



(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)



1-0051775

(51)^{2021.01} H04N 19/60; H04N 19/137; H04N
19/70; H04N 19/46; H04N 19/132;
H04N 19/176

(13) B

(21) 1-2022-06124

(22) 25/02/2021

(86) PCT/KR2021/002409 25/02/2021

(87) WO 2021/172912 02/09/2021

(30) 62/982,742 27/02/2020 US

(45) 25/09/2025 450

(43) 25/11/2022 416A

(73) LG Electronics Inc. (KR)

128, Yeoui-daero, Yeongdeungpo-gu Seoul 07336, Republic of Korea

(72) YOO, Sunmi (KR); CHOI, Jungah (KR); HEO, Jin (KR); CHOI, Jangwon (KR).

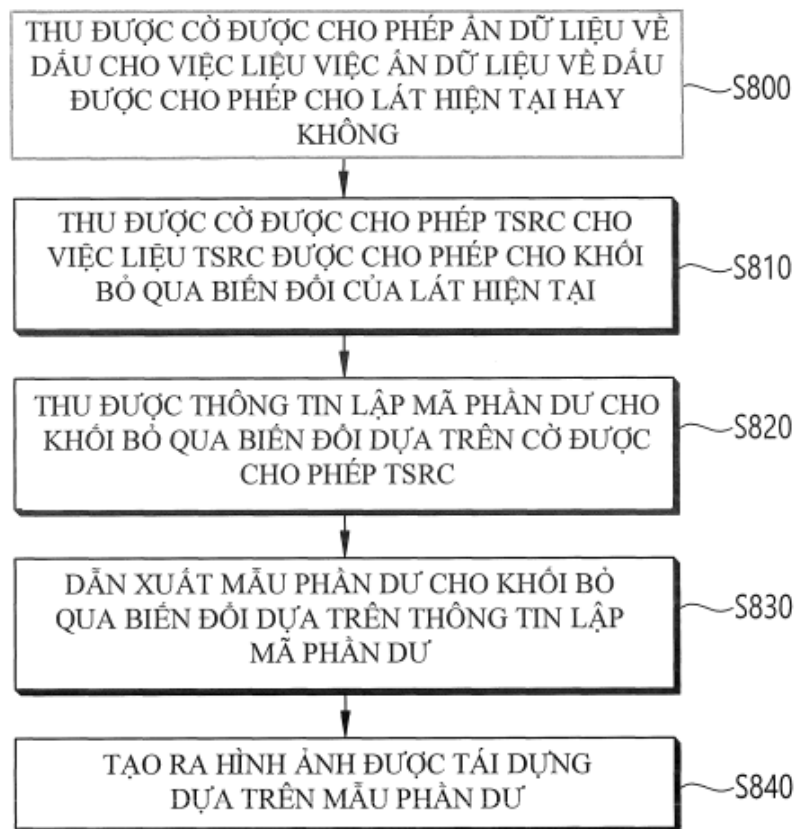
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ VÀ MÃ HÓA ẢNH, PHƯƠNG PHÁP TRUYỀN DỮ
LIỆU CHO THÔNG TIN ẢNH VÀ PHƯƠNG TIỆN LƯU TRỮ PHI CHUYÊN
TIẾP ĐƯỢC ĐƯỢC BỞI MÁY TÍNH

(21) 1-2022-06124

(57) Sáng chế đề cập đến phương pháp giải mã ảnh, phương pháp mã hóa ảnh, phương pháp truyền dữ liệu cho thông tin ảnh và phương tiện lưu trữ đọc được bởi máy tính phi chuyển tiếp. Phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã theo sáng chế này bao gồm các bước là: thu được cờ khả dụng việc ẩn dữ liệu về dấu chỉ báo liệu việc ẩn dữ liệu về dấu là khả dụng cho lát hiện tại hay không; thu được cờ khả dụng lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) chỉ báo liệu TSRC là khả dụng cho khối bỏ qua biến đổi của lát hiện tại hay không; thu được thông tin lập mã phần dư về khối bỏ qua biến đổi dựa trên cờ khả dụng TSRC; dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư; và tạo ra hình ảnh tái dựng dựa trên mẫu phần dư, trong đó cờ khả dụng TSRC thu được dựa trên cờ khả dụng ẩn dữ liệu về dấu.

FIG. 8



Lĩnh vực kỹ thuật được đề cập

Sáng chế này đề cập đến công nghệ lập mã ảnh, và cụ thể hơn là, đề cập đến phương pháp và thiết bị giải mã video trong đó thông tin cờ về liệu TSRC được cho phép hay không được cho phép được lập mã dựa trên thông tin cờ về liệu SDH được cho phép khi lập mã dữ liệu phần dư của khối trong hệ thống lập mã ảnh.

Tình trạng kỹ thuật của sáng chế

Gần đây, nhu cầu về các ảnh độ phân giải cao, chất lượng cao, chẳng hạn các ảnh độ phân giải cao (High Definition, HD) và các ảnh độ phân giải siêu cao (Ultra High Definition, UHD), đã và đang ngày càng tăng trong các lĩnh vực khác nhau. Vì dữ liệu ảnh này có độ phân giải cao và chất lượng cao, nên lượng thông tin hoặc lượng bit cần được truyền tải so với dữ liệu ảnh kế thừa. Do đó, khi dữ liệu ảnh được truyền sử dụng các phương tiện như đường băng rộng dùng dây/không dây thông thường, hoặc dữ liệu ảnh được lưu trữ sử dụng phương tiện lưu trữ hiện có, thì chi phí truyền và chi phí lưu trữ bị tăng lên.

Theo đó, cần có kỹ thuật nén ảnh hiệu quả cao để truyền, lưu trữ, và tái tạo một cách hiệu quả thông tin của các ảnh có độ phân giải cao và chất lượng cao.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất phương pháp và thiết bị để nâng cao hiệu quả lập mã ảnh.

Sáng chế cũng đề xuất phương pháp và thiết bị để cải thiện hiệu quả lập mã phần dư.

Theo một phương án của sáng chế này, phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã được cung cấp. Phương pháp bao gồm bước thu được cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không, bước thu được cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại hay không, bước thu được thông tin lập mã phần dư cho khối bỏ

qua biến đổi dựa trên cờ được phép TSRC, bước dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư, và bước tạo ra hình ảnh được tái dựng dựa trên mẫu phần dư, trong đó cờ được phép TSRC thu được dựa trên cờ được phép ẩn dữ liệu về dấu.

Theo phương án khác của tài liệu này, thiết bị giải mã để thực hiện việc giải mã ảnh được cung cấp. Thiết bị giải mã bao gồm bộ giải mã entropy được định cấu hình để thu được cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không, để thu được cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại hay không, và để thu được thông tin lập mã phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC, bộ xử lý phần dư được định cấu hình để dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư, và bộ cộng được định cấu hình để tạo ra hình ảnh được tái dựng dựa trên mẫu phần dư, trong đó cờ được phép TSRC thu được dựa trên cờ được phép ẩn dữ liệu về dấu.

Theo phương án khác nữa của tài liệu này, phương pháp mã hóa video mà được thực hiện bởi thiết bị mã hóa được cung cấp. Phương pháp bao gồm bước mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại, bước mã hóa cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại dựa trên cờ được phép ẩn dữ liệu về dấu, bước mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC, và bước tạo ra luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ được phép TSRC và thông tin phần dư.

Theo một phương án khác nữa, sáng chế đề xuất thiết bị mã hóa video. Thiết bị mã hóa bao gồm bộ mã hóa entropy được định cấu hình để mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không, để mã hóa cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại dựa trên cờ được phép ẩn dữ liệu về dấu hay không, để mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC, và để tạo ra luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ được phép TSRC và thông tin phần dư.

Theo phương án khác nữa, có phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính mà lưu trữ luồng bit bao gồm thông tin ảnh mà làm cho thiết bị giải mã thực hiện phương pháp giải mã ảnh. Trong phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính, phương pháp giải mã ảnh bao gồm bước thu được cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không, bước thu được cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại hay không, bước thu được thông tin lập mã phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC, bước dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư, và bước tạo ra hình ảnh được tái dựng dựa trên mẫu phần dư, trong đó cờ được phép TSRC thu được dựa trên cờ được phép ẩn dữ liệu về dấu.

Theo tài liệu này, có thể nâng cao hiệu quả của việc lập mã phần dư.

Theo tài liệu này, cờ được phép TSRC có thể được báo hiệu tùy thuộc vào cờ được phép ẩn dữ liệu về dấu, và thông qua điều này, hiệu quả lập mã có thể được nâng cao bằng việc ngăn việc ẩn dữ liệu về dấu không được sử dụng cho khối bỏ qua biến đổi mà TSRC không được cho phép cho, và hiệu quả lập mã phần dư tổng thể có thể được nâng cao bằng cách giảm lượng bit cần được lập mã.

Theo tài liệu này, cờ được phép TSRC có thể được báo hiệu tùy thuộc vào việc cờ được phép bỏ qua biến đổi và cờ được phép ẩn dữ liệu về dấu, và thông qua điều này, hiệu quả lập mã có thể được nâng cao bằng việc ngăn việc ẩn dữ liệu về dấu không được sử dụng cho khối bỏ qua biến đổi mà TSRC không được cho phép cho, và hiệu quả lập mã phần dư tổng thể có thể được nâng cao bằng cách giảm lượng bit để được lập mã.

Mô tả vắn tắt các hình vẽ

Fig.1 là hình vẽ minh họa vắn tắt ví dụ về thiết bị lập mã video/ảnh mà các phương án của sáng chế có thể áp dụng với.

Fig.2 là giản đồ sơ lược minh họa cấu hình của thiết bị mã hóa video/ảnh mà phương án (các phương án) của sáng chế này có thể được áp dụng với.

Fig.3 là sơ đồ giản lược minh họa cấu hình của thiết bị giải mã video/ảnh mà phương án (các phương án) của sáng chế có thể được áp dụng với.

Fig.4 là hình vẽ thể hiện theo cách ví dụ việc lập mã số học nhị phân thích ứng ngữ cảnh (CABAC) để mã hóa phân tử cú pháp.

Fig.5 là sơ đồ thể hiện các hệ số biến đổi làm ví dụ bên trong khối 4x4.

Fig.6 minh họa một cách vắn tắt phương pháp mã hóa ảnh được thực hiện bởi thiết bị mã hóa theo sáng chế.

Fig.7 minh họa một cách vắn tắt thiết bị mã hóa để thực hiện phương pháp mã hóa ảnh theo sáng chế.

Fig.8 minh họa một cách vắn tắt phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã theo sáng chế.

Fig.9 minh họa một cách vắn tắt thiết bị giải mã để thực hiện phương pháp giải mã ảnh theo sáng chế.

Fig.10 minh họa sơ đồ cấu trúc của hệ thống tạo luồng nội dung mà sáng chế được áp dụng vào đó.

Mô tả chi tiết sáng chế

Sáng chế có thể được điều chỉnh theo nhiều dạng khác nhau, và các phương án thực hiện cụ thể của sáng chế sẽ được mô tả và được minh họa trên các hình vẽ. Tuy nhiên, các phương án này không nhằm để giới hạn sáng chế. Các thuật ngữ được sử dụng trong phần mô tả sau đây là chỉ được dùng để mô tả các phương án cụ thể, chứ không nhằm giới hạn sáng chế. Biểu diễn của số ít cũng bao gồm biểu diễn của số nhiều, miễn là nó được đọc khác nhau một cách rõ ràng. Các từ ngữ như “bao gồm” và “có” là nhằm thể hiện các dấu hiệu, các số lượng, các bước, các hoạt động, các phần tử, các thành phần, hoặc những sự kết hợp của chúng, mà được sử dụng trong phần mô tả sau đây, là tồn tại, và do đó, cần được hiểu rằng khả năng tồn tại hoặc bổ sung của một hoặc nhiều dấu hiệu, số lượng, bước, hoạt động, phần tử, thành phần khác, hoặc những sự kết hợp của chúng, là không bị loại trừ.

Trong lúc đó, các phần tử trên các hình vẽ mà được mô tả theo sáng chế là được vẽ độc lập để thuận tiện cho việc mô tả các chức năng cụ thể khác nhau, và không có nghĩa rằng các phần tử này là được biểu hiện bởi phần cứng độc lập hoặc phần mềm độc lập. Ví dụ, hai hoặc nhiều phần tử trong số các phần tử là có thể được tổ hợp để tạo thành

một phần tử, hoặc một phần tử có thể được phân vùng thành nhiều phần tử. Các phương án mà trong đó các phần tử được kết hợp và/hoặc được phân vùng là thuộc sáng chế chứ không nằm ngoài phạm vi của sáng chế.

Dưới đây, các phương án về sáng chế này sẽ được mô tả chi tiết với việc tham chiếu đến các hình vẽ kèm theo. Bên cạnh đó, các số chỉ dẫn giống nhau được dùng để chỉ các thành phần giống nhau trong suốt các hình vẽ, và các phần mô tả giống nhau của các thành phần giống nhau này sẽ được lược bỏ.

Fig.1 là hình vẽ minh họa vắn tắt ví dụ về thiết bị lập mã video/ảnh mà các phương án của sáng chế có thể áp dụng với.

Tham chiếu đến Fig.1, hệ thống lập mã video/ảnh có thể bao gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị thu). Thiết bị nguồn có thể phân phối thông tin hoặc dữ liệu video/ảnh được mã hóa ở dạng của tệp hoặc tạo luồng đến thiết bị thu thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng.

Thiết bị nguồn có thể bao gồm nguồn video, thiết bị mã hóa, và bộ truyền. Thiết bị thu có thể bao gồm bộ thu, thiết bị giải mã, và bộ kết xuất. Thiết bị mã hóa có thể được gọi là thiết bị mã hóa video/ảnh, và thiết bị giải mã có thể được gọi là thiết bị giải mã video/ảnh. Bộ truyền có thể được bao gồm trong thiết bị mã hóa. Bộ nhận có thể được gồm trong thiết bị giải mã. Bộ kết xuất có thể gồm bộ phận hiển thị, bộ phận hiển thị có thể được tạo cấu hình như là thiết bị tách rời hoặc thành phần ngoài.

Nguồn video có thể giành được video/ảnh thông qua quy trình chụp, tổng hợp, hoặc tạo ra video/ảnh. Nguồn video có thể gồm thiết bị chụp video/hình ảnh và/hoặc thiết bị tạo ra video/ảnh. Thiết bị chụp video/ảnh có thể bao gồm, ví dụ, một hoặc nhiều camera, các phần lưu trữ video/ảnh chứa các video/ảnh đã được chụp trước đó, và dạng tương tự. Thiết bị tạo ra video/ảnh có thể bao gồm, ví dụ, máy tính, máy tính bảng, và điện thoại thông minh, và có thể tạo ra các video/ảnh (theo cách điện tử). Ví dụ, video/hình ảnh ảo có thể được tạo ra qua máy tính hoặc dạng tương tự. Trong trường hợp này, quy trình chụp video/ảnh có thể được thay thế bằng quy trình tạo ra dữ liệu liên quan.

Thiết bị mã hóa có thể mã hóa ảnh/hình ảnh đầu vào. Thiết bị mã hóa có thể thực hiện chuỗi các thủ tục như dự đoán, biến đổi, và lượng tử hóa cho việc nén và mã hóa

hiệu quả. Dữ liệu được mã hóa (thông tin video/ảnh được mã hóa) có thể được xuất ra dưới dạng luồng bit.

Bộ truyền có thể truyền thông tin hoặc dữ liệu ảnh/video được mã hóa được xuất ra ở dạng luồng bit đến bộ thu của thiết bị thu qua phương tiện lưu trữ số hoặc mạng ở dạng của tệp hoặc tạo luồng. Phương tiện lưu trữ kỹ thuật số có thể gồm các phương tiện lưu trữ khác nhau như là USB, SD, CD, DVD, Blu-ray, HDD, SSD, và tương tự. Bộ truyền có thể gồm phần tử để tạo ra tệp tin phương tiện thông qua định dạng tệp tin định trước và có thể gồm phần tử để truyền thông qua mạng phát rộng/truyền thông. Bộ nhận có thể nhận/trích xuất luồng bit và truyền luồng bit nhận được đến thiết bị giải mã.

Thiết bị giải mã có thể giải mã video/hình ảnh bằng cách thực hiện một loạt các thủ tục như là giải lượng tử hóa, biến đổi ngược, và dự đoán tương ứng với hoạt động vận hành của thiết bị mã hóa.

Bộ kết xuất có thể kết xuất video/ảnh được giải mã. Video/ảnh đã kết xuất có thể được hiển thị thông qua bộ hiển thị.

Sáng chế liên quan đến việc lập mã video/hình ảnh. Ví dụ, các phương pháp/các phương án được bộc lộ trong sáng chế này có thể được áp dụng cho phương pháp được bộc lộ trong chuẩn lập mã video đa năng (Versatile video coding, VVC), chuẩn lập mã video cốt yếu (Essential video coding, EVC), chuẩn audio video 1 (AOMedia Video 1, AV1), chuẩn lập mã video thế hệ thứ hai (2nd generation of audio video coding standard, AVS2), hoặc chuẩn lập mã video/ảnh thế hệ tiếp theo (ví dụ, H.267 hoặc H.268, v.v.)

Sáng chế này thể hiện các phương án khác nhau của việc lập mã video/ảnh, và các phương án có thể được thực hiện kết hợp với nhau trừ khi được đề cập theo cách khác.

Trong sáng chế này, video có thể đề cập đến chuỗi các ảnh theo thời gian. Hình ảnh nói chung đề cập đến đơn vị đang biểu diễn một ảnh trong vùng thời gian cụ thể, và hình ảnh con/lát/ngói là đơn vị cấu thành một phần của hình ảnh trong khi lập mã. Hình ảnh con/lát/ngói có thể gồm một hoặc nhiều đơn vị cây lập mã (Coding tree unit, CTU). Một hình ảnh có thể bao gồm một hoặc nhiều hình ảnh con/lát/ngói. Một hình ảnh có thể bao gồm một hoặc nhiều nhóm ngôi. Một nhóm ngôi có thể chứa một hoặc nhiều ngôi. Gạch có thể biểu diễn vùng hình chữ nhật của các hàng CTU bên trong ngôi trong hình ảnh. Ngôi có thể được phân chia thành nhiều gạch, trong số đó mỗi gạch bao gồm

một hoặc nhiều hàng CTU bên trong ngói. Ngói không được phân vùng thành nhiều gạch cũng có thể được gọi là gạch. Việc quét gạch là việc lập thứ tự các CTU theo trình tự cụ thể đang phân vùng hình ảnh trong đó các CTU được lập thứ tự liên tiếp theo quét mảnh CTU trong gạch, các gạch bên trong ngói được lập thứ tự liên tiếp theo quét mảnh các gạch của ngói, và các ngói trong ảnh được lập thứ tự liên tiếp theo việc quét mảnh các ngói của hình ảnh. Ngoài ra, hình ảnh con có thể biểu diễn vùng hình chữ nhật gồm một hoặc nhiều lát bên trong hình ảnh. Tức là, hình ảnh con chứa một hoặc nhiều lát mà nhìn chung bao gồm cả vùng hình chữ nhật của hình ảnh. Ngói là vùng hình chữ nhật của các CTU nằm trong cột ngói cụ thể và hàng ngói cụ thể trong hình ảnh. Cột ngói là vùng hình chữ nhật của các CTU có chiều cao bằng chiều cao của hình ảnh và chiều rộng được chỉ rõ bởi các phần tử cú pháp trong tập hợp tham số hình ảnh. Hàng ngói là vùng hình chữ nhật của các CTU có chiều cao được đặc tả bằng các phần tử cú pháp trong tập tham số hình ảnh và chiều rộng bằng chiều rộng của hình ảnh. Quét ngói là việc lập thứ tự các CTU theo trình tự cụ thể đang phân chia hình ảnh trong đó các CTU được lập thứ tự liên tiếp trong quét mảnh CTU trong ngói, trái lại các ngói trong hình ảnh được lập thứ tự liên tiếp trong quét mảnh các ngói của hình ảnh. Lát bao gồm một số gạch nguyên của hình ảnh mà có thể được chứa dành riêng theo đơn vị NAL đơn lẻ. Lát có thể gồm hoặc là số các ngói hoàn chỉnh hoặc chỉ là chuỗi liên tiếp của các viên gạch hoàn chỉnh của một ngói. Các nhóm ngói và lát có thể được sử dụng thay thế cho nhau trong sáng chế này. Ví dụ, trong sáng chế, phần đầu nhóm ngói/nhóm lát có thể được gọi là lát/phần đầu lát.

Điểm ảnh (pixel) hoặc pel có thể nghĩa là đơn vị nhỏ nhất cấu thành một hình ảnh (hoặc ảnh). Ngoài ra, "mẫu" có thể được sử dụng như là thuật ngữ tương ứng với điểm ảnh. Mẫu nhìn chung có thể biểu diễn điểm ảnh hoặc giá trị của điểm ảnh, và có thể biểu diễn chỉ điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ.

Đơn vị có thể biểu diễn đơn vị cơ sở của việc xử lý ảnh. Đơn vị có thể gồm ít nhất một trong số vùng cụ thể của hình ảnh và thông tin liên quan đến vùng. Một đơn vị có thể bao gồm một khối độ chói và hai khối sắc độ (ví dụ, cb, cr). Đơn vị có thể được sử dụng theo kiểu thay thế cho nhau theo nghĩa như là khối hoặc khu vực trong một số trường hợp. Trong trường hợp nói chung, khối $M \times N$ có thể bao gồm các mẫu (hoặc các

mảng mẫu) hoặc tập (hoặc mảng) các hệ số biến đổi của M cột và N hàng.

Trong sáng chế, “A hoặc B” có thể nghĩa là “chỉ A”, “chỉ B” hoặc “cả A và B”. Nói cách khác, trong sáng chế này, “A hoặc B” có thể được diễn dịch như “A và/hoặc B”. Ví dụ, thuật ngữ “A, B hoặc C” ở đây có nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “bất kỳ và tổ hợp bất kỳ của A, B và C”.

Dấu gạch chéo (/) hoặc dấu phẩy được sử dụng trong sáng chế có nghĩa là “và/hoặc”. Ví dụ, “A/B” có thể có nghĩa là “A và/hoặc B”. Theo đó, “A/B” có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ví dụ, “A, B, C” có thể có nghĩa là “A, B hoặc C”.

Trong sáng chế, “ít nhất một trong số A và B” có thể chỉ “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ngoài ra, theo sáng chế này, cách biểu diễn “ít nhất một trong số A hoặc B” hoặc “ít nhất một trong số A và/hoặc B” có thể được diễn dịch giống như “ít nhất một trong số A và B”.

Ngoài ra, trong sáng chế này, “ít nhất một trong số A, B và C” nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “sự kết hợp bất kỳ của A, B và C”. Cũng như vậy, “ít nhất một trong số A, B hoặc C” hoặc “ít nhất một trong số A, B và/hoặc C” có thể có nghĩa là “ít nhất một trong số A, B và C”.

Ngoài ra, các dấu ngoặc đơn được sử dụng trong sáng chế này có thể nghĩa là “ví dụ”. Cụ thể là, khi “việc dự đoán (nội dự đoán)” được chỉ báo, thì “nội dự đoán” có thể được đề xuất như ví dụ của “dự đoán”. Nói cách khác, “việc dự đoán” trong sáng chế này có thể không bị giới hạn ở “việc nội dự đoán”, và “việc nội dự đoán” có thể được đề xuất như ví dụ của “việc dự đoán”. Cũng như vậy, thậm chí khi “việc dự đoán (nội dự đoán)” được thể hiện, thì “nội dự đoán” có thể được đề xuất như ví dụ của “dự đoán”.

Trong sáng chế này, các đặc điểm kỹ thuật mà được mô tả riêng lẻ trong phạm vi một hình vẽ có thể được thực hiện riêng lẻ hoặc có thể được thực hiện cùng lúc.

Các hình vẽ sau đây được tạo ra để giải thích ví dụ cụ thể của sáng chế. Bởi vì các tên của các thiết bị cụ thể được mô tả trong các hình vẽ hoặc các tên của các tín hiệu/thông điệp/trường cụ thể được biểu thị theo cách ví dụ, nên các đặc điểm kỹ thuật của sáng chế này không được giới hạn ở các tên cụ thể được sử dụng trong các hình vẽ sau đây.

Fig.2 là giản đồ sơ lược minh họa cấu hình của thiết bị mã hóa video/ảnh mà phương án (các phương án) của sáng chế này có thể được áp dụng với. Dưới đây, thiết bị mã hóa video có thể bao gồm thiết bị mã hóa ảnh.

Tham chiếu đến Fig.2, thiết bị mã hóa 200 bao gồm bộ phân vùng ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, và bộ mã hóa entropy 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể bao gồm bộ liên dự đoán 221 và bộ nội dự đoán 222. Bộ xử lý phần dư 230 có thể bao gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ giải lượng tử hóa 234, và bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn bao gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ tái dựng hoặc bộ tạo ra khối được tái dựng. Bộ phân vùng ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropy 240, bộ cộng 250, và bộ lọc 260 có thể được tạo cấu hình bởi ít nhất một thành phần phần cứng (ví dụ, bộ chip bộ mã hóa hoặc bộ xử lý) theo phương án. Ngoài ra, bộ nhớ 270 có thể bao gồm bộ đệm hình ảnh được giải mã (DPB) hoặc có thể được tạo cấu hình bằng phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn bao gồm bộ nhớ 270 như thành phần bên trong/bên ngoài.

Bộ phân vùng ảnh 210 có thể phân vùng ảnh đầu vào (hoặc hình ảnh hoặc khung) được nhập vào thiết bị mã hóa 200 vào trong một hoặc nhiều bộ xử lý. Ví dụ, bộ xử lý có thể được gọi là đơn vị lập mã (Coding unit, CU). Trong trường hợp này, đơn vị lập mã có thể được phân vùng theo cách đệ quy theo cấu trúc cây tứ phân nhị phân tam phân (Quad-tree binary-tree ternary-tree, QTBTBT) từ đơn vị cây lập mã (Coding tree unit, CTU) hoặc đơn vị lập mã lớn nhất (Largest coding unit, LCU). Ví dụ, một đơn vị tạo mã có thể được phân vùng thành nhiều đơn vị tạo mã có độ sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc tam phân. Trong trường hợp này, ví dụ, cấu trúc cây tứ phân có thể được áp dụng trước tiên và cấu trúc cây nhị phân và/hoặc cấu trúc tam phân có thể được áp dụng sau. Theo cách thay thế, cấu trúc cây nhị phân có thể được áp dụng trước tiên. Thủ tục tạo mã theo sáng chế có thể được thực hiện dựa trên đơn vị tạo mã cuối cùng mà không được phân vùng thêm nữa. Trong trường hợp này, đơn vị tạo mã lớn nhất có thể được dùng làm đơn vị tạo mã cuối cùng dựa trên hiệu quả tạo mã theo các đặc tính của hình ảnh, hoặc nếu cần thiết, đơn vị tạo mã có thể được phân vùng theo cách đệ quy thành các đơn vị tạo mã có chiều sâu sâu hơn, và đơn vị tạo mã có kích cỡ tối ưu có thể được sử dụng làm đơn vị tạo mã cuối cùng. Ở đây, thủ

tục tạo mã có thể gồm thủ tục dự đoán, biến đổi, và tái tạo, vốn sẽ được mô tả sau. Như ví dụ khác, bộ xử lý có thể còn bao gồm đơn vị dự đoán (Prediction unit, PU) hoặc đơn vị biến đổi (Transform Unit, TU). Trong trường hợp này, đơn vị dự đoán và đơn vị biến đổi có thể được chia tách và được phân vùng từ đơn vị tạo mã cuối cùng được đề cập ở trên. Đơn vị dự đoán có thể là đơn vị của mẫu dự đoán, và đơn vị biến đổi có thể là đơn vị để dẫn xuất hệ số biến đổi và/hoặc đơn vị để dẫn xuất tín hiệu phần dư từ hệ số biến đổi.

Đơn vị có thể được sử dụng hoán đổi cho nhau bằng các thuật ngữ như là khối hoặc khu vực trong một số trường hợp. Trong trường hợp nói chung, khối $M \times N$ có thể biểu diễn tập gồm các mẫu hoặc các hệ số biến đổi bao gồm M cột và N hàng. Mẫu nói chung là có thể biểu diễn điểm ảnh hoặc giá trị của điểm ảnh, có thể biểu diễn chỉ điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc biểu diễn chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ. Mẫu có thể được sử dụng như thuật ngữ tương ứng với một hình ảnh (hoặc ảnh) cho điểm ảnh hoặc pel.

Trong thiết bị mã hóa 200, tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán) được xuất ra từ bộ liên dự đoán 221 hoặc bộ nội dự đoán 222 được trừ khỏi tín hiệu ảnh đầu vào (khối gốc, mảng mẫu gốc) để tạo ra khối phần dư, tín hiệu phần dư, mảng mẫu phần dư, và tín hiệu phần dư được tạo ra được truyền đến bộ biến đổi 232. Trong trường hợp này, như được thể hiện, đơn vị để trừ tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán) khỏi tín hiệu ảnh đầu vào (khối gốc, mảng mẫu gốc) trong bộ mã hóa 200 có thể được gọi là bộ trừ 231. Bộ dự đoán có thể thực hiện việc dự đoán trên khối cần được xử lý (ở dưới đây, được đề cập đến như là khối hiện tại) và tạo ra khối được dự đoán bao gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định liệu việc nội dự đoán hay liên dự đoán được áp dụng trên cơ sở khối hiện tại hay CU. Như được mô tả sau trong phần mô tả về mỗi chế độ dự đoán, bộ dự đoán có thể tạo ra thông tin khác nhau liên quan đến dự đoán, như thông tin chế độ dự đoán, và truyền thông tin được tạo ra đến bộ mã hóa entropy 240. Thông tin về việc dự đoán có thể được mã hóa trong bộ mã hóa entropy 240 và được xuất ra dưới dạng luồng bit.

Bộ nội dự đoán 222 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong hình ảnh hiện tại. Trong trường hợp chung, khối $M \times N$ có thể biểu diễn bộ các mẫu hoặc các hệ số biến đổi được tạo thành từ M cột và N hàng. Trong nội dự đoán,

các chế độ dự đoán có thể bao gồm nhiều chế độ vô hướng và nhiều chế độ có hướng. Các chế độ không hướng có thể bao gồm, ví dụ, chế độ DC và chế độ phẳng. Chế độ có hướng có thể bao gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng theo mức độ chi tiết của hướng dự đoán. Tuy nhiên, điều này chỉ là ví dụ, nhiều hoặc ít chế độ dự đoán hơn có thể được sử dụng phụ thuộc vào việc cài đặt. Bộ nội dự đoán 222 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng cách sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ liên dự đoán 221 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên hình ảnh tham chiếu. Ở đây, để làm giảm lượng thông tin chuyển động được truyền trong chế độ liên dự đoán, thông tin chuyển động có thể được dự đoán trong các đơn vị là các khối, các khối con, hoặc các mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng liên dự đoán (dự đoán L0, dự đoán L1, song dự đoán, v.v.). Trong trường hợp liên dự đoán, khối lân cận có thể bao gồm khối lân cận không gian có mặt trong hình ảnh hiện tại và khối lân cận thời gian có mặt trong hình ảnh tham chiếu. Hình ảnh tham chiếu bao gồm khối tham chiếu và hình ảnh tham chiếu bao gồm khối lân cận thời gian có thể là giống hoặc khác nhau. Khối lân cận thời gian có thể được gọi là khối tham chiếu được sắp xếp vào một chỗ, CU được đặt cùng vị trí (Co-located CU, colCU), và tương tự, và hình ảnh tham chiếu bao gồm khối lân cận thời gian có thể được gọi là hình ảnh được sắp xếp vào một chỗ (Collocated picture, colPic). Ví dụ, bộ liên dự đoán 221 có thể tạo cấu trúc danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và tạo ra thông tin chỉ báo ứng viên nào được sử dụng để dẫn xuất vector chuyển động và/hoặc chỉ số hình ảnh tham chiếu của khối hiện tại. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau. Ví dụ, trong trường hợp chế độ bỏ qua và chế độ hợp nhất, bộ liên dự đoán 221 có thể sử dụng thông tin chuyển động của khối lân cận như thông tin chuyển động của khối hiện tại. Trong chế độ bỏ qua, không giống chế độ hợp nhất, tín hiệu phần dư có thể không được truyền. Trong trường hợp của chế độ dự đoán vector chuyển động (Motion vector prediction, MVP), vector chuyển động của khối lân cận có thể được sử dụng làm bộ dự đoán vector chuyển động và vector chuyển động

của khối hiện tại có thể được chỉ thị bằng cách báo hiệu mức chênh lệch vector chuyển động.

