(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) **1-0028185**

(51)[7] **H04N 19/70**

(13) **B**

(21) 1-2016-04710 (22) 04/04/2014
(62) 1-2015-04133
(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) 25/05/2021 398 (43) 25/01/2017 346A
(73) Dolby International AB (SE)
Apollo Building, 3E Herikerbergweg 1-35 Amsterdam Zuidoost, Netherlands 1101 CN
(72) DESHPANDE, Sachin G. (US).
(74) Công ty TNHH Quốc tế D&N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP GIẢI MÃ DÒNG BIT VIĐEO

(57) Sáng chế đề cập đến phương pháp giải mã dòng bit viđeo, phương pháp này bao gồm các bước: thu tập hợp các thông số định thời hình ảnh bao gồm thông số trễ loại bỏ bộ đệm hình ảnh được mã hóa để loại bỏ gói truy nhập từ bộ đệm hình ảnh được mã hóa ở cấp độ gói truy nhập; thu dữ liệu đại diện của ảnh viđeo được mã hóa; lưu trữ dữ liệu này trong bộ đệm hình ảnh được mã hóa; và xác định liệu gói truy nhập này có cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa này hay không để giải mã ở cấp độ gói truy nhập hoặc gói giải mã có cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa hay không để giải mã ở cấp độ hình ảnh phụ.



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++) {	
addlClsIdx = addl_output_layer_sets_idx[i]	ue(v)
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlClsIdx][j])	
output_layer_flag[addlClsIdx][j]	u(1)
}	
}	

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp giải mã dòng bit viđeo.

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 cải tiến. 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, trình đa các phương tiện thu, mạch tích hợp v.v...

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

Tính phổ biến ngày càng tăng của các phương tiện thu số đã đặt ra một số vấn đề. Ví dụ, các phương tiện thu số chất lượng cao đối với lưu trữ phải thể hiện có hiệu quả, việc truyền và phát lại nhanh chóng cho thấy có một số thách thức. Có thể quan sát thấy từ sáng chế là, các hệ thống và các phương pháp mà chúng đại diện cho các phương tiện thu số với tính năng cải tiến có hiệu quả có thể có tính kích hoạt.

Các thông tin nêu trên và các mục đích khác, các đặc trưng và ưu điểm của sáng chế sẽ được hiểu rõ hơn khi xem xét phần mô tả chi tiết dưới đây của sáng chế, có liên quan đến các hình vẽ kèm theo.

Bản chất kỹ thuật của sáng chế

Cách giải quyết vấn đề

Một khía cạnh của sáng chế đề xuất phương pháp để giải mã dòng bit viđeo bao gồm các bước: (a) thu dòng bit cơ sở đại diện cho dãy viđeo được mã hóa; (b) thu các

dòng bit nâng cao đại diện cho dãy viđeo được mã hóa đã nêu; (c) thu tập hợp tham số viđeo có chứa các thành phần 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 viđeo đã nêu chứa một phần tử cú pháp để truyền tín hiệu phần mở rộng tập hợp tham số viđeo; (d) thu phần mở rộng tập hợp tham số viđeo đã nêu chứa các thành phần 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 ra bao gồm thông tin chỉ báo thay đổi trong ít nhất một tập hợp lớp ra.

Một khía cạnh của sáng chế đề xuất phương pháp để giải mã dòng bit viđeo bao gồm các bước: (a) thu dòng bit cơ sở đại diện cho dãy viđeo được mã hóa; (b) thu các dòng bit nâng cao đại diện cho dãy viđeo được mã hóa đã nêu; (c) thu tập hợp tham số viđeo có chứa các thành phần 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 viđeo đã nêu chứa một phần tử cú pháp để truyền tín hiệu phần mở rộng tập hợp tham số viđeo; (d) thu phần tập hợp tham số viđeo đã nêu chứa các thành phần cú pháp mà nó bao gồm các thông số liên quan đến bộ đệm hình ảnh được mã hóa đối với bộ đệm hình ảnh được mã hóa cho ít nhất một trong các dòng bit nâng cao đã nêu.

Một khía cạnh của sáng chế đề xuất phương pháp dùng để mã hóa viđeo, bao gồm các bước: bắt đầu phân tách tiêu đề phiến thứ nhất của hình ảnh hiện thời; xác định các bước nào được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) sẽ dựa trên hình ảnh và các bước nào sẽ dựa vào gói truy nhập (AU); thực hiện việc loại bỏ từ DPB; thực hiện việc kết xuất hình ảnh từ DPB; thực hiện việc giải mã và lưu trữ hình ảnh được mã hóa hiện thời trong DPB; đánh dấu hình ảnh được mã hóa hiện thời trong DPB; và thực hiện kết xuất hình ảnh bổ sung từ DPB.

Một khía cạnh của sáng chế đề xuất thiết bị điện tử được tạo cấu hình dùng để mã hóa viđeo, 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 tách tiêu đề phiến thứ nhất của hình ảnh hiện thời; xác định các bước nào được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) sẽ dựa vào hình ảnh và các bước nào sẽ dựa vào gói truy nhập (AU);

thực hiện việc loại bỏ từ DPB; thực hiện việc kết xuất hình ảnh từ DPB; thực hiện việc giải mã và lưu trữ hình ảnh được mã hóa hiện thời trong DPB; đánh dấu hình ảnh được mã hóa hiện thời trong DPB; và thực hiện kết xuất hình ảnh bổ sung từ DPB.

Mô tả vắn tắt các hình vẽ

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

Fig.1B thể hiện 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 để gửi thông báo và bước đệm dòng bit có thể được thực hiện.

Fig.2 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp dùng để gửi thông báo.

Fig.3 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp dùng để xác định một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập.

Fig.4 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp dùng để đệm dòng bit;

Fig.5 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp dùng để xác định một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập.

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

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

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

Fig.8 minh họa một số thành phần có thể được sử dụng trong thiết bị điện tử truyền phát.
Fig.9 thể hiện sơ đồ khối minh họa một số thành phần có thể được sử dụng trong thiết bị điện tử thu nhận.

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

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

Fig.12 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp để vận hành bộ đệm hình ảnh được mã hóa.

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

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

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

Fig.14 minh họa cú pháp chung của Gói NAL.

Fig.15 minh họa tập hợp thông số viđeo hiện thời.

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

Fig.17 minh họa tập hợp thông số viđeo có tính ví dụ.

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

Fig.19 minh họa tập hợp thông số viđeo có tính ví dụ.

Fig.20 minh họa tập hợp thông số viđeo hiện thời.

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

Fig.22 minh họa tập hợp thông số viđeo có tính ví dụ.

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

Fig.24 minh họa tập hợp thông số viđeo có tính ví dụ.

Fig.25 minh họa tập hợp thông số viđeo có tính ví dụ.

Fig.26 minh họa tập hợp thông số viđeo có tính ví dụ.

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

Fig.28 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.29 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.30 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.31 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.32 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.33 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.34 minh họa cú pháp của tập hợp thông số viđeo có tính ví dụ.

Fig.35 minh họa cú pháp phần mở rộng tập hợp thông số viđeo có tính ví dụ.

Fig.36 minh họa cú pháp thay đổi các tập hợp lớp ra có tính ví dụ.

Fig.37 minh họa cú pháp thay đổi các tập hợp lớp ra có tính ví dụ.

Fig.38A minh họa cú pháp phần mở rộng thông số viđeo có tính ví dụ.

Fig.38B minh họa cú pháp phần mở rộng thông số viđeo có tính ví dụ.

Fig.39A minh họa cú pháp op_dpb_info_parameters(j) có tính ví dụ.

Fig.39B minh họa cú pháp op_dpb_info_parameters(j) có tính ví dụ.

Fig.40 minh họa cú pháp phần mở rộng thông số viđeo có tính ví dụ khác.

Fig.41 minh họa cú pháp oop_dpb_info_parameters(j) có tính ví dụ khác.

Fig.42 minh họa cú pháp oop_dpb_info_parameters(j) có tính ví dụ khác.

Fig.43 minh họa cú pháp num_dpb_info_parameters có tính ví dụ.

Fig.44 minh họa cú pháp oop_dpb_info_parameters(j) có tính ví dụ khác.

Fig.45 minh họa cú pháp num_dpb_info_parameters có tính ví dụ khác.

Fig.46 minh họa cú pháp num_dpb_info_parameters có tính ví dụ khác.

Fig.47 minh họa cú pháp phần mở rộng thông số viđeo có tính ví dụ khác và layer_dpb_info(i).

Fig.48 minh họa cú pháp oop_dpb_info_parameters andlayer_dpb_info(i) có tính ví dụ.

Fig.49A minh họa vps_extension() có tính ví dụ khác.

Fig.49B minh họa vps_extension() có tính ví dụ khác.

Fig.50 minh họa oop_dpb_maxbuffering_parameters(i) có tính ví dụ.

Fig.51 minh họa layer_dpb_info_parameters(i) có tính ví dụ.

Fig.52 minh họa vps_extension() có tính ví dụ khác.

Fig.53 minh họa vps_extension() có tính ví dụ khác.

Fig.54 minh họa oop_dpb_maxbuffering_parameters(i,k) có tính ví dụ.

Fig.55 minh họa oop_dpb_maxbuffering_parameters(i,k) có tính ví dụ.

Fig.56 minh họa vps_extension() có tính ví dụ khác.

Fig.57 minh họa oop_dpb_maxbuffring_parameters(i,k) có tính ví dụ.

Fig.58 minh họa oop_dpb_maxbuffring_parameters(i,k) có tính ví dụ.

Fig.59 minh họa oop_dpb_maxbuffring_parameters(i,k) có tính ví dụ.

Fig.60 minh họa oop_dpb_maxbuffring_parameters(i,k) có tính ví dụ.

Fig.61 minh họa oop_dpb_maxbuffering_parameters(i,k) có tính ví dụ.

Fig.62 minh họa seq_parameter_set_rbsp() có tính ví dụ.

Fig.63 thể hiện sơ đồ khối minh họa việc mã hóa viđeo giữa nhiều thiết bị điện tử.

Fig.64 thể hiện sơ đồ luồng về phương pháp để vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp.

Fig.65 thể hiện sơ đồ luồng về phương pháp khác để vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp.

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

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

Fig.67B thể hiện sơ đồ khối minh họa việc sử dụng bộ đệm hình ảnh (DPB) được chia sẻ và các môđun vận hành của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp được chia sẻ dùng cho lớp cơ sở và lớp nâng cao.

Fig.68 thể hiện biểu đồ định thời mô tả việc vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp.

Fig.69 thể hiện sơ đồ khối minh họa kết cấu và định thời cho gói của lớp mạng trừu tượng (NAL) của các lớp cho các hình ảnh được mã hóa và các gói truy nhập (AU) khi lớp nâng cao thứ hai (EL2) có tốc độ ảnh thấp hơn lớp cơ sở thứ nhất (BL) và lớp nâng cao thứ nhất (EL1).

Fig.70 thể hiện sơ đồ khối minh họa kết cấu và định thời dùng cho các gói của lớp mạng trừu tượng (NAL) của các lớp cho các hình ảnh được mã hóa và các gói truy nhập (AU) khi lớp cơ sở thứ nhất (BL) có tốc độ ảnh thấp hơn lớp nâng cao thứ nhất (EL1) và lớp nâng cao thứ hai (EL2).

Mô tả chi tiết sáng chế

Ví dụ 1

Thiết bị điện tử dùng để gửi thông báo được mô tả dưới đây. Thiết bị điện tử bao gồm bộ xử lý và các lệnh được lưu trữ trong bộ nhớ mà nó truyền thông điện tử với bộ xử lý. Thiết bị điện tử này sẽ xác định, khi bộ đệm hình ảnh được mã hóa (CPB) hỗ trợ hoạt động ở cấp độ hình ảnh phụ, liệu có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung trong thông báo thông tin nâng cao bổ sung (SEI) định thời hình ảnh hay không. Thiết bị điện tử này cũng sẽ tạo ra, khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ được bao gồm trong thông báo SEI định thời hình ảnh (hoặc thông báo SEI nào đó hoặc tập hợp thông số khác nào đó ví dụ tập hợp thông số hình ảnh hoặc tập hợp thông số

chuỗi hoặc tập hợp thông số viđeo hoặc tập hợp thông số tương thích), thông số trễ loại bỏ CPB của gói giải mã chung, trong đó thông số trễ loại bỏ CPB của gói giải mã chung được áp dụng cho tất cả các gói giải mã trong gói truy nhập từ CPB. Thiết bị điện tử này cũng sẽ tạo ra, khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ không được bao gồm trong thông báo SEI định thời hình ảnh, thông số trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập. Thiết bị điện tử này cũng sẽ gửi thông báo SEI định thời hình ảnh với thông số trễ loại bỏ CPB của gói giải mã chung hoặc các thông số trễ loại bỏ CPB của gói giải mã.

Thông số trễ loại bỏ CPB của gói giải mã chung có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB của gói giải mã ngay trước đó trước khi loại bỏ từ CPB gói giải mã hiện thời trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Ngoài ra, khi gói giải mã là gói giải mã trong gói truy nhập, thông số trễ loại bỏ CPB của gói giải mã chung có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB gói giải mã sau cùng trong gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm gần nhất trong gói truy nhập trước đó trước khi loại bỏ từ CPB gói giải mã thứ nhất trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Ngược lại, khi gói giải mã là gói giải mã không phải là thứ nhất trong gói truy nhập, thông số trễ loại bỏ CPB của gói giải mã chung có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB gói giải mã trước đó trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh trước khi loại bỏ từ CPB gói giải mã hiện thời trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Các thông số trễ loại bỏ CPB của gói giải mã có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB gói giải mã sau cùng trước khi loại bỏ từ CPB gói giải mã thứ i trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Thiết bị điện tử này có thể tính toán các thông số trễ loại bỏ CPB của gói giải mã theo phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun trong đó cpb_removal_delay_length_minus1 + 1 là độ dài của thông số trễ loại bỏ CPB của gói giải mã chung.

Thiết bị điện tử này cũng có thể tạo ra, khi CPB hỗ trợ hoạt động ở cấp độ gói truy nhập, thông báo SEI định thời hình ảnh bao gồm thông số trễ loại bỏ CPB mà nó xác định bao nhiêu tích tắc đồng hồ để chờ sau khi loại bỏ từ CPB của gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm gần nhất trong gói truy nhập trước đó trước khi loại bỏ từ CPB dữ liệu của gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Thiết bị điện tử này có thể xác định liệu CPB có hỗ trợ hoạt động ở cấp độ hình ảnh phụ hoặc cấp độ gói truy nhập hay không. Việc này có thể bao gồm việc xác định cờ hiệu định thời hình ảnh mà nó chỉ báo liệu bộ đệm hình ảnh được mã hóa (CPB) có tạo ra các thông số hỗ trợ hoạt động ở cấp độ hình ảnh phụ hay không dựa trên trị số của cờ hiệu định thời hình ảnh. Cờ hiệu định thời hình ảnh có thể được bao gồm trong thông báo SEI định thời hình ảnh.

Việc xác định liệu có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung có thể bao gồm việc cài đặt cờ hiệu trễ loại bỏ CPB của gói giải mã chung là 1 khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ được bao gồm trong thông báo SEI định thời hình ảnh. Điều này cũng có thể bao gồm việc cài đặt cờ hiệu trễ loại bỏ CPB của gói giải mã chung là 0 khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ không được bao gồm trong thông báo SEI định thời hình ảnh. Cờ hiệu trễ loại bỏ CPB của gói giải mã chung có thể được bao gồm trong thông báo SEI định thời hình ảnh.

Thiết bị điện tử cũng có thể tạo ra, khi CPB hỗ trợ hoạt động ở cấp độ hình ảnh phụ, các thông số liên quan đến các bộ của lớp mạng trừu tượng riêng rẽ mà nó chỉ báo số lượng, được dịch vị đi một, các gói NAL dùng cho mỗi gói giải mã trong gói truy nhập. Theo phương án khác, hoặc thêm vào đó, thiết bị điện tử có thể tạo ra thông số NAL

chung mà nó chỉ báo số lượng, được dịch vị đi một, các gói NAL thông thường đối với mỗi gói giải mã trong gói truy nhập.

Thiết bị điện tử dùng để đệm dòng bit cũng sẽ 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 truyền thông điện tử với bộ xử lý. Thiết bị điện tử này xác định rằng CPB truyền tín hiệu các thông số ở cấp độ hình ảnh phụ cho gói truy nhập. Thiết bị điện tử này cũng sẽ xác định, khi thông báo Thông tin nâng cao bổ sung (SEI) định thời hình ảnh nhận được bao gồm cờ hiệu trễ loại bỏ bộ đệm hình ảnh (CPB) được mã hóa của bộ mã hóa chung, thông số trễ loại bỏ CPB của gói giải mã chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập. Thiết bị điện tử này cũng sẽ xác định, khi thông báo SEI định thời hình ảnh không bao gồm cờ hiệu trễ loại bỏ CPB của gói giải mã chung, thông số trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập. Thiết bị điện tử này cũng loại bỏ các gói giải mã từ CPB bằng cách sử dụng thông số trễ loại bỏ CPB của gói giải mã chung hoặc các thông số trễ riêng rẽ loại bỏ CPB của gói giải mã. Thiết bị điện tử này cũng giải mã các gói giải mã trong gói truy nhập.

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

Biểu thức 1

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

trong đó du_cpb_removal_delay[i] là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số.

Theo phương án khác, thiết bị điện tử này có thể cài đặt thông số trễ loại bỏ CPB, cpb_removal_delay, và du_cpb_removal_delay[num_decoding_units_minus1] để thỏa mãn phương trình

Biểu thức 2

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

trong đó du_cpb_removal_delay[i] là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số.

Theo phương án khác, thiết bị điện tử có thể cài đặt thông số trễ loại bỏ 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 thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một.

Theo một cấu hình, thiết bị điện tử này xác định rằng cờ hiệu định thời hình ảnh được cài đặt trong thông báo SEI định thời hình ảnh. Thiết bị điện tử này cũng có thể cài đặt các thông số trễ loại bỏ CPB, cpb_removal_delay, và du_cpb_removal_delay[num_decoding_units_minus1] để thỏa mãn phương trình: -1<=(cpb_removal_delay*t_c - du_cpb_removal_delay[num_decoding_units_minus1]* $t_{c,sub}$) <=1 trong đó du_cpb_removal_delay[num_decoding_units_minus1] là thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một.

Biến số ClockDiff có để 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 lượng các đơn vị thời gian của đồng hồ hoạt động tại tần số time_scale Hz tương ứng với một lượng tăng của bộ đếm tích tắc đồng hồ, num_units_in_sub_tick là số lượng các đơn vị thời gian của đồng hồ hoạt động tại tần số time_scale Hz tương ứng với một lượng tăng của tích tắc đồng hồ của hình ảnh phụ

counter, num_decoding_units_minus1+1 là số lượng các gói giải mã trong gói truy nhập, và time_scale là số lượng các đơn vị thời gian đi qua trong một giây.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1, CPB hoạt động tại cấp độ hình ảnh phụ và ClockDiff lớn hơn zero, thời gian loại bỏ đối với gói 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 loại bỏ danh định của gói giải mã m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1, CPB hoạt động ở cấp độ gói truy nhập và ClockDiff lớn hơn zero, thời gian loại bỏ dùng cho gói truy nhập 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 loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, Ceil() là phương trình trần và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + max((t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c)))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) +$

max((t_{c_sub} * Ceil((t_{af}(m) - $t_{r,n}$(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - $t_{r,n}$(n)) / t_c))) trong đó $t_{r,n}$(m) là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, t_{af}(m) là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}$(n) là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và t_{af}(n) là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}$(m) < t_{af}(m), thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã sau cùng m của gói truy nhập, t_r(m) theo: t_r(m) = $t_{r,n}$(m) + min((t_{c_sub} * Ceil((t_{af}(m) - $t_{r,n}$(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - $t_{r,n}$(n)) / t_c))) trong đó $t_{r,n}$(m) là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, t_{af}(m) là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}$(n) là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và t_{af}(n) là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}$(n) < t_{af}(n), thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, t_r(n) theo: t_r(n) = $t_{r,n}$(n) + min((t_{c_sub} * Ceil((t_{af}(m) - $t_{r,n}$(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - $t_{r,n}$(n)) / t_c))) trong đó $t_{r,n}$(m) là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, t_{af}(m) là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}$(n) là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và t_{af}(n) là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}$(m) < t_{af}(m), thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã sau cùng m của gói truy nhập, t_r(m) theo: t_r(m) = $t_{r,n}$(m) + (t_c * Ceil((t_{af}(n) - $t_{r,n}$(n)) / t_c)) trong đó $t_{r,n}$(m) là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, t_{af}(m) là thời gian

tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập 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 loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ dùng cho gói giải mã m mà nó không phải là gói giải mã sau cùng được cài đặt tại $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ dùng cho gói giải mã m mà nó là gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n, và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ dùng cho gói giải mã m mà nó không phải là gói giải mã sau cùng được cài đặt tại $t_r(m) = t_{af}(m)$, trong đó

$t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ dùng cho gói giải mã m mà nó là gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n, và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ dùng cho gói giải mã m được cài đặt tại $t_r(m) = t_{af}(m)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n, và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{af}(n)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Ngoài ra trong một số trường hợp cờ hiệu có thể được gửi tromg một phần của dòng bit để đánh tính hiệu mà trong số các phương trình thay thế nêu trên được sử dụng để quyết định thời gian loại bỏ của các gói giải mã và thời gian loại bỏ của gói truy nhập.

Trong một trường hợp cờ hiệu này có thể được gọi là du_au_cpb_alignment_mode_flag. Nếu du_au_cpb_alignment_mode_flag là 1 thì các phương trình nói trên mà mó sắp đặt hoạt động của CPB, CPB này hoạt động ở chế độ dựa vào hình ảnh phụ với CPB mà nó hoạt động bằng chế độ của gói truy nhập được sử dụng. Nếu du_au_cpb_alignment_mode_flag là 0 thì các phương trình nói trên, các phương trình này không sắp đặt hoạt động của CPB mà nó hoạt động ở chế độ dựa vào hình ảnh phụ với CPB mà nó hoạt động bằng chế độ của gói truy nhập được sử dụng.

Trong một trường hợp cờ hiệu du_au_cpb_alignment_mode_flag có thể được truyền tín hiệu bằng thông tin khả dụng viđeo (VUI). Trong trường hợp khác cờ hiệu du_au_cpb_alignment_mode_flag có thể được gửi trong thông báo SEI định thời hình ảnh. Trong trường hợp khác nữa cờ hiệu du_au_cpb_alignment_mode_flag có thể được gửi trong phần quy chuẩn 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

```
pic_timing( payloadSize ) {
    if( CpbDpbDelaysPresentFlag ) {
        cpb_removal_delay
        dpb_output_delay
        if( sub_pic_cpb_params_present_flag ) {
            num_decoding_units_minus1
            du_au_cpb_alignment_mode_flag
            for( i = 0; i <= num_decoding_units_minus1; i++ ) {
                num_nalus_in_du_minus1[i]
                du_cpb_removal_delay[i]
            }
        }
    }
}
```

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 $t_r(n)$ của gói truy nhập n có thể được gọi là CpbRemovalTime(n), $t_r(m)$ của gói 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 gói

truy nhập m có thể được gọi là FinalArrivalTime(n) của gói truy nhập n, t_{af}(m) của gói giải mã m có thể được gọi là FinalArrivalTime(m), $t_{r,n}$(n) có thể được gọi là NominalRemovalTime(n) của gói truy nhập n, $t_{r,n}$(m) có thể được gọi là NominalRemovalTime(m) của gói giải mã m.

Phương pháp dùng để 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 được mã hóa (CPB) hỗ trợ hoạt động ở cấp độ hình ảnh phụ, liệu có bao gồm thông số trễ loại bỏ CPB của gói 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 thông số trễ loại bỏ CPB của gói giải mã chung sẽ được bao gồm trong thông báo SEI định thời hình ảnh, thông số trễ loại bỏ CPB của gói giải mã chung, trong đó thông số trễ loại bỏ CPB của gói giải mã chung được áp dụng cho tất cả các gói giải mã trong gói truy nhập từ CPB. Phương pháp này cũng bao gồm bước tạo ra, khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ không được bao gồm trong thông báo SEI định thời hình ảnh, thông số trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập. Phương pháp này cũng bao gồm bước gửi thông báo SEI định thời hình ảnh cùng với thông số trễ loại bỏ CPB của gói giải mã chung hoặc các thông số trễ loại bỏ CPB của gói giải mã.

Phương pháp dùng để 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 thông số ở cấp độ hình ảnh phụ dugnf cho gói truy nhập. 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 hiệu trễ loại bỏ bộ đệm hình ảnh (CPB) được mã hóa của bộ mã hóa chung, thông số trễ loại bỏ CPB của gói giải mã chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập. Phương pháp này cũng bao gồm bước xác định, khi thông báo SEI định thời hình ảnh không bao gồm cờ hiệu trễ loại bỏ CPB của gói giải mã chung, thông số trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập. Phương pháp này cũng bao gồm bước loại bỏ các gói giải mã từ CPB bằng cách sử dụng thông số trễ loại bỏ CPB của gói giải mã chung hoặc các thông số trễ riêng rẽ loại bỏ

CPB của gói giải mã. Phương pháp này còn bao gồm bước giải mã các gói giải mã trong gói truy nhập.

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 thông 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 bước truyền tín hiệu các thông số của bộ mã hóa tham chiếu giả thiết (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 đổi 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 bước truyền tín hiệu gọn nhẹ hơn đối với các thông số khi mỗi hình ảnh phụ đi tới và được loại bỏ từ CPB theo những quãng thời gian đều nhau.

Ngoài ra, khi các thông số trễ loại bỏ CPB ở cấp độ hình ảnh phụ hiện diện, Bộ đệm hình ảnh được mã hóa (CPB) có thể hoạt động ở cấp độ gói truy nhập hoặc cấp độ 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 cấp độ hình ảnh phụ và hoạt động của CPB của cấp độ gói truy nhập dẫn đến định thời giống nhau về việc loại bỏ gói giải mã. Cụ thể là định thời loại bỏ gói giải mã sau cùng trong gói truy nhập khi hoạt động ở chế độ hình ảnh phụ và định thời loại bỏ của gói truy nhập khi hoạt động ở chế độ của gói truy nhập sẽ là giống nhau.

Cần lưu ý rằng mặc dù thuật ngữ "giải 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 - HECV). 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 viđeo (VCL) và các gói NAL của dữ liệu đột. Dòng bit Loại II có thể chứa các gói NAL và các thành phần cú pháp bổ sung khác.

Tài liệu của Nhóm hợp tác về mã hóa viđeo (Joint Collaborative Team on Video Coding - JCTVC) JCTVC-I0333 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 viđeo hiệu suất cao (HEVC) (JCTVC-I1003), được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này. 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 được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này. 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 được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này.

