



(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)



1-0051782

(51)^{2021.01} H04N 19/60; H04N 19/129; H04N
19/132; H04N 19/70; H04N 19/18;
H04N 19/122; H04N 19/176

(13) B

(21) 1-2022-03469

(22) 09/11/2020

(86) PCT/KR2020/015587 09/11/2020

(87) WO2021/096172 20/05/2021

(30) 62/933,972 11/11/2019 US; 62/933,951 11/11/2019 US

(45) 25/09/2025 450

(43) 25/08/2022 413A

(73) LG ELECTRONICS INC. (KR)

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

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

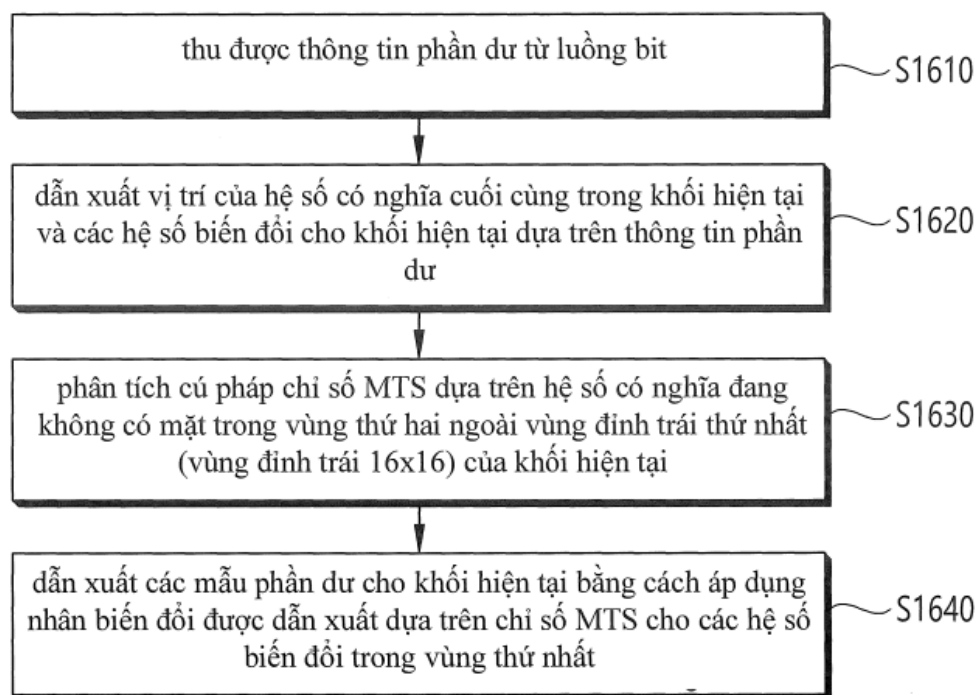
(74) Công ty Luật TNHH T&G (TGVN)

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

(21) 1-2022-03469

(57) Sáng chế đề cập tới phương pháp giải mã hình ảnh bao gồm bước áp dụng việc biến đổi sơ cấp ngược vào hệ số biến đổi để dẫn xuất các mẫu phần dư cho khối hiện tại, trong đó: việc biến đổi sơ cấp ngược được thực hiện trên cơ sở chỉ số chọn nhiều biến đổi (MTS) nhận được từ luồng bit; chỉ số MTS được phân tích cú pháp trên cơ sở hệ số hiệu quả hiện không tồn tại trong khu vực thứ hai ngoại trừ khu vực thứ nhất trên đỉnh bên trái của khối hiện tại; và khu vực thứ nhất là khu vực 16X16 trên đỉnh bên trái của khối hiện tại. Sáng chế này còn đề cập tới phương pháp mã hóa, phương pháp truyền dữ liệu cho thông tin hình ảnh, và phương tiện lưu trữ kỹ thuật số được bởi máy tính.

FIG. 16



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến kỹ thuật lập mã hình ảnh và, cụ thể hơn là đến phương pháp và thiết bị để lập mã hình ảnh dựa trên việc biến đổi trong hệ thống lập mã hình ảnh.

Tình trạng kỹ thuật của sáng chế

Hiện nay, nhu cầu về các hình ảnh/video độ phân giải cao, chất lượng cao như là hình ảnh/video 4K, 8K hoặc độ phân giải siêu cao (Ultra High Definition, UHD) khác đang gia tăng trong nhiều lĩnh vực khác nhau. Vì dữ liệu hình ảnh/video trở nên có độ phân giải cao hơn và chất lượng cao hơn, nên lượng thông tin hoặc lượng bit được truyền sẽ tăng lên khi so với dữ liệu hình ảnh thông thường. Do đó, khi dữ liệu ảnh được truyền bằng cách sử dụng các 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, thì chi phí truyền và chi phí lưu trữ sẽ tăng lên.

Hơn nữa, hiện nay, sự quan tâm và nhu cầu đối với truyền thông nhập vai như là thực tế ảo (Virtual Reality, VR), nội dung thực tế nhân tạo (Artificial Reality, AR) hoặc ảnh ba chiều, hoặc dạng tương tự đang tăng lên, và việc truyền quảng bá cho các hình ảnh/các video có đặc điểm hình ảnh khác với các hình ảnh thực, như là hình ảnh trong trò chơi đang tăng lên.

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

Bản chất kỹ thuật của sáng chế

Khía cạnh kỹ thuật của sáng chế này là để đề xuất phương pháp và thiết bị để tăng hiệu suất lập mã hình ảnh.

Khía cạnh kỹ thuật khác của sáng chế này là để đề xuất phương pháp và thiết bị để tăng hiệu suất trong việc lập mã chỉ số biến đổi.

Khía cạnh kỹ thuật khác nữa của sáng chế này là để đề xuất phương pháp lập mã hình ảnh và thiết bị lập mã hình ảnh sử dụng MTS.

Khía cạnh kỹ thuật khác nữa của sáng chế này là để đề xuất phương pháp lập mã hình ảnh và thiết bị lập mã hình ảnh cho việc báo hiệu của chỉ số MTS.

Theo phương án thực hiện của sáng chế này, sáng chế đề xuất phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã. Phương pháp có thể bao gồm bước dẫn xuất các mẫu phần dư cho khối hiện tại bằng cách áp dụng việc biến đổi sơ cấp ngược vào các hệ số biến đổi, trong đó việc biến đổi sơ cấp ngược được thực hiện dựa trên chỉ số MTS nhận được từ luồng bit, trong đó chỉ số MTS được phân tích cú pháp dựa trên hệ số có nghĩa đang không ở trong vùng thứ hai khác với vùng thứ nhất trên đỉnh bên trái của khối hiện tại, và trong đó vùng thứ nhất là vùng 16 x 16 trên đỉnh bên trái của khối hiện tại.

Chỉ số MTS được phân tích cú pháp dựa trên thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không, và trong đó thông tin cờ được dẫn xuất bằng cách xác định xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không bởi đơn vị của khối con quét của việc quét hệ số có nghĩa.

Khi trị số của cờ, đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong khối con quét hay không, là 1 và hệ số có nghĩa đang tồn tại trong khối con quét được định vị trong vùng thứ hai, thì thông tin cờ chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Khi trị số của cờ, đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong khối con quét hay không, là 1 và khối con quét được định vị trong vùng thứ hai, thì thông tin cờ chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Thông tin cờ ban đầu được đặt là 1, và khi hệ số có nghĩa được định vị trong vùng thứ hai, thì thông tin cờ được thay đổi thành 0.

Khối con quét là khối 4×4 , và khối con quét được quét theo hướng quét đường chéo ngược từ vị trí của hệ số có nghĩa cuối cùng trong khối hiện tại.

Theo phương án thực hiện khác của sáng chế này, sáng chế đề xuất phương pháp mã hóa được thực hiện bởi thiết bị mã hóa. Phương pháp có thể bao gồm bước dẫn xuất các hệ số biến đổi cho khối hiện tại bằng cách áp dụng MTS vào các mẫu phần dư; đặt bằng không vùng thứ hai khác với vùng thứ nhất trên đỉnh bên trái của khối hiện tại; dẫn xuất thông tin phần dư dựa trên hệ số biến đổi của vùng thứ nhất; và xây dựng thông tin hình ảnh để phân tích cú pháp chỉ số MTS đang chỉ thị nhân biến đổi MTS dựa trên hệ số có nghĩa tồn tại trong vùng thứ hai; trong đó vùng thứ nhất là vùng 16×16 trên đỉnh bên trái của khối hiện tại.

Theo phương án thực hiện khác nữa của sáng chế này, sáng chế đề xuất phương tiện lưu trữ kỹ thuật số lưu trữ dữ liệu hình ảnh chứa thông tin hình ảnh đã được mã hóa và luồng bit được sinh ra theo phương pháp mã hóa được thực hiện bởi thiết bị mã hóa.

Theo phương án thực hiện khác nữa của sáng chế này, sáng chế đề xuất phương tiện lưu trữ kỹ thuật số lưu trữ dữ liệu hình ảnh chứa thông tin hình ảnh đã được mã hóa và luồng bit để làm cho thiết bị giải mã thực hiện phương pháp giải mã hình ảnh.

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

Theo sáng chế này, có thể tăng hiệu suất trong việc lập mã chỉ số MTS.

Theo sáng chế này, có thể tăng hiệu suất của phương pháp lập mã hình ảnh sử dụng MTS.

Các hiệu quả có thể thu được qua các ví dụ cụ thể của sáng chế này không bị hạn chế vào các hiệu quả đã được liệt kê ở trên. Ví dụ, có thể có các hiệu quả kỹ thuật khác nhau mà người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng có thể hiểu hoặc dẫn xuất từ sáng chế này. Theo đó, các hiệu quả cụ thể của sáng chế này không bị giới hạn ở những hiệu quả được mô tả một cách rõ ràng trong sáng chế và có thể bao gồm các hiệu quả khác nhau vốn 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 một cách sơ lược ví dụ của hệ thống lập mã video/hình ảnh mà sáng chế này có thể áp dụng được vào đó.

Fig.2 là giản đồ minh họa một cách sơ lược cấu hình của thiết bị mã hóa video/hình ảnh mà sáng chế này có thể áp dụng được vào đó.

Fig.3 là giản đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh mà sáng chế này có thể áp dụng được vào đó.

Fig.4 là minh họa một cách sơ lược mẫu đa biến đổi theo phương án thực hiện của tài liệu này.

Fig.5 thể hiện theo cách làm ví dụ các chế độ theo hướng nội của 65 hướng dự đoán.

Fig.6 là giản đồ để giải thích RST theo phương án thực hiện của sáng chế này.

Fig.7 là giản đồ minh họa trình tự của việc sắp xếp dữ liệu đầu ra của việc biến đổi sơ cấp xuôi thành vectơ một chiều theo một ví dụ.

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

Fig.9 là giản đồ minh họa các chế độ nội dự đoán góc rộng theo phương án thực hiện của tài liệu này.

Fig.10 là giản đồ minh họa hình dạng khối mà LFNST được áp dụng vào đó.

Fig.11 là giản đồ minh họa việc sắp xếp của dữ liệu đầu ra của LFNST xuôi theo một ví dụ.

Fig.12 là giản đồ minh họa rằng số của dữ liệu đầu ra cho LFNST xuôi bị hạn chế vào số tối đa là 16 theo một ví dụ.

Fig.13 là giản đồ minh họa việc đặt bằng không trong khối mà LFNST 4x4 được áp dụng vào đó, theo một ví dụ.

Fig.14 là giản đồ minh họa việc đặt bằng không trong khối mà LFNST 8x8 được áp dụng vào đó, theo một ví dụ.

Fig.15 minh họa việc quét của khối biến đổi 32 x 32 theo phương án thực hiện của sáng chế này.

Fig.16 là lưu đồ để giải thích phương pháp giải mã hình ảnh theo một ví dụ.

Fig.17 là lưu đồ để giải thích phương pháp mã hóa theo một ví dụ.

Fig.18 minh họa cấu trúc của hệ thống tạo luồng nội dung mà sáng chế này được áp dụng vào đó.

Mô tả chi tiết sáng chế

Mặc dù sáng chế có thể dễ có các sự thay đổi khác nhau và chứa các phương án thực hiện khác nhau, nhưng các phương án thực hiện cụ thể của nó đã được thể hiện trên các hình vẽ theo cách làm ví dụ và sẽ được mô tả chi tiết. Tuy nhiên, nó không nhằm giới hạn sáng chế ở các phương án thực hiện cụ thể được bộc lộ ở đây. Thuật ngữ được sử dụng ở đây chỉ nhằm mục đích mô tả các phương án thực hiện cụ thể và không nhằm giới hạn ý tưởng kỹ thuật của sáng chế này. Các dạng số ít có thể bao

gồm cả các dạng số nhiều, trừ khi ngữ cảnh nêu rõ ngược lại. Thuật ngữ như “chứa” và “có” nhằm để chỉ ra rằng các dấu hiệu, các số, các bước, các thao tác, các phần tử, các thành phần, hoặc các tổ hợp của chúng được sử dụng phân mô tả sau đây sẽ tồn tại và sẽ được hiểu là có khả năng tồn tại hoặc việc thêm vào một hoặc nhiều dấu hiệu, các số, các bước, các thao tác, các phần tử, các thành phần, hoặc các tổ hợp của chúng là không 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 tạo thuận tiện cho việc mô tả để phân biệt các chức năng khác nhau, và, tuy nhiên, nó không có nghĩa là từng thành phần được hiện thực hóa bởi phần cứng hoặc phần mềm tách rời. Ví dụ, hai hoặc hơn hai thành phần bất kỳ trong số các thành phần 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 thực hiện mà trong đó các thành phần được tổ hợp và/hoặc được phân chia sẽ thuộc vào phạm vi bảo hộ của sáng chế theo sáng chế này miễn là chúng không tách khỏi nguyên lý của sáng chế.

Ở dưới đây, các phương án thực hiện được ưu tiên của sáng chế sẽ được mô tả chi tiết hơn vẫn đề cập tới các hình vẽ kèm theo. Thêm vào đó, các sự ký hiệu tham chiếu giống nhau được sử dụng cho các chi tiết giống nhau trong các hình vẽ, và các phần mô tả lặp lại của các chi tiết giống nhau này sẽ được bỏ qua.

Tài liệu này liên quan tới việc lập mã video/hình ảnh. Ví dụ, phương pháp/ví dụ được bộc lộ trong tài liệu này có thể liên quan tới 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 tới việc tạo mã video khác (ví dụ, tiêu chuẩn tạo mã video hiệu suất 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 tài liệu này, nhiều phương án thực hiện liên quan đến việc lập mã video/hình ảnh có thể được đề xuất, và, trừ khi được chỉ rõ theo cách ngược lại, còn không thì các phương án thực hiện có thể được tổ hợp với nhau và được thực hiện.

Trong tài liệu này, video có thể có nghĩa là tập hợp của các chuỗi hình ảnh theo thời gian. Nói chung, ảnh có nghĩa là đơn vị biểu diễn hình ảnh tại vùng thời gian cụ thể, lát/phiến là đơn vị cấu thành một phần của ảnh. Lát/phiến có thể bao gồm một hoặc nhiều đơn vị cây lập mã (Coding tree unit, CTU). Một ảnh có thể được tạo nên bởi một hoặc nhiều lát/phiến. Một ảnh có thể được tạo nên bởi một hoặc nhiều nhóm phiến. Một nhóm phiến có thể bao gồm một hoặc nhiều phiến.

Điểm ảnh (pixel hoặc pel) có thể có nghĩa là đơn vị nhỏ nhất cấu thành nên một ảnh (hoặc hình ảnh). Cũng vậy, “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 trị số của điểm ảnh, và có thể biểu diễn chỉ điểm ảnh/trị số điểm ảnh của thành phần độ chói hoặc chỉ điểm ảnh/trị số điểm ảnh của thành phần sắc độ. Theo cách khác, mẫu có thể đề cập đến trị số điểm ảnh trong miền không gian, hoặc khi trị số điểm ảnh này được chuyển đổi thành miền tần số, thì nó có thể đề cập đến 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 này có thể gồm ít nhất một vùng trong số vùng cụ thể và thông tin liên quan đến vùng này. Một đơn vị có thể gồm một khối độ chói và hai khối sắc độ (ví dụ, cb, cr). Đơn vị và thuật ngữ như là khối, khu vực, hoặc dạng tương tự có thể được sử dụng thay thế cho nhau tùy theo hoàn cảnh. Trong trường hợp chung, khối $M \times N$ có thể gồm tập hợp (hoặc mảng) các mẫu (hoặc các mảng mẫu) hoặc hệ số biến đổi tạo nên M cột và N hàng.

Trong tài liệu này, thuật ngữ “/” và “,” nên được diễn đạt là để chỉ “và/hoặc”. Ví dụ, biểu diễn “A/B” có thể nghĩa là “A và/hoặc B”. Thêm nữa, “A, B” có thể có nghĩa là “A và/hoặc B”. Hơn nữa, “A/B/C” có thể có nghĩa là “ít nhất một hành phần

trong số A, B, và/hoặc C”. Cũng vậy, “A/B/C” có thể có nghĩa là “ít nhất một thành phần trong số A, B, và/hoặc C”.

Hơn nữa, trong tài liệu này, thuật ngữ “hoặc” nên được diễn dịch là để chỉ thị “và/hoặc”. Ví dụ, cách thể hiện “A hoặc B” có thể bao gồm 1) chỉ có A, 2) chỉ có B, và/hoặc 3) cả hai A và B. Nói cách khác, thuật ngữ “hoặc” trong tài liệu này nên được diễn dịch là để chỉ ra “theo cách bổ sung vào hoặc theo cách thay thế”.

Trong sáng chế này, “ít nhất một thành phần trong số A và B” có thể có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Thêm vào đó, theo sáng chế này, các thể hiện “ít nhất một thành phần trong số A hoặc B” hoặc “ít nhất một thành phần trong số A và/hoặc B” có thể được diễn dịch như là “ít nhất một trong số A và B”.

Thêm vào đó, theo sáng chế này, “ít nhất một thành phần trong số A, B và C” có thể có nghĩa là “chỉ A”, “chỉ B”, “chỉ C”, hoặc “tổ hợp bất kỳ của A, B và C”. Thêm vào đó, “ít nhất một thành phần trong số A, B hoặc C” hoặc “ít nhất một thành phần trong số A, B và/hoặc C” có thể có nghĩa là “ít nhất một thành phần trong số A, B và C”.

Thêm vào đó, dấu ngoặc đơn được sử dụng theo sáng chế này có thể có nghĩa là “ví dụ”. Cụ thể là, khi được chỉ ra như là “việc dự đoán (việc nội dự đoán)”, thì nó có thể có nghĩa là “việc nội dự đoán” như được đề xuất là ví dụ của “việc dự đoán”. Nói cách khác, “việc dự đoán” của sáng chế này không bị hạn chế vào “việc nội dự đoán”, và “việc nội dự đoán” có thể được đề xuất như là ví dụ của “việc dự đoán”. Thêm vào đó, khi được chỉ thị như là “việc dự đoán (tức là, việc nội dự đoán)”, nó cũng có thể có nghĩa là “việc nội dự đoán” như được đề xuất như là ví dụ của “việc dự đoán”.

Các dấu hiệu kỹ thuật được mô tả trong một hình vẽ theo sáng chế này có thể được áp dụng một cách riêng rẽ hoặc có thể được áp dụng một cách đồng thời.

Fig.1 minh họa một cách sơ lược ví dụ của hệ thống lập mã video/hình ảnh mà sáng chế này có thể áp dụng được vào đó.

Đề cập tới Fig.1, hệ thống lập mã video/hình ảnh có thể bao gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị nhận). Thiết bị nguồn có thể phân phát thông tin video/hình ảnh đã được mã hóa hoặc ở dạng của tệp hoặc việc tạo luồng tới thiết bị nhận thông qua phương tiện lưu trữ kỹ thuật số hoặc mạng.

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

Nguồn video có thể thu được video/hình ảnh qua quy trình chụp, tổng hợp, hoặc sinh ra video/hình ảnh. Nguồn video có thể bao gồm thiết bị chụp video/hình ảnh và/hoặc thiết bị sinh ra video/hình ảnh. Thiết bị chụp video/hình ảnh có thể bao gồm, ví dụ, một hoặc nhiều camera, các phần lưu trữ video/hình ảnh chứa các video/hình ảnh đã được chụp từ trước, hoặc dạng tương tự. Thiết bị sinh ra video/hình ảnh có thể bao gồm, ví dụ, máy tính, máy tính bảng và điện thoại thông minh, và có thể sinh ra video/hình ảnh (theo cách điện tử). Ví dụ, video/hình ảnh ảo có thể được sinh ra qua máy tính hoặc dạng tương tự. Trong trường hợp này, quy trình chụp video/hình ảnh có thể được thay thế bằng quy trình sinh 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 các chuỗi thủ tục như dự đoán, biến đổi, và lượng tử hóa để nén và lập mã một cách hiệu quả. Dữ liệu đã được mã hóa (thông tin video/hình ảnh đã được mã hóa) có thể được đưa ra ở dạng của luồng bit.

Bộ truyền có thể truyền thông tin video/hình ảnh đã được mã hóa hoặc dữ liệu được đưa ra ở dạng của luồng bit tới bộ thu của thiết bị nhận qua phương tiện lưu trữ kỹ thuật số hoặc mạng ở dạng của tệp hoặc việc tạo luồng. Phương tiện lưu trữ kỹ thuật số có thể bao gồm các phương tiện lưu trữ bất kỳ như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và dạng tương tự. Bộ truyền có thể bao gồm phần tử để tạo ra tệp phương tiện qua định dạng tệp đã được xác định từ trước, và có thể bao gồm phần tử cho việc truyền qua mạng truyền quảng bá/truyền thông. Bộ thu có thể nhận/tách luồng bit và truyền luồng bit nhận được/được tách tới thiết bị giải mã.

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

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

Fig.2 là giản đồ minh họa một cách sơ lược cấu hình của thiết bị mã hóa video/hình ảnh mà sáng chế này có thể áp dụng được vào đó. Dưới đây, phần được đề cập tới như là thiết bị mã hóa video có thể bao gồm thiết bị mã hóa hình ảnh.

Đề cập tới Fig.2, thiết bị mã hóa 200 có thể bao gồm bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropy 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể bao gồm bộ liên dự đoán 221 và bộ dự đoán nội 222. Bộ xử lý phần dư 230 có thể bao gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ giải lượng tử hóa 234, bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn chứa bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ phận tái dựng hoặc bộ phận sinh khối được tái dựng. Bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropy 240, bộ cộng 250, và bộ lọc 260, vốn đã được mô tả ở trên, có thể được tạo nên bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip hoặc các bộ xử lý bộ mã hóa) theo phương án thực hiện. Hơn nữa, bộ nhớ 270

có thể bao gồm bộ đệm ảnh được giải mã (decoded picture buffer, DPB), và có thể được tạo nên bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn chứa bộ nhớ 270 làm thành phần bên trong/bên 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 nhị phân tam phân (Quad-tree binary-tree ternary-tree, QTBTNT). 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. Theo cách khác, cấu trúc cây nhị phân có thể được áp dụng đầu tiên. Thủ tục lập mã theo sáng chế này có thể được thực hiện dựa trên đơn vị lập mã cuối cùng vốn không thể được phân vùng thêm nữa. Trong trường hợp này, đơn vị lập mã tối đa có thể được sử dụng một cách trực tiếp như là đơn vị lập mã cuối cùng dựa trên hiệu suất lập mã theo đặc tính của hình ảnh. Theo cách khác, đơn vị lập mã có thể được phân vùng theo cách đệ quy thành các đơn vị lập mã có độ sâu sâu hơn nữa nếu cần, sao cho đơn vị lập mã có kích cỡ tối ưu có thể được sử dụng làm đơn vị lập mã cuối cùng. Ở đây, thủ tục lập mã có thể bao gồm các thủ tục như dự đoán, biến đổi, và tái dựng, vốn sẽ được mô tả sau. Như là ví dụ khác, đơn vị xử lý có thể còn chứa 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 như là khối, khu vực, hoặc dạng tương tự có thể được sử dụng thay thế cho nhau tùy theo hoàn cảnh. Trong trường hợp chung, khối $M \times N$ có thể biểu diễn cho tập hợp các mẫu hoặc các hệ số biến đổi đang tạo thành M cột và N hàng. Mẫu có thể biểu diễn chung cho điểm ảnh hoặc trị số của điểm ảnh, và có thể chỉ biểu diễn điểm ảnh/trị số điểm ảnh của thành phần độ chói, hoặc chỉ điểm ảnh/trị số 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 từ tín hiệu được dự đoán (khối được dự đoán, mảng mẫu được dự đoán) được đưa ra từ bộ dự đoán 220 từ tín hiệu hình ảnh đầu vào (khối gốc, mảng mẫu gốc) để sinh ra tín hiệu phần dư (khối phần dư, mảng mẫu phần dư), và tín hiệu phần dư được sinh ra được truyền tới bộ biến đổi 232. Bộ dự đoán 220 có thể thực hiện việc dự đoán trên khối đích xử lý (dưới đây, sẽ được đề cập tới như là “khối hiện tại”), và có thể sinh ra khối được dự đoán chứa các mẫu dự đoán cho khối hiện tại. Bộ dự đoán 220 có thể xác định xem liệu việc nội dự đoán hay dự đoán liên thông sẽ được áp dụng trên khối hiện tại hoặc trên cơ sở CU. Như được thảo luận sau đây trong phần mô tả của mỗi chế độ dự đoán, bộ dự đoán có thể tạo ra các thông tin khác nhau liên quan đến việc dự đoán, như là thông tin chế độ dự đoán, và truyền thông tin được sinh ra đến bộ mã hóa entropy 240. Thông tin về việc dự đoán có thể được mã hóa trong bộ mã hóa entropy 240 và xuất ra trong dạng luồng bit.

Bộ dự đoán trong ảnh 222 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 trong vùng lân cận của hoặc cách xa khối hiện tại theo chế độ dự đoán. Trong việc nội dự đoán, các chế độ dự đoán có thể bao gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Các chế độ không có hướng có thể bao gồm, ví dụ, chế độ DC và chế độ phẳng. Chế độ có hướng có thể bao gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng theo mức độ chi tiết của hướng dự đoán. Tuy nhiên, nó chỉ đơn thuần là ví dụ, và nhiều hoặc ít chế độ dự đoán có hướng hơn có thể được sử

dụng phụ thuộc vào việc thiết lập. Bộ dự đoán nội 222 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng cách sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

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

khối lân cận có thể được sử dụng làm bộ dự đoán vectơ chuyển động và vectơ chuyển động của khối hiện tại có thể được chỉ thị bởi việc báo hiệu khác biệt vectơ chuyển động.

Bộ dự đoán 220 có thể sinh ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc dự đoán trên một khối, cũng như, có thể áp dụng việc nội dự đoán và việc liên dự đoán tại cùng một thời điểm. Nó có thể được gọi là việc liên dự đoán và nội được kết hợp (combined inter and intra prediction, CIIP). Hơn nữa, bộ dự đoán có thể dựa trên chế độ dự đoán sao chép khối nội (intra block copy, IBC), hoặc chế độ bảng màu (palette) 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 dạng tương tự, như việc lập mã nội dung màn (screen content coding, SCC). Mặc dù IBC về cơ bản là thực hiện việc dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện một cách tương tự với việc liên dự đoán mà trong đó nó dẫn xuất khối tham chiếu trong khối hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một kỹ thuật trong số các kỹ thuật dự đoán liên thông được mô tả trong sáng chế này.

Tín hiệu dự đoán được sinh ra thông qua bộ liên dự đoán 221 và/hoặc bộ dự đoán nội 222 có thể được sử dụng để sinh ra tín hiệu được tái dựng hoặc để sinh ra tín hiệu phần dư. Bộ biến đổi 232 có thể sinh ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu phần dư. Ví dụ, kỹ thuật biến đổi có thể gồm ít nhất một kỹ thuật trong số biến đổi cosin rời rạc (Discrete Cosine Transform, DCT), biến đổi sin rời rạc (Discrete Sine Transform, DST), biến đổi Karhunen-Loève (Karhunen-Loève Transform, KLT), biến đổi dựa trên đồ thị (Graph-Based Transform, GBT), hoặc biến đổi phi tuyến có điều kiện (Conditionally Non-Linear Transform, CNT). Tại đây, GBT nghĩa là biến đổi được thu nhận từ đồ thị khi thông tin quan hệ giữa các điểm ảnh được biểu diễn bởi đồ thị. CNT đề cập đến sự biến đổi được thu nhận dựa trên tín hiệu dự đoán được tạo ra nhờ sử dụng tất cả các điểm ảnh được tái dựng

trước đó. Thêm vào đó, quy trình biến đổi có thể được áp dụng cho các khối điểm ảnh hình vuông có cùng kích cỡ hoặc có thể được áp dụng cho các khối có kích cỡ biến đổi khác với hình vuông.

