



(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)



1-0050601

(51)^{2021.01} H04N 19/60; H04N 19/176; H04N
19/70; H04N 19/186; H04N 19/124;
H04N 19/18

(13) B

(21) 1-2022-05832

(22) 22/02/2021

(86) PCT/KR2021/002182 22/02/2021

(87) WO2021/167421 26/08/2021

(30) 62/980,082 21/02/2020 US; 62/992,098 19/03/2020 US; 63/001,456 29/03/2020 US

(45) 25/08/2025 449

(43) 25/11/2022 416A

(73) LG ELECTRONICS INC. (KR)

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

(72) KOO, Moonmo (KR); LIM, Jaehyun (KR); KIM, Seunghwan (KR).

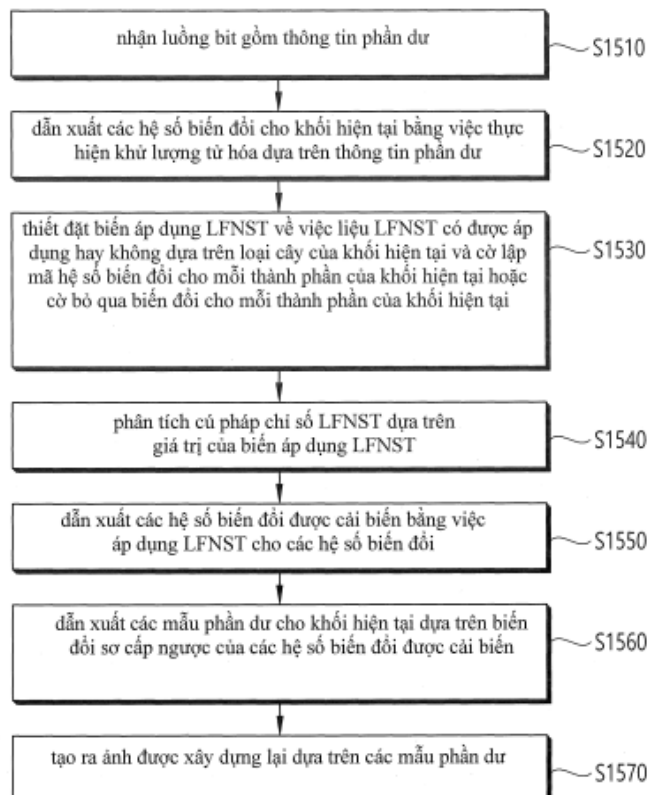
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ HÌNH ẢNH, PHƯƠNG PHÁP MÃ HÓA HÌNH ẢNH,
PHƯƠNG TIỆN LƯU TRỮ KỸ THUẬT SỐ ĐƯỢC ĐỌC BỞI MÁY TÍNH VÀ
PHƯƠNG PHÁP TRUYỀN DỮ LIỆU CHO THÔNG TIN HÌNH ẢNH

(21) 1-2022-05832

(57) Sáng chế đề cập đến phương pháp giải mã hình ảnh, phương pháp mã hóa hình ảnh, phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính và phương pháp truyền dữ liệu cho thông tin hình ảnh. Phương pháp giải mã hình ảnh theo sáng chế bao gồm bước dẫn xuất các hệ số biến đổi được cải biến, trong đó bước dẫn xuất các hệ số biến đổi được cải biến bao gồm bước thiết đặt biến áp dụng biến đổi không thể phân tách tần số thấp (low-frequency non-separable transform, LFNST) về việc liệu có áp dụng LFNST hay không trên cơ sở của loại cây của khối hiện tại, cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại, hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại, và có thể bao gồm bước phân tích cú pháp chỉ số LFNST trên cơ sở của giá trị của biến áp dụng LFNST.

FIG. 15



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập chung đến công nghệ lập mã hình ảnh và, cụ thể hơn, đề cập đến phương pháp lập mã hình ảnh dựa trên việc biến đổi trong hệ thống lập mã hình ảnh và thiết bị của nó.

Tình trạng kỹ thuật của sáng chế

Ngày nay, nhu cầu về các hình ảnh/các video chất lượng cao và độ phân giải cao chẳng hạn như các hình ảnh/các video 4K, 8K hoặc độ phân giải cực cao (ultra high definition, UHD) khác đang gia tăng trong các lĩnh vực khác nhau. Do dữ liệu hình ảnh/video trở nên có độ phân giải cao hơn và chất lượng cao hơn, lượng thông tin hoặc lượng bit được truyền tăng khi so sánh với dữ liệu hình ảnh thông thường. Do đó, khi dữ liệu hình ảnh được truyền sử dụng phương tiện như là đường băng rộng có dây/không dây thông thường hoặc dữ liệu hình ảnh/video được lưu trữ sử dụng phương tiện lưu trữ sẵn có, chi phí truyền và chi phí lưu trữ của chúng tăng.

Ngoài ra, ngày nay, sự quan tâm và nhu cầu về phương tiện nhập vai chẳng hạn như nội dung thực tế ảo (virtual reality, VR), thực tế nhân tạo (artificial reality, AR) hoặc ảnh ba chiều, hoặc tương tự đang tăng, và việc phát rộng cho các hình ảnh/các video có các đặc tính hình ảnh khác với của các hình ảnh thực, chẳng hạn như hình ảnh trò chơi đang tăng.

Theo đó, có đòi hỏi về kỹ thuật nén hình ảnh/video hiệu quả cao để nén và truyền hoặc lưu trữ, và tái tạo một cách hiệu quả thông tin của các hình ảnh/các video chất lượng cao và độ phân giải cao có các đặc tính khác nhau như được mô tả ở trên.

Bản chất kỹ thuật của sáng chế

Một khía cạnh kỹ thuật của sáng chế là nhằm đề xuất phương pháp và thiết bị để tăng hiệu quả lập mã hình ảnh.

Khía cạnh kỹ thuật khác của sáng chế là để đề xuất phương pháp và thiết bị để làm tăng hiệu quả trong việc lập mã chỉ số biến đổi không thể phân tách tần số thấp (low-frequency non-separable transform, LFNST).

Khía cạnh kỹ thuật khác nữa của sáng chế là để đề xuất phương pháp và thiết bị để làm tăng hiệu quả trong việc lập mã chỉ số LFNST dựa trên thông tin cờ về điều biến lập mã xung delta dựa trên khối (block-based delta pulse code modulation, BDPCM) và sự tồn tại của hệ số biến đổi.

Theo một phương án của sáng chế, có đề xuất phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã. Phương pháp có thể gồm dẫn xuất các hệ số biến đổi được cải biến, trong đó việc dẫn xuất của các hệ số biến đổi được cải biến có thể gồm thiết đặt biến áp dụng LFNST về việc liệu LFNST có được áp dụng hay không dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại, và phân tích cú pháp chỉ số LFNST dựa trên giá trị của biến áp dụng LFNST.

Giá trị của biến áp dụng LFNST có thể được thiết đặt là 1, mà chỉ ra rằng chỉ số LFNST được phân tích cú pháp, dựa trên cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại là 0.

Giá trị của biến áp dụng LFNST có thể được thiết đặt là 1, mà chỉ ra rằng chỉ số LFNST được phân tích cú pháp, dựa trên cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại là 0.

Khi loại cây là cây đơn, cờ bỏ qua biến đổi cho thành phần độ chói của khối hiện tại là 0, thành phần sắc độ của khối hiện tại là trong chế độ BDPCM, giá trị của biến áp dụng LFNST có thể được thiết đặt là 1, mà chỉ ra rằng chỉ số LFNST được phân tích cú pháp.

Cờ lập mã hệ số biến đổi cho thành phần sắc độ có thể là 0.

Cờ bỏ qua biến đổi cho thành phần sắc độ có thể được suy ra là 1.

Theo một phương án của sáng chế, có đề xuất phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa. Phương pháp có thể gồm dẫn xuất các hệ số biến đổi được cải biến từ các hệ số biến đổi bằng việc áp dụng LFNST, trong đó việc dẫn xuất của các hệ số biến đổi được cải biến có thể gồm thiết đặt biến áp dụng LFNST về việc liệu LFNST có được áp dụng hay không dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại, và áp dụng LFNST dựa trên giá trị của biến áp dụng LFNST.

Theo một phương án khác của sáng chế, có đề xuất phương tiện lưu trữ kỹ thuật số mà lưu trữ dữ liệu hình ảnh gồm thông tin hình ảnh được mã hóa và luồng bit được tạo ra theo phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa.

Theo một phương án khác nữa của sáng chế, có đề xuất phương tiện lưu trữ kỹ thuật số mà lưu trữ dữ liệu hình ảnh gồm thông tin hình ảnh được mã hóa và luồng bit để khiến thiết bị giải mã thực hiện phương pháp giải mã hình ảnh.

Theo sáng chế, có thể tăng hiệu quả nén hình ảnh/video tổng.

Theo sáng chế, có thể tăng hiệu quả trong việc lập mã chỉ số LFNST.

Theo sáng chế, có thể tăng hiệu quả trong việc lập mã chỉ số LFNST dựa trên thông tin cờ về BDPCM và sự tồn tại của hệ số biến đổi.

Các hiệu quả mà có thể thu nhận được thông qua các ví dụ cụ thể của sáng chế không bị giới hạn ở các hiệu quả được liệt kê ở trên. Ví dụ, có thể có các hiệu quả kỹ thuật khác nhau người có trình độ trung bình trong kỹ thuật liên quan có thể hiểu hoặc dẫn xuất từ sáng chế. Theo đó, các hiệu quả cụ thể của sáng chế không bị giới hạn ở các hiệu quả được mô tả rõ ràng trong sáng chế và có thể gồm các hiệu quả khác nhau mà có thể được hiểu hoặc được dẫn xuất từ các dấu hiệu kỹ thuật của sáng chế.

Mô tả vắn tắt các hình vẽ

Fig.1 minh họa sơ lược ví dụ của hệ thống lập mã video/hình ảnh mà cho nó sáng chế là có thể ứng dụng.

Fig.2 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị mã hóa video/hình ảnh với nó sáng chế có thể áp dụng.

Fig.3 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh với nó sáng chế có thể áp dụng.

Fig.4 minh họa cấu trúc của hệ thống phát luồng nội dung mà cho nó sáng chế được áp dụng.

Fig.5 minh họa sơ lược kỹ thuật đa biến đổi theo một phương án của sáng chế.

Fig.6 thể hiện ví dụ các chế độ hướng nội của 65 chế độ dự đoán.

Fig.7 là sơ đồ giải thích RST theo một phương án của sáng chế.

Fig.8 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của việc biến đổi sơ cấp xuôi thành vector một chiều theo một ví dụ.

Fig.9 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của việc biến đổi thứ cấp xuôi thành khối hai chiều theo một ví dụ.

Fig.10 là sơ đồ minh họa các chế độ dự đoán nội góc rộng theo một phương án của sáng chế.

Fig.11 là sơ đồ minh họa hình dạng khối mà cho nó LFNST được áp dụng.

Fig.12 là sơ đồ minh họa việc sắp xếp của dữ liệu đầu ra của LFNST xuôi theo một ví dụ.

Fig.13 minh họa việc đặt bằng không trong khối mà cho nó LFNST 4 x 4 được áp dụng theo một ví dụ.

Fig.14 minh họa việc đặt bằng không trong khối mà cho nó LFNST 8 x 8 được áp dụng theo một ví dụ.

Fig.15 là lưu đồ minh họa thao tác của thiết bị giải mã video theo một phương án của sáng chế.

Fig.16 là lưu đồ minh họa thao tác của thiết bị mã hóa video theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Trong khi sáng chế có thể là khả dĩ với các cải biến khác nhau và bao gồm các phương án khác nhau, các phương án cụ thể của nó đã được thể hiện trong các hình vẽ làm ví dụ và sau đây sẽ được mô tả chi tiết. Tuy nhiên, điều này không nhằm giới hạn sáng chế ở các phương án cụ thể được bộc lộ ở đây. Thuật ngữ được sử dụng ở đây chỉ nhằm mục đích mô tả các phương án cụ thể, và không nhằm giới hạn bản chất kỹ thuật của sáng chế. Các dạng số ít có thể gồm các dạng số nhiều trừ khi ngữ cảnh được chỉ ra rõ ràng khác đi. Các từ ngữ như “gồm” và “có” được nhằm để chỉ ra rằng các dấu hiệu, các số lượng, các bước, các hoạt động, các phần tử, các thành phần, hoặc các tổ hợp của chúng, mà được sử dụng trong phần mô tả sau đây, là tồn tại, và do đó không nên 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 tổ hợp của chúng, là bị loại trừ từ trước.

Trong khi đó, mỗi thành phần trên các hình vẽ được mô tả ở đây được minh họa một cách độc lập để thuận tiện cho việc mô tả cho các chức năng đặc tính khác nhau, và tuy nhiên, nó không có nghĩa là mỗi thành phần đạt được bằng phần cứng hoặc phần mềm tách biệt. Ví dụ, hai hoặc nhiều thành phần bất kỳ này có thể được kết hợp để tạo thành thành phần đơn, và thành phần đơn bất kỳ có thể được chia thành nhiều thành phần. Các phương án trong đó các thành phần được kết hợp và/hoặc được phân chia sẽ thuộc về phạm vi của quyền sáng chế của sáng chế miễn là chúng không trệt khỏi bản chất của sáng chế.

Sau đây, các phương án ưu tiên của sáng chế sẽ được giải thích chi tiết hơn trong khi tham chiếu tới các hình vẽ kèm theo. Ngoài ra, các ký hiệu chỉ dẫn giống nhau được sử dụng cho các bộ phận giống nhau trong các hình vẽ, và các phần mô tả lặp lại cho các bộ phận giống nhau sẽ được bỏ qua.

Sáng chế liên quan đến việc lập mã video/hình ảnh. Ví dụ, phương pháp/ví dụ được bộc lộ trong bản mô tả này có thể liên quan đến tiêu chuẩn lập mã video vạn năng (Versatile Video Coding, VVC) (ITU-T Rec. H.266), tiêu chuẩn lập mã video/hình ảnh thế hệ tiếp theo sau VVC, hoặc các tiêu chuẩn liên quan đến việc lập mã video khác (ví dụ, tiêu chuẩn lập mã video hiệu quả cao (High Efficiency Video Coding, HEVC) (ITU-T Rec. H.265), tiêu chuẩn lập mã video thiết yếu (essential video coding, EVC), tiêu chuẩn AVS2, v.v.).

Trong bản mô tả này, các phương án khác nhau liên quan đến việc lập mã video/hình ảnh có thể được đề xuất, và, trừ khi được nêu rõ khác đi, các phương án có thể được kết hợp với nhau và được thực hiện.

Trong bản mô tả này, video có thể nghĩa là tập các chuỗi hình ảnh theo thời gian. Thông thường ảnh có nghĩa là đơn vị biểu diễn hình ảnh ở khoảng thời gian cụ thể, và lát/phiên là đơn vị cấu thành một phần của ảnh. Lát/phiên có thể gồm một hoặc nhiều đơn vị cây lập mã (coding tree unit, CTU). Một ảnh có thể được cấu thành từ một hoặc nhiều lát/phiên. Một ảnh có thể được cấu thành từ một hoặc nhiều nhóm phiên. Một nhóm phiên có thể gồm một hoặc nhiều phiên.

Điểm ảnh (pixel) hoặc pel có thể có nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc hình ảnh). Ngoài ra, ‘mẫu’ có thể được sử dụng như là thuật ngữ tương ứng với điểm ảnh. Mẫu có thể biểu diễn chung cho điểm ảnh hoặc giá trị của điểm ảnh, và có thể chỉ biểu diễn điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ. Mặt khác, mẫu có thể dùng để chỉ giá trị điểm ảnh trong miền không gian, hoặc khi giá trị điểm ảnh này được chuyển đổi thành miền tần số, nó có thể dùng để chỉ hệ số biến đổi trong miền tần số.

Đơn vị có thể biểu diễn đơn vị cơ bản của việc xử lý hình ảnh. Đơn vị có thể gồm ít nhất một thành phần trong số vùng cụ thể và thông tin liên quan đến vùng. Một đơn vị có thể gồm một khối độ chói và hai khối sắc độ (ví dụ cb, cr). Đơn vị và thuật ngữ chẳng hạn như khối, khu vực, hoặc tương tự có thể được sử dụng thay thế cho nhau theo các hoàn cảnh. Trong trường hợp chung, khối $M \times N$ có thể gồm tập (hoặc mảng) của các mẫu (hoặc các mảng ví dụ) hoặc các hệ số biến đổi cấu thành từ M cột và N hàng.

Trong bản mô tả này, thuật ngữ "/" và "," nên được hiểu là để chỉ ra "và/hoặc." Ví dụ, việc diễn đạt "A/B" có thể nghĩa là "A và/hoặc B." Ngoài ra, "A, B" có thể nghĩa là "A và/hoặc B." Ngoài ra, "A/B/C" có thể nghĩa là "ít nhất một trong số A, B, và/hoặc C." Ngoài ra, "A/B/C" có thể nghĩa là "ít nhất một trong số A, B, và/hoặc C."

Ngoài ra, trong sáng chế, thuật ngữ "hoặc" nên được hiểu là để chỉ ra "và/hoặc." Ví dụ, việc diễn đạt "A hoặc B" có thể gồm 1) chỉ A, 2) chỉ B, và/hoặc 3) cả A và B. Nói cách khác, thuật ngữ "hoặc" trong bản mô tả này nên được hiểu là để chỉ ra "ngoài ra hoặc mặt khác."

Trong sáng chế, "ít nhất một trong số A và B" có thể nghĩa là "chỉ A", "chỉ B", hoặc "cả A và B". Ngoài ra, trong sáng chế, việc diễn đạt "ít nhất một trong số A hoặc B" hoặc "ít nhất một trong số A và/hoặc B" có thể được diễn giải là "ít nhất một trong số A và B".

Ngoài ra, trong sáng chế, "ít nhất một trong số A, B, và C" có thể nghĩa là "chỉ A", "chỉ B", "chỉ C", hoặc "kết hợp bất kỳ của A, B, và C". Ngoài ra, "ít nhất một trong số A, B, hoặc C" hoặc "ít nhất một trong số A, B, và/hoặc C" có thể nghĩa là "ít nhất một trong số A, B, và C".

Ngoài ra, dấu ngoặc đơn được sử dụng trong sáng chế có thể nghĩa là "ví dụ". Cụ thể là, khi được chỉ ra dưới dạng "việc dự đoán (việc dự đoán nội)", có thể nghĩa là "việc dự đoán nội" được đề xuất như ví dụ của "việc dự đoán". Nghĩa là, "việc dự đoán"

trong sáng chế không bị giới hạn ở “việc dự đoán nội”, và “việc dự đoán nội” có thể được đề xuất như là ví dụ của “việc dự đoán”. Ngoài ra, khi được chỉ ra dưới dạng “việc dự đoán (nghĩa là, việc dự đoán nội)”, nó có thể cũng nghĩa là “việc dự đoán nội” được đề xuất như ví dụ của “việc dự đoán”.

Các dấu hiệu kỹ thuật được mô tả riêng lẻ trong một hình vẽ trong sáng chế có thể được thực hiện một cách riêng lẻ hoặc có thể được thực hiện đồng thời.

Fig.1 minh họa sơ lược ví dụ của hệ thống lập mã video/hình ảnh mà cho nó sáng chế là có thể ứng dụng.

Tham khảo đến Fig.1, hệ thống lập mã video/hình ảnh có thể gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị nhận). Thiết bị nguồn có thể chuyển thông tin video/hình ảnh được mã hóa hoặc dữ liệu dưới dạng tệp hoặc phát luồng tới thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng.

Thiết bị nguồn có thể gồm nguồn video, thiết bị mã hóa, và bộ truyền. Thiết bị nhận có thể gồm bộ nhận, thiết bị giải mã, và bộ kết xuất. Thiết bị mã hóa có thể được gọi là thiết bị mã hóa video/hình ảnh, và thiết bị giải mã có thể được gọi là thiết bị giải mã video/hình ảnh. Bộ truyền có thể được 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 là thiết bị riêng biệt hoặc thành phần ngoài.

Nguồn video có thể thu nhận video/hình ảnh thông qua quy trình chụp, tổng hợp, hoặc tạo ra video/hình ảnh. Nguồn video có thể gồm thiết bị chụp video/hình ảnh và/hoặc thiết bị tạo ra video/hình ảnh. Thiết bị chụp video/hình ảnh có thể gồm, ví dụ, một hoặc nhiều camera, kho lưu trữ video/hình ảnh gồm các video/hình ảnh đã được chụp trước đó, hoặc tương tự. Thiết bị tạo ra video/hình ảnh có thể gồm, ví dụ, máy tính, máy tính bảng và điện thoại thông minh, và có thể (theo cách điện) tạo ra video/hình ảnh. Ví dụ, video/hình ảnh ảo có thể được tạo ra thông qua máy tính hoặc tương tự.

Trong trường hợp này, quy trình chụp video/hình ảnh có thể được thay thế bởi quy trình tạo ra dữ liệu liên quan.

Thiết bị mã hóa có thể mã hóa video/hình ảnh đầu vào. Thiết bị mã hóa có thể thực hiện một loạt các thủ tục như là dự đoán, biến đổi, và lượng tử hóa cho hiệu quả nén và lập mã. Dữ liệu được mã hóa (thông tin video/hình ảnh được mã hóa) có thể được xuất ra trong dạng luồng bit.

Bộ truyền có thể truyền thông tin video/hình ảnh được mã hóa hoặc dữ liệu được xuất ra trong dạng luồng bit đến bộ nhận của thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng trong dạng tệp hoặc phát 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 phương tiện thông qua định dạng tệp được xác 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 được nhận/được trích xuất đến thiết bị giải mã.

Thiết bị giải mã có thể giải mã video/hình ảnh bằng việc thực hiện chuỗi các thủ tục chẳng hạn như khử lượng tử hóa, biến đổi ngược, dự đoán, và tương tự tương ứng với thao tác của thiết bị mã hóa.

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

Fig.2 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị mã hóa video/hình ảnh với nó sáng chế có thể áp dụng. Sau đây, phần được gọi là thiết bị mã hóa video có thể gồm thiết bị mã hóa hình ảnh.

Tham khảo đến Fig.2, thiết bị mã hóa 200 có thể gồm bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropi 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể gồm bộ dự đoán liên 221 và bộ dự đoán nội 222. Bộ xử lý phần dư 230 có thể gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ khử

lượng tử hóa 234, bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ xây dựng lại hoặc bộ tạo khối được xây dựng lại. Bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropi 240, bộ cộng 250, và bộ lọc 260, mà đã được mô tả ở trên, có thể được cấu thành bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip bộ mã hóa hoặc các bộ xử lý) theo một phương án. Ngoài ra, bộ nhớ 270 có thể gồm bộ đệm ảnh được giải mã (DPB), và có thể được cấu thành bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 270 như là thành phần trong/ngoài.

Bộ phân vùng hình ảnh 210 có thể phân vùng hình ảnh đầu vào (hoặc ảnh hoặc khung) được nhập vào thiết bị mã hóa 200 thành một hoặc nhiều đơn vị xử lý. Như một ví dụ, đơn vị xử lý có thể được gọi là đơn vị lập mã (coding unit, CU). Trong trường hợp này, bắt đầu với đơn vị cây lập mã (coding tree unit, CTU) hoặc đơn vị lập mã lớn nhất (largest coding unit, LCU), đơn vị lập mã có thể được phân vùng theo cách đệ quy theo cấu trúc cây tứ phân cây nhị phân cây tam phân (Quad-tree binary-tree ternary-tree, QTBTBT). Ví dụ, một đơn vị lập mã có thể được chia thành nhiều đơn vị lập 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 đầu tiên và cấu trúc cây nhị phân và/hoặc cấu trúc tam phân có thể được áp dụng sau đó. Mặt khác, cấu trúc cây nhị phân có thể được áp dụng đầu tiên. Thủ tục lập mã theo sáng chế có thể được thực hiện dựa trên đơn vị lập mã cuối cùng mà không được phân vùng tiếp. Trong trường hợp này, đơn vị lập mã tối đa có thể được sử dụng một cách trực tiếp làm đơn vị lập mã cuối cùng dựa trên hiệu quả lập mã theo đặc tính hình ảnh. Mặt khác, đơn vị lập mã có thể được phân vùng theo cách đệ quy thành các đơn vị lập mã của độ sâu sâu hơn nữa nếu cần thiết, sao cho đơn vị lập mã có kích thước tối ưu có thể được sử dụng làm đơn vị lập mã cuối cùng. Ở đây, thủ tục lập mã có thể gồm các thủ tục chẳng hạn như dự đoán, biến đổi, và xây dựng lại, mà sẽ được mô tả sau. Theo một ví dụ khác, đơn vị xử lý có thể còn 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 hoặc được phân vùng từ đơn vị lập mã cuối cùng được mô tả ở trên. Đơn vị dự đoán có thể là đơn vị của việc dự đoán mẫu, 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 và thuật ngữ chẳng hạn như khối, khu vực, hoặc tương tự có thể được sử dụng thay thế cho nhau theo các hoàn cảnh. Trong trường hợp chung, khối $M \times N$ có thể thể hiện tập của các mẫu hoặc các hệ số biến đổi bao gồm M cột và N hàng. Mẫu có thể biểu diễn chung điểm ảnh hoặc giá trị của điểm ảnh, và có thể thể hiện 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. Mẫu có thể được sử dụng làm thuật ngữ tương ứng với điểm ảnh hoặc pel của một ảnh (hoặc hình ảnh).

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

Bộ dự đoán nội 222 có thể dự đoán khối hiện tại bằng việc tham chiếu tới các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt ở lân cận hoặc cách

xa với khối hiện tại theo chế độ dự đoán. Trong việc dự đoán nội, các chế độ dự đoán có thể gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Các chế độ không có hướng có thể gồm, ví dụ, chế độ DC và chế độ phẳng. Chế độ có hướng có thể 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, đây chỉ đơn thuần là ví dụ, và nhiều hoặc ít chế độ dự đoán định hướng hơn có thể được sử dụng phụ thuộc vào việc thiết đặt. Bộ dự đoán nội 222 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng việc sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên 221 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vector chuyển động trên ảnh tham chiếu. Tại thời điểm này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động có thể được dự đoán trên cơ sở khối, khối con, hoặc mẫu dựa trên sự tương liên của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm vector chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin hướng dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, v.v.). Trong trường hợp dự đoán liên, khối lân cận có thể gồm khối lân cận theo không gian tồn tại trong ảnh hiện tại và khối lân cận theo thời gian có mặt trong ảnh tham chiếu. Ảnh tham chiếu gồm khối tham chiếu và ảnh tham chiếu gồm khối lân cận theo thời gian có thể là giống nhau hoặc khác nhau. Khối lân cận theo thời gian có thể được gọi là khối tham chiếu được đặt cùng một chỗ, CU được đặt cùng một chỗ (collocated CU, colCU), và tương tự, và ảnh tham chiếu gồm khối lân cận theo thời gian có thể được gọi là ảnh được đặt cùng một chỗ (collocated picture, colPic). Ví dụ, bộ dự đoán liên 221 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và tạo ra thông tin đang chỉ ra ứng viên nào được sử dụng để dẫn xuất vector chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau. Ví dụ, trong trường hợp của chế độ bỏ qua và chế độ hợp nhất, bộ dự đoán liên 221 có

thể sử dụng thông tin chuyển động của khối lân cận làm thông tin chuyển động của khối hiện tại. Trong chế độ bỏ qua, không giống với chế độ hợp nhất, tín hiệu dư có thể không được truyền. Trong trường hợp chế độ dự đoán thông tin 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 ra bằng việc báo hiệu sự chênh lệch vector chuyển động.

Bộ dự đoán 220 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc dự đoán nội hoặc việc dự đoán liên cho việc dự đoán trên một khối, và còn có thể áp dụng việc dự đoán nội và việc dự đoán liên cùng một lúc. Việc này có thể được gọi là việc dự đoán nội và liên được kết hợp (inter and intra prediction, CIIP). Ngoài ra, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép khối nội (intra block copy, IBC), hoặc chế độ bảng màu để thực hiện việc dự đoán trên khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc lập mã hình ảnh/video nội dung của trò chơi hoặc tương tự, chẳng hạn như lập mã nội dung màn (screen content coding, SCC). Mặc dù IBC về cơ bản thực hiện việc dự đoán trong khối hiện tại, nó có thể được thực hiện tương tự như việc dự đoán liên ở chỗ nó dẫn xuất khối tham chiếu trong khối hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật dự đoán liên được mô tả trong sáng chế.

Tín hiệu dự đoán được tạo ra thông qua bộ dự đoán liên 221 và/hoặc bộ dự đoán nội 222 có thể được sử dụng để tạo ra tín hiệu được xây dựng lại hoặc để tạo ra tín hiệu 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 dư. Ví dụ, kỹ thuật biến đổi có thể gồm ít nhất một kỹ thuật trong số biến đổi côsin 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 tùy theo điều kiện (Conditionally Non-Linear Transform, CNT). Ở đây, GBT nghĩa là việc biến đổi thu nhận được từ đồ thị khi thông tin quan hệ giữa các điểm ảnh được biểu diễn bởi

đồ thị. CNT đề cập đến việc biến đổi thu được dựa trên tín hiệu dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được xây dựng lại trước đó. Bên cạnh đó, quy trình biến đổi có thể được áp dụng cho các khối điểm ảnh hình vuông có cùng kích thước hoặc có thể được áp dụng cho các khối có kích thước khác hình vuông.

Bộ lượng tử hóa 233 có thể lượng tử hóa các hệ số biến đổi và truyền chúng đến bộ mã hóa entropi 240, và bộ mã hóa entropi 240 có thể mã hóa tín hiệu được lượng tử hóa (thông tin về các hệ số biến đổi được lượng tử hóa) và xuất ra tín hiệu được mã hóa trong luồng bit. Thông tin về các hệ số biến đổi được lượng tử hóa có thể được gọi là thông tin phần dư. Bộ lượng tử hóa 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa loại khối thành dạng vectơ một chiều dựa trên thứ tự quét hệ số, và tạo ra thông tin về các hệ số biến đổi được lượng tử hóa dựa trên các hệ số biến đổi được lượng tử hóa của dạng vectơ một chiều. Bộ mã hóa entropi 240 có thể thực hiện các phương pháp mã hóa khác nhau, ví dụ, như Golomb hàm số mũ, lập mã chiều dài biến đổi thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding, CAVLC), lập mã số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, CABAC), và tương tự. Bộ mã hóa entropi 240 có thể mã hóa thông tin cần thiết cho việc xây dựng lại video/hình ảnh khác với các hệ số biến đổi được lượng tử hóa (ví dụ các giá trị của các phần tử cú pháp, v.v.) cùng nhau hoặc một cách riêng biệt. Thông tin được mã hóa (ví dụ, thông tin video/hình ảnh được mã hóa) có thể được truyền hoặc được lưu trữ trên cơ sở đơn vị của lớp trừu tượng mạng (network abstraction layer, NAL) dưới dạng luồng bit. Thông tin video/hình ảnh có thể còn gồm thông tin về các tập thông số khác nhau chẳng hạn như tập thông số thích ứng (adaptation parameter set, APS), tập thông số ảnh (picture parameter set, PPS), tập thông số trình tự (sequence parameter set, SPS), tập thông số video (video parameter set, VPS) hoặc tương tự. Ngoài ra, thông tin video/hình ảnh có thể còn gồm thông tin ràng buộc chung. Theo sáng chế, thông tin và/hoặc các phần tử cú pháp mà được truyền/được báo hiệu tới thiết bị giải mã từ thiết bị mã hóa có thể được gồm trong thông tin video/hình 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 được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Ở đây, mạng có thể gồm mạng phát rộng, mạng truyền thông và/hoặc tương tự, và 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 (không được thể hiện) mà truyền tín hiệu được xuất từ bộ mã hóa entropi 240 và/hoặc bộ lưu trữ (không được thể hiện) mà lưu trữ có thể tạo cấu hình dưới dạng thành phần trong/ngoài của thiết bị mã hóa 200, hoặc bộ truyền có thể được gồm trong bộ mã hóa entropi 240.

Các hệ số biến đổi được lượng tử hóa được xuất từ bộ lượng tử hóa 233 có thể được sử dụng để tạo ra tín hiệu dự đoán. Ví dụ, bằng việc áp dụng việc khử lượng tử hóa và việc biến đổi ngược cho các hệ số biến đổi được lượng tử hóa thông qua bộ khử lượng tử hóa 234 và bộ biến đổi ngược 235, tín hiệu phần dư (khối phần dư hoặc các mẫu phần dư) có thể được xây dựng lại. Bộ cộng 155 cộng tín hiệu phần dư được xây dựng lại vào tín hiệu dự đoán được xuất từ bộ dự đoán liên 221 hoặc bộ dự đoán nội 222, sao cho tín hiệu được xây dựng lại (ảnh được xây dựng lại, khối được xây dựng lại, mảng mẫu được xây dựng lại) có thể được tạo ra. Khi không có phần dư cho khối mục tiêu xử lý như trong trường hợp trong đó chế độ bỏ qua được áp dụng, khối được dự đoán có thể được sử dụng làm khối được xây dựng lại. Bộ cộng 250 có thể được gọi là bộ xây dựng lại hoặc bộ tạo khối được xây dựng lại. Tín hiệu được xây dựng lại được tạo ra có thể được sử dụng cho việc dự đoán nội của khối mục tiêu xử lý tiếp theo trong khối hiện tại, và như được mô tả sau đây, có thể được sử dụng cho việc dự đoán liên của ảnh tiếp theo qua việc lọc.

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

Bộ lọc 260 có thể cải thiện chất lượng video chủ quan/khách quan bằng việc áp dụng việc lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 260 có thể tạo ra ảnh được

xây dựng lại được cải biến bằng việc áp dụng các phương pháp lọc khác nhau cho ảnh được xây dựng lại, và có thể lưu trữ ảnh được xây dựng lại được cải biến trong bộ nhớ 270, cụ thể trong DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch chuyển thích ứng mẫu, bộ lọc vòng lặp thích ứng, bộ lọc song phương hoặc tương tự. Như được thảo luận sau trong phần mô tả của mỗi phương pháp lọc, bộ lọc 260 có thể tạo ra thông tin khác nhau liên quan đến việc lọc, và truyền thông tin được tạo ra tới bộ mã hóa entropi 240. Thông tin về việc lọc có thể được mã hóa trong bộ mã hóa entropi 240 và xuất ra dưới dạng luồng bit.

Ảnh được xây dựng lại được cải biến mà đã được truyền tới bộ nhớ 270 có thể được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên 221. Thông qua việc này, thiết bị mã hóa có thể tránh sự không khớp của việc dự đoán trong thiết bị mã hóa 100 và thiết bị giải mã khi việc dự đoán liên được áp dụng, và có thể cũng cải thiện hiệu quả lập mã.

Bộ nhớ 270 DPB có thể lưu trữ ảnh được xây dựng lại được cải biến để sử dụng chúng làm ảnh tham chiếu trong bộ dự đoán liên 221. Bộ nhớ 270 có thể lưu trữ thông tin chuyển động của khối trong ảnh hiện tại, mà từ đó thông tin chuyển động đã được dẫn xuất (hoặc được mã hóa) và/hoặc thông tin chuyển động của các khối trong ảnh đã được xây dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền tới bộ dự đoán liên 221 để được sử dụng như là thông tin chuyển động của khối lân cận hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ nhớ 270 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong ảnh hiện tại, và truyền chúng tới bộ dự đoán nội 222.

Fig.3 là sơ đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh với nó sáng chế có thể áp dụng.

Tham khảo đến Fig.3, thiết bị giải mã video 300 có thể gồm bộ giải mã entropi 310, bộ xử lý phân dư 320, bộ dự đoán 330, bộ cộng 340, bộ lọc 350, và bộ nhớ 360. Bộ dự đoán 330 có thể gồm bộ dự đoán liên 331 và bộ dự đoán nội 332. Bộ xử lý phân dư

320 có thể gồm bộ khử lượng tử hóa 321 và bộ biến đổi ngược 321. Bộ giải mã entropi 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, mà đã được mô tả ở trên, có thể được cấu thành bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip bộ giải mã hoặc các bộ xử lý) theo một phương án. Ngoài ra, bộ nhớ 360 có thể gồm bộ đệm ảnh được giải mã (DPB), và có thể được cấu thành bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 360 như là thành phần trong/ngoài.

Khi luồng bit gồm thông tin video/hình ảnh được nhập, thiết bị giải mã 300 có thể xây dựng lại hình ảnh tương ứng với quy trình bởi đó thông tin video/hình ảnh đã 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 tới việc phân vùng khối thu nhận được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã bằng việc sử dụng đơn vị xử lý được áp dụng trong thiết bị mã hóa. Do đó, đơn vị xử lý của việc giải mã có thể là, ví dụ, đơn vị lập mã, mà có thể được phân vùng cùng 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 xuất từ đơn vị lập mã. Và, tín hiệu hình ảnh được xây dựng lại được giải mã và được xuất ra thông qua thiết bị giải mã 300 có thể được xây dựng lại thông qua bộ tái tạo.