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

```
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_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 (1)

Các ví dụ về ngữ nghĩa thông báo SEI khoảng thời gian đệm theo các hệ thống và các phương pháp được bộc lộ trong sáng chế được trình bày dưới đây. Cụ thể, các chi tiết

bổ sung về ngữ nghĩa của các thành phần cú pháp được cải biến được trình bày dưới đây. Khi NalHrdBpPresentFlag hoặc VclHrdBpPresentFlag bằng nhau là 1, thông báo SEI khoảng thời gian đệm có thể được liên kết với gói truy nhập bất kỳ trong dòng bit, và thông báo SEI khoảng thời gian đệm có thể được liên kết với từng gói truy nhập IDR, với từng gói truy nhập CRA, và với từng gói truy nhập được liên kết với thông báo SEI điểm phục hồi. Đối với một số ứng dụng, sự hiện diện thường xuyên của thông báo SEI khoảng thời gian đệm có thể được mong muốn. Khoảng thời gian đệm được xác định là tập hợp của các gói truy nhập giữa hai thời điểm của thông báo SEI khoảng thời gian đệm trong thứ tự mã hóa.

'seq_parameter_set_id' xác định tập hợp thông số chuỗi có chứa các đặc tính HDR dãy. Trị số seq_parameter_set_id có thể bằng trị số seq_parameter_set_id trong tập hợp thông số hình ảnh được tham chiếu bởi hình ảnh được mã hóa sơ cấp được liên kết với thông báo SEI khoảng thời gian đệm. Trị số seq_parameter_set_id có thể nằm trong khoảng từ 0 đến 31, bao gồm cả trị số 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 được mã hóa được liên kết với gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm và thời gian loại bỏ từ CPB của dữ liệu được mã hóa được liên kết với cùng một gói truy nhập, đối với khoảng thời gian đệm thứ nhất sau khi khởi động HRD. Thành phần cú pháp có các độ dài theo bít được xác định bằng initial_cpb_removal_delay_length_minus1 + 1. Nó được tính theo đơn vị xung nhịp 90 kHz. initial_cpb_removal_delay[SchedSelIdx] có thể không bằng 0 và có thể không vượt quá 90000 * (CpbSize[SchedSelIdx] / BitRate[SchedSelIdx]), đương lượng thời gian của kích thước CPB được tính theo các đơn vị thời gian 90 kHz.

'initial_cpb_removal_delay_offset'[SchedSelIdx] được dùng cho SchedSelIdx-th CPB phối hợp với cpb_removal_delay để xác định thời gian cung cấp ban đầu của các gói truy nhập được mã hóa tới CPB. initial_cpb_removal_delay_offset[SchedSelIdx] là theo đơn vị xung nhịp 90 kHz. Thành phần cú pháp initial_cpb_removal_delay_offset[

SchedSelIdx] là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng initial_cpb_removal_delay_length_minus1 + 1. Thành phần cú pháp này không được sử dụng bởi các bộ giải mã và 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 dãy viđeo được mã hóa, tổng của initial_cpb_removal_delay[SchedSelIdx] và initial_cpb_removal_delay_offset[SchedSelIdx] có thể không đổi đối với mỗi trị số của SchedSelIdx.

'initial_du_cpb_removal_delay'[SchedSelIdx] xác định khoảng trễ cho SchedSelIdx-th CPB giữa thời gian đi đến CPB của bit thứ nhất của dữ liệu được mã hóa được liên kết với gói giải mã thứ nhất trong gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm và thời gian loại bỏ từ CPB của dữ liệu được mã hóa được liên kết với cùng một gói giải mã, đối với khoảng thời gian đệm thứ nhất sau khi khởi động HRD. Thành phần cú pháp có các độ dài theo bít được xác định bằng initial_cpb_removal_delay_length_minus1 + 1. Nó được tính theo đơn vị xung nhịp 90 kHz. initial_du_cpb_removal_delay[SchedSelIdx] có thể không bằng 0 và có thể không vượt quá 90000 * (CpbSize[SchedSelIdx] / BitRate[SchedSelIdx]), đương lượng thời gian của kích thước CPB được tính theo các đơn vị thời gian 90 kHz.

'initial_du_cpb_removal_delay_offset'[SchedSelIdx] được dùng cho SchedSelIdx-th CPB phối hợp với cpb_removal_delay để xác định thời gian cung cấp ban đầu của các gói giải mã tới CPB.

initial_cpb_removal_delay_offset[SchedSelIdx] được tính theo xung nhịp 90 kHz. Thành phần cú pháp initial_du_cpb_removal_delay_offset[SchedSelIdx] là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng initial_cpb_removal_delay_length_minus1 + 1. Thành phần cú pháp này không được sử dụng bởi các bộ giải mã và 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 dãy video được 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 trị số 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 trình bày dưới đây. Cụ thể, các chi tiết bổ sung về ngữ nghĩa của các thành phần cú pháp được cải biến sẽ được trình bày dưới đây.

Cú pháp của thông báo SEI định thời hình ảnh phụ thuộc vào nội dung của tập hợp thông số chuỗi mà nó có tính kích hoạt đối với hình ảnh được mã hóa được liên kết với thông báo SEI định thời hình ảnh. Tuy nhiên, trừ phi thông báo SEI định thời hình ảnh của gói truy nhập làm tươi giải mã tức thời (IDR) được đề cập trước bởi thông báo SEI khoảng thời gian đệm bên trong cùng một gói truy nhập, việc kích hoạt tập hợp thông 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 được mã hóa đó là hình ảnh IDR) không xuất hiện cho tới khi mã hóa gói của lớp mạng trừu tượng (NAL) của phiến được mã hóa của hình ảnh được mã hóa đó. Do gói NAL của phiến được mã hóa của hình ảnh được mã hóa sẽ theo sau thông báo SEI định thời hình ảnh trong thứ tự gói NAL, có thể có các trường hợp trong đó cần thiết để cho bộ giải mã lưu trữ chuỗi byte thô (RBSP) có chứa thông báo SEI định thời hình ảnh cho tới khi xác định các thông số của thông số chuỗi có tính kích hoạt đối với hình ảnh được mã hóa, và và sau đó thực hiện việc phân tách thông báo SEI định thời hình ảnh.

Sự hiện diện của thông báo SEI định thời hình ảnh trong dòng bit được xác định dưới đây. Nếu CpbDpbDelaysPresentFlag bằng 1, một thông báo SEI định thời hình ảnh có thể hiện diện trong từng gói truy nhập của dãy video được mã hóa. Khác đi thì (CpbDpbDelaysPresentFlag bằng 0), không có các thông báo SEI định thời hình ảnh có thể hiện diện trong bất kỳ gói truy nhập nào của dãy video được mã hóa.

'cpb_removal_delay' xác định có bao nhiêu tích tắc đồng hồ (xem khoản E.2.1 của JCTVC-I1003) phải chờ sau khi loại bỏ từ CPB gói truy nhập được liên kết với thông báo

SEI khoảng thời gian đệm gần đây nhất trong gói truy nhập trước đó trước khi loại bỏ từ bộ đệm dữ liệu gói truy nhập được liên kết với thông báo SEI định thời hình ảnh. Trị số này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu gói truy nhập vào CPB đối với HSS, như được xác định trong Phụ lục C của JCTVC-I1003. Thành phần cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng cpb_removal_delay_length_minus1 + 1. cpb_removal_delay là phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun.

Trị số cpb_removal_delay_length_minus1 mà nó xác định độ dài (bằng bit) của thành phần cú pháp cpb_removal_delay là trị số cpb_removal_delay_length_minus1 được mã hóa trong tập hợp thông số chuỗi mà nó có tính kích hoạt đối với hình ảnh được mã hóa sơ cấp được liên kết với thông báo SEI định thời hình ảnh, mặc dù cpb_removal_delay xác định số lượng tích tắc đồng hồ liên quan đến thời gian loại bỏ của gói truy nhập trước đó có chứa thông báo SEI khoảng thời gian đệm, mà nó có thể là gói truy nhập của dãy video được mã hóa khác.

'dpb_output_delay' được sử dụng để tính toán thời gian kết xuất của bộ đệm hình ảnh được mã hóa (DPB) của hình ảnh. Nó xác định có bao nhiêu tích tắc đồng hồ phải chờ sau khi loại bỏ gói giải mã sau cùng trong gói truy nhập từ CPB trước khi hình ảnh được mã hóa được kết xuất từ DPB (xem khoản C.2 của JCTVC-I1003).

Liên quan đến DPB, hình ảnh không được loại bỏ từ DPB tại thời gian kết xuất của hình ảnh khi hình 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 hình ảnh được mã hóa. Độ dài của thành phần cú pháp dpb_output_delay được tính theo bit bằng dpb_output_delay_length_minus1 + 1. Khi max_dec_pic_buffering[max_temporal_layers_minus1] bằng 0, 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 mã hóa phù hợp định thời kết xuất như được xác định trong khoản 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 dãy viđeo được mã hóa tiếp theo trong thứ tự mã hóa. Thứ tự

kết xuất hình ảnh được thiết lập bởi các trị số của thành phần cú pháp này có thể giống với thứ tự như được thiết lập bởi các trị số của 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, trong thứ tự mã hóa, hình ảnh IDR cùng với no_output_of_prior_pics_flag bằng 1 hoặc được kết luận bằng 1, thời gian kết xuất nhận được từ từ dpb_output_delay có thể tăng lên với việc tăng trị số của PicOrderCnt( ) liên quan đến tất cả các hình ảnh trong cùng dãy video được mã hóa.

'num_decoding_units_minus1' cộng 1 xác định số lượng của các gói giải mã trong gói truy nhập mà thông báo SEI định thời hình ảnh được liên kết. Trị số num_decoding_units_minus1 có thể nằm trong khoảng từ 0 đến PicWidthInCtbs * PicHeightInCtbs - 1, bao gồm cả trị số cuối.

'common_du_cpb_removal_delay_flag' bằng 1 xác định rằng thành phần cú pháp common_du_cpb_removal_delay có hiện diện.

'common_du_cpb_removal_delay_flag' bằng 0 xác định rằng thành phần cú pháp common_du_cpb_removal_delay không hiện diện.

'common_du_cpb_removal_delay' xác định thông tin như sau: Nếu gói giải mã là gói giải mã thứ nhất trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh thì common_du_cpb_removal_delay xác định có bao nhiêu tích tắc đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) phải chờ sau khi loại bỏ từ CPB gói giải mã sau cùng trong gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm gần đây nhất trong gói truy nhập trước đó trước khi loại bỏ từ CPB gói giải mã thứ nhất trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh.

Khác đi common_du_cpb_removal_delay xác định có bao nhiêu tích tắc đồng hồ của hình ảnh phụs (xem khoản E.2.1 của JCTVC-I1003) phải chờ sau khi loại bỏ từ CPB gói giải mã trước đó trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh trước khi loại bỏ từ CPB gói giải mã hiện thời trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh. Trị số này còn được sử dụng để tính toán thời gian đi

đến sớm nhất có thể của dữ liệu gói giải mã vào CPB đối với HSS, như được xác định trong Phụ lục C. Thành phần cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng cpb_removal_delay_length_minus1 + 1. common_du_cpb_removal_delay là phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun.

Một cách thay thế để xác định 'common_du_cpb_removal_delay' như sau.

common_du_cpb_removal_delay xác định có bao nhiêu tích tắc đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) để chờ sau khi loại bỏ từ CPB gói giải mã sau cùng trước khi loại bỏ từ CPB gói giải mã hiện thời trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh. Trị số này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu gói giải mã vào CPB đối với HSS, như được xác định trong Phụ lục C. Thành phần cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng cpb_removal_delay_length_minus1 + 1. common_du_cpb_removal_delay là phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun.

Trị số cpb_removal_delay_length_minus1 mà nó xác định độ dài (bằng bit) của thành phần cú pháp common_du_cpb_removal_delay là trị số cpb_removal_delay_length_minus1 được mã hóa trong tập hợp thông số chuỗi mà nó có tính kích hoạt đối với hình ảnh được mã hóa được liên kết với thông báo SEI định thời hình ảnh, mặc dù common_du_cpb_removal_delay xác định số lượng tích tắc đồng hồ của các hình ảnh phụ liên quan đến thời gian loại bỏ của gói giải mã thứ nhất trong gói truy nhập trước đó có chứa thông báo SEI khoảng thời gian đệm, mà nó có thể là gói truy nhập của dãy viđeo được mã hóa khác.

'num_nalus_in_du_minus1[i]' cộng 1 xác định số lượng các gói NAL trong gói giải mã thứ i của gói truy nhập mà thông báo SEI định thời hình ảnh được liên kết. Trị số num_nalus_in_du_minus1[i] có thể nằm trong khoảng từ 0 đến PicWidthInCtbs * PicHeightInCtbs - 1, bao gồm cả trị số cuối.

Gói giải mã thứ nhất của gói truy nhập gồm num_nalus_in_du_minus1[0] + 1 gói NAL liên tiếp thứ nhất trong thứ tự mã hóa trong gói truy nhập. Gói mã hóa thứ i (với i lớn hơn 0) của gói truy nhập gồm num_nalus_in_du_minus1[i] + 1 gói NAL liên tiếp đầu tiên ngay sau gói NAL cuối cùng trong gói mã hóa trước đó của gói truy nhập, trong thứ tự mã hóa. Có thể có ít nhất một gói NAL VCL trong mỗi gói giải mã. Tất cả các gói NAL không VCL được liên kết với gói NAL VCL có thể được bao gồm trong cùng một gói giải mã.

'du_cpb_removal_delay[i]' xác định có bao nhiêu tích tắc đồng hồ của các hình ảnh phụ (xem khoản E.2.1 của JCTVC-I1003) phải chờ sau khi loại bỏ từ CPB gói giải mã thứ nhất trong gói truy nhập được liên kết với thông báo SEI khoảng thời gian đệm gần đây nhất trong gói truy nhập trước đó trước khi loại bỏ từ CPB gói giải mã thứ i trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh. Trị số này còn được sử dụng để tính toán thời gian đi đến sớm nhất có thể của dữ liệu gói giải mã vào CPB đối với HSS (ví dụ, như được xác định trong Phụ lục C của JCTVC-I1003). Thành phần cú pháp là mã có độ dài cố định mà độ dài của nó theo bit được xác định bằng cpb_removal_delay_length_minus1 + 1. du_cpb_removal_delay[i] là phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun.

Trị số cpb_removal_delay_length_minus1 mà nó xác định độ dài (bằng bit) của thành phần cú pháp du_cpb_removal_delay[i] là trị số cpb_removal_delay_length_minus1 được mã hóa trong tập hợp thông số chuỗi mà nó có tính kích hoạt đối với hình ảnh được mã hóa được liên kết với thông báo SEI định thời hình ảnh, mặc dù du_cpb_removal_delay[i] xác định số lượng tích tắc đồng hồ của các hình ảnh phụ liên quan đến thời gian loại bỏ của gói giải mã thứ nhất trong gói truy nhập trước đó có chứa thông báo SEI khoảng thời gian đệm, mà nó có thể là gói truy nhập của dãy viđeo được mã hóa khác.

Theo một cấu hình, định thời loại bỏ gói giải mã và mã hóa các gói giải mã có thể được thực hiện như sau.

Nếu SubPicCpbFlag bằng 0, biến CpbRemovalDelay(m) được cài đặt bằng trị số cpb_removal_delay trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập nghĩa là gói giả mã m, và biến T_c được cài đặt bằng t_c. Khác đi nếu SubPicCpbFlag bằng 1 và common_du_cpb_removal_delay_flag bằng 0 thì biến CpbRemovalDelay(m) được cài đặt bằng trị số du_cpb_removal_delay[i] đối với gói giải mã m (với m nằm trong khoảng từ 0 đến num_decoding_units_minus1) trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập có chứa gói giả mã m, và biến T_c được cài đặt bằng t_{c_sub}.

Trong một số trường hợp, khác đi nếu SubPicCpbFlag bằng 1 và common_du_cpb_removal_delay_flag bằng 0 thì biến CpbRemovalDelay(m) được cài đặt bằng trị số (m+1)*du_cpb_removal_delay[i] đối với gói giải mã m (với m nằm trong khoảng từ 0 đến num_decoding_units_minus1) trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập có chứa gói giả mã m, và biến T_c được cài đặt bằng t_{c_sub}.

Khác đi nếu SubPicCpbFlag bằng 1 và common_du_cpb_removal_delay_flag bằng 1 thì biến CpbRemovalDelay(m) được cài đặt bằng trị số common_du_cpb_removal_delay đối với gói giải mã m trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập có chứa gói giả mã m, và biến T_c được cài đặt bằng t_{c_sub}.

Khi gói giải mã m là gói giải mã với n bằng 0 (gói giải mã thứ nhất của gói truy nhập mà nó khởi động HRD), thời gian loại bỏ danh định của gói giải mã từ CPB được xác định bằng $t_{r,n}(0)$ = InitCpbRemovalDelay[SchedSelIdx] / 90000.

Khi gói giải mã m là gói giải mã thứ nhất của gói truy nhập thứ nhất của khoảng thời gian đệm mà nó không khởi động HRD, thời gian loại bỏ danh định của gói giải mã từ CPB được xác định bằng $t_{r,n}(m) = t_{r,n}(m_b) + T_c$ * CpbRemovalDelay(m), trong đó $t_{r,n}(m_b)$ là thời gian loại bỏ danh định của gói giải mã thứ nhất của khoảng thời gian đệm trước đó.

Khi gói giải mã m là gói giải mã thứ nhất của khoảng thời gian đệm, m_b được cài đặt bằng m tại thời gian loại bỏ $t_{r,n}(m)$ của gói giải mã m. Thời gian loại bỏ danh định $t_{r,n}(m)$ của gói giải mã m mà nó không phải là gói giải mã thứ nhất của khoảng thời gian đệm được xác định bằng $t_{r,n}(m) = t_{r,n}(m_b) + T_c$ * CpbRemovalDelay(m), trong đó $t_{r,n}(m_b)$ là thời gian loại bỏ danh định của gói giải mã thứ nhất của khoảng thời gian đệm hiện thời.

Thời gian loại bỏ của gói giải mã m được xác định như sau. Nếu low_delay_hrd_flag bằng 0 hoặc $t_{r,n}(m) >= t_{af}(m)$, thời gian loại bỏ của gói giải mã m được xác định bằng $t_r(m) = t_{r,n}(m)$. Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian loại bỏ của gói giải mã m được xác định bằng $t_r(m) = t_{r,n}(m) + T_c$ * 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 gói giải mã m, b(m), lớn tới mức nó ngăn cản việc loại bỏ tại thời gian loại bỏ danh định.

Trong trường hợp khác thời gian loại bỏ của gói giải mã m được xác định như sau. If low_delay_hrd_flag bằng 0 hoặc $t_{r,n}(m) >= t_{af}(m)$, thời gian loại bỏ của gói giải mã m được xác định by $t_r(m) = t_{r,n}(m)$. Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thời gian loại bỏ của gói giải mã m mà nó không phải là gói giải mã sau cùng trong gói truy nhập được xác định by $t_r(m) = t_{af}(m)$, và thời gian loại bỏ của gói giải mã m mà nó là gói giải mã sau cùng trong gói truy nhập $t_r(m) = t_{r,n}(m) + T_c$ * 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 gói giải mã m, b(m), lớn tới mức nó ngăn cản việc loại bỏ tại thời gian loại bỏ danh định.

Trong trường hợp khác thời gian loại bỏ của gói giải mã m được xác định như sau. Nếu low_delay_hrd_flag bằng 0 hoặc $t_{r,n}(m) >= t_{af}(m)$, thời gian loại bỏ của gói giải mã m được xác định bằng $t_r(m) = t_{r,n}(m)$. Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian loại bỏ của gói giải mã m mà nó không phải là gói giải mã sau cùng trong gói truy nhập được xác định bằng $t_r(m) = t_{af}(m)$, và thời gian loại bỏ của gói giải mã m mà nó là gói giải mã sau cùng trong gói truy nhập $t_r(m) = t_{r,n}(m) + t_c$ *

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 gói giải mã m, b(m), lớn tới mức nó ngăn cản việc loại bỏ tại thời gian loại bỏ danh định.

Trong trường hợp khác thời gian loại bỏ của gói giải mã m được xác định như sau. Nếu low_delay_hrd_flag bằng 0 hoặc $t_{r,n}$(m) >= t_{af}(m), thì thời gian loại bỏ của gói giải mã m được xác định bằng t_r(m) = $t_{r,n}$(m). Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}$(m) < t_{af}(m)), thời gian loại bỏ của gói giải mã m được xác định bằng t_r(m) = t_{af}(m). Trường hợp sau (low_delay_hrd_flag bằng 1 và $t_{r,n}$(m) < t_{af}(m)) chỉ báo rằng kích thước của gói giải mã m, b(m), lớn tới mức nó ngăn cản việc loại bỏ tại thời gian loại bỏ danh định.

Khi SubPicCpbFlag bằng 1, thì thời gian loại bỏ CPB danh định của gói truy nhập n $t_{r,n}$(n) được cài đặt bằng thời gian loại bỏ CPB danh định của gói giải mã sau cùng trong gói truy nhập n, thời gian loại bỏ CPB của gói truy nhập n t_r(n) được cài đặt bằng thời gian loại bỏ CPB của gói giải mã sau cùng trong gói truy nhập n.

Khi SubPicCpbFlag bằng 0, thì mỗi gói giả mã là gói truy nhập, do vậy thời gian loại bỏ CPB danh định và thời gian loại bỏ CPB của gói truy nhập n là thời gian loại bỏ CPB danh định và thời gian loại bỏ CPB của gói giải mã n.

Tại thời gian loại bỏ CPB của gói giải mã m, gói giải mã sẽ được mã hóa tức thời.

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 nét 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 thành phần cú pháp common_num_nalus_in_du_minus1, thành phần này nó có thể được sử dụng để xác định ba0 nhiêu dữ liệu cần được loại bỏ từ CPB khi loại bỏ gói giải mã. 'common_num_nalus_in_du_minus1' cộng 1 xác định số lượng các gói NAL trong mỗi gói giải mã của gói truy nhập mà thông báo SEI định thời hình ảnh được liên kết. Trị số common_num_nalus_in_du_minus1 có thể nằm trong khoảng từ 0 đến PicWidthInCtbs * PicHeightInCtbs - 1, bao gồm cả trị số cuối.

Gói giải mã thứ nhất của gói truy nhập gồm common_num_nalus_in_du_minus1 + 1 bộ NAL liên tiếp thứ nhất trong thứ tự mã hóa trong gói truy nhập. Gói mã hóa thứ i (với i lớn hơn 0) của gói truy nhập gồm common_num_nalus_in_du_minus1 + 1 bộ NAL liên tiếp thứ nhất ngay sau gói NAL cuối cùng trong gói mã hóa trước đó của gói truy nhập, trong thứ tự mã hóa. Có thể có ít nhất một gói NAL VCL trong mỗi gói giải mã. Tất cả các gói NAL không VCL được liên kết với gói NAL VCL có thể được bao gồm trong cùng một gói 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

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

Bảng (3)

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

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

Thêm vào đó các cải biến đối với các thành phần 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 hạn chế dòng bit sao cho hoạt động CPB dựa trên hình ảnh phụ và hoạt động CPB cấp độ gói truy nhập dẫn đến định thời giống nhau về việc loại bỏ gói giải mã.

Khi sub_pic_cpb_params_present_flag bằng1 theo đó các thông số trễ loại bỏ CPB cấp độ hình ảnh phụ hiện diện thì CPB có thể hoạt động ở cấp độ gói truy nhập hoặc cấp độ hình ảnh phụ. sub_pic_cpb_params_present_flag bằng 0 thì xác định rằng các thông số trễ loại bỏ CPB cấp độ hình ảnh phụ không hiện diện và CPB hoạt động ở cấp độ gói truy nhập. Khi sub_pic_cpb_params_present_flag không hiện diện, trị số của nó được kết luận bằng 0.

Để trợ giúp hoạt động tại cả cấp gói truy nhập hoặc cấp độ hình ảnh phụ, các hạn chế dòng bit sau đây có thể được sử dụng: Nếu sub_pic_cpb_params_present_flag bằng 1 thì cần phải có sự phù hợp dòng bit theo đó các hạn chế sau phải được hoàn thành khi truyền tín hiệu các trị số cho cpb_removal_delay và du_cpb_removal_delay[i] đối với mọi i:

Biểu thức 3

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

trong đó du_cpb_removal_delay[i] là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số. Theo một số phương án một thông số cho phép có thể được bổ sung để thỏa mãn hạn chế nêu trên.

Để trợ giúp hoạt động tại cả cấp gói truy nhập hoặc cấp độ hình ảnh phụ, các hạn chế dòng bit dưới đây có thể được sử dụng: biến $T_{du}(k)$ được xác định như sau:

Biểu thức 4

$$T_{du}(k) = T_{du}(k-1) + t_{c_sub} * \sum_{i=0}^{num_decoding_units_minus1_k} (du_cpb_removal_delay_minus1_k[i]+1)$$

trong đó du_cpb_removal_delay_minus1$_k$[i] và num_decoding_units_minus1$_k$ là các thông số cho gói giải mã thứ I của gói truy nhập thứ k (với k=0 cho gói truy nhập 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 thông số trễ loại bỏ CPB của gói giải mã cho gói giải mã thứ I của gói truy nhập thứ k, và num_decoding_units_minus1$_k$ là số lượng của các gói giải mã trong gói truy nhập thứ k, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, và i và k là các chỉ số. Sau đó khi cờ hiệu định thời hình ảnh (ví dụ, sub_pic_cpb_params_present_flag) được cài đặt là 1, thì hạn chế sau phải 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 loại bỏ CPB. Do đó trong trường hợp này khoảng trễ loại bỏ CPB (au_cpb_removal_delay_minus1 + 1) được cài đặt sao cho việc vận hành hoạt động CPB dựa trên hình ảnh phụ và hoạt động CPB dựa trên gói truy nhập dẫn đến định thời giống nhau của việc loại bỏ gói truy nhập và gói giải mã cuối cùng của việc loại bỏ gói truy nhập.

Để trợ giúp hoạt động tại cả cấp gói truy nhập hoặc cấp độ hình ảnh phụ, các hạn chế dòng bit sau đây có thể được sử dụng: Nếu sub_pic_cpb_params_present_flag bằng 1 thì cần phải có sự phù hợp dòng bit theo đó các hạn chế sau phải được hoàn thành khi truyền tín hiệu các trị số cho cpb_removal_delay và du_cpb_removal_delay[i] đối với mọi i:

Biểu thức 5

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

trong đó du_cpb_removal_delay[i] là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số.

Để trợ giúp hoạt động tại cả cấp gói truy nhập hoặc cấp độ hình ảnh phụ, các hạn chế dòng bit sau đây có thể được sử dụng: Nếu sub_pic_cpb_params_present_flag bằng 1

thì cần phải có sự phù hợp dòng bit theo đó các hạn chế sau phải được hoàn thành khi truyền tín hiệu các trị số cho cpb_removal_delay và du_cpb_removal_delay[num_decoding_units_minus1]: 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 thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một. Theo một số phương án một thông số cho phép có thể được bổ sung để thỏa mãn hạn chế nêu trên.

Để trợ giúp hoạt động tại cả cấp gói truy nhập hoặc cấp độ hình ảnh phụ, các hạn chế dòng bit sau đây có thể được sử dụng: Nếu sub_pic_cpb_params_present_flag bằng 1 thì cần phải có sự phù hợp dòng bit theo đó các hạn chế sau phải được hoàn thành khi truyền tín hiệu các trị số cho cpb_removal_delay và du_cpb_removal_delay[i] đối với mọi i: -1<=(cpb_removal_delay*t_c - du_cpb_removal_delay[num_decoding_units_minus1]* $t_{c,sub}$) <=1 trong đó du_cpb_removal_delay[num_decoding_units_minus1] là thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một.

Ngoài ra, các hệ thống và các phương pháp hiện thời có thể cải biến định thời về việc loại bỏ gói giải mã. Khi các thông số trễ loại bỏ CPB cấp độ hình ảnh phụ hiện diện, thời gian loại bỏ của gói 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 chênh lệch có thể xuất hiện do bộ đếm tích tắc đồng hồ và bộ đếm tích tắc đồng hồ của hình ảnh phụ.

Khi sub_pic_cpb_params_present_flag bằng1 thì các thông số trễ loại bỏ CPB cấp độ hình ảnh phụ hiện diện và CPB có thể hoạt động ở cấp độ gói truy nhập hoặc cấp độ hình ảnh phụ. sub_pic_cpb_params_present_flag bằng 0 xác định rằng các thông số trễ loại bỏ CPB cấp độ hình ảnh phụ không hiện diện và CPB hoạt động ở cấp độ gói truy

nhập. Khi sub_pic_cpb_params_present_flag không hiện diện, trị số của nó được kết luận bằng 0.

Cụ thể là, một ví dụ về định thời về việc loại bỏ gói giải mã và giải mã đối với việc thực hiện gói giải mã được trình bày sau đây. Biến SubPicCpbPreferredFlag hoặc được xác định bằng các công cụ bên ngoài, hoặc khi không được xác định bằng các công cụ bên ngoài, thì được cài đặt bằng 0. Biến SubPicCpbFlag được rút ra như sau: SubPicCpbFlag = SubPicCpbPreferredFlag && sub_pic_cpb_params_present_flag. Nếu SubPicCpbFlag bằng 0, thì CPB hoạt động ở cấp độ gói truy nhập và mỗi gói giả mã là gói truy nhập. Khác đi CPB hoạt động tại cấp độ hình ảnh phụ và mỗi gói giả mã là tập hợp phụ của gói truy nhập.

Nếu SubPicCpbFlag bằng 0, biến CpbRemovalDelay(m) được cài đặt bằng trị số cpb_removal_delay trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập mà nó là gói giải mã m, và biến T_c được cài đặt bằng t_c. Khác đi biến CpbRemovalDelay(m) được cài đặt bằng trị số du_cpb_removal_delay[i] đối với gói giải mã m trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập có chứa gói giải mã m, và biến T_c được cài đặt bằng t_{c_sub}.

Khi gói giải mã m là gói giải mã với n bằng 0 (gói giải mã thứ nhất của gói truy nhập mà nó khởi động HRD), thời gian loại bỏ danh định của gói giải mã từ CPB được xác định by $t_{r,n}(0)$ = InitCpbRemovalDelay[SchedSelIdx] / 90000.

Khi gói giải mã m là gói giải mã thứ nhất của gói truy nhập thứ nhất của khoảng thời gian đệm mà nó không khởi động HRD, thời gian loại bỏ danh định của gói giải mã từ CPB được xác định by $t_{r,n}(m) = t_{r,n}(m_b) + T_c$ * CpbRemovalDelay(m) trong đó $t_{r,n}(m_b)$ là thời gian loại bỏ danh định của gói giải mã thứ nhất của khoảng thời gian đệm trước đó.

Khi gói giải mã m là gói giải mã thứ nhất của khoảng thời gian đệm, m_b được cài đặt bằng m tại thời gian loại bỏ $t_{r,n}(m)$ của gói giải mã m.

Thời gian loại bỏ danh định $t_{r,n}(m)$ của gói giải mã m mà nó không phải là gói giải mã thứ nhất của khoảng thời gian đệm được xác định bằng $t_{r,n}(m) = t_{r,n}(m_b) + T_c *$ CpbRemovalDelay(m) trong đó $t_{r,n}(m_b)$ là thời gian loại bỏ danh định của gói giải mã thứ nhất của khoảng thời gian đệm hiện thời.

Thời gian loại bỏ của gói giải mã m được xác định dưới đây. Biến ClockDiff được xác định là ClockDiff = (num_units_in_tick-(num_units_in_sub_tick* (num_decoding_units_minus1+1)) / time_scale). Trong một số trường hợp có thể cần có sự phù hợp dòng bit nghĩa là các thông số num_units_in_tick, num_units_in_sub_tick, num_decoding_units_minus1 được truyền tín hiệu sao cho thỏa mãn được phương trình sau đây. (num_units_in_tick-(num_units_in_sub_tick*(num_decoding_units_minus1+1))) >=0

Trong trường hợp khác nào đó có thể cần có sự phù hợp dòng bit nghĩa là các thông 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 phương trình dưới đây. (num_units_in_tick-(num_units_in_sub_tick*(num_decoding_units_minus1+1))) <=0 If low_delay_hrd_flag bằng 0 hoặc $t_{r,n}(m) >= t_{af}(m)$, thời gian loại bỏ của gói giải mã m được xác định bằng $t_r(m) = t_{r,n}(m)$.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), và khi sub_pic_cpb_params_present_flag bằng1 và CPB hoạt động tại cấp độ hình ảnh phụ, và nếu ClockDiff lớn hơn zero thì thời gian loại bỏ của gói giải mã m khi đó là gói giải mã sau cùng của gói truy nhập n được xác định bằng $t_r(m) = t_{r,n}(m) + T_c *$ Ceil(($t_{af}(m)$ - $t_{r,n}(m)$) / T_c) + ClockDiff.

Khác đi (low_delay_hrd_flag bằng 1 và tr,n(m) < taf(m)), và khi sub_pic_cpb_params_present_flag bằng1 và CPB hoạt động ở cấp độ gói truy nhập và nếu ClockDiff nhỏ hơn zero thì thời gian loại bỏ của gói truy nhập n được xác định bằng $t_r(m) = t_{r,n}(m) + t_c *$ Ceil(($t_{af}(m)$ - $t_{r,n}(m)$) / t_c) - ClockDiff.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$), thì thời gian loại bỏ của gói giải mã m được xác định bằng $t_r(m) = t_{r,n}(m) + T_c * Ceil((t_{af}(m) - t_{r,n}(m)) / T_c)$. Trường hợp sau (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) chỉ báo rằng kích thước của gói giải mã m, b(m), lớn tới mức nó ngăn cản việc loại bỏ tại thời gian loại bỏ danh định.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và khi cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + min((t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c)))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và khi cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + min((t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c)))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh

định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã mà nó không phải là gói giải mã sau cùng của gói truy nhập được cài đặt as $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. và thời gian loại bỏ của gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ được cài đặt theo: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m trong gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã mà nó không phải là gói giải mã sau cùng của gói truy nhập được cài đặt là $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. và thời gian loại bỏ của gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ được cài 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là

thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m trong gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(m) < t_{af}(m)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ đối với gói giải mã được cài đặt as $t_r(m) = t_{af}(m)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m trong gói truy nhập n.

Khác đi (low_delay_hrd_flag bằng 1 và $t_{r,n}(n) < t_{af}(n)$) và cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{af}(n)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi SubPicCpbFlag bằng 1, thì thời gian loại bỏ CPB danh định của gói truy nhập n $t_{r,n}(n)$ được cài đặt bằng thời gian loại bỏ CPB danh định của gói giải mã sau cùng trong gói truy nhập n, thời gian loại bỏ CPB của gói truy nhập n $t_r(n)$ được cài đặt bằng thời gian loại bỏ CPB của gói giải mã sau cùng trong gói truy nhập n.

Khi SubPicCpbFlag bằng 0, mỗi gói giả mã là gói truy nhập, do vậy thời gian loại bỏ CPB danh định và thời gian loại bỏ CPB của gói truy nhập n là thời gian loại bỏ CPB danh định và thời gian loại bỏ CPB của gói giải mã n. Tại thời gian loại bỏ CPB của gói giải mã m, gói giải mã sẽ được mã hóa tức thời.

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 thông 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 chuẩn HEVC.

Để tiện lợi, một số định nghĩa được trình bày dưới đây, mà nó 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 dòng dữ liệu (ví dụ, dòng bit) trong đó việc mã hóa dòng bit không đòi hỏi phải truy nhập bất kỳ điểm nào trong dòng bit đứng trước điểm truy nhập ngẫu nhiên để mã hóa hình ảnh hiện thời và tất cả các hình ảnh tiếp theo sau hình ảnh hiện thời trong thứ tự kết xuất.

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

Thông báo SEI khoảng thời gian đệm có thể là thông báo SEI liên quan đến khoảng thời gian đệm. Thông báo SEI định thời hình ảnh có thể là thông báo SEI liên quan đến thời điểm loại bỏ 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 loại bỏ hình ảnh được mã hóa.

Bộ đệm hình ảnh được mã hóa (CPB) có thể là bộ đệm vào trước ra trước có chứa các gói truy nhập trong thứ tự mã hóa được xác định trong bộ mã hóa tham chiếu giả thiết

(HRD). Gói truy nhập có thể là tập hợp các gói của Lớp truy nhập mạng (NAL), các gói này là liên tiếp trong thứ tự mã hóa và chứa chính xác một hình ảnh được mã hóa. Bên cạnh gói NAL của các phiến được mã hóa của hình ảnh được mã hóa, gói truy nhập cũng có thể chứa các gói NAL khác không chứa các phiến của hình ảnh được mã hóa. Việc mã hóa gói truy nhập luôn dẫn đến kết quả là hình ảnh được mã hóa. Gói NAL có thể là kết cấu 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.

Trong bản mô tả này, thuật ngữ "chung" nói chung để chỉ thành phần cú pháp hoặc biến mà nó có thể áp dụng cho những thứ nhiều hơn một. Ví dụ, trong ngữ cảnh các thành phần cú pháp trong thông báo SEI định thời hình ảnh, thuật ngữ "chung"" có thể có nghĩa rằng thành phần cú pháp đó (ví dụ, common_du_cpb_removal_delay) được áp dụng cho tất cả các gói giải mã trong gói truy nhập được liên kết với thông báo SEI định thời hình ảnh. Ngoài ra, các gói dữ liệu được mô tả theo "n" và "m" nói chung để chỉ lần lượt các gói truy nhập và các gói giải mã.

Một số cấu hình sẽ được mô tả có liên quan đến 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 sắp xếp 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 trình bày trên các hình vẽ, không phải là để hạn chế 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 thể hiện sơ đồ khối minh họa ví dụ về một hoặc nhiều các thiết bị điện tử 102 trong đó các hệ thống và các phương pháp để 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 đơn lẻ 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 các hình ảnh đầu vào 106. Trong một số cấu hình, (các) hình ảnh đầu vào 106 có thể được bắt giữ trên thiết bị điện tử A 102a bằng cách sử dụng cảm biến hình ảnh, có thể được gọi ra 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 được mã hóa. Ví dụ, bộ mã hóa 104 có thể mã hóa một loạt các hình ảnh đầu vào 106 (ví dụ, viđeo). Theo một cấu hình, bộ mã hóa 104 có thể là bộ mã hóa HEVC. Dữ liệu được mã hóa này có thể là dữ liệu số (ví dụ, phần của 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 mà nó hỗ trợ hoạt động ở cấp độ hình ảnh phụ, thiết bị điện tử 102 có thể gửi các thông số hình ảnh phụ, (ví dụ, thông số trễ loại bỏ CPB). Cụ thể là, thiết bị điện tử 102 (ví dụ, bộ mã hóa 104) có thể xác định liệu có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung trong thông báo SEI định thời hình ảnh hay không. Ví dụ, thiết bị điện tử có thể cài đặt cờ hiệu (ví dụ, common_du_cpb_removal_delay_flag) bằng một khi bộ mã hóa 104 được bao gồm thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) trong thông báo SEI định thời hình ảnh. Khi thông số trễ loại bỏ CPB của gói giải mã chung được bao gồm, thiết bị điện tử có thể tạo ra thông số trễ loại bỏ CPB của gói giải mã chung theo đó có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập. Nói cách khác, thay vì bao gồm thông số trễ loại bỏ CPB của gói giải mã cho mỗi gói giải mã trong gói truy nhập, thông số chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập mà cùng với nó thông báo SEI định thời hình ảnh được liên kết.

Ngược lại, khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ không được bao gồm trong thông báo SEI định thời hình ảnh, thiết bị điện tử 102 có thể tạo ra khoảng trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập mà cùng với nó thông báo SEI định thời hình ảnh được liên kết. Môđun tạo thông báo 108 có thể thực hiện một hoặc nhiều quy trình được mô tả dưới đây có liên quan 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 as part của dòng bit 114. Trong một số cấu hình thiết bị điện tử A 102a có thể gửi thông báo đến thiết bị điện tử B 102b bằng cách truyền riêng rẽ 110. Ví dụ, việc truyền riêng rẽ này có thể không phải là phần của dòng bit 114. Chẳng hạn, thông báo SEI định thời hình ảnh hoặc thông báo khác có thể được gửi bằng cách 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 được mã hóa dựa trên (các) hình ảnh đầu vào 106. Trong một số cấu hình, dòng bit 114 có thể bao gồm dữ liệu bổ sung, như thông báo SEI định thời hình ảnh hoặc thông báo khác, (các) tiêu đề phiến, (các) PPS v.v... Do các hình ảnh đầu vào 106 bổ sung được mã hóa, dòng bit 114 có thể bao gồm một hoặc nhiều các hình ảnh được mã hóa. Chẳng hạn, dòng bit 114 có thể bao gồm một hoặc nhiều hình ảnh được mã hóa tương ứng với dữ liệu bổ sung (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 bằng cách sử dụng kết nối hữu tuyến hoặc không dây. Trong một số trường hợp, việc này có thể được thực hiện qua mạng, như Internet hoặc mạng cục bộ (LAN). Như được minh họa trên Fig.1A, bộ giải mã 112 có thể được thực hiện trên thiết bị điện tử B 102b riêng rẽ 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 tới bộ giải mã 112 hoặc được lưu trữ trong bộ nhớ để lấy ra nhờ 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 được mã hóa 118 dựa trên dòng bit 114. (Các) hình ảnh được mã hóa 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 được mã hóa. CPB 120 này có thể sử dụng các thông số được tìm thấy thông báo SEI định thời hình ảnh để xác định khi loại bỏ dữ liệu. Khi CPB 120 hỗ trợ hoạt động ở cấp độ hình ảnh phụ, các gói giải mã đơn lẻ có thể được loại bỏ ở một thời gian nào đó chứ không phải là các gói truy nhập nguyên vẹn. Bộ giải mã 112 có thể bao gồm bộ đệm hình ảnh được mã hóa (DPB) 122. Mỗi hình ảnh được mã hóa được đặt vào DPB 122 để được tham chiếu nhờ quy trình mã hóa cũng như để kết xuất và cắt xén. Hình ảnh được mã hóa được loại bỏ từ DPB ở phần sau thời gian kết xuất DPB hoặc thời gian mà tại đó không còn cần dùng để tham chiếu dự đoán giữa các hình.

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 thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) hay không. Điều này có thể bao gồm việc xác định cờ hiệu (ví dụ, common_du_cpb_removal_delay_flag) theo đó nó được cài đặt khi thông số chung hiện diện trong thông báo SEI định thời hình ảnh. Nếu thông số chung hiện diện, bộ giải mã 112 có thể xác định thông số trễ loại bỏ CPB của gói giải mã chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập. Nếu thông số chung không hiện diện, bộ giải

mã 112 có thể xác định thông số trễ riêng rẽ loại bỏ CPB của gói giải mã dùng cho mỗi gói giải mã trong gói truy nhập. Bộ giải mã 112 cũng có thể loại bỏ các gói giải mã từ CPB 120 bằng cách sử dụng hoặc thông số trễ loại bỏ CPB của gói giải mã chung hoặc thông số trễ loại bỏ CPB của các gói giải mã riêng rẽ. CPB 120 có thể thực hiện một hoặc nhiều quy trình được mô tả dưới đây liên quan đến Fig.4 và Fig.5.

HRD được mô tả trên đây có thể là một ví dụ về bộ giải mã 112 được minh họa 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 phần của nó được bao gồm trong (các) thiết bị điện tử 102 có thể được thực hiện trong phần cứng. Ví dụ, một hoặc nhiều thành phần này hoặc phần 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 bằng cách 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 bằng cách sử dụng bộ chip, mạch tích hợp chuyên dụng (ASIC), Mạch tích hợp cỡ lớn (LSI) hoặc mạch tích hợp v.v...

Fig.1B thể hiện sơ đồ khối minh họa 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 đơn lẻ 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 viđeo 1908 phù hợp cho việc mã hóa viđeo có khả năng điều chỉnh và mã hóa viđeo đa hình ảnh, 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 viđeo hiệu suất cao (HEVC), bao gồm hình ảnh có thể điều chỉnh được và/hoặc đa hình ảnh. 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 bắt giữ trên thiết bị điện tử A 1902 bằng cách sử dụng cảm biến hình ảnh, được lấy ra 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 một loạt hình ảnh (ví dụ, viđeo) trong nguồn 1906. Cụ thể, để mã hóa viđeo có thể điều chỉnh được đối với khả năng điều chỉnh SNR cũng được biết tới là khả năng điều chỉnh chất lượng 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, đối với mã hóa viđeo có thể điều chỉnh được dùng cho khả năng điều chỉnh không gian nguồn được giảm mẫu có thể được sử dụng cho bộ mã hóa lớp cơ sở. Cụ thể, để mã hóa đa hình ảnh một nguồn hình 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 liên quan đến Fig.6B.

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

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 viđeo 1972 là phù hợp cho giải mã viđeo có thể điều chỉnh được và mã hóa viđeo đa hình ảnh. Theo một ví dụ, các dòng bit 1934, 1936 có thể được truyền đến thiết bị điện tử B 1970 bằng cách sử dụng kết nối hữu tuyến hoặc không dây. Trong một số trường hợp, điều này có thể được thực hiện qua mạng, như Internet hoặc Mạng cục bộ(LAN). Như được minh họa trên Fig.1B, bộ giải mã 1972 có thể được thực hiện trên thiết bị điện tử B 1970 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 việc 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 tới bộ giải mã 1972 hoặc được lưu trữ trong bộ nhớ để lấy ra nhờ bộ giải mã 1972. Bộ giải mã 1972 có thể cung cấp lớp cơ sở được giải mã 1992 và (các) hình ảnh của lớp nâng cao được 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 viđeo hiệu suất cao (HEVC), bao gồm có thể điều chỉnh được và/hoặc đa hình ảnh. 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 liên quan đến 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 liên quan đến 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 được mã hóa và/hoặc bộ đệm hình ảnh được mã hóa, như được mô tả liên quan đến 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, tùy trường hợp áp dụng.

Fig.2 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp 200 dùng để gửi thông báo. Phương pháp 200 này có thể được thực hiện nhờ bộ mã hóa 104 hoặc một trong số 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 hiệu định thời hình ảnh (ví dụ, sub_pic_cpb_params_present_flag) mà nó chỉ báo liệu CPB 120 hỗ trợ hoạt động ở cấp độ hình ảnh phụ hay không. Ví dụ, khi cờ hiệu định thời hình ảnh được cài đặt là 1, CPB 120 có thể hoạt động ở cấp độ gói truy nhập hoặc cấp độ hình ảnh phụ. Cần lưu ý rằng ngay cả khi khi cờ hiệu định thời hình ảnh được cài đặt là 1, thì quyết định về việc liệu có hoạt động ở cấp độ hình ảnh phụ hay không được để cho bộ giải mã 112 tự làm.

Bộ mã hóa 104 có thể tại bước 204 xác định một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập. Ví dụ, bộ mã hóa 104 có thể xác định một thông số trễ loại bỏ CPB gói giải mã chung (ví dụ, common_du_cpb_removal_delay) được áp dụng cho tất cả các gói giải mã trong gói truy nhập từ CPB 120. Theo phương án khác,

bộ mã hóa 104 có thể xác định khoảng trễ riêng rẽ loại bỏ CPB của gói giải mã (ví dụ, du_cpb_removal_delay[i]) dùng cho mỗi gói giải mã trong gói truy nhập.

Bộ mã hóa 104 có thể tại bước 206 xác định một hoặc nhiều thông số NAL mà nó chỉ báo số lượng, được dịch vị đi một, các gói NAL trong mỗi gói giải mã ở điểm truy nhập. Ví dụ, bộ mã hóa 104 có thể xác định một thông số NAL chung (ví dụ, common_num_nalus_in_du_minus1) được áp dụng cho tất cả các gói giải mã trong gói truy nhập từ CPB 120. Theo phương án khác, bộ mã hóa 104 có thể xác định khoảng trễ riêng rẽ loại bỏ CPB của gói giải mã (ví dụ, num_nalus_in_du_minus1[i]) dùng cho mỗi gói giải mã trong gói truy nhập.

Bộ mã hóa 104 cũng có thể tại bước 208 gửi thông báo SEI định thời hình ảnh bao gồm cờ hiệu định thời hình ảnh, các khoảng trễ loại bỏ và các thông số NAL. Thông báo SEI định thời hình ảnh có thể bao gồm các thông 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 cách truyền không dây, 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 phần của dòng bit 114. Trong một số cấu hình, thiết bị điện tử A 102a có thể gửi 208 thông báo đến thiết bị điện tử B 102b theo cách truyền riêng rẽ 110 (mà nó không phải là phần của dòng bit 114).

Chẳng hạn, thông báo đó có thể được gửi bằng cách sử dụng thiết bị ngoài băng tần nào đó. Trong trường hợp nào đó 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 hợp thông số ví dụ tập hợp thông số viđeo và/hoặc tập hợp thông số chuỗi và/hoặc tập hợp thông số hình ảnh và/hoặc tập hợp thông số tương thích và/hoặc tiêu đề phiến.

Fig.3 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp 300 để xác định một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập. Nói cách khác, phương pháp này 300 được minh họa trên Fig.3 có thể tiếp tục minh họa bước 204 trong phương pháp 200 được minh họa trên Fig.2. Phương pháp này 300 có thể được thực

hiện bởi bộ mã hóa 104. Bộ mã hóa 104 có thể tại bước 302 xác định liệu có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) hay không.

Việc này có thể bao gồm việc xác định liệu cờ hiệu trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay_flag) được cài đặt hay không. Bộ mã hóa 104 có thể gửi thông số chung trong trường hợp các gói giải mã được loại bỏ từ CPB theo các khoảng thời gian đều nhau. Đây có thể là trường hợp, ví dụ, khi mỗi gói giả mã tương ứng với số lượng dòng nhất định của hình ảnh hoặc có kết cấu bình thường khác.

Ví dụ, cờ hiệu trễ loại bỏ CPB của gói giải mã chung có thể được cài đặt là 1 khi thông số trễ loại bỏ CPB của gói giải mã chung sẽ được bao gồm trong thông báo SEI định thời hình ảnh và 0 khi nó không được bao gồm. Nếu đúng (ví dụ, cờ hiệu được cài đặt là 1), thì bộ mã hóa 104 có thể tại bước 304 xác định thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) được áp dụng cho tất cả các gói giải mã trong gói truy nhập. Nếu sai (ví dụ, cờ hiệu được cài đặt là 0), thì bộ mã hóa 104 có thể tại bước 306 xác định thông số trễ riêng rẽ loại bỏ CPB của các gói giải mã (ví dụ, du_cpb_removal_delay[i]) dùng cho mỗi gói giải mã trong gói truy nhập.

Nếu thông số trễ loại bỏ CPB của gói giải mã chung hiện diện trong thông báo SEI định thời hình ảnh, thì nó có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB 120 gói giải mã ngay trước đó trước khi loại bỏ từ CPB 120 gói giải mã hiện thời trong gói truy nhập kết hợp với thông báo SEI định thời hình ảnh.

Ví dụ, khi gói giải mã là gói giải mã trong gói truy nhập, thì thông số trễ loại bỏ CPB 120 của bộ giải mã chung có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB 120 của gói giải mã sau cùng trong gói truy nhập kết hợp với thông báo SEI khoảng thời gian đệm gần nhất trong gói truy nhập trước đó trước khi loại bỏ từ CPB 120 gói giải mã thứ nhất trong gói truy nhập kết hợp với thông báo SEI định thời hình ảnh.

Khi gói giải mã là gói giải mã không phải là thứ nhất trong gói truy nhập, thì thông số trễ loại bỏ CPB của gói giải mã chung có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB 120 gói giải mã trước đó trong gói truy nhập kết hợp với thông báo SEI định thời hình ảnh trước khi loại bỏ từ CPB gói giải mã hiện thời trong gói truy nhập kết hợp với thông báo SEI định thời hình ảnh.

Ngược lại, khi thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) không được gửi trong thông báo SEI định thời hình ảnh, thì thông số trễ riêng rẽ loại bỏ CPB của các gói giải mã (ví dụ, du_cpb_removal_delay[i]) có thể được bao gồm trong thông báo SEI định thời hình ảnh dùng cho mỗi gói giải mã trong gói truy nhập. Các thông số trễ loại bỏ CPB của gói giải mã (ví dụ, du_cpb_removal_delay[i]) có thể xác định số lượng tích tắc đồng hồ của hình ảnh phụ để chờ sau khi loại bỏ từ CPB 120 của gói giải mã sau cùng trước khi loại bỏ từ CPB 120 gói giải mã thứ i trong gói truy nhập kết hợp với thông báo SEI định thời hình ảnh. Các thông số trễ loại bỏ CPB của gói giải mã có thể được tính toán theo phần còn lại của bộ đếm $2^{(cpb_removal_delay_length_minus1 + 1)}$ môđun trong đó cpb_removal_delay_length_minus1 + 1 là độ dài của thông số trễ loại bỏ CPB của gói giải mã chung.

Fig.4 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp 400 dùng để đệm 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), mà nó có thể tại bước 402 thu thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác). Ví dụ, thiết bị điện tử 102 có thể tại bước 402 thu thông báo qua một hoặc nhiều cách truyền không dây, truyền hữu tuyến, bus thiết bị, mạng v.v... Chẳng hạn, thiết bị điện tử B 102b có thể thu 402 thông báo từ thiết bị điện tử A 102a. Thông báo có thể là phần của 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 cách truyền riêng rẽ 110 (mà nó không phải là phần của dòng bit 114, ví dụ). Chẳng hạn, thông báo SEI định thời hình ảnh có thể được thu bằng cách sử dụng thiết bị ngoài băng tần nào đó. Trong một số cấu hình, thông báo có thể bao gồm một hoặc nhiều cờ hiệu định thời

hình ảnh, một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập và một hoặc nhiều các thông số NAL. Do vậy, tại bước 402 việc thu thông báo có thể bao gồm thu một hoặc nhiều cờ hiệu định thời hình ảnh, một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập và một hoặc nhiều các thông số NAL.

Bộ giải mã 112 có thể xác định 404 liệu CPB 120 có hoạt động ở cấp độ gói truy nhập hoặc cấp độ 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 phương án khác, quyết định có thể là dựa trên việc liệu bộ giải mã 112 có đủ các nguồn để trợ giúp hoạt động dựa trên hình ảnh phụ hay không. Nếu CPB 120 hoạt động ở cấp độ hình ảnh phụ, thì bộ giải mã có thể xác định 406 một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập. Ví dụ, bộ giải mã 112 có thể xác định một thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) được áp dụng cho tất cả các gói giải mã trong gói truy nhập. Theo phương án khác, bộ giải mã 112 có thể xác định khoảng trễ riêng rẽ loại bỏ CPB của gói giải mã (ví dụ, du_cpb_removal_delay[i]) cho mỗi gói giải mã trong gói truy nhập. Nói cách khác, thông báo SEI định thời hình ảnh có thể bao gồm thông số chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập hoặc các thông số riêng rẽ cho từng gói giải mã.

Bộ giải mã 112 cũng có thể tại bước 408 loại bỏ các gói giải mã dựa trên các khoảng trễ loại bỏ cho các gói giải mã, tức là, bằng cách sử dụng hoặc thông số chung có thể áp dụng cho tất cả các gói giải mã trong gói truy nhập hoặc các thông số riêng rẽ cho từng gói giải mã. Bộ giải mã 112 cũng có thể tại bước 410 giải mã các gói giải mã.

Bộ giải mã 112 có thể sử dụng ClockDiff biến đổi khi xác định thời gian loại bỏ để được xác định từ một số thông số đã được truyền tín hiệu. Cụ thể là, 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 lượng các đơn vị thời gian của đồng hồ hoạt động tại tần số time_scale Hz tương ứng với một lượng tăng của bộ đếm tích tắc đồng hồ, num_units_in_sub_tick là số lượng các đơn vị thời gian của đồng hồ hoạt động tại tần số

time_scale Hz tương ứng với một lượng tăng của tích tắc đồng hồ của hình ảnh phụ counter, num_decoding_units_minus1+1 là số lượng các gói giải mã trong gói truy nhập, và time_scale là số lượng các đơn vị thời gian đi qua trong một giây.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1, CPB hoạt động tại cấp độ hình ảnh phụ và ClockDiff lớn hơn zero, thời gian loại bỏ đối với gói 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 loại bỏ danh định của gói giải mã m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1, CPB hoạt động tại cấp độ gói truy nhập và ClockDiff lớn hơn zero, thời gian loại bỏ dùng cho gói truy nhập 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 loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, Ceil() là phương trình trần và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + max((t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub})), (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c)))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp

độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập 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 loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập 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 loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub}

là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{r,n}(n) + (t_c * Ceil((t_{af}(n) - t_{r,n}(n)) / t_c))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã mà nó không phải là gói giải mã sau cùng của gói truy nhập được cài đặt as $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. và thời gian loại bỏ cho gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ theo: $t_r(m) = t_{r,n}(m) + (t_{c_sub} * Ceil((t_{af}(m) - t_{r,n}(m)) / t_{c_sub}))$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã mà nó không phải là gói giải mã sau cùng của gói truy nhập được cài đặt là $t_r(m) = t_{af}(m)$, trong đó $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m. và thời gian loại bỏ

cho gói giải mã sau cùng m của gói truy nhập, $t_r(m)$ 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 loại bỏ danh định của gói giải mã sau cùng m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp (ví dụ, low_delay_hrd_flag) được cài đặt là 1, $t_{r,n}(m) < t_{af}(m)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động tại cấp độ hình ảnh phụ, thời gian loại bỏ cho gói giải mã được cài đặt as $t_r(m) = t_{af}(m)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã m, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ, $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n và $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã trong gói truy nhập n.

Khi cờ hiệu của bộ giải mã tham khảo giả thiết (HRD) trễ thấp được cài đặt là 1, $t_{r,n}(n) < t_{af}(n)$, thì cờ hiệu định thời hình ảnh được cài đặt là 1 và CPB hoạt động ở cấp độ gói truy nhập, thời gian loại bỏ dùng cho gói truy nhập n, $t_r(n)$ theo: $t_r(n) = t_{af}(n)$ trong đó $t_{r,n}(m)$ là thời gian loại bỏ danh định của gói giải mã sau cùng n, t_{c_sub} là tích tắc đồng hồ của hình ảnh phụ, Ceil() là phương trình trần, $t_{af}(m)$ là thời gian tới cuối cùng của gói giải mã sau cùng m, $t_{r,n}(n)$ là thời gian loại bỏ danh định của gói truy nhập n, t_c là tích tắc đồng hồ và $t_{af}(n)$ là thời gian tới cuối cùng của gói truy nhập n.

Nếu CPB hoạt động ở cấp độ gói truy nhập, bộ giải mã 112 có thể tại bước 412 xác định thông số trễ loại bỏ CPB. Điều này có thể được bao gồm 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 loại bỏ gói truy nhập dựa trên thông số khoảng trễ loại bỏ CPB và tại bước 416 giải mã gói truy nhập. Nói cách khác, bộ giải mã 112 có thể một lúc giải mã toàn bộ các gói truy nhập, thay vì các gói giải mã bên trong gói truy nhập.

Fig.5 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp 500 cho bước xác định một hoặc nhiều khoảng trễ loại bỏ cho các gói giải mã trong gói truy nhập. Nói cách khác, phương pháp 500 được minh họa trên Fig.5 có thể tiếp tục minh họa bước 406 trong phương pháp 400 được minh họa trên Fig.4. Phương pháp 500 có thể được thực hiện bởi bộ giải mã 112. Bộ giải mã 112 có thể tại bước 502 xác định liệu thông báo SEI định thời hình ảnh thu được có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung hay không. Việc này có thể bao gồm việc xác định liệu cờ hiệu trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay_flag) có được cài đặt hay không. Nếu đúng, thì bộ giải mã 112 có thể tại bước 504 xác định thông số trễ loại bỏ CPB của gói giải mã chung (ví dụ, common_du_cpb_removal_delay) được áp dụng cho tất cả các gói giải mã trong gói truy nhập. Nếu sai, thì bộ giải mã 112 có thể tại bước 506 xác định thông số trễ riêng rẽ loại bỏ CPB của các gói giải mã (ví dụ, du_cpb_removal_delay[i]) dùng cho mỗi gói giải mã trong gói truy nhập.

Bên cạnh việc cải biến thông báo SEI định thời hình ảnh ngữ nghĩa, các hệ thống và các phương pháp hiện thời cũng có thể áp đặt giới hạn dòng bit sao cho việc vận hành hoạt động CPB dựa trên hình ảnh phụ và hoạt động CPB dựa trên gói truy nhập dẫn đến định thời giống nhau về việc loại bỏ gói giải mã. Cụ thể là, khi cờ hiệu định thời hình ảnh (ví dụ, sub_pic_cpb_params_present_flag) được cài đặt là 1, khoảng trễ loại bỏ CPB có thể được cài đặt theo

Biểu thức 6

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

trong đó du_cpb_removal_delay[i] là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số.

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

Biểu thức 7

$$T_{du}(k) = T_{du}(k-1) + t_{c_sub} * \sum_{i=0}^{num_decoding_units_minus1_k} (du_cpb_removal_delay_minus1_k[i]+1)$$

trong đó $du_cpb_removal_delay_minus1_k[i]$ và $num_decoding_units_minus1_k$ là các thông số cho gói giải mã thứ I của gói truy nhập thứ k (với k=0 cho gói truy nhập 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 thông số trễ loại bỏ CPB của gói giải mã cho gói giải mã thứ I của gói truy nhập thứ k, và $num_decoding_units_minus1_k$ là số lượng của các gói giải mã trong gói truy nhập thứ k, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, và i và k là các chỉ số. Sau đó khi cờ hiệu định thời hình ảnh (ví dụ, sub_pic_cpb_params_present_flag) được cài đặt là 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 loại bỏ CPB. Do đó trong trường hợp này khoảng trễ loại bỏ CPB (au_cpb_removal_delay_minus1 + 1) được cài đặt sao cho việc vận hành hoạt động CPB dựa trên hình ảnh phụ và hoạt động CPB dựa trên gói truy nhập dẫn đến định thời giống nhau của việc loại bỏ gói truy nhập và gói giải mã cuối cùng của việc loại bỏ gói truy nhập.

Theo phương án khác, khoảng trễ loại bỏ CPB có thể được cài đặt theo

Biểu thức 8

$$-1 \le [cpb_removal_delay * t_c - (\sum_{i=0}^{num_decoding_units_minus1} du_cpb_removal_delay[i]) * t_{c_sub})] \le 1$$

trong đó $du_cpb_removal_delay[i]$ là các thông số trễ loại bỏ CPB của gói giải mã, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một, và i là chỉ số.

Theo phương án khác, cpb_removal_delay và du_cpb_removal_delay[num_decoding_units_minus1] có thể được cài đặt 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 thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một.

Bên cạnh việc cải biến thông báo SEI định thời hình ảnh ngữ nghĩa, các hệ thống và các phương pháp hiện thời cũng có thể áp đặt giới hạn dòng bit sao cho việc vận hành hoạt động CPB dựa trên hình ảnh phụ và hoạt động CPB dựa trên gói truy nhập dẫn đến định thời giống nhau về việc loại bỏ gói giải mã. Cụ thể là, khi cờ hiệu định thời hình ảnh (ví dụ, sub_pic_cpb_params_present_flag) được cài đặt là 1, thì các trị số cho cpb_removal_delay và du_cpb_removal_delay[num_decoding_units_minus1] có thể được cài đặt để thỏa mãn: -1<=(cpb_removal_delay*t_c - du_cpb_removal_delay[num_decoding_units_minus1]* $t_{c,sub}$) <=1 trong đó du_cpb_removal_delay[num_decoding_units_minus1] là thông số trễ loại bỏ CPB của gói giải mã dùng cho gói giải mã thứ num_decoding_units_minus1, t_c là tích tắc đồng hồ, $t_{c,sub}$ là tích tắc đồng hồ của hình ảnh phụ, num_decoding_units_minus1 là số lượng các gói giải mã trong gói truy nhập được dịch vị đi một.

Fig.6A thể hiện sơ đồ khối minh họa 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, mà nó 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 nối thông điện tử với bộ nhớ với các lệnh có thể thực hiện được, phần sụn, mảng cổng lập trình được dạng trường (FPGA) 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 hình ảnh hoặc dữ liệu ảnh (ví dụ, viđeo) 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 không dây, các cổng v.v...

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

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

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

Hoặc tín hiệu trong 626 hoặc liên tín hiệu 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 được biến đổi 634 để cung cấp cho môđun lượng tử hóa 636. Môđun lượng tử hóa 636 này sẽ lượng tử hóa tín hiệu được biến đổi 634 để tạo ra các hệ số được biến đổi và lượng tử hóa (TQC) 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 này sẽ thực hiện việc lượng tử hóa ngược trên TQCs 638 để tạo ra tín hiệu được lượng tử hóa ngược 652 để cung cấp cho môđun biến đổi hóa ngược 654. Môđun biến đổi hóa ngược 654 này sẽ giải nén tín hiệu được lượng tử hóa ngược 652 để tạo ra tín hiệu được giải nén 656 để cung cấp cho môđun tái thiết lập 658.

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

