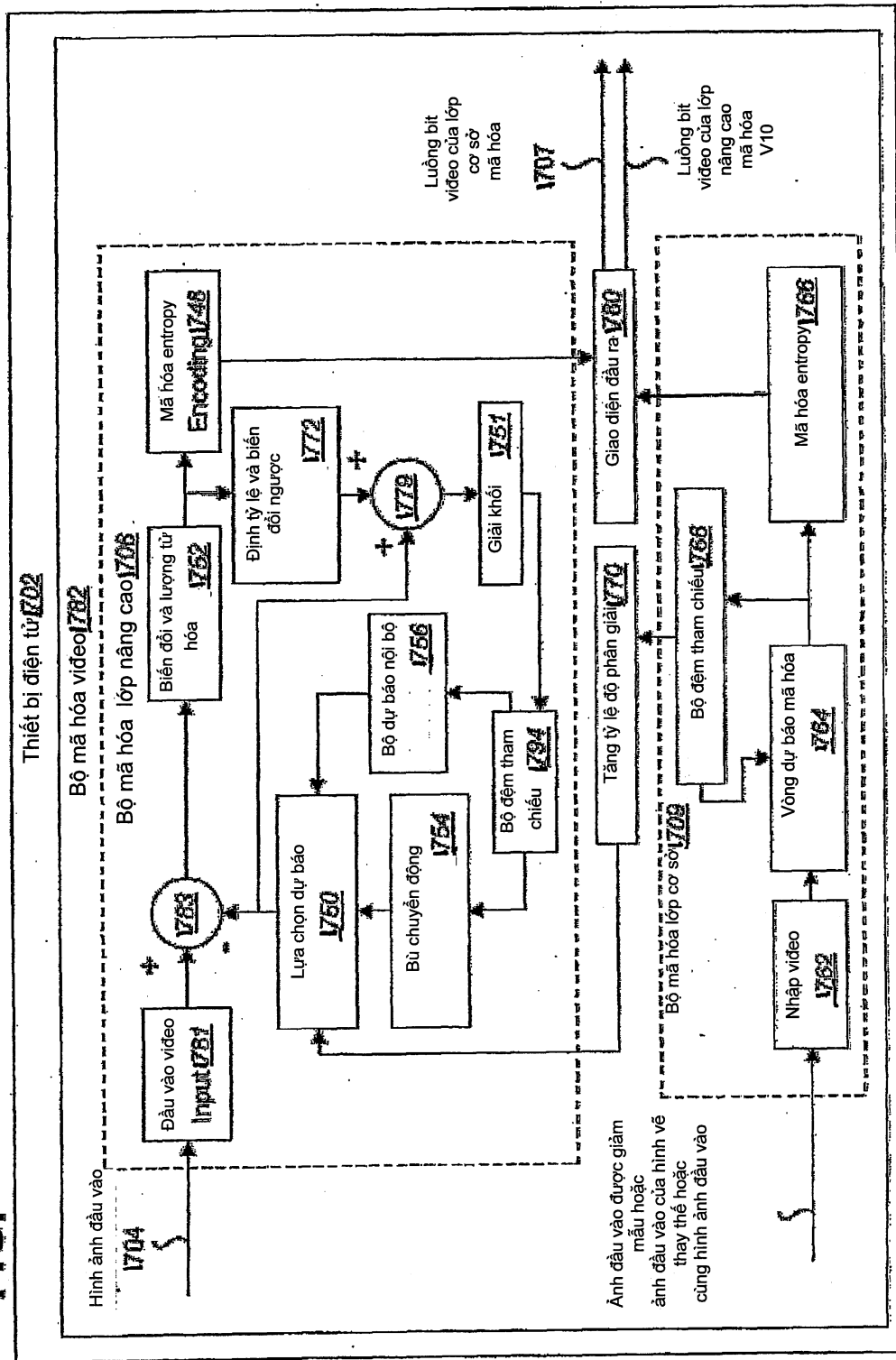


FIG. 6A

FIG. 6B



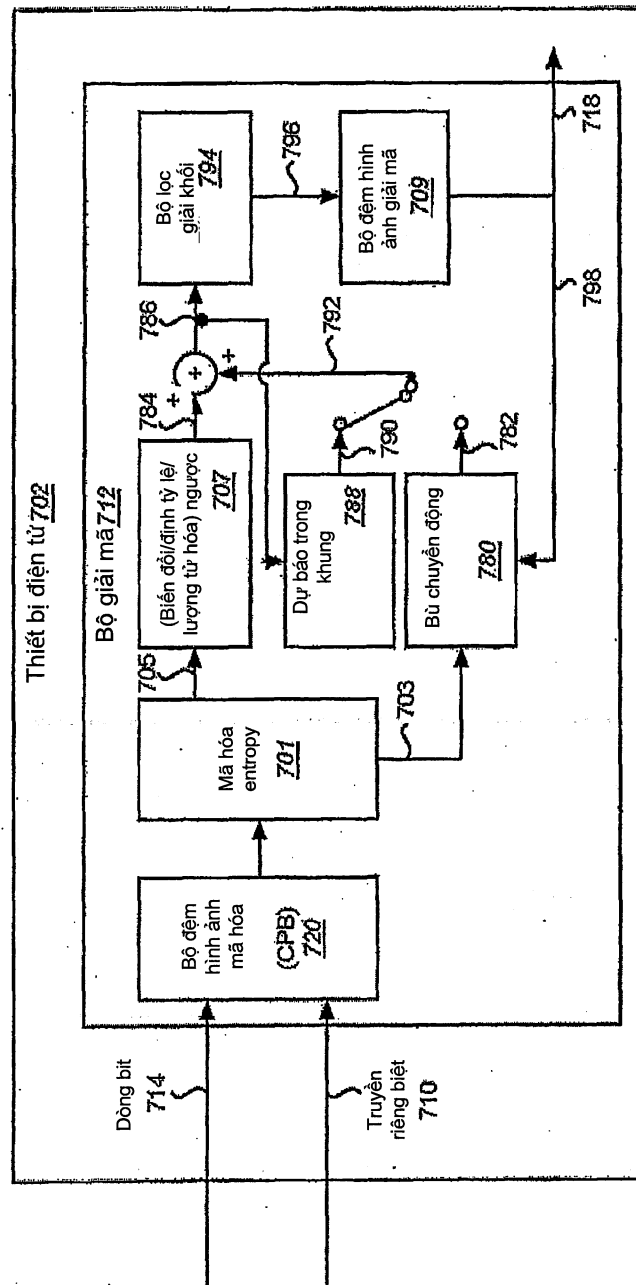
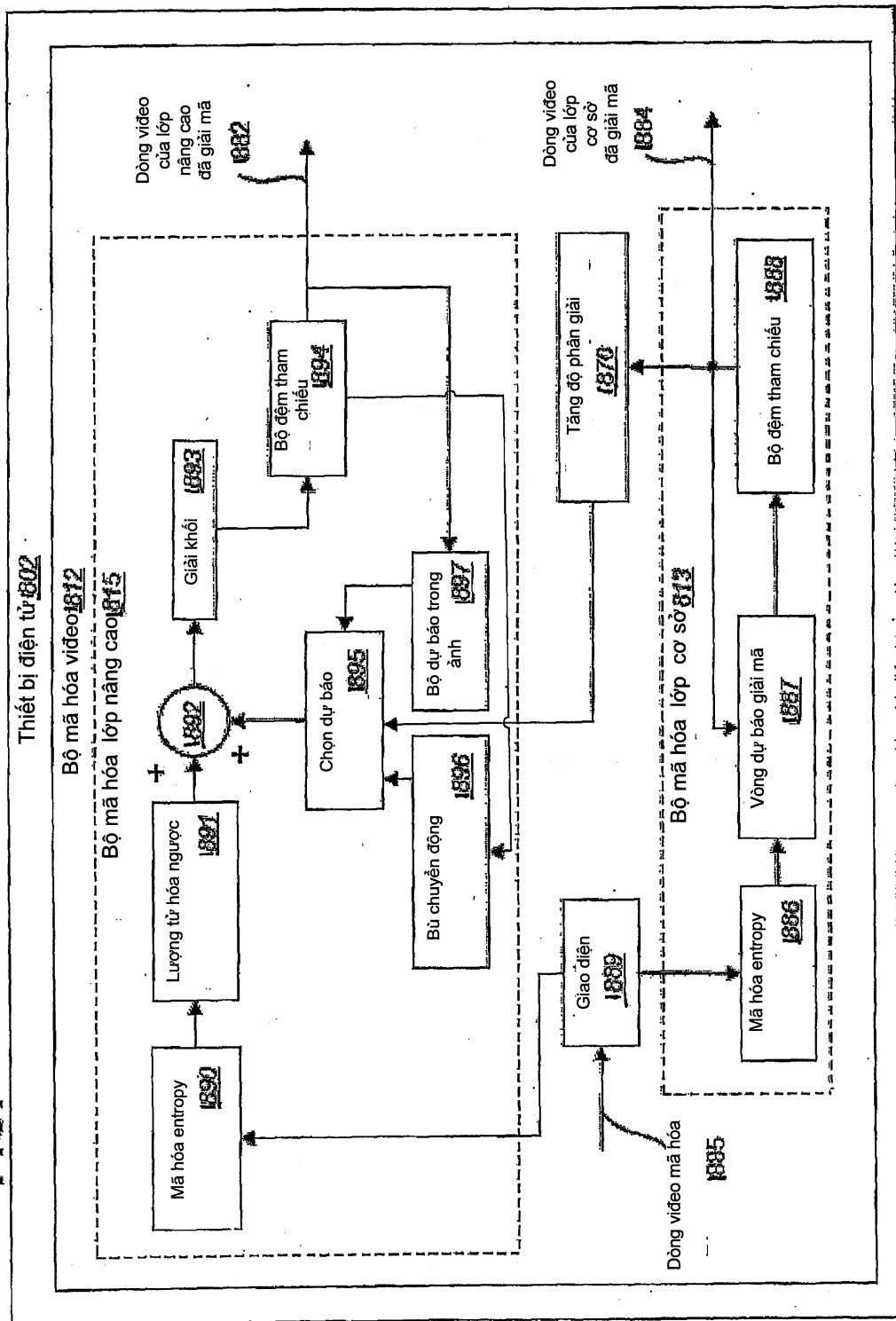


