(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) **1-0048256**

(51)[2020.01] **H04N 19/132**; H04N 19/18; H04N 19/176; H04N 19/105

(13) **B**

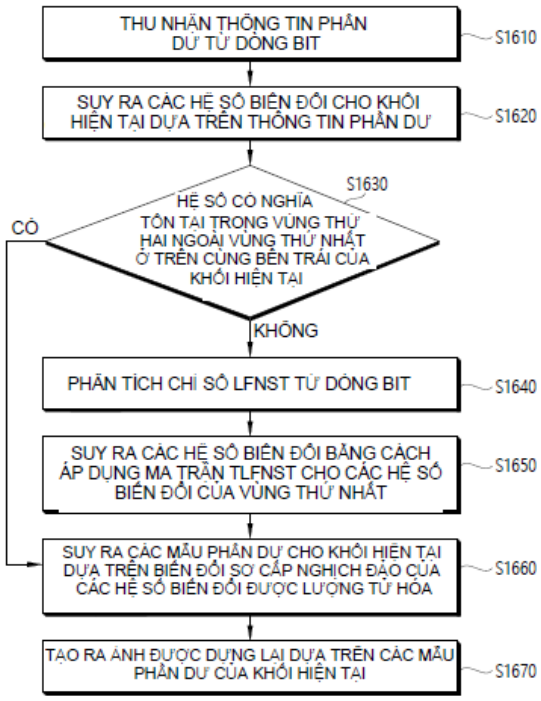
(21) 1-2021-07917
(22) 19/06/2020
(86) PCT/KR2020/007992 19/06/2020
(87) WO2020/256482 24/12/2020
(30) 62/863,833 19/06/2019 US; 62/865,133 21/06/2019 US
(45) 25/07/2025 448
(43) 25/03/2022 408A
(73) LG ELECTRONICS INC. (KR)
128, Yeoui-daero, Yeongdeungpo-Gu Seoul 07336, Korea
(72) KOO, Moonmo (KR); LIM, Jaehyun (KR); NAM, Junghak (KR); KIM, Seunghwan (KR).
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ VÀ MÃ HÓA HÌNH ẢNH, PHƯƠNG PHÁP TRUYỀN VÀ PHƯƠNG TIỆN LƯU TRỮ SỐ ĐỌC ĐƯỢC BẰNG MÁY TÍNH

(21) 1-2021-07917

(57) Sáng chế đề cập đến phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã, phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa, phương pháp truyền và phương tiện lưu trữ số đọc được bằng máy tính. Phương pháp giải mã hình ảnh, theo sáng chế, có thể bao gồm các bước: suy ra các hệ số biến đổi cho khối hiện tại trên cơ sở thông tin phần dư; xác định liệu hệ số có nghĩa có mặt hay không trong vùng thứ hai không gồm vùng thứ nhất ở trên cùng bên trái của khối hiện tại; phân tích chỉ số biến đổi không tách được tần số thấp (low-frequency non-separable transform, LFNST) từ dòng bit nếu hệ số có nghĩa không có mặt trong vùng thứ hai; suy ra các hệ số biến đổi được sửa đổi bằng cách áp dụng ma trận LFNST, được suy ra trên cơ sở chỉ số LFNST, để biến đổi các hệ số của vùng thứ nhất; và suy ra các mẫu phần dư của khối hiện tại trên cơ sở biến đổi sơ cấp ngược của các hệ số biến đổi được sửa đổi.

Fig.16



Lĩnh vực kỹ thuật được đề cập

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

Tình trạng kỹ thuật của sáng chế

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

Hơn nữa, ngày nay, sự quan tâm và nhu cầu đối với các phương tiện truyền thông nhập vai chẳng hạn như nội dung hoặc ảnh toàn ký thực tế ảo (virtual reality, VR), thực tế nhân tạo (artificial reality, AR), hoặc tương tự ngày càng tăng, và việc truyền phát các hình ảnh/các viđeo có các đặc điểm hình ảnh khác với các đặc điểm của các hình ảnh thực tế, chẳng hạn như hình ảnh trò chơi ngày càng tăng.

Theo đó, có nhu cầu đối với kỹ thuật nén hình ảnh/viđeo hiệu quả cao để nén và truyền hoặc lưu trữ hiệu quả, và tái lập thông tin của các hình ảnh/các viđeo 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

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

Một khía cạnh kỹ thuật khác của sáng chế là đề xuất phương pháp và thiết bị để tăng hiệu quả trong tạo mã chỉ số biến đổi.

Một khía cạnh kỹ thuật khác nữa của sáng chế là đề xuất phương pháp và thiết bị tạo mã hình ảnh sử dụng biến đổi không tách được tần số thấp (low-frequency non-separable transform, LFNST).

Một khía cạnh kỹ thuật khác nữa của sáng chế là đề xuất phương pháp và máy tạo mã hình ảnh để sự đặt bằng không được thực hiện khi áp dụng LFNST.

Theo một phương án của sáng chế, phương pháp giải mã hình ảnh được thực hiện bởi thiết bị giải mã được đề xuất. Phương pháp này có thể gồm các bước: thu nhận thông tin phần dư từ dòng bit; suy ra các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư; xác định liệu hệ số có nghĩa có mặt trong vùng thứ hai khác với vùng thứ nhất ở trên cùng bên trái của khối hiện tại hay không; phân tích chỉ số LFNST từ dòng bit khi hệ số có nghĩa không có mặt trong vùng thứ hai; suy ra các hệ số biến đổi được sửa đổi bằng cách áp dụng ma trận LFNST được suy ra dựa trên chỉ số LFNST cho các hệ số biến đổi trong vùng thứ nhất; suy ra mẫu phần dư cho khối hiện tại dựa trên biến đổi sơ cấp ngược cho các hệ số biến đổi được sửa đổi; và tạo ra ảnh được dựng lại dựa trên các mẫu phần dư cho khối hiện tại.