Tín hiệu lọc 664 này có thể được cung cấp cho môđun độ lệch thích ứng mẫu (SAO) 666. Môđun SAO 666 có thể tạo ra thông tin SAO 668 để cung cấp cho môđun mã hóa entropy 642 và tín hiệu SAO 670 để cung cấp cho bộ lọc vòng tương thích (ALF) 672. ALF 672 tạo ra tín hiệu ALF 674 để cung cấp cho bộ đệm hình ảnh được mã hóa 676. Tín hiệu ALF 674 có thể bao gồm dữ liệu từ một hoặc nhiều các 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 được mã hóa). Ví dụ, môđun mã hóa entropy 642 có thể mã hóa các TQC 638 bằng cách sử dụng kỹ thuật Mã hóa độ dài biến đổi thuộc tính thích ứng (CAVLC) hoặc Mã hóa số học nhị phân thuộc tính thích ứng (CABAC). Cụ thể, môđun mã hóa entropy 642 có thể mã hóa các TQC 638 dựa trên một hoặc nhiều thông tin trong khung 640, thông tin liên chế độ 648 và thông tin SAO 668. Dòng bit A 614a (ví dụ, dữ liệu hình ảnh được mã hóa) 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 liên quan đến Fig.1. Ngoài ra hoặc theo phương án khác, môđun tạo thông báo 608 có thể thực hiện một hoặc nhiều quy trình được mô tả liên quan đến 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 thông số hình ảnh phụ. Các thông số hình ảnh phụ có thể bao gồm một hoặc nhiều khoảng trễ loại bỏ dùng cho các gói giải mã (ví dụ, common_du_cpb_removal_delay hoặc du_cpb_removal_delay[i]) và một hoặc nhiều các thông 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 lồng 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 lồng vào trong dòng bit A 614a (trong trường hợp đó dòng bit B 614b có thể là giống với dòng bit A 614a), nhưng có thể được cung cấp theo cách truyền riêng rẽ 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 được cung cấp cho giao diện truyền thông, giao diện mạng, máy truyền phát không dây, 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 thể hiện sơ đồ khối minh họa một cấu hình của bộ mã hóa viđeo 1782 trên thiết bị điện tử 1702. Bộ mã hóa viđeo 1782 này có thể bao gồm bộ mã hóa lớp nâng cao 1706, bộ mã hóa lớp cơ sở 1709, khối nâng cao tỷ lệ phân giải 1770 và giao diện đầu ra 1780. Bộ mã hóa viđeo trên Fig.6B, ví dụ, phù hợp cho việc mã hóa viđeo có khả năng điều chỉnh và mã hóa viđeo đa hình ảnh, như được mô tả trong bản mô tả này.

Bộ mã hóa lớp nâng cao 1706 có thể bao gồm đầu vào viđeo 1781 để thu hình ảnh đầu vào 1704. Kết xuất của đầu vào viđeo 1781 có thể được cung cấp cho bộ cộng/bộ trừ 1783 để thu kết xuất của lựa chọn dự báo 1750. Kết xuất của bộ cộng/bộ trừ 1783 có thể được cung cấp cho khối biến đổi và lượng tử hóa 1752. Kết xuất của khối biến đổi và lượng tử hóa 1752 này có thể được cung cấp cho khối mã hóa entropy 1748 và khối tạo tỷ lệ và biến đổi ngược1772. Sau khi mã hóa entropy 1748 được thực hiện, kết xuất 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 viđeo của lớp cơ sở được mã hóa 1707 và dòng bit viđeo của lớp nâng cao được mã hóa 1710.

Kết xuất của khối tạo tỷ lệ và biến đổi ngược1772 có thể được cung cấp cho bộ cộng 1779. Bộ cộng 1779 cũng có thể thu kết xuất của lựa chọn dự báo 1750. Kết xuất của bộ cộng 1779 có thể được cung cấp cho khối giải khối 1751. Kết xuất của khối giải khối 1751 có thể được cung cấp cho bộ đệm tham chiếu 1794. Kết xuất của bộ đệm tham chiếu 1794 có thể được cung cấp cho khối bù chuyển động 1754. Kết xuất của khối bù chuyển động 1754 có thể được cung cấp cho lựa chọn dự báo 1750. Kết xuất của bộ đệm tham chiếu 1794 cũng có thể được cung cấp cho bộ dự báo trong 1756. Kết xuất của bộ dự báo trong 1756 có thể được cung cấp cho lựa chọn dự báo 1750. Lựa chọn dự báo 1750 cũng có thể receive kết xuất củakhối nâng cao tỷ lệ phân giải 1770.

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

Fig.7A thể hiện sơ đồ khối minh họa một cấu hình của bộ giải mã 712 trên thiết bị điện tử 702. Bộ giải mã 712 có thể được bao gồm 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 được bao gồm 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 các hình ảnh được mã hóa và dữ liệu bổ sung được bao gồm trong dòng bit 714) để mã hóa. Trong một số cấu hình, dòng bit thu được 714 có thể bao gồm dữ liệu bổ sung thu được, như thông báo (ví dụ, thông báo SEI định thời hình ảnh hoặc thông báo khác), tiêu đề phiến, PPS v.v... Trong một số cấu hình, bộ giải mã 712 còn có thể thu truyền riêng rẽ 710. Việc truyền riêng rẽ 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 theo cách truyền riêng rẽ 710 thay vì trong dòng bit 714. Tuy nhiên, cần lưu ý rằng việc truyền riêng rẽ 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 liên quan đến Fig.1 nêu trên. Ngoài ra hoặc theo phương án khác, bộ giải mã 712 có thể thực hiện một hoặc nhiều quy trình được mô tả liên quan đến 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 thông số hình ảnh phụ và loại bỏ và giải mã các gói giải mã trong gói truy nhập dựa trên các thông số hình ảnh phụ. Cần lưu ý rằng một hoặc nhiều các gói truy nhập có thể được bao gồm trong dòng bit và có thể bao gồm một hoặc nhiều dữ liệu hình ảnh được mã hóa và dữ liệu bổ sung.

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

Tín hiệu thông tin chuyển động 703 có thể được kết hợp với một phần của tín hiệu khung tham chiếu 798 từ bộ đệm hình ảnh được mã hóa 709 tại môđun bù chuyển động 780, mà nó có thể tạo ra tín hiệu dự báo liên khung 782. Các hệ số được lượng tử hóa, được tạo tỷ lệ và/hoặc được biến đổi 705 có thể được lượng tử hóa ngược, được tạo tỷ lệ và được biến đổi ngược bởi môđun ngược 707, nhờ đó tạo ra tín hiệu dư được giải mã

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

Fig.7B thể hiện sơ đồ khối minh họa một cấu hình của bộ giải mã viđeo 1812 trên thiết bị điện tử 1802. Bộ giải mã viđeo 1812 này có thể bao gồm bộ giải mã lớp nâng cao 1815 và bộ giải mã lớp cơ sở 1813. Bộ giải mã viđeo 812 này có thể bao gồm giao diện 1889 và nâng cao tỷ lệ phân giải 1870. Bộ giải mã viđeo trên Fig.7B, ví dụ, là phù hợp cho giải mã viđeo có thể điều chỉnh được và mã hóa viđeo đa hình ảnh, như được mô tả trong bản mô tả này.

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

Giao diện 1889 cũng có thể cung cấp một phần hoặc toàn bộ dòng viđeo được mã hóa 1885 cho khối giải mã entropy 1890 trong bộ giải mã lớp nâng cao 1815. Kết xuất của khối giải mã entropy 1890 có thể được cung cấp cho khối lượng tử hóa ngược1891. Kết xuất của khối lượng tử hóa ngược1891 có thể được cung cấp cho bộ cộng 1892. Bộ cộng 1892 này có thể bổ sung kết xuất của khối lượng tử hóa ngược1891 và kết xuất của khối lựa chọn dự báo 1895. Kết xuất của bộ cộng 1892 này có thể được cung cấp cho khối giải khối 1893. Kết xuất 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 viđeo được giải mã lớp nâng cao 1882. Kết xuất của bộ đệm tham chiếu 1894 cũng có thể được cung cấp cho bộ dự báo trong 1897. Bộ giải mã lớp nâng cao 1815 có thể bao gồm bù chuyển động 1896. Bù chuyển động 1896 có thể được thực hiện sau khi nâng cao tỷ lệ phân giải 1870. Khối lựa chọn dự báo 1895 có thể thu kết xuất của bộ dự báo trong 1897 và kết xuất của bù chuyển động 1896. Ngoài ra, bộ giải mã có thể bao gồm một hoặc nhiều bộ đệm hình ảnh được mã hóa, nếu cần, như đi kèm với giao diện 1889.

Fig.8 minh họa một số thành phần có thể được sử dụng trong thiết bị điện tử truyền phát 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 phát 802 được minh họa trên Fig.8.

Thiết bị điện tử truyền phát 802 bao gồm bộ xử lý 817 mà nó 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, mà nó có thể bao gồm cả bộ nhớ chỉ đọc (ROM), bộ nhớ truy cập ngẫu nhiên (RAM) hoặc loại thiết bị bất kỳ mà nó có thể lưu trữ thông tin, cung cấp các lệnh 813a (ví dụ, các lệnh thực hiện) và dữ liệu 815a cho bộ xử lý 817. Một phần của bộ nhớ 811 có thể bao gồm bộ nhớ truy cập ngẫu nhiên không thay đổi (NVRAM). 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 lưu trú 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 mà chúng đã được tải 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 triển khai 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 phát 802 có thể bao gồm một hoặc nhiều các giao diện truyền thông 819 để truyền thông với các thiết bị điện tử khác (ví dụ, thiết bị điện tử thu nhận). Các giao diện truyền thông 819 có thể là dựa trên công nghệ truyền thông hữu tuyến, công nghệ truyền thông không dây, 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), bộ thích ứng Ethernet, bus giao diện 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 cộng tác thế hệ thứ 3 (3GPP) và tương tự.

Thiết bị điện tử truyền phát 802 có thể bao gồm một hoặc nhiều các thiết bị kết xuất 823 và một hoặc nhiều thiết bị đầu vào 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 được bao gồm trong thiết bị điện tử 802 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 đoán ảnh phù hợp bất kỳ, như đèn tia âm cực (CRT), màn hình tinh thể lỏng (LCD), đi-ốt phát quang (LED), plasma khí, điện quang hoặc loại tương tự. Bộ kiểm soát màn hình 827 có thể được cung cấp 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 trình bày trên màn hình 825. Các ví dụ về các thiết bị đầu vào 821 bao gồm bàn phím, chuột, micrô, thiết bị điều khiển từ xa, nút bấm, cần điều khiển, bi xoay, bảng xúc giác, màn hình cảm ứng, bút quang v.v...

Một số thành phần của thiết bị điện tử truyền phát 802 được ghép nối với nhau nhờ hệ thống bus 829, mà nó có thể bao gồm bus công suất, bus tín hiệu điều khiển và bus tín hiệu trạng thái, thêm vào đó bus dữ liệu. Tuy nhiên, để cho dễ hiểu, một số bus

được minh họa trên Fig.8 dưới dạng hệ thống bus 829. Thiết bị điện tử truyền phát 802 được minh họa trên Fig.8 là biểu đồ khối chức năng thay vì liệt kê các phần tử cụ thể.

Fig.9 thể hiện sơ đồ khối minh họa một số thành phần có thể được sử dụng trong thiết bị điện tử thu nhận 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 nhận 902 được minh họa trên Fig.9.

Thiết bị điện tử thu nhận 902 bao gồm bộ xử lý 917 mà nó điều khiển hoạt động của thiết bị điện tử 902. Bộ xử lý 917 cũng có thể được gọi là CPU. Bộ nhớ 911, mà nó có thể bao gồm cả bộ nhớ chỉ đọc (ROM), bộ nhớ truy cập ngẫu nhiên (RAM) hoặc loại thiết bị bất kỳ mà nó có thể lưu trữ thông tin, cung cấp các lệnh 913a (ví dụ, các lệnh thực hiện) và dữ liệu 915a cho bộ xử lý 917. Một phần của bộ nhớ 911 có thể bao gồm bộ nhớ truy cập ngẫu nhiên không thay đổi (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 lưu trú 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 mà chúng đã được tải để thực hiện hoặc xử lý bởi bộ xử lý 917. Các lệnh 913b có thể được thực hiện bởi bộ xử lý 917 để triển khai 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 nhận 902 có thể bao gồm một hoặc nhiều các giao diện truyền thông 919 để truyền thông với các thiết bị điện tử khác (ví dụ, thiết bị điện tử truyền phát). Giao diện truyền thông 919 có thể là dựa trên công nghệ truyền thông hữu tuyến, công nghệ truyền thông không dây, 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 cộng tác thế hệ thứ 3 (3GPP) và tương tự.

Thiết bị điện tử thu nhận 902 có thể bao gồm một hoặc nhiều các thiết bị kết xuất 923 và một hoặc nhiều các thiết bị đầu vào 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 được bao gồm trong thiết bị điện tử 902 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 đoán ảnh phù hợp bất kỳ, như đèn tia âm cực (CRT), màn hình tinh thể lỏng (LCD), đi-ốt phát quang (LED), plasma khí, điện quang hoặc loại tương tự. Bộ kiểm soát màn hình 927 có thể được cung cấp 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 trình bày trên màn hình 925. Các ví dụ về các thiết bị đầu vào 921 bao gồm bàn phím, chuột, micrô, thiết bị điều khiển từ xa, nút bấm, cần điều khiển, bi xoay, bảng xúc giác, màn hình cảm ứng, bút quang v.v...

Một số thành phần của thiết bị điện tử thu nhận 902 được ghép nối với nhau nhờ hệ thống bus 929, mà nó có thể bao gồm bus công suất, bus tín hiệu điều khiển và bus tín hiệu trạng thái, bên cạnh bus dữ liệu. Tuy nhiên, để cho dễ hiểu, một số bus được minh họa trên Fig.9 dưới dạng hệ thống bus 929. Thiết bị điện tử thu nhận 902 được minh họa trên Fig.9 là biểu đồ khối chức năng thay cho liệt kê các phần tử cụ thể.

Fig.10 thể hiện sơ đồ khối minh họa một cấu hình của thiết bị điện tử 1002 trong đó các hệ thống và các phương pháp dùng để gửi thông báo có thể được thực hiện. Thiết bị điện tử 1002 bao gồm các phương tiện thu mã hóa 1031 và các phương tiện thu truyền 1033. Các các phương tiện thu mã hóa 1031 và các phương tiện thu 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 Fig.1, Fig.2, Fig.3, Fig.6 và Fig.8 nêu trên. Ví dụ, các phương tiện thu mã hóa 1031 và các phương tiện thu 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 thể hiện sơ đồ khối minh họa một cấu hình của thiết bị điện tử 1102 trong đó các hệ thống và các phương pháp dùng để 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 giải mã 1137. Các phương tiện thu 1135 và các phương 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 Fig.1, Fig.4, Fig.5, Fig.7 và Fig.9 nêu trên. Ví dụ, các phương tiện thu 1135 và các phương 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 thể hiện sơ đồ khối minh họa một cấu hình của phương pháp 1200 để vận hành bộ đệm hình ảnh được giải mã (DPB). Phương pháp 1200 này có thể được thực hiện bởi bộ mã hóa 104 hoặc một trong số các bộ phận phụ của nó (ví dụ, mô đun của bộ đệm hình ảnh được mã hóa 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 phương án khác phương pháp này 1200 có thể được thực hiện bởi bộ giải mã 712 hoặc một trong số các bộ phận phụ của nó (ví dụ, môđun của bộ đệm hình ảnh được mã hóa 709). Bộ giải mã này có thể phân tách tiêu đề phiến thứ nhất của hình ảnh 1202. Việc kết xuất và loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thời khi gói giải mã thứ nhất của gói truy nhập có chứa hình ảnh hiện thời được loại bỏ từ CPB và tiếp tục diễn ra như sau.

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

- Dòng bit của viđeo có thể bao gồm kết cấu cú pháp mà nó được đặt vào các bó dữ liệu lôgic được gọi chung là các gói của Lớp mạng trừu tượng (NAL). Mỗi gói NAL bao gồm tiêu đề gói NAL, như tiêu đề gói NAL hai byte (ví dụ, 16 bit), để xác định mục đích của chuỗi dữ liệu liên kết. Ví dụ, mỗi phiến (và/hoặc hình ảnh) được mã hóa có thể được mã hóa thành một hoặc nhiều gói NAL của phiến (và/hoặc hình ảnh). Các gói NAL khác có thể được bao gồm dùng cho các hạng mục dữ liệu khác, như ví dụ, thông tin nâng cao bổ sung, phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời (TSA), phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời từng bước (STSA), phiến được mã hóa không TSA, hình ảnh kế tiếp không STSA, phiến được mã hóa của hình ảnh truy nhập liên kết gãy, phiến được mã hóa của hình ảnh làm tươi giải mã tức thời, phiến được mã hóa của hình ảnh truy nhập ngẫu nhiên sạch, phiến được mã hóa của hình ảnh chính có thể giải mã, phiến được mã hóa của được dán nhãn đối với hình ảnh loại bỏ, tập hợp thông số viđeo, tập hợp thông số chuỗi, tập hợp thông số hình ảnh, dấu phân tách gói truy nhập, đuôi của dãy, đuôi của dòng bit, dữ liệu bộ lọc, và/hoặc thông báo thông tin nâng cao dãy. Bảng (4) minh họa một ví dụ về các mã của gói NAL và các lớp của loại gói NAL. Các loại gói NAL khác có thể được bao gồm, nếu cần. Cũng cần phải hiểu là các trị số loại gói NAL đối với các gói NAL được trình bày trong bảng (4) có thể được trộn lại và được ấn định lại. Ngoài ra các loại gói NAL bổ sung cũng có thể được thêm vào. Ngoài ra một số loại gói NAL cũng có thể được loại bỏ.

Hình ảnh của điểm truy nhập ngẫu nhiên (IRAP) là hình ảnh được mã hóa mà đối với nó mỗi gói NAL lớp mã hóa viđeo có nal_unit_type nằm trong khoảng từ BLA_W_LP đến RSV_IRAP_VCL23, bao gồm cả trị số cuối như được trình bày tại bảng (4). Hình ảnh IRAP chỉ chứa Các phiến mã hóa trong (I). Hình ảnh làm tươi giải mã tức thời (IDR) là hình ảnh IRAP mà đối với nó mỗi gói NAL lớp mã hóa viđeo có nal_unit_type bằng IDR_W_RADL hoặc IDR_N_LP như được trình bày tại bảng (4). Hình ảnh làm tươi giải mã tức thời (IDR) chỉ chứa các phiến I, và có thể là hình ảnh thứ nhất trong dòng bit trong thứ tự mã hóa, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh IDR là hình ảnh thứ nhất của dãy viđeo được mã hóa (CVS) trong thứ tự mã

hóa. Hình ảnh truy nhập liên kết gãy (BLA) là hình ảnh IRAP mà đối với nó mỗi gói NAL lớp mã hóa viđeo có nal_unit_type bằng BLA_W_LP, BLA_W_RADL, hoặc BLA_N_LP như được trình bày tại bảng (4). Hình ảnh BLA chỉ chứa các phiến I, và có thể là hình ảnh thứ nhất trong dòng bit trong thứ tự mã hóa, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh BLA bắt đầu dãy viđeo được 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 thành phần cú pháp để xác định tập hợp hình ảnh tham chiếu không trống.

Bảng 4

nal_unit_type	Tên của nal_unit_type	Nội dung gói NAL và kết cấu cú pháp chuỗi byte thô (RBSP)	Lớp của loại gói NAL
0 1	TRAIL_ N TRAIL_R	Phân đoạn phiến được mã hóa của hình ảnh kéo không TSA, không STSA slice_segment_layer_rbsp()	Lớp mã hóa viđeo (VCL)
2 3	TSA_N TSA_R	Phân đoạn phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời (TSA) slice_segment_layer_rbsp()	VCL
4 5	STSA_N STSA_R	Phân đoạn phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời từng bước (STSA) slice_segment_layer_rbsp()	VCL
6 7	RADL_N RADL_R	Phân đoạn phiến được mã hóa 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 phiến được mã hóa của hình ảnh chính 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 gói NAL VCL không tham chiếu lớp phụ không IRAP dự trữ	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Các loại gói NAL VCL tham chiếu lớp phụ không IRAP dự trữ	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_ N_LP	Phân đoạn phiến được mã hóa của hình ảnh truy nhập liên kết gãy (BLA) slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Phân đoạn phiến được mã hóa của hình ảnh làm tươi giải mã tức thời (IDR) slice. segment_layer_rbsp()	VCL
21	CRA_NUT	Phân đoạn phiến được mã hóa hình ảnh truy nhập ngẫu nhiên sạch (CRA) slice_segment_layer_rbsp()	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Các loại gói NAL VCL IRAP dự trữ	VCL
24..31	RSV_VCL24.. RSV_VCL31	Các loại gói NAL VCL không IRAP dự trữ	VCL

32	VPS_NUT	Tập hợp thông số viđeo Video_parameter_set_rbsp()	lớp mã hóa không viđeo (không VCL)
33	SPS_NUT	Tập hợp thông số dãy seq parameter_set_rbsp()	không VCL
34	PPS_NUT	Tập hợp thông số hình ảnh pic_parameter_ set rbsp()	không VCL
35	AUD_NUT	Dấu phân tách gói truy nhập access_unit_delimiter_rbsp()	không VCL
36	EOS_NUT	Đuôi của dãy 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 trữ	không VCL
48..63	UNSPEC48.. UNSPEC83	Không xác định	không VCL

Bảng (4)

Trên bảng (5), cú pháp tiêu đề gói NAL có thể bao gồm hai byte dữ liệu, tức 16 bit. Bit thứ nhất là "forbidden_zero_bit" luôn luôn được cài đặt ở zero tại điểm bắt đầu của gói NAL. Sáu bit tiếp theo là "nal_unit_type" xác định loại kết cấu dữ liệu của các dãy byte thô ("RBSP") được chứa trong gói NAL như được trình bày tại 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 đặc điểm cơ sở của tiêu chuẩn. Trong các phần mở rộng mã hóa viđeo có thể điều chỉnh được và/hoặc các phần mở rộng cú pháp thì nuh_layer_id có thể xác định rằng gói NAL cụ thể này thuộc về lớp được xác định bởi trị số của 6 bit này. Thành phần cú pháp tiếp theo là "nuh_temporal_id_plus1". nuh_temporal_id_plus1 trừ 1 đó có thể xác định nhãn hiệu tạm thời cho gói NAL đó. Nhãn hiệu tạm thời biến đổi được TemporalId có thể được xác định là TemporalId = nuh_temporal_id_plus1 - 1. Nhãn hiệu tạm thời TemporalId được sử dụng để xác định lớp phụ tạm thời. Biến HighestTid xác định lớp phụ tạm thời cao nhất sẽ phải giải mã.

Bảng 5

nal_unit_header() {	Tên
forbiden_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 kết cấu cú pháp tập hợp thông số chuỗi (SPS) có tính ví dụ.

pic_width_in_luma_samples xác định rằng chiều rộng của mỗi hình ảnh được mã hóa trong các gói luma samples. pic_width_in_luma_samples không bằng 0.

pic_height_in_luma_samples xác định rằng chiều cao mỗi hình ảnh được mã hóa trong các gói luma samples. pic_height_in_luma_samples không bằng 0.

sps_max_sub_layers_minus1 cộng 1 xác định số lượng tối đa của các lớp phụ tạm thời mà nó có thể hiện diện trong mỗi CVS tham chiếu đến SPS. Trị số sps_max_sub_layers_minus1 phải nằm trong khoảng từ 0 đến 6, bao gồm cả trị số cuối.

Cờ hiệu sps_sub_layer_ordering_info_present_flag bằng 1 xác định rằng các thành phần cú pháp sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], và sps_max_latency_increase_plus1[i] hiện diện đối với các lớp phụ sps_max_sub_layers_minus1 + 1. Còn sps_sub_layer_ordering_info_present_flag bằng 0 xác định rằng các trị số 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] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng i. Trị số sps_max_dec_pic_buffering_minus1[i] phải nằm trong khoảng từ 0 đến MaxDpbSize - 1, bao gồm cả trị số cuối trong đó MaxDpbSize xác định kích thước tối đa của bộ đệm hình ảnh được giải mã trong các gói của các bộ đệm lưu trữ hình ảnh. Khi i lớn hơn 0, thì sps_max_dec_pic_buffering_minus1[i] phải lớn

hơn hoặc bằng sps_max_dec_pic_buffering_minus1[i - 1]. Khi sps_max_dec_pic_buffering_minus1[i] không hiện diện với i nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag being bằng 0, thì nó được kết luận bằng sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1].

sps_max_num_reorder_pics[i] chỉ báo số lượng tối đa cho phép của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS trong thứ tự mã hóa và tiếp theo sau hình ảnh đó trong thứ tự kết xuất khi HighestTid bằng i. Trị số sps_max_num_reorder_pics[i] phải nằm trong khoảng từ 0 đến sps_max_dec_pic_buffering_minus1[i], bao gồm cả trị số cuối. Khi i lớn hơn 0, thì sps_max_num_reorder_pics[i] phải lớn hơn hoặc bằng sps_max_num_reorder_pics[i - 1]. Khi sps_max_num_reorder_pics[i] không hiện diện với i nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag bằng 0, thì nó được kết luận bằng 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 trị số SpsMaxLatencyPictures[i], mà nó xác định số lượng tối đa của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS trong thứ tự kết xuất và tiếp theo sau hình ảnh đó trong thứ tự mã hóa khi HighestTid bằng i.

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

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 biểu diễn.

Trị số sps_max_latency_increase_plus1[i] phải nằm trong khoảng từ 0 đến 2^{32} - 2, bao gồm cả trị số cuối. Khi sps_max_latency_increase_plus1[i] không hiện diện với i

nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag being bằng 0, thì nó được kết luận bằng sps_max_latency_increase_plus1[sps_max_sub_layers_minus1].

Bảng 6

seq_parameter_set_rbsp() {
...
sps_max_sub_layers_minus1
...
pic_width_in_luma_samples
pic_height_in_luma_samples
...
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]
sps_max_num_reorder_pics[i]
sps_max_latency_increase_plus1[i]
}
...
}

Bảng (6)

Khi hình 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 trong thứ tự mã hóa, hoặc hình ảnh thứ nhất tiếp theo sau đuôi gói NAL của dãy trong thứ tự mã hóa, thì biến NoRaslOutputFlag được cài đặt bằng 1.

- Khác đi, nếu một số thiết bị bên ngoài là có thể sử dụng được để cài đặt biến HandleCraAsBlaFlag ở trị số cho hình ảnh hiện thời, thì biến HandleCraAsBlaFlag được cài đặt bằng trị số được cung cấp bởi các thiết bị bên ngoài đó và biến NoRaslOutputFlag được cài đặt bằng HandleCraAsBlaFlag.

- Khác đi, biến HandleCraAsBlaFlag được cài đặt bằng 0 và biến NoRaslOutputFlag được cài đặt bằng 0.

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

1. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm như sau:

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

- Khác đi, nếu trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] nhận được từ SPS hoạt động lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động cho hình ảnh đứng trước, thì NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được cài đặt là 1 nhờ bộ giải mã đang xét nghiệm, không quan tâm đến trị số no_output_of_prior_pics_flag.

- Khác đi, thì NoOutputOfPriorPicsFlag được cài đặt bằng no_output_of_prior_pics_flag.

2. Trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét 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 trống không cần đến kết xuất của các hình ảnh chúng chứa, và độ đầy của DPB được cài đặt bằng 0.

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

- Khác đi (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 hình ảnh được đánh dấu là "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh mà nó được làm trống, độ đầy của 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 độ đầy của DPB đi một đối với mỗi bộ

đệm lưu trữ hình ảnh bổ sung được làm trống, cho tới khi không có điều kiện nào sau đây là đúng:

1. Số lượng hình ảnh có trị số nuh_layer_id cụ thể trong DPB được đánh dấu là "cần kết xuất" lớn hơn sps_max_num_reorder_pics[HighestTid] từ tập hợp thông số chuỗi hoạt động (khi trị số nuh_layer_id cụ thể này bằng 0) hoặc từ tập hợp thông số chuỗi của lớp hoạt động đối với trị số nuh_layer_id cụ thể đó.

2. Nếu sps_max_latency_increase_plus1[HighestTid] từ tập hợp thông số chuỗi hoạt động (khi trị số nuh_layer_id cụ thể này bằng 0) hoặc từ tập hợp thông số chuỗi của lớp hoạt động đối với trị số nuh_layer_id cụ thể đó không bằng 0 và có ít nhất một hình ảnh có trị số nuh_layer_id cụ thể trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì PicLatencyCount có thể thay đổi được liên kết lớn hơn hoặc bằng SpsMaxLatencyPictures[HighestTid] đối với trị số nuh_layer_id cụ thể đó.

3. Số lượng hình ảnh có trị số nuh_layer_id cụ thể trong DPB lớn hơn hoặc bằng sps_max_dec_pic_buffering[HighestTid] + 1 từ tập hợp thông số chuỗi hoạt động (khi trị số nuh_layer_id cụ thể này bằng 0) hoặc từ tập hợp thông số chuỗi của lớp hoạt động đối với trị số 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 thời khi gói giải mã sau cùng của gói truy nhập có chứa hình ảnh hiện thời được loại bỏ từ CPB.

Đối với mỗi hình ảnh có trị số nuh_layer_id bằng trị số nuh_layer_id của hình ảnh hiện thời trong DPB được đánh dấu là "cần kết xuất", thì PicLatencyCount có thể thay đổi được liên kết được cài đặt bằng PicLatencyCount + 1.

Hình ảnh hiện thời được coi như đã được giải mã sau khi gói giải mã sau cùng của hình ảnh được giải mã. Hình ảnh được mã hóa 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 được mã hóa hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu là "cần kết xuất" và PicLatencyCount thay đổi được liên kết được cài đặt bằng 0.

- Khác đi (hình ảnh được mã hóa hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu là "không cần kết xuất".

Hình ảnh được mã hóa 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 trị số nuh_layer_id bằng trị số 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 hợp thông số chuỗi hoạt động (khi trị số nuh_layer_id của hình ảnh hiện thời bằng 0) hoặc từ tập hợp thông số chuỗi của lớp hoạt động đối với trị số nuh_layer_id của hình ảnh hiện thời.

- sps_max_latency_increase_plus1[HighestTid] từ tập hợp thông số chuỗi hoạt động (khi trị số nuh_layer_id của hình ảnh hiện thời bằng 0) hoặc từ tập hợp thông số chuỗi của lớp hoạt động đối với trị số nuh_layer_id của hình ảnh hiện thời không bằng 0 và có ít nhất một hình ảnh có trị số nuh_layer_id cụ thể trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì PicLatencyCount có thể thay đổi được liên kết lớn hơn hoặc bằng SpsMaxLatencyPictures[HighestTid] đối với trị số nuh_layer_id cụ thể đó.

Quy trình "đệm" 1204 và quy trình đệm bổ sung 1208 là giống hệt nhau về các bước và gồm các bước theo thứ tự sau đây: Các hình ảnh mà chúng là thứ nhất cho kết xuất được lựa chọn là những hình ảnh có trị số 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 có thể biến đổi mà nó kết hợp với mỗi hình ảnh, xác định một cách duy nhất hình ảnh được liên kết giữa tất cả các hình ảnh trong CVS, và, khi hình ảnh được liên kết đó được kết xuất từ bộ đệm hình ảnh được mã hóa, chỉ báo vị trí của hình ảnh liên kết đó trong thứ tự kết xuất liên quan đến các vị trí thứ tự kết xuất của hình ảnh khác trong cùng CVS mà nó sẽ là đầu ra từ bộ đệm hình ảnh được mã hóa.

- Các hình ảnh này được cắt xén, bằng cách sử dụng cửa sổ cắt xén phù hợp được xác định trong tập hợp thông số chuỗi hoạt động đối với hình ảnh có nuh_layer_id bằng 0 hoặc trong tập hợp thông số chuỗi của lớp hoạt động đối với trị số nuh_layer_id bằng trị số nuh_layer_id của hình ảnh, các hình ảnh được cắt xén đó là đầu ra theo thứ tự tăng của nuh_layer_id, và các hình ảnh đó được đánh dấu là "không cần kết xuất".

- Mỗi bộ đệm lưu trữ hình ảnh có chứa hình ảnh được đánh dấu là "không sử dụng để tham chiếu" và theo đó bao gồm một trong số các hình ảnh mà nó đã được cắt xén và kết xuất được làm trống.

Trên Fig.13A, do cú pháp tiêu đề gói 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 luôn được cài đặt ở zero tại điểm bắt đầu của gói NAL. Sáu bit tiếp theo là "nal_unit_type" sẽ xác định loại kết cấu dữ liệu của các dãy byte thô ("RBSP") chứa trong gói NAL. 6 bit tiếp theo là "nuh_reserved_zero_6bits". nuh_reserved_zero_6bits này có thể bằng 0 theo đặc điểm cơ sở của tiêu chuẩn. Các trị số khác của nuh_reserved_zero_6bits có thể được xác định nếu cần. Cá bộ giải mã có thể bỏ qua (nghĩa là, loại bỏ từ dòng bit và vứt bỏ) tất cả các gói NAL có các trị số nuh_reserved_zero_6bits không bằng 0 khi khi thao tác dòng dựa trên đặc điểm cơ sở của tiêu chuẩn. Trong nuh_reserved_zero_6bits có thể điều chỉnh được hoặc các phần mở rộng khác có thể xác định các trị số khác, để truyền tín hiệu mã hóa viđeo có thể điều chỉnh được và/hoặc các phần mở rộng cú pháp. Trong một số trường hợp thành phần cú pháp nuh_reserved_zero_6bits có thể được gọi là reserved_zero_6bits. Trong một số trường hợp thành phần cú pháp nuh_reserved_zero_6bits thể được gọi là layer_id_plus1 hoặc layer_id, như được minh họa trên Fig.13B và Fig.13C. Trong trường hợp này thành phần layer_id sẽ là layer_id_plus1 trừ 1. Trong trường hợp này nó có thể được sử dụng để truyền tín hiệu thông tin liên quan đến viđeo được mã hóa có thể điều chỉnh được. Thành phần cú pháp tiếp theo là "nuh_temporal_id_plus1". nuh_temporal_id_plus1 trừ 1 có thể xác định nhãn hiệu tạm thời cho gói NAL đó. Nhãn hiệu tạm thời biến đổi được TemporalId có thể được xác định là TemporalId = nuh_temporal_id_plus1 - 1.

Trên Fig.14, kết cấu cú pháp chung của gói NAL được minh họa. Cú pháp hai byte của tiêu đề gói NAL trên Fig.13 được bao gồm liên quan đến nal_unit_header() trên Fig.14. Phần còn lại của cú pháp gói NAL chủ yếu lien quan đến RBSP.

Một kỹ thuật hiện có đối với việc sử dụng "nuh_reserved_zero_6bits" là truyền thông tin tín hiệu mã hóa viđeo có thể điều chỉnh 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 hình, và cờ hiệu chiều sâu, mỗi loại để chỉ việc xác định lớp khác nhau của viđeo được mã hóa có thể điều chỉnh được. Do đó 6 bit đó chỉ báo gói NAL cụ thể này thuộc lớp nào của kỹ thuật mã hóa có thể điều chỉnh được. Sau đó trong chuỗi dữ liệu, như cú pháp phần mở rộng của tập hợp tham số viđeo ("VPS") ("scalability_type") như được minh họa trên Fig.15, thông tin về lớp đó được xác định. Cú pháp phần mở rộng VPS trên Fig.15 bao gồm 4 bit cho loại có thể điều chỉnh được (thành phần cú pháp scalability_type) mà nó xác định các loại có thể điều chỉnh được đang được sử dụng trong dãy viđeo được mã hóa và kích thước được truyền tín hiệu thông qua layer_id_plus1 (or layer_id) trong tiêu đề gói NAL. Khi loại có thể điều chỉnh bằng 0, dãy viđeo được mã hóa phù hợp với chuẩn cơ sở, do vậy layer_id_của tất cả các gói NAL bằng 0 và không có gói NAL nào thuộc lớp hoặc hình nâng cao hoặc. Các trị số cao hơn của loại có thể điều chỉnh được trình bày như được minh họa trên Fig.16.

layer_id_dim_len[i] xác định độ dài, theo bit, của ID kích thước có thể điều chỉnh thứ i. Tổng của các trị số layer_id_dim_len[i] đối với mọi trị số i nằm trong khoảng từ 0 đến 7 sẽ nhỏ hơn hoặc bằng 6. vps_extension_byte_alignment_reserved_zero_bit là zero. vps_layer_id[i] xác định trị số 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 các 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 nhãn hiệu điều chỉnh được trong gói NAL và trong tập hợp thông số viđeo để phân bổ các bit trong số các loại khả năng điều chỉnh được liệt kê trên Fig.16. Sau đó đối với mỗi loại khả năng điều

chỉnh, Fig.16 xác định bao nhiêu kích thước được trợ giúp. Ví dụ, loại khả năng điều chỉnh 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 các bit được phân bổ cho mỗi trong hai kích thước này, trong đó tổng cộng tất cả các trị số của layer_id_dim_len[i] nhỏ hơn hoặc bằng 6, mà là số lượng các bit trong nuh_reserved_zero_6bits của tiêu đề gói NAL. Do vậy, để kết hợp kỹ thuật này xác định các loại nào điều chỉnh được đang được sử dụng và 6 bit của tiêu đề gói NAL được phân bổ như thế nào cho khả năng điều chỉnh.

Với sự kết hợp cố định như vậy của các kích thước của khả năng điều chỉnh khác nhau, như được minh họa 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 phần mở rộng của tập hợp thông số viđeo được cải biến xác định loại khả năng điều chỉnh cho mỗi bit trong thành phần cú pháp nuh_reserved_zero_6bits . vps_extension_byte_alignment_reserved_zero_bit được cài đặt bằng 0. max_num_layers_minus1_bits chỉ báo tổng số lượng các bit được sử dụng cho thành phần cú pháp trong hai byte đầu tiên của tiêu đề gói NAL trên Fig.13 được gọi là layer_id_plus1 hoặc nuh_reserved_zero_6bits. scalability_map[i] xác định loại khả năng điều chỉnh cho mỗi bit trong thành phần cú pháp layer_id_plus1. Trong trường hợp nào đó thành phần cú pháp layer_id_plus1 có thể còn được gọi là thành phần cú pháp nuh_reserved_zero_6bits hoặc rserved_zero_6bits . Ánh xạ khả năng điều chỉnh đối với tất cả các bit của thành phần cú pháp layer_id_plus1 cùng xác định khả năng điều chỉnh đang được sử dụng trong dãy viđeo được mã hóa. Trị số thực của nhãn hiệu đối với mỗi loại khả năng điều chỉnh được truyền tín hiệu thông qua các bit tương ứng này trong trường layer_id_plus1 (nuh_reserved_zero_6bits) trong tiêu đề gói NAL. Khi scalability_map[i] bằng 0 đối với mọi trị số i, thì dãy viđeo được mã hóa phù hợp với chuẩn cơ sở, do vậy trị số layer_id_plus1 của các gói NAL bằng 0 và không có gói NAL nào thuộc lớp hoặc hình nâng cao. vps_layer_id[i] xác định trị số 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 các 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 trị số cao hơn của scalability_map[i] được trình bày như được thể hiện trên Fig.18. Ánh xạ khả năng điều chỉnh [i] bao gồm kích thước của khả năng điều chỉnh (0) không có gì; (1) không gian; (2) chất lượng; (3) độ sâu; (4) đa hình; (5) không xác định; (6) dự trữ; và (7) dự trữ.

Do đó mỗi bit trong tiêu đề gói NAL được diễn giải dựa trên 3 bit trong tập hợp thông số viđeo về kích thước khả năng điều chỉnh là gì (ví dụ, không có gì, không gian, chất lượng, độ sâu, đa hình, không xác định, dự trữ). Ví dụ, để truyền tín hiệu rằng tất cả các bit trong layer_id_plus1 tương ứng với khả năng điều chỉnh không gian, các trị số scalability_map trong VPS có thể được mã hóa là 001 001 001 001 001 001 cho 6 bit của tiêu đề gói NAL. Ngoài ra ví dụ, để truyền tín hiệu rằng 3 bit trong layer_id_plus1 tương ứng với khả năng điều chỉnh không gian và 3 bit tương ứng với khả năng điều chỉnh chất lượng, các trị số scalability_map trong VPS có thể được mã hóa là 001 001 001 010 010 010 cho 6 bit của tiêu đề gói NAL.

Trên Fig.19, phương án khác bao gồm tập hợp thông số viđeo truyền tín hiệu số lượng kích thước của khả năng điều chỉnh trong 6 bit của tiêu đề gói NAL bằng cách sử dụng num_scalability_dimensions_minus1. num_scalability_dimensions_minus1 cộng 1 chỉ báo số lượng kích thước của khả năng điều chỉnh được truyền tín hiệu thông qua các thành phần 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 liên quan đến Fig.17. num_bits_for_scalability_map[i] xác định độ dài theo bit đối với kích thước của khả năng điều chỉnh thứ i. Tổng của tất cả num_bits_for_scalability_map[i] đối với i=0, ..num_scalability_dimensions_minus1 bằng sáu (hoặc khác đi bằng số lượng bit được sử dụng cho các thành phần cú pháp layer_id_plus1; vps_reserved_zero_6bits; max_num_layers_minus1; reserved_zero_6bits; nuh_reserved_zero_6bits).

Trên các hình vẽ Fig.17 và Fig.19 là 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 trị số cao hơn của scalability_map[i] có thể được xác định là dự trữ cho các dòng bit phù hợp với hồ sơ cụ thể của kỹ thuật viđeo. Ví dụ, ví dụ các trị số ánh xạ khả năng điều chỉnh 6..15 có thể được xác định là ‘‘dự trữ” khi truyền tín hiệu scalability_map[i] với u(4). Ví dụ theo phương án khác, scalability_map[i] có thể được truyền tín hiệu với ue(v) hoặc biểu đồ mã hóa khác nào đó. Ví dụ theo phương án khác, hạn chế có thể được xác định sao cho các trị số scalability_map[i] được sắp xếp theo thứ tự đơn điệu không giảm (hoặc không tăng). Việc này tạo ra một số trường kích thước của khả năng điều chỉnh trong trường layer_id_plus1 trong tiêu đề gói NAL có liên quan.

Một kỹ thuật hiện có khác đối với truyền tín hiệu mã hóa viđeo có thể điều chỉnh được bằng cách sử dụng thành phần cú pháp "layer_id_plus1" hoặc "nuh_reserved_zero_6bits" ánh xạ layer_id_plus1 trong tiêu đề gói NAL đến định danh lớp bằng cách truyền tín hiệu bảng dò tìm chung trong tập hợp thông số viđeo. Trên Fig.20, kỹ thuật hiện có này bao gồm tập hợp tham số viđeo để xác định số lượng các loại kích thước và các định danh kích thước đối với lớp thứ i của bảng dò tìm đó. Cụ thể, vps_extension_byte_alignment_reserved_zero_bit là zero. num_dimensions_minus1[i] cộng 1 xác định số lượng các loại kích thước (dimension_type[i][j]) và các nhãn hiệu kích thước (dimension_id[i][j]) đối với lớp thứ i. dimension_type[i][j] xác định loại kích thước của khả năng điều chỉnh thứ j của lớp thứ i, mà nó có layer_id hoặc layer_id_plus1 bằng i, như được xác định trên Fig.31. Như được minh họa trên Fig.21, các kích thước này được xác định bao gồm (0) idx thứ tự hình; (1) cờ hiệu độ sâu; (2) ID phụ thuộc; (3) ID chất lượng; (4)-(15) dự trữ. dimension_id[i][j] xác định nhãn hiệu loại kích thước của khả năng điều chỉnh thứ j của lớp thứ i, mà nó khi không hiện diện thì được kết luận bằng 0. num_direct_ref_layers[i] xác định số lượng các 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 minh họa trên Fig.20 tạo ra bảng dò tìm to không nhìn thấy được.

Trên Fig.22, một mở rông tập hợp thông số viđeo được cải biến bao gồm mặt nạ của khả năng điều chỉnh mà nó được sử dụng kết hợp với kích thước của khả năng điều chỉnh. scalability_mask truyền tín hiệu sơ đồ gồm các bit 0 và 1 với mỗi bit tương ứng với một kích thước của khả năng điều chỉnh như được chỉ báo bởi cú pháp ánh xạ khả năng điều chỉnh trên Fig.23. Trị số 1 đối với kích thước của khả năng điều chỉnh cụ thể chỉ báo rằng kích thước của khả năng điều chỉnh này hiện diện trong lớp này (lớp thứ I). Trị số 0 đối với kích thước của khả năng điều chỉnh cụ thể chỉ báo rằng kích thước của khả năng điều chỉnh này không hiện diện trong lớp này (lớp thứ I). Ví dụ, tập hợp các bit 00100000 liên quan đến khả năng điều chỉnh chất lượng. Trị số nhãn hiệu thực tế của kích thước của khả năng điều chỉnh cụ thể mà nó hiện diện được chỉ báo bởi trị số scalability_id[j] được truyền tín hiệu. Các trị số num_scalability_types[i] bằng tổng của số lượng các bit trong scalability_mask có trị số là 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 trị số nhãn hiệu của kích thước của khả năng điều chỉnh thứ j đối với loại của các trị số của khả năng điều chỉnh mà chúng được truyền tín hiệu nhờ trị số scalability_mask.

Trên Fig.24, một cải biến của Fig.22, bao gồm mặt nạ của khả năng điều chỉnh được truyền tín hiệu ngoài vòng. Việc này tạo ra một mặt nạ chung cho mỗi lần định danh lớp. Trên Fig.25, trong cải biến này tập hợp thông số viđeo tương ứng có tính ví dụ có thể bao gồm việc định danh có thể điều chỉnh được với mặt nạ của khả năng điều chỉnh hiện không được bao gồm. Trong trường hợp này thành phần cú pháp scalable_id[j] có cùng biểu hiện là thành phần cú pháp scalability_id[j] trên Fig.22.

Trên Fig.26 một cải biến của Fig.22 bao gồm mặt nạ của khả năng điều chỉnh (scalability_mask) được truyền tín hiệu ngoài vòng. Việc này tạo ra một mặt nạ chung cho mỗi lần định danh lớp. scalability_mask truyền tín hiệu sơ đồ gồm các bit 0 và 1 với mỗi bit tương ứng với một kích thước của khả năng điều chỉnh như được chỉ ra bởicú

pháp ánh xạ khả năng điều chỉnh trên Fig.27. Trị số 1 đối với kích thước của khả năng điều chỉnh cụ thể chỉ báo rằng kích thước của khả năng điều chỉnh này có hiện diện trong lớp này (lớp thứ I). Trị số 0 đối với kích thước của khả năng điều chỉnh cụ thể chỉ báo rằng kích thước của khả năng điều chỉnh này không hiện diện trong lớp này (lớp thứ I). Ví dụ, tập hợp các bit 00100000 liên quan đến khả năng điều chỉnh chất lượng. Trị số nhãn hiệu thực tế của kích thước của khả năng điều chỉnh cụ thể mà nó hiện diện được chỉ báo bằng trị số scalability_id[j] được truyền tín hiệu. Các trị số của num_scalability_types[i] bằng tổng của số lượng các bit trong scalability_mask có trị số là 1. Do vậy

Biểu thức 10

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

Trong trường hợp này biến scalability_id[j] còn được gọi là biến dimension_id[i][j]. dimension_id[i][j] xác định nhãn hiệu của khả năng điều chỉnh của kích thước của khả năng điều chỉnh thứ j của lớp thứ i. Sau đó biến ScalabilityId[i][j] được rút 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 khả năng điều chỉnh 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 điều chỉnh không gian cho lớp thứ i, QualityId[i][2] là ID chất lượng đối với kích thước khả năng điều chỉnh chất lượng đối với lớp thứ i, depthFlag[i][3] là cờ hiệu độ sâu / ID độ sâu đối với kích thước khả năng điều chỉnh độ sâu cho lớp thứ i, và ViewId[i][4] là ID hình đối với kích thước khả năng điều chỉnh đa hình cho lớp thứ i.

Ngoài ra 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] mà nó truyền tín hiệu trị số layer_id trong tiêu đề gói NAL được truyền tín hiệu.

Theo phương án khác một hoặc nhiều thành phần cú pháp scalability_mask[i], scalability_mask, scalability_id[j] có thể được truyền tín hiệu bằng cách sử dụng số lượng các 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 phương án khác một hoặc nhiều thành phần cú pháp có thể được truyền tín hiệu với ue(v). Theo phương án khác scalability_mask có thể được truyền tín hiệu trong tiêu đề gói NAL trong các thành phần 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 gói NAL VPS, hoặc chỉ cho các gói NAL không VPS, hoặc cho tất cả các gói NAL. Theo phương án khác nữa scalability_mask có thể được truyền tín hiệu từng hình ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ nó có thể được truyền tín hiệu trong tiêu đề phiến, tập hợp thông số hình ảnh, tập hợp thông số viđeo, hoặc bất kỳ tập hợp thông số khác hoặc bất kỳ phần quy chuẩn 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 thành phần cú pháp nuh_reserved_zero_6bits hoặc layer_id_plus1 trong tiêu đề gói NAL trên Fig.13 có 6 bit. Tuy nhiên tất cả các mô tả nêu trên có thể được cải biến phù hợp nếu thành phần cú pháp đó sử dụng số lượng bit khác với 6 bit. Ví dụ nếu thành phần cú pháp đó (nuh_reserved_zero_6bits hoặc layer_id_plus1) sử dụng 9 bit thì sau đó trên Fig.17 trị số bit max_num_layer_minus1 sẽ là 9 và scalability_map[i] sẽ được truyền tín hiệu cho từng 9 bit thay vì 6 bit.

Trên Fig.24 một cải biến của Fig.22 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Thành phần cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm các bit 0 và 1 với độ dài bằng vps_max_layers_minus1. Trị số 0 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp độc lập. Trị số 1 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp phụ thuộc mà nó phụ thuộc vào một hoặc nhiều lớp khác.

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

Biểu thức 11

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

Trên Fig.29 một cải biến của Fig.26 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Thành phần cú pháp mới layer_dependency_flag[i] được xác định. layer_dependency_flag[i] đánh tín hiệu nếu một lớp phụ thuộc vào các lớp khác. Trị số 0 đối với cờ hiệu chỉ báo rằng lớp với layer_id i là lớp độc lập. Trị số 1 cho bit thứ i chỉ báo rằng lớp với layer_id i là lớp phụ thuộc.

Trên Fig.30 một cải biến của Fig.26 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Thành phần cú pháp mới layer_dependency_map[i] được xác định.

layer_dependency_map[i] truyền tín hiệu sơ đồ gồm các bit 0 và 1 với độ dài bằng vps_max_layers_minus1. Trị số 0 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i không phụ thuộc vào lớp với layer_id (k+1). Trị số 1 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i phụ thuộc vào lớp với layer_id (k+1).

Trên Fig.31 một cải biến của Fig.26 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Thành phần cú pháp mới layer_dependency_information_pattern được xác định.

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

Biểu thức 12

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

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

Trên Fig.32 một cải biến của Fig.26 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Fig.28 là cú pháp biến thể dựa trên cú pháp trên Fig.27. Thành phần cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm các bit 0 và 1 với độ dài bằng vps_max_layers_minus1. Trị số 0 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp độc lập. Trị số 1 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp phụ thuộc mà nó phụ thuộc vào một hoặc nhiều lớp khác.

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

Biểu thức 13

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

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

Trên Fig.33 một cải biến của Fig.26 đưa ra cú pháp để đánh tín hiệu thông tin phụ thuộc lớp. Fig.29 là cú pháp biến thể dựa trên cú pháp trên Fig.31. Thành phần cú pháp mới layer_dependency_information_pattern được xác định.

layer_dependency_information_pattern truyền tín hiệu sơ đồ gồm các bit 0 và 1 với độ dài bằng vps_max_layers_minus1. Trị số 0 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp độc lập. Trị số 1 cho bit thứ i chỉ báo rằng lớp với layer_id (i+1) là lớp phụ thuộc mà nó phụ thuộc vào một hoặc nhiều lớp khác.

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

Biểu thức 14

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

layer_dependency_map[i] truyền tín hiệu sơ đồ gồm các bit 0 và 1 với độ dài bằng vps_max_layers_minus1. Trị số 0 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i không phụ thuộc vào lớp với layer_id (k+1). Trị số 1 cho bit thứ k layer_dependency_map[i] chỉ báo rằng lớp i phụ thuộc vào lớp with layer_id (k+1). Các thành phần cú pháp layer_dependency_map[i] được truyền tín hiệu chỉ khi layer_dependency_information_pattern(i) có trị số 1. Trong đó

layer_depdndency_information_pattern(i) là là bit thứ i của thành phần cú pháp layer_dependency_pattern.

Theo phương án khác thành phần cú pháp layer_dependency_information_pattern có thể được truyền tín hiệu dưới dạng tập hợp của các trị số cờ hiệu 1 bit. Trong trường hợp này tổng các trị số 1 bit vps_max_layers_minus1 sẽ được đánh tín hiệu là:

Bảng 9

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

Theo phương án khác layer_dependency_map[i] thành phần cú pháp có thể được truyền tín hiệu dưới dạng tập hợp của các trị số cờ hiệu 1 bit. Trong trường hợp này tổng các trị số 1 bit vps_max_layers_minus1 sẽ được đánh tín hiệu là:

Bảng 10

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

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

Theo phương án khác một hoặc nhiều trong số hoặc nhiều các thành phần cú pháp layer_dependency_information_pattern, layer_dependency_map có thể được truyền tín hiệu với ue(v) hoặc biểu đồ mã hóa khác nào đó.

Theo phương án khác tên của các thành phần 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 thành phần cú pháp khác như layer_dependency_information_pattern, layer_dependency_map, layer_dependency_flag[i] v.v.. có thể được truyền tín hiệu từng hình ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ nó có thể được truyền tín hiệu trong tiêu đề phiến, pps/ sps/ vps/ aps hoặc bất kỳ tập hợp thông số khác hoặc phần quy chuẩn khác của dòng bit.

Như đã mô tả ở phần trên, mã hóa viđeo có thể điều chỉnh được là kỹ thuật mã hóa dòng bit viđeo mà nó cũng chứa một hoặc nhiều dòng bit của tập hợp phụ. Dòng bit viđeo tập hợp phụ có thể nhận được bằng cách dẹp bỏ các gói từ viđeo 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 viđeo chất lượng thấp. Ví dụ, dòng bit viđeo 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, et al., "Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)" JCTVC-L0453, Shanghai, October 2012, được kết hợp ở đây bằng cách tham khảo toàn bộ trong bản mô tả này. Chen, et al., "SHVC Draft Text 1," JCTVC-L1008, Geneva, March, 2013, được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này. Wang, et al., "AHG9: On VPS and SPS in HEVC 3DV and scalable extensions," JCTVC-M0268, Incheon, April 2013 được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này.

Như đã mô tả ở phần trên, mã hóa viđeo đa hình ảnh là kỹ thuật mã hóa dòng bit viđeo mà nó cũng chứa một hoặc nhiều các dòng bit khác đại diện cho các hình thay thế. Ví dụ, đa hình có thể là cặp hình đối với viđeo ba chiều. Ví dụ, đa hình có thể đại diện cho đa hình của cùng một cảnh từ nhiều điểm quan sát khác nhau. Đa hình nhìn chung có chứa số 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 hình ảnh 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 bên cạnh. Hannuksela, et al., "Common specification text for scalable and multi-view extensions," JCTVC-L0452, Geneva, January 2013, được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này. Tech, et. al. "MV-HEVC Draft Text 3 (ISO/IEC 23008-2:201x/PDAM2)," JCT3V-C1004_d3, Geneva, January 2013, được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này.

Chen, et al., "SHVC Draft Text 1," JCTVC-L1008, Geneva, January 2013; Hannuksela, et al. "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; từng tài liệu này được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này. JCTVC-L0452 và JCTVC-L0453 mỗi tài liệu đều có khái niệm các điểm hoạt động đầu ra. Khái niệm này sau đó được đổi thành khái niệm các tập hợp lớp ra trong JCTVC-L1008. Đối với các thành phần cú pháp num_output_operation_points, output_op_point_index[], và output_layer_flag[][] được xác định dối với các điểm hoạt động đầu ra đối với JCTVC-L0453 và LCTVC-L0452, và được cải biến 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 gói truy nhập hiện thời có chứa thông báo SEI.

Chen, et al., "SHVC Draft Text 1," JCTVC-L1008, Geneva, January 2013; Hannuksela, et al. "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; mỗi tài liệu này được kết hợp bằng cách tham khảo toàn bộ trong bản mô tả này, mỗi tài liệu đều bộ đệm hình ảnh được giải mã (DPB) của lệnh kết xuất mà nó hoạt động dựa trên việc sử dụng các thành phần 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 loại bỏ các hình ảnh 0 từ DPB. Thông tin này được truyền tín hiệu trong tập hợp thông số viđeo dùng cho lớp cơ sở, mà nó cung cấp tạo ra thông tin đệm cho nội dung viđeo bao gồm các lớp nâng cao, nếu có.

Trên Fig.34, ví dụ khác về tập hợp tham số viđeo được minh họa. Cụ thể, tập hợp thông số viđeo trên Fig.34 bao gồm tham chiếu đến phần mở rộng tập hợp thông số viđeo 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.

Trên Fig.35, ví dụ khác về phần mở rộng tập hợp thông số viđeo được minh họa. Cụ thể, phần mở rộng tập hợp thông số viđeo trên Fig.35 bao gồm tham chiếu đến num_output_layer_sets, mà nó xác định những lớp của tập hợp các lớp của dòng bit mà nó có thể là đầu ra cho người xem nhờ bộ giải mã. num_output_layer_sets xác định số lượng các tập hợp lớp mà đối với nó các lớp đầu ra được xác định với output_layer_set_index[i] và output_layer_flag[lsIdx][j]. Khi không hiện diện, trị số num_output_layer_sets được kết luận 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 hiện diện.

output_layer_flag[lsIdx][j] bằng 1 xác định rằng lớp với nuh_layer_id bằng j là lớp đầu ra mục tiêu của tập hợp lớp thứ lsIdx. Trị số output_layer_flag[lsIdx][j] bằng 0 xác định rằng lớp với nuh_layer_id bằng j không phải là lớp đầu ra mục tiêu 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 hợp thông số viđeo mới đi kèm với phần mở rộng tập hợp thông số viđeo tương ứng. Tiếc là, việc gửi tập hợp thông số viđeo mới đi kèm với phần mở rộng tập hợp thông số viđeo tương ứng

gây ra việc thuyên giảm mạnh về băng tần khả dụng. Ngoài ra tập hợp thông số viđeo mới có thể chỉ được kích hoạt tại một số loại hình ảnh nhất định, ví dụ tại các loại hình 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 ra xác định các lớp cụ thể của tập hợp lớp mà nó có thể được cung cấp làm đầu ra. Ví dụ, tập hợp lớp ra 1 có thể là lớp 0 là đầu ra và lớp 1 không phải là đầu ra. Ví dụ, tập hợp lớp ra 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 ra 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 ra 4 có thể là lớp 0 là đầu ra và lớp 4 không phải là đầu ra. Số lượng giới hạn các lớp ra 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 ra. Ví dụ sự thay đổi phụ thuộc lớp có thể cần một số ít lớp phải được giải mã để giải mã lớp mục tiêu 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 ra mà không phải gửi tập hợp thông số viđeo mới.

Trên Fig.36, cú pháp thông báo SEI có thể được sử dụng để hỗ trợ các thay đổi của các tập hợp lớp ra, gọi là, output_layer_sets_change(payloadSize). Thông báo SEI này chỉ báo rằng các thay đổi thông tin của các tập hợp lớp ra bắt đầu với gói truy nhập hiện thời có chứa thông báo SEI và được diễn giải liên quan đến tập hợp thông số viđeo hoạt động. Hơn một tập hợp thông số viđeo có thể được bao gồm với dòng bit, và tập hợp

thông số viđeo hoạt động sẽ được thay đổi, nếu cần. Khi hiện diện, thông báo SEI thay đổi của các tập hợp lớp ra áp dụng cho gói truy nhập mục tiêu mà nó bao gồm gói truy nhập hiện thời và tất cả các gói truy nhập theo sau, trong thứ tự mã hóa, cho tới khi có thông báo SEI thay đổi của các tập hợp lớp ra tiếp theo hoặc kết thúc của dãy viđeo được mã hóa, tùy cái nào xảy ra sớm hơn trong thứ tự mã hóa. Tốt hơn nếu các thông báo SEI thay đổi của các tập hợp lớp ra không có tác động cộng dồn.

'active_vps_id' xác định tập hợp thông số viđeo hoạt động có chứa thông tin các tập hợp lớp ra mà vì dó thay đổi hoặc bổ sung được áp dụng. Trị số active_vps_id bằng trị số video_parameter_set_id của tập hợp thông số viđeo hoạt động đối với các gói NAL VCL của gói truy nhập có chứa thông báo SEI.

'num_changed_output_layer_sets' xác định số lượng các tập hợp lớp ra thay đổi mà đối với nó các lớp ra được xác định trong đoạn mở rộng của tập hợp thông số viđeo hoạt động được xác định bởi active_vps_id. Trị số num_changed_output_layer_sets phải nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số cuối. Khi không hiện diện, trị số num_changed_output_layer_sets được kết luận bằng 0.

'changed_output_ layer_sets_idx_entry'[i] xác định mục nhập chỉ số trong danh sách của các mục nhập tập hợp lớp ra trong phần mở rộng vps được xác định bởi active_vps_id mà mà thay đổi output_layer_flag[clsIdx][j] áp dụng cho nó. Trị số changed_output_layer_set_idx_entry[i] phải nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số 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 với nuh_layer_id bằng j là lớp đầu ra mục tiêu của tập hợp lớp thứ clsIdx. Trị số output_layer_flag[clsIdx][j] bằng 0 xác định rằng lớp với nuh_layer_id bằng j không phải là lớp đầu ra mục tiêu của tập hợp lớp thứ clsIdx đó.

Khi output_layer_flag[clsIdx][j] không hiện diện đối với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm

trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, output_layer_flag[clsIdx][j] được kết luận bằng (j = = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

'num_addl_output_layer_sets' xác định số lượng các tập hợp lớp bổ sung mà đối với nó các lớp đầu ra được xác định với addl_output_layer_set_idx[i] và output_layer_flag[addllsIdx][j]. Khi không hiện diện, trị số num_addl_output_layer_sets được kết luận 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 hiện diện.

'output_layer_flag'[addllsIdx][j] bằng 1 xác định rằng lớp với nuh_layer_id bằng j là lớp đầu ra mục tiêu của tập hợp lớp thứ addllsIdx. Trị số output_layer_flag[addllsIdx][j] bằng 0 xác định rằng lớp với nuh_layer_id bằng j không phải là lớp đầu ra mục tiêu của tập hợp lớp thứ addllsIdx.

Khi output_layer_flag[addllsIdx][j] không hiện diện đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, thì output_layer_flag[addllsIdx][j] được kết luận bằng (j = = LayerSetLayerIdList[addllsIdx][NumLayersInIdList[addllsIdx] - 1]).

Khi output_layer_flag[addllsIdx][j] không hiện diện đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, và khi output_layer_flag[clsIdx][j] không hiện diện đối với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, thì output_layer_flag[addllsIdx][j] được kết luận bằng (j = = LayerSetLayerIdList[addllsIdx][NumLayersInIdList[addllsIdx] - 1]).

Khi output_layer_flag[addllsIdx][j] không hiện diện đối với addllsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, và khi output_layer_flag[clsIdx][

j] không hiện diện đôi với clsIdx nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối và đối với j nằm trong khoảng từ 0 đến 63, bao gồm cả trị số cuối, thì output_layer_flag[clsIdx][j] được kết luận bằng (j = = LayerSetLayerIdList[clsIdx][NumLayersInIdList[clsIdx] - 1]).

Danh sách mục tiêu của các lớp ra có thể nhận được, nếu muốn. Danh sách nhãn hiệu của lớp ra đích, targetOpLayerIdList, xác định danh sách cửtrị số nuh_layer_ids, theo thứ tự tăng lên của các trị số nuh_layer_id mà chúng là các lớp ra đối với các tập hợp lớp ra được lựa chọn có thể nhận được từ cách sau:

Đối với tập hợp lớp ra được lựa chọn oLsIdx, danh sách nhãn hiệu của lớp ra targetOpLayerIdList nhận được 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 ra mục tiêu có thể được xác định bởi danh sách nhãn hiệu lớp ra mục tiêu được liên kết targetOpLayerIdList.

Tập hợp lớp ra được lựa 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 ra.

Tập hợp lớp ra được lựa 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 ra hiện diện.

Danh sách mục tiêu của các lớp được giải mã có thể nhận được, nếu muốn. Danh sách nhãn hiệu lớp được giải mã đích, targetDLayerIdList, xác định danh sách của các trị số nuh_layer_id, mà chúng cần được giải mã đối với các tập hợp lớp ra được lựa chọn có thể được suy ra như sau:

Đối với tập hợp lớp ra được lựa chọn oLsIdx, danh sách nhãn hiệu của lớp ra targetOpLayerIdList nhận được 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 sách nhãn hiệu lớp được giải mã mục tiêu 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 trường hợp nào đó targetDLayerIdList cuối cùng có thể thu được bằng cách phân loại targetDLayerIdList nêu trên theo thứ tự tăng lên của các trị số nuh_layer_id.

Tập hợp nhãn hiệu lớp được giải mã mục tiêu được xác định bởi danh sách nhãn hiệu lớp được giải mã mục tiêu được liên kết targetDLayerIdList. Trong trường hợp nào đó targetDLayerIdList này có thể giống với danh sách nhãn hiệu lớp TargetDecLayerIdList từ JCTVC-L1008, mà nó xác định danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id, của các gói NAL sẽ phải được giải mã:

Tập hợp lớp ra được lựa 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 ra.

Tập hợp lớp ra được lựa 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 ra có hiện diện.

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

'num_deleted_output_layer_sets' xác định số lượng các tập hợp lớp ra mà chúng được xóa và không còn hiện diện. Trị số num_deleted_output_layer_sets nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số cuối. Khi không hiện diện, thì trị số num_deleted_output_layer_sets được kết luận bằng 0.

'deleted_output_layer_sets_idx_entry'[i] xác định mục nhập chỉ số trong danh sách của các mục nhập tập hợp lớp ra mà nó được chỉ báo phải được xóa và không còn hiện diện. Trị số deleted_output_layer_set_idx_entry[i] phải nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số 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 các tập hợp lớp ra thay đổi mà đối với nó các lớp ra được xác định theo đoạn mở rộng vps của tập hợp thông số viđeo hoạt động được xác định bởi active_vps_id. Trị số num_changed_output_layer_sets phải nằm trong khoảng từ 0 đến num_output_layer_sets - num_deleted_output_layer_sets, bao gồm cả trị số cuối. Khi không hiện diện, trị số num_changed_output_layer_sets được kết luận bằng 0.

Danh sách mục tiêu của các lớp ra có thể nhận được, nếu muốn. Danh sách nhãn hiệu của lớp ra đích, targetOpLayerIdList, xác định danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id mà chúng là các lớp ra đối với các tập hợp lớp ra được lựa chọn có thể được suy ra như sau:

Đối với tập hợp lớp ra được lựa chọn oLsIdx, danh sách nhãn hiệu của lớp ra targetOpLayerIdList nhận được là:

Bảng 14

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 ra mục tiêu được xác định bởi danh sách nhãn hiệu lớp ra mục tiêu được liên kết targetOpLayerIdList.

Tập hợp lớp ra được lựa 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 ra.

Tập hợp lớp ra được lựa 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 ra có hiện diện.

Danh sách mục tiêu của các lớp được giải mã có thể nhận được, nếu muốn.

Danh sách nhãn hiệu lớp được giải mã đích, TargetDLayerIdList, xác định danh sách của các trị số nuh_layer_id, mà chúng cần được giải mã đối với các tập hợp lớp ra được lựa chọn có thể được suy ra như sau:

Đối với tập hợp lớp ra được lựa chọn oLsIdx, danh sách nhãn hiệu của lớp ra targetOpLayerIdList có thể được suy ra là:

Bảng 15

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

Sau đó danh sách nhãn hiệu lớp được giải mã mục tiêu targetDLayerIdList có thể được suy ra là:

Bảng 16