Thiết bị giải mã 300 có thể nhận tín hiệu được xuất ra từ thiết bị mã hóa trên Fig.2 trong dạng luồng bit, và tín hiệu nhận được có thể được giải mã thông qua bộ giải mã entropi 310. Ví dụ, bộ giải mã entropi 310 có thể phân tích cú pháp luồng bit để dẫn xuất thông tin (ví dụ, thông tin video/hình ảnh) được yêu cầu cho việc xây dựng lại hình ảnh (hoặc việc xây dựng lại ảnh). Thông tin video/hình ảnh có thể còn gồm thông tin về các tập thông số khác nhau chẳng hạn như tập thông số thích ứng (APS), tập thông số ảnh (PPS), tập thông số trình tự (SPS), tập thông số video (VPS) hoặc tương tự. Ngoài ra, thông tin video/hình ảnh có thể còn gồm thông tin ràng buộc chung. Thiết bị giải mã có thể giải mã ảnh thêm dựa trên thông tin về tập thông số và/hoặc thông tin ràng buộc

chung. Trong sáng chế, thông tin được báo hiệu/được nhận và/hoặc các phần tử cú pháp, mà sẽ được mô tả sau, có thể được giải mã thông qua thủ tục giải mã và được thu nhận từ luồng bit. Ví dụ, bộ giải mã entropi 310 có thể giải mã thông tin trong luồng bit dựa trên phương pháp lập mã chẳng hạn như mã hóa Golomb hàm số mũ, CAVLC, CABAC, hoặc tương tự, và có thể xuất ra giá trị của phần tử cú pháp cần thiết cho việc xây dựng lại hình ảnh và được lượng tử hóa các giá trị của hệ số biến đổi liên quan đến phần dư. Cụ thể hơn là, phương pháp giải mã entropi CABAC có thể nhận ngán tương ứng với mỗi phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh sử dụng thông tin phần tử cú pháp mục tiêu giải mã và không tin giải mã các khối mục tiêu giải mã và lân cận, hoặc thông tin của ký hiệu/ngán được giải mã trong bước trước đó, dự đoán khả năng tạo ra ngán theo mô hình ngữ cảnh được xác định và thực hiện việc giải mã số học của ngán để tạo ra ký hiệu tương ứng với mỗi giá trị phần tử cú pháp. Ở đây, phương pháp giải mã entropi CABAC có thể cập nhật mô hình ngữ cảnh sử dụng thông tin của ký hiệu/ngán được giải mã cho mô hình ngữ cảnh của ký hiệu/ngán tiếp theo sau việc xác định của mô hình ngữ cảnh. Thông tin về việc dự đoán trong số thông tin được giải mã trong bộ giải mã entropi 310 có thể được cung cấp cho bộ dự đoán (bộ dự đoán liên 332 và bộ dự đoán nội 331), và các giá trị phần dư, nghĩa là, các hệ số biến đổi được lượng tử hóa, trên đó việc giải mã entropi đã được thực hiện trong bộ giải mã entropi 310, và thông tin thông số được kết hợp có thể được nhập 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 trong bộ giải mã entropi 310 có thể được cung cấp cho bộ lọc 350. Trong khi đó, bộ nhận (không được thể hiện) mà nhận tín hiệu được xuất từ thiết bị mã hóa có thể còn cấu thành thiết bị giải mã 300 dưới dạng thành phần trong/ngoài, và bộ nhận có thể là thành phần của bộ giải mã entropi 310. Trong lúc đó, thiết bị giải mã theo sáng chế có thể được gọi là thiết bị lập mã video/hình ảnh/ảnh, và thiết bị giải mã có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin video/hình ảnh/ảnh) và bộ giải mã mẫu (bộ giải mã mẫu video/hình ảnh/ảnh). Bộ giải mã thông tin có thể gồm bộ giải mã entropi 310,

và bộ giải mã mẫu có thể gồm ít nhất một thành phần trong số bộ khử lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ dự đoán liên 332, và bộ dự đoán nội 331.

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

Bộ khử lượng tử hóa 322 thu nhận tín hiệu phần dư (khối phần dư, mảng mẫu phần dư) bằng việc biến đổi ngược các hệ số biến đổi.

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

Bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc dự đoán nội hoặc việc dự đoán liên cho việc dự đoán trên một khối, và còn có thể áp dụng việc dự đoán nội và việc dự đoán liên cùng một lúc. Việc này có thể được gọi là việc dự đoán nội và liên được kết hợp (CIIP). Ngoài ra, bộ dự đoán có thể thực hiện sao chép khối nội (IBC) cho việc dự đoán trên khối. Sao chép khối nội có thể được sử dụng cho việc lập mã hình ảnh/video nội dung của trò chơi hoặc tương tự, chẳng hạn như lập mã nội dung màn (SCC). Mặc dù IBC về cơ bản thực hiện việc dự đoán trong khối hiện tại, nó có thể được thực hiện tương tự như việc dự đoán liên ở chỗ nó dẫn xuất khối tham chiếu trong khối hiện tại.

Nghĩa là, IBC có thể sử dụng ít nhất một trong số các kỹ thuật dự đoán liên được mô tả trong sáng chế.

Bộ dự đoán nội 331 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt ở lân cận hoặc cách xa với khối hiện tại theo chế độ dự đoán. Trong việc dự đoán nội, các chế độ dự đoán có thể gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Bộ dự đoán nội 331 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng việc sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ dự đoán liên 332 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vectơ chuyển động trên ảnh tham chiếu. Tại thời điểm này, để làm giảm lượng thông tin chuyển động được truyền trong chế độ dự đoán liên, thông tin chuyển động có thể được dự đoán trên cơ sở khối, khối con, hoặc mẫu dựa trên sự tương liên của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm vectơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin hướng dự đoán liên (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, v.v.). Trong trường hợp dự đoán liên, khối lân cận có thể gồm khối lân cận theo không gian tồn tại trong ảnh hiện tại và khối lân cận theo thời gian có mặt trong ảnh tham chiếu. Ví dụ, bộ dự đoán liên 332 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận, và dẫn xuất vectơ chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại dựa trên thông tin lựa chọn ứng viên được nhận. Việc dự đoán liên có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể gồm thông tin chỉ ra chế độ của việc dự đoán liên cho khối hiện tại.

Bộ cộng 340 có thể tạo ra tín hiệu được xây dựng lại (ảnh được xây dựng lại, khối được xây dựng lại, mảng mẫu được xây dựng lại) bằng cách cộng tín hiệu dư thu nhận được vào tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán) được xuất ra từ bộ dự đoán 330. Khi không có phần dư cho khối mục tiêu xử lý như trong trường hợp

trong đó chế độ bỏ qua được áp dụng, khối được dự đoán có thể được sử dụng làm khối được xây dựng lại.

Bộ cộng 340 có thể được gọi là bộ xây dựng lại hoặc bộ tạo khối được xây dựng lại. Tín hiệu được xây dựng lại được tạo ra có thể được sử dụng cho việc dự đoán nội của khối mục tiêu xử lý tiếp theo trong khối hiện tại, và như được mô tả sau đây, có thể được xuất ra qua việc lọc hoặc được sử dụng cho việc dự đoán liên của ảnh tiếp theo.

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

Bộ lọc 350 có thể cải thiện chất lượng video chủ quan/khách quan bằng việc áp dụng việc lọc cho tín hiệu được xây dựng lại. Ví dụ, bộ lọc 350 có thể tạo ra ảnh được xây dựng lại được cải biến bằng việc áp dụng các phương pháp lọc khác nhau cho ảnh được xây dựng lại, và có thể truyền ảnh được xây dựng lại được cải biến trong bộ nhớ 360, cụ thể trong DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc khử khối, dịch chuyển thích ứng mẫu, bộ lọc vòng lặp thích ứng, bộ lọc song phương hoặc tương tự.

Ảnh được xây dựng lại (được cải biến) mà đã được lưu trữ trong DPB của bộ nhớ 360 có thể được sử dụng làm ảnh tham chiếu trong bộ dự đoán liên 332. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối trong ảnh hiện tại, từ đó thông tin chuyển động đã được dẫn xuất (hoặc được giải mã) và/hoặc thông tin chuyển động của các khối trong ảnh đã được xây dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền tới bộ dự đoán liên 260 để được sử dụng như là thông tin chuyển động của khối lân cận hoặc thông tin chuyển động của khối lân cận theo thời gian. Bộ nhớ 360 có thể lưu trữ các mẫu được xây dựng lại của các khối được xây dựng lại trong ảnh hiện tại, và truyền chúng tới bộ dự đoán nội 331.

Trong bản mô tả này, các ví dụ được mô tả trong bộ dự đoán 330, bộ khử lượng tử hóa 321, bộ biến đổi ngược 322, và bộ lọc 350 của thiết bị giải mã 300 có thể được

áp dụng tương tự hoặc tương ứng cho bộ dự đoán 220, bộ khử lượng tử hóa 234, bộ biến đổi ngược 235, và bộ lọc 260 của thiết bị mã hóa 200, một cách tương ứng.

Như được mô tả ở trên, việc dự đoán được thực hiện để tăng hiệu quả nén trong việc thực hiện lập mã video. Thông qua việc này, khối được dự đoán gồm các mẫu dự đoán cho khối hiện tại, mà là khối mục tiêu lập mã, có thể được tạo ra. Ở đây, khối được dự đoán gồm các mẫu dự đoán trong miền không gian (hoặc miền điểm ảnh). Khối được dự đoán có thể được dẫn xuất giống nhau trong thiết bị mã hóa và thiết bị giải mã, và thiết bị mã hóa có thể tăng hiệu quả lập mã hình ảnh bằng việc báo hiệu tới thiết bị giải mã không phải giá trị mẫu ban đầu của chính khối gốc mà thông tin về phần dư (thông tin phần dư) giữa khối gốc và khối được dự đoán. Thiết bị giải mã có thể dẫn xuất khối phần dư gồm các mẫu phần dư dựa trên thông tin phần dư, tạo ra khối được xây dựng lại gồm các mẫu được xây dựng lại bằng việc cộng khối phần dư vào khối được dự đoán, và tạo ra ảnh được xây dựng lại gồm các khối được xây dựng lại.

Thông tin phần dư có thể được tạo ra thông qua các thủ tục biến đổi và lượng tử hóa. Ví dụ, thiết bị mã hóa có thể dẫn xuất khối phần dư giữa khối gốc và khối được dự đoán, dẫn xuất các hệ số biến đổi bằng việc thực hiện thủ tục biến đổi trên các mẫu phần dư (mảng mẫu phần dư) được gồm trong khối phần dư, và dẫn xuất các hệ số biến đổi được lượng tử hóa bằng việc thực hiện thủ tục lượng tử hóa về các hệ số biến đổi, sao cho nó có thể báo hiệu thông tin phần dư được kết hợp tới thiết bị giải mã (thông qua luồng bit). Ở đây, thông tin phần dư có thể gồm thông tin giá trị, thông tin vị trí, kỹ thuật biến đổi, nhân biến đổi, thông số lượng tử hóa hoặc tương tự của các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể thực hiện thủ tục lượng tử hóa/khử lượng tử hóa và dẫn xuất các mẫu phần dư (hoặc khối mẫu phần dư), dựa trên thông tin phần dư. Thiết bị giải mã có thể tạo ra khối được xây dựng lại dựa trên khối được dự đoán và khối phần dư. Thiết bị mã hóa có thể dẫn xuất khối phần dư bằng việc khử lượng tử hóa/biến đổi ngược các hệ số biến đổi được lượng tử hóa để tham chiếu cho việc dự đoán liên của ảnh tiếp theo, và có thể tạo ra ảnh được xây dựng lại dựa trên điều này.

Fig.4 minh họa cấu trúc của hệ thống phát luồng nội dung mà cho nó sáng chế được áp dụng.

Ngoài ra, hệ thống phát luồng nội dung mà cho nó sáng chế được áp dụng có thể bao gồm máy chủ mã hóa, máy chủ phát luồng, máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị nhập đa phương tiện.

Máy chủ mã hóa thực hiện chức năng nén tới dữ liệu kỹ thuật số các nội dung được nhập từ các thiết bị đầu vào đa phương tiện, chẳng hạn như điện thoại thông minh, camera, máy quay video và tương tự, để tạo ra luồng bit, và để truyền nó vào máy chủ phát luồng. Theo một ví dụ khác, trong trường hợp trong đó thiết bị đầu vào đa phương tiện, chẳng hạn như, điện thoại thông minh, camera, máy quay video hoặc tương tự, trực tiếp tạo ra luồng bit, máy chủ mã hóa có thể được bỏ qua. Luồng bit có thể được tạo ra bằng phương pháp mã hóa hoặc phương pháp tạo ra luồng bit mà cho nó sáng chế được áp dụng. Và máy chủ phát luồng có thể lưu trữ luồng bit tạm thời trong quá trình xử lý để truyền hoặc nhận luồng bit.

Máy chủ phát luồng truyền dữ liệu đa phương tiện tới thiết bị người dùng trên cơ sở của yêu cầu của người dùng thông qua máy chủ web, mà đóng vai trò là phương tiện mà thông tin cho người dùng về việc dịch vụ là gì. Khi người dùng yêu cầu dịch vụ mà người dùng muốn, máy chủ web truyền yêu cầu tới máy chủ phát luồng, và máy chủ phát luồng truyền dữ liệu đa phương tiện tới người dùng. Theo đó, hệ thống phát luồng nội dung có thể gồm máy chủ điều khiển tách biệt, và trong trường hợp này, máy chủ điều khiển đóng vai trò điều khiển các lệnh/các phản hồi giữa các thiết bị tương ứng trong hệ thống phát luồng nội dung.

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

Ví dụ, 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, trợ lý kỹ thuật số cá nhân (personal digital assistant, PDA), máy phát đa phương tiện cầm tay (portable multimedia player, PMP), bộ điều hướng, PC dạng tấm, PCT máy tính bảng, máy tính ultrabook, thiết bị đeo được (ví dụ, đầu cuối kiểu đồng hồ (đồng hồ thông minh), đầu cuối kiểu kính (kính thông minh), bộ hiển thị gắn trên đầu (head mounted display, HMD)), TV kỹ thuật số, máy tính để bàn, biển báo kỹ thuật số hoặc tương tự. Mỗi trong số các máy chủ trong hệ thống phát luồng nội dung có thể được thao tác như máy chủ được phân phối, và trong trường hợp này, dữ liệu được nhận bởi mỗi máy chủ có thể được xử lý theo cách được phân phối.

Fig.5 minh họa sơ lược kỹ thuật đa biến đổi theo một phương án của sáng chế.

Tham khảo đến Fig.5, bộ biến đổi có thể tương ứng với bộ biến đổi trong thiết bị mã hóa nên trên của Fig.2, và bộ biến đổi ngược có thể tương ứng với bộ biến đổi ngược trong thiết bị mã hóa nên trên của Fig.2, hoặc với bộ biến đổi ngược trong thiết bị giải mã của Fig.3.

Bộ biến đổi có thể dẫn xuất các hệ số biến đổi (sơ cấp) bằng việc thực hiện việc biến đổi sơ cấp dựa trên các mẫu phần dư (mảng mẫu phần dư) trong khối phần dư (S510). Việc biến đổi sơ cấp này có thể được gọi là việc biến đổi lõi. Ở đây, việc biến đổi sơ cấp có thể dựa trên việc lựa chọn đa biến đổi (multiple transform selection, MTS), và khi việc đa biến đổi được áp dụng làm việc biến đổi sơ cấp, có thể được gọi là đa biến đổi lõi.

Việc đa biến đổi lõi có thể thể hiện phương pháp biến đổi sử dụng bổ sung biến đổi cosin rời rạc (discrete cosine transform, DCT) loại 2 và biến đổi sin rời rạc (discrete sine transform, DST) loại 7, DCT loại 8, và/hoặc DST loại 1. Nghĩa là, việc đa biến đổi lõi có thể thể hiện phương pháp biến đổi để biến đổi tín hiệu phần dư (hoặc khối phần dư) của miền không gian thành các hệ số biến đổi (hoặc các hệ số biến đổi sơ cấp) của miền tần số dựa trên nhiều các nhân biến đổi được lựa chọn từ trong số DCT loại 2, DST

loại 7, DCT loại 8 và DST loại 1. Ở đây, các hệ số biến đổi sơ cấp có thể được gọi là các hệ số biến đổi theo thời gian từ quan điểm bộ biến đổi.

Nghĩa là, khi phương pháp biến đổi thông thường được áp dụng, các hệ số biến đổi có thể được tạo ra bằng việc áp dụng biến đổi từ miền không gian cho miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT loại 2. Tuy nhiên, khi việc đa biến đổi lõi được áp dụng, các hệ số biến đổi (hoặc các hệ số biến đổi sơ cấp) có thể được tạo ra bằng việc áp dụng biến đổi từ miền không gian cho miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT loại 2, DST loại 7, DCT loại 8, và/hoặc DST loại 1. Ở đây, DCT loại 2, DST loại 7, DCT loại 8, và DST loại 1 có thể được gọi là các loại biến đổi, các nhân biến đổi, hoặc các lõi biến đổi. Các loại DCT/DST này có thể được xác định dựa trên các chức năng cơ bản.

Nếu việc đa biến đổi lõi được thực hiện, thì nhân biến đổi dọc và nhân biến đổi ngang cho khối mục tiêu có thể được lựa chọn từ trong số các nhân biến đổi, việc biến đổi dọc cho khối mục tiêu có thể được thực hiện dựa trên nhân biến đổi dọc, và việc biến đổi ngang cho khối mục tiêu có thể được thực hiện dựa trên nhân biến đổi ngang. Ở đây, việc biến đổi ngang có thể thể hiện việc biến đổi cho các thành phần ngang của khối mục tiêu, và việc biến đổi dọc có thể thể hiện việc biến đổi cho các thành phần dọc của khối mục tiêu. Nhân biến đổi dọc/nhân biến đổi ngang có thể được xác định thích ứng dựa trên chế độ dự đoán và/hoặc chỉ số biến đổi của khối mục tiêu (CU hoặc khối con) gồm khối phần dư.

Ngoài ra, theo một ví dụ, nếu việc biến đổi sơ cấp được thực hiện bằng việc áp dụng MTS, mối tương quan ánh xạ cho các nhân biến đổi có thể được thiết đặt bằng việc thiết đặt các chức năng cơ sở cụ thể tới các giá trị được xác định trước và kết hợp các chức năng cơ sở cần được áp dụng trong việc biến đổi dọc hoặc việc biến đổi ngang. Ví dụ, khi nhân biến đổi ngang được diễn đạt dưới dạng $trTypeHor$ và việc dự đoán dọc nhân biến đổi được diễn đạt dưới dạng $trTypeVer$, giá trị $trTypeHor$ hoặc $trTypeVer$ là 0 có thể được thiết đặt cho DCT2, giá trị $trTypeHor$ hoặc $trTypeVer$ là 1 có thể được

thiết đặt cho DST-7, và giá trị trTypeHor hoặc trTypeVer là 2 có thể được thiết đặt cho DCT-8.

Trong trường hợp này, thông tin chỉ số MTS có thể được mã hóa và được báo hiệu tới thiết bị giải mã để chỉ ra bất kỳ trong số nhiều tập nhân biến đổi. Ví dụ, chỉ số MTS là 0 có thể chỉ ra rằng cả giá trị trTypeHor và giá trị trTypeVer là 0, chỉ số MTS là 1 có thể chỉ ra rằng cả giá trị trTypeHor và giá trị trTypeVer là 1, chỉ số MTS là 2 có thể chỉ ra rằng giá trị trTypeHor là 2 và giá trị trTypeVer là 1, chỉ số MTS là 3 có thể chỉ ra rằng giá trị trTypeHor là 1 và giá trị trTypeVer là 2, và chỉ số MTS là 4 có thể chỉ ra rằng cả giá trị trTypeHor và giá trị trTypeVer là 2.

Trong một ví dụ, các tập nhân biến đổi theo thông tin chỉ số MTS được minh họa trong bảng sau đây.

[Bảng 1]

tu_mts_idx[x0][y0]	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

Bộ biến đổi có thể dẫn xuất các hệ số biến đổi (thứ cấp) được cải biến bằng việc thực hiện việc biến đổi thứ cấp dựa trên các hệ số biến đổi (sơ cấp) (S520). Việc biến đổi sơ cấp là biến đổi từ miền không gian thành miền tần số, và việc biến đổi thứ cấp dùng để chỉ việc biến đổi thành biểu thức nén thêm bằng việc sử dụng sự tương liên tồn tại giữa các hệ số biến đổi (sơ cấp). Việc biến đổi thứ cấp có thể gồm việc biến đổi không thể phân tách được. Trong trường hợp này, việc biến đổi thứ cấp có thể được gọi là việc biến đổi thứ cấp không thể phân tách được (non-separable secondary transform, NSST), hoặc việc biến đổi thứ cấp không thể phân tách được phụ thuộc vào chế độ (mode-dependent non-separable secondary transform, MDNSST). Việc biến đổi thứ cấp không thể phân tách được có thể thể hiện việc biến đổi mà tạo thành các hệ số biến đổi

được cải biến (hoặc các hệ số biến đổi thứ cấp) cho tín hiệu phần dư bằng việc biến đổi thứ cấp, dựa trên ma trận biến đổi không thể phân tách được, các hệ số biến đổi (sơ cấp) được dẫn xuất qua việc biến đổi sơ cấp. Tại thời điểm này, việc biến đổi dọc và việc biến đổi ngang có thể không được áp dụng tách biệt (hoặc các biến đổi ngang và dọc có thể không được áp dụng một cách độc lập) cho các hệ số biến đổi (sơ cấp), nhưng các việc biến đổi có thể được áp dụng cùng lúc dựa trên ma trận biến đổi không thể phân tách được. Nói cách khác, việc biến đổi thứ cấp không thể phân tách được có thể thể hiện phương pháp biến đổi trong đó không được áp dụng tách biệt trong việc dự đoán dọc và việc dự đoán ngang cho các hệ số biến đổi (sơ cấp), và ví dụ, các tín hiệu hai chiều (các hệ số biến đổi) được sắp xếp lại thành tín hiệu một chiều qua hướng được xác định nhất định (ví dụ, chiều hàng thứ nhất hoặc chiều cột thứ nhất), và sau đó các hệ số biến đổi được cải biến (hoặc các hệ số biến đổi thứ cấp) được tạo ra dựa trên ma trận biến đổi không thể phân tách được. Ví dụ, theo thứ tự hàng thứ nhất, $M \times N$ khối được bố trí theo đường theo thứ tự hàng thứ nhất, hàng thứ hai, ..., và hàng thứ N . Theo thứ tự cột thứ nhất, $M \times N$ khối được bố trí theo đường theo thứ tự cột thứ nhất, cột thứ hai, ..., và cột thứ N . Việc biến đổi thứ cấp không thể phân tách được có thể được áp dụng cho vùng trên cùng bên trái của khối được tạo cấu hình với các hệ số biến đổi (sơ cấp) (sau đây, có thể được gọi là khối hệ số biến đổi). Ví dụ, nếu độ rộng (W) và độ cao (H) của khối hệ số biến đổi là đều bằng với hoặc lớn hơn 8, việc biến đổi thứ cấp không thể phân tách 8×8 có thể được áp dụng cho vùng 8×8 trên cùng bên trái của khối hệ số biến đổi. Ngoài ra, nếu độ rộng (W) và độ cao (H) của khối hệ số biến đổi là đều bằng với hoặc lớn hơn 4, và độ rộng (W) hoặc độ cao (H) của khối hệ số biến đổi là nhỏ hơn 8, thì việc biến đổi thứ cấp không thể phân tách 4×4 có thể được áp dụng cho vùng $\min(8, W) \times \min(8, H)$ trên cùng bên trái của khối hệ số biến đổi. Tuy nhiên, phương án không bị giới hạn ở đó, và ví dụ, ngay cả nếu chỉ điều kiện rằng độ rộng (W) hoặc độ cao (H) của khối hệ số biến đổi là bằng hoặc lớn hơn 4 được thỏa mãn, việc biến đổi thứ cấp không thể phân tách 4×4 có thể được áp dụng cho vùng $\min(8, W) \times \min(8, H)$ trên cùng bên trái của khối hệ số biến đổi.

Cụ thể là, ví dụ, nếu khối đầu vào 4x4 được sử dụng, việc biến đổi thứ cấp không thể phân tách được có thể được thực hiện như sau đây.

Khối đầu vào 4x4 X có thể được biểu diễn như sau đây.

[Phương trình 1]

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

Nếu X được biểu diễn dưới dạng vector, vector $\vec{X}$ có thể được biểu diễn như dưới đây.

[Phương trình 2]

$$\vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T$$

Trong Phương trình 2, vector $\vec{X}$ là vector một chiều thu nhận được bằng việc sắp xếp lại khối hai chiều X của Phương trình 1 theo thứ tự hàng thứ nhất.

Trong trường hợp này, việc biến đổi không thể phân tách thứ cấp có thể được tính toán như dưới đây.

[Phương trình 3]

$$\vec{F} = T \cdot \vec{X}$$

Trong phương trình này, $\vec{F}$ biểu diễn vector hệ số biến đổi, và T biểu diễn ma trận biến đổi (không thể phân tách) 16x16.

Qua Phương trình 3 nêu trên, vector hệ số biến đổi $\vec{F}$ 16x1 có thể được dẫn xuất, và $\vec{F}$ có thể có thể được tổ chức lại thành khối 4x4 qua thứ tự quét (ngang, dọc, chéo và tương tự). Tuy nhiên, việc tính toán được mô tả ở trên là ví dụ, và việc biến đổi

hypercube-Givens (hypercube-Givens transform, HyGT) hoặc tương tự có thể được sử dụng cho việc tính toán của việc biến đổi thứ cấp không thể phân tách được để làm giảm độ phức tạp tính toán của việc biến đổi thứ cấp không thể phân tách được.

Trong khi đó, trong việc biến đổi thứ cấp không thể phân tách được, nhân biến đổi (hoặc lõi biến đổi, loại biến đổi) có thể được lựa chọn để phụ thuộc vào chế độ. Trong trường hợp này, chế độ có thể gồm chế độ dự đoán nội và/hoặc chế độ dự đoán liên.

Như được mô tả ở trên, việc biến đổi thứ cấp không thể phân tách được có thể được thực hiện dựa trên việc biến đổi 8×8 hoặc việc biến đổi 4×4 được xác định dựa trên độ rộng (W) và độ cao (H) của khối hệ số biến đổi. Việc biến đổi 8×8 dùng để chỉ việc biến đổi mà là có thể ứng dụng cho vùng 8×8 được gồm trong khối hệ số biến đổi khi cả W và H là bằng hoặc lớn hơn 8, và vùng 8×8 có thể là vùng 8×8 trên cùng bên trái trong khối hệ số biến đổi. Tương tự, việc biến đổi 4×4 dùng để chỉ việc biến đổi mà là có thể ứng dụng cho vùng 4×4 được gồm trong khối hệ số biến đổi khi cả W và H là bằng hoặc lớn hơn 4, và vùng 4×4 có thể là vùng 4×4 trên cùng bên trái trong khối hệ số biến đổi. Ví dụ, ma trận nhân biến đổi 8×8 có thể là ma trận $64 \times 64 / 16 \times 64$, và ma trận nhân biến đổi 4×4 có thể là ma trận $16 \times 16 / 8 \times 16$.

Ở đây, để lựa chọn nhân biến đổi phụ thuộc chế độ, hai nhân biến đổi thứ cấp không thể phân tách mỗi tập biến đổi cho việc biến đổi thứ cấp không thể phân tách được có thể tạo cấu hình cho cả việc biến đổi 8×8 và việc biến đổi 4×4 , và có thể có bốn tập biến đổi. Nghĩa là, bốn tập biến đổi có thể tạo cấu hình cho việc biến đổi 8×8 , và bốn tập biến đổi có thể tạo cấu hình cho việc biến đổi 4×4 . Trong trường hợp này, mỗi trong số bốn tập biến đổi cho việc biến đổi 8×8 có thể gồm hai nhân biến đổi 8×8 , và mỗi trong số bốn tập biến đổi cho việc biến đổi 4×4 có thể gồm hai nhân biến đổi 4×4 .

Tuy nhiên, đối với kích thước của việc biến đổi, nghĩa là, kích thước của vùng mà với nó việc biến đổi được áp dụng, có thể là, ví dụ, kích thước ngoài 8×8 hoặc $4 \times$

4, số lượng của các tập có thể là n , và số lượng của các nhân biến đổi trong mỗi tập có thể là k .

Tập biến đổi có thể được gọi là tập NSST hoặc tập LFNST. Tập cụ thể trong số các tập biến đổi có thể được lựa chọn, ví dụ, dựa trên chế độ dự đoán nội của khối hiện tại (CU hoặc khối con). Việc biến đổi không thể phân tách tần số thấp (low-frequency non-separable transform, LFNST) có thể là ví dụ của việc biến đổi không thể phân tách được giảm, mà sẽ được mô tả sau, và biểu diễn việc biến đổi không thể phân tách được cho thành phần tần số thấp.

Để tham chiếu, ví dụ, chế độ dự đoán nội có thể gồm hai chế độ dự đoán nội không có hướng (hoặc không có góc) và 65 chế độ dự đoán nội có hướng (hoặc có góc). Các chế độ dự đoán nội không có hướng có thể gồm chế độ dự đoán nội phẳng số 0 và chế độ dự đoán nội DC số 1, và các chế độ dự đoán nội có hướng có thể gồm 65 chế độ dự đoán nội các số từ 2 đến 66. Tuy nhiên, đây là ví dụ, và bản mô tả này có thể được áp dụng ngay cả khi số lượng của các chế độ dự đoán nội là khác nhau. Trong khi đó, trong một số trường hợp, chế độ dự đoán nội số 67 có thể được sử dụng thêm, và chế độ dự đoán nội số 67 có thể thể hiện chế độ mô hình tuyến tính (linear model, LM).

Fig.6 thể hiện ví dụ các chế độ hướng nội của 65 chế độ dự đoán.

Tham khảo đến Fig.6, trên cơ sở của chế độ dự đoán nội 34 có hướng dự đoán chéo hướng lên bên trái, các chế độ dự đoán nội có thể được chia thành các chế độ dự đoán nội có hướng ngang và các chế độ dự đoán nội có hướng dọc. Trên Fig.6, H và V biểu thị hướng ngang và hướng dọc, một cách tương ứng, và các số -32 đến 32 chỉ ra các sự dịch chuyển trong 1/32 đơn vị trên vị trí lưới mẫu. Các số này có thể thể hiện sự dịch chuyển cho giá trị chỉ số chế độ. Các chế độ dự đoán nội 2 đến 33 có hướng ngang, và các chế độ dự đoán nội 34 đến 66 có hướng dọc. Nói nghiêm ngặt, chế độ dự đoán nội 34 có thể được coi là không phải ngang cũng như dọc, nhưng có thể được phân loại như thuộc về hướng ngang trong việc xác định tập biến đổi của việc biến đổi thứ cấp. Đó là bởi vì dữ liệu đầu vào được chuyển vị để được sử dụng cho chế độ hướng dọc đối

xúng trên cơ sở của chế độ dự đoán nội 34, và phương pháp căn chỉnh dữ liệu đầu vào cho chế độ ngang được sử dụng cho chế độ dự đoán nội 34. Chuyển đổi dữ liệu đầu vào có nghĩa là các hàng và các cột của dữ liệu khối $M \times N$ hai chiều được chuyển thành dữ liệu $N \times M$. Chế độ dự đoán nội 18 và chế độ dự đoán nội 50 có thể thể hiện chế độ dự đoán nội ngang và chế độ dự đoán nội dọc, một cách tương ứng, và chế độ dự đoán nội 2 có thể được gọi là chế độ dự đoán nội chéo hướng lên bên phải bởi vì chế độ dự đoán nội 2 có điểm ảnh tham chiếu bên trái và thực hiện việc dự đoán theo hướng lên bên phải. Tương tự, chế độ dự đoán nội 34 có thể được gọi là chế độ dự đoán nội chéo hướng xuống bên phải, và chế độ dự đoán nội 66 có thể được gọi là chế độ dự đoán nội chéo hướng xuống bên trái.

Theo một ví dụ, bốn tập biến đổi theo chế độ dự đoán nội có thể được ánh xạ, ví dụ, như được thể hiện trong bảng sau đây.

[Bảng 2]

predModeIntra	lnstTrSetIdx
predModeIntra < 0	1
$0 \leq \text{predModeIntra} \leq 1$	0
$2 \leq \text{predModeIntra} \leq 12$	1
$13 \leq \text{predModeIntra} \leq 23$	2
$24 \leq \text{predModeIntra} \leq 44$	3
$45 \leq \text{predModeIntra} \leq 55$	2
$56 \leq \text{predModeIntra} \leq 80$	1

Như được thể hiện trong Bảng 2, bất kỳ trong số bốn tập biến đổi, nghĩa là, lnstTrSetIdx, có thể được ánh xạ bất kỳ trong số bốn chỉ số, nghĩa là, 0 đến 3, theo chế độ dự đoán nội.

Khi được xác định rằng tập cụ thể được sử dụng cho việc biến đổi không thể phân tách được, một trong số k nhân biến đổi trong tập cụ thể có thể được lựa chọn qua chỉ số biến đổi thứ cấp không thể phân tách được. Thiết bị mã hóa có thể dẫn xuất chỉ số biến đổi thứ cấp không thể phân tách được chỉ ra nhân biến đổi cụ thể dựa trên việc kiểm tra tốc độ-nhiều loạn (rate-distortion, RD) và có thể báo hiệu chỉ số biến đổi thứ cấp không thể phân tách được tới thiết bị giải mã. Thiết bị giải mã có thể lựa chọn một

trong số k nhân biến đổi trong tập cụ thể dựa trên chỉ số biến đổi thứ cấp không thể phân tách được. Ví dụ, giá trị chỉ số $l_{\text{nst}} 0$ có thể dùng để chỉ nhân biến đổi thứ cấp không thể phân tách thứ nhất, giá trị chỉ số $l_{\text{nst}} 1$ có thể dùng để chỉ nhân biến đổi thứ cấp không thể phân tách thứ hai, và giá trị chỉ số $l_{\text{nst}} 2$ có thể dùng để chỉ nhân biến đổi thứ cấp không thể phân tách thứ ba. Mặt khác, giá trị chỉ số $l_{\text{nst}} 0$ có thể chỉ ra rằng việc biến đổi thứ cấp không thể phân tách thứ nhất không được áp dụng cho khối mục tiêu, và các giá trị chỉ số $l_{\text{nst}} 1$ đến 3 có thể chỉ ra ba nhân biến đổi.

Bộ biến đổi có thể thực hiện việc biến đổi thứ cấp không thể phân tách được dựa trên các nhân biến đổi được lựa chọn, và có thể thu nhận các hệ số biến đổi (thứ cấp) được cải biến. Như được mô tả ở trên, các hệ số biến đổi được cải biến có thể được dẫn xuất như các hệ số biến đổi được lượng tử hóa qua bộ lượng tử hóa, và có thể được mã hóa và được báo hiệu tới thiết bị giải mã và được chuyển tới bộ khử lượng tử hóa/bộ biến đổi ngược trong thiết bị mã hóa.

Trong khi đó, như được mô tả ở trên, nếu việc biến đổi thứ cấp được bỏ qua, các hệ số biến đổi (sơ cấp), mà là đầu ra của việc biến đổi (có thể phân tách) sơ cấp, có thể được dẫn xuất như các hệ số biến đổi được lượng tử hóa qua bộ lượng tử hóa như được mô tả ở trên, và có thể được mã hóa và được báo hiệu tới thiết bị giải mã và được chuyển tới bộ khử lượng tử hóa/bộ biến đổi ngược trong thiết bị mã hóa.

Bộ biến đổi ngược có thể thực hiện chuỗi các thủ tục theo thứ tự ngược với thứ tự trong đó chúng đã được thực hiện trong bộ biến đổi được mô tả ở trên. Bộ biến đổi ngược có thể nhận các hệ số bộ biến đổi (được khử lượng tử hóa), và dẫn xuất các hệ số biến đổi (sơ cấp) bằng việc thực hiện việc biến đổi (ngược) thứ cấp (S550), và có thể thu nhận khối phần dư (các mẫu phần dư) bằng việc thực hiện việc biến đổi (ngược) sơ cấp trên các hệ số biến đổi (sơ cấp) (S560). Theo sự kết nối này, các hệ số biến đổi sơ cấp có thể được gọi là các hệ số biến đổi được cải biến từ quan điểm bộ biến đổi ngược. Như được mô tả ở trên, thiết bị mã hóa và thiết bị giải mã có thể tạo ra khối được xây dựng lại dựa trên khối phần dư và khối được dự đoán, và có thể tạo ra ảnh được xây

dựng lại dựa trên khối được xây dựng lại.

Thiết bị giải mã có thể còn gồm bộ xác định ứng dụng biến đổi ngược thứ cấp (hoặc bộ phận để xác định xem có áp dụng việc biến đổi ngược thứ cấp hay không) và bộ xác định biến đổi ngược thứ cấp (hoặc bộ phận để xác định việc biến đổi ngược thứ cấp). Bộ xác định ứng dụng biến đổi ngược thứ cấp có thể xác định xem có áp dụng việc biến đổi ngược thứ cấp hay không. Ví dụ, việc biến đổi ngược thứ cấp có thể là NSST, RST, hoặc LFNST và bộ xác định ứng dụng biến đổi ngược thứ cấp có thể xác định xem có áp dụng việc biến đổi ngược thứ cấp hay không dựa trên cờ biến đổi thứ cấp được thu nhận bằng việc phân tích cú pháp luồng bit. Trong một ví dụ khác, bộ xác định ứng dụng biến đổi ngược thứ cấp có thể xác định xem có áp dụng việc biến đổi ngược thứ cấp hay không dựa trên hệ số biến đổi của khối phần dư.

Bộ xác định biến đổi ngược thứ cấp có thể xác định việc biến đổi ngược thứ cấp. Trong trường hợp này, bộ xác định biến đổi ngược thứ cấp có thể xác định việc biến đổi ngược thứ cấp được áp dụng cho khối hiện tại dựa trên tập biến đổi LFNST (NSST hoặc RST) được định rõ theo chế độ dự đoán nội. Trong một phương án, phương pháp xác định biến đổi thứ cấp có thể được xác định phụ thuộc vào phương pháp xác định biến đổi sơ cấp. Các kết hợp khác nhau của các biến đổi sơ cấp và các biến đổi thứ cấp có thể được xác định theo chế độ dự đoán nội. Ngoài ra, trong một ví dụ, bộ xác định biến đổi ngược thứ cấp có thể xác định vùng mà với nó việc biến đổi ngược thứ cấp được áp dụng dựa trên kích thước của khối hiện tại.