Bộ lượng tử hóa 233 có thể lượng tử hóa các hệ số biến đổi và truyền chúng đến bộ mã hóa entropy 240, và bộ mã hóa entropy 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 này có thể được đề cập đến như thông tin phần dư. Bộ lượng tử hóa 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa loại khối thành dạng vectơ một chiều dựa trên thứ tự quét hệ số, và sinh 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 entropy 240 có thể thực hiện các phương pháp mã hóa khác nhau như, ví dụ, 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 dạng tương tự. Bộ mã hóa entropy 240 có thể mã hóa thông tin cần thiết cho việc tái dựng video/hình ảnh khác với các hệ số biến đổi được lượng tử hóa (ví dụ các trị số của các thành phần cú pháp, v.v.) cùng nhau hoặc theo cách tách biệt. Thông tin được mã hóa (ví dụ, thông tin video/hình ảnh đã 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) ở dạng của luồng bit. Thông tin video/hình ảnh có thể còn chứa thông tin về các bộ thông số khác nhau như tập hợp thông số thích ứng (Adaptation Parameter Set, APS), tập hợp thông số ảnh (Picture Parameter Set, PPS), tập hợp thông số chuỗi (Sequence Parameter Set, SPS), hoặc tập hợp thông số video (Video Parameter Set, VPS) hoặc dạng tương tự. Hơn nữa, thông tin video/hình ảnh có thể còn chứa thông tin ràng buộc chung. Theo sáng chế này, thông tin và/hoặc các thành phần cú pháp vốn được truyền/được báo hiệu tới thiết bị giải mã từ thiết bị mã hóa có thể được chứa trong thông tin video/hình ảnh.

Thông tin video/hình ảnh có thể được mã hóa qua thủ tục mã hóa được mô tả ở trên và được chứa 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 bao gồm mạng truyền quảng bá, mạng truyền thông và/hoặc dạng tương tự, và phương tiện lưu trữ kỹ thuật số có thể bao gồm các phương tiện lưu trữ khác nhau như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và dạng tương tự. Bộ truyền (không được thể hiện) truyền tín hiệu được đưa ra từ bộ mã hóa entropy 240 và/hoặc bộ phận lưu trữ (không được thể hiện) lưu trữ nó, có thể được tạo cấu hình như là thành phần bên trong/bên ngoài của thiết bị mã hóa 200, hoặc bộ truyền có thể được chứa trong bộ mã hóa entropy 240.

Các hệ số biến đổi được lượng tử hóa được đưa ra từ bộ lượng tử hóa 233 có thể được sử dụng để sinh ra tín hiệu dự đoán. Ví dụ, bằng cách áp dụng việc giải lượng tử hóa hoặc việc biến đổi ngược cho các hệ số biến đổi được lượng tử hóa qua bộ giải 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 tái dựng. Bộ cộng 155 thêm tín hiệu phần dư được tái dựng vào tín hiệu dự đoán được đưa ra từ bộ liên dự đoán 221 hoặc bộ dự đoán nội 222, sao cho tín hiệu được tái dựng (ảnh được tái dựng, khối được tái dựng, mảng mẫu được tái dựng) có thể được sinh ra. Khi không có phần dư cho việc xử lý khối đích như trong trường hợp mà trong đó chế độ bỏ qua được áp dụng, thì khối dự đoán có thể được sử dụng làm khối được tái dựng. Bộ cộng 250 có thể được gọi là bộ phận tái dựng hoặc bộ phận sinh khối được tái dựng. Tín hiệu được tái dựng được sinh ra có thể được sử dụng cho việc nội dự đoán của việc xử lý tiếp theo của khối đích trong khối hiện tại, và như sẽ được mô tả sau, có thể được sử dụng cho việc liên dự đoán của ảnh tiếp theo qua việc lọc.

Trong khi đó, trong quy trình mã hóa và/hoặc tái dựng 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 cách áp dụng việc lọc cho tín hiệu được tái dựng. Ví dụ, bộ lọc 260 có thể sinh ra ảnh được tái dựng, được biến đổi bằng cách áp dụng nhiều phương pháp lọc khác nhau cho ảnh được tái dựng, và có thể lưu ảnh được tái dựng, được biến đổi trong bộ nhớ 270, cụ thể là trong DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể bao gồm, ví dụ, việc lọc khử khối, dịch vị tương thích mẫu, bộ lọc vòng lặp tương thích, bộ lọc song phương hoặc dạng tương tự. Như sẽ được thảo luận sau trong phần mô tả của từng phương pháp lọc, bộ lọc 260 có thể sinh ra thông tin khác nhau liên quan tới việc lọc, và truyền thông tin được sinh ra tới bộ mã hóa entropy 240. Thông tin về việc lọc có thể được mã hóa trong bộ mã hóa entropy 240 và đưa ra ở dạng của luồng bit.

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

Bộ nhớ 270 DPB có thể lưu trữ ảnh được tái dựng, được sửa đổi để sử dụng nó làm ảnh tham chiếu trong bộ dự đoán liên liên thông 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 vừa được tái dựng. Thông tin chuyển động được lưu có thể được truyền tới bộ liên dự đoán 221 để được tận 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 giữ các mẫu được tái dựng của các khối được tái dựng trong ảnh hiện tại, và truyền chúng tới bộ dự đoán nội 222.

Fig.3 là giản đồ minh họa một cách sơ lược cấu hình của thiết bị giải mã video/hình ảnh mà sáng chế này có thể áp dụng được vào đó.

Đề cập tới Fig.3, thiết bị giải mã video 300 có thể bao gồm bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, bộ lọc 350 và bộ nhớ 360. Bộ dự đoán 330 có thể bao gồm bộ liên dự đoán 331 và bộ dự đoán nội 332. Bộ xử lý phần dư 320 có thể bao gồm bộ giải lượng tử hóa 321 và bộ biến đổi ngược 321. Bộ giải mã entropy 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, vốn đã được mô tả ở trên, có thể được tạo nên bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip hoặc bộ giải mã hoặc các bộ xử lý) theo phương án thực hiện. Hơn nữa, bộ nhớ 360 có thể bao gồm bộ đệm ảnh được giải mã (decoded picture buffer, DPB), và có thể được tạo nên bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn chứa bộ nhớ 360 làm thành phần bên trong/bên ngoài.

Khi luồng bit chứa thông tin video/hình ảnh là đầu vào, thì thiết bị giải mã 300 có thể tái dựng hình ảnh tương ứng với quy trình mà nhờ 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 được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã bằng cách 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, ví dụ, là đơn vị lập mã, vốn có thể được phân vùng dọc theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây lập mã hoặc đơn vị lập mã lớn nhất. Một hoặc nhiều đơn vị biến đổi có thể được dẫn xuất từ đơn vị lập mã. Và, tín hiệu hình ảnh được tái dựng được giải mã và được đưa ra qua thiết bị giải mã 300 có thể được tái tạo qua bộ phận tái tạo.

Thiết bị giải mã 300 có thể nhận tín hiệu đưa ra từ thiết bị mã hóa của Fig.2 ở dạng của luồng bit, và tín hiệu nhận được có thể được giải mã qua bộ giải mã entropy

310. Ví dụ, bộ giải mã entropy 310 có thể phân tích cú pháp luồng bit để dẫn xuất thông tin (ví dụ, thông tin video/hình ảnh) được yêu cầu cho việc tái dựng hình ảnh (hoặc tái dựng ảnh). Thông tin video/hình ảnh có thể còn chứa thông tin về các bộ thông số khác nhau như tập hợp thông số thích ứng (Adaptation Parameter Set, APS), tập hợp thông số ảnh (Picture Parameter Set, PPS), tập hợp thông số chuỗi (Sequence Parameter Set, SPS), hoặc tập hợp thông số video (Video Parameter Set, VPS) hoặc dạng tương tự. Hơn nữa, thông tin video/hình ảnh có thể còn chứa thông tin ràng buộc chung. Thiết bị giải mã có thể giải mã ảnh còn dựa trên thông tin về bộ thông số và/hoặc thông tin ràng buộc chung. Theo sáng chế này, thông tin và/hoặc các thành phần cú pháp được báo hiệu/nhận được, vốn sẽ được mô tả sau, có thể được giải mã qua thủ tục giải mã và được thu từ luồng bit. Ví dụ, bộ giải mã entropy 310 có thể giải mã thông tin trong luồng bit dựa trên phương pháp lập mã như là mã hóa Golomb hàm số mũ, CAVLC, CABAC, hoặc dạng tương tự, và có thể đưa ra trị số của thành phần cú pháp cần thiết cho việc tái dựng hình ảnh và các trị số được lượng tử hóa của hệ số biến đổi mà không quan tâm tới phần dư. Cụ thể hơn là, phương pháp giải mã entropy CABAC có thể nhận bin tương ứng với từng thành phần cú pháp trong luồng bit, xác định mô hình ngữ cảnh sử dụng thông tin thành phần cú pháp đích giải mã và thông tin việc giải mã của các khối lân cận và các khối đích giải mã, hoặc thông tin về ký hiệu/bin được giải mã trong bước trước, dự đoán khả năng sinh ra bin theo mô hình ngữ cảnh được xác định và thực hiện việc giải mã số hoặc của bin để sinh ra ký hiệu tương ứng với từng trị số thành phần cú pháp. Ở đây, phương pháp giải mã entropy CABAC có thể cập nhật mô hình ngữ cảnh sử dụng thông tin của ký hiệu/bin được giải mã cho mô hình ngữ cảnh của ký hiệu/bin 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 entropy 310 có thể được cung cấp tới bộ dự đoán (bộ liên dự đoán 332 và bộ dự đoán nội 331), và các trị số phần dư, nghĩa là, các hệ số biến đổi được lượng tử hóa, mà việc giải mã entropy đã được thực hiện trên đó trong bộ giải mã entropy

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). Hơn nữa, thông tin về việc lọc trong số thông tin được giải mã trong bộ giải mã entropy 310 có thể được cung cấp cho bộ lọc 350. Trong khi đó, bộ thu (không được thể hiện) vốn nhận tín hiệu được đưa ra từ thiết bị mã hóa có thể còn tạo nên thiết bị giải mã 300 như là chi tiết bên trong/bên ngoài, và bộ thu có thể là thành phần của bộ giải mã entropy 310. Trong khi đó, thiết bị giải mã theo sáng chế này có thể được 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 bao gồm bộ giải mã entropy 310, và bộ giải mã mẫu có thể bao gồm ít nhất một bộ trong số bộ giải lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ liên dự đoán 332, và bộ dự đoán nội 331.

Bộ giải lượng tử hóa 321 có thể đưa ra các hệ số biến đổi bằng cách giải lượng tử hóa các hệ số biến đổi được lượng tử hóa. Bộ giải lượng tử hóa 321 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa ở dạng của 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 vốn đã được thực hiện trong thiết bị mã hóa. Bộ giải lượng tử hóa 321 có thể thực hiện việc giải lượng tử hóa trên các hệ số biến đổi được lượng tử hóa bằng cách sử dụng thông số lượng tử hóa (ví dụ, thông tin kích cỡ bước lượng tử hóa), và thu được các hệ số biến đổi.

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

Bộ dự đoán có thể thực hiện việc dự đoán trên khối hiện tại, và sinh ra khối được dự đoán chứa các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định xem liệu việc nội dự đoán hay dự đoán liên thông được áp dụng cho khối hiện tại dựa

trên thông tin về việc dự đoán được đưa ra từ bộ giải mã entropy 310, và cụ thể là có thể xác định chế độ nội/liên dự đoán.

Bộ dự đoán có thể sinh ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc dự đoán trên một khối, cũng như, có thể áp dụng việc nội dự đoán và việc liên dự đoán tại cùng một thời điểm. Nó có thể được gọi là việc liên dự đoán và nội được kết hợp (combined inter and intra prediction, CIIP). Thêm vào đó, bộ dự đoán có thể thực hiện việc sao chép khối nội (intra block copy, IBC) để dự đoán trên khối. Việc sao chép khối nội có thể được sử dụng cho việc lập mã hình ảnh/video nội dung của trò chơi hoặc dạng tương tự, như việc lập mã nội dung màn (screen content coding, SCC). Mặc dù IBC về cơ bản là thực hiện việc dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện một cách tương tự với việc liên dự đoán mà trong đó nó dẫn xuất khối tham chiếu trong khối hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một kỹ thuật trong số các kỹ thuật dự đoán liên thông được mô tả trong sáng chế này.

Bộ dự đoán nội 331 có thể dự đoán khối hiện tại bằng cách đề cập tới các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt trong vùng lân cận của hoặc cách xa khối khối hiện tại theo chế độ dự đoán. Trong việc nội dự đoán, các chế độ dự đoán có thể bao 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 cách sử dụng chế độ dự đoán được áp dụng cho khối lân cận.

Bộ liên dự đoán 332 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ rõ bởi vectơ chuyển động trên ảnh tham chiếu. Tại thời điểm này, để làm giảm lượng của thông tin chuyển động được truyền trong chế độ dự đoán, 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 bao gồm vectơ chuyển

động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn chứa thông tin hướng dự đoán liên thông (việc dự đoán L0, việc dự đoán L1, việc dự đoán Bi, v.v.). Trong trường hợp của việc liên dự đoán, khối lân cận có thể bao gồm khối lân cận theo không gian đang tồn tại trong ảnh hiện tại và khối lân cận theo thời gian đang tồn tại trong ảnh tham chiếu. Ví dụ, bộ liên dự đoán 332 có thể đặt 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 nhận được thông tin chọn ứng viên. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể bao gồm thông tin đang chỉ thị chế độ của việc liên dự đoán cho khối hiện tại.

Bộ cộng 340 có thể sinh ra tín hiệu được tái dựng (ảnh được tái dựng, khối được tái dựng, mảng mẫu được tái dựng) bằng cách cộng tín hiệu phần dư thu được vào tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán) được đưa ra từ bộ dự đoán 330. Khi không có phần dư cho việc xử lý khối đích như trong trường hợp mà trong đó chế độ bỏ qua được áp dụng, thì khối dự đoán có thể được sử dụng làm khối được tái dựng.

Bộ cộng 340 có thể được gọi là bộ phận tái dựng hoặc bộ phận sinh khối được tái dựng. Tín hiệu được tái dựng được sinh ra có thể được sử dụng cho việc nội dự đoán của khối đích xử lý tiếp theo trong khối hiện tại, và như sẽ được mô tả sau, có thể được đưa ra qua việc lọc hoặc được sử dụng cho việc nội dự đoán của ảnh tiếp theo.

Trong khi đó, trong quy trình giải mã 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 350 có thể cải thiện chất lượng video chủ quan/khách quan bằng cách áp dụng việc lọc cho tín hiệu được tái dựng. Ví dụ, bộ lọc 350 có thể sinh ra ảnh được tái dựng, được biến đổi bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh

được tái dựng, và có thể truyền ảnh được tái dựng, được biến đổi trong bộ nhớ 360, cụ thể là trong DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể bao gồm, ví dụ, việc lọc khử khối, dịch vị tương thích mẫu, bộ lọc vòng lặp tương thích, bộ lọc song phương hoặc dạng tương tự.

Ảnh được tái dựng (được biến đổi) đã được lưu trong DPB của bộ nhớ 360 có thể được sử dụng như là ảnh tham chiếu trong bộ liên dự đoán 332. Bộ nhớ 360 có thể lưu thông tin chuyển động của khối trong ảnh hiện tại, mà 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 hiện đã được tái dựng. Thông tin chuyển động được lưu trữ có thể được truyền tới bộ liên dự đoán 260 để được tận 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 giữ các mẫu được tái dựng của các khối được tái dựng 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 giải 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 với bộ dự đoán 220, bộ giải lượng tử hóa 234, bộ biến đổi ngược 235, và bộ lọc 260 của thiết bị mã hóa 200, theo cách tương ứng.

Như được mô tả ở trên, việc dự đoán được thực hiện để tăng hiệu suất nén trong việc thực hiện việc lập mã video. Qua đó, khối được dự đoán chứa các mẫu dự đoán cho khối hiện tại, vốn là khối đích lập mã, có thể được sinh ra. Ở đây, khối được dự đoán chứa các mẫu dự đoán trong miền không gian (hoặc miền điểm ảnh). Khối được dự đoán có thể được dẫn xuất theo cách trùng lặp trong thiết bị mã hóa và thiết bị giải mã, và thiết bị mã hóa có thể tăng hiệu suất lập mã hình ảnh bằng cách báo hiệu cho thiết bị giải mã không phải là trị số mẫu gốc của tự bản thân khối gốc mà là 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 chứa các mẫu phần dư dựa trên thông tin phần dư, sinh

ra khỏi được tái dựng chứa các mẫu được tái dựng bằng cách thêm khối phần dư vào khối được dự đoán, và sinh ra ảnh được tái dựng chứa các khối được tái dựng.

Thông tin phần dư có thể được sinh ra 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 cách thực hiện thủ tục biến đổi trên các mẫu phần dư (mảng mẫu phần dư) được chứa trong khối phần dư, và dẫn xuất các hệ số biến đổi được lượng tử hóa bằng cách thực hiện thủ tục lượng tử hóa trên các hệ số biến đổi, sao cho nó có thể báo hiệu thông tin phần dư được kết hợp cho thiết bị giải mã (qua luồng bit). Ở đây, thông tin phần dư có thể bao gồm thông tin trị số, thông tin vị trí, kỹ thuật biến đổi, nhân biến đổi, thông số lượng tử hóa hoặc dạng 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/giải 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 sinh ra khối được tái dựng dựa trên khối được dự đoán và khối phần dư. Thiết bị mã hóa có thể dẫn xuất khối phần dư bằng cách giải lượng tử hóa/biến đổi ngược các hệ số biến đổi được lượng tử hóa để tham khảo cho việc liên dự đoán của ảnh tiếp theo, và có thể sinh ra ảnh được tái dựng dựa trên đó.

Fig.4 minh họa một cách sơ lược nhiều kỹ thuật biến đổi theo phương án thực hiện của sáng chế này.

Đề cập tới Fig.4, bộ biến đổi có thể tương ứng với bộ biến đổi trong thiết bị mã hóa của Fig.2 nêu trên, và bộ biến đổi ngược có thể tương ứng với bộ biến đổi ngược trong thiết bị mã hóa của Fig.2 nêu trên, 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 cách thực hiện việc biến đổi sơ cấp dựa trên các mẫu phần dư (mảng mẫu phần dư) trong khối phần dư (S410). Việc biến đổi sơ cấp này có thể được đề cập tới như là việc biến đổi lõi. Ở

đây, việc biến đổi sơ cấp có thể dựa trên việc chọn nhiều biến đổi (multiple transform selection, MTS), và khi nhiều biến đổi được áp dụng làm việc biến đổi sơ cấp, thì nó có thể được đề cập tới như là việc biến đổi nhiều lần.

Việc biến đổi nhiều lần có thể biểu diễn phương pháp biến đổi còn sử dụng 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 biến đổi nhiều lần có thể biểu diễn 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 nhân biến đổi được chọn 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 tạm thời từ góc nhìn của bộ biến đổi.

Nói cách khác, khi phương pháp biến đổi truyền thống được áp dụng, thì các hệ số biến đổi có thể được sinh ra bằng cách áp dụng việc biến đổi từ miền không gian sang miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT loại 2. Không giống như vậy, khi việc biến đổi nhiều nhân được áp dụng, các hệ số biến đổi (hoặc các hệ số biến đổi sơ cấp) có thể được sinh ra bằng cách áp dụng việc biến đổi từ miền không gian thành 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 loại biến đổi, nhân biến đổi hoặc lõi biến đổi. Các loại biến đổi DCT/DST này có thể được định nghĩa dựa trên cơ sở các chức năng.

Khi việc biến đổi nhiều nhân được thực hiện, nhân biến đổi thẳng đứng và nhân biến đổi nằm ngang cho khối đích có thể được chọn từ trong số các nhân biến đổi, việc biến đổi thẳng đứng có thể được thực hiện trên khối đích dựa trên nhân biến đổi thẳng đứng, và việc biến đổi nằm ngang có thể được thực hiện trên khối đích dựa trên nhân biến đổi nằm ngang. Ở đây, việc biến đổi nằm ngang có thể chỉ thị việc biến đổi

trên các thành phần nằm ngang của khối đích, và việc biến đổi thẳng đứng có thể chỉ thị việc biến đổi trên các thành phần thẳng đứng của khối đích. Nhân biến đổi thẳng đứng/nhân biến đổi nằm ngang có thể được xác định theo cách tương thích, dựa trên chế độ dự đoán và/hoặc chỉ số biến đổi cho khối đích (CU hoặc khối con) chứa khối phần dư.

Hơn nữa, theo một ví dụ, nếu việc biến đổi sơ cấp được thực hiện bằng cách áp dụng MTS, thì quan hệ ánh xạ cho các nhân biến đổi có thể được đặt bằng cách thiết lập các chức năng cơ sở cụ thể cho các trị số được định trước và kết hợp các chức năng cơ sở cần được áp dụng trong việc biến đổi thẳng đứng hoặc việc biến đổi nằm ngang. Ví dụ, khi nhân biến đổi nằm ngang được biểu diễn là $trTypeHor$ và nhân biến đổi hướng thẳng đứng được biểu diễn là $trTypeVer$, thì trị số $trTypeHor$ hoặc $trTypeVer$ bằng 0 có thể được đặt cho DCT2, trị số $trTypeHor$ hoặc $trTypeVer$ bằng 1 có thể được đặt cho DST7, và trị số $trTypeHor$ hoặc $trTypeVer$ 2 có thể được đặt cho DCT8.

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 thị một bộ bất kỳ trong số nhiều bộ nhân biến đổi. Ví dụ, chỉ số MTS là 0 có thể chỉ thị rằng cả hai trị số $trTypeHor$ và $trTypeVer$ là 0, chỉ số MTS là 1 có thể chỉ thị rằng cả hai trị số $trTypeHor$ và $trTypeVer$ là 1, chỉ số MTS là 2 có thể chỉ thị rằng trị số $trTypeHor$ là 2 và trị số $trTypeVer$ là 1, chỉ số MTS là 3 có thể chỉ thị rằng trị số $trTypeHor$ là 1 và trị số $trTypeVer$ là 2, và chỉ số MTS là 4 có thể chỉ thị rằng cả hai trị số $trTypeHor$ và $trTypeVer$ là 2.

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

[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 biến đổi, bằng cách thực hiện việc biến đổi thứ cấp dựa trên các hệ số biến đổi (sơ cấp) (S420). Việc biến đổi sơ cấp là biến đổi từ miền không gian sang miền tần số, và việc biến đổi thứ cấp đề cập tới việc biến đổi thành biểu diễn nén hơn nữa bằng cách sử dụng việc tương liên đang tồn tại giữa các hệ số biến đổi (sơ cấp). Biến đổi thứ cấp có thể bao gồm biến đổi không tách riêng được. Trong trường hợp này, biến đổi thứ cấp có thể được gọi là biến đổi thứ cấp không tách riêng được (non-separable secondary transform, NSST), hoặc biến đổi thứ cấp không tách riêng được phụ thuộc vào chế độ (mode-dependent non-separable secondary transform, MDNSST). Biến đổi thứ cấp không tách riêng được có thể biểu diễn biến đổi sinh ra các hệ số biến đổi đã được biến đổi (hoặc các hệ số biến đổi thứ cấp) cho tín hiệu phần dư bằng cách biến đổi thứ cấp, dựa trên ma trận biến đổi không tách riêng được, các hệ số biến đổi (sơ cấp) được dẫn xuất qua việc biến đổi sơ cấp. Tại thời điểm này, việc biến đổi thẳng đứng và việc biến đổi nằm ngang có thể không được áp dụng một cách tách biệt (hoặc các việc biến đổi thẳng đứng và biến đổi thẳng đứng 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 một lần dựa trên ma trận biến đổi không tách riêng được. Nói cách khác, việc biến đổi thứ cấp không tách riêng được có thể biểu diễn phương pháp biến đổi mà trong đó nó không được áp dụng một cách tách biệt theo hướng thẳng đứng và hướng nằm 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 thành tín hiệu một chiều qua hướng được xác định cụ thể (ví dụ,

hướng hàng là đầu tiên hoặc hướng cột là đầu tiên), và sau đó các hệ số biến đổi đã được biến đổi (hoặc các hệ số biến đổi thứ cấp) được sinh ra dựa trên ma trận biến đổi không tách riêng được. Ví dụ, theo thứ tự hàng là đầu tiên, thì các khối $M \times N$ được bố trí theo đường theo thứ tự của hàng thứ nhất, hàng thứ hai, ..., và hàng thứ N . Theo thứ tự cột là đầu tiên, thì các khối $M \times N$ được bố trí trong đường theo thứ tự của cột thứ nhất, cột thứ hai, ..., và cột thứ N . Biến đổi thứ cấp không tách riêng được có thể được áp dụng cho vùng trên đỉnh bên trái của khối được tạo cấu hình với các hệ số biến đổi (sơ cấp) (dưới đây, có thể được đề cập tới như là khối hệ số biến đổi). Ví dụ, nếu chiều rộng (width, W) và chiều cao (height, H) của khối hệ số biến đổi tất cả đều là bằng với hoặc lớn hơn 8, thì biến đổi thứ cấp không tách riêng được 8×8 có thể được áp dụng cho vùng 8×8 trên đỉnh bên trái của khối hệ số biến đổi. Hơn nữa, nếu chiều rộng (width, W) và chiều cao (height, H) của khối hệ số biến đổi tất cả đều là bằng với hoặc lớn hơn 4, và chiều rộng (width, W) hoặc chiều cao (height, H) của khối hệ số biến đổi là nhỏ hơn 8, thì sau đó biến đổi thứ cấp không tách riêng được 4×4 có thể được áp dụng cho vùng $\min(8, W) \times \min(8, H)$ trên đỉnh bên trái của khối hệ số biến đổi. Tuy nhiên, phương án thực hiện không bị hạn chế vào đó, và ví dụ, thậm chí nếu chỉ có điều kiện là chiều rộng (width, W) hoặc chiều cao (height, H) của khối hệ số biến đổi là bằng với hoặc lớn hơn 4 được thỏa mãn, thì biến đổi thứ cấp không tách riêng được 4×4 có thể được áp dụng cho vùng $\min(8, W) \times \min(8, H)$ trên đỉnh bên trái của khối hệ số biến đổi.

Cụ thể là, ví dụ, nếu khối đầu vào 4×4 được sử dụng, thì biến đổi thứ cấp không tách riêng được có thể được thực hiện như sau.

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

[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 ở dạng của véctor, thì véctor $\vec{X}$ có thể được biểu diễn như sau.

[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, véctor $\vec{X}$ là véctor một chiều thu được bằng cách sắp xếp lại khối hai chiều X của phương trình 1 theo thứ tự hàng là đầu tiên.

Trong trường hợp này, việc biến đổi không tách riêng được, thứ cấp có thể được tính toán như sau.

[Phương trình 3]

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

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

Qua phương trình 3 nêu trên, véctor hệ số biến đổi 16x1 $\vec{F}$ có thể được dẫn xuất, và $\vec{F}$ có thể tổ chức lại thành khối 4x4 qua thứ tự quét (nằm ngang, thẳng đứng, theo đường chéo và dạng tương tự). Tuy nhiên, việc tính toán nêu trên là ví dụ, và phép biến đổi hypercube-Givens (Hypercube-Givens Transform, HyGT) hoặc dạng tương tự có thể được sử dụng cho việc tính toán của biến đổi thứ cấp không tách riêng được để làm giảm độ phức tạp tính toán của biến đổi thứ cấp không tách riêng được.