Ở đây, vùng thứ nhất được suy ra dựa trên kích thước của khối hiện tại, khi kích thước của khối hiện tại là 4x4 hoặc 8x8, thì vùng thứ nhất là từ trên cùng bên trái của khối hiện tại đến vị trí mẫu thứ 8 theo hướng quét, và khi kích thước của khối hiện tại không phải là 4x4 hoặc 8x8, thì vùng thứ nhất là khu vực 4x4 ở trên cùng bên trái của khối hiện tại.

Hướng quét là hướng quét đường chéo.

Số lượng xác định trước của các hệ số biến đổi được sửa đổi được suy ra dựa trên kích thước của khối hiện tại, khi chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, thì 48 hệ số biến đổi được sửa đổi được suy ra, và khi chiều rộng

và chiều cao của khối hiện tại là 4 hoặc lớn hơn và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, thì 16 hệ số biến đổi sửa đổi được suy ra.

48 hệ số biến đổi được sửa đổi được sắp xếp ở các khu vực 4x4 trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái trong phạm vi khu vực 8x8 trên cùng bên trái của khối hiện tại.

16 hệ số biến đổi được sửa đổi được sắp xếp trong khu vực 4x4 trên cùng bên trái của khối hiện tại.

Theo một phương án khác của sáng chế, phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa được đề xuất. Phương pháp này có thể gồm các bước: suy ra các mẫu dự đoán cho khối hiện tại; suy ra mẫu các phần dư cho khối hiện tại dựa trên mẫu dự đoán; suy ra các hệ số biến đổi cho khối hiện tại dựa trên các hệ số biến đổi sơ cấp cho các mẫu phần dư; suy ra hệ số biến đổi đã sửa đổi cho khối hiện tại dựa trên các hệ số biến đổi của vùng thứ nhất ở trên cùng bên trái của khối hiện tại và ma trận LFNST được thông qua lượng tử hóa các hệ số biến đổi đã sửa đổi và chỉ số LFNST xác định trước; đặt bằng không vùng thứ hai của khối hiện tại mà trong đó các hệ số biến đổi đã sửa đổi không có mặt; và mã hóa thông tin phần dư được suy ra qua phép lượng tử hóa của các hệ số biến đổi và chỉ số LFNST chỉ ra ma trận LFNST.

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

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

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

Theo sáng chế, có thể tăng hiệu quả trong tạo mã chỉ số biến đổi.

Một khía cạnh kỹ thuật của sáng chế có thể đề xuất phương pháp và thiết bị tạo mã hình ảnh sử dụng LFNST.

Một khía cạnh kỹ thuật của sáng chế có thể đề xuất phương pháp và máy tạo mã hình ảnh để sự đặt bằng không được thực hiện khi áp dụng LFNST.

Các hiệu quả mà có thể được thu nhận qua các ví dụ cụ thể của sáng chế không bị giới hạn với các hiệu quả được liệt kê ở trên. Ví dụ, có thể có nhiều hiệu quả kỹ thuật khác nhau mà người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể hiểu hoặc suy ra từ sáng chế. Do đó, các hiệu quả cụ thể của sáng chế không bị giới hạn ở các hiệu quả được mô tả rõ ràng trong sáng chế và có thể gồm các hiệu quả khác nhau mà có thể được hiểu hoặc được suy ra từ các dấu hiệu kỹ thuật của sáng chế.

Mô tả vắn tắt các hình vẽ

Fig.1 minh họa dạng giản đồ ví dụ của hệ thống tạo mã viđeo/hình ảnh mà sáng chế có thể áp dụng.

Fig.2 là sơ đồ minh họa dạng giản đồ cấu hình của thiết bị mã hóa viđeo/hình ảnh mà sáng chế có thể áp dụng.

Fig.3 là sơ đồ minh họa dạng giản đồ cấu hình của thiết bị giải mã viđeo/hình ảnh mà sáng chế có thể áp dụng.

Fig.4 minh họa dạng giản đồ lược đồ đa biến đổi theo một phương án của sáng chế.

Fig.5 thể hiện ví dụ các chế độ định hướng nội ảnh của 65 hướng dự đoán.

Fig.6 là sơ đồ giải thích RST theo một phương án của sáng chế.

Fig.7 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của biến đổi sơ cấp thuận thành véc-tơ một chiều theo một ví dụ.

Fig.8 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của biến đổi thứ cấp thuận thành véc-tơ hai chiều theo một ví dụ.

Fig.9 là sơ đồ minh họa các chế độ nội dự đoán góc rộng theo một phương án của sáng chế.

Fig.10 là sơ đồ minh họa hình dạng khối mà LFNST được áp dụng.

Fig.11 là sơ đồ minh họa cách sắp xếp dữ liệu đầu ra của một LFNST thuận theo một ví dụ.

Fig.12 là sơ đồ minh họa rằng số lượng dữ liệu đầu ra cho một LFNST thuận được giới hạn ở giá trị tối đa là 16 theo một ví dụ.

Fig.13 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 4x4 được áp dụng theo một ví dụ.

Fig.14 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 8x8 được áp dụng theo một ví dụ.

Fig.15 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 8x8 được áp dụng theo một ví dụ khác.

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

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

Fig.18 minh họa cấu trúc của hệ thống tạo dòng nội dung mà sáng chế được áp dụng.