FIG. 7A

Fig. 7B



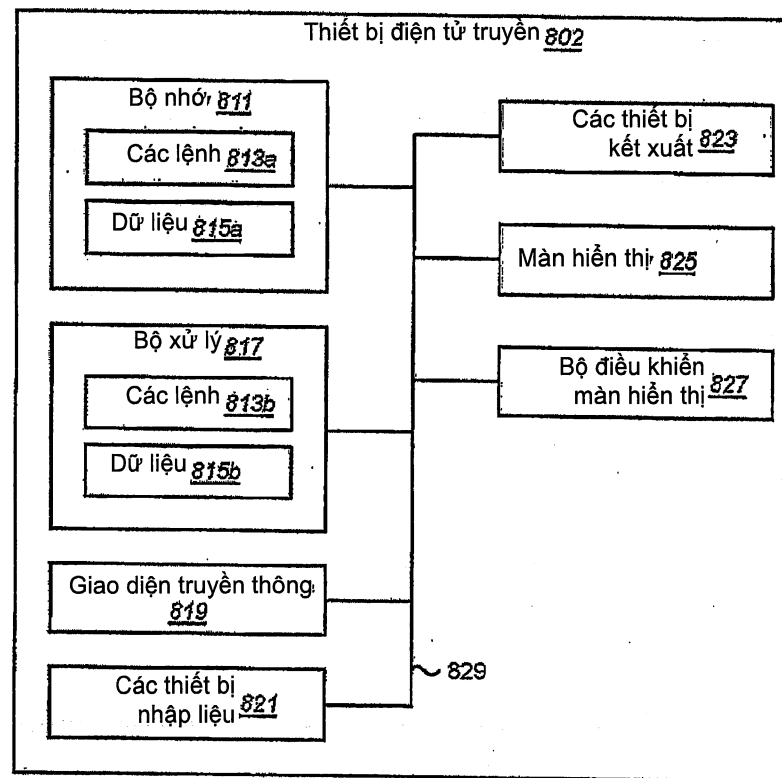


FIG. 8

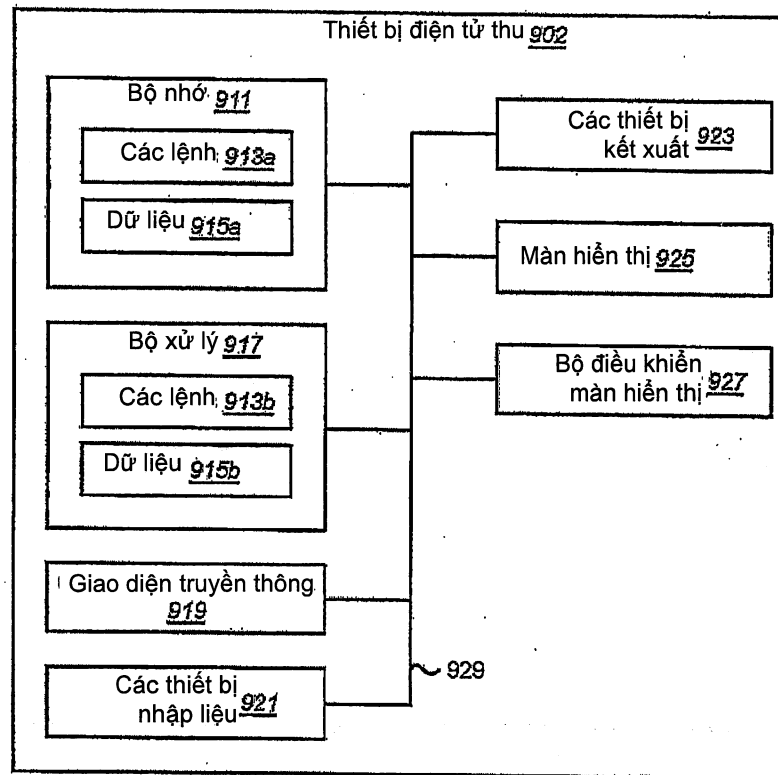


FIG. 9

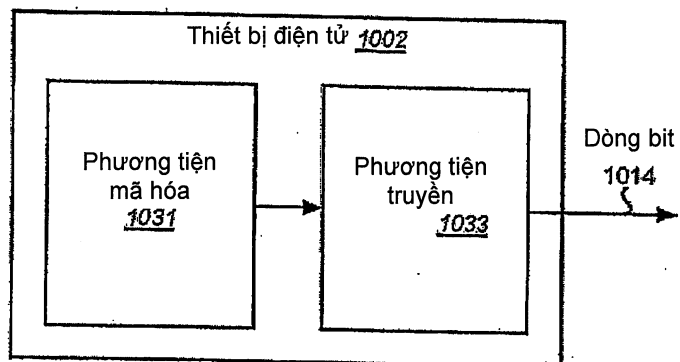


FIG. 10

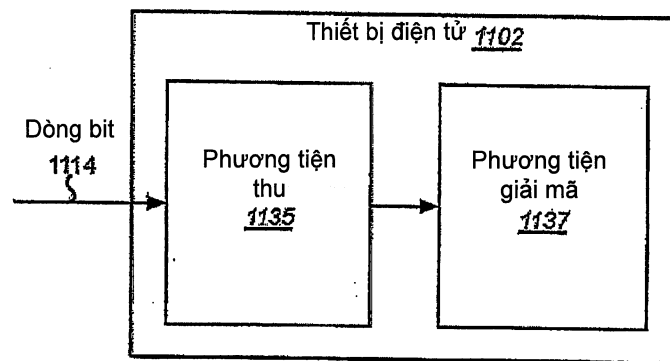


FIG. 11

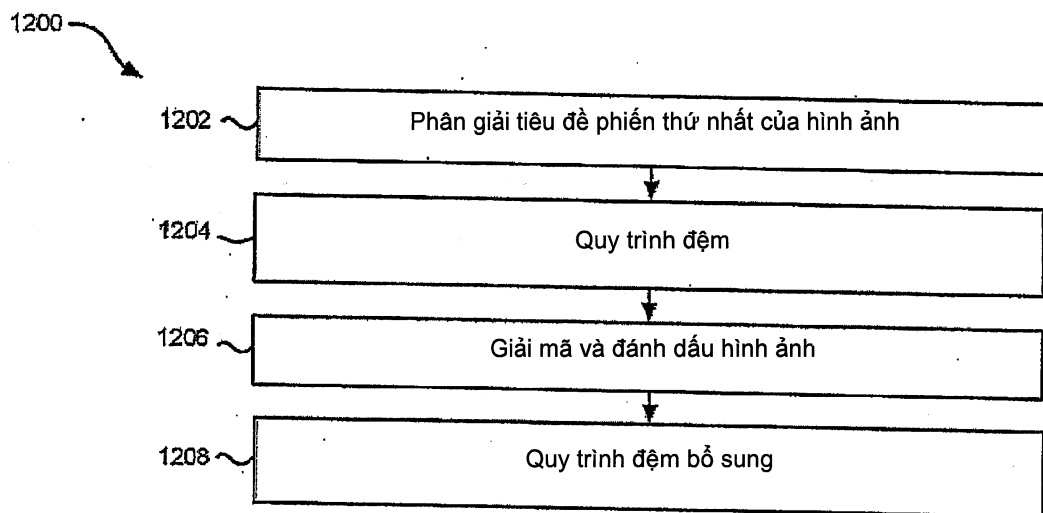


FIG. 12

<code>nal_unit_header()</code> {	Tên
<code>forbidden_zero_bit</code>	<code>f(1)</code>
<code>nal_unit_type</code>	<code>u(6)</code>
<code>nuh_reserved_zero_6bits</code>	<code>u(6)</code>
<code>nuh_temporal_id_plus1</code>	<code>u(3)</code>
}	

Cú pháp của tiêu đề đơn vị NAL

FIG. 13A

nal_unit_header() {	Tên
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
layer_id_plus1	u(6)
nuh_temporal_id_plus1	u(3)
}	

Cú pháp của tiêu đề đơn vị NAL

FIG. 13B

nal_unit_header() {	Tên
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

Cú pháp của tiêu đề đơn vị NAL

FIG. 13C

nal_unit(NumBytesInNALunit) {	Tên
nal_unit_header()	
NumBytesInRBSP = 0	
for(i = 2; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++]	b(8)
rbsp_byte[NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++]	b(8)
}	
}	

Cú pháp của đơn vị NAL chung

FIG. 14

vps_extension() {	Tên
while(!byte_aligned())	
vps_extausion_byte_alignment_reserved_zero_bit	u(1)
// scalability type and layer_id partitioning method	
scalability_type	u(4)
for(i=0; i < MaxDlm(scalability_type); i++)	
layer_id_dlm_len[i]	u(3)
// layer specific information	
for(i=0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j=0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Cú pháp phần mở rộng của tập tham số video hiện tại