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

Như được mô tả ở trên, biến đổi thứ cấp không tách riêng được có thể được thực hiện dựa trên biến đổi 8×8 hoặc biến đổi 4×4 được xác định dựa trên chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi. Biến đổi 8×8 đề cập tới biến đổi có khả năng áp dụng cho vùng 8×8 được chứa trong khối hệ số biến đổi khi cả W và H là bằng với hoặc lớn hơn 8, và vùng 8×8 có thể là vùng 8×8 trên đỉnh bên trái trong khối hệ số biến đổi. Tương tự, việc biến đổi 4×4 đề cập tới việc biến đổi có khả năng áp dụng cho vùng 4×4 được chứa trong khối hệ số biến đổi khi cả W và H là bằng với hoặc lớn hơn 4, và vùng 4×4 có thể là vùng 4×4 trên đỉnh bên trái trong khối hệ số biến đổi. Ví dụ, ma trận nhân biến đổi 8×8 có thể là ma trận 64×64 / 16×64 , và ma trận nhân biến đổi 4×4 có thể là ma trận 16×16 / 8×16 .

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

Tuy nhiên, khi kích cỡ của việc biến đổi, nghĩa là, kích cỡ của vùng mà việc biến đổi được áp dụng vào đó, có thể, ví dụ, là kích cỡ khác với 8×8 hoặc 4×4 , thì số các bộ có thể là n, và số các nhân biến đổi trong từng bộ có thể là k.

Tập biến đổi có thể được đề cập tới như là bộ NSST hoặc bộ LFNST. Tập cụ thể trong số các tập biến đổi có thể được chọn, ví dụ, dựa trên chế độ nội dự đoán của

khối hiện tại (CU hoặc khối con). Việc biến đổi không tách biệt tần số thấp (low-frequency non-separable transform, LFNST) có thể là ví dụ của việc biến đổi không tách biệt được làm giảm, vốn sẽ được mô tả sau, và biểu diễn việc biến đổi không tách biệt cho thành phần tần số thấp.

Để tham khảo, ví dụ, chế độ nội dự đoán có thể bao gồm hai chế độ nội dự đoán không có hướng (hoặc không có góc) và 65 chế độ nội dự đoán có hướng (hoặc có góc). Các chế độ nội dự đoán không có hướng có thể bao gồm chế độ nội dự đoán phẳng số 0 và chế độ nội dự đoán DC số 1, và các chế độ nội dự đoán có hướng có thể bao gồm 65 chế độ nội dự đoán số 2 đến 66. Tuy nhiên, nó là ví dụ, và tài liệu này có thể được áp dụng thậm chí khi số các chế độ nội dự đoán là số khác. Trong khi đó, trong một số trường hợp, chế độ nội dự đoán số 67 có thể còn được sử dụng, và chế độ nội dự đoán số 67 có thể biểu diễn cho chế độ mô hình tuyến tính (linear model, LM).

Fig.5 thể hiện theo cách làm ví dụ các chế độ theo hướng nội của 65 hướng dự đoán.

Đề cập tới Fig.5, trên cơ sở chế độ nội dự đoán 34 đang có hướng dự đoán theo đường chéo lên phía trên bên trái, các chế độ nội dự đoán có thể được chia thành các chế độ nội dự đoán đang có tính chất hướng là nằm ngang và các chế độ nội dự đoán đang có tính chất hướng là thẳng đứng. Trên Fig.5, H và V mô tả tính chất hướng nằm ngang và tính chất hướng thẳng đứng, theo cách tương ứng, và các số -32 đến 32 chỉ thị các độ dịch chuyển theo các đơn vị $1/32$ trên vị trí lưới mẫu. Các số này có thể biểu diễn dịch chuyển cho trị số chỉ số chế độ. Các chế độ nội dự đoán 2 đến 33 có tính chất hướng nằm ngang, và các chế độ nội dự đoán 34 đến 66 có tính chất hướng thẳng đứng. Nói một cách chặt chẽ, chế độ nội dự đoán 34 có thể được xem xét như đang là không phải là nằm ngang lẫn thẳng đứng, nhưng có thể được phân loại như đang phụ thuộc vào tính chất hướng nằm ngang trong việc xác định tập biến

đổi của biến đổi thứ cấp. Đó là do dữ liệu đầu vào được chuyển vị để được sử dụng cho chế độ hướng thẳng đứng, theo cách đối xứng trên cơ sở chế độ nội dự đoán 34, và phương pháp sắp thẳng dữ liệu đầu vào cho chế độ nằm ngang được sử dụng cho chế độ nội dự đoán 34. Việc chuyển vị dữ liệu đầu vào nghĩa là các hàng và các cột của dữ liệu khối $M \times N$ hai chiều được chuyển đổi thành dữ liệu $N \times M$. Chế độ nội dự đoán 18 và chế độ nội dự đoán 50 có thể biểu diễn cho chế độ nội dự đoán nằm ngang và chế độ nội dự đoán thẳng đứng, theo cách tương ứng, và chế độ nội dự đoán 2 có thể được đề cập tới như là chế độ nội dự đoán đường chéo hướng lên trên, bên phải do chế độ nội dự đoán 2 có điểm ảnh tham chiếu bên trái và thực hiện việc dự đoán theo hướng hướng lên trên bên phải. Tương tự, chế độ nội dự đoán 34 có thể được đề cập tới như là chế độ nội dự đoán đường chéo hướng xuống bên phải, và chế độ nội dự đoán 66 có thể được đề cập tới như chế độ nội dự đoán đường chéo hướng xuống bên trái.

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

[Bảng 2]

predModeIntra	lnstTrSetIdx
$\text{predModeIntra} < 0$	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 trên bảng 2, tập bất kỳ trong số bốn tập biến đổi, nghĩa là, **lnstTrSetIdx**, có thể được ánh xạ tới một chỉ số trong số bốn chỉ số, nghĩa là, 0 tới 3, theo chế độ nội dự đoán.

Khi nó được xác định rằng bộ cụ thể được sử dụng cho việc biến đổi không tách biệt, thì chỉ một trong số k nhân biến đổi trong bộ cụ thể có thể được chọn qua chỉ số biến đổi thứ cấp không tách riêng được. Thiết bị mã hóa có thể dẫn xuất chỉ số biến đổi thứ cấp không tách riêng được đang chỉ thị nhân biến đổi cụ thể dựa trên kiểm tra tỉ lệ-nhiều loạn (rate-distortion, RD) và có thể báo hiệu chỉ số biến đổi thứ cấp không tách riêng được tới thiết bị giải mã. Thiết bị giải mã có thể chọn một nhân trong số k nhân biến đổi trong bộ cụ thể dựa trên chỉ số biến đổi thứ cấp không tách riêng được. Ví dụ, trị số chỉ số $lnst$ 0 có thể đề cập tới nhân biến đổi thứ cấp không tách biệt được thứ nhất, trị số chỉ số $lnst$ 1 có thể đề cập tới nhân biến đổi thứ cấp không tách biệt được thứ hai, và trị số chỉ số $lnst$ 2 có thể đề cập tới nhân biến đổi thứ cấp không tách biệt được thứ ba. Theo cách khác, trị số chỉ số $lnst$ 0 có thể chỉ thị rằng biến đổi thứ cấp không tách riêng được thứ nhất là không được áp dụng cho khối đích, và trị số chỉ số $lnsts$ 1 đến 3 có thể chỉ thị ba nhân biến đổi.

Bộ biến đổi có thể thực hiện biến đổi thứ cấp không tách riêng được dựa trên các nhân biến đổi được chọn, và có thể thu được các hệ số biến đổi được biến đổi (thứ cấp). Như được mô tả ở trên, các hệ số biến đổi đã được biến đổi có thể được dẫn xuất như là 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 truyền tới bộ giải lượng tử hóa/bộ biến đổi ngược trong thiết bị mã hóa.

Trong khi đó, như được mô tả ở trên, nếu biến đổi thứ cấp được bỏ qua, thì các hệ số biến đổi (sơ cấp), vốn là đầu ra của biến đổi sơ cấp (có thể tách biệt được), có thể được dẫn xuất như là 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 truyền tới bộ giải lượng tử hóa/bộ biến đổi ngược trong thiết bị mã hóa.

Bộ biến đổi ngược có thể thực hiện các chuỗi của các thủ tục trong thứ tự ngược mà chúng đã được thực hiện với nó trong bộ biến đổi được mô tả ở trên. Bộ biến đổi

ngược có thể nhận các hệ số bộ biến đổi (được giải lượng tử hóa), và dẫn xuất các hệ số biến đổi (sơ cấp) bằng cách thực hiện việc biến đổi thứ cấp (ngược) (S450), và có thể thu được khối phần dư (các mẫu phần dư) bằng cách thực hiện việc biến đổi sơ cấp (ngược) trên các hệ số biến đổi (sơ cấp) (S460). Liên quan tới vấn đề này, các hệ số biến đổi sơ cấp có thể được gọi là các hệ số biến đổi đã được biến đổi từ góc nhìn của bộ biến đổi ngược. Như được mô tả ở trên, thiết bị mã hóa và thiết bị giải mã có thể sinh ra khối được tái dựng dựa trên khối phần dư và khối được dự đoán, và có thể sinh ra ảnh được tái dựng dựa trên khối được tái dựng.

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

Bộ xác định việc biến đổi ngược thứ cấp có thể xác định việc biến đổi ngược thứ cấp. Trong trường hợp này, bộ xác định việc 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 chỉ rõ theo chế độ nội dự đoán. Theo một phương án thực hiện, phương pháp xác định việc biến đổi thứ cấp có thể được xác định phụ thuộc vào phương pháp xác định việc biến đổi sơ cấp. Các tổ hợp khác nhau của các việc biến đổi sơ cấp và biến đổi thứ cấp có thể được xác định theo chế độ nội dự đoán. Hơn nữa, theo ví dụ, bộ xác định việc biến đổi ngược thứ cấp có thể xác định vùng

mà việc biến đổi ngược thứ cấp được áp dụng vào đó dựa trên kích cỡ của khối hiện tại.

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

Trong khi đó, theo sáng chế này, biến đổi thứ cấp được giảm (reduced secondary transform, RST) mà trong đó kích cỡ của ma trận biến đổi (nhân) được làm giảm có thể được áp dụng trong khái niệm của NSST để làm giảm lượng tính toán và bộ nhớ cần thiết cho biến đổi thứ cấp không tách riêng được.

Trong khi đó, nhân biến đổi, ma trận biến đổi, và hệ số đang tạo nên ma trận nhân biến đổi, nghĩa là, hệ số của nhân hoặc hệ số của ma trận, được mô tả theo sáng chế này có thể được biểu diễn bởi 8 bit. Nó có thể là điều kiện để áp dụng trong thiết bị giải mã và thiết bị mã hóa, và có thể làm giảm lượng bộ nhớ cần thiết cần được yêu cầu để lưu nhân biến đổi với việc suy giảm hiệu quả hoạt động có thể được chứa theo cách hợp lý, khi so sánh với 9 bit hoặc 10 bit đang tồn tại. Thêm vào đó, việc biểu diễn của ma trận nhân trong 8 bit có thể cho phép bộ phận nhân nhỏ được sử dụng, và có thể là thích hợp hơn cho các lệnh nhiều dữ liệu, lệnh đơn (single instruction multiple data, SIMD) được sử dụng cho việc áp dụng phần mềm tối ưu.

Trong sáng chế này, thuật ngữ “RST” có nghĩa là việc biến đổi được thực hiện trên các mẫu phần dư cho khối đích dựa trên ma trận biến đổi mà kích cỡ của nó được làm giảm theo nhân tố giảm. Trong trường hợp của việc thực hiện của việc biến đổi được làm giảm, lượng tính toán cần thiết cho việc biến đổi có thể được làm giảm do việc giảm trong kích cỡ của ma trận biến đổi. Nghĩa là, RST có thể được sử dụng để

giải quyết vấn đề độ phức tạp tính toán đang xuất hiện tại việc biến đổi không tách riêng được hoặc việc biến đổi của khối có kích cỡ lớn.

RST có thể được đề cập tới như là các thuật ngữ khác, như việc biến đổi được làm giảm, biến đổi thứ cấp được làm giảm, việc biến đổi giảm, việc biến đổi được đơn giản hóa, việc biến đổi đơn giản, và dạng tương tự, và tên mà RST có thể được đề cập tới như là không bị hạn chế cho các ví dụ được liệt kê. Theo cách khác, do RST chủ yếu được thực hiện trong vùng tần số thấp chứa hệ số không bằng không trong khối biến đổi, nên nó có thể được đề cập tới như là việc biến đổi không tách riêng được, tần số thấp (Low-Frequency Non-Separable Transform, LFNST). Chỉ số biến đổi có thể được đề cập tới như 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 bao gồm bộ phận biến đổi thứ cấp được giảm, ngược vốn dẫn xuất các hệ số biến đổi đã được biến đổi dựa trên RST ngược của các hệ số biến đổi, và bộ biến đổi sơ cấp ngược vốn dẫn xuất các mẫu phần dư cho khối đích dựa trên việc biến đổi sơ cấp ngược cho các hệ số biến đổi đã được biến đổi. Việc biến đổi sơ cấp ngược đề cập tới việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Theo sáng chế này, việc dẫn xuất hệ số biến đổi dựa trên việc biến đổi có thể đề cập tới việc dẫn xuất hệ số biến đổi bằng cách áp dụng việc biến đổi.

Fig.6 là giản đồ minh họa RST theo phương án thực hiện của sáng chế này.

Theo sáng chế này, “khối đích” có thể đề cập tới khối hiện tại cần được lập mã, khối phần dư, hoặc khối biến đổi.

Trong RST theo một ví dụ, vectơ N chiều có thể được ánh xạ tới vectơ R chiều nằm 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 nghĩa là hình vuông có độ dài của cạnh của khối mà việc biến đổi được áp dụng vào đó, hoặc số tổng cộng của các hệ số biến đổi tương

ứng với khối mà việc biến đổi được áp dụng vào đó, và nhân tố giảm có thể có nghĩa là trị số R/N . Nhân tố giảm có thể được đề cập tới như là nhân tố giảm, nhân tố làm giảm, nhân tố được đơn giản hóa, nhân tố đơn giản hoặc các thuật ngữ khác. Trong khi đó, R có thể được đề cập tới như là hệ số được làm giảm, nhưng theo các tình huống khác, nhân tố giảm có thể có nghĩa là R . Hơn nữa, theo các tình huống, nhân tố giảm có thể có nghĩa là trị số N/R .

Theo một ví dụ, nhân tố giảm hoặc hệ số được làm giảm có thể được báo hiệu qua luồng bit, nhưng ví dụ không bị hạn chế vào đó. Ví dụ, trị số được định nghĩa từ trước cho nhân tố giảm hoặc hệ số được làm giảm có thể được lưu trong từng thiết bị trong số thiết bị mã hóa 200 và thiết bị giải mã 300, và trong trường hợp này, nhân tố được làm giảm hoặc hệ số được làm giảm có thể không được báo hiệu theo cách tách biệt.

Kích cỡ của ma trận biến đổi được làm giảm theo một ví dụ có thể là $R \times N$ nhỏ hơn $N \times N$, kích cỡ của ma trận biến đổi thông thường, và có thể được định nghĩa như trong phương trình 4 dưới đây.

[Phương trình 4]

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

Ma trận T trong khối biến đổi được làm giảm được thể hiện trên Fig.6(a) có thể có nghĩa là ma trận $TR \times N$ của phương trình 4. Như được thể hiện trên Fig.6(a), khi ma trận biến đổi được làm giảm $TR \times N$ được nhân với các mẫu phần dư cho khối đích, thì các hệ số biến đổi cho khối đích có thể được dẫn xuất.

Theo một ví dụ, nếu kích cỡ của khối mà việc biến đổi được áp dụng vào đó là 8×8 và $R=16$ (tức là, $R/N = 16/64 = 1/4$), thì RST theo Fig.6(a) có thể được biểu diễn như là phép toán ma trận như được thể hiện trên phương trình 5 bên dưới. Trong trường hợp này, bộ nhớ và việc tính toán nhân có thể được làm giảm xấp xỉ bằng $1/4$, nhờ nhân tổ được làm giảm.

Theo sáng chế này, phép toán ma trận có thể được hiểu như là thao tác nhân vector cột với ma trận, được bố trí trên bên trái của vector cột, để thu được 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 đích và có thể là các hệ số biến đổi cụ thể được sinh ra bằng cách áp dụng việc biến đổi sơ cấp. Như là kết quả của việc tính toán của phương trình 5 các hệ số biến đổi c_i cho khối đích có thể được dẫn xuất, và quy trình dẫn xuất c_i có thể như ở trong phương trình 6.

[Phương trình 6]

Đối với i từ đến R :

$$c_i = 0$$

Đối với j từ 1 đến N :

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

Như là kết quả của việc tính toán của phương trình 6, các hệ số biến đổi c_1 đến c_R cho khối đích có thể được dẫn xuất. Nghĩa là, khi $R=16$, các hệ số biến đổi c_1 đến c_{16} cho khối đích có thể được dẫn xuất. Nếu, thay cho RST, việc biến đổi thông thường được áp dụng và ma trận biến đổi có kích cỡ 64×64 ($N \times N$) được nhân tới kích cỡ các mẫu phần dư là 64×1 ($N \times 1$), thì chỉ 16 (R) hệ số biến đổi được dẫn xuất cho khối đích do RST đã được áp dụng, mặc dù 64 (N) hệ số biến đổi được dẫn xuất cho khối đích. Do tổng số các hệ số biến đổi cho khối đích được làm giảm từ N đến R , nên số lượng dữ liệu được truyền bởi thiết bị mã hóa 200 tới thiết bị giải mã 300 giảm, sao cho hiệu suất 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 góc nhìn của kích cỡ của ma trận biến đổi, kích cỡ của ma trận biến đổi thông thường là 64×64 ($N \times N$), nhưng kích cỡ của ma trận biến đổi được làm giảm được làm giảm tới 16×64 ($R \times N$), để việc sử dụng bộ nhớ trong trường hợp của việc thực hiện của RST có thể được làm giảm bởi tỉ lệ R/N khi được so sánh với trường hợp thực hiện việc biến đổi thông thường. Thêm vào đó, khi được so sánh với số việc tính toán nhân $N \times N$ trong trường hợp của việc sử dụng ma trận biến đổi thông thường, việc sử dụng của ma trận biến đổi được làm giảm có thể làm giảm số việc tính toán nhân theo tỉ lệ R/N ($R \times N$).

Theo một ví dụ, bộ biến đổi 232 của thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi cho khối đích bằng cách thực hiện việc biến đổi sơ cấp và biến đổi thứ cấp dựa trên RTS trên các mẫu phần dư cho khối đích. Các hệ số biến đổi này có thể được truyền tới 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 đã được biến đổi dựa trên biến đổi thứ cấp được làm giảm (reduced secondary transform, RST) ngược cho các hệ số biến đổi, và có thể dẫn xuất các mẫu phần dư cho khối đích dựa trên việc biến đổi sơ cấp ngược cho các hệ số biến đổi đã được biến đổi.

Kích cỡ của ma trận RST ngược $TN \times R$ theo một ví dụ là $N \times R$ nhỏ hơn kích cỡ $N \times N$ của ma trận biến đổi ngược thông thường, và là trong quan hệ chuyển vị với ma trận biến đổi được làm giảm $TR \times N$ được thể hiện trên phương trình 4.

Ma trận Tt trong Reduced Inv. Khối biến đổi được thể hiện trên Fig.6(b) có thể có nghĩa là ma trận RST ngược $TR \times NT$ (chữ T hoa có nghĩa là chuyển vị). Khi ma trận RST ngược $TR \times NT$ được nhân tới các hệ số biến đổi cho khối đích như được thể hiện trên Fig.6(b), các hệ số biến đổi đã được biến đổi cho khối đích hoặc các mẫu phần dư cho khối hiện tại có thể được dẫn xuất. Ma trận RST ngược $TR \times NT$ có thể được biểu diễn như là $(TR \times NT)N \times R$.

Cụ thể hơn, khi RST ngược được áp dụng như là việc biến đổi ngược thứ cấp, thì các hệ số biến đổi đã được biến đổi cho khối đích có thể được dẫn xuất khi ma trận RST ngược $TR \times NT$ được nhân cho các hệ số biến đổi cho khối đích. 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 đích có thể được dẫn xuất khi ma trận RST ngược $TR \times NT$ được nhân tới các hệ số biến đổi cho khối đích.

Theo một ví dụ, nếu kích cỡ của khối mà việc biến đổi ngược được áp dụng vào đó là 8×8 và $R = 16$ (tức là, $R/N = 16/64 = 1/4$), sau đó RST theo Fig.6(b) có thể được biểu diễn như là phép toán ma trận như được thể hiện trên phương trình 7 bên dưới.

[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 đích. Như là kết quả của việc tính toán của phương trình 7, r_i đang biểu diễn các hệ số biến đổi đã được biến đổi cho khối đích hoặc các mẫu phần dư cho khối đích có thể được dẫn xuất, và quy trình dẫn xuất r_i có thể như trong phương trình 8.

[Phương trình 8]

Đối với i từ 1 đến N $r_i = 0$ Đối với j từ 1 đến R $r_i += t_{ji} * c_j$

Như là kết quả của việc tính toán của phương trình 8, r_1 đến r_N đang biểu diễn các hệ số biến đổi đã được biến đổi cho khối đích hoặc các mẫu phần dư cho khối đích có thể được dẫn xuất. Khi được xem xét từ góc nhìn của kích cỡ của ma trận biến đổi ngược, kích cỡ của ma trận biến đổi ngược thông thường là 64×64 ($N \times N$), nhưng kích cỡ của ma trận biến đổi được làm giảm được làm giảm tới 64×16 ($R \times N$), sao cho việc sử dụng bộ nhớ trong trường hợp của việc thực hiện RST ngược có thể được làm giảm theo tỉ lệ R/N khi được so sánh với trường hợp thực hiện việc biến đổi ngược thông thường. Thêm vào đó, khi được so sánh với số lần tính toán nhân $N \times N$ trong trường hợp của việc sử dụng ma trận biến đổi ngược thông thường, thì việc sử dụng của ma trận biến đổi ngược được làm giảm có thể làm giảm số việc tính toán nhân theo tỉ lệ R/N ($N \times R$).

Cấu hình tập biến đổi được thể hiện trên bảng 2 cũng có thể được áp dụng cho RST 8×8 . Nghĩa là, RST 8×8 có thể được áp dụng theo tập biến đổi trong bảng 2. Do một tập biến đổi chứa hai hoặc ba phần biến đổi (nhân) theo chế độ nội dự đoán, nên có thể được tạo cấu hình để chọn một cho tới bốn việc biến đổi chứa nó trong trường hợp không có biến đổi thứ cấp nào được áp dụng. Trong việc biến đổi mà không có

biến đổi thứ cấp được áp dụng, nó có thể được xem như là áp dụng ma trận đồng nhất. Giả sử rằng các chỉ số 0, 1, 2, và 3 theo cách tương ứng được gán cho bốn việc biến đổi (ví dụ, chỉ số 0 có thể được cấp phát cho trường hợp mà trong đó ma trận đồng nhất được áp dụng, nghĩa là, trường hợp không có biến đổi thứ cấp được áp dụng), chỉ số biến đổi hoặc chỉ số $lfnst$ như là thành phần cú pháp có thể được báo hiệu cho từng khối hệ số biến đổi, nhờ đó gán việc biến đổi cần được áp dụng. Nghĩa là, với khối 8×8 trên đỉnh bên trái, qua chỉ số biến đổi, có thể gán RST 8×8 RST trong cấu hình RST, hoặc để gán $lfnst$ 8×8 khi LFNST được áp dụng. $lfnst$ 8×8 và RST 8×8 đề cập tới các việc biến đổi có thể áp dụng được cho vùng 8×8 được chứa trong khối hệ số biến đổi khi cả W và H của khối đích cần được biến đổi là bằng với hoặc lớn hơn 8, và vùng 8×8 có thể là vùng 8×8 đỉnh bên trái trong khối hệ số biến đổi. Tương tự, $lfnst$ 4×4 và RST 4×4 đề cập tới các việc biến đổi có thể áp dụng được cho vùng 4×4 được chứa trong khối hệ số biến đổi khi cả W và H của khối đích là bằng với hoặc lớn hơn 4, và vùng 4×4 có thể là vùng 4×4 đỉnh bên trái trong khối hệ số biến đổi.

Theo phương án thực hiện của sáng chế này, cho việc biến đổi trong việc xử lý mã hóa, chỉ 48 mảnh dữ liệu là có thể được chọn và ma trận nhân biến đổi 16×48 tối đa có thể được áp dụng vào đó, hơn là áp dụng ma trận nhân biến đổi 16×64 thành 64 mảnh dữ liệu đang tạo thành vùng 8×8 . Ở đây, "tối đa" có nghĩa là có trị số tối đa là 16 trong ma trận nhân biến đổi $m \times 48$ cho việc sinh ra m hệ số. Nghĩa là, khi RST được thực hiện bởi việc áp dụng ma trận nhân biến đổi $m \times 48$ ($m \leq 16$) cho vùng 8×8 , 48 mảnh dữ liệu được nhập vào và m hệ số sẽ được sinh ra. Khi m là 16, 48 mảnh dữ liệu được nhập vào và 16 hệ số được sinh ra. Nghĩa là, giả sử rằng 48 mảnh dữ liệu tạo thành vectơ 48×1 , ma trận 16×48 và vectơ 48×1 được nhân một cách liên tiếp, nhờ đó sinh ra vectơ 16×1 . Ở đây, 48 mảnh dữ liệu đang tạo thành vùng 8×8 có thể được sắp xếp phù hợp, nhờ đó tạo nên vectơ 48×1 . Ví dụ, vectơ 48×1 có thể được xây dựng dựa trên 48 mảnh dữ liệu đang tạo nên vùng ngoại trừ vùng 4×4 đáy

bên phải trong số các vùng 8×8 . Ở đây, khi phép toán ma trận được thực hiện bằng cách áp dụng ma trận nhân biến đổi 16×48 tối đa, 16 hệ số biến đổi đã được biến đổi được sinh ra, và 16 hệ số biến đổi đã được biến đổi có thể được sắp xếp trong vùng 4×4 đỉnh bên trái theo thứ tự quét, và vùng 4×4 đỉnh bên phải và vùng 4×4 đáy bên trái có thể được điền với các số không.

Với việc biến đổi ngược trong quy trình giải mã, ma trận 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 việc xử lý biến đổi ngược được thực hiện bởi thiết bị giải mã, dữ liệu hệ số đầu vào mà RST ngược được áp dụng vào đó được đặt cấu hình trong vectơ một chiều theo thứ tự sắp xếp được định nghĩa từ trước, và vectơ hệ số được biến đổi được thu bằng cách nhân vectơ một chiều và ma trận RST ngược tương ứng trên bên trái của vectơ một chiều có thể được sắp xếp trong khối hai chiều theo thứ tự sắp xếp được định nghĩa từ trước.

Tóm lại, trong quy trình biến đổi, khi RST hoặc LFNST được áp dụng cho vùng 8×8 , phép toán ma trận của 48 các hệ số biến đổi trong các đỉnh bên trái, đỉnh bên phải, và đáy bên trái của vùng 8×8 ngoại trừ vùng đáy bên phải trong số các hệ số biến đổi trong vùng 8×8 và ma trận nhân biến đổi 16×48 . Với phép toán ma trận, 48 hệ số biến đổi là đầu vào trong mảng một chiều. Khi phép toán ma trận được thực hiện, 16 hệ số biến đổi đã được biến đổi được dẫn xuất, và các hệ số biến đổi đã được biến đổi có thể được sắp xếp trong vùng đỉnh bên trái của vùng 8×8 .