Mô tả chi tiết sáng chế

Mặc dù sáng chế có thể dễ bị các sửa đổi khác nhau và gồm các phương án khác nhau, nhưng các phương án cụ thể của nó đã được thể hiện trong các hình vẽ bằng ví dụ và bây giờ sẽ được mô tả chi tiết. Tuy nhiên, điều này không nhằm mục đích giới hạn sáng chế với các phương án cụ thể được bộc lộ ở đây. Thuật ngữ được sử dụng ở đây chỉ nhằm mục đích mô tả các phương án cụ thể, và không nhằm giới hạn ý tưởng kỹ thuật của sáng chế. Các dạng số ít có thể gồm các dạng số nhiều trừ khi ngữ cảnh chỉ ra rõ ràng theo cách khác. Các thuật ngữ chẳng hạn như "gồm" và "có" nhằm chỉ ra rằng các đặc điểm, các số lượng, các bước, các hoạt động, các phần tử, các thành phần, hoặc các kết hợp của chúng được sử dụng trong mô tả sau

đây có mặt, và do đó không nên được hiểu rằng khả năng có mặt hoặc bổ sung một hoặc nhiều đặc điểm, số lượng, bước, hoạt động vận hành, phần tử, thành phần khác nhau, hoặc các kết hợp khác nhau của chúng được loại trừ trước.

Trong khi đó, mỗi thành phần trên các hình vẽ được mô tả ở đây được minh họa độc lập để thuận tiện mô tả như với các chức năng đặc trưng khác biệt với nhau, và tuy nhiên, không có nghĩa là mỗi thành phần được thực hiện bởi phần cứng hoặc phần mềm riêng biệt. Ví dụ, bất kỳ hai hoặc nhiều hơn các thành phần có thể được kết hợp để tạo thành thành phần đơn lẻ, và thành phần đơn lẻ bất kỳ cũng có thể được chia thành nhiều thành phần. Các phương án mà trong đó các thành phần được kết hợp và/hoặc được chia sẽ thuộc về phạm vi của sáng chế miễn là chúng không rời khỏi bản chất của sáng chế.

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

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

Trong sáng chế, nhiều phương án khác nhau liên quan đến tạo mã viđeo/hình ảnh có thể được đề xuất, và, trừ khi được quy định ngược lại, các phương án có thể được kết hợp với nhau và được thực hiện.

Trong sáng chế, viđeo có thể có nghĩa là tập hợp của một loạt các hình ảnh theo thời gian. Nói chung, ảnh có nghĩa là đơn vị biểu diễn hình ảnh tại một múi

thời gian cụ thể, và mảnh/ô xếp là đơn vị cấu thành một phần của ảnh. Mảnh/ô xếp có thể gồm một hoặc nhiều đơn vị cây tạo mã (coding tree unit, CTU). Một ảnh có thể được cấu thành bởi một hoặc nhiều mảnh/ô xếp. Một ảnh có thể được cấu thành bởi một hoặc nhiều nhóm ô xếp. Một nhóm ô xếp có thể gồm một hoặc nhiều ô xếp.

Pixel (điểm ảnh) hoặc pel có thể có nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc hình ảnh). Ngoài ra, "mẫu" có thể được sử dụng như thuật ngữ tương ứng với pixel. Mẫu thường có thể biểu diễn pixel hoặc giá trị của pixel, và có thể chỉ biểu diễn pixel/giá trị pixel của thành phần luma (độ sáng) hoặc chỉ pixel/giá trị pixel của thành phần chroma (sắc độ). Ngoài ra, mẫu có thể tham chiếu đến giá trị pixel trong miền không gian, hoặc khi giá trị pixel này được biến đổi sang miền tần số, thì nó có thể tham chiếu đến hệ số biến đổi trong miền tần số.

Đơn vị có thể biểu diễn đơn vị cơ sở của quá trình xử lý hình ảnh. Đơn vị có thể gồm ít nhất một trong vùng cụ thể và thông tin liên quan đến vùng đó. Một đơn vị có thể gồm một khối luma và hai khối chroma (ví dụ, cb, cr). Đơn vị và thuật ngữ chẳng hạn như khối, khu vực, hoặc tương tự có thể được sử dụng thay thế cho nhau tùy theo các tình huống. Trong trường hợp tổng quát, khối M × N có thể gồm tập hợp (hoặc mảng) các mẫu (hoặc các mảng mẫu) hoặc các hệ số biến đổi gồm có M cột và N hàng.

Trong sáng chế, thuật ngữ "/" và "," nên được hiểu là chỉ ra "và/hoặc". Ví dụ, cụm từ "A/B" có thể có nghĩa là "A và/hoặc B". Hơn 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 trong A, B, và/hoặc C". Ngoài ra, "A/B/C" có thể có nghĩa là "ít nhất một trong A, B, và/hoặc C".

Hơn nữa, trong sáng chế, thuật ngữ "hoặc" nên được hiểu là chỉ ra "và/hoặc". Chẳng hạn, phương trình "A hoặc B" có thể gồm 1) chỉ A, 2) chỉ B, và/hoặc 3) cả A và B. Nói cách khác, thuật ngữ "hoặc" trong sáng chế nên được hiểu là chỉ ra "bổ sung hoặc thay thế".

Trong sáng chế, “ít nhất một trong A và B” có thể có nghĩa là “chỉ A”, “chỉ B”, hoặc “cả A và B”. Ngoài ra, trong sáng chế, cụm từ “ít nhất một trong A hoặc B” hoặc “ít nhất một trong A và/hoặc B” có thể được hiểu giống như “ít nhất một trong A và B”.

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

Ngoài ra, dấu ngoặc đơn được sử dụng trong sáng chế có thể có nghĩa là “ví dụ”. Cụ thể, khi được chỉ ra là “sự dự đoán (sự nội dự đoán)”, nó có thể có nghĩa là “sự nội dự đoán” được đề xuất là ví dụ của “sự dự đoán”. Nói cách khác, “sự dự đoán” của sáng chế không bị giới hạn với “sự nội dự đoán”, và “sự nội dự đoán” có thể được đề xuất là một ví dụ của “sự dự đoán”. Ngoài ra, khi được chỉ ra là “sự dự đoán (tức là, sự nội dự đoán)”, nó cũng có thể có nghĩa là “sự nội dự đoán” được đề xuất là ví dụ của “sự dự đoán”.

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

Fig.1 minh họa dạng giản đồ ví dụ của hệ thống tạo mã viđeo/hình ảnh mà sáng chế có thể áp dụng.