```
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 trường hợp nào đó targetDLayerIdList cuối cùng có thể thu được bằng cách phân loại targetDLayerIdList nêu trên theo thứ tự tăng lên của các trị số nuh_layer_id.

Tập hợp nhãn hiệu lớp được giải mã mục tiêu được xác định bởi danh sách nhãn hiệu lớp được giải mã mục tiêu được liên kết targetDLayerIdList. Trong trường hợp nào đó targetDLayerIdList này có thể giống với danh sách nhãn hiệu lớp TargetDecLayerIdList từ JCTVC-L1008, mà nó xác định danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id, của các gói NAL sẽ phải được giải mã:

Tập hợp lớp ra được lựa 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 ra.

Tập hợp lớp ra được lựa 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 ra có hiện diện.

Thông báo SEI thay đổi của các tập hợp lớp ra được xác định sao cho tác động của nó không có tính cộng dồn. Theo phương án 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 (from 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 phương án khác một hoặc nhiều thành phần cú pháp có thể được truyền tín hiệu bằng cách sử dụng số lượng các bit cố định đã biết thay cho u(v) thay cho ue(v). Ví dụ chúng có thể được truyền tín hiệu bằng cách sử dụng u(8) hoặc u(16) hoặc u(32) hoặc u(64) v.v...

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

Theo phương án khác tên của các thành phần 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 thành phần cú pháp khác được bao gồm trong thông báo SEI của các tập hợp lớp ra có thể được truyền tín hiệu từng hình 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 phiến, tập hợp thông số tương thích pps/ sps/ vps/ hoặc bất kỳ tập hợp thông số khác hoặc phần quy chuẩn khác của dòng bit.

Theo phương án khác nữa các thành phần cú pháp khác có thể được truyền tín hiệu từng hình 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 phiến, tập hợp thông số tương thích pps/ sps/ vps/ hoặc bất kỳ tập hợp thông số khác hoặc phần quy chuẩn khác của dòng bit.

Theo các phương án khác nữa tất cả các khái niệm được xác định trong sáng chế liên quan đến các tập hợp lớp ra 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 về sáng chế được mô tả dưới đây. Lưu ý là, để tiện lợi, các bộ phận có các chức năng giống với chức năng của các bộ phận tương ứng được minh họa trong Ví dụ thứ nhất được trình bày bằng các số dẫn chiếu giống nhau, và 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 được giải mã (DPB) của thứ tự kết xuất dựa trên việc sử dụng các thành phần 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 loại bỏ các hình ảnh từ DPB không ghi chép các đặc điểm của bộ đệm mà có thể phát sinh từ việc mã hóa viđeo có thể điều chỉnh được, chẳng hạn như khi các số lượng khác nhau của các lớp nâng cao được sử dụng mà chúng có khuynh hướng thay đổi sau khi nội dung đã được mã hóa dựa vào các ưu tiên kiểm tra hình ảnh của người sử dụng, và các lớp nâng cao đa hình mà chúng có khuynh hướng thay đổi sau khi nội dung đã được mã hóa dựa vào các ưu tiên kiểm tra hình ảnh của người sử dụng. Cũng đã xác định rằng việc truyền tín hiệu cho bộ đệm hình ảnh được giải mã (DPB) của thứ tự kết xuất dựa trên bằng cách sử dụng các thành phần 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 loại bỏ các hình ảnh từ 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 ra được lựa chọn. Để tạo điều kiện cho các khác biệt như vậy trong các ưu tiên kiểm tra hình ảnh, bộ đệm hình ảnh được giải mã (DPB) của thứ tự kết xuất có thể còn và/hoặc theo phương án khác phải dựa vào các thành phần cú pháp như vậy được bao gồm cùng với phần mở rộng tập hợp thông số viđeo (phần mở rộng VPS) để tạo ra các thành phần cú pháp cho một hoặc nhiều lớp nâng cao. Theo cách này các thành phần cú pháp có thể được lựa chọn để đặc biệt phù hợp đối với điểm hoạt động cụ thể hoặc tập hợp lớp ra, mà nó có khuynh hướng tương ứng với các ưu tiên kiểm tra hình ảnh của người sử dụng.

Các thông 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 ra 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 thông số đệm DPB được truyền tín hiệu nêu trên từ phần mở rộng VPS nếu chúng được truyền tín hiệu cho điểm hoạt động đang xét nghiệm hoặc đối với các tập hợp lớp ra được lựa chọn. Khác đi các thông số cấp SPS tương ứng từ SPS hoạt động (khi currLayerId mà nó 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 thông 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 thông 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 các tập hợp lớp ra trong phần mở rộng VPS. Các tập hợp lớp ra có thể tương ứng với các điểm hoạt động. Các thông 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 thông số đệm DPB được truyền tín hiệu nêu trên từ phần mở rộng VPS nếu chúng được truyền tín hiệu cho điểm hoạt động đang xét nghiệm hoặc đối với các tập hợp lớp ra được lựa chọn. Khác đi các thông số cấp SPS tương ứng từ SPS hoạt động (khi currLayerId mà nó 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.

Trên Fig.38A, vps_extension được cải biến có tính 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 cải biến này có thể được xác định về mặt điểm hoạt động mà nó là dòng bit được tạo ra từ dòng bit khác nhờ hoạt động của quy trình trích dòng bit phụ với dòng bit khác, TemporalId cao nhất mục tiêu, và danh sách nhãn hiệu lớp mục tiêu dưới dạng các dữ liệu đầu vào.

num_output_layer_sets xác định số lượng các tập hợp lớp mà đối với nó các lớp đầu ra được xác định với output_layer_set_index[i] và output_layer_flag[lsIdx][j]. Khi không hiện diện, thì trị số num_output_layer_sets được kết luận 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 hiện diện.

output_layer_flag[lsIdx][j] bằng 1 xác định rằng lớp với nuh_layer_id bằng j là lớp đầu ra mục tiêu của tập hợp lớp thứ lsIdx. Trị số output_layer_flag[lsIdx][j] bằng 0 xác định rằng lớp với nuh_layer_id bằng j không phải là lớp đầu ra mục tiêu của tập hợp lớp thứ lsIdx.

num_op_dpb_info_parameters xác định số lượng các kết cấu cú pháp op_dpb_parameters( ) hiện diện trong RBSP của phần mở rộng VPS, được xác định về mặt 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 trị số cuối.

operation_point_layer_set_idx[i] xác định chỉ số, đến danh sách của các tập hợp lớp được xác định bởi các điểm hoạt động mà kết cấu cú pháp op_dpb_info_parameters( ) thứ i trong phần mở rộng VPS tác động đến. Trị số operation_point_layer_set_idx[i] có thể nằm trong khoảng từ 0 đến vps_num_layer_sets_minus1, bao gồm cả trị số cuối. Để có được sự 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.

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] cộng 1 chỉ báo có bao nhiêu lớp phụ được bao gồm. vps_max_sub_layers_minus1[j] cộng 1 xác định số lượng tối đa của các lớp phụ tạm thời có thể hiện diện trong CVS cho lớp với nuh_layer_id bằng j. Trị số vps_max_sub_layers_minus1[j] nằm trong khoảng từ 0 đến 6, bao gồm cả trị số 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 từng 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 hiện diện cho lớp với 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 trị số 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 với nuh_layer_id bằng j.

vps_max_dec_pic_buffering_minus1[j][k] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS cho lớp với nuh_layer_id bằng j trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng k. Trị số vps_max_dec_pic_buffering_minus1[j][k] phải nằm trong khoảng từ 0 đến MaxDpbSize - 1 (như được xác định trong khoản A.4), bao gồm cả trị số cuối. Khi k lớn hơn 0, thì vps_max_dec_pic_buffering_minus1[j][k] phải lớn hơn hoặc bằng vps_max_dec_pic_buffering_minus1[j][k - 1]. Khi vps_max_dec_pic_buffering_minus1[j][k] không hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[j] bằng 0, thì nó được kết luận bằng vps_max_dec_pic_buffering_minus1[j] [vps_max_sub_layers_minus1[j]].

vps_max_num_reorder_pics[j][k] chỉ báo số lượng tối đa cho phép của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS cho lớp với nuh_layer_id bằng j

trong thứ tự mã hóa và tiếp theo sau hình ảnh đó trong thứ tự kết xuất khi HighestTid bằng k. Trị số vps_max_num_reorder_pics[j][k] phải nằm trong khoảng từ 0 đến vps_max_dec_pic_buffering_minus1[j][k], bao gồm cả trị số 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 hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[j] bằng 0, thì nó được kết luận bằng 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 trị số VpsMaxLatencyPictures[j][k], mà nó xác định số lượng tối đa của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS cho lớp với nuh_layer_id bằng j trong thứ tự kết xuất và tiếp theo sau hình ảnh đó trong thứ tự mã hóa khi HighestTid bằng k.

Khi vps_max_latency_increase_plus1[j][k] không bằng 0, trị số 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 biểu diễn.

Trị số vps_max_latency_increase_plus1[j][k] nằm trong khoảng từ 0 đến 2^{32} - 2, bao gồm cả trị số cuối. Khi vps_max_latency_increase_plus1[j][k] không hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[j] bằng 0, thì nó được kết luận bằng vps_max_latency_increase_plus1[j][vps_max_sub_layers_minus1[j]].

'vps_max_sub_layers_minus1'[id] [j] cộng 1 xác định số lượng tối đa của các lớp phụ tạm thời mà nó có thể hiện diện trong CVS cho lớp với nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id. Trị số vps_max_sub_layers_minus1[id] [j] phải nằm trong khoảng từ 0 đến 6, bao gồm cả trị số 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 hiện diện cho lớp với 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 trị số 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 với nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id.

'vps_max_dec_pic_buffering_minus1'[id][j][k] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS cho lớp với nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng k. Trị số vps_max_dec_pic_buffering_minus1[id][j][k] phải nằm trong khoảng từ 0 đến MaxDpbSize - 1 (như được xác định trong khoản A.4), bao gồm cả trị số cuối. Khi k lớn hơn 0, thì vps_max_dec_pic_buffering_minus1[id][j][k] phải 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 hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[id][j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[id][j] bằng 0, thì nó được kết luận bằng 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 số lượng tối đa cho phép của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS cho lớp với nuh_layer_id bằng j cho điểm hoạt động kết hợp với chỉ số id trong thứ tự mã hóa và tiếp theo sau hình ảnh đó trong thứ tự kết xuất khi HighestTid bằng k. Trị số vps_max_num_reorder_pics[id][j][k] phải nằm trong khoảng từ 0 đến vps_max_dec_pic_buffering_minus1[id][j][k], bao gồm cả trị số cuối. Khi k lớn hơn 0, thì vps_max_num_reorder_pics[id][j][k

] phải lớn hơn hoặc bằng vps_max_num_reorder_pics[id][j][k- 1]. Khi vps_max_num_reorder_pics[id][j][k] không hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[id][j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[id][j] bằng 0, thì nó được kết luận bằng vps_max_num_reorder_pics[id][j][vps_max_sub_layers_minus1[id][j]].

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

Khi vps_max_latency_increase_plus1[id][j][k] không bằng 0, trị số VpsMaxLatencyPictures[id][j][k] được xác định như sau:

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

Khi vps_max_latency_increase_plus1[id][j][k] bằng 0, thì không có giới hạn tương ứng nào được biểu diễn.

Trị số vps_max_latency_increase_plus1[id][j][k] phải nằm trong khoảng từ 0 đến 2^{32} - 2, bao gồm cả trị số cuối. Khi vps_max_latency_increase_plus1[id][j][k] không hiện diện cho k nằm trong khoảng từ 0 đến vps_max_sub_layers_minus1[id][j] - 1, bao gồm cả trị số cuối, do vps_sub_layer_ordering_info_present_flag[id][j] bằng 0, thì nó được kết luận bằng vps_max_latency_increase_plus1[id][j][vps_max_sub_layers_minus1[id][j]].

Trên Fig.39 B, op_dpb_info_parameters có thể được cải biến như được thể hiện là op_dpb_info_parameters(id,j). Trong trường hợp này cú pháp của phần mở rộng VPS có thể được minh họa trên Fig.38B. Bộ giải mã tham chiếu giả thiết (HRD) được sử dụng để kiểm tra sự phù hợp của dòng bit và bộ giải mã. Có hai loại dòng bit hoặc tập hợp phụ của dòng bit phụ thuộc vào việc kiểm tra sự phù hợp HRD theo Nhóm hợp tác về mã hóa

viđeo (JCT-VC). Loại thứ nhất, gọi là dòng bit Loại I, là dòng gói NAL chỉ chứa các gói NAL VCL và các gói NALvới nal_unit_type bằng FD_NUT (các gói NAL dữ liệu bộ lọc) cho tất cả các gói truy nhập trong dòng bit. Loại thứ hai, gọi là dòng bit Loại II, có chứa, bên cạnh các gói NAL VCL và các gói NAL dữ liệu bộ lọc đó cho tất cả các gói truy nhập trong dòng bit, ít nhất một trong số (a) các gói NAL không VCL bổ sung khác với các gói NAL dữ liệu bộ lọc, và (b) tất cả các thành phần cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, và trailing_zero_8bits mà chúng tạo thành dòng byte từ dòng gói NAL.

Các thành phần cú pháp của các gói NAL không VCL (hoặc các trị số mặc định đối với một số thành phần 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 hợp thông số HRD (các thông số HRD NAL và các thông số HRD VCL) được sử dụng. Các tập hợp thông số HRD được truyền tín hiệu thông qua kết cấu cú pháp hrd_parameters( ), mà nó có thể là phần của kết cấu cú pháp SPS hoặc kết cấu cú pháp VPS.

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

(1) Điểm hoạt động đang xét nghiệm, ký hiệu là TargetOp, được lựa chọn. Danh sách nhãn hiệu lớp OpLayerIdList của TargetOp gồm danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id, hiện diện trong tập hợp phụ của dòng bit kết hợp với TargetOp, mà nó là tập hợp phụ của các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm. OpTid của TargetOp bằng TemporalId cao nhất hiện diện trong tập hợp phụ của dòng bit được liên kết với TargetOp.

[0448]

(2) TargetDecLayerIdList được cài đặt bằng OpLayerIdList của TargetOp, HighestTid được cài đặt bằng OpTid của TargetOp, và quy trình trích dòng bit phụ như được xác

định trong điều 10 được hủy bỏ với dòng bit đang xét nghiệm, HighestTid, và TargetDecLayerIdList dưới dạng các dữ liệu đầu vào, và đầu ra được chỉ định đến BitstreamToDecode.

(3) Kết cấu cú pháp hrd_parameters( ) và kết cấu cú pháp sub_layer_hrd_parameters( ) có thể áp dụng cho TargetOp được lựa chọn. Nếu TargetDecLayerIdList chứa tất cả các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm, thì kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) được lựa chọn. Khác đi, kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) mà nó áp dụng cho TargetOp được lựa chọn. Bên trong kết cấu cú pháp hrd_parameters( ) được lựa chọn, nếu BitstreamToDecode là dòng bit Loại I, thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo ngay sau điều kiện "if(vcl_hrd_parameters_present_flag)" được lựa chọn và biến NalHrdModeFlag được cài đặt bằng 0; khác đi (BitstreamToDecode là dòng bit Loại II), thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo sau hoặc điều kiện "if(vcl_hrd_parameters_present_flag)" (trong trường hợp này biến NalHrdModeFlag được cài đặt bằng 0) hoặc điều kiện "if(nal_hrd_parameters_present_flag)" (trong trường hợp này biến NalHrdModeFlag được cài đặt bằng 1) được lựa chọn. Khi BitstreamToDecode là dòng bit Loại II và NalHrdModeFlag bằng 0, tất cả các gói NAL không VCL trừ các gói NAL dữ liệu bộ lọc, và tất cả các thành phần cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, và trailing_zero_8bits mà chúng tạo thành dòng byte từ dòng gói NAL (như được xác định trong Phụ lục B), khi hiện diện, được vứt bỏ ra khỏi BitstreamToDecode, và và dòng bit còn lại được chỉ định đến BitstreamToDecode.

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

(1) Tập hợp lớp ra đang xét nghiệm, ký hiệu là TargetOpLs được lựa chọn. Điểm hoạt động được tham chiếu trong TargetOpLs nhờ output_layer_set_idx[] xác định điểm hoạt động đang xét nghiệm. Danh sách nhãn hiệu lớp ra OpLayerIdList của TargetOpLs gồm danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id, hiện diện trong tập hợp phụ của dòng bit kết hợp với TargetOp và TargetOpLs, mà nó là tập hợp phụ của các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm. OpTid của TargetOp bằng TemporalId cao nhất hiện diện trong tập hợp phụ của dòng bit kết hợp với TargetOp.

[0452]

(2) TargetDecLayerIdList được cài đặt bằng danh sách nhãn hiệu lớp được mã hóa mục tiêu targetDLayerIdList đối với các tập hợp lớp ra được lựa chọn TargetOpLs, HighestTid được cài đặt bằng OpTid của TargetOp, và quy trình trích dòng bit phụ như được xác định trong điều 10 được hủy bỏ với dòng bit đang xét nghiệm, HighestTid, và TargetDecLayerIdList dưới dạng các dữ liệu đầu vào, và đầu ra được chỉ định đến BitstreamToDecode.

(3) Kết cấu cú pháp hrd_parameters( ) và kết cấu cú pháp sub_layer_hrd_parameters( ) có thể áp dụng cho TargetOp được lựa chọn. Nếu TargetDecLayerIdList chứa tất cả các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm, kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) được lựa chọn. Khác đi, kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) mà nó áp dụng cho TargetOp được lựa chọn. Bên trong kết cấu cú pháp hrd_parameters( ) được lựa chọn, nếu BitstreamToDecode là dòng bit Loại I, thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo ngay sau điều kiện "if(vcl_hrd_parameters_present_flag)" được lựa chọn và biến NalHrdModeFlag được cài đặt bằng 0; khác đi (BitstreamToDecode là dòng bit Loại II), thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo ngay sau hoặc điều kiện "if(vcl_hrd_parameters_present_flag)" (trong trường hợp này

biến NalHrdModeFlag được cài đặt bằng 0) hoặc điều kiện "if(nal_hrd_parameters_present_flag)" (trong trường hợp này biến NalHrdModeFlag được cài đặt bằng 1) được lựa chọn. khi BitstreamToDecode là dòng bit Loại II và NalHrdModeFlag bằng 0, tất cả các gói NAL không VCL trừ các gói NAL dữ liệu bộ lọc, và tất cả các thành phần cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, và trailing_zero_8bits mà chúng tạo thành dòng byte từ dòng gói NAL (như được xác định trong Phụ lục B), khi hiện diện, được vứt bỏ kra khỏi BitstreamToDecode, và và dòng bit còn lại được chỉ định đến 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 sự phù hợp với hồ sơ, tầng và cấp độ cụ thể phải có khả năng giải mã thành công tất cả các dòng bit có phù hợp với các yêu cầu về phù hợp dòng bit được xác định trong khoản 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 gói NAL VCL, và khoảng thời gian 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 gói NAL không VCL), hoặc nhờ các thiết bị bên ngoài không được xác định trong bản mô tả này.

(2) Khi dòng bit chứa các thành phần cú pháp có các trị số mà chúng được xác định là dự trữ và nó được xác định rằng các bộ giải mã cần bỏ qua các trị số của các thành phần cú pháp hoặc Các gói NAL có chứa các thành phần cú pháp có các trị số dự trữ, 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 thành phần cú pháp hoặc các gói NAL có chứa các thành phần cú pháp có các trị số dự trữ 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 sự phù hợp của bộ giải mã, các dòng bit phù hợp với hồ sơ, tầng và cấp độ được đòi hỏi, như được xác định trong khoản C.4 được cung cấp bởi bộ lập lịch trình dòng giả thiết (HSS) cho cả HRD và bộ giải mã đang xét 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 hình ảnh với PicOutputFlag bằng 1, và, đối với mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT, các trị số của tất cả các mẫu được kết xuất phải bằng các trị số của các mẫu được tạo ra bởi quy trình giải mã được xác định.

Để phù hợp bộ mã hóa định thời kết xuất, HSS hoạt động như được mô tả trên đây, với các lịch trình được lựa chọn chỉ từ tập hợp phụ của các trị số SchedSelIdx mà đối với nó tốc độ bit và kích thước CPB được hạn chế như được xác định trong Phụ lục A đối với các hồ sơ, tầng và cấp độ, hoặc với các lịch trình cung cấp "nội suy" như được xác định dưới dây mà đối với nó tốc độ bit và kích thước CPB được hạn chế như được xác định trong Phụ lục A. Lịch trình cung cấp giống nhau được sử dụng cho cả HRD và DUT.

Khi các thông số HRD và các thông báo SEI khoảng thời gian đệm hiện diện với cpb_cnt_minus1[HighestTid] lớn hơn 0, bộ giải mã sẽ có khả năng giải mã dòng bit như được cung cấp từ việc vận hành HSS bằng cách 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 loại bỏ CPB ban đầu

Biểu thức 15

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

dưới đây:

Bảng 17

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

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

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

cho bất kỳ SchedSelIdx > 0 và r sao cho BitRate[SchedSelIdx - 1] <= r <= BitRate[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 hồ sơ, tầng và cấp độ. InitCpbRemovalDelay[SchedSelIdx] có thể khác biệt cho từng khoảng thời gian đệm và cần được tính toán lại.

Để phù hợp 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 đầu ra hình ảnh là giống nhau cho 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 đến DUT "theo yêu cầu" từ DUT, nghĩa là HSS cung cấp các bit (trong thứ tự mã hóa) chỉ khi DUT cần nhiều bit hơn để tiếp tục quá trình xử lý của nó. Điều này có nghĩa là đối với xét nghiệm này, bộ đệm hình ảnh được mã hóa của DUT có thể nhỏ bằng kích thước của gói 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 nhở một trong số các lịch trình được xác định trong dòng bit BitstreamToDecode sao cho tốc độ bit và kích thước CPB được hạn chế như được xác định trong Phụ lục A. Thứ tự kết xuất các hình ảnh là giống nhau cho cả HRD và DUT.

(3) Kích thước CPB HRD được xác định bằng CpbSize[SchedSelIdx] như được xác định trong khoản E.2.3, trong đó SchedSelIdx và các thông số HRD được lựa chọn như được xác định trong khoản C.1. Kích thước DPB được xác định bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi nuh_layer_id cho hình ảnh được mã hóa hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho trị số nuh_layer_id của hình ảnh được mã hóa hiện thời. Trong một số trường hợp, nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện đối với các tập hợp lớp ra được lựa chọn, thì kích thước DPB được xác định bằng vps_max_dec_pic_buffering_minus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid] đối với

currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0, trong đó currLayerId là nuh_layer_id của hình ảnh được mã hóa hiện thời. Khác đi nếu điểm hoạt động DPB information parameters op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, thì kích thước DPB được xác định bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi nuh_layer_id cho hình ảnh được mã hóa hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho trị số nuh_layer_id của hình ảnh được mã hóa hiện thời.

Trong một số trường hợp, nếu các các thông số thông tin DPB của tập hợp lớp ra oop_dpb_info_parameters() hiện diện đối với các tập hợp lớp ra được lựa chọn, kích thước DPB được xác định bằng vps_max_dec_pic_buffering_minus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid] đối với currLayerId đối với các tập hợp lớp ra được lựa chọn, trong đó currLayerId là nuh_layer_id của hình ảnh được mã hóa hiện thời. Khác đi nếu các các thông số thông tin DPB của tập hợp lớp ra oop_dpb_info_parameters() không hiện diện đối với các tập hợp lớp ra được lựa chọn, thì kích thước DPB được xác định bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi nuh_layer_id cho hình ảnh được mã hóa hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho trị số nuh_layer_id của hình ảnh được mã hóa hiện thời.

Thời gian loại bỏ từ 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 được mã hóa chứa các bộ đệm lưu trữ hình ảnh. Số lượng các bộ đệm lưu trữ hình ảnh cho nuh_layer_id bằng 0 bắt nguồn từ SPS hoạt động. Số lượng các bộ đệm lưu trữ hình ảnh cho cho mỗi trị số nuh_layer_id không phải là zero bắt nguồn từ SPS lớp hoạt động cho trị số nuh_layer_id không phải zero đó. Mỗi trong số các bộ đệm lưu trữ hình ảnh chứa hình ảnh được mã hóa được đánh dấu là "được sử dụng để tham chiếu" hoặc được giữ lại để kết xuất trong tương lai. Quy trình để kết xuất và loại bỏ các hình ảnh từ DPB như được xác định trong khoản F.13.5.2.2 được hủy bỏ, tiếp theo sau là

yêu cầu của quy trình giải mã hình ảnh, đánh dấu, đệm bổ sung, và lưu trữ như được xác định trong khoản F.13.5.2.3. Quy trình "đệm" được xác định trong khoản F.13.5.2.4 và được hủy bỏ như được xác định trong các khoản F.13.5.2.2 và F.13.5.2.3.

Kết xuất và loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thời khi gói giải mã thứ nhất của gói truy nhập có chứa hình ảnh hiện thời được loại bỏ từ 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 mà nó không phải là hình ảnh 0, các bước theo thứ tự như sau được áp dụng:

(A) Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm theo cách sau:

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

(ii) Khác đi, nếu trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] nhận được từ SPS hoạt động lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động cho hình ảnh đứng trước, NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được cài đặt là 1 nhờ bộ giải mã đang xét nghiệm, không quan tâm đến trị số no_output_of_prior_pics_flag. mặc dù việc cài đặt NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã đang xét nghiệm được phép cài đặt NoOutputOfPriorPicsFlag là 1 trong trường hợp này.

(iii) Khác đi, NoOutputOfPriorPicsFlag được cài đặt bằng no_output_of_prior_pics_flag.

(B) Trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét 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 trống không cần đến kết xuất của các hình ảnh chúng chứa, và độ đầy của DPB được cài đặt bằng 0.

(ii) Khác đi (NoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa hình ảnh được đánh dấu là "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (không kết xuất), và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm trống bởi yêu cầu lặp đi lặp lại quy trình "đệm" được xác định trong khoản F.13.5.2.4, và độ đầy của DPB được cài đặt bằng 0.

(iii) Khác đi (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 hình ảnh được đánh dấu là "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh mà nó được làm trống, độ đầy của DPB được giảm bớt một. Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được mã hóa 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 đang xét nghiệm:

(1) Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm TargetOp, MaxNumReorderPics[TargetOp] [currLayerId][HighestTid] được cài đặt bằng vps_max_num_reorder_pics[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_num_reorder_pics[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động

op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, thì MaxNumReorderPics[TargetOp][currLayerId][HighestTid] được cài đặt bằng sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

(2) Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm TargetOp, thì MaxLatencyIncreasePlus1[TargetOp][currLayerId][HighestTid] được cài đặt bằng vps_max_latency_increase_plus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_latency_increase_plus1[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm, MaxLatencyPictures[TargetOp][currLayerId][HighestTid] được cài đặt bằng VpsMaxLatencyPictures[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng VpsMaxLatencyPictures[TargetOp][CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, thì MaxLatencyIncreasePlus1[TargetOp][currLayerId][HighestTid] được cài đặt bằng sps_max_latency_increase_plus1[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS lớp hoạt động cho trị số currLayerId và MaxLatencyPictures[TargetOp] [currLayerId][HighestTid] được cài đặt bằng SpsMaxLatencyPictures[HighestTid] nhận được từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

(3) Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện đối với điểm hoạt động được lựa chọn đang xét nghiệm TargetOp, thì MaxDecPicBufferingMinus1[TargetOp] [currLayerId][HighestTid] được cài đặt bằng vps_max_dec_pic_buffering_minus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_dec_pic_buffering_minus1[TargetOp] [CurrLayerId][HighestTid]

cho currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, MaxDecPicBufferingMinus1[TargetOp][currLayerId][HighestTid] được cài đặt bằng sps_max_dec_pic_buffering_minus1[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình "đệm" được xác định trong khoản F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt độ đầy của DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm trống, 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì biến liên kết 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 các tức thời khi gói giải mã sau cùng của gói truy nhập n có chứa hình ảnh hiện thời được loại bỏ từ CPB.

Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được mã hóa hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là "cần kết xuất" và mà nó có nuh_layer_id value bằng currLayerId, thì biến liên kết PicLatencyCount[currLayerId] được cài đặt bằng PicLatencyCount[currLayerId] + 1.

Hình ảnh hiện thời được coi như đã được giải mã sau khi gói giải mã sau cùng của hình ảnh được giải mã. Hình ảnh được mã hóa 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 được mã hóa hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu là "cần kết xuất" và Biến liên kết PicLatencyCount[currLayerId] được cài đặt bằng 0.

(B) Khác đi (hình ảnh được mã hóa hiện thời has PicOutputFlag bằng 0), thì nó được đánh dấu là "không cần kết xuất".

Hình ảnh được mã hóa 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 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì biến liên kết 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 thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm, MaxNumReorderPics[currLayerId][HighestTid] được cài đặt bằng vps_max_num_reorder_pics[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_num_reorder_pics[CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId

lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm MaxNumReorderPics[currLayerId][HighestTid] được cài đặt bằng sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

(2) Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm, MaxLatencyIncreasePlus1[currLayerId][HighestTid] được cài đặt bằng vps_max_latency_increase_plus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_latency_increase_plus1[CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện cho điểm hoạt động đang xét nghiệm, MaxLatencyPictures[currLayerId][HighestTid] được cài đặt bằng VpsMaxLatencyPictures [HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng VpsMaxLatencyPictures [CurrLayerId][HighestTid] đối với currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, MaxLatencyIncreasePlus1[currLayerId][HighestTid] được cài đặt bằng sps_max_latency_increase_plus1[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS lớp hoạt động cho trị số currLayerId và MaxLatencyPictures[currLayerId][HighestTid] được cài đặt bằng SpsMaxLatencyPictures [HighestTid] nhận được từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

(3) Nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() hiện diện đối với điểm hoạt động được lựa chọn đang xét nghiệm, MaxDecPicBufferingMinus1[currLayerId][HighestTid] được cài đặt bằng vps_max_dec_pic_buffering_minus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng vps_max_dec_pic_buffering_minus1[CurrLayerId][HighestTid] đối với

currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động op_dpb_info_parameters() không hiện diện cho điểm hoạt động đang xét nghiệm, MaxDecPicBufferingMinus1[currLayerId][HighestTid] được cài đặt bằng sps_max_dec_pic_buffering_minus1[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình "đệm" được xác định theo khoản F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt độ đầy của DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm trống, 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì biến liên kết 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 thời khi gói giải mã sau cùng của gói truy nhập n có chứa hình ảnh hiện thời được loại bỏ từ CPB.

Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được mã hóa hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là "cần kết xuất" và mà nó có nuh_layer_id value bằng currLayerId, thì biến liên kết PicLatencyCount[currLayerId] được cài đặt bằng PicLatencyCount[currLayerId] + 1.

Hình ảnh hiện thời được coi như đã được giải mã sau khi gói giải mã sau cùng của hình ảnh được giải mã. Hình ảnh được mã hóa 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 được mã hóa hiện thời has PicOutputFlag bằng 1, thì nó được đánh dấu là "cần kết xuất" và biến liên kết PicLatencyCount[currLayerId] được cài đặt bằng 0.

(B) Khác đi (hình ảnh được mã hóa hiện thời has PicOutputFlag bằng 0), thì nó được đánh dấu là "không cần kết xuất".

Hình ảnh được mã hóa 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 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó biến liên kết 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 chúng là thứ nhất cho kết xuất được lựa chọn làm các hình ảnh có trị số 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, bằng cách 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 đối với trị số nuh_layer_id bằng trị số đó của hình ảnh, các hình ảnh được cắt

xén đó là đầu ra theo thứ tự tăng của nuh_layer_id, và các hình ảnh đó được đánh dấu là "không cần kết xuất".

(C) Mỗi bộ đệm lưu trữ hình ảnh có chứa hình ảnh được đánh dấu là "không sử dụng để tham chiếu" và nó bao gồm một trong số các hình ảnh mà nó đã được cắt xén và kết xuất được làm trống.

Phần mở rộng VPS có thể có các cải biến bổ sung, nếu muốn.

Trên Fig.40, cải biến bổ sung có thể bao gồm các thông số DPB đang được gửi trong phần mở rộng VPS đối với các tập hợp lớp ra thay vì đối với các điểm hoạt động, trong đó oops_dpb_info_parameters(j) được minh họa trên Fig.41.

num_dpb_info_parameters xác định số lượng các kết cấu cú pháp op_dpb_parameters( ) hiện diện trong RBSP của phần mở rộng VPS. num_dpb_info_parameters của các bộ giải mã phải nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số cuối.

output_point_layer_set_idx[i] xác định chỉ số, vào danh sách của các tập hợp lớp ra mục tiêu mà kết cấu cú pháp oop_dpb_info_parameters( ) thứ i trong phần mở rộng VPS tác động đến.

Trị số output_point_layer_set_idx[i] phải nằm trong khoảng từ 0 đến num_output_layer_sets, bao gồm cả trị số cuối. Yêu cầu về sự phù hợp dòng bit là output_point_layer_set_idx [i] không bằng output_point_layer_set_idx [j] cho j bất kỳ không bằng i.

Trên Fig.42, oop_dpb_info_paremters(c) có thể còn được cải biến, trong đó cú pháp trong phần mở rộng VPS có thể như được minh họa trên Fig.43.

Trên Fig.44, oop_dpb_info_paremters(c) có thể còn được cải biến, trong đó cú pháp trong phần mở rộng VPS có thể như được minh họa trên Fig.45 hoặc Fig.46.

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

Bảng 18

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

có thể được thay đổi thành

Bảng 19

for(j = 0; j <= vps_max_layers_minus1; j++)
oop_dpb_info_parameters(j)

vps_max_layer_id xác định trị số nuh_layer_id tối đa của tất cả các gói NAL trong CVS. vps_max_layers_minus1, xác định số lượng tối đa của các lớp mà chúng có thể hiện diện trong CVS, trong đó một lớp có thể ví dụ là lớp có thể điều chỉnh không gian, lớp có thể điều chỉnh chất lượng, hình hoa văn hoặc hình độ sâu.

Phương án thay thể mang tính ví dụ khác đối với cú pháp trong phần mở rộng VPS là

Bảng 20

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

có thể được thay đổi thành

Bảng 21

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

trong đó numOutputLayers đối với chỉ số tập hợp lớp ra được lựa chọn oplsIdx nhận được là:

Bảng 22

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

Phương án thay thể mang tính 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 oplsIdx được lựa chọn nhận được là:

Bảng 25

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

Sau đó danh sách nhãn hiệu lớp được giải mã mục tiêu targetDLayerIdList và numDecodedLayers đối với oplsIdx được lựa chọn nhận được 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 hiệu 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

for(j = 0; j <= vps_max_layer_id; j++) {	
vps_ layer_ info_present_flag[j]	u(1)
if(vps_layer info_present_flag)	
oop_dpb_info_parameters(j)	
}	

vps_layer_info_present_flag[j] bằng 1 xác định rằng oop_dpb_info_parameters hiện diện đối với lớp thứ j cho tập hợp lớp ra cụ thể. vps_layer_info_present_flag[j] bằng 0 xác định rằng oop_dpb_info_parameters không hiện diện đối với lớp thứ j cho tập hợp lớp ra cụ thể.

Theo phương án khác num_dpb_info_parameters của các bộ giải mã phải nằm trong khoảng từ 0 đến 1024, bao gồm cả trị số 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 trị số cuối.

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

Trên Fig.48, layer_dpb_info(i) cải biến có thể được sử dụng trong đó thành phần 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 rẽ trong oop_dpb_info_parameters(id) / op_dpb_info_parameters(id).

Trên Fig.49A và 49B, vps_extension được cải biến có tính 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 các tập hợp lớp

mà đối với nó các lớp đầu ra được xác định với output_layer_set_index[i] và output_layer_flag[lsIdx][j]. Khi không hiện diện, thì trị số num_output_layer_sets được kết luận 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 kết luận 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 dứt khoát. Trị số max_sub_layers_vps_predict_flag[0] được kết luận bằng 0.

'max_sub_layers_vps_minus1'[i] cộng 1 xác định số lượng tối đa của các lớp phụ tạm thời mà nó có thể hiện diện trong CVS cho lớp với nuh_layer_id bằng i. Trị số max_vps_sub_layers_vps_minus1[i] phải nằm trong khoảng từ 0 đến 6, bao gồm cả trị số cuối. Trong một số trường hợp max_sub_layers_vps_minus1[i] được dùng cho sự suy luận của thành phần 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 kết luận bằng max_sub_layers_vps_minus1[i - 1]. Trị số max_sub_layers_vps_minus1[0] được kết luận bằng vps_max_sub_layers_minus1.

Biến MaxSubLayers[setId] cho setId nằm trong khoảng từ 0 đến num_dpb_info_parameters- 1, bao gồm cả trị số cuối, được rút 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 các kết cấu cú pháp oop_dpb_maxbuffering_parameters(i) hiện diện trong RBSP của phần mở rộng VPS. num_dpb_info_parameters của các bộ giải mã phải nằm trong khoảng từ 0 đến numOutputLayerSets, bao gồm cả trị số cuối.

'output_point_layer_set_idx'[i] xác định chỉ số, vào danh sách của các tập hợp lớp ra mà kết cấu cú pháp oop_dpb_maxbuffering_parameters(i) thứ i trong phần mở rộng VPS tác động đến.

Trị số output_point_layer_set_idx[i] phải nằm trong khoảng từ 0 đến numOutputLayerSets, bao gồm cả trị số cuối. Yêu cầu về sự phù hợp dòng bit là

output_point_layer_set_idx [i] không bằng output_point_layer_set_idx [j] cho j bất kỳ không bằng i.

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 hiện diện đối với các lớp phụ MaxSubLayers[i]. sub_layer_vps_buf_info_present_flag[i] bằng 0 xác định rằng các trị số 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] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS cho lớp với nuh_layer_id bằng highestLayerId trong tập hợp lớp ra kết hợp với chỉ số i trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng j. Trị số max_vps_dec_pic_buffering_minus1[i][j] phải nằm trong khoảng từ 0 đến MaxDpbSize - 1 (như được xác định trong khoản A.4), bao gồm cả trị số cuối. Khi j lớn hơn 0, max_vps_dec_pic_buffering_minus1[i][j] phải 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 cho sự suy luận của các trị số của các thành phần cú pháp SPS sps_max_dec_pic_buffering_minus1[j]. Khi

max_vps_dec_pic_buffering_minus1[i][j] không hiện diện cho i nằm trong khoảng từ 0 đến MaxSubLayers[i] - 2, bao gồm cả trị số cuối, do sub_layer_vps_buf_info_present_flag[i] bằng 0, thì nó được kết luận bằng max_vps_dec_pic_buffering_minus1[i][MaxSubLayers[i] - 1].

Trị số max_vps_dec_pic_buffering_minus1[0][j] đối với mỗi trị số của j được kết luận bằng vps_max_dec_pic_buffering_minus1[j].

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] hiện diện 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 trị số 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 số lượng tối đa cho phép của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS cho lớp với nuh_layer_id bằng i trong thứ tự mã hóa và tiếp theo sau hình ảnh đó trong thứ tự kết xuất khi HighestTid bằng j. Trị số max_vps_num_reorder_pics[i][j] phải nằm trong khoảng từ 0 đến max_vps_dec_pic_buffering_minus1[i][j], bao gồm cả trị số cuối. Khi j lớn hơn 0, thì max_vps_num_reorder_pics[i][j] phải 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 cho sự suy luận của các trị số của thành phần cú pháp SPS sps_max_num_reorder_pics[j]. Khi max_vps_num_reorder_pics[i][j] không hiện diện cho i nằm trong khoảng từ 0 đến max_sub_layers_vps_minus1[i] - 1, bao gồm cả trị số cuối, do

sub_layer_vps_ordering_info_present_flag[i] bằng 0, thì nó được kết luận bằng 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 trị số VpsMaxLatencyPictures[i][j], mà nó xác định số lượng tối đa của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS cho lớp với nuh_layer_id bằng i trong thứ tự kết xuất và tiếp theo sau hình ảnh đó trong thứ tự mã hóa khi HighestTid bằng j.

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

VpsMaxLatencyPictures[i][j] = max_vps_num_reorder_pics[i][j] + 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 biểu diễn.

Trị số max_vps_latency_increase_plus1[j][k] phải nằm trong khoảng từ 0 đến 2^{32} - 2, bao gồm cả trị số cuối. Trong một số trường hợp max_vps_latency_increase_plus1[i][j] được sử dụng cho sự suy luận của các trị số của các thành phần cú pháp SPS sps_max_latency_increase_plus1[j]. Khi max_vps_latency_increase_plus1[i][j] không hiện diện cho i nằm trong khoảng từ 0 đến max_sub_layers_vps_minus1[i] - 1, bao gồm cả trị số cuối, do sub_layer_vps_ordering_info_present_flag[i] bằng 0, thì nó được kết luận bằng max_vps_latency_increase_plus1[i][max_sub_layers_vps_minus1[i]].

Trên Fig.52, vps_extension được cải biến có tính 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, sub_layer_vps_buf_info_predict_flag[i], sub_layer_vps_ordering_info_predict_flag[i] mà chúng được truyền tín hiệu một cách có điều kiện. sub_layer_vps_buf_info_predict_flag[i] bằng 1 xác định rằng max_vps_dec_pic_buffering_minus1[i][j] được kết luận bằng max_vps_dec_pic_buffering_minus1[i - 1][j] đối với mỗi trị số của j.

sub_layer_vps_buf_info_predict_flag[i] bằng 0 xác định rằng max_vps_dec_pic_buffering_minus1[i][j] cho ít nhất một trị số j đã được truyền tín hiệu một cách dứt khoát. sub_layer_vps_buf_info_predict_flag[0] được kết luận bằng 0. Khi không hiện diện, thì sub_layer_vps_buf_info_predict_flag[i] được kết luận bằng 0.

sub_layer_vps_ordering_info_predict_flag[i] bằng 1 xác định rằng các thành phần cú pháp sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics[i][j], và max_vps_latency_increase_plus1[i][j] lần lượt được kết luận bằng sub_layer_vps_ordering_info_present_flag[i - 1], max_vps_num_reorder_pics[i - 1][j], và max_vps_latency_increase_plus1[i 1][j]. sub_layer_vps_ordering_info_predict_flag[i] bằng 0 chỉ báo rằng các thành phần cú pháp sub_layer_vps_ordering_info_present_flag[i], max_vps_num_reorder_pics[i][j], và max_vps_latency_increase_plus1[i][j] đã được truyền tín hiệu một cách dứt khoát. Khi không hiện diện, thì trị số sub_layer_vps_ordering_info_predict_flag[i] được cài đặt bằng 0.