FIG. 15

scalability_type	MaxDlm(scalability_type)	Kích thước khả năng điều chỉnh
0	1	không (HEVC cơ sở)
1	2	không gian và chất lượng
2	3	không gian, chất lượng, không xác định
3	4	không gian, chất lượng, không xác định, không xác định
4	2	đa hình ảnh và độ sâu
5	3	đa hình ảnh, độ sâu, không xác định
6	4	đa hình ảnh, độ sâu, không xác định, không xác định
7	4	đa hình ảnh, không gian, chất lượng và độ sâu
8	5	đa hình ảnh, không gian, chất lượng, độ sâu, không xác định
9	6	đa hình ảnh, không gian, chất lượng, độ sâu, không xác định, không xác định
10..15	reserved	dự phòng

Các loại khả năng mở rộng hiện tại

FIG. 16

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// Scalability map and layer_id_plus1 mapping method	
for (i=0; i<= max_num_layers_minus1_bits; i++) {	
scalability_map[i]	u(3)
}	
// layer specific information	
for(l = 0; l <= max_num_layers_minus1; l++) {	
vps_layer_id[l]	u(6)
// layer dependency	
num_direct_ref_layers[l]	u(6)
for(j = 0; j < num_direct_ref_layers[l]; j++)	
ref_layer_id[l][j]	u(6)
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.17

Scalability_map[i]	Kích thước khả năng mở rộng
0	không (HEVC cơ sở)
1	Không gian
2	Chất lượng
3	Độ sâu
4	Đa hình ảnh
5	Không xác định
6, 7	Dự phòng

Cú pháp ánh xạ khả năng mở rộng

FIG.18

vps_extension() {	Tên
while(lbyte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// scalability map and layer_id_plus1 mapping method	
num_scalability_dimensions_minus1	u(3)
for(i=0; i <= num_scalability_dimensions_minus1; i++) {	
scalability_map[i]	u(3)
num_bits_for_scalability_map[i]	u(3)
}	
// layer specific information	
for(i=0; i <= max_num_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j=0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.19

vps_extension() {	Tên
while(lbyte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// layer specific information	
for(i=1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
num_dimensions_minus1[i]	u(4)
for(j=0; j <= num_dimensions_minus1; j++) {	
dimension_type[i][j]	u(4)
dimension_id[i][j]	u(8)
}	
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j=0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Cú pháp phần mở rộng của tập tham số video hiện tại

FIG.20

dimension_type[i][j]	dimension_id[i][j]
0	view order idx
1	depth flag
2	dependency ID
3	quality ID
4..15	reserved

Loại kích thước hiện tại, cú pháp ID kích thước

FIG.21

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// layer specific information	
for(i = 1; i <= max_num_layers_minus1; i++) {	
// mask signalling scalability types that are present for this layer ID	
scalability_mask	u(8)
for(j = 0; j <= num_scalability_types[j]; j++) {	
scalability_id[j]	u(8)
}	
// layer dependency	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Cú pháp phân mở rộng của tập tham số video

FIG. 22

scalability_mask_bit[i]	Kích thước khả năng điều chỉnh
0	Không gian
1	Chất lượng
2	Độ sâu
3	Đa hình ảnh
4-15	Dự phòng

Cú pháp ánh xạ khả năng mở rộng

FIG.23

video_parameter_set_rbsp() {	Tên
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
reserved_zero_2bits	u(2)
max_num_layers_minus1 // reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_level(1, vps_max_sub_layers_minus1)	
next_essential_info_byte_offset // reserved_zero_12bits	u(12)
for(i=0; i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
num_hrd_parameters	ue(v)
for(i=0; i < num_hrd_parameters; i++) {	
if(i > 0)	
op_point(i)	
hrd_parameters(i == 0, vps_max_sub_layers_minus1)	
}	
bit_equal_to_one	u(1)
scalability_mask	u(8)
for(i=0; i <= max_num_layers_minus1; i++) {	
vps_extension_data()	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.24

vps_extension_data() {	Tên
vps_extension_type	u(2)
layer_id	u(6)
layer_max_sub_layers_minus1	u(3)
new_profile_level_flag	u(1)
if (new_profile_level_flag == 0) {	
profile_reference_layer_id	u(6)
}	
else {	
profile_level(1, layer_max_sub_layers_minus1)	
}	
for (j = 0; j <= num_scalability_types; j++) {	
scalable_id[j]	u(8)
}	
num_reference_layers	u(6)
for (n = 0; n < num_reference_layers; n++)	
direct_coding_dependency_layer_id_plus1[n]	u(6)
}	

Cú pháp phần mở rộng của tập tham số video

FIG.25

<code>vps_extension()</code> {	Tên
<code>while(!byte_aligned())</code>	
<code>vps_extension_byte_alignment_reserved_one_bit</code>	u(1)
<code>avc_base_codec_flag</code>	u(1)
<code>scalability_mask</code>	u(16)
<code>for(i = 0; i < NumScalabilityTypes; i++) {</code>	
<code> dimension_id_layer_minus1[i]</code>	u(3)
<code>}</code>	
<code>vps_nuh_layer_id_present_flag</code>	u(1)
<code>// layer specific information</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++) {</code>	
<code> // mapping of layer ID to scalability dimension IDs</code>	
<code> if(vps_nuh_layer_id_present_flag)</code>	
<code> layer_id_in_nuh[i]</code>	u(6)
<code> for(j = 0; j < NumScalabilityTypes; j++)</code>	
<code> dimension_id[i][j]</code>	u(16)
<code> }</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++)</code>	
<code> profile_for_layer(1, vps_max_sub_layers_minus1)</code>	
<code>for(i = 1; i <= vps_max_layers_minus1; i++) {</code>	
<code> // layer dependancy</code>	
<code> num_direct_ref_layers[i]</code>	u(4)
<code> for(j = 0; j < num_direct_ref_layers[i]; j++)</code>	
<code> ref_layer_id[i][j]</code>	u(6)
<code> }</code>	
<code>}</code>	

Cú pháp phân mở rộng tập tham số video

FIG.26

scalability_mask's K' th bit	Kích thước khả năng mở rộng
0	Không gian
1	Chất lượng
2	Độ sâu
3	Đa hình ảnh
4-15	Dự trữ

Cú pháp Scalability_mask

FIG.27

vps_extension() {	Tên
while(1byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1 ; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i = 1; i <= NumDepLayers; i++) {	
// layer dependency information signaling	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.28

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_allgment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(i, vps_max_sub_layers_minus1)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// layer dependency information signalling	
layer_dependency_flag[i]	u(1)
if(layer_dependency_flag[i]) {	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.29

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(i, vps_max_sub_layers_minus1)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// layer dependency information signaling	
layer_dependency_map[i]	u(v)
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.30

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1 ; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i = 1; i <= NumDepLayers; i++)	
layer_dependency_map[i]	u(v)
}	

Cú pháp phân mở rộng của tập tham số video

FIG.31

vps_extension() {	Tên
while(1byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i = 0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
profile_tier_level(i, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(layer_dependency_information_pattern(i)) {	
// layer dependency information signaling	
num_direct_ref_layers[i]	u(6)
for(j = 0; j < num_direct_ref_layers[i]; j++)	
ref_layer_id[i][j]	u(6)
}	
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.32

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_codec_flag	u(1)
scalability_mask	u(16)
for(i=0; i < NumScalabilityTypes; i++) {	
dimension_id_len_minus1[i]	u(3)
}	
vps_nuh_layer_id_present_flag	u(1)
// layer specific information	
for(i=1; i <= vps_max_layers_minus1; i++) {	
// mapping of layer ID to scalability dimension IDs	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j=0; j <= num_dimensions_minus1; j++)	
dimension_id[i][j]	u(v)
}	
for(i=1; i <= vps_max_layers_minus1 ; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
layer_dependency_information_pattern	u(v)
for(i=1; i <= vps_max_layers_minus1; i++) {	
if(layer_dependency_information_pattern(i)) {	
layer_dependency_map[i]	u(v)
}	
}	

Cú pháp phần mở rộng của tập tham số video

FIG.33

video_parameter_set_rbsp() {	Tên
vps_video_parameter_set_id	u(4)
vps_reserved_three_2bits	u(2)
vps_max_layers_minus1	u(6)
vps_max_sub_layers_minus1	u(3)
vps_temporal_id_nesting_flag	u(1)
vps_extension_offset // vps_reserved 0xffff 16bits	u(16)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_sub_layer_ordering_info_present_flag	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase_plus1[i]	ue(v)
}	
vps_max_layer_id	u(6)
vps_num_layer_sets_minus1	ue(v)
for(i = 1; i <= vps_num_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layer_id; j++)	
layer_id_included_flag[i][j]	u(1)
vps_timing_info_present_flag	u(1)
if(vps_timing_info_present_flag) {	
vps_num_units_in_tick	u(32)
vps_time_scale	u(32)
vps_poc_proportional_to_timing_flag	u(1)
if(vps_poc_proportional_to_timing_flag)	
vps_num_ticks_poc_diff_one_minus1	ue(v)
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
hrd_layer_set_idx[i]	ue(v)
if(i > 0)	
cprms_present_flag[i]	u(1)
hrd_parameters(cprms_present_flag[i], vps_max_sub_layers_minus1)	
}	
}	
vps_extension_flag	u(1)
if(vps_extension_flag) {	
vps_extension()	
vps_extension2_flag	u(1)
if(vps_extension2_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