Tham chiếu đến Fig.1, hệ thống tạo mã viđeo/hình ảnh có thể gồm thiết bị thứ nhất (thiết bị nguồn) và thiết bị thứ hai (thiết bị nhận). Thiết bị nguồn có thể phân phối thông tin hoặc dữ liệu viđeo/hình ảnh được mã hóa ở dạng tệp hoặc tạo dòng đến thiết bị nhận thông qua mạng hoặc phương tiện lưu trữ số.

Thiết bị nguồn có thể gồm nguồn viđeo, thiết bị mã hóa, và bộ truyền. Thiết bị nhận có thể gồm bộ nhận, thiết bị giải mã, và bộ kết xuất. Thiết bị mã hóa có thể được gọi là thiết bị mã hóa viđeo/hình ảnh, và thiết bị giải mã có thể được gọi là thiết bị giải mã viđeo/hình ảnh. Bộ truyền có thể được gồm trong thiết bị mã hóa.

Bộ nhận có thể được gồm trong thiết bị giải mã. Bộ kết xuất có thể gồm bộ hiển thị, và bộ hiển thị có thể được tạo cấu hình là thiết bị riêng biệt hoặc thành phần bên ngoài.

Bộ nguồn viđeo có thể thu nhận viđeo/hình ảnh qua quá trình thu nạp, tổng hợp, hoặc tạo viđeo/hình ảnh. Bộ nguồn viđeo có thể gồm thiết bị thu nạp viđeo/hình ảnh và/hoặc thiết bị tạo viđeo/hình ảnh. Thiết bị thu nạp viđeo/hình ảnh có thể gồm, ví dụ, một hoặc nhiều camera, kho lưu trữ viđeo/hình ảnh gồm viđeo/các hình ảnh đã thu nạp trước đó, hoặc dạng tương tự. Thiết bị tạo viđeo/hình ảnh có thể gồm, ví dụ, máy tính, máy tính bảng, và điện thoại thông minh, và có thể tạo ra (kiểu điện tử) dữ liệu viđeo/hình ảnh. Ví dụ, viđeo/hình ảnh ảo có thể được tạo ra qua máy tính hoặc thiết bị tương tự. Trong trường hợp này, quá trình thu nạp viđeo/hình ảnh có thể được thay thế bằng quá trình tạo dữ liệu liên quan.

Thiết bị mã hóa 12 có thể mã hóa viđeo/hình ảnh đầu vào. Thiết bị mã hóa có thể thực hiện một loạt các quy trình chẳng hạn như dự đoán, biến đổi, và lượng tử hóa để nén và tạo mã hiệu quả. Dữ liệu được mã hóa (thông tin viđeo/hình ảnh được mã hóa) có thể được xuất ra ở dạng dòng bit.

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

Thiết bị giải mã có thể giải mã viđeo/hình ảnh bằng cách thực hiện một loạt các quy trình chẳng hạn như giải lượng tử hóa, biến đổi ngược, dự đoán, và quy trình tương tự tương ứng với hoạt động của thiết bị mã hóa.

Bộ kết xuất có thể trình diễn viđeo/hình ảnh được giải mã. Viđeo/hình ảnh được trình diễn có thể được hiển thị qua bộ hiển thị.

Fig.2 là sơ đồ minh họa dạng giản đồ cấu hình của thiết bị mã hóa viđeo/hình ảnh mà sáng chế có thể áp dụng. Sau đây, thứ được tham chiếu là thiết bị mã hóa viđeo có thể gồm thiết bị mã hóa hình ảnh.

Tham chiếu đến Fig.2, thiết bị mã hóa 200 có thể gồm bộ chia hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entrôpi 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể gồm bộ liên dự đoán 221 và bộ nội dự đoán 222. Bộ xử lý phần dư 230 có thể gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ giải lượng tử hóa 234, và bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ dựng lại hoặc bộ tạo khối được dựng lại. Bộ chia hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entrôpi 240, bộ cộng 250, và bộ lọc 260, mà đã được mô tả ở trên, có thể được cấu thành bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip (chipset) hoặc các bộ xử lý mã hóa) theo một phương án. Hơn nữa, bộ nhớ 270 có thể gồm bộ đệm ảnh được giải mã (decoded picture buffer, DPB) hoặc có thể được cấu thành bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 270 dưới dạng thành phần bên trong/bên ngoài.

Bộ chia hình ảnh 210 có thể phân chia hình ảnh đầu vào (hoặc ảnh hoặc khung hình) được đưa vào thiết bị mã hóa 200 thành một hoặc nhiều đơn vị xử lý. Một ví dụ, đơn vị xử lý có thể được gọi là đơn vị tạo mã (coding unit, CU). Trong trường hợp này, bắt đầu với đơn vị cây tạo mã (coding tree unit, CTU) hoặc đơn vị tạo mã lớn nhất (largest coding unit, LCU), đơn vị tạo mã có thể được phân chia đệ quy theo cấu trúc cây tứ phân nhị phân cây tam phân (Quad-tree binary-tree ternary-tree, QTBTTT). Ví dụ, một đơn vị tạo mã có thể được chia thành nhiều đơn vị tạo mã có độ sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc tam phân. Trong trường hợp này, ví dụ, cấu trúc cây tứ có thể được áp dụng