Khối lân cận thời gian có thể được gọi là khối tham chiếu được sắp xếp vào một chỗ, CU được đặt cùng vị trí (Co-located CU, colCU), và tương tự, và hình ảnh tham chiếu bao gồm khối lân cận thời gian có thể được gọi là hình ảnh được sắp xếp vào một chỗ (Collocated picture, colPic). Khối lân cận thời gian có thể được gọi là khối tham chiếu được sắp xếp vào một chỗ, CU được đặt cùng vị trí (Co-located CU, colCU), và tương tự, và hình ảnh tham chiếu bao gồm khối lân cận thời gian có thể được gọi là hình ảnh được sắp xếp vào một chỗ (Collocated picture, colPic). Điều này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (Combined inter and intra prediction, CIIP). Bên cạnh đó, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép nội khối (IBC) hoặc chế độ bảng màu cho việc dự đoán khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc mã hóa hình ảnh/video nội dung của trò chơi hoặc tương tự, ví dụ, tạo mã nội dung màn (screen content coding, SCC). Về cơ bản, IBC thực hiện việc dự đoán trong hình ảnh hiện tại nhưng có thể được thực hiện tương tự cho việc liên dự đoán trong đó khối tham chiếu được dẫn xuất trong hình ảnh hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong các kỹ thuật liên dự đoán được mô tả trong sáng chế. Chế độ bảng màu có thể được coi như là ví dụ về việc tạo mã trong ảnh hoặc việc nội dự đoán. Khi chế độ bảng màu (palette) được áp dụng, thì giá trị mẫu nằm trong hình ảnh có thể được báo hiệu dựa trên thông tin về bảng của bảng màu (palette) và chỉ số bảng màu (palette).

Tín hiệu dự đoán được tạo ra bởi bộ dự đoán (bao gồm bộ liên dự đoán 221 và/hoặc bộ nội dự đoán 222) có thể được sử dụng để tạo ra tín hiệu được tái dựng hoặc để tạo ra tín hiệu phần dư. Bộ biến đổi 232 có thể tạo ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu phần dư. Ví dụ, kỹ thuật biến đổi có thể bao gồm ít nhất một kỹ thuật trong số biến đổi cosin rời rạc (discrete cosine transform, DCT), biến đổi sin rời rạc (discrete sine transform, DST), biến đổi Karhunen-Loève (Karhunen-Loève transform, KLT), biến đổi dựa trên đồ thị (graph-based transform, GBT), hoặc biến đổi phi tuyến có điều kiện (Conditionally non-linear transform, CNT). Ở đây, GBT có nghĩa là phép biến đổi thu nhận được từ đồ thị khi thông tin mối quan hệ giữa các điểm ảnh được biểu diễn bởi đồ thị. CNT đề cập đến phép biến đổi được tạo ra dựa trên tín hiệu

dự đoán được tạo ra sử dụng tất cả các điểm ảnh được tái dựng từ trước. Bên cạnh đó, quy trình biến đổi có thể được áp dụng cho các khối điểm ảnh hình vuông có cùng kích cỡ hoặc có thể được áp dụng cho các khối có kích cỡ có thể thay đổi được khác hình vuông.

Bộ lượng tử hóa 233 có thể lượng tử hóa các hệ số biến đổi và truyền chúng tới bộ mã hóa entropy 240 và bộ mã hóa entropy 240 có thể mã hóa tín hiệu đã lượng tử hóa (thông tin về các hệ số biến đổi đã được lượng tử hóa) và xuất ra luồng bit. Thông tin về các hệ số biến đổi đã được lượng tử hóa có thể được đề cập đến như thông tin phần dư. Bộ lượng tử hóa 233 có thể sắp xếp lại các hệ số biến đổi đã được lượng tử hóa dạng khối thành dạng vector một chiều dựa trên thứ tự quét hệ số và tạo ra thông tin về các hệ số biến đổi đã được lượng tử hóa dựa trên các hệ số biến đổi đã được lượng tử hóa ở dạng vector một chiều. Thông tin về các hệ số biến đổi có thể được tạo ra. Bộ mã hóa entropy 240 có thể thực hiện các phương pháp mã hóa khác nhau, ví dụ, như Golomb hàm số mũ, mã hoá chiều dài biến đổi thích ứng ngữ cảnh (Context-Adaptive Length Coding, CAVLC), mã hoá số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC), và tương tự. Bộ mã hóa entropy 240 có thể mã hóa thông tin cần thiết cho việc dựng lại video/ảnh khác với các hệ số biến đổi đã được lượng tử hóa (ví dụ, các giá trị của các phân tử cú pháp, v.v.) cùng nhau hoặc tách biệt. Thông tin được mã hoá (ví dụ, thông tin video/ảnh được mã hóa) có thể được truyền hoặc được lưu giữ trong các đơn vị là lớp trừu tượng mạng (Network Abstraction Layer, NAL) ở dạng luồng bit. Thông tin video/ảnh có thể còn bao gồm thông tin về các tập tham số khác nhau như tập tham số thích ứng (Adaptation parameter set, APS), tập tham số hình ảnh (Picture parameter set, PPS), tập tham số trình tự (Sequence parameter set, SPS), hoặc tập tham số video (Video parameter set, VPS). Thêm vào đó, thông tin video/hình ảnh có thể còn bao gồm thông tin ràng buộc chung. Trong sáng chế này, thông tin và/hoặc các phân tử cú pháp được truyền/được báo hiệu từ thiết bị mã hóa đến thiết bị giải mã có thể được gồm trong thông tin video/ảnh. Thông tin video/hình ảnh có thể được mã hóa thông qua thủ tục mã hóa được mô tả ở trên và được gồm trong luồng bit. Luồng bit có thể được truyền qua mạng hoặc có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Mạng có thể gồm mạng phát rộng và/hoặc mạng truyền thông, và phương tiện lưu trữ kỹ thuật số có thể gồm các phương tiện lưu trữ khác nhau như USB,

SD, CD, DVD, Blu-ray, HDD, SSD, và tương tự. Bộ truyền (không được thể hiện) truyền tín hiệu xuất ra từ bộ mã hóa entropy 240 và/hoặc đơn vị lưu trữ (không được thể hiện) đang lưu tín hiệu có thể được bao gồm như thành phần bên trong/bên ngoài của thiết bị mã hóa 200, và theo cách khác, bộ truyền có thể được bao gồm trong bộ mã hóa entropy 240.

Các hệ số biến đổi đã được lượng tử hóa được xuất ra từ bộ lượng tử hóa 233 có thể được sử dụng để tạo ra tín hiệu dự đoán. Ví dụ, tín hiệu phần dư (khối phần dư hoặc các mẫu phần dư) có thể được tái dựng bằng cách áp dụng việc giải lượng tử hóa và biến đổi ngược đối với các hệ số biến đổi đã được lượng tử hóa qua bộ giải lượng tử hóa 234 và bộ biến đổi ngược 235. Bộ cộng 250 cộng tín hiệu phần dư được tái dựng vào tín hiệu dự đoán được xuất ra từ bộ liên dự đoán 221 hoặc bộ nội dự đoán 222 để tạo ra tín hiệu được tái dựng (hình ảnh được tái dựng, khối được tái dựng, mảng mẫu được tái dựng). Nếu không có phần dư nào cho khối cần được xử lý, như trường hợp mà trong đó chế độ bỏ qua được áp dụng, thì khối được dự đoán có thể được sử dụng làm khối được tái dựng. Bộ cộng 250 có thể được gọi là bộ tái dựng hoặc bộ tạo ra khối được tái dựng. Tín hiệu được tái dựng được tạo ra có thể được sử dụng cho việc nội dự đoán khối tiếp theo cần được xử lý trong hình ảnh hiện tại và có thể được sử dụng cho việc liên dự đoán hình ảnh tiếp theo qua việc lọc như được mô tả bên dưới.

Trong khi đó, việc ánh xạ độ chói với việc định cỡ sắc độ (luma mapping with chroma scaling, LMCS) có thể được áp dụng trong suốt quá trình mã hóa và/hoặc tái dựng ảnh.

Bộ lọc 260 có thể nâng cao chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được tái dựng. Ví dụ, bộ lọc 260 có thể tạo ra hình ảnh được tái dựng được điều chỉnh bằng cách áp dụng các phương pháp lọc khác nhau đối với hình ảnh được tái dựng và lưu trữ hình ảnh được tái dựng được điều chỉnh trong bộ nhớ 270, cụ thể, DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể bao gồm, ví dụ, lọc khử khối, bù trừ tương ứng mẫu, lọc vòng tương thích, lọc hai bên, và tương tự. Bộ lọc 260 có thể tạo ra thông tin khác nhau liên quan tới việc lọc và truyền thông tin được tạo ra đến bộ mã hóa entropy 240 như sẽ được mô tả sau trong phần mô tả về mỗi phương pháp lọc. Thông tin liên quan đến việc lọc có thể được mã hóa bởi bộ mã hóa entropy 240 và được xuất ra trong dạng luồng bit.

Ảnh được tái dựng được điều chỉnh mà được truyền đến bộ nhớ 270 là có thể được sử dụng làm ảnh tham chiếu ở bộ liên dự đoán 221. Khi việc liên dự đoán được áp dụng qua thiết bị mã hóa, thì việc không khớp dự đoán giữa thiết bị mã hóa 200 và thiết bị giải mã 300 có thể tránh được và hiệu quả mã hóa có thể được nâng cao.

DPB của bộ nhớ DPB 270 có thể lưu hình ảnh được tái dựng được điều chỉnh để sử dụng như là hình ảnh tham chiếu trong bộ liên dự đoán 221. Bộ nhớ 270 có thể lưu trữ thông tin chuyển động của khối mà thông tin chuyển động trong ảnh hiện tại được dẫn xuất (hoặc được mã hóa) từ đó và/hoặc thông tin chuyển động của các khối trong hình ảnh mà đã được tái dựng. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ liên dự đoán 221 và được sử dụng như là thông tin chuyển động của khối lân cận không gian hoặc thông tin chuyển động của khối lân cận thời gian. Bộ nhớ 270 có thể lưu trữ các mẫu được tái dựng của các khối được tái dựng trong hình ảnh hiện tại, và có thể truyền các mẫu được tái dựng này đến bộ nội dự đoán 222.

Fig.3 là sơ đồ giản lược minh họa cấu hình của thiết bị giải mã video/ảnh mà phương án (các phương án) của sáng chế có thể được áp dụng với.

Tham chiếu đến Fig.3, thiết bị giải mã 300 có thể chứa bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, bộ lọc 350, và bộ nhớ 360. Bộ dự đoán 330 có thể bao gồm bộ liên dự đoán 331 và bộ nội dự đoán 332. Bộ xử lý phần dư 320 có thể bao gồm bộ giải lượng tử hóa 321 và bộ biến đổi ngược 322. Bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350 có thể được đặt cấu hình bởi thành phần phần cứng (ví dụ bộ chip hoặc bộ xử lý bộ giải mã) theo một phương án thực hiện. Ngoài ra, bộ nhớ 360 có thể bao gồm bộ đệm hình ảnh được giải mã (DPB) hoặc có thể được tạo cấu hình bằng phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn bao gồm bộ nhớ 360 như thành phần bên trong/bên ngoài.

Khi luồng bit bao gồm thông tin video/ảnh được nhập vào, thì thiết bị giải mã 300 có thể tái dựng ảnh tương ứng với quy trình trong đó thông tin video/ảnh được xử lý trong thiết bị mã hóa của Fig.2. Ví dụ, thiết bị giải mã 300 có thể dẫn xuất các đơn vị/các khối dựa trên thông tin liên quan đến sự phân vùng khối thu được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã sử dụng bộ xử lý được áp dụng trong thiết bị mã hóa. Do đó, bộ xử lý việc giải mã có thể đơn vị lập mã chẳng hạn và đơn vị lập mã

có thể được phân vùng theo cấu trúc cây tứ phân, cấu trúc cây nhị phân và/hoặc cấu trúc cây tam phân từ đơn vị cây lập mã hoặc đơn vị lập mã lớn nhất. Một hoặc nhiều đơn vị biến đổi có thể được dẫn ra từ đơn vị lập mã. Tín hiệu ảnh được tái dựng được giải mã và xuất ra qua thiết bị giải mã 300 có thể được tái tạo thông qua thiết bị tái tạo.

Thiết bị giải mã 300 có thể thu tín hiệu mà được xuất ra từ thiết bị mã hóa trên Fig.2 dưới dạng luồng bit, và tín hiệu thu được có thể được giải mã thông qua bộ giải mã entropy 310. Ví dụ, bộ giải mã entropy 310 có thể phân tích cú pháp luồng bit để dẫn xuất thông tin (ví dụ, thông tin video/ảnh) cần thiết cho việc tái dựng ảnh (hoặc tái dựng hình ảnh). Thông tin video/ảnh có thể còn bao gồm thông tin về các tập tham số khác nhau như tập tham số thích ứng (Adaptation parameter set, APS), tập tham số hình ảnh (Picture parameter set, PPS), tập tham số trình tự (Sequence parameter set, SPS), hoặc tập tham số video (Video parameter set, VPS). Thêm vào đó, thông tin video/hình ảnh có thể còn bao gồm thông tin ràng buộc chung. Thiết bị giải mã còn có thể giải mã ảnh dựa trên thông tin về tập hợp thông số và/hoặc thông tin ràng buộc chung. Thông tin được báo hiệu/nhận được và/hoặc các phần tử cú pháp được mô tả sau trong sáng chế có thể được giải mã có thể giải mã thủ tục giải mã và thu nhận được từ luồng bit. Ví dụ, bộ giải mã entropy 310 giải mã thông tin trong luồng bit dựa trên phương pháp lập mã như lập mã Golomb hàm số mũ, CAVLC hoặc CABAC, và xuất ra các phần tử cú pháp được yêu cầu cho việc tái dựng ảnh và các giá trị được lượng tử hóa của các hệ số biến đổi cho phần dư. Cụ thể hơn, phương pháp giải mã entropy CABAC có thể thu ngắn (bin) tương ứng với mỗi phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh sử dụng thông tin về phần tử cú pháp mục tiêu giải mã, thông tin về khối mục tiêu giải mã hoặc thông tin về ký hiệu/bin được giải mã trong pha trước, và thực hiện việc giải mã số học trên bin này bằng cách dự đoán xác suất xuất hiện bin theo mô hình ngữ cảnh được xác định, và tạo ra ký hiệu tương ứng với giá trị của mỗi phần tử cú pháp. Trong trường hợp này, phương pháp giải mã entropy CABAC có thể cập nhật mô hình ngữ cảnh bằng cách sử dụng thông tin về ký hiệu/bin được giải mã cho mô hình ngữ cảnh của ký hiệu/bin tiếp theo sau khi xác định mô hình ngữ cảnh. Thông tin liên quan đến việc dự đoán trong số thông tin được giải mã bởi bộ giải mã entropy 310 có thể được cung cấp đến bộ dự đoán (bộ liên dự đoán 332 và bộ nội dự đoán 331), và giá trị phần dư mà việc giải mã entropy được thực hiện trên đó trong bộ giải mã entropy 310,

tức là, các hệ số biến đổi được lượng tử hóa và thông tin tham số liên quan, có thể được cấp vào bộ xử lý phần dư 320. Bộ xử lý phần dư 320 có thể dẫn xuất tín hiệu phần dư (khối phần dư, các mẫu phần dư, mảng mẫu phần dư). Ngoài ra, thông tin về việc lọc trong số thông tin được giải mã bởi bộ giải mã entropy 310 có thể được cung cấp đến bộ lọc 350. Trong khi đó, bộ thu (không được thể hiện) để thu tín hiệu được xuất ra từ thiết bị mã hóa có thể còn được tạo cấu hình như phần tử bên trong/bên ngoài của thiết bị giải mã 300, hoặc bộ thu có thể là thành phần của bộ giải mã entropy 310. Trong khi đó, thiết bị giải mã theo sáng chế này có thể được đề cập đến như là thiết bị giải mã video/ảnh/hình ảnh, và thiết bị giải mã có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin video/ảnh/hình ảnh) và bộ giải mã mẫu (bộ giải mã mẫu video/ảnh/hình ảnh). Bộ giải mã thông tin có thể bao gồm bộ giải mã entropy 310, và bộ giải mã mẫu có thể bao gồm ít nhất một trong bộ giải lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ liên dự đoán 332, và bộ nội dự đoán 331.

Bộ giải lượng tử hóa 321 có thể giải lượng tử hóa các hệ số biến đổi được lượng tử hóa và xuất ra các hệ số biến đổi. Bộ giải lượng tử hóa 321 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa dưới dạng của loại khối hai chiều. Trong trường hợp này, việc sắp xếp lại có thể được thực hiện dựa trên thứ tự quét hệ số được thực hiện trong thiết bị mã hóa. Bộ giải lượng tử hóa 321 có thể thực hiện việc giải lượng tử hóa trên các hệ số biến đổi được lượng tử hóa bằng cách sử dụng tham số lượng tử hóa (ví dụ, thông tin kích thước bước lượng tử hóa) và thu được các hệ số biến đổi.

Bộ biến đổi ngược 322 biến đổi ngược các hệ số biến đổi để thu được tín hiệu phần dư (khối phần dư, mảng mẫu phần dư).

Bộ dự đoán có thể thực hiện việc dự đoán trên khối hiện tại và tạo ra khối được dự đoán bao gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định liệu việc nội dự đoán hoặc việc liên dự đoán được áp dụng cho khối hiện tại dựa trên thông tin về dự đoán được xuất ra từ bộ giải mã entropy 310 hay không và có thể xác định chế độ nội/liên dự đoán cụ thể.

Bộ dự đoán 320 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau được mô tả bên dưới. Ví dụ, bộ dự đoán có thể không chỉ áp dụng việc nội

dự đoán hoặc liên dự đoán để dự đoán một khối mà còn có thể áp dụng đồng thời việc nội dự đoán và liên dự đoán. Điều này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (Combined inter and intra prediction, CIIP). Bên cạnh đó, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép nội khối (IBC) hoặc chế độ bảng màu cho việc dự đoán khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc mã hóa hình ảnh/video nội dung của trò chơi hoặc tương tự, ví dụ, tạo mã nội dung màn (screen content coding, SCC). Về cơ bản, IBC thực hiện việc dự đoán trong hình ảnh hiện tại nhưng có thể được thực hiện tương tự cho việc liên dự đoán trong đó khối tham chiếu được dẫn xuất trong hình ảnh hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong các kỹ thuật liên dự đoán được mô tả trong sáng chế. Chế độ bảng màu có thể được coi như là ví dụ về việc nội lập mã hoặc việc nội dự đoán. Khi chế độ bảng màu (palette) được áp dụng, thì giá trị mẫu nằm trong hình ảnh có thể được báo hiệu dựa trên thông tin về bảng của bảng màu (palette) và chỉ số bảng màu (palette).

Bộ nội dự đoán 331 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong hình ảnh hiện tại. Trong trường hợp chung, khối $M \times N$ có thể biểu diễn bộ các mẫu hoặc các hệ số biến đổi được tạo thành từ M cột và N hàng. Trong nội dự đoán, các chế độ dự đoán có thể bao gồm nhiều chế độ vô hướng và nhiều chế độ có hướng. Bộ nội dự đoán 331 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng cách sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ liên dự đoán 332 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên hình ảnh tham chiếu. Trong trường hợp này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ liên dự đoán, thông tin chuyển động có thể được dự đoán trong các đơn vị là các khối, các khối con, hoặc các mẫu dựa trên mối tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng liên dự đoán (dự đoán $L0$, dự đoán $L1$, song dự đoán, v.v.). Trong trường hợp liên dự đoán, khối lân cận có thể bao gồm khối lân cận không gian có mặt trong hình ảnh hiện tại và khối lân cận thời gian có mặt trong hình ảnh tham chiếu. Ví dụ, bộ liên dự đoán 332 có thể tạo cấu trúc danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và dẫn xuất vector chuyển động của khối hiện tại và/hoặc

chỉ số hình ảnh tham chiếu dựa trên thông tin chọn ứng viên nhận được. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể bao gồm thông tin chỉ báo chế độ liên dự đoán cho khối hiện tại.

Bộ cộng 340 có thể tạo ra tín hiệu được tái dựng (hình ảnh được tái dựng, khối được tái dựng, mảng mẫu được tái dựng) bằng cộng tín hiệu phần dư thu được vào tín hiệu dự đoán (khối được dự đoán, mảng mẫu được dự đoán) được xuất ra từ bộ dự đoán (bao gồm bộ liên dự đoán 332 và/hoặc bộ nội dự đoán 331). Nếu không có phần dư nào cho khối cần được xử lý, như khi chế độ bỏ qua được áp dụng, thì khối được dự đoán có thể được sử dụng như khối được tái dựng.

Bộ cộng 340 có thể được gọi là bộ tái dựng hoặc bộ tạo khối được tái dựng. Tín hiệu được tái dựng được tạo ra có thể được sử dụng cho việc nội dự đoán khối tiếp theo cần được xử lý trong hình ảnh hiện tại, có thể được xuất ra qua bước lọc được mô tả ở trên, hoặc có thể được sử dụng cho việc liên dự đoán hình ảnh tiếp theo.

Trong khi đó, phép ánh xạ độ chói cùng với việc định tỷ lệ sắc độ (LMCS) có thể được áp dụng trong quy trình giải mã ảnh.

Bộ lọc 350 có thể nâng cao chất lượng hình ảnh chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được tái dựng. Ví dụ, bộ lọc 350 có thể tạo ra hình ảnh được tái dựng được điều chỉnh bằng cách áp dụng các phương pháp lọc khác nhau đối với hình ảnh được tái dựng và lưu trữ hình ảnh được tái dựng được điều chỉnh trong bộ nhớ 360, cụ thể, DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể bao gồm, ví dụ, lọc khử khối, bù trừ tương ứng mẫu, lọc vòng tương thích, lọc hai bên, và tương tự.

Hình ảnh được tái dựng (được điều chỉnh) được lưu trữ trong DPB của bộ nhớ 360 có thể được sử dụng như hình ảnh tham chiếu trong bộ liên dự đoán 332. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối mà thông tin chuyển động trong hình ảnh hiện tại được dẫn xuất (hoặc được giải mã) từ đó và/hoặc thông tin chuyển động của các khối trong hình ảnh mà đã được tái dựng. Thông tin chuyển động được lưu trữ có thể được truyền đến bộ liên dự đoán 260 để được sử dụng như thông tin chuyển động của khối lân cận không gian hoặc thông tin chuyển động của khối lân cận thời gian. Bộ nhớ 360 có thể lưu trữ các mẫu được tái dựng của các khối được tái dựng trong hình ảnh

hiện tại và chuyển các mẫu được tái dựng đến bộ nội dự đoán 331.

Trong sáng chế này, các phương án được mô tả trong bộ lọc 260, bộ liên dự đoán 221, và bộ nội dự đoán 222 của thiết bị mã hóa 200 có thể là giống như hoặc được áp dụng lần lượt để tương ứng với bộ lọc 350, bộ liên dự đoán 332, và bộ nội dự đoán 331 của thiết bị giải mã 300. Nguyên tắc tương tự có thể cũng áp dụng với đơn vị 332 và bộ nội dự đoán 331.

Trong sáng chế này, ít nhất một trong việc lượng tử hóa/lượng tử hóa ngược và/hoặc biến đổi/biến đổi ngược có thể được lược bỏ. Khi việc lượng tử hóa/lượng tử hóa ngược được lược bỏ, thì các hệ số biến đổi được lượng tử hóa có thể được gọi là các hệ số biến đổi. Khi việc biến đổi/biến đổi ngược được lược bỏ, thì các hệ số biến đổi có thể được gọi là các hệ số hoặc các hệ số phần dư, hoặc có thể vẫn được gọi là các hệ số biến đổi cho sự đồng nhất về diễn đạt.

Trong sáng chế, hệ số biến đổi được lượng tử hóa và hệ số biến đổi có thể được đề cập đến lần lượt như hệ số biến đổi và hệ số biến đổi được định tỷ lệ. Trong trường hợp này, thông tin phần dư có thể gồm thông tin về hệ số (các hệ số) biến đổi, và thông tin về hệ số (các hệ số) biến đổi có thể được báo hiệu thông qua cú pháp tạo mã phần dư. Các hệ số biến đổi có thể được dẫn xuất dựa trên thông tin dư (hoặc thông tin về hệ số (các hệ số) biến đổi), và các hệ số biến đổi được định tỷ lệ có thể được dẫn xuất bằng cách biến đổi ngược (định tỷ lệ) trên các hệ số biến đổi. Các mẫu phần dư có thể được dẫn xuất dựa trên việc biến đổi ngược (biến đổi) trên các hệ số biến đổi được định tỷ lệ. Điều này có thể cũng được áp dụng/được thể hiện trong các phần khác của sáng chế này.

Như được mô tả ở trên, thiết bị mã hóa có thể thực hiện các phương pháp mã hóa khác nhau như là Golomb hàm số mũ, tập mã chiều dài biến đổi thích ứng ngữ cảnh (Context-Adaptive biến Length Coding, CAVLC), và lập mã số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC). Ngoài ra, thiết bị giải mã có thể giải mã thông tin trong luồng bit dựa trên phương pháp lập mã như là lập mã Golomb hàm số mũ, CAVLC hoặc CABAC, và xuất ra giá trị của phần tử cú pháp cần thiết cho việc xây dựng lại hình ảnh và các giá trị được lượng tử hóa của các hệ số biến đổi liên quan đến các phần dư.

Ví dụ, các phương pháp lập mã được mô tả ở trên có thể được thực hiện như được mô tả dưới đây.

Fig.4 là hình vẽ thể hiện theo cách ví dụ việc lập mã số học nhị phân thích ứng ngữ cảnh (CABAC) để mã hóa phần tử cú pháp. Ví dụ, trong quy trình mã hóa CABAC, khi tín hiệu đầu vào là phần tử cú pháp, chứ không phải giá trị nhị phân, thì thiết bị mã hóa có thể chuyển đổi tín hiệu đầu vào thành giá trị nhị phân bằng cách nhị phân hóa giá trị của tín hiệu đầu vào. Ngoài ra, khi tín hiệu đầu vào đã là giá trị nhị phân (tức là, khi giá trị của tín hiệu đầu vào là giá trị nhị phân), thì việc nhị phân hóa có thể không được thực hiện và có thể được vòng qua. Ở đây, mỗi số nhị phân 0 hoặc 1 cấu thành giá trị nhị phân có thể được đề cập đến như bin. Ví dụ, nếu chuỗi nhị phân sau khi nhị phân hóa là 110, thì mỗi trong số 1, 1, và 0 được gọi là một bin. Bin (các bin) cho một phần tử cú pháp có thể thể hiện giá trị của phần tử cú pháp.

Sau đây, các bin được nhị phân hóa của các phần tử cú pháp có thể được nhập vào công cụ tạo mã thông thường hoặc công cụ tạo mã đi vòng qua. Bộ máy lập mã thông thường của thiết bị mã hóa có thể phân bổ mô hình ngữ cảnh phản ánh giá trị xác suất đến bin tương ứng, và có thể mã hóa bin tương ứng dựa trên mô hình ngữ cảnh được phân bổ. Bộ máy lập mã thông thường của thiết bị mã hóa có thể cập nhật mô hình ngữ cảnh của mỗi bin sau khi thực hiện việc mã hóa trên mỗi bin. Bin được mã hóa như được mô tả ở trên có thể được đề cập đến như bin được lập mã ngữ cảnh.

Trong khi đó, khi các bin của phần tử cú pháp được nhị phân hóa được cấp vào bộ máy lập mã vòng, thì chúng có thể được lập mã như sau. Ví dụ, bộ máy lập mã vòng của thiết bị mã hóa lược bỏ thủ tục ước lượng xác suất đối với bin đầu vào và thủ tục cập nhật mô hình xác suất được áp dụng cho bin sau khi mã hóa. Khi việc mã hóa vòng được áp dụng, thì thiết bị mã hóa có thể mã hóa bin đầu vào bằng cách áp dụng sự phân bổ xác suất đồng nhất thay vì phân bổ mô hình ngữ cảnh, nhờ đó nâng cao tốc độ mã hóa. Bin được mã hóa như được mô tả ở trên có thể được đề cập đến như bin vòng.

Việc giải mã entropy có thể biểu diễn quy trình để thực hiện quy trình giống như việc mã hóa entropy được mô tả ở trên theo thứ tự ngược.

Ví dụ, khi phần tử cú pháp được giải mã dựa trên mô hình ngữ cảnh, thì thiết bị giải mã có thể thu bin tương ứng với phần tử cú pháp thông qua luồng bit, xác định mô

hình ngữ cảnh sử dụng phần tử cú pháp và giải mã thông tin của khối mục tiêu giải mã hoặc khối lân cận hoặc thông tin về ký hiệu/bin được giải mã trong pha trước, dự đoán xác suất xuất hiện của bin thu được theo mô hình ngữ cảnh được xác định, và thực hiện việc giải mã số học trên bin để dẫn xuất giá trị của phần tử cú pháp. Sau đó, mô hình ngữ cảnh của bin mà được giải mã tiếp theo có thể được cập nhật với mô hình ngữ cảnh xác định.

Cũng như vậy, ví dụ, khi phần tử cú pháp được giải mã vòng, thì thiết bị giải mã có thể thu bin tương ứng với phần tử cú pháp thông qua luồng bit, và giải mã bin đầu vào bằng cách áp dụng sự phân bố xác suất đồng nhất. Trong trường hợp này, thủ tục của thiết bị giải mã để suy ra mô hình ngữ cảnh của phần tử cú pháp và thủ tục để cập nhật mô hình ngữ cảnh được áp dụng cho bin sau khi giải mã có thể được lược bỏ.

Như được mô tả ở trên, các mẫu dư có thể được suy ra như các hệ số biến đổi được lượng tử hóa thông qua các quy trình biến đổi và lượng tử hóa. Các hệ số biến đổi được lượng tử hóa có thể cũng được đề cập đến như các hệ số biến đổi. Trong trường hợp này, các hệ số biến đổi trong khối có thể được báo hiệu dưới dạng thông tin phần dư. Thông tin phần dư có thể bao gồm cú pháp lập mã phần dư. Tức là, thiết bị mã hóa có thể tạo cấu hình cú pháp lập mã phần dư với thông tin phần dư, mã hóa phần tử này, và xuất nó ra dưới dạng luồng bit, và thiết bị giải mã có thể giải mã cú pháp lập mã dư từ luồng bit và dẫn xuất các hệ số biến đổi (được lượng tử hóa) phần dư. Cú pháp lập mã dư có thể bao gồm các phần tử cú pháp biểu diễn liệu việc biến đổi được áp dụng cho khối tương ứng hay không, vị trí của hệ số biến đổi hiệu quả cuối cùng trong khối hay không, liệu hệ số biến đổi hiệu quả tồn tại trong khối con hay không, kích thước/dấu hiệu của hệ số biến đổi hiệu quả hay không, và tương tự, như sẽ được mô tả sau.

Ví dụ, các phần tử cú pháp liên quan đến việc mã hóa/giải mã dữ liệu phần dư có thể được trình bày như được thể hiện trong bảng sau đây.