Các thành phần cú pháp khác và ngữ nghĩa trên Fig.52 là giống với ngữ nghĩa trên Fig.49.

Trên Fig.53, vps_extension được cải biến có tính ví dụ được minh họa. Phần mở rộng vps bao gồm cải biến nữa về cú pháp trên Fig.49. Trên Fig.53 oop_dpb_maxbuffering_parameters(i,j) được truyền tín hiệu cho mỗi lớp j đối với tập hợp lớp ra cụ thể i so với Fig.49 mà nó truyền tín hiệu tập hợp đơn lẻ các thông số oop_dpb_maxbuffering_parameters(i) cho lớp cao nhất id trong tập hợp lớp ra i.

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] hiện diện 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 trị số 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] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS cho lớp với nuh_layer_id bằng k cho tập hợp lớp ra kết hợp với chỉ số i trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng j. Trị số max_vps_dec_pic_buffering_minus1[i][k][j] phải nằm trong khoảng từ 0 đến MaxDpbSize - 1 (như được xác định trong khoản A.4), bao gồm cả trị số cuối.

Khi j lớn hơn 0, max_vps_dec_pic_buffering_minus1[i][k][j] phải lớn hơn hoặc bằng max_vps_dec_pic_buffering_minus1[i][k][j - 1] được sử dụng cho sự suy luận của các trị số của các thành phần cú pháp SPS sps_max_dec_pic_buffering_minus1[j].

Khi max_vps_dec_pic_buffering_minus1[i][j] không hiện diện cho i nằm trong khoảng từ 0 đến max_sub_layers_vps_minus1[k] - 1, bao gồm cả trị số cuối, do sub_layer_vps_buf_info_present_flag[i][k] being bằng 0, thì nó được kết luận bằng max_vps_dec_pic_buffering_minus1[i][k] [max_sub_layers_vps_minus1[k]].

Trị số max_vps_dec_pic_buffering_minus1[i][0][j] đối với mỗi trị số của i và j được kết luận bằng vps_max_dec_pic_buffering_minus1[j].

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 thể cho oop_dpb_maxbuffering_parameters so với cú pháp trên Fig.54 cho oop_dpb_maxbuffering_parameters.

Biến MaxSubLayers[setId][k] cho setId nằm trong khoảng từ 0 đến num_dpb_info_parameters - 1, bao gồm cả trị số cuối, nhận được như sau:

Bảng 29

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

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

Trên Fig.56, vps_extension được cải biến có tính ví dụ được minh họa. Phần mở rộng vps bao gồm cải biến nữa về cú pháp trên Fig.52 với cú pháp mới. Trên Fig.53 theo biến thể này oop_dpb_maxbuffering_parameters(i,j) được truyền tín hiệu cho mỗi lớp j đối với tập hợp lớp ra cụ thể i so với Fig.52 mà nó truyền tín hiệu tập hợp đơn lẻ các thông số oop_dpb_maxbuffering_parameters(i) cho lớp cao nhất id trong tập hợp lớp ra i.

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

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 thể cho oop_dpb_maxbuffering_parameters so với cú pháp trên Fig.57 cho oop_dpb_maxbuffering_parameters.

Biến MaxSubLayers[setId][k] cho setId nằm trong khoảng từ 0 đến num_dpb_info_parameters - 1, bao gồm cả trị số cuối, được rút ra như sau:

Bảng 30

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

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

Trên Fig.59, vps_extension được cải biến có tính ví dụ được minh họa. Phần mở rộng vps bao gồm cải biến nữa của cú pháp trên Fig.56 với cú pháp mới, gọi là, cờ hiệu 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 thành phần cú pháp sub_layer_vps_ordering_info_predict_flag[i] và layer_dpb_info_parameters(i) hiện diện đối với các lớp vps_max_num_layers_minus1. layer_dpb_info_parameters_presence_flag[i] bằng 0 xác định rằng các thành phần cú pháp sub_layer_vps_ordering_info_predict_flag[i] và layer_dpb_info_parameters(i) không hiện diện đối với các lớp vps_max_num_layers_minus1.

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

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 thể cho cho oop_dpb_maxbuffering_parameters so với cú pháp trên Fig.60 cho oop_dpb_maxbuffering_parameters.

Biến MaxSubLayers[setId][k] cho setId nằm trong khoảng từ 0 đến num_dpb_info_parameters - 1, bao gồm cả trị số cuối, nhận được như sau:

Bảng 31

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

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

Một phương án thay đổi có tính ví dụ đối vớicú pháp trong phần mở rộng VPS là

Bảng 32

for(k = 1; j <= vps_max_layers_minus1; k++)
oop_dpb_maxbuffering_parameters(k)

có thể được thay đổi thành

Bảng 33

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

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

Một phương án thay đổi có tính 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)

The vps_max_layer_id xác định trị số tối đa nuh_layer_id của tất cả các gói NAL trong CVS. vps_max_layers_minus1, xác định số lượng tối đa của các lớp mà chúng có thể hiện diện trong CVS, trong đó một lớp có thể ví dụ be lớp có thể điều chỉnh không gian, lớp có thể điều chỉnh chất lượng, hình hoa văn hoặc hình độ sâu.

Phương án thay thể mang tính 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 nhận được 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 hiệu 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 hiện diện đối với lớp thứ k cho tập hợp lớp ra cụ thể. vps_layer_info_present_flag[k] bằng 0 xác định rằng oop_dpb_maxbuffering_parameters không hiện diện đối với lớp thứ k cho tập hợp lớp ra cụ thể đó.

Các thành phần cú pháp của các gói NAL không VCL (hoặc các trị số mặc định đối với một số các thành phần 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 hợp thông số HRD (Các thông số HRD NAL và Các thông số HRD VCL) được sử dụng. Các tập hợp thông số HRD được truyền tín hiệu thông qua kết cấu cú pháp hrd_parameters( ), mà nó có thể là phần của kết cấu cú pháp SPS hoặc kết cấu cú pháp VPS.

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

(1) Tập hợp lớp ra đang xét nghiệm, ký hiệu là TargetOpLs được lựa chọn. Điểm hoạt động được tham chiếu trong TargetOpLs by output_layer_set_idx[] xác định điểm hoạt động đang xét nghiệm. Danh sách nhãn hiệu lớp ra OpLayerIdList của TargetOpLs gồm danh sách của các trị số nuh_layer_id, theo thứ tự tăng lên của các trị số nuh_layer_id, hiện diện trong tập hợp phụ của dòng bit kết hợp với TargetOp và TargetOpLs, mà nó là tập hợp phụ của các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm. OpTid của TargetOp bằng TemporalId cao nhất hiện diện trong tập hợp phụ của dòng bit kết hợp với TargetOp.

(2) TargetDecLayerIdList được cài đặt bằng danh sách nhãn hiệu lớp được mã hóa mục tiêu targetDLayerIdList đối với các tập hợp lớp ra được lựa chọn TargetOpLs, HighestTid được cài đặt bằng OpTid của TargetOp, và quy trình trích dòng bit phụ như được xác định trong điều 10 được hủy bỏ với dòng bit đang xét nghiệm, HighestTid, và

TargetDecLayerIdList dưới dạng các dữ liệu đầu vào, và đầu ra được chỉ định đến BitstreamToDecode.

(3) Kết cấu cú pháp hrd_parameters( ) và kết cấu cú pháp sub_layer_hrd_parameters( ) có thể áp dụng cho TargetOp được lựa chọn. Nếu TargetDecLayerIdList chứa tất cả các trị số nuh_layer_id hiện diện trong dòng bit đang xét nghiệm, kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) được lựa chọn. Khác đi, kết cấu cú pháp hrd_parameters( ) trong SPS hoạt động (hoặc được cung cấp thông qua các thiết bị bên ngoài không được xác định trong bản mô tả này) mà nó áp dụng cho TargetOp được lựa chọn. Trong kết cấu cú pháp hrd_parameters( ) được lựa chọn, nếu BitstreamToDecode là dòng bit Loại I, thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo ngay sau điều kiện "if(vcl_hrd_parameters_present_flag)" được lựa chọn và biến NalHrdModeFlag được cài đặt bằng 0; khác đi (BitstreamToDecode là dòng bit Loại II), thì kết cấu cú pháp sub_layer_hrd_parameters(HighestTid) mà nó theo ngay sau hoặc điều kiện "if(vcl_hrd_parameters_present_flag)" (trong trường hợp này biến NalHrdModeFlag được cài đặt bằng 0) hoặc điều kiện "if(nal_hrd_parameters_present_flag)" (trong trường hợp này biến NalHrdModeFlag được cài đặt bằng 1) được lựa chọn. Khi BitstreamToDecode là dòng bit Loại II và NalHrdModeFlag bằng 0, tất cả các gói NAL không VCL trừ các gói NAL dữ liệu bộ lọc, và tất cả các thành phần cú pháp leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, và trailing_zero_8bits mà chúng tạo thành dòng byte từ dòng gói NAL (như được xác định trong Phụ lục B), khi hiện diện, được vứt bỏ ra khỏi BitstreamToDecode, và và dòng bit còn lại được chỉ định đến 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 sự phù hợp với hồ sơ, tầng và cấp độ cụ thể phải có khả năng giải mã thành công tất cả các dòng bit có phù hợp với các yêu cầu về phù hợp dòng bit được xác định trong khoản 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 gói NAL VCL, và khoảng thời gian 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 gói NAL không VCL), hoặc nhờ các thiết bị bên ngoài không được xác định trong bản mô tả này.

(2) Khi dòng bit chứa các thành phần cú pháp có các trị số mà chúng được xác định là dự trữ và nó được xác định rằng các bộ giải mã cần bỏ qua các trị số của các thành phần cú pháp hoặc các gói NAL có chứa các thành phần cú pháp có các trị số dự trữ, và dòng bit theo cách khác phù hợp với bản mô tả này, bộ giải mã phù hợp sẽ phải giải mã dòng bit theo cùng cách thức giống như khi nó giải mã dòng bit phù hợp và phải bỏ qua các thành phần cú pháp hoặc các gói NAL có chứa các thành phần cú pháp có các trị số dự trữ 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 sự phù hợp của bộ giải mã, các dòng bit phù hợp với hồ sơ, tầng và cấp độ được đòi hỏi, như được xác định trong khoản C.4 được cung cấp bởi bộ lập lịch trình dòng giả thiết (HSS) cho cả HRD và bộ giải mã đang xét 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 hình ảnh với PicOutputFlag bằng 1, và, đối với mỗi hình ảnh giải mã được cắt xén được kết xuất bởi DUT, các trị số của tất cả các mẫu được kết xuất phải bằng các trị số của các mẫu được tạo ra bởi quy trình giải mã được xác định.

Đối với sự phù hợp của bộ mã hóa định thời kết xuất, HSS hoạt động như được mô tả trên đây, với các lịch trình được lựa chọn chỉ từ tập hợp phụ của các trị số SchedSelIdx mà đối với nó tốc độ bit và kích thước CPB được hạn chế như được xác định trong Phụ lục A đối với hồ sơ, tầng và cấp độ 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 hạn chế như được xác định trong Phụ lục A. Lịch trình cung cấp giống nhau được sử dụng cho cả HRD và DUT.

Khi các thông số HRD và các thông báo SEI khoảng thời gian đệm hiện diện với cpb_cnt_minus1[HighestTid] lớn hơn 0, bộ giải mã sẽ có khả năng giải mã dòng bit như được cung cấp từ việc vận hành HSS bằng cách 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 loại bỏ CPB ban đầu

Biểu thức 16

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

như sau:

Bảng 41

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

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

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

cho bất kỳ SchedSelIdx > 0 và r sao cho BitRate[SchedSelIdx - 1] <= r <= BitRate[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 hồ sơ, tầng và cấp độ được xác định. InitCpbRemovalDelay[SchedSelIdx] có thể khác biệt cho từng khoảng thời gian đệm và cần được tính toán lại.

Đối với sự 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 đầu ra hình ảnh là giống nhau cho 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 đến DUT "theo yêu cầu" từ DUT, nghĩa là HSS cung cấp các bit (trong thứ tự mã hóa) chỉ khi DUT cần nhiều bit hơn để tiếp tục quá trình xử lý của nó. Điều này có nghĩa là đối với xét nghiệm này, bộ đệm hình ảnh được mã hóa của DUT có thể nhỏ bằng kích thước của gói 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 nhở một trong số các lịch trình được xác định trong dòng bit BitstreamToDecode sao cho tốc độ bit và kích thước CPB được hạn chế như được xác định trong Phụ lục A. Thứ tự kết xuất các hình ảnh là giống nhau cho cả HRD và DUT.

(3) Kích thước CPB HRD được xác định bằng CpbSize[SchedSelIdx] như được xác định trong khoản E.2.3, trong đó SchedSelIdx và các thông số HRD được lựa chọn như được xác định trong khoản C.1. Kích thước DPB được xác định bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi nuh_layer_id cho hình ảnh được mã hóa hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho trị số nuh_layer_id của hình ảnh được mã hóa hiện thời.

Trong một số trường hợp nếu các các thông số thông tin DPB của tập hợp lớp ra oop_dpb_maxbuffering_parameters() hiện diện đối với các tập hợp lớp ra được lựa chọn, Kích thước DPB được xác định bằng max_vps_dec_pic_buffering_minus1[CurrLayerId][HighestTid] trong đó currLayerId là nuh_layer_id của hình ảnh được mã hóa hiện thời. Khác đi nếu các các thông số thông tin DPB của tập hợp lớp ra oop_dpb_maxbuffering_parameters () không hiện diện đối với các tập hợp lớp ra được lựa chọn, kích thước DPB được xác định bằng sps_max_dec_pic_buffering_minus1[HighestTid] + 1 từ SPS hoạt động (khi nuh_layer_id cho hình ảnh được mã hóa hiện thời bằng 0) hoặc từ SPS lớp hoạt động cho trị số nuh_layer_id của hình ảnh được mã hóa hiện thời.)

Thời gian loại bỏ từ 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 được mã hóa chứa các bộ đệm lưu trữ hình ảnh. Số lượng các bộ đệm lưu trữ hình ảnh cho nuh_layer_id bằng 0 bắt nguồn từ SPS hoạt động. Số lượng các bộ đệm lưu trữ hình ảnh cho mỗi trị số nuh_layer_id không phải là zero bắt nguồn từ SPS lớp hoạt động cho trị số nuh_layer_id không phải zero đó. Mỗi trong số các bộ đệm lưu trữ hình ảnhchứa hình ảnh được mã hóa được đánh dấu là "được sử dụng để tham chiếu"

hoặc được giữ lại để kết xuất trong tương lai. Quy trình để kết xuất và loại bỏ các hình ảnh từ DPB như được xác định trong khoản F.13.5.2.2 được hủy bỏ, tiếp theo sau là yêu cầu của quy trình giải mã hình ảnh, đánh dấu, đệm bổ sung, và lưu trữ như được xác định theo khoản F.13.5.2.3. Quy trình "đệm" được xác định theo khoản F.13.5.2.4 và được hủy bỏ như được xác định theo các khoản F.13.5.2.2 và F.13.5.2.3.

Kết xuất và loại bỏ các hình ảnh từ DPB trước khi giải mã của hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra một cách tức thời khi gói giải mã thứ nhất của gói truy nhập có chứa hình ảnh hiện thời được loại bỏ từ 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 mà nó không phải là hình ảnh 0, các bước theo thứ tự như sau được áp dụng:

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

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

(ii) Khác đi, nếu trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid] nhận được từ SPS hoạt động lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minus1[HighestTid], nhận được từ SPS hoạt động cho hình ảnh đứng trước, NoOutputOfPriorPicsFlag có thể (nhưng không bắt buộc) được cài đặt là 1 nhờ bộ giải mã đang xét nghiệm, không quan tâm đến trị số no_output_of_prior_pics_flag. Mặc dù việc cài đặt NoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã đang xét nghiệm được phép cài đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

(iii) Khác đi, NoOutputOfPriorPicsFlag được cài đặt bằng no_output_of_prior_pics_flag.

(B) Trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HRD theo cách sau:

(i) Nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh trong DPB được làm trống không cần đến kết xuất của các hình ảnh chúng chứa, và độ đầy của DPB được cài đặt bằng 0.

(ii) Khác đi (NoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh có chứa hình ảnh được đánh dấu là "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (không kết xuất), và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm trống bởi yêu cầu lặp đi lặp lại quy trình "đệm" được xác định trong khoản F.13.5.2.4, và độ đầy của DPB được cài đặt bằng 0.

(iii) Khác đi (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 hình ảnh được đánh dấu là "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (không kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh mà nó được làm trống, độ đầy của DPB được giảm bớt một. Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được mã hóa hiện thời.

Các biến MaxNumReorderPics[currLayerId][HighestTid], MaxLatencyIncreasePlus1[currLayerId][HighestTid], MaxLatencyPictures[currLayerId][HighestTid], MaxDecPicBufferingMinus1[currLayerId][HighestTid] nhận được theo cách sau:

Nếu các thông số thông tin DPB của lớp layer_dpb_info_parameters() hiện diện trong VPS, thì MaxNumReorderPics[currLayerId][HighestTid] được cài đặt bằng vps_max_num_reorder_pics[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng max_vps_num_reorder_pics[CurrLayerId][HighestTid] cho currLayerId khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của lớp

layer_dpb_info_parameters() không hiện diện, thì MaxNumReorderPics[currLayerId][HighestTid] được cài đặt bằng sps_max_num_reorder_pics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

Nếu các thông số thông tin DPB của lớp layer_dpb_info_parameters() hiện diện trong VPS, thì MaxLatencyIncreasePlus1[currLayerId][HighestTid] được cài đặt bằng vps_max_latency_increase_plus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng max_vps_latency_increase_plus1[CurrLayerId][HighestTid] cho currLayerId khi currLayerId lớn hơn 0. Nếu các thông số thông tin DPB của lớp layer_dpb_info_parameters() hiện diện trong VPS, thì MaxLatencyPictures[currLayerId][HighestTid] được cài đặt bằng SpsMaxLatencyPictures [HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng VpsMaxLatencyPictures [CurrLayerId][HighestTid] cho currLayerId khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của lớp layer_dpb_info_parameters() không hiện diện đối với điểm hoạt động đang xét nghiệm, MaxLatencyIncreasePlus1[currLayerId][HighestTid] được cài đặt bằng sps_max_latency_increase_plus1[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS lớp hoạt động cho trị số currLayerId và MaxLatencyPictures[currLayerId][HighestTid] được cài đặt bằng SpsMaxLatencyPictures[currLayerId][HighestTid] nhận được từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

Nếu các thông số thông tin DPB của điểm hoạt động oop_dpb_maxbuffering_parameters () hiện diện đối với các tập hợp lớp ra được lựa chọn, thì MaxDecPicBufferingMinus1[currLayerId][HighestTid] được cài đặt bằng vps_max_dec_pic_buffering_minus1[HighestTid] khi currLayerId bằng 0 hoặc được cài đặt bằng max_vps_dec_pic_buffering_minus1[CurrLayerId][HighestTid] cho currLayerId cho điểm hoạt động đang xét nghiệm khi currLayerId lớn hơn 0. Khác đi nếu các thông số thông tin DPB của điểm hoạt động oop_dpb_maxbuffering_parameters () không hiện diện cho điểm hoạt động đang xét nghiệm, thì MaxDecPicBufferingMinus1[currLayerId][HighestTid] được cài đặt bằng sps_max_dec_pic_buffering_minus1[

HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS lớp hoạt động cho trị số currLayerId.

Khi một hoặc nhiều điều kiện sau đây là đúng, quy trình "đệm" được xác định trong khoản F.13.5.2.4 được hủy bỏ một cách lặp đi lặp lại trong khi tiếp tục giảm bớt độ đầy của DPB đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung được làm trống, 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì biến liên kết 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 trong khoản này xảy ra một các tức thời khi gói giải mã sau cùng của gói truy nhập n có chứa hình ảnh hiện thời được loại bỏ từ CPB.

Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được mã hóa hiện thời.

Đối với mỗi hình ảnh trong DPB được đánh dấu là "cần kết xuất" và mà nó có trị số nuh_layer_id bằng currLayerId, thì biến liên kết PicLatencyCount[currLayerId] được cài đặt bằng PicLatencyCount[currLayerId] + 1.

Hình ảnh hiện thời được coi như đã được giải mã sau khi gói giải mã sau cùng của hình ảnh được giải mã. Hình ảnh được mã hóa 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 được mã hóa hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu là "cần kết xuất" và biến liên kết PicLatencyCount [currLayerId] được cài đặt bằng 0.

(B) Khác đi (hình ảnh được mã hóa hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu là "không cần kết xuất".

Hình ảnh được mã hóa 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 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 hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu là "cần kết xuất" mà đối với nó thì biến liên kết 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 chúng là thứ nhất cho kết xuất được lựa chọn làm các hình ảnh có trị số 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, bằng cách 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 trị số nuh_layer_id bằng trị số đó của hình ảnh, các hình ảnh được cắt xén đó là đầu ra theo thứ tự tăng của nuh_layer_id, và các hình ảnh đó được đánh dấu là "không cần kết xuất".

(C) Mỗi bộ đệm lưu trữ hình ảnh có chứa hình ảnh được đánh dấu là "không sử dụng để tham chiếu" và nó bao gồm một trong số các hình ảnh mà nó đã được cắt xén và kết xuất được làm trống.

Trên FIG. 62, cú pháp của tập hợp thông số chuỗi (sps) cải biến mang tính 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 trị số cờ hiệu này và dựa trên trị số nuh_layer_id một số các thành phần 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 thành phần 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 hiện diện trong SPS này. Trị số của các thông số này được cài đặt bằng trị số của các thông 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 của SPS hoạt động. sps_dpb_params_present_flag bằng 1 xác định rằng, cho SPS với nuh_layer_id >0 thì các thành phần 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] hiện diện trong SPS này.

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

Theo phương án khác một hoặc nhiều thành phần cú pháp có thể được truyền tín hiệu với ue(v) hoặc biểu đồ mã hóa khác nào đó thay cho số lượng cố định các bit như mã hóa u(v).

Theo phương án khác tên của các thành phần 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 thành phần cú pháp khác có thể được truyền tín hiệu từng hình ảnh bất kỳ chỗ nào trong dòng bit. Ví dụ chúng có thể được truyền tín hiệu trong tiêu đề phân đoạn phiến, pps/ sps/ vps/ hoặc bất kỳ tập hợp thông số khác hoặc phần quy chuẩn khác của dòng bit.

Theo các phương án khác nữa tất cả các khái niệm được xác định trong sáng chế liên quan đến các tập hợp lớp ra 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 dùng để mã hóa viđeo được bộc lộ. Phương pháp này bao gồm bắt đầu phân tách tiêu đề phiến thứ nhất của hình ảnh hiện thời. Đã xác định được những bước nào được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) sẽ dựa vào hình ảnh và các bước nào sẽ dựa vào gói truy nhập (AU). Việc loại bỏ từ DPB được thực hiện. Kết xuất hình ảnh từ DPB được thực hiện. Bước giải mã và lưu trữ hình ảnh được mã hóa hiện thời trong DPB được thực hiện. Hình ảnh được mã hóa hiện thời trong DPB được đánh dấu. Kết xuất hình ảnh bổ sung từ DPB cũng được thực hiện.

Trong một số cấu hình, việc loại bỏ và kết xuất từ DPB có thể là dựa trên ít nhất một cờ hiệu kết xuất của AU, như cờ hiệu kết xuất của AU, cờ hiệu kết xuất không rasl của AU và/hoặc cờ hiệu của các hình ảnh trước không kết xuất của AU. Cờ hiệu kết xuất của AU có thể được suy ra dựa trên các thành phần cú pháp được truyền tín hiệu trong dòng bit và các điều kiện khác. Các cờ hiệu AU đại diện cho các cờ hiệu được rút ra và được áp dụng tại cấp độ gói truy nhập (AU). Trong một số trường hợp, các cờ hiệu đó sẽ là các cờ hiệu khác với các cờ hiệu tương tự mà chúng được truyền tín hiệu hoặc rút ra tại cấp độ hình ảnh, như cờ hiệu kết xuất hình ảnh (ví dụ, pic_output_flag), cờ hiệu kết xuất

không rasl của hình ảnh (ví dụ, NoRaslOutputFlag) và/hoặc cờ hiệu không kết xuất hình ảnh của các hình ảnh trước (ví dụ, no_output_of_prior_pics flag).

Theo một cấu hình, cờ hiệu kết xuất của AU thứ nhất có thể nhận được từ và được sử dụng bởi DPB và cờ hiệu kết xuất của AU thứ hai nhận được từ và được sử dụng bởi DPB. Theo cấu hình khác, tất cả các trị số thành phần cú pháp của cờ hiệu kết xuất hình ảnh có thể được hạn chế ở cùng trị số đối với tất cả hình ảnh được mã hóa AU khi kết xuất hình ảnh flag có hiện diện trong tiêu đề phiến thứ nhất. Trong một số cấu hình, tất cả các trị số thành phần cú pháp của cờ hiệu kết xuất hình ảnh có thể được hạn chế ở cùng trị số đối với tất cả hình ảnh được mã hóa AU khi khi không có cờ hiệu kết xuất hình ảnh nào hiện diện trong tiêu đề phiến thứ nhất.

Theo một cấu hình, việc loại bỏ có thể dựa trên hình ảnh, việc kết xuất hình ảnh có thể dựa trên AU (gói truy nhập), 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. Bước loại bỏ từ SPB có thể loại bỏ một hoặc nhiều hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời.

Theo cấu hình khác, việc loại 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 loại 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 loại 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 loại 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 được mã hóa 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. Độ đầy của DPB có thể được tăng thêm một khi hình ảnh được mã hóa được lưu trữ trong DPB trong bộ đệm lưu trữ trống. Độ đầy của DPB có thể được giảm bớt một khi một hình ảnh được kết xuất

từ DPB. Độ đầy của DPB có thể được theo dõi theo từng lớp. Độ đầy của DPB cũng có thể được theo dõi đối với tập hợp lớp ra.

DPB có thể bao gồm các bộ đệm hình ảnh được xác định và quản lý một cách riêng rẽ đối với các hình ảnh được mã hóa 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 được mã hóa 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, hình ảnh được mã hóa có thể được lưu trữ trong một khe của bộ đệm hình 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 mà nó phù hợp với chuẩn mã hóa viđeo hiệu suất cao có thể điều chỉnh (SHVC). Phương pháp này cũng có thể được thực hiện nhờ bộ giải mã bên trong thiết bị điện tử mà nó phù hợp với chuẩn mã hóa viđeo hiệu suất cao đa hình ảnh (MV-HEVC).

Thiết bị điện tử được tạo cấu hình dùng để mã hóa viđeo 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 tách tiêu đề phiến 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 được mã hóa (DPB) sẽ dựa vào hình ảnh và các bước nào sẽ dựa vào gói truy nhập (AU). Các lệnh trong bộ nhớ còn có thể tiến hành thực hiện việc loại bỏ từ 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 được mã hóa hiện thời trong DPB. Các lệnh trong bộ nhớ cũng có thể tiến hành đánh dấu hình ảnh được mã hóa hiện thời trong DPB. Các lệnh trong bộ nhớ còn có thể tiến hành thực hiện kết xuất hình ảnh bổ sung từ DPB.

Một số cấu hình sẽ được mô tả có liên quan đến 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 sắp xếp 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 trình bày trên các hình vẽ, không phải là để hạn chế 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 thể hiện sơ đồ khối minh họa việc mã hóa viđeo 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 viđeo và/hoặc để giải mã viđeo. Các thiết bị điện tử 102 có thể được tạo cấu hình để sử dụng tác vụ của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp. Tác vụ của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp này tham chiếu đến các kịch bản trong đó nhiều bước của việc loại bỏ, kết xuất (đệm), lưu trữ, đánh dấu và kết xuất (đệm) bổ sung được thực hiện trên bộ đệm hình ảnh được mã hóa (DPB) 116 xảy ra hoặc trên cơ sở hình ảnh hoặc trên cơ sở gói truy nhập (AU). Mặc dù các kết hợp của các bước này được tham chiếu đến như được thực hiện trên cơ sở hình ảnh hoặc trên cơ sở gói truy nhập (AU), tất cả các kết hợp có thể của việc thực hiện mỗi bước này được trợ giúp riêng lẻ hoặc trên cơ sở hình ảnh hoặc trên cơ sở gói truy nhập (AU).

Trong bản mô tả này, gói truy nhập (AU) tham chiếu đến tập hợp các gói của lớp mạng trừu tượng (NAL) mà chúng được liên kết với nhau theo quy tắc phân loại xác định, mà chúng có tính liên tiếp trong thứ tự mã hóa, và chúng bao gồm các gói NAL của lớp mã hóa viđeo (VCL) của tất cả các hình ảnh mã hóa liên kết với cùng thời gian kết xuất và các gói NAL không VCL liên kết của chúng. Lớp cơ sở thứ nhất là lớp trong đó tất cả các gói NAL VCL có nuh_layer_id bằng 0. Hình ảnh được mã hóa là sự trình bày được mã hóa của hình ảnh mà nó bao gồm các gói NAL VCL với trị số nuh_layer_id và nó bao gồm tất cả các gói dạng cây mã hóa của hình ảnh. Trong một số trường hợp, hình ảnh được mã hóa có thể được gọi là thành phần lớp. Các chi tiết bổ sung về các bước dựa trên hình ảnh hoặc dựa trên gói truy nhập (AU) được trình bày như sau có liên quan đến 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 viđeo hiệu suất cao đa hình ảnh (MV-HEVC), chuẩn mã hóa viđeo hiệu suất cao có thể điều chỉnh (SHVC) hoặc chuẩn mã hóa viđeo hiệu suất cao đa hình ảnh (MV-HEVC). Chuẩn HEVC là chuẩn nén viđeo mà nó hoạt động như chuẩn kế tiếp của định dạng H.264/MPEG-4 AVC (Mã hóa viđeo tiên tiến) và mà nó tạo ra chất lượng viđeo cải tiến và các tỷ lệ nén dữ liệu nâng cao. Trong bản mô tả này, hình 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 mầu sắc 4:2:0, 4:2:2 và 4:4:4 hoặc định dạng khác nào đó. Hoạt động của bộ mã hóa tham chiếu giả thiết (HRD) và hoạt động của bộ đệm hình ảnh được 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 bổ sung 2112. Thiết bị điện tử thứ nhất 2102a có thể nhận được hình ảnh đầu vào 2106. Trong một số cấu hình, hình ảnh đầu vào 2106 này có thể được bắt giữ trên thiết bị điện tử thứ nhất 2102a bằng cách sử dụng cảm biến hình ảnh, được lấy ra 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 được mã hóa 2110. Ví dụ, bộ mã hóa 2108 này có thể mã hóa một loạt các hình ảnh đầu vào 2106 (ví dụ, viđeo). Dữ liệu được mã hóa 2110 có thể là dữ liệu số (ví dụ, dòng bit).

Môđun truyền tín hiệu bổ sung 2112 có thể tạo ra việc truyền tín hiệu bổ sung dựa trên dữ liệu được mã hóa 2110. Ví dụ, môđun truyền tín hiệu bổ sung 2112 có thể bổ sung dữ liệu bổ sung cho dữ liệu được mã hóa 2110 dưới dạng thông tin tiêu đề phiến, thông tin tập hợp thông số viđeo (VPS), thông tin tập hợp thông số chuỗi (SPS), thông tin tập hợp thông số hình ảnh (PPS), số đếm tứ tự hình ảnh (POC), ký hiệu hình ảnh tham chiếu v.v... Trong một số cấu hình, môđun truyền tín hiệu bổ sung 2112 có thể tạo ra chỉ số bao gói chỉ báo sự quá độ giữa hai tập hợp hình ảnh.

Bộ mã hóa 2108 (và môđun truyền tín hiệu bổ sung 2112, ví dụ) có thể tạo ra dòng bit 2114. Dòng bit 2114 này có thể bao gồm dữ liệu hình ảnh được mã hóa dựa trên hình ảnh đầu vào 2106. Trong một số cấu hình, dòng bit 2114 này có thể bao gồm dữ liệu bổ sung, như thông tin tiêu đề phiến, thông tin VPS, thông tin SPS, thông tin PPS v.v... Do các hình ảnh đầu vào bổ sung 2106 được mã hóa, dòng bit 2114 có thể bao gồm một hoặc nhiều các hình ảnh được mã hóa. Chẳng hạn, dòng bit 2114 có thể bao gồm một hoặc nhiều các hình ảnh tham chiếu được mã hóa 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 bằng cách sử dụng kết nối hữu tuyến hoặc không dây. Trong một số trường hợp, việc này có thể được thực hiện qua mạng, như Internet hoặc mạng cục bộ(LAN). Như được minh họa trên Fig.63, bộ giải mã 2104 có thể được thực hiện trên thiết bị điện tử thứ hai 2102b riêng rẽ 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 tới bộ giải mã 2104 hoặc được lưu trữ trong bộ nhớ để lấy ra nhờ bộ giải mã 2104.

Bộ giải mã 2104 có thể thu (ví dụ, có được) dòng bit 2114. Bộ giải mã 2104 có thể tạo ra hình ảnh được mã hóa 2118 (ví dụ, một hoặc nhiều hình ảnh được mã hóa 2118) dựa trên dòng bit 2114. Hình ảnh được mã hóa 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 được mã hóa (DPB) 116. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể là bộ đệm giữ lại các hình ảnh được giải mã để tham chiếu, sắp xếp lại thứ tự kết xuất hoặc khoảng trễ kết xuất được xác định cho bộ mã hóa tham chiếu giả thiết (HRD). Trên thiết bị điện tử 102, bộ đệm hình ảnh được mã hóa (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 liên dự báo. Khi các hình ảnh được giải mã không đúng thứ tự, thì các

hình ảnh này có thể được lưu trữ trong bộ đệm hình ảnh được mã hóa (DPB) 116 theo đó sau này chúng có thể được hiển thị theo thứ tự.

Trong H.264 hoặc chuẩn mã hóa viđeo tiên tiến (AVC), việc quản lý bộ đệm hình ảnh được giải mã (DPB) 116 (ví dụ, xóa, bổ sung hình ảnh, sắp xếp lại thứ tự hình ảnh v.v...) được thực hiện bằng cách sử dụng các hoạt động kiểm soát quản lý bộ nhớ (MMCO). Nhiều phương pháp quản lý bộ đệm hình ảnh được 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 được mã hóa (DPB) hỗn hợp 2120. Môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 2120 chú ý đến các phương pháp quản lý bộ đệm hình ảnh được giải mã (DPB) 116 với các bước dựa trên cơ sở hình ảnh 2122 và/hoặc các phương pháp quản lý bộ đệm hình ảnh được giải mã (DPB) 116 với các bước dựa trên gói truy nhập (AU) 2124. Ví dụ, một ưu điểm của việc sử dụng các bước dựa trên hình ảnh 2124 đối với việc loại bỏ, lưu trữ và đánh dấu tham chiếu là bộ nhớ tối ưu của bộ đệm hình ảnh được 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 gói truy nhập (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 được mã hóa (DPB) hỗn hợp 116. Phương pháp 2200 này có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 dưới dạng 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 được mã hóa (DPB) hỗn hợp 2120. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể bắt đầu bước phân tách 2202 tiêu đề phiến thứ nhất của hình ảnh hiện thời. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể xác định 2204 các bước hoạt động nào của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp sẽ dựa vào hình ảnh và các bước nào sẽ dựa vào gói truy nhập (AU).

Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể thực hiện 2206 việc loại bỏ (không có kết xuất) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Việc loại bỏ này có thể loại bỏ các hình ảnh từ bộ đệm hình ảnh được mã hóa (DPB) 116 trước khi giải mã hình ảnh hiện thời. Bộ đệm hình ảnh được mã hóa (DPB) 116 này có thể thực hiện 2208 việc kết xuất hình ảnh (đệm) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Kết xuất hình ảnh (đệm) có thể chỉ việc kết xuất các hình ảnh từ bộ đệm hình ảnh được mã hóa (DPB) 116 vào thời gian loại bỏ của bộ đệm hình ảnh được mã hóa (CPB). Trong một số cấu hình, thuật ngữ đệm có thể được sử dụng để chỉ việc kết xuất của một hoặc nhiều hình ảnh từ bộ đệm hình ảnh được mã hóa (DPB) 116. Do vậy, các thuật ngữ đệm và kết xuất có thể được sử dụng thay đổi cho nhau.

Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể giải mã 2210 và lưu trữ hình ảnh hiện thời trong bộ đệm hình ảnh được mã hóa (DPB) 116. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể đánh dấu 2212 hình ảnh được mã hóa hiện thời được lưu trữ trong bộ đệm hình ảnh được mã hóa (DPB) 116 đó. Ví dụ, bộ đệm hình ảnh được mã hóa (DPB) 116 này có thể đánh dấu 2212 hình ảnh được mã hóa hiện thời là "không sử dụng để tham chiếu," "được sử dụng để tham chiếu" "cần kết xuất" hoặc "không cần kết xuất." Bộ đệm hình ảnh được mã hóa (DPB) 116 này cũng có thể thực hiện 2214 việc kết xuất hình ảnh khác nữa (đệm bổ sung) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Trong một số cấu hình, việc loại bỏ/đệm lặp lại của các hình ảnh từ bộ đệm hình ảnh được mã hóa (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 được mã hóa (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 được mã hóa (DPB) hỗn hợp 116. Phương pháp này 2300 có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 là 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 được mã hóa (DPB) hỗn hợp 2120. Bộ đệm hình ảnh được mã hóa (DPB) 116 này có thể bắt đầu bước phân tách 2302 tiêu đề phiến 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 được mã hóa (DPB) 116 được thực hiện trên cơ sở hình ảnh và một số bước hoạt động của bộ đệm hình ảnh được mã hóa (DPB) 116 được thực hiện trên cơ sở gói truy nhập (AU). Bộ đệm hình ảnh được mã hóa (DPB) 116 này có thể thực hiện 2304 việc loại bỏ dựa trên hình ảnh (không có kết xuất) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể thực hiện 2306 việc kết xuất (đệm) hình ảnh dựa trên gói truy nhập (AU) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể thực hiện 2308 việc giải mã dựa trên hình ảnh và lưu trữ của hình ảnh hiện thời trong bộ đệm hình ảnh được mã hóa (DPB).

Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể thực hiện 310 việc đánh dấu dựa trên hình ảnh đối với hình ảnh được mã hóa hiện thời trong bộ đệm hình ảnh được mã hóa (DPB). Bước đánh dấu được thực hiện nhờ bộ đệm hình ảnh được mã hóa (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 được mã hóa trong bộ đệm hình ảnh được mã hóa (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 khoảnh khắc được quy định bất kỳ trong khi vận hành quy trình giải mã. Việc ấn định một trong số các dấu hiệu này cho hình ảnh sẽ loại bỏ dứt khoát các dấu hiệu khác mà chúng được ấn định cho hình ảnh. Khi hình ảnh được đánh dấu là "được sử dụng để tham chiếu" thì điều này để chỉ chung cho hình 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 được mã hóa (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 hình ảnh hoặc trên cơ sở gói truy nhập (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 hình ảnh hoặc trên cơ sở gói truy nhập (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 gói truy nhập (AU).

Bộ đệm hình ảnh được mã hóa (DPB) 116 cũng có thể thực hiện 2312 việc đánh dấu kết xuất dựa trên gói truy nhập (AU) hình ảnh được mã hóa hiện thời. Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể thực hiện 2314 việc loại bỏ hình ảnh dựa trên gói truy nhập (AU) khác nữa (đệm bổ sung) từ bộ đệm hình ảnh được mã hóa (DPB) 116. Trong một số cấu hình, việc loại bỏ/đệm lặp lại của các hình ảnh từ bộ đệm hình ảnh được mã hóa (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 hạn chế sự phù hợp của dòng bit đối với các cờ hiệu trong tiêu đề phân đoạn phiến. Trong một số trường hợp, các hạn chế này có thể được áp dụng cho các cờ hiệu đối với các hình ảnh thuộc cùng một gói truy nhập (AU). Ví dụ, các cờ hiệu pic_output_flag và/hoặc no_output_of_prior_pics_flag có thể là cần thiết để tuân thủ các hạn chế sự phù hợp của dòng bit. Chẳng hạn, JCTVC-L1003, JCTVC-M1008 và JCT3V-D1004 mỗi tài liệu đều mô tả việc truyền tín hiện trong tiêu đề phân đoạn phiến bằng cách sử dụng pic_output_flag và no_output_of_prior_pics_flag. Noài ra, các cờ hiệu như PicOutputFlag, NoRaslOutputFlag và NoOutputOfPriorPicsFlag có thể suy ra được dựa trên các thành phần cú pháp và các loại gói 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 điều chỉnh của HEVC. JCT3V-D1004 MV-HEVC Draft Text 4 mô tả dự thảo văn bản cho phần mở rộng đa hình ảnh của HEVC.

Dòng bit được mã hóa viđeo, theo JCTVC-L1003, JCTVC-M1008 và/hoặc JCT3V-D1004, có thể bao gồm kết cấu cú pháp mà nó được đặt vào các bó dữ liệu lôgic được gọi chung là các gói của Lớp mạng trừu tượng (NAL). Mỗi gói NAL bao gồm tiêu đề gói NAL, như tiêu đề gói NAL hai byte (ví dụ, 16 bit), để xác định mục đích của chuỗi dữ liệu liên kết. Ví dụ, mỗi phiến được mã hóa (và/hoặc hình ảnh) có thể là mã hóa thành

một hoặc nhiều gói NAL của phiến (và/hoặc hình ảnh). Các gói NAL khác có thể được bao gồm dùng cho các hạng mục dữ liệu khác, như ví dụ, thông tin nâng cao bổ sung, phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời (TSA), phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời từng bước (STSA), phiến được mã hóa không TSA, hình ảnh kế tiếp không STSA, phiến được mã hóa của hình ảnh truy nhập liên kết gãy, phiến được mã hóa của hình ảnh làm tươi giải mã tức thời, phiến được mã hóa của hình ảnh truy nhập ngẫu nhiên sạch, phiến được mã hóa của hình ảnh chính có thể giải mã, phiến được mã hóa của được dán nhãn đối với hình ảnh loại bỏ, tập hợp thông số viđeo, tập hợp thông số chuỗi, tập hợp thông số hình ảnh, dấu phân tách gói truy nhập, đuôi của dãy, đuôi của dòng bit, dữ liệu bộ lọc và/hoặc thông báo thông tin nâng cao dãy. Bảng (7) dưới đây minh họa một ví dụ về các mã của gói NAL và các lớp của loại gói NAL. Các loại gói NAL khác cũng có thể được bao gồm, nếu cần.

Cũng cần phải hiểu là các trị số loại gói NAL đối với các gói NAL được trình bày trong bảng (7) có thể là được sắp xếp lại và được ấn định lại. Ngoài ra, các loại gói NAL bổ sung có thể được bổ sung hoặc loại bỏ.

Hình ảnh của điểm truy nhập ngẫu nhiên trong (IRAP) là hình ảnh được mã hóa mà đối với nó mỗi gói NAL lớp mã hóa viđeo có nal_unit_type nằm trong khoảng từ BLA_W_LP đến RSV_IRAP_VCL23, bao gồm cả trị số cuối, như được trình bày tại bảng (7) dưới đây. Hình ảnh IRAP chỉ bao gồm các phiến mã hóa trong (I).

Hình ảnh làm tươi giải mã tức thời (IDR) là hình ảnh IRAP mà đối với nó mỗi gói NAL lớp mã hóa viđeo has nal_unit_type bằng IDR_W_RADL hoặc IDR_N_LP, như được trình bày tại bảng (7). Hình ảnh làm tươi giải mã tức thời (IDR) chỉ bao gồm các phiến I và có thể là hình ảnh thứ nhất trong dòng bit trong thứ tự mã hóa, hoặc có thể xuất hiện sau trong dòng bit.

Mỗi hình ảnh IDR là hình ảnh thứ nhất của dãy viđeo được mã hóa (CVS) trong thứ tự mã hóa. Hình ảnh truy nhập liên kết gãy (BLA) là hình ảnh IRAP mà đối với nó mỗi gói NAL lớp mã hóa viđeo có nal_unit_type bằng BLA_W_LP, BLA_W_RADL, hoặc BLA_N_LP, như được trình bày tại bảng (7).

Hình ảnh BLA chỉ bao gồm các phiến I, và có thể là hình ảnh thứ nhất trong dòng bit trong thứ tự mã hóa, hoặc có thể xuất hiện sau trong dòng bit. Mỗi hình ảnh BLA bắt đầu dãy viđeo được 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 thành phần cú pháp mà chúng tập hợp hình ảnh tham chiếu không trống.

Bảng 42

nal_unit_type	**Tên của nal_unit_type**	**Nội dung gói NAL và kết cấu cú pháp chuỗi byte thô (RBSP)**	**Lớp của loại gói NAL**
0 1	TRAIL_ N TRAIL_R	Phân đoạn phiến được mã hóa của hình ảnh kéo không TSA, không STSA slice_segment_layer_rbsp()	Lớp mã hóa viđeo (VCL)
2 3	TSA_N TSA_R	Phân đoạn phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời (TSA) slice_segment_layer_rbsp()	VCL
4 5	STSA_N STSA_R	Phân đoạn phiến được mã hóa của hình ảnh truy nhập lớp phụ tạm thời từng bước (STSA) slice_segment_layer_rbsp()	VCL
6 7	RADL_N RADL_R	Phân đoạn phiến được mã hóa 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 phiến được mã hóa của hình ảnh chính 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 gói NAL VCL không tham chiếu lớp phụ không IRAP dự trữ	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Các loại gói NAL VCL tham chiếu lớp phụ không IRAP dự trữ	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_ N_LP	Phân đoạn phiến được mã hóa của hình ảnh truy nhập liên kết gãy (BLA) slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Phân đoạn phiến được mã hóa của hình ảnh làm tươi giải mã tức thời (IDR) slice. segment_layer_rbsp()	VCL
21	CRA_NUT	Phân đoạn phiến được mã hóa hình ảnh truy nhập ngẫu nhiên sạch (CRA) slice_segment_layer_rbsp()	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Các loại gói NAL VCL IRAP dự trữ	VCL
24..31	RSV_VCL24.. RSV_VCL31	Các loại gói NAL VCL không IRAP dự trữ	VCL
32	VPS_NUT	Tập hợp thông số viđeo Video_parameter_set_rbsp()	lớp mã hóa không viđeo (không VCL)
33	SPS_NUT	Tập hợp thông số dãy seq parameter_set_rbsp()	không VCL

34	PPS_NUT	Tập hợp thông số hình ảnh pic_parameter_ set rbsp()	không VCL
35	AUD_NUT	Dấu phân tách gói truy nhập access_unit_delimiter_rbsp()	không VCL
36	EOS_NUT	Đuôi của dãy 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 trữ	không VCL
48..63	UNSPEC48.. UNSPEC83	Không xác định	không VCL

Bảng (7)

Trên bảng (8) dưới đây, cú pháp tiêu đề gói NAL có thể bao gồm hai byte dữ liệu, gọi là, 16 bit. Bit thứ nhất có thể là "forbidden_zero_bit" mà nó luôn được cài đặt bằng zero tại điểm bắt đầu của gói NAL. Sáu bit tiếp theo có thể là "nal_unit_type" mà nó xác định loại kết cấu dữ liệu của các dãy byte thô (“RBSP”) được bao gồm trong gói NAL như được trình bày tại 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 đặc điểm cơ sở của tiêu chuẩn. Trong mã hóa viđeo có thể điều chỉnh được và/hoặc các phần mở rộng cú pháp, nuh_layer_id có thể xác định rằng gói NAL cụ thể này thuộc về lớp được xác định bởi trị số 6 bit này.

Thành phần cú pháp tiếp theo có thể là "nuh_temporal_id_plus1". nuh_temporal_id_plus1 trừ 1 có thể xác định nhãn hiệu tạm thời cho gói NAL đó. Nhãn hiệu tạm thời biến đổi được TemporalId có thể được xác định là TemporalId = nuh_temporal_id_plus1 - 1. Nhãn hiệu tạm thời TemporalId được sử dụng để xác định lớp phụ tạm thời. Biến HighestTid xác định lớp phụ tạm thời cao nhất sẽ phải 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 trình bày kết cấu cú pháp tập hợp thông số chuỗi (SPS) có tính ví dụ. pic_width_in_luma_samples xác định chiều rộng của mỗi hình ảnh được mã hóa trong các gói luma samples. pic_width_in_luma_samples không bằng 0. pic_height_in_luma_samples xác định chiều cao mỗi hình ảnh được mã hóa trong các gói luma samples. pic_height_in_luma_samples không bằng 0.

sps_max_sub_layers_minus1 cộng 1 xác định số lượng tối đa của các lớp phụ tạm thời mà nó có thể hiện diện trong mỗi CVS tham chiếu đến SPS. Trị số sps_max_sub_layers_minus1 nằm trong khoảng từ 0 đến 6, bao gồm cả trị số cuối.