đầu tiên và cấu trúc cây nhị phân và/hoặc cấu trúc cây tam phân có thể được áp dụng sau đó. Ngoài ra, cấu trúc cây nhị phân có thể được áp dụng đầu tiên. Quy trình tạo mã theo sáng chế có thể được thực hiện dựa trên đơn vị tạo mã cuối cùng mà không được phân chia nữa. Trong trường hợp này, đơn vị tạo mã cực đại có thể được sử dụng trực tiếp là đơn vị tạo mã cuối cùng dựa trên hiệu quả tạo mã theo đặc điểm hình ảnh. Ngoài ra, đơn vị tạo mã có thể được phân chia đệ quy thành các đơn vị tạo mã có độ sâu hơn nữa nếu cần, sao cho đơn vị tạo mã có kích thước tối ưu có thể được sử dụng là đơn vị tạo mã cuối cùng. Ở đây, quy trình tạo mã có thể gồm các quy trình chẳng hạn như dự đoán, biến đổi, và dựng lại, mà sẽ được mô tả sau đây. Một ví dụ khác, đơn vị xử lý có thể còn gồm đơn vị dự đoán (prediction unit, PU) hoặc đơn vị biến đổi (transform unit, TU). Trong trường hợp này, đơn vị dự đoán và đơn vị biến đổi có thể được tách ra hoặc được phân chia từ đơn vị tạo mã cuối cùng được mô tả ở trên. Đơn vị dự đoán có thể là đơn vị dự đoán mẫu, và đơn vị biến đổi có thể là đơn vị để suy ra hệ số biến đổi và/hoặc đơn vị để suy ra tín hiệu phần dư từ hệ số biến đổi.

Đơn vị và thuật ngữ chẳng hạn như khối, khu vực, hoặc tương tự có thể được sử dụng thay thế cho nhau tùy theo các tình huống. Trong trường hợp tổng quát, khối M × N có thể biểu diễn tập hợp các mẫu hoặc các hệ số biến đổi gồm có M cột và N hàng. Mẫu thường có thể biểu diễn pixel hoặc giá trị của pixel, và có thể chỉ biểu diễn pixel/giá trị pixel của thành phần luma hoặc chỉ pixel/giá trị pixel của thành phần chroma. Mẫu có thể được sử dụng là thuật ngữ tương ứng với pixel hoặc pel của ảnh (hoặc hình ảnh).

Bộ trừ 231 trừ đầu ra tín hiệu dự đoán (khối dự đoán, mảng mẫu dự đoán) từ bộ dự đoán 220 khỏi tín hiệu hình ảnh đầu vào (khối ban đầu, mảng mẫu ban đầu) để tạo ra tín hiệu phần dư (khối phần dư, mảng mẫu phần dư), và tín hiệu phần dư được truyền đến bộ biến đổi 232. Bộ dự đoán 220 có thể thực hiện sự dự đoán trên khối đích xử lý (sau đây, được tham chiếu là “khối hiện tại”), và có thể tạo ra khối

được dự đoán gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán 220 có thể xác định liệu sự nội dự đoán hay sự liên dự đoán được áp dụng trên cơ sở khối hiện tại hoặc CU. Như được mô tả 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 sự dự đoán, chẳng hạn như thông tin chế độ dự đoán, và truyền thông tin được tạo ra đến bộ mã hóa entrôpi 240. Thông tin về dự đoán có thể được mã hóa trong bộ mã hóa entrôpi 240 và được xuất ra ở dạng dòng bit.

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

Bộ liên dự đoán 221 có thể suy ra khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được chỉ định bởi véc-tơ chuyển động trên ảnh tham chiếu. Tại thời điểm này, để giảm lượng thông tin chuyển động được truyền trong chế độ liên dự đoán, thông tin chuyển động có thể được dự đoán theo trên cơ sở khối, khối con, hoặc mẫu dựa trên sự tương quan của thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể gồm véc-tơ chuyển động và chỉ số ảnh tham chiếu. Thông tin chuyển động có thể còn gồm thông tin hướng liên dự đoán (dự đoán L0, dự đoán L1, song dự đoán (Bi), v.v.). Trong trường hợp liên dự đoán, khối lân cận có thể gồm khối lân cận không gian hiện hữu trong ảnh hiện tại và khối lân cận thời gian hiện hữu trong ảnh tham chiếu.

Ảnh tham chiếu gồm khối tham chiếu và ảnh tham chiếu gồm khối lân cận thời gian có thể giống nhau hoặc khác nhau. Khối lân cận thời gian có thể được gọi là khối tham chiếu đồng vị trí, CU đồng vị trí (colocated CU, colCU), và tương tự, và ảnh tham chiếu gồm khối lân cận thời gian có thể được gọi là ảnh đồng vị trí (collocated picture, colPic). Ví dụ, bộ liên dự đoán 221 có thể tạo cấu hình danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và tạo ra thông tin chỉ ra ứng viên nào được sử dụng để suy ra véc-tơ chuyển động và/hoặc chỉ số ảnh tham chiếu của khối hiện tại. Sự 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 đi. Trong trường hợp của chế độ dự đoán thông tin chuyển động (chế độ dự đoán véc-tơ chuyển động (motion vector prediction, MVP)), véc-tơ chuyển động của khối lân cận có thể được sử dụng như bộ dự đoán véc-tơ chuyển động và véc-tơ chuyển động của khối hiện tại có thể được chỉ ra bằng cách tín hiệu hóa sự khác biệt véc-tơ chuyển động.

Bộ dự đoán 220 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng sự nội dự đoán hoặc sự liên dự đoán cho sự dự đoán trên một khối, và, cũng như, có thể áp dụng sự nội dự đoán và sự liên dự đoán tại cùng thời điểm. Điều này có thể được gọi là sự liên dự đoán và nội ảnh được kết hợp (combined inter and intra prediction, CIIP). Hơn nữa, bộ dự đoán có thể dựa trên chế độ dự đoán bản sao khối nội ảnh (intra block copy, IBC), hoặc chế độ bảng màu để thực hiện sự dự đoán trên khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng để tạo mã hình ảnh/viđeo nội dung của trò chơi hoặc nội dung tương tự, chẳng hạn như tạo mã nội dung màn hình (screen content coding, SCC). Mặc dù IBC về cơ bản thực hiện sự dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện tương tự như sự liên dự đoán ở chỗ nó suy ra