[Bảng 1]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Descriptor
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions - 1) {	
xC = CbPosX[chType][x0][y0]	
yC = CbPosY[chType][x0][y0]	
wC = CbWidth[chType][x0][y0] / SubWidthC	
hC = CbHeight[chType][x0][y0] / SubHeightC	
} else {	
xC = x0	
yC = y0	
wC = tbWidth / SubWidthC	
hC = tbHeight / SubHeightC	
}	
chromaAvailable = treeType != DUAL_TREE_LUMA && sps_chroma_format_idc != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1))	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && sps_chroma_format_idc != 0 && ((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag)))) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex == NumIntraSubPartitions - 1))))) {	
tu_cb_coded_flag[xC][yC]	ae(v)
tu_cr_coded_flag[xC][yC]	ae(v)
}	

if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag &	
&	
((subTuIndex == 0 && cu_sbt_pos_flag)	
(subTuIndex == 1 && !cu_sbt_pos_flag)) &&	
((CuPredMode[chType][x0][y0] == MODE_INTRA &&	
!cu_act_enabled_flag[x0][y0])	
(chromaAvailable && (tu_cb_coded_flag[xC][yC]	
tu_cr_coded_flag[xC][yC]))	
CbWidth[chType][x0][y0] > MaxTbSizeY	
CbHeight[chType][x0][y0] > MaxTbSizeY))	
(IntraSubPartitionsSplitType != ISP_NO_SPLIT &&	
(subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfLuma)))	
tu_y_coded_flag[x0][y0]	ae(v)
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT)	
InferTuCbfLuma = InferTuCbfLuma && !tu_y_coded_flag[x0][y0]	
}	
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y	
0] > 64	
tu_y_coded_flag[x0][y0] (chromaAvailable && (tu_cb_coded_flag	
[xC][yC]	
tu_cr_coded_flag[xC][yC])) && treeType != DUAL_TREE_CHRO	
MA &&	
pps_cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y	
0] > 64	
(chromaAvailable && (tu_cb_coded_flag[xC][yC]	
tu_cr_coded_flag[xC][yC]))) &&	
treeType != DUAL_TREE_LUMA && sh_cu_chroma_qp_offset_enable	
d_flag &&	
!IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && pps_chroma_qp_offset_list_len_minus1 >	
0)	
cu_chroma_qp_offset_idx	ae(v)
}	

if(sps_joint_cbr_enabled_flag && ((CuPredMode[chType][x0][y0] == MODE_INTRA && (tu_cb_coded_flag[xC][yC] tu_cr_coded_flag[xC][yC])) (tu_cb_coded_flag[xC][yC] && tu_cr_coded_flag[xC][yC])) && chromaAvailable)	
tu_joint_cbr_residual_flag[xC][yC]	ae(v)
if(tu_y_coded_flag[x0][y0] && treeType != DUAL_TREE_CHROMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplitType == ISP_NO_SPLIT) && !cu_sbt_flag)	
transform_skip_flag[x0][y0][0]	ae(v)
if(!transform_skip_flag[x0][y0][0] sh_ts_residual_coding_disabled_fla g)	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
if(tu_cb_coded_flag[xC][yC] && treeType != DUAL_TREE_LUMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][1]	ae(v)
if(!transform_skip_flag[xC][yC][1] sh_ts_residual_coding_disabled_fl ag)	
residual_coding(xC, yC, Log2(wC), Log2(hC), 1)	
else	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 1)	
}	
if(tu_cr_coded_flag[xC][yC] && treeType != DUAL_TREE_LUMA && !(tu_cb_coded_flag[xC][yC] && tu_joint_cbr_residual_flag[xC][y C])) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][2]	ae(v)
if(!transform_skip_flag[xC][yC][2] sh_ts_residual_coding_disabled_fl ag)	
residual_coding(xC, yC, Log2(wC), Log2(hC), 2)	
else	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 2)	
}	
}	

transform_skip_flag chỉ báo liệu việc biến đổi được bỏ qua trong khối liên quan hay không. transform_skip_flag có thể là phần tử cú pháp của cờ bỏ qua việc biến đổi. Khối liên quan có thể là khối mã hóa (Coding block, CB) hoặc khối biến đổi (Transform block, TB). Liên quan đến các thủ tục biến đổi (và lượng tử hóa) và lập mã phân dư, CB và TB có thể được sử dụng hoán đổi cho nhau. Ví dụ, như được mô tả ở trên, các mẫu dư có thể được suy ra cho CB, và các hệ số biến đổi (được lượng tử hóa) có thể được

suy ra thông qua việc biến đổi và lượng tử hóa cho các mẫu dư, và thông qua thông tin, thủ tục lập mã dư, (ví dụ, các phần tử cú pháp) chỉ báo hiệu quả vị trí, độ lớn, dấu hiệu, v.v. của các hệ số biến đổi (được lượng tử hóa) có thể được tạo ra và được báo hiệu. Các hệ số biến đổi được lượng tử hóa có thể được gọi đơn giản là các hệ số biến đổi. Nhìn chung, khi CB không lớn hơn TB lớn nhất, thì kích thước của CB có thể giống như kích thước của TB, và trong trường hợp này, khối mục tiêu cần được biến đổi (và được lượng tử hóa) và phần dư được lập mã có thể được gọi là CB hoặc TB. Trong khi đó, khi CB là lớn hơn TB tối đa, thì khối đích cần được biến đổi (và được lượng tử hóa) và phần dư được lập mã có thể được gọi là TB. Ở dưới đây, các phần tử cú pháp liên quan đến việc lập mã phần dư được báo hiệu trong các đơn vị là các khối biến đổi (TB) sẽ được mô tả nhưng đây là ví dụ và TB có thể được sử dụng hoán đổi với các khối lập mã (CB) như được mô tả ở trên.

Cùng lúc đó, các phần tử cú pháp mà được báo hiệu sau cờ bỏ qua biến đổi được báo hiệu có thể là giống như các phần tử cú pháp được bộc lộ trong Bảng 2 và/hoặc Bảng 3 dưới đây, và các phần mô tả chi tiết về các phần tử cú pháp được mô tả dưới đây.

[Bảng 2]

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth == 5 && log2TbHeight < 6)	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min(log2TbWidth, 5)	
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth < 6 && log2TbHeight == 5)	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3)	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	

do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfntDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfntZeroOutSigCoeffFlag = 0	
if((lastSubBlock > 0 lastScanPos > 0) && cIdx == 0)	
MtsDcOnly = 0	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[i][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[i][1]	
inferSbDcSigCoeffFlag = 0	
if(i < lastSubBlock && i > 0) {	
sb coded flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	

if(sb_coded_flag[xS][yS] && (xS > 3 yS > 3) && cIdx == 0)	
MtsZeroOutSigCoeffFlag = 0	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = firstPosMode0	
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
if(sb_coded_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) &	
&	
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY))	
{	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
abs_level_gtx_flag[n][0]	ae(v)
remBinsPass1--	
if(abs_level_gtx_flag[n][0]) {	
par_level_flag[n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag[n][1]	ae(v)
remBinsPass1--	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	

AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag [n] + abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n] [1]	
if(sh_dep_quant_used_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
firstPosMode1 = n - 1	
}	
for(n = firstPosMode0; n > firstPosMode1; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n] [0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n] [1]	
if(abs_level_gtx_flag[n][1])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
}	
for(n = firstPosMode1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n] [0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n] [1]	
if(sb_coded_flag[xS][yS])	
dec abs_level[n]	ae(v)
if(AbsLevel[xC][yC] > 0) {	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
if(sh_dep_quant_used_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
if(sh_dep_quant_used_flag !sh_sign_data_hiding_used_flag)	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	

for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if((AbsLevel[xC][yC] > 0) && (!signHidden (n != firstSigScanPosSb)))	
coeff sign flag[n]	ae(v)
}	
if(sh dep quant used flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(AbsLevel[xC][yC] > 0)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff sign flag[n])	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	

if(AbsLevel[xC][yC] > 0) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff sign flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)	
)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
}	

[Bảng 3]

	Descriptor
residual ts coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3)	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
inferSbCbf = 1	
RemCpbs = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
for(i = 0; i <= lastSubBlock; i++) {	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
if(i != lastSubBlock !inferSbCbf)	
sb coded flag [xS][yS]	ae(v)
if(sb coded flag[xS][yS] && i < lastSubBlock)	
inferSbCbf = 0	
/* First scan pass */	
inferSbSigCoeffFlag = 1	
lastScanPosPass1 = -1	
for(n = 0; n <= numSbCoeff - 1 && RemCpbs >= 4; n++) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
lastScanPosPass1 = n	

if(sb_coded_flag[xS][yS] && (n != numSbCoeff - 1 !inferSbSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
RemCbs--	
if(sig_coeff_flag[xC][yC])	
inferSbSigCoeffFlag = 0	
}	
CoeffSignLevel[xC][yC] = 0	
if(sig_coeff_flag[xC][yC]) {	
coeff_sign_flag[n]	ae(v)
RemCbs--	
CoeffSignLevel[xC][yC] = (coeff_sign_flag[n] > 0 ? -1 : 1)	
abs_level_gtx_flag[n][0]	ae(v)
RemCbs--	
if(abs_level_gtx_flag[n][0]) {	
par_level_flag[n]	ae(v)
RemCbs--	
}	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag [n][0]	
}	
/* Greater than X scan pass (numGtXFlags=5) */	
lastScanPosPass2 = -1	
for(n = 0; n <= numSbCoeff - 1 && RemCbs >= 4; n++) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n] [0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n] [1]	
AbsLevelPass2[xC][yC] = AbsLevelPass1[xC][yC]	
for(j = 1; j < 5; j++) {	
if(abs_level_gtx_flag[n][j - 1]) {	
abs_level_gtx_flag[n][j]	ae(v)
RemCbs--	
}	
AbsLevelPass2[xC][yC] += 2 * abs_level_gtx_flag[n][j]	
}	
lastScanPosPass2 = n	
}	

/* remainder scan pass */	
for(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
if((n <= lastScanPosPass2 && AbsLevelPass2[xC][yC] >= 10) (n > lastScanPosPass2 && n <= lastScanPosPass1 && AbsLevelPass1[xC][yC] >= 2) (n > lastScanPosPass1 && sb_coded_flag[xS][yS]))	
abs_remainder[n]	ae(v)
if(n <= lastScanPosPass2)	
AbsLevel[xC][yC] = AbsLevelPass2[xC][yC] + 2 * abs_remainder	
else if(n <= lastScanPosPass1)	
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder	
else { /* bypass */	
AbsLevel[xC][yC] = abs_remainder[n]	
if(abs_remainder[n])	
coeff_sign_flag[n]	ae(v)
}	
if(BdpcmFlag[x0][y0][cIdx] == 0 && n <= lastScanPosPass1) {	
absLeftCoeff = xC > 0 ? AbsLevel[xC - 1][yC] : 0	
absAboveCoeff = yC > 0 ? AbsLevel[xC][yC - 1] : 0	
predCoeff = Max(absLeftCoeff, absAboveCoeff)	
if(AbsLevel[xC][yC] == 1 && predCoeff > 0)	
AbsLevel[xC][yC] = predCoeff	
else if(AbsLevel[xC][yC] > 0 && AbsLevel[xC][yC] <= predCo	
eff)	
AbsLevel[xC][yC] --	
}	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (1 - 2 * coeff_sign_flag	
[n]) * AbsLevel[xC][yC]	
}	
}	
}	

Theo phương án này, như được thể hiện trong Bảng 1, việc lập mã phần dư có thể được chia theo giá trị của phần tử cú pháp transform_skip_flag của cờ bỏ qua việc biến đổi. Tức là, phần tử cú pháp khác nhau có thể được sử dụng cho việc lập mã phần dư dựa trên giá trị của cờ bỏ qua việc biến đổi (dựa trên liệu việc biến đổi được bỏ qua hay không). Việc lập mã phần dư được sử dụng khi việc bỏ qua biến đổi không được áp dụng (tức là, khi việc biến đổi được áp dụng) có thể được gọi là lập mã phần dư thông thường (regular residual coding, RRC), và việc lập mã phần dư được sử dụng khi việc bỏ qua biến đổi được áp dụng (tức là, khi việc biến đổi không được áp dụng) có thể được gọi là

lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC). Ngoài ra, việc lập mã phần dư thông thường có thể được gọi là việc lập mã phần dư chung. Ngoài ra, việc lập mã phần dư thông thường có thể được gọi là cấu trúc cú pháp lập mã phần dư thông thường, và việc bỏ qua biến đổi việc lập mã phần dư có thể được gọi là cấu trúc cú pháp lập mã phần dư bỏ qua biến đổi. Bảng 2 nêu trên có thể thể hiện phần tử cú pháp của lập mã phần dư khi giá trị của `transform_skip_flag` là 0, tức là, khi việc biến đổi được áp dụng, và Bảng 3 nêu trên có thể thể hiện phần tử cú pháp của việc lập mã phần dư khi giá trị của `transform_skip_flag` là 1, tức là, khi việc biến đổi không được áp dụng.

Cụ thể, ví dụ, cờ bỏ qua việc biến đổi chỉ báo liệu để bỏ qua việc biến đổi khối biến đổi có thể được phân tích cú pháp, và liệu cờ bỏ qua việc biến đổi là 1 có thể được xác định. Nếu giá trị của cờ bỏ qua biến đổi là 0, như được thể hiện trong Bảng 2, các phần tử cú pháp `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, `last_sig_coeff_y_suffix`, `sb_coded_flag`, `sig_coeff_flag`, `abs_level_gtx_flag`, `par_level_flag`, `abs_remainder`, `coeff_sign_flag` và/hoặc `dec_abs_level` cho hệ số phần dư của khối biến đổi có thể được phân tích cú pháp, và hệ số phần dư có thể được dẫn xuất dựa trên các phần tử cú pháp. Trong trường hợp này, các phần tử cú pháp có thể được phân tích cú pháp tuần tự, và thứ tự phân tích cú pháp có thể được thay đổi. Ngoài ra, `abs_level_gtx_flag` có thể thể hiện `abs_level_gt1_flag` và/hoặc `abs_level_gt3_flag`. Ví dụ, `abs_level_gtx_flag[n][0]` có thể là ví dụ của cờ mức hệ số biến đổi thứ nhất (`abs_level_gt1_flag`), và `abs_level_gtx_flag[n][1]` có thể là ví dụ của cờ mức hệ số biến đổi thứ hai (`abs_level_gt3_flag`).

Tham khảo Bảng 2 ở trên, `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, `last_sig_coeff_y_suffix`, `sb_coded_flag`, `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gt3_flag`, `abs_remainder`, `coeff_sign_flag`, và/hoặc `dec_abs_level` có thể được mã hóa/được giải mã. Trong khi đó, `sb_coded_flag` có thể được biểu diễn là `coded_sub_block_flag`.

Trong một phương án, thiết bị mã hóa có thể mã hóa (x, y) thông tin vị trí của hệ số biến đổi khác không cuối cùng trong khối biến đổi dựa trên các phần tử cú pháp `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, và `last_sig_coeff_y_suffix`. Cụ thể hơn, `last_sig_coeff_x_prefix` biểu diễn tiền tố của vị trí

cột của hệ số đáng kể cuối cùng theo thứ tự quét trong khối biến đổi, `last_sig_coeff_y_prefix` biểu diễn tiền tố của vị trí hàng của hệ số đáng kể cuối cùng theo thứ tự quét trong khối biến đổi, `last_sig_coeff_x_suffix` biểu diễn hậu tố của vị trí cột của hệ số đáng kể cuối cùng theo thứ tự quét trong khối biến đổi, và `last_sig_coeff_y_suffix` biểu diễn hậu tố của vị trí hàng của hệ số đáng kể cuối cùng theo thứ tự quét trong khối biến đổi. Ở đây, hệ số có nghĩa có thể biểu diễn hệ số khác không (0). Ngoài ra, thứ tự quét có thể là thứ tự quét đường chéo bên phải. Theo cách khác, thứ tự quét có thể là thứ tự quét ngang hoặc thứ tự quét thẳng. Thứ tự quét có thể được xác định dựa trên liệu việc nội/liên dự đoán có được áp dụng cho khối mục tiêu (CB hoặc CB gồm TB), và/hoặc chế độ nội/liên dự đoán cụ thể hay không.

Ở dưới đây, thiết bị mã hóa có thể chia khối biến đổi thành các khối con 4x4, và sau đó chỉ báo liệu có hệ số khác không trong khối con hiện tại sử dụng phần tử cú pháp 1-bit `coded_sub_block_flag` cho mỗi khối con 4x4 hay không.

Nếu giá trị của `coded_sub_block_flag` là 0, thì không có nhiều thông tin hơn cần được truyền, và do đó, thiết bị mã hóa có thể chấm dứt quy trình mã hóa trên khối con hiện tại. Ngược lại, nếu giá trị của `coded_sub_block_flag` là 1, thì thiết bị mã hóa có thể tiếp tục thực hiện quy trình mã hóa trên `sig_coeff_flag`. Bởi vì khối con bao gồm hệ số khác không cuối cùng không đòi hỏi việc mã hóa cho `coded_sub_block_flag` và khối con bao gồm thông tin DC của khối biến đổi có xác suất cao của việc chứa hệ số khác không, thì `coded_sub_block_flag` có thể không được lập mã và giá trị của nó có thể được giải định là 1.

Nếu giá trị của `coded_sub_block_flag` là 1 và do đó xác định được rằng hệ số khác không tồn tại trong khối con hiện tại, thì thiết bị mã hóa có thể mã hóa `sig_coeff_flag` có giá trị nhị phân theo thứ tự quét ngược. Thiết bị mã hóa có thể mã hóa phần tử cú pháp 1-bit `sig_coeff_flag` cho mỗi hệ số biến đổi theo thứ tự quét. Nếu trị số của hệ số biến đổi tại vị trí quét hiện tại không phải là 0, thì giá trị của `sig_coeff_flag` có thể là 1. Ở đây, trong trường hợp của khối con bao gồm hệ số khác không cuối cùng, `sig_coeff_flag` không cần được mã hóa cho hệ số khác không cuối cùng, vì vậy quy trình lập mã cho khối con có thể được lược bỏ. Việc lập mã thông tin theo mức có thể được thực hiện chỉ khi `sig_coeff_flag` là 1, và bốn phần tử cú pháp có thể được sử dụng trong quy trình mã hóa thông tin theo mức. Cụ thể hơn, mỗi `sig_coeff_flag[xC][yC]` có thể

chỉ báo liệu mức (giá trị) của hệ số biến đổi tương ứng tại mỗi vị trí hệ số biến đổi (xC, yC) trong TB hiện tại là khác không hay không. Trong một phương án, sig_coeff_flag có thể tương ứng với ví dụ về phân tử cú pháp của cờ hệ số có nghĩa chỉ báo liệu hệ số biến đổi được lượng tử hóa là hệ số có nghĩa khác không hay không.

Việc duy trì giá trị mức sau khi mã hóa cho sig_coeff_flag có thể được dẫn xuất như được thể hiện trong phương trình sau đây. Tức là, phân tử cú pháp remAbsLevel chỉ báo giá trị mức cần được mã hóa có thể được dẫn xuất từ phương trình sau đây.

[Phương trình 1]

$$\text{remAbsLevel} = |\text{coeff}| - 1$$

Ở đây, coeff nghĩa là giá trị hệ số biến đổi thực sự.

Ngoài ra, abs_level_gt1_flag có thể chỉ báo liệu remAbsLevel' của vị trí quét tương ứng (n) là lớn hơn 1 hay không. Ví dụ, khi giá trị của abs_level_gt1_flag là 0, thì giá trị tuyệt đối của hệ số biến đổi của vị trí tương ứng có thể là 1. Ngoài ra, khi giá trị của abs_level_gt1_flag là 1, remAbsLevel chỉ ra giá trị mức cần được mã hóa sau có thể được cập nhật như được thể hiện trong phương trình sau đây.

[Phương trình 2]

$$\text{remAbsLevel} = \text{remAbsLevel} - 1$$

Ngoài ra, giá trị hệ số ít nghĩa nhất (Least significant coefficient, LSB) của remAbsLevel được mô tả trong Phương trình 2 được mô tả ở trên có thể được mã hóa như trong Phương trình 3 dưới đây thông qua par_level_flag.

[Phương trình 3]

$$\text{par_level_flag} = |\text{coeff}| \& 1$$

Ở đây, par_level_flag[n] có thể chỉ báo tính chẵn lẻ của (giá trị) mức hệ số biến đổi ở vị trí quét n.

Giá trị mức hệ số biến đổi remAbsLevel mà là để được mã hóa sau khi thực hiện việc mã hóa par_level_flag có thể được cập nhật như được thể hiện dưới đây trong phương trình sau đây.

[Phương trình 4]

remAbsLevel = remAbsLevel >> 1

abs_level_gt3_flag có thể chỉ báo liệu remAbsLevel' của vị trí quét tương ứng (n) là lớn hơn 3 hay không. Việc mã hóa cho abs_remainder có thể được thực hiện chỉ trong trường hợp trong đó rem_abs_gt3_flag là bằng với 1. Mỗi quan hệ giữa giá trị hệ số biến đổi thực sự coeff và mỗi phần tử cú pháp có thể được thể hiện dưới đây trong phương trình sau đây.

[Phương trình 5]

$$|coeff| = sig_coeff_flag + abs_level_gt1_flag + par_level_flag + 2 * (abs_level_gt3_flag + abs_remainder)$$

Ngoài ra, bảng sau đây chỉ báo các ví dụ liên quan đến Phương trình 5 được mô tả ở trên.

[Bảng 4]

coeff[n]	sig_coeff_flag[n]	abs_level_gtx_flag[n][0]	par_level_flag[n]	abs_level_gtx_flag[n][1]	abs_remainder[n]
0	0				
1	1	0			
2	1	1	0	0	
3	1	1	1	0	
4	1	1	0	1	0
5	1	1	1	1	0
6	1	1	0	1	1
7	1	1	1	1	1
8	1	1	0	1	2
9	1	1	1	1	2
10	1	1	0	1	3
11	1	1	1	1	3
...	...	...	...		

Ở đây, |coeff| chỉ báo (giá trị) mức hệ số biến đổi và có thể cũng được chỉ báo như AbsLevel cho hệ số biến đổi. Ngoài ra, dấu hiệu của mỗi hệ số có thể được mã hóa bằng cách sử dụng coeff_sign_flag, vốn là ký hiệu 1-bit.

Ngoài ra, nếu giá trị của cờ bỏ qua biến đổi là 1, như được thể hiện trong Bảng 3, các phần tử cú pháp sb_coded_flag, sig_coeff_flag, coeff_sign_flag, abs_level_gtx_flag, par_level_flag và/hoặc abs_remainder cho hệ số phần dư của khối biến đổi có thể được phân tích cú pháp, và hệ số phần dư có thể được dẫn xuất dựa trên các phần tử cú pháp. Trong trường hợp này, các phần tử cú pháp có thể được phân tích cú pháp tuần tự, và thứ tự phân tích cú pháp có thể được thay đổi. Ngoài ra, abs_level_gtx_flag có thể biểu diễn abs_level_gt1_flag, abs_level_gt3_flag, abs_level_gt5_flag, abs_level_gt7_flag, và/hoặc abs_level_gt9_flag. Ví dụ, abs_level_gtx_flag[n][j] có thể là cờ chỉ ra xem giá

trị tuyệt đối hoặc mức (giá trị) của hệ số biến đổi tại vị trí quét n có lớn hơn $(j < 1) + 1$ hay không. Điều kiện $(j < 1) + 1$ có thể được thay thế theo cách tùy chọn với ngưỡng cụ thể như ngưỡng thứ nhất, ngưỡng thứ hai, hoặc tương tự.

Trong khi đó, CABAC cung cấp hiệu suất cao, nhưng có nhược điểm là hiệu suất thông lượng kém. Điều này bị gây ra bởi bộ máy lập mã thông thường của CABAC. Việc mã hóa thông thường (tức là, việc lập mã qua bộ máy lập mã thông thường của CABAC) thể hiện sự phụ thuộc dữ liệu cao bởi vì nó sử dụng trạng thái và phạm vi xác suất được cập nhật thông qua việc lập mã bin trước, và có thể mất nhiều thời gian để đọc khoảng xác suất và xác định trạng thái hiện tại. Vấn đề về lưu lượng của CABAC có thể được giải quyết bằng cách giới hạn số lượng bin được lập mã ngưỡng cảnh. Ví dụ, như được thể hiện trong Bảng 2 được mô tả ở trên, tổng của các ngăn được sử dụng để trình bày `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, và `abs_level_gt3_flag` có thể được giới hạn ở số lượng của các ngăn phụ thuộc vào kích thước của khối tương ứng. Ngoài ra, ví dụ, như được thể hiện trong Bảng 3 được mô tả ở trên, tổng của các ngăn được sử dụng để trình bày `sig_coeff_flag`, `coeff_sign_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gt3_flag`, `abs_level_gt5_flag`, `abs_level_gt7_flag`, `abs_level_gt9_flag` có thể được giới hạn ở số lượng của các ngăn phụ thuộc vào kích thước của khối tương ứng. Ví dụ, nếu khối tương ứng khối có kích thước 4×4 , tổng của các ngăn cho `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gt3_flag` hoặc `sig_coeff_flag`, `coeff_sign_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gt3_flag`, `abs_level_gt5_flag`, `abs_level_gt7_flag`, `abs_level_gt9_flag` có thể được giới hạn ở 32 (hoặc ví dụ 28), và nếu khối tương ứng khối có kích thước 2×2 , tổng của các ngăn cho `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gt3_flag` có thể được giới hạn ở 8 (hoặc ví dụ 7). Số lượng bin được giới hạn có thể được biểu diễn bằng `remBinsPass1` hoặc `RemCcb`s. Hoặc, ví dụ, đối với thông lượng CABAC cao hơn, số lượng của các bin được lập mã ngưỡng cảnh có thể được giới hạn cho khối (CB hoặc TB) gồm CG mục tiêu lập mã. Nói cách khác, số lượng bin được lập mã ngưỡng cảnh có thể được giới hạn trong các đơn vị là các khối (CB hoặc TB). Ví dụ, khi kích thước của khối hiện tại là 16×16 , thì số lượng bin được lập mã ngưỡng cảnh cho khối hiện tại có thể được giới hạn đến 1,75 lần số lượng điểm ảnh của khối hiện tại, tức là, 448, bất kể CG hiện tại.

Trong trường hợp này, nếu tất cả các bin được lập mã ngữ cảnh mà có số lượng được giới hạn được sử dụng khi phân tử ngữ cảnh được lập mã, thì thiết bị mã hóa có thể nhị phân hóa các hệ số còn lại thông qua phương pháp nhị phân hóa hệ số như được mô tả dưới đây, thay vì sử dụng việc lập mã ngữ cảnh, và có thể thực hiện việc mã hóa vòng. Nói cách khác, ví dụ, nếu số lượng của các ngăn được lập mã ngữ cảnh mà được lập mã cho 4x4 CG là 32 (hoặc ví dụ 28), hoặc nếu số lượng của các ngăn được lập mã ngữ cảnh mà được lập mã cho 2x2 CG là 8 (hoặc ví dụ 7), sig_coeff_flag, abs_level_gt1_flag, par_level_flag, abs_level_gt3_flag mà được lập mã với ngăn được lập mã ngữ cảnh có thể không còn được lập mã, và có thể được lập mã một cách trực tiếp thành dec_abs_level. Hoặc, ví dụ, khi số lượng của các ngăn được lập mã ngữ cảnh được lập mã cho khối 4x4 bằng 1,75 lần số lượng của các điểm ảnh của toàn bộ khối, nghĩa là, khi được giới hạn ở 28, sig_coeff_flag, abs_level_gt1_flag, par_level_flag, và abs_level_gt3_flag được lập mã dưới dạng các ngăn được lập mã ngữ cảnh có thể không được lập mã nữa, và có thể được lập mã trực tiếp dưới dạng dec_abs_level như được thể hiện trong Bảng 5 dưới đây.

[Bảng 5]

coeff n	dec_abs_level n
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
...	...

Giá trị |coeff| có thể được dẫn xuất dựa trên dec_abs_level. Trong trường hợp này, giá trị hệ số biến đổi, tức là, |coeff|, có thể được dẫn xuất như được thể hiện trong phương trình sau đây.

[Phương trình 6]

$$|coeff| = dec_abs_level$$

Bên cạnh đó, `coeff_sign_flag` có thể chỉ ra dấu của mức hệ số biến đổi ở vị trí quét tương ứng n . Nghĩa là, `coeff_sign_flag` có thể chỉ ra dấu của hệ số biến đổi ở vị trí quét tương ứng n .

Fig.5 thể hiện ví dụ về các hệ số biến đổi trong khối 4×4 .

Khối 4×4 của Fig.5 biểu diễn ví dụ về các hệ số được lượng tử hóa. Khối của Fig.5 có thể là khối biến đổi 4×4 , hoặc khối con 4×4 của khối biến đổi 8×8 , 16×16 , 32×32 , hoặc 64×64 . Khối 4×4 của Fig.5 có thể biểu diễn khối độ chói hoặc khối sắc độ.

Trong lúc đó, như được mô tả ở trên, khi tín hiệu đầu vào không phải là giá trị nhị phân mà là phần tử cú pháp, thì thiết bị mã hóa có thể biến đổi tín hiệu đầu vào thành giá trị nhị phân bằng cách nhị phân hóa giá trị của tín hiệu đầu vào. Ngoài ra, thiết bị giải mã có thể giải mã phần tử cú pháp để suy ra giá trị được nhị phân hóa (tức là, bin được nhị phân hóa) của phần tử cú pháp, và có thể giải nhị phân hóa giá trị được nhị phân hóa để suy ra giá trị của phần tử cú pháp. Quy trình nhị phân hóa có thể được thực hiện như quy trình nhị phân hóa rice bị cắt cụt (Truncated Rice, TR), quy trình nhị phân hóa Exp-Golomb thứ tự thứ k (Egk), Exp-Golomb thứ tự thứ k giới hạn (EGk giới hạn), quy trình nhị phân hóa độ dài cố định (Fixed-Length, FL), hoặc tương tự. Ngoài ra, quy trình giải nhị phân hóa có thể biểu diễn quy trình được thực hiện dựa trên quy trình nhị phân hóa TR, quy trình nhị phân hóa EGk, hoặc quy trình nhị phân hóa FL để dẫn xuất giá trị của phần tử cú pháp.

Ví dụ, quy trình nhị phân hóa TR có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa TR có thể là `cMax` và `cRiceParam` cho phần tử cú pháp và yêu cầu cho việc nhị phân hóa TR. Ngoài ra, đầu ra của quy trình nhị phân hóa TR có thể là việc nhị phân hóa TR cho `symbolVal` mà là giá trị tương ứng với chuỗi bin.

Cụ thể, ví dụ, khi có mặt chuỗi bin hậu tố cho phần tử cú pháp, thì chuỗi bin TR cho phần tử cú pháp có thể là sự ghép chuỗi bin tiền tố và chuỗi bin hậu tố; và khi vắng mặt chuỗi bin hậu tố, thì chuỗi bin TR cho phần tử cú pháp có thể là chuỗi bin tiền tố. Ví dụ, chuỗi bin tiền tố có thể được suy ra như được mô tả dưới đây.

Giá trị tiền tố của `symbolVal` cho phần tử cú pháp có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 7]

$$\text{prefixVal} = \text{symbolVal} \gg \text{cRiceParam}$$

Ở đây, prefixVal có thể biểu thị giá trị tiền tố của symbolVal . Tiền tố (tức là, chuỗi bin tiền tố) của chuỗi bin TR của phần tử cú pháp có thể được dẫn xuất như được mô tả dưới đây.

Ví dụ, nếu prefixVal là thấp hơn $\text{cMax} \gg \text{cRiceParam}$, thì chuỗi bin tiền tố có thể là chuỗi bit có chiều dài là $\text{prefixVal}+1$ được lập chỉ số bằng binIdx . Tức là, nếu prefixVal là thấp hơn $\text{cMax} \gg \text{cRiceParam}$, thì chuỗi bin tiền tố có thể là chuỗi bit mà số lượng bit là $\text{prefixVal}+1$, được chỉ báo bằng binIdx . bin cho binIdx nhỏ hơn prefixVal có thể là bằng 1. Ngoài ra, bin cho binIdx giống như prefixVal có thể là bằng 0.