Cờ hiệu sps_sub_layer_ordering_info_present_flag bằng 1 xác định rằng sps_max_dec_pic_buffering_minus1[i], sps_max_num_reorder_pics[i], và sps_max_latency_increase_plus1[i] các thành phần cú pháp hiện diện đối với các lớp phụ sps_max_sub_layers_minus1 + 1. sps_sub_layer_ordering_info_present_flag bằng 0 xác định rằng các trị số 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] cộng 1 xác định kích thước cần thiết tối đa của bộ đệm hình ảnh được mã hóa đối với CVS trong các gói của các bộ đệm lưu trữ hình ảnh khi HighestTid bằng i. Trị số sps_max_dec_pic_buffering_minus1[i] nằm trong khoảng từ 0 đến MaxDpbSize - 1, bao gồm cả trị số cuối trong đó MaxDpbSize xác định kích thước tối đa của bộ đệm hình ảnh được giải mã trong các gói của các bộ đệm lưu trữ hình ảnh. Khi i lớn hơn 0, thì sps_max_dec_pic_buffering_minus1[i] phải lớn

hơn hoặc bằng sps_max_dec_pic_buffering_minus1[i - 1]. Khi sps_max_dec_pic_buffering_minus1[i] không hiện diện cho i nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag bằng 0, thì nó được kết luận bằng sps_max_dec_pic_buffering_minus1[sps_max_sub_layers_minus1].

sps_max_num_reorder_pics[i] chỉ báo số lượng tối đa cho phép của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS trong thứ tự mã hóa và tiếp theo sau hình ảnh đó trong thứ tự kết xuất khi HighestTid bằng i. Trị số sps_max_num_reorder_pics[i] nằm trong khoảng từ 0 đến sps_max_dec_pic_buffering_minus1[i], bao gồm cả trị số cuối. Khi i lớn hơn 0, thì sps_max_num_reorder_pics[i] có thể là lớn hơn hoặc bằng sps_max_num_reorder_pics[i - 1]. Khi sps_max_num_reorder_pics[i] không hiện diện cho i nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag being bằng 0, thì nó được kết luận bằng sps_max_num_reorder_pics[sps_max_sub_layers_minus1].

sps_max_latency_increase_plus1[i] không bằng 0 có thể được sử dụng để tính toán trị số SpsMaxLatencyPictures[i], trị số này xác định số lượng tối đa của các hình ảnh mà chúng đứng trước hình ảnh bất kỳ trong CVS trong thứ tự kết xuất và tiếp theo sau hình ảnh đó trong thứ tự mã hóa khi HighestTid bằng i. Khi sps_max_latency_increase_plus1[i] không bằng 0, thì trị số 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 biểu diễn.

Trị số sps_max_latency_increase_plus1[i] nằm trong khoảng từ 0 đến 2^{32} - 2, bao gồm cả trị số cuối. Khi sps_max_latency_increase_plus1[i] không hiện diện cho i nằm trong khoảng từ 0 đến sps_max_sub_layers_minus1 - 1, bao gồm cả trị số cuối, do sps_sub_layer_ordering_info_present_flag being bằng 0, thì nó được kết luận bằng sps_max_latency_increase_plus1[sps_max_sub_layers_minus1].

Bảng 44

seq_parameter_set_rbsp() {
...
sps_max_sub_layers_minus1
...
pic_width_in_luma_samples
pic_height_in_luma_samples
...
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]
sps_max_num_reorder_pics[i]
sps_max_latency_increase_plus1[i]
}
...
}

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 trình bày trong bảng (10) dưới đây:

Bảng 45

CÚ PHÁP CỦA TIÊU ĐỀ PHÂN ĐOẠN PHIẾN CHUNG

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
...
...
}

Bảng (10)

Trong bảng (10), no_output_of_prior_pics_flag tác động đến kết xuất của các hình ảnh được mã hóa trước đây trong bộ đệm hình ảnh được mã hóa (DPB) sau bước giải mã

của IDR hoặc hình ảnh BLA mà nó 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 thành phần cú pháp pic_output_flag có hiện diện trong các tiêu đề phiến liên kết. output_flag_present_flag bằng 0 chỉ báo rằng thành phần cú pháp pic_output_flag không hiện diện trong các tiêu đề phiến liên kết. pic_output_flag tác động tới các quy trình kết xuất hình ảnh được mã hóa và loại bỏ như được xác định trong Phụ lục C của JCTVC-L1003. Khi pic_output_flag không hiện diện, thì nó được kết luận bằng 1.

Đối với qúa trình giải mã chung (như được nêu tại 8.1 của JCTVC-L1003), PicOutputFlag được cài đặt như sau:

- Nếu hình ảnh hiện thời là hình ảnh RASL và NoRaslOutputFlag của hình ảnh IRAP liên kết bằng 1, thì PicOutputFlag có thể được cài đặt bằng 0.

- Khác đi, PicOutputFlag có thể được cài đặt bằng pic_output_flag.

Ngoài ra, trong qúa 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), trị số PicOutputFlag cho hình ảnh tạo ra có thể được cài đặt bằng 0 trong các điều kiện nhất định.

Khi hình 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 trong thứ tự mã hóa hoặc hình ảnh thứ nhất tiếp theo sau đuôi gói NAL của dãy trong thứ tự mã hóa, biến NoRaslOutputFlag có thể được cài đặt bằng 1.

- Khác đi, nếu một số thiết bị bên ngoài chưa được xác định trong JCTVC-L1003 có thể sử dụng được để cài đặt biến HandleCraAsBlaFlag tại trị số cho hình ảnh hiện thời, biến HandleCraAsBlaFlag có thể được cài đặt bằng trị số được tạo ra bởi các thiết bị bên ngoài và biến NoRaslOutputFlag có thể được cài đặt bằng HandleCraAsBlaFlag.

- Khác đi, biến HandleCraAsBlaFlag có thể được cài đặt bằng 0 và biến NoRaslOutputFlag có thể được cài đặt bằng 0.

Như đã mô tả trên đây, gói truy nhập (AU) tham chiếu đến tập hợp các gói của lớp mạng trừu tượng (NAL) mà chúng được liên kết với nhau theo quy tắc phân loại xác định, mà chúng có tính liên tiếp trong thứ tự mã hóa, và chúng bao gồm các gói NAL của lớp mã hóa viđeo (VCL) của tất cả hình ảnh được mã hóa kết hợp với cùng thời gian kết xuất và các gói NAL không VCL liên kết của chúng. Lớp cơ sở thứ nhất là lớp trong đó tất cả các gói NAL VCL có nuh_layer_id bằng 0. Hình ảnh được mã hóa là sự trình bày được mã hóa của hình ảnh mà nó bao gồm các gói NAL VCL với trị số nuh_layer_id và chúng bao gồm tất cả các gói dạng cây mã hóa của hình ảnh. Trong một số trường hợp, hình ảnh được mã hóa có thể được gọi là thành phần lớp. Các chi tiết bổ sung về các bước dựa trên hình ảnh hoặc dựa trên gói truy nhập (AU) được trình bày liên quan đến các hình vẽ Fig.69 và Fig.70 dưới đây.

Trong một số cấu hình, các hạn chế sự 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 được mã hóa trong gói truy nhập (AU). Ngoài ra, ba cờ hiệu kết xuất mới của gói truy nhập, cờ hiệu kết xuất của AU (ví dụ, AuOutputFlag), cờ hiệu kết xuất không RASL AU (ví dụ, AuNoRaslOutputFlag) và không kết xuất AU của cờ hiệu các hình ảnh trước (ví dụ, AuNoOutputOfPriorPicsFlag), có thể được suy ra cho AU dựa trên trị số các thành phần cú pháp khác và các loại gói NAL đối với các hình ảnh được mã hóa trong AU. Trong một số cấu hình, việc kết xuất và loại bỏ các hình ảnh có thể dựa trên ba cờ hiệu bày (ví dụ, AuOutputFlag, AuNoRaslOutputFlag và AuNoOutputOfPriorPicsFlag) cho SHVC và HEVC đa hình ảnh.

Ví dụ, các hạn chế sự phù hợp của dòng bit đối với các phần mở rộng HEVC có thể tiếp theo sau như được mô tả trong các hệ thống và các phương pháp nêu trong bản mô tả này. Cụ thể, các hạn chế sự phù hợp của dòng bit có thể được áp dụng cho dòng bit SHVC. Ngoài ra, các hạn chế sự phù hợp của dòng bit có thể được áp dụng cho dòng bit MV-HEVC.

Theo một cấu hình, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến pic_output_flag có thể cần phải giống nhau trong tất cả các tiêu đề phân đoạn

phiến của các hình ảnh được mã hóa trong gói truy nhập (AU). Theo cấu hình khác, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến pic_output_flag, khi hiện diện, có thể giống nhau trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa trong gói truy nhập (AU) khi các hình ảnh được mã hóa có cùng gói NAL.

Theo một cấu hình, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến pic_output_flag cho các phân đoạn phiến với nuh_layer_id bằng trị số nuh_layer_id của lớp mục tiêu có thể giống nhau trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa đó trong gói truy nhập (AU). Theo cấu hình khác, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến pic_output_flag cho các phân đoạn phiến với nuh_layer_id không bằng trị số nuh_layer_id của lớp mục tiêu có thể bằng 0 trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa đó trong gói truy nhập (AU).

Theo một cấu hình, lớp mục tiêu có thể là lớp thuộc một tập hợp lớp hoặc tập hợp lớp mục tiêu hoặc tập hợp lớp ra như được xác định trong JCTVC-L1003, JCTVC-M1008, hoặc JCT3V-D1004. Theo cấu hình khác, lớp mục tiêu có thể là lớp dự định sẽ được giải mã. Theo cấu hình khác nữa, lớp mục tiêu có thể là lớp dự định sẽ được giải mã và kết xuất (được hiển thị hoặc khác đi thì được gửi để kết xuất).

Theo một số cấu hình, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến no_output_of_prior_pics_flag có thể giống nhau trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa trong gói truy nhập (AU). Trong các cấu hình khác, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến no_output_of_prior_pics_flag có thể giống nhau trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa trong gói truy nhập (AU) khi các hình ảnh được mã hóa có cùng gói NAL. Theo một cấu hình, khi hiện diện, trị số các thành phần cú pháp của tiêu đề phân đoạn phiến no_output_of_prior_pics_flag cho các phân đoạn phiến với nuh_layer_id bằng trị số nuh_layer_id của lớp mục tiêu có thể giống nhau trong tất cả các tiêu đề phân đoạn phiến của các hình ảnh được mã hóa đó trong gói truy nhập (AU).

Theo cấu hình khác, các thành phần 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 trị số cho các lớp với nuh_layer_id > 0 có thể được kết luận bằng các trị số được truyền tín hiệu đối với nuh_layer_id bằng 0.

Theo một số cấu hình, các cờ hiệu bổ sung, như AuOutputFlag và AuNoRaslOutputFlag có thể được sử dụng. Các cờ hiệu AuOutputFlag và AuNoRaslOutputFlag có thể nhận được theo nhiều phương pháp. Theo một phương pháp hoặc cấu hình, hai cờ hiệu, AuOutputFlag và AuNoRaslOutputFlag có thể nhận được từ và được sử dụng cho hoạt động của DPB. AuOutputFlag có thể được cài đặt bằng 1 nếu PicOutputFlag bằng 1 cho tất cả các hình ảnh trong AU. Khác đi AuOutputFlag có thể được cài đặt bằng 0.

Theo phương pháp khác, AuOutputFlag có thể được cài đặt bằng 1 nếu PicOutputFlag bằng 1 cho ít nhất một hình ảnh trong AU. Khác đi AuOutputFlag có thể được cài đặt bằng 0. Do vậy, trong trường hợp này, AuOutputFlag có thể được cài đặt bằng 0 nếu PicOutputFlag bằng 0 cho tất cả các hình ảnh trong AU.

Theo phương pháp khác nữa, AuOutputFlag có thể được cài đặt bằng 1 nếu PicOutputFlag bằng 1 cho các hình ảnh thuộc tất cả các lớp kết xuất mục tiêu trong AU. Khác đi thì AuOutputFlag có thể được cài đặt bằng 0.

Theo phương pháp khác nữa, AuOutputFlag có thể được cài đặt bằng 1 nếu PicOutputFlag bằng 1 cho hình ảnh thuộc ít nhất một trong số các lớp kết xuất mục tiêu trong AU. Khác đi AuOutputFlag có thể được cài đặt bằng 0.

Theo một phương pháp, AuNoRaslOutputFlag có thể được cài đặt bằng 1 nếu NoRaslOutputFlag bằng 1 cho tất cả các hình ảnh trong AU. Khác đi AuNoRaslOutputFlag có thể được cài đặt bằng 0.

Theo phương pháp khác, AuNoRaslOutputFlag có thể được cài đặt bằng 1 nếu NoRaslOutputFlag bằng 1 cho ít nhất một hình ảnh trong AU. Khác đi AuNoRaslOutputFlag có thể được cài đặt bằng 0. Do vậy, trong trường hợp này,

AuNoRaslOutputFlag có thể được cài đặt bằng 0 nếu NoRaslOutputFlag bằng 0 cho tất cả các hình ảnh trong AU.

Theo phương pháp khác nữa, AuNoRaslOutputFlag có thể được cài đặt bằng 1 nếu NoRaslOutputFlag bằng 1 cho các hình ảnh thuộc tất cả các lớp kết xuất mục tiêu trong AU. Khác đi AuNoRaslOutputFlag có thể được cài đặt bằng 0.

Theo phương pháp khác nữa, AuNoRaslOutputFlag có thể được cài đặt bằng 1 nếu NoRaslOutputFlag bằng 1 cho hình ảnh thuộc ít nhất một các lớp kết xuất mục tiêu trong AU. Khác đi AuNoRaslOutputFlag có thể được cài 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 hiệu kết xuất của AU, như AuOutputFlag và AuNoRaslOutputFlag, theo các hệ thống và các phương pháp hiện thời, được đưa ra dưới đây trong Danh sách (1A) và Danh sách (2A) dưới đây. Ngoài ra, như được mô tả dưới đây trong Danh sách (1), Danh sách (1A), Danh sách (2) và Danh sách (2A), cờ hiệu kết xuất của AU AuNoOutputOfPriorPicsFlag có thể nhận được 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 đến các tài liệu tiêu chuẩn. Tài liệu (1) dưới đây cung cấp các phần của JCTVC-L1003 mà chúng có thể được thay đổi để điều chỉnh các hệ thống và các phương pháp hiện thời.

Tài liệu 1

C.3 Hoạt động của bộ đệm hình ảnh được giải mã (DPB)

C.3.1 Những vấn đề chung

Các đặc tính kỹ thuật theo mục này áp dụng một cách độc lập đối với mỗi tập hợp các thông số DPB được lựa chọn như được chỉ ra ở mục C.1. DPB hoạt động riêng rẽ hoặc độc lập đối với mỗi lớp. Do vậy, các bước sau đây sẽ tiến hành riêng rẽ đối với mỗi

hình ảnh được giải mã với trị số cụ thể nuh_layer_id.

Bộ đệm hình ảnh được 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 liên kết với trị số nuh_layer_id của lớp đó. Mỗi bộ đệm lưu trữ hình ảnh có thể chứa hình ảnh được mã hóa là hình ảnh được đánh dấu là "sử dụng để tham chiếu" hoặc được giữ lại để kết xuất trong tương lai. Các quy trình được cài đặt theo các mục C.3.2, C.3.3 và C.3.4 được áp dụng theo trình tự cụ thể dưới đây.

C.3.2 Loại bỏ các hình ảnh từ DPB

Việc loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra tức thời tại thời gian loại bỏ CBP của gói giải mã thứ nhất của hình ảnh hiện thời thuộc gói truy nhập n (chứa hình ảnh hiện thời) và sẽ tiến hành như sau:

- Quy trình giải mã đối với RPS như được xác định ở mục 8.3.2 được viện dẫn.

- Khi hình ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1 mà nó không phải là hình ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm như sau:

- Nếu hình ảnh hiện thời là hình ảnh CRA, NoOutputOfPriorPicsFlag được cài đặt bằng 1 (không quan tâm đến trị số no_output_of_Prior_pics_flag).

- Khác đi, nếu trị số pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minusl[HighestTid] lần lượt được suy ra từ SPS hoạt động tương ứng với trị số nuh_layer_id của hình ảnh hiện thời lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minusl[HighestTid] được suy ra từ SPS hoạt động cho hình ảnh đứng trước với trị số nuh_layer_id bằng trị số nuh_Iayer_id của hình ảnh hiện thời, NoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được cài đặt bằng 1 bởi bộ giải mã đang xét nghiệm, không quan tâm đến trị số 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 đang xét nghiệm được cho phép cài đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

- Khác đi, thì NoOutputOfPriorPicsFlag được cài đặt bằng no_output_of_prior_pics_flag.

2. Trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HRD, sao cho khi trị số NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh_layer_id của hình ảnh hiện thời trong DPB được làm trống mà không có kết xuất của các hình ảnh mà chúng chứa và độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được cài đặt bằng 0.

Theo một phương án, trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR, sao cho khi trị số NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả trị số nuh_layer_id trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB cho tất cả các trị số nuh_layer_id được cài đặt bằng 0.

Theo một phương án trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR, sao cho khi trị số NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] với trị số currLayerId bằng nuh _layer _id của hình ảnh hiện thời trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB DPBFullness[currLayerId] cho trị số nuh_layer_id currLayerId của hình ảnh hiện thời được cài đặt bằng 0.

Theo một phương án trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR, sao cho khi trị số NoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] cho tất cả trị số nuh_layer_id trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB DPBFullness[nuh_layer_id] cho tất cả các trị số nuh_layer_id được cài 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 trị số nuh_layer_id của hình ảnh hiện thời trong DPB, tất cả các hình ảnh k đó trong DPB được loại bỏ từ DPB:

- hình ảnh k được đánh dấu "không sử dụng để tham chiếu"

- hình ảnh k có PicOutputFlag bằng 0 hoặc thời gian kết xuất DPB của nó nhỏ hơn hoặc bằng thời gian loại bỏ CPB của gói giải mã thứ nhất (được ký hiệu là gói 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 loại bỏ từ DPB, độ đầy của DPB bị giảm đi một.

- Theo một phương án, đối với mỗi hình ảnh k có trị số nuh_layer_id nuhLayerldk mà nó được loại bỏ từ DPB, độ đầy của DPB DPBfuliness[nuhLayerldk] bị giảm đi một.

C.3.3 Kết xuất hình ảnh

Các quy trình được xác định trong mục này xảy ra tức thì tại thời gian loại bỏ CBP của gói truy nhập n, AuCpbRemovalTime[n].

Khi hình 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 gói truy nhập n là gói truy nhập thứ nhất của khoảng thời gian đệm và khác đi 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 trị số của pic_dpb_output_delay trong thông báo SEI định thời hình ảnh liên quan với gói truy nhập n và picSptDpbOutputDuDelay là trị số của picsptDpbOutputdudelay, khi hiện diện, trong các thông báo SEI thông tin gói giải mã được liên kết với gói truy nhập n hoặc trị số pic_dpb_output_du_delay trong thông báo SEI định thời hình ảnh liên kết với gói truy nhập n khi không có thông báo SEI thông tin gói giải mã được liên kết với gói truy nhập n hoặc không có thông báo SEI thông tin gói giải mã được liên kết với gói truy nhập n có pic_spt_dpb_output_du_delay hiện diện.

Lưu ý - Khi thành phần cú pháp pic_spt_dpb_output_du_delay không hiện diện trong thông báo SEI thông tin gói giải mã bất kỳ được liên kết với gói truy nhập n, thì trị số được suy ra bằng pic_dpb_output_du_delay trong thông báo SEI định thời hình ảnh được liên kết với gói truy nhập n.

Việc kết xuất hình ảnh hiện thời nếu nuh_layer_id của nó thuộc lớp trong TargetDecLayerldList được xác định cụ thể như sau:

- Nếu PicOutputFlag bằng 1 và DpbOutputTime[n] bằng AuCpbRemovalTime[n], thì hình ảnh hiện thời được kết xuất.

- Khác đi, nếu PicOutputFlag bằng 0, thì hình ảnh hiện thời không được kết xuất, mà sẽ được lưu trữ trong bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh_layer_id của hình ảnh hiện thời trong DPB như được xác định ở mục C.3.4.

Theo một phương án: Khác đi, nếu PicOutputFlag bằng 0, thì hình 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 trị số nuh _laye _rid currLayerld của hình ảnh hiện thời trong DPB như được xác định ở mục C.3.4.

- Khác đi (PicOutputFlag bằng 1 và DpbOutputTime[n] lớn hơn AuCpbRemovalTime[n]), thì hình ảnh hiện thời được kết xuất trễ hơn và sẽ được lưu trữ vào trong bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh _layer _ id của hình ảnh hiện thời trong DPB (như được xác định ở mục C.3.4) và được kết xuất tại thời gian DpbOutputTime[n] trừ phi được chỉ báo là sẽ không được kết xuất bằng cách giải mã

hoặc suy luận no_output_of_prior_pics_flag bằng 1 tại thời gian đứng 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 ra 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 của hình ảnh hiện thời (không cần kiểm tra nếu như nó thuộc TargetDecLayerldList hoặc tập hợp lớp ra đối với điểm hoạt động hiện thời).

Khi kết xuất, thì hình ảnh bị cắt xén, bằng cách sử dụng cửa sổ cắt xén phù hợp được xác định trong SPS hoạt động cho hình ảnh đó.

Khi hình ảnh n là hình ảnh mà nó được kết xuất và không phải là hình ảnh cuối cùng của dòng bit được kết xuất, thì trị số của biến DpbOutputInterval[n] được suy ra như sau:

DpbOutputInterval[n] = DpbOutputTime[nextPicInOutputOrder] − DpbOutputTime[n](C-17)

trong đó nextPicInOutputOrder là hình ảnh tiếp theo sau hình ảnh n theo thứ tự kết xuất và có PicOutputFlag bằng 1.

C.3.4 Đánh dấu và lưu hình ảnh được giải mã hiện thời

Quy trình được xác định trong mục này xảy ra tức thời tại thời gian loại bỏ CBP của gói truy nhập n, CpbRemovalTime[n].

Hình ảnh được giải mã hiện thời được lưu trữ trong DPB trong bộ đệm lưu trữ hình ảnh trống tương ứng với trị số nuh_layer_id của hình ảnh hiện thời, độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được tăng thêm một và hình ảnh hiện thời được đánh dấu "sử dụng để tham chiếu ngắn hạn."

Theo một phương án:

Hình ảnh được 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 trị số nuh_layer_id của hình ảnh hiện thời trong DPB, độ đầy của DPB cho trị số nuh _layer _id của hình ảnh hiện thời DPBFullness[currLayerId] được tăng thêm một và hình ảnh hiện thời được đánh dấu "sử dụng để tham chiếu ngắn hạn."

C.4 Sự phù hợp của dòng bit

Các đặc tính kỹ thuật ở mục C.4 áp dụng.

C.5 Sự phù hợp của bộ giải mã

F.8.1.1C.5.1 Những vấn đề chung

Các đặc tính kỹ thuật ở mục C.5.1 áp dụng.

Tài liệu (1)

Như được sử dụng trong Tài liệu (1) nêu trên, PSB là để chỉ bộ đệm lưu trữ hình ảnh. DPBFullness là để chỉ biến được sử dụng để mô tả độ đầy của bộ đệm hình ảnh được giải mã (DPB) 116.

Tài liệu (1A) dưới đây đề xuất phương pháp thay thế cho phần C.3.2 của Tài liệu (1) để điều chỉnh các hệ thống và các phương pháp hiện thời. Trong một số cấu hình, Tài liệu (1A) có thể chỉ trình bày các thay đổi cho phần C.3.2 trong JCTVC-L1003. Tài liệu (1A) có thể sử dụng các cờ hiệu AuNoOutputOfPriorPicsFlag và AuNoRaslOutputFlag được xác định như phần trên đây.

Tài liệu 1A

C.3.2 Loại bỏ các hình ảnh từ DPB

Việc loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra tức thời tại thời gian loại bỏ CBP của gói giải mã thứ nhất của hình ảnh hiện thời thuộc gói truy nhập *n* (chứa hình ảnh hiện thời) và tiến hành như sau:

- Quy trình giải mã đối với RPS như được xác định ở mục 8.3.2 được viện dẫn.

- Khi hình ảnh hiện thời là hình ảnh IRAP với AuNoRaslOutputFlag bằng 1 mà nó không phải là hình ảnh 0, thì các bước theo thứ tự sau đây được áp dụng:

- Theo một cấu hình khác, khi hình ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1 mà nó không phải là hình ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm như sau:

- Nếu hình ảnh hiện thời là hình ảnh CRA, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 (không quan tâm đến trị số no__output_of_prior_pics_flag cho hình ảnh hiện thời hoặc các hình ảnh khác trong AU).

- Khác đi, nếu trị số pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_miiiusl[HighestTid] được suy ra từ SPS hoạt động lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_decj)ic_buffermg_mmusl[HighestTid], được suy ra từ SPS hoạt động cho hình ảnh đứng trước, thì AuNoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được cài đặt bằng 1 bởi bộ giải mã đang xét nghiệm, không quan tâm đến trị số no_output_ofjprior_pics_flag.

Lưu ý - mặc dù việc cài đặt AuNoOutputOfPriorPicsFlag bằng no_output_of_prior_pics_flag được ưu tiên trong các điều kiện này, bộ giải mã đang xét nghiệm được cho phép cài đặt AuNoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

- Khác đi, AuNoOutputOfPriorPicsFlag được cài đặt dựa trên trị số no_output_of_prior_pics_flag cho hình ảnh hiện thời và các hình ảnh khác trong AU như sau:

  - AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_ofjprior_pics_flag bằng 1 cho ít nhất một hình ảnh trong AU. Khác đi, AuNoOutputOfPriorPicsFlag được cài đặt bằng 0. Do đó trong trường hợp này AuNoOutputOfPriorPicsFlag được cài đặt bằng 0 nếu

no_output_of_prior_pics_flag bằng 0 cho tất cả các hình ảnh trong AU.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_ofjprior_pics_flag bằng 1 cho hình ảnh hiện thời. Trường hợp khác, AuNoOutputOfPriorPicsFlag giữ nguyên không đổi.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho tất cả các hình ảnh trong AU. Trường hợp khác AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of _prior _pics _flag bằng 1 cho tất cả các hình ảnh thuộc các lớp ra mục tiêu trong AU. Khác đi AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho ít nhất một hình ảnh thuộc các lớp ra mục tiêu trong AU. Khác đi AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

2. Trị số AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR, sao cho khi trị số AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả các trị số nul _ layer _id trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB cho tất cả trị số nuh_layer_id được cài đặt bằng 0.

Theo phương án khác, trị số AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR, sao cho khi trị số AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh_layer_id của hình ảnh hiện thời trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được cài đặt bằng 0.

Tài liệu (1A)

Tài liệu (2) dưới đây cung cấp các phần của JCTVC-L1008 mà chúng có thể được thay đổi để điều chỉnh các hệ thống và các phương pháp hiện thời.

Tài liệu 2

F.13 Bộ giải mã tham chiếu giả thiết

F.13.1 Những vấn đề chung

Các đặc tính kỹ thuật ở mục C. 1 được áp dụng.

F.13.2 Hoạt động của bộ đệm hình ảnh được giải mã (CPB)

Các đặc tính kỹ thuật in mục C.2 được áp dụng.

F.13.3 Hoạt động của bộ đệm hình ảnh được giải mã (DPB)

Các đặc tính kỹ thuật ở mục C.3 được áp dụng riêng rẽ cho từng tập hợp của các hình ảnh được mã hóa với trị số cụ thể nuh_layer_id.

PicOutputFlag cho các hình ảnh không được bao gồm trong lớp ra mục tiêu được cài đặt bằng 0.

Các hình ảnh được mã hóa với cùng thời gian kết xuất DPB và với PicOutputFlag bằng 1 được kết xuất theo thứ tự tăng lên của trị số nuh_layer_ids của các hình ảnh được mã hóa này.

F.13.5 Sự phù hợp của bộ giải mã

F. 13.5.1 Những vấn đề chung

Các đặc tính kỹ thuật ở 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 được 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 liên kết với trị số nuh _layer _id của lớp đó. Số lượng các bộ đệm lưu

trữ hình ảnh cho trị số nuh_layer_id bằng 0 được suy ra từ SPS hoạt động của lớp với trị số nuh_layer_id bằng 0. Số lượng các bộ đệm lưu trữ hình ảnh cho mỗi trị số nuh_layer_id khác 0 được suy ra từ SPS của lớp hoạt động cho trị số nuh_layer_id khác 0. Mỗi bộ đệm lưu trữ hình ảnh chứa hình ảnh được mã hóa được đánh dấu "sử dụng để tham chiếu" hoặc được giữ lại để kết xuất trong tương lai. Quy trình để kết xuất và loại bỏ các hình ảnh từ DPB như được xác định ở mục F.13.5.2.2 được viện dẫn, tiếp theo là sự viện dẫn quy trình đối với việc giải mã hình ảnh, việc đánh dấu, việc đệm bổ sung và việc lưu trữ như được xác định ở mục F.13.5.2.3. Quy trình "đệm" được xác định ở mục F.13.5.2.4 và được viện dẫn như được xác định ở các mục F.13.5.2.2 và F.13.5.2.3.

F. 13.5.2.2 Kết xuất và loại bỏ các hình ảnh từ DPB

Việc kết xuất và loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra tức thời khi gói giải mã thứ nhất của hình ảnh hiện thời thuộc gói truy nhập chứa hình ảnh hiện thời được loại bỏ từ CPB và tiến hành như sau.

Quy trình giải mã cho RPS như được xác định ở mục 8.3.2 được viện dẫn.

- Nếu hình ảnh hiện thời là hình ảnh IRAP với NoRaslOutputFlag bằng 1 và với trị số nuh_layer_id bằng 0 mà nó không phải là hình ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm như sau:

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

- Khác đi, khi trị số pic_width_in_luma_samples, pic_heightin_luma_samples hoặc sps max_dcc_pic_buffering_minus I [HighestTid] được suy ra từ SPS hoạt động tương ứng với trị số nuh_layer__id của hình ảnh hiện thời lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples hoặc sps_max_dec_pic_buffering_minusl[HighestTid], được suy ra từ SPS hoạt động

cho hình ảnh đứng trước với trị số nuh_layer_id bằng trị số nuh_layer_id của hình ảnh hiện thời, thì NoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được cài đặt bằng 1 bởi bộ giải mã đang xét nghiệm, không quan tâm đến trị số 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 đang xét nghiệm được cho phép cài đặt NoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

- Khác đi, NoOutputOfPriorPicsFlag được cài đặt bằng no_output_of_prior_pics_flag.

2. Trị số NoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm được áp dụng cho HDR như sau:

- Nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh_layer_id của hình ảnh hiện thời trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được cài 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 trị số nuh_layer_id trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB cho tất cả trị số nuh_layer_id được cài 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 trị số nuh_layer_id của hình ảnh hiện thời trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB DPBFullness[nuh_layer_id] được cài đặt bằng 0.

Theo một phương án, nếu NoOutputOfPriorPicsFlag bằng 1, tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh layer jd] cho tất cả các trị số nuh_layer_jd trong DPB được làm trống mà không có kết xuất của các hình ảnh chúng chứa và độ đầy của DPB

DPBFullness[nuh_layer_id] cho tất cả trị số nuh_layer_id được cài đặt bằng 0.

- Khác đi (NoOutputOfPriorPicsFlag bằng 0), thì tất cả các bộ đệm lưu trữ hình ảnh chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB tương ứng với trị số nuh_layer_id của hình ảnh hiện thời được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình "đệm" được xác định ở mục F. 13.5.2.4 và độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được cài đặt bằng 0.

Theo phương án khác, khi NoOutputOfPriorPicsFlag bằng 0, thì tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả các trị số nul _ layer _id chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống tương ứng với tất cả các trị số nul _ layer _id trong DPB được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình "đệm" được xác định ở mục F. 13.5.2.4 và độ đầy của DPB cho tất cả trị số nuh_layer_id được cài đặt bằng 0.

- Theo phương án khác, khác đi (NoOutputOfPriorPicsFlag bằng 0), thì tất cả các bộ đệm lưu trữ hình ảnh chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (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 trị số nuh_layer_id currLayerId của hình ảnh hiện thời được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở mục F.13.5.2.4 và độ đầy của DPB DPBFullness[nuh_layer_id] cho trị số nuh_layer_id của hình ảnh hiện thời được cài đặt bằng 0.

- Theo phương án khác, khi NoOutputOfPriorPicsFlag bằng 0, tất cả các bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id] cho tất cả trị số nuh_layer_id trong DPB chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (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 trị số nuh_layer_ids trong

DPB được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở mục F.13.5.2.4 và độ đầy của DPB DPBFullness[nuh_layer_id] cho tất cả trị số nuh_layer_id được cài đặt bằng 0.

- Khác đi (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 trị số nuh_layer_id của hình ảnh hiện thời chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm trống, độ đầy của DPB tương ứng với trị số nuh_layer_id của hình ảnh được giải mã hiện thời bị giảm đi một. Biến currLayerId được cài đặt bằng nuh_layer_id của hình ảnh được giải mã hiện thời và khi một hoặc một số các điều kiện sau đây là đúng, quy trình đệm được xác định ở mục F.13.5.2.4 được viện dẫn lặp đi lặp lại trong khi độ đầy của DPB bị giảm tiếp đi một đối với mỗi bộ đệm lưu trữ hình ảnh bổ sung tương ứng với trị số nuh_layer_id của hình ảnh được giải mã hiện thời được làm trống, 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_reorderjpics[HighestTid] từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho trị số currLayerId.

  - sps_max_latency_increase_plusl[HighestTid] của SPS hoạt động (khi currLayerId bằng 0) hoặc SPS của lớp hoạt động cho trị số currLayerId là không bằng 0 và có ít nhất một hình ảnh với nuh _layer _id bằng currLayerId trong DPB được đánh dấu "không cần kết xuất" mà đối với nó biến liên kết 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 trị số currLayerId.

  - Số lượng hình ảnh với nuh_layer_id bằng currLayerId trong DPB lớn hơn hoặc bằng sps_max_dec_pic_buffering_minusl[HighestTid] + 1 từ SPS hoạt động (khi currLayerId bằng 0) hoặc từ SPS của lớp hoạt động cho trị số currLayerId.

Theo phương án khác: Khác đi (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 trị số nuh_layer_id chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm trống, độ đầy của DPB tương ứng với trị số nuh_layer_id của hình ảnh bị giảm đi một.

Theo phương án khác: Khác đi (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 trị số nuh_layer_id currLayerId của hình ảnh hiện thời chứa hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm trống, độ đầy của DPB DPBfullness[nuh_layer_id] tương ứng với trị số nuh_layer_id của hình ảnh được giải mã hiện thời bị giảm đi một.

Theo phương án khác: Khác đi (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] cho tất cả trị số nuh_layer_id trong DPB chứa hình ảnh mà nó được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất). Đối với mỗi bộ đệm lưu trữ hình ảnh được làm trống, độ đầy của DPB DPBFullness[nuh_layer_id] tương ứng với trị số nuh_layer_id của hình ảnh trống bị giảm đi một.

F.13.5.2.3 Giải mã hình ảnh, đánh dấu, đệm bổ sung và lưu trữ

Các quy trình được xác định cụ thể trong mục này xảy ra tức thì khi gói giải mã cuối cùng của gói truy nhập n chứa hình ảnh hiện thời được loại bỏ từ CPB.

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

Đối với mỗi hình ảnh trong DPB được đánh dấu "không cần kết xuất" và có trị số nuh_layer_id bằng currLayerId, biến liên kết PicLatencyCo unt[currLayerId] được cài

đặt bằng PicLatencyCount[currLayerId] + 1.

Hình ảnh hiện thời được xem như là đã được giải mã sau khi gói giải mã cuối cùng của hình ảnh được giải mã. Hình ảnh được giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống tương ứng với currLayerId (trị số nuh_layer_id của hình ảnh hiện thời) trong DPB và áp dụng theo quy tắc như sau:

- Nếu hình ảnh được giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu "không cần kết xuất" và biến liên kết của nó PicLatencyCount[currLayerId] được cài đặt bằng 0.

- Khác đi (hình ảnh được giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu "không cần kết xuất."

Theo một phương án, hình ảnh được giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống tương ứng với currLayerId (trị số nuh _layer _id của hình ảnh hiện thời) trong DPB và áp dụng theo quy tắc như sau:

- Nếu hình ảnh được giải mã hiện thời có PicOutputFlag bằng 1, thì nó được đánh dấu "không cần kết xuất" và biến liên kết của nó PicLatencyCount[currLayerId] được cài đặt bằng 0. Tất cả các bộ đệm lưu trữ hình ảnh chứa hình ảnh có cùng trị số đếm thứ tự hình ảnh (PicOrderCntVal) như là hình ảnh được giải mã hiện thời được đánh dấu "không cần kết xuất."

- Trong trường hợp khác (hình ảnh được giải mã hiện thời có PicOutputFlag bằng 0), thì nó được đánh dấu "không cần kết xuất."

Theo một phương án, hình ảnh được giải mã hiện thời được lưu trữ trong bộ đệm lưu trữ hình ảnh trống PSB[curreLayerId] tương ứng với currLayerId (trị số nuh_layer_id của hình ảnh hiện thời) trong DPB, và độ đầy của DPB cho trị số nuh_layer _jd của hình ảnh hiện thời DPBFullness[nuh_layer_id] được tăng thêm một và áp dụng quy tắc như sau:

Hình ảnh được 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 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 các 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 trị số currLayerId.

- sps_max_latency_increase_plusl[HighestTid] không bằng 0 và có ít nhất một hình ảnh với nuh_layer_id bằng currLayerId trong DPB được đánh dấu "không cần kết xuất" cho biến liên kết 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 trị số 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 trước hết để kết xuất được lựa chọn làm các hình ảnh có trị số 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, bằng cách 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 trị số nuh_layer_id bằng trị số của hình ảnh, các hình ảnh được cắt xén được kết xuất theo thứ tự tăng lên của nuh_layer_id và các hình ảnh đó được đánh dấu "không cần kết xuất"

3. Mỗi bộ đệm lưu trữ hình ảnh chứa hình ảnh được đánh dấu "không sử dụng để tham chiếu" và mỗi bộ đệm lưu trữ hình ảnh đó bao gồm một hình ảnh mà đã được cắt xén và được kết xuất được làm trống.

Tài liệu (2)

Tài liệu (2A) dưới đây đề xuất phương pháp thay thế cho phần F.13.5.2.2 của Tài liệu (2) để điều chỉnh các hệ thống và các phương pháp hiện thời. Trong một số cấu hình, Tài liệu (2A) có thể chỉ trình bày các thay đổi cho phần F.13.5.2.2 trong JCTVC-L1008. Tài liệu (2A) có thể sử dụng các cờ hiệu AuNoOutputOfPriorPicsFlag và AuNoRaslOutputFlag được xác định như phần trên đây.

Tài liệu 2A

F.13.5.2.2 Kết xuất và loại bỏ các hình ảnh từ DPB

Việc kết xuất và loại bỏ các hình ảnh từ DPB trước khi giải mã hình ảnh hiện thời (nhưng sau khi phân tách tiêu đề phiến của phiến thứ nhất của hình ảnh hiện thời) xảy ra tức thời khi gói giải mã thứ nhất của gói truy nhập chứa hình ảnh hiện thời được loại bỏ từ CPB và tiến hành như sau:

Quy trình giải mã đối với RPS như được xác định ở 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 mà nó không phải là hình 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 mà nó không phải là hình ảnh 0, các bước theo thứ tự sau đây được áp dụng:

1. Biến AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét nghiệm như sau:

- Nếu hình ảnh hiện thời có nuh _layer _id bằng 0, thì AuNoOutputOfPriorPicsFlag được bắt đầu bằng 0. Khi đó:

- Nếu hình ảnh hiện thời là hình ảnh CRA, thì AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 (không quan tâm đến trị số no_output_of_prior_pics_flag cho hình ảnh hiện thời hoặc các hình ảnh khác trong AU).

- Khác đi, nếu trị số pic_width_in_luma_samples,

pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minusl[HighestTid] được suy ra từ SPS hoạt động lần lượt khác với trị số pic_width_in_luma_samples, pic_height_in_luma_samples, hoặc sps_max_dec_pic_buffering_minusl[HighestTid], được suy ra từ SPS hoạt động cho hình ảnh đứng trước, thì AuNoOutputOfPriorPicsFlag có thể (mà lẽ ra là không) được cài đặt bằng 1 bởi bộ giải mã đang xét nghiệm, không quan tâm đến trị số 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 đang xét nghiệm được cho phép cài đặt AuNoOutputOfPriorPicsFlag bằng 1 trong trường hợp này.

- Khác đi, AuNoOutputOfPriorPicsFlag được cài đặt dựa trên trị số no_output_of_prior_pics_flag cho hình ảnh hiện thời và các hình ảnh khác trong AU như sau:

- AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho ít nhất một hình ảnh trong AU. Khác đi, AUNoOutputOfPriorPicsFlag được cài đặt bằng 0. Do đó trong trường hợp này AuNoOutputOfPriorPicsFlag được cài đặt bằng 0 nếu no_output_of_prior_pics_flag bằng 0 cho tất cả các hình ảnh trong AU.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho hình ảnh hiện thời. Khác đi, AUNoOutputOfPriorPicsFlag giữ nguyên không đổi.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho tất cả các hình ảnh trong AU. Khác đi, AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1

nếu no_output_of_prior_pics_flag bằng 1 cho tất cả các hình ảnh thuộc các lớp ra mục tiêu trong AU. Khác đi, AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

- Theo phương án khác, AuNoOutputOfPriorPicsFlag được cài đặt bằng 1 nếu no_output_of_prior_pics_flag bằng 1 cho ít nhất một hình ảnh thuộc lớp ra mục tiêu trong AU. Khác đi, AUNoOutputOfPriorPicsFlag được cài đặt bằng 0.

2. Trị số AuNoOutputOfPriorPicsFlag được suy ra cho bộ giải mã đang xét 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 trị số nuh_layer_id được làm trống mà không có kết xuất của các hình ảnh mà chúng bao gồm và độ đầy của DPB được cài đặt bằng 0 cho tất cả trị số nuh_layer_id.
- Theo phương án khác, nếu AuNoOutputOfPriorPicsFlag bằng 1, thì tất cả các bộ đệm lưu trữ hình ảnh trong DPB tương ứng với tất cả trị số nuh_layer_id của hình ảnh hiện thời được làm trống mà không có kết xuất của các hình ảnh mà chúng bao gồm và độ đầy của DPB được cài đặt bằng 0 cho trị số nuh_layer_id của hình ảnh hiện thời.

Khác đi (AuNoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với tất cả trị số nuh_layer_id bao gồm hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống (mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở mục F.13.5.2.4 và độ đầy của DPB cho tất cả trị số nuh_layer_id được cài đặt bằng 0.

Khác đi, (AuNoOutputOfPriorPicsFlag bằng 0), tất cả các bộ đệm lưu trữ hình ảnh tương ứng với trị số nuh_layer_id của hình ảnh hiện thời bao gồm hình ảnh được đánh dấu "không cần kết xuất" và "không sử dụng để tham chiếu" được làm trống

(mà không có kết xuất) và tất cả các bộ đệm lưu trữ hình ảnh không trống trong DPB được làm trống bởi sự viện dẫn lặp đi lặp lại quy trình đệm được xác định ở mục F.13.5.2.4 và độ đầy của DPB cho trị số nuh_layer_id của hình ảnh hiện thời được cài đặt bằng 0.

Tài liệu (2A)

Theo một cấu hình, đối với văn bản dự kiến trong Tài liệu (1), Tài liệu (1A), Tài liệu (2) và Tài liệu (2A) nêu trên, tất cả những chỗ xuất hiện "hình ảnh hiện thời" có thể được thay thế bằng "hình ảnh được giải mã hiện thời".

Tài liệu (3) dưới đây cung cấp phần bổ sung đối với việc mã hóa để điều chỉnh các hệ thống và các phương pháp hiện thời. Tài liệu (3) cung cấp các phần của JCTVC-L1003 (nghĩa là tiêu chuẩn HEVC phiên bản 34) mà chúng có thể được thay đổi để điều chỉnh các hệ thống và các phương pháp hiện thời.

Tài liệu 3

8.3.2. Quy trình giải mã dùng cho tập hợp hình ảnh tham chiếu

Khi hình 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 trị số 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 hình 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 trị số nuh_layer_id (nếu có) được đánh dấu “không sử dụng để tham chiếu.”

Tài liệu (3)

Nhiều biến thể khác nhau có thể được sử dụng để xác định hoạt động của các bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 116 với nhiều bước khác nhau dựa trên hình ảnh hoặc dựa trên gói truy nhập. Các bảng (11) đến (15) dưới đây liệt kê nhiều biến thể của các bước đó và tương ứng với các bố trí dựa trên hình ảnh/dựa trên gói truy nhập (AU).

Bảng 46

Bước	**Dựa trên hình ảnh / Dựa trên AU**
Loại 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**
Loại 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**
Loại 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
Loại 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
Loại 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 thể hiện sơ đồ khối minh họa một cấu hình của bộ giải mã 2404. Bộ giải mã 2404 có thể được bao gồm trong thiết bị điện tử 2402. Ví dụ, bộ giải mã 2404 có thể là bộ giải mã mã hóa viđeo hiệu suất cao (HEVC). Bộ giải mã 2404 và/hoặc một hoặc nhiều thành phần được minh họa như được bao gồm 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 các hình ảnh được mã hóa được bao gồm trong dòng bit 2414) để mã hóa. Trong một số cấu hình, dòng bit thu được 2414 có thể bao gồm thông tin bổ sung thu được, như tiêu đề phiến thu được, PPS thu được, thông tin mô tả bộ đệm thu được v.v... Các hình ảnh được mã hóa được bao gồm trong dòng bit 2414 có thể bao gồm một hoặc nhiều các hình ảnh tham chiếu được mã hóa và/hoặc một hoặc nhiều các hình ảnh được mã hóa khác.

Các biểu tượng thu được (trong một hoặc nhiều các hình ảnh được mã hóa được bao gồm 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 được lượng tử hóa, được tạo tỷ lệ và/hoặc được biến đổi 458.

Tín hiệu thông tin chuyển động 456 có thể được kết hợp với một phần của 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 được lượng tử hóa, được tạo tỷ lệ và/hoặc được biến đổi 458 có thể được lượng tử hóa ngược, được tạo tỷ lệ và được biến đổi ngược bởi môđun ngược 462, nhờ đó tạo ra tín hiệu dư được giải mã 470. Tín hiệu dư được 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 lựa 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 lựa 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 được 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 được lọc sinh ra 482 có thể được ghi vào bộ nhớ khung 464. Tín hiệu được lọc sinh ra 482 có thể bao gồm hình ảnh được mã hóa.

Bộ nhớ khung 464 này có thể bao gồm bộ đệm hình ảnh được mã hóa (DPB) 2416 như được mô tả trong bản mô tả này. Bộ đệm hình ảnh được mã hóa (DPB) 2416 có thể có khả năng thực hiện các hoạt động của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 116. Bộ đệm hình ảnh được mã hóa (DPB) 2416 có thể bao gồm một hoặc nhiều hình ảnh được mã hóa 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 được mã hóa. Ví dụ, bộ nhớ khung 464 có thể bao gồm các tiêu đề phiến, thông tin tập hợp thông số viđeo (VPS), thông tin tập hợp thông số chuỗi (SPS), thông tin tập hợp thông số hình ảnh (PPS), các thông 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, bổ mã hóa 2108, môđun truyền tín hiệu bổ sung 2112).

Fig.67A thể hiện sơ đồ khối minh họa việc sử dụng cả lớp nâng cao và lớp cơ sở dùng để mã hóa viđeo với các bộ đệm hình ảnh được mã hóa (DPB) riêng rẽ 516a-b và các môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 520a-b tách biệt dùng 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 viđeo 508 mà nó 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 thứ nhất 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 nhận được hình ảnh đầu vào 2506. Trong một số cấu hình, hình ảnh đầu vào 2506 có thể được bắt giữ trên thiết bị điện tử thứ nhất 502a bằng cách sử dụng cảm biến hình ảnh, được lấy ra 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 được mã hóa. Ví dụ, bộ mã hóa lớp nâng cao 526 có thể mã hóa một loạt các hình ảnh đầu vào 2506 (ví dụ, viđeo). Dữ liệu được mã hóa có thể được bao gồm trong dòng bit viđeo nâng cao của lớp được 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 viđeo của lớp nâng cao 534 có thể bao gồm bộ đệm hình ảnh được mã hóa (DPB) 516a và môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 520a. Tương tự, bộ giải mã lớp cơ sở thứ nhất 536 có thể bao gồm bộ đệm hình ảnh được mã hóa (DPB) 516b và môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 520b.

Bộ mã hóa lớp cơ sở thứ nhất 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 thứ nhất 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 thứ nhất 528. Ví dụ, cho khả năng điều chỉnh tỷ lệ tín hiệu - nhiễu (còn được gọi là khả năng điều chỉnh 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 thứ nhất 528. Theo ví dụ khác, cho khả năng điều chỉnh không gian, hình ảnh được giảm mẫu có thể được sử dụng bởi bộ mã hóa lớp cơ sở thứ nhất 528. Theo ví dụ khác nữa, cho khả năng điều chỉnh đa hình ảnh, hình ảnh quan sát khác có thể được sử dụng bởi bộ mã hóa lớp cơ sở thứ nhất 528. Bộ mã hóa lớp cơ sở thứ nhất 528 có thể tạo ra dữ liệu được mã hóa được bao gồm trong dòng bit viđeo của lớp cơ sở được mã hóa 532.

Mỗi dòng bit viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532 có thể bao gồm dữ liệu được mã hóa dựa trên hình ảnh đầu vào 2506. Theo một ví dụ, dòng bit viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532 có thể bao gồm dữ liệu hình ảnh được mã hóa. Trong một số cấu hình, dòng bit viđeo của lớp nâng cao được mã hóa 530 và/hoặc dòng bit viđeo của lớp cơ sở được mã hóa 532 này có thể bao gồm dữ liệu bổ sung, như thông tin tập hợp thông số chuỗi (SPS), thông tin tập hợp thông số hình ảnh (PPS), thông tin tập hợp thông số viđeo (VPS), thông tin tiêu đề phiến v.v...

Dòng bit viđeo của lớp nâng cao được mã hóa 530 có thể được cung cấp cho thiết bị điện tử thứ hai 502b. Tương tự, dòng bit viđeo của lớp cơ sở được mã hóa 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 viđeo 2504. Bộ giải mã viđeo 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 viđeo của lớp cơ sở được mã hóa 530 được giải mã bởi bộ giải mã lớp cơ sở thứ nhất 536 trong khi dòng bit viđeo của lớp nâng cao được mã hóa 530 được giải mã bởi bộ giải mã lớp nâng cao 534.

Theo một ví dụ, dòng bit viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532 có thể được truyền đến thiết bị điện tử thứ hai 502b bằng cách sử dụng kết nối hữu tuyến hoặc không dây. Trong một số trường hợp, việc nàys 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 thứ nhất 528) và bộ

giải mã 2504 (ví dụ, bộ giải mã lớp cơ sở thứ nhất 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 phần 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 viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532 có thể sử dụng được cho bộ giải mã viđeo 2504 theo nhiều cách thức. Ví dụ, dòng bit viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532 có thể được cung cấp qua bus tới bộ giải mã viđeo 2504 hoặc được lưu trữ trong bộ nhớ để lấy ra nhờ bộ giải mã viđeo 2504.