Trái lại, trong việc xử lý biến đổi ngược, khi RST ngược hoặc LFNST được áp dụng cho vùng 8×8 , 16 hệ số biến đổi tương ứng với vùng đỉnh bên trái của vùng 8×8 trong số các hệ số biến đổi trong vùng 8×8 có thể là đầu vào trong mảng một chiều theo thứ tự quét và có thể được đưa vào phép toán ma trận với ma trận nhân biến đổi 48×16 . Nghĩa là, phép toán ma trận có thể được biểu diễn như là (ma trận 48×16) * (vectơ hệ số biến đổi 16×1) = (vectơ hệ số biến đổi được biến đổi 48×1). Ở đây,

véc tơ $n \times 1$ có thể được diễn dịch là có cùng một ý nghĩa như ma trận $n \times 1$ và do đó có thể được biểu diễn như là véc tơ cột $n \times 1$. Hơn nữa, * mô tả việc nhân ma trận. Khi phép toán ma trận được thực hiện, 48 hệ số biến đổi đã được biến đổi có thể được dẫn xuất, và 48 hệ số biến đổi đã được biến đổi có thể được sắp xếp trong các vùng đỉnh bên trái, đỉnh bên phải, đáy bên trái của vùng 8×8 ngoại trừ vùng đáy 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 bao gồm bộ biến đổi thứ cấp được làm giảm, ngược để dẫn xuất các hệ số biến đổi đã được biến đổi dựa trên RST ngược trên các hệ số biến đổi và bộ biến đổi sơ cấp ngược để dẫn xuất các mẫu phần dư cho khối đích dựa trên việc biến đổi sơ cấp ngược trên các hệ số biến đổi đã được biến đổi. Việc biến đổi sơ cấp ngược đề cập tới việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Theo sáng chế này, việc dẫn xuất hệ số biến đổi dựa trên việc biến đổi có thể đề cập tới việc dẫn xuất hệ số biến đổi bằng cách áp dụng việc biến đổi.

Việc biến đổi không tách riêng được được mô tả ở trên, LFNST, sẽ được mô tả chi tiết như sau. LFNST có thể bao 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 sơ cấp (lỗi) làm đầu vào, và áp dụng việc biến đổi thứ cấp xuôi (biến đổi thứ cấp).

[Phương trình 9]

$$y = G^T$$

Trong phương trình 9, x và y là các đầu vào và các đầu ra của biến đổi thứ cấp, theo cách tương ứng, và G là ma trận đang biểu diễn biến đổi thứ cấp, và các véc tơ cơ sở biến đổi được tạo nên từ các véc tơ cột. Trong trường hợp của LFNST ngược,

khi chiều của ma trận biến đổi G được biểu diễn như là [số hàng x số cột], trong trường hợp của LFNST xuôi, việc chuyển vị của ma trận G trở thành chiều của GT .

Với LFNST ngược, các kích thước của ma trận G là [48 x 16], [48 x 8], [16 x 16], [16 x 8], và ma trận [48 x 8] và ma trận [16 x 8] là các ma trận riêng phần đã lấy mẫu 8 véctor cơ sở biến đổi từ bên trái của ma trận [48 x 16] và ma trận [16 x 16], theo cách tương ứng.

Mặt khác, cho LFNST xuôi, các kích thước của ma trận GT là [16 x 48], [8 x 48], [16 x 16], [8 x 16], và ma trận [8 x 48] và ma trận [8 x 16] là các ma trận riêng phần thu được bằng cách lấy mẫu 8 véctor cơ sở biến đổi từ đỉnh của ma trận [16 x 48] và ma trận [16 x 16], theo cách tương ứng.

Do đó, trong trường hợp của LFNST xuôi, véctor [48 x 1] hoặc véctor [16 x 1] là khả thi để làm đầu vào x , và véctor [16 x 1] hoặc véctor [8 x 1] là khả thi để làm 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 (two-dimensional, 2D), để xây dựng nên véctor [48 x 1] hoặc véctor [16 x 1] làm đầu vào x , véctor một chiều phản được xây dựng một cách phù hợp bằng cách sắp xếp dữ liệu 2D là đầu ra của việc biến đổi xuôi.

Fig.7 là giản đồ minh họa trình tự của việc sắp xếp dữ liệu đầu ra của việc biến đổi sơ cấp xuôi thành véctor một chiều theo một ví dụ. Các giản đồ bên trái của (a) và (b) của Fig.7 thể hiện trình tự cho việc xây dựng véctor [48 x 1], và các giản đồ bên phải của (a) và (b) của Fig.7 thể hiện trình tự cho việc xây dựng véctor [16 x 1]. Trong trường hợp của LFNST, véctor một chiều x có thể được thu bởi việc sắp xếp liên tiếp dữ liệu 2D theo cùng một thứ tự như trong (a) và (b) của Fig.7.

Hướng sắp xếp của dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được xác định theo chế độ nội dự đoán của khối hiện tại. Ví dụ, khi chế độ nội dự đoán của khối hiện tại là trong hướng nằm ngang so với hướng đường chéo, dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được sắp xếp theo thứ tự của (a) trên Fig.7, và khi

chế độ nội dự đoán của khối hiện tại là trong hướng thẳng đứng so với hướng đường chéo, thì dữ liệu đầu ra của việc biến đổi sơ cấp xuôi có thể được sắp xếp theo thứ tự của (b) của Fig.7.

Theo một ví dụ, thứ tự sắp xếp khác với các thứ tự sắp xếp của (a) và (b) Fig.7 có thể được áp dụng, và để dẫn xuất cùng một kết quả (véctor y) như khi các thứ tự sắp xếp của (a) và (b) trên Fig.7 được áp dụng, các véctor cột của ma trận G có thể được sắp xếp theo thứ tự sắp xếp này. Nghĩa là, có thể sắp xếp lại các véctor cột của G sao cho từng phần tử đang tạo nên véctor x luôn được nhân bởi cùng một véctor cơ sở biến đổi.

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

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

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

Theo ví dụ khác, do thứ tự mà trong đó véctor đầu ra y được xử lý trong việc thực hiện việc lượng tử hóa hoặc việc lập mã phần dư có thể được đặt từ trước, véctor đầu ra y có thể không được sắp xếp trong khối 2D như được thể hiện trên Fig.8. Tuy

nhien, trong trường hợp của việc lập mã phân dư, việc lập mã dữ liệu có thể được thực hiện trong các đơn vị của khối 2D (ví dụ, 4x4) 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 đường chéo của Fig.8.

Trong khi đó, thiết bị giải mã có thể đặt cấu hình vectơ đầu vào một chiều y bằng cách sắp xếp đầu ra dữ liệu hai chiều qua quy trình giải lượng tử hóa hoặc dạng tương tự theo thứ tự quét được đặt trước cho thông tin ngược. Vectơ đầu vào y có thể được đưa ra như là vectơ đầu ra x bởi phương trình sau.

[Phương trình 10]

$$x = Gy$$

Trong trường hợp của LFNST ngược, vectơ đầu ra x có thể được dẫn xuất bằng cách nhân vectơ đầu vào y, vốn là vectơ [16 x 1] hoặc vectơ [8 x 1], bởi ma trận G. Với LFNST ngược, vectơ đầu ra x có thể hoặc là vectơ [48 x 1] hoặc là vectơ [16 x 1].

Vectơ đầu ra x được sắp xếp trong khối hai chiều theo thứ tự được thể hiện trên Fig.7 và được sắp xếp như là 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 của quy trình biến đổi thứ cấp xuôi như là một tổng thể, và trong trường hợp của việc biến đổi ngược, không giống như trong hướng xuôi, việc biến đổi thứ cấp ngược được áp dụng đầu tiên, và sau đó việc biến đổi sơ cấp ngược sẽ được áp dụng.

Trong LFNST ngược, một ma trận trong số 8 ma trận [48 x 16] và 8 ma trận [16 x 16] có thể được chọn như là ma trận biến đổi G. Việc xem liệu có áp dụng ma trận [48 x 16] hay ma trận [16 x 16] phụ thuộc vào kích cỡ và hình dạng của khối.

Thêm vào đó, 8 ma trận có thể được dẫn xuất từ bốn tập biến đổi như được thể hiện trên bảng 2 ở trên, và mỗi tập biến đổi có thể tạo thành từ hai ma trận. Tập biến đổi nào được sử dụng trong số 4 tập biến đổi sẽ được xác định theo chế độ nội dự đoán, và cụ thể hơn là, tập biến đổi được xác định dựa trên trị số của chế độ nội dự đoán được mở rộng bằng cách xem xét việc nội dự đoán góc rộng (Wide Angle Intra Prediction, WAIP). Ma trận nào sẽ được chọn trong số hai ma trận đang tạo nên tập biến đổi được chọn sẽ được dẫn xuất qua việc báo hiệu chỉ số. Cụ thể hơn là, 0, 1, và 2 là có thể làm trị số chỉ số được truyền, 0 có thể chỉ thị rằng LFNST là không được áp dụng, và 1 và 2 có thể chỉ thị một ma trận bất kỳ trong số các ma trận biến đổi đang tạo thành tập biến đổi được chọn dựa trên trị số chế độ nội dự đoán.

Fig.9 là giản đồ minh họa các chế độ nội dự đoán góc rộng theo phương án thực hiện của tài liệu này.

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

Khi chiều rộng của khối hiện tại dự đoán là lớn hơn chiều cao, thì các điểm ảnh tham chiếu cao hơn nói chung là gần hơn với vị trí ở bên trong khối cần được dự đoán. Do đó, có thể chính xác hơn để dự đoán theo hướng đáy bên trái hơn là hướng đỉnh bên phải. Ngược lại, khi chiều cao của khối là lớn hơn chiều rộng, thì các điểm ảnh tham chiếu nói chung là gần hơn với các vị trí ở bên trong khối cần được dự đoán. Do đó, có thể chính xác hơn để dự đoán theo hướng đỉnh bên phải hơn là theo hướng đáy bên trái. Do đó, có thể có lợi để áp dụng việc ánh xạ lại, tức là, việc điều chỉnh chỉ số chế độ, tới chỉ số của chế độ nội dự đoán góc rộng.

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

Bảng 3 bên dưới thể hiện quy trình dẫn xuất chế độ nội được biến đổi bằng cách ánh xạ lại chế độ nội dự đoán tới chế độ nội dự đoán góc rộng.

[Bảng 3]

Inputs to this process are:

- a variable `predModeIntra` specifying the intra prediction mode,
- a variable `nTbW` specifying the transform block width,
- a variable `nTbH` specifying the transform block height,
- a variable `cIdx` specifying the colour component of the current block.

Output of this process is the modified intra prediction mode `predModeIntra`.

The variables `nW` and `nH` are derived as follows:

- If `IntraSubPartitionsSplitType` is equal to `ISP_NO_SPLIT` or `cIdx` is not equal to 0, the following applies:

$$nW = nTbW \quad (8-97)$$

$$nH = nTbH \quad (8-98)$$

- Otherwise (`IntraSubPartitionsSplitType` is not equal to `ISP_NO_SPLIT` and `cIdx` is equal to 0), the following applies:

$$nW = nCbW \quad (8-99)$$

$$nH = nCbH \quad (8-100)$$

The variable `whRatio` is set equal to $\text{Abs}(\text{Log2}(nW / nH))$.

For non-square blocks (`nW` is not equal to `nH`), the intra prediction mode `predModeIntra` is modified as follows:

- If all of the following conditions are true, `predModeIntra` is set equal to $(\text{predModeIntra} + 65)$.
 - `nW` is greater than `nH`
 - `predModeIntra` is greater than or equal to 2
 - `predModeIntra` is less than $(\text{whRatio} > 1) ? (8 + 2 * \text{whRatio}) : 8$
- Otherwise, if all of the following conditions are true, `predModeIntra` is set equal to $(\text{predModeIntra} - 67)$.
 - `nH` is greater than `nW`
 - `predModeIntra` is less than or equal to 66
 - `predModeIntra` is greater than $(\text{whRatio} > 1) ? (60 - 2 * \text{whRatio}) : 60$

Trong bảng 3, trị số chế độ nội dự đoán được mở rộng cuối cùng sẽ được lưu trong biến `predModeIntra`, và `ISP_NO_SPLIT` chỉ thị rằng khối CU không được chia thành phần phân vùng con bởi kỹ thuật nội phân vùng con (Intra Sub Partitions, ISP) hiện được chấp thuận trong tiêu chuẩn VVCd, và các trị số biến `cIdx` là 0, 1, và 2 chỉ thị trường hợp của độ chói, các thành phần Cb, và Cr, theo cách tương ứng. Hàm `Log2` được thể hiện trên bảng 3 trả lại trị số log với cơ số 2, và hàm `Abs` trả lại trị số tuyệt đối.

Biến `predModeIntra` chỉ thị chế độ nội dự đoán và chiều cao và chiều rộng của khối biến đổi, v.v. được sử dụng như là các trị số đầu vào của quy trình ánh xạ chế

độ nội dự đoán góc rộng, và trị số đầu ra là chế độ nội dự đoán được điều chỉnh predModeIntra . Chiều cao và chiều rộng này của khối biến đổi hoặc khối lập mà có thể là chiều cao và chiều rộng của khối hiện tại cho việc ánh xạ lại của chế độ nội dự đoán. Tại thời điểm này, biến whRatio đang phản ánh tỉ lệ của chiều rộng với chiều rộng có thể được đặt tới $\text{Abs}(\text{Log2}(\text{nW} / \text{nH}))$.

Với khối không vuông, chế độ nội dự đoán có thể được chia thành hai trường hợp và được điều chỉnh.

Đầu tiên, nếu tất cả các điều kiện (1) ~ (3) được thỏa mãn, (1) thì chiều rộng của khối hiện tại là lớn hơn chiều cao, (2) chế độ nội dự đoán trước khi điều chỉnh là bằng với hoặc lớn hơn 2, (3) chế độ nội dự đoán là nhỏ hơn trị số được dẫn xuất từ $(8 + 2 * \text{whRatio})$ khi biến whRatio là lớn hơn 1, và là nhỏ hơn 8 khi biến whRatio là nhỏ hơn hoặc bằng với 1 [predModeIntra là nhỏ hơn $(\text{whRatio} > 1)$? $(8 + 2 * \text{whRatio}) : 8$], chế độ nội dự đoán được đặt tới trị số 65 lớn hơn chế độ nội dự đoán [predModeIntra được đặt bằng với $(\text{predModeIntra} + 65)$].

Nếu khác với phần trên, nghĩa là, các điều kiện (1) ~ (3) sau được thỏa mãn, thì (1) chiều cao của khối hiện tại là lớn hơn chiều rộng, (2) chế độ nội dự đoán trước khi điều chỉnh là nhỏ hơn hoặc bằng với 66, (3) chế độ nội dự đoán là lớn hơn trị số được dẫn xuất từ $(60 - 2 * \text{whRatio})$ khi biến whRatio là lớn hơn 1, và là lớn hơn 60 khi biến whRatio là nhỏ hơn hoặc bằng với 1 [predModeIntra là lớn hơn $(\text{whRatio} > 1)$? $(60 - 2 * \text{whRatio}) : 60$], chế độ nội dự đoán được đặt tới trị số 67 nhỏ hơn chế độ nội dự đoán [predModeIntra được đặt bằng với $(\text{predModeIntra} - 67)$].

Bảng 2 bên trên thể hiện cách tập biến đổi được chọn dựa trên trị số chế độ nội dự đoán được mở rộng bởi WAIP trong LFNST. Như được thể hiện trên Fig.9, các chế độ 14 đến 33 và các chế độ 35 đến 80 là đối xứng so với hướng dự đoán quanh chế độ 34. Ví dụ, chế độ 14 và chế độ 54 là đối xứng so với hướng tương ứng với chế

độ 34. Do đó, cùng một tập biến đổi được áp dụng cho các chế độ nằm trong các hướng đối xứng lẫn nhau, và việc đối xứng này cũng được phản ánh trong bảng 2.

Trong khi đó, nó được giả sử rằng dữ liệu nhập LFNST xuôi cho chế độ 54 là đối xứng với dữ liệu nhập LFNST cho chế độ 14. Ví dụ, với chế độ 14 và chế độ 54, dữ liệu hai chiều được sắp xếp lại thành dữ liệu một chiều theo thứ tự sắp xếp được thể hiện trong (a) của Fig.7 và (b) của Fig.7, theo cách tương ứng. Thêm vào đó, như có thể thấy rằng các mẫu hình trong thứ tự được thể hiện trên (a) của Fig.7 và (b) của Fig.7 là đối xứng so với hướng (hướng đường chéo) được chỉ thị bởi chế độ 34.

Trong khi đó, như được mô tả ở trên, vốn biến đổi ma trận của ma trận $[48 \times 16]$ và ma trận $[16 \times 16]$ được áp dụng cho LFNST được xác định bởi kích cỡ và hình dạng của khối đích biến đổi.

Fig.10 là giản đồ minh họa hình dạng khối mà LFNST được áp dụng vào đó. (a) của Fig.10 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.10, các khối với các biên dày hơn chỉ thị các vùng mà LFNST được áp dụng vào đó. Với các khối của các hình vẽ Fig.10 (a) và (b), LFNST được áp dụng cho vùng 4×4 trên đỉnh bên trái, và cho khối của Fig.10 (c), LFNST được áp dụng một cách riêng rẽ cho các vùng 4×4 trên đỉnh bên trái được sắp xếp một cách liên tục. Trên (a), (b), và (c) của Fig.10, do LFNST được áp dụng theo các đơn vị của các vùng 4×4 , LFNST này dưới đây sẽ được đề cập tới như là “LFNST 4×4 ”. Để làm ma trận biến đổi tương ứng, ma trận $[16 \times 16]$ hoặc $[16 \times 8]$ có thể được áp dụng dựa trên chiều ma trận cho G trong các phương trình 9 và 10.

Cụ thể hơn, ma trận $[16 \times 8]$ được áp dụng vào khối 4×4 (4×4 TU hoặc 4×4 CU) của Fig.10 (a) và ma trận $[16 \times 16]$ được áp dụng cho các khối trong (b) và (c) của

Fig.10. Nó là để điều chỉnh độ phức tạp tính toán cho trường hợp tệ nhất cho 8 lần nhân cho mỗi mẫu.

Liên quan tới (d) và (e) của Fig.10, LFNST được áp dụng cho vùng 8x8 trên đỉnh bên trái, và LFNST này ở đây được đề cập tới như là “LFNST 8x8”. Như là ma trận tạo thành tương ứng, ma trận [48 x 16] hoặc ma trận [48 x 8] có thể được áp dụng. Trong trường hợp của LFNST xuôi, do vector [48 x 1] (vector x trong phương trình 9) được nhập vào như là dữ liệu đầu vào, nên tất cả các trị số mẫu của vùng 8x8 trên đỉnh bên trái không được sử dụng làm các trị số đầu vào của LFNST xuôi. Nghĩa là, như có thể được thấy trong thứ tự bên trái của Fig.7 (a) hoặc thứ tự bên trái của Fig.7 (b), vector [48 x 1] có thể được xây dựng dựa trên các mẫu đang thuộc về 3 khối 4x4 còn lại trong khi vẫn để lại khối 4x4 dưới đáy bên phải nguyên vẹn như là nó vốn có.

Ma trận [48 x 8] có thể được áp dụng cho khối 8x8 (8x8 TU hoặc 8x8 CU) trong Fig.10 (d), và ma trận [48x16] có thể được áp dụng cho khối 8x8 trên Fig.10(e). Nó cũng để điều chỉnh độ phức tạp tính toán cho trường hợp xấu nhất là tới 8 lần nhân cho mỗi mẫu.

Phụ thuộc vào hình dạng khối, khi LFNST xuôi tương ứng (LFNST 4x4 hoặc LFNST 8x8) được áp dụng, 8 hoặc 16 dữ liệu đầu ra (vector y trong phương trình 9, vector [8 x 1] hoặc [16 x 1]) sẽ được sinh ra. Trong LFNST xuôi, số dữ liệu đầu ra là bằng với hoặc nhỏ hơn số dữ liệu đầu vào do các đặc tính của ma trận GT.

Fig.11 là giản đồ minh họa việc sắp xếp của dữ liệu đầu ra của LFNST xuôi theo một ví dụ, và thể hiện khối mà trong đó dữ liệu đầu ra của LFNST xuôi được sắp xếp theo hình khối.

Khu vực được gạch chéo tại phần đỉnh bên trái của khối được thể hiện trên Fig.11 tương ứng với khu vực mà trong đó dữ liệu đầu ra của LFNST xuôi nằm tại đó, các vị trí được đánh dấu với 0 chỉ thị các mẫu được điền với các trị số 0, và khu

vực còn lại thể hiện các vùng không bị thay đổi bởi LFNST xuôi. Trong khu vực không bị thay đổi bởi LFNST, dữ liệu đầu vào của việc biến đổi sơ cấp xuôi vẫn giữ nguyên không đổi.

Như được mô tả ở trên, do kích cỡ của ma trận biến đổi được áp dụng thay đổi theo hình dạng của khối, nên số dữ liệu đầu ra cũng thay đổi. Như Fig.11, dữ liệu đầu ra của LFNST xuôi có thể không điền hoàn toàn khối 4x4 trên đỉnh bên trái. Trong trường hợp của (a) và (d) của Fig.11, ma trận $[16 \times 8]$ và ma trận $[48 \times 8]$ được áp dụng cho khối được chỉ thị bởi đường dày hoặc vùng một phần bên trong khối, theo cách tương ứng, và vectơ $[8 \times 1]$ làm đầu ra của LFNST xuôi sẽ được sinh ra. Nghĩa là, theo thứ tự quét được thể hiện trên (b) của Fig.8, chỉ 8 dữ liệu đầu ra có thể được điền như được thể hiện trên (a) và (d) của Fig.11, và 0 có thể được điền trong 8 vị trí còn lại. Trong trường hợp của khối LFNST được áp dụng của Fig.10 (d), như được thể hiện trên Fig.11(d), hai khối 4x4 trong phần đỉnh bên phải và phần đáy bên trái liền kề với khối 4x4 đỉnh bên phải cũng được điền với các trị số 0.

Như được mô tả ở trên, cơ bản là, bằng cách báo hiệu chỉ số LFNST, việc xem liệu có áp dụng LFNST hay không và ma trận biến đổi cần được áp dụng sẽ được chỉ rõ. Như được thể hiện trên Fig.11, khi LFNST được áp dụng, do số dữ liệu đầu ra của LFNST xuôi có thể bằng với hoặc nhỏ hơn số của dữ liệu đầu vào, nên vùng được điền với các trị số bằng không sẽ xuất hiện như sau.

1) Như được thể hiện trên (a) của Fig.11, các mẫu từ thứ 8 và các vị trí sau trong thứ tự quét trong khối 4x4 trên đỉnh bên trái, nghĩa là, các mẫu từ thứ 9 tới thứ 16.

2) Như được thể hiện trên (d) và (e) của Fig.11, khi ma trận $[16 \times 48]$ hoặc ma trận $[8 \times 48]$ được áp dụng, hai khối 4x4 liền kề với khối 4x4 trên đỉnh bên trái hoặc các khối 4x4 thứ hai hoặc thứ ba trong thứ tự quét.

Do đó, nếu dữ liệu không bằng không tồn tại bằng cách kiểm tra các khu vực 1) và 2), thì rõ ràng là 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 của LFNST được chấp thuận trong tiêu chuẩn VVC, do việc báo hiệu của chỉ số LFNST được thực hiện sau việc lập mã phân dư, nên thiết bị mã hóa có thể không biết xem liệu có dữ liệu không bằng không (các hệ số có nghĩa) cho tất cả các vị trí nằm trong TU hoặc khối CU qua việc lập mã phân dư hay không. Theo đó, thiết bị mã hóa có thể xác định xem liệu có thực hiện việc báo hiệu trên chỉ số LFNST dựa trên sự tồn tại của dữ liệu không bằng không hay không, và thiết bị giải mã có thể xác định xem liệu chỉ số LFNST có được phân tích cú pháp hay không. Khi dữ liệu không bằng không không tồn tại trong khu vực được gán trong 1) và 2) nêu trên, thì việc báo hiệu của chỉ số LFNST sẽ được thực hiện.

Do mã đơn phân bị cắt cụt được áp dụng như là phương pháp nhị phân hóa cho chỉ số LFNST, nên chỉ số LFNST được tạo thành từ lên tới hai bin, và 0, 10, và 11 được gán như là các mã nhị phân cho các trị số chỉ số LFNST khả thi là 0, 1, và 2, theo cách tương ứng. Theo một ví dụ, việc lập mã CABAC dựa trên ngữ cảnh có thể được áp dụng cho bin thứ nhất (việc lập mã thông thường), và việc lập mã CABAC dựa trên ngữ cảnh cũng có thể được áp dụng cho bin thứ hai. Việc lập mã của chỉ số LFNST được thể hiện trong bảng sau.

[Bảng 4]

Syntax element	binIdx					
	0	1	2	3	4	>= 5
.....						
lfnst_idx[][]	(treeType != SINGLE_TREE) ? 1 : 0	2	na	na	na	na
.....						

Như được thể hiện trên bảng 4, cho bin thứ nhất ($\text{binIdx} = 0$), ngữ cảnh 0 được áp dụng trong trường hợp của cây đơn, và ngữ cảnh 1 có thể được áp dụng trong trường hợp của cây không phải là cây đơn. Cũng vậy, như được thể hiện trên bảng 4, ngữ cảnh 2 có thể được áp dụng cho bin thứ hai ($\text{binIdx} = 1$). Nghĩa là, hai ngữ cảnh có thể được cấp phát cho bin thứ nhất, một ngữ cảnh có thể được cấp phát cho bin thứ hai, và mỗi ngữ cảnh có thể được phân biệt bởi trị số ctxInc (0, 1, 2).

Ở đây, cây đơn có nghĩa là thành phần độ chói và thành phần sắc độ được lập mã với cùng một cấu trúc lập mã. Khi đơn vị lập mã được chia trong khi đang có cùng một cấu trúc lập mã, và kích cỡ của đơn vị lập mã trở nên nhỏ hơn hoặc bằng với ngưỡng cụ thể, và thành phần độ chói và thành phần sắc độ được lập mã với cấu trúc cây tách biệt, thì đơn vị lập mã tương ứng được coi là cây kép, và do đó ngữ cảnh của bin thứ nhất có thể được xác định. Nghĩa là, như được thể hiện trên bảng 4, ngữ cảnh 1 có thể được cấp phát.

Theo cách khác, khi trị số của biến treeType được gán là SINGLE_TREE cho bin thứ nhất, thì ngữ cảnh 0 có thể được sử dụng, trái lại ngữ cảnh 1 có thể được sử dụng.

Trong khi đó, cho LFNST được chấp thuận, các phương pháp làm đơn giản hóa sau có thể được áp dụng.

(i) Theo một ví dụ, số dữ liệu đầu ra cho LFNST xuôi có thể bị hạn chế vào số tối đa là 16.

Trong trường hợp của (c) trên Fig.10, LFNST 4x4 có thể được áp dụng cho hai vùng 4x4 liền kề với phần trên đỉnh bên trái, theo cách tương ứng, và trong trường hợp này, tối đa là dữ liệu đầu ra 32 LFNST có thể được sinh ra. Khi số dữ liệu đầu ra cho LFNST xuôi bị hạn chế vào số tối đa là 16, trong trường hợp của các khối 4xN/Nx4 ($N \geq 16$) (TU hoặc CU), thì LFNST 4x4 chỉ được áp dụng tới một vùng 4x4 trong phần đỉnh bên trái, và LFNST có thể được áp dụng chỉ một lần cho tất cả các

khối của Fig.10. Thông qua điều này, việc áp dụng của việc lập mã hình ảnh có thể được đơn giản hóa.

Fig.12 thể hiện rằng số của dữ liệu đầu ra cho LFNST xuôi bị hạn chế vào số tối đa là 16, theo một ví dụ. Như trên Fig.12, khi LFNST được áp dụng cho vùng 4×4 trên đỉnh bên trái nhất trong khối $4 \times N$ hoặc $N \times 4$ mà trong đó N là 16 hoặc lớn hơn, thì dữ liệu đầu ra của LFNST xuôi trở thành 16 mảnh.

(ii) Theo một ví dụ, việc đặt bằng không có thể còn được áp dụng cho vùng mà LFNST không được áp dụng vào đó. Trong tài liệu này, việc đặt bằng không có thể có nghĩa là việc điền các trị số của các vị trí đang thuộc về vùng cụ thể với trị số là 0. Nghĩa là, việc đặt bằng không có thể được áp dụng cho vùng không bị 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 , nên việc đặt bằng không có thể được chia thành hai loại ((ii)-(A) và (ii)-(B)) như sau.

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

Như được thể hiện trên Fig.13, liên quan tới khối mà LFNST 4×4 được áp dụng vào đó, nghĩa là, cho tất cả các khối trong các khối trong (a), (b) và (c) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng vào đó, có thể được điền với các số không.

Mặt khác, (d) của Fig.13 thể hiện rằng khi trị số tối đa của số của dữ liệu đầu ra của LFNST xuôi bị hạn chế vào 16 như được thể hiện trên Fig.12, thì việc đặt bằng không được thực hiện trên các khối còn lại mà LFNST 4×4 không được áp dụng vào đó.