Trong khi đó, như được mô tả ở trên, nếu việc biến đổi (ngược) thứ cấp được bỏ qua, các hệ số biến đổi (được khử lượng tử hóa) có thể được nhận, việc biến đổi ngược (có thể phân tách) sơ cấp có thể được thực hiện, và khối phần dư (các mẫu phần dư) có thể được thu nhận. Như được mô tả ở trên, thiết bị mã hóa và thiết bị giải mã có thể tạo ra khối được xây dựng lại dựa trên khối phần dư và khối được dự đoán, và có thể tạo ra ảnh được xây dựng lại dựa trên khối được xây dựng lại.

Trong khi đó, trong sáng chế, việc biến đổi thứ cấp dư (reduced secondary

transform, RST) trong đó kích thước của ma trận biến đổi (nhân) được giảm có thể được áp dụng trong khái niệm của NSST để làm giảm lượng tính toán và bộ nhớ được yêu cầu cho việc biến đổi thứ cấp không thể phân tách được.

Trong khi đó, nhân biến đổi, ma trận biến đổi, và hệ số cấu thành ma trận nhân biến đổi, nghĩa là, hệ số nhân hoặc hệ số ma trận, được mô tả trong sáng chế có thể được thể hiện trong 8 bit. Điều này có thể là điều kiện cho việc thực hiện trong thiết bị giải mã và thiết bị mã hóa, và có thể làm giảm lượng bộ nhớ được yêu cầu để lưu trữ nhân biến đổi với sự suy giảm hiệu quả mà có thể được cung cấp một cách hợp lý so sánh với 9 bit hoặc 10 bit hiện có. Ngoài ra, việc thể hiện của ma trận nhân trong 8 bit có thể cho phép hệ số nhân nhỏ được sử dụng, và có thể thích hợp hơn cho các lệnh đa dữ liệu lệnh đơn (single instruction multiple data, SIMD) được sử dụng cho việc thực hiện phần mềm tối ưu.

Trong bản mô tả này, thuật ngữ “RST” có thể nghĩa là việc biến đổi mà được thực hiện trên các mẫu phần dư cho khối mục tiêu dựa trên ma trận biến đổi mà kích thước của nó được giảm theo hệ số được giảm. Trong trường hợp thực hiện việc biến đổi giảm, lượng tính toán được yêu cầu cho việc biến đổi có thể được giảm do việc giảm về kích thước của ma trận biến đổi. Nghĩa là, RST có thể được sử dụng để giải quyết vấn đề độ phức tạp tính toán xảy ra ở việc biến đổi không thể phân tách được hoặc việc biến đổi của khối có kích thước lớn.

RST có thể được gọi bởi nhiều thuật ngữ khác nhau, chẳng hạn như việc biến đổi được giảm, việc biến đổi thứ cấp được giảm, việc biến đổi giảm, việc biến đổi được đơn giản hóa, việc biến đổi đơn giản, và tương tự, và tên mà RST có thể được gọi là không bị giới hạn ở các ví dụ được liệt kê. Mặt khác, do RST chủ yếu được thực hiện trong vùng tần số thấp gồm hệ số khác không trong khối biến đổi, có thể được gọi là việc biến đổi không thể phân tách tần số thấp (Low-Frequency Non-Separable Transform, LFNST). Hệ số biến đổi có thể được gọi là chỉ số LFNST.

Trong khi đó, khi việc biến đổi ngược thứ cấp được thực hiện dựa trên RST, bộ

biến đổi ngược 235 của thiết bị mã hóa 200 và bộ biến đổi ngược 322 của thiết bị giải mã 300 có thể gồm bộ biến đổi thứ cấp được giảm ngược mà dẫn xuất các hệ số biến đổi được cải biến dựa trên RST ngược của các hệ số biến đổi, và bộ biến đổi sơ cấp ngược mà dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên việc biến đổi sơ cấp ngược của các hệ số biến đổi được cải biến. Việc biến đổi sơ cấp ngược dùng để chỉ việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Trong sáng chế, dẫn xuất hệ số biến đổi dựa trên việc biến đổi có thể dùng để chỉ dẫn xuất hệ số biến đổi bằng việc áp dụng việc biến đổi.

Fig.7 là sơ đồ minh họa RST theo một phương án của sáng chế.

Trong sáng chế, "khối mục tiêu" có thể dùng để chỉ khối hiện tại cần được lập mã, khối phần dư, hoặc khối biến đổi.

Trong RST theo một ví dụ, vector chiều N có thể được ánh xạ tới vector chiều R được đặt trong không gian khác, sao cho ma trận biến đổi được giảm có thể được xác định, trong đó R là nhỏ hơn N. N có thể nghĩa là bình phương của chiều dài của cạnh của khối mà cho nó việc biến đổi được áp dụng, hoặc tổng số lượng của các hệ số biến đổi tương ứng với khối mà cho nó việc biến đổi được áp dụng, và hệ số được giảm có thể nghĩa là giá trị R/N . Hệ số được giảm có thể được gọi là hệ số được giảm, hệ số giảm, hệ số được đơn giản hóa, hệ số đơn giản hoặc nhiều thuật ngữ khác. Trong khi đó, R có thể được gọi là hệ số được giảm, nhưng theo các hoàn cảnh, hệ số được giảm có thể nghĩa là R. Ngoài ra, theo các hoàn cảnh, hệ số được giảm có thể nghĩa là giá trị N/R .

Trong một ví dụ, hệ số được giảm hoặc hệ số được giảm có thể được báo hiệu qua luồng bit, nhưng ví dụ không bị giới hạn ở đây. Ví dụ, giá trị được xác định trước cho hệ số được giảm hoặc hệ số được giảm có thể được lưu trữ trong mỗi trong số thiết bị mã hóa 200 và thiết bị giải mã 300, và trong trường hợp này, hệ số được giảm hoặc hệ số được giảm có thể không được báo hiệu một cách tách biệt.

Kích thước của ma trận biến đổi được giảm theo một ví dụ có thể là $R \times N$ nhỏ

hơn $N \times N$, kích thước của ma trận biến đổi thông thường, và có thể được xác định như trong Phương trình 4 dưới đây.

[Phương trình 4]

$$T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1N} \\ t_{21} & t_{22} & t_{23} & & t_{2N} \\ & \vdots & & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \cdots & t_{RN} \end{bmatrix}$$

Ma trận T trong khối biến đổi được giảm được thể hiện trên Fig.7(a) có thể nghĩa là ma trận $T_{R \times N}$ của Phương trình 4. Như được thể hiện trên Fig.7(a), khi ma trận biến đổi được giảm $T_{R \times N}$ được nhân với các mẫu phần dư cho khối mục tiêu, các hệ số biến đổi cho khối mục tiêu có thể được dẫn xuất.

Trong một ví dụ, nếu kích thước của khối mà cho nó việc biến đổi được áp dụng là 8×8 và $R = 16$ (nghĩa là, $R/N = 16/64 = 1/4$), thì RST theo Fig.7(a) có thể được thể hiện dưới dạng thao tác ma trận như được thể hiện trong Phương trình 5 dưới đây. Trong trường hợp này, bộ nhớ và phép tính nhân có thể được giảm tới xấp xỉ $1/4$ bởi hệ số được giảm.

Trong sáng chế, thao tác ma trận có thể được hiểu là thao tác nhân vector cột cho ma trận, được bố trí trên bên trái của vector cột, để thu nhận vector cột.

[Phương trình 5]

$$\begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \cdots & t_{1,64} \\ t_{2,1} & t_{2,2} & t_{2,3} & & t_{2,64} \\ & \vdots & & \ddots & \vdots \\ t_{16,1} & t_{16,2} & t_{16,3} & \cdots & t_{16,64} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{64} \end{bmatrix}$$

Trong Phương trình 5, r_1 đến r_{64} có thể biểu diễn các mẫu phần dư cho khối mục tiêu và có thể cụ thể là các hệ số biến đổi được tạo ra bằng việc áp dụng việc biến đổi sơ cấp. Như là kết quả tính toán của Phương trình 5 các hệ số biến đổi c_i cho khối mục tiêu có thể được dẫn xuất, và phương pháp dẫn xuất c_i có thể là như trong Phương trình 6.

[Phương trình 6]

```
for i from 1 to R:
```

```
     $c_i = 0$ 
```

```
    for j from 1 to N
```

```
         $c_i += t_{ij} * r_j$ 
```

Như là kết quả tính toán của Phương trình 6, các hệ số biến đổi c_1 đến c_R cho khối mục tiêu có thể được dẫn xuất. Nghĩa là, khi $R=16$, các hệ số biến đổi c_1 đến c_{16} cho khối mục tiêu có thể được dẫn xuất. Nếu, thay vì RST, việc biến đổi thông thường được áp dụng và ma trận biến đổi có kích thước 64×64 ($N \times N$) được nhân với các mẫu phần dư có kích thước 64×1 ($N \times 1$), thì chỉ 16 (R) hệ số biến đổi được dẫn xuất cho khối mục tiêu bởi vì RST được áp dụng, mặc dù 64 (N) hệ số biến đổi được dẫn xuất cho khối mục tiêu. Do tổng số lượng của các hệ số biến đổi cho khối mục tiêu được giảm từ N thành R , lượng dữ liệu được truyền bởi thiết bị mã hóa 200 tới thiết bị giải mã 300 giảm, do đó hiệu quả của việc truyền giữa thiết bị mã hóa 200 và thiết bị giải mã 300 có thể được cải thiện.

Khi được xem xét từ quan điểm kích thước của ma trận biến đổi, kích thước của ma trận biến đổi thông thường là 64×64 ($N \times N$), nhưng kích thước của ma trận biến đổi được giảm được giảm tới 16×64 ($R \times N$), nên việc sử dụng bộ nhớ trong trường hợp thực hiện RST có thể được giảm tới tỷ lệ R/N khi được so sánh với trường hợp thực hiện việc biến đổi thông thường. Ngoài ra, khi so sánh với số lượng của các phép tính nhân $N \times N$ trong trường hợp sử dụng ma trận biến đổi thông thường, việc sử dụng của ma trận biến đổi được giảm có thể làm giảm số lượng của các phép tính nhân tới tỷ lệ R/N ($R \times N$).

Trong một ví dụ, bộ biến đổi 232 của thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi cho khối mục tiêu bằng việc thực hiện việc biến đổi sơ cấp và việc biến đổi thứ cấp dựa trên RST trên các mẫu phần dư cho khối mục tiêu. Các hệ số biến đổi này có thể được truyền đến bộ biến đổi ngược của thiết bị giải mã 300, và bộ biến đổi ngược

322 của thiết bị giải mã 300 có thể dẫn xuất các hệ số biến đổi được cải biến dựa trên việc biến đổi thứ cấp được giảm (RST) ngược cho các hệ số biến đổi, và có thể dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên việc biến đổi sơ cấp ngược cho các hệ số biến đổi được cải biến.

Kích thước của ma trận RST ngược $T_{N \times R}$ theo một ví dụ là $N \times R$ nhỏ hơn kích thước $N \times N$ của ma trận biến đổi ngược thông thường, và là trong mối quan hệ chuyển vị với ma trận biến đổi được giảm $T_{R \times N}$ được thể hiện trong Phương trình 4.

Ma trận T^T trong khối biến đổi ngược được giảm được thể hiện trên Fig.7(b) có thể nghĩa là ma trận RST ngược $T_{R \times N}^T$ (chỉ số trên T có nghĩa là chuyển vị). Khi ma trận RST ngược $T_{R \times N}^T$ được nhân với các hệ số biến đổi cho khối mục tiêu như được thể hiện trên Fig.7(b), các hệ số biến đổi được cải biến cho khối mục tiêu hoặc các mẫu phần dư cho khối hiện tại có thể được dẫn xuất. Ma trận RST ngược $T_{R \times N}^T$ có thể được thể hiện dưới dạng $(T_{R \times N}^T)_{N \times R}$.

Cụ thể hơn là, khi RST ngược được áp dụng làm việc biến đổi ngược thứ cấp, các hệ số biến đổi được cải biến cho khối mục tiêu có thể được dẫn xuất khi ma trận RST ngược $T_{R \times N}^T$ được nhân với các hệ số biến đổi cho khối mục tiêu. Trong khi đó, RST ngược có thể được áp dụng làm việc biến đổi sơ cấp ngược, và trong trường hợp này, các mẫu phần dư cho khối mục tiêu có thể được dẫn xuất khi ma trận RST ngược $T_{R \times N}^T$ được nhân với các hệ số biến đổi cho khối mục tiêu.

Trong một ví dụ, nếu kích thước của khối mà cho nó việc biến đổi ngược được áp dụng là 8×8 và $R = 16$ (nghĩa là, $R/N = 16/64 = 1/4$), thì RST theo Fig.7(b) có thể được thể hiện dưới dạng thao tác ma trận như được thể hiện trong Phương trình 7 dưới đây.

[Phương trình 7]

$$\begin{bmatrix} t_{1,1} & t_{2,1} & & t_{16,1} \\ t_{1,2} & t_{2,2} & \dots & t_{16,2} \\ t_{1,3} & t_{2,3} & & t_{16,3} \\ \vdots & \vdots & & \vdots \\ & \vdots & \ddots & \vdots \\ t_{1,64} & t_{2,64} & \dots & t_{16,64} \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{16} \end{bmatrix}$$

Trong Phương trình 7, c_1 đến c_{16} có thể biểu diễn các hệ số biến đổi cho khối mục tiêu. Như là kết quả tính toán của Phương trình 7, r_i biểu diễn các hệ số biến đổi được cải biến cho khối mục tiêu hoặc các mẫu phần dư cho khối mục tiêu có thể được dẫn xuất, và quy trình dẫn xuất r_i có thể là như trong Phương trình 8.

[Phương trình 8]

```

For i from 1 to N
   $r_i = 0$ 
  for j from 1 to R
     $r_i += t_{ji} * c_j$ 
  
```

Như là kết quả tính toán của Phương trình 8, r_1 đến r_N biểu diễn các hệ số biến đổi được cải biến cho khối mục tiêu hoặc các mẫu phần dư cho khối mục tiêu có thể được dẫn xuất. Khi được xem xét từ quan điểm kích thước của ma trận biến đổi ngược, kích thước của ma trận biến đổi ngược thông thường là 64×64 ($N \times N$), nhưng kích thước của ma trận biến đổi ngược được giảm được giảm tới 64×16 ($R \times N$), nên việc sử dụng bộ nhớ trong trường hợp thực hiện RST ngược có thể được giảm tới tỷ lệ R/N khi được so sánh với trường hợp thực hiện việc biến đổi ngược thông thường. Ngoài ra, khi so sánh với số lượng của các phép tính nhân $N \times N$ trong trường hợp sử dụng ma trận biến đổi ngược thông thường, việc sử dụng ma trận biến đổi ngược được giảm có thể làm giảm số lượng của các phép tính nhân tới tỷ lệ R/N ($N \times R$).

Cấu hình tập biến đổi được thể hiện trong Bảng 2 cũng có thể được áp dụng cho 8×8 RST. Nghĩa là, 8×8 RST có thể được áp dụng theo tập biến đổi trong Bảng 2. Do một tập biến đổi gồm hai hoặc ba biến đổi (nhân) theo chế độ dự đoán nội, có thể tạo cấu hình để lựa chọn một trong số tới bốn biến đổi gồm mà trong trường hợp trong đó

không việc biến đổi thứ cấp nào được áp dụng. Trong việc biến đổi trong đó không việc biến đổi thứ cấp nào được áp dụng, nó có thể được xem xét để áp dụng ma trận xác thực. Giả định rằng các chỉ số 0, 1, 2, và 3 một cách tương ứng được gán cho bốn biến đổi (ví dụ, chỉ số 0 có thể được phân phối cho trường hợp trong đó ma trận xác thực được áp dụng, nghĩa là, trường hợp trong đó không có việc biến đổi thứ cấp nào được áp dụng), chỉ số biến đổi hoặc chỉ số lfnst dưới dạng phân tử cú pháp có thể được báo hiệu cho mỗi khối hệ số biến đổi, nhờ đó chỉ định việc biến đổi cần được áp dụng. Nghĩa là, đối với khối 8×8 trên cùng bên trái, thông qua chỉ số biến đổi, có thể chỉ định 8×8 RST trong cấu hình RST, hoặc chỉ định 8×8 lfnst khi LFNST được áp dụng. 8×8 lfnst và 8×8 RST dùng để chỉ các biến đổi có thể ứng dụng cho vùng 8×8 được gồm trong khối hệ số biến đổi khi cả W và H của khối mục tiêu cần được biến đổi là bằng hoặc lớn hơn 8, và vùng 8×8 có thể là vùng 8×8 trên cùng bên trái trong khối hệ số biến đổi. Tương tự, 4×4 lfnst và 4×4 RST dùng để chỉ các việc biến đổi có thể ứng dụng cho vùng 4×4 được gồm trong khối hệ số biến đổi khi cả W và H của khối mục tiêu là bằng hoặc lớn hơn 4, và vùng 4×4 có thể là vùng 4×4 trên cùng bên trái trong khối hệ số biến đổi.

Theo một phương án của sáng chế, cho việc biến đổi trong quy trình mã hóa, chỉ 48 mẫu dữ liệu có thể được lựa chọn và tối đa 16×48 ma trận nhân biến đổi có thể được áp dụng cho nó, thay vì áp dụng 16×64 ma trận nhân biến đổi cho 64 mẫu dữ liệu tạo thành vùng 8×8 . Ở đây, "tối đa" có nghĩa là m có giá trị tối đa là 16 trong $m \times 48$ ma trận nhân biến đổi để tạo ra m hệ số. Nghĩa là, khi RST được thực hiện bằng việc áp dụng $m \times 48$ ma trận nhân biến đổi ($m \leq 16$) cho vùng 8×8 , 48 mẫu dữ liệu được nhập và m hệ số được tạo ra. Khi m là 16, 48 mẫu dữ liệu được nhập và 16 hệ số được tạo ra. Nghĩa là, giả định rằng 48 mẫu dữ liệu tạo thành 48×1 vector, 16×48 ma trận và 48×1 vector được nhân lần lượt, nhờ đó tạo ra 16×1 vector. Ở đây, 48 mẫu dữ liệu tạo thành vùng 8×8 có thể được sắp xếp hợp lý, nhờ đó tạo thành 48×1 vector. Ví dụ, 48×1 vector có thể được xây dựng dựa trên 48 mẫu dữ liệu cấu thành vùng ngoại trừ vùng 4×4 dưới cùng bên phải trong số các vùng 8×8 . Ở đây, khi thao tác ma trận được thực

hiện bằng việc áp dụng tối đa 16×48 ma trận nhân biến đổi, 16 hệ số biến đổi được cải biến được tạo ra, và 16 hệ số biến đổi được cải biến có thể được sắp xếp trong vùng 4×4 trên cùng bên trái theo thứ tự quét, và vùng 4×4 trên cùng bên phải và vùng 4×4 dưới cùng bên trái có thể được điền với các số không.

Đối với việc biến đổi ngược trong quy trình giải mã, ma trận được chuyển vị của ma trận nhân biến đổi nêu trên có thể được sử dụng. Nghĩa là, khi RST ngược hoặc LFNST được thực hiện trong quy trình biến đổi ngược được thực hiện bởi thiết bị giải mã, dữ liệu hệ số đầu vào cho nó RST ngược được áp dụng được tạo cấu hình trong vector một chiều theo thứ tự sắp xếp được xác định trước, và vector hệ số được cải biến được thu nhận bằng việc nhân vector một chiều và ma trận RST ngược tương ứng ở bên trái của vector một chiều có thể được sắp xếp trong khối hai chiều theo thứ tự sắp xếp được xác định trước.

Tóm lại, trong quy trình biến đổi, khi RST hoặc LFNST được áp dụng cho vùng 8×8 , thao tác ma trận của 48 hệ số biến đổi trong các vùng trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của vùng 8×8 ngoại trừ vùng dưới cùng bên phải trong số các hệ số biến đổi trong vùng 8×8 và ma trận nhân biến đổi 16×48 . Đối với thao tác ma trận, 48 hệ số biến đổi được nhập trong mảng một chiều. Khi thao tác ma trận được thực hiện, 16 hệ số biến đổi được cải biến được dẫn xuất, và các hệ số biến đổi được cải biến có thể được sắp xếp trong vùng trên cùng bên trái của vùng 8×8 .

Ngược lại, trong quy trình biến đổi ngược, khi RST ngược hoặc LFNST được áp dụng cho vùng 8×8 , 16 hệ số biến đổi tương ứng với vùng trên cùng bên trái của vùng 8×8 trong số các hệ số biến đổi trong vùng 8×8 có thể được nhập vào trong mảng một chiều theo thứ tự quét và có thể được tiến hành thao tác ma trận với ma trận nhân biến đổi 48×16 . Nghĩa là, thao tác ma trận có thể được thể hiện dưới dạng (ma trận 48×16) * (vector hệ số biến đổi 16×1) = (vector hệ số biến đổi được cải biến 48×1). Ở đây, vector $n \times 1$ có thể được diễn giải là có cùng ý nghĩa như ma trận $n \times 1$ và có thể do đó được thể hiện dưới dạng vector cột $n \times 1$. Ngoài ra, * biểu thị phép nhân ma trận. Khi

thao tác ma trận được thực hiện, 48 hệ số biến đổi được cải biến có thể được dẫn xuất, và 48 hệ số biến đổi được cải biến có thể được sắp xếp trong các vùng trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của vùng 8x8 ngoại trừ vùng dưới cùng bên phải.

Khi việc biến đổi ngược thứ cấp là dựa trên RST, bộ biến đổi ngược 235 của thiết bị mã hóa 200 và bộ biến đổi ngược 322 của thiết bị giải mã 300 có thể gồm bộ biến đổi thứ cấp được giảm ngược để dẫn xuất các hệ số biến đổi được cải biến dựa trên RST ngược về các hệ số biến đổi và bộ biến đổi sơ cấp ngược để dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên việc biến đổi sơ cấp ngược trên các hệ số biến đổi được cải biến. Việc biến đổi sơ cấp ngược dùng để chỉ việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Trong sáng chế, dẫn xuất hệ số biến đổi dựa trên việc biến đổi có thể dùng để chỉ dẫn xuất hệ số biến đổi bằng việc áp dụng việc biến đổi.

Trong việc biến đổi không được phân tách được mô tả ở trên, LFNST, sẽ được mô tả chi tiết như sau đây. LFNST có thể gồm việc biến đổi xuôi bởi thiết bị mã hóa và việc biến đổi ngược bởi thiết bị giải mã.

Thiết bị mã hóa nhận kết quả (hoặc một phần của kết quả) được dẫn xuất sau khi áp dụng việc biến đổi (lỗi) sơ cấp dưới dạng đầu vào, và áp dụng việc biến đổi thứ cấp xuôi (biến đổi thứ cấp).

[Phương trình 9]

$$y = G^T x$$

Trong Phương trình 9, x và y là các đầu vào và các đầu ra của việc biến đổi thứ cấp, một cách tương ứng, và G là ma trận biểu diễn việc biến đổi thứ cấp, và các vector cơ bản biến đổi được cấu thành từ các vector cột. Trong trường hợp LFNST ngược, khi kích thước của ma trận biến đổi G được diễn đạt dưới dạng [số hàng x số cột], trong trường hợp LFNST xuôi, sự chuyển vị của ma trận G trở thành kích thước của G^T .

Đối với LFNST ngược, các kích thước của ma trận G là $[48 \times 16]$, $[48 \times 8]$, $[16 \times 16]$, $[16 \times 8]$, và ma trận $[48 \times 8]$ và ma trận $[16 \times 8]$ là các ma trận một phần mà lấy mẫu 8 vector cơ bản biến đổi từ bên trái của ma trận $[48 \times 16]$ và ma trận $[16 \times 16]$, một cách tương ứng.

Mặt khác, đối với LFNST xuôi, các kích thước của ma trận G^T là $[16 \times 48]$, $[8 \times 48]$, $[16 \times 16]$, $[8 \times 16]$, và ma trận $[8 \times 48]$ và ma trận $[8 \times 16]$ là các ma trận một phần được thu thập bằng việc lấy mẫu 8 vector cơ bản biến đổi từ trên cùng của ma trận $[16 \times 48]$ và ma trận $[16 \times 16]$, một cách tương ứng.

Do đó, trong trường hợp LFNST xuôi, vector $[48 \times 1]$ hoặc vector $[16 \times 1]$ là có thể dưới dạng đầu vào x , và vector $[16 \times 1]$ hoặc vector $[8 \times 1]$ là có thể dưới dạng đầu ra y . Trong việc lập mã và giải mã video, đầu ra của việc biến đổi sơ cấp xuôi là dữ liệu hai chiều (2D), để xây dựng vector $[48 \times 1]$ hoặc vector $[16 \times 1]$ dưới dạng đầu vào x , vector một chiều phải được xây dựng bằng việc sắp xếp thích hợp dữ liệu 2D mà là đầu ra của việc biến đổi xuôi.

Fig.8 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của việc biến đổi sơ cấp xuôi thành vector một chiều theo một ví dụ. Các biểu đồ bên trái của (a) và (b) của Fig.8 thể hiện trình tự cho việc xây dựng vector $[48 \times 1]$, và các biểu đồ bên phải của (a) và (b) của Fig.8 thể hiện trình tự cho việc xây dựng vector $[16 \times 1]$. Trong trường hợp LFNST, vector một chiều x có thể được thu nhận bằng việc sắp xếp tuần tự dữ liệu 2D trong cùng thứ tự như trong (a) và (b) của Fig.8.

Hướng sắp xếp của dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được xác định theo chế độ dự đoán nội của khối hiện tại. Ví dụ, khi chế độ dự đoán nội của khối hiện tại là trong việc dự đoán ngang đối với hướng chéo, dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được sắp xếp trong thứ tự của (a) của Fig.8, và khi chế độ dự đoán nội của khối hiện tại là trong việc dự đoán dọc đối với hướng chéo, dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được sắp xếp trong thứ tự của (b) của Fig.8.

Theo một ví dụ, thứ tự sắp xếp khác với các thứ tự sắp xếp của (a) và (b) Fig.8 có thể được áp dụng, và theo thứ tự để dẫn xuất cùng kết quả (vector y) như khi các thứ tự sắp xếp của (a) và (b) Fig.8 được áp dụng, các vector cột của ma trận G có thể được sắp xếp lại theo thứ tự sắp xếp. Nghĩa là, nó có thể sắp xếp lại các vector cột của G sao cho mỗi thành phần cấu thành vector x luôn được nhân với cùng vector cơ bản biến đổi.

Do đầu ra y được dẫn xuất qua Phương trình 9 là vector một chiều, khi dữ liệu hai chiều được yêu cầu dưới dạng dữ liệu đầu vào trong quy trình sử dụng kết quả của việc biến đổi thứ cấp xuôi dưới dạng đầu vào, ví dụ, trong quy trình thực hiện việc lượng tử hóa hoặc lập mã phần dư, vector đầu ra y của Phương trình 9 phải được sắp xếp hợp lý dưới dạng dữ liệu 2D một lần nữa.

Fig.9 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của việc biến đổi thứ cấp xuôi thành khối hai chiều theo một ví dụ.

Trong trường hợp LFNST, các giá trị đầu ra có thể được sắp xếp trong khối 2D theo thứ tự quét được xác định trước. (a) của Fig.9 thể hiện rằng khi đầu ra y là vector $[16 \times 1]$, các giá trị đầu ra được sắp xếp ở 16 vị trí của khối 2D theo thứ tự quét chéo. (b) của Fig.9 thể hiện rằng khi đầu ra y là vector $[8 \times 1]$, các giá trị đầu ra được sắp xếp ở 8 vị trí của khối 2D theo thứ tự quét chéo, và 8 vị trí còn lại được điền với các số không. X trong (b) của Fig.9 chỉ ra rằng nó được điền với số không.

Theo một ví dụ khác, do thứ tự trong đó vector đầu ra y được xử lý trong việc thực hiện việc lượng tử hóa hoặc lập mã phần dư có thể được thiết đặt trước, vector đầu ra y có thể không được sắp xếp trong khối 2D như được thể hiện trên Fig.9. Tuy nhiên, trong trường hợp việc lập mã phần dư, việc lập mã dữ liệu có thể được thực hiện trong các đơn vị khối 2D (ví dụ, 4×4) chẳng hạn như nhóm hệ số (Coefficient Group, CG), và trong trường hợp này, dữ liệu được sắp xếp theo thứ tự cụ thể như trong thứ tự quét chéo của Fig.9.

Trong khi đó, thiết bị giải mã có thể tạo cấu hình vector đầu vào một chiều y bằng việc sắp xếp dữ liệu hai chiều được xuất thông qua quy trình khử lượng tử hóa hoặc tương tự theo thứ tự quét được thiết đặt trước cho việc biến đổi ngược. Vector đầu vào y có thể được xuất ra dưới dạng vector đầu ra x bằng phương trình sau.

[Phương trình 10]

$$x = Gy$$

Trong trường hợp LFNST ngược, vector đầu ra x có thể được dẫn xuất bằng việc nhân vector đầu vào y , mà là vector $[16 \times 1]$ hoặc vector $[8 \times 1]$, bởi ma trận G . Đối với LFNST ngược, vector đầu ra x có thể là hoặc vector $[48 \times 1]$ hoặc vector $[16 \times 1]$.

Vector đầu ra x được sắp xếp trong khối hai chiều theo thứ tự được thể hiện trên Fig.8 và được sắp xếp dưới dạng dữ liệu hai chiều, và dữ liệu hai chiều này trở thành dữ liệu đầu vào (hoặc một phần của dữ liệu đầu vào) của việc biến đổi sơ cấp ngược.

Theo đó, việc biến đổi thứ cấp ngược là ngược với quy trình biến đổi thứ cấp xuôi một cách tổng thể, và trong trường hợp việc biến đổi ngược, không giống trong hướng xuôi, việc biến đổi thứ cấp ngược được áp dụng đầu tiên, và sau đó việc biến đổi sơ cấp ngược được áp dụng.

Trong LFNST ngược, một trong số 8 ma trận $[48 \times 16]$ và 8 ma trận $[16 \times 16]$ có thể được lựa chọn làm ma trận biến đổi G . Việc lựa chọn áp dụng ma trận $[48 \times 16]$ hay ma trận $[16 \times 16]$ phụ thuộc vào kích thước và hình dạng của khối.

Ngoài ra, 8 ma trận có thể được dẫn xuất từ bốn tập biến đổi như được thể hiện trong Bảng 2 ở trên, và mỗi tập biến đổi có thể cấu thành từ hai ma trận. Tập biến đổi nào để sử dụng trong số 4 tập biến đổi được xác định theo chế độ dự đoán nội, và cụ thể hơn là, tập biến đổi được xác định dựa trên giá trị của chế độ dự đoán nội được mở rộng bằng việc xem xét việc dự đoán nội góc rộng (Wide Angle Intra Prediction, WAIP). Ma trận nào để lựa chọn từ trong số hai ma trận cấu thành tập biến đổi được lựa chọn được dẫn xuất qua việc báo hiệu chỉ số. Cụ thể hơn là, 0, 1, và 2 là có thể dưới dạng giá trị

chỉ số được truyền, 0 có thể chỉ ra rằng LFNST không được áp dụng, và 1 và 2 có thể chỉ ra bất kỳ một trong số hai ma trận biến đổi cấu thành tập biến đổi được lựa chọn dựa trên giá trị chế độ dự đoán nội.

Fig.10 là sơ đồ minh họa các chế độ dự đoán nội góc rộng theo một phương án của sáng chế.

Giá trị chế độ dự đoán nội chung có thể có các giá trị từ 0 đến 66 và 81 đến 83, và giá trị chế độ dự đoán nội được mở rộng do WAIP có thể có giá trị từ -14 đến 83 như được thể hiện. Các giá trị từ 81 đến 83 chỉ ra chế độ mô hình tuyến tính thành phần chéo (CCLM, Cross Component Linear Model), và các giá trị từ -14 đến -1 và các giá trị từ 67 đến 80 chỉ ra chế độ dự đoán nội được mở rộng do ứng dụng WAIP.

Khi độ rộng của khối hiện tại dự đoán là lớn hơn độ cao, các điểm ảnh tham chiếu bên trên là thường gần hơn với các vị trí bên trong khối cần được dự đoán. Do đó, có thể chính xác hơn khi dự đoán theo chiều đáy-bên trái so với theo chiều đỉnh-bên phải. Ngược lại, khi độ cao của khối là lớn hơn độ rộng, các điểm ảnh tham chiếu bên trái là thường gần với các vị trí bên trong khối cần được dự đoán. Do đó, có thể chính xác hơn khi dự đoán theo chiều đỉnh-bên phải so với theo chiều đáy-bên trái. Do đó, có thể có ưu điểm khi áp dụng việc ánh xạ lại, nghĩa là, việc cải biến chỉ số chế độ, cho chỉ số của chế độ dự đoán nội góc rộng.

Khi việc dự đoán nội góc rộng được áp dụng, thông tin về việc dự đoán nội hiện tồn tại có thể được báo hiệu, và sau khi thông tin được phân tích cú pháp, thông tin có thể được ánh xạ lại cho chỉ số của chế độ dự đoán nội góc rộng. Do đó, tổng số lượng của các chế độ dự đoán nội cho khối cụ thể (ví dụ, khối không vuông có kích thước cụ thể) có thể không thay đổi, và nghĩa là, tổng số lượng của các chế độ dự đoán nội là 67, và việc lập mã chế độ dự đoán nội cho khối cụ thể có thể không được thay đổi.

Trong khi đó, như được mô tả ở trên, ma trận biến đổi nào trong số ma trận $[48 \times 16]$ và ma trận $[16 \times 16]$ được áp dụng cho LFNST được xác định bởi kích thước và hình dạng của khối mục tiêu biến đổi.

Fig.11 là sơ đồ minh họa hình dạng khối mà cho nó LFNST được áp dụng. (a) của Fig.11 thể hiện các khối 4×4 , (b) thể hiện các khối 4×8 và 8×4 , (c) thể hiện các khối $4 \times N$ hoặc $N \times 4$ trong đó N là 16 hoặc lớn hơn, (d) thể hiện các khối 8×8 , (e) thể hiện các khối $M \times N$ trong đó $M \geq 8$, $N \geq 8$, và $N > 8$ hoặc $M > 8$.

Trên Fig.11, các khối với các biên dày chỉ ra các vùng mà cho nó LFNST được áp dụng. Đối với các khối của Fig.13 (a) và Fig.13 (b), LFNST được áp dụng cho vùng 4×4 trên cùng bên trái, và đối với khối của Fig.11 (c), LFNST được áp dụng riêng lẻ hai vùng 4×4 trên cùng bên trái được sắp xếp liên tục. Trong (a), (b), và (c) của Fig.11, do LFNST được áp dụng trong các đơn vị của các vùng 4×4 , LFNST này sau đây sẽ được gọi là “LFNST 4×4 ”. Dựa trên kích thước ma trận cho G, ma trận $[16 \times 16]$ hoặc $[16 \times 8]$ có thể được áp dụng.

Cụ thể hơn là, ma trận $[16 \times 8]$ được áp dụng cho khối 4×4 (4×4 TU hoặc 4×4 CU) của Fig.11 (a) và ma trận $[16 \times 16]$ được áp dụng cho các khối trong (b) và (c) của Fig.11. Việc này là để điều chỉnh độ phức tạp tính toán cho trường hợp xấu nhất tới 8 phép nhân mỗi mẫu.

Đối với (d) và (e) của Fig.11, LFNST được áp dụng cho vùng 8×8 trên cùng bên trái, và LFNST này sau đây được gọi là “LFNST 8×8 ”. Đối với ma trận biến đổi tương ứng, ma trận $[48 \times 16]$ hoặc ma trận $[48 \times 8]$ có thể được áp dụng. Trong trường hợp LFNST xuôi, do vector $[48 \times 1]$ (x vector trong Phương trình 9) được nhập dưới dạng dữ liệu đầu vào, tất cả các giá trị mẫu của vùng 8×8 trên cùng bên trái không được sử dụng dưới dạng các giá trị đầu vào của LFNST xuôi. Nghĩa là, như có thể được thấy ở bên trái của Fig.8 (a) hoặc bên trái của Fig.8 (b), vector $[48 \times 1]$ có thể được xây dựng dựa trên các mẫu thuộc về 3 khối 4×4 còn lại trong khi để nguyên khối 4×4 dưới cùng bên phải.

Ma trận $[48 \times 8]$ có thể được áp dụng cho khối 8×8 (8×8 TU hoặc 8×8 CU) trên Fig.11 (d), và ma trận $[48 \times 16]$ có thể được áp dụng cho khối 8×8 trên Fig.11(e). Việc này cũng để điều chỉnh độ phức tạp tính toán cho trường hợp xấu nhất tới 8 phép nhân mỗi mẫu.

Phụ thuộc vào hình dạng khối, khi LFNST xuôi tương ứng (LFNST 4×4 hoặc LFNST 8×8) được áp dụng, 8 hoặc 16 dữ liệu đầu ra (vector y trong Phương trình 9, vector $[8 \times 1]$ hoặc $[16 \times 1]$) được tạo ra. Trong LFNST xuôi, số lượng của dữ liệu đầu ra là bằng hoặc nhỏ hơn số lượng của dữ liệu đầu vào do các tính chất của ma trận G^T .

Fig.12 là sơ đồ minh họa việc sắp xếp của dữ liệu đầu ra của LFNST xuôi theo một ví dụ, và thể hiện khối trong đó dữ liệu đầu ra của LFNST xuôi được sắp xếp theo hình dạng khối.

Diện tích mờ ở trên cùng bên trái của khối được thể hiện trên Fig.12 tương ứng với khu vực tại đó dữ liệu đầu ra của LFNST xuôi được đặt, các vị trí được đánh dấu với 0 chỉ ra các mẫu được điền với các giá trị 0, và diện tích còn lại biểu diễn các vùng mà không được thay đổi bởi LFNST xuôi. Trong khu vực không được thay đổi bởi LFNST, dữ liệu đầu ra của việc biến đổi sơ cấp xuôi duy trì không được thay đổi.