Bộ giải mã viđeo 2504 có thể tạo ra một hoặc nhiều hình ảnh được mã hóa dựa trên dòng bit viđeo của lớp nâng cao được mã hóa 530 và dòng bit viđeo của lớp cơ sở được mã hóa 532. Hình ảnh được mã hóa 2118 (nó có thể bao gồm hình ảnh được giải mã của lớp nâng cao 538 và hình ảnh được 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 được mã hóa 2118 có thể được truyền đến thiết bị khác hoặc quay trở về thiết bị điện tử thứ nhất 502a. Hình ảnh được mã hóa 2118 cũng có thể được lưu trữ hoặc khác đi 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 được mã hóa 2118. Trong các cấu hình khác, hình ảnh được mã hóa 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 được mã hóa 2118 có thể được bao gồm trong chuỗi hình 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 thể hiện sơ đồ khối minh họa việc sử dụng bộ đệm hình ảnh (DPB) được chia sẻ 516c và môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 520c được chia sẻ 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 viđeo của lớp nâng cao 534 và bộ giải mã lớp cơ

sở thứ nhất 536 chia sẻ cả bộ đệm hình ảnh được mã hóa (DPB) 516c và môđun vận hành bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 520c.

Fig.68 thể hiện biểu đồ định thời mô tả hoạt động của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 116. Các hoạt động của bộ đệm hình ảnh được mã hóa (DPB) hỗn hợp 116 trên Fig.68 trình bày các bước của biến thể được ưu tiên trong đó việc loại 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 gói truy nhập (AU). Bộ đệm hình ảnh được giải mã (DPB) 116 lý tưởng và bộ giải mã tham chiếu giả thiết (HRD) có thể hoạt động theo cách thức mà nhiều bước đơn lẻ được thể hiện (ví dụ, bước loại bỏ 621a-b, bước kết xuất 623, bóc lưu trữ 625a-c, bước đánh dấu 627a-c, 629 và bước kết xuất bổ sung 631) tất cả đều được thực hiện tức thời. 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 sự phù hợp của bộ mã hóa 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 cho cả bộ giải mã tham chiếu giả thiết (HRD) và bộ giải mã đang xét nghiệm (DUT) cho tới khoảng trễ cố định. Do đó, dịch vị định thời được minh họa có thể xảy ra cho DUT mà nó tạo ra khoảng trễ cố định so với HRD.

Các bước cho bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a, bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b và bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644 được minh họa. Thời gian loại bỏ của bộ đệm hình ảnh được mã hóa (CPB) 2646 được minh họa. Sau thời gian loại bỏ 2646 của bộ đệm hình ảnh được mã hóa (CPB), việc loại bỏ (không có kết xuất) dựa trên hình ảnh 621a có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b và việc loại bỏ (không có kết xuất) dựa trên hình ảnh 621b có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644. Sau khi dịch vị định thời, việc kết xuất (đệm) hình ảnh dựa trên gói truy nhập (AU) 623 có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644, bộ đệm hình ảnh được mã

hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b.

Sau dịch vị định thời khác, quá trình 613 liên quan đến hình ảnh được mã hóa hiện thời được minh họa. Bước lưu trữ dựa trên hình ảnh 625a-c có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644, bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh được mã hóa (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 thứ nhất (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, hình ảnh được mã hóa được lưu trữ trong bộ đệm hình ảnh được mã hóa (DPB) 116 trong bộ đệm lưu trữ trống và độ đầy của bộ đệm hình ảnh được mã hóa (DPB) được tăng thêm một. Ngoài ra, khi hình ảnh được loại bỏ (không có kết xuất) từ bộ đệm hình ảnh được mã hóa (DPB) 116, thì độ đầy của bộ đệm hình ảnh được mã hóa (DPB) được giảm bớt một. Tương tự, khi hình ảnh được đệm từ bộ đệm hình ảnh được mã hóa (DPB) 116 (trong hoặc bước đệm hoặc bước đệm bổ sung), độ đầy của bộ đệm hình ảnh được mã hóa (DPB) được giảm bớt một.

Bộ đệm hình ảnh được mã hóa (DPB) 116 có thể bao gồm các bộ đệm hình ảnh được xác định và quản lý một cách riêng rẽ đối với các hình ảnh được mã hóa có các đặc điểm khác nhau. Ví dụ, bộ đệm hình ảnh được mã hóa (DPB) 116 có thể bao gồm các bộ đệm hình ảnh được xác định và quản lý một cách riêng rẽ đối với các hình ảnh được mã hóa với các độ phân giải khác nhau, độ sâu khác nhau và/hoặc màu sắc khác nhau.

Hình ảnh được mã hóa có thể được lưu trữ trong một nhóm bộ đệm lưu trữ hình ảnh chung trong bộ đệm hình ảnh được mã hóa (DPB) 116. Ví dụ, hai trường hợp phụ bổ sung có thể được sử dụng để xác định các hạn chế kích thước của bộ đệm hình ảnh được mã hóa (DPB) 116 mà nó tác động quá trình đệm/loại bỏ và xác định cấp độ. Theo hạn chế của bộ đệm hình ảnh được giải mã (DPB) 116 dựa trên byte, thì hình ảnh được mã hóa có thể được lưu trữ với cân nhắc về kích thước dựa trên độ phân giải và/hoặc độ sâu

bit. Các hạn chế kích thước của bộ đệm hình ảnh được mã hóa (DPB) 116 có để được xác định theo giới hạn byte mà nó cân nhắc độ phân giải và độ sâu bit của mỗi hình ảnh được mã hóa. Theo hạn chế của bộ đệm hình ảnh được giải mã (DPB) 116 dựa trên đơn vị hình ảnh, thì hình ảnh được mã hóa có thể được lưu trữ (và được cân nhắc chiếm một khe của bộ đệm hình ảnh). Các hạn chế kích thước của bộ đệm hình ảnh được mã hóa (DPB) 116 sau đó có thể được xác định theo giới hạn số lượng các khe hình mà không cần cân nhắc đến độ phân giải và độ sâu bit của mỗi hình ảnh được mã hóa.

Theo một cấu hình, độ đầy của bộ đệm hình ảnh được mã hóa (DPB) có thể được theo dõi theo từng lớp. Ví dụ, các hạn chế kích thước của bộ đệm hình ảnh được mã hóa (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 độ đầy của bộ đệm hình ảnh được mã hóa (DPB) của mỗi lớp. Khi hình ảnh được loại bỏ từ lớp với trị số ID của lớp bằng nuh_layer_id, thì biến DPBFullness[nuh_layer_id] có thể được cài đặt bằng DPBFullness[nuh_layer_id]-1 (nghĩa là DPBFullness[nuh_layer_id] có thể được giảm bớt một). Trong trường hợp này, thì hình ảnh đó đã được loại bỏ từ bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id].

Tương tự, khi hình ảnh được giải mã hiện thời với trị số ID của lớp bằng nuh_layer_id được lưu trữ trong bộ đệm hình ảnh được mã hóa (DPB) 116, thì biến DPBFullness[nuh_layer_id] được cài đặt bằng DPBFullness[nuh_layer_id]+1 (nghĩa là DPBFullness[nuh_layer_id] được tăng thêm một). Trong trường hợp này, hình ảnh đó đã được lưu trữ vào bộ đệm lưu trữ hình ảnh PSB[nuh_layer_id].

Độ đầy của bộ đệm hình ảnh được mã hóa (DPB) cũng có thể được theo dõi đối với tập hợp lớp ra. Các hạn chế kích thước của bộ đệm hình ảnh được mã hóa (DPB) 116 sau đó có thể được truyền tín hiệu, và bước đệm có thể được áp dụng, dựa trên các hạn chế được xác định cho tập hợp lớp ra. Trị số DPBFullness có thể được theo dõi đối với tập hợp lớp ra mà nó được kết hợp với điểm hoạt động đang xét nghiệm. Do vậy, khi hình ảnh được loại bỏ từ lớp thuộc tập hợp lớp ra, thì trị số độ đầy của bộ đệm hình ảnh được mã hóa (DPB) có thể được giảm bớt một vì DPBFullness = DPBFullness-1. Tương tự,

khi hình ảnh được giải mã hiện thời được lưu trữ trong bộ đệm hình ảnh được mã hóa (DPB) 116, thì độ đầy của bộ đệm hình ảnh được mã hóa (DPB) có thể được giảm bớt một vì DPBFullness = DPBFullness+1.

Trong quá trình 613 liên quan đến hình ảnh được mã hóa 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 được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644, bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b. Sau dịch vị định thời khác, bước đánh dấu kết xuất dựa trên gói truy nhập (AU) 629 có thể được thực hiện bởi bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644, bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b. Khi quá trình 613 liên quan đến hình ảnh được mã hóa hiện thời được hoàn thành, bước 631 của việc kết xuất hình ảnh dựa trên gói truy nhập (AU) (đệm bổ sung) có thể được thực hiện nhờ bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp cơ sở thứ nhất (BL) 2644, bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ nhất (EL1) 2642a và bộ đệm hình ảnh được mã hóa (DPB) 116 của lớp nâng cao thứ hai (EL2) 2642b.

Fig.69 thể hiện sơ đồ khối minh họa kết cấu và định thời cho các gói của lớp mạng trừu tượng (NAL) của các lớp cho các hình ảnh được mã hóa và các gói truy nhập (AU) khi lớp nâng cao thứ hai (EL2) 942b có tốc độ ảnh thấp hơn lớp cơ sở thứ nhất (BL) 944 và lớp nâng cao thứ nhất (EL1) 942a. Các gói NAL của hình ảnh được mã hóa EL 1 953a được minh họa cùng với lớp nâng cao thứ nhất (EL1) 942a. Các gói NAL của hình ảnh được mã hóa EL 2 953b được minh họa cùng với lớp nâng cao thứ hai (EL2) 942b. Các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c được minh họa cùng với lớp cơ sở thứ nhất (BL) 944.

Tại thời gian t1, các gói NAL của hình ảnh được mã hóa EL 1 953a, các gói NAL của hình ảnh được mã hóa EL 2 953b và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c là phần của gói truy nhập (AU) 955a. Tại thời gian t2, các gói NAL của hình ảnh được mã hóa EL 1 953a và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c là phần của gói truy nhập (AU) 955b. Tại thời gian t3, các gói NAL của hình ảnh được mã hóa EL 1 953a, các gói NAL của hình ảnh được mã hóa EL 2 953b và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c là phần của gói truy nhập (AU) 955c. Tại thời gian t4, các gói NAL của hình ảnh được mã hóa EL 1 953a và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c là phần của gói truy nhập (AU) 955d.

Fig.70 thể hiện sơ đồ khối minh họa kết cấu và định thời cho các gói của lớp mạng trừu tượng (NAL) của các lớp cho các hình ảnh được mã hóa và các gói truy nhập (AU) khi lớp cơ sở thứ nhất (BL) 1044 có tốc độ ảnh thấp hơn lớp nâng cao thứ nhất (EL1) 1042a và lớp nâng cao thứ hai (EL2) 1042b. Các gói NAL của hình ảnh được mã hóa EL 1 1053a được minh họa cùng với lớp nâng cao thứ nhất (EL1) 1042a. Các gói NAL của hình ảnh được mã hóa EL 2 1053b được minh họa cùng với lớp nâng cao thứ hai (EL2) 1042b. Các gói NAL của hình ảnh được mã hóa của lớp cơ sở 1053c được minh họa cùng với lớp cơ sở thứ nhất (BL) 1044.

Tại thời gian t1, các gói NAL của hình ảnh được mã hóa EL 1 1053a, các gói NAL của hình ảnh được mã hóa EL 2 1053b và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 1053c là phần của gói truy nhập (AU) 1055a. Tại thời gian t2, các gói NAL của hình ảnh được mã hóa EL 1 1053a và các gói NAL của hình ảnh được mã hóa EL2 1053b là phần của gói truy nhập (AU) 1055b. Tại thời gian t3, các gói NAL của hình ảnh được mã hóa EL 1 1053a, các gói NAL của hình ảnh được mã hóa EL 2 1053b và các gói NAL của hình ảnh được mã hóa của lớp cơ sở 1053c là phần của gói truy nhập (AU) 1055c. Tại thời gian t4, các gói NAL của hình ảnh được mã hóa EL 1 1053a và các gói NAL của hình ảnh được mã hóa EL 1 1053b là phần của gói truy nhập (AU) 1055d.

Thuật ngữ "vật ghi đọc được bằng máy tính" tham chiếu đến vật ghi bất kỳ có thể sử dụng được mà nó 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, vật ghi đọc được bằng máy tính và/hoặc vật ghi đọc được bằng bộ xử lý mà chúng có tính chất không tạm thời và hữu hình. Bằng cách ví dụ, và không hạn chế, 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), đĩa laze, đĩa quang, đĩa đa năng số (DVD), đĩa mềm và đĩa bluray (nhãn hiệu được đăng ký) trong đó đĩa mềm thường sao chép lại dữ liệu bằng phương pháp từ, trong khi các đĩa kia sao chéo dữ liệu theo phương pháp quang với tia laze.

Cần lưu ý rằng một hoặc nhiều các phương pháp được mô tả trong bản mô tả này có thể được thực hiện 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. Nói cách khác, trừ phi thứ tự cụ thể của các bước hoặc các hoạt động là cần thiết đối với việc vận hành chính xác hoạt động của phương pháp này mà nó đang được mô tả, thứ tự và/hoặc việc sử dụng các bước và /hoặc các hoạt động cụ thể đó có thể được cải biến mà không nằm ngoài phạm vi bảo hộ.

Cần hiểu rằng 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.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dòng bit viđeo, phương pháp này bao gồm các bước:

(a) thu tập hợp các thông số định thời hình ảnh bao gồm thông số trễ loại bỏ bộ đệm hình ảnh được mã hóa để loại bỏ gói truy nhập từ bộ đệm hình ảnh được mã hóa ở cấp độ gói truy nhập;

(b) thu dữ liệu đại diện của ảnh viđeo được mã hóa;

(c) lưu trữ dữ liệu này trong bộ đệm hình ảnh được mã hóa;

(d) xác định liệu gói truy nhập này có cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa này hay không để giải mã ở cấp độ gói truy nhập hoặc gói giải mã có cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa hay không để giải mã ở cấp độ hình ảnh phụ;

(e) khi gói giải mã này cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa để giải mã ở cấp độ hình ảnh phụ thì:

(i) thu cờ chỉ báo liệu dòng bit có bao gồm thông số đại diện cho độ trễ loại bỏ thông thường cho tất cả các gói giải mã trong gói truy nhập,

(ii) nếu cờ này chỉ báo rằng thông số đại diện cho độ trễ loại bỏ thông thường cho tất cả các gói giải mã có ở trong dòng bit, thì thu thông số đại diện cho độ trễ loại bỏ thông thường cho tất cả các gói giải mã trong đó độ trễ loại bỏ thông thường này là khoảng thời gian giữa thời điểm loại bỏ gói giải mã hiện thời ra khỏi bộ đệm hình ảnh được mã hóa và thời điểm loại bỏ gói giải mã kề ngay trước đó ra khỏi bộ đệm hình ảnh được mã hóa,

(iii) nếu cờ này chỉ báo rằng thông số đại diện cho độ trễ loại bỏ thông thường cho tất cả các gói giải mã không ở trong dòng bit, thì thu các thông số trễ riêng rẽ của gói giải mã mà mỗi thông số này đại diện cho độ trễ giữa các gói giải mã liên tiếp trong gói truy nhập,

(iv) loại bỏ gói giải mã khỏi bộ đệm hình ảnh được mã hóa dựa trên thông số đại diện cho hoặc độ trễ loại bỏ thông thường hoặc các thông số trễ riêng rẽ của gói giải mã, và

(v) giải mã gói giải mã được loại bỏ này; và

(f) khi gói truy nhập cần được loại bỏ khỏi bộ đệm hình ảnh được mã hóa để giải mã ở cấp độ gói truy nhập thì:

(i) xác định độ trễ loại bỏ của gói truy nhập đối với gói truy nhập này trong dữ liệu,

(ii) loại bỏ gói truy nhập này khỏi bộ đệm hình ảnh được mã hóa này dựa trên độ trễ của gói truy nhập này, và

(iii) giải mã gói truy nhập được loại bỏ này,

trong đó

gói giải mã là tập hợp con của gói truy nhập này, và

trị số tích tắc đồng hồ, t_c, nhân với thông số trễ loại bỏ bộ đệm hình ảnh được mã hóa dùng để loại bỏ gói truy nhập khỏi bộ đệm hình ảnh được mã hóa này ở cấp độ gói truy nhập là bằng với trị số tích tắc đồng hồ của hình ảnh phụ, $t_{c,sub}$, nhân với tổng của tất cả các thông số trễ riêng rẽ của gói giải mã trong gói truy nhập này, trong đó độ trễ loại bỏ gói truy nhập được biểu diễn bằng số lượng các tích tắc đồng hồ, và các thông số trễ của gói giải mã được biểu diễn bằng các số lượng tích tắc đồng hồ của hình ảnh phụ.

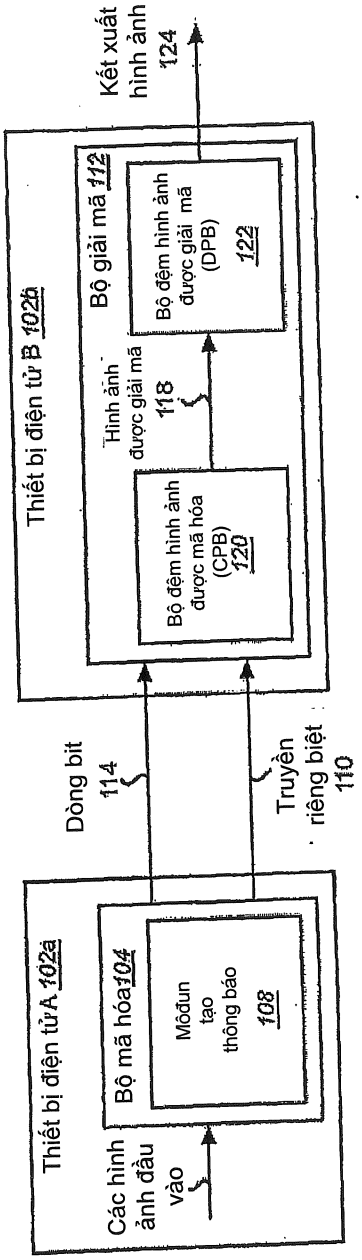

Thiết bị điện tử A 102a
Các hình ảnh đầu vào
Bộ mã hóa 104
Modun tạo thông báo 108
Dòng bit 114
Truyền riêng biệt 110
Thiết bị điện tử B 102b
Bộ giải mã 112
Bộ đệm hình ảnh được mã hóa (CPB) 120
Hình ảnh được giải mã 118
Bộ đệm hình ảnh được giải mã (DPB) 122
Kết xuất hình ảnh 124

FIG. 1A

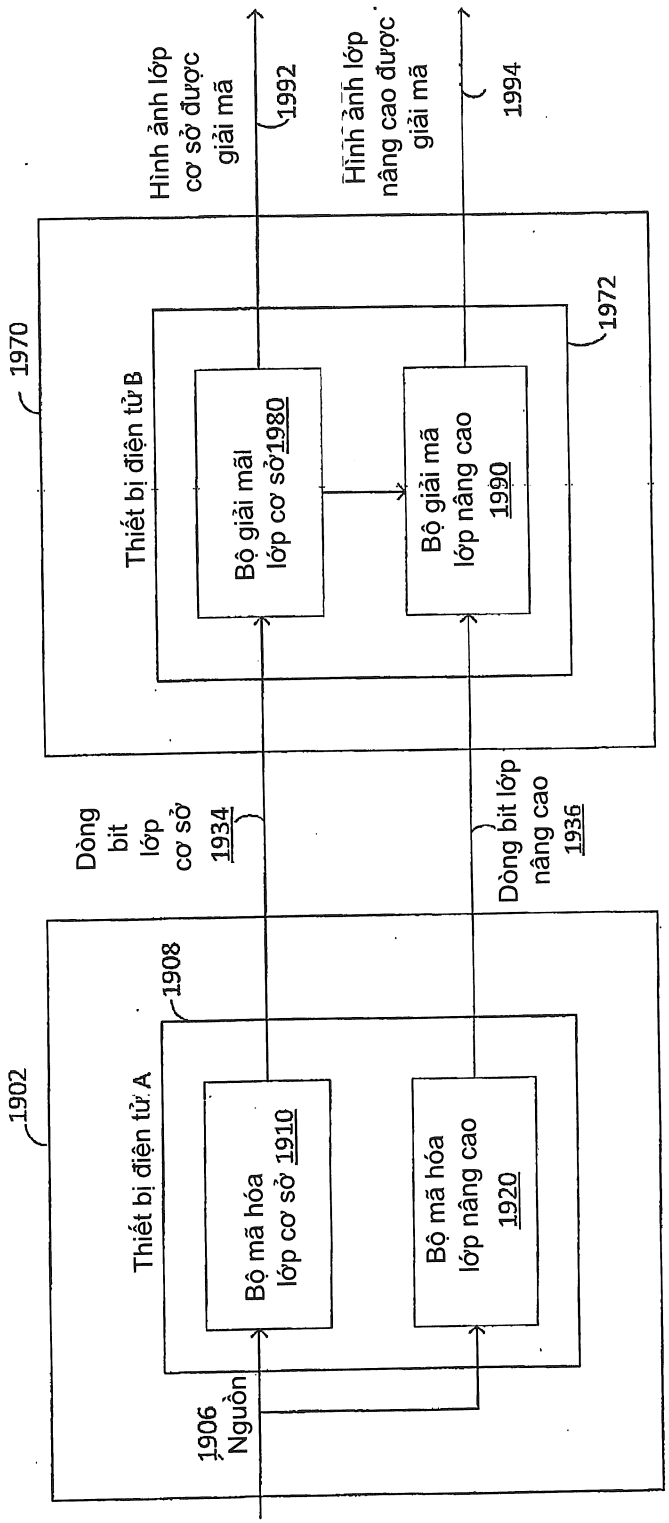

1902
Thiết bị điện tử A
1908
Bộ mã hóa lớp cơ sở 1910
Bộ mã hóa lớp nâng cao 1920
1906 Nguồn
Dòng bit lớp cơ sở 1934
Dòng bit lớp nâng cao 1936
1970
Thiết bị điện tử B
1972
Bộ giải mã lớp cơ sở 1980
Bộ giải mã lớp nâng cao 1990
Hình ảnh lớp cơ sở được giải mã
1992
Hình ảnh lớp nâng cao được giải mã
1994

FIG. 1B

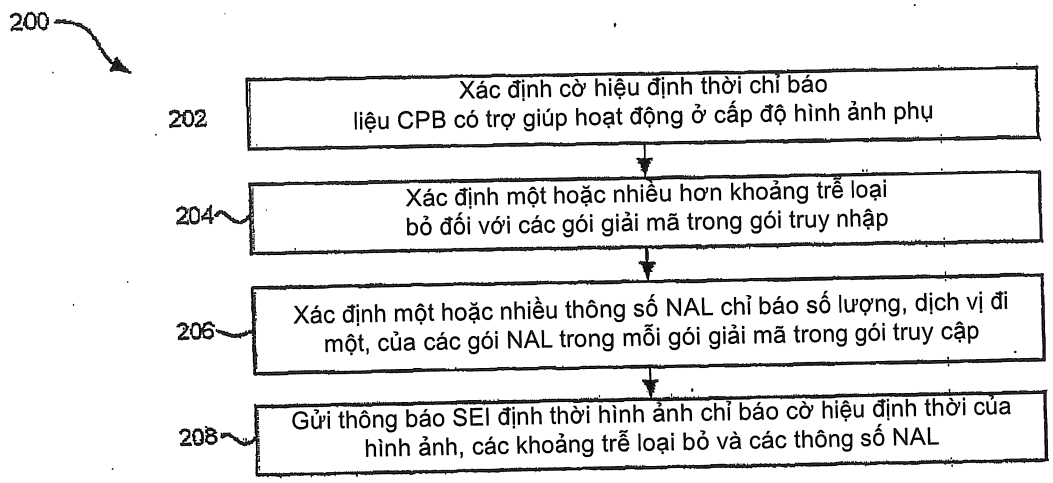

200
202
Xác định cờ hiệu định thời chỉ báo liệu CPB có trợ giúp hoạt động ở cấp độ hình ảnh phụ
204
Xác định một hoặc nhiều hơn khoảng trễ loại bỏ đối với các gói giải mã trong gói truy nhập
206
Xác định một hoặc nhiều thông số NAL chỉ báo số lượng, dịch vị đi một, của các gói NAL trong mỗi gói giải mã trong gói truy cập
208
Gửi thông báo SEI định thời hình ảnh chỉ báo cờ hiệu định thời của hình ảnh, các khoảng trễ loại bỏ và các thông số NAL

FIG. 2

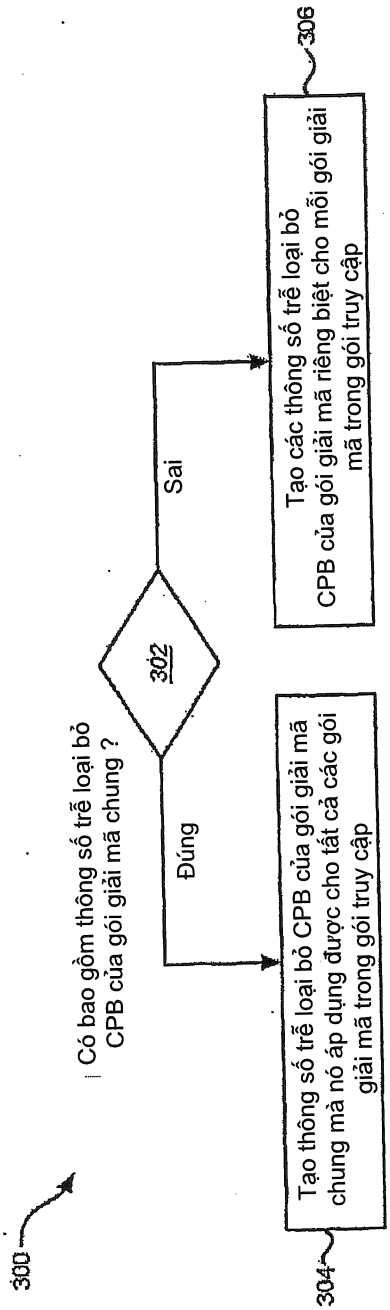

300
Có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung ?
302
Đúng
Sai
304
Tạo thông số trễ loại bỏ CPB của gói giải mã chung mà nó áp dụng được cho tất cả các gói giải mã trong gói truy cập
306
Tạo các thông số trễ loại bỏ CPB của gói giải mã riêng biệt cho mỗi gói giải mã trong gói truy cập

FIG. 3

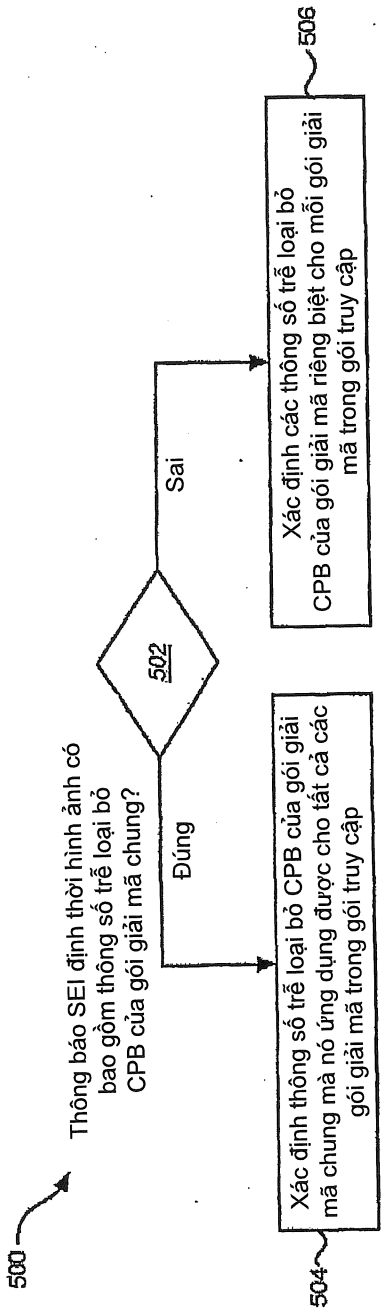

500
Thông báo SEI định thời hình ảnh có bao gồm thông số trễ loại bỏ CPB của gói giải mã chung?
502
Đúng
Sai
504
Xác định thông số trễ loại bỏ CPB của gói giải mã chung mà nó ứng dụng được cho tất cả các gói giải mã trong gói truy cập
506
Xác định các thông số trễ loại bỏ CPB của gói giải mã riêng biệt cho mỗi gói giải mã trong gói truy cập

FIG. 5

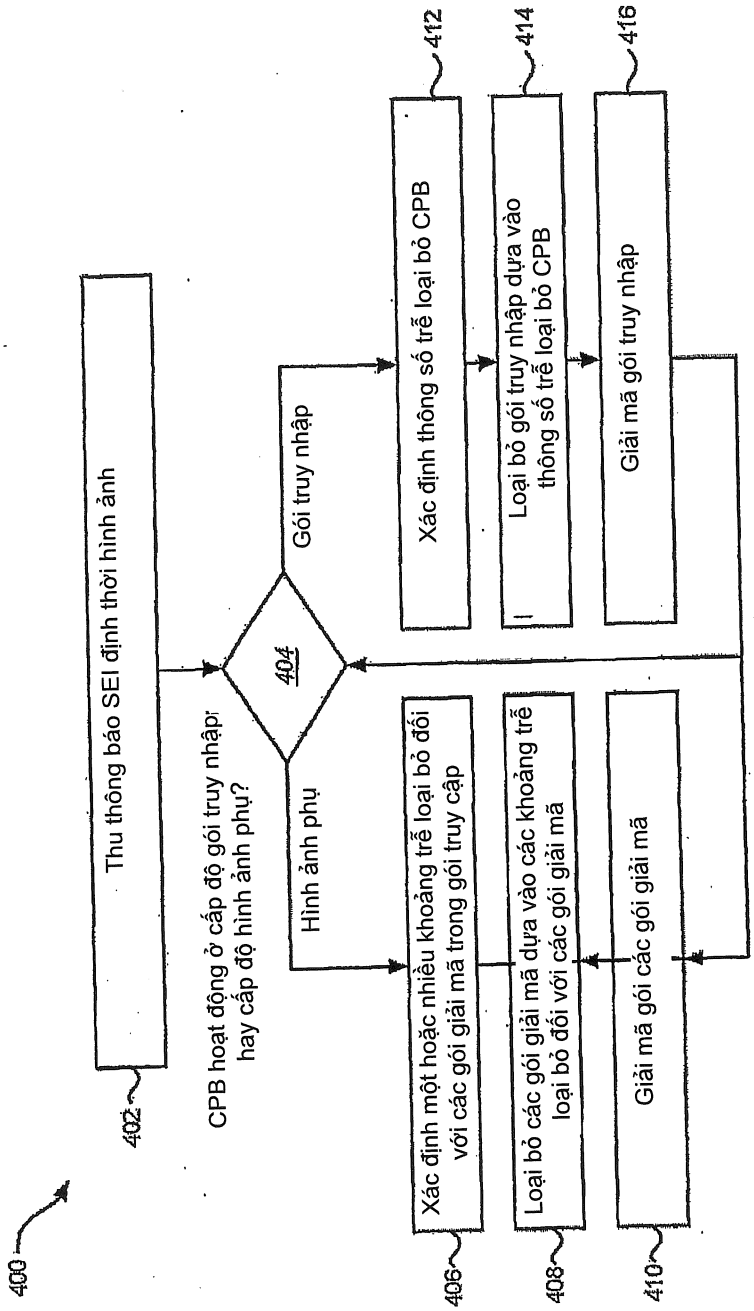

400
Thu thông báo SEI định thời hình ảnh
402
CPB hoạt động ở cấp độ gói truy nhập hay cấp độ hình ảnh phụ?
404
Gói truy nhập
Hình ảnh phụ
Xác định thông số trễ loại bỏ CPB
412
Loại bỏ gói truy nhập dựa vào thông số trễ loại bỏ CPB
414
Giải mã gói truy nhập
416
Xác định một hoặc nhiều khoảng trễ loại bỏ đối với các gói giải mã trong gói truy cập
406
Loại bỏ các gói giải mã dựa vào các khoảng trễ loại bỏ đối với các gói giải mã
408
Giải mã gói các gói giải mã
410

FIG. 4

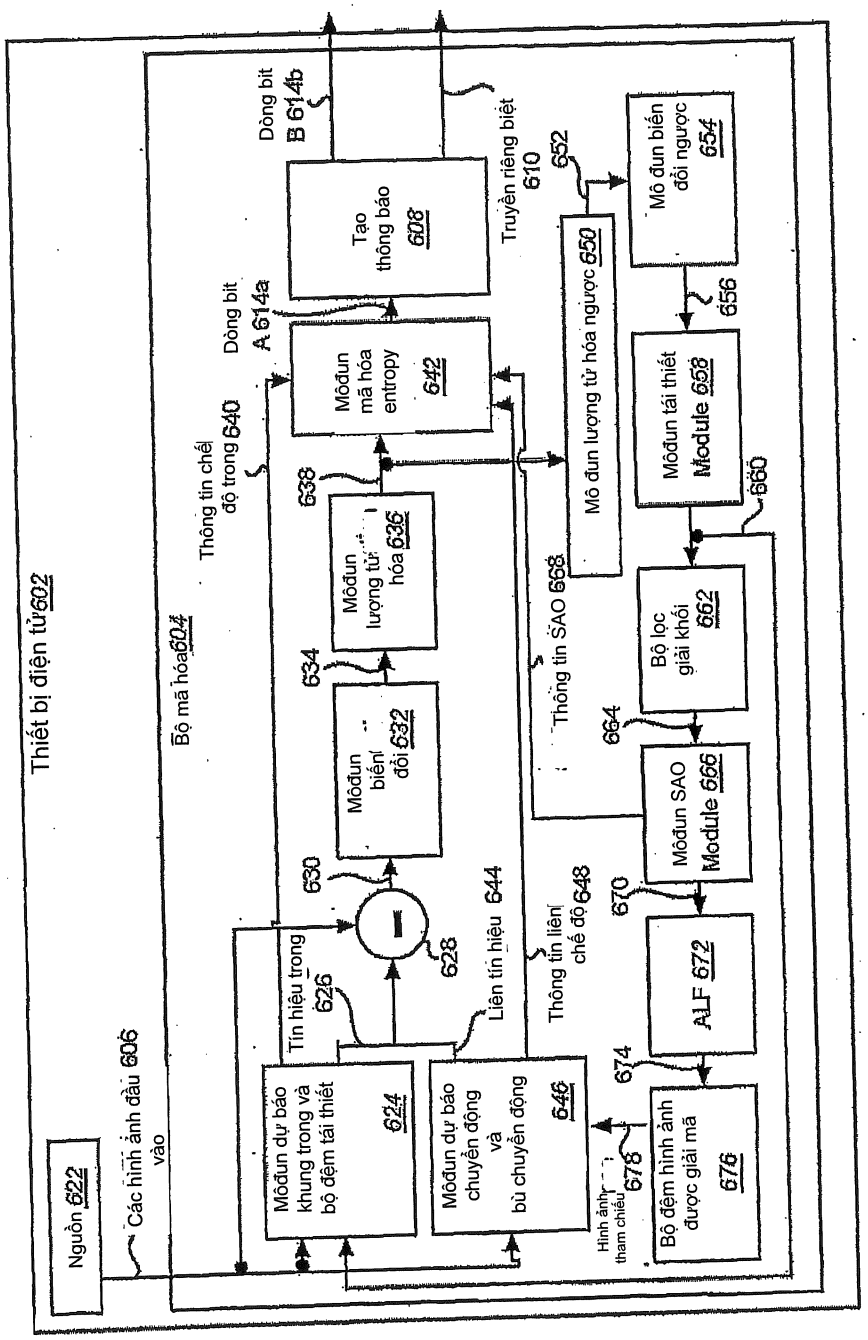
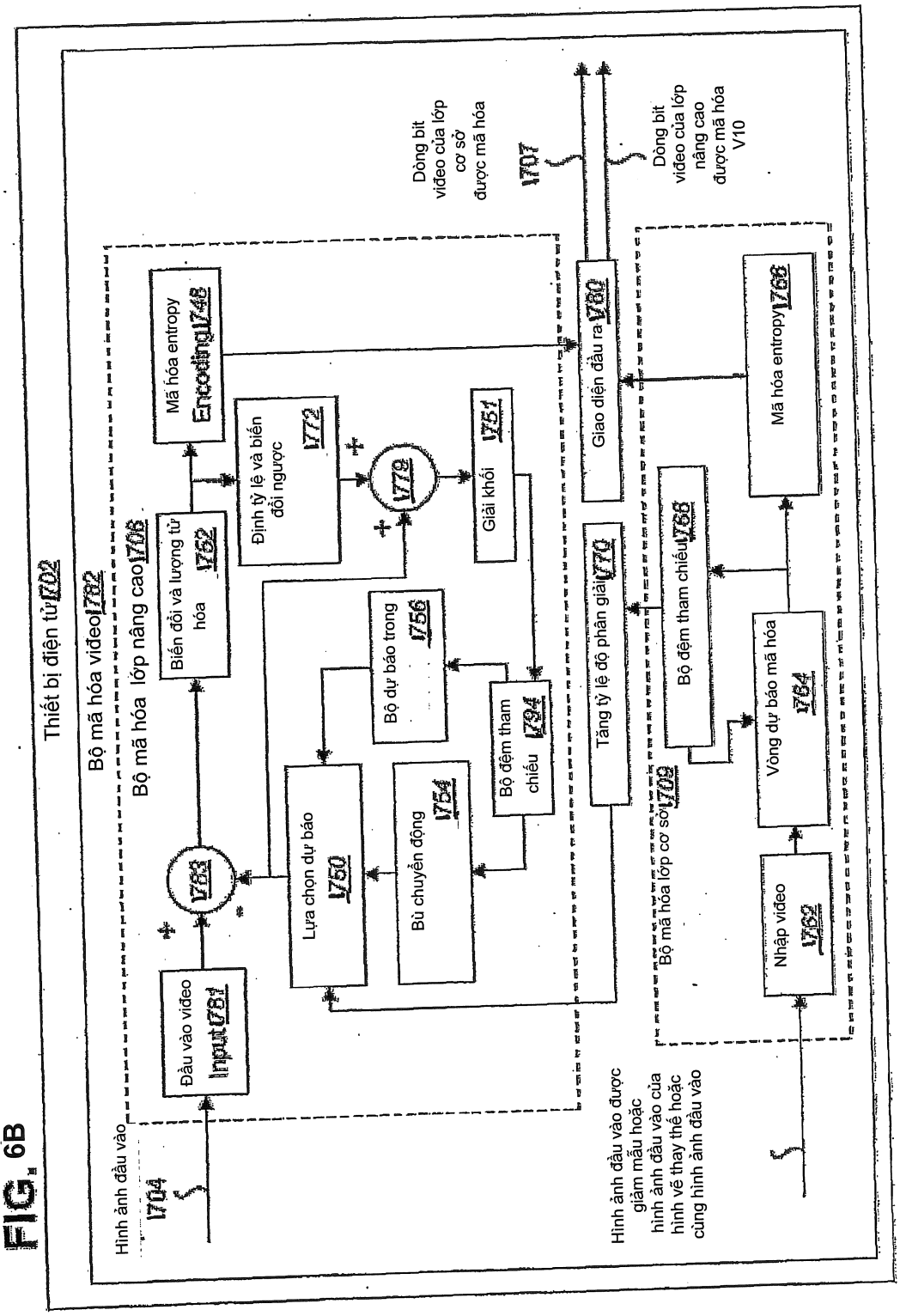

Thiết bị điện tử 602
Nguồn 622
Các hình ảnh đầu vào 606
Bộ mã hóa 604
Thông tin chế độ trong 640
Dòng bit A 614a
Dòng bit B 614b
Truyền riêng biệt 610
Tạo thông báo 608
Mô đun mã hóa entropy 642
Mô đun lượng tử hóa 636
Mô đun biến đổi 632
Tín hiệu trong 626
Liên tín hiệu 644
Thông tin liên chế độ 648
Thông tin SAO 668
Mô đun lượng tử hóa ngược 650
Mô đun biến đổi ngược 654
Mô đun tái thiết Module 658
Bộ lọc giải khối 662
Mô đun SAO Module 666
ALF 672
Mô đun dự báo khung trong và bộ đệm tái thiết 624
Mô đun dự báo chuyển động và bù chuyển động 646
Hình ảnh tham chiếu
Bộ đệm hình ảnh được giải mã 676

FIG. 6A

FIG. 6B


Thiết bị điện tử 1702
Bộ mã hóa video 1782
Bộ mã hóa lớp nâng cao 1708
Hình ảnh đầu vào
1704
Đầu vào video Input 1781
1783
Biến đổi và lượng tử hóa 1762
Mã hóa entropy Encoding 1748
Định tỷ lệ và biến đổi ngược 1772
1779
Giải khối 1751
Lựa chọn dự báo 1750
Bù chuyển động 1754
Bộ dự báo trong 1756
Bộ đếm tham chiếu 1794
Tăng tỷ lệ độ phân giải 1770
Giao diện đầu ra 1780
Dòng bit video của lớp cơ sở được mã hóa 1707
Dòng bit video của lớp nâng cao được mã hóa V10
Bộ mã hóa lớp cơ sở 1709
Bộ đệm tham chiếu 1766
Mã hóa entropy 1768
Vòng dự báo mã hóa 1764
Nhập video 1762
Hình ảnh đầu vào được giảm mẫu hoặc hình ảnh đầu vào của hình vẽ thay thế hoặc cùng hình ảnh đầu vào

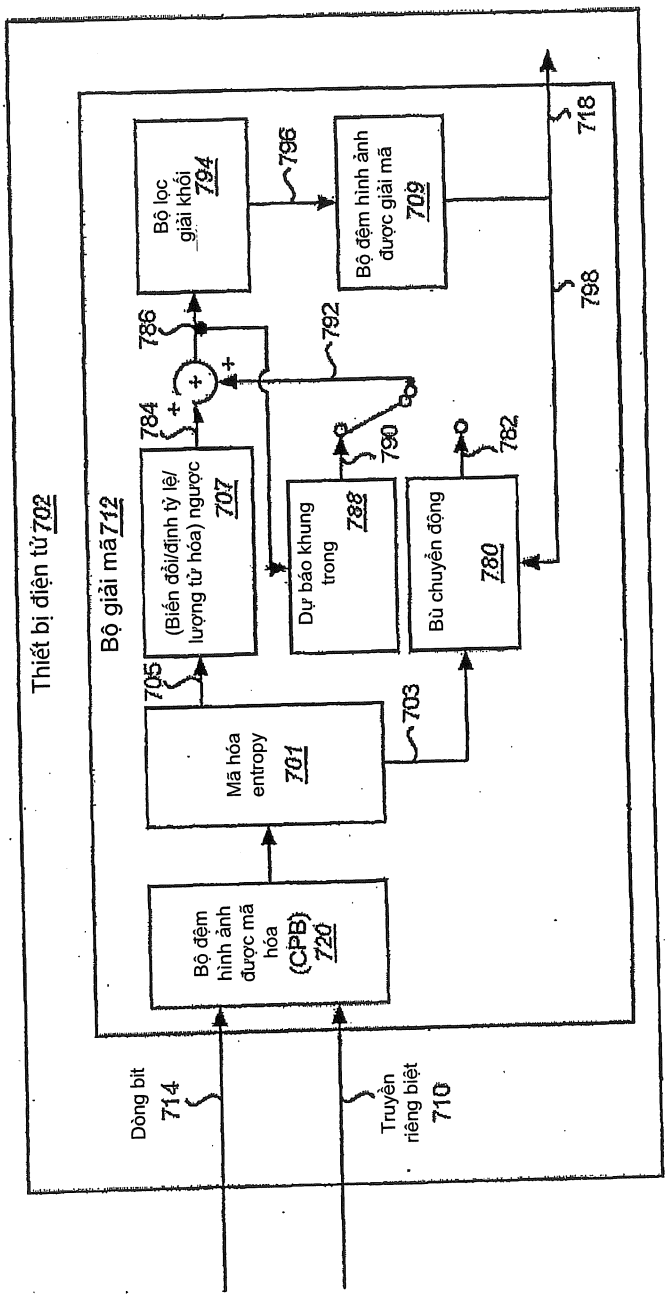
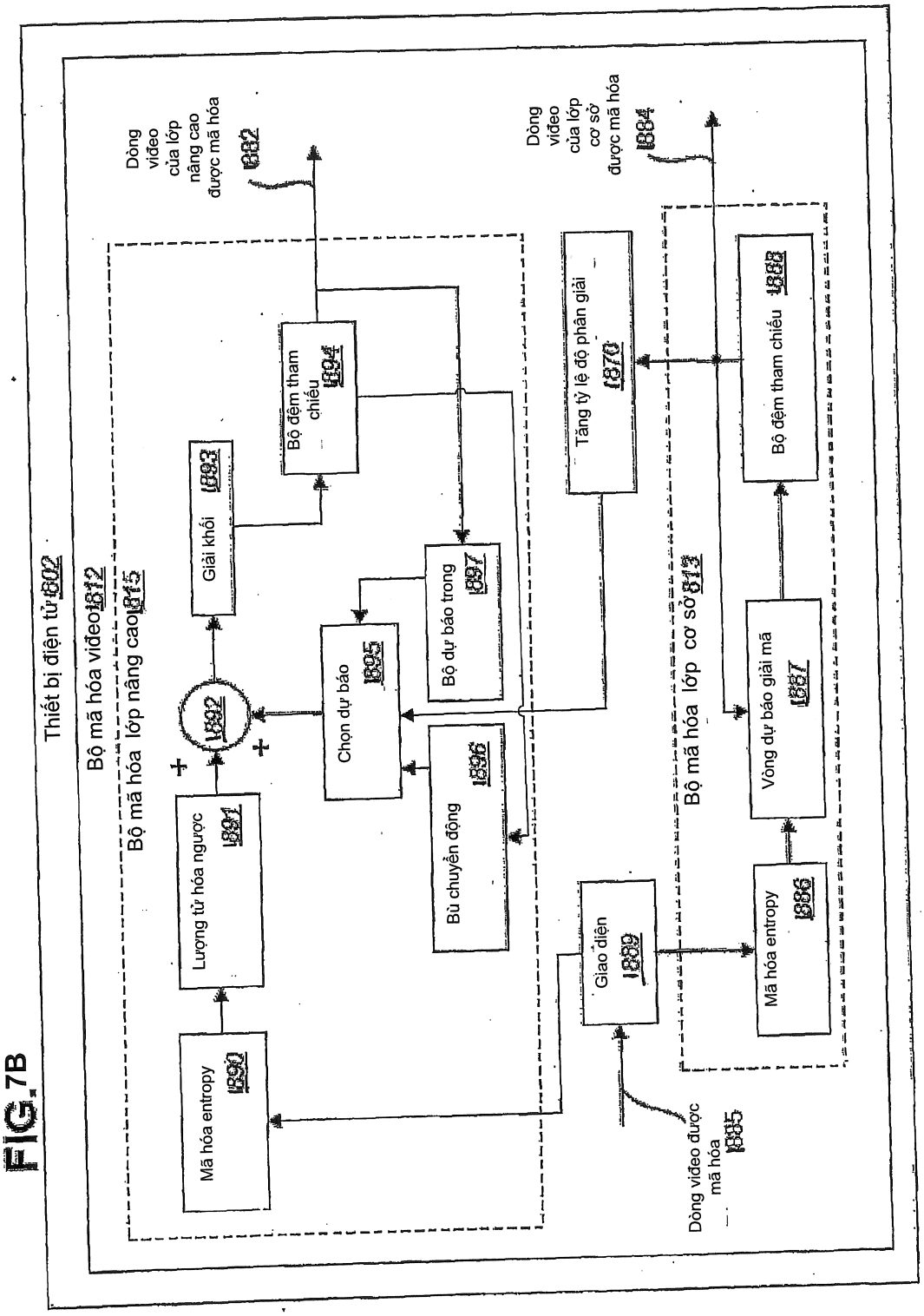

Thiết bị điện tử 702
Bộ giải mã 712
Bộ đệm hình ảnh được mã hóa (CPB) 720
Mã hóa entropy 701
(Biến đổi/định tỷ lệ/ lượng tử hóa) ngược 707
Dự báo khung trong 788
Bù chuyển động 780
Bộ lọc giải khối 794
Bộ đệm hình ảnh được giải mã 709
705
703
784
786
790
782
792
796
798
718
Dòng bit 714
Truyền riêng biệt 710

FIG. 7A

FIG.7B


Thiết bị điện tử 802
Bộ mã hóa video 812
Bộ mã hóa lớp nâng cao 815
Dòng video của lớp nâng cao được mã hóa 882
Bộ đệm tham chiếu 894
Giải khối 893
Lượng tử hóa ngược 891
Mã hóa entropy 890
Chọn dự báo 895
Bộ dự báo trong 897
Bù chuyển động 896
Dòng video của lớp cơ sở được mã hóa 884
Tăng tỷ lệ độ phân giải 870
Bộ đệm tham chiếu 888
Bộ mã hóa lớp cơ sở 813
Vòng dự báo giải mã 887
Giao diện 889
Mã hóa entropy 886
Dòng video được mã hóa 885

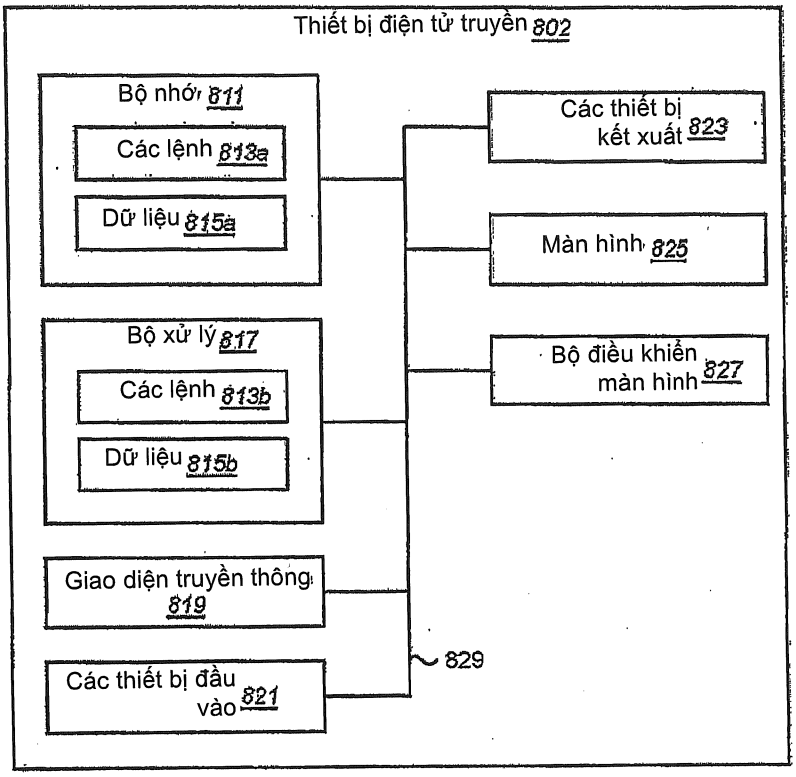

Thiết bị điện tử truyền 802
Bộ nhớ 811
Các lệnh 813a
Dữ liệu 815a
Bộ xử lý 817
Các lệnh 813b
Dữ liệu 815b
Giao diện truyền thông 819
Các thiết bị đầu vào 821
829
Các thiết bị kết xuất 823
Màn hình 825
Bộ điều khiển màn hình 827