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

Như được thể hiện trên Fig.14, liên quan tới khối mà LFNST 8×8 được áp dụng vào đó, nghĩa là, cho tất cả các khối trong các khối trong (d), và (e) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng vào đó, có thể được điền với các số không.

(iii) Do việc đặt bằng không được biểu diễn trong (ii) nêu trên, nên khu vực được điền với các số không có thể không giống nhau khi LFNST được áp dụng. Theo đó, có thể kiểm tra xem liệu dữ liệu không bằng không có tồn tại theo việc đặt bằng không được đề xuất trong (ii) qua khu vực rộng hơn trường hợp của LFNST của Fig.11 hay không.

Ví dụ, khi (ii)-(B) được áp dụng, sau khi việc kiểm tra xem liệu dữ liệu không bằng không có tồn tại hay không trong đó khu vực được điền với các trị số không trong (d) và (e) của Fig.11 bên cạnh khu vực được điền với 0 bên cạnh Fig.14, thì việc báo hiệu cho chỉ số LFNST có thể được thực hiện chỉ khi dữ liệu không bằng không không tồn tại.

Tất nhiên, thậm chí nếu việc đặt bằng không được đề xuất trong (ii) được áp dụng, thì có khả năng kiểm tra xem liệu dữ liệu không bằng không có tồn tại theo cùng một cách như đang tồn tại việc báo hiệu chỉ số LFNST hay không. Nghĩa là, sau khi kiểm tra xem liệu dữ liệu không bằng không có tồn tại trong khối được điền với các số không trên Fig.11 hay không, thì 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 sử việc đặt bằng không, nghĩa là, kiểm tra chỉ xem liệu dữ liệu không bằng không có tồn tại trong khu vực được đánh dấu rõ là 0 trong Fig.11 hay không, có thể thực hiện việc phân tích cú pháp chỉ số LFNST.

Các phương án thực hiện khác nhau mà trong đó các tổ hợp của các phương pháp đơn giản hóa ((i), (ii)-(A), (ii)-(B), (iii)) cho LFNST được áp dụng có thể được dẫn xuất. Tất nhiên, các tổ hợp của các phương pháp đơn giản hóa nêu trên không bị hạn chế vào các phương án thực hiện dưới đây, và tổ hợp bất kỳ có thể được áp dụng cho LFNST.

Phương án thực hiện

- Giới hạn số dữ liệu đầu ra cho LFNST xuôi tới tối đa là 16 $\rightarrow$ (i)
- Khi LFNST 4x4 được áp dụng, thì tất cả các khu vực mà LFNST 4x4 không được áp dụng với là $\rightarrow$ (ii)-(A) được đặt bằng không
- Khi LFNST 8x8 được áp dụng, tất cả các khu vực mà LFNST 8x8 không được áp dụng với là $\rightarrow$ (ii)-(B) được đặt bằng không
- Sau khi kiểm tra xem liệu dữ liệu không bằng không có tồn tại hay không, cũng đang tồn tại khu vực được điền với trị số không và khu vực được điền với các số không do việc đặt bằng không bổ sung ((ii)-(A), (ii)-(B)), chỉ số LFNST được báo hiệu chỉ khi dữ liệu không bằng không không tồn tại $\rightarrow$ (iii)

Trong trường hợp của phương án thực hiện này, khi LFNST được áp dụng, khu vực mà trong đó dữ liệu đầu ra không bằng không có thể tồn tại bị hạn chế vào khu vực 4x4 trên đỉnh 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 mà trong đó dữ liệu không bằng 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 (tức là, vị trí của gờ đáy bên phải của khối 4x4 trên đỉnh bên trái) là vị trí cuối cùng mà trong đó dữ liệu khác với 0 có thể tồn tại.

Do đó, khi LFNST được áp dụng, sau khi việc kiểm tra xem liệu dữ liệu không bằng không có tồn tại trong vị trí mà tại đó việc xử lý việc lập mã phần dư không

được cho phép (tại vị trí vượt quá vị trí cuối cùng) hay không, thì nó có thể được xác định xem liệu chỉ số LFNST có được báo hiệu hay không.

Trong trường hợp của phương pháp đặt bằng không được đề xuất trong (ii), do số dữ liệu cuối cùng được sinh ra khi việc biến đổi sơ cấp và LFNST được áp dụng, lượng tính toán cần thiết để thực hiện toàn bộ quy trình biến đổi có thể được làm giảm. Nghĩa là, khi LFNST được áp dụng, do việc đặt bằng không được áp dụng cho việc biến đổi sơ cấp xuôi đưa ra dữ liệu đang tồn tại trong vùng mà trong đó LFNST không được áp dụng, nên không cần phải sinh ra dữ liệu cho vùng trở thành phần đặt bằng không trong suốt quá trình thực hiện việc biến đổi sơ cấp xuôi. Theo đó, có thể làm giảm lượng tính toán cần thiết để sinh ra dữ liệu tương ứng. Các tác dụng bổ sung của phương pháp đặt bằng không được đề xuất trong (ii) được tóm tắt như sau.

Đầu tiên, như được mô tả ở trên, lượng tính toán cần thiết để thực hiện toàn bộ quy trình biến đổi sẽ được làm giảm.

Cụ thể là, khi (ii)-(B) được áp dụng, lượng tính toán cho trường hợp xấu nhất được làm giảm, sao cho quy trình biến đổi có thể được làm nhẹ. Nói cách khác, nói chung, lượng tính toán lớn là cần thiết để thực hiện việc biến đổi sơ cấp kích cỡ lớn. Bằng cách áp dụng (ii)-(B), số dữ liệu được dẫn xuất như là kết quả của việc thực hiện LFNST xuôi có thể được giảm tới 16 hoặc nhỏ hơn. Thêm vào đó, khi kích cỡ của toàn bộ khối (TU hoặc CU) tăng, thì tác động của việc làm giảm lượng vận hành biến đổi còn tiếp tục được tăng.

Thứ hai, lượng tính toán cần thiết cho toàn bộ quy trình biến đổi có thể được làm giảm, nhờ đó làm giảm việc tiêu thụ công suất cần thiết để thực hiện việc biến đổi.

Thứ ba, trễ tham gia vào trong quy trình biến đổi sẽ được làm giảm.

Việc biến đổi thứ cấp như là LFNST bổ sung lượng tính toán vào việc biến đổi sơ cấp đang tồn tại, do đó làm tăng thời gian trễ tổng thể tham gia vào trong việc

thực hiện việc biến đổi. Cụ thể là, trong trường hợp của việc nội dự đoán, do dữ liệu được tái dựng của các khối lân cận được sử dụng trong quy trình dự đoán, nên trong suốt quy trình mã hóa, việc tăng độ trễ do việc biến đổi thứ cấp dẫn tới việc tăng độ trễ cho tới khi tái dựng. Nó có thể dẫn tới việc tăng trong 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, thì thời gian trễ của việc thực hiện việc biến đổi sơ cấp có thể được giảm lớn khi LFNST được áp dụng, thời gian trễ cho toàn bộ việc biến đổi sẽ được duy trì hoặc được làm giảm, sao cho thiết bị mã hóa có thể được áp dụng đơn giản hơn.

Trong khi đó, trong việc nội dự đoán thông thường, khối đích lập mã được coi như là một đơn vị lập mã, và việc lập mã được thực hiện mà không có việc phân vùng của chúng. Tuy nhiên, việc lập mã nội các phân vùng con (Intra Sub-Partition, ISP) đề cập tới việc thực hiện việc lập mã dự đoán nội với khối đích lập mã đang được phân vùng theo hướng nằm ngang hoặc hướng thẳng đứng. Trong trường hợp này, khối được tái dựng có thể được sinh ra bằng cách thực hiện việc mã hóa/giải mã theo các đơn vị của các khối được phân vùng, và khối được tái dựng 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 nội dự đoán được thực hiện trên một khối con bằng cách đề cập tới trị số điểm ảnh được tái dựng của khối con được định vị liền kề với cạnh trái hoặc cạnh trên đỉnh của chúng. Dưới đây, thuật ngữ “lập mã” có thể được sử dụng như là khái niệm cả cả việc lập mã được thực hiện bởi thiết bị mã hóa và việc giải mã được thực hiện bởi thiết bị giải mã.

Dưới đây, việc báo hiệu của chỉ số LFNST và chỉ số MTS sẽ được mô tả.

Bảng cú pháp đơn vị lập mã, bảng cú pháp đơn vị biến đổi, và bảng cú pháp lập mã phân dư liên quan tới việc báo hiệu của chỉ số LFNST và chỉ số MTS theo một ví

dụ được thể hiện như bên dưới. Theo bảng 5, chỉ số MTS di chuyển từ cú pháp mức đơn vị biến đổi tới cú pháp mức đơn vị lập mã, và được báo hiệu sau khi chỉ số LFNST được báo hiệu. Hơn nữa, ràng buộc của việc không cho phép LFNST khi ISP được áp dụng cho đơn vị lập mã sẽ được loại bỏ. Do ràng buộc của việc không cho phép LFNST khi ISP được áp dụng cho đơn vị lập mã được loại bỏ, nên LFNST có thể được áp dụng cho tất cả các khối dự đoán nội. Thêm vào đó, cả chỉ số MTS và chỉ số LFNST được báo hiệu theo cách có điều kiện tại đầu cuối trong mức đơn vị lập mã.

[Bảng 5]

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>
<code>...</code>
<code> LfnstDcOnly = 1</code>
<code> LfnstZeroOutSigCoeffFlag = 1</code>
<code> MtsZeroOutSigCoeffFlag = 1</code>
<code> transform_tree(x0, y0, cbWidth, cbHeight, treeType)</code>
<code> lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC</code> <code> : (IntraSubPartitionsSplitType ==</code> <code> ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth</code>
<code> lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC</code> <code> : (IntraSubPartitionsSplitType ==</code> <code> ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight</code>
<code> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 &&</code> <code> CuPredMode[chType][x0][y0] == MODE_INTRA &&</code> <code> (!intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) &&</code> <code> Max(cbWidth, cbHeight) <= MaxTbSizeY) {</code>
<code> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) &&</code> <code> LfnstZeroOutSigCoeffFlag == 1)</code>
<code> lfnst_idx[x0][y0]</code>
<code> }</code>
<code> if(treeType != DUAL_TREE_CHROMA && lfnst_idx[x0][y0] == 0 &&</code> <code> transform_skip_flag[x0][y0] == 0 && Max(cbWidth, cbHeight) <= 32 &&</code> <code> IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT && (!cu_sbt_flag) &&</code> <code> MtsZeroOutSigCoeffFlag == 1 && tu_cbf_luma[x0][y0]) {</code>
<code> if(((CuPredMode[chType][x0][y0] == MODE_INTER &&</code> <code> sps_explicit_mts_inter_enabled_flag)</code> <code> (CuPredMode[chType][x0][y0] == MODE_INTRA &&</code> <code> sps_explicit_mts_intra_enabled_flag)))</code>
<code> mts_idx[x0][y0]</code>
<code> }</code>
<code>...</code>

[Bảng 6]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {
...
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)
transform_skip_flag[x0][y0]
}
...

[Bảng 7]

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {
...
if((cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6) && cIdx == 0 && log2TbWidth > 4)
log2ZoTbWidth = 4
else
log2ZoTbWidth = Min(log2TbWidth, 5)
MaxCcbbs = 2 * (1 << log2TbWidth) * (1 << log2TbHeight)
if((cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6) && cIdx == 0 && log2TbHeight > 4)
log2ZoTbHeight = 4
else
log2ZoTbHeight = Min(log2TbHeight, 5)
...
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))
LfnstZeroOutSigCoeffFlag = 0
if((LastSignificantCoeffX > 15 LastSignificantCoeffY > 15) && cIdx == 0)
MtsZeroOutSigCoeffFlag = 0
...

Các ý nghĩa của các biến chính được thể hiện trên bảng là như sau.

1. cbWidth, cbHeight: chiều rộng và chiều cao của khối lập mã hiện tại
2. log2TbWidth, log2TbHeight: trị số log cơ số 2 cho chiều rộng và chiều cao của khối biến đổi hiện tại, nó có thể được giảm, bằng cách phản ánh việc đặt bằng

không, cho vùng trên đỉnh bên trái mà trong đó hệ số không bằng không có thể tồn tại.

3. `sps_lfnst_enabled_flag`: cờ đang chỉ thị xem liệu có hay không việc LFNST được cho phép thực hiện, nếu trị số cờ là 0, thì nó chỉ thị rằng LFNST là không được cho phép thực hiện, và nếu trị số cờ là 1, thì nó chỉ thị rằng LFNST được cho phép thực hiện. Nó được định nghĩa trong tập tham số trình tự (sequence parameter set - SPS).

4. `CuPredMode[ chType ][ x0 ][ y0 ]`: chế độ dự đoán tương ứng với biến `chType` và vị trí `(x0, y0)`, `chType` có thể có các trị số là 0 và 1, trong đó 0 chỉ thị thành phần độ chói và 1 chỉ thị thành phần sắc độ. Vị trí `(x0, y0)` chỉ thị vị trí trên ảnh, và `MODE_INTRA` (intra prediction) và `MODE_INTER` (inter prediction) có thể là trị số của `CuPredMode[ chType ][ x0 ][ y0 ]`.

5. `IntraSubPartitionsSplit[ x0 ][ y0 ]`: các nội dung của vị trí `(x0, y0)` là giống như trong mục 4. Nó chỉ thị việc phân vùng ISP nào tại vị trí `(x0, y0)` được áp dụng, `ISP_NO_SPLIT` chỉ thị rằng đơn vị lập mã tương ứng với vị trí `(x0, y0)` không được chia thành các khối phân vùng.

6. `intra_mip_flag[ x0 ][ y0 ]`: các nội dung của vị trí `(x0, y0)` là giống như trong mục 4 ở trên. `intra_mip_flag` là cờ đang chỉ thị xem liệu có chế độ dự đoán dự đoán nội dựa trên ma trận (matrix-based intra prediction, MIP) được áp dụng hay không. Nếu trị số cờ là 0, thì chỉ thị rằng MIP không được cho phép thực hiện, và nếu trị số cờ là 1, thì chỉ thị rằng MIP được cho phép thực hiện.

7. `cIdx`: trị số là 0 chỉ thị độ chói, và các trị số 1 và 2 chỉ thị Cb và Cr vốn tương ứng là các thành phần sắc độ.

8. `treeType`: chỉ thị cây đơn và cây kép, v.v. (`SINGLE_TREE`: cây đơn, `DUAL_TREE_LUMA`: cây kép cho thành phần độ chói, `DUAL_TREE_CHROMA`: cây kép cho thành phần sắc độ)

9. `tu_cbf_cb[ x0 ][ y0 ]`: các nội dung của vị trí $(x0, y0)$ là giống như trong mục 4. Nó chỉ thị cờ khối được lập mã (coded block flag, CBF) cho thành phần Cb. Nếu trị số của nó là 0, thì có nghĩa rằng không có các hệ số không bằng không nào có mặt trong đơn vị biến đổi tương ứng cho thành phần Cb, và nếu trị số của nó là 1, thì nó chỉ thị rằng các hệ số không bằng không là có mặt trong đơn vị biến đổi tương ứng cho thành phần Cb.

10. `lastSubBlock`: Nó chỉ thị vị trí trong thứ tự quét của khối con (nhóm hệ số (CG, Coefficient Group)) mà hệ số khác không cuối cùng nằm trong đó. 0 chỉ thị khối còn mà thành phần DC được chứa trong đó, và trong trường hợp đang lớn hơn 0, nó không phải là khối con mà thành phần DC được chứa ở trong đó.

11. `lastScanPos`: Nó chỉ thị vị trí mà trong đó hệ số có nghĩa cuối cùng là trong thứ tự quét nằm trong khối con. Nếu một khối con chứa 16 vị trí, thì các trị số từ 0 đến 15 là khả thi.

12. `lnst_idx[ x0 ][ y0 ]`: Thành phần cú pháp chỉ số LFNST cần được phân tích cú pháp. Nếu nó không được phân tích cú pháp, thì nó được suy ra như là trị số 0. Nghĩa là, trị số mặc định được đặt là 0, đang chỉ thị rằng LFNST không được áp dụng.

13. `cu_sbt_flag`: Cờ đang chỉ thị xem liệu việc biến đổi khối con (subblock transform, SBT) được chứa trong tiêu chuẩn VVC hiện tại là có thể áp dụng được hay không. Trị số cờ bằng với 0 chỉ thị rằng SBT là không thể áp dụng được, và trị số cờ bằng với 1 chỉ thị rằng SBT được áp dụng.

14. `sps_explicit_mts_inter_enabled_flag`, `sps_explicit_mts_intra_enabled_flag`: Các cờ đang chỉ thị xem liệu MTS rõ ràng có được áp dụng vào CU liên thông và CU nội, theo cách tương ứng, hay không. Trị số cờ bằng với 0 chỉ thị MTS là không áp dụng được cho CU liên thông hoặc CU nội, và trị số cờ bằng với 1 chỉ thị rằng MTS là có thể áp dụng được.

15. `tu_mts_idx[ x0 ][ y0 ]`: Thành phần cú pháp chỉ số MTS cần được phân tích

cú pháp. Khi không được phân tích, thì thành phần này được suy ra như là trị số 0. Nghĩa là, thành phần được đặt tới trị số mặc định là 0, vốn chỉ thị rằng DCT-2 được áp dụng theo cả cách nằm ngang và thẳng đứng.

Như được thể hiện trên bảng 4, nhiều điều kiện được kiểm tra khi lập mã $mts_idx[x0][y0]$, và $tu_mts_idx[x0][y0]$ được báo hiệu chỉ khi $lnst_idx[x0][y0]$ là bằng với 0.

$tu_cbf_luma[x0][y0]$ là cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại cho thành phần độ chói hay không.

Theo bảng 5, khi cả chiều rộng và chiều cao của đơn vị lập mã cho thành phần độ chói là 32 hoặc nhỏ hơn, thì $mts_idx[x0][y0]$ được báo hiệu ($\text{Max}(cbWidth, cbHeight) \leq 32$), nghĩa là, việc xem liệu MTS có được áp dụng hay không được xác định bởi chiều rộng và chiều cao của đơn vị lập mã cho thành phần độ chói.

Hơn nữa, theo bảng 5, nó có thể được tạo cấu hình để báo hiệu $lnst_idx[x0][y0]$ thậm chí trong chế độ ISP ($\text{IntraSubPartitionsSplitType} \neq \text{ISP_NO_SPLIT}$), và cùng một trị số chỉ số LFNST có thể được áp dụng cho tất cả các khối phân vùng ISP.

Tuy nhiên, $mts_idx[x0][y0]$ có thể được tạo tính hiệu chỉ trong trường hợp khác với chế độ ISP ($\text{IntraSubPartitionsSplit}[x0][y0] = \text{ISP_NO_SPLIT}$).

Như được thể hiện trên bảng 7, việc kiểm tra trị số của $mts_idx[x0][y0]$ có thể bị bỏ qua trong quy trình xác định $\log2ZoTbWidth$ và $\log2ZoTbHeight$ (trong đó $\log2ZoTbWidth$ và $\log2ZoTbHeight$ theo cách tương ứng mô tả các trị số logarit cơ số 2 của chiều rộng và chiều cao của vùng trên đỉnh bên trái còn lại sau khi việc đặt bằng không được thực hiện).

Theo một ví dụ, điều kiện kiểm tra $sps_mts_enable_flag$ có thể được thêm vào khi xác định $\log2ZoTbWidth$ và $\log2ZoTbHeight$ trong việc lập mã phân dư.

Biến `LfnstZeroOutSigCoeffFlag` trong bảng 5 là 0 nếu có hệ số có nghĩa tại vị trí đặt bằng không khi LFNST được áp dụng, và là 1 nếu theo cách khác. Biến `LfnstZeroOutSigCoeffFlag` có thể đặt theo nhiều điều kiện được thể hiện trên bảng 7.

Theo một ví dụ, biến `LfnstDcOnly` trong bảng 5 là 1 khi tất cả các hệ số có nghĩa cuối cùng cho các khối biến đổi mà với nó cờ khối được ghi mã tương ứng (CBF, vốn là 1 khi có ít nhất một hệ số có nghĩa trong khối tương ứng, và là 0 theo cách khác) là 1 là tại các vị trí DC (các vị trí trên đỉnh bên trái), và là 0 theo cách khác. Cụ thể là, vị trí của hệ số có nghĩa cuối cùng được kiểm tra liên quan tới một khối biến đổi độ chói trong độ chói cây kép, và vị trí của hệ số có nghĩa cuối cùng được kiểm tra liên quan tới cả khối biến đổi cho Cb và khối biến đổi cho Cr trong sắc độ cây kép. Trong cây đơn, vị trí của hệ số có nghĩa cuối cùng có thể được kiểm tra liên quan tới các khối biến đổi cho độ chói, Cb, và Cr.

Trong bảng 5, `MtsZeroOutSigCoeffFlag` ban đầu được đặt là 1, và trị số này có thể được thay đổi trong việc lập mã phần dư của bảng 6. Biến `MtsZeroOutSigCoeffFlag` được thay đổi từ 1 đến 0 khi có hệ số có nghĩa trong vùng cần được điền đầy với 0s bằng cách đặt bằng không ($\text{LastSignificantCoeffX} > 15 \mid \mid \text{LastSignificantCoeffY} > 15$), trong trường hợp đó chỉ số MTS không được báo hiệu như được thể hiện trên bảng 5.

Như được thể hiện trên bảng 5, khi `tu_cbf_luma[ x0 ][ y0 ]` là 0, thì việc lập mã `mts_idx[ x0 ][ y0 ]` có thể được bỏ qua. Nghĩa là, khi trị số CBF của thành phần độ chói là 0, thì không việc biến đổi nào được áp dụng và do đó chỉ số MTS không cần việc báo hiệu. Do đó, việc lập mã chỉ số MTS có thể được bỏ qua.

Theo một ví dụ, dấu hiệu kỹ thuật nêu trên có thể được áp dụng trong cú pháp điều kiện khác. Ví dụ, sau khi MTS được thực hiện, biến đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng khác với vùng DC của khối hiện tại hay không có thể được dẫn xuất, và khi biến chỉ thị rằng hệ số có nghĩa tồn tại trong vùng loại trừ vùng

DC, thì chỉ số MTS có thể được báo hiệu. Nghĩa là, sự tồn tại của hệ số có nghĩa trong vùng khác với vùng DC của khối hiện tại chỉ thị rằng trị số của $tu_cbf_luma[x0][y0]$ là 1, và trong trường hợp này, chỉ số MTS có thể được báo hiệu.

Biến có thể được biểu diễn như là `MtsDcOnly`, và sau khi biến `MtsDcOnly` ban đầu được đặt là 1 tại mức đơn vị lập mã, thì trị số được thay đổi thành 0 khi nó được xác định rằng hệ số có nghĩa là có mặt trong vùng ngoại trừ vùng DC của khối hiện tại trong mức lập mã phần dư. Khi biến `MtsDcOnly` là 0, thì thông tin hình ảnh có thể được tạo cấu hình sao cho chỉ số MTS được báo hiệu.

Khi $tu_cbf_luma[x0][y0]$ là 0, do cú pháp lập mã phần dư không được gọi tại mức đơn vị biến đổi của bảng 6, nên trị số ban đầu là 1 của biến `MtsDcOnly` được duy trì. Trong trường hợp này, do biến `MtsDcOnly` không bị thay đổi thành 0, nên thông tin hình ảnh có thể được tạo cấu hình sao cho chỉ số MTS không được báo hiệu. Nghĩa là, chỉ số MTS không được phân tích cú pháp và được báo hiệu.

Trong khi đó, thiết bị giải mã có thể xác định chỉ số màu sắc `cIdx` của hệ số biến đổi để dẫn xuất biến `MtsZeroOutSigCoeffFlag` của bảng 7. Chỉ số màu sắc `cIdx` là 0 nghĩa là thành phần độ chói.

Theo một ví dụ, do MTS có thể được áp dụng chỉ cho thành phần độ chói của khối hiện tại, nên thiết bị giải mã có thể xác định xem liệu chỉ số màu sắc là độ chói hay không khi dẫn xuất biến `MtsZeroOutSigCoeffFlag` cho việc xác định xem liệu có phân tích cú pháp chỉ số MTS hay không.

Biến `MtsZeroOutSigCoeffFlag` là biến đang chỉ thị xem liệu việc đặt bằng không có được thực hiện khi MTS được áp dụng hay không. Nó chỉ thị xem liệu hệ số biến đổi có tồn tại trong vùng trên đỉnh bên trái mà trong đó hệ số có nghĩa cuối cùng có thể tồn tại do việc đặt bằng không sau khi MTS được thực hiện, nghĩa là, trong vùng khác với vùng 16x16 trên đỉnh bên trái. Biến `MtsZeroOutSigCoeffFlag` ban đầu được đặt là 1 tại mức đơn vị lập mã như được thể hiện trên bảng 4

(MtsZeroOutSigCoeffFlag = 1), và khi hệ số biến đổi tồn tại trong vùng khác với vùng 16X16, thì trị số của nó có thể được thay đổi từ 1 tới 0 trong mức lập mã phần dư như được thể hiện trên bảng 7 (MtsZeroOutSigCoeffFlag = 0). Khi trị số của biến MtsZeroOutSigCoeffFlag là 0, thì chỉ số MTS không được báo hiệu.

Như được thể hiện trên bảng 7, tại mức lập mã phần dư, vùng không được đặt bằng không trong đó hệ số biến đổi không bằng không có thể tồn tại có thể được đặt phụ thuộc việc xem liệu có hay không có việc đặt bằng không theo MTS được thực hiện, và thậm chí trong trường hợp này, chỉ số màu sắc (cIdx) là 0, thì vùng không được đặt bằng không có thể được đặt tới vùng 16x16 trên đỉnh bên trái của khối hiện tại.

Như vậy, khi dẫn xuất biến xác định xem liệu chỉ số MTS có được phân tích cú pháp hay không, thì nó được xác định xem liệu thành phần màu sắc là độ chói hay sắc độ. Tuy nhiên, do LFNST có thể được áp dụng cho cả thành phần độ chói và thành phần sắc độ của khối hiện tại, nên thành phần màu sắc không được xác định khi dẫn xuất biến để xác định xem liệu có phân tích cú pháp chỉ số LFNST hay không.

Ví dụ, bảng 5 thể hiện biến LfnstZeroOutSigCoeffFlag có thể chỉ thị rằng việc đặt bằng không được thực hiện khi LFNST được áp dụng. Biến LfnstZeroOutSigCoeffFlag chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai ngoại trừ vùng thứ nhất tại đỉnh bên trái của khối hiện tại hay không. Trị số này ban đầu được đặt là 1, và khi hệ số có nghĩa là có mặt trong vùng thứ hai, thì trị số có thể được thay đổi thành 0. Chỉ số LFNST có thể được phân tích cú pháp chỉ khi trị số của biến được đặt từ đầu LfnstZeroOutSigCoeffFlag được duy trì tại 1. Khi việc xác định và dẫn xuất xem liệu trị số của biến LfnstZeroOutSigCoeffFlag có là 1 hay không, do LFNST có thể được áp dụng cho cả thành phần độ chói và thành phần sắc độ của khối hiện tại, nên chỉ số màu sắc của khối hiện tại không được xác định.

Như được thể hiện trên bảng 5, chỉ số MTS được báo hiệu cho từng đơn vị lập

mã, và `mts_idx` (chỉ số MTS) được báo hiệu chỉ khi `MtsZeroOutSigCoeffFlag` là bằng với 1.

Hơn nữa, như được thể hiện trên bảng 7, trị số của `MtsZeroOutCoeffFlag` có thể được xác định theo điều kiện cụ thể (`if( ( LastSignificantCoeffX > 15 || LastSignificantCoeffY > 15 ) && cIdx == 0 )`) trong mức lập mã phần dư.

Trị số của `MtsZeroOutSigCoeffFlag` được đặt tới 0 khi tọa độ x (`LastSignificantCoeffX`) của hệ số khác không cuối cùng là lớn hơn 15 và tọa độ y (`LastSignificantCoeffY`) của hệ số khác không cuối cùng là lớn hơn 15. Ở đây, các tọa độ x tăng từ bên trái sang bên phải trong khối biến đổi, và các tọa độ y tăng từ đỉnh xuống đáy trong khối biến đổi. Vị trí trên đỉnh bên trái của khối biến đổi là (0, 0).