Như được mô tả ở trên, do kích thước của ma trận biến đổi được áp dụng thay đổi theo hình dạng của khối, số lượng của dữ liệu đầu ra cũng thay đổi. Theo Fig.12, dữ liệu đầu ra của LFNST xuôi có thể không điền hoàn toàn khối 4×4 trên cùng bên trái. Trong trường hợp (a) và (d) của Fig.12, ma trận $[16 \times 8]$ và ma trận $[48 \times 8]$ được áp dụng cho khối được chỉ ra bởi đường đậm hoặc một phần vùng bên trong khối, một cách tương ứng, và vector $[8 \times 1]$ dưới dạng đầu ra của LFNST xuôi được tạo ra. Nghĩa là, theo thứ tự quét được thể hiện trong (b) của Fig.9, chỉ 8 dữ liệu đầu ra có thể được điền như được thể hiện trong (a) và (d) của Fig.12, và 0 có thể được điền vào 8 vị trí còn lại. Trong trường hợp khối được áp dụng LFNST của Fig.11 (d), như được thể hiện trên Fig.12(d), hai khối 4×4 ở trên cùng bên phải và dưới cùng bên trái lân cận với khối 4×4 trên cùng bên trái cũng được điền với các giá trị 0.

Như được mô tả ở trên, về cơ bản, bằng việc báo hiệu chỉ số LFNST, việc liệu có áp dụng LFNST hay không và ma trận biến đổi cần được áp dụng được cụ thể hóa. Như được thể hiện trên Fig.12, khi LFNST được áp dụng, do số lượng của dữ liệu đầu ra của LFNST xuôi có thể bằng với hoặc nhỏ hơn số lượng của dữ liệu đầu vào, vùng được điền với giá trị không xảy ra như sau đây.

1) Như được thể hiện trên (a) của Fig.12, các mẫu từ vị trí thứ 8 và sau đó trong thứ tự quét trong khối 4x4 trên cùng bên trái, nghĩa là, các mẫu từ thứ 9 đến thứ 16.

2) Như được thể hiện trên (d) và (e) của Fig.12, khi ma trận [48 x 16] hoặc ma trận [48 x 8] được áp dụng, hai khối 4x4 liền kề với khối 4x4 trên cùng bên trái hoặc các khối 4x4 thứ hai và thứ ba trong thứ tự quét.

Do đó, nếu dữ liệu khác không tồn tại bằng việc kiểm tra các diện tích 1) và 2), rõ ràng rằng LFNST không được áp dụng, sao cho việc báo hiệu của chỉ số LFNST tương ứng có thể được bỏ qua.

Theo một ví dụ, ví dụ, trong trường hợp LFNST được áp dụng trong tiêu chuẩn VVC, do việc báo hiệu của chỉ số LFNST được thực hiện sau khi việc lập mã phần dư, thiết bị mã hóa có thể biết liệu có dữ liệu khác không (các hệ số đáng kể) cho tất cả các vị trí trong khối TU hoặc CU hay không thông qua việc lập mã phần dư. Theo đó, thiết bị mã hóa có thể xác định xem có thực hiện việc báo hiệu trên chỉ số LFNST hay không dựa trên sự tồn tại của dữ liệu khác không, và thiết bị giải mã có thể xác định xem chỉ số LFNST có được phân tích cú pháp hay không. Khi dữ liệu khác không không tồn tại trong khu vực được chỉ định trong 1) và 2) ở trên, việc báo hiệu của chỉ số LFNST được thực hiện.

Trong khi đó, đối với LFNST được áp dụng, các phương pháp đơn giản hóa sau đây có thể được áp dụng.

(i) Theo một ví dụ, số lượng của dữ liệu đầu ra cho LFNST xuôi có thể được giới hạn ở tối đa là 16.

Trong trường hợp (c) của Fig.11, LFNST 4×4 có thể được áp dụng cho hai vùng 4×4 liền kề với trên cùng bên trái, một cách tương ứng, và trong trường hợp này, tối đa là 32 dữ liệu đầu ra LFNST có thể được tạo ra. Khi số lượng của dữ liệu đầu ra cho LFNST xuôi được giới hạn ở tối đa là 16, trong trường hợp các khối $4 \times N/N \times 4$ ($N \geq 16$) (TU hoặc CU), LFNST 4×4 chỉ được áp dụng cho một vùng 4×4 ở trên cùng bên trái, LFNST có thể được áp dụng chỉ một lần cho tất cả các khối của Fig.11. Thông qua việc này, việc thực hiện của việc lập mã hình ảnh có thể được đơn giản hóa.

(ii) Theo một ví dụ, việc đặt bằng không có thể được áp dụng bổ sung cho vùng mà với nó LFNST không được áp dụng. Trong bản mô tả này, việc đặt bằng không có thể nghĩa là điền các giá trị của tất cả các vị trí thuộc về vùng cụ thể với giá trị của 0. Nghĩa là, việc đặt bằng không có thể được áp dụng cho vùng mà không được thay đổi do LFNST và duy trì kết quả của việc biến đổi sơ cấp xuôi. Như được mô tả ở trên, do LFNST được chia thành LFNST 4×4 và LFNST 8×8 , việc đặt bằng không có thể được chia thành hai loại ((ii)-(A) và (ii)-(B)) như sau đây.

(ii)-(A) Khi LFNST 4×4 được áp dụng, vùng mà với nó LFNST 4×4 không được áp dụng có thể được đặt bằng không. Fig.13 là sơ đồ minh họa việc đặt bằng không trong khối mà cho nó LFNST 4×4 được áp dụng theo một ví dụ.

Như được thể hiện trên Fig.13, đối với khối mà cho nó LFNST 4×4 được áp dụng, nghĩa là, cho tất cả các khối trong (a), (b) và (c) của Fig.12, toàn bộ vùng mà cho nó LFNST không được áp dụng có thể được điền với các số không.

Mặt khác, (d) của Fig.13 thể hiện rằng khi giá trị tối đa của số lượng của dữ liệu đầu ra của LFNST xuôi được giới hạn ở 16 theo một ví dụ, việc đặt bằng không được thực hiện trên các khối còn lại mà cho nó LFNST 4×4 không được áp dụng.

(ii)-(B) Khi LFNST 8×8 được áp dụng, vùng mà với nó LFNST 8×8 không được áp dụng có thể được đặt bằng không. Fig.14 là sơ đồ minh họa việc đặt bằng không trong khối mà cho nó LFNST 8×8 được áp dụng theo một ví dụ.

Như được thể hiện trên Fig.14, đối với khối mà cho nó LFNST 8x8 được áp dụng, nghĩa là, cho tất cả các khối trong (d) và (e) của Fig.12, toàn bộ vùng mà cho nó LFNST không được áp dụng có thể được điền với các số không.

(iii) Do việc đặt bằng không được biểu diễn trong (ii) ở trên, khu vực được điền với các số không có thể là không giống nhau khi LFNST được áp dụng. Theo đó, có thể kiểm tra xem dữ liệu khác không có tồn tại hay không theo việc đặt bằng không được đề xuất trong (ii) trong diện tích rộng hơn so với trường hợp LFNST của Fig.12.

Ví dụ, khi (ii)-(B) được áp dụng, sau khi kiểm tra xem dữ liệu khác không có tồn tại hay không trong đó khu vực được điền với các giá trị không trong (d) và (e) của Fig.12 bên cạnh khu vực được điền với 0 một cách bổ sung trên Fig.14, việc báo hiệu cho chỉ số LFNST có thể được thực hiện chỉ khi dữ liệu khác không không tồn tại.

Đương nhiên, ngay cả nếu việc đặt bằng không được đề xuất trong (ii) được áp dụng, có thể kiểm tra xem dữ liệu khác không có tồn tại theo cùng cách như việc báo hiệu chỉ số LFNST đang tồn tại hay không. Nghĩa là, sau khi kiểm tra xem dữ liệu khác không có tồn tại trong khối được điền với các số không trên Fig.12 hay không, việc báo hiệu chỉ số LFNST có thể được áp dụng. Trong trường hợp này, thiết bị mã hóa chỉ thực hiện việc đặt bằng không và thiết bị giải mã không giả định việc đặt bằng không, nghĩa là, chỉ kiểm tra xem dữ liệu khác không có tồn tại chỉ trong khu vực được đánh dấu tường minh là 0 trên Fig.12 hay không, có thể thực hiện việc phân tích cú pháp chỉ số LFNST.

Các phương án khác nhau trong đó các kết hợp của các phương pháp đơn giản hóa ((i), (ii)-(A), (ii)-(B), (iii)) cho LFNST được áp dụng có thể được dẫn xuất. Đương nhiên, các kết hợp của các phương pháp đơn giản hóa nêu trên không bị giới hạn ở phương án sau đây, và kết hợp bất kỳ có thể được áp dụng cho LFNST.

Phương án

- Giới hạn số lượng của dữ liệu đầu ra cho LFNST xuôi tới tối đa là 16 → (i)

- Khi LFNST 4x4 được áp dụng, tất cả các khu vực mà cho nó LFNST 4x4 không được áp dụng được đặt bằng không $\rightarrow$ (ii)-(A)

- Khi LFNST 8x8 được áp dụng, tất cả các khu vực mà cho nó LFNST 8x8 không được áp dụng được đặt bằng không $\rightarrow$ (ii)-(B)

- Sau khi kiểm tra xem dữ liệu khác không có cũng tồn tại khu vực đang tồn tại được điền với giá trị không và khu vực được điền với các số không do các việc đặt bằng không bổ sung ((ii)-(A), (ii)-(B)) hay không, chỉ số LFNST được báo hiệu chỉ khi dữ liệu khác không không tồn tại $\rightarrow$ (iii)

Trong trường hợp phương án này, khi LFNST được áp dụng, khu vực trong đó dữ liệu đầu ra khác không có thể tồn tại được giới hạn ở bên trong của diện tích 4x4 trên cùng bên trái. Chi tiết hơn, trong trường hợp của Fig.13 (a) và Fig.14 (a), vị trí thứ 8 trong thứ tự quét là vị trí cuối cùng trong đó dữ liệu khác không có thể tồn tại. Trong trường hợp của Fig.13 (b) và (c) và Fig.14 (b), vị trí thứ 16 trong thứ tự quét (nghĩa là, vị trí của mép dưới cùng bên phải của khối 4x4 trên cùng bên trái) là vị trí cuối cùng trong đó dữ liệu khác 0 có thể tồn tại.

Do đó, khi LFNST được áp dụng, sau khi kiểm tra xem dữ liệu khác không có tồn tại trong vị trí trong đó quy trình lập mã phân dư không được cho phép (ở vị trí bên ngoài vị trí cuối cùng) hay không, nó có thể được xác định xem chỉ số LFNST có được báo hiệu hay không.

Trong trường hợp phương pháp đặt bằng không được đề xuất trong (ii), do số lượng của dữ liệu cuối cùng được tạo ra khi cả việc biến đổi sơ cấp và LFNST được áp dụng, lượng tính toán được yêu cầu để thực hiện toàn bộ quy trình biến đổi có thể được giảm. Nghĩa là, khi LFNST được áp dụng, do việc đặt bằng không được áp dụng cho việc biến đổi sơ cấp xuôi dữ liệu đầu ra tồn tại trong vùng mà với nó LFNST không được áp dụng, không cần phải tạo ra dữ liệu cho vùng mà trở thành đặt bằng không trong khi thực hiện việc biến đổi sơ cấp xuôi. Theo đó, nó có thể làm giảm lượng tính toán

cần thiết để tạo ra dữ liệu tương ứng. Các hiệu quả bổ sung của phương pháp đặt bằng không được đề xuất trong (ii) được tóm tắt như sau đây.

Đầu tiên, như được mô tả ở trên, lượng tính toán được yêu cầu để thực hiện toàn bộ quy trình biến đổi được giảm.

Cụ thể là, khi (ii)-(B) được áp dụng, lượng tính toán cho trường hợp xấu nhất được giảm, sao cho quy trình biến đổi có thể được nhẹ bớt. Nói cách khác, nói chung, lượng tính toán lớn được yêu cầu để thực hiện việc biến đổi sơ cấp kích thước lớn. Bằng việc áp dụng (ii)-(B), số lượng của dữ liệu được dẫn xuất dưới dạng kết quả thực hiện LFNST xuôi có thể được giảm tới 16 hoặc nhỏ hơn. Ngoài ra, do kích thước của toàn bộ khối (TU hoặc CU) tăng, hiệu quả giảm lượng thao tác biến đổi được tăng thêm nữa.

Thứ hai, lượng tính toán được yêu cầu cho toàn bộ quy trình biến đổi có thể được giảm, nhờ đó làm giảm lượng tiêu thụ năng lượng được yêu cầu để thực hiện việc biến đổi.

Thứ ba, độ trễ có mặt trong quy trình biến đổi được giảm.

Việc biến đổi thứ cấp chẳng hạn như LFNST cộng lượng tính toán vào việc biến đổi sơ cấp đang tồn tại, do đó làm tăng thời gian trễ tổng thể có mặt trong khi thực hiện việc biến đổi. Cụ thể là, trong trường hợp việc dự đoán nội, do dữ liệu được xây dựng lại của các khối lân cận được sử dụng trong quy trình dự đoán, trong khi mã hóa, sự tăng về độ trễ do việc biến đổi thứ cấp dẫn đến sự tăng về độ trễ cho đến khi xây dựng lại. Điều này có thể dẫn đến việc tăng về độ trễ tổng thể của việc mã hóa dự đoán nội.

Tuy nhiên, nếu việc đặt bằng không được đề xuất trong (ii) được áp dụng, thời gian trễ để thực hiện việc biến đổi sơ cấp có thể được giảm đáng kể khi LFNST được áp dụng, thời gian trễ cho toàn bộ việc biến đổi được duy trì hoặc được giảm, sao cho thiết bị mã hóa có thể được thực hiện đơn giản hơn.

Trong khi đó, trong việc dự đoán nội thông thường, khối mục tiêu lập mã được xem là một đơn vị lập mã, và việc lập mã được thực hiện mà không có việc phân vùng

của nó. Tuy nhiên, việc lập mã các phân vùng con nội (Intra Sub-Partitions, ISP) dùng để chỉ việc thực hiện việc lập mã dự đoán nội với khối mục tiêu lập mã được phân vùng trong chiều ngang hoặc chiều dọc. Trong trường hợp này, khối được xây dựng lại có thể được tạo ra bằng việc thực hiện mã hóa/giải mã trong các đơn vị của các khối được phân vùng, và khối được xây dựng lại có thể được sử dụng làm khối tham chiếu của khối được phân vùng tiếp theo. Theo một ví dụ, trong việc lập mã ISP, một khối lập mã có thể được phân vùng thành hai hoặc bốn khối con và được lập mã, và trong ISP, việc dự đoán nội được thực hiện trên một khối con bằng việc tham chiếu tới giá trị điểm ảnh được xây dựng lại của khối con được đặt liền kề với phía trên cùng hoặc bên trái của nó. Sau đây, thuật ngữ “lập mã” có thể được sử dụng làm khái niệm gồm cả việc lập mã được thực hiện bởi thiết bị mã hóa và việc giải mã được thực hiện bởi thiết bị giải mã.

ISP phân vùng khối được dự đoán dưới dạng độ chói nội thành hai hoặc bốn phân vùng con theo chiều dọc hoặc chiều ngang theo kích thước của khối. Ví dụ, kích thước khối tối thiểu mà cho nó ISP có thể được áp dụng là 4×8 hoặc 8×4 . Nếu kích thước khối là lớn hơn 4×8 hoặc 8×4 , khối được phân vùng thành bốn phân vùng con.

Khi ISP được áp dụng, các khối con được lập mã lần lượt theo loại phân vùng, chẳng hạn như, ngang hoặc dọc, từ trái sang phải, hoặc từ trên cùng đến dưới cùng, và việc lập mã cho khối con tiếp theo có thể được thực hiện sau khi thực hiện cho tới quy trình khôi phục lại thông qua việc biến đổi ngược và việc dự đoán nội cho một khối con. Đối với khối con ngoài cùng bên trái hoặc trên cùng, như trong phương pháp dự đoán nội thông thường, điểm ảnh được xây dựng lại của khối lập mã mà đã được lập mã được tham chiếu đến. Ngoài ra, nếu khối con trước đó không liền kề với mỗi cạnh của khối con bên trong sau nó, theo thứ tự để dẫn xuất các điểm ảnh tham chiếu liền kề với cạnh tương ứng, như trong phương pháp dự đoán nội thông thường, điểm ảnh được xây dựng lại của khối lập mã liền kề đã được lập mã được tham chiếu đến.

Trong chế độ lập mã ISP, tất cả các khối con có thể được lập mã với cùng chế độ dự đoán nội, và còn chỉ ra xem có sử dụng việc lập mã ISP hay không và còn chỉ ra

trong hướng nào (ngang hoặc dọc) việc phân vùng cần được thực hiện có thể được báo hiệu. Tại lúc này, số lượng của các khối con có thể được điều chỉnh thành 2 hoặc 4 phụ thuộc vào hình dạng khối, và khi kích thước (chiều rộng x chiều cao) của một khối con là nhỏ hơn 16, việc phân vùng có thể không được cho phép cho các khối con tương ứng, cũng như việc ứng dụng của chính việc lập mã ISP có thể bị giới hạn.

Trong khi đó, trong trường hợp chế độ dự đoán ISP, một đơn vị lập mã được phân vùng thành hai hoặc bốn khối phân vùng, nghĩa là, các khối con, và được dự đoán, và cùng chế độ dự đoán nội được áp dụng cho hai hoặc bốn khối phân vùng do đó được phân vùng.

Như được mô tả ở trên, cả chiều ngang (nếu đơn vị lập mã $M \times N$ có chiều dài ngang và chiều dài dọc của M và N , một cách tương ứng, được chia trong việc dự đoán ngang, nó được chia thành các khối $M \times (N/2)$ khi được chia thành hai, và thành các khối $M \times (N/4)$ khi được chia thành bốn) và chiều dọc (nếu đơn vị lập mã $M \times N$ được chia trong việc dự đoán dọc, nó được chia thành các khối $(M/2) \times N$ khi được chia thành hai, và được chia thành các khối $(M/4) \times N$ khi được chia thành bốn) là có thể như hướng phân vùng. Khi được phân vùng trong việc dự đoán ngang, các khối phân vùng được lập mã theo thứ tự từ đỉnh xuống dưới, và khi được phân vùng trong việc dự đoán dọc, các khối phân vùng được lập mã theo thứ tự từ trái sang phải. Khối phân vùng được lập mã hiện tại có thể được dự đoán bằng việc tham chiếu tới điểm ảnh được xây dựng lại các giá trị của khối phân vùng trên cùng (bên trái) trong trường hợp việc phân vùng hướng ngang (dọc).

Việc biến đổi có thể được áp dụng cho tín hiệu phần dư được tạo ra bằng phương pháp dự đoán ISP trong các đơn vị của các khối phân vùng. Công nghệ lựa chọn đa biến đổi (Multiple Transform Selection, MTS) dựa trên sự kết hợp DST-7/DCT-8 cũng như DCT-2 hiện tồn tại có thể được áp dụng cho việc biến đổi sơ cấp (biến đổi lõi hoặc biến đổi sơ cấp) dựa trên hướng xuôi, và việc biến đổi không thể phân tách tần số thấp (Low

Frequency Non-Separable Transform, LFNST) có thể được áp dụng cho hệ số biến đổi được tạo ra theo việc biến đổi sơ cấp để tạo ra hệ số biến đổi được cải biến cuối cùng.

Nghĩa là, LFNST cũng có thể được áp dụng cho các khối phân vùng được phân chia bằng việc áp dụng chế độ dự đoán ISP, và cùng chế độ dự đoán nội được áp dụng cho các khối phân vùng được phân chia như được mô tả ở trên. Theo đó, khi lựa chọn tập LFNST được dẫn xuất dựa trên chế độ dự đoán nội, tập LFNST được dẫn xuất có thể được áp dụng cho tất cả các khối phân vùng. Nghĩa là, cùng chế độ dự đoán nội được áp dụng cho tất cả các khối phân vùng, và nhờ đó cùng tập LFNST có thể được áp dụng cho tất cả các khối phân vùng.

Trong khi đó, theo một ví dụ, LFNST có thể được áp dụng chỉ cho các khối biến đổi có cả chiều dài ngang và dọc là 4 hoặc nhiều hơn. Do đó, khi chiều dài ngang hoặc dọc của khối phân vùng được phân vùng theo phương pháp dự đoán ISP là nhỏ hơn 4, LFNST không được áp dụng và chỉ số LFNST không được báo hiệu. Ngoài ra, khi LFNST được áp dụng cho mỗi khối phân vùng, khối phân vùng tương ứng có thể được coi như là một khối biến đổi. Đương nhiên, khi phương pháp dự đoán ISP không được áp dụng, LFNST có thể được áp dụng cho khối lập mã.

Việc ứng dụng của LFNST cho mỗi khối phân vùng được mô tả chi tiết như sau đây.

Theo một ví dụ, sau khi áp dụng LFNST xuôi tới khối phân vùng riêng lẻ, và sau khi chỉ để tới 16 hệ số (8 hoặc 16) trong vùng 4x4 trên cùng bên trái theo thứ tự quét hệ số biến đổi, việc đặt bằng không của việc điền tất cả các vị trí và các vùng còn lại với giá trị của 0 có thể được áp dụng.

Mặt khác, theo một ví dụ, khi chiều dài của một cạnh của khối phân vùng là 4, LFNST được áp dụng chỉ cho vùng 4x4 trên cùng bên trái, và khi chiều dài của tất cả các cạnh của khối phân vùng, nghĩa là, độ rộng và độ cao, là 8 hoặc nhiều hơn, LFNST

có thể được áp dụng cho 48 hệ số còn lại ngoại trừ vùng 4x4 dưới cùng bên phải trong vùng 8x8 trên cùng bên trái.

Mặt khác, theo một ví dụ, để điều chỉnh độ phức tạp tính toán của trường hợp xấu nhất tới 8 phép nhân mỗi mẫu, khi mỗi khối phân vùng là 4x4 hoặc 8x8, chỉ 8 hệ số biến đổi có thể được xuất ra sau khi áp dụng LFNST xuôi. Nghĩa là, nếu khối phân vùng là 4x4, ma trận 8x16 có thể được áp dụng làm ma trận biến đổi, và nếu khối phân vùng là 8x8, ma trận 8x48 có thể được áp dụng làm ma trận biến đổi.

Trong khi đó, trong tiêu chuẩn VVC hiện tại, việc báo hiệu chỉ số LFNST được thực hiện trong các đơn vị của các đơn vị lập mã. Theo đó, khi chế độ dự đoán ISP được sử dụng và LFNST được áp dụng cho tất cả các khối phân vùng, thì cùng giá trị chỉ số LFNST có thể được áp dụng cho các khối phân vùng tương ứng. Nghĩa là, khi giá trị chỉ số LFNST được truyền một lần ở mức đơn vị lập mã, chỉ số LFNST tương ứng có thể được áp dụng cho tất cả các khối phân vùng trong đơn vị lập mã. Như được mô tả ở trên, giá trị chỉ số LFNST có thể có các giá trị của 0, 1, và 2, 0 chỉ ra trường hợp trong đó LFNST không được áp dụng, và 1 và 2 chỉ ra hai ma trận biến đổi có mặt trong một tập LFNST khi LFNST được áp dụng.

Như được mô tả ở trên, tập LFNST được xác định bởi chế độ dự đoán nội, và do tất cả các khối phân vùng trong đơn vị lập mã được dự đoán trong cùng chế độ dự đoán nội trong trường hợp chế độ dự đoán ISP, các khối phân vùng có thể dùng để chỉ cùng tập LFNST.

Theo một ví dụ khác, việc báo hiệu chỉ số LFNST vẫn được thực hiện trong các đơn vị của các đơn vị lập mã, nhưng trong trường hợp chế độ dự đoán ISP, mà không xác định xem có áp dụng LFNST một cách đồng nhất cho tất cả các khối phân vùng hay không, việc liệu có áp dụng giá trị chỉ số LFNST được báo hiệu ở mức đơn vị lập mã cho mỗi khối phân vùng hay không áp dụng LFNST có thể được xác định qua điều kiện phân tách. Ở đây, điều kiện phân tách có thể được báo hiệu dưới dạng cờ cho mỗi khối phân vùng qua luồng bit, và khi giá trị cờ là 1, giá trị chỉ số LFNST được báo hiệu ở

mức đơn vị lập mã có thể được áp dụng, và khi giá trị cờ là 0, LFNST có thể không được áp dụng.

Sau đây, phương pháp duy trì độ phức tạp tính toán cho trường hợp xấu nhất khi LFNST được áp dụng cho chế độ ISP sẽ được mô tả.

Trong trường hợp chế độ ISP, để duy trì số lượng của các phép nhân mỗi mẫu (hoặc mỗi hệ số, hoặc mỗi vị trí) ở giá trị nhất định hoặc nhỏ hơn khi LFNST được áp dụng, việc ứng dụng của LFNST có thể được giới hạn. Phụ thuộc vào kích thước của khối phân vùng, số lượng của các phép nhân mỗi mẫu (hoặc mỗi hệ số, hoặc mỗi vị trí) có thể được duy trì ở 8 hoặc nhỏ hơn bằng việc áp dụng LFNST như sau đây.

1. Khi cả chiều dài ngang và chiều dài dọc của khối phân vùng là bằng hoặc lớn hơn 4, cùng phương pháp như phương pháp điều chỉnh độ phức tạp tính toán cho trường hợp xấu nhất cho LFNST trong tiêu chuẩn VVC hiện tại có thể được áp dụng.

Nghĩa là, khi khối phân vùng là khối 4x4, thay vì ma trận 16x16, theo hướng xuôi, ma trận 8x16 được thu thập bằng việc lấy mẫu 8 hàng trên cùng từ ma trận 16x16 có thể được áp dụng, và theo hướng ngược, ma trận 16x8 được thu thập bằng việc lấy mẫu 8 cột bên trái từ ma trận 16x16 có thể được áp dụng. Ngoài ra, khi khối phân vùng là khối 8x8, theo hướng xuôi, thay vì ma trận 16x48, ma trận 8x48 được thu thập bằng việc lấy mẫu 8 hàng trên cùng từ ma trận 16x48 có thể được áp dụng, và theo hướng ngược, thay vì ma trận 48x16, ma trận 48x8 được thu thập bằng việc lấy mẫu 8 cột bên trái từ ma trận 48x16 có thể được áp dụng.

Trong trường hợp khối 4xN hoặc Nx4 ($N > 4$), khi việc biến đổi xuôi được thực hiện, 16 hệ số được tạo ra sau khi áp dụng ma trận 16x16 chỉ cho khối 4x4 trên cùng bên trái được sắp xếp trong vùng 4x4 trên cùng bên trái, và các vùng khác có thể được điền với các giá trị 0. Ngoài ra, thực hiện việc biến đổi ngược, 16 hệ số được đặt trong khối 4x4 trên cùng bên trái có thể được sắp xếp trong thứ tự quét để tạo cấu hình vector đầu vào, và sau đó 16 dữ liệu đầu ra có thể được tạo ra bằng việc nhân ma trận 16x16.

Dữ liệu đầu ra được tạo ra có thể được sắp xếp trong vùng 4x4 trên cùng bên trái, và các vùng còn lại ngoại trừ vùng 4x4 trên cùng bên trái có thể được điền với các số không.

Trong trường hợp khối 8xN hoặc Nx8 ($N > 8$), khi việc biến đổi xuôi được thực hiện, 16 hệ số được tạo ra sau khi áp dụng ma trận 16x48 chỉ cho vùng ROI trong khối 8x8 trên cùng bên trái (các vùng còn lại ngoại trừ khối 4x4 dưới cùng bên phải từ khối 8x8 trên cùng bên trái) có thể được sắp xếp trong diện tích 4x4 trên cùng bên trái, và các vùng khác có thể được điền với các giá trị 0. Ngoài ra, thực hiện việc biến đổi ngược, 16 hệ số được đặt trong khối 4x4 trên cùng bên trái có thể được sắp xếp trong thứ tự quét để tạo cấu hình vector đầu vào, và sau đó 48 dữ liệu đầu ra có thể được tạo ra bằng việc nhân ma trận 48x16. Dữ liệu đầu ra được tạo ra có thể được điền trong vùng ROI, và các vùng khác có thể được điền với các giá trị 0.

Theo một ví dụ khác, để duy trì số lượng của các phép nhân mỗi mẫu (hoặc mỗi hệ số, hoặc mỗi vị trí) ở giá trị nhất định hoặc nhỏ hơn, số lượng của các phép nhân mỗi mẫu (hoặc mỗi hệ số, hoặc mỗi vị trí) dựa trên kích thước đơn vị lập mã ISP thay vì kích thước của khối phân vùng ISP có thể được duy trì ở 8 hoặc nhỏ hơn. Nếu chỉ có một khối trong số các khối phân vùng ISP, mà thỏa mãn điều kiện trong đó LFNST được áp dụng, việc tính toán độ phức tạp cho trường hợp xấu nhất của LFNST có thể được áp dụng dựa trên kích thước đơn vị lập mã tương ứng thay vì kích thước của khối phân vùng. Ví dụ, khi khối lập mã độ chói cho đơn vị lập mã nhất định được phân vùng thành 4 khối phân vùng có kích thước 4x4 và được lập mã bởi ISP, và khi không có hệ số biến đổi khác không nào tồn tại cho hai khối phân vùng trong số chúng, hai khối phân vùng khác có thể được thiết đặt một cách tương ứng để tạo ra 16 hệ số biến đổi thay vì 8 (dựa trên bộ mã hóa).

Sau đây, phương pháp báo hiệu chỉ số LFNST trong trường hợp chế độ ISP sẽ được mô tả.

Như được mô tả ở trên, chỉ số LFNST có thể có các giá trị của 0, 1, và 2, trong đó 0 chỉ ra rằng LFNST không được áp dụng, và 1 và 2 một cách tương ứng chỉ ra hoặc

một trong số hai ma trận nhân LFNST được gồm trong tập LFNST được lựa chọn. LFNST được áp dụng dựa trên ma trận nhân LFNST được lựa chọn bởi chỉ số LFNST. Phương pháp truyền chỉ số LFNST trong tiêu chuẩn VVC hiện tại sẽ được mô tả như sau đây.

1. Chỉ số LFNST có thể được truyền một lần cho mỗi đơn vị lập mã (CU), và trong trường hợp cây đôi, các chỉ số LFNST riêng lẻ có thể được báo hiệu cho khối độ chói và khối sắc độ, một cách tương ứng.

2. Khi chỉ số LFNST không được báo hiệu, giá trị chỉ số LFNST được suy ra là giá trị mặc định là 0. Trường hợp trong đó giá trị chỉ số LFNST được suy ra là 0 là như sau đây.

A. Trong trường hợp chế độ trong đó không có việc biến đổi nào được áp dụng (ví dụ, bỏ qua biến đổi, BDPCM, lập mã không tổn hao, v.v.)

B. Khi việc biến đổi sơ cấp không phải là DCT-2 (DST7 hoặc DCT8), nghĩa là, khi việc biến đổi trong việc dự đoán ngang hoặc việc biến đổi trong việc dự đoán dọc không phải là DCT-2

C. Khi chiều dài ngang hoặc chiều dài dọc cho khối độ chói của đơn vị lập mã vượt quá kích thước của biến đổi độ chói tối đa có thể biến đổi, ví dụ, khi kích thước của biến đổi độ chói tối đa có thể biến đổi là 64, và khi kích thước cho khối độ chói của khối lập mã là bằng 128x16, LFNST không thể được áp dụng.

Trong cây đôi, việc liệu mỗi trong số đơn vị lập mã cho thành phần độ chói và đơn vị lập mã cho thành phần sắc độ có vượt quá kích thước biến đổi độ chói tối đa hay không được xác định. Nghĩa là, việc liệu khối độ chói có vượt quá kích thước biến đổi độ chói tối đa cho việc biến đổi hay không được kiểm tra, và việc liệu độ rộng hoặc độ dài của khối độ chói tương ứng theo định dạng màu sắc đối với khối sắc độ có vượt quá kích thước biến đổi độ chói tối đa cho việc biến đổi tối đa hay không được kiểm tra. Ví dụ, khi định dạng màu sắc là 4:2:0, độ rộng/độ dài của khối độ chói tương ứng là bằng

hai lần độ rộng/độ dài của khối sắc độ, và kích thước biến đổi của khối độ chói tương ứng là bằng hai lần độ rộng/độ dài của khối sắc độ. Trong một ví dụ khác, khi định dạng màu sắc là 4:4:4, độ rộng/độ dài và kích thước biến đổi của khối độ chói tương ứng là giống như của khối sắc độ.

Việc biến đổi độ dài 64 hoặc việc biến đổi độ dài 32 có thể nghĩa là việc biến đổi được áp dụng cho độ rộng hoặc độ cao có chiều dài là 64 hoặc 32, một cách tương ứng, và “kích thước biến đổi” có thể nghĩa là 64 hoặc 32 như chiều dài tương ứng.

Trong trường hợp cây đơn, sau khi kiểm tra xem chiều dài ngang hoặc chiều dài dọc của khối độ chói có vượt quá kích thước khối biến đổi độ chói có thể biến đổi tối đa hay không, nếu nó vượt quá, việc báo hiệu chỉ số LFNST có thể được bỏ qua.

D. Chỉ số LFNST có thể được truyền chỉ khi cả chiều dài ngang và chiều dài dọc của đơn vị lập mã là bằng hoặc lớn hơn 4.

Trong trường hợp cây đôi, chỉ số LFNST có thể được báo hiệu chỉ khi cả các chiều dài ngang và dọc cho thành phần tương ứng (nghĩa là, thành phần độ chói hoặc sắc độ) là bằng hoặc lớn hơn 4.

Trong trường hợp cây đơn, chỉ số LFNST có thể được báo hiệu khi cả các chiều dài ngang và dọc cho thành phần độ chói là bằng hoặc lớn hơn 4.

E. Nếu vị trí của hệ số khác không cuối cùng không phải là vị trí DC (vị trí trên cùng bên trái của khối), và nếu vị trí của hệ số khác không cuối cùng không phải là vị trí DC, trong trường hợp khối độ chói của loại cây đôi, chỉ số LFNST được truyền. Trong trường hợp khối sắc độ loại cây đôi, nếu bất kỳ một trong số vị trí của hệ số khác không cuối cùng cho Cb và vị trí của hệ số khác không cuối cùng cho Cr không phải là vị trí DC, chỉ số LFNST tương ứng được truyền.

Trong trường hợp loại cây đơn, nếu vị trí của hệ số khác không cuối cùng của bất kỳ một trong số thành phần độ chói, thành phần Cb, và thành phần Cr không phải là vị trí DC, chỉ số LFNST được truyền.

Ở đây, nếu giá trị cờ khối được lập mã (CBF) chỉ ra xem hệ số biến đổi cho một khối biến đổi tồn tại có là 0 hay không, vị trí của hệ số khác không cuối cùng cho khối biến đổi tương ứng không được kiểm tra để xác định xem chỉ số LFNST có được báo hiệu hay không. Nghĩa là, khi giá trị CBF tương ứng là 0, do không có việc biến đổi nào được áp dụng cho khối tương ứng, vị trí của hệ số khác không cuối cùng có thể không được xem xét khi kiểm tra điều kiện cho việc báo hiệu chỉ số LFNST.

Ví dụ, 1) trong trường hợp loại cây đôi và thành phần độ chói, nếu giá trị CBF tương ứng là 0, chỉ số LFNST không được báo hiệu, 2) trong trường hợp loại cây đôi và thành phần sắc độ, nếu giá trị CBF cho C_b là 0 và giá trị CBF cho C_r là 1, chỉ vị trí của hệ số khác không cuối cùng cho C_r được kiểm tra và chỉ số LFNST tương ứng được truyền, 3) trong trường hợp loại cây đơn, vị trí của hệ số khác không cuối cùng được kiểm tra chỉ cho các thành phần có giá trị CBF là 1 cho mỗi trong số độ chói, C_b , và C_r .

F. Khi được xác nhận rằng hệ số biến đổi tồn tại ở vị trí ngoài vị trí trong đó LFNST hệ số biến đổi có thể tồn tại, việc báo hiệu chỉ số LFNST có thể được bỏ qua. Trong trường hợp khối biến đổi 4×4 và khối biến đổi 8×8 , các hệ số biến đổi LFNST có thể tồn tại ở tám vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC, và các vị trí còn lại được điền với các số không. Ngoài ra, khi khối biến đổi 4×4 và khối biến đổi 8×8 là không phải, các hệ số biến đổi LFNST có thể tồn tại trong mười sáu vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC, và các vị trí còn lại được điền với các số không.

Theo đó, nếu các hệ số biến đổi khác không tồn tại trong vùng mà nên được điền với giá trị không sau tiến trình lập mã phân dư, việc báo hiệu chỉ số LFNST có thể được bỏ qua.

Trong khi đó, chế độ ISP có thể cũng được áp dụng chỉ cho khối độ chói, hoặc có thể được áp dụng cho cả khối độ chói và khối sắc độ. Như được mô tả ở trên, khi việc dự đoán ISP được áp dụng, đơn vị lập mã tương ứng có thể được chia thành hai hoặc bốn khối phân vùng và được dự đoán, và việc biến đổi có thể được áp dụng cho mỗi

trong số các khối phân vùng. Do đó, cũng khi xác định điều kiện báo hiệu chỉ số LFNST trong các đơn vị của các đơn vị lập mã, cần phải xem xét đến thực tế rằng LFNST có thể được áp dụng cho các khối phân vùng tương ứng. Ngoài ra, khi chế độ dự đoán ISP được áp dụng chỉ cho thành phần cụ thể (ví dụ, khối độ chói), chỉ số LFNST phải được báo hiệu xem xét đến thực tế rằng chỉ thành phần được chia thành các khối phân vùng. Các phương pháp báo hiệu chỉ số LFNST sẵn có trong chế độ ISP được tóm tắt như sau đây.