FIG. 34

vps_extension() {	Ten
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nuh[i]	u(6)
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(lsIdx = 1; lsIdx <= vps_num_layer_sets_minus1; lsIdx ++) {	
vps_profile_present_flag[lsIdx]	u(1)
if(!vps_profile_present_flag[lsIdx])	
profile_layer_set_ref_minus1[lsIdx]	ue(v)
profile_tier_level(vps_profile_present_flag[lsIdx], vps_max_sub_layers_minus1)	
}	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 35

output_layer_sets_change(payloadSize) {	Tên
active_vps_id	u(4)
num_changed_output_layer_sets	ue(v)
for(i = 0; i < num_changed_output_layer_sets; i++) {	
changed_output_layer_set_idx_entry[i]	ue(v)
clsIdx = output_layer_set_idx[changed_output_layer_set_idx_entry[i]]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[clsIdx][j])	
output_layer_flag[clsIdx][j]	u(1)
}	
num_addl_output_layer_sets	
for(i = num_output_layer_sets; i < num_output_layer_sets + num_addl_output_layer_sets; i++) {	
addlIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIdx][j])	
output_layer_flag[addlIdx][j]	u(1)
}	
}	

FIG. 36

output_layer_sets_change(payloadSize) {	Tên
active_vps_id	u(4)
num_deleted_output_layer_sets	ue(v)
for(i = 0; i < num_deleted_output_layer_sets; i++)	
deleted_output_layer_sets_idx_entry[i]	ue(v)
num_changed_output_layer_sets	ue(v)
for(i = 0; i < num_changed_output_layer_sets; i++) {	
changed_output_layer_set_idx_entry[i]	ue(v)
clsIdx = output_layer_set_idx[changed_output_layer_set_idx_entry[i]]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[clsIdx][j])	
output_layer_flag[clsIdx][j]	u(1)
}	
num_addl_output_layer_sets	
for(i = num_output_layer_sets - num_deleted_output_layer_sets; i < num_output_layer_sets -	
num_deleted_output_layer_sets + num_addl_output_layer_sets; i++) {	
addlIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIdx][j])	
output_layer_flag[addlIdx][j]	u(1)
}	
}	

FIG. 37

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
ave_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_op_dpb_info_parameters	ue(v)
for(i = 0; i < num_op_dpb_info_parameters; i++) {	
operation_point_layer_set_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
op_dpb_info_parameters(j)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 38A

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_op_dpb_info_parameters	ue(v)
for(i = 0; i < num_op_dpb_info_parameters; i++) {	
operation_point_layer_set_idx[i]	ue(v)
for(j = 0; j <= vps_max_layer_id; j++)	
op_dpb_info_parameters(i,j)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 38B

op_dpb_info_parameters(j) {	
vps_max_sub_layers_minus1[j]	u(3)
vps_sub_layer_ordering_info_present_flag[j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[j] ? 0 : vps_max_sub_layers_minus1[j]);	
k <= vps_max_sub_layers_minus1[j]; k++) {	
vps_max_dec_pic_buffering_minus1[j][k]	ue(v)
vps_max_num_reorder_pics[j][k]	ue(v)
vps_max_latency_increase_plus1[j][k]	ue(v)
}	
}	

FIG. 39A

op_dpb_info_parameters(id, j) {	
vps_max_sub_layers_minus1[id][j]	u(3)
vps_sub_layer_ordering_info_present_flag[id][j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[id][j] ? 0 : vps_max_sub_layers_minus1[id][j]); k <= vps_max_sub_layers_minus1[id][j]; k++) {	
vps_max_dec_pic_buffering_minus1[id][j][k]	ue(v)
vps_max_num_reorder_pics[id][j][k]	ue(v)
vps_max_latency_increase_plus1[id][j][k]	ue(v)
}	
}	

FIG. 39B

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx = output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(i)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 40

oop_dpb_info_parameters() {	
vps_max_sub_layers_minus1[j]	u(3)
vps_sub_layer_ordering_info_present_flag[j]	u(1)
for(k=(vps_sub_layer_ordering_info_present_flag[j]?0: vps_max_sub_layers_minus1[j]); k <= vps_max_sub_layers_minus1[j]; k++){	
vps_max_dec_pic_buffering_minus1[j][k]	ue(v)
vps_max_num_reorder_pics[j][k]	ue(v)
vps_max_latency_increase_plus1[j][k]	ue(v)
}	
}	

FIG. 41

oop_dpb_info_parameters() {	
vps_max_sub_layers_minus1	u(3)
vps_sub_layer_ordering_info_present_flag	u(1)
for(k=(vps_sub_layer_ordering_info_present_flag?0: vps_max_sub_layers_minus1); k <= vps_max_sub_layers_minus1; k++){	
vps_max_dec_pic_buffering_minus1[k]	ue(v)
vps_max_num_reorder_pics[k]	ue(v)
vps_max_latency_increase_plus1[k]	ue(v)
}	
}	

FIG. 42

num_dpb_info_parameters	ue(v)
for(i=0; i<num_dpb_info_parameters; i++){	
output_point_layer_set_idx[i]	ue(v)
oplsIdx=output_point_layer_set_idx[i]	
for(j=0; j<=vps_max_layer_id; j++){	
oop_dpb_info_parameters()	
}	
}	

FIG. 43

oop_dpb_info_parameters(id, j) {	
vps_max_sub_layers_minus1[id][j]	u(3)
vps_sub_layer_ordering_info_present_flag[id][j]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[id][j] ? 0 : vps_max_sub_layers_minus1[id][j]); k <= vps_max_sub_layers_minus1[id][j]; k++) {	
vps_max_dec_pic_buffering_minus1[id][j][k]	ue(v)
vps_max_num_reorder_pics[id][j][k]	ue(v)
vps_max_latency_increase_plus1[id][j][k]	ue(v)
}	
}	

FIG. 44

num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx=output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(oplsIdx, j)	
}	

FIG. 45

num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oplsIdx=output_point_layer_set_idx[i]	
for(j = 0; j <= vps_max_layer_id; j++)	
oop_dpb_info_parameters(i, j)	
}	

FIG. 46

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
avc_base_layer_flag	u(1)
splitting_flag	u(1)
for(i = 0, NumScalabilityTypes = 0; i < 16; i++) {	
scalability_mask[i]	u(1)
NumScalabilityTypes += scalability_mask[i]	
}	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id_len_minus1[j]	u(3)
...	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info(i)	
}	
}	

layer_dpb_info(i) {	
vps_max_sub_layers_minus1[i]	u(3)
vps_sub_layer_ordering_info_present_flag[i]	u(1)
for(k = (vps_sub_layer_ordering_info_present_flag[i] ? 0 : vps_max_sub_layers_minus1[i]);	
k <= vps_max_sub_layers_minus1[i]; k++) {	
vps_max_dec_pic_buffering_minus1[i][k]	ue(v)
vps_max_num_reorder_pics[i][k]	ue(v)
vps_max_latency_increase_plus1[i][k]	ue(v)
}	
}	

FIG. 47

oop_dpb_info_parameters(id) / op_dpb_info_parameters(id)/ layer_dpb_info(id) {	
vps_sub_layer_ordering_info_present_flag[id]	u(1)
for(i = (vps_sub_layer_ordering_info_present_flag ? 0 : vps_max_sub_layers_minus1);	
i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering_minus1[id][i]	ue(v)
vps_max_num_reorder_pics[id][i]	ue(v)
vps_max_latency_increase_plus1[id][i]	ue(v)
}	
}	

FIG. 48

vps_extension() {	]	Tên
while(!byte_aligned())		
vps_extension_byte_alignment_reserved_one_bit		u(1)
avc_base_layer_flag		u(1)
splitting_flag		u(1)
for(i = 0; NumScalabilityTypes = 0; i < 16; i++) {		
scalability_mask[i]		u(1)
NumScalabilityTypes += scalability_mask[i]		
}		
for(j = 0; j < NumScalabilityTypes; j++)		
dimension_id_len_minus1[j]		u(3)
vps_nuh_layer_id_present_flag		u(1)
for(i = 1; i <= vps_max_layers_minus1; i++) {		
if(vps_nuh_layer_id_present_flag)		
layer_id_in_nuh[i]		u(6)
for(j = 0; j < NumScalabilityTypes; j++)		
dimension_id[i][j]		u(v)
}		
for(i = 1; i <= vps_max_layers_minus1; i++) {		
max_sub_layers_vps_predict_flag[i]		u(1)
if(!max_sub_layers_vps_predict_flag[i])		
max_sub_layers_vps_minus1[i]		u(3)
}		
vps_number_layer_sets_minus1		u(10)
vps_num_profile_tier_level_minus1		u(6)
for(i = 1; i <= vps_num_profile_tier_level_minus1; i++) {		
vps_profile_present_flag[i]		u(1)
if(!vps_profile_present_flag[i])		
profile_ref_minus1[i]		u(6)
profile_tier_level(vps_profile_present_flag[i], vps_max_sub_layers_minus1)		
}		

FIG. 49A

multiple_output_layer_sets_in_layer_setmore_output_layer_sets_than_default_flag	u(1)
if(!multiple_output_layer_sets_in_layer_setmore_output_layer_sets_than_default_flag)	
numOutputLayerSets = vps_number_layer_sets_minus1 + 1	
else {	
num_add_output_layer_sets	u(10)
numOutputLayerSets =	
num_add_output_layer_sets + vps_number_layer_sets_minus1 + 1	
}	
if(numOutputLayerSets > 1)	
default_one_target_output_layer_flag	u(1)
for(i = 1; i < numOutputLayerSets; i++) {	
if(i > vps_num_layer_sets_minus1) {	
output_layer_set_idx_minus1[i]	u(v)
lsIdx = output_layer_set_idx_minus1[i] + 1	
for(j = 0 ; j < NumLayersInIdList[lsIdx] - 1; j++)	
output_layer_flag[i][j]	u(1)
}	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
oop_dpb_maxbuffering_parameters(i)	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info_parameters(i)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