Trị số của `MtsZeroOutSigCoeffFlag` được khởi tạo tới 1 trong bảng 5, và được duy trì là 1 nếu điều kiện có mặt trong bảng 7 không được thỏa mãn và do đó trị số của `MtsZeroOutSigCoeffFlag` không được đặt tới 0.

Khi việc biến đổi sơ cấp được áp dụng theo hướng nằm ngang là A và việc biến đổi sơ cấp được áp dụng theo hướng thẳng đứng là B, thì việc biến đổi sơ cấp có thể được biểu diễn bởi (A, B). Trong tiêu chuẩn VVC hiện tại, trong trường hợp của (DST-7, DST-7), (DST-7, DCT-8), (DCT-8, DST-7), và (DCT-8, DCT-8), hệ số biến đổi bị hạn chế để chỉ tồn tại trong vùng 16 x 16 trên đỉnh bên trái của khối biến đổi. Do đó, điều kiện của (`LastSignificantCoeffX > 15 || LastSignificantCoeffY > 15`) trong bảng 7 chỉ thị rằng hệ số khác không cuối cùng được định vị bên ngoài vùng 16 x 16 trên đỉnh bên trái. Trong bảng 7, biến `cIdx` mô tả thành phần màu sắc, và biến `cIdx` trị số là 0 chỉ thị thành phần độ chói.

Trong tiêu chuẩn VVC hiện tại, chỉ số MTS (`mts_idx`) được phân tích cú pháp sau việc phân tích cú pháp cho việc lập mã phần dư. Cụ thể là, bảng 5 là đơn vị lập mã là chức năng phân tích cú pháp cho cú pháp cho đơn vị lập mã, chỉ số MTS được

phân tích cú pháp sau chức năng cây biến đổi, và chức năng của việc phân tích cú pháp việc lập mã phần dư trong chức năng cây biến đổi sẽ được gọi ra. Do đó, chỉ số MTS được phân tích cú pháp sau việc phân tích cú pháp còn dư (chức năng phân tích cú pháp còn dư được minh họa trên bảng 6 là chịu trách nhiệm cho việc phân tích cú pháp việc lập mã phần dư), và do thông tin chỉ số MTS là không được biết từ góc độ của việc giải mã trong suốt việc lập mã phần dư, nên nó có thể là bất khả thi để biết xem việc biến đổi sơ cấp nào được áp dụng. Tuy nhiên, khi MTS là có thể áp dụng được và SBT hoặc ISP được áp dụng hoặc khi MTS ngầm được áp dụng, thì MTS được áp dụng ngầm, và do đó có thể biết về việc biến đổi sơ cấp trước khi phân tích cú pháp cho việc lập mã phần dư.

Điều kiện được minh họa trên bảng 7 cần kiểm tra, do tiêu chuẩn VVC hiện tại đặt hạn chế cho việc đưa ra chỉ 16 hệ số biến đổi khi việc biến đổi tới được áp dụng trong DST-7 chiều dài 32 hoặc DCT-8 chiều dài 32. Ví dụ, khi trị số của chỉ số MTS là lớn hơn 0 và do đó chỉ thị rằng việc biến đổi sơ cấp được áp dụng là một trong số (DST-7, DST-7), (DST-7, DCT-8), (DCT-8, DST-7), (DCT-8, DCT-8), thì hệ số biến đổi không bằng không có thể tồn tại chỉ trong vùng 16 x 16 trên đỉnh bên trái của khối biến đổi.

Theo đó, khi vị trí của hệ số biến đổi không bằng không cuối cùng là ở bên ngoài vùng 16 x 16 trên đỉnh bên trái, vốn tương ứng với trường hợp mà trong đó trị số chỉ số MTS là 0, thì việc báo hiệu của chỉ số MTS có thể được bỏ qua như được thể hiện trên bảng 5. Tuy nhiên, thậm chí như vậy, vị trí của hệ số biến đổi không bằng không cuối cùng là nằm trong vùng 16 x 16 trên đỉnh bên trái, hệ số không bằng không có thể tồn tại bên ngoài vùng 16 x 16 trên đỉnh bên trái. Trong trường hợp này, chỉ số MTS có thể được báo hiệu như là 0 thậm chí trị số chỉ số MTS là 0, vốn được mô tả chi tiết hơn như sau.

Khi việc biến đổi sơ cấp là (DCT-2, DCT-2), thì vị trí của hệ số khác không

cuối cùng tồn tại trong vùng 16×16 trên đỉnh bên trái và hệ số biến đổi không bằng không cũng có thể tồn tại bên ngoài vùng 16×16 trên đỉnh bên trái, do trong thứ tự quét nằm trong khối biến đổi, vùng bên ngoài vùng 16×16 trên đỉnh bên trái, nghĩa là, vùng khác với vùng 16×16 trên đỉnh bên trái, có thể được quét và sau đó vùng 16×16 trên đỉnh bên trái có thể được quét, hoặc vùng 16×16 trên đỉnh bên trái có thể được quét và sau đó vùng bên ngoài vùng 16×16 trên đỉnh bên trái có thể được quét.

Fig.15 minh họa việc quét của khối biến đổi 32×32 theo phương án thực hiện của sáng chế này, vốn thể hiện rõ rằng khối biến đổi 32×32 được chia thành các nhóm hệ số (coefficient group, CG) 4×4 và được quét. Theo sáng chế này, các CG được chia để được quét được đề cập tới như là “các khối con quét”.

Số được chỉ thị cho từng CG mô tả thứ tự quét xuôi, và các CG có thể được quét bởi thiết bị giải mã theo thứ tự quét ngược lại từ vị trí mà tại đó hệ số biến đổi không bằng không cuối cùng tồn tại. Trên Fig.15, có thể giả sử rằng hệ số biến đổi không bằng không cuối cùng tồn tại trong CG 25, và CG 25 tồn tại trong vùng 16×16 trên đỉnh bên trái.

Như được thể hiện trên Fig.15, khi hệ số biến đổi không bằng không cuối cùng tồn tại trong CG 25, do thiết bị giải mã quét các CG theo thứ tự quét ngược từ CG mà trong đó hệ số biến đổi không bằng không cuối cùng tồn tại, nên thiết bị giải mã quét các CG theo thứ tự 25, 24, 23, ..., 3, 2, 1 dựa trên các số được chỉ thị trên Fig.15. Do các CG 24, 23, 22, 21, 20, 17, 16, 15, và 11 được định vị bên ngoài vùng 16×16 trên đỉnh bên trái, nên không thể biết xem liệu hệ số không bằng không chỉ tồn tại trong vùng 16×16 trên đỉnh bên trái hay không, chỉ bằng cách kiểm tra xem liệu LastSignificantCoeffX và LastSignificantCoeffY có tương ứng với vùng 16×16 trên đỉnh bên trái như được thể hiện trên bảng 6 hay không.

Do đó, khi thực hiện việc phân tích cú pháp cho việc lập mã phân dư, nếu việc kiểm tra xem liệu nhóm hệ số 4×4 tương ứng có thuộc về vùng 16×16 trên đỉnh

bên trái hay không, bất cứ khi nào hệ số biến đổi không bằng không được quét, thì trị số của `MtsZeroOutSigCoeffFlag` có thể được đặt bằng 0 chỉ trong trường hợp mà trong đó hệ số không bằng không tồn tại bên ngoài vùng 16 x 16 trên đỉnh bên trái. Trong trường hợp này, cú pháp lập mã phần dư là như được thể hiện trên bảng dưới đây.

[Bảng 8]

<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {</code>	Bộ mô tả
<code>if(((sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbWidth > 4)</code>	
<code>log2ZoTbWidth = 4</code>	
<code>else</code>	
<code>log2ZoTbWidth = Min(log2TbWidth, 5)</code>	
<code>if((sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbHeight > 4)</code>	
<code>log2ZoTbHeight = 4</code>	
<code>else</code>	
<code>log2ZoTbHeight = Min(log2TbHeight, 5)</code>	
<code>if(log2TbWidth > 0)</code>	
<code>last_sig_coeff_x_prefix</code>	<code>ae(v)</code>
<code>if(log2TbHeight > 0)</code>	
<code>last_sig_coeff_y_prefix</code>	<code>ae(v)</code>
<code>if(last_sig_coeff_x_prefix > 3)</code>	
<code>last_sig_coeff_x_suffix</code>	<code>ae(v)</code>

if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3) {	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 – log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 – log2SbH	
}	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight – (log2SbW + log2SbH))) – 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock – –	

}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth – log2SbW][log2TbHeight – log2SbH] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth – log2SbW][log2TbHeight – log2SbH] [lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
}.while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfstZeroOutSigCoeffFlag = 0	
if((LastSignificantCoeffX > 15 LastSignificantCoeffY > 15) && cIdx == 0)	
MtsZeroOutSigCoeffFlag = 0	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth – log2SbW][log2TbHeight – log2SbH] [i][0]	

$yS = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][i][1]$	
$\text{inferSbDcSigCoeffFlag} = 0$	
$\text{if}((i < \text{lastSubBlock}) \ \&\& \ (i > 0)) \{$	
coded_sub_block_flag $[xS][yS]$	$\text{ae}(v)$
$\text{inferSbDcSigCoeffFlag} = 1$	
$\}$	
$\text{firstSigScanPosSb} = \text{numSbCoeff}$	
$\text{lastSigScanPosSb} = -1$	
$\text{firstPosMode0} = (i == \text{lastSubBlock} ? \text{lastScanPos} : \text{numSbCoeff} - 1)$	
$\text{firstPosMode1} = -1$	
$\text{for}(n = \text{firstPosMode0}; n \geq 0 \ \&\& \ \text{remBinsPass1} \geq 4; n--) \{$	
$xC = (xS \ll \log_2\text{SbW}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][n][0]$	
$yC = (yS \ll \log_2\text{SbH}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][n][1]$	
$\text{if}(\text{coded_sub_block_flag}[xS][yS] \ \&\& \ (n > 0 \ \ !\text{inferSbDcSigCoeffFlag}) \ \&\& \ (xC \neq \text{LastSignificantCoeffX} \ \ yC \neq \text{LastSignificantCoeffY})) \{$	
sig_coeff_flag $[xC][yC]$	$\text{ae}(v)$
$\text{remBinsPass1}--$	
$\text{if}(\text{sig_coeff_flag}[xC][yC])$	
$\text{inferSbDcSigCoeffFlag} = 0$	
$\}$	
$\text{if}(\text{sig_coeff_flag}[xC][yC]) \{$	
$\text{if}((xC > 15 \ \ yC > 15) \ \&\& \ \text{cIdx} == 0)$	

MtsZeroOutSigCoeffFlag = 0	
abs_level_gtx_flag[n][0]	ae(v)
remBinsPass1--	
if(abs_level_gtx_flag[n][0]) {	
par_level_flag[n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag[n][1]	ae(v)
remBinsPass1--	
}	
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n][1]	
if(pic_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
if(remBinsPass1 < 4)	
firstPosModel = n - 1	
}	
.....	
}	
}	

Như được thể hiện trên bảng 8, bất cứ khi nào hệ số biến đổi không bằng không

được tìm thấy, thì việc xem liệu hệ số biến đổi không bằng không có được định vị trong vùng 16 x 16 trên đỉnh bên trái ($\text{if}((x_C > 15 \mid y_C > 15) \&\& \text{cIdx} == 0)$) có thể được kiểm tra, nhờ đó thiết lập trị số `MtsZeroOutSigCoeffFlag` tới 0. Ở đây, điều kiện này được kiểm tra khi trị số của `sig_coeff_flag[xC][yC]`, vốn là thông tin còn đang chỉ thị xem liệu hệ số biến đổi có tồn tại hay không, là 1 ($\text{if}(\text{sig_coeff_flag}[xC][yC])$), trong đó `xC` và `yC` theo cách tương ứng thể hiện tọa độ x và tọa độ y theo các đơn vị mẫu nằm trong khối biến đổi. Vị trí trên đỉnh bên trái của khối biến đổi là (0, 0).

Bảng 8 chứa điều kiện kiểm tra được hiện diện trong bảng 7 và cập nhật của `MtsZeroOutSigCoeffFlag` ($\text{if}((\text{LastSignificantCoeffX} > 15 \mid \text{LastSignificantCoeffY} > 15) \&\& \text{cIdx} == 0) \text{ MtsZeroOutSigCoeffFlag} = 0$). Nếu trị số của `sig_coeff_flag[xC][yC]` cho hệ số biến đổi không bằng không cuối cùng được suy ra là 1, thì điều kiện kiểm tra và việc cập nhật của `MtsZeroOutSigCoeffFlag` có thể được loại bỏ khỏi bảng 8.

Tóm lại, khi báo tín hiệu chỉ số MTS trong bảng 5 qua cấu hình cú pháp được thể hiện trên bảng 8, chỉ số MTS chỉ được báo hiệu khi hệ số biến đổi không bằng không tồn tại trong vùng 16 x 16 trên đỉnh bên trái của khối biến đổi độ chói là không quan tâm tới việc biến đổi sơ cấp nào được áp dụng. Khi chỉ số MTS không được báo hiệu, trị số của chỉ số MTS được suy ra như là 0, nghĩa là, (DCT-2, DCT-2) được áp dụng.

Như được mô tả ở trên, trong ba trường hợp dưới đây, việc biến đổi sơ cấp được áp dụng có thể được biết trước khi chỉ số MTS được báo hiệu.

- 1) Khi việc biến đổi khối con (subblock transform, SBT) được áp dụng
- 2) Khi việc phân vùng con nội (intra sub-partition, ISP) được áp dụng
- 3) Khi MTS ngầm được áp dụng

Trong trường hợp (1), do chỉ số MTS được đặt cấu hình để được báo hiệu chỉ khi trị số của `cu_sbt_flag` đang chỉ thị xem liệu SBT được áp dụng trong bảng 5 là 0 hay không, nghĩa là, do chỉ số MTS được báo hiệu chỉ khi SBT không được áp dụng, thì trường hợp mà trong đó SBT được áp dụng là không liên quan tới việc biến đổi của bảng 7.

Trong trường hợp (2), do chỉ số MTS được báo hiệu chỉ khi ISP không được áp dụng trong bảng 4 (điều kiện nếu `IntraSubPartitionsSplit[ x0 ][ y0 ] == ISP_NO_SPLIT` được kiểm tra khi báo hiệu chỉ số MTS trong bảng 5), trường hợp mà trong đó ISP được áp dụng cũng là không liên quan tới việc biến đổi của bảng 8.

Trong trường hợp (3), do chỉ số MTS được báo hiệu chỉ khi MTS rõ ràng được cho phép thực hiện trong bảng 5, trường hợp mà trong đó MTS ngầm được áp dụng là cũng không liên quan tới việc biến đổi của bảng 7. Nghĩa là, do chỉ số MTS được báo hiệu chỉ khi trị số của `sps_explicit_mts_intra_enabled_flag` là 1 trong MTS (liên MTS) được áp dụng cho khối phần dư được sinh ra bởi việc nội dự đoán và trị số của `sps_explicit_mts_inter_enabled_flag` là 1 trong MTS (liên MTS) được áp dụng cho khối phần dư được sinh ra bởi việc liên dự đoán, việc biến đổi của bảng 8 không ảnh hưởng tới trường hợp mà trong đó MTS ngầm được áp dụng.

Theo ví dụ khác, như được thể hiện trên bảng dưới đây, nó có thể được kiểm tra xem liệu hệ số không bằng không có tồn tại bên ngoài vùng 16 x 16 trên đỉnh bên trái hay không bởi đơn vị của CG.

[Bảng 9]

<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {</code>	Bộ mô tả
<code>if(((sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbWidth > 4)</code>	
<code>log2ZoTbWidth = 4</code>	

else	
$\log_2 \text{ZoTbWidth} = \text{Min}(\log_2 \text{TbWidth}, 5)$	
$\text{if}((\text{sps_mts_enabled_flag} \ \&\& \ \text{cu_sbt_flag} \ \&\& \\ \log_2 \text{TbWidth} < 6 \ \&\& \ \log_2 \text{TbHeight} < 6)) \\ \&\& \ \text{cIdx} == 0 \ \&\& \ \log_2 \text{TbHeight} > 4)$	
$\log_2 \text{ZoTbHeight} = 4$	
else	
$\log_2 \text{ZoTbHeight} = \text{Min}(\log_2 \text{TbHeight}, 5)$	
$\text{if}(\log_2 \text{TbWidth} > 0)$	
last_sig_coeff_x_prefix	ae(v)
$\text{if}(\log_2 \text{TbHeight} > 0)$	
last_sig_coeff_y_prefix	ae(v)
$\text{if}(\text{last_sig_coeff_x_prefix} > 3)$	
last_sig_coeff_x_suffix	ae(v)
$\text{if}(\text{last_sig_coeff_y_prefix} > 3)$	
last_sig_coeff_y_suffix	ae(v)
$\log_2 \text{TbWidth} = \log_2 \text{ZoTbWidth}$	
$\log_2 \text{TbHeight} = \log_2 \text{ZoTbHeight}$	
$\text{remBinsPass1} = ((1 \ll (\log_2 \text{TbWidth} + \log_2 \text{TbHeight})) * 7) \gg 2$	
$\log_2 \text{SbW} = (\text{Min}(\log_2 \text{TbWidth}, \log_2 \text{TbHeight}) < 2 \ ? \ 1 \ : \ 2)$	
$\log_2 \text{SbH} = \log_2 \text{SbW}$	
$\text{if}(\log_2 \text{TbWidth} + \log_2 \text{TbHeight} > 3) \{$	
$\text{if}(\log_2 \text{TbWidth} < 2) \{$	
$\log_2 \text{SbW} = \log_2 \text{TbWidth}$	

$\log_2\text{SbH} = 4 - \log_2\text{SbW}$	
} else if($\log_2\text{TbHeight} < 2$) {	
$\log_2\text{SbH} = \log_2\text{TbHeight}$	
$\log_2\text{SbW} = 4 - \log_2\text{SbH}$	
}	
}	
$\text{numSbCoeff} = 1 \ll (\log_2\text{SbW} + \log_2\text{SbH})$	
$\text{lastScanPos} = \text{numSbCoeff}$	
$\text{lastSubBlock} = (1 \ll (\log_2\text{TbWidth} + \log_2\text{TbHeight} - (\log_2\text{SbW} + \log_2\text{SbH}))) - 1$	
do {	
if($\text{lastScanPos} == 0$) {	
$\text{lastScanPos} = \text{numSbCoeff}$	
$\text{lastSubBlock}--$	
}	
$\text{lastScanPos}--$	
$\text{xS} = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][\text{lastSubBlock}][0]$	
$\text{yS} = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][\text{lastSubBlock}][1]$	
$\text{xC} = (\text{xS} \ll \log_2\text{SbW}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][\text{lastScanPos}][0]$	
$\text{yC} = (\text{yS} \ll \log_2\text{SbH}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][\text{lastScanPos}][1]$	
} while(($\text{xC} \neq \text{LastSignificantCoeffX}$) ($\text{yC} \neq \text{LastSignificantCoeffY}$))	

<pre> if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0) </pre>	
<pre>LfstDcOnly = 0</pre>	
<pre> if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight)) </pre>	
<pre>LfstZeroOutSigCoeffFlag = 0</pre>	
<pre> if((LastSignificantCoeffX > 15 LastSignificantCoeffY > 15) && cIdx == 0) </pre>	
<pre>MtsZeroOutSigCoeffFlag = 0</pre>	
<pre>QState = 0</pre>	
<pre>for(i = lastSubBlock; i >= 0; i--) {</pre>	
<pre>startQStateSb = QState</pre>	
<pre> xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [i][0] </pre>	
<pre> yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [i][1] </pre>	
<pre>inferSbDcSigCoeffFlag = 0</pre>	
<pre>if((i < lastSubBlock) && (i > 0)) {</pre>	
<pre>coded_sub_block_flag[xS][yS]</pre>	ae(v)
<pre>inferSbDcSigCoeffFlag = 1</pre>	
<pre>}</pre>	
<pre> if(coded_sub_block_flag[xS][yS] && ((xS << log2SbW) > 15 (yS << log2SbH) > 15) && cIdx == 0) </pre>	
<pre>MtsZeroOutSigCoeffFlag = 0</pre>	
<pre>firstSigScanPosSb = numSbCoeff</pre>	

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

lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n][1]	
if(pic_dep_quant_enabled_flag)	
QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
if(remBinsPass1 < 4)	
firstPosMode1 = n - 1	
}	
.....	
}	
}	

Trong bảng 9, xS và yS thể hiện vị trí tọa độ x và vị trí tọa độ y trong từng đơn vị CG bên trong khối biến đổi hiện tại, và cần phải được chuyển đổi thành các tọa độ trong các đơn vị mẫu để kiểm tra xem liệu hiệu số biến đổi có được định vị bên ngoài vùng 16 x 16 trên đỉnh bên trái hay không. Nghĩa là, xS và yS được chuyển đổi thành các tọa độ trong các đơn vị mẫu như trong $(xS \ll \log_2 SbW)$ và $(yS \ll \log_2 SbH)$. Ở đây, $\log_2 SbW$ và $\log_2 SbH$ theo cách tương ứng mô tả các trị số logarit cơ số 2 của chiều rộng và chiều cao của CG tương ứng, các tọa độ x tăng từ bên trái tới bên phải trong khối biến đổi, và các tọa độ y tăng từ trên đỉnh xuống dưới đáy. Các tọa độ của vị trí trên đỉnh bên trái của khối biến đổi là (0, 0).

Như được thể hiện trên bảng 9, theo việc quét bởi đơn vị của CG, khi hệ số biến đổi không bằng không tồn tại bên trong từng CG (khi trị số của coded_sub_block_flag[xS][yS] là 1) và CG được nhận diện là cần được định vị

bên ngoài vùng 16×16 trên đỉnh bên trái của khối biến đổi ($(xS \ll \log_2 SbW) > 15 \mid (yS \ll \log_2 SbH) > 15$)), thì trị số của `MtsZeroOutSigCoeffFlag` được đặt tới 0.

Như được thể hiện trên bảng 5, do trị số của biến `MtsZeroOutSigCoeffFlag` được khởi tạo tới 1, nên trị số của `MtsZeroOutSigCoeffFlag` được duy trì tại 1 trừ khi trị số của `MtsZeroOutSigCoeffFlag` được đặt là 0. Khi trị số của `coded_sub_block_flag[xS][yS]` cho CG mà hệ số biến đổi không bằng không cuối cùng thuộc vào đó được suy ra là 1, thì phần kiểm tra điều kiện trong bảng 7 và cập nhật của `MtsZeroOutSigCoeffFlag` (`if ( ( LastSignificantCoeffX > 15 | | LastSignificantCoeffY > 15 ) && cIdx == 0 ) MtsZeroOutSigCoeffFlag = 0`) có thể được loại bỏ khỏi bảng 9.

Các hình vẽ dưới đây được tạo ra để mô tả các ví dụ cụ thể của sáng chế này. Do các thuật ngữ cụ thể cho các thiết bị hoặc các thuật ngữ cụ thể cho các tín hiệu/các tín hiệu/các trường trong các hình vẽ được tạo ra để minh họa, nên các dấu hiệu kỹ thuật của sáng chế này không bị hạn chế vào các thuật ngữ cụ thể được sử dụng trong các hình vẽ dưới đây.

Fig.16 là lưu đồ minh họa việc vận hành của video thiết bị giải mã theo phương án thực hiện của sáng chế này.

Mỗi quy trình được bộc lộ trên Fig.16 là dựa trên một số các chi tiết được mô tả với tham khảo tới Fig.4 đến Fig.15. Do đó, phần mô tả của các chi tiết cụ thể chồng lấn với các phần đã được mô tả liên quan tới Fig.3 đến Fig.15 sẽ được bỏ qua hoặc sẽ được tạo ra một cách sơ lược.

Thiết bị giải mã 300 theo phương án thực hiện có thể nhận luồng bit chứa thông tin phần dư và có thể dẫn xuất thông tin phần dư, ví dụ, các hệ số biến đổi được lượng tử hóa, cho khối hiện tại, nghĩa là, khối biến đổi cần được biến đổi, từ luồng bit (S1610).

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 đích 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 đích có thể được chứa trong tập thông số trình tự (sequence parameter set, SPS) hoặc phần đầu phần đầu lát và có thể bao gồm ít nhất một thông tin về việc xem liệu RST có được áp dụng hay không, thông tin về yếu tố được làm giảm, thông tin về kích cỡ biến đổi tối thiểu để áp dụng RST, thông tin về kích cỡ biến đổi tối đa cho việc áp dụng RST, kích cỡ RST ngược, và thông tin về chỉ số biến đổi đang chỉ thị ma trận bất kỳ trong số các ma trận nhân biến đổi được chứa trong tập biến đổi.

Thiết bị giải mã 300 có thể dẫn xuất vị trí của hệ số có nghĩa cuối cùng trong khối hiện tại và các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư (S1620). Thiết bị giải mã 300 có thể dẫn xuất các hệ số biến đổi bằng cách giải lượng tử hóa các hệ số biến đổi được lượng tử hóa khối hiện tại.

Các hệ số biến đổi được dẫn xuất có thể được sắp xếp hai chiều trong khối hiện tại, và thiết bị giải mã có thể dẫn xuất thông tin về dữ liệu không bằng không, nghĩa là, hệ số có nghĩa không bằng không, trong khối hiện tại qua việc lập mã phần dư này. Nghĩa là, thiết bị giải mã có thể nhận diện thông tin vị trí cuối cùng về hệ số có nghĩa không bằng không trong khối hiện tại.

Các hệ số biến đổi được dẫn xuất dựa trên thông tin phần dư trong S1620 có thể là các hệ số biến đổi được giải lượng tử hóa như được mô tả ở trên hoặc có thể các hệ số biến đổi được lượng tử hóa. Nghĩa là, các hệ số biến đổi chỉ cần là dữ liệu cho việc nhận diện dữ liệu không bằng không trong khối hiện tại và vị trí của hệ số có nghĩa mà không quan tâm tới việc lượng tử hóa.

Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi đã được biến đổi bằng cách áp dụng LFNST cho các hệ số biến đổi hoặc có thể không thực hiện LFNST.

Thiết bị giải mã có thể dẫn xuất các mẫu phần dư bằng cách thực hiện việc biến

đổi sơ cấp ngược của các hệ số biến đổi hoặc các hệ số biến đổi đã được biến đổi. Thiết bị giải mã có thể sử dụng DCT-2 chung làm nhân biến đổi hoặc có thể áp dụng MTS nêu trên cho việc biến đổi sơ cấp ngược.

MTS có thể được thực hiện ngầm hoặc có thể được thực hiện dựa trên việc báo hiệu rõ ràng của chỉ số MTS.

Thiết bị giải mã có thể kiểm tra các điều kiện được xác định từ trước để phân tích cú pháp chỉ số MTS. Theo một ví dụ, thiết bị giải mã có thể phân tích cú pháp chỉ số MTS dựa trên hệ số có nghĩa đang không ở trong vùng thứ hai khác với vùng thứ nhất trên đỉnh bên trái của khối hiện tại (S1630).

Chỉ số MTS có thể được phân tích cú pháp bằng cách dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không, trong đó thông tin cờ có thể được dẫn xuất bằng cách xác định xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không bởi đơn vị của khối con quét của việc quét hệ số có nghĩa.

Thiết bị giải mã có thể thực hiện việc biến đổi sơ cấp ngược trên các hệ số biến đổi trong vùng thứ nhất trên đỉnh bên trái của khối hiện tại cần được biến đổi. Nghĩa là, vùng thứ hai khác với vùng thứ nhất trên đỉnh bên trái của khối hiện tại được đặt bằng không và do đó không chứa hệ số biến đổi.

Theo một ví dụ, khi MTS được áp dụng, thì vùng thứ nhất có thể là vùng 16 x 16 trên đỉnh bên trái của khối hiện tại, trong trường hợp đó vùng thứ hai có thể là vùng khác với vùng 16 x 16 trên đỉnh bên trái.

Khối con quét có thể là khối 4 x 4 và có thể được quét theo hướng quét đường chéo ngược từ vị trí của hệ số có nghĩa cuối cùng trong khối hiện tại như được thể hiện trên Fig.15.

Để phân tích cú pháp chỉ số MTS sau khi xác định xem liệu hệ số có nghĩa có

tồn tại trong vùng đặt bằng không hay không, thiết bị giải mã có thể dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không và có thể phân tích cú pháp chỉ số MTS dựa trên thông tin cờ.