1. Chỉ số LFNST có thể được truyền một lần cho mỗi đơn vị lập mã (CU), và trong trường hợp cây đôi, các chỉ số LFNST riêng lẻ có thể được báo hiệu cho khối độ chói và khối sắc độ, một cách tương ứng.

2. Khi chỉ số LFNST không được báo hiệu, giá trị chỉ số LFNST được suy ra là giá trị mặc định là 0. Trường hợp trong đó giá trị chỉ số LFNST được suy ra là 0 là như sau đây.

A. Trong trường hợp chế độ trong đó không có việc biến đổi nào được áp dụng (ví dụ, bỏ qua biến đổi, BDPCM, lập mã không tổn hao, v.v.)

B. Khi chiều dài ngang hoặc chiều dài dọc cho khối độ chói của đơn vị lập mã vượt quá kích thước của biến đổi độ chói tối đa có thể biến đổi, ví dụ, khi kích thước của biến đổi độ chói tối đa có thể biến đổi là 64, và khi kích thước cho khối độ chói của khối lập mã là bằng 128x16, LFNST không thể được áp dụng.

Việc liệu có báo hiệu chỉ số LFNST hay không có thể được xác định dựa trên kích thước của khối phân vùng thay vì đơn vị lập mã. Nghĩa là, nếu chiều dài ngang hoặc dọc của khối phân vùng cho khối độ chói tương ứng vượt quá kích thước của việc biến đổi độ chói tối đa có thể biến đổi, việc báo hiệu chỉ số LFNST có thể được bỏ qua và giá trị chỉ số LFNST có thể được suy ra là 0.

Trong cây đôi, việc liệu mỗi trong số đơn vị lập mã hoặc khối phân vùng cho thành phần độ chói và đơn vị lập mã hoặc khối phân vùng cho thành phần sắc độ có

vượt quá kích thước khối biến đổi tối đa hay không được xác định. Nghĩa là, khi mỗi trong số độ rộng và độ dài của đơn vị lập mã hoặc khối phân vùng cho độ chói được so sánh với kích thước biến đổi độ chói tối đa và thậm chí một trong số chúng là lớn hơn kích thước biến đổi độ chói tối đa, LFNST không được áp dụng. Trong đơn vị lập mã hoặc khối phân vùng cho sắc độ, độ rộng/độ dài của khối độ chói tương ứng theo định dạng màu sắc được so sánh với kích thước biến đổi độ chói tối đa cho việc biến đổi tối đa. Ví dụ, khi định dạng màu sắc là 4:2:0, độ rộng/độ dài của khối độ chói tương ứng là bằng hai lần độ rộng/độ dài của khối sắc độ, và kích thước biến đổi tối đa của khối độ chói tương ứng là bằng hai lần kích thước biến đổi tối đa của khối sắc độ. Trong một ví dụ khác, khi định dạng màu sắc là 4:4:4, độ rộng/độ dài và kích thước biến đổi của khối độ chói tương ứng là giống như của khối sắc độ.

Trong trường hợp cây đơn, sau khi kiểm tra xem chiều dài ngang hoặc chiều dài dọc cho khối độ chói (đơn vị lập mã hoặc khối phân vùng) có vượt quá kích thước khối biến đổi độ chói có thể biến đổi tối đa hay không, nếu nó vượt quá, việc báo hiệu chỉ số LFNST có thể được bỏ qua.

C. Nếu LFNST được gồm trong tiêu chuẩn VVC hiện tại được áp dụng, chỉ số LFNST có thể được truyền chỉ khi cả chiều dài ngang và chiều dài dọc của khối phân vùng là bằng hoặc lớn hơn 4.

Nếu LFNST cho khối $2 \times M$ ($1 \times M$) hoặc $M \times 2$ ($M \times 1$) được áp dụng bên cạnh LFNST được gồm trong tiêu chuẩn VVC hiện tại, chỉ số LFNST có thể được truyền chỉ khi kích thước của khối phân vùng là bằng hoặc lớn hơn khối $2 \times M$ ($1 \times M$) hoặc $M \times 2$ ($M \times 1$). Ở đây, việc diễn đạt “khối $P \times Q$ là bằng hoặc lớn hơn khối $R \times S$ ” có nghĩa là $P \geq R$ và $Q \geq S$.

Tóm lại, chỉ số LFNST có thể được truyền chỉ khi khối phân vùng là bằng hoặc lớn hơn kích thước tối thiểu mà cho nó LFNST là có thể ứng dụng. Trong trường hợp cây đôi, chỉ số LFNST có thể được báo hiệu chỉ khi khối phân vùng cho thành phần độ chói hoặc sắc độ là bằng hoặc lớn hơn kích thước tối thiểu mà cho nó LFNST là có thể

ứng dụng. Trong trường hợp cây đơn, chỉ số LFNST có thể được báo hiệu chỉ khi khối phân vùng cho thành phần độ chói là bằng hoặc lớn hơn kích thước tối thiểu mà cho nó LFNST là có thể ứng dụng.

Trong bản mô tả này, việc diễn đạt “khối $M \times N$ là lớn hơn hoặc bằng với khối $K \times L$ ” có nghĩa là M là lớn hơn hoặc bằng với K và N là lớn hơn hoặc bằng với L . Việc diễn đạt “khối $M \times N$ là lớn hơn khối $K \times L$ ” có nghĩa là M là lớn hơn hoặc bằng với K và N là lớn hơn hoặc bằng với L , và M là lớn hơn K hoặc N là lớn hơn L . Việc diễn đạt “khối $M \times N$ nhỏ hơn hoặc bằng với khối $K \times L$ ” có nghĩa là M là nhỏ hơn hoặc bằng với K và N là nhỏ hơn hoặc bằng với L , trong khi việc diễn đạt “khối $M \times N$ là nhỏ hơn khối $K \times L$ ” có nghĩa là M là nhỏ hơn hoặc bằng với K và N là nhỏ hơn hoặc bằng với L , và M là nhỏ hơn K hoặc N là nhỏ hơn L .

D. Nếu vị trí của hệ số khác không cuối cùng không phải là vị trí DC (vị trí trên cùng bên trái của khối), và nếu vị trí của hệ số khác không cuối cùng không phải là vị trí DC trong bất kỳ một trong số tất cả các khối phân vùng. Trong trường hợp của khối độ chói loại cây đôi, chỉ số LFNST được truyền. Trong trường hợp loại cây đôi và khối sắc độ, nếu ít nhất một trong số vị trí của hệ số khác không cuối cùng của tất cả các khối phân vùng cho C_b (nếu chế độ ISP không được áp dụng cho thành phần sắc độ, số lượng của các khối phân vùng được coi là một) và vị trí của hệ số khác không cuối cùng của tất cả các khối phân vùng cho C_r (nếu chế độ ISP không được áp dụng cho thành phần sắc độ, số lượng của các khối phân vùng được coi là một) không phải là vị trí DC, chỉ số LFNST tương ứng có thể được truyền.

Trong trường hợp loại cây đơn, nếu vị trí của hệ số khác không cuối cùng của bất kỳ một trong số tất cả các khối phân vùng cho thành phần độ chói, thành phần C_b và thành phần C_r không phải là vị trí DC, chỉ số LFNST tương ứng có thể được truyền.

Ở đây, nếu giá trị của cờ khối được lập mã (CBF) chỉ ra xem hệ số biến đổi có tồn tại cho mỗi khối phân vùng hay không là 0, vị trí của hệ số khác không cuối cùng cho khối phân vùng tương ứng không được kiểm tra để kiểm tra xem chỉ số LFNST có

được báo hiệu hay không. Nghĩa là, khi giá trị CBF tương ứng là 0, do không có việc biến đổi nào được áp dụng cho khối tương ứng, vị trí của hệ số khác không cuối cùng cho khối phân vùng tương ứng không được xem xét khi kiểm tra điều kiện cho việc báo hiệu chỉ số LFNST.

Ví dụ, 1) trong trường hợp loại cây đôi và thành phần độ chói, nếu giá trị CBF tương ứng cho mỗi khối phân vùng là 0, khối phân vùng được loại trừ khi xác định xem có báo hiệu chỉ số LFNST hay không, 2) trong trường hợp loại cây đôi và thành phần sắc độ, nếu giá trị CBF cho C_b là 0 và giá trị CBF cho C_r là 1 cho mỗi khối phân vùng, chỉ vị trí của hệ số khác không cuối cùng cho C_r được kiểm tra để xác định xem có báo hiệu chỉ số LFNST hay không, 3) trong trường hợp loại cây đơn, có thể xác định xem có báo hiệu chỉ số LFNST hay không bằng việc kiểm tra vị trí của hệ số khác không cuối cùng chỉ cho các khối có giá trị CBF là 1 cho tất cả các khối phân vùng của thành phần độ chói, thành phần C_b , và thành phần C_r .

Trong trường hợp chế độ ISP, thông tin hình ảnh có thể cũng được tạo cấu hình sao cho vị trí của hệ số khác không cuối cùng không được kiểm tra, và phương án của nó là như sau đây.

i. Trong trường hợp chế độ ISP, việc báo hiệu chỉ số LFNST có thể được cho phép mà không kiểm tra vị trí của hệ số khác không cuối cùng cho cả khối độ chói và khối sắc độ. Nghĩa là, ngay cả nếu vị trí của hệ số khác không cuối cùng cho tất cả các khối phân vùng là vị trí DC hoặc giá trị CBF tương ứng là 0, việc báo hiệu chỉ số LFNST có thể được cho phép.

ii. Trong trường hợp chế độ ISP, việc kiểm tra của vị trí của hệ số khác không cuối cùng chỉ cho khối độ chói có thể được bỏ qua, và trong trường hợp của khối sắc độ, việc kiểm tra của vị trí của hệ số khác không cuối cùng có thể được thực hiện trong cách được mô tả ở trên. Ví dụ, trong trường hợp loại cây đôi và khối độ chói, việc báo hiệu chỉ số LFNST được cho phép mà không kiểm tra vị trí của hệ số khác không cuối cùng, và trong trường hợp loại cây đôi và khối sắc độ, xem chỉ số LFNST tương ứng có được

báo hiệu hay không có thể được xác định bằng việc kiểm tra xem vị trí DC có tồn tại cho vị trí của hệ số khác không cuối cùng hay không trong cách được mô tả ở trên.

iii. Trong trường hợp chế độ ISP và loại cây đơn, phương pháp i hoặc ii có thể được áp dụng. Nghĩa là, trong trường hợp chế độ ISP và khi số i được áp dụng cho loại cây đơn, có thể bỏ qua việc kiểm tra của vị trí của hệ số khác không cuối cùng cho cả khối độ chói và khối sắc độ và cho phép việc báo hiệu chỉ số LFNST. Mặt khác, bằng việc áp dụng phần ii, cho các khối phân vùng cho thành phần độ chói, việc kiểm tra của vị trí của hệ số khác không cuối cùng được bỏ qua, và cho các khối phân vùng cho thành phần sắc độ (nếu ISP không được áp dụng cho thành phần sắc độ, số lượng của các khối phân vùng có thể được coi là 1), vị trí của hệ số khác không cuối cùng được kiểm tra trong cách được mô tả ở trên, nhờ đó xác định xem có báo hiệu chỉ số LFNST hay không.

E. Khi được xác nhận rằng hệ số biến đổi tồn tại ở vị trí ngoài vị trí trong đó LFNST hệ số biến đổi có thể tồn tại ngay cả cho một khối phân vùng trong số tất cả các khối phân vùng, việc báo hiệu chỉ số LFNST có thể được bỏ qua.

Ví dụ, trong trường hợp khối phân vùng 4×4 và khối phân vùng 8×8 , các hệ số biến đổi LFNST có thể tồn tại ở tám vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC, và các vị trí còn lại được điền với các số không. Ngoài ra, nếu nó là bằng hoặc lớn hơn 4×4 và không phải là khối phân vùng 4×4 cũng như khối phân vùng 8×8 , các hệ số biến đổi LFNST có thể tồn tại ở 16 vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC, và tất cả các vị trí còn lại được điền với các số không.

Theo đó, nếu các hệ số biến đổi khác không tồn tại trong vùng mà nên được điền với giá trị không sau tiến trình lập mã phần dư, việc báo hiệu chỉ số LFNST có thể được bỏ qua.

Trong khi đó, trong trường hợp chế độ ISP, điều kiện chiều dài được quan sát độc lập cho việc dự đoán ngang và việc dự đoán dọc, và DST-7 được áp dụng thay vì

DCT-2 mà không báo hiệu cho chỉ số MTS. Được xác định xem chiều dài ngang hoặc dọc có lớn hơn hoặc bằng với 4 và nhỏ hơn hoặc bằng với 16 hay không, và nhân biến đổi sơ cấp được xác định theo kết quả xác định. Theo đó, trong trường hợp chế độ ISP, khi LFNST có thể được áp dụng, cấu hình kết hợp biến đổi sau đây là có thể.

1. Khi chỉ số LFNST là 0 (gồm trường hợp trong đó chỉ số LFNST được suy ra là 0), điều kiện quyết định việc biến đổi sơ cấp tại thời điểm của ISP được gồm trong tiêu chuẩn VVC hiện tại có thể được tuân theo. Nói cách khác, có thể được kiểm tra xem điều kiện chiều dài (là bằng với hoặc lớn hơn 4 hoặc bằng với hoặc nhỏ hơn 16) có được thỏa mãn một cách độc lập cho các hướng ngang và dọc, một cách tương ứng hay không, và nếu nó được thỏa mãn, DST-7 có thể được áp dụng thay vì DCT-2 cho việc biến đổi sơ cấp, trong khi, nếu nó không được thỏa mãn, DCT-2 có thể được áp dụng.

2. Đối với trường hợp trong đó chỉ số LFNST là lớn hơn 0, hai cấu hình sau đây có thể là có thể cho việc biến đổi sơ cấp.

A. DCT-2 có thể được áp dụng cho cả các hướng ngang và dọc.

B. Điều kiện quyết định biến đổi sơ cấp tại thời điểm của ISP được gồm trong tiêu chuẩn VVC hiện tại có thể được tuân theo. Nói cách khác, có thể được kiểm tra xem điều kiện chiều dài (là bằng với hoặc lớn hơn 4 hoặc bằng với hoặc nhỏ hơn 16) có được thỏa mãn một cách độc lập cho các hướng ngang và dọc, một cách tương ứng hay không, và nếu nó được thỏa mãn, DST-7 có thể được áp dụng thay vì DCT-2, trong khi, nếu nó không được thỏa mãn, DCT-2 có thể được áp dụng.

Trong chế độ ISP, thông tin hình ảnh có thể tạo cấu hình sao cho chỉ số LFNST được truyền cho mỗi khối phân vùng thay vì cho mỗi đơn vị lập mã. Trong trường hợp này, trong phương pháp báo hiệu chỉ số LFNST được mô tả ở trên, có thể được xem là chỉ một khối phân vùng tồn tại trong đơn vị trong đó chỉ số LFNST được truyền, và có thể được xác định xem có báo hiệu chỉ số LFNST hay không.

Theo một phương án, sơ đồ điều biến lập mã xung chên lệch khối hoặc điều

biến lập mã xung delta dựa trên khối (block-based delta pulse code modulation, BDPCM) có thể được sử dụng. BDPCM có thể được gọi là việc điều biến lập mã xung delta dựa trên khối phần dư (residual block-based delta pulse code modulation, RDPCM) được lượng tử hóa.

Khi khối được dự đoán bằng việc áp dụng BDPCM, các mẫu được xây dựng lại được sử dụng để dự đoán hàng hoặc cột của khối từng đường. Ở đây, các điểm ảnh tham chiếu được sử dụng có thể là các mẫu không được lọc. Hướng BDPCM có thể chỉ ra xem việc dự đoán hướng thẳng hay hướng ngang được sử dụng. Lỗi dự đoán được lượng tử hóa trong miền không gian, và điểm ảnh được xây dựng lại bằng việc bổ sung lỗi dự đoán được khử lượng tử hóa cho việc dự đoán. Như một thay thế cho BDPCM này, BDPCM miền phần dư được lượng tử hóa có thể được đề xuất, trong đó hướng dự đoán hoặc việc báo hiệu có thể là giống như trong BDPCM được áp dụng cho miền không gian. Nghĩa là, các hệ số lượng tử hóa chính chúng có thể được tích lũy thông qua BDPCM miền phần dư được lượng tử hóa như trong điều biến lập mã xung delta (DPCM), sau đó phần dư có thể được xây dựng lại thông qua việc khử lượng tử hóa. Theo đó, BDPCM miền phần dư được lượng tử hóa có thể được sử dụng để áp dụng DPCM trong giai đoạn lập mã phần dư. Miền phần dư được lượng tử hóa được sử dụng trong sáng chế chỉ ra rằng phần dư được dẫn xuất dựa trên việc dự đoán được lượng tử hóa mà không có việc biến đổi, và dùng để chỉ miền cho mẫu phần dư được lượng tử hóa.

Theo một ví dụ, việc liệu có áp dụng chế độ điều biến lập mã xung delta dựa trên khối (BDPCM) có thể được báo hiệu trong bảng cú pháp cho đơn vị lập mã như được thể hiện trong Bảng 3.

[Bảng 3]

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
chType = treeType == DUAL_TREE_CHROMA ? 1 : 0	
.....	
if(CuPredMode[chType][x0][y0] == MODE_INTRA	
CuPredMode[chType][x0][y0] == MODE_PLT) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag)	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
else {	
if(sps_bdpcm_enabled_flag &&	
cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
.....	
}	
}	
if((treeType == SINGLE_TREE treeType ==	
DUAL_TREE_CHROMA) &&	
ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC,	
treeType)	
else if(!pred_mode_plt_flag) {	
if(!cu_act_enabled_flag) {	
if(cbWidth / SubWidthC <= MaxTsSize && cbHeight / SubHeightC	
<= MaxTsSize	
&& sps_bdpcm_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
.....	
}	
}	
}	
}	

<code>} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */</code>	
<code>.....</code>	
<code>}</code>	
<code>if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag &&</code>	
<code> general_merge_flag[x0][y0] == 0)</code>	
<code> cu_coded_flag</code>	<code>ae(v)</code>
<code> if(cu_coded_flag) {</code>	
<code> </code>	
<code> LfnstDcOnly = 1</code>	
<code> LfnstZeroOutSigCoeffFlag = 1</code>	
<code> MtsDcOnly = 1</code>	
<code> MtsZeroOutSigCoeffFlag = 1</code>	
<code> transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)</code>	
<code> lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType == ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)</code>	
<code> lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : ((IntraSubPartitionsSplitType == ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight)</code>	
<code> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0))</code>	
<code> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {</code>	
<code> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)</code>	
<code> lfnst_idx</code>	<code>ae(v)</code>
<code> }</code>	
<code> </code>	
<code> }</code>	
<code>}</code>	

Trong Bảng 3, `intra_bdpcm_luma_flag` là cờ chỉ ra xem BDPCM có được áp dụng cho thành phần độ chói của đơn vị lập mã hay không, và `intra_bdpcm_chroma_flag` là cờ chỉ ra xem BDPCM có được áp dụng cho thành phần sắc độ của đơn vị lập mã hay không. Giá trị cờ là 1 chỉ ra rằng BDPCM được áp dụng, và giá trị cờ là 0 chỉ ra rằng BDPCM không được áp dụng. Giá trị của biến `BdpcmFlag[ x ][ y ][ cIdx ]` được xác định theo các giá trị của `intra_bdpcm_luma_flag` và `intra_bdpcm_chroma_flag` như trong Bảng 4.

[Bảng 4]

.....
intra_bdpcm_luma_flag equal to 1 specifies that BDPCM is applied to the current luma coding block at the location (x_0, y_0) , i.e. the transform is skipped, the intra luma prediction mode is specified by **intra_bdpcm_luma_dir_flag**. **intra_bdpcm_luma_flag** equal to 0 specifies that BDPCM is not applied to the current luma coding block at the location (x_0, y_0) .
 When **intra_bdpcm_luma_flag** is not present it is inferred to be equal to 0.
 The variable **BdpcmFlag**[x][y][cIdx] is set equal to **intra_bdpcm_luma_flag** for $x = x_0..x_0 + \text{cbWidth} - 1$, $y = y_0..y_0 + \text{cbHeight} - 1$ and $\text{cIdx} = 0$.
intra_bdpcm_luma_dir_flag equal to 0 specifies that the BDPCM prediction direction is horizontal. **intra_bdpcm_luma_dir_flag** equal to 1 specifies that the BDPCM prediction direction is vertical.
 The variable **BdpcmDir**[x][y][cIdx] is set equal to **intra_bdpcm_luma_dir_flag** for $x = x_0..x_0 + \text{cbWidth} - 1$, $y = y_0..y_0 + \text{cbHeight} - 1$ and $\text{cIdx} = 0$.

intra_bdpcm_chroma_flag equal to 1 specifies that BDPCM is applied to the current chroma coding blocks at the location (x_0, y_0) , i.e. the transform is skipped, the intra chroma prediction mode is specified by **intra_bdpcm_chroma_dir_flag**. **intra_bdpcm_chroma_flag** equal to 0 specifies that BDPCM is not applied to the current chroma coding blocks at the location (x_0, y_0) .
 When **intra_bdpcm_chroma_flag** is not present it is inferred to be equal to 0.
 The variable **BdpcmFlag**[x][y][cIdx] is set equal to **intra_bdpcm_chroma_flag** for $x = x_0..x_0 + \text{cbWidth} - 1$, $y = y_0..y_0 + \text{cbHeight} - 1$ and $\text{cIdx} = 1..2$.
intra_bdpcm_chroma_dir_flag equal to 0 specifies that the BDPCM prediction direction is horizontal. **intra_bdpcm_chroma_dir_flag** equal to 1 specifies that the BDPCM prediction direction is vertical.
 The variable **BdpcmDir**[x][y][cIdx] is set equal to **intra_bdpcm_chroma_dir_flag** for $x = x_0..x_0 + \text{cbWidth} - 1$, $y = y_0..y_0 + \text{cbHeight} - 1$ and $\text{cIdx} = 1..2$.

Trong **BdpcmFlag**[x][y][cIdx] của Bảng 4, giá trị của x là giá trị tọa độ x (vị trí điểm ảnh ngang) dựa trên vị trí tọa độ ngang trên cùng bên trái (tọa độ x) của ảnh độ chói như tham chiếu (thiết đặt là 0), và các tọa độ x tăng từ trái sang phải. Trong **BdpcmFlag**[x][y][cIdx] của Bảng 4, giá trị của y là giá trị tọa độ y (vị trí điểm ảnh dọc) dựa trên vị trí tọa độ dọc trên cùng bên trái (tọa độ y) của ảnh độ chói như tham chiếu (thiết đặt là 0), và các tọa độ y tăng từ trên xuống dưới.

Trong **BdpcmFlag**[x][y][cIdx] của Bảng 4, giá trị của cIdx chỉ ra thành phần màu sắc (Y, Cb, hoặc Cr), 0 chỉ ra thành phần độ chói, và 1 và 2 chỉ ra thành phần sắc độ (1 cho Cb và 2 cho Cr). Trong Bảng 4, mặc dù giá trị của cIdx là thành phần sắc độ thay vì 0, tọa độ x và tọa độ y được biểu diễn dựa trên ảnh độ chói. Ví dụ, khi độ phân giải ngang của thành phần sắc độ là thấp hơn hai lần so với độ phân giải ngang của thành phần độ chói (ví dụ, khi định dạng màu sắc cho ảnh là 4:2:0 hoặc 4:2:2), các giá trị tọa độ x thực cho ảnh sắc độ là một nửa các tọa độ tương ứng dựa trên ảnh độ chói.

Trong Bảng 4, **cbWidth** và **cbHeight** một cách tương ứng chỉ ra độ rộng và độ

dài của khối lập mã dựa trên thành phần độ chói. Tương tự, khi độ phân giải ngang (đọc) của thành phần sắc độ là thấp hơn hai lần so với độ phân giải ngang (đọc) của thành phần độ chói, độ rộng thực (độ dài) của khối lập mã sắc độ tương ứng bằng một nửa độ rộng thực (độ dài) của khối lập mã độ chói.

Trong Bảng 4, $x0$ và $y0$ một cách tương ứng chỉ ra giá trị tọa độ x và giá trị tọa độ y của vị trí trên cùng bên trái trong khối lập mã độ chói khi vị trí trên cùng bên trái của ảnh độ chói được thiết đặt là 0. Ở đây, khối lập mã sắc độ tương ứng với vị trí độ chói được đặt cùng một chỗ khi được chuyển đổi thành thành phần độ chói.

Như được thể hiện trong Bảng 4, do tất cả các giá trị $BdpcmFlag[x][y][cIdx]$ cho hai thành phần sắc độ (Cb và Cr) được thiết đặt là `intra_bdpcm_chroma_flag`, khi giá trị của `intra_bdpcm_chroma_flag` là 1, BDPCM có thể được áp dụng cho cả hai thành phần sắc độ (Cb và Cr) của đơn vị lập mã được lập mã hiện tại.

Như được thể hiện trong Bảng 5, khi BDPCM được áp dụng cho mỗi thành phần, giá trị của cờ bỏ qua biến đổi (`transform_skip_flag`) cho thành phần được suy ra là 1.

[Bảng 5]

.....

transform_skip_flag[$x0$][$y0$][$cIdx$] specifies whether a transform is applied to the associated transform block or not. The array indices $x0$, $y0$ specify the location ($x0$, $y0$) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture. The array index $cIdx$ specifies an indicator for the colour component; it is equal to 0 for Y, 1 for Cb, and 2 for Cr. **transform_skip_flag**[$x0$][$y0$][$cIdx$] equal to 1 specifies that no transform is applied to the associated transform block. **transform_skip_flag**[$x0$][$y0$][$cIdx$] equal to 0 specifies that the decision whether transform is applied to the associated transform block or not depends on other syntax elements.

When **transform_skip_flag**[$x0$][$y0$][$cIdx$] is not present, it is inferred as follows:

- If **BdpcmFlag**[$x0$][$y0$][$cIdx$] is equal to 1, **transform_skip_flag**[$x0$][$y0$][$cIdx$] is inferred to be equal to 1.
- Otherwise (**BdpcmFlag**[$x0$][$y0$][$cIdx$] is equal to 0), **transform_skip_flag**[$x0$][$y0$][$cIdx$] is inferred to be equal to 0.

Trong Bảng 5, `transform_skip_flag[x0][y0][cIdx]` là cờ chỉ ra xem khối lập mã có được lập mã hay không với bỏ qua biến đổi. Giá trị cờ là 1 chỉ ra rằng việc bỏ qua

biến đổi được áp dụng, và giá trị còn là 0 chỉ ra rằng việc bỏ qua biến đổi không được áp dụng. Ở đây, các ý nghĩa của $x0$, $y0$, và $cIdx$ cũng được định rõ trong Bảng 5 và đã được mô tả tham chiếu tới Bảng 4.

Trong Bảng 5, khi giá trị của $BdpcmFlag[x0][y0][cIdx]$ là 1, $transform_skip_flag[x0][y0][cIdx]$ được suy ra là 1 (nghĩa là, khi BPPCM được áp dụng cho thành phần được định rõ bởi $cIdx$, bỏ qua biến đổi cho thành phần được thiết đặt là 1), và như được thể hiện trong Bảng 4, do tất cả các giá trị $BdpcmFlag[x0][y0][cIdx]$ cho hai thành phần sắc độ (Cb và Cr) được thiết đặt là $intra_bdpcm_chroma_flag$, các giá trị $transform_skip_flag[x0][y0][cIdx]$ cho hai thành phần sắc độ được thiết đặt ngang bằng là 1 cho trường hợp trong đó BDPCM được áp dụng cho các thành phần sắc độ.

Bảng 6 là bảng cú pháp cho đơn vị biến đổi, mà thể hiện việc báo hiệu hoặc việc phân tích cú pháp của $transform_skip_flag[x0][y0][cIdx]$ và giá trị còn khối được lập mã (coded block flag, CBF) của khối biến đổi cho mỗi thành phần màu sắc.

[Bảng 6]

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 && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1))	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT && ((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 (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]	
}	
.....	

<pre> 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) </pre>	
<pre>tu_joint_cbr_residual_flag[xC][yC]</pre>	ae(v)
<pre>if(tu_y_coded_flag[x0][y0] && treeType != DUAL_TREE_CHROMA) {</pre>	
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplitType == ISP_NO_SPLIT) && !cu_sbt_flag) </pre>	
<pre>transform_skip_flag[x0][y0][0]</pre>	ae(v)
<pre>.....</pre>	
<pre>}</pre>	
<pre>if(tu_cb_coded_flag[xC][yC] && treeType != DUAL_TREE_LUMA) {</pre>	
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag) </pre>	
<pre>transform_skip_flag[xC][yC][1]</pre>	ae(v)
<pre>.....</pre>	
<pre>}</pre>	
<pre> if(tu_cr_coded_flag[xC][yC] && treeType != DUAL_TREE_LUMA && !(tu_cb_coded_flag[xC][yC] && tu_joint_cbr_residual_flag[xC][yC])) { </pre>	
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag) </pre>	
<pre>transform_skip_flag[xC][yC][2]</pre>	ae(v)
<pre>.....</pre>	
<pre>}</pre>	
<pre>}</pre>	

Trong Bảng 6, khi báo hiệu `tu_y_coded_flag[x0][y0]` (hoặc có thể được biểu diễn là `tu_cbf_luma[x0][y0]`), mà là CBF cho thành phần độ chói, và `tu_cb_coded_flag[xC][yC]` (CBF cho thành phần Cb hoặc `tu_cbf_cb[ xC ][ yC ]`) và `tu_cr_coded_flag[ xC ][ yC ]` (CBF cho thành phần Cr hoặc `tu_cbf_cr[ xC ][ yC ]`), mà là các CBF cho thành phần sắc độ, việc liệu BDPCM có được áp dụng cho mỗi thành phần hay không không được kiểm tra. Nghĩa là, giá trị của `BdpcmFlag[ x ][ y ][ cIdx ]` không được kiểm tra.

Ở đây, cờ khởi được lập mã (CBF) chỉ ra xem hệ số biến đổi có tồn tại trong khối biến đổi hay không, trong đó giá trị của 1 chỉ ra rằng hệ số biến đổi tồn tại và giá trị của 0 chỉ ra rằng hệ số biến đổi không tồn tại. Do đó, mặc dù BDPCM được áp dụng cho mỗi thành phần, giá trị CBF tương ứng có thể là 0.

Theo phương pháp thao tác BDPCM, tín hiệu phần dư được xây dựng lại bằng việc tích lũy các giá trị được truyền dưới dạng các hệ số biến đổi cho mỗi vị trí theo hướng được lựa chọn (hướng ngang hoặc dọc trong VVC, được định rõ bởi

`intra_bdpcm_luma_dir_flag` cho thành phần độ chói trong Bảng 4, và được định rõ bởi `intra_bdpcm_chroma_dir_flag` cho thành phần sắc độ). Ở đây, giá trị CBF của 0 có thể được coi là giống về hoạt động như trong phương pháp dự đoán nội thông thường mà không có tín hiệu phần dư.

Tuy nhiên, khi giá trị CBF là 0, chế độ BDPCM của việc báo hiệu `intra_bdpcm_luma_flag`, `intra_bdpcm_luma_dir_flag`, `intra_bdpcm_chroma_flag`, và `intra_bdpcm_chroma_dir_flag` như được thể hiện trong Bảng 3 có thể là có ưu điểm hơn so với việc lập mã dự đoán nội xét về chi phí RD. Do đó, khi giá trị CBF là 0, việc lập mã có thể được thực hiện trong chế độ BDPCM, và do đó có thể tồn tại trường hợp trong đó giá trị CBF là 0 ngay cả trong chế độ BDPCM.

Chỉ số LFNST (`lfnst_idx`) có thể được báo hiệu trong đơn vị lập mã của Bảng 3, và như được thể hiện trong Bảng 3, chỉ số LFNST có thể được báo hiệu chỉ khi giá trị của biến `lfnstNotTsFlag` là 1. Trong Bảng 3, khi bất kỳ trong số các cờ bỏ qua biến đổi cho nhiều thành phần là 1, giá trị của `lfnstNotTsFlag` được thiết đặt là 1. Nghĩa là, biến `lfnstNotTsFlag` có thể được thiết đặt là 1 chỉ khi tất cả các giá trị cờ bỏ qua biến đổi cho tất cả các thành phần thuộc về đơn vị lập mã hiện tại là 0.

Như được thể hiện trong Bảng 3, trong cây đơn, giá trị của `lfnstNotTsFlag` có thể là 1 chỉ khi các cờ bỏ qua biến đổi cho cả thành phần độ chói và thành phần sắc độ là 0, trong cây tách biệt độ chói (`DUAL_TREE_LUMA`), giá trị của biến `lfnstNotTsFlag` có thể là 1 khi chỉ cờ bỏ qua biến đổi cho thành phần độ chói là 0, và trong cây tách biệt sắc độ (`DUAL_TREE_CHROMA`), giá trị của biến `lfnstNotTsFlag` có thể là 1 chỉ khi tất cả các cờ bỏ qua biến đổi cho thành phần Cb và thành phần Cr là 0.

Như được mô tả ở trên, khi chế độ BDPCM được áp dụng cho thành phần và giá trị CBF là 0, cờ bỏ qua biến đổi cho thành phần có thể được suy ra là 1. Tuy nhiên, do hệ số biến đổi không tồn tại cho thành phần, thành phần là không liên quan đến quy trình biến đổi. Thành phần là không liên quan đến quy trình biến đổi và do đó cần phải không liên quan đến việc báo hiệu của chỉ số LFNST. Trong trường hợp này, tuy nhiên,

trong Bảng 3, do cờ bỏ qua biến đổi cho thành phần được suy ra là 1 và do đó giá trị của `lfnstNotTsFlag` được thiết đặt là 0, chỉ số LFNST không được báo hiệu. Cho việc minh họa cụ thể, các trường hợp sau đây có thể được minh họa.

1. Trong cây tách biệt sắc độ, nếu cờ tách biệt chỉ ra xem BDPCM có được áp dụng hay không tồn tại cho mỗi trong số thành phần Cb và thành phần Cr một cách tương ứng tồn tại không giống Bảng 3 (ví dụ, nếu `intra_bdpcm_cb_flag` và `intra_bdpcm_cr_flag` tồn tại thay vì `intra_bdpcm_chroma_flag` trong Bảng 3), khi BDPCM được áp dụng cho thành phần Cb hoặc thành phần Cr và giá trị CBF cho khối biến đổi tương ứng là 0, cờ bỏ qua biến đổi cho thành phần Cb hoặc thành phần Cr cũng là 1, và do đó giá trị của `lfnstNotTsFlag` trong Bảng 3 là 0. Do đó, chỉ số LFNST không được báo hiệu.

Tuy nhiên, khi BDPCM được áp dụng cho hoặc thành phần Cr hoặc thành phần Cb, giá trị CBF cho thành phần còn lại là khác 0, và giá trị của cờ bỏ qua biến đổi cho thành phần còn lại là 0, LFNST có thể được áp dụng cho thành phần còn lại, và do đó được mong muốn cho phép việc báo hiệu chỉ số LFNST.

Tuy nhiên, như được thể hiện trong Bảng 3, nếu các cờ chỉ ra xem BDPCM được áp dụng cho thành phần Cb và thành phần Cr có giống như `intra_bdpcm_chroma_flag` hay không, khi `intra_bdpcm_chroma_flag` là 1, BDPCM được áp dụng cho thành phần Cb và thành phần Cr cùng một lúc. Do đó, trường hợp trong đó BDPCM được áp dụng cho hoặc thành phần Cr hoặc thành phần Cb, giá trị CBF cho thành phần còn lại là khác 0, và giá trị cờ bỏ qua biến đổi cho thành phần còn lại là 0 không xảy ra, và chỉ số LFNST không được báo hiệu.

2. Trong cây đơn, khi BDPCM được áp dụng cho thành phần bất kỳ trong số Y, Cb, và Cr và giá trị CBF cho khối biến đổi tương ứng là 0, giá trị cờ bỏ qua biến đổi cho thành phần cũng là 1, và do đó giá trị của `lfnstNotTsFlag` là 0.

Tuy nhiên, khi giá trị CBF cho bất kỳ trong số hai thành phần còn lại là khác 0

và giá trị cờ bỏ qua biến đổi cho thành phần là 0, LFNST có thể được áp dụng cho (các) thành phần, và do đó được mong muốn cho phép việc báo hiệu chỉ số LFNST.

Như được thể hiện trong Bảng 3, được coi là liệu BDPCM có được áp dụng cho thành phần độ chói (Y) hay không được xác định bởi `intra_luma_bdpcm_flag` (khi giá trị của `intra_luma_bdpcm_flag` là 1, BDPCM được áp dụng, và khi giá trị của `intra_luma_bdpcm_flag` là 0, BDPCM không được áp dụng) và liệu BDPCM có được áp dụng cho hai thành phần sắc độ (Cb và Cr) hay không được xác định bởi `intra_chroma_bdpcm_flag` (khi giá trị của `intra_chroma_bdpcm_flag` là 1, BDPCM được áp dụng, và khi giá trị của `intra_chroma_bdpcm_flag` là 0, BDPCM không được áp dụng).

Trong trường hợp này, khi việc lập mã được thực hiện trong cây đơn, (1) trong trường hợp trong đó BDPCM được áp dụng cho thành phần độ chói, giá trị CBF tương ứng là 0, giá trị CBF tương ứng cho bất kỳ trong số hai thành phần sắc độ còn lại là khác 0, và giá trị cờ bỏ qua biến đổi cho thành phần là 0, LFNST có thể được áp dụng cho (các) thành phần sắc độ. Tuy nhiên, khi BDPCM được áp dụng cho thành phần độ chói và giá trị CBF tương ứng là 0, chỉ số LFNST có thể được báo hiệu chỉ khi tất cả các giá trị cờ bỏ qua biến đổi cho hai thành phần sắc độ là 0 trong tiêu chuẩn VVC hiện tại, và cuối cùng, BDPCM có thể không được áp dụng cho hai thành phần sắc độ. Ngoài ra, trong tiêu chuẩn VVC hiện tại, do LFNST được áp dụng chỉ cho thành phần độ chói trong cây đơn, LFNST không được áp dụng cho các thành phần sắc độ khi việc lập mã được thực hiện trong cây đơn.