FIG. 49B

oop_dpb_maxbuffering_parameters(i) {	
sub_layer_vps_buf_info_present_flag[i]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i] - 1); j <= MaxSubLayers[i] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][j]	ue(v)
}	

FIG. 50

layer_dpb_info_parameters(i) {	
sub_layer_vps_ordering_info_present_flag[i]	u(1)
for(j = (sub_layer_vps_ordering_info_present_flag[i] ? 0 :	
max_sub_layers_vps_minus1[i]); j <= max_sub_layers_vps_minus1[i]; j++) {	
max_vps_num_reorder_pics[i][j]	ue(v)
max_vps_latency_increase_plus1[i][j]	ue(v)
}	
}	

FIG. 51

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i > 0 && (MaxSubLayers[i] == MaxSubLayers[i-1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
oop_dpb_maxbuffering_parameters(i)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 52

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; i++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 53

oop_dpb_maxbuffering_parameters(i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i][k] ? 0 :	
max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 54

oop_dpb_maxbuffering_parameters(i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 55

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i > 0 && (MaxSubLayers[i] == MaxSubLayers[i - 1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; k++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
...	
}	

FIG. 56

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i][k] ? 0 : max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++) {	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 57

oop_dpb_maxbuffering_parameters (i, k) {	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++) {	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 58

vps_extension() {	Tên
while(!byte_aligned())	
vps_extension_byte_alignment_reserved_one_bit	u(1)
...	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
max_sub_layers_vps_predict_flag[i]	u(1)
if(!max_sub_layers_vps_predict_flag[i])	
max_sub_layers_vps_minus1[i]	u(3)
}	
...	
profile_level_tier_idx[i]	u(v)
}	
num_dpb_info_parameters	ue(v)
for(i = 0; i < num_dpb_info_parameters; i++) {	
if(i > 0 && (MaxSubLayers[i] == MaxSubLayers[i - 1]))	
sub_layer_vps_buf_info_predict_flag[i]	u(1)
if(!sub_layer_vps_buf_info_predict_flag[i]) {	
output_point_layer_set_idx[i]	ue(v)
for(k = 1; k <= vps_max_layers_minus1; i++) {	
oop_dpb_maxbuffering_parameters(i, k)	
}	
}	
}	
layer_dpb_info_parameters_presence_flag	u(1)
If(layer_dpb_info_parameters_presence_flag) {	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
if(max_sub_layers_vps_predict_flag[i])	
sub_layer_vps_ordering_info_predict_flag[i]	u(1)
if(!sub_layer_ordering_info_predict_flag[i])	
layer_dpb_info_parameters(i)	
}	
}	
}	
...	
}	

FIG. 59

oop_dpb_maxbuffering_parameters(i,k){	
sub_layer_vps_maxbuffering_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_maxbuffering_info_present_flag[i][k] ? 0 : max_sub_layers_vps_minus1[k]); j <= max_sub_layers_vps_minus1[k]; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 60

oop_dpb_maxbuffering_parameters(i,k){	
sub_layer_vps_buf_info_present_flag[i][k]	u(1)
for(j = (sub_layer_vps_buf_info_present_flag[i] ? 0 : MaxSubLayers[i][k] - 1); j <= MaxSubLayers[i][k] - 1; j++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
}	

FIG. 61

seq_parameter_set_rbsp() {	Tên ₁
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
if(nuh_layer_id == 0)	
profile_tier_level(1, sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
...	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
if(nuh_layer_id > 0)	
sps_dpb_params_present_flag	u(1)
if((nuh_layer_id == 0) ((nuh_layer_id > 0) && sps_dpb_params_present_flag)) {	
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1);	
i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
}	
log2_min_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_luma_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
...	
sps_extension_flag	u(1)
if(sps_extension_flag) {	
sps_extension()	
sps_extension2_flag	u(1)
if(sps_extension2_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

FIG. 62

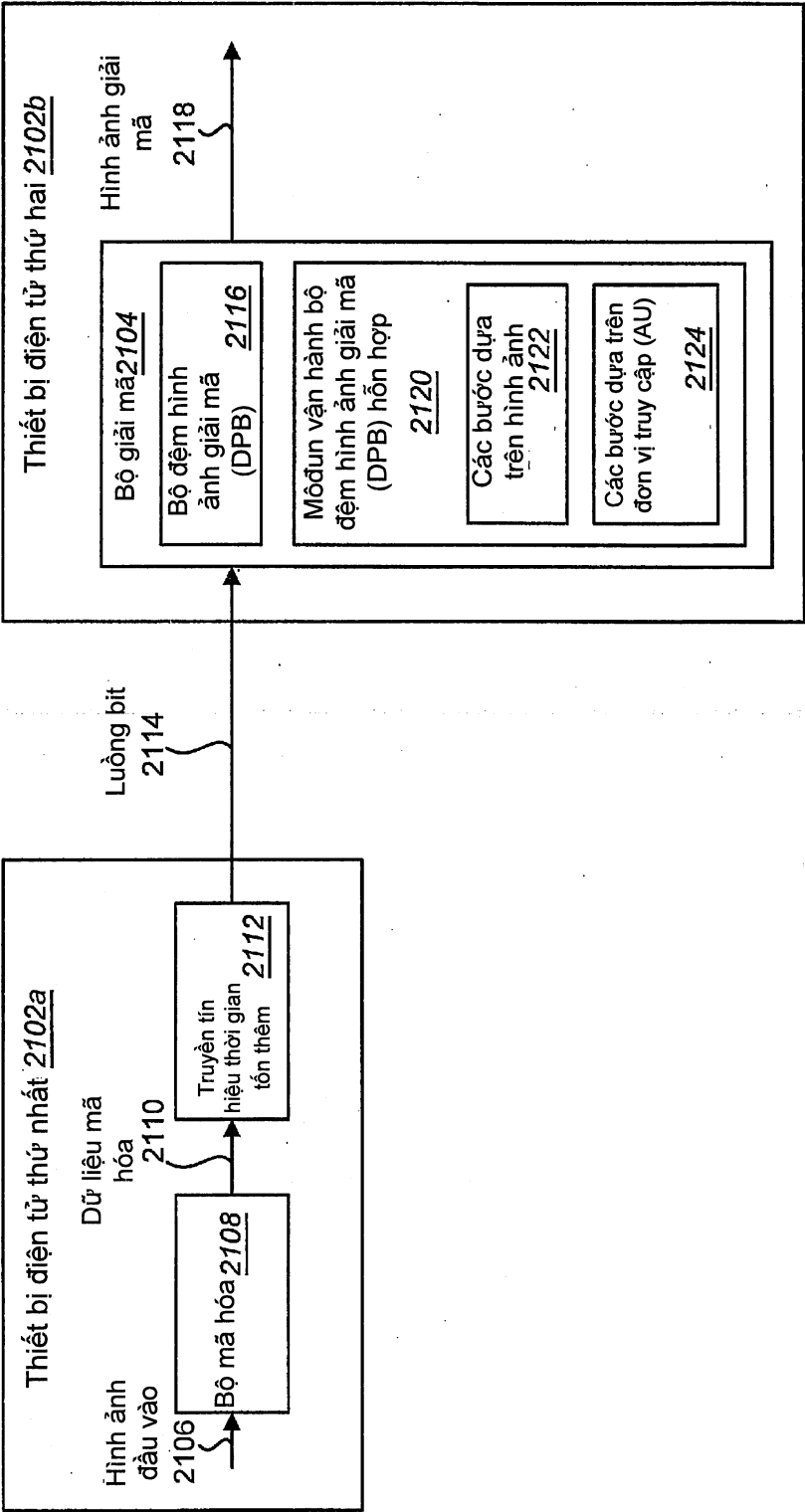
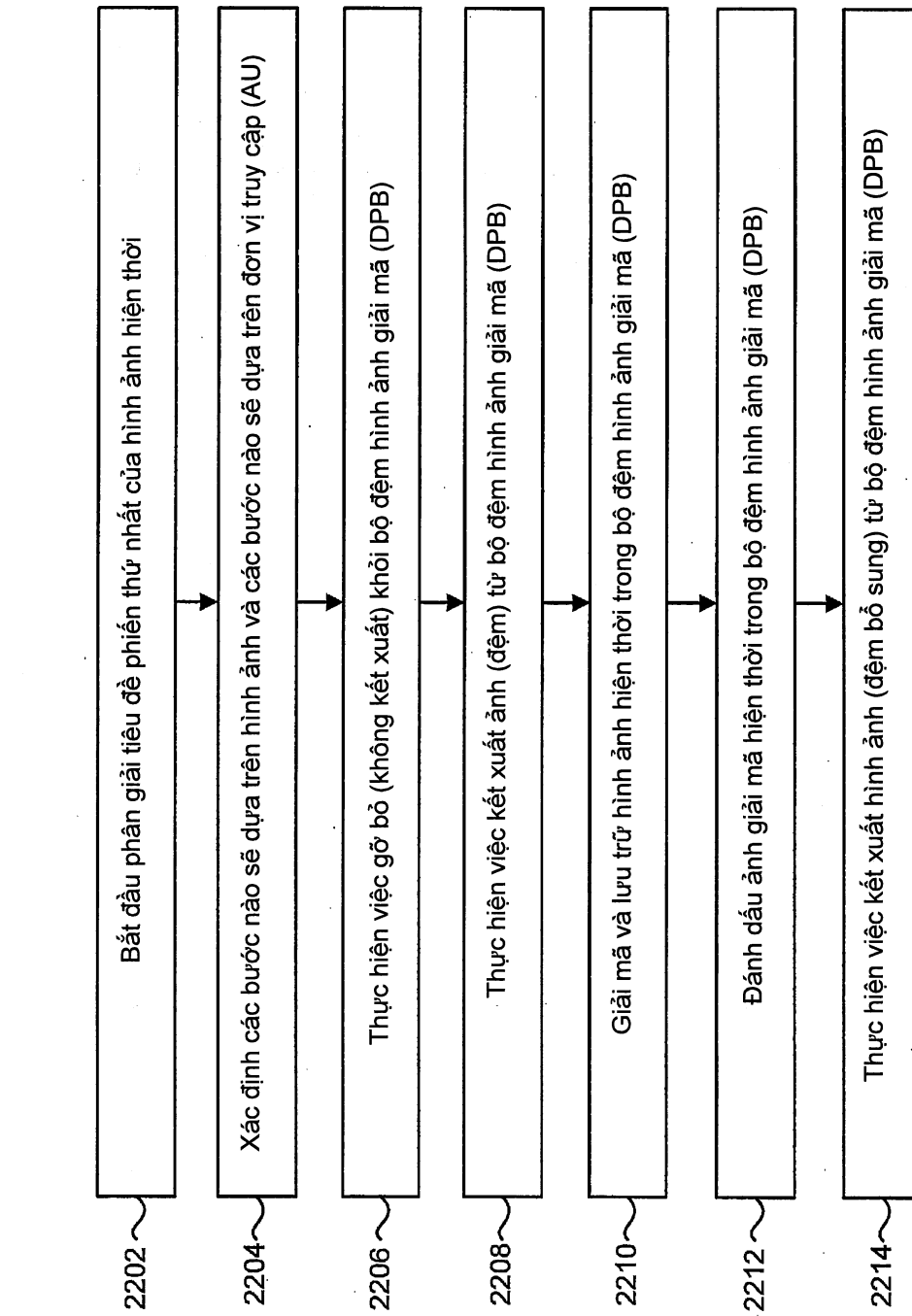