Thông tin cờ có thể là biến `MtsZeroOutSigCoeffFlag`, và việc xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không có thể được xác định bởi đơn vị của khối con quét của việc quét hệ số có nghĩa.

Khi cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong khối con quét (`coded_sub_block_flag` or `sb_coded_flag`) được minh họa trên bảng 9 hay không là bằng với trị số 1 và hệ số có nghĩa đang tồn tại trong khối con quét được định vị trong vùng thứ hai ($((xS \ll \log_2 SbW) > 15 \mid (yS \ll \log_2 SbH) > 15)$), thì biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Theo cách khác, biến `MtsZeroOutSigCoeffFlag` có thể được dẫn xuất bằng cách kiểm tra xem liệu khối con quét 4x4 có thuộc về vùng thứ nhất hay không bất cứ khi nào quét các hệ số biến đổi như trong bảng 8. Nghĩa là, khi cờ đang chỉ thị xem liệu hệ số có nghĩa tồn tại trong khối con quét (`sig_coeff_flag`) là bằng với trị số 1 hay không và hệ số có nghĩa đang tồn tại trong khối con quét được định vị trong vùng thứ hai ($(xC > 15 \mid yC > 15)$), biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Theo cách khác, khi cờ đang chỉ thị xem liệu hệ số có nghĩa tồn tại trong khối con quét là bằng với trị số 1 hay không và khối con quét được định vị trong vùng thứ hai, thì biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Biến `MtsZeroOutSigCoeffFlag` có thể được đặt từ đầu tới 1, có thể được duy trì như là 1 khi hệ số có nghĩa không được định vị trong vùng thứ hai, và có thể được thay đổi thành 0 khi hệ số có nghĩa được định vị trong vùng thứ hai.

Tóm lại, thiết bị giải mã có thể phân tích cú pháp chỉ số MTS dựa trên thông tin

cờ đang chỉ thị rằng không có hệ số có nghĩa nào tồn tại trong vùng thứ hai và khi trị số của biến `MtsZeroOutSigCoeffFlag`, được đặt từ đầu tới 1, được duy trì, thì chỉ số MTS có thể được phân tích cú pháp.

Thiết bị giải mã có thể dẫn xuất các mẫu phân dư cho khôi hiện tại bằng cách áp dụng nhân biến đổi được dẫn xuất dựa trên chỉ số MTS cho các hệ số biến đổi trong vùng thứ nhất (S1640).

Tiếp theo, thiết bị giải mã 300 có thể sinh ra các mẫu được tái dựng dựa trên các mẫu phân dư cho khôi hiện tại và các mẫu dự đoán cho khôi hiện tại.

Các hình vẽ dưới đây được tạo ra để mô tả các ví dụ cụ thể của sáng chế này. Do các thuật ngữ cụ thể cho các thiết bị hoặc các thuật ngữ cụ thể cho các tín hiệu/các tin nhắn/các trường trong các hình vẽ được tạo ra để minh họa, nên các dấu hiệu kỹ thuật của sáng chế này không bị hạn chế vào các thuật ngữ cụ thể được sử dụng trong các hình vẽ dưới đây.

Fig.17 là lưu đồ minh họa việc vận hành của thiết bị mã hóa video theo phương án thực hiện của sáng chế này.

Mỗi quy trình được bộc lộ trên Fig.16 là dựa trên một số các chi tiết được mô tả với tham khảo tới Fig.4 đến Fig.15. Do đó, phần mô tả của các chi tiết cụ thể đang chồng lấn các phần đã được mô tả tham khảo tới Fig.2 và Fig.4 đến Fig.15 sẽ được bỏ qua hoặc được tạo ra một cách sơ lược.

Thiết bị mã hóa 100 theo phương án thực hiện có thể dẫn xuất các mẫu dự đoán dựa trên chế độ nội dự đoán được áp dụng cho khôi hiện tại (S1710).

Thiết bị mã hóa 100 theo phương án thực hiện có thể dẫn xuất các mẫu phân dư cho khôi hiện tại dựa trên các mẫu dự đoán (S1720).

Thiết bị mã hóa 100 theo phương án thực hiện có thể dẫn xuất các hệ số biến đổi cho khôi hiện tại bằng cách áp dụng MIS cho các mẫu phân dư (S1730).

Theo một ví dụ, thiết bị mã hóa có thể áp dụng MTS nêu trên cho việc biến đổi sơ cấp hoặc có thể sử dụng DCT-2 chung làm nhân biến đổi.

MTS có thể được thực hiện theo cách ngấm hoặc có thể được thực hiện dựa trên việc báo hiệu rõ ràng của chỉ số MTS.

Thiết bị mã hóa có thể xác định xem liệu có thực hiện MTS cho việc biến đổi sơ cấp hay không, và có thể dẫn xuất các hệ số biến đổi bằng cách áp dụng DCT-8 hoặc DST-7 cho các mẫu phần dư khi MTS được xác định là cần được thực hiện.

Theo một ví dụ, khi MTS được áp dụng cho việc biến đổi sơ cấp, thì thiết bị mã hóa có thể đặt bằng không vùng thứ hai khác với vùng thứ nhất trên đỉnh bên trái của khối hiện tại (S1740).

Theo một ví dụ, vùng thứ nhất có thể vùng 16 x 16 trên đỉnh bên trái của khối hiện tại, trong trường hợp đó vùng thứ hai có thể vùng khác với vùng 16 x 16 trên đỉnh bên trái.

Do việc đặt bằng không này, nên có thể làm giảm lượng tính toán được yêu cầu để thực hiện toàn bộ quy trình biến đổi và để làm giảm lượng vận hành được yêu cầu cho toàn bộ quy trình biến đổi, dẫn tới làm giảm việc tiêu thụ công suất cần thiết để thực hiện việc biến đổi. Hơn nữa, độ trễ được sinh ra trong quy trình biến đổi có thể được giảm, do đó tăng hiệu suất lập mã hình ảnh.

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi được biến đổi bằng cách tiếp tục áp dụng LFNST cho các hệ số biến đổi được dẫn xuất sau việc biến đổi sơ cấp.

Thiết bị mã hóa có thể dẫn xuất thông tin phần dư dựa trên các hệ số biến đổi hoặc các hệ số biến đổi đã được biến đổi cho khối hiện tại (S1750).

Nghĩa là, thiết bị mã hóa có thể sinh ra thông tin phần dư chứa thông tin trên các hệ số biến đổi đã được lượng tử hóa. Thông tin phần dư có thể bao gồm thành phần thông tin/cú pháp liên quan tới việc biến đổi trước đó. Thiết bị mã hóa có thể

mã hóa thông tin hình ảnh/video chứa thông tin phần dư và có thể đưa ra thông tin hình ảnh/video đã được mã hóa ở dạng của luồng bit.

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

Cũng vậy, thiết bị mã hóa có thể đặt cấu hình thông tin hình ảnh sao cho chỉ số MTS đang chỉ thị nhân biến đổi của MTS được phân tích cú pháp dựa trên việc xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không (S1760).

Hơn nữa, thiết bị mã hóa có thể xây dựng thông tin hình ảnh để dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không theo đơn vị của khối con quét của việc quét hệ số có nghĩa.

Nghĩa là, thiết bị mã hóa có thể xây dựng thông tin hình ảnh sao cho thông tin hình ảnh được minh họa trên bảng 8 hoặc bảng 9 có thể được phân tích cú pháp trong thiết bị giải mã.

Theo một ví dụ, khối con quét có thể là khối 4 x 4 và có thể được quét theo hướng quét đường chéo ngược từ vị trí của hệ số có nghĩa cuối cùng trong khối hiện tại cũng trong thiết bị giải mã như được thể hiện trên Fig.15.

Thông tin cờ có thể là biến `MtsZeroOutSigCoeffFlag`, và việc xem liệu hệ số có nghĩa có tồn tại trong vùng thứ hai hay không có thể được xác định bởi đơn vị của khối con quét của việc quét hệ số có nghĩa.

Khi cờ đang chỉ thị việc xem liệu hệ số có nghĩa có tồn tại trong khối con quét (`coded_sub_block_flag`) hay không được minh họa trên bảng 9 là bằng với trị số 1 và hệ số có nghĩa đang tồn tại trong khối con quét được định vị trong vùng thứ hai ($((xS \ll \log_2 SbW) > 15 \mid (yS \ll \log_2 SbH) > 15))$), thì biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Theo cách khác, biến `MtsZeroOutSigCoeffFlag` có thể được dẫn xuất bằng cách kiểm tra xem liệu khối con quét 4x4 có thuộc về vùng thứ nhất hay không bất cứ khi nào việc quét các hệ số biến đổi như trong bảng 8. Nghĩa là, khi cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong khối con quét (`sig_coeff_flag`) hay không là bằng với trị số 1 và hệ số có nghĩa đang tồn tại trong khối con quét được định vị trong vùng thứ hai (`if( ( xC > 15 || yC > 15 )`), thì biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Theo cách khác, khi cờ đang chỉ thị xem liệu hệ số có nghĩa có tồn tại trong khối con quét hay không là bằng với trị số 1 và khối con quét được định vị trong vùng thứ hai, thì biến `MtsZeroOutSigCoeffFlag` có thể chỉ thị rằng hệ số có nghĩa tồn tại trong vùng thứ hai.

Biến `MtsZeroOutSigCoeffFlag` có thể được đặt từ đầu tới 1, có thể được duy trì là 1 khi hệ số có nghĩa không được định vị trong vùng thứ hai, và có thể được thay đổi thành 0 khi hệ số có nghĩa được định vị trong vùng thứ hai.

Thiết bị mã hóa có thể mã hóa thông tin phần dư được dẫn xuất dựa trên các hệ số biến đổi, và chỉ số MTS dựa trên thông tin cờ đang chỉ thị rằng không có hệ số có nghĩa tồn tại trong vùng thứ hai và sau đó có thể đưa chúng ra.

Theo sáng chế này, ít nhất một việc trong số lượng tử hóa/giải lượng tử hóa và/hoặc biến đổi/biến đổi ngược có thể được bỏ qua. Khi việc lượng tử hóa/giải lượng tử hóa được bỏ qua, thì hệ số biến đổi được lượng tử hóa có thể được đề cập tới như là hệ số biến đổi. Khi việc biến đổi/biến đổi ngược được bỏ qua, thì hệ số biến đổi có thể được đề cập tới như là hệ số hoặc hệ số còn dư, hoặc có thể vẫn được đề cập tới như là hệ số biến đổi để tạo thống nhất cho việc biểu diễn.

Thêm vào đó, theo sáng chế này, hệ số biến đổi được lượng tử hóa và hệ số biến đổi có thể được đề cập tới như là hệ số biến đổi và hệ số biến đổi được định tỉ lệ, theo cách tương ứng. Trong trường hợp này, thông tin phần dư có thể bao gồm thông tin

về hệ số biến đổi (các hệ số biến đổi), và thông tin về hệ số biến đổi (các hệ số biến đổi) có thể được báo hiệu 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 hệ số biến đổi (các hệ số biến đổi)), và các hệ số biến đổi được định tỉ lệ có thể được dẫn xuất qua việc biến đổi ngược (việc định tỉ lệ) của các hệ số biến đổi. Các mẫu phần dư có thể được dẫn xuất dựa trên việc biến đổi ngược (việc biến đổi) của các hệ số biến đổi được định tỉ lệ. Các chi tiết này có thể cũng được áp dụng/được biểu diễn trong các phần khác của sáng chế này.

Theo các phương án thực hiện được mô tả ở trên, các phương pháp được giải thích trên cơ sở các lưu đồ nhờ các phương tiện của các chuỗi của các bước và các khối, nhưng sáng chế này không bị hạn chế vào thứ tự của các bước, và bước cụ thể có thể được thực hiện theo thứ tự hoặc bước khác với phần đã được mô tả, hoặc đồng thời với bước khác. Hơn nữa, người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng có thể hiểu rằng các bước được thể hiện trên lưu đồ là không có tính loại trừ, và rằng bước khác là có thể được hợp nhất hoặc một hoặc nhiều bước của lưu đồ là có thể được loại bỏ mà không ảnh hưởng đến phạm vi của sáng chế này.

Các phương pháp nêu trên theo sáng chế này có thể được áp dụng ở dạng phần mềm, và thiết bị mã hóa và/hoặc thiết bị giải mã theo phần bộc lộ có thể được chứa trong thiết bị để xử lý hình ảnh, như TV, máy tính, điện thoại thông minh, hộp giải mã tín hiệu truyền hình, thiết bị hiển thị hoặc dạng tương tự.

Khi các phương án thực hiện theo sáng chế được biểu hiện bởi phần mềm, thì các phương pháp được mô tả ở trên có thể được biểu hiện dưới dạng các môđun (các quy trình, các chức năng hoặc dạng tương tự) để thực hiện các chức năng được mô tả ở trên. Các môđun có thể được lưu trong bộ nhớ và có thể được thực thi bởi bộ xử lý. Bộ nhớ có thể ở bên trong hoặc bên ngoài bộ xử lý và có thể được kết nối tới bộ xử lý theo nhiều cách đã biết rõ khác nhau. Bộ xử lý có thể bao gồm mạch tích hợp ứng

dụng cụ thể (application-specific integrated circuit, ASIC), chipset khác, mạch logic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ có thể bao 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, vật ghi, và/hoặc thiết bị lưu trữ khác. Nghĩa là, các phương án thực hiện được mô tả trong sáng chế này có thể được ứng dụng và được thực hiện bởi bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip. Ví dụ, các đơn vị chức năng được thể hiện trên mỗi hình vẽ là có thể được áp dụng và thực hiện trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip.

Hơn nữa, thiết bị giải mã và thiết bị mã hoá mà sáng chế được áp dụng vào đó có thể được chứa trong bộ thu phát truyền quảng bá đa phương tiện, thiết bị đầu cuối truyền thông di động, thiết bị video rạp chiếu phim gia đình, thiết bị video rạp chiếu phim kỹ thuật số, camera giám sát, thiết bị chat video, thiết bị truyền thông thời gian thực như truyền thông video, thiết bị tạo luồng di động, phương tiện lưu trữ, máy ghi hình cầm tay, thiết bị cung cấp dịch vụ video theo yêu cầu (Video on Demand, VoD), thiết bị video của dịch vụ cung cấp nội dung trên nền mạng viễn thông (Over The Top, OTT), thiết bị cung cấp dịch vụ phát luồng Internet, thiết bị video ba chiều (Three Dimensional, 3D), thiết bị điện thoại video, và thiết bị video y tế, và có thể được dùng để xử lý tín hiệu video hoặc tín hiệu dữ liệu. Ví dụ, thiết bị video của dịch vụ cung cấp nội dung trên nền mạng viễn thông (OTT) có thể gồm máy chơi trò chơi, máy chơi đĩa Blu-ray, tivi truy cập Internet, hệ thống rạp hát gia đình, điện thoại thông minh, máy tính bảng, đầu ghi video kỹ thuật số (Digital Video Recorder, DVR), và dạng tương tự.v

Thêm vào đó, phương pháp xử lý mà sáng chế được áp dụng vào đó là có thể được tạo ra dưới dạng chương trình được thực thi bởi máy tính, và được lưu trữ trên phương tiện lưu trữ đọc được bởi máy tính. Dữ liệu đa phương tiện mà có cấu trúc dữ liệu theo sáng chế thì cũng có thể được lưu trữ trong phương tiện lưu trữ đọc được bởi máy tính. Phương tiện lưu trữ đọc được bởi máy tính chứa tất cả các loại thiết bị

lưu trữ và các thiết bị lưu trữ phân tán mà dữ liệu đọc được bởi máy tính được lưu trữ trong đó. Phương tiện lưu trữ đọc được bởi máy tính có thể bao gồm, ví dụ, đĩa Blu-ray (Blu-ray Disc, BD), bus nối tiếp vạn năng (universal serial bus, USB), ROM, PROM, EPROM, EEPROM, RAM, CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang học. Thêm nữa, phương tiện lưu trữ đọc được bởi máy tính chứa phương tiện được áp dụng ở dạng của sóng mang (ví dụ, việc truyền qua Internet). Thêm vào đó, luồng bit được sinh ra bởi phương pháp mã hóa này là có thể được lưu trữ trên phương tiện ghi đọc được bởi máy tính hoặc được truyền thông qua mạng truyền thông có dây hoặc không dây. Thêm vào đó, các phương án thực hiện của sáng chế là có thể được thực hiện dưới dạng sản phẩm chương trình máy tính bởi các mã chương trình, và các mã chương trình này có thể được thực thi trên máy tính bởi các phương án thực hiện của sáng chế này. Các mã chương trình có thể được lưu trữ trên vật mang đọc được bởi máy tính.

Fig.18 minh họa cấu trúc của hệ thống tạo luồng nội dung mà sáng chế này được áp dụng vào đó.

Hơn nữa, hệ thống tạo luồng nội dung mà sáng chế được áp dụng vào đó là có thể về quy mô lớn gồm máy chủ mã hoá, máy chủ tạo luồng, máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị đầu vào đa phương tiện.

Máy chủ mã hóa có chức năng nén thành các dữ liệu kỹ thuật số, các nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện, như là điện thoại thông minh, camera, máy ghi hình cầm tay, và dạng tương tự, để sinh ra luồng bit, và truyền nó đến máy chủ tạo luồng. Như là một ví dụ khác, trong trường hợp mà tại đó thiết bị đầu vào đa phương tiện, như là điện thoại thông minh, camera, máy ghi hình cầm tay, hoặc tương tự, trực tiếp sinh ra luồng bit, thì máy chủ mã hóa có thể được bỏ qua. Luồng bit có thể được sinh ra bởi phương pháp mã hóa hoặc phương pháp sinh ra

luồng bit mà sáng chế này được áp dụng vào đó. Và máy chủ tạo luồng có thể tạm thời lưu trữ luồng bit này trong suốt quá trình truyền hoặc nhận luồng bit này.

Máy chủ tạo luồng truyền dữ liệu đa phương tiện đến thiết bị người dùng dựa trên cơ sở yêu cầu của người dùng thông qua máy chủ web, có chức năng như trang thiết bị để thông báo cho người dùng biết có dịch vụ gì. Khi người dùng yêu cầu dịch vụ mà mình cần, thì máy chủ web truyền yêu cầu đó đến máy chủ tạo luồng, và máy chủ tạo luồng truyền dữ liệu đa phương tiện đến người dùng. Liên quan tới vấn đề này, thì hệ thống tạo luồng nội dung có thể bao gồm máy chủ điều khiển tách rời, và trong trường hợp này, thì máy chủ điều khiển này có chức năng điều khiển các lệnh/các phản hồi giữa các thiết bị tương ứng trong hệ thống tạo luồng nội dung này.

Máy chủ tạo luồng có thể nhận các nội dung từ bộ lưu trữ phương tiện và/hoặc máy chủ mã hoá. Ví dụ, trong trường hợp các nội dung này được nhận từ máy chủ mã hóa, thì các nội dung này có thể nhận được theo thời gian thực. Trong trường hợp này, máy chủ tạo luồng có thể lưu trữ luồng bit này trong một khoảng thời gian định trước để cung cấp dịch vụ tạo luồng một cách 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 truyền quảng bá kỹ thuật số, thiết bị trợ giúp cá nhân kỹ thuật số (Personal Digital Assistant, PDA), máy chơi đa phương tiện di động (Portable Multimedia Player, PMP), thiết bị dẫn đường, máy tính bảng lai laptop, máy tính bảng, máy tính xách tay siêu mỏng nhẹ, thiết bị đeo được (ví dụ, thiết bị đầu cuối kiểu đồng hồ (đồng hồ thông minh), thiết bị đầu cuối kiểu kính mắt (kính thông minh), thiết bị hiển thị gắn trên đầu (Head Mounted Display, HMD)), tivi kỹ thuật số, máy tính để bàn, biển chỉ dẫn kỹ thuật số, hoặc dạng tương tự. Mỗi máy chủ trong số các máy chủ trong hệ thống tạo luồng nội dung này có thể được vận hành như là máy chủ phân tán, và trong trường hợp này, thì dữ liệu nhận được bởi mỗi máy chủ có thể được xử lý theo cách phân tán.

Các yêu cầu bảo hộ được bộc lộ ở đây có thể được kết hợp theo cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật của các yêu cầu bảo hộ dạng phương pháp của sáng chế có thể được kết hợp để được triển khai hoặc được thực hiện trong thiết bị, và các dấu hiệu kỹ thuật của các yêu cầu bảo hộ dạng thiết bị có thể được kết hợp để được triển khai hoặc được thực hiện trong phương pháp. Hơn nữa, các dấu hiệu kỹ thuật các yêu cầu bảo hộ dạng phương pháp và các yêu cầu bảo hộ dạng thiết bị có thể được kết hợp để được triển khai hoặc được thực hiện trong thiết bị, và các dấu hiệu kỹ thuật của các yêu cầu bảo hộ dạng phương pháp và các yêu cầu bảo hộ dạng thiết bị có thể được kết hợp để được triển khai hoặc được thực hiện trong phương pháp.

YÊU CẦU BẢO HỘ

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

thu được thông tin phần dư từ luồng bit;

dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư; và

dẫn xuất các mẫu phần dư cho khối hiện tại bằng cách áp dụng việc biến đổi sơ cấp ngược vào các hệ số biến đổi dựa trên chỉ số chọn nhiều biến đổi (multiple transform selection, MTS);

trong đó bước dẫn xuất hệ số biến đổi bao gồm:

dẫn xuất xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không bằng cách quét khối con quét theo đơn vị khối con quét thiết đặt trước theo thứ tự quét theo đường chéo ngược; và

dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong vùng thứ hai hay không dựa trên (i) việc liệu hệ số có nghĩa là có mặt tại khối con quét hay không, (ii) việc liệu khối con quét có thuộc về vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái của khối hiện tại hay không, và (iii) việc liệu chỉ số màu sắc của khối hiện tại là bằng 0 hay không, và

chỉ số MTS được phân tích cú pháp dựa trên thông tin cờ đang chỉ thị rằng hệ số có nghĩa là không có mặt trong vùng thứ hai.

2. Phương pháp giải mã hình ảnh theo điểm 1, trong đó chỉ số MTS được suy ra là 0 khi chỉ số MTS không được phân tích cú pháp.

3. Phương pháp giải mã hình ảnh theo điểm 1, trong đó bước dẫn xuất xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không bao gồm việc phân tích cú pháp cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không.

4. Phương pháp giải mã hình ảnh theo điểm 1, trong đó thông tin cờ ban đầu được đặt là 1, và

trong đó, khi hệ số có nghĩa được định vị trong vùng thứ hai, thì thông tin cờ được thay đổi thành 0.

5. Phương pháp giải mã hình ảnh theo điểm 1, trong đó khối con quét là khối 4×4 , và

trong đó khối con quét được quét theo thứ tự quét theo đường chéo ngược từ khối con quét cuối cùng chứa hệ số có nghĩa cuối cùng trong khối hiện tại.

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

dẫn xuất các mẫu dự đoán cho khối hiện tại;

dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán;

dẫn xuất các hệ số biến đổi cho khối hiện tại bằng cách áp dụng việc biến đổi sơ cấp vào các mẫu phần dư;

dẫn xuất thông tin phần dư dựa trên hệ số biến đổi của vùng 16×16 trên đỉnh bên trái của khối hiện tại; và

xây dựng thông tin hình ảnh để phân tích cú pháp chỉ số chọn nhiều biến đổi (multiple transform selection, MTS) đang chỉ thị nhân biến đổi sơ cấp dựa trên hệ số có nghĩa đang có mặt trong vùng thứ hai khác với vùng 16×16 trên đỉnh bên trái,

trong đó bước xây dựng thông tin hình ảnh bao gồm:

mã hóa cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không bằng cách quét khối con quét theo đơn vị khối con quét thiết đặt trước theo thứ tự quét theo đường chéo ngược; và

dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong vùng thứ hai hay không dựa trên (i) việc liệu hệ số có nghĩa là có mặt tại khối con quét hay không, (ii) việc liệu khối con quét có thuộc về vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái của khối hiện tại hay không, và (iii) việc liệu chỉ số màu sắc của khối hiện tại là bằng 0 hay không, và

mã hóa chỉ số MTS dựa trên thông tin cờ đang chỉ thị rằng hệ số có nghĩa là không có mặt trong vùng thứ hai.

7. Phương pháp mã hóa hình ảnh theo điểm 6, trong đó chỉ số MTS được suy ra là 0 khi chỉ số MTS không được mã hóa.

8. Phương pháp mã hóa theo điểm 6, trong đó thông tin cờ ban đầu được đặt là 1, và trong đó, khi hệ số có nghĩa được định vị trong vùng thứ hai, thì thông tin cờ được thay đổi thành 0.

9. Phương pháp mã hóa theo điểm 7, trong đó khối con quét là khối 4 x 4, và

trong đó khối con quét được quét theo thứ tự quét theo đường chéo ngược từ khối con quét cuối cùng chứa hệ số có nghĩa cuối cùng trong khối hiện tại.

10. Phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính lưu trữ luồng bit được sinh ra bởi phương pháp, phương pháp này bao gồm các bước:

dẫn xuất các mẫu dự đoán cho khối hiện tại;

dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán;

dẫn xuất các hệ số biến đổi cho khối hiện tại bằng cách áp dụng việc biến đổi sơ cấp vào các mẫu phần dư;

dẫn xuất thông tin phần dư dựa trên hệ số biến đổi của vùng 16x16 trên đỉnh bên trái của khối hiện tại; và

xây dựng thông tin hình ảnh để phân tích cú pháp chỉ số chọn nhiều biến đổi (multiple transform selection, MTS) đang chỉ thị nhân biến đổi sơ cấp dựa trên hệ số có nghĩa đang có mặt trong vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái,

trong đó bước xây dựng thông tin hình ảnh bao gồm:

mã hóa cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không bằng cách quét khối con quét theo đơn vị khối con quét thiết đặt trước theo thứ tự quét theo đường chéo ngược; và

dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong vùng thứ hai hay không dựa trên (i) việc liệu hệ số có nghĩa là có mặt tại khối con quét hay không, (ii) việc liệu khối con quét có thuộc về vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái của khối hiện tại hay không, và (iii) việc liệu chỉ số màu sắc của khối hiện tại là bằng 0 hay không, và

mã hóa chỉ số MTS dựa trên thông tin cờ đang chỉ thị rằng hệ số có nghĩa là không có mặt trong vùng thứ hai.

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

thu được luồng bit cho hình ảnh, trong đó luồng bit được sinh ra dựa trên việc dẫn xuất các mẫu dự đoán cho khối hiện tại; dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán; dẫn xuất các hệ số biến đổi cho khối hiện tại bằng cách áp dụng việc biến đổi sơ cấp vào các mẫu phần dư; dẫn xuất thông tin phần dư dựa trên hệ số biến đổi của vùng 16x16 trên đỉnh bên trái của khối hiện tại; và xây dựng thông tin hình ảnh để phân tích cú pháp chỉ số chọn nhiều biến đổi (multiple transform selection, MTS) đang chỉ thị nhân biến đổi sơ cấp dựa trên hệ số có nghĩa đang có mặt trong vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái; và

truyền dữ liệu bao gồm luồng bit,

trong đó bước xây dựng thông tin hình ảnh bao gồm:

mã hóa cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong khối con quét hay không bằng cách quét khối con quét theo đơn vị khối con quét thiết đặt trước theo thứ tự quét theo đường chéo ngược; và

dẫn xuất thông tin cờ đang chỉ thị xem liệu hệ số có nghĩa là có mặt trong vùng thứ hai hay không dựa trên (i) việc liệu hệ số có nghĩa là có mặt tại khối con quét hay không, (ii) việc liệu khối con quét có thuộc về vùng thứ hai khác với vùng 16x16 trên đỉnh bên trái của khối hiện tại hay không, và (iii) việc liệu chỉ số màu sắc của khối hiện tại là bằng 0 hay không, và

mã hóa chỉ số MTS dựa trên thông tin cờ đang chỉ thị rằng hệ số có nghĩa là không có mặt trong vùng thứ hai.