FIG. 8

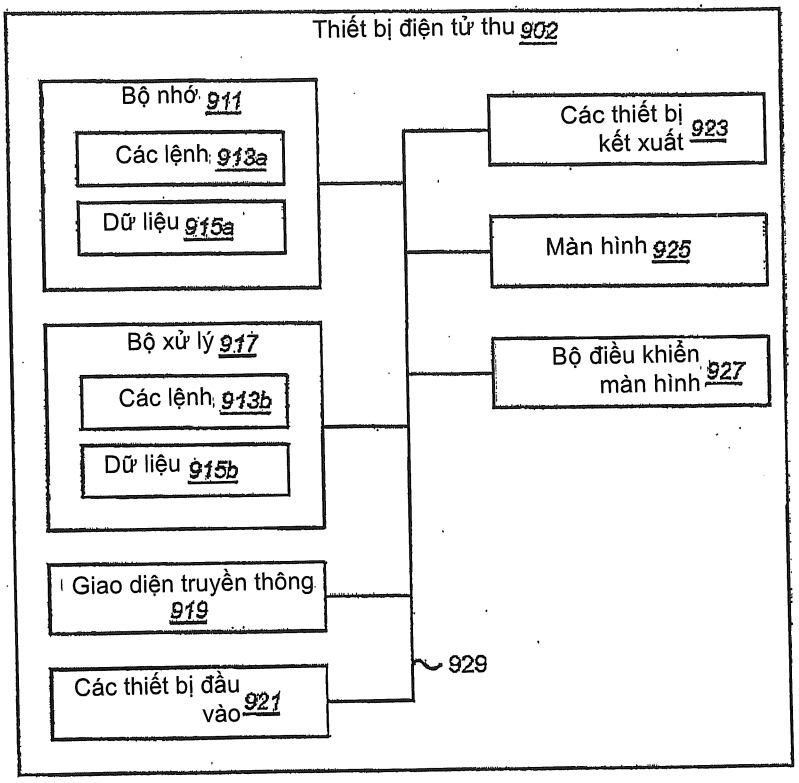

Thiết bị điện tử thu 902
Bộ nhớ 911
Các lệnh 913a
Dữ liệu 915a
Bộ xử lý 917
Các lệnh 913b
Dữ liệu 915b
Giao diện truyền thông 919
Các thiết bị đầu vào 921
929
Các thiết bị kết xuất 923
Màn hình 925
Bộ điều khiển màn hình 927

FIG. 9

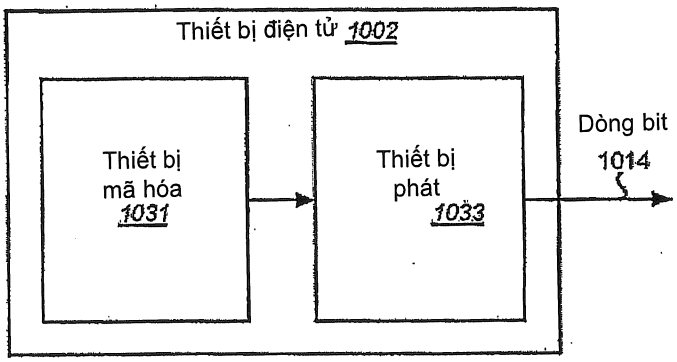

Thiết bị điện tử 1002
Thiết bị mã hóa 1031
Thiết bị phát 1033
Dòng bit 1014

FIG. 10

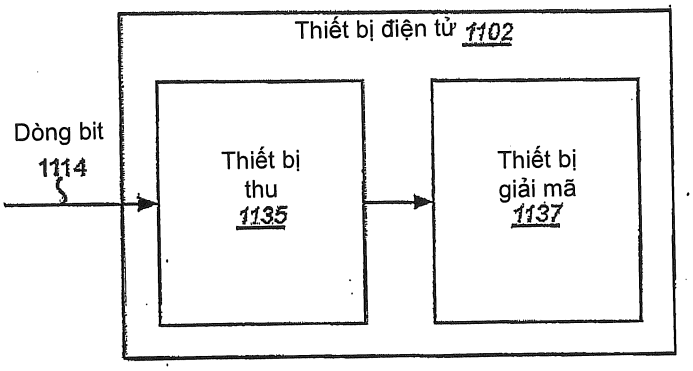


FIG. 11

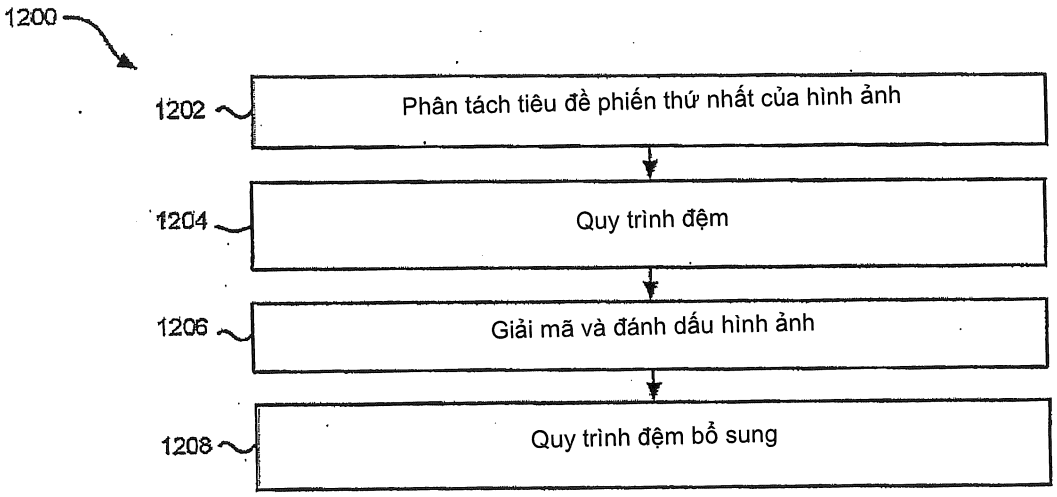


FIG. 12

nal_unit_header( ) {	Tên
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_reserved_zero_6bits	u(6)
nuh_temporal_id_plus1	u(3)
}	

Cú pháp tiêu đề của gói 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 tiêu đề của gói 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 tiêu đề của gói 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 gói NAL chung

FIG. 14

vps_extension( ) {	Tên
while(!byte_aligned( ))	
vps_extension_byte_alignment_reserved_zero_bit	u(1)
// scalability type and layer_id partitioning method	
scalability_type	u(4)
for(i = 0; i < MaxDim(scalability_type); i++)	
layer_id_dim_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 hợp thông số viđeo hiện tại

FIG. 15

scalability_type	MaxDim(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 trữ

Các loại khả năng điều chỉnh 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(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 hợp thông số viđeo

FIG.17

Scalability_map[i]	Kích thước khả năng điều chỉnh
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 trữ

Cú pháp ánh xạ khả năng điều chỉnh

FIG.18

vps_extension( ) {	Tên
while(!byte_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 hợp thông số viđeo

FIG.19

vps_extension( ) {	Tên
while(!byte_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 hợp thông số viđeo 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[i]; 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 phần mở rộng của tập hợp thông số viđeo

FIG. 22

scalability_mask_bit[k]	Kích thước khả năng điều chỉnh
0	Không gian
1	Chất lượng
2	Độ sâu
3	Đa hình ảnh
4-15	Dự trữ

Cú pháp ánh xạ khả năng điều chỉnh

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 hợp thông số viđeo

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 hợp thông số viđeo

FIG. 25

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 <NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v8)
}	
for(i = 1; i <= vps_max_layers_minus1 ; i++)	
profile_tier_level(1, vps_max_sub_layers_minus1)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// 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 tập hợp thông số viđeo

FIG.26

scalability_mask's k'th bit	Kích thước khả năng điều chỉnh
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(!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 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 hợp thông số viđeo

FIG.28

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)	
for(i = 1; i <= vps_max_layers_minus1; i++) {	
// layer dependency information signaling	
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 hợp thông số viđeo

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(1, 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 hợp thông số viđeo

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 phần mở rộng của tập hợp thông số viđeo

FIG.31

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 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 hợp thông số viđeo

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 hợp thông số viđeo

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( ) {	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(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++) {	
addlIsIdx = **addl_output_layer_sets_idx**[i]	ue(v)
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIsIdx][j])	
output_layer_flag[addlIsIdx][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++) {	
addlIsIdx = **addl_output_layer_sets_idx**[i]	ue(v)
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[addlIsIdx][j])	
output_layer_flag[addlIsIdx][j]	u(1)
}	
}	

FIG. 37

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(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(j)	
}	
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(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. 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( ) {	Bộ mô tả
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

	Bộ mô tả
vps_extension( ) {	
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_buff_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( ) {	Bộ mô tả
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++); 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_buff_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)	
}	
}	
}	
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++); 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_buff_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
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

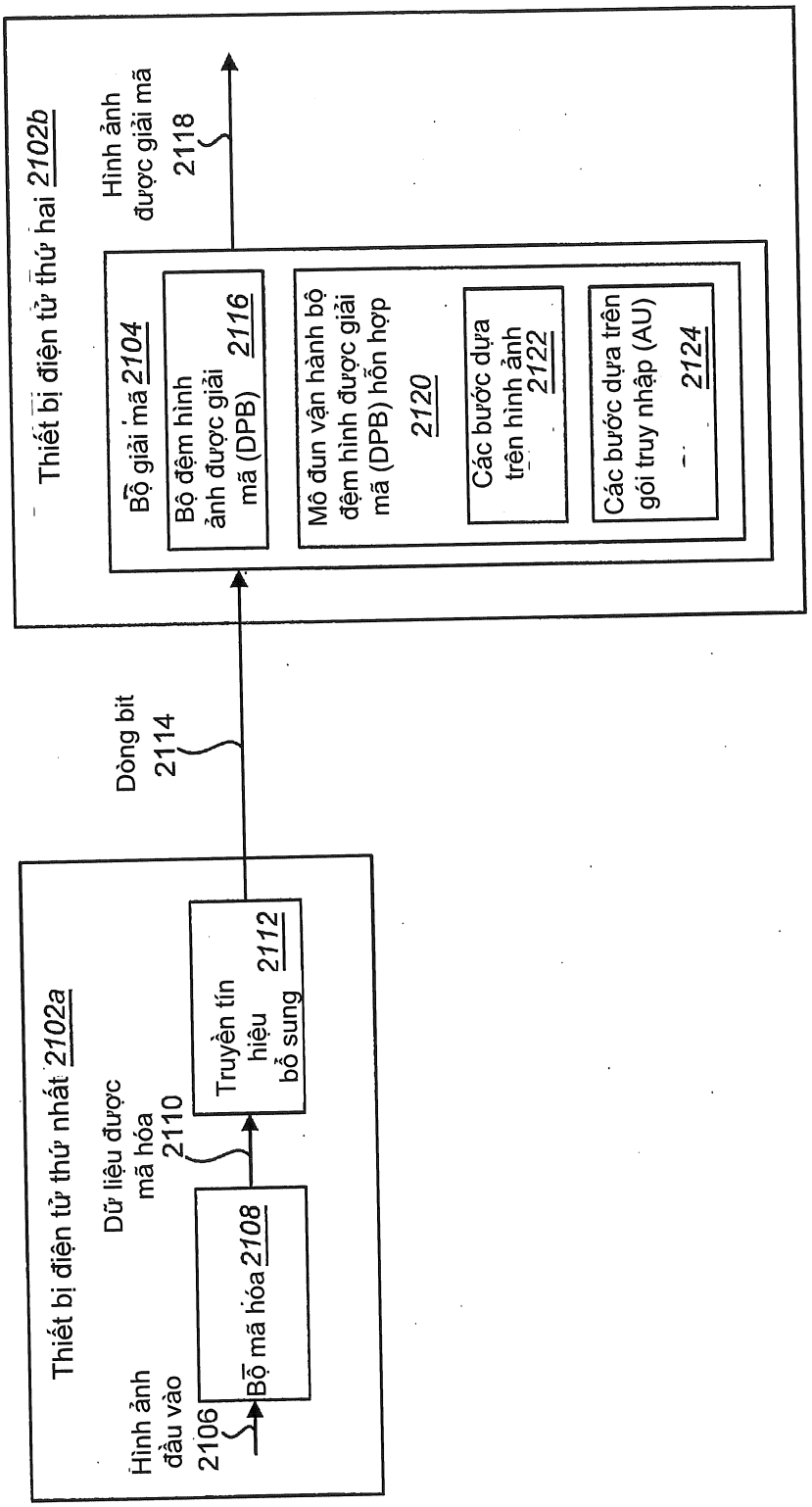

Thiết bị điện tử thứ nhất 2102a
Hình ảnh đầu vào 2106
Bộ mã hóa 2108
Dữ liệu được mã hóa 2110
Truyền tín hiệu bổ sung 2112
Dòng bit 2114
Thiết bị điện tử thứ hai 2102b
Bộ giải mã 2104
Bộ đệm hình ảnh được giải mã (DPB) 2116
Mô đun vận hành bộ đệm hình được giải mã (DPB) hỗn hợp 2120
Các bước dựa trên hình ảnh 2122
Các bước dựa trên gói truy nhập (AU) 2124
Hình ảnh được giải mã 2118

FIG. 63

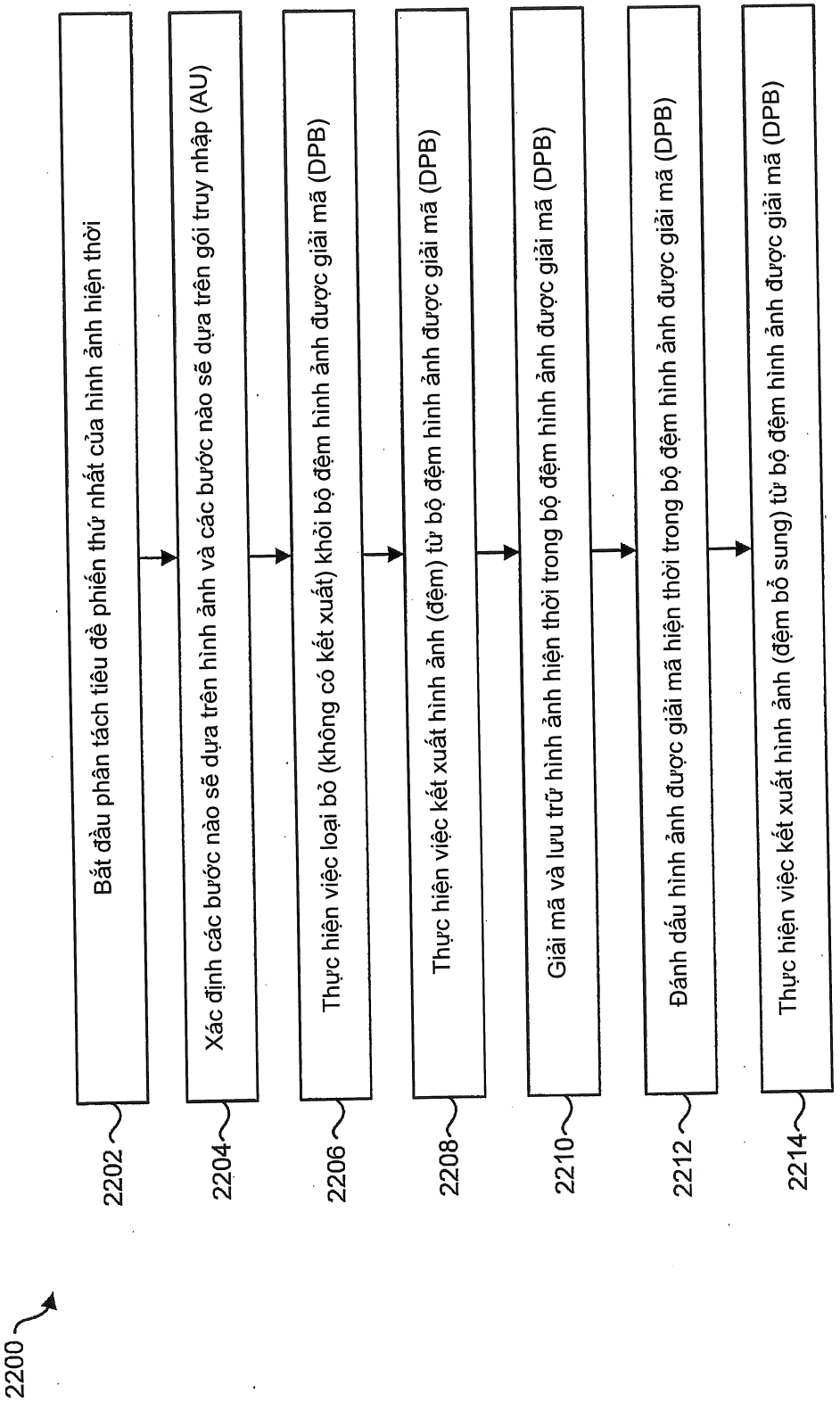

2200
2202
Bắt đầu phân tách tiêu đề phiến thứ nhất của hình ảnh hiện thời
2204
Xác định các bước nào sẽ dựa trên hình ảnh và các bước nào sẽ dựa trên gói truy nhập (AU)
2206
Thực hiện việc loại bỏ (không có kết xuất) khỏi bộ đệm hình ảnh được giải mã (DPB)
2208
Thực hiện việc kết xuất hình ảnh (đệm) từ bộ đệm hình ảnh được giải mã (DPB)
2210
Giải mã và lưu trữ hình ảnh hiện thời trong bộ đệm hình ảnh được giải mã (DPB)
2212
Đánh dấu hình ảnh được giải mã hiện thời trong bộ đệm hình ảnh được giải mã (DPB)
2214
Thực hiện việc kết xuất hình ảnh (đệm bổ sung) từ bộ đệm hình ảnh được giải mã (DPB)

FIG. 64

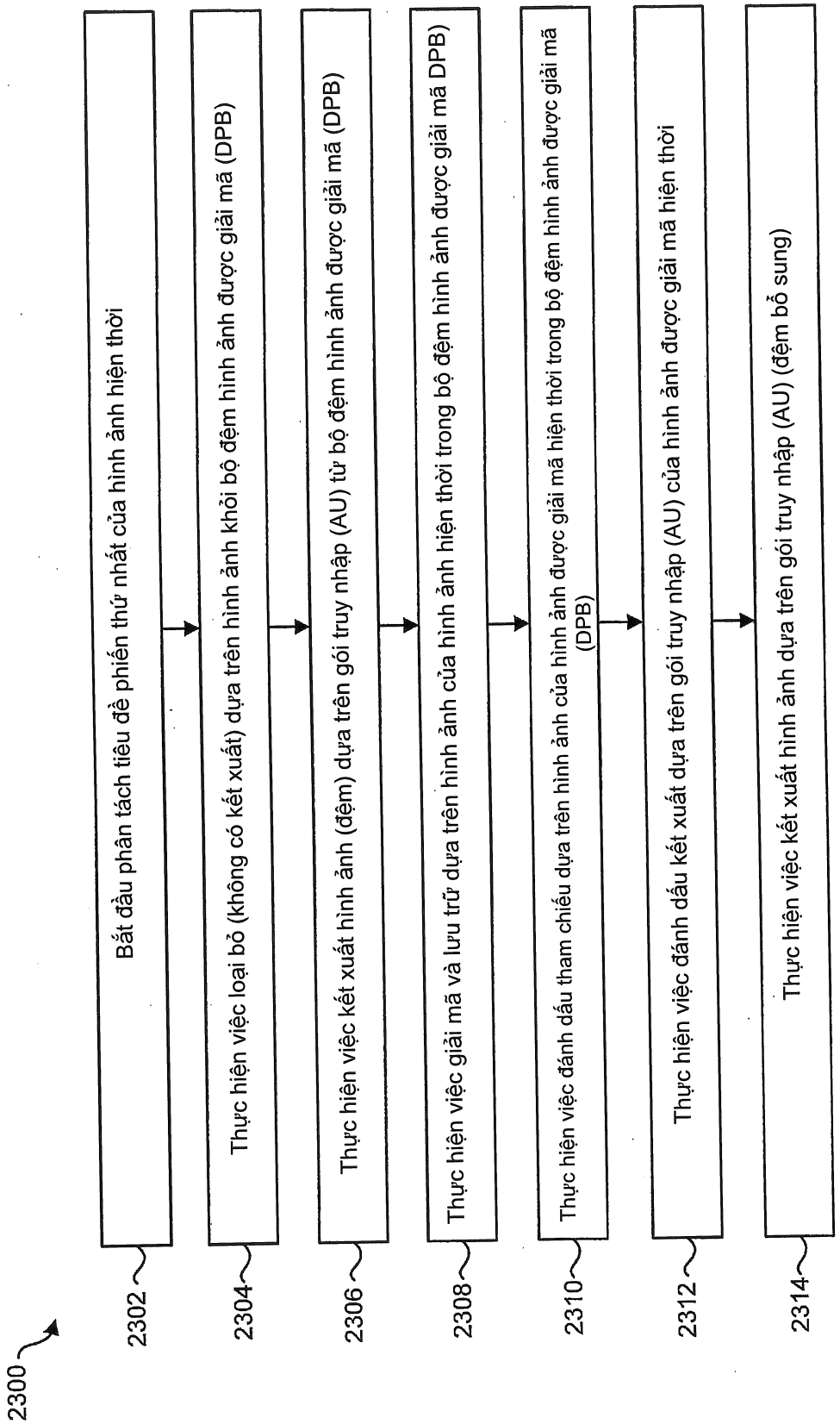

2300
2302
Bắt đầu phân tách tiêu đề phiên thứ nhất của hình ảnh hiện thời
2304
Thực hiện việc loại bỏ (không có kết xuất) dựa trên hình ảnh khỏi bộ đệm hình ảnh được giải mã (DPB)
2306
Thực hiện việc kết xuất hình ảnh (đệm) dựa trên gói truy nhập (AU) từ bộ đệm hình ảnh được giải mã (DPB)
2308
Thực hiện việc giải mã và lưu trữ dựa trên hình ảnh của hình ảnh hiện thời trong bộ đệm hình ảnh được giải mã DPB)
2310
Thực hiện việc đánh dấu tham chiếu dựa trên hình ảnh của hình ảnh được giải mã hiện thời trong bộ đệm hình ảnh được giải mã (DPB)
2312
Thực hiện việc đánh dấu kết xuất dựa trên gói truy nhập (AU) của hình ảnh được giải mã hiện thời
2314
Thực hiện việc kết xuất hình ảnh dựa trên gói truy nhập (AU) (đệm bổ sung)

FIG. 65

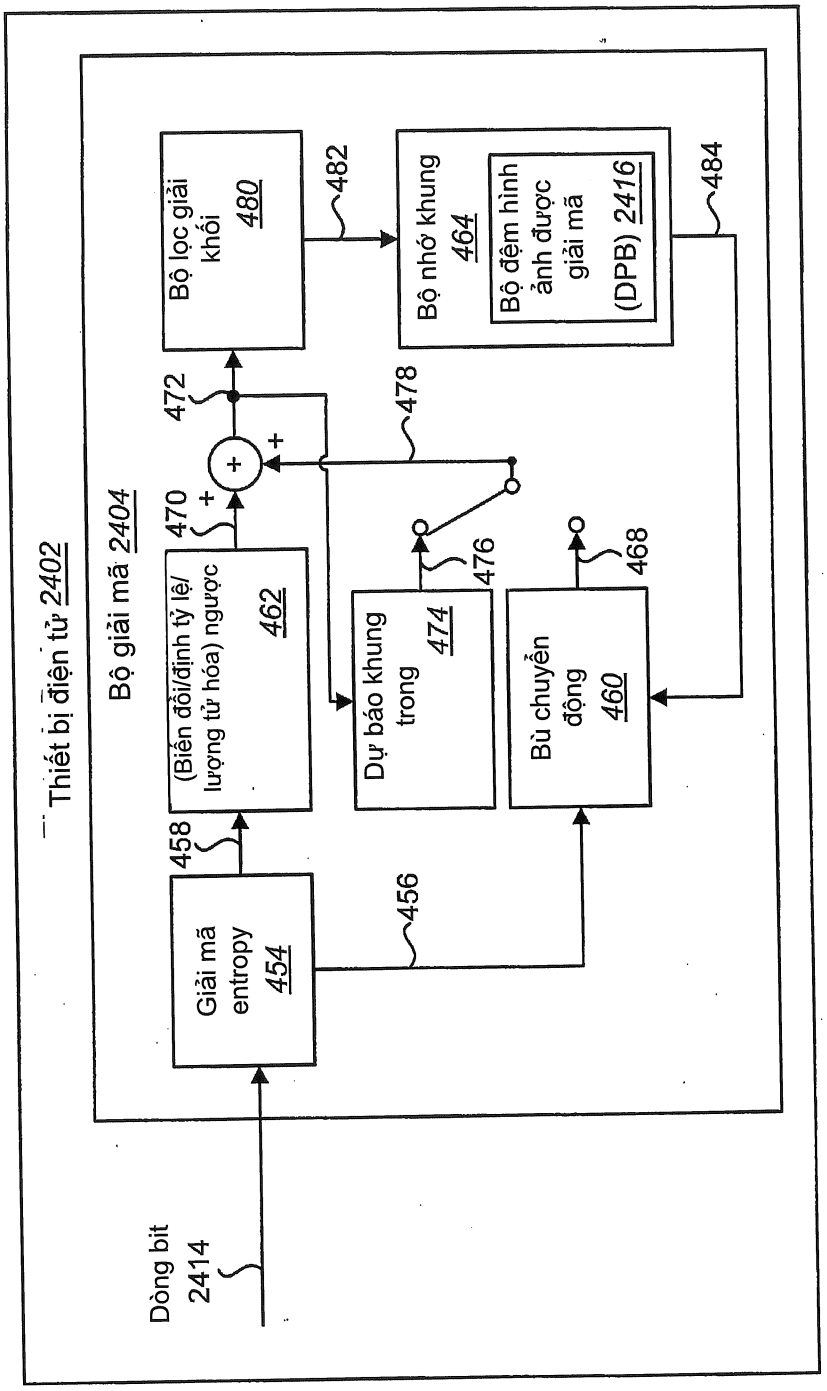

Thiết bị điện tử 2402
Bộ giải mã 2404
Dòng bit 2414
Giải mã entropy 454
458
456
(Biến đổi/định tỷ lệ/ lượng tử hóa) ngược 462
470
472
478
Dự báo khung trong 474
476
Bù chuyển động 460
468
Bộ lọc giải khối 480
482
Bộ nhớ khung 464
Bộ đệm hình ảnh được giải mã (DPB) 2416
484

FIG. 66

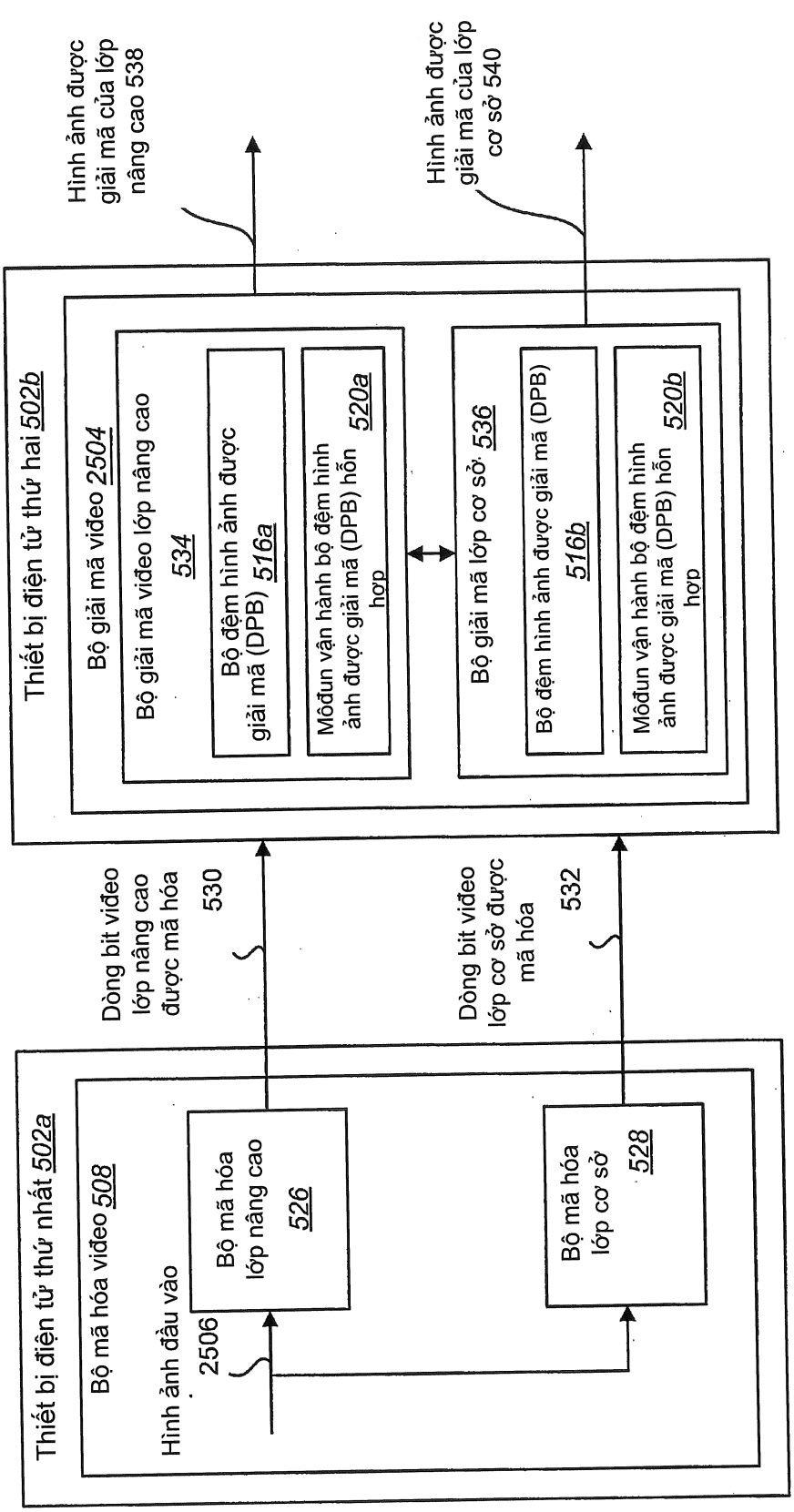

Thiết bị điện tử thứ nhất 502a
Bộ mã hóa video 508
Hình ảnh đầu vào
2506
Bộ mã hóa lớp nâng cao 526
Bộ mã hóa lớp cơ sở 528
Dòng bit video lớp nâng cao được mã hóa 530
Dòng bit video lớp cơ sở được mã hóa 532
Thiết bị điện tử thứ hai 502b
Bộ giải mã video 2504
Bộ giải mã video lớp nâng cao 534
Bộ đệm hình ảnh được giải mã (DPB) 516a
Môđun vận hành bộ đệm hình ảnh được giải mã (DPB) hỗn hợp 520a
Bộ giải mã lớp cơ sở 536
Bộ đệm hình ảnh được giải mã (DPB) 516b
Môđun vận hành bộ đệm hình ảnh được giải mã (DPB) hỗn hợp 520b
Hình ảnh được giải mã của lớp nâng cao 538
Hình ảnh được giải mã của lớp cơ sở 540

FIG. 67A

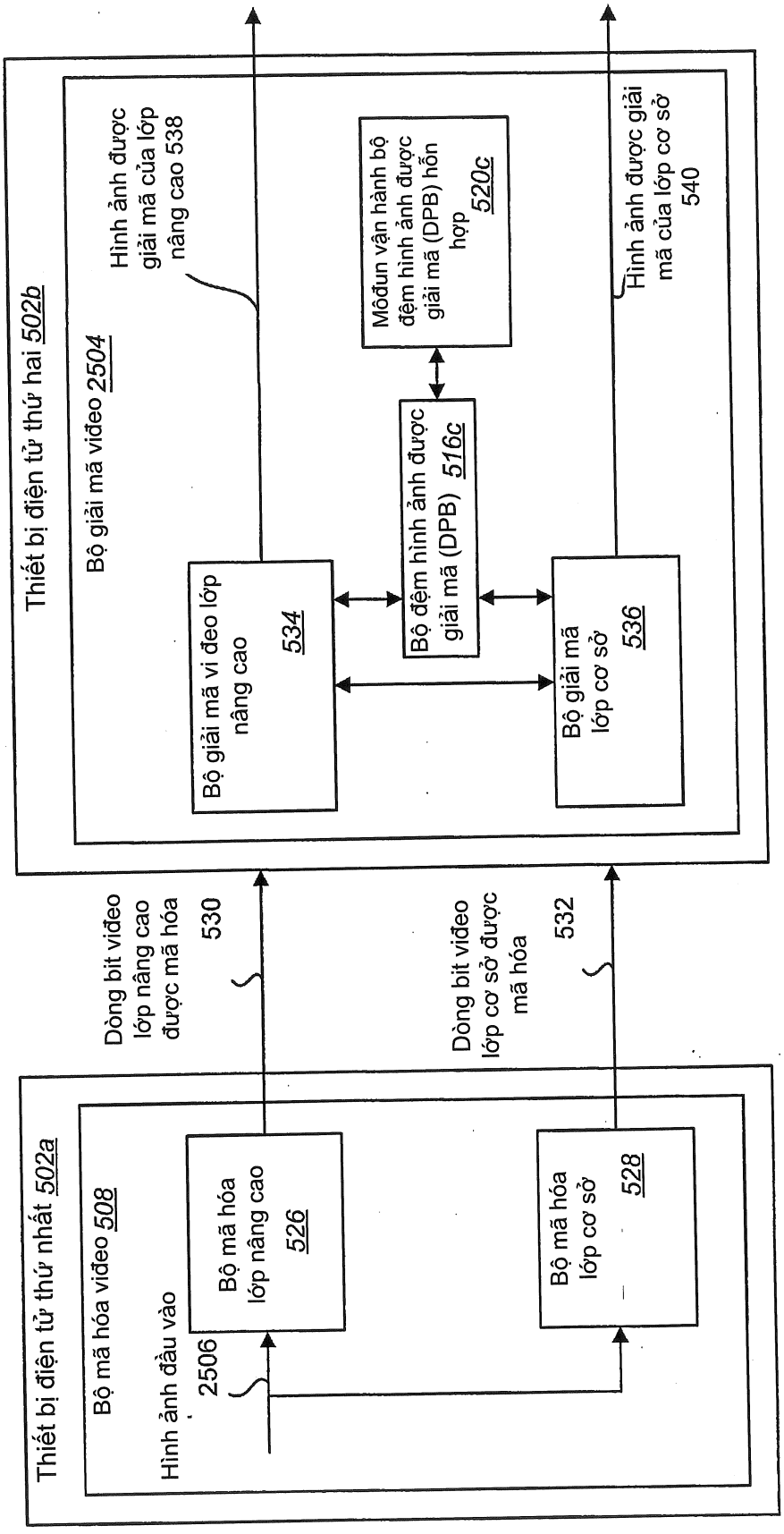

Thiết bị điện tử thứ nhất 502a
Bộ mã hóa video 508
Hình ảnh đầu vào
2506
Bộ mã hóa lớp nâng cao 526
Bộ mã hóa lớp cơ sở 528
Dòng bit video lớp nâng cao được mã hóa 530
Dòng bit video lớp cơ sở được mã hóa 532
Thiết bị điện tử thứ hai 502b
Bộ giải mã video 2504
Hình ảnh được giải mã của lớp nâng cao 538
Bộ giải mã vi đeo lớp nâng cao 534
Bộ đệm hình ảnh được giải mã (DPB) 516c
Môđun vận hành bộ đệm hình ảnh được giải mã (DPB) hỗn hợp 520c
Bộ giải mã lớp cơ sở 536
Hình ảnh được giải mã của lớp cơ sở 540

FIG. 67B

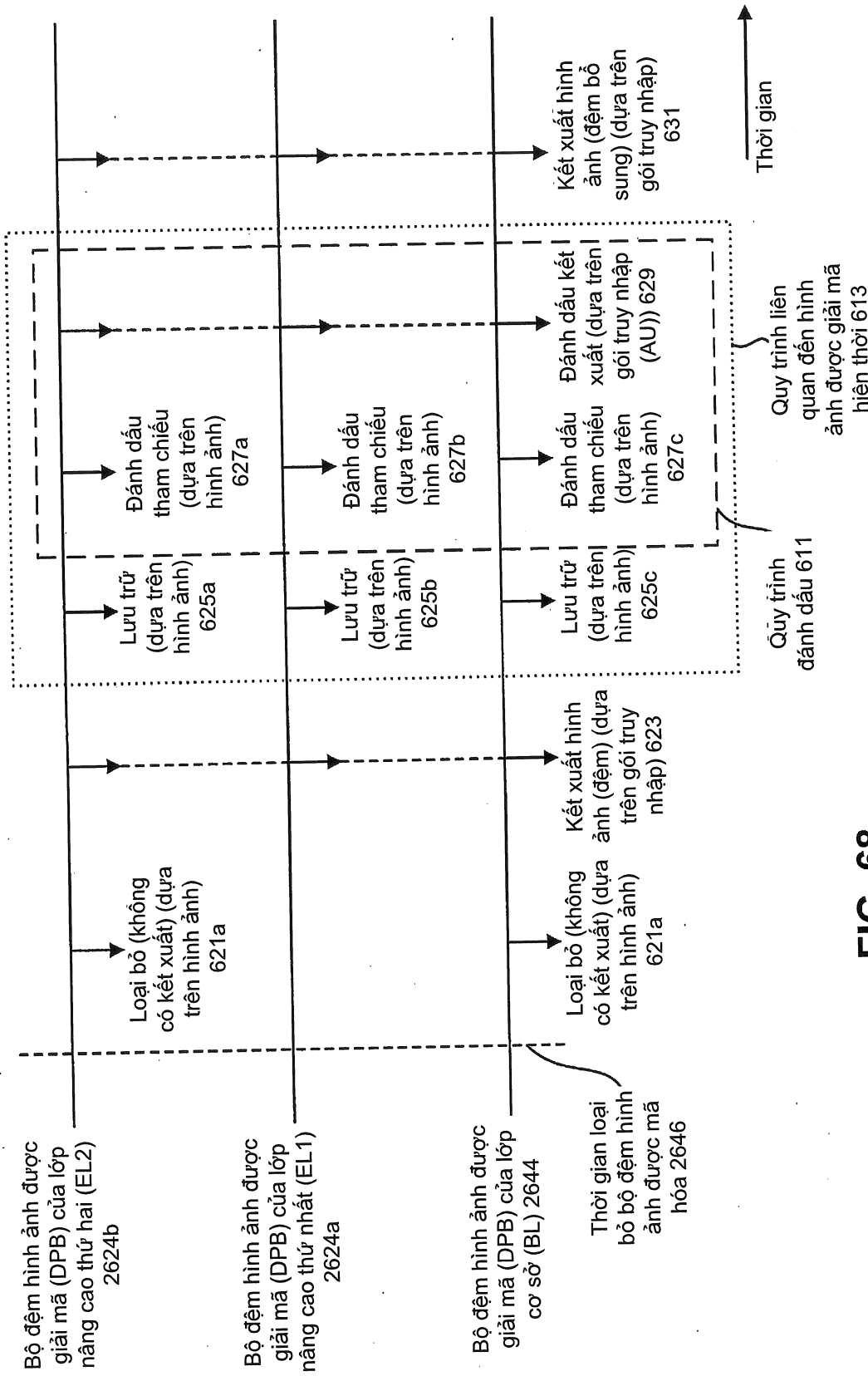

Bộ đệm hình ảnh được giải mã (DPB) của lớp nâng cao thứ hai (EL2) 2624b
Bộ đệm hình ảnh được giải mã (DPB) của lớp nâng cao thứ nhất (EL1) 2624a
Bộ đệm hình ảnh được giải mã (DPB) của lớp cơ sở (BL) 2644
Thời gian loại bỏ bộ đệm hình ảnh được mã hóa 2646
Loại bỏ (không có kết xuất) (dựa trên hình ảnh) 621a
Lưu trữ (dựa trên hình ảnh) 625a
Đánh dấu tham chiếu (dựa trên hình ảnh) 627a
Loại bỏ (không có kết xuất) (dựa trên hình ảnh) 621a
Lưu trữ (dựa trên hình ảnh) 625b
Đánh dấu tham chiếu (dựa trên hình ảnh) 627b
Kết xuất hình ảnh (đệm) (dựa trên gói truy nhập) 623
Lưu trữ (dựa trên hình ảnh) 625c
Đánh dấu tham chiếu (dựa trên hình ảnh) 627c
Đánh dấu kết xuất (dựa trên gói truy nhập (AU)) 629
Kết xuất hình ảnh (đệm bổ sung) (dựa trên gói truy nhập) 631
Quy trình đánh dấu 611
Quy trình liên quan đến hình ảnh được giải mã hiện thời 613
Thời gian

FIG. 68

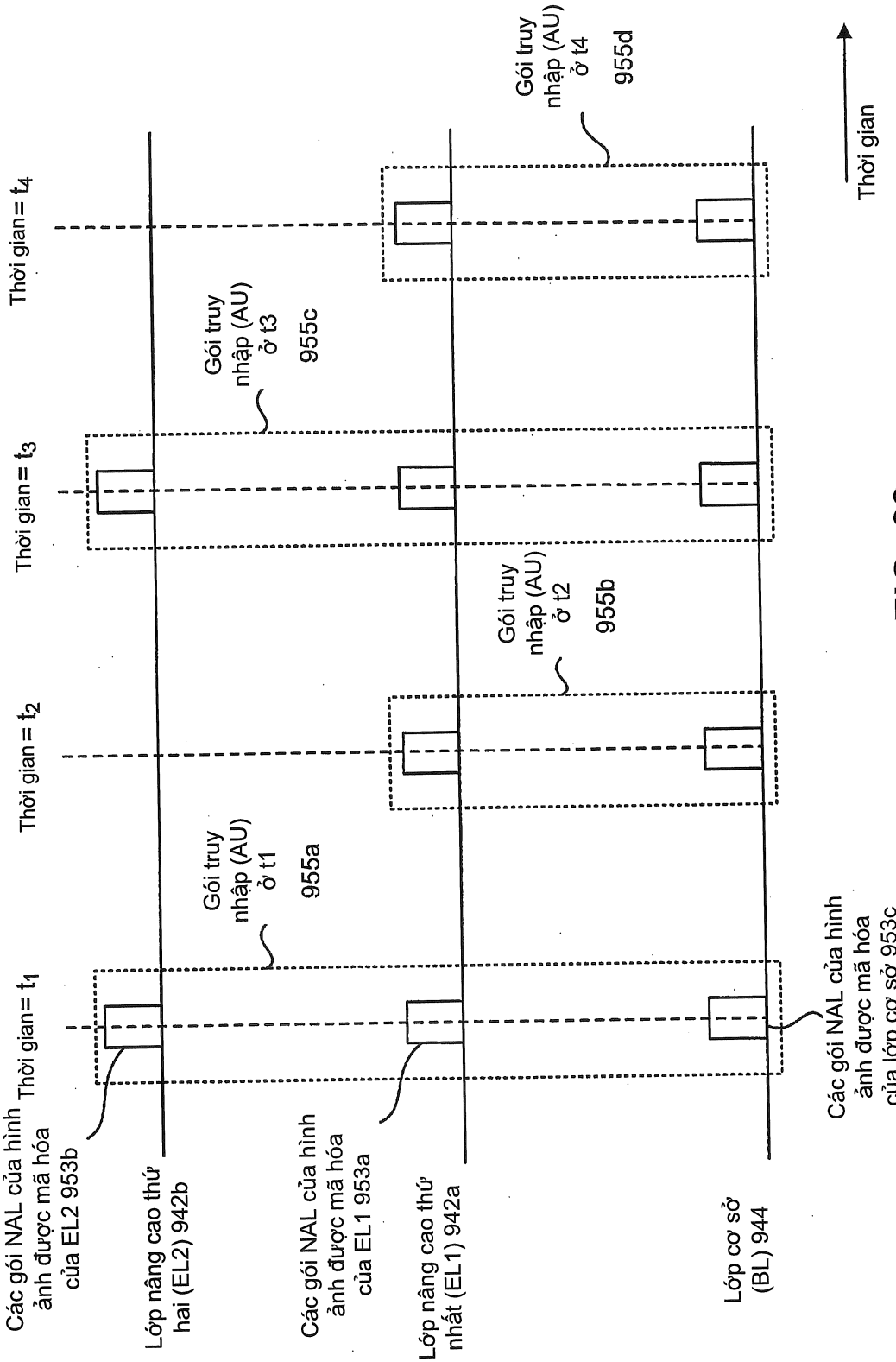

Thời gian= t1
Thời gian = t2
Thời gian= t3
Thời gian= t4
Thời gian
Gói truy nhập (AU) ở t1
955a
Gói truy nhập (AU) ở t2
955b
Gói truy nhập (AU) ở t3
955c
Gói truy nhập (AU) ở t4
955d
Các gói NAL của hình ảnh được mã hóa của EL2 953b
Các gói NAL của hình ảnh được mã hóa của EL1 953a
Các gói NAL của hình ảnh được mã hóa của lớp cơ sở 953c
Lớp nâng cao thứ hai (EL2) 942b
Lớp nâng cao thứ nhất (EL1) 942a
Lớp cơ sở (BL) 944

FIG. 69

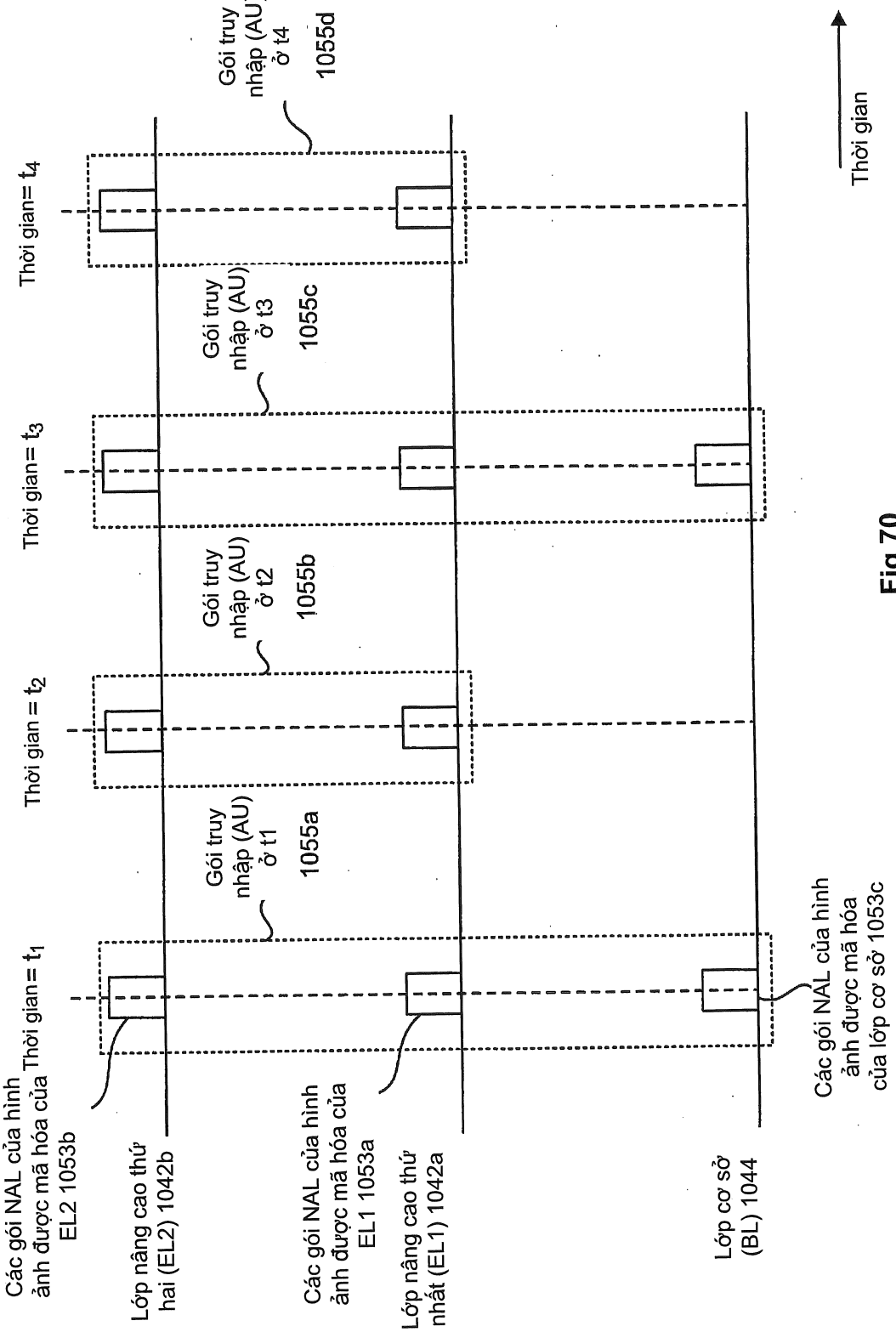

Thời gian= t_4
Thời gian= t_3
Thời gian = t_2
Thời gian= t_1
Gói truy nhập (AU) ở t4
1055d
Gói truy nhập (AU) ở t3
1055c
Gói truy nhập (AU) ở t2
1055b
Gói truy nhập (AU) ở t1
1055a
Thời gian
Các gói NAL của hình ảnh được mã hóa của EL2 1053b
Lớp nâng cao thứ hai (EL2) 1042b
Các gói NAL của hình ảnh được mã hóa của EL1 1053a
Lớp nâng cao thứ nhất (EL1) 1042a
Lớp cơ sở (BL) 1044
Các gói NAL của hình ảnh được mã hóa của lớp cơ sở 1053c

Fig.70