FIG. 63

**FIG. 64**

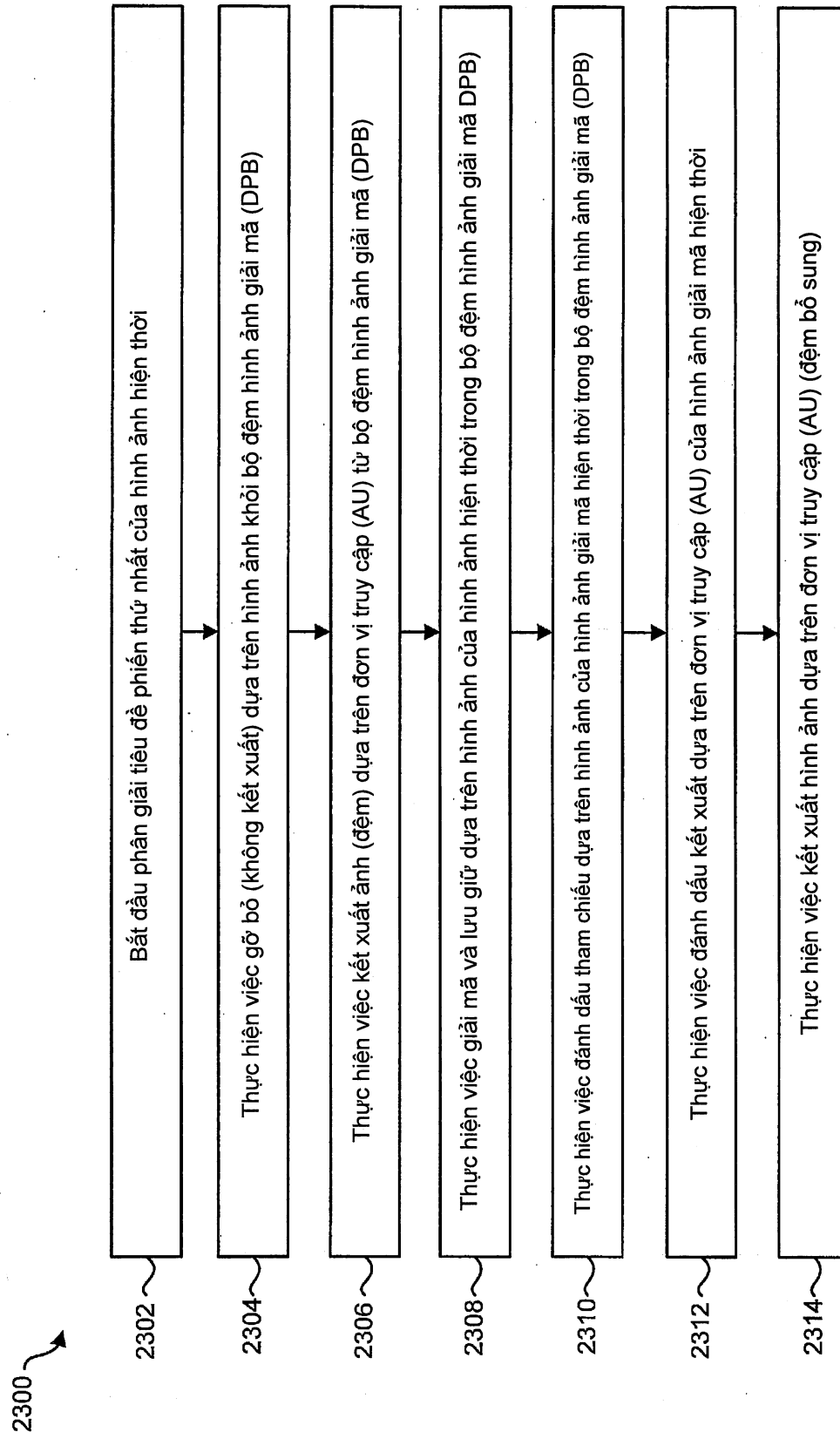


FIG. 65

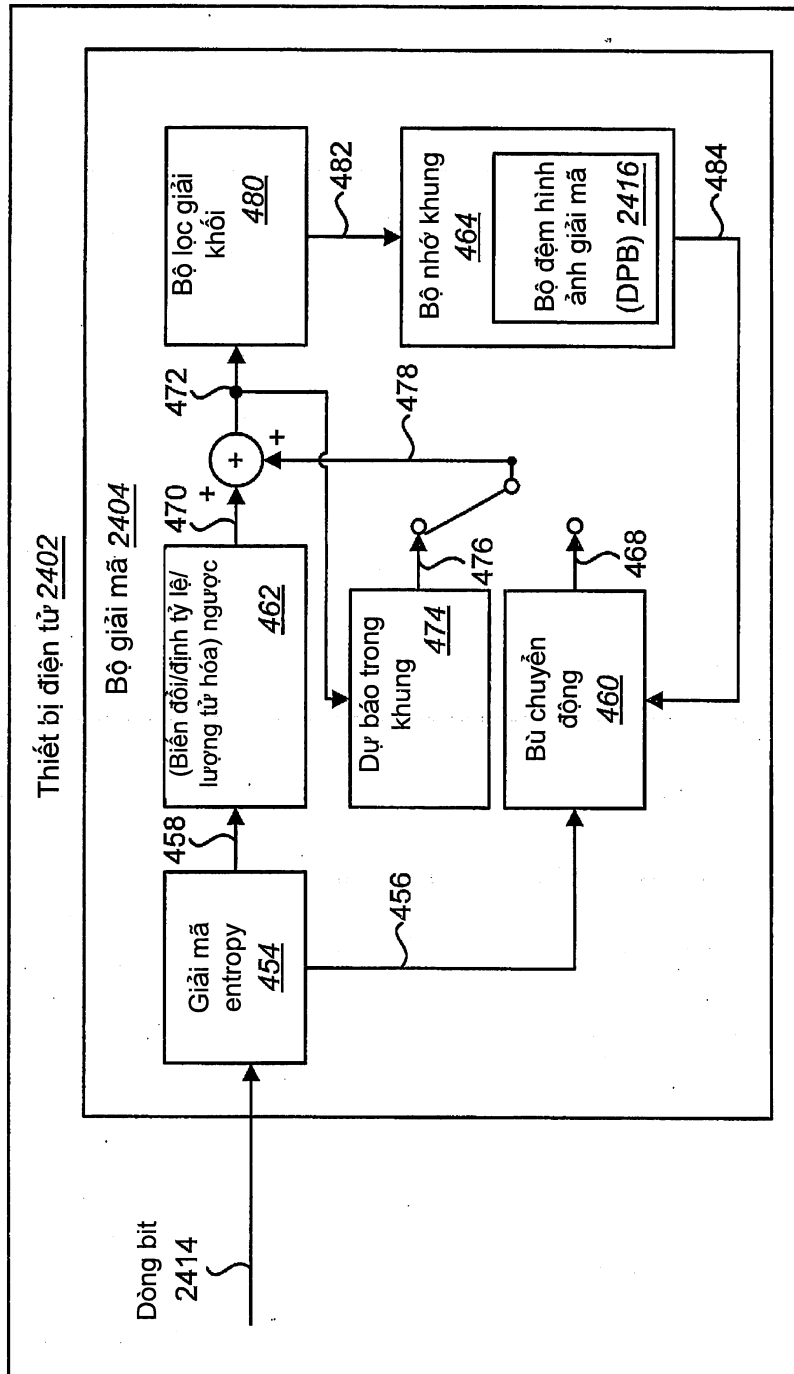


FIG. 66