FIG. 1

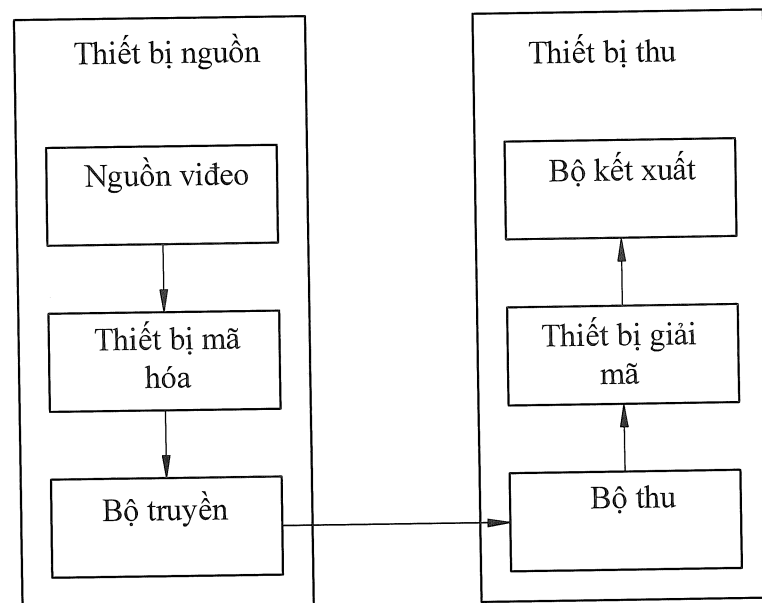
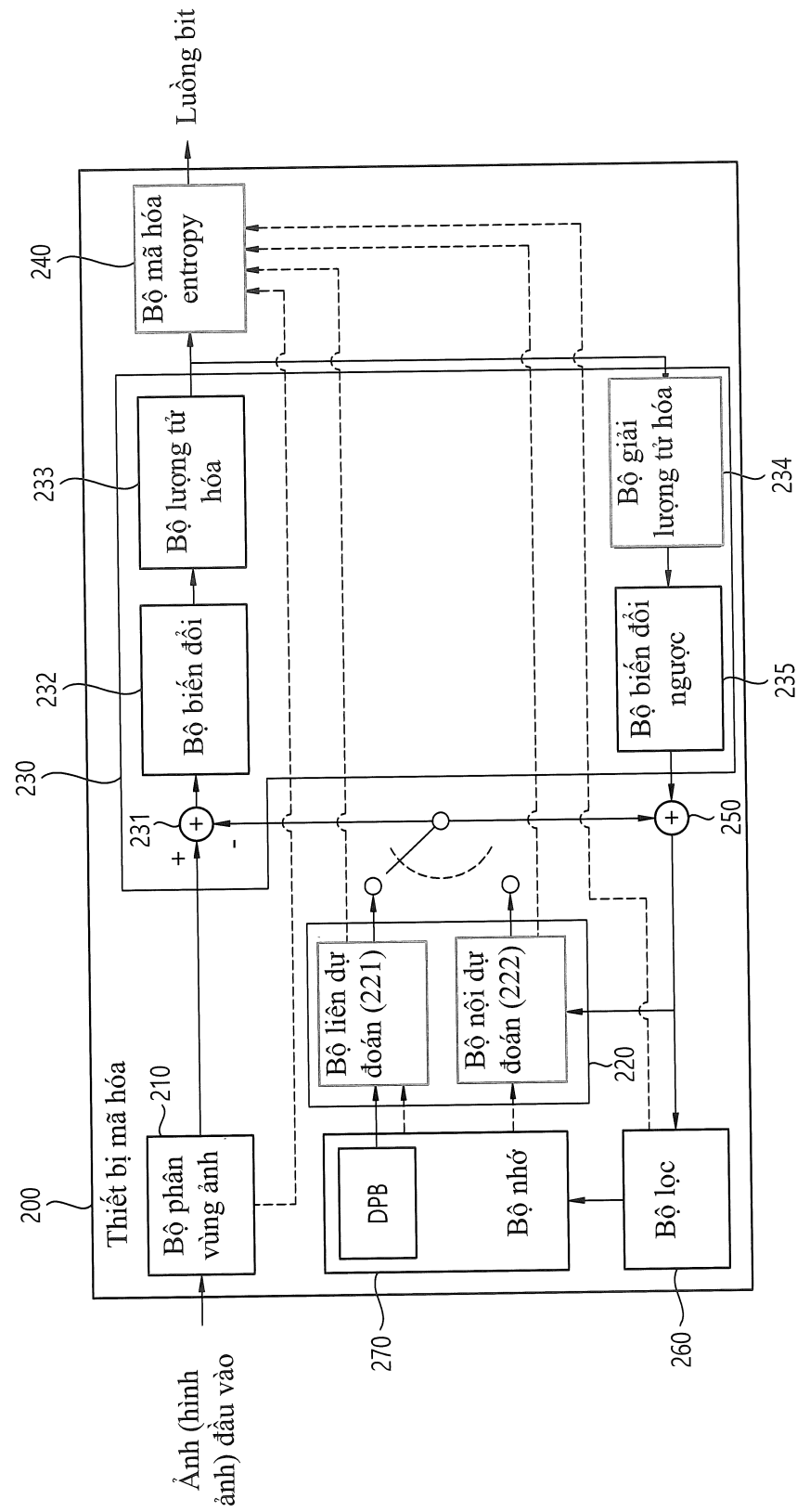


FIG. 2



3/18

FIG. 3

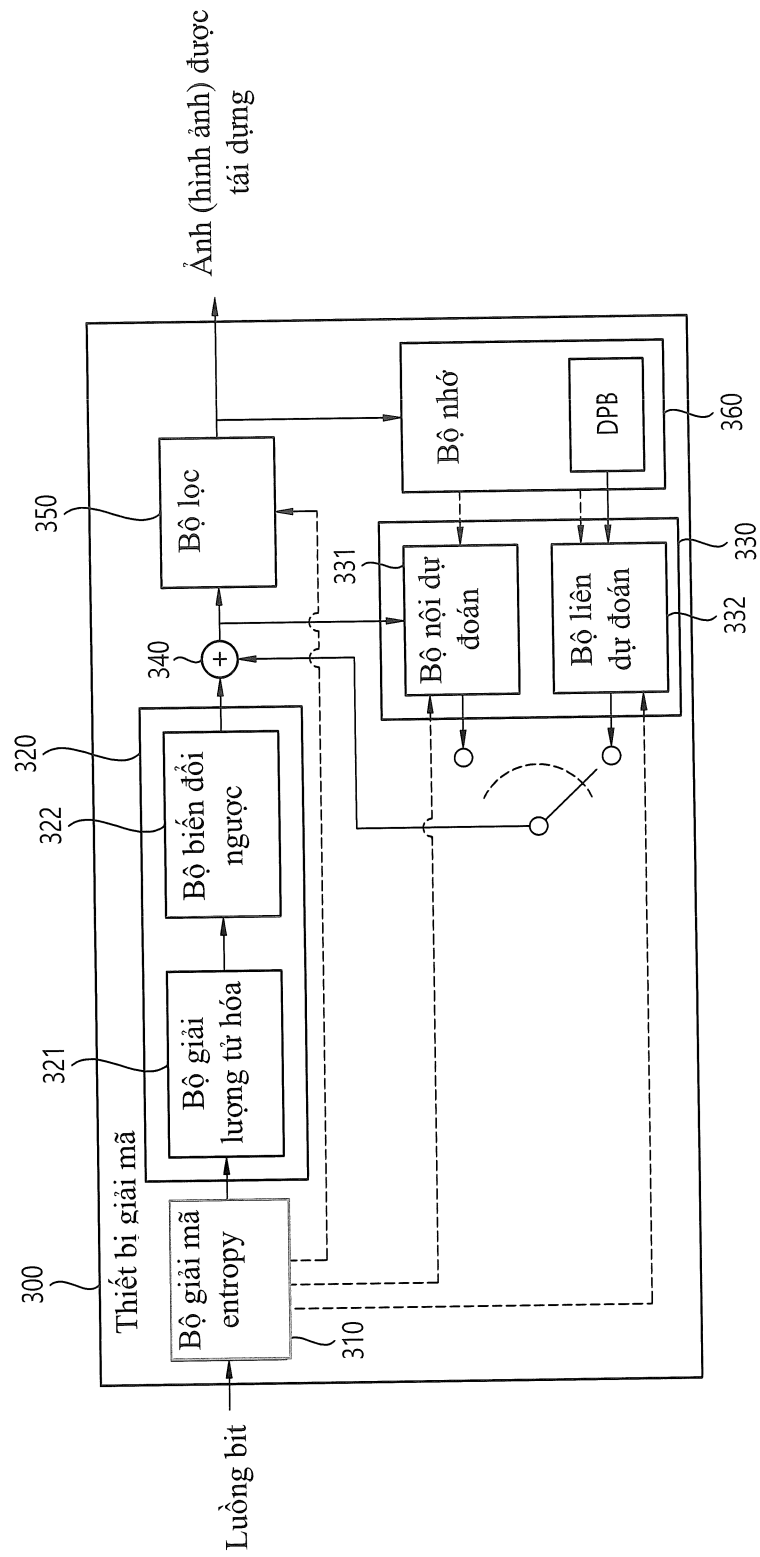
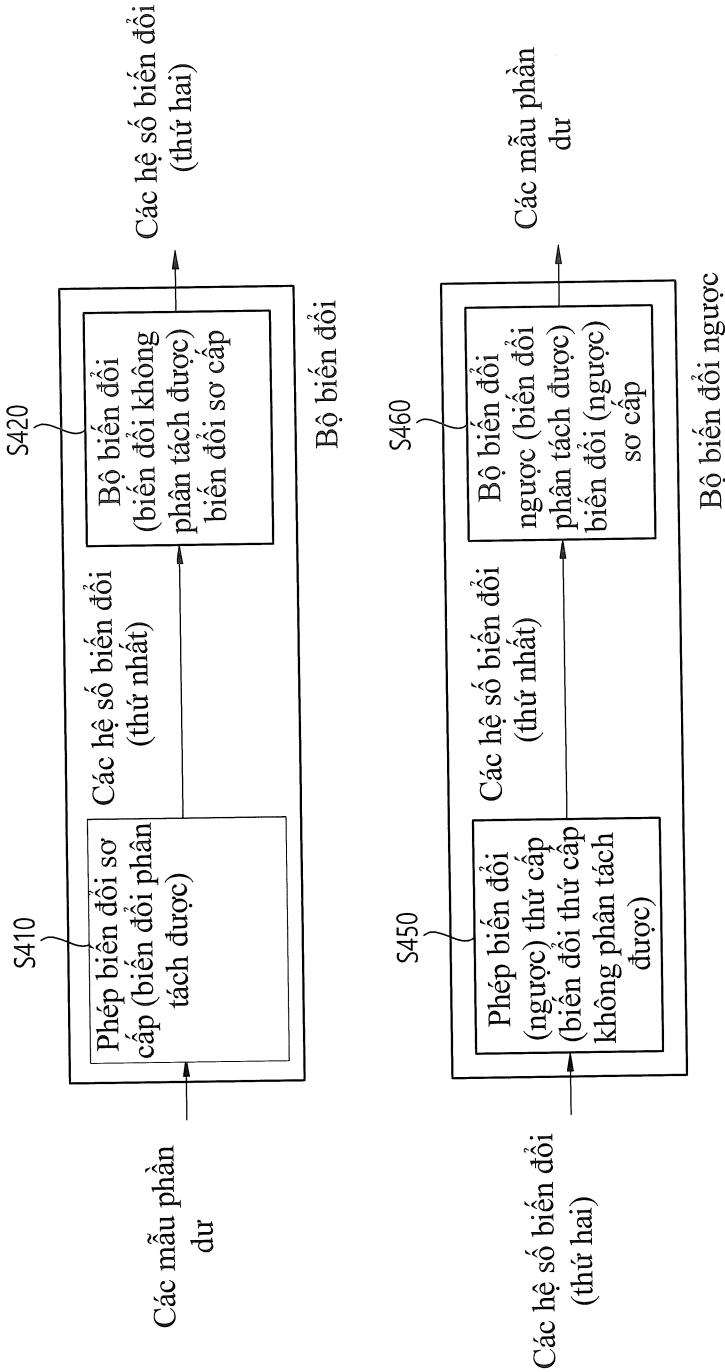


FIG. 4



5/18

FIG. 5

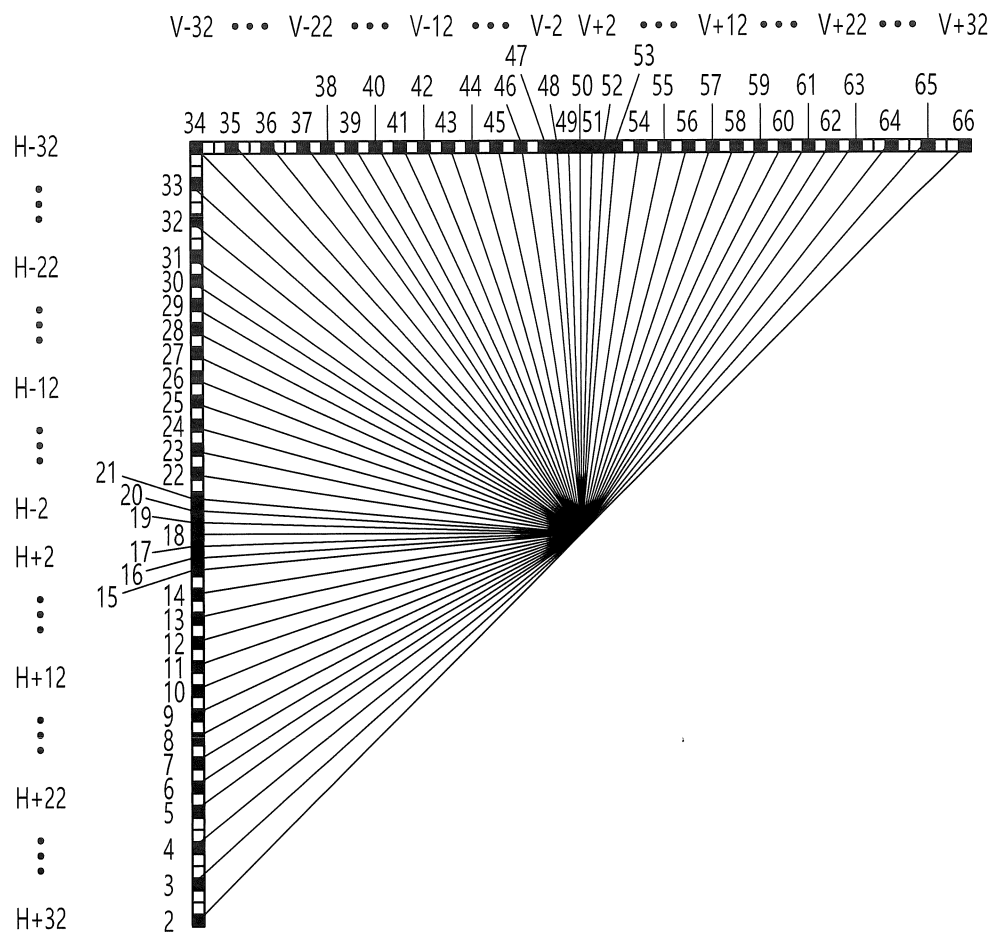


FIG. 6

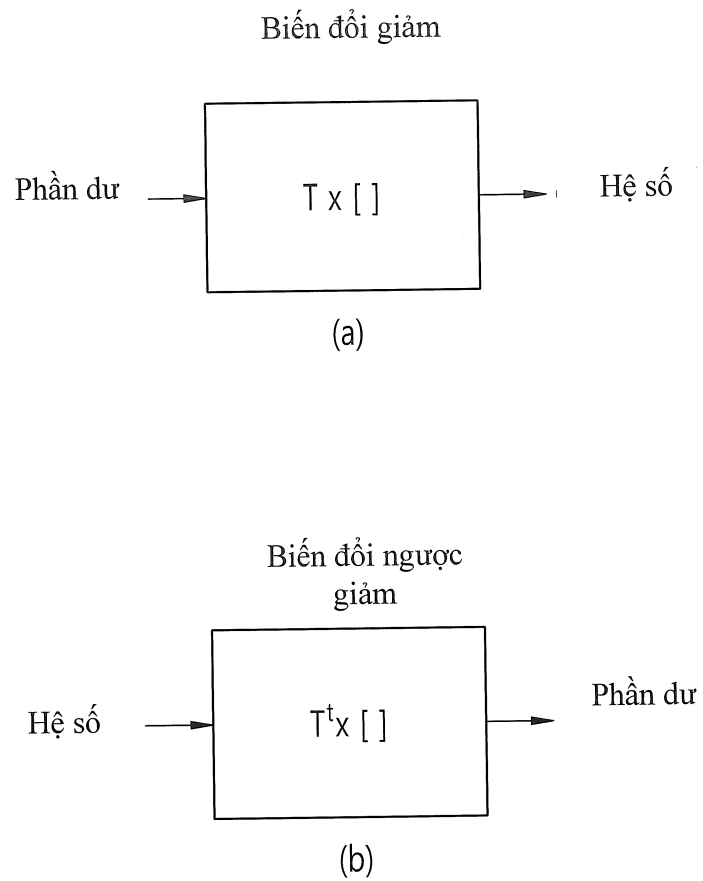


FIG. 7

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. 8

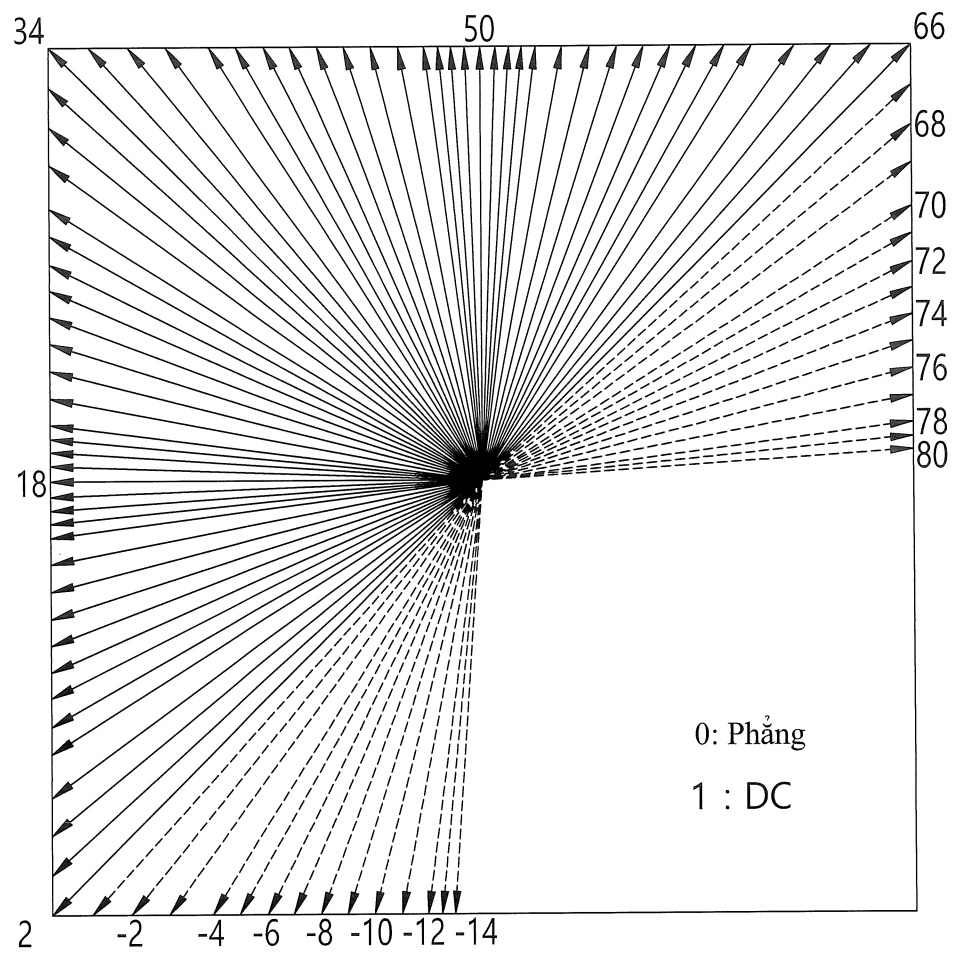
1	3	6	10
2	5	9	13
4	8	12	15
7	11	14	16

(a)

1	3	6	
2	5		
4	8		
7			

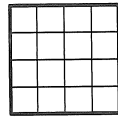
(b)

FIG. 9

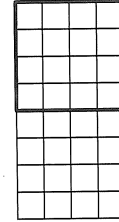
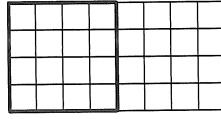


10/18

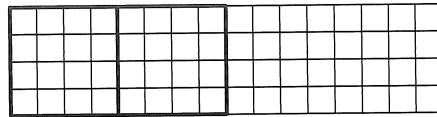
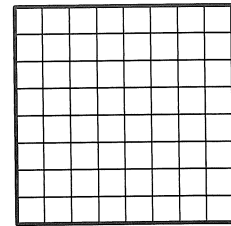
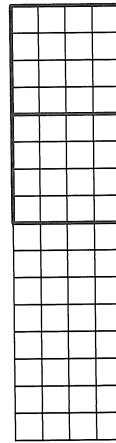
FIG. 10



(a) 4x4



(b) 8x4 / 4x8

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

(d) 8x8

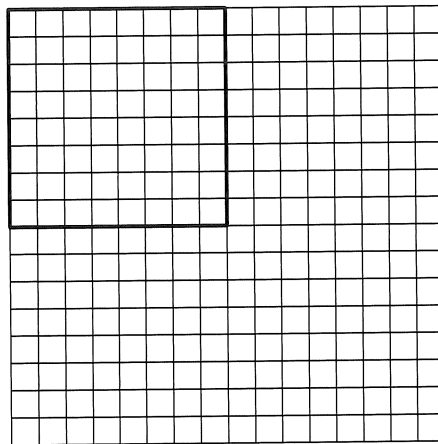
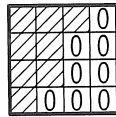
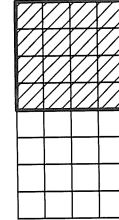
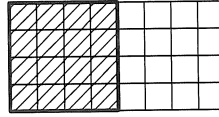
(e) $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

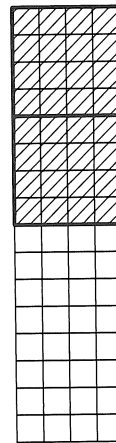
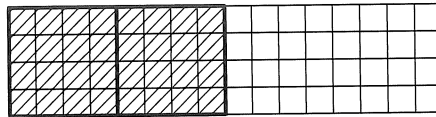
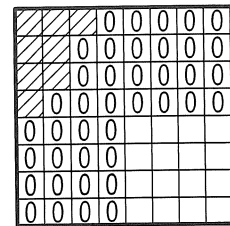
FIG. 11



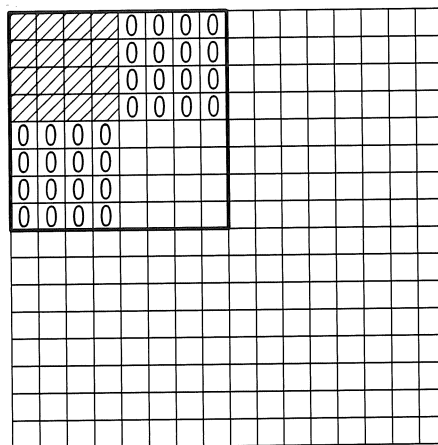
(a) 4x4



(b) 8x4 / 4x8

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

(d) 8x8

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

12/18

FIG. 12

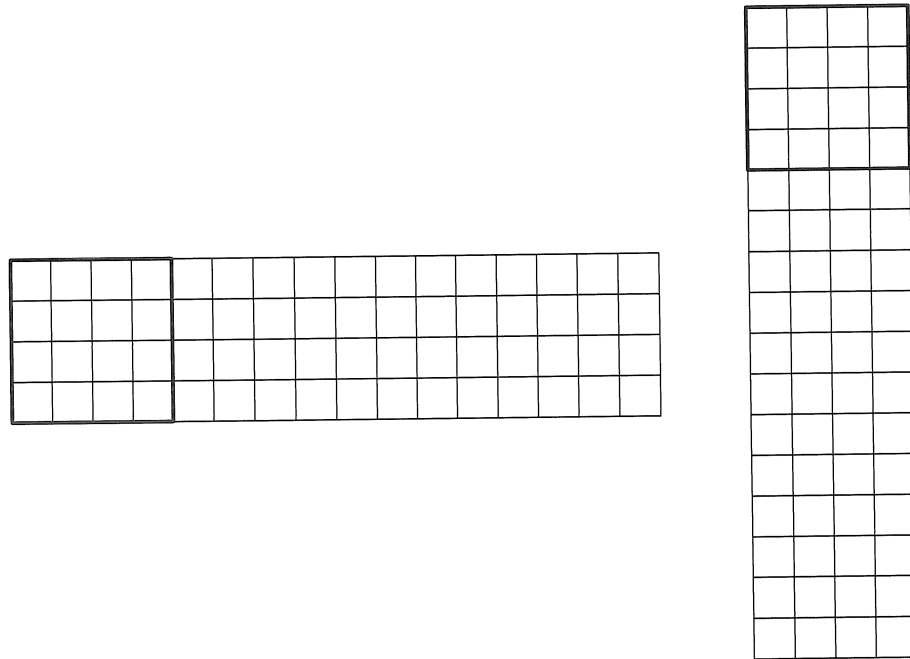
 $4 \times N / N \times 4, \text{ khi } N \geq 16$

FIG. 13

			0
		0	0
		0	0
0	0	0	

(a) 4x4

				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

(b) 8x4 / 4x8

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

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

				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

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
0	0	0	0

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

			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

1	3	6	10	15	21	28	36
2	5	9	14	20	27	35	43
4	8	13	19	26	34	42	49
7	12	18	25	33	41	48	54
11	17	24	32	40	47	53	58
16	23	31	39	46	52	57	61
22	30	38	45	51	56	60	63
29	37	44	50	55	59	62	64

FIG. 16

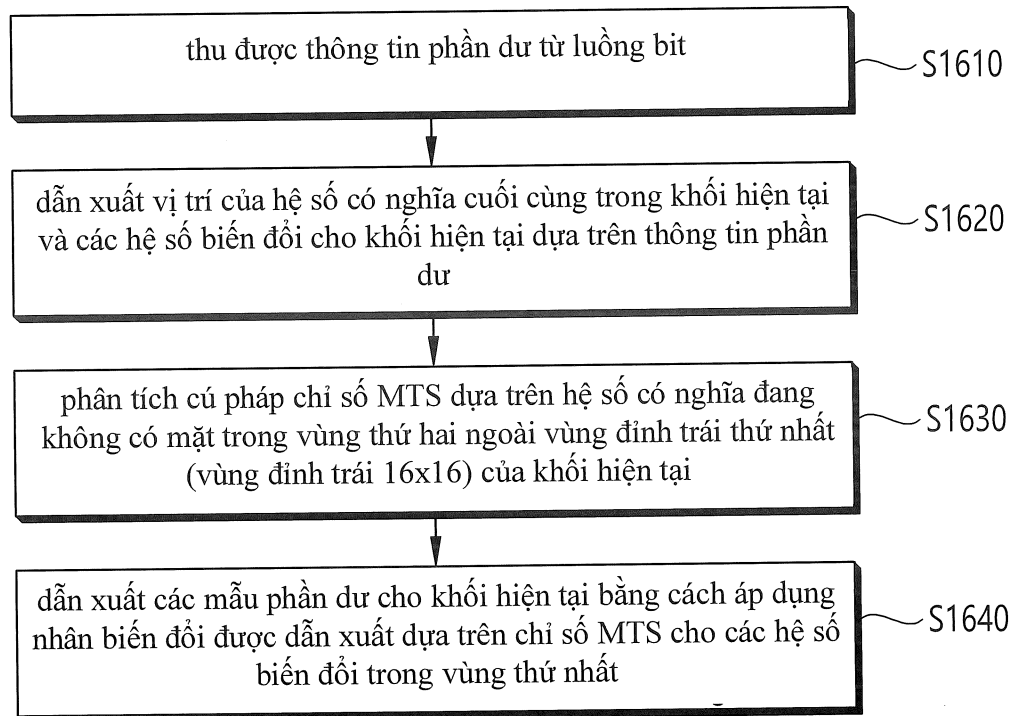


FIG. 17