(2) Trong trường hợp khác của lập mã trong cây đơn, trong đó BDPCM được áp dụng cho hai thành phần sắc độ và các giá trị CBF cho cả các thành phần sắc độ là 0, khi giá trị CBF cho thành phần độ chói là khác 0 và giá trị cờ bỏ qua biến đổi do đó là 0, LFNST có thể được áp dụng cho thành phần độ chói. Nghĩa là, mặc dù việc bỏ qua biến đổi không được áp dụng cho thành phần độ chói, BDPCM được áp dụng cho hai thành phần sắc độ, và tất cả các giá trị CBF của các thành phần sắc độ là 0, chỉ số LFNST

cho thành phần độ chói có thể được báo hiệu.

Theo đó, khi biến `lfnstNotTsFlag` trong Bảng 3 được thiết đặt như được thể hiện trong Bảng 7 và Bảng 8, chỉ số LFNST có thể được báo hiệu trong hai trường hợp ((1) và (2)).

[Bảng 7]

```

lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
  ( !tu_y_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 ) )
  && ( treeType == DUAL_TREE_LUMA ||
  ( ( !tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 )
    && ( !tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 ) ) ) )

```

[Bảng 8]

```

lfnstNotTsFlag = ( !tu_y_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 )
  && ( !tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 )
  && ( !tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 )

```

Như được thể hiện trong Bảng 7 và Bảng 8, khi giá trị CBF cho thành phần là 0, giá trị cờ bỏ qua biến đổi cho thành phần là không liên quan mặc dù là giá trị bất kỳ và do đó không cấu thành vào việc thay đổi giá trị của `lfnstNotTsFlag` thành 0.

Nói cách khác, khi các giá trị CBF cho tất cả các thành phần thuộc về đơn vị lập mã là 0 hoặc các giá trị cờ bỏ qua biến đổi cho nó là 0, giá trị của `lfnstNotTsFlag` có thể được thiết đặt là 1.

Trong Bảng 7, các thành phần khác nhau mà cho nó các điều kiện được kiểm tra được tạo cấu hình phụ thuộc vào cây đơn, cây tách biệt độ chói, và cây tách biệt sắc độ. Nghĩa là, giá trị CBF hoặc giá trị cờ bỏ qua biến đổi được kiểm tra cho Y/Cb/Cr trong cây đơn, chỉ cho Y trong cây tách biệt độ chói, và cho Cb và Cr trong cây tách biệt sắc độ, và được kiểm tra cho tất cả các thành phần trong Bảng 8.

Trong Bảng 8, được coi là CBF hoặc cờ bỏ qua biến đổi được suy ra là 0 cho thành phần không được kết hợp với đơn vị lập mã hiện tại. Nghĩa là, trong cây tách biệt

độ chói, `tu_cb_coded_flag[x0][y0]` hoặc `transform_skip_flag[x0][y0][1]` cần phải được suy ra là 0, và `tu_cr_coded_flag[x0][y0]` hoặc `transform_skip_flag[x0][2]` cần phải được suy ra là 0. Trong cây tách biệt sắc độ, `tu_y_coded_flag[x0][y0]` hoặc `transform_skip_flag[x0][y0][0]` cần phải được suy ra là 0.

Trong tiêu chuẩn VVC hiện tại, LFNST có thể được áp dụng cho khối lập mã mà cho nó chế độ ISP được áp dụng. Khi chế độ ISP được áp dụng và do đó khối lập mã cho thành phần Y được phân tách thành N (N = 2 hoặc 4) khối phân vùng, mỗi khối phân vùng có giá trị CBF. Nghĩa là, `tu_y_coded_flag` được báo hiệu cho mỗi khối phân vùng.

Khi chế độ ISP được áp dụng trong tiêu chuẩn VVC hiện tại, được đảm bảo rằng ít nhất một trong số các khối phân vùng có CBF là 1, và một trong số các khối phân vùng có CBF là 1 không nhất thiết phải là khối phân vùng thứ nhất theo thứ tự lập mã.

Khi ISP được áp dụng, do `x0` và `y0` trong Bảng 7 và Bảng 8 là tọa độ x và tọa độ y của đơn vị lập mã, `tu_y_coded_flag[x0][y0]` trong Bảng 7 và Bảng 8 là CBF cho khối phân vùng thứ nhất.

Do đó, khi chế độ ISP được áp dụng, không thể kiểm tra xem hệ số biến đổi có tồn tại trong khối lập mã độ chói hay không chỉ với điều kiện `!"tu_y_coded_flag[x0][y0]"` trong Bảng 7 và Bảng 8.

Theo một ví dụ, khi chế độ ISP được áp dụng, Bảng 7 và Bảng 8 có thể được mô tả như Bảng 9 và Bảng 10, một cách tương ứng, để phản ánh tình huống trong đó không thể kiểm tra rõ ràng xem hệ số biến đổi có tồn tại trong khối lập mã độ chói hay không. Trong Bảng 9 và Bảng 10, điều kiện `"IntraSubPartitionsSplitType == ISP_NO_SPLIT"` chỉ ra trường hợp trong đó chế độ ISP không được áp dụng.

[Bảng 9]

```

lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
    ((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !tu_y_coded_flag[ x0 ][ y0 ] )
    || transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0)) &&
    (treeType == DUAL_TREE_LUMA ||
    ((!tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0) &&
    (!tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)))

```

[Bảng 10]

```

lfnstNotTsFlag = ( ( IntraSubPartitionsSplitType == ISP_NO_SPLIT
    && !tu_y_coded_flag[ x0 ][ y0 ] ) || transform_skip_flag[ x0 ][ y0 ][ 0 ] ==
    0) && (!tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ]
    == 0) && (!tu_cr_coded_flag[ x0 ][ y0 ] ||
    transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)

```

Tuy nhiên, do việc bỏ qua biến đổi không thể được áp dụng khi chế độ ISP được áp dụng trong tiêu chuẩn VVC hiện tại, chế độ ISP không được áp dụng trừ khi điều kiện "transform_skip_flag[x0][y0][0] == 0" được thỏa mãn.

Khi chỉ một trong số "!tu_y_coded_flag[x0][y0]" và "transform_skip_flag[x0][y0][0] == 0" trong điều kiện "!tu_y_coded_flag[x0][y0] || transform_skip_flag[x0][y0][0] == 0" của Bảng 7 được thỏa mãn logic, điều kiện là đúng, và do đó việc kiểm tra "!tu_y_coded_flag[x0][y0]" là có ý nghĩa chỉ khi "transform_skip_flag[x0][y0][0] == 0" không được thỏa mãn.

Tuy nhiên, như được mô tả ở trên, khi "transform_skip_flag[x0][y0][0] == 0" không được thỏa mãn, chắc chắn rằng chế độ ISP không được áp dụng, và do đó việc liệu chế độ ISP có được áp dụng hay không không cần phải được xem xét khi kiểm tra "!tu_y_coded_flag[x0][y0]". Do đó, Bảng 7 và Bảng 8 được xem là vẫn có hiệu lực.

Theo một ví dụ, trong VVC hiện tại, LFNST 1) có thể được áp dụng trong cây

tách biệt độ chói, 2) có thể được áp dụng trong cây tách biệt sắc độ, và 3) có thể được áp dụng chỉ cho thành phần độ chói trong cây đơn. Như được mô tả ở trên, do việc biến đổi không được áp dụng cho thành phần mà BDPCM được áp dụng cho và có CBF là 0, thành phần có thể được xem là không liên quan đến LFNST.

Trong cây tách biệt độ chói hoặc cây đơn, khi BDPCM được áp dụng cho thành phần độ chói và CBF là 0, LFNST về cơ bản không được áp dụng, và do đó không cần thiết phải kiểm tra giá trị CBF cho thành phần độ chói. Do đó, trong cây tách biệt độ chói và cây đơn, `lfnstNotTsFlag` có thể được dẫn xuất như được thể hiện trong Bảng 11 hoặc Bảng 12 mà không kiểm tra giá trị CBF cho thành phần độ chói.

[Bảng 11]

```
lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
    transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 ) &&
    ( treeType == DUAL_TREE_LUMA ||
    ( ( !tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 ) &&
    ( !tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 ) ) )
```

[Bảng 12]

```
lfnstNotTsFlag = transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 &&
    ( !tu_cb_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 )
    && ( !tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 )
```

Khi Bảng 11 hoặc Bảng 12 được áp dụng, các thao tác trong tất cả các trường hợp mà cho nó BDPCM và LFNST được áp dụng được mô tả như dưới đây.

1. BDPCM không được áp dụng cho độ chói và BDPCM không được áp dụng cho sắc độ: Cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện.

2. BDPCM được áp dụng cho độ chói và BDPCM không được áp dụng cho sắc độ

2-1) Cây tách biệt độ chói: Cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện. Nghĩa là, trong cả hai trường hợp trong đó CBF cho thành phần độ chói là 0 và trong đó CBF cho thành phần độ chói là 1, chỉ số LFNST được suy ra là 0 mà không được báo hiệu.

2-2) Cây tách biệt sắc độ: Cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện.

2-3) Cây đơn: Cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện, bởi vì CBF cho thành phần độ chói không được kiểm tra trong Bảng 12.

3. BDPCM không được áp dụng cho độ chói và BDPCM được áp dụng cho sắc độ

3-1) Cây tách biệt độ chói: Cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện.

3-2) Cây tách biệt sắc độ: Khi giá trị CBF là 0 cho cả thành phần Cb và thành phần Cr, chỉ số LFNST được suy ra là 0 mà không được báo hiệu, và do đó cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện.

3-3) Cây đơn: Mặc dù giá trị CBF là 0 cho cả thành phần Cb và thành phần Cr, BDPCM không được áp dụng cho thành phần độ chói, và do đó LFNST có thể được áp dụng. Theo đó, chỉ số LFNST có thể được báo hiệu. Tuy nhiên, trong tiêu chuẩn VVC hiện tại, LFNST không được áp dụng trong trường hợp này.

4. BDPCM được áp dụng cho độ chói và BDPCM được áp dụng cho sắc độ: Do BDPCM được áp dụng cho thành phần độ chói, cờ bỏ qua biến đổi tương ứng là 1 và giá trị của `lnstNotTsFlag` là 0, và do đó chỉ số LFNST không được báo hiệu. Cuối cùng, cùng thao tác như trong tiêu chuẩn VVC hiện tại, nghĩa là, Bảng 3, được thực hiện.

Như được đề cập ở trên, theo tiêu chuẩn VVC hiện tại, trong cây tách biệt độ chói và cây đơn, LFNST có thể được áp dụng chỉ cho thành phần độ chói. Trong cây tách biệt độ chói, khi giá trị CBF cho thành phần độ chói là 0, giá trị của giá trị biến `LnstDcOnly` được duy trì là 1 trong Bảng 3, và do đó chỉ số LFNST được suy ra là 0 mà không được báo hiệu.

Tuy nhiên, trong cây đơn, khi giá trị của `LnstDcOnly` được cập nhật thành 0 bởi thành phần sắc độ (giá trị của `LnstDcOnly` được cập nhật thành 0 khi việc lập mã phần dư tương ứng được thực hiện và do đó hệ số biến đổi khác không cuối cùng tồn tại bên ngoài vị trí DC) mặc dù giá trị CBF cho thành phần độ chói là 0 và khi điều kiện đặt bằng không cho LFNST được thỏa mãn đối với thành phần sắc độ và do đó giá trị của `LnstZeroOutSigCoeffFlag` được duy trì là 0 (trong trường hợp trong đó kích thước của khối biến đổi là 4×4 và 8×8 như trường hợp trong đó hệ số biến đổi tồn tại chỉ trong vùng trong đó hệ số biến đổi có thể tồn tại khi LFNST xuôi được áp dụng, hệ số biến đổi có thể tồn tại ở các vị trí từ vị trí DC đến vị trí thứ tám theo thứ tự quét xuôi, và trong các trường hợp khác, các hệ số biến đổi có thể tồn tại trong vùng 4×4 trên cùng bên trái của TB. Ở đây, vị trí DC dùng để chỉ vị trí trên cùng bên trái của khối biến đổi), chỉ số LFNST có thể được báo hiệu.

Do đó, trong tiêu chuẩn VVC hiện tại, trong cây đơn, mặc dù chỉ số LFNST được báo hiệu, giá trị CBF cho thành phần độ chói là 0, và do đó LFNST không được áp dụng cho thành phần bất kỳ, và do đó chỉ số LFNST có thể được báo hiệu dư.

Bằng việc cải biến giá trị của biến `lnstNotTsFlag` được thiết đặt trong Bảng 3 như được thể hiện trong Bảng 13, chỉ số LFNST có thể được tạo cấu hình để được suy ra là 0 mà không được báo hiệu khi giá trị CBF cho thành phần độ chói là 0 trong cây đơn.

Ở đây, điều kiện "`IntraSubPartitionsSplitType != ISP_NO_SPLIT`" trong Bảng 13 chỉ ra rằng chế độ ISP được áp dụng. Như được mô tả ở trên, khi chế độ ISP được áp dụng, CBF cho ít nhất một trong số các khối phân vùng là khác 0 và

`tu_y_coded_flag[ x0 ][ y0 ]` là chỉ CBF cho khối phân vùng thứ nhất trong thứ tự lập mã, và do đó `tu_y_coded_flag[x0][y0]` được kiểm tra chỉ khi điều kiện "IntraSubPartitionsSplitType != ISP_NO_SPLIT" được phân tách độc lập và chế độ ISP không được áp dụng.

[Bảng 13]

```

lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
                  ((IntraSubPartitionsSplitType != ISP_NO_SPLIT || tu_y_coded_flag[ x0 ][ y0 ])
                  && transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0)) &&
                  ( treeType == DUAL_TREE_LUMA
                  || ( transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 &&
                      transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 ))

```

Như được thể hiện trong Bảng 7 đến Bảng 10, khi việc báo hiệu chỉ số LFNST được cho phép chỉ khi BDPCM được áp dụng và giá trị CBF là 0 (nghĩa là, được coi là khối biến đổi mà cho nó việc biến đổi không được áp dụng trong trường hợp này), biến `lfnstNotTsFlag` có thể được thiết đặt như được thể hiện trong Bảng 14.

[Bảng 14]

```

lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
                  ((IntraSubPartitionsSplitType != ISP_NO_SPLIT ||
                    tu_y_coded_flag[ x0 ][ y0 ]) &&
                    transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0)) &&
                  ( treeType == DUAL_TREE_LUMA || (( !tu_cb_coded_flag[ x0 ][ y0 ] ||
                    transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0) &&
                    ( !tu_cr_coded_flag[ x0 ][ y0 ] || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 )))

```

Khi Bảng 14 được áp dụng, 1) trong cây tách biệt độ chói, chỉ số LFNST được

báo hiệu chỉ khi giá trị CBF cho thành phần độ chói là 1 giống như trong Bảng 3, và 2) trong cây tách biệt sắc độ, chỉ số LFNST không được báo hiệu khi tất cả các giá trị CBF cho các thành phần sắc độ (Cb và Cr) là 0 như trong Bảng 7 đến Bảng 12, ví dụ, khi BDPCM được áp dụng và tất cả các giá trị CBF cho thành phần Cb và thành phần Cr là 0. Ngoài ra, 3) trong cây đơn, chỉ số LFNST có thể được báo hiệu khi giá trị CBF cho thành phần độ chói là 1 và 3-1) việc bỏ qua biến đổi không được áp dụng cho cả thành phần Cb và thành phần Cr hoặc 3-2) tất cả các giá trị CBF cho thành phần Cb và thành phần Cr là 0 khi BDPCM được áp dụng cho thành phần sắc độ.

Sau đây, phương án trong đó giá trị CBF cho khối biến đổi được suy ra là 1 khi chế độ BDPCM được áp dụng và phương pháp báo hiệu chỉ số LFNST hiện có được sử dụng như sẽ được mô tả.

Khi mỗi thành phần màu sắc được lập mã trong chế độ BDPCM, phương pháp báo hiệu chỉ số LFNST được thể hiện trong Bảng 3, nghĩa là, thiết đặt hiện có cho biến `lnstNotTsFlag`, có thể được sử dụng cho như chính nó bằng việc suy giá trị CBF cho thành phần là 1. Theo phương án này, bảng cú pháp cho đơn vị biến đổi của Bảng 6 có thể được thay đổi như được thể hiện trong Bảng 15.

[Bảng 15]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Descriptor
<pre> if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions - 1) { </pre>	
<pre> xC = CbPosX[chType][x0][y0] </pre>	
<pre> yC = CbPosY[chType][x0][y0] </pre>	
<pre> wC = CbWidth[chType][x0][y0] / SubWidthC </pre>	
<pre> hC = CbHeight[chType][x0][y0] / SubHeightC </pre>	
<pre> } else { </pre>	
<pre> xC = x0 </pre>	
<pre> yC = y0 </pre>	
<pre> wC = tbWidth / SubWidthC </pre>	
<pre> hC = tbHeight / SubHeightC </pre>	
<pre> } </pre>	
<pre> chromaAvailable = treeType != DUAL_TREE_LUMA && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1)) </pre>	
<pre> if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag))) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex == NumIntraSubPartitions - 1))) { </pre>	
<pre> if(BdpcmFlag[xC][yC][1] == 0) </pre>	
<pre> tu_cb_coded_flag[xC][yC] </pre>	ae(v)
<pre> if(BdpcmFlag[xC][yC][2] == 0) </pre>	
<pre> tu_cr_coded_flag[xC][yC] </pre>	ae(v)
<pre> } </pre>	

<pre> 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 (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))) && !BdpcmFlag[x0][y0][0]) </pre>	
<pre> tu_y_coded_flag[x0][y0] </pre>	ae(v)
<pre> if(IntraSubPartitionsSplitType != ISP_NO_SPLIT) </pre>	
<pre> InferTuCbfLuma = InferTuCbfLuma && !tu_y_coded_flag[x0][y0] </pre>	
<pre> } </pre>	
<pre> </pre>	
<pre> 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) </pre>	
<pre> tu_joint_cbr_residual_flag[xC][yC] </pre>	ae(v)
<pre> </pre>	
<pre> } </pre>	

Như được thể hiện trong Bảng 3, do ISP và BDPCM là loại trừ lẫn nhau, nghĩa là, `intra_subpartitions_mode_flag[ x0 ][ y0 ]` có thể được phân tích cú pháp chỉ khi `intra_bdpcm_luma_flag` là 0, ISP không được cho phép khi BDPCM được cho phép, và BDPCM không được cho phép khi ISP được cho phép. Trong Bảng 15, điều kiện "`IntraSubPartitionsSplitType != ISP_NO_SPLIT`" chỉ ra rằng chế độ ISP được áp dụng, và điều kiện `!BdpcmFlag[ x0 ][ y0 ][ 0 ]` có thể được biểu diễn như được thể hiện trong Bảng 16.

[Bảng 16]

<pre> 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 (chromaAvailable && (tu_cb_coded_flag[xC][yC] tu_cr_coded_flag[xC][yC]))) CbWidth[chType][x0][y0] > MaxTbSizeY CbHeight[chType][x0][y0] > MaxTbSizeY) && !BdpcmFlag[x0][y0][0]) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfluma))) </pre>	
<code>tu_y_coded_flag[x0][y0]</code>	<code>ae(v)</code>

Ngoài ra đến Bảng 15, nội dung mà giá trị CBF tương ứng được suy ra là 1 khi BDPCM được áp dụng có thể được phản ánh trong văn bản đặc tả cho CBF (Bảng 17).

[Bảng 17]

.....

tu_cb_coded_flag[x0][y0] equal to 1 specifies that the Cb transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the top-left location (x0, y0) of the considered transform block.

When **tu_cb_coded_flag[x0][y0]** is not present, its value is inferred as follows:

- If **BdpcmFlag[x0][y0][1]** is equal to 1, **tu_cb_coded_flag[x0][y0]** is inferred to be equal to 1.
- Otherwise (**BdpcmFlag[x0][y0][1]** is equal to 0), **tu_cb_coded_flag[x0][y0]** is inferred to be equal to 0.

tu_cr_coded_flag[x0][y0] equal to 1 specifies that the Cr transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the top-left location (x0, y0) of the considered transform block.

When **tu_cr_coded_flag[x0][y0]** is not present, its value is inferred as follows:

- If **BdpcmFlag[x0][y0][2]** is equal to 1, **tu_cr_coded_flag[x0][y0]** is inferred to be equal to 1.
- Otherwise (**BdpcmFlag[x0][y0][2]** is equal to 0), **tu_cr_coded_flag[x0][y0]** is inferred to be equal to 0.

tu_y_coded_flag[x0][y0] equal to 1 specifies that the luma transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

When **tu_y_coded_flag[x0][y0]** is not present, its value is inferred as follows:

- If **cu_sbt_flag** is equal to 1 and one of the following conditions is true, **tu_y_coded_flag[x0][y0]** is inferred to be equal to 0:
 - **subTuIndex** is equal to 0 and **cu_sbt_pos_flag** is equal to 1.
 - **subTuIndex** is equal to 1 and **cu_sbt_pos_flag** is equal to 0.
- Otherwise, if **treeType** is equal to **DUAL_TREE_CHROMA**, **tu_y_coded_flag[x0][y0]** is inferred to be equal to 0.
- Otherwise, **tu_y_coded_flag[x0][y0]** is inferred to be equal to 1.

...

Khi BDPCM được áp dụng cho sắc độ cho đơn vị lập mã được lập mã hiện tại, Cb-Cr liên kết có thể được cho phép như chính nó hoặc có thể được bất hoạt. Trong Bảng 15, **tu_joint_cbr_residual_flag[xC][yC]**, mà là phần tử cú pháp chỉ ra xem Cb-Cr liên kết có được cho phép hay không, được báo hiệu.

Như được thể hiện trong Bảng 17, khi BDPCM được áp dụng cho thành phần sắc độ (nghĩa là, khi cả **BdpcmFlag[x0][y0][1]** và **BdpcmFlag[x0][y0][2]** là 1), nếu các giá trị CBF cho thành phần Cb và thành phần Cr được suy ra là 1, Cb-Cr liên kết có thể được áp dụng chỉ khi cả **tu_cb_coded_flag[xC][yC]** và **tu_cr_coded_flag[xC][yC]** là 1. Bảng 18 thể hiện đặc tả kỹ thuật liên quan đến thao tác Cb-Cr liên kết.

[Bảng 18]

.....

ph_joint_cbr_residual_flag specifies whether, in transform units with **tu_joint_cbr_residual_flag**[x0][y0] equal to 1, the collocated residual samples of both chroma components have inverted signs. When **tu_joint_cbr_residual_flag**[x0][y0] equal to 1 for a transform unit, **ph_joint_cbr_residual_flag** equal to 0 specifies that the sign of each residual sample of the Cr (or Cb) component is identical to the sign of the collocated Cb (or Cr) residual sample and **ph_joint_cbr_residual_flag** equal to 1 specifies that the sign of each residual sample of the Cr (or Cb) component is given by the inverted sign of the collocated Cb (or Cr) residual sample.

.....

tu_joint_cbr_residual_flag[x0][y0] specifies whether the residual samples for both chroma components Cb and Cr are coded as a single transform block. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

tu_joint_cbr_residual_flag[x0][y0] equal to 1 specifies that the transform unit syntax includes the transform coefficient levels for a single transform block from which the residual samples for both Cb and Cr are derived. **tu_joint_cbr_residual_flag**[x0][y0] equal to 0 specifies that the transform coefficient levels of the chroma components are coded as indicated by the syntax elements **tu_cb_coded_flag**[x0][y0] and **tu_cr_coded_flag**[x0][y0].

When **tu_joint_cbr_residual_flag**[x0][y0] is not present, it is inferred to be equal to 0.

Depending on **tu_joint_cbr_residual_flag**[x0][y0], **tu_cb_coded_flag**[x0][y0], and **tu_cr_coded_flag**[x0][y0], the variable **TuCRResMode**[x0][y0] is derived as follows:

- If **tu_joint_cbr_residual_flag**[x0][y0] is equal to 0, the variable **TuCRResMode**[x0][y0] is set equal to 0.
- Otherwise, if **tu_cb_coded_flag**[x0][y0] is equal to 1 and **tu_cr_coded_flag**[x0][y0] is equal to 0, the variable **TuCRResMode**[x0][y0] is set equal to 1.
- Otherwise, if **tu_cb_coded_flag**[x0][y0] is equal to 1, the variable **TuCRResMode**[x0][y0] is set equal to 2.
- Otherwise, the variable **TuCRResMode**[x0][y0] is set equal to 3.

.....

8.7.2 Scaling and transformation process

Inputs to this process are:

- a luma location ($xTbY$, $yTbY$) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable $cIdx$ specifying the colour component of the current block,
- a variable $predMode$ specifying the prediction mode of the coding unit,
- a variable $nTbW$ specifying the transform block width,
- a variable $nTbH$ specifying the transform block height.

Output of this process is the $(nTbW) \times (nTbH)$ array of residual samples $resSamples[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

The variable $codedCIdx$ is derived as follows:

- If $cIdx$ is equal to 0 or $TuCRMode[xTbY][yTbY]$ is equal to 0, $codedCIdx$ is set equal to $cIdx$.
- Otherwise, if $TuCRMode[xTbY][yTbY]$ is equal to 1 or 2, $codedCIdx$ is set equal to 1.
- Otherwise, $codedCIdx$ is set equal to 2.

The variable $cSign$ is set equal to $(1 - 2 * ph_joint_cbr_sign_flag)$.

The $(nTbW) \times (nTbH)$ array of residual samples $resSamples$ is derived as follows:

1. The scaling process for transform coefficients as specified in clause 8.7.3 is invoked with the transform block location ($xTbY$, $yTbY$), the transform block width $nTbW$ and the transform block height $nTbH$, the prediction mode $predMode$, and the colour component variable $cIdx$ being set equal to $codedCIdx$ as inputs, and the output is an $(nTbW) \times (nTbH)$ array of scaled transform coefficients d .
2. The $(nTbW) \times (nTbH)$ array of residual samples res is derived as follows:
 - If $transform_skip_flag[xTbY][yTbY][codedCIdx]$ is equal to 1, the residual sample array values $res[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived as follows:

$$res[x][y] = d[x][y] \quad (1150)$$
 - Otherwise ($transform_skip_flag[xTbY][yTbY][codedCIdx]$ is equal to 0), the transformation process for scaled transform coefficients as specified in clause 8.7.4.1 is invoked with the transform block location ($xTbY$, $yTbY$), the transform block width $nTbW$ and the transform block height $nTbH$, the colour component variable $cIdx$ being set equal to $codedCIdx$ and the $(nTbW) \times (nTbH)$ array of scaled transform coefficients d as inputs, and the output is an $(nTbW) \times (nTbH)$ array of residual samples res .
3. The residual samples $resSamples[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived as follows:
 - If $cIdx$ is equal to $codedCIdx$, the following applies:

$$resSamples[x][y] = res[x][y] \quad (1151)$$
 - Otherwise, if $TuCRMode[xTbY][yTbY]$ is equal to 2, the following applies:

$$resSamples[x][y] = cSign * res[x][y] \quad (1152)$$
 - Otherwise, the following applies:

$$resSamples[x][y] = (cSign * res[x][y]) >> 1 \quad (1153)$$

.....

Như được thể hiện trong Bảng 18, khi chế độ Cb-Cr liên kết được áp dụng, hệ số biến đổi cho một trong số các thành phần sắc độ được báo hiệu và được lập mã, nghĩa là, thành phần được định rõ bởi $codedCIdx$ trong Bảng 18 được lập mã. $codedCIdx$ có thể có giá trị của 1 hoặc 2, trong đó 1 chỉ ra thành phần Cb và 2 chỉ ra thành phần Cr. Ở đây, dữ liệu phần dư cho thành phần khác được xác định như giá trị của dữ liệu phần dư cho thành phần được lập mã được nhân với giá trị của $cSign$ và sau đó được dịch chuyển bởi 1 sang phải hoặc không được dịch chuyển.

Khi BDPCM được áp dụng cho thành phần sắc độ, do giá trị của $TuCResMode[x0][y0]$ được xác định là 2 như được thể hiện trong Bảng 18, dữ liệu phần dư cho thành phần khác được xác định như giá trị không được dịch chuyển bởi 1 sang phải ($resSamples[x][y] = cSign * res[x][y]$) và giá trị của $cSign$ được xác định như $(1 - 2 * ph_joint_cbr_sign_flag)$ dựa trên phần tử cú pháp $ph_joint_cbr_sign_flag$.

Ngoài ra, như được thể hiện trong Bảng 19, khi BDPCM không được áp dụng cho thành phần sắc độ, chế độ Cb-Cr liên kết có thể không được áp dụng. Trong Bảng 19, chỉ một phần của việc báo hiệu $tu_joint_cbr_residual_flag[xC][yC]$ trong Bảng 15 được thay đổi. Nghĩa là, khi BDPCM không được áp dụng cho thành phần sắc độ, chế độ Cb-Cr liên kết được giới hạn không được áp dụng.

[Bảng 19]

<pre> 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 && intra_bdpcm_chroma_flag == 0) </pre>	
$tu_joint_cbr_residual_flag[xC][yC]$	ae(v)

Sử dụng biến $BdpcmFlag[x0][y0][1]$ và biến $BdpcmFlag[x0][y0][2]$, thông tin cú pháp mà thao tác theo cùng cách như trong Bảng 19 có thể được tạo cấu hình như được thể hiện trong Bảng 20.

[Bảng 20]

<pre> 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 && !BdpcmFlag[x0][y0][1] && !BdpcmFlag[x0][y0][2]) </pre>	
$tu_joint_cbr_residual_flag[xC][yC]$	ae(v)

Theo một ví dụ, khi chế độ BDPCM được áp dụng, ràng buộc thích ứng luồng bit có thể được gán cho tín hiệu giá trị CBF cho khối biến đổi tương ứng là 1.

Khi chế độ BDPCM được áp dụng và giá trị CBF là 0, cờ bỏ qua biến đổi cho thành phần tương ứng được suy ra là 1. Trong trường hợp này, mặc dù thành phần là không liên quan đến quy trình biến đổi do sự không tồn tại của hệ số biến đổi, cờ bỏ qua biến đổi được suy ra là 1, và do đó chỉ số LFNST không được báo hiệu. Để ngăn ngừa vấn đề này, khi chế độ BDPCM được áp dụng, ràng buộc thích ứng luồng bit có thể được gán cho tín hiệu giá trị CBF cho khối biến đổi tương ứng là 1.

Bảng 21 thể hiện rằng ràng buộc thích ứng luồng bit được bổ sung vào các ngữ nghĩa cho mỗi CBF.

[Bảng 21]

.....

tu_cb_coded_flag[x0][y0] equal to 1 specifies that the Cb transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the top-left location (x0, y0) of the considered transform block.

When **tu_cb_coded_flag**[x0][y0] is not present, its value is inferred to be equal to 0.

It is a requirement of bitstream conformance that **tu_cb_coded_flag**[x0][y0] shall be equal to 1 when **tu_cb_coded_flag**[x0][y0] is present and **BdpcmFlag**[x0][y0][1] is equal to 1.

tu_cr_coded_flag[x0][y0] equal to 1 specifies that the Cr transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the top-left location (x0, y0) of the considered transform block.

When **tu_cr_coded_flag**[x0][y0] is not present, its value is inferred to be equal to 0.

It is a requirement of bitstream conformance that **tu_cr_coded_flag**[x0][y0] shall be equal to 1 when **tu_cr_coded_flag**[x0][y0] is present and **BdpcmFlag**[x0][y0][2] is equal to 1.

tu_y_coded_flag[x0][y0] equal to 1 specifies that the luma transform block contains one or more transform coefficient levels not equal to 0. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered transform block relative to the top-left luma sample of the picture.

When **tu_y_coded_flag**[x0][y0] is not present, its value is inferred as follows:

- If **cu_sbt_flag** is equal to 1 and one of the following conditions is true, **tu_y_coded_flag**[x0][y0] is inferred to be equal to 0:
 - **subTuIndex** is equal to 0 and **cu_sbt_pos_flag** is equal to 1.
 - **subTuIndex** is equal to 1 and **cu_sbt_pos_flag** is equal to 0.
- Otherwise, if **treeType** is equal to **DUAL_TREE_CHROMA**, **tu_y_coded_flag**[x0][y0] is inferred to be equal to 0.
- Otherwise, **tu_y_coded_flag**[x0][y0] is inferred to be equal to 1.

It is a requirement of bitstream conformance that **tu_y_coded_flag**[x0][y0] shall be equal to 1 when **tu_y_coded_flag**[x0][y0] is present and **BdpcmFlag**[x0][y0][0] is equal to 1.

.....

Các hình vẽ sau đây được cung cấp để mô tả các ví dụ cụ thể của sáng chế. Vì các thuật ngữ cụ thể cho các thiết bị hoặc các thuật ngữ cụ thể cho các tín hiệu/các tin nhắn/các trường cụ thể được minh họa trong các hình vẽ được cung cấp để minh họa,

các dấu hiệu kỹ thuật của sáng chế không bị giới hạn ở các thuật ngữ cụ thể được sử dụng trong các hình vẽ sau.

Fig.15 là lưu đồ minh họa thao tác của thiết bị giải mã video theo một phương án của sáng chế.

Mỗi quy trình được bộc lộ trên Fig.15 là dựa trên một số chi tiết được mô tả tham chiếu tới Fig.5 đến Fig.14. Do đó, phần mô tả của các chi tiết cụ thể trùng lặp với các chi tiết được mô tả tham chiếu tới Fig.3 và Fig.5 đến Fig.14 sẽ được bỏ qua hoặc sẽ được thực hiện sơ lược.

Thiết bị giải mã 300 theo phương án có thể nhận thông tin phần dư từ luồng bit (S1510).

Cụ thể là, thiết bị giải mã 300 có thể giải mã thông tin về các hệ số biến đổi được lượng tử hóa cho khối hiện tại từ luồng bit, và có thể dẫn xuất các hệ số biến đổi được lượng tử hóa cho khối mục tiêu dựa trên thông tin về các hệ số biến đổi được lượng tử hóa cho khối hiện tại. Thông tin về các hệ số biến đổi được lượng tử hóa cho khối mục tiêu có thể được gồm trong tập thông số trình tự (SPS) hoặc phần đầu lát và có thể gồm ít nhất một trong số thông tin về liệu RST có được áp dụng hay không, thông tin về hệ số được giảm, thông tin về kích thước biến đổi tối thiểu để áp dụng RST, thông tin về kích thước biến đổi tối đa để áp dụng RST, kích thước RST ngược, và thông tin về chỉ số biến đổi chỉ ra bất kỳ một trong số các ma trận nhân biến đổi được gồm trong tập biến đổi.

Thiết bị giải mã có thể còn nhận thông tin về chế độ dự đoán nội cho khối hiện tại và thông tin về việc liệu ISP có được áp dụng cho khối hiện tại hay không. Thiết bị giải mã có thể nhận và phân tích cú pháp thông tin chỉ ra xem liệu áp dụng việc lập mã ISP hay chế độ ISP, bởi đó dẫn xuất xem khối hiện tại có được phân chia thành số lượng được xác định trước của các khối biến đổi phân vùng con hay không. Ở đây, khối hiện tại có thể là khối lập mã. Ngoài ra, thiết bị giải mã có thể dẫn xuất kích thước và số

lượng của các khối phân vùng con được phân chia thông qua thông tin cờ chỉ ra hướng trong đó khối hiện tại được phân chia.

Theo một ví dụ, chỉ số LFNST có thể được nhận thông qua luồng bit. Chỉ số LFNST là giá trị để định rõ ma trận LFNST khi LFNST được áp dụng như việc biến đổi không thể phân tách thứ cấp ngược và có thể có giá trị nằm trong khoảng từ 0 đến 2. Ví dụ, giá trị chỉ số LFNST là 0 có thể chỉ ra rằng không có LFNST nào được áp dụng cho khối hiện tại, giá trị chỉ số LFNST là 1 có thể chỉ ra ma trận LFNST thứ nhất, và giá trị chỉ số LFNST là 2 có thể chỉ ra ma trận LFNST thứ hai.

Thông tin về ISP và chỉ số LFNST có thể được nhận trong mức đơn vị lập mã.

Thiết bị giải mã 300 có thể dẫn xuất các hệ số biến đổi cho khối hiện tại bằng việc thực hiện việc khử lượng tử hóa dựa trên thông tin phân dư (S1520).

Ở đây, khối hiện tại có thể là khối biến đổi, mà là đơn vị biến đổi, và khi loại cây của khối hiện tại là cây đơn, khối hiện tại có thể gồm khối biến đổi cho thành phần độ chói, khối biến đổi cho thành phần sắc độ thứ nhất, và khối biến đổi cho thành phần sắc độ thứ hai. Khi loại cây của khối hiện tại là độ chói cây đôi, khối hiện tại có thể gồm khối biến đổi cho thành phần độ chói, và khi loại cây của khối hiện tại là sắc độ cây đôi, khối hiện tại có thể gồm khối biến đổi cho thành phần sắc độ thứ nhất và khối biến đổi cho thành phần sắc độ thứ hai.

Các hệ số biến đổi được dẫn xuất có thể được sắp xếp trong các đơn vị khối 4 x 4 theo thứ tự quét chéo ngược, và các hệ số biến đổi trong khối 4 x 4 cũng có thể được sắp xếp theo thứ tự quét chéo ngược. Nghĩa là, các hệ số biến đổi được khử lượng tử hóa có thể được sắp xếp theo thứ tự quét ngược được áp dụng trong codec video, chẳng hạn như trong VVC hoặc HEVC.

Thiết bị giải mã có thể thiết đặt biến áp dụng LFNST về việc liệu LFNST có được áp dụng hay không dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của

khởi hiện tại (S1530).