Ví dụ, chuỗi bin được suy ra thông qua việc nhị phân hóa đơn phân cho prefixVal có thể được thể hiện trong bảng sau đây.

[Bảng 6]

prefixVal	Bin string					
0	0					
1	1	0				
2	1	1	0			
3	1	1	1	0		
4	1	1	1	1	0	
5	1	1	1	1	1	0
...						
binIdx	0	1	2	3	4	5

Trong khi đó, nếu prefixVal không nhỏ hơn $\text{cMax} \gg \text{cRiceParam}$, thì chuỗi bin tiền tố có thể là chuỗi bit trong đó chiều dài là $\text{cMax} \gg \text{cRiceParam}$ và tất cả các bit là 1.

Ngoài ra, nếu cMax là lớn hơn symbolVal và nếu cRiceParam là lớn hơn 0, thì chuỗi bin hậu tố bin của chuỗi bin TR có thể là có mặt. Ví dụ, chuỗi bin hậu tố có thể được suy ra như được mô tả dưới đây.

Giá trị hậu tố của symbolVal cho phần tử cú pháp có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 8]

$$\text{suffixVal} = \text{symbolVal} - ((\text{prefixVal}) \ll \text{cRiceParam})$$

Ở đây, suffixVal có thể biểu thị giá trị hậu tố của symbolVal .

Hậu tố của chuỗi bin TR (tức là, chuỗi bin hậu tố) có thể được suy ra dựa trên quy trình nhị phân hóa FL cho suffixVal mà có giá trị cMax là $(1 \ll cRiceParam) - 1$.

Trong khi đó, nếu giá trị của tham số đầu vào, tức là, cRiceParam, là 0, thì quy trình nhị phân hóa TR có thể là nhị phân hóa đơn phân được cắt cụt chính xác, và có thể luôn sử dụng giá trị cMax giống như giá trị lớn nhất có thể của phần tử cú pháp cần được giải mã.

Ngoài ra, ví dụ, quy trình nhị phân hóa EGk có thể được thực hiện như sau. Phần tử cú pháp được lập mã với ue(v) có thể là phần tử cú pháp được trải qua lập mã Exp-Golomb.

Ví dụ, quy trình nhị phân hóa Exp-Golomb thứ tự thứ 0 (0-th order Exp-Golomb, EG0) có thể được thực hiện như sau.

Quy trình phân tích cú pháp cho phần tử cú pháp có thể bắt đầu bằng việc đọc bit bao gồm bit khác không thứ nhất tại vị trí hiện tại của luồng bit và tính số lượng bit dẫn đầu bằng 0. Quy trình có thể được biểu diễn như được thể hiện trong bảng sau đây.

[Bảng 7]

```
leadingZeroBits = -1
for( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 )
```

Ngoài ra, biến số 'codeNum' có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 9]

$$\text{codeNum} = 2^{\text{leadingZeroBits}} - 1 + \text{read_bits}(\text{leadingZeroBits})$$

Ở đây, giá trị được trả lại từ read_bits(leadingZeroBits), tức là, giá trị được chỉ báo bằng read_bits(leadingZeroBits), có thể được diễn dịch như sự biểu diễn nhị phân của số nguyên không dấu cho bit có nhiều nghĩa nhất được ghi đầu tiên.

Cấu trúc của mã Exp-Golomb trong đó chuỗi bit được chia thành bit “tiền tố” và bit “hậu tố” có thể được biểu diễn như được thể hiện trong bảng sau đây.

[Bảng 8]

Bit string form	Range of codeNum
1	0
0 1 x_0	1..2
0 0 1 $x_1 x_0$	3..6
0 0 0 1 $x_2 x_1 x_0$	7..14
0 0 0 0 1 $x_3 x_2 x_1 x_0$	15..30
0 0 0 0 0 1 $x_4 x_3 x_2 x_1 x_0$	31..62
...	...

Bit “tiền tố” có thể là bit được phân tích cú pháp như được mô tả ở trên để tính toán `leadingZeroBits` và có thể được biểu diễn bằng 0 hoặc 1 của chuỗi bit trong Bảng 8. Tức là, chuỗi bit được bọc lờ bằng 0 hoặc 1 trong Bảng 8 nêu trên có thể biểu diễn chuỗi bit tiền tố. Bit “hậu tố” có thể là bit được phân tích cú pháp khi tính toán `codeNum`, và có thể được biểu diễn bằng x_i trong Bảng 8 nêu trên. Tức là, chuỗi bit được bọc lờ như x_i trong Bảng 8 nêu trên có thể biểu diễn chuỗi bit hậu tố. Ở đây, i có thể là giá trị trong phạm vi của `LeadingZeroBits-1`. Ngoài ra, mỗi x_i có thể là bằng 0 hoặc 1.

Chuỗi bit được gán cho `codeNum` có thể được thể hiện trong bảng sau đây.

[Bảng 9]

Bit string	codeNum
1	0
0 1 0	1
0 1 1	2
0 0 1 0 0	3
0 0 1 0 1	4
0 0 1 1 0	5
0 0 1 1 1	6
0 0 0 1 0 0 0	7
0 0 0 1 0 0 1	8
0 0 0 1 0 1 0	9
...	...

Nếu bộ mô tả của phần tử cú pháp là `ue(v)`, tức là, nếu phần tử cú pháp được lập mã với `ue(v)`, thì giá trị của phần tử cú pháp có thể là bằng `codeNum`.

Ngoài ra, ví dụ, quy trình nhị phân hóa EGk có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa EGk có thể là yêu cầu cho việc nhị phân hóa EGk. Ngoài ra, đầu ra của quy trình nhị phân hóa EGk là nhị phân hóa EGk cho `symbolVal`, tức là, giá trị tương ứng với chuỗi bin.

Chuỗi bit của quy trình nhị phân hóa EGk cho `symbolVal` có thể được dẫn xuất như sau.

[Bảng 10]

```

absV = Abs( symbolVal )
stopLoop = 0
do
    if( absV >= ( 1 << k ) ) {
        put( 1 )
        absV = absV - ( 1 << k )
        k++
    } else {
        put( 0 )
        while( k-- )
            put( ( absV >> k ) & 1 )
        stopLoop = 1
    }
while( !stopLoop )

```

Tham chiếu đến Bảng 10 nêu trên, giá trị nhị phân X có thể được thêm vào cuối của chuỗi bin thông qua mỗi lần gọi put(X). Ở đây, X có thể là 0 hoặc 1.

Ngoài ra, ví dụ, quy trình nhị phân hóa EGk giới hạn có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa EGk giới hạn có thể là yêu cầu cho việc nhị phân hóa EGk giới hạn, tham số Rice riceParam, log2TransformRange như biến số biểu diễn lôgarit nhị phân có giá trị tối đa, và maxPreExtLen như biến số biểu diễn chiều rộng mở rộng tiền tố lớn nhất. Ngoài ra, đầu ra của quy trình nhị phân hóa EGk giới hạn có thể là việc nhị phân hóa EGk giới hạn cho symbolVal như giá trị tương ứng với chuỗi rỗng.

Chuỗi bit của quy trình nhị phân hóa EGk giới hạn cho symbolVal có thể được dẫn xuất như sau.

[Bảng 11]

```

codeValue = symbolVal >> riceParam
PrefixExtensionLength = 0
while( ( PrefixExtensionLength < maxPrefixExtensionLength ) &&
      ( codeValue > ( ( 2 << PrefixExtensionLength ) - 2 ) ) ) {
    PrefixExtensionLength++
    put( 1 )
}
if( PrefixExtensionLength == maxPrefixExtensionLength )
    escapeLength = log2TransformRange
else {
    escapeLength = PrefixExtensionLength + riceParam
    put( 0 )
}
symbolVal = symbolVal - ( ( ( 1 << PrefixExtensionLength ) - 1 ) << riceParam )
while( ( escapeLength-- ) > 0 )
    put( ( symbolVal >> escapeLength ) & 1 )

```

Ngoài ra, ví dụ, quy trình nhị phân hóa FL có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa FL có thể là yêu cầu cho việc nhị phân hóa FL và cMax cho phần tử cú pháp. Ngoài ra, đầu ra của quy trình nhị phân hóa FL có thể là việc nhị phân hóa FL cho symbolVal như giá trị tương ứng với chuỗi bin.

Việc nhị phân hóa FL có thể được tạo cấu hình bằng cách sử dụng chuỗi bit mà có số lượng bit có chiều dài cố định là symbolVal. Ở đây, bit có chiều dài cố định có thể là chuỗi bit số nguyên không dấu. Tức là, chuỗi bit cho symbolVal như giá trị ký hiệu có thể được suy ra thông qua việc nhị phân hóa FL, và chiều dài bit (tức là, số lượng bit) của chuỗi bit có thể là chiều dài cố định.

Ví dụ, chiều dài cố định có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 10]

$$\text{fixedLength} = \text{Ceil}(\text{Log2}(\text{cMax} + 1))$$

Việc lập chỉ số các bin cho việc nhị phân hóa FL có thể là phương pháp sử dụng giá trị mà tăng theo thứ tự từ bit có nhiều nghĩa nhất đến bit có ít nghĩa nhất. Ví dụ, chỉ số bin liên quan đến bit có nghĩa nhất có thể là binIdx = 0.

Trong khi đó, ví dụ, quy trình nhị phân hóa cho phần tử cú pháp abs_remainder trong thông tin phần dư có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa cho abs_remainder có thể là yêu cầu cho việc nhị phân hóa phần tử cú pháp abs_remainder[n], thành phần màu cIdx, và vị trí độ chói

(x_0, y_0) . Vị trí độ chói (x_0, y_0) có thể thể hiện mẫu phía trên-bên trái của khối biến đổi độ chói hiện tại dựa trên mẫu độ chói phía trên-bên trái của hình ảnh.

Đầu ra của quy trình nhị phân hóa cho `abs_remainder` có thể là việc nhị phân hóa của `abs_remainder` (tức là, chuỗi bin được nhị phân hóa của `abs_remainder`). Các chuỗi bin sẵn có cho `abs_remainder` có thể được suy ra thông qua quy trình nhị phân hóa.

Thông số `rice cRiceParam` cho `abs_remainder[n]` có thể được dẫn xuất thông qua quy trình dẫn xuất thông số `rice` được thực hiện bằng việc nhập thành phần màu sắc `cIdx` và vị trí độ chói (x_0, y_0) , vị trí quét hệ số hiện tại (xC, yC) , `log2TbWidth`, mà là logarit nhị phân của độ rộng của khối biến đổi, và `log2TbHeight`, mà là logarit nhị phân của độ cao của khối biến đổi. Phần mô tả chi tiết của quy trình dẫn xuất thông số `rice` sẽ được mô tả sau.

Ngoài ra, ví dụ, `cMax for abs_remainder[n]` cần được lập mã hiện tại có thể được dẫn xuất dựa trên tham số `rice cRiceParam`. `cMax` có thể được dẫn xuất như được thể hiện trong phương trình sau đây.

[Phương trình 11]

$$cMax = 6 \ll cRiceParam$$

Trong khi đó, việc nhị phân hóa cho `abs_remainder`, tức là, chuỗi bin cho `abs_remainder`, có thể là sự ghép nối của chuỗi bin tiền tố và chuỗi bin hậu tố khi có mặt chuỗi bin hậu tố. Ngoài ra, khi không có mặt chuỗi bin hậu tố, chuỗi bin cho `abs_remainder` có thể là chuỗi bin tiền tố.

Ví dụ, chuỗi bin tiền tố có thể được dẫn xuất như được mô tả dưới đây.

Giá trị tiền tố `prefixVal` của `abs_remainder[n]` có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 12]

$$prefixVal = \text{Min}(cMax, abs_remainder[n])$$

Tiền tố của chuỗi bin (tức là, chuỗi bin tiền tố) của `abs_remainder[n]` có thể được suy ra thông qua quy trình nhị phân hóa TR cho `prefixVal`, trong đó `cMax` và `cRiceParam` được sử dụng như đầu vào.

Nếu chuỗi bin tiền tố là giống với chuỗi bit trong đó tất cả các bit là 1 và chiều dài bit là 6, thì chuỗi bin hậu tố của chuỗi bin của `abs_remainder[n]` có thể tồn tại, và có thể được dẫn xuất như được mô tả dưới đây.

Quy trình dẫn xuất tham số rice cho `dec_abs_level[n]` có thể là như sau.

Đầu vào của quy trình dẫn xuất tham số rice có thể là chỉ số thành phần màu `cIdx`, vị trí độ chói (x_0 , y_0), vị trí quét hệ số hiện tại (x_C , y_C), $\log_2 \text{TbWidth}$ như lôgarit nhị phân của chiều rộng của khối biến đổi, và $\log_2 \text{TbHeight}$ như lôgarit nhị phân của chiều cao của khối biến đổi. Vị trí độ chói (x_0 , y_0) có thể chỉ báo mẫu phía trên-bên trái của khối biến đổi độ chói hiện tại dựa trên mẫu độ chói phía trên-bên trái của hình ảnh. Ngoài ra, đầu ra của quy trình suy ra tham số rice có thể là tham số rice `cRiceParam`.

Ví dụ, biến `locSumAbs` có thể được dẫn xuất một cách tương tự với mã giả được bộc lộ trong bảng sau đây, dựa trên mảng `AbsLevel[x][y]` cho khối biến đổi có chỉ số thành phần đã cho `cIdx` và vị trí độ chói trên cùng bên trái (x_0 , y_0).

[Bảng 12]

```

locSumAbs = 0
if(  $x_C < (1 \ll \log_2 \text{TbWidth}) - 1$  ) {
    locSumAbs += AbsLevel[  $x_C + 1$  ][  $y_C$  ]
    if(  $x_C < (1 \ll \log_2 \text{TbWidth}) - 2$  )
        locSumAbs += AbsLevel[  $x_C + 2$  ][  $y_C$  ]
    if(  $y_C < (1 \ll \log_2 \text{TbHeight}) - 1$  )
        locSumAbs += AbsLevel[  $x_C + 1$  ][  $y_C + 1$  ]
}
if(  $y_C < (1 \ll \log_2 \text{TbHeight}) - 1$  ) {
    locSumAbs += AbsLevel[  $x_C$  ][  $y_C + 1$  ]
    if(  $y_C < (1 \ll \log_2 \text{TbHeight}) - 2$  )
        locSumAbs += AbsLevel[  $x_C$  ][  $y_C + 2$  ]
}
locSumAbs = Clip3( 0, 31, locSumAbs - baseLevel * 5 )

```

(1532)

Sau đó, dựa trên biến đã cho `locSumAbs`, thông số rice `cRiceParam` có thể được dẫn xuất như được thể hiện trong bảng dưới đây.

[Bảng 13]

locSumAbs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cRiceParam	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2
locSumAbs	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cRiceParam	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3

Ngoài ra, ví dụ, trong quy trình dẫn xuất thông số rice cho `abs_remainder[n]`, `baseLevel` có thể được thiết đặt là 4.

Mặt khác, ví dụ, thông số rice `cRiceParam` có thể được xác định dựa trên việc liệu việc bỏ qua biến đổi có được áp dụng cho khối hiện tại hay không. Tức là, nếu việc biến đổi không được áp dụng cho TB hiện tại bao gồm CG hiện tại, nói cách khác, nếu việc bỏ qua biến đổi được áp dụng cho TB hiện tại bao gồm CG hiện tại, thì tham số rice `cRiceParam` có thể được dẫn xuất là 1.

Ngoài ra, giá trị hậu tố `suffixVal` của `abs_remainder` có thể được dẫn xuất như được thể hiện trong phương trình sau đây.

[Phương trình 13]

$$\text{suffixVal} = \text{abs_remainder}[n] - \text{cMax}$$

Chuỗi bin hậu tố của chuỗi bin của `abs_remainder` có thể được dẫn xuất thông qua quy trình nhị phân hóa EGk giới hạn cho `suffixVal` trong đó k được thiết đặt là `cRiceParam+1`, `riceParam` được thiết đặt là `cRiceParam`, và `log2TransformRange` được thiết đặt là 15, và `maxPreExtLen` được thiết đặt là 11.

Trong khi đó, ví dụ, quy trình nhị phân hóa cho phần tử cú pháp `dec_abs_level` trong thông tin dư có thể được thực hiện như sau.

Đầu vào của quy trình nhị phân hóa cho `dec_abs_level` có thể là yêu cầu cho việc nhị phân hóa phần tử cú pháp `dec_abs_level[n]`, thành phần màu `cIdx`, vị trí độ sáng (`x0`, `y0`), vị trí quét hệ số hiện tại (`xC`, `yC`), `log2TbWidth` như lôgarit nhị phân của chiều rộng của khối biến đổi, và `log2TbHeight` như lôgarit nhị phân của chiều cao của khối biến đổi. Vị trí độ chói (`x0`, `y0`) có thể chỉ báo mẫu phía trên-bên trái của khối biến đổi độ chói hiện tại dựa trên mẫu độ chói phía trên-bên trái của hình ảnh.

Đầu ra của quy trình nhị phân hóa cho `dec_abs_level` có thể là việc nhị phân hóa `dec_abs_level` (tức là, chuỗi bin được nhị phân hóa của `dec_abs_level`). Các chuỗi bin sẵn có cho `dec_abs_level` có thể được suy ra thông qua quy trình nhị phân hóa.

Tham số rice `cRiceParam` cho `dec_abs_level[n]` có thể được suy ra thông qua quy trình suy ra tham số rice được thực hiện có đầu vào là thành phần màu `cIdx`, vị trí độ chói (`x0`, `y0`), vị trí quét hệ số hiện tại (`xC`, `yC`), `log2TbWidth` như lôgarit nhị phân của

chiều rộng của khối biến đổi, và $\log_2 \text{TbHeight}$ như lôgarit nhị phân của chiều cao của khối biến đổi. Quy trình dẫn xuất tham số rice sẽ được mô tả dưới đây một cách chi tiết.

Ngoài ra, ví dụ, cMax cho $\text{dec_abs_level}[n]$ có thể được suy ra dựa trên tham số rice cRiceParam . cMax có thể được suy ra như được thể hiện trong bảng sau đây.

[Phương trình 14]

$$\text{cMax} = 6 \ll \text{cRiceParam}$$

Trong khi đó, việc nhị phân hóa cho $\text{dec_abs_level}[n]$, tức là, chuỗi bin cho $\text{dec_abs_level}[n]$, có thể là sự ghép nối chuỗi bin tiền tố và chuỗi bin hậu tố khi có mặt chuỗi bin hậu tố. Ngoài ra, khi không có mặt chuỗi bin hậu tố, thì chuỗi bin cho $\text{dec_abs_level}[n]$ có thể là chuỗi bin tiền tố.

Ví dụ, chuỗi bin tiền tố có thể được dẫn xuất như được mô tả dưới đây.

Giá trị tiền tố prefixVal của $\text{dec_abs_level}[n]$ có thể được suy ra như được thể hiện trong phương trình sau đây.

[Phương trình 15]

$$\text{prefixVal} = \text{Min}(\text{cMax}, \text{dec_abs_level}[n])$$

Tiền tố của chuỗi bin (tức là, chuỗi bin tiền tố) của $\text{dec_abs_level}[n]$ có thể được suy ra thông qua quy trình nhị phân hóa TR cho prefixVal , trong đó cMax và cRiceParam được sử dụng như đầu vào.

Nếu chuỗi bin tiền tố là giống với chuỗi bit trong đó tất cả các bit là 1 và chiều dài bit là 6, thì chuỗi bin hậu tố của chuỗi bin của $\text{dec_abs_level}[n]$ có thể tồn tại, và có thể được dẫn xuất như được mô tả dưới đây.

Quy trình dẫn xuất tham số rice cho $\text{dec_abs_level}[n]$ có thể là như sau.

Đầu vào của quy trình dẫn xuất tham số rice có thể là chỉ số thành phần màu cIdx , vị trí độ chói (x_0, y_0), vị trí quét hệ số hiện tại (x_C, y_C), $\log_2 \text{TbWidth}$ như lôgarit nhị phân của chiều rộng của khối biến đổi, và $\log_2 \text{TbHeight}$ như lôgarit nhị phân của chiều cao của khối biến đổi. Vị trí độ chói (x_0, y_0) có thể chỉ báo mẫu phía trên-bên trái của khối biến đổi độ chói hiện tại dựa trên mẫu độ chói phía trên-bên trái của hình ảnh. Ngoài ra, đầu ra của quy trình dẫn xuất tham số rice có thể là tham số rice cRiceParam .

Ví dụ, biến `locSumAbs` có thể được dẫn xuất tương tự với mã giả được bọc lỏ trong bảng dưới đây, dựa trên mảng `AbsLevel[x][y]` cho khối biến đổi có chỉ số thành phần đã biết `cIdx` và vị trí độ chói trên cùng-bên trái (x_0, y_0).

[Bảng 14]

```

locSumAbs = 0
if( xC < (1 << log2TbWidth) - 1 ) {
    locSumAbs += AbsLevel[ xC + 1 ][ yC ]
    if( xC < (1 << log2TbWidth) - 2 )
        locSumAbs += AbsLevel[ xC + 2 ][ yC ]
    if( yC < (1 << log2TbHeight) - 1 )
        locSumAbs += AbsLevel[ xC + 1 ][ yC + 1 ]
}
if( yC < (1 << log2TbHeight) - 1 ) {
    locSumAbs += AbsLevel[ xC ][ yC + 1 ]
    if( yC < (1 << log2TbHeight) - 2 )
        locSumAbs += AbsLevel[ xC ][ yC + 2 ]
}
locSumAbs = Clip3( 0, 31, locSumAbs - baseLevel * 5 )

```

(1532)

Sau đó, dựa trên biến đã cho `locSumAbs`, thông số rice `cRiceParam` có thể được dẫn xuất như được thể hiện trong bảng dưới đây.

[Bảng 15]

locSumAbs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cRiceParam	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2
locSumAbs	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
cRiceParam	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3

Ngoài ra, ví dụ, trong quy trình dẫn xuất tham số rice cho `dec_abs_level[n]`, `baseLevel` có thể được thiết đặt là 0, và `ZeroPos[n]` có thể được dẫn xuất như sau.

[Phương trình 16]

$$\text{ZeroPos}[n] = (\text{QState} < 2 ? 1 : 2) << \text{cRiceParam}$$

Ngoài ra, giá trị hậu tố `suffixVal` của `dec_abs_level[n]` có thể được dẫn xuất như được thể hiện trong phương trình sau.

[Phương trình 17]

$$\text{suffixVal} = \text{dec_abs_level}[n] - \text{cMax}$$

Chuỗi bin hậu tố của chuỗi bin của `dec_abs_level[n]` có thể được dẫn xuất thông qua quy trình nhị phân hóa EGk giới hạn cho `suffixVal` trong đó `k` được thiết đặt là

cRiceParam+1, truncSuffixLen được thiết đặt là 15, và maxPreExtLen được thiết đặt là 11.

Cùng lúc đó, RRC và TSRC có thể có các sự khác biệt sau đây.

- Ví dụ, tham số Rice cho phần tử cú pháp `abs_remainder[]` trong TSRC có thể được dẫn xuất như 1. Tham số Rice cRiceParam của phần tử cú pháp `abs_remainder[]` in RRC có thể được dẫn xuất dựa trên `lastAbsRemainder` và `lastRiceParam` như được mô tả nêu trên, nhưng tham số Rice cRiceParam của phần tử cú pháp `abs_remainder[]` in TSRC có thể được dẫn xuất như 1. Tức là, ví dụ, khi việc bỏ qua biến đổi được áp dụng cho khối hiện tại (ví dụ, TB hiện tại), thì tham số Rice cRiceParam cho `abs_remainder[]` của TSRC cho khối hiện tại có thể được dẫn xuất là 1.

- Ngoài ra, ví dụ, tham khảo Bảng 3 và Bảng 4, trong RRC, `abs_level_gtx_flag[n][0]` và/hoặc `abs_level_gtx_flag[n][1]` có thể được báo hiệu, nhưng trong TSRC, `abs_level_gtx_flag[n][0]`, `abs_level_gtx_flag[n][1]`, `abs_level_gtx_flag[n][2]`, `abs_level_gtx_flag[n][3]`, và `abs_level_gtx_flag[n][4]` có thể được báo hiệu. Ở đây, `abs_level_gtx_flag[n][0]` có thể được thể hiện dưới dạng `abs_level_gt1_flag` hoặc cờ mức hệ số thứ nhất, `abs_level_gtx_flag[n][1]` có thể được thể hiện dưới dạng `abs_level_gt3_flag` hoặc cờ mức hệ số thứ hai, `abs_level_gtx_flag[n][2]` có thể được thể hiện dưới dạng `abs_level_gt5_flag` hoặc cờ mức hệ số thứ ba, `abs_level_gtx_flag[n][3]` có thể được thể hiện dưới dạng `abs_level_gt7_flag` hoặc cờ mức hệ số thứ tư, và `abs_level_gtx_flag[n][4]` có thể được thể hiện dưới dạng `abs_level_gt9_flag` hoặc cờ mức hệ số thứ năm. Cụ thể là, cờ mức hệ số thứ nhất có thể là cờ về việc liệu mức hệ số có lớn hơn ngưỡng thứ nhất (ví dụ, 1) hay không, cờ mức hệ số thứ hai có thể là cờ về việc liệu mức hệ số có lớn hơn ngưỡng thứ hai (ví dụ, 3) hay không, cờ mức hệ số thứ ba có thể là cờ về việc liệu mức hệ số có lớn hơn ngưỡng thứ ba (ví dụ, 5) hay không, cờ mức hệ số thứ tư có thể là cờ về việc liệu mức hệ số có lớn hơn ngưỡng thứ tư (ví dụ, 7) hay không, cờ mức hệ số thứ năm có thể là cờ về việc liệu mức hệ số có lớn hơn ngưỡng thứ năm (ví dụ, 9) hay không. Như được mô tả ở trên, trong TSRC, được so sánh với RRC, `abs_level_gtx_flag[n][0]`, `abs_level_gtx_flag[n][1]`, và `abs_level_gtx_flag[n][2]`, `abs_level_gtx_flag[n][3]`, `abs_level_gtx_flag[n][4]` có thể được còn được bao gồm.

- Ngoài ra, ví dụ, trong RRC, phần tử cú pháp `coeff_sign_flag` có thể được lập mã đi vòng qua, nhưng trong TSRC, phần tử cú pháp `coeff_sign_flag` có thể được lập mã đi vòng qua hoặc được lập mã ngừng cảnh.

- Cũng như vậy, ví dụ, khi bin được lập mã ngừng cảnh cho khối hiện tại bị cạn kiệt, trong RRC, thì nó có thể được lập mã như phần tử cú pháp `dec_abs_level`, nhưng trong TSRC, nó có thể được lập mã như phần tử cú pháp `abs_remainder`.

- Cũng như vậy, ví dụ, thứ tự để phân tích cú pháp các hệ số biến đổi của RRC có thể được phân tích cú pháp theo thứ tự hứa hẹn theo hướng bên trái dưới cùng sang bên phải trên cùng dựa trên hệ số khác không cuối cùng, nhưng trong trường hợp của TSRC, thì nó có thể được phân tích cú pháp theo thứ tự hứa hẹn theo hướng bên phải bên trái-dưới cùng, và vị trí của hệ số khác không cuối cùng có thể được lược bỏ.

- Cũng như vậy, ví dụ, trong RRC, việc lượng tử hóa phụ thuộc (dependent quantization, DQ) hoặc phương pháp của phương pháp ẩn dữ liệu về dấu (sign data hiding method, SDH) có thể được áp dụng, nhưng trong TSRC, việc lượng tử hóa phụ thuộc và các phương pháp ẩn dữ liệu về dấu có thể không được sử dụng.

Cũng như vậy, phương pháp ẩn dữ liệu về dấu (SDH) có thể được đề xuất liên quan đến việc lập mã phần dư. Phương pháp ẩn dữ liệu về dấu có thể là như sau.

Khi dẫn xuất hệ số biến đổi, thì dấu của hệ số biến đổi có thể được dẫn xuất dựa trên cờ dấu 1 bit (phần tử cú pháp `coeff_sign_flag` được mô tả ở trên). Về mặt này, SDH có thể chỉ báo kỹ thuật để lược bỏ việc báo hiệu rõ ràng đối với `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất trong khối con/nhóm hệ số (coefficient group, CG) để nâng cao hiệu quả lập mã. Ở đây, giá trị của `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất dựa trên tổng số của các mức tuyệt đối (tức là, các giá trị tuyệt đối) của các hệ số biến đổi có nghĩa trong khối con/nhóm hệ số tương ứng. Tức là, dấu của hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất dựa trên tổng của các mức tuyệt đối của các hệ số biến đổi có nghĩa trong khối con/nhóm hệ số tương ứng. Trong khi đó, hệ số biến đổi có nghĩa có thể đề cập đến hệ số biến đổi khác không mà giá trị (tuyệt đối) của nó không là 0. Ví dụ, khi tổng của các mức tuyệt đối cho các hệ số biến đổi có nghĩa là chẵn, thì giá trị của `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như 1, trong khi, khi tổng của các mức tuyệt đối của các hệ số biến

đổi có nghĩa là lẻ, thì giá trị của `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như 0. Nói cách khác, ví dụ, khi tổng của các mức tuyệt đối cho các hệ số biến đổi có nghĩa là chẵn, thì dấu cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị âm, trong khi, khi tổng của các mức tuyệt đối của các hệ số biến đổi có nghĩa là lẻ, thì dấu cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị dương. Thay vào đó, ví dụ, khi tổng của các mức tuyệt đối cho các hệ số biến đổi có nghĩa là chẵn, thì giá trị của `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như 0, trong khi, khi tổng của các mức tuyệt đối của các hệ số biến đổi có nghĩa là lẻ, thì giá trị của `coeff_sign_flag` cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như 1. Nói cách khác, ví dụ, khi tổng của các mức tuyệt đối cho các hệ số biến đổi có nghĩa là chẵn, thì dấu cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị dương, trong khi, khi tổng của các mức tuyệt đối của các hệ số biến đổi có nghĩa là lẻ, thì dấu cho hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị âm.

Ví dụ, SDH trong cú pháp phần dư có thể được biểu diễn như được thể hiện trong bảng sau đây.

[Bảng 16]

}	
<code>signHiddenFlag = sh_sign_data_hiding_used_flag && (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)</code>	
<code>for(n = numSbCoeff - 1; n >= 0; n--) {</code>	
<code>xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]</code>	
<code>yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]</code>	
<code>if((AbsLevel[xC][yC] > 0) && (!signHiddenFlag (n != firstSigScanPosSb)))</code>	
<code>coeff_sign_flag[n]</code>	<code>ae(v)</code>

Tham chiếu đến Bảng 16, biến `signHiddenFlag` có thể chỉ báo liệu SDH được áp dụng hay không. Biến `signHiddenFlag` có thể cũng được gọi là `signHidden`. Ví dụ, khi giá trị của biến `signHiddenFlag` là 0, thì biến `signHiddenFlag` có thể chỉ báo rằng SDH không được áp dụng, trong khi, khi giá trị của biến `signHiddenFlag` là 1, thì biến `signHiddenFlag` có thể chỉ báo rằng SDH được áp dụng. Ví dụ, giá trị của biến `signHiddenFlag` có thể được thiết đặt dựa trên thông tin cờ được báo hiệu (ví dụ, `sh_sign_data_hiding_used_flag` hoặc `pic_sign_data_hiding_enabled_flag` hoặc `sps_sign_data_hiding_enabled_flag`). Cũng như vậy, ví dụ, giá trị của biến

signHiddenFlag có thể được thiết đặt dựa trên lastSigScanPosSb và firstSigScanPosSb. Ở đây, lastSigScanPosSb có thể chỉ báo vị trí của hệ số biến đổi có nghĩa cuối cùng được tìm kiếm trong khối con/nhóm hệ số tương ứng theo thứ tự quét, và firstSigScanPosSb có thể chỉ báo vị trí của hệ số biến đổi có nghĩa thứ nhất được tìm kiếm trong khối con/nhóm hệ số tương ứng theo thứ tự quét. Nhìn chung, lastSigScanPosSb có thể được đặt trong vùng thành phần tần số tương đối cao so với firstSigScanPosSb. Theo đó, khi lastSigScanPosSb - firstSigScanPosSb là lớn hơn ngưỡng định trước, giá trị signHidden có thể được dẫn xuất như 1 (tức là, SDH được áp dụng), trong khi trái lại giá trị signHidden có thể được dẫn xuất như 0 (tức là, SDH không được áp dụng). Ở đây, ví dụ, tham chiếu đến Bảng 35, giá trị ngưỡng có thể được thiết đặt là 3.