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

Tín hiệu dự đoán được tạo ra qua bộ liên dự đoán 221 và/hoặc bộ nội dự đoán 222 có thể được sử dụng để tạo ra tín hiệu được dựng lại hoặc để tạo ra tín hiệu phần dư. Bộ biến đổi 232 có thể tạo ra các hệ số biến đổi bằng cách áp dụng kỹ thuật biến đổi cho tín hiệu phần dư. Ví dụ, kỹ thuật biến đổi có thể gồm ít nhất một trong 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 tính có điều kiện (conditionally non-linear transform, CNT). Ở đây, GBT có nghĩa là biến đổi được thu nhận từ đồ thị khi thông tin mối quan hệ giữa các pixel được biểu diễn bằng đồ thị. CNT đề cập đến biến đổi được thu nhận dựa trên tín hiệu dự đoán được tạo ra sử dụng tất cả các pixel được dựng lại trước đó. Ngoài ra, quá trình biến đổi có thể được áp dụng cho các khối pixel vuông có cùng kích thước hoặc có thể được áp dụng cho các khối có kích thước có thể thay đổi thay vì kích thước vuông.

Bộ lượng tử hóa 233 có thể lượng tử hóa các hệ số biến đổi và truyền chúng đến bộ mã hóa entrôpi 240, và bộ mã hóa entrôpi 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 dòng bit. Thông tin về các hệ số biến đổi được lượng tử hóa có thể được tham chiếu là thông tin phần dư. Bộ lượng tử hóa 233 có thể sắp xếp lại các hệ số biến đổi được lượng tử hóa kiểu khối thành dạng véc-tơ một chiều dựa trên thứ tự quét hệ số, và tạo ra thông tin về các hệ số biến đổi được lượng tử hóa dựa trên các hệ số biến đổi được lượng tử hóa của dạng véc-tơ một chiều. Bộ mã hóa entrôpi 240 có thể thực hiện các phương pháp mã hóa khác nhau, chẳng hạn như Golomb theo cấp số nhân, tạo mã độ dài biến đổi thích ứng theo ngữ cảnh (context-adaptive variable length coding, CAVLC), tạo mã số học nhị phân thích

ứng theo ngữ cảnh (context-adaptive binary arithmetic coding, CABAC), và phương pháp tương tự. Bộ mã hóa entrôpi 240 có thể mã hóa thông tin cần thiết cho sự dựng lại viđeo/hình ảnh khác với các hệ số biến đổi được lượng tử hóa (ví dụ, giá trị của các phần tử cú pháp, v.v.) cùng nhau hoặc riêng biệt. Thông tin được mã hóa (ví dụ, thông tin viđeo/hình ảnh được mã hóa) có thể được truyền hoặc được lưu trữ trong đơn vị cơ sở của lớp trừu tượng mạng (network abstraction layer, NAL) ở dạng dòng bit. Thông tin viđeo/hình ảnh có thể còn gồm thông tin về các tập hợp thông số khác nhau chẳng hạn như tập hợp thông số thích ứng (adaptation parameter set, APS), tập hợp thông số ảnh (picture parameter set, PPS), tập hợp thông số trình tự (sequence parameter set, SPS), hoặc tập hợp thông số viđeo (viđeo parameter set, VPS) hoặc dạng tương tự. Hơn nữa, thông tin viđeo/hình ảnh có thể còn gồm thông tin ràng buộc chung. Trong sáng chế, thông tin và/hoặc các phần tử cú pháp mà được truyền/được tín hiệu hóa đến thiết bị giải mã từ thiết bị mã hóa có thể được gồm trong thông tin viđeo/hình ảnh. Thông tin viđeo/hình ảnh có thể được mã hóa qua quy trình mã hóa được mô tả ở trên và được gồm trong dòng bit. Dòng bit có thể được truyền qua mạng, hoặc được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Mạng có thể gồm mạng phát rộng, mạng truyền thông và/hoặc loại tương tự, và phương tiện lưu trữ kỹ thuật số có thể gồm các phương tiện lưu trữ khác nhau chẳng hạn như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và loại tương tự. Bộ truyền (không được thể hiện) mà truyền tín hiệu được xuất ra từ bộ mã hóa entrôpi 240 và/hoặc bộ lưu trữ (không được thể hiện) mà lưu trữ tín hiệu có thể được tạo cấu hình như phần tử bên trong/bên ngoài của thiết bị mã hóa 200, hoặc bộ truyền có thể được gồm trong bộ mã hóa entrôpi 240.

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

Trong khi đó, trong quá trình mã hóa và/hoặc dựng lại ảnh, sự ánh xạ luma với sự tỷ lệ hóa chroma (luma mapping with chroma scaling, LMCS) có thể được áp dụng.

Bộ lọc 260 có thể cải thiện chất lượng viđeo chủ quan/khách quan bằng cách áp dụng quá trình lọc cho tín hiệu được dựng lại. Ví dụ, bộ lọc 260 có thể tạo ra ảnh được dựng lại được sửa đổi bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được dựng lại, và có thể lưu trữ ảnh được dựng lại được sửa đổi trong bộ nhớ 270, cụ thể là trong DPB của bộ nhớ 270. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc tách khối, bù thích ứng mẫu, bộ lọc vòng lặp thích ứng, bộ lọc hai chiều hoặc tương tự. Như được thảo luận sau đây trong phần mô tả của từng phương pháp lọc, bộ lọc 260 có thể tạo ra nhiều thông tin khác nhau liên quan đến quá trình lọc, và truyền thông tin đã tạo ra đến bộ mã hóa entrôpi 240. Thông tin về quá trình lọc có thể được mã hóa trong bộ mã hóa entrôpi 240 và được xuất ra ở dạng dòng bit.

Ảnh được dựng lại được sửa đổi mà đã được truyền đến bộ nhớ 270 có thể được sử dụng làm ảnh tham chiếu trong bộ liên dự đoán 221. Qua cách này, thiết bị

mã hóa có thể tránh sự không phù hợp dự đoán trong thiết bị mã hóa 100 và thiết bị giải mã khi áp dụng sự liên dự đoán, và cũng có thể cải thiện hiệu quả tạo mã.