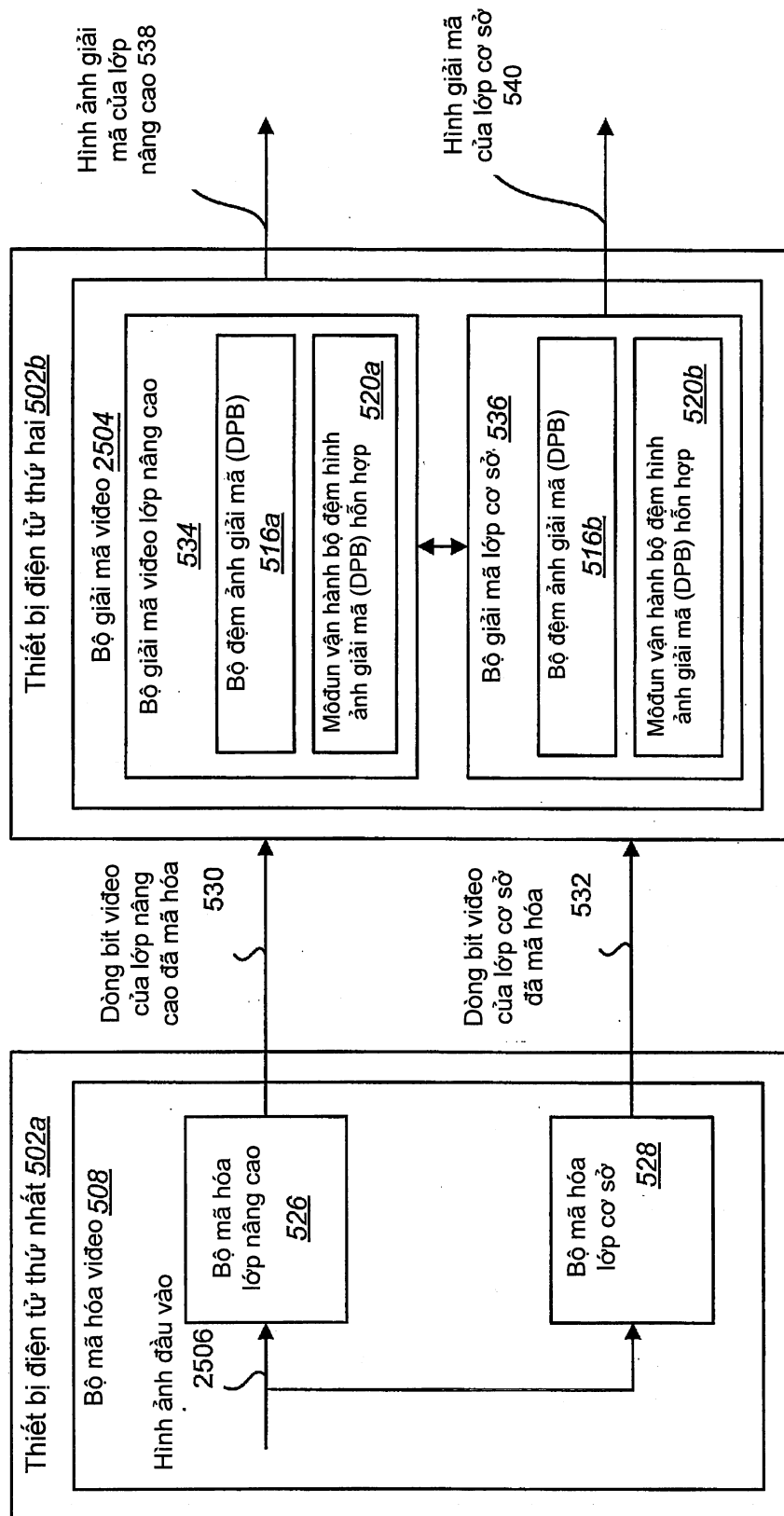


FIG. 67A

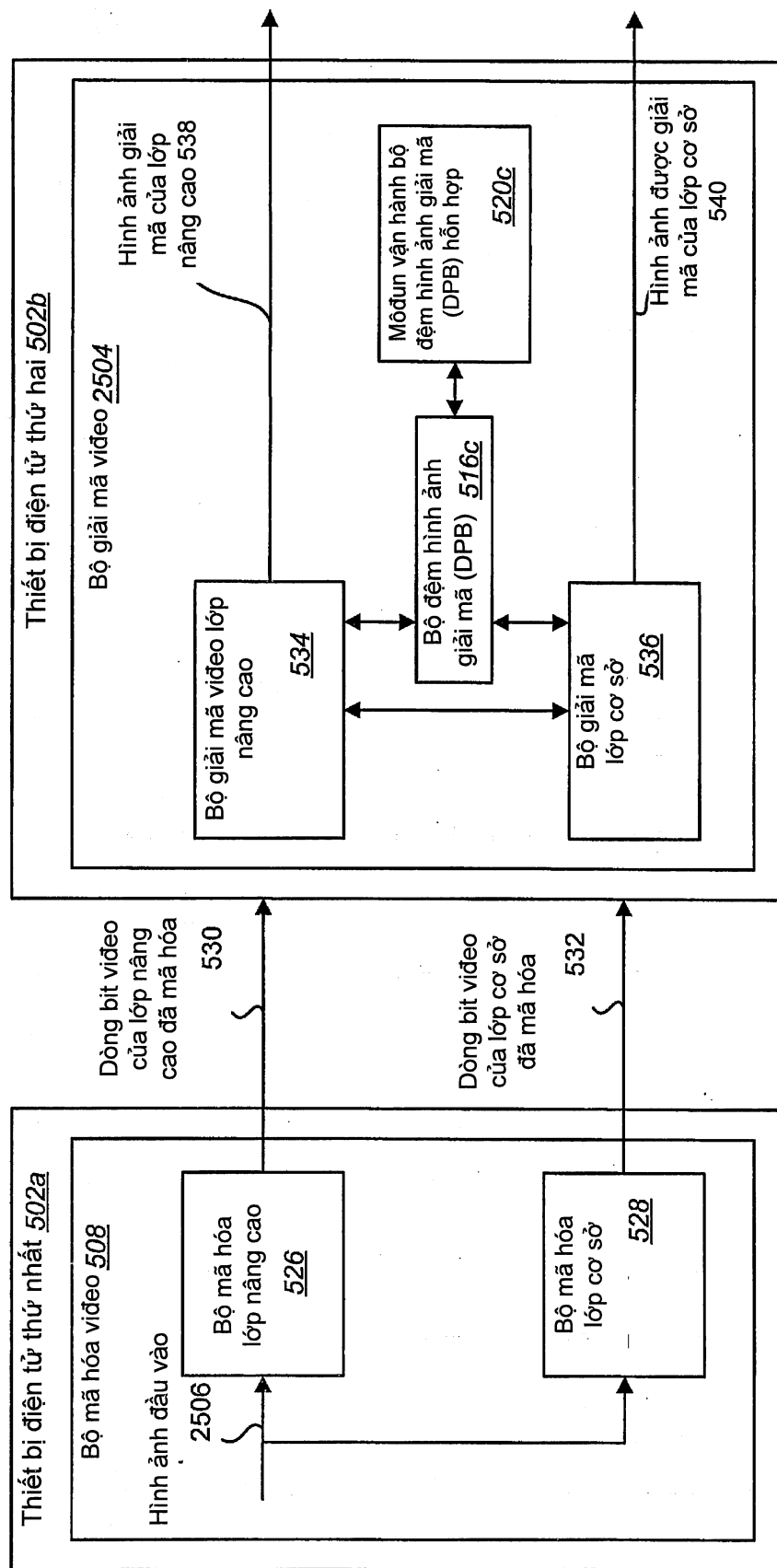


FIG. 67B

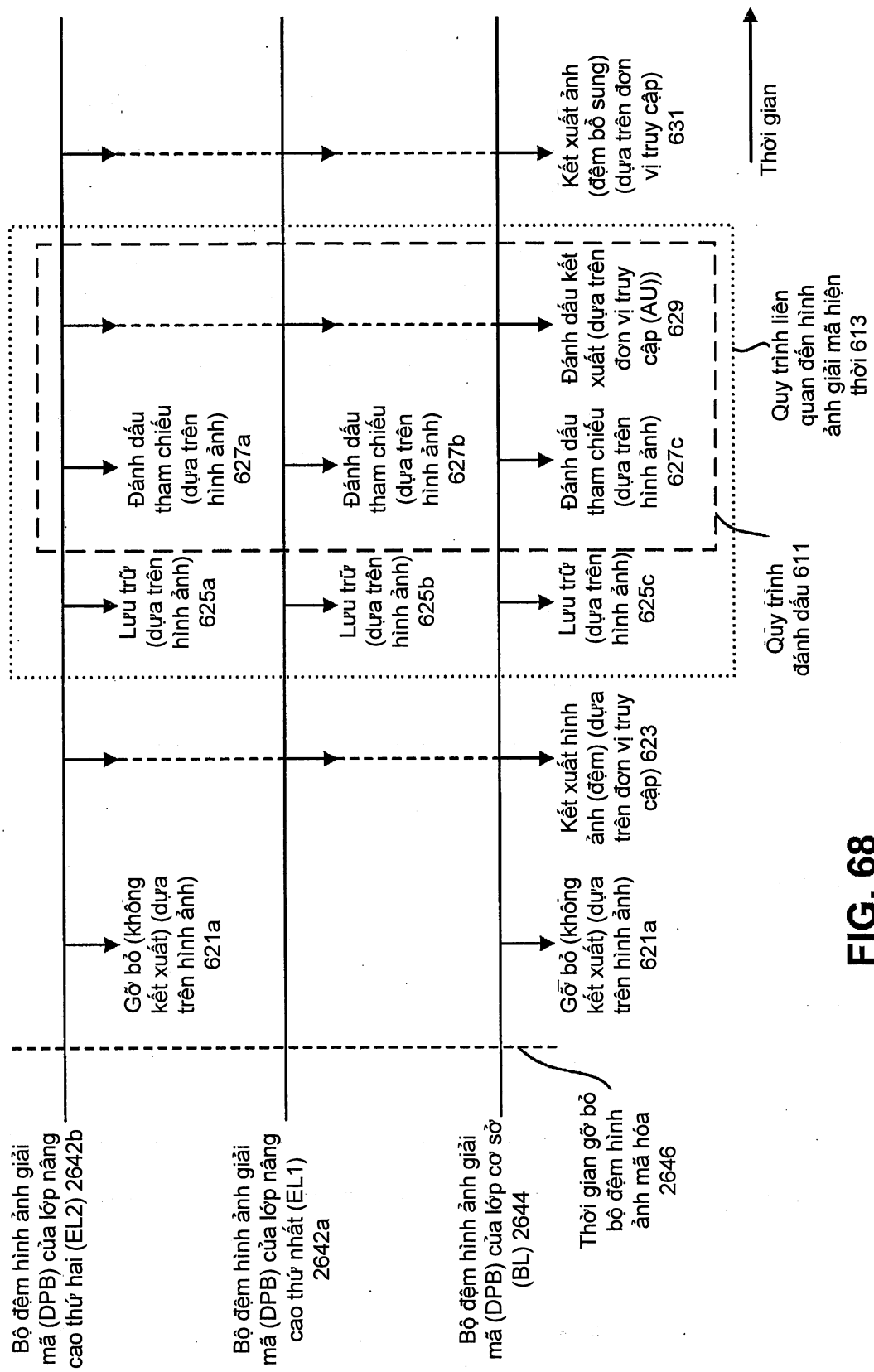