Ngoài ra, tham chiếu đến Bảng 16, thậm chí nếu giá trị của signHiddenFlag là 0 (tức là, !signHiddenFlag), nếu hệ số hiện tại không là hệ số có nghĩa thứ nhất trong khối (khối con) theo thứ tự quét (tức là, $n \neq \text{firstSigScanPosSb}$), sau đó coeff_sign_flag[n] cho hệ số hiện tại có thể được báo hiệu rõ ràng.

Cũng như vậy, tham chiếu đến Bảng 16, nếu giá trị của signHiddenFlag là 1, và hệ số hiện tại là hệ số có nghĩa thứ nhất trong khối (khối con) theo thứ tự quét (tức là, $n = \text{first SigScanPosSb}$), sau đó việc báo hiệu rõ ràng coeff_sign_flag[n] cho hệ số hiện tại có thể được lược bỏ. Trong trường hợp này, giá trị của coeff_sign_flag[n] cho hệ số hiện tại (tức là, hệ số có nghĩa thứ nhất) có thể được dẫn xuất như sau. Ví dụ, giá trị của coeff_sign_flag[n] cho hệ số có nghĩa thứ nhất có thể được dẫn xuất dựa trên các giá trị coeff_sign_flag[n] cho các hệ số có nghĩa trong khối (khối con) tương ứng. Ví dụ, khi tổng của các giá trị coeff_sign_flag[n] cho các hệ số có nghĩa là chẵn, coeff_sign_flag[n] cho hệ số có nghĩa thứ nhất có thể được dẫn xuất như 1, trong khi, khi tổng của các giá trị coeff_sign_flag[n] cho các hệ số có nghĩa còn lại ngoại trừ hệ số có nghĩa thứ nhất là lẻ, coeff_sign_flag[n] cho hệ số có nghĩa thứ nhất có thể được dẫn xuất như 0. Thay vào đó, khi tổng của các giá trị coeff_sign_flag[n] cho các hệ số có nghĩa là chẵn, thì coeff_sign_flag[n] cho hệ số có nghĩa thứ nhất có thể được dẫn xuất như 0, trong khi, khi tổng của các giá trị coeff_sign_flag[n] cho các hệ số có nghĩa là lẻ, thì coeff_sign_flag[n] cho hệ số có nghĩa thứ nhất có thể được dẫn xuất như 1.

Trong khi đó, nếu trong cú pháp mức cao (VPS, SPS, PPS, cú pháp tiêu đề lát, v.v.) hoặc cú pháp mức thấp (cú pháp dữ liệu lát, cú pháp đơn vị lập mã, cú pháp đơn vị biến

đổi, v.v.), việc ẩn dữ liệu về dấu được mô tả ở trên được kích hoạt, và nếu `sh_ts_residual_coding_disabled_flag` là 1, thì quy trình ẩn dữ liệu về dấu của RRC có thể được sử dụng khi lập mã không suy hao. Theo đó, việc lập mã không suy hao có thể trở thành không thể do các thiết đặt không đúng trong thiết bị mã hóa. Thay vào đó, nếu việc lập mã suy hao (tức là, phương pháp lập mã không thể đảo ngược lại) chứ không phải lập mã không suy hao được áp dụng, và tín hiệu phần dư mà việc bỏ qua biến đổi đã được áp dụng cho được lập mã với RRC trong khi đó cùng lúc BDPCM được áp dụng, thì sau đó BDPCM có thể phải chịu sự suy hao lập mã bởi vì SDH được thực hiện theo điều kiện áp dụng SDH, bất kể thực tế là khoảng mà giá trị phần dư trở thành 0 xuất hiện thường xuyên hơn so với trường hợp thông thường do sự chênh lệch giữa các phần dư. Cụ thể, ví dụ, nếu các hệ số biến đổi có nghĩa (dữ liệu phần dư khác không) tồn tại lần lượt ở các vị trí 0 và 15 trong CG, và các giá trị của các hệ số biến đổi tại các vị trí còn lại trong CG là 0, sau đó SDH có thể được áp dụng cho CG theo các điều kiện áp dụng SDH được mô tả ở trên, và do đó dữ liệu về dấu (tức là, việc lập mã cờ dấu) cho hệ số biến đổi có nghĩa thứ nhất của CG có thể được lược bỏ. Theo đó, trong trường hợp này, tính chẵn lẻ của chỉ hai dữ liệu phần dư của CG có thể được điều chỉnh trong bước lượng tử hóa để lược bỏ dữ liệu về dấu, và do đó nhiều suy hao lập mã có thể xuất hiện so với trường hợp SDH không được áp dụng. Các trường hợp này có thể cũng xuất hiện trong các khối mà BDPCM không được áp dụng với, nhưng do các đặc điểm của BDPCM, mức được hạ thấp thông qua sự chênh lệch với phần dư lân cận, vì vậy các trường hợp bất lợi có thể xuất hiện thường xuyên hơn khi áp dụng SDH.

Theo đó, trong trường hợp này, để ngăn sự suy hao lập mã không được mong muốn hoặc sự hư hỏng bị gây ra bởi việc sử dụng SDH và lập mã phần dư cùng nhau khi `sh_ts_residual_coding_disabled_flag` = 1 (tức là, lập mã các mẫu phần dư của khối bỏ qua biến đổi trong lát hiện tại với RRC), thì có các phương án để thiết đặt sự phụ thuộc/ràng buộc giữa hai kỹ thuật được mô tả ở trên được cung cấp.

Trong khi đó, như được mô tả ở trên, phương pháp lập mã dữ liệu phần dư có thể bao gồm việc lập mã phần dư thông thường (regular residual coding, RRC) và việc lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC).

Như được thể hiện trong Bảng 1, phương pháp lập mã dữ liệu phần dư cho khối hiện tại trong số hai phương pháp nêu trên có thể được xác định dựa trên các giá trị của

transform_skip_flag và sh_ts_residual_coding_disabled_flag. Ở đây, phần tử cú pháp sh_ts_residual_coding_disabled_flag có thể thể hiện liệu TSRC có được cho phép hay không. Theo đó, thậm chí khi transform_skip_flag chỉ báo rằng việc biến đổi được bỏ qua, nếu sh_ts_residual_coding_disabled_flag chỉ báo rằng TSRC không được cho phép, thì sau đó các phần tử cú pháp theo RRC so với khối bỏ qua biến đổi có thể được báo hiệu. Tức là, khi giá trị của transform_skip_flag là 0 hoặc giá trị của sh_ts_residual_coding_disabled_flag là 1, thì RRC có thể được sử dụng, trong khi trái lại TSRC có thể được sử dụng.

Tài liệu này đề xuất như một phương án phương pháp trong đó sh_ts_residual_coding_disabled_flag phụ thuộc vào pic_sign_data_hiding_enabled_flag. Ví dụ, các phần tử cú pháp được đề xuất trong phương án này có thể như được thể hiện trong Bảng sau đây.

[Bảng 17]

slice header() {	Descriptor
picture header in slice header flag	u(1)
if(picture header in slice header flag)	
picture header structure()	
(...)	
if(!pic sign data hiding enabled flag)	
sh ts residual coding disabled flag	u(1)
if(ph lmcs enabled flag)	
slice lmcs enabled flag	u(1)
if(pic scaling list enabled flag)	
slice scaling list present flag	u(1)
if(NumEntryPoints > 0) {	
offset len minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry point offset minus1[i]	u(v)
}	
if(slice header extension present flag) {	
slice header extension length	ue(v)
for(i = 0; i < slice header extension length; i++)	
slice header extension data byte[i]	u(8)
}	
byte alignment()	
}	

Ở đây, ví dụ, pic_sign_data_hiding_enabled_flag có thể là cờ cho việc liệu việc ẩn dữ liệu về dấu được cho phép hay không. Ví dụ, pic_sign_data_hiding_enabled_flag có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép. Tức là, ví dụ,

`pic_sign_data_hiding_enabled_flag` có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép cho các khối của các hình ảnh cho trình tự hay cấu trúc tiêu đề hình ảnh (tức là, `picture_header_structure()`). Ví dụ, `pic_sign_data_hiding_enabled_flag` có thể chỉ báo liệu có được sử dụng việc ẩn dữ liệu về dấu chỉ báo liệu việc ẩn dữ liệu về dấu được sử dụng cho lát hiện tại hay không có thể là có mặt. Ví dụ, `pic_sign_data_hiding_enabled_flag` mà giá trị của nó là 1 có thể chỉ báo rằng việc ẩn dữ liệu về dấu được cho phép, trong khi `pic_sign_data_hiding_enabled_flag` mà giá trị của nó là 0 có thể chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép. Ví dụ, `pic_sign_data_hiding_enabled_flag` mà giá trị của nó là 1 có thể chỉ báo rằng có về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng có thể là có mặt, trong khi `pic_sign_data_hiding_enabled_flag` mà giá trị của nó là 0 có thể chỉ báo rằng có về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng với là không có mặt.

Theo Bảng 17 được mô tả ở đây, `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu chỉ khi việc ẩn dữ liệu về dấu không được cho phép. Thêm vào đó, khi việc ẩn dữ liệu về dấu được cho phép, thì `sh_ts_residual_coding_disabled_flag` có thể không được báo hiệu, và giá trị của `sh_ts_residual_coding_disabled_flag` có thể được suy ra là 0 (lập mã các mẫu phần dư của khối bỏ qua biến đổi trong lát hiện tại với cú pháp TSRC) hoặc 1 (lập mã các mẫu phần dư của khối bỏ qua biến đổi trong lát hiện tại với cú pháp RRC).

Ở đây, ví dụ, `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu như cú pháp tiêu đề hình ảnh hoặc cú pháp tiêu đề lát. Ví dụ, khi `pic_sign_data_hiding_enabled_flag` được báo hiệu như cú pháp không phải cú pháp tiêu đề hình ảnh, thì nó có thể được gọi tên khác. Ví dụ, `pic_sign_data_hiding_enabled_flag` có thể được biểu diễn như `slice_sign_data_hiding_enabled_flag`. Thêm vào đó, `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu như cú pháp tiêu đề lát, hoặc có thể được báo hiệu trong cú pháp mức cao (HLS)(eg, cú pháp SPS/cú pháp VPS/cú pháp PPS/cú pháp PH/cú pháp DPS, v.v) hoặc mức thấp (CU/TU) không phải cú pháp tiêu đề lát. Khi phương pháp lập mã phần dư được xác định bằng việc liệu SDH được cho phép hay không bất kể mối quan hệ trên/dưới của các cú pháp được báo hiệu hoặc vị trí về cú pháp, thì có thể được diễn dịch như tương thích với phương án này.

Trong khi đó, theo việc lập mã ảnh/video thông thường, trong cú pháp mức cao (cú pháp SPS/cú pháp VPS/cú pháp PPS/cú pháp DPS/cú pháp tiêu đề hình ảnh/cú pháp tiêu đề lát, v.v.) hoặc mức thấp (CU/TU), SDH được cho phép, và khi `sh_ts_residual_coding_disabled_flag` là 1, thì SDH trong RRC được mô tả nêu trên có thể được sử dụng cho việc lập mã không suy hao, vì vậy việc lập mã không suy hao có thể trở nên không thể do sự thiết đặt không đúng trong thiết bị mã hóa. Theo đó, trong tài liệu này, để ngăn sự suy hao lập mã không chủ đích hoặc sự hư hỏng bị gây ra bằng việc sử dụng SDH và c phần dư cùng nhau khi `sh_ts_residual_coding_disabled_flag` = 1 (tức là, lập mã các mẫu phần dư các khối bỏ qua biến đổi trong lát hiện tại với RRC), thì có phương án trong đó SDH không được sử dụng khi lập mã mức của hệ số biến đổi khi giá trị của `transform_skip_flag` là 1. Cú pháp lập mã phần dư theo phương án được đề xuất có thể là như trong Bảng sau đây.

[Bảng 18]

residual coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth == 5 && log2TbHeight < 6)	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min(log2TbWidth, 5)	
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth < 6 && log2TbHeight == 5)	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last sig coeff x prefix	ae(v)
if(log2TbHeight > 0)	
last sig coeff y prefix	ae(v)
if(last sig coeff x prefix > 3)	
last sig coeff x suffix	ae(v)
if(last sig coeff y prefix > 3)	
last sig coeff y suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3)	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	

numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfntDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfntZeroOutSigCoeffFlag = 0	
if((lastSubBlock > 0 lastScanPos > 0) && cIdx == 0)	
MtsDcOnly = 0	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[i][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]	
[i][1]	
inferSbDcSigCoeffFlag = 0	

if(i < lastSubBlock && i > 0) {	
coded_sub_block_flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
if(coded_sub_block_flag[xS][yS] && (xS > 3 yS > 3) && cIdx == 0)	
MtsZeroOutSigCoeffFlag = 0	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = firstPosMode0	
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag [xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag [xC][yC]) {	
abs_level_gtx_flag [n][0]	ae(v)
remBinsPass1--	
if(abs_level_gtx_flag [n][0]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag [n][1]	ae(v)
remBinsPass1--	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	

AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n][1]	
if(ph_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
firstPosMode1 = n - 1	
}	
for(n = firstPosMode0; n > firstPosMode1; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(abs_level_gtx_flag[n][1])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
}	
for(n = firstPosMode1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(coded_sub_block_flag[xS][yS])	
dec_abs_level[n]	ae(v)
if(AbsLevel[xC][yC] > 0) {	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
if(ph_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
if(ph_dep_quant_enabled_flag !pic_sign_data_hiding_enabled_flag transform_skip_flag[x0][y0][cIdx])	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	

for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if((AbsLevel[xC][yC] > 0) && (!signHidden (n != firstSigScanPosSb)))	
coeff sign flag[n]	ae(v)
}	
if(ph dep quant enabled flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(AbsLevel[xC][yC] > 0)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff sign flag[n])	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(AbsLevel[xC][yC] > 0) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff sign flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)	
)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
}	

Tham chiếu đến Bảng 18 được mô tả ở trên, biến signHidden chỉ báo liệu SDH được áp dụng có thể được dẫn xuất dựa trên giá trị của transform_skip_flag. Ví dụ, khi giá trị của transform_skip_flag là 1, thì giá trị của signHidden có thể được dẫn xuất như 0. Tức là, ví dụ, khi giá trị của transform_skip_flag là 1, thì SDH có thể không được áp dụng khi dẫn xuất dấu của hệ số biến đổi của khối hiện tại.

Thêm vào đó, trong tài liệu này, để ngăn sự suy hao lập mã không chủ đích hoặc sự hư hỏng bị gây ra bằng việc sử dụng SDH và các phần dư cùng nhau khi $\text{sh_ts_residual_coding_disabled_flag} = 1$ (tức là, lập mã các mẫu phần dư các khối bỏ qua biến đổi trong lát hiện tại với RRC), thì có phương án trong đó SDH không được sử dụng khi lập mã mức của hệ số biến đổi khi giá trị của $\text{transform_skip_flag}$ là 1. Cách pháp lập mã phần dư theo phương án được đề xuất có thể là như trong Bảng sau đây.

[Bảng 19]

<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {</code>	Descriptor
<code>if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth == 5 && log2TbHeight < 6)</code>	
<code>log2ZoTbWidth = 4</code>	
<code>else</code>	
<code>log2ZoTbWidth = Min(log2TbWidth, 5)</code>	
<code>if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth < 6 && log2TbHeight == 5)</code>	
<code>log2ZoTbHeight = 4</code>	
<code>else</code>	
<code>log2ZoTbHeight = Min(log2TbHeight, 5)</code>	
<code>if(log2TbWidth > 0)</code>	
<code>last_sig_coeff_x_prefix</code>	ae(v)
<code>if(log2TbHeight > 0)</code>	
<code>last_sig_coeff_y_prefix</code>	ae(v)
<code>if(last_sig_coeff_x_prefix > 3)</code>	
<code>last_sig_coeff_x_suffix</code>	ae(v)
<code>if(last_sig_coeff_y_prefix > 3)</code>	
<code>last_sig_coeff_y_suffix</code>	ae(v)
<code>log2TbWidth = log2ZoTbWidth</code>	
<code>log2TbHeight = log2ZoTbHeight</code>	
<code>remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2</code>	
<code>log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)</code>	
<code>log2SbH = log2SbW</code>	
<code>if(log2TbWidth + log2TbHeight > 3)</code>	
<code>if(log2TbWidth < 2) {</code>	
<code>log2SbW = log2TbWidth</code>	
<code>log2SbH = 4 - log2SbW</code>	
<code>} else if(log2TbHeight < 2) {</code>	
<code>log2SbH = log2TbHeight</code>	
<code>log2SbW = 4 - log2SbH</code>	
<code>}</code>	

numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH][lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH][lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfstZeroOutSigCoeffFlag = 0	
if((lastSubBlock > 0 lastScanPos > 0) && cIdx == 0)	
MtsDcOnly = 0	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH][i][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH][i][1]	
inferSbDcSigCoeffFlag = 0	

if(i < lastSubBlock && i > 0) {	
coded_sub_block_flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
if(coded_sub_block_flag[xS][yS] && (xS > 3 yS > 3) && cIdx == 0)	
MtsZeroOutSigCoeffFlag = 0	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = firstPosMode0	
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) &&	
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag [xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag [xC][yC]) {	
abs_level_gtx_flag [n][0]	ae(v)
remBinsPass1--	
if(abs_level_gtx_flag [n][0]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag [n][1]	ae(v)
remBinsPass1--	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	

AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n][1]	
if(ph_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
firstPosMode1 = n - 1	
}	
for(n = firstPosMode0; n > firstPosMode1; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(abs_level_gtx_flag[n][1])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
}	
for(n = firstPosMode1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
if(coded_sub_block_flag[xS][yS])	
dec_abs_level[n]	ae(v)
if(AbsLevel[xC][yC] > 0) {	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
if(ph_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
}	
if(ph_dep_quant_enabled_flag !pic_sign_data_hiding_enabled_flag BdpcmFlag[x0][y0][cIdx])	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	

for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if((AbsLevel[xC][yC] > 0) && (!signHidden (n != firstSigScanPosSb)))	
coeff sign flag[n]	ae(v)
}	
if(ph dep quant enabled flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(AbsLevel[xC][yC] > 0)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff sign flag[n])	
QState = QStateTransTable[QState][AbsLevel[xC][yC] & 1]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n]	
[0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n]	
[1]	
if(AbsLevel[xC][yC] > 0) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff sign flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1)	
)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
}	
}	

Tham chiếu đến Bảng 19 nêu trên, biến signHidden chỉ báo liệu SDH được áp dụng có thể được dẫn xuất dựa trên giá trị của biến BdpcmFlag chỉ báo liệu BDPCM được áp dụng hay không. Ví dụ, khi giá trị của BdpcmFlag là 1, thì giá trị của signHidden có thể được dẫn xuất như 0. Tức là, ví dụ, khi giá trị của BdpcmFlag là 1 (khi BDPCM

được áp dụng cho khối hiện tại), thì SDH có thể không được áp dụng khi dẫn xuất đầu của hệ số biến đổi của khối hiện tại.

Tham chiếu đến Bảng 19, khi BdpcmFlag là 1, thì SDH cho TSRC được cho phép nếu việc lập mã suy hao được áp dụng, nhưng SDH có thể không được sử dụng nếu BDPCM được áp dụng.

Ngoài ra, tài liệu này đề xuất các phương án khác nhau liên quan đến việc báo hiệu phần tử cú pháp `sh_ts_residual_coding_disabled_flag` được mô tả ở trên.

Ví dụ, như được mô tả ở trên, bởi vì `sh_ts_residual_coding_disabled_flag` là phần tử cú pháp xác định liệu TSRC không được cho phép hay không, thì nó có thể không cần được báo hiệu khi khối bỏ qua biến đổi không được sử dụng. Tức là, nó có thể là ý nghĩa để báo hiệu `sh_ts_residual_coding_disabled_flag` chỉ khi phần tử cú pháp cho việc liệu khối bỏ qua biến đổi được sử dụng chỉ báo rằng khối bỏ qua biến đổi được sử dụng hay không.

Do đó, tài liệu này đề xuất phương án trong đó `sh_ts_residual_coding_disabled_flag` được báo hiệu chỉ khi `sps_transform_skip_enabled_flag` là 1. Cú pháp theo phương án này được thể hiện trong Bảng sau đây.

[Bảng 20]

slice header() {	Descriptor
(...)	
if(sps transform skip enabled flag)	
sh ts residual coding disabled flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 20, khi `sps_transform_skip_enabled_flag` là 1, thì `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi đó, khi `sps_transform_skip_enabled_flag` là 0, `sh_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Ở đây, ví dụ, `sps_transform_skip_enabled_flag` có thể chỉ báo liệu khối bỏ qua biến đổi được sử dụng. Tức là, ví dụ, `sps_transform_skip_enabled_flag` có thể chỉ báo liệu việc bỏ qua biến đổi được cho phép. Ví dụ, khi giá trị của `sps_transform_skip_enabled_flag` là 1, thì `sps_transform_skip_enabled_flag` có thể chỉ báo rằng cờ bỏ qua biến đổi (`transform_skip_flag`) có thể là có mặt trong cú pháp đơn

vị biến đổi, trong khi đó, khi giá trị của `sps_transform_skip_enabled_flag` là 0, `sps_transform_skip_enabled_flag` có thể chỉ báo rằng cờ bỏ qua biến đổi là không có mặt trong cú pháp đơn vị biến đổi. Trong khi đó, khi `sh_ts_residual_coding_disabled_flag` không được báo hiệu, thì `sh_ts_residual_coding_disabled_flag` có thể được dẫn xuất là 0. Ngoài ra, `sps_transform_skip_enabled_flag` được mô tả ở trên có thể được báo hiệu trong SPS, hoặc có thể được báo hiệu trong cú pháp mức cao (VPS, PPS, cú pháp tiêu đề hình ảnh, cú pháp tiêu đề lát, hoặc tương tự) hoặc cú pháp mức thấp (cú pháp dữ liệu lát, cú pháp đơn vị lập mã, cú pháp đơn vị biến đổi, hoặc tương tự) không phải SPS. Cũng như vậy, nó có thể được báo hiệu trước `sh_ts_residual_coding_disabled_flag`.

Thêm vào đó, tài liệu này đề xuất phương án kết hợp các phương án được mô tả nêu trên kết hợp với việc báo hiệu `sh_ts_residual_coding_disabled_flag`. ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 21]

slice_header() {	Descriptor
(...)	
if(!pic_sign_data_hiding_enabled_flag && sps_transform_skip_enabled_flag)	
sh ts residual coding disabled flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 21, khi `sps_transform_skip_enabled_flag` là 1 và `pic_sign_data_hiding_enabled_flag` là 0, thì `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `sh_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Trong khi đó, khi `sh_ts_residual_coding_disabled_flag` không được báo hiệu, thì `sh_ts_residual_coding_disabled_flag` có thể được dẫn xuất là 0.

Thay vào đó, ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 22]

picture header structure() {	Descriptor
(...)	
ph_dep_quant_enabled_flag	
if(!pic_sign_data_hiding_enabled_flag sps_transform_skip_enabled_flag)	
ph_ts_residual_coding_disabled_flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 22, khi `pic_sign_data_hiding_enabled_flag` là 0 và `sps_transform_skip_enabled_flag` là 1, thì `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `sh_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Trong khi đó, khi `sh_ts_residual_coding_disabled_flag` không được báo hiệu, thì `sh_ts_residual_coding_disabled_flag` có thể được dẫn xuất là 0.

Cũng như vậy, ví dụ, theo phương án này, phương pháp để báo hiệu các phân tử cú pháp `ph_dep_quant_enabled_flag` và `sh_ts_residual_coding_disabled_flag` trong cùng một cú pháp mức cao hoặc cú pháp mức thấp có thể được đề xuất. Ví dụ, tham chiếu đến Bảng 22 nêu trên, cả `ph_dep_quant_enabled_flag` và `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh. Trong trường hợp này, `sh_ts_residual_coding_disabled_flag` có thể được đề cập đến như `ph_ts_residual_coding_disabled_flag`. Trong khi đó, `ph_dep_quant_enabled_flag` có thể là cờ chỉ báo liệu việc lượng tử hóa phụ thuộc được cho phép hay không. Ví dụ, `ph_dep_quant_enabled_flag` có thể chỉ báo liệu việc lượng tử hóa phụ thuộc được cho phép hay không. Tức là, ví dụ, `ph_dep_quant_enabled_flag` có thể chỉ báo liệu việc lượng tử hóa phụ thuộc được cho phép cho các khối của các hình ảnh theo trình tự. Ví dụ, `ph_dep_quant_enabled_flag` có thể chỉ báo liệu cờ được sử dụng việc lượng tử hóa phụ thuộc chỉ báo liệu việc lượng tử hóa phụ thuộc được sử dụng cho lát hiện tại có thể là có mặt. Ví dụ, `ph_dep_quant_enabled_flag` mà giá trị của nó là 1 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc là được phép, trong khi `ph_dep_quant_enabled_flag` mà giá trị của nó là 0 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc không được cho phép. Cũng như vậy, ví dụ, `ph_dep_quant_enabled_flag` có thể được gọi là `sh_dep_quant_enabled_flag` theo cú pháp được báo hiệu.

Thay vào đó, ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 23]

picture header structure() {	Descriptor
(...)	
ph_dep_quant_enabled_flag	
if(!pic_sign_data_hiding_enabled_flag && sps_transform_skip_enabled_flag)	
ph_ts_residual_coding_disabled_flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 23, khi `pic_sign_data_hiding_enabled_flag` là 0 và `sps_transform_skip_enabled_flag` là 1, thì `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `sh_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Trong khi đó, khi `sh_ts_residual_coding_disabled_flag` không được báo hiệu, thì `sh_ts_residual_coding_disabled_flag` có thể được dẫn xuất là 0. Cũng như vậy, ví dụ, tham chiếu đến Bảng 23 nêu trên, cả `ph_dep_quant_enabled_flag` và `sh_ts_residual_coding_disabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh. Trong trường hợp này, `sh_ts_residual_coding_disabled_flag` có thể được đề cập đến như `ph_ts_residual_coding_disabled_flag`.

Ngoài ra, tài liệu này đề xuất phương án trong đó các phần tử cú pháp `ph_dep_quant_enabled_flag`, `pic_sign_data_hiding_enabled_flag` và/hoặc `sh_ts_residual_coding_disabled_flag` được mô tả nêu trên được báo hiệu trong cùng một cú pháp mức cao (VPS, SPS, PPS, tiêu đề hình ảnh, tiêu đề lát, hoặc tương tự) hoặc cú pháp mức thấp (dữ liệu lát, đơn vị lập mã, đơn vị biến đổi, hoặc tương tự).

Ví dụ, như được thể hiện trong Bảng sau đây, phương án trong đó cả `pic_sign_data_hiding_enabled_flag` và `sh_ts_residual_coding_disabled_flag` được báo hiệu trong cú pháp tiêu đề hình ảnh có thể được đề xuất.

[Bảng 24]

picture header structure() {	Descriptor
(...)	
if(sps_sign_data_hiding_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(!pic_sign_data_hiding_enabled_flag)	
ph_ts_residual_coding_disabled_flag	u(1)
(...)	
}	

Trong trường hợp này, `sh_ts_residual_coding_disabled_flag` có thể được đề cập đến như `ph_ts_residual_coding_disabled_flag`.

Theo phương án này, phần tử cú pháp (tức là, `sh_ts_residual_coding_disabled_flag`) chỉ báo liệu việc lập mã phần dư (tức là, TSRC) cho khối bỏ qua biến đổi được cho phép có thể được báo hiệu chỉ khi giá trị của phần tử cú pháp (tức là, `pic_sign_data_hiding_enabled_flag`) chỉ báo liệu SDH được cho phép trong HLS hay không là 0. Ví dụ, tham chiếu đến Bảng 24, `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh, và khi giá trị của `pic_sign_data_hiding_enabled_flag` là 0, thì `ph_ts_residual_coding_disabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh. Trong khi đó, ví dụ, khi giá trị của `pic_sign_data_hiding_enabled_flag` là 1, thì `ph_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Khi `sh_ts_residual_coding_disabled_flag` không được báo hiệu, thì `sh_ts_residual_coding_disabled_flag` có thể được suy ra là 0. Cũng như vậy, khi giá trị của `sps_sign_data_hiding_enabled_flag` là 1, thì `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh.

Phương án theo Bảng 24 được mô tả nêu trên chỉ là ví dụ, và hai phần tử cú pháp có thể được báo hiệu trong cú pháp mức cao (VPS, SPS, PPS, tiêu đề lát, hoặc tương tự) hoặc cú pháp mức thấp (dữ liệu lát, đơn vị lập mã, đơn vị biến đổi, hoặc tương tự) không phải tiêu đề hình ảnh.

Thay vào đó, ví dụ, như được thể hiện trong Bảng sau đây, có thể có phương án trong đó phần tử cú pháp chỉ báo liệu SDH được cho phép được báo hiệu chỉ khi giá trị của phần tử cú pháp chỉ báo liệu việc lập mã phần dư (tức là, TSRC) cho khối bỏ qua biến đổi được cho phép là 0 (tức là, khi TSRC được cho phép) hay không.

[Bảng 25]

picture header structure() {	Descriptor
(...)	
ph ts residual coding disabled flag	
if(!ph ts residual coding disabled flag)	
pic sign data hiding enabled flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 25, khi giá trị của `ph_ts_residual_coding_disabled_flag` là 0, thì `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh. Trong khi đó, ví dụ, khi giá trị của `ph_ts_residual_coding_disabled_flag` là 1, thì `pic_sign_data_hiding_enabled_flag` có thể không được báo hiệu. Cũng như vậy, ví

dụ, khi `pic_sign_data_hiding_enabled_flag` không được báo hiệu, thì `pic_sign_data_hiding_enabled_flag` có thể được suy ra là 0 trong thiết bị giải mã.

Phương án theo Bảng 25 được mô tả nêu trên chỉ là ví dụ, và hai phần tử cú pháp có thể được báo hiệu trong cú pháp mức cao (VPS, SPS, PPS, tiêu đề lát, hoặc tương tự) hoặc cú pháp mức thấp (dữ liệu lát, đơn vị lập mã, đơn vị biến đổi, hoặc tương tự) không phải tiêu đề hình ảnh.

Thay vào đó, ví dụ, các phương pháp để giới hạn `pic_sign_data_hiding_enabled_flag` và/hoặc `ph_dep_quant_enabled_flag` dựa trên `ph_ts_residual_coding_disabled_flag` có thể được đề xuất.

Ví dụ, như được thể hiện trong Bảng sau đây, có thể có phương án được cung cấp trong đó `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` được báo hiệu chỉ khi giá trị của `ph_ts_residual_coding_disabled_flag` là 0.

[Bảng 26]

picture header structure() {	Descriptor
(...)	
ph ts residual coding disabled flag	
if(!ph ts residual coding disabled flag) {	
pic sign data hiding enabled flag	u(1)
ph dep quant enabled flag	u(1)
}	
(...)	
}	

Tham chiếu đến Bảng 26, khi giá trị của `ph_ts_residual_coding_disabled_flag` là 0, thì `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh. Trong khi đó, ví dụ, khi giá trị của `ph_ts_residual_coding_disabled_flag` là 1, thì `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể không được báo hiệu. Cũng như vậy, ví dụ, khi `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` không được báo hiệu, `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể được suy ra là 0 trong thiết bị giải mã.