Bộ nhớ 270 DPB có thể lưu trữ ảnh được dựng lại được sửa đổi để sử dụng nó làm ảnh tham chiếu trong bộ liên dự đoán 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 suy ra (hoặc được mã hóa) và/hoặc thông tin chuyển động của các khối trong ảnh đã được dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền tới đơn vị liên dự đoán 221 để được sử dụng làm thông tin chuyển động của khối lân cận hoặc thông tin chuyển động của khối lân cận thời gian. Bộ nhớ 270 có thể lưu trữ các mẫu được dựng lại của các khối được dựng lại trong ảnh hiện tại, và truyền chúng đến bộ nội dự đoán 222.

Fig.3 là sơ đồ minh họa dạng giản đồ cấu hình của thiết bị giải mã viđeo/hình ảnh mà sáng chế có thể áp dụng.

Tham chiếu đến Fig.3, thiết bị giải mã viđeo 300 có thể gồm bộ giải mã entrôpi 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 gồm bộ liên dự đoán 331 và bộ nội dự đoán 332. Bộ xử lý phần dư 320 có thể gồm bộ giải lượng tử hóa 321 và bộ biến đổi ngược 321. Bộ giải mã entrôpi 310, bộ xử lý phần dư 320, bộ dự đoán 330, bộ cộng 340, và bộ lọc 350, mà đã được mô tả ở trên, có thể được cấu thành bởi một hoặc nhiều thành phần phần cứng (ví dụ, các chipset hoặc các bộ xử lý bộ giải mã) theo một phương án. Hơn nữa, bộ nhớ 360 có thể gồm bộ đệm ảnh được giải mã (decoded picture buffer, DPB) hoặc có thể được cấu thành bởi phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn gồm bộ nhớ 360 dưới dạng thành phần bên trong/bên ngoài.

Khi dòng bit gồm thông tin viđeo/hình ảnh được đưa vào, thiết bị giải mã 300 có thể dựng lại hình ảnh tương ứng với quá trình mà theo đó thông tin viđeo/hình ảnh được xử lý trong thiết bị mã hóa của Fig.2. Ví dụ, thiết bị giải mã 300 có thể

suy ra các đơn vị/các khối dựa trên thông tin liên quan đến sự phân chia khối được thu nhận từ dòng bit. Thiết bị giải mã 300 có thể thực hiện sự giải mã bằng cách sử dụng bộ xử lý được áp dụng trong thiết bị mã hóa. Do đó, đơn vị xử lý giải mã có thể là, ví dụ, đơn vị tạo mã, mà có thể được phân chia dọc theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây tạo mã hoặc đơn vị tạo mã lớn nhất. Một hoặc nhiều đơn vị biến đổi có thể được suy ra từ đơn vị tạo mã. Và, tín hiệu hình ảnh được dựng lại được giải mã và được xuất ra qua thiết bị giải mã 300 có thể được tái tạo thông qua bộ tái tạo.

Thiết bị giải mã 300 có thể nhận tín hiệu được xuất ra từ thiết bị mã hóa của Fig.2 ở dạng dòng bit, và tín hiệu được nhận có thể được giải mã qua bộ giải mã entrôpi 310. Ví dụ, bộ giải mã entrôpi 310 có thể phân tích dòng bit để suy ra thông tin (ví dụ, thông tin viđeo/hình ảnh) cần thiết cho sự dựng lại hình ảnh (hoặc sự dựng lại ảnh). Thông tin viđeo/hình ảnh có thể còn gồm thông tin về các tập hợp thông số khác nhau chẳng hạn như tập hợp thông số thích ứng (adaptation parameter set, APS), tập hợp thông số ảnh (picture parameter set, PPS), tập hợp thông số trình tự (sequence parameter set, SPS), hoặc tập hợp thông số viđeo (viđeo parameter set, VPS) hoặc dạng tương tự. Hơn nữa, thông tin viđeo/hình ảnh có thể còn gồm thông tin ràng buộc chung. Thiết bị giải mã có thể còn giải mã ảnh dựa trên thông tin về tập hợp thông số và/hoặc thông tin ràng buộc chung. Trong sáng chế, thông tin và/hoặc các phần tử cú pháp được tín hiệu hóa/được nhận, mà được mô tả sau đây, có thể được giải mã qua quy trình giải mã và được thu nhận từ dòng bit. Ví dụ, bộ giải mã entrôpi 310 có thể giải mã thông tin trong dòng bit dựa trên phương pháp tạo mã chẳng hạn như mã hóa Golomb theo số mũ, CAVLC, CABAC, hoặc tương tự, và có thể xuất ra giá trị của phần tử cú pháp cần thiết cho sự dựng lại ảnh và các giá trị được lượng tử hóa của hệ số biến đổi liên quan đến phần dư. Cụ thể hơn, phương pháp giải mã entrôpi CABAC có thể nhận bin tương ứng với mỗi phần tử cú pháp trong dòng bit, xác định mô hình ngữ cảnh sử dụng thông tin phần