Biến áp dụng LFNST có thể là biến cờ, ví dụ, `lfnstNotTsFlag`, và `lfnstNotTsFlag` có thể được dẫn xuất dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại. Biến áp dụng LFNST bằng 1 chỉ ra rằng LFNST là có thể áp dụng, và biến áp dụng LFNST bằng 0 chỉ ra rằng không có LFNST nào có thể áp dụng. Nghĩa là, khi biến áp dụng LFNST là 1, chỉ số LFNST có thể được phân tích cú pháp, và khi biến áp dụng LFNST là 0, chỉ số LFNST có thể không được phân tích cú pháp.

Cờ lập mã hệ số biến đổi dùng để chỉ CBF chỉ ra xem hệ số biến đổi cho khối biến đổi có tồn tại hay không, trong đó khi CBF bằng 1, mà chỉ ra rằng hệ số biến đổi cho khối biến đổi tồn tại, quy trình biến đổi có thể được thực hiện, và khi CBF bằng 0, không có quy trình biến đổi nào được thực hiện.

Theo một ví dụ, giá trị của biến áp dụng LFNST có thể được dẫn xuất là 1 dựa trên các cờ lập mã hệ số biến đổi cho tất cả các thành phần của khối hiện tại là 0. Trong trường hợp này, giá trị của biến áp dụng LFNST được thiết đặt là 1, nhưng khi CBF là 0, không có quy trình biến đổi thực nào được thực hiện và việc lập mã phần dư không được nhập, và do đó biến `lfnstDcOnly` sẽ được mô tả sau không được cập nhật thành 0. Nghĩa là, khi các cờ lập mã hệ số biến đổi cho tất cả các thành phần của khối hiện tại là 0, không có LFNST nào được áp dụng.

Theo một ví dụ, giá trị của biến áp dụng LFNST có thể được thiết đặt là 1 dựa trên các cờ bỏ qua biến đổi cho tất cả các thành phần của khối hiện tại là 0. Nghĩa là, khi bỏ qua biến đổi không được áp dụng cho tất cả các thành phần của khối hiện tại, LFNST có thể được áp dụng.

Theo một ví dụ, khi loại cây của khối hiện tại là cây đơn, chế độ BDPCM có thể được áp dụng cho mỗi trong số thành phần độ chói và thành phần sắc độ. Nghĩa là, chế độ BDPCM có thể được áp dụng cho thành phần độ chói và có thể không được áp dụng

cho thành phần sắc độ. Ngược lại, chế độ BDPCM có thể được áp dụng cho thành phần sắc độ và có thể không được áp dụng cho thành phần độ chói.

Theo tiêu chuẩn hiện tại, trong cây đơn, LFNST có thể được áp dụng chỉ cho thành phần độ chói mà không được áp dụng cho thành phần sắc độ. Ở đây, cho thành phần độ chói, chế độ BDPCM không được áp dụng và hệ số biến đổi tồn tại, và do đó việc biến đổi có thể được thực hiện (nghĩa là, trong đó chế độ BDPCM không được áp dụng cho thành phần độ chói, CBF là khác 0, và cờ bỏ qua biến đổi là khác 0), và có trường hợp trong đó chế độ BDPCM là có thể áp dụng cho thành phần sắc độ. Khi chế độ BDPCM được áp dụng, cờ bỏ qua biến đổi được suy ra là 1.

Một cách thông thường, khi dẫn xuất biến áp dụng LFNST, nếu cờ bỏ qua biến đổi cho thành phần bất kỳ được gồm trong khối là 1, biến áp dụng LFNST được thiết đặt là 0, và do đó chỉ số LFNST không được phân tích cú pháp. Nghĩa là, trong trường hợp loại cây đơn, mặc dù LFNST là có thể áp dụng cho thành phần độ chói, biến áp dụng LFNST được thiết đặt là 0 do thành phần sắc độ mà cho nó chế độ BDPCM được áp dụng (do cờ bỏ qua biến đổi được suy ra là 1 khi chế độ BDPCM được áp dụng).

Theo một ví dụ, khi chế độ BDPCM được áp dụng cho khối cụ thể, có thể là trường hợp trong đó CBF là 0, và do đó vấn đề nêu trên có thể được giải quyết sử dụng trường hợp này. Thiết bị giải mã có thể áp dụng LFNST cho thành phần độ chói mặc dù chế độ BDPCM được áp dụng cho thành phần sắc độ bằng việc bổ sung điều kiện rằng CBF là 0 cho điều kiện cho việc dẫn xuất biến áp dụng LFNST.

Thiết bị giải mã có thể thiết đặt điều kiện cho việc dẫn xuất biến áp dụng LFNST là "(treeType == DUAL_TREE_CHROMA || (!tu_y_coded_flag[x0][y0] || transform_skip_flag[x0][y0][0] == 0)) && (treeType == DUAL_TREE_LUMA || ((!tu_cb_coded_flag[x0][y0] || transform_skip_flag[x0][y0][1] == 0) && (!tu_cr_coded_flag[x0][y0] || transform_skip_flag[x0][y0][2] == 0)))".

Theo điều kiện, do loại cây của khối hiện tại là cây đơn ("treeType == DUAL_TREE_CHROMA" không được thỏa mãn, và do đó điều kiện "(!tu_y_coded_flag[x0][y0] || transform_skip_flag[x0][y0][0] == 0)" được kiểm tra) và cờ bỏ qua biến đổi cho thành phần độ chói là 0, điều kiện "transform_skip_flag[x0][y0][0] == 0" được thỏa mãn.

Ngoài ra, khi chế độ BDPCM được áp dụng cho thành phần sắc độ của khối hiện tại và cờ lập mã hệ số biến đổi là 0, điều kiện ((!tu_cb_coded_flag[x0][y0] || transform_skip_flag[x0][y0][1] == 0) && (!tu_cr_coded_flag[x0][y0] || transform_skip_flag[x0][y0][2] == 0)) được thỏa mãn. Nghĩa là, mặc dù chế độ BDPCM được áp dụng cho thành phần sắc độ và do đó cờ bỏ qua biến đổi được suy ra là 1, giá trị của biến áp dụng LFNST có thể được thiết đặt là 1, mà chỉ ra rằng chỉ số LFNST có thể được phân tích cú pháp, theo điều kiện cờ lập mã hệ số biến đổi.

Như được mô tả ở trên, bằng việc thay đổi điều kiện cho việc thiết đặt biến áp dụng LFNST, có thể giải quyết vấn đề rằng LFNST không được áp dụng cho tất cả các khối biến đổi bất kể giá trị CBF khi chế độ BDPCM được áp dụng cho khối biến đổi bất kỳ. Ngoài ra, khi kiểm tra điều kiện để xác định xem có phân tích cú pháp chỉ số LFNST hay không, cờ bỏ qua biến đổi và giá trị CBF được kiểm tra cùng nhau cho mỗi thành phần màu sắc. Do đó, khi giá trị CBF là 0 mặc dù bất kỳ trong số các thành phần màu sắc là trong chế độ BDPCM, việc phân tích cú pháp của chỉ số LFNST cho thành phần khác không bị ảnh hưởng.

Thiết bị giải mã có thể phân tích cú pháp chỉ số LFNST dựa trên giá trị của biến áp dụng LFNST (S1540).

Thiết bị giải mã có thể phân tích cú pháp chỉ số LFNST dựa trên giá trị của biến áp dụng LFNST là 1, và có thể dẫn xuất các biến khác nhau ngoài biến áp dụng LFNST để phân tích cú pháp chỉ số LFNST.

Ví dụ, thiết bị giải mã có thể dẫn xuất biến thứ nhất (biến `LfnstDcOnly`) chỉ ra xem hệ số đáng kể có tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối hiện tại hay không và biến thứ hai (biến `LfnstZeroOutSigCoeffFlag`) chỉ ra xem hệ số biến đổi có tồn tại trong vùng thứ hai ngoài vùng thứ nhất trên cùng bên trái của khối hiện tại hay không. Ngoài ra, thiết bị giải mã có thể xác định việc liệu có áp dụng LFNST hay không, nghĩa là, xem có phân tích cú pháp chỉ số LFNST hay không, dựa trên loại cây và kích thước của khối hiện tại.

Biến thứ nhất và biến thứ hai được thiết đặt ban đầu là 1, trong đó biến thứ nhất có thể được cập nhật là 0 khi hệ số đáng kể tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối hiện tại, và biến thứ hai có thể được cập nhật là 0 khi hệ số biến đổi tồn tại trong vùng thứ hai.

Khi biến thứ nhất được cập nhật thành 0 và biến thứ hai được duy trì là 1, LFNST có thể được áp dụng cho khối hiện tại.

Đối với thành phần độ chói mà cho nó chế độ phân vùng con nội (ISP) là có thể áp dụng, chỉ số LFNST có thể được phân tích cú pháp mà không dẫn xuất biến `LfnstDcOnly`.

Cụ thể là, trong trường hợp trong đó chế độ ISP được áp dụng và cờ bỏ qua biến đổi, nghĩa là, `transform_skip_flag[x0][y0][0]`, cho thành phần độ chói là 0, khi loại cây của khối hiện tại là cây đơn hoặc cây đôi cho độ chói, chỉ số LFNST có thể được báo hiệu bất kể giá trị của biến `LfnstDcOnly`.

Tuy nhiên, cho thành phần sắc độ mà cho nó chế độ ISP không được áp dụng, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 theo giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cb, `transform_skip_flag[x0][y0][1]`, và giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cr, `transform_skip_flag[x0][y0][2]`. Nghĩa là, khi giá trị của `cIdx` trong `transform_skip_flag[x0][y0][cIdx]` là 1, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của

`transform_skip_flag[ x0 ][ y0 ][ 1 ]` là 0, và khi giá trị của `cIdx` là 2, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 2 ]` là 0. Khi giá trị của biến `LfnstDcOnly` là 0, thiết bị giải mã có thể phân tích cú pháp chỉ số LFNST, và nếu không, chỉ số LFNST có thể được suy ra là 0 mà không được báo hiệu.

Biến thứ hai có thể là biến `LfnstZeroOutSigCoeffFlag` mà có thể chỉ ra rằng việc đặt bằng không được thực hiện khi LFNST được áp dụng. Biến thứ hai có thể ban đầu được thiết đặt là 1, và có thể được thay đổi thành 0 khi hệ số đáng kể tồn tại trong vùng thứ hai.

Biến `LfnstZeroOutSigCoeffFlag` có thể được dẫn xuất là 0 khi chỉ số của khối con trong đó hệ số khác không cuối cùng tồn tại là lớn hơn 0 và cả chiều rộng và chiều cao của khối biến đổi là bằng hoặc lớn hơn 4 hoặc khi vị trí của hệ số khác không cuối cùng trong khối con trong đó hệ số khác không cuối cùng tồn tại là lớn hơn 7 và kích thước của khối biến đổi là 4 x 4 hoặc 8 x 8. Khối con dùng để chỉ khối 4 x 4 được sử dụng làm đơn vị lập mã trong việc lập mã phần dư và có thể được gọi là nhóm hệ số (CG). Chỉ số khối con của 0 dùng để chỉ khối con 4 x 4 trên cùng bên trái.

Nghĩa là, khi hệ số khác không được dẫn xuất trong vùng ngoài vùng trên cùng bên trái trong đó hệ số biến đổi LFNST có thể tồn tại trong khối biến đổi hoặc hệ số khác không tồn tại ở vị trí ngoài vị trí thứ tám trong thứ tự quét cho khối 4 x 4 hoặc khối 8 x 8, biến `LfnstZeroOutSigCoeffFlag` được thiết đặt là 0.

Thiết bị giải mã có thể xác định tập LFNST bao gồm các ma trận LFNST dựa trên chế độ dự đoán nội được dẫn xuất từ thông tin về chế độ dự đoán nội, và có thể lựa chọn bất kỳ trong số nhiều ma trận LFNST dựa trên tập LFNST và chỉ số LFNST.

Ở đây, cùng tập LFNST và cùng chỉ số LFNST có thể được áp dụng cho các khối biến đổi phân vùng con mà khối hiện tại được phân chia thành. Nghĩa là, do cùng chế độ dự đoán nội được áp dụng cho các khối biến đổi phân vùng con, tập LFNST được

xác định dựa trên chế độ dự đoán nội cũng có thể được áp dụng ngang bằng cho tất cả các khối biến đổi phân vùng con. Ngoài ra, do chỉ số LFNST được báo hiệu trong mức đơn vị lập mã, cùng ma trận LFNST có thể được áp dụng cho các khối biến đổi phân vùng con mà khối hiện tại được phân chia thành.

Như được mô tả ở trên, tập biến đổi có thể được xác định theo chế độ dự đoán nội cho khối biến đổi cần được biến đổi, và LFNST ngược có thể được thực hiện dựa trên ma trận nhân biến đổi, nghĩa là, bất kỳ một trong các ma trận LFNST, được bao gồm trong tập biến đổi được chỉ báo bởi chỉ số LFNST. Ma trận được áp dụng cho LFNST ngược có thể được gọi là ma trận LFNST ngược hoặc ma trận LFNST, và được gọi bởi thuật ngữ bất kỳ miễn là ma trận là chuyển vị của ma trận được sử dụng cho LFNST xuôi.

Trong một ví dụ, ma trận LFNST ngược có thể là ma trận không vuông trong đó số lượng các cột là nhỏ hơn số lượng các hàng.

Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi được cải biến từ các hệ số biến đổi dựa trên chỉ số LFNST và ma trận LFNST cho LFNST, nghĩa là, bằng việc áp dụng LFNST (S1550).

LFNST là biến đổi không thể phân tách trong đó việc biến đổi được áp dụng cho các hệ số mà không phân tách các hệ số theo hướng cụ thể, không giống việc biến đổi sơ cấp để phân tách ngang hoặc dọc các hệ số cần được biến đổi và biến đổi chúng. Việc biến đổi không thể phân tách này có thể là biến đổi không thể phân tách tần số thấp để áp dụng việc biến đổi xuôi chỉ cho vùng tần số thấp thay vì toàn bộ diện tích của khối.

Thiết bị giải mã có thể dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên biến đổi ngược sơ cấp của hệ số biến đổi được cải biến (S1560).

Ở đây, do biến đổi ngược sơ cấp, biến đổi có thể phân tách thông thường có thể được sử dụng, hoặc MTS nêu trên có thể được sử dụng.

Sau đó, thiết bị giải mã 300 có thể tạo ra các mẫu được xây dựng lại dựa trên

các mẫu phần dư cho khôi hiện tại và các mẫu dự đoán cho khôi hiện tại (S1570).

Các hình vẽ sau đây được cung cấp để mô tả các ví dụ cụ thể của sáng chế. Vì các thuật ngữ cụ thể cho các thiết bị hoặc các thuật ngữ cụ thể cho các tín hiệu/các tín nhân/các trường cụ thể được minh họa trong các hình vẽ được cung cấp để minh họa, các dấu hiệu kỹ thuật của sáng chế không bị giới hạn ở các thuật ngữ cụ thể được sử dụng trong các hình vẽ sau.

Fig.16 là lưu đồ minh họa thao tác của thiết bị mã hóa video theo một phương án của sáng chế.

Mỗi quy trình được bộc lộ trên Fig.16 là dựa trên một số chi tiết được mô tả tham chiếu tới Fig.4 đến Fig.14. Do đó, phần mô tả của các chi tiết cụ thể trùng lặp với các chi tiết được mô tả tham chiếu tới Fig.2 và Fig.4 đến Fig.14 sẽ được bỏ qua hoặc sẽ được thực hiện sơ lược.

Thiết bị mã hóa 200 theo một phương án có thể dẫn xuất các mẫu dự đoán cho khôi hiện tại dựa trên chế độ dự đoán nội được áp dụng cho khôi hiện tại (S1610).

Khi ISP được áp dụng cho khôi hiện tại, thiết bị mã hóa có thể thực hiện việc dự đoán bởi mỗi khối biến đổi phân vùng con.

Thiết bị mã hóa có thể xác định xem có áp dụng việc lập mã ISP hoặc chế độ ISP cho khôi hiện tại hay không, nghĩa là, khối lập mã, và có thể xác định hướng trong đó khôi hiện tại được phân chia và có thể dẫn xuất kích thước và số lượng của các khối con được phân chia theo kết quả xác định.

Cùng chế độ dự đoán nội có thể được áp dụng cho các khối biến đổi phân vùng con mà khôi hiện tại được phân chia thành, và thiết bị mã hóa có thể dẫn xuất mẫu dự đoán cho mỗi khối biến đổi phân vùng con. Nghĩa là, thiết bị mã hóa thực hiện tuần tự dự đoán nội, ví dụ, ngang hoặc dọc, hoặc từ trái sang phải hoặc từ đỉnh đến đáy, theo dạng được phân chia của các khối biến đổi phân vùng con. Đối với khối con ngoài cùng bên trái hoặc trên cùng, điểm ảnh được xây dựng lại của khối lập mã đã được lập mã

được tham chiếu tới như trong chế độ dự đoán nội thông thường. Ngoài ra, đối với mỗi phía của khối biến đổi phân vùng con nội tuần tự, mà không liên kết với khối biến đổi phân vùng con trước đó, để dẫn xuất các điểm ảnh tham chiếu liên kết với phía, điểm ảnh được xây dựng lại của khối lập mã liên kết đã được lập mã được tham chiếu tới như trong chế độ dự đoán nội thông thường.

Thiết bị mã hóa 200 có thể dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán (S1620).

Thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi cho khối hiện tại bằng việc áp dụng ít nhất một trong số LFNST hoặc MTS cho các mẫu phần dư và có thể sắp xếp các hệ số biến đổi theo thứ tự quét được xác định trước.

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên quy trình biến đổi, chẳng hạn như việc biến đổi sơ cấp và/hoặc việc biến đổi thứ cấp, trên các mẫu phần dư, có thể áp dụng LFNST khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần độ chói, và có thể không áp dụng LFNST khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ.

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên việc biến đổi sơ cấp của các mẫu phần dư (S1630).

Việc biến đổi sơ cấp có thể được thực hiện qua nhiều các nhân biến đổi như trong MTS, trong trường hợp đó nhân biến đổi có thể được lựa chọn dựa trên chế độ dự đoán nội.

Thiết bị mã hóa 200 có thể thiết đặt biến áp dụng LFNST về việc liệu LFNST có được áp dụng hay không dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại (S1640).

Biến áp dụng LFNST có thể là biến cờ, ví dụ, `lnstNotTsFlag`, và `lnstNotTsFlag` có thể được dẫn xuất dựa trên loại cây của khối hiện tại và cờ lập mã hệ số biến đổi cho

mỗi thành phần của khối hiện tại hoặc cờ bỏ qua biến đổi cho mỗi thành phần của khối hiện tại. Biến áp dụng LFNST bằng 1 chỉ ra rằng LFNST là có thể áp dụng, và biến áp dụng LFNST bằng 0 chỉ ra rằng không có LFNST nào có thể áp dụng. Nghĩa là, khi biến áp dụng LFNST là 1, LFNST có thể được áp dụng, và khi biến áp dụng LFNST là 0, LFNST có thể không được áp dụng.

Cờ lập mã hệ số biến đổi dùng để chỉ CBF chỉ ra xem hệ số biến đổi cho khối biến đổi có tồn tại hay không, trong đó khi CBF bằng 1, mà chỉ ra rằng hệ số biến đổi cho khối biến đổi tồn tại, quy trình biến đổi có thể được thực hiện, và khi CBF bằng 0, không có quy trình biến đổi nào được thực hiện.

Theo một ví dụ, giá trị của biến áp dụng LFNST có thể được dẫn xuất là 1 dựa trên các cờ lập mã hệ số biến đổi cho tất cả các thành phần của khối hiện tại là 0. Trong trường hợp này, giá trị của biến áp dụng LFNST được thiết đặt là 1, nhưng khi CBF là 0, không có quy trình biến đổi thực nào được thực hiện, và biến `LfnstDcOnly` sẽ được mô tả sau không được cập nhật thành 0. Nghĩa là, khi các cờ lập mã hệ số biến đổi cho tất cả các thành phần của khối hiện tại là 0, không có LFNST nào được áp dụng.

Theo một ví dụ, giá trị của biến áp dụng LFNST có thể được thiết đặt là 1 dựa trên các cờ bỏ qua biến đổi cho tất cả các thành phần của khối hiện tại là 0. Nghĩa là, khi bỏ qua biến đổi không được áp dụng cho tất cả các thành phần của khối hiện tại, LFNST có thể được áp dụng.

Theo một ví dụ, khi loại cây của khối hiện tại là cây đơn, chế độ BDPCM có thể được áp dụng cho mỗi trong số thành phần độ chói và thành phần sắc độ. Nghĩa là, chế độ BDPCM có thể được áp dụng cho thành phần độ chói và có thể không được áp dụng cho thành phần sắc độ. Ngược lại, chế độ BDPCM có thể được áp dụng cho thành phần sắc độ và có thể không được áp dụng cho thành phần độ chói.

Theo tiêu chuẩn hiện tại, trong cây đơn, LFNST có thể được áp dụng chỉ cho thành phần độ chói mà không được áp dụng cho thành phần sắc độ. Ở đây, cho thành

phần độ chói, chế độ BDPCM không được áp dụng và hệ số biến đổi tồn tại, và do đó việc biến đổi có thể được thực hiện (nghĩa là, trong đó chế độ BDPCM không được áp dụng cho thành phần độ chói, CBF là khác 0, và cờ bỏ qua biến đổi là khác 0), và có trường hợp trong đó chế độ BDPCM là có thể áp dụng cho thành phần sắc độ. Khi chế độ BDPCM được áp dụng, cờ bỏ qua biến đổi được suy ra là 1.

Một cách thông thường, khi dẫn xuất biến áp dụng LFNST, nếu cờ bỏ qua biến đổi cho thành phần bất kỳ được gồm trong khối là 1, biến áp dụng LFNST được thiết đặt là 0, và do đó chỉ số LFNST không được phân tích cú pháp. Nghĩa là, mặc dù LFNST là có thể áp dụng cho thành phần độ chói, biến áp dụng LFNST được thiết đặt là 0 do thành phần sắc độ mà cho nó chế độ BDPCM được áp dụng (do cờ bỏ qua biến đổi được suy ra là 1 khi chế độ BDPCM được áp dụng).

Theo một ví dụ, khi chế độ BDPCM được áp dụng cho khối cụ thể, có thể là trường hợp trong đó CBF là 0, và do đó vấn đề nêu trên có thể được giải quyết sử dụng trường hợp này. Thiết bị mã hóa có thể áp dụng LFNST cho thành phần độ chói mặc dù chế độ BDPCM được áp dụng cho thành phần sắc độ bằng việc bổ sung điều kiện rằng CBF là 0 cho điều kiện cho việc dẫn xuất biến áp dụng LFNST.

Thiết bị mã hóa có thể thiết đặt điều kiện cho việc dẫn xuất biến áp dụng LFNST là "(treeType == DUAL_TREE_CHROMA || (!tu_y_coded_flag[x0][y0] || transform_skip_flag[x0][y0][0] == 0)) && (treeType == DUAL_TREE_LUMA || ((!tu_cb_coded_flag[x0][y0] || transform_skip_flag[x0][y0][1] == 0) && (!tu_cr_coded_flag[x0][y0] || transform_skip_flag[x0][y0][2] == 0)))".

Theo điều kiện, do loại cây của khối hiện tại là cây đơn ("treeType == DUAL_TREE_CHROMA" không được thỏa mãn, và do đó điều kiện "(!tu_y_coded_flag[x0][y0] || transform_skip_flag[x0][y0][0] == 0)" được kiểm tra) và cờ bỏ qua biến đổi cho thành phần độ chói là 0, điều kiện "transform_skip_flag[x0][y0][0] == 0" được thỏa mãn.

Ngoài ra, khi chế độ BDPCM được áp dụng cho thành phần sắc độ của khối hiện tại và cờ lập mã hệ số biến đổi là 0, điều kiện $((!tu_cb_coded_flag[x0][y0] \parallel transform_skip_flag[x0][y0][1] = 0) \&\& (!tu_cr_coded_flag[x0][y0] \parallel transform_skip_flag[x0][y0][2] = 0))$ được thỏa mãn. Nghĩa là, mặc dù chế độ BDPCM được áp dụng cho thành phần sắc độ và do đó cờ bỏ qua biến đổi cho thành phần sắc độ được suy ra là 1, giá trị của biến áp dụng LFNST có thể được thiết đặt là 1, mà chỉ ra rằng LFNST là có thể áp dụng, theo điều kiện cờ lập mã hệ số biến đổi.

Như được mô tả ở trên, bằng việc thay đổi điều kiện cho việc thiết đặt biến áp dụng LFNST, có thể giải quyết vấn đề rằng LFNST không được áp dụng cho tất cả các khối biến đổi bất kể giá trị CBF khi chế độ BDPCM được áp dụng cho khối biến đổi bất kỳ. Ngoài ra, khi kiểm tra điều kiện để xác định xem có phân tích cú pháp chỉ số LFNST hay không, cờ bỏ qua biến đổi và giá trị CBF được kiểm tra cùng nhau cho mỗi thành phần màu sắc. Do đó, khi giá trị CBF là 0 mặc dù bất kỳ trong số các thành phần màu sắc là trong chế độ BDPCM, việc phân tích cú pháp của chỉ số LFNST cho thành phần khác không bị ảnh hưởng.

Thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi được cải biến bằng việc áp dụng LFNST dựa trên giá trị của biến áp dụng LFNST (S1650).

LFNST là biến đổi không thể phân tách trong đó việc biến đổi được áp dụng cho các hệ số mà không phân tách các hệ số theo hướng cụ thể, không giống việc biến đổi sơ cấp để phân tách ngang hoặc dọc các hệ số cần được biến đổi và biến đổi chúng. Việc biến đổi không thể phân tách này có thể là biến đổi không thể phân tách tần số thấp để áp dụng biến đổi chỉ cho vùng tần số thấp thay vì toàn bộ khối mục tiêu cần được biến đổi.

Thiết bị mã hóa có thể áp dụng LFNST cho khối hiện tại dựa trên giá trị của biến áp dụng LFNST là 1, và có thể dẫn xuất các biến khác nhau ngoài biến áp dụng LFNST để áp dụng LFNST.

Ví dụ, thiết bị mã hóa có thể dẫn xuất biến thứ nhất (biến `LfnstDcOnly`) chỉ ra xem hệ số đáng kể có tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối hiện tại hay không và biến thứ hai (biến `LfnstZeroOutSigCoeffFlag`) chỉ ra xem hệ số biến đổi có tồn tại trong vùng thứ hai ngoài vùng thứ nhất trên cùng bên trái của khối hiện tại hay không. Ngoài ra, thiết bị mã hóa có thể xác định việc liệu có áp dụng LFNST hay không dựa trên loại cây và kích thước của khối hiện tại. Vùng thứ nhất trên cùng bên trái biểu thị vùng gồm tám vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC trong khối biến đổi 4×4 và khối biến đổi 8×8 , và biểu thị vùng gồm 16 vị trí từ vị trí DC theo thứ tự quét hệ số biến đổi trong tiêu chuẩn VVC trong khối biến đổi ngoài khối biến đổi 4×4 và khối biến đổi 8×8 .

Biến thứ nhất và biến thứ hai được thiết đặt ban đầu là 1, trong đó biến thứ nhất có thể được cập nhật là 0 khi hệ số đáng kể tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối hiện tại, và biến thứ hai có thể được cập nhật là 0 khi hệ số biến đổi tồn tại trong vùng thứ hai.

Khi biến thứ nhất được cập nhật thành 0 và biến thứ hai được duy trì là 1, LFNST có thể được áp dụng cho khối hiện tại.

Đối với thành phần độ chói mà cho nó chế độ phân vùng con nội (ISP) là có thể áp dụng, LFNST có thể được áp dụng mà không dẫn xuất biến `LfnstDcOnly`.

Cụ thể là, trong trường hợp trong đó chế độ ISP được áp dụng và cờ bỏ qua biến đổi, nghĩa là, `transform_skip_flag[x0][y0][0]`, cho thành phần độ chói là 0, khi loại cây của khối hiện tại là cây đơn hoặc cây đôi cho độ chói, LFNST có thể được áp dụng bất kể giá trị của biến `LfnstDcOnly`.

Tuy nhiên, cho thành phần sắc độ mà cho nó chế độ ISP không được áp dụng, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 theo giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cb, `transform_skip_flag[x0][y0][1]`, và giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cr, `transform_skip_flag[x0][y0][2]`. Nghĩa là,

khi giá trị của `cIdx` trong `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` là 1, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 1 ]` là 0, và khi giá trị của `cIdx` là 2, giá trị của biến `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 2 ]` là 0. Khi giá trị của biến `LfnstDcOnly` là 0, thiết bị mã hóa có thể áp dụng LFNST, và nếu không, thiết bị mã hóa có thể không áp dụng LFNST.

Biến thứ hai có thể là biến `LfnstZeroOutSigCoeffFlag` mà có thể chỉ ra rằng việc đặt bằng không được thực hiện khi LFNST được áp dụng. Biến thứ hai có thể ban đầu được thiết đặt là 1, và có thể được thay đổi thành 0 khi hệ số đáng kể tồn tại trong vùng thứ hai.

Biến `LfnstZeroOutSigCoeffFlag` có thể được dẫn xuất là 0 khi chỉ số của khối con trong đó hệ số khác không cuối cùng tồn tại là lớn hơn 0 và cả chiều rộng và chiều cao của khối biến đổi là bằng hoặc lớn hơn 4 hoặc khi vị trí của hệ số khác không cuối cùng trong khối con trong đó hệ số khác không cuối cùng tồn tại là lớn hơn 7 và kích thước của khối biến đổi là 4 x 4 hoặc 8 x 8. Khối con dùng để chỉ khối 4 x 4 được sử dụng làm đơn vị lập mã trong việc lập mã phần dư và có thể được gọi là nhóm hệ số (CG). Chỉ số khối con của 0 dùng để chỉ khối con 4 x 4 trên cùng bên trái.

Nghĩa là, khi hệ số khác không được dẫn xuất trong vùng ngoài vùng trên cùng bên trái trong đó hệ số biến đổi LFNST có thể tồn tại trong khối biến đổi hoặc hệ số khác không tồn tại ở vị trí ngoài vị trí thứ tám trong thứ tự quét cho khối 4 x 4 hoặc khối 8 x 8, biến `LfnstZeroOutSigCoeffFlag` được thiết đặt là 0.

Thiết bị mã hóa có thể xác định tập LFNST gồm các ma trận LFNST dựa trên chế độ dự đoán nội được dẫn xuất từ thông tin về chế độ dự đoán nội, và có thể lựa chọn bất kỳ một trong số nhiều ma trận LFNST.

Ở đây, cùng tập LFNST và cùng chỉ số LFNST có thể được áp dụng cho các

khối biến đổi phân vùng con mà khối hiện tại được phân chia thành. Nghĩa là, do cùng chế độ dự đoán nội được áp dụng cho các khối biến đổi phân vùng con, tập LFNST được xác định dựa trên chế độ dự đoán nội cũng có thể được áp dụng ngang bằng cho tất cả các khối biến đổi phân vùng con. Ngoài ra, do chỉ số LFNST được báo hiệu trong mức đơn vị lập mã, cùng ma trận LFNST có thể được áp dụng cho các khối biến đổi phân vùng con mà khối hiện tại được phân chia thành.

Như được mô tả ở trên, tập biến đổi có thể được xác định theo chế độ dự đoán nội cho khối biến đổi cần được biến đổi, và LFNST có thể được thực hiện dựa trên ma trận nhân biến đổi, nghĩa là, bất kỳ trong số các ma trận LFNST, được gồm trong tập biến đổi LFNST. Ma trận được áp dụng cho LFNST có thể được gọi là ma trận LFNST hoặc, và được gọi bởi thuật ngữ bất kỳ miễn là ma trận là chuyển vị của ma trận được sử dụng cho LFNST ngược.

Trong một ví dụ, ma trận LFNST có thể là ma trận không vuông trong đó số lượng của các hàng là ít hơn số lượng của các cột.

Khi chỉ số LFNST là lớn hơn 0 và loại cây của khối hiện tại là sắc độ cây đôi, LFNST có thể áp dụng được cho khối hiện tại, và do đó thiết bị mã hóa có thể mã hóa giá trị của cờ thành 1 sao cho danh sách định cỡ không được áp dụng cho thành phần sắc độ. Cờ có thể là cờ thuộc về tập phần tử cú pháp mức cao, chẳng hạn như tập thông số trình tự (sequence parameter set, SPS).

Theo một ví dụ, khi chỉ số LFNST là lớn hơn 0 và loại cây của khối hiện tại là độ chói cây đôi, LFNST có thể áp dụng được cho khối hiện tại, và do đó thiết bị mã hóa có thể mã hóa giá trị của cờ thành 1 sao cho danh sách định cỡ không được áp dụng cho thành phần độ chói.

Thiết bị mã hóa có thể dẫn xuất thông tin phần dư bằng việc thực hiện việc lượng tử hóa dựa trên các hệ số biến đổi được cải biến cho khối hiện tại và có thể mã hóa và xuất ra thông tin phần dư và chỉ số LFNST (S1660).

Thiết bị mã hóa có thể tạo ra thông tin phần dư gồm thông tin về các hệ số biến đổi được lượng tử hóa. Thông tin phần dư có thể gồm thông tin/phần tử cú pháp liên quan đến việc biến đổi nêu trên. Thiết bị mã hóa có thể mã hóa thông tin hình ảnh/video gồm thông tin phần dư, và có thể xuất ra thông tin hình ảnh/video dưới dạng luồng bit.

Cụ thể là, thiết bị mã hóa 200 có thể tạo ra thông tin về các hệ số biến đổi được lượng tử hóa và có thể mã hóa thông tin được lượng tử hóa về các hệ số biến đổi được tạo ra.

Phần tử cú pháp của chỉ số LFNST theo phương án này có thể chỉ ra bất kỳ trong số liệu LFNST (ngược) có được áp dụng hay không và bất kỳ một trong các ma trận LFNST được bao gồm trong tập LFNST, và khi tập LFNST bao gồm hai ma trận nhân biến đổi, phần tử cú pháp của chỉ số LFNST có thể có ba giá trị.

Theo một ví dụ, khi cấu trúc cây phân chia của khối hiện tại là loại cây đôi, chỉ số LFNST có thể được mã hóa cho mỗi trong số khối độ chói và khối sắc độ.

Theo một phương án, các giá trị của phần tử cú pháp của chỉ số biến đổi có thể bao gồm 0 chỉ ra rằng không có LFNST (ngược) nào được áp dụng cho khối hiện tại, 1 chỉ ra ma trận LFNST thứ nhất trong số các ma trận LFNST, và 2 chỉ ra ma trận LFNST thứ hai trong số các ma trận LFNST.

Trong sáng chế, ít nhất một hoạt động trong số việc lượng tử hóa/việc khử lượng tử hóa và/hoặc việc biến đổi/việc biến đổi ngược có thể được lược bỏ. Khi việc lượng tử hóa/việc khử lượng tử hóa được bỏ qua, hệ số biến đổi được lượng tử hóa có thể được gọi là hệ số biến đổi. Khi việc biến đổi/việc biến đổi ngược được bỏ qua, hệ số biến đổi có thể được gọi là hệ số hoặc hệ số phần dư, hoặc có thể vẫn được gọi là hệ số biến đổi để thống nhất trong diễn đạt.

Ngoài ra, trong sáng chế, hệ số biến đổi được lượng tử hóa và hệ số biến đổi có thể được gọi là hệ số biến đổi và hệ số biến đổi được định cỡ, một cách tương ứng. Trong trường hợp này, thông tin phần dư có thể gồm thông tin về (các) hệ số biến đổi,

và thông tin về (các) hệ số biến đổi có thể được báo hiệu thông qua cú pháp lập mã phần dư. Các hệ số biến đổi có thể được dẫn xuất dựa trên thông tin phần dư (hoặc thông tin về (các) hệ số biến đổi), và các hệ số biến đổi được định cỡ có thể được dẫn xuất thông qua việc biến đổi ngược (định cỡ) của các hệ số biến đổi. Các mẫu phần dư có thể được dẫn xuất dựa trên việc biến đổi ngược (biến đổi) của các hệ số biến đổi được định cỡ. Các chi tiết này có thể cũng được áp dụng/được thể hiện trong các phần khác của sáng chế.

Trong các phương án được mô tả ở trên, các phương pháp được giải thích trên cơ sở của các lưu đồ bởi chuỗi các bước hoặc các khối, nhưng sáng chế không bị giới hạn ở thứ tự của các bước, và bước nhất định có thể được thực hiện theo thứ tự hoặc bước khác với được mô tả ở trên, hoặc đồng thời với bước khác. Ngoài ra, có thể được hiểu bởi người có trình độ trung bình trong lĩnh vực rằng các bước được thể hiện trong lưu đồ không phải là bắt buộc, và bước khác có thể được kết hợp hoặc một hoặc nhiều bước của lưu đồ có thể được loại bỏ mà không ảnh hưởng đến phạm vi của sáng chế.

Các phương pháp được mô tả ở trên theo sáng chế có thể được thực hiện dưới dạng phần mềm, và thiết bị mã hóa và/hoặc thiết bị giải mã theo sáng chế có thể được gồm trong thiết bị xử lý hình ảnh, chẳng hạn như, TV, máy tính, điện thoại thông minh, hộp thu tín hiệu truyền hình, thiết bị hiển thị hoặc tương tự.

Khi các phương án trong sáng chế được triển khai bởi phần mềm, các phương pháp được mô tả ở trên có thể được triển khai dưới dạng các môđun (các quy trình, các chức năng hoặc tương tự) để thực hiện các chức năng được mô tả ở trên. Các môđun có thể được lưu trữ trong bộ nhớ và có thể được thực thi bởi bộ xử lý. Bộ nhớ có thể là phía trong hoặc phía ngoài bộ xử lý và có thể được kết nối với bộ xử lý theo các cách khác nhau đã biết. Bộ xử lý có thể gồm mạch tích hợp ứng dụng cụ thể (application-specific integrated circuit, ASIC), bộ chip khác, mạch logic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ có thể gồm bộ nhớ chỉ đọc (read-only memory, ROM), bộ nhớ truy cập ngẫu nhiên (random access memory, RAM), bộ nhớ tác động nhanh, thẻ nhớ, phương tiện lưu trữ,

và/hoặc thiết bị lưu trữ khác. Nghĩa là, các phương án được mô tả trong sáng chế có thể được triển khai và 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 trong mỗi hình vẽ có thể được triển khai và được thực hiện trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip.