Cũng như vậy, ví dụ, tham chiếu đến Bảng 26 nêu trên, `ph_ts_residual_coding_disabled_flag`, `pic_sign_data_hiding_enabled_flag`, và `ph_dep_quant_enabled_flag` có thể đều được báo hiệu trong cú pháp tiêu đề hình ảnh.

Thêm vào đó, tài liệu này đề xuất phương án kết hợp các phương án được mô tả nêu trên kết hợp với việc báo hiệu `sh_ts_residual_coding_disabled_flag`. Ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 27]

picture header structure() {	Descriptor
(...)	
if(sps_sign_data_hiding_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(!pic_sign_data_hiding_enabled_flag sps_transform_skip_enabled_flag)	
ph_ts_residual_coding_disabled_flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 27, khi `pic_sign_data_hiding_enabled_flag` là 0 hoặc `sps_transform_skip_enabled_flag` là 1, thì `ph_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `ph_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Trong khi đó, khi `ph_ts_residual_coding_disabled_flag` không được báo hiệu, thì `ph_ts_residual_coding_disabled_flag` có thể được suy ra là 0 trong thiết bị giải mã. Cũng như vậy, khi giá trị của `sps_sign_data_hiding_enabled_flag` là 1, thì `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh.

Thay vào đó, ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 28]

picture header structure() {	Descriptor
(...)	
if(sps_sign_data_hiding_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(!pic_sign_data_hiding_enabled_flag && sps_transform_skip_enabled_flag)	
ph_ts_residual_coding_disabled_flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 28, khi `pic_sign_data_hiding_enabled_flag` là 0 và `sps_transform_skip_enabled_flag` là 1, thì `ph_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `ph_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Trong khi đó, khi `ph_ts_residual_coding_disabled_flag` không được báo hiệu,

thì `ph_ts_residual_coding_disabled_flag` có thể được suy ra là 0 trong thiết bị giải mã. Cũng như vậy, khi giá trị của `sps_sign_data_hiding_enabled_flag` là 1, thì `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu trong cú pháp tiêu đề hình ảnh.

Thay vào đó, ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 29]

picture header structure() {	Descriptor
(...)	
if(sps transform skip enabled flag)	
ph ts residual coding disabled flag	
if(!ph ts residual coding disabled flag)	
pic sign data hiding enabled flag	u(1)
(...)	
}	

Tham chiếu đến Bảng 29, khi `sps_transform_skip_enabled_flag` là 1, thì `ph_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `ph_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Cũng như vậy, tham chiếu đến Bảng 29, khi `ph_ts_residual_coding_disabled_flag` là 0, `pic_sign_data_hiding_enabled_flag` có thể được báo hiệu, trong khi trái lại `pic_sign_data_hiding_enabled_flag` có thể không được báo hiệu. Trong khi đó, khi `ph_ts_residual_coding_disabled_flag` không được báo hiệu, thì `ph_ts_residual_coding_disabled_flag` có thể được suy ra là 0 trong thiết bị giải mã. Cũng như vậy, khi `pic_sign_data_hiding_enabled_flag` không được báo hiệu, thì `pic_sign_data_hiding_enabled_flag` có thể được suy ra là 0 trong thiết bị giải mã.

Thay vào đó, ví dụ, phương án để báo hiệu `sh_ts_residual_coding_disabled_flag` có thể được đề xuất như được thể hiện trong Bảng sau đây.

[Bảng 30]

picture header structure() {	Descriptor
(...)	
if(sps transform skip enabled flag)	
ph ts residual coding disabled flag	
if(!ph ts residual coding disabled flag){	
pic sign data hiding enabled flag	u(1)
ph dep quant enabled flag	u(1)
}	
(...)	
}	

Tham chiếu đến Bảng 30, khi `sps_transform_skip_enabled_flag` là 1, thì `ph_ts_residual_coding_disabled_flag` có thể được báo hiệu, trong khi trái lại `ph_ts_residual_coding_disabled_flag` có thể không được báo hiệu. Ngoài ra, tham chiếu đến Bảng 30, khi `ph_ts_residual_coding_disabled_flag` là 0, `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể được báo hiệu, trong đó trái lại `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể không được báo hiệu. Trong khi đó, khi `ph_ts_residual_coding_disabled_flag` không được báo hiệu, thì `ph_ts_residual_coding_disabled_flag` có thể được suy ra là 0 trong thiết bị giải mã. Cũng như vậy, khi `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` không được báo hiệu, `pic_sign_data_hiding_enabled_flag` và `ph_dep_quant_enabled_flag` có thể được suy ra 0 trong thiết bị giải mã.

Cùng lúc đó, như được mô tả ở trên, thông tin (phần tử cú pháp) trong bảng cú pháp được bộc lộ trong tài liệu này có thể được bao gồm trong thông tin hình ảnh/video, và có thể được định cấu hình/được mã hóa trong thiết bị mã hóa và được truyền tới thiết bị giải mã dưới dạng luồng bit. Thiết bị giải mã có thể phân tích cú pháp/giải mã thông tin (phần tử cú pháp) trong bảng cú pháp tương ứng. Thiết bị giải mã có thể thực hiện quy trình tái dựng khối/hình ảnh/video dựa trên thông tin được giải mã.

Fig.6 minh họa một cách vắn tắt phương pháp mã hóa ảnh được thực hiện bởi thiết bị mã hóa theo sáng chế. Phương pháp được bộc lộ trên Fig.6 có thể được thực hiện bởi thiết bị mã hóa được bộc lộ trên Fig.2. Cụ thể, ví dụ, S600 đến S630 của Fig.6 có thể được thực hiện bằng bộ mã hóa entropy của thiết bị mã hóa. Thêm vào đó, mặc dù không được thể hiện, quy trình để dẫn xuất mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ dự đoán của thiết bị mã hóa, quy trình để dẫn xuất mẫu phần dư cho

khối bỏ qua biến đổi dựa trên mẫu gốc và mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ trừ của thiết bị mã hóa, và quy trình để tạo ra mẫu được tái dựng và hình ảnh được tái dựng cho khối bỏ qua biến đổi dựa trên mẫu phần dư và mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ cộng của thiết bị mã hóa.

Thiết bị mã hóa mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không (S600). Thiết bị mã hóa có thể mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không. Thông tin ảnh có thể bao gồm cờ được phép ẩn dữ liệu về dấu. Ví dụ, thiết bị mã hóa có thể xác định liệu việc ẩn dữ liệu về dấu được cho phép cho các khối của các hình ảnh theo trình tự, có thể mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép. Ví dụ, cờ được phép ẩn dữ liệu về dấu có thể là cờ cho việc liệu việc ẩn dữ liệu về dấu được cho phép. Ví dụ, cờ được phép ẩn dữ liệu về dấu có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép. Tức là, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép cho các khối của các hình ảnh trong trình tự. Ví dụ, cờ được phép ẩn dữ liệu về dấu có thể chỉ báo liệu cờ được sử dụng cho việc ẩn dữ liệu về dấu chỉ báo liệu việc ẩn dữ liệu về dấu được sử dụng cho lát hiện tại có thể là có mặt hay không. Ví dụ, cờ được phép ẩn dữ liệu về dấu mà có giá trị là 1 có thể chỉ báo rằng việc ẩn dữ liệu về dấu được cho phép, trong khi cờ được phép ẩn dữ liệu về dấu là có giá trị là 0 có thể chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép. Ví dụ, cờ được phép ẩn dữ liệu về dấu mà có giá trị là 1 có thể chỉ báo rằng cờ về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng cho có thể là có mặt, trong khi cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0 có thể chỉ báo rằng cờ về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng là không có mặt. Cũng như vậy, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể được báo hiệu trong cú pháp tập tham số trình tự (sequence parameter set, SPS). Thay vào đó, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể được báo hiệu trong cú pháp tiêu đề hình ảnh hoặc cú pháp tiêu đề lát. Phần tử cú pháp của cờ được phép ẩn dữ liệu về dấu có thể là `sps_sign_data_hiding_enabled_flag` được mô tả nêu trên.

Thiết bị mã hóa mã hóa cờ được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến

đôi trong lát hiện tại dựa trên cờ được phép ẩn dữ liệu về dấu hay không (S610). Thông tin ảnh có thể bao gồm cờ được phép TSRC.

Ví dụ, thiết bị mã hóa có thể mã hóa cờ được phép TSRC dựa trên cờ được phép ẩn dữ liệu về dấu. Ví dụ, cờ được phép TSRC có thể được mã hóa dựa trên cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (tức là, khi cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), thì cờ được phép TSRC có thể được mã hóa. Nói cách khác, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (tức là, khi cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), thì cờ được phép TSRC có thể được báo hiệu. Cũng như vậy, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, thì cờ được phép TSRC có thể không được mã hóa, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0 trong thiết bị giải mã. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, thì cờ được phép TSRC có thể không được báo hiệu, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0 trong thiết bị giải mã.

Ở đây, ví dụ, cờ được phép TSRC có thể là cờ về việc liệu TSRC có được cho phép hay không. Tức là, ví dụ, cờ được phép TSRC có thể là cờ chỉ báo liệu TSRC được cho phép cho các khối trong lát hay không. Nói cách khác, ví dụ, cờ được phép TSRC có thể là cờ chỉ báo liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hay không. Ở đây, khối có thể là khối lập mã (coding block, CB) hoặc khối biến đổi (transform block, TB). Ví dụ, cờ được phép TSRC mà có giá trị là 1 có thể thể hiện rằng TSRC là không được cho phép, và cờ được phép TSRC mà có giá trị là 0 có thể thể hiện rằng TSRC là được cho phép. Cũng như vậy, ví dụ, cờ được phép TSRC có thể được báo hiệu trong cú pháp phần đầu lát. Phần tử cú pháp của cờ được phép TSRC có thể là `sh_ts_residual_coding_disabled_flag` được mô tả ở trên. Cờ được phép TSRC có thể được đề cập đến như cờ không được phép TSRC.

Trong khi đó, ví dụ, thiết bị mã hóa có thể xác định xem liệu việc lượng tử hóa phụ thuộc có được cho phép hay không cho các khối của các ảnh trong trình tự, và có thể mã hóa cờ được phép lượng tử hóa phụ thuộc cho việc liệu việc lượng tử hóa phụ thuộc có được cho phép hay không. Thông tin hình ảnh có thể gồm cờ được phép lượng tử hóa phụ thuộc. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể là cờ cho việc liệu việc

lượng tử hóa phụ thuộc có được cho phép hay không. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể chỉ báo liệu việc lượng tử hóa phụ thuộc có được cho phép hay không. Nghĩa là, ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể biểu diễn liệu việc lượng tử hóa phụ thuộc có được cho phép hay không cho các khối của các hình ảnh trong trình tự. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể chỉ báo liệu cờ được sử dụng việc lượng tử hóa phụ thuộc chỉ báo liệu việc lượng tử hóa phụ thuộc có được sử dụng cho lát hiện tại hay không có thể có mặt hay không. Ví dụ, cờ được phép lượng tử hóa phụ thuộc mà có giá trị là 1 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc là được cho phép, và cờ được phép lượng tử hóa phụ thuộc mà có giá trị là 0 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc là không được cho phép. Cũng như vậy, ví dụ, cờ được phép việc lượng tử hóa phụ thuộc có thể được báo hiệu trong cú pháp SPS, cú pháp tiêu đề lát hoặc tương tự. Phần tử cú pháp của cờ được phép việc lượng tử hóa phụ thuộc có thể là `sps_dep_quant_enabled_flag` được mô tả nêu trên.

Cũng như vậy, ví dụ, thiết bị mã hóa có thể mã hóa cờ được phép việc bỏ qua biến đổi cho việc liệu việc bỏ qua biến đổi được cho phép cho lát hiện tại hay không. Thông tin ảnh có thể bao gồm cờ được phép việc bỏ qua biến đổi. Ví dụ, thiết bị mã hóa có thể xác định liệu việc bỏ qua biến đổi được cho phép cho các khối của các hình ảnh trong trình tự hay không, và có thể mã hóa cờ được phép việc bỏ qua biến đổi cho việc liệu việc bỏ qua biến đổi được cho phép. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể là cờ cho việc liệu việc bỏ qua biến đổi được cho phép hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể chỉ báo liệu việc bỏ qua biến đổi được cho phép hay không. Tức là, ví dụ, cờ được phép việc bỏ qua biến đổi có thể chỉ báo liệu việc bỏ qua biến đổi được cho phép cho các khối của các hình ảnh trong trình tự hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể chỉ báo liệu cờ bỏ qua biến đổi là có mặt hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi là có giá trị là 1 có thể chỉ báo rằng việc bỏ qua biến đổi được cho phép, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 0 có thể chỉ báo rằng việc bỏ qua biến đổi không được cho phép. Tức là, ví dụ, cờ được phép việc bỏ qua biến đổi mà có giá trị là 1 có thể chỉ báo rằng cờ bỏ qua biến đổi có thể là có mặt, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 0 có thể chỉ báo rằng cờ bỏ qua biến đổi là không có mặt. Cũng như vậy, ví dụ, cờ được phép việc bỏ qua biến đổi có thể được báo hiệu trong cú pháp tập tham số trình tự (sequence parameter set,

SPS). Phần tử cú pháp của cờ được phép việc bỏ qua biến đổi có thể là `sps_transform_skip_enabled_flag` được mô tả nêu trên.

Cũng như vậy, ví dụ, cờ được phép TSRC có thể được mã hóa dựa trên cờ được phép ẩn dữ liệu về dấu và/hoặc cờ được phép việc bỏ qua biến đổi. Ví dụ, cờ được phép TSRC có thể được mã hóa dựa trên cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 1. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (Tức là, cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), và giá trị của cờ được phép việc bỏ qua biến đổi là 1 (tức là, khi cờ được phép việc bỏ qua biến đổi chỉ báo rằng việc bỏ qua biến đổi được cho phép), thì cờ được phép TSRC có thể được mã hóa được (hoặc được báo hiệu). Cũng như vậy, ví dụ, khi giá trị của cờ được phép việc bỏ qua biến đổi là 0, cờ được phép TSRC có thể không được mã hóa, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0. Tức là, ví dụ, khi giá trị của cờ được phép việc bỏ qua biến đổi là 0, cờ được phép TSRC có thể không được báo hiệu, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0.

Thiết bị mã hóa mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC (S620). Thiết bị mã hóa có thể mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC.

Ví dụ, thiết bị mã hóa có thể xác định cú pháp lập mã phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC. Ví dụ, thiết bị mã hóa có thể xác định cú pháp lập mã phần dư cho khối bỏ qua biến đổi như một trong số cú pháp lập mã phần dư thông thường (Regular Residual Coding, RRC) và cú pháp luồng bit phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) dựa trên cờ được phép TSRC. Cú pháp RRC có thể chỉ báo cú pháp theo RRC, và cú pháp TSRC có thể chỉ báo cú pháp theo TSRC.

Ví dụ, cú pháp lập mã phần dư cho khối bỏ qua biến đổi có thể được xác định như cú pháp lập mã phần dư thông thường (RRC) dựa trên cờ được phép TSRC mà có giá trị là 1. Trong trường hợp này, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối bỏ qua biến đổi được bỏ qua biến đổi có thể được mã hóa, và giá trị của cờ bỏ qua biến đổi có thể là 1. Ví dụ, thông tin ảnh có thể bao gồm cờ bỏ qua biến đổi cho khối bỏ qua biến đổi. Cờ

bỏ qua biến đổi có thể chỉ báo liệu khối bỏ qua biến đổi được bỏ qua biến đổi hay không. Tức là, cờ bỏ qua biến đổi có thể chỉ báo liệu việc biến đổi được áp dụng cho các hệ số biến đổi của khối bỏ qua biến đổi hay không. Phần tử cú pháp biểu diễn cờ bỏ qua biến đổi có thể là `transform_skip_flag` được mô tả nêu trên. Ví dụ, khi giá trị của cờ bỏ qua biến đổi là 1, thì cờ bỏ qua biến đổi có thể chỉ báo rằng việc biến đổi không được áp dụng cho khối bỏ qua biến đổi (tức là, được bỏ qua biến đổi), trong khi đó, khi giá trị của cờ bỏ qua biến đổi là 0, thì cờ bỏ qua biến đổi có thể chỉ báo rằng việc biến đổi được áp dụng cho khối bỏ qua biến đổi. Ví dụ, khi khối trong lát hiện tại là khối bỏ qua biến đổi, thì giá trị của cờ bỏ qua biến đổi cho khối có thể là 1.

Cũng như vậy, ví dụ, cú pháp lập mã phần dư cho khối bỏ qua biến đổi có thể được xác định như cú pháp lập mã phần dư bỏ qua biến đổi (TSRC) dựa trên cờ được phép TSRC mà có giá trị là 0. Cũng như vậy, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối bỏ qua biến đổi được bỏ qua biến đổi có thể được mã hóa, và cú pháp lập mã phần dư cho khối bỏ qua biến đổi có thể được xác định như cú pháp lập mã bỏ qua biến đổi (TSRC) dựa trên cờ bỏ qua biến đổi mà có giá trị là 1 và cờ được phép TSRC mà có giá trị là 0. Cũng như vậy, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối được bỏ qua biến đổi có thể được mã hóa, và cú pháp lập mã phần dư cho khối có thể được xác định như cú pháp lập mã thông thường (RRC) dựa trên cờ bỏ qua biến đổi mà có giá trị là 0 và cờ được phép TSRC mà có giá trị là 0.

Sau đó, ví dụ, thiết bị mã hóa có thể mã hóa thông tin phần dư của cú pháp lập mã phần dư được xác định cho khối bỏ qua biến đổi. Thiết bị mã hóa có thể dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi, và có thể mã hóa thông tin phần dư của cú pháp lập mã phần dư được xác định cho mẫu phần dư của khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp lập mã phần dư thông thường (RRC) cho khối bỏ qua biến đổi có thể được mã hóa dựa trên cờ được phép TSRC mà có giá trị là 1, và thông tin phần dư của cú pháp TSRC cho khối bỏ qua biến đổi có thể được mã hóa dựa trên cờ được phép TSRC mà có giá trị là 0. Thông tin ảnh có thể bao gồm thông tin phần dư.

Ví dụ, thiết bị mã hóa có thể xác định liệu để thực hiện việc liên dự đoán hay nội dự đoán trong khối bỏ qua biến đổi, và có thể xác định chế độ liên dự đoán cụ thể hoặc chế độ nội dự đoán cụ thể dựa trên chi phí RD. Theo chế độ được xác định, thiết bị mã hóa có thể dẫn xuất các mẫu dự đoán cho khối bỏ qua biến đổi, và có thể dẫn xuất các

mẫu phần dư cho khối bỏ qua biến đổi thông qua phép trừ các mẫu dự đoán cho các mẫu gốc cho khối bỏ qua biến đổi.

Sau đó, ví dụ, thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi của khối bỏ qua biến đổi dựa trên các mẫu phần dư. Ví dụ, thiết bị mã hóa có thể xác định liệu việc biến đổi có được áp dụng cho khối bỏ qua biến đổi hay không. Nghĩa là, thiết bị mã hóa có thể xác định liệu việc biến đổi có được áp dụng cho các mẫu phần dư của khối bỏ qua biến đổi hay không. Thiết bị mã hóa có thể xác định xem có áp dụng việc biến đổi cho khối bỏ qua biến đổi hay không xem xét đến hiệu quả lập mã. Ví dụ, thiết bị mã hóa có thể xác định rằng không biến đổi được áp dụng cho khối bỏ qua biến đổi hay không. Trong khi đó, khối mà cho nó việc biến đổi không được áp dụng có thể được gọi là khối bỏ qua biến đổi.

Khi việc biến đổi không được áp dụng cho khối bỏ qua biến đổi, tức là, khi việc biến đổi không được áp dụng cho các mẫu phần dư, thì thiết bị mã hóa có thể dẫn xuất các mẫu phần dư được dẫn xuất như các hệ số biến đổi của khối bỏ qua biến đổi. Ngoài ra, khi việc biến đổi được áp dụng cho khối bỏ qua biến đổi, tức là, khi việc biến đổi được áp dụng cho các mẫu phần dư, thì thiết bị mã hóa có thể thực hiện việc biến đổi trên các mẫu phần dư để dẫn xuất các hệ số biến đổi của khối bỏ qua biến đổi. Khối bỏ qua biến đổi có thể bao gồm nhiều khối con hoặc nhóm hệ số (CG). Ngoài ra, kích thước của khối con của khối bỏ qua biến đổi có thể là kích thước 4x4 hoặc kích thước 2x2. Tức là, khối con của khối bỏ qua biến đổi có thể bao gồm lên đến 16 hệ số biến đổi khác không hoặc lên đến 4 hệ số biến đổi khác không. Ở đây, khối bỏ qua biến đổi có thể là khối lập mã (coding block, CB) hoặc khối biến đổi (transform block, TB). Cũng như vậy, hệ số biến đổi có thể được đề cập đến như hệ số phần dư.

Ví dụ, khi cú pháp lập mã phần dư cho khối bỏ qua biến đổi được xác định như cú pháp RRC, thì thiết bị mã hóa có thể mã hóa thông tin phần dư của cú pháp RRC cho khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp RRC có thể gồm các phần tử cú pháp được thể hiện trong Bảng 2 ở trên.

Ví dụ, thông tin phần dư của cú pháp RRC có thể bao gồm các phần tử cú pháp cho các hệ số biến đổi của khối bỏ qua biến đổi. Ở đây, hệ số biến đổi có thể được đề cập đến như hệ số phần dư.

Ví dụ, các phần tử cú pháp có thể bao gồm các phần tử cú pháp như `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, `last_sig_coeff_y_suffix`, `sb_coded_flag`, `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gtX_flag`, `abs_remainder`, `dec_abs_level`, và/hoặc `coeff_sign_flag`.

Cụ thể là, ví dụ, các phần tử cú pháp có thể gồm thông tin vị trí chỉ báo vị trí của hệ số biến đổi khác không cuối cùng trong mảng hệ số phần dư của khối bỏ qua biến đổi. Tức là, các phần tử cú pháp có thể bao gồm thông tin vị trí chỉ báo vị trí của hệ số biến đổi khác không cuối cùng theo thứ tự quét của khối bỏ qua biến đổi. Thông tin vị trí có thể bao gồm thông tin chỉ báo tiền tố của vị trí cột của hệ số biến đổi khác không cuối cùng, thông tin chỉ báo tiền tố của vị trí hàng của hệ số biến đổi khác không cuối cùng, thông tin chỉ báo hậu tố của vị trí cột của hệ số biến đổi khác không cuối cùng, và thông tin chỉ báo hậu tố của vị trí hàng của hệ số biến đổi khác không cuối cùng. Các phần tử cú pháp cho thông tin vị trí có thể là `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, và `last_sig_coeff_y_suffix`. Trong khi đó, hệ số biến đổi khác không có thể được đề cập đến như hệ số có nghĩa.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm cờ khối con được lập mã chỉ báo liệu khối con của khối bỏ qua biến đổi bao gồm hệ số biến đổi khác không; cờ hệ số có nghĩa chỉ báo liệu hệ số biến đổi của khối bỏ qua biến đổi là hệ số biến đổi khác không hay không; cờ mức hệ số thứ nhất cho việc liệu mức hệ số cho hệ số biến đổi là lớn hơn ngưỡng thứ nhất hay không; cờ mức chặn lẻ cho sự chặn lẻ của mức hệ số; và/hoặc cờ mức hệ số thứ hai cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ hai hay không. Ở đây, cờ khối con được lập mã có thể là `sb_coded_flag` hoặc `coded_sub_block_flag`; cờ hệ số có nghĩa có thể là `sig_coeff_flag`; cờ mức hệ số thứ nhất có thể là `abs_level_gt1_flag` hoặc `abs_level_gtx_flag`; cờ mức chặn lẻ có thể là `par_level_flag`; và cờ mức hệ số thứ hai có thể là `abs_level_gt3_flag` hoặc `abs_level_gtx_flag`.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm thông tin liên quan đến giá trị hệ số cho giá trị của hệ số biến đổi của khối bỏ qua biến đổi. Thông tin liên quan đến giá trị hệ số có thể là `abs_remainder` và/hoặc `dec_abs_level`.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm cờ về dấu chỉ báo dấu của hệ số biến đổi. Cờ dấu có thể là `coeff_sign_flag`.

Trong khi đó, ví dụ, khi việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi, thì cờ về dấu của hệ số biến đổi có nghĩa thứ nhất của nhóm hệ số hiện tại (CG) trong khối bỏ qua biến đổi có thể không được mã hóa và báo hiệu. Tức là, ví dụ, khi việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi, thì các phần tử cú pháp có thể không bao gồm cờ về dấu chỉ báo dấu của hệ số biến đổi có nghĩa thứ nhất. trong khi đó, ví dụ, liệu việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi có thể được dẫn xuất dựa trên cờ được phép ẩn dữ liệu về dấu hay không, và/hoặc vị trí của hệ số biến đổi có nghĩa thứ nhất và vị trí của hệ số biến đổi có nghĩa cuối cùng của CG hiện tại của khối bỏ qua biến đổi. Ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, và giá trị thu được bằng cách trừ vị trí hệ số biến đổi có nghĩa thứ nhất từ vị trí hệ số biến đổi có nghĩa cuối cùng là lớn hơn 3 (tức là, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, và số lượng các hệ số biến đổi có nghĩa trong CG hiện tại là lớn hơn 3), thì việc ẩn dữ liệu về dấu có thể được áp dụng cho CG hiện tại của khối bỏ qua biến đổi.

Thêm vào đó, ví dụ, khi cú pháp lập mã phần dư cho khối bỏ qua biến đổi được xác định như cú pháp TSRC, thì thiết bị mã hóa có thể mã hóa thông tin phần dư của cú pháp TSRC cho khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp TSRC có thể gồm các phần tử cú pháp được thể hiện trong Bảng 3 ở trên.

Ví dụ, thông tin phần dư của cú pháp TSRC có thể bao gồm các phần tử cú pháp cho các hệ số biến đổi của khối bỏ qua biến đổi. Ở đây, hệ số biến đổi có thể được đề cập đến như hệ số phần dư.

Ví dụ, các phần tử cú pháp có thể bao gồm các phần tử cú pháp được lập mã ngữ cảnh và/hoặc các phần tử cú pháp được lập mã vòng cho hệ số biến đổi. Các phần tử cú pháp có thể bao gồm các phần tử cú pháp như `sig_coeff_flag`, `coeff_sign_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gtX_flag` và/hoặc `abs_remainder`.

Ví dụ, các phần tử cú pháp được lập mã ngữ cảnh cho hệ số biến đổi có thể gồm cờ hệ số có nghĩa chỉ báo liệu hệ số biến đổi là hệ số biến đổi khác không hay không, cờ dấu chỉ báo dấu của hệ số biến đổi, cờ mức hệ số thứ nhất cho việc liệu mức hệ số cho hệ số biến đổi có lớn hơn ngưỡng thứ nhất hay không, và/hoặc cờ mức chặn lẻ cho

tính chẵn lẻ của mức hệ số cho hệ số biến đổi. Bên cạnh đó, ví dụ, các phần tử cú pháp được lập mã ngữ cảnh có thể bao gồm cờ mức hệ số thứ hai cho việc liệu mức hệ số của hệ số biến đổi có lớn hơn ngưỡng thứ hai hay không, cờ mức hệ số thứ ba cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ ba hay không, cờ mức hệ số thứ tư cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ tư hay không, và/hoặc cờ mức hệ số thứ năm cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ năm hay không. Ở đây, cờ hệ số có nghĩa có thể là `sig_coeff_flag`; cờ về dấu có thể là `coeff_sign_flag`; cờ mức hệ số thứ nhất có thể là `abs_level_gt1_flag`; và cờ mức chẵn lẻ có thể là `par_level_flag`. Ngoài ra, cờ mức hệ số thứ hai có thể là `abs_level_gt3_flag` hoặc `abs_level_gtx_flag`; cờ mức hệ số thứ ba có thể là `abs_level_gt5_flag` hoặc `abs_level_gtx_flag`; cờ mức hệ số thứ tư có thể là `abs_level_gt7_flag` hoặc `abs_level_gtx_flag`; và cờ mức hệ số thứ năm có thể là `abs_level_gt9_flag` hoặc `abs_level_gtx_flag`.

Cũng như vậy, ví dụ, các phần tử cú pháp được lập mã vòng cho hệ số biến đổi có thể bao gồm thông tin mức hệ số cho giá trị (hoặc mức hệ số) của hệ số biến đổi, và/hoặc cờ về dấu chỉ báo dấu cho hệ số biến đổi. Thông tin mức hệ số có thể là `abs_remainder` và/hoặc `dec_abs_level`, và cờ dấu có thể là `coeff_sign_flag`.

Thiết bị mã hóa tạo ra luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ được phép TSRC và thông tin phần dư (S630). Ví dụ, thiết bị mã hóa có thể xuất ra, như luồng bit, thông tin ảnh bao gồm cờ được phép ẩn dữ liệu về dấu, cờ được phép TSRC, và thông tin phần dư. Luồng bit có thể bao gồm cờ được phép ẩn dữ liệu về dấu, cờ được phép TSRC, và thông tin phần dư. Ngoài ra, luồng bit có thể còn bao gồm cờ được phép việc lượng tử hóa phụ thuộc và/hoặc cờ được phép việc bỏ qua biến đổi.

Cùng lúc đó, thông tin hình ảnh có thể gồm thông tin liên quan đến việc dự đoán cho khối bỏ qua biến đổi. Thông tin liên quan đến dự đoán có thể bao gồm thông tin chế độ dự đoán về chế độ liên dự đoán hoặc chế độ nội dự đoán được thực hiện trên khối bỏ qua biến đổi.

Trong khi đó, luồng bit có thể được truyền đến thiết bị giải mã thông qua mạng hoặc phương tiện lưu trữ (kỹ thuật số). Ở đây, mạng có thể gồm mạng phát rộng, mạng truyền thông và/hoặc tương tự, và các phương tiện lưu trữ kỹ thuật số có thể gồm các

phương tiện lưu trữ khác nhau, như là bus nối tiếp đa năng (Universal Serial Bus, USB), bảo mật kỹ thuật số (Secure Digital, SD), đĩa compact (Compact Disk, CD), đĩa video kỹ thuật số (Digital Video Disk, DVD), Blu-ray, ổ đĩa đĩa cứng (Hard Disk Drive, HDD), ổ đĩa trạng thái rắn (Solid State Drive, SSD), và tương tự.

Fig.7 minh họa một cách vắn tắt thiết bị mã hóa để thực hiện phương pháp mã hóa ảnh theo sáng chế. Phương pháp được bộc lộ trên Fig.6 có thể được thực hiện bởi thiết bị mã hóa được bộc lộ trên Fig.7. Cụ thể, ví dụ, bộ mã hóa entropy của thiết bị mã hóa của Fig.7 có thể thực hiện S600 đến S630 trên Fig.6. Thêm vào đó, mặc dù không được thể hiện, quy trình để dẫn xuất mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ dự đoán của thiết bị mã hóa, quy trình để dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên mẫu gốc và mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ trừ của thiết bị mã hóa, và quy trình để tạo ra mẫu được tái dựng và hình ảnh được tái dựng cho khối bỏ qua biến đổi dựa trên mẫu phần dư và mẫu dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ cộng của thiết bị mã hóa.

Fig.8 minh họa một cách vắn tắt phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã theo sáng chế. Phương pháp được bộc lộ trên Fig.8 có thể được thực hiện bởi thiết bị giải mã được bộc lộ ở Fig.3. Cụ thể, ví dụ, S800 đến S820 của Fig.8 có thể được thực hiện bởi bộ giải mã entropy của thiết bị giải mã, S830 của Fig.8 có thể được thực hiện bởi bộ xử lý phần dư của thiết bị giải mã, và S840 có thể được thực hiện bởi bộ cộng của thiết bị giải mã. Ngoài ra, mặc dù không được thể hiện, quy trình nhận thông tin dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bởi bộ giải mã entropi của thiết bị giải mã, và quy trình dẫn xuất mẫu dự đoán của khối bỏ qua biến đổi có thể được thực hiện bởi bộ dự đoán của thiết bị giải mã.