FIG. 68

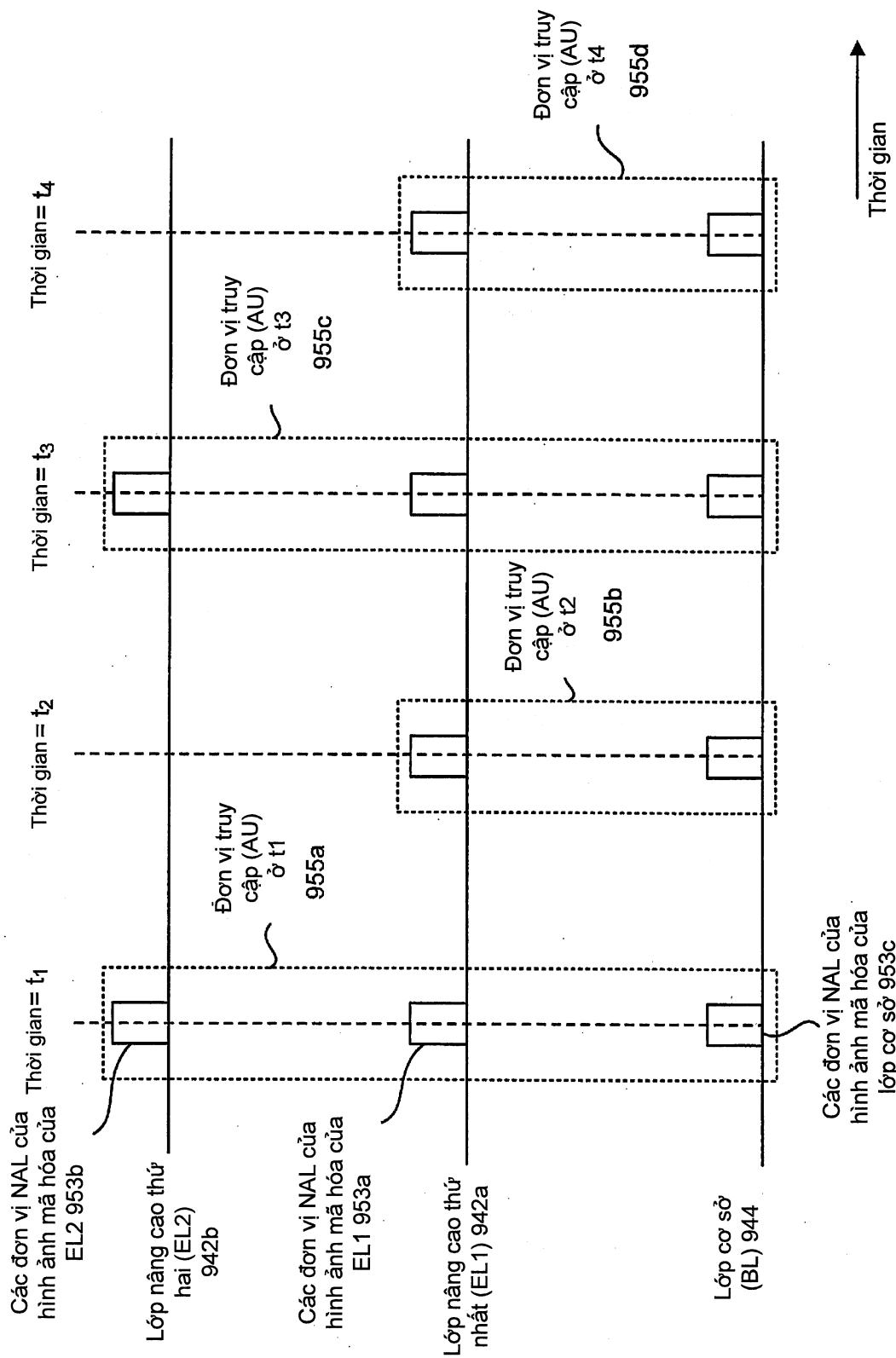


FIG. 69

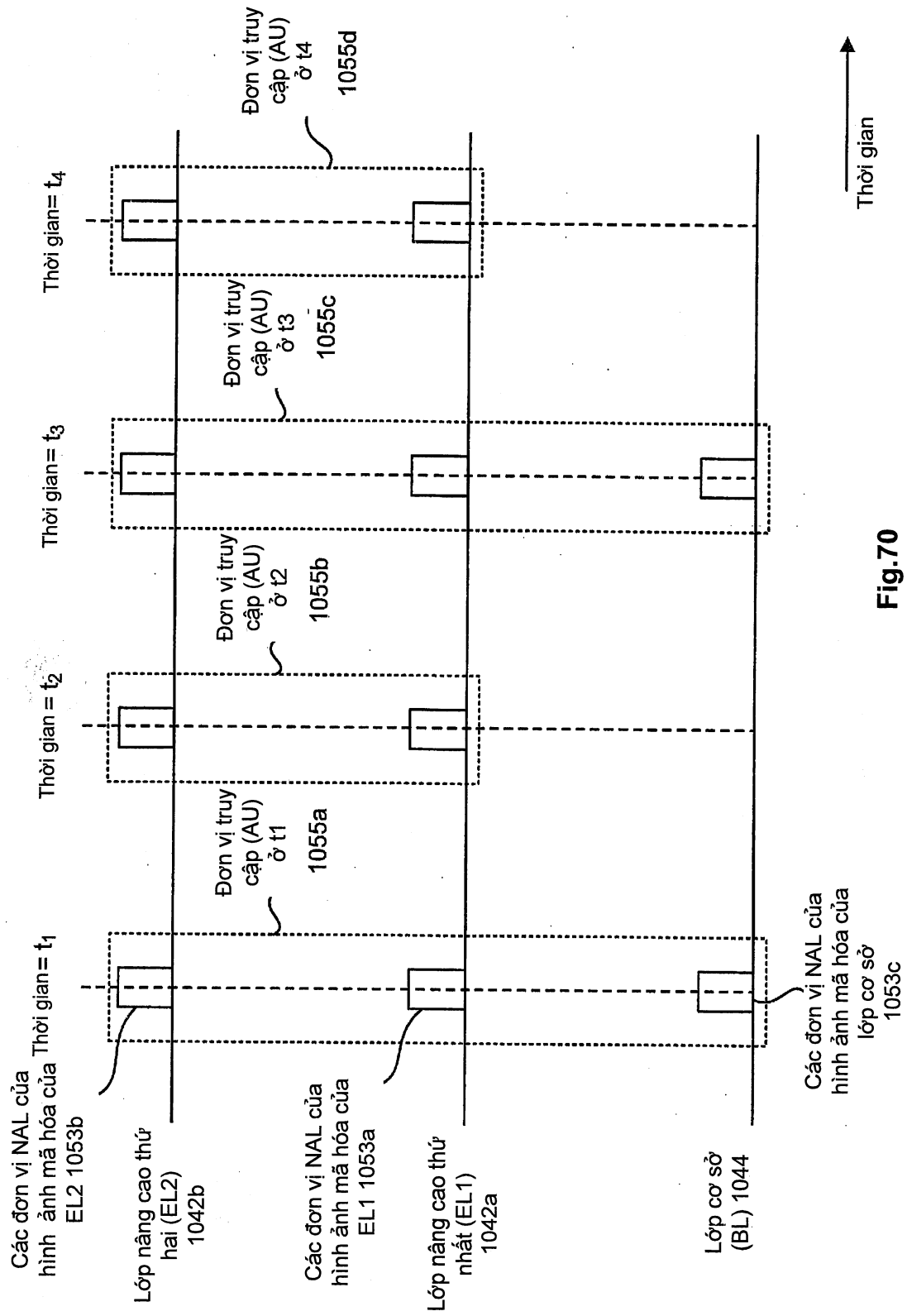


Fig.70