Ngoài ra, thiết bị giải mã và thiết bị mã hóa mà cho nó sáng chế được áp dụng, có thể được gồm trong bộ thu phát phát rộng đa phương tiện, đầu cuối truyền thông di động, thiết bị video điện ảnh tại nhà, thiết bị video điện ảnh kỹ thuật số, camera giám sát, thiết bị trò chuyện video, thiết bị truyền thông thời gian thực chẳng hạn như truyền thông video, thiết bị phát luồng di động, phương tiện lưu trữ, máy quay video, 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 điện thoại video, và thiết bị video y tế, và có thể được sử dụng để xử lý tín hiệu video hoặc tín hiệu dữ liệu. Ví dụ, thiết bị cung cấp dịch vụ nội dung trên nền mạng viễn thông (OTT) có thể gồm máy chơi trò chơi, máy đọc phát Blu-ray, TV kết nối Internet, hệ thống rạp hát gia đình, điện thoại thông minh, PC máy tính bảng, đầu ghi video kỹ thuật số (digital video recorder, DVR) và tương tự.

Ngoài ra, phương pháp xử lý mà cho nó sáng chế được áp dụng, có thể được tạo ra dưới dạng chương trình được thực thi bởi máy tính, và được lưu trữ trong phương tiện ghi đọc được bởi máy tính. Dữ liệu đa phương tiện có cấu trúc dữ liệu theo sáng chế cũng có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính. Phương tiện ghi đọc được bởi máy tính gồm tất cả các loại thiết bị lưu trữ và các thiết bị lưu trữ được phân phối trong đó dữ liệu đọc được bởi máy tính được lưu trữ. Phương tiện ghi đọc được bởi máy tính có thể gồm, ví dụ, đĩa Blu-ray (Blu-ray Disc, BD), bus nối tiếp vạn năng (USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang. Ngoài ra, phương tiện ghi đọc được bởi máy tính gồm phương tiện được triển khai dưới dạng sóng mang (ví dụ, truyền qua Internet). Ngoài ra,

luồng bit được tạo ra bởi phương pháp mã hóa có thể được lưu trữ trong phương tiện ghi đọc được bởi máy tính hoặc được truyền thông qua mạng truyền thông có dây hoặc không dây. Ngoài ra, các phương án của sáng chế có thể được triển khai dưới dạng sản phẩm chương trình máy tính bởi các mã chương trình, và các mã chương trình có thể được thực thi trên máy tính bởi các phương án của sáng chế. Các mã chương trình có thể được lưu trữ trên vật mang đọc được bởi máy tính.

YÊU CẦU BẢO HỘ

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

nhận thông tin phần dư từ luồng bit;

dẫn xuất các hệ số biến đổi cho thành phần thứ nhất của khối hiện tại bằng việc thực hiện việc khử lượng tử hóa dựa trên thông tin phần dư, thành phần thứ nhất là đại diện của một trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại; và

dẫn xuất các hệ số biến đổi được cải biến bằng việc áp dụng biến đổi không thể phân tách tần số thấp (low-frequency non-separable transform, LFNST) cho các hệ số biến đổi,

trong đó việc dẫn xuất các hệ số biến đổi được cải biến bao gồm:

đáp ứng lại trường hợp trong đó loại cây của khối hiện tại là cây đơn, thiết đặt cờ áp dụng LFNST dựa trên cờ bỏ qua biến đổi cho thành phần thứ nhất của khối hiện tại, và cờ lập mã hệ số biến đổi cho thành phần thứ hai của khối hiện tại, thành phần thứ hai là đại diện của thành phần còn lại trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại; và

phân tích cú pháp, từ luồng bit, chỉ số LFNST dựa trên giá trị của cờ áp dụng LFNST;

trong đó giá trị của cờ áp dụng LFNST được thiết đặt bằng 1 dựa trên việc cờ bỏ qua biến đổi cho thành phần thứ nhất là bằng 0 và cờ lập mã hệ số biến đổi cho thành phần thứ hai là bằng 0, và

trong đó chỉ số LFNST được phân tích cú pháp từ luồng bit dựa trên việc giá trị của cờ áp dụng LFNST là bằng 1.

2. Phương pháp giải mã hình ảnh theo điểm 1, trong đó thành phần thứ hai của khối hiện tại được lập mã trong chế độ BDPCM trong khi thành phần thứ nhất của khối hiện tại không được lập mã trong chế độ BDPCM.

3. Phương pháp giải mã hình ảnh theo điểm 2, trong đó, cờ bỏ qua biến đổi cho thành phần thứ hai của khối hiện tại được suy ra là 1.

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

dẫn xuất các mẫu dự đoán cho thành phần thứ nhất của khối hiện tại, thành phần thứ nhất là đại diện của một trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại;

dẫn xuất các mẫu phần dư cho thành phần thứ nhất của khối hiện tại dựa trên các mẫu dự đoán;

dẫn xuất các hệ số biến đổi cho thành phần thứ nhất của khối hiện tại dựa trên việc biến đổi sơ cấp của các mẫu phần dư; và

dẫn xuất các hệ số biến đổi được cải biến từ các hệ số biến đổi bằng việc áp dụng biến đổi không thể phân tách tần số thấp (LFNST),

trong đó việc dẫn xuất các hệ số biến đổi được cải biến bao gồm:

đáp ứng lại trường hợp trong đó loại cây của khối hiện tại là cây đơn, thiết đặt cờ áp dụng LFNST dựa trên cờ bỏ qua biến đổi cho thành phần thứ nhất của khối hiện tại, và cờ lập mã hệ số biến đổi cho thành phần thứ hai của khối hiện tại, thành phần thứ hai là đại diện của thành phần còn lại trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại; và

áp dụng LFNST dựa trên giá trị của cờ áp dụng LFNST;

trong đó giá trị của cờ áp dụng LFNST được thiết đặt bằng 1 dựa trên việc cờ bỏ qua biến đổi cho thành phần thứ nhất là bằng 0 và cờ lập mã hệ số biến đổi cho thành phần thứ hai là bằng 0, và

trong đó LFNST được áp dụng dựa trên việc giá trị của cờ áp dụng LFNST là bằng 1.

5. Phương pháp mã hóa hình ảnh theo điểm 4, trong đó thành phần thứ hai của khối hiện tại được lập mã trong chế độ BDPCM trong khi thành phần thứ nhất của khối hiện tại không được lập mã trong chế độ BDPCM.

6. Phương pháp mã hóa hình ảnh theo điểm 5, trong đó, cờ bỏ qua biến đổi cho thành phần thứ hai của khối hiện tại được suy ra là 1.

7. Phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính để lưu trữ luồng bit được tạo ra bởi phương pháp mã hóa hình ảnh,

phương pháp mã hóa hình ảnh bao gồm các bước:

dẫn xuất các mẫu dự đoán cho thành phần thứ nhất của khối hiện tại, thành phần thứ nhất là đại diện của một trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại;

dẫn xuất các mẫu phần dư cho thành phần thứ nhất của khối hiện tại dựa trên các mẫu dự đoán;

dẫn xuất các hệ số biến đổi cho thành phần thứ nhất của khối hiện tại dựa trên việc biến đổi sơ cấp của các mẫu phần dư; và

dẫn xuất các hệ số biến đổi được cải biến từ các hệ số biến đổi bằng việc áp dụng biến đổi không thể phân tách tần số thấp (LFNST),

trong đó việc dẫn xuất các hệ số biến đổi được cải biến bao gồm:

đáp ứng lại trường hợp trong đó loại cây của khối hiện tại là cây đơn, thiết đặt cờ áp dụng LFNST dựa trên cờ bỏ qua biến đổi cho thành phần thứ nhất của

khối hiện tại, và cờ lập mã hệ số biến đổi cho thành phần thứ hai của khối hiện tại, thành phần thứ hai là đại diện của thành phần còn lại trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại; và

áp dụng LFNST dựa trên giá trị của cờ áp dụng LFNST;

trong đó giá trị của cờ áp dụng LFNST được thiết đặt bằng 1 dựa trên việc cờ bỏ qua biến đổi cho thành phần thứ nhất là bằng 0 và cờ lập mã hệ số biến đổi cho thành phần thứ hai là bằng 0, và

trong đó LFNST được áp dụng dựa trên việc giá trị của cờ áp dụng LFNST là bằng 1.

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

dẫn xuất các mẫu dự đoán cho thành phần thứ nhất của khối hiện tại, thành phần thứ nhất là đại diện của một trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại;

dẫn xuất các mẫu phần dư cho thành phần thứ nhất của khối hiện tại dựa trên các mẫu dự đoán;

dẫn xuất các hệ số biến đổi cho thành phần thứ nhất của khối hiện tại dựa trên việc biến đổi sơ cấp của các mẫu phần dư;

dẫn xuất các hệ số biến đổi được cải biến từ các hệ số biến đổi bằng việc áp dụng biến đổi không thể phân tách tần số thấp (LFNST);

dẫn xuất các hệ số biến đổi được lượng tử hóa bằng việc thực hiện việc lượng tử hóa trên các hệ số biến đổi được cải biến;

mã hóa thông tin phần dư cho các hệ số biến đổi được lượng tử hóa để tạo ra luồng bit; và

truyền dữ liệu bao gồm luồng bit,

trong đó việc dẫn xuất các hệ số biến đổi được cải biến bao gồm:

đáp ứng lại trường hợp trong đó loại cây của khối hiện tại là cây đơn, thiết đặt cờ áp dụng LFNST dựa trên cờ bỏ qua biến đổi cho thành phần thứ nhất của khối hiện tại, và cờ lập mã hệ số biến đổi cho thành phần thứ hai của khối hiện tại, thành phần thứ hai là đại diện của thành phần còn lại trong số thành phần độ chói hoặc thành phần sắc độ của khối hiện tại; và

áp dụng LFNST dựa trên giá trị của cờ áp dụng LFNST;

trong đó giá trị của cờ áp dụng LFNST được thiết đặt bằng 1 dựa trên việc cờ bỏ qua biến đổi cho thành phần thứ nhất là bằng 0 và cờ lập mã hệ số biến đổi cho thành phần thứ hai là bằng 0, và

trong đó LFNST được áp dụng dựa trên việc giá trị của cờ áp dụng LFNST là bằng

1.

FIG. 1

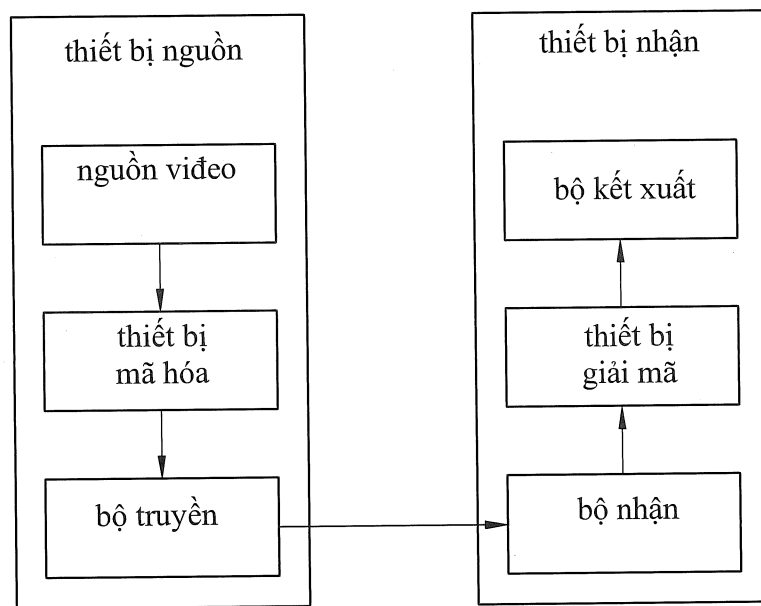


FIG. 3

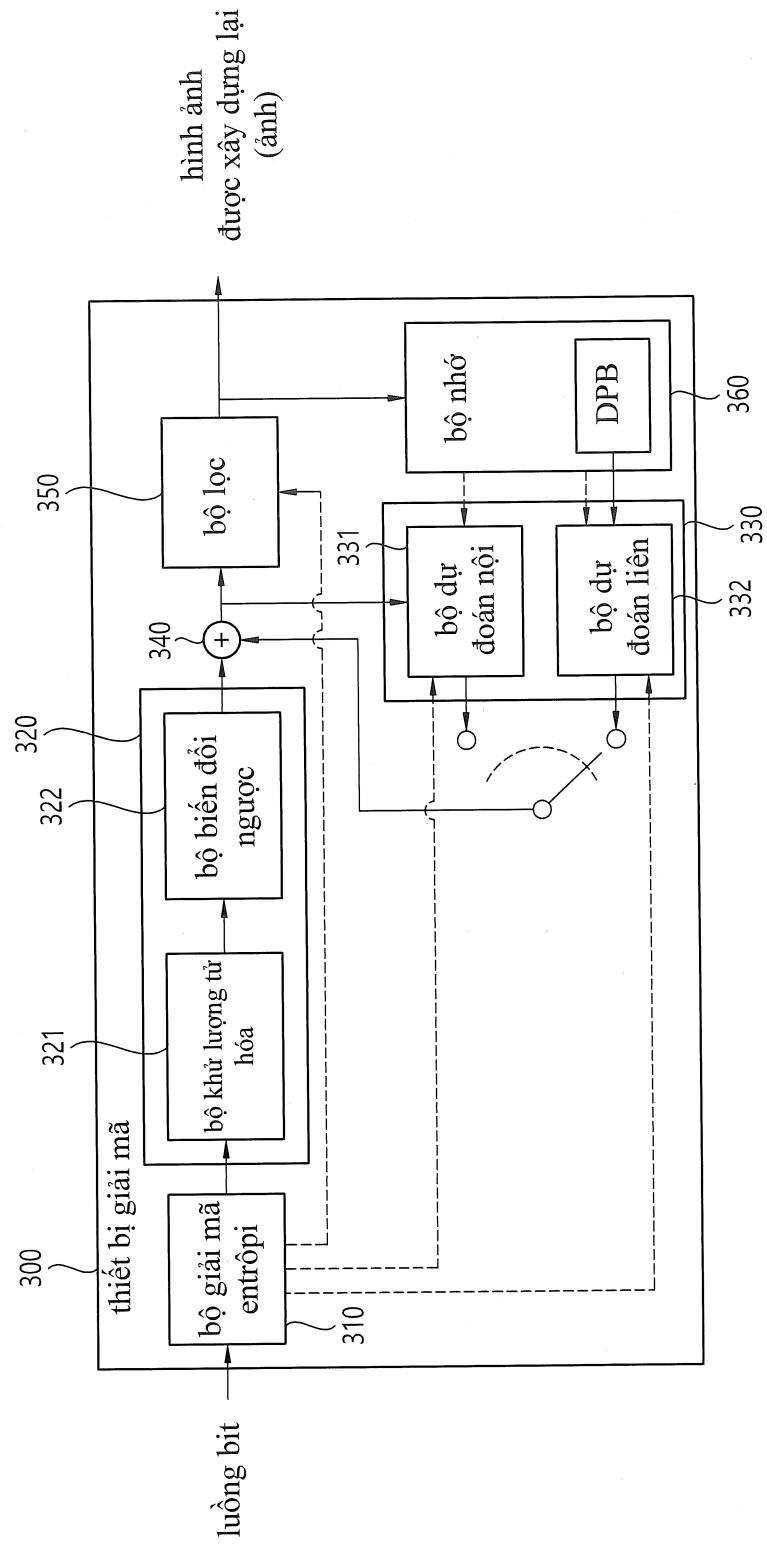


FIG. 4

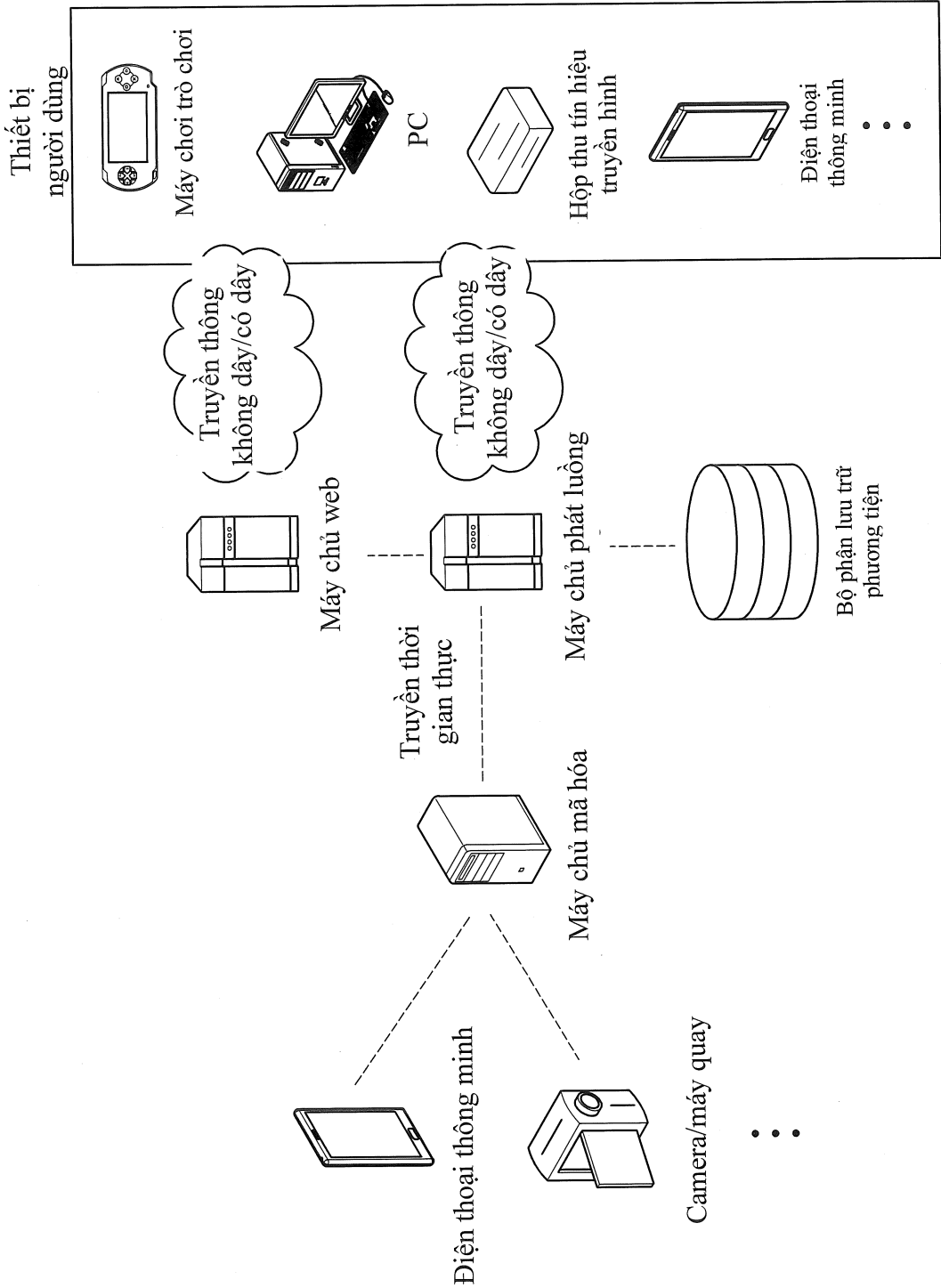


FIG. 5

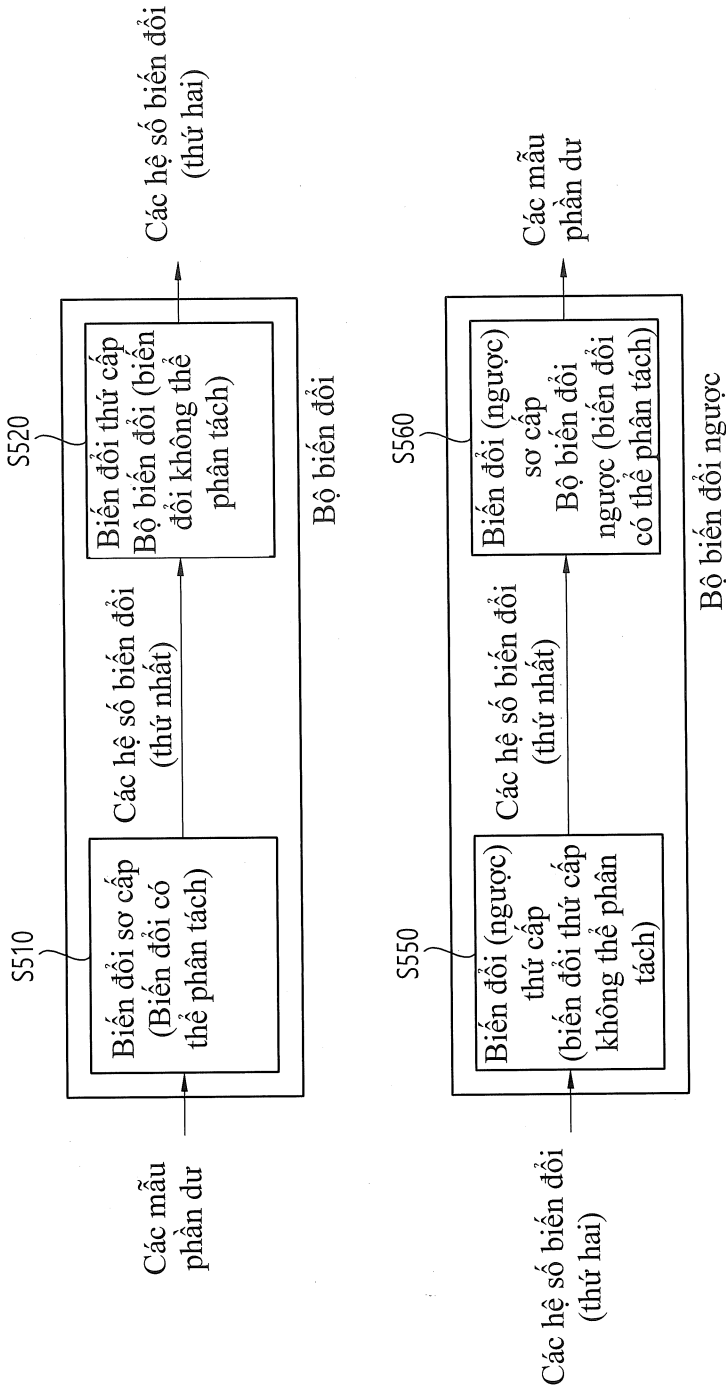


FIG. 6

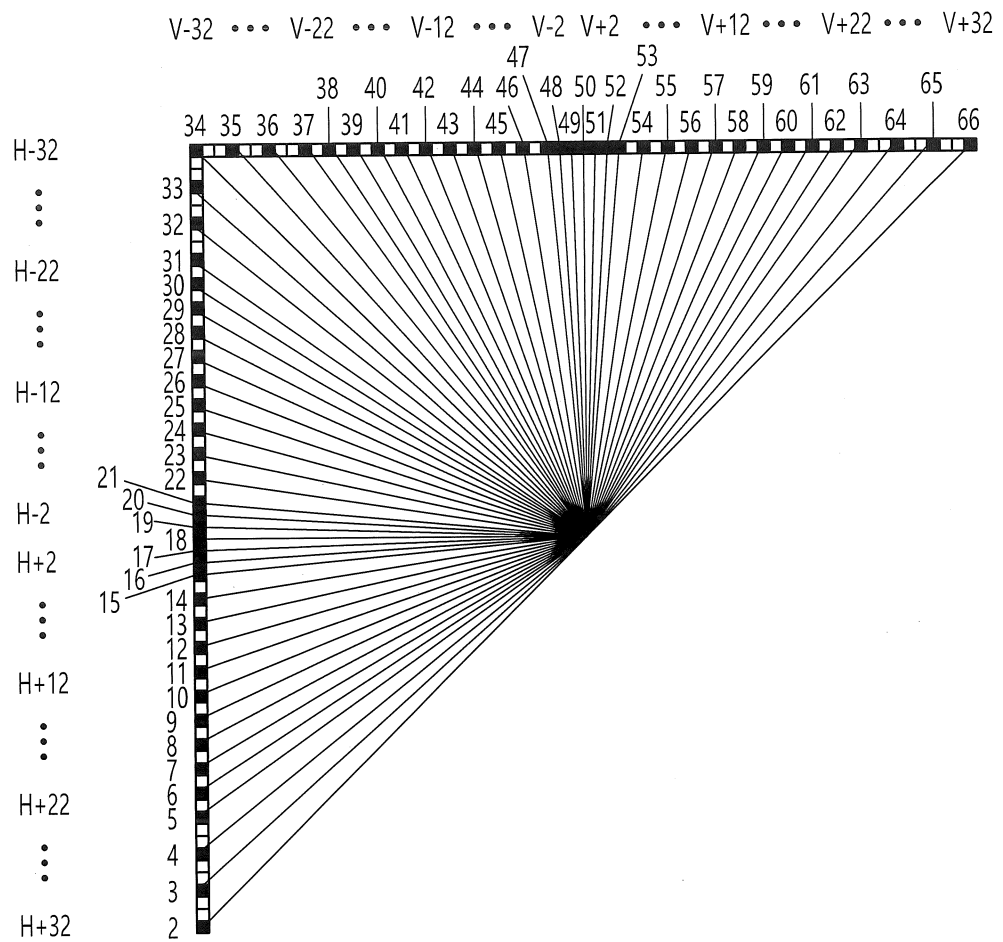


FIG. 7

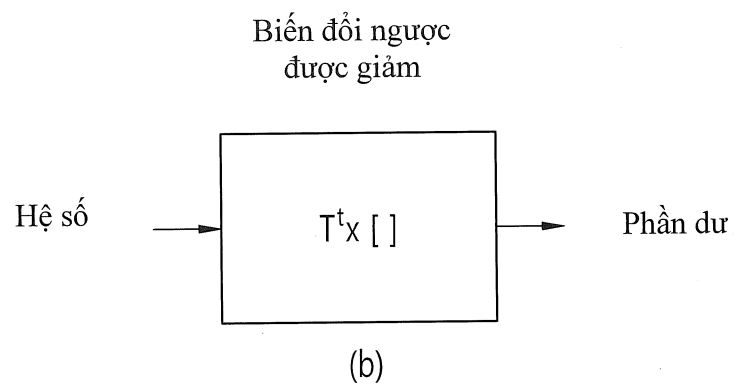
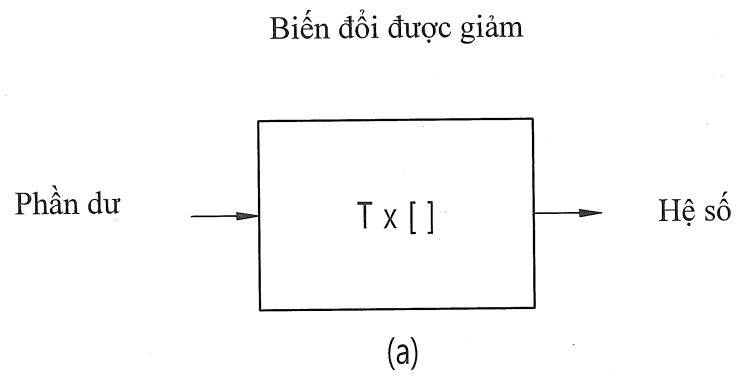


FIG. 8

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36				
37	38	39	40				
41	42	43	44				
45	46	47	48				

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(a)

1	9	17	25	33	37	41	45
2	10	18	26	34	38	42	46
3	11	19	27	35	39	43	47
4	12	20	28	36	40	44	48
5	13	21	29				
6	14	22	30				
7	15	23	31				
8	16	24	32				

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

(b)

FIG. 9

1	3	6	10
2	5	9	13
4	8	12	15
7	11	14	16

(a)

1	3	6	
2	5		
4	8		
7			

(b)

FIG. 10

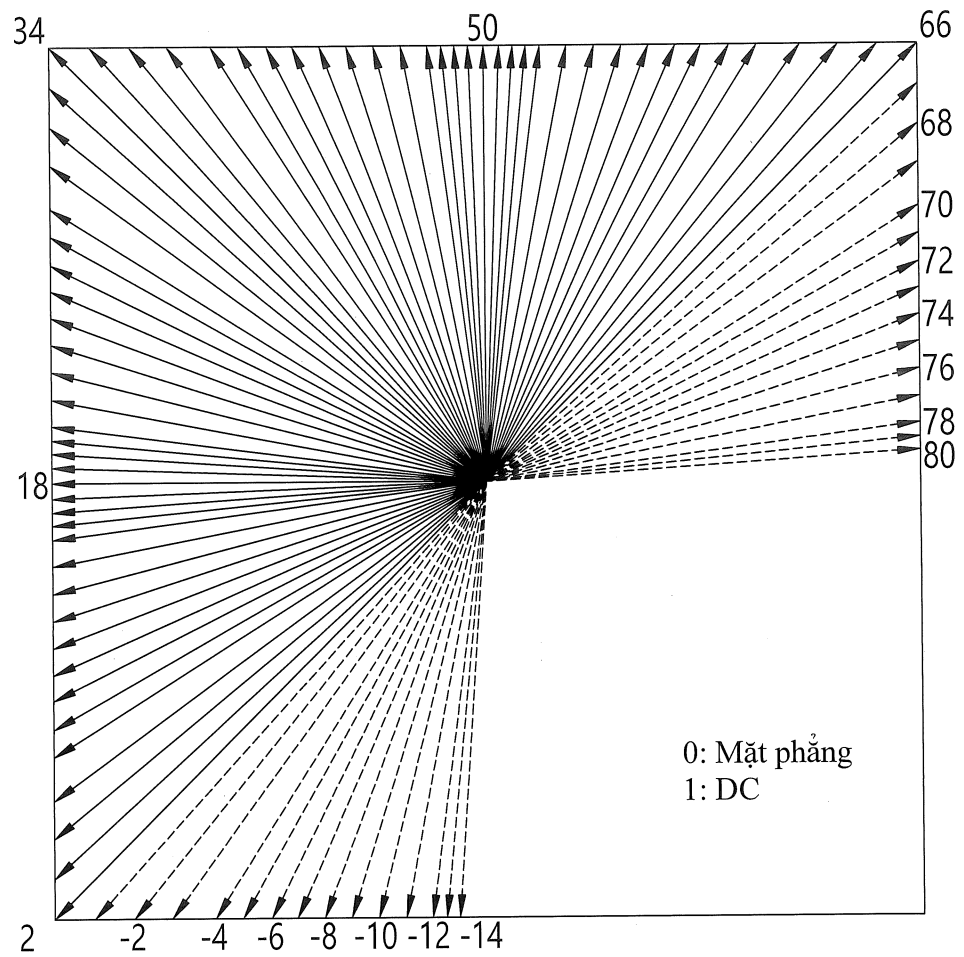
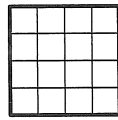
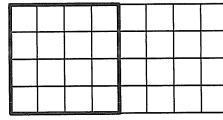


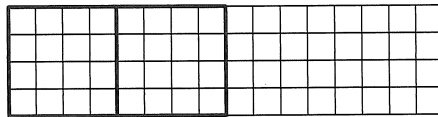
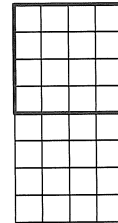
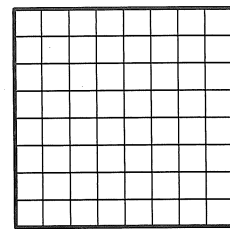
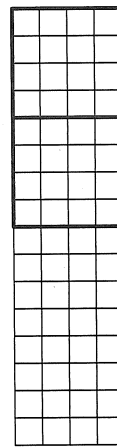
FIG. 11



(a) 4x4



(b) 8x4 / 4x8

(c) 4xN / Nx4, khi $N \geq 16$ 

(d) 8x8

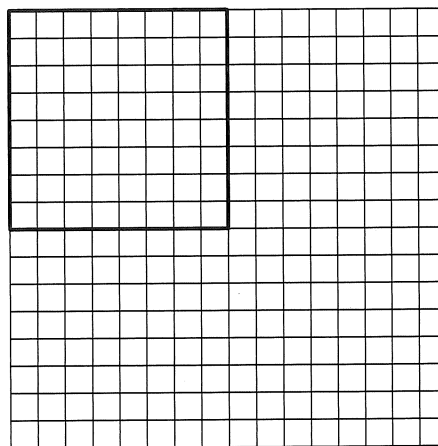
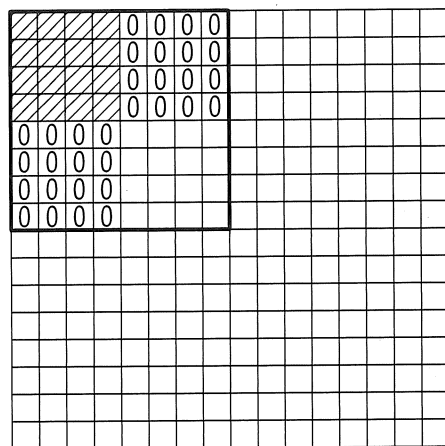
(e) $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

Figure 1 shows four schematic diagrams of the basic building blocks of the 2D lattice:

- (a) 4×4 block: A 4×4 grid with a 2×2 sub-block of zeros in the top-right corner.
- (b) $8 \times 4 / 4 \times 8$ block: An 8×4 grid with a 4×4 sub-block of zeros in the top-right corner.
- (c) $4 \times N / N \times 4$ block: A $4 \times N$ grid with a 4×4 sub-block of zeros in the top-right corner.
- (d) 8×8 block: An 8×8 grid with a 4×4 sub-block of zeros in the top-right corner.



(e) $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

FIG. 14

			0	0	0	0
		0	0	0	0	0
		0	0	0	0	0
	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(a) 8x8

[illegible]

(b) $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

FIG. 15

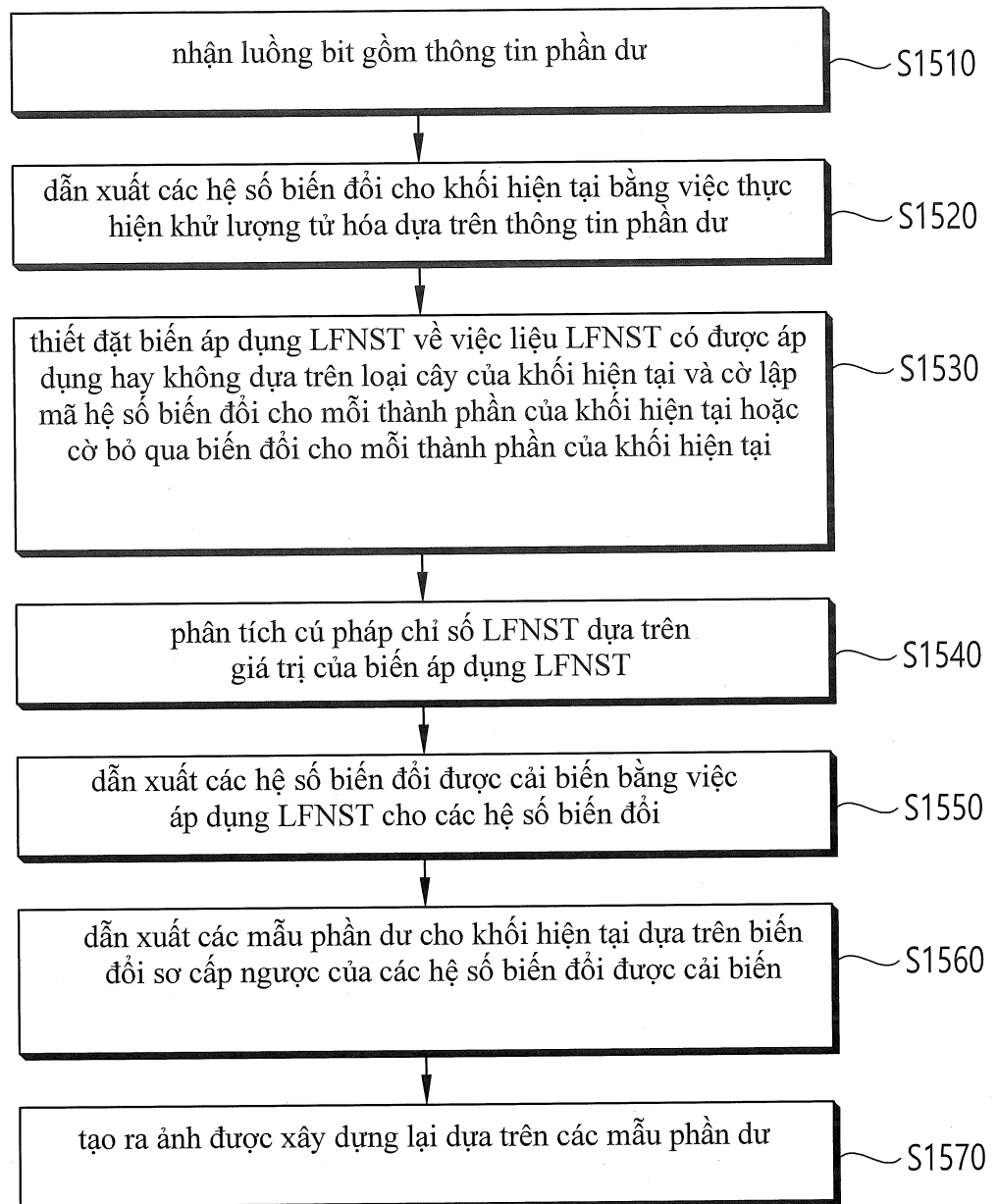


FIG. 16