Thiết bị giải mã thu được cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép cho lát hiện tại hay không (S800). Thiết bị giải mã có thể thu được thông tin ảnh bao gồm cờ được phép ẩn dữ liệu về dấu thông qua luồng bit. Thông tin ảnh có thể bao gồm cờ được phép ẩn dữ liệu về dấu. Ví dụ, cờ được phép ẩn dữ liệu về dấu có thể là cờ cho việc liệu việc ẩn dữ liệu về dấu được cho phép. Ví dụ, cờ được phép ẩn dữ liệu về dấu có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép. Tức là, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể chỉ báo liệu việc ẩn dữ liệu về dấu được cho phép cho các khối của các hình ảnh trong trình tự. Ví dụ, cờ được phép ẩn dữ liệu

về dấu có thể chỉ báo liệu cờ được sử dụng cho việc ẩn dữ liệu về dấu chỉ báo liệu việc ẩn dữ liệu về dấu được sử dụng cho lát hiện tại có thể là có mặt hay không. Ví dụ, cờ được phép ẩn dữ liệu về dấu mà có giá trị là 1 có thể chỉ báo rằng việc ẩn dữ liệu về dấu được cho phép, trong khi cờ được phép ẩn dữ liệu về dấu là có giá trị là 0 có thể chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép. Ví dụ, cờ được phép ẩn dữ liệu về dấu mà có giá trị là 1 có thể chỉ báo rằng cờ về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng cho có thể là có mặt, trong khi cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0 có thể chỉ báo rằng cờ về dấu mà việc ẩn dữ liệu về dấu đã được áp dụng là không có mặt. Cũng như vậy, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể được báo hiệu trong cú pháp tập tham số trình tự (sequence parameter set, SPS). Thay vào đó, ví dụ, cờ được phép ẩn dữ liệu về dấu có thể được báo hiệu trong cú pháp tiêu đề hình ảnh hoặc cú pháp tiêu đề lát. Phần tử cú pháp của cờ được phép ẩn dữ liệu về dấu có thể là `sps_sign_data_hiding_enabled_flag` được mô tả nêu trên.

Thiết bị giải mã thu được cờ được phép lập mã phần dư bỏ qua biến đổi (TSRC) cho việc liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hiện tại hay không (S810). Thông tin ảnh có thể bao gồm cờ được phép TSRC.

Ví dụ, thiết bị giải mã có thể thu được cờ được phép TSRC dựa trên cờ được phép ẩn dữ liệu về dấu. Ví dụ, cờ được phép TSRC có thể thu được dựa trên cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (tức là, khi cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), thì cờ được phép TSRC có thể thu được. Nói cách khác, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (tức là, khi cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), thì cờ được phép TSRC có thể được báo hiệu. Cũng như vậy, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, thì cờ được phép TSRC có thể không thu được, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, thì cờ được phép TSRC có thể không được báo hiệu, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0.

Ở đây, ví dụ, cờ được phép TSRC có thể là cờ về việc liệu TSRC có được cho phép hay không. Tức là, ví dụ, cờ được phép TSRC có thể là cờ chỉ báo liệu TSRC được cho phép cho các khối trong lát hay không. Nói cách khác, ví dụ, cờ được phép TSRC có thể

là cờ chỉ báo liệu TSRC được cho phép cho khối bỏ qua biến đổi trong lát hay không. Ở đây, khối có thể là khối lập mã (coding block, CB) hoặc khối biến đổi (transform block, TB). Ví dụ, cờ được phép TSRC mà có giá trị là 1 có thể thể hiện rằng TSRC là không được cho phép, và cờ được phép TSRC mà có giá trị là 0 có thể thể hiện rằng TSRC là được cho phép. Cũng như vậy, ví dụ, cờ được phép TSRC có thể được báo hiệu trong cú pháp tiêu đề lát. Phần tử cú pháp của cờ được phép TSRC có thể là `sh_ts_residual_coding_disabled_flag` được mô tả ở trên. Cờ được phép TSRC có thể được đề cập đến như cờ không được phép TSRC.

Trong khi đó, ví dụ, thiết bị giải mã có thể thu được cờ được phép việc lượng tử hóa phụ thuộc. Thiết bị giải mã có thể thu nhận thông tin hình ảnh gồm cờ được phép lượng tử hóa phụ thuộc thông qua luồng bit. Thông tin hình ảnh có thể gồm cờ được phép lượng tử hóa phụ thuộc. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể là cờ cho việc liệu việc lượng tử hóa phụ thuộc có được cho phép hay không. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể chỉ báo liệu việc lượng tử hóa phụ thuộc có được cho phép hay không. Nghĩa là, ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể biểu diễn liệu việc lượng tử hóa phụ thuộc có được cho phép hay không cho các khối của các hình ảnh trong trình tự. Ví dụ, cờ được phép lượng tử hóa phụ thuộc có thể chỉ báo liệu cờ được sử dụng việc lượng tử hóa phụ thuộc chỉ báo liệu việc lượng tử hóa phụ thuộc có được sử dụng cho lát hiện tại hay không có thể có mặt hay không. Ví dụ, cờ được phép lượng tử hóa phụ thuộc mà có giá trị là 1 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc là được cho phép, và cờ được phép lượng tử hóa phụ thuộc mà có giá trị là 0 có thể chỉ báo rằng việc lượng tử hóa phụ thuộc là không được cho phép. Cũng như vậy, ví dụ, cờ được phép việc lượng tử hóa phụ thuộc có thể được báo hiệu trong cú pháp SPS, cú pháp tiêu đề lát hoặc tương tự. Phần tử cú pháp của cờ được phép việc lượng tử hóa phụ thuộc có thể là `sps_dep_quant_enabled_flag` được mô tả nêu trên.

Cũng như vậy, ví dụ, thiết bị giải mã có thể thu được cờ được phép việc bỏ qua biến đổi. Thiết bị giải mã có thể thu được thông tin ảnh bao gồm cờ được phép việc bỏ qua biến đổi thông qua luồng bit. Thông tin ảnh có thể bao gồm cờ được phép việc bỏ qua biến đổi. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể là cờ cho việc liệu việc bỏ qua biến đổi được cho phép hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể chỉ báo liệu việc bỏ qua biến đổi được cho phép hay không. Tức là, ví dụ, cờ được

phép việc bỏ qua biến đổi có thể chỉ báo liệu việc bỏ qua biến đổi được cho phép cho các khối của các hình ảnh trong trình tự hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi có thể chỉ báo liệu cờ bỏ qua biến đổi là có mặt hay không. Ví dụ, cờ được phép việc bỏ qua biến đổi là có giá trị là 1 có thể chỉ báo rằng việc bỏ qua biến đổi được cho phép, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 0 có thể chỉ báo rằng việc bỏ qua biến đổi không được cho phép. Tức là, ví dụ, cờ được phép việc bỏ qua biến đổi mà có giá trị là 1 có thể chỉ báo rằng cờ bỏ qua biến đổi có thể là có mặt, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 0 có thể chỉ báo rằng cờ bỏ qua biến đổi là không có mặt. Cũng như vậy, ví dụ, cờ được phép việc bỏ qua biến đổi có thể được báo hiệu trong cú pháp tập tham số trình tự (sequence parameter set, SPS). Phần tử cú pháp của cờ được phép việc bỏ qua biến đổi có thể là `sps_transform_skip_enabled_flag` được mô tả nêu trên.

Cũng như vậy, ví dụ, cờ được phép TSRC có thể thu được dựa trên cờ được phép ẩn dữ liệu về dấu và/hoặc cờ được phép việc bỏ qua biến đổi. Ví dụ, cờ được phép TSRC có thể thu được dựa trên cờ được phép ẩn dữ liệu về dấu mà có giá trị là 0, và cờ được phép việc bỏ qua biến đổi mà có giá trị là 1. Tức là, ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 0 (Tức là, cờ được phép ẩn dữ liệu về dấu chỉ báo rằng việc ẩn dữ liệu về dấu không được cho phép), và giá trị của cờ được phép việc bỏ qua biến đổi là 1 (tức là, khi cờ được phép việc bỏ qua biến đổi chỉ báo rằng việc bỏ qua biến đổi được cho phép), thì cờ được phép TSRC có thể thu được (hoặc được báo hiệu). Cũng như vậy, ví dụ, khi giá trị của cờ được phép việc bỏ qua biến đổi là 0, cờ được phép TSRC có thể không thu được, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0. Tức là, ví dụ, khi giá trị của cờ được phép việc bỏ qua biến đổi là 0, cờ được phép TSRC có thể không được báo hiệu, và giá trị của cờ được phép TSRC có thể được dẫn xuất như 0.

Thiết bị giải mã thu được thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC (S820). Thiết bị giải mã có thể thu được thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ được phép TSRC.

Ví dụ, thiết bị giải mã có thể xác định cú pháp lập mã phần dư cho khối bỏ qua biến đổi trong lát hiện tại dựa trên cờ được phép TSRC. Ví dụ, thiết bị giải mã có thể xác định cú pháp lập mã phần dư cho khối bỏ qua biến đổi như một trong số cú pháp lập mã phần dư thông thường (Regular Residual Coding, RRC) và cú pháp luồng bit phần

được bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) dựa trên cờ được phép TSRC. Cú pháp RRC có thể chỉ báo cú pháp theo RRC, và cú pháp TSRC có thể chỉ báo cú pháp theo TSRC.

Ví dụ, cú pháp lập mã phần dư cho khối bỏ qua biến đổi trong lát hiện tại có thể được xác định như cú pháp lập mã phần dư thông thường (RRC) dựa trên cờ được phép TSRC mà có giá trị là 1. Trong trường hợp này, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối bỏ qua biến đổi là được bỏ qua biến đổi hay không có thể thu được dựa trên cờ được phép việc bỏ qua biến đổi mà có giá trị là 1, và giá trị của cờ bỏ qua biến đổi có thể là 1. Ví dụ, thông tin ảnh có thể bao gồm cờ bỏ qua biến đổi cho khối bỏ qua biến đổi. Cờ bỏ qua biến đổi có thể chỉ báo liệu khối bỏ qua biến đổi được bỏ qua biến đổi hay không. Tức là, cờ bỏ qua biến đổi có thể chỉ báo liệu việc biến đổi được áp dụng cho các hệ số biến đổi của khối bỏ qua biến đổi hay không. Phần tử cú pháp biểu diễn cờ bỏ qua biến đổi có thể là `transform_skip_flag` được mô tả nêu trên. Ví dụ, khi giá trị của cờ bỏ qua biến đổi là 1, thì cờ bỏ qua biến đổi có thể chỉ báo rằng việc biến đổi không được áp dụng cho khối bỏ qua biến đổi (tức là, được bỏ qua biến đổi), trong khi đó, khi giá trị của cờ bỏ qua biến đổi là 0, thì cờ bỏ qua biến đổi có thể chỉ báo rằng việc biến đổi được áp dụng cho khối bỏ qua biến đổi. Ví dụ, giá trị của cờ bỏ qua biến đổi cho khối bỏ qua biến đổi có thể là 1.

Cũng như vậy, ví dụ, cú pháp lập mã phần dư cho khối bỏ qua biến đổi có thể được xác định như cú pháp lập mã phần dư bỏ qua biến đổi (TSRC) dựa trên cờ được phép TSRC mà có giá trị là 0. Cũng như vậy, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối bỏ qua biến đổi được bỏ qua biến đổi có thể thu được, và cú pháp lập mã phần dư cho khối bỏ qua biến đổi có thể được xác định như cú pháp lập mã bỏ qua biến đổi (TSRC) dựa trên cờ bỏ qua biến đổi mà có giá trị là 1 và cờ được phép TSRC mà có giá trị là 0. Cũng như vậy, ví dụ, cờ bỏ qua biến đổi cho việc liệu khối hiện tại được bỏ qua biến đổi có thể thu được, và cú pháp lập mã phần dư cho khối hiện tại có thể được xác định như cú pháp lập mã thông thường (RRC) dựa trên cờ bỏ qua biến đổi mà có giá trị là 0 và cờ được phép TSRC mà có giá trị là 0.

Sau đó, ví dụ, thiết bị giải mã có thể thu nhận thông tin phần dư của cú pháp lập mã phần dư được xác định cho khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp lập mã phần dư thông thường (RRC) có thể thu được dựa trên cờ được phép TSRC

mà có giá trị là 1, và thông tin phần dư của cú pháp TSRC có thể thu được dựa trên cờ được phép TSRC mà có giá trị là 0. Thông tin ảnh có thể bao gồm thông tin phần dư.

Ví dụ, khi cú pháp lập mã phần dư cho khối bỏ qua biến đổi được xác định như cú pháp RRC, thì thiết bị giải mã có thể thu được thông tin phần dư của cú pháp RRC cho khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp RRC có thể gồm các phần tử cú pháp được thể hiện trong Bảng 2 ở trên.

Ví dụ, thông tin phần dư của cú pháp RRC có thể bao gồm các phần tử cú pháp cho các hệ số biến đổi của khối bỏ qua biến đổi. Ở đây, hệ số biến đổi có thể được đề cập đến như hệ số phần dư.

Ví dụ, các phần tử cú pháp có thể bao gồm các phần tử cú pháp như `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, `last_sig_coeff_y_suffix`, `sb_coded_flag`, `sig_coeff_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gtX_flag`, `abs_remainder`, `dec_abs_level`, và/hoặc `coeff_sign_flag`.

Cụ thể là, ví dụ, các phần tử cú pháp có thể gồm thông tin vị trí chỉ báo vị trí của hệ số biến đổi khác không cuối cùng trong mảng hệ số phần dư của khối bỏ qua biến đổi. Tức là, các phần tử cú pháp có thể bao gồm thông tin vị trí chỉ báo vị trí của hệ số biến đổi khác không cuối cùng theo thứ tự quét của khối bỏ qua biến đổi. Thông tin vị trí có thể bao gồm thông tin chỉ báo tiền tố của vị trí cột của hệ số biến đổi khác không cuối cùng, thông tin chỉ báo tiền tố của vị trí hàng của hệ số biến đổi khác không cuối cùng, thông tin chỉ báo hậu tố của vị trí cột của hệ số biến đổi khác không cuối cùng, và thông tin chỉ báo hậu tố của vị trí hàng của hệ số biến đổi khác không cuối cùng. Các phần tử cú pháp cho thông tin vị trí có thể là `last_sig_coeff_x_prefix`, `last_sig_coeff_y_prefix`, `last_sig_coeff_x_suffix`, và `last_sig_coeff_y_suffix`. Trong khi đó, hệ số biến đổi khác không có thể được đề cập đến như hệ số có nghĩa.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm cờ khối con được lập mã chỉ báo liệu khối con của khối bỏ qua biến đổi bao gồm hệ số biến đổi khác không; cờ hệ số có nghĩa chỉ báo liệu hệ số biến đổi của khối bỏ qua biến đổi là hệ số biến đổi khác không hay không; cờ mức hệ số thứ nhất cho việc liệu mức hệ số cho hệ số biến đổi là lớn hơn ngưỡng thứ nhất hay không; cờ mức chuẩn lẻ cho sự chuẩn lẻ của mức hệ số;

và/hoặc cờ mức hệ số thứ hai cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ hai hay không. Ở đây, cờ khối con được lập mã có thể là `sb_coded_flag` hoặc `coded_sub_block_flag`; cờ hệ số có nghĩa có thể là `sig_coeff_flag`; cờ mức hệ số thứ nhất có thể là `abs_level_gt1_flag` hoặc `abs_level_gtx_flag`; cờ mức chẵn lẻ có thể là `par_level_flag`; và cờ mức hệ số thứ hai có thể là `abs_level_gt3_flag` hoặc `abs_level_gtx_flag`.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm thông tin liên quan đến giá trị hệ số cho giá trị của hệ số biến đổi của khối bỏ qua biến đổi. Thông tin liên quan đến giá trị hệ số có thể là `abs_remainder` và/hoặc `dec_abs_level`.

Cũng như vậy, ví dụ, các phần tử cú pháp có thể bao gồm cờ về dấu chỉ báo dấu của hệ số biến đổi. Cờ dấu có thể là `coeff_sign_flag`.

Trong khi đó, ví dụ, khi việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi, thì cờ về dấu của hệ số biến đổi có nghĩa thứ nhất của nhóm hệ số hiện tại (CG) trong khối bỏ qua biến đổi có thể không được báo hiệu. Tức là, ví dụ, khi việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi, thì các phần tử cú pháp có thể không bao gồm cờ về dấu chỉ báo dấu của hệ số biến đổi có nghĩa thứ nhất. Trong khi đó, ví dụ, liệu việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi có thể được dẫn xuất dựa trên cờ được phép ẩn dữ liệu về dấu hay không, và/hoặc vị trí của hệ số biến đổi có nghĩa thứ nhất và vị trí của hệ số biến đổi có nghĩa cuối cùng của CG hiện tại. Ví dụ, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, và giá trị thu được bằng cách trừ vị trí hệ số biến đổi có nghĩa thứ nhất từ vị trí hệ số biến đổi có nghĩa cuối cùng là lớn hơn 3 (tức là, khi giá trị của cờ được phép ẩn dữ liệu về dấu là 1, và số lượng các hệ số biến đổi có nghĩa trong CG hiện tại là lớn hơn 3), thì việc ẩn dữ liệu về dấu có thể được áp dụng cho CG hiện tại của khối bỏ qua biến đổi.

Thêm vào đó, ví dụ, khi cú pháp lập mã phần dư cho khối bỏ qua biến đổi được xác định như cú pháp TSRC, thì thiết bị giải mã có thể thu được thông tin phần dư của cú pháp TSRC cho khối bỏ qua biến đổi. Ví dụ, thông tin phần dư của cú pháp TSRC có thể gồm các phần tử cú pháp được thể hiện trong Bảng 3 ở trên.

Ví dụ, thông tin phần dư của cú pháp TSRC có thể bao gồm các phần tử cú pháp cho các hệ số biến đổi của khối bỏ qua biến đổi. Ở đây, hệ số biến đổi có thể được đề cập đến như hệ số phần dư.

Ví dụ, các phần tử cú pháp có thể bao gồm các phần tử cú pháp được lập mã ngữ cảnh và/hoặc các phần tử cú pháp được lập mã vòng cho hệ số biến đổi. Các phần tử cú pháp có thể bao gồm các phần tử cú pháp như `sig_coeff_flag`, `coeff_sign_flag`, `abs_level_gt1_flag`, `par_level_flag`, `abs_level_gtX_flag` và/hoặc `abs_remainder`.

Ví dụ, các phần tử cú pháp được lập mã ngữ cảnh cho hệ số biến đổi có thể gồm cờ hệ số có nghĩa chỉ báo liệu hệ số biến đổi là hệ số biến đổi khác không hay không, cờ dấu chỉ báo dấu của hệ số biến đổi, cờ mức hệ số thứ nhất cho việc liệu mức hệ số cho hệ số biến đổi có lớn hơn ngưỡng thứ nhất hay không, và/hoặc cờ mức chặn lẻ cho tính chặn lẻ của mức hệ số cho hệ số biến đổi. Bên cạnh đó, ví dụ, các phần tử cú pháp được lập mã ngữ cảnh có thể bao gồm cờ mức hệ số thứ hai cho việc liệu mức hệ số của hệ số biến đổi có lớn hơn ngưỡng thứ hai hay không, cờ mức hệ số thứ ba cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ ba hay không, cờ mức hệ số thứ tư cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ tư hay không, và/hoặc cờ mức hệ số thứ năm cho việc liệu mức hệ số của hệ số biến đổi là lớn hơn ngưỡng thứ năm hay không. Ở đây, cờ hệ số có nghĩa có thể là `sig_coeff_flag`; cờ về dấu có thể là `coeff_sign_flag`; cờ mức hệ số thứ nhất có thể là `abs_level_gt1_flag`; và cờ mức chặn lẻ có thể là `par_level_flag`. Ngoài ra, cờ mức hệ số thứ hai có thể là `abs_level_gt3_flag` hoặc `abs_level_gtx_flag`; cờ mức hệ số thứ ba có thể là `abs_level_gt5_flag` hoặc `abs_level_gtx_flag`; cờ mức hệ số thứ tư có thể là `abs_level_gt7_flag` hoặc `abs_level_gtx_flag`; và cờ mức hệ số thứ năm có thể là `abs_level_gt9_flag` hoặc `abs_level_gtx_flag`.

Cũng như vậy, ví dụ, các phần tử cú pháp được lập mã vòng cho hệ số biến đổi có thể bao gồm thông tin mức hệ số cho giá trị (hoặc mức hệ số) của hệ số biến đổi, và/hoặc cờ về dấu chỉ báo dấu cho hệ số biến đổi. Thông tin mức hệ số có thể là `abs_remainder` và/hoặc `dec_abs_level`, và cờ dấu có thể là `coeff_sign_flag`.

Thiết bị giải mã dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư (S830). Ví dụ, thiết bị giải mã có thể dẫn xuất các hệ số biến đổi của

khối bỏ qua biến đổi dựa trên thông tin phần dư, và có thể dẫn xuất các mẫu phần dư của khối bỏ qua biến đổi dựa trên các hệ số biến đổi.

Ví dụ, thiết bị giải mã có thể dẫn xuất các hệ số biến đổi của khối bỏ qua biến đổi dựa trên các phần tử cú pháp của thông tin phần dư. Sau đó, thiết bị giải mã có thể dẫn xuất các mẫu phần dư của khối bỏ qua biến đổi dựa trên các hệ số biến đổi. Ví dụ, khi dẫn xuất được rằng việc biến đổi không được áp dụng cho khối bỏ qua biến đổi dựa trên cờ bỏ qua biến đổi, nghĩa là, khi giá trị của cờ bỏ qua biến đổi là 1, thì thiết bị giải mã có thể dẫn xuất các hệ số biến đổi như các mẫu phần dư của khối bỏ qua biến đổi. Thay vào đó, ví dụ, khi được dẫn xuất mà không áp dụng việc biến đổi cho khối bỏ qua biến đổi dựa trên cờ bỏ qua biến đổi, tức là, khi giá trị của cờ bỏ qua biến đổi là 1, thì thiết bị giải mã có thể giải lượng tử hóa các hệ số biến đổi để dẫn xuất các mẫu phần dư của khối bỏ qua biến đổi. Thay vào đó, ví dụ, khi được dẫn xuất trong khi áp dụng việc biến đổi cho khối trong lát hiện tại dựa trên cờ bỏ qua biến đổi, tức là, khi giá trị của cờ bỏ qua biến đổi cho khối là 0, thì thiết bị giải mã có thể biến đổi ngược các hệ số biến đổi để dẫn xuất các mẫu phần dư của khối. Thay vào đó, ví dụ, khi được dẫn xuất trong khi áp dụng việc biến đổi cho khối dựa trên cờ bỏ qua biến đổi, tức là, khi giá trị của cờ bỏ qua biến đổi là 0, thì thiết bị giải mã có thể giải lượng tử hóa các hệ số biến đổi, và biến đổi ngược các hệ số biến đổi được giải lượng tử hóa để dẫn xuất các mẫu phần dư của khối hiện tại.

Trong khi đó, ví dụ, khi việc ẩn dữ liệu về dấu được áp dụng cho khối bỏ qua biến đổi, thì dấu của hệ số biến đổi có nghĩa thứ nhất của CG hiện tại trong khối bỏ qua biến đổi có thể được dẫn xuất dựa trên tổng của các giá trị tuyệt đối của các hệ số biến đổi có nghĩa trong CG hiện tại. Ví dụ, khi tổng của các giá trị tuyệt đối của các hệ số biến đổi có nghĩa là chẵn, thì dấu của hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị dương, trong khi, khi tổng của các giá trị tuyệt đối của các hệ số biến đổi có nghĩa là lẻ, thì dấu của hệ số biến đổi có nghĩa thứ nhất có thể được dẫn xuất như giá trị âm.

Thiết bị giải mã tạo ra ảnh được tái dựng dựa trên mẫu phần dư (S840). Ví dụ, thiết bị giải mã có thể tạo ra mẫu được tái dựng và/hoặc ảnh được tái dựng của khối bỏ qua biến đổi trong lát hiện tại dựa trên mẫu phần dư. Ví dụ, thiết bị giải mã có thể dẫn xuất mẫu dự đoán bằng cách thực hiện chế độ liên dự đoán hoặc chế độ nội dự đoán cho khối

bỏ qua biến đổi dựa trên thông tin dự đoán thu được qua luồng bit, và có thể tạo ra mẫu được dựng lại thông qua việc cộng mẫu dự đoán và mẫu phần dư.

Sau quá trình này, như đã mô tả trên đây, thì thủ tục lọc vòng lặp trong, chẳng hạn thủ tục ALF, SAO (Sample Adaptive Offset - dịch thích ứng mẫu) và/hoặc lọc khử khối, có thể được áp dụng, nếu cần, cho hình ảnh được tái tạo, để nâng cao chất lượng video chủ quan/khách quan.

Fig.9 minh họa một cách vắn tắt thiết bị giải mã để thực hiện phương pháp giải mã ảnh theo sáng chế. Phương pháp được bộc lộ trên Fig.8 có thể được thực hiện bởi thiết bị giải mã được bộc lộ ở Fig.9. Cụ thể, ví dụ, bộ giải mã entropy của thiết bị giải mã của Fig.9 có thể thực hiện S800 đến S820 của Fig.8; bộ xử lý phần dư của thiết bị giải mã của Fig.9 có thể thực hiện S830 của Fig.8; và bộ cộng của thiết bị giải mã của Fig.9 có thể thực hiện S840 của Fig.8. Thêm vào đó, mặc dù không được thể hiện, quy trình để nhận thông tin dự đoán cho khối bỏ qua biến đổi có thể được thực hiện bằng bộ giải mã entropy của thiết bị giải mã của Fig.9, và quy trình dẫn xuất mẫu dự đoán của khối hiện tại có thể được thực hiện bởi bộ dự đoán của thiết bị giải mã được thể hiện của Fig.9.

Theo tài liệu này, như được mô tả nêu trên, có thể để nâng cao hiệu quả của việc lập mã phần dư.

Thêm vào đó, theo tài liệu này, cờ được phép TSRC có thể được báo hiệu tùy thuộc vào cờ được phép ẩn dữ liệu về dấu, và thông qua điều này, hiệu quả lập mã có thể được nâng cao bằng việc ngăn việc ẩn dữ liệu về dấu không được sử dụng cho khối bỏ qua biến đổi mà TSRC không được cho phép cho, và hiệu quả lập mã phần dư tổng thể có thể được nâng cao bằng cách giảm lượng bit để được lập mã.

Thêm vào đó, theo tài liệu này, cờ được phép TSRC có thể được báo hiệu tùy thuộc vào việc cờ được phép bỏ qua biến đổi và cờ được phép ẩn dữ liệu về dấu, và thông qua điều này, hiệu quả lập mã có thể được nâng cao bằng việc ngăn việc ẩn dữ liệu về dấu không được sử dụng cho khối bỏ qua biến đổi mà TSRC không được cho phép cho, và hiệu quả lập mã phần dư tổng thể có thể được nâng cao bằng cách giảm lượng bit để được lập mã.

Theo phương án được mô tả nêu trên, các phương pháp được mô tả dựa trên sơ đồ tiến trình có chuỗi các bước hoặc khối. Sáng chế không được giới hạn ở thứ tự của các

bước hoặc các khối ở trên. Một số bước hoặc khối có thể xuất hiện đồng thời hoặc theo trật tự khác với các bước hoặc các khối khác được mô tả ở trên. Thêm nữa, người có hiểu biết trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng các bước được thể hiện trong sơ đồ tiến trình ở trên không phải là riêng biệt và các bước thêm nữa cũng thể được bao gồm, hoặc một hoặc nhiều bước trong các sơ đồ tiến trình làm ví dụ có thể được xóa mà không ảnh hưởng tới phạm vi bảo hộ của sáng chế này.

Các phương án được mô tả trong bản mô tả này có thể được thực hiện trên bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip. Ví dụ, các đơn vị chức năng được thể hiện ở mỗi hình vẽ có thể được thực hiện bằng cách được triển khai trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip. Trong trường hợp này, thông tin cho việc triển khai (ví dụ, thông tin về các lệnh) hoặc thuật toán có thể được lưu trữ trong phương tiện lưu trữ kỹ thuật số.

Ngoài ra, thiết bị giải mã và thiết bị mã hóa mà sáng chế được áp dụng với có thể được bao gồm trong thiết bị truyền/thu quảng bá đa phương tiện, thiết bị đầu cuối truyền thông di động, thiết bị video chiếu phim gia đình, thiết bị video chiếu phim số, camera giám sát, thiết bị trò chuyện video, thiết bị truyền thông thời gian thực như truyền thông video, thiết bị tạo luồng (streaming) di động, phương tiện lưu trữ, máy ghi hình cầm tay, thiết bị cung cấp dịch vụ video theo yêu cầu (Video on Demand, VoD), thiết bị cung cấp dịch vụ nội dung trên nền mạng viễn thông (Over The Top, OTT), thiết bị cung cấp dịch vụ tạo luồng Internet, thiết bị video ba chiều (Three Dimensional, 3D), thiết bị video dùng cho các cuộc họp tổ chức qua điện thoại, thiết bị người dùng cho vận tải (ví dụ, thiết bị người dùng cho xe cộ, thiết bị người dùng cho máy bay, thiết bị người dùng cho tàu, v.v.) và thiết bị video y tế và có thể được dùng để xử lý các tín hiệu video hoặc các tín hiệu dữ liệu. Ví dụ, thiết bị video của dịch vụ cung cấp nội dung trên nền mạng viễn thông (Over The Top, OTT) có thể bao gồm bảng điều khiển chơi trò chơi, máy đọc đĩa Blu-ray, ti vi truy cập Internet, hệ thống rạp hát gia đình, điện thoại thông minh, máy tính bảng, đầu ghi video kỹ thuật số (Digital Video Recorder, DVR), và tương tự.

Ngoài ra, phương pháp xử lý mà sáng chế được áp dụng với có thể được tạo ra dưới dạng chương trình mà để được thực thi bởi máy tính, và có thể được lưu trữ trên phương tiện ghi có thể đọc được bằng máy tính. Dữ liệu đa phương tiện mà có cấu trúc dữ liệu theo sáng chế cũng có thể được lưu trữ trên phương tiện ghi đọc được bởi máy

tính. Phương tiện ghi đọc được bởi máy tính bao gồm tất cả các loại thiết bị lưu trữ mà dữ liệu đọc được bởi máy tính được lưu trong đó. Phương tiện ghi đọc được bằng máy tính này có thể bao gồm BD, thiết bị buýt nối tiếp vạn năng (Universal Serial Bus, USB), ROM, PROM, EPROM, RAM, CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang học chẳng hạn. Hơn nữa, phương tiện ghi đọc được bởi máy tính chứa phương tiện được thực hiện ở dạng của các sóng bộ mang (ví dụ, truyền qua Internet). Hơn nữa, luồng bit được tạo ra bằng phương pháp mã hóa có thể được lưu trong phương tiện ghi đọc được bởi máy tính hoặc có thể được truyền qua các mạng truyền thông có dây hoặc không dây.

Ngoài ra, các phương án theo sáng chế có thể được thực hiện với sản phẩm chương trình máy tính bằng các mã chương trình, và các mã chương trình này có thể được thực hiện trên máy tính bởi các phương án theo sáng chế. Các mã chương trình có thể được lưu trên vật mang đọc được bởi máy tính.

Fig.10 minh họa sơ đồ cấu trúc của hệ thống tạo luồng nội dung mà sáng chế được áp dụng vào đó.

Hệ thống tạo luồng nội dung mà phương án thực hiện (các phương án thực hiện) của sáng chế này được áp dụng với, có thể phần lớn bao gồm máy chủ mã hóa, máy chủ tạo luồng, máy chủ web, bộ phận lưu trữ phương tiện, thiết bị người dùng, và thiết bị nhập vào đa phương tiện.

Máy chủ mã hóa nén nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện như điện thoại thông minh, camera, máy quay video, v.v. thành dữ liệu dạng số để tạo ra luồng bit và truyền luồng bit tới máy chủ tạo luồng. Như ví dụ khác, khi thiết bị đầu vào đa phương tiện như là điện thoại thông minh, camera, máy ghi hình cầm tay, v.v. trực tiếp tạo ra luồng bit, thì máy chủ mã hóa có thể được lược bỏ.

Luồng bit có thể được tạo ra bởi phương pháp tạo mã hoặc phương pháp tạo ra luồng bit mà phương án (các phương án) của sáng chế. này được áp dụng vào đó, và máy chủ phát luồng có thể tạm thời lưu trữ luồng bit trong quy trình truyền hoặc nhận luồng bit.

Máy chủ tạo luồng truyền dữ liệu đa phương tiện đến thiết bị người dùng dựa trên yêu cầu người dùng thông qua máy chủ web, và máy chủ web đóng vai trò là phương

tiện để thông báo cho người dùng về dịch vụ. Khi người dùng yêu cầu dịch vụ mong muốn từ máy chủ web, thì máy chủ web chuyển nó đến máy chủ tạo luồng, và máy chủ tạo luồng truyền dữ liệu đa phương tiện đến người dùng. Trong trường hợp này, hệ thống tạo luồng nội dung có thể gồm máy chủ điều khiển tách biệt. Trong trường hợp này, máy chủ điều khiển có chức năng để điều khiển mệnh lệnh/hỏi đáp giữa các thiết bị trong hệ thống tạo luồng các nội dung.

Máy chủ tạo luồng có thể thu các nội dung từ bộ lưu trữ phương tiện và/hoặc máy chủ mã hóa. Ví dụ, khi nội dung được nhận từ máy chủ mã hóa, thì nội dung có thể được nhận theo thời gian thực. Trong trường hợp này, để cung cấp dịch vụ tạo luồng trơn tru, thì máy chủ tạo luồng có thể lưu trữ luồng bit trong thời gian được xác định trước.

Các ví dụ của thiết bị người dùng có thể gồm điện thoại di động, điện thoại thông minh, máy tính xách tay, thiết bị đầu cuối phát rộng kỹ thuật số, thiết bị trợ giúp số cá nhân (Personal Digital Assistant, PDA), thiết bị chơi đa phương tiện di động (Portable Multimedia Player, PMP), thiết bị định vị, PC dạng ngói, các PC dạng bảng, các máy tính siêu di động, các thiết bị đeo được (ví dụ, các đồng hồ thông minh, các kính thông minh, các bộ hiển thị gắn trên đầu), các ti vi số, máy tính để bàn, bảng ký hiệu dạng số và tương tự. Mỗi máy chủ trong hệ thống tạo luồng nội dung có thể được vận hành như là máy chủ phân tán, mà trong trường hợp đó dữ liệu được thu từ từng máy chủ có thể được phân tán.

Các yêu cầu bảo hộ được mô tả trong sáng chế này có thể được tổ hợp theo các cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật của các yêu cầu bảo hộ dạng phương pháp của sáng chế có thể được tổ hợp và được triển khai dưới dạng thiết bị, và các dấu hiệu kỹ thuật của các yêu cầu bảo hộ dạng thiết bị của sáng chế có thể được tổ hợp và được triển khai dưới dạng phương pháp. Ngoài ra, các đặc điểm kỹ thuật của yêu cầu bảo hộ phương pháp của sáng chế và các đặc điểm kỹ thuật của yêu cầu bảo hộ thiết bị có thể được kết hợp để được thực hiện như thiết bị, và các đặc điểm kỹ thuật của yêu cầu bảo hộ phương pháp của sáng chế và các đặc điểm kỹ thuật của yêu cầu bảo hộ thiết bị có thể được kết hợp để được thực hiện như phương pháp.

YÊU CẦU BẢO HỘ

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

thu được cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép hay không;

thu được cờ không được phép việc lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được sử dụng cho khối bỏ qua biến đổi trong lát hiện tại hay không;

thu được thông tin lập mã phần dư cho khối bỏ qua biến đổi dựa trên cờ không được phép TSRC;

dẫn xuất mẫu phần dư cho khối bỏ qua biến đổi dựa trên thông tin lập mã phần dư; và

tạo ra hình ảnh được tái dựng dựa trên mẫu phần dư,

trong đó cờ không được phép TSRC thu được dựa trên cờ được phép ẩn dữ liệu về dấu.

2. Phương pháp theo điểm 1, trong đó cờ được phép ẩn dữ liệu về dấu bằng 1 biểu diễn rằng việc ẩn dữ liệu về dấu được cho phép, và

trong đó cờ được phép ẩn dữ liệu về dấu bằng 0 biểu diễn rằng việc ẩn dữ liệu về dấu không được cho phép.

3. Phương pháp theo điểm 2, trong đó, để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 1, dấu của hệ số biến đổi có nghĩa thứ nhất của nhóm hệ số (coefficient group, CG) hiện tại trong khối bỏ qua biến đổi được dẫn xuất dựa trên tổng của các giá trị tuyệt đối của các hệ số biến đổi có nghĩa trong CG hiện tại.

4. Phương pháp theo điểm 3, trong đó, để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 1, cờ về dấu cho hệ số biến đổi có nghĩa thứ nhất không được báo hiệu.

5. Phương pháp theo điểm 2, trong đó cờ không được phép TSRC thu được từ luồng bit để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 0.

6. Phương pháp theo điểm 5, trong đó, để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 1, thì cờ không được phép TSRC không thu được từ luồng bit, và giá trị của cờ không được phép TSRC được dẫn xuất như 0.

7. Phương pháp theo điểm 1, còn bao gồm bước thu được cờ được phép việc bỏ qua biến đổi cho việc liệu việc bỏ qua biến đổi được cho phép hay không,

trong đó cờ không được phép TSRC thu được dựa trên cờ được phép ẩn dữ liệu về dấu và cờ được phép việc bỏ qua biến đổi.

8. Phương pháp theo điểm 7, trong đó cờ không được phép TSRC thu được từ luồng bit để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 0 và giá trị của cờ được phép việc bỏ qua biến đổi là bằng 1.

9. Phương pháp theo điểm 8, trong đó, để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 1 hoặc giá trị của cờ được phép bỏ qua biến đổi là bằng 0, thì cờ không được phép TSRC không thu được từ luồng bit, và giá trị của cờ không được phép TSRC được dẫn xuất như 0.

10. Phương pháp mã hóa ảnh được thực hiện bởi thiết bị mã hóa, phương pháp này bao gồm các bước:

mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép hay không;

mã hóa, dựa trên cờ được phép ẩn dữ liệu về dấu, cờ không được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được sử dụng cho khối bỏ qua biến đổi trong lát hiện tại hay không;

mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ không được phép TSRC; và

tạo ra luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ không được phép TSRC, và thông tin phần dư.

11. Phương pháp theo điểm 10, trong đó cờ được phép ẩn dữ liệu về dấu bằng 1 biểu diễn rằng việc ẩn dữ liệu về dấu được cho phép, và

trong đó cờ được phép ẩn dữ liệu về dấu bằng 0 biểu diễn rằng việc ẩn dữ liệu về dấu không được cho phép.

12. Phương pháp theo điểm 11, trong đó cờ không được phép TSRC được mã hóa thành luồng bit để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 0.

13. Phương pháp theo điểm 12, trong đó cờ không được phép TSRC không được mã hóa thành luồng bit để phản hồi về giá trị của cờ được phép ẩn dữ liệu về dấu là bằng 1.

14. Phương pháp theo điểm 10, còn bao gồm bước mã hóa cờ được phép bỏ qua biến đổi cho việc liệu việc bỏ qua biến đổi được cho phép hay không,

trong đó cờ không được phép TSRC được mã hóa thành luồng bit dựa trên cờ được phép ẩn dữ liệu về dấu và cờ được phép việc bỏ qua biến đổi.

15. Phương tiện lưu trữ phi chuyển tiếp đọc được bởi máy tính để lưu trữ luồng bit được tạo ra bởi phương pháp mã hóa ảnh, phương pháp mã hóa ảnh này bao gồm các bước:

mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu có được cho phép hay không;

mã hóa, dựa trên cờ được phép ẩn dữ liệu về dấu, cờ không được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC có được sử dụng cho khối bỏ qua biến đổi trong lát hiện tại hay không; và

mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ không được phép TSRC,

trong đó luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ không được phép TSRC, và thông tin phần dư.

16. Phương pháp truyền dữ liệu cho thông tin ảnh bao gồm các bước:

mã hóa cờ được phép ẩn dữ liệu về dấu cho việc liệu việc ẩn dữ liệu về dấu được cho phép hay không;

mã hóa, dựa trên cờ được phép ẩn dữ liệu về dấu, cờ không được phép lập mã phần dư bỏ qua biến đổi (Transform Skip Residual Coding, TSRC) cho việc liệu TSRC được sử dụng cho khối bỏ qua biến đổi trong lát hiện tại hay không;

mã hóa thông tin phần dư cho khối bỏ qua biến đổi dựa trên cờ không được phép TSRC;

tạo ra luồng bit bao gồm cờ được phép ẩn dữ liệu về dấu, cờ không được phép TSRC, và thông tin phần dư; và
truyền dữ liệu bao gồm luồng bit.

FIG. 1

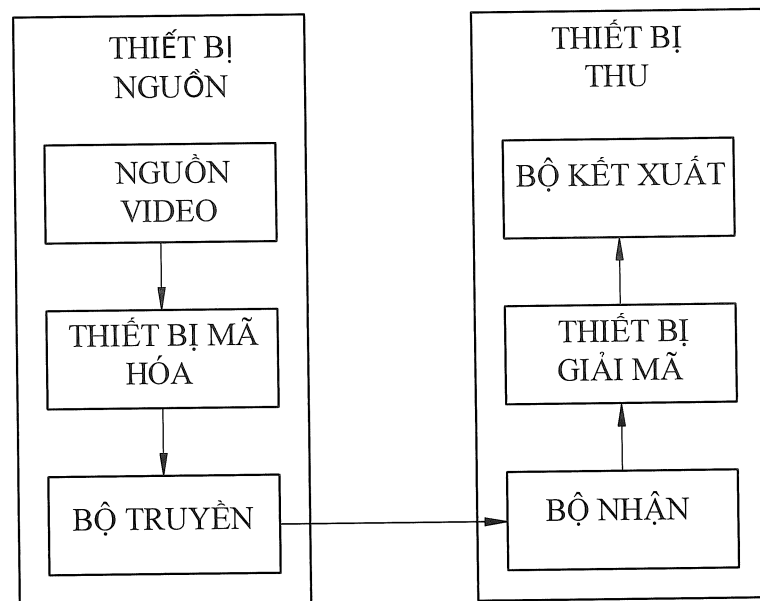


FIG. 2

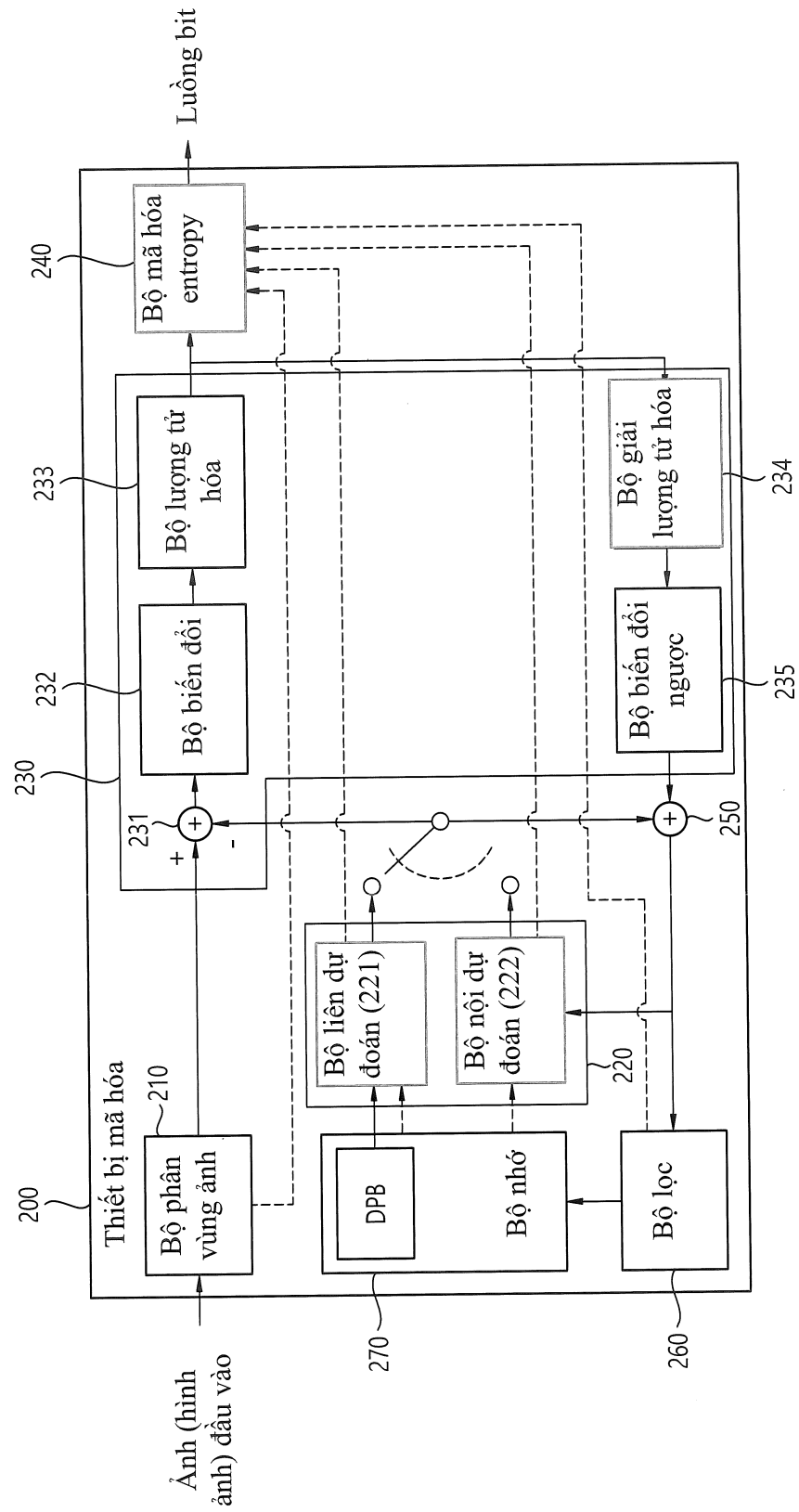


FIG. 3

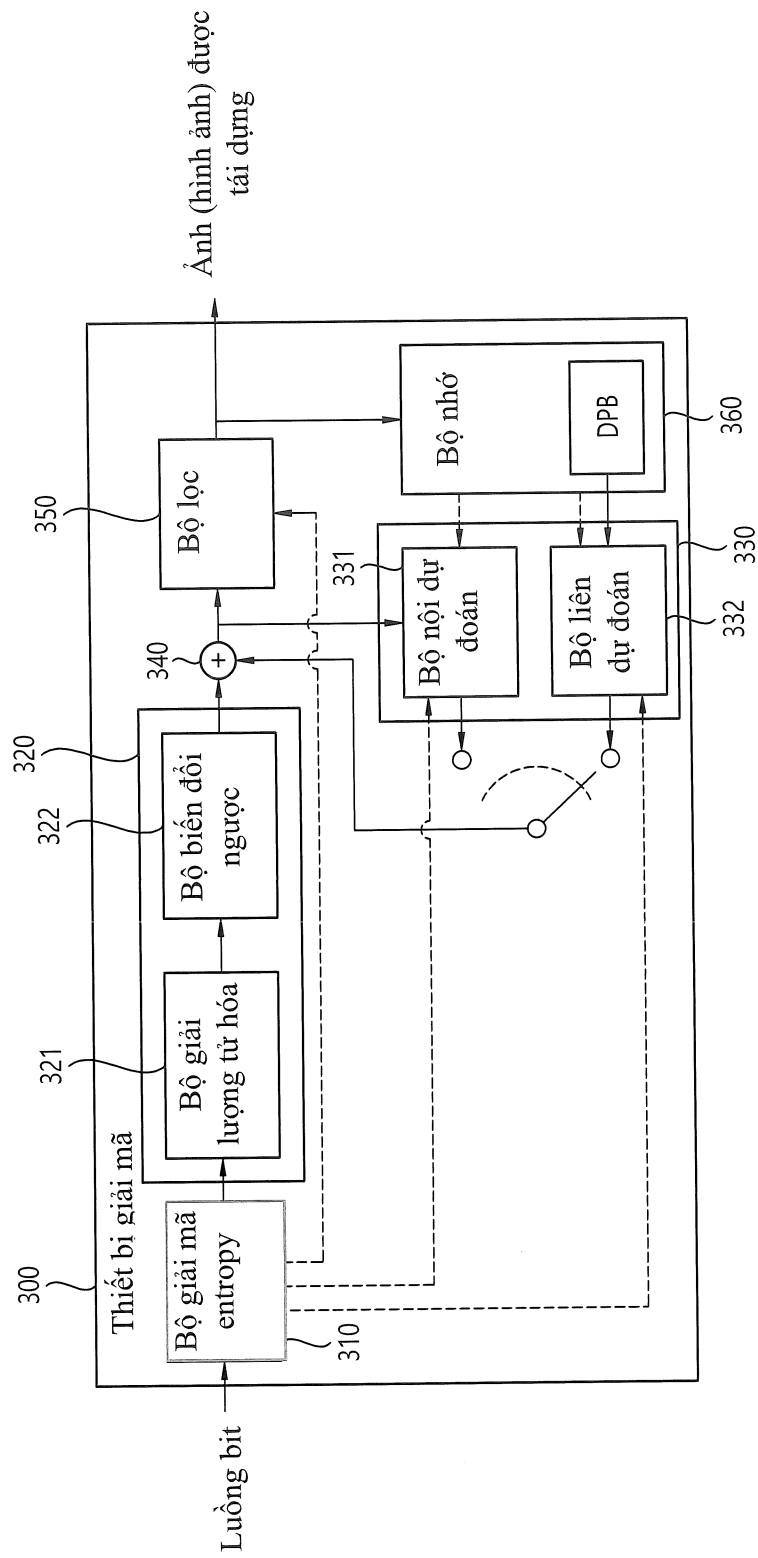


FIG. 4

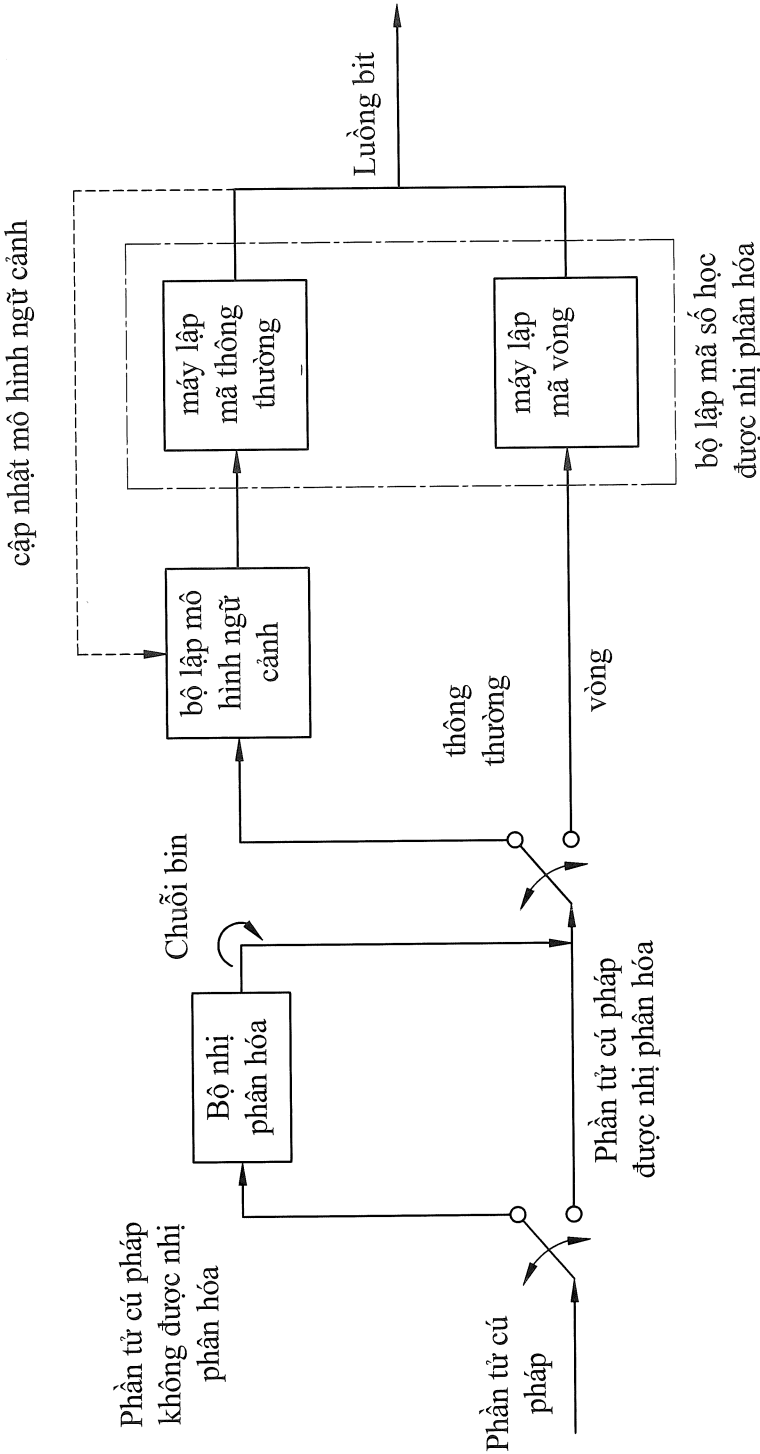


FIG. 5

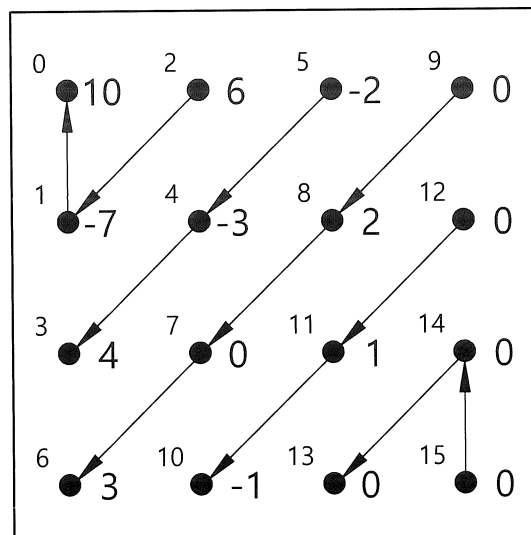


FIG. 6

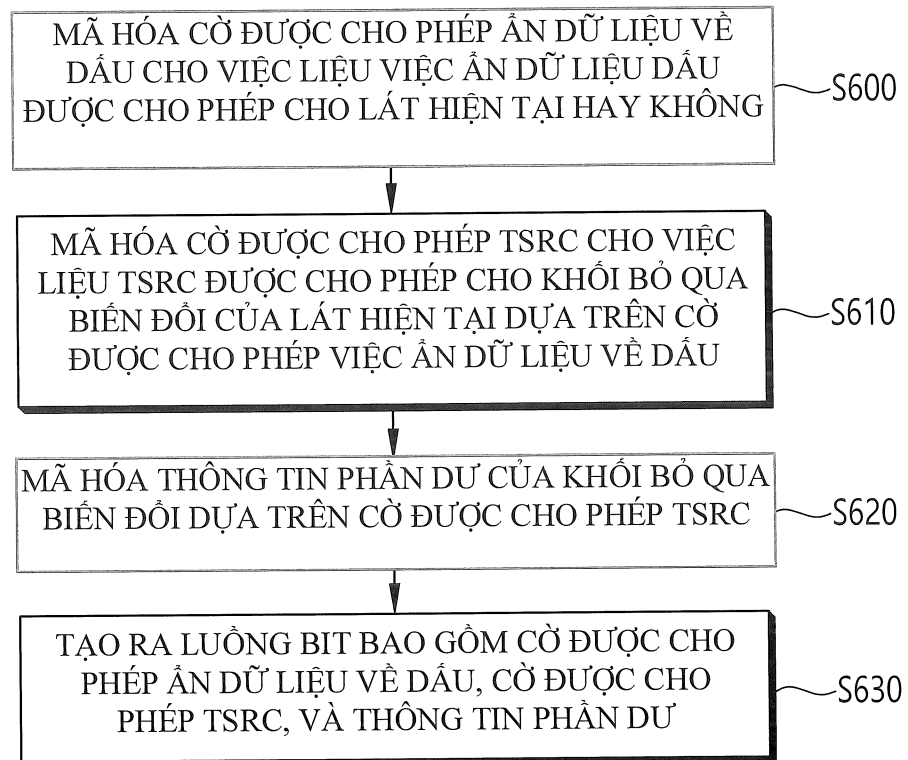


FIG. 7

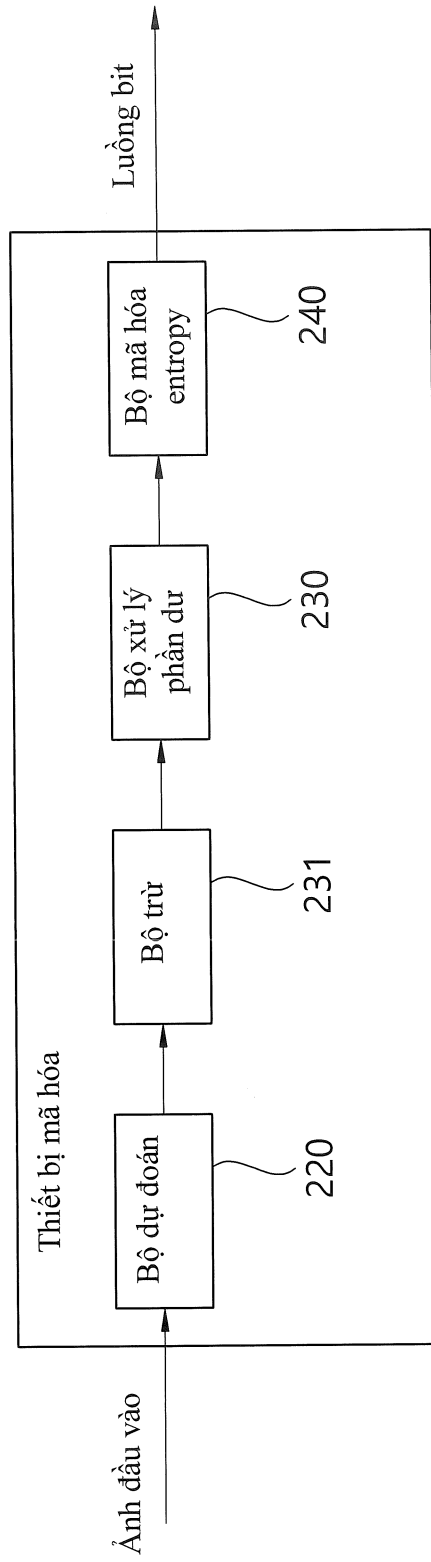


FIG. 8

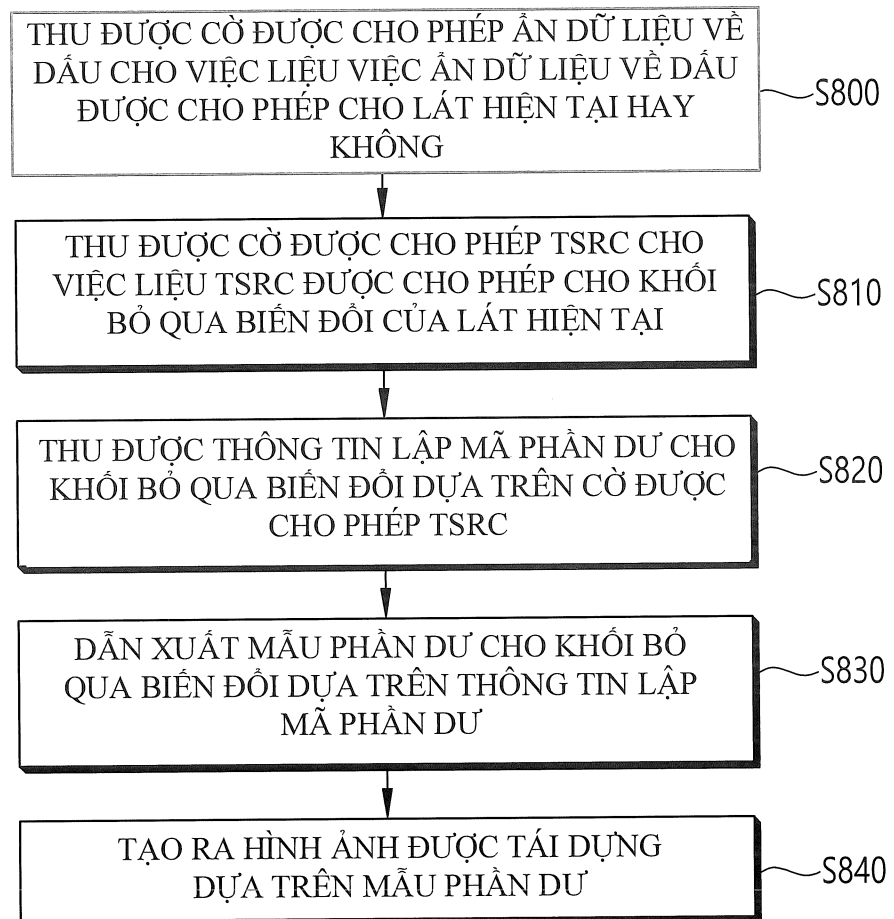


FIG. 9

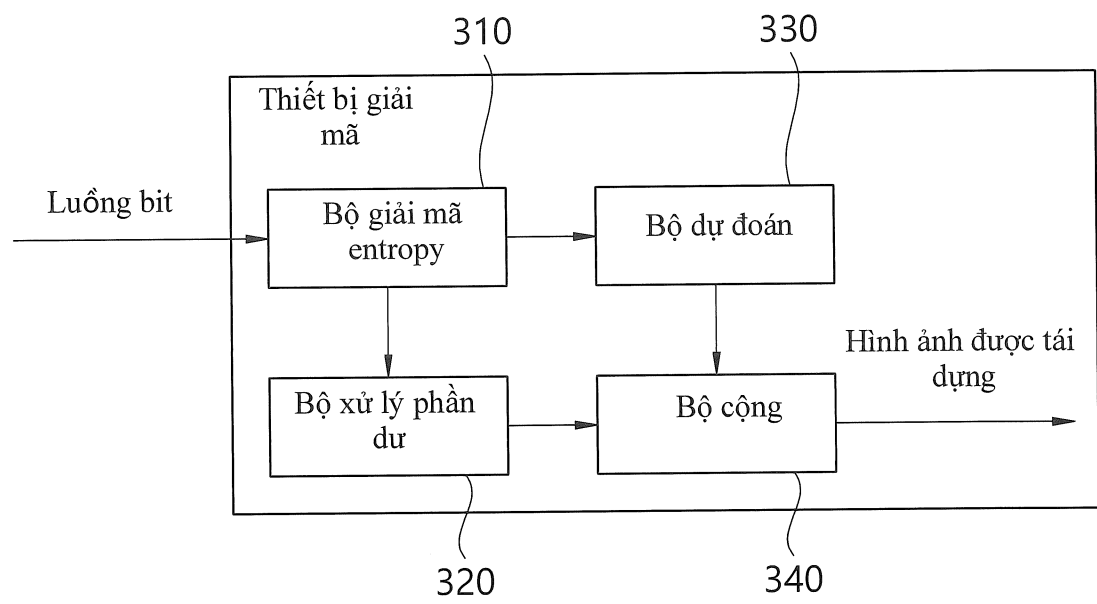


FIG. 10