tử cú pháp đích giải mã và thông tin giải mã của các khối đích giải mã và lân cận, hoặc thông tin của ký hiệu (symbol)/ nhị phân (bin) được giải mã trong bước trước, dự đoán xác suất tạo ra bin theo mô hình ngữ cảnh được xác định và thực hiện sự giải mã số học của bin để tạo ra ký hiệu tương ứng với mỗi giá trị phần tử cú pháp. Ở đây, phương pháp giải mã entrôpi 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 khi xác định mô hình ngữ cảnh. Thông tin về sự dự đoán trong số các thông tin được giải mã trong bộ giải mã entrôpi 310 có thể được cung cấp cho bộ dự đoán (bộ liên dự đoán 332 và bộ nội dự đoán 331), và các giá trị phần dư, nghĩa là, các hệ số biến đổi được lượng tử hóa, mà trên đó sự giải mã entrôpi đã được thực hiện bởi bộ giải mã 310, và các thông tin thông số được liên kết có thể được đưa vào bộ xử lý phần dư 320. Bộ xử lý phần dư 320 có thể suy ra tín hiệu phần dư (khối phần dư, các mẫu phần dư, mảng mẫu phần dư). Hơn nữa, thông tin về quá trình lọc trong số các thông tin được giải mã bởi bộ giải mã entrôpi 310 có thể được cung cấp cho bộ lọc 350. Trong khi đó, bộ nhận (không được thể hiện) mà nhận tín hiệu được xuất ra từ thiết bị mã hóa có thể cấu thành thêm thiết bị giải mã 300 như phần tử bên trong/bên ngoài, và bộ nhận có thể là thành phần của bộ giải mã entrôpi 310. Trong khi đó, thiết bị giải mã theo sáng chế có thể được gọi là máy tạo mã viđeo/hình ảnh/ảnh, và thiết bị giải mã có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin viđeo/hình ảnh/ảnh) và bộ giải mã mẫu (bộ giải mã mẫu viđeo/hình ảnh/ảnh). Bộ giải mã thông tin có thể gồm bộ giải mã entrôpi 310, và bộ giải mã mẫu có thể gồm ít nhất một trong bộ giải lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ liên dự đoán 332, và bộ nội dự đoán 331.

Bộ giải lượng tử hóa 331 có thể xuất 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 khối hai chiều. Trong

trường hợp này, bộ sắp xếp lại có thể thực hiện sự sắp xếp lại dựa trên thứ tự quét hệ số mà đã được thực hiện trong thiết bị mã hóa. Bộ giải lượng tử hóa 321 có thể thực hiện sự giải lượng tử hóa trên các hệ số biến đổi được lượng tử hóa sử dụng thông số lượng tử hóa (ví dụ, thông tin kích thước bước lượng tử hóa), và thu nhận các hệ số biến đổi.

Bộ biến đổi ngược 322 có thể thu nhận 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 sự dự đoán trên khối hiện tại, và tạo ra khối được dự đoán gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán có thể xác định liệu sự nội dự đoán hay sự liên dự đoán được áp dụng cho khối hiện tại dựa trên thông tin về sự dự đoán được xuất ra từ bộ giải mã entrôpi 310, và cụ thể là có thể xác định chế độ nội dự đoán/liên ảnh.

Bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng sự nội dự đoán hoặc sự liên dự đoán cho sự dự đoán trên một khối, và, cũng như, có thể áp dụng sự nội dự đoán và sự liên dự đoán tại cùng thời điểm. Điều này có thể được gọi là sự liên dự đoán và nội dự đoán được kết hợp (combined inter and intra prediction, CIIP). Ngoài ra, bộ dự đoán có thể thực hiện sự sao chép nội khối (intra block copy, IBC) cho sự dự đoán trên khối. Sự sao chép khối nội ảnh có thể được sử dụng để tạo mã hình ảnh/viđeo nội dung của trò chơi hoặc nội dung tương tự, chẳng hạn như tạo mã nội dung màn hình (screen content coding, SCC). Mặc dù IBC về cơ bản thực hiện sự dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện tương tự như sự liên dự đoán ở chỗ nó suy ra khối tham chiếu trong khối hiện tại. Nghĩa là, IBC có thể sử dụng ít nhất một trong các kỹ thuật liên dự đoán được mô tả trong sáng chế.

Bộ nội dự đoán 331 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt bên cạnh hoặc cách khối hiện tại tùy theo chế độ dự đoán. Trong nội dự đoán, các chế độ dự đoán

có thể gồm nhiều chế độ không định hướng và nhiều chế độ định hướng. Bộ nội dự đoán 331 có thể xác định chế độ dự đoán được áp dụng cho khối hiện tại bằng cách sử dụng chế độ dự đoán áp dụng cho khối lân cận.

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

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

Bộ cộng 340 có thể được gọi là bộ dựng lại hoặc bộ tạo khối được dựng lại. Tín hiệu được dựng lại được tạo ra có thể được sử dụng cho sự nội dự đoán của

khối đích xử lý tiếp theo trong khối hiện tại, và như được mô tả sau đây, có thể được xuất ra qua quá trình lọc hoặc được sử dụng cho sự liên dự đoán của các ảnh tiếp theo.

Trong khi đó, trong quá trình giải mã ảnh, sự ánh xạ luma với sự định tỷ lệ chroma (luma mapping with chroma scaling, LMCS) có thể được áp dụng.

Bộ lọc 350 có thể cải thiện chất lượng viđeo chủ quan/khách quan bằng cách áp dụng quá trình lọc cho tín hiệu được dựng lại. Ví dụ, bộ lọc 350 có thể tạo ra ảnh được dựng lại được sửa đổi bằng cách áp dụng các phương pháp lọc khác nhau cho ảnh được dựng lại, và có thể truyền ảnh được dựng lại được sửa đổi trong bộ nhớ 360, cụ thể là trong DPB của bộ nhớ 360. Các phương pháp lọc khác nhau có thể gồm, ví dụ, lọc tách khối, bù thích ứng mẫu, bộ lọc vòng lặp thích ứng, bộ lọc hai chiều hoặc tương tự.

Ảnh được dựng lại (được sửa đổi) mà đã được lưu trữ trong DPB của bộ nhớ 360 có thể được sử dụng làm ảnh tham chiếu trong bộ liên dự đoán 332. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối trong ảnh hiện tại, mà từ đó thông tin chuyển động đã được suy ra (hoặc được giải mã) và/hoặc thông tin chuyển động của các khối trong ảnh đã được dựng lại. Thông tin chuyển động được lưu trữ có thể được truyền tới đơn vị liên dự đoán 260 để được sử dụng làm thông tin chuyển động của khối lân cận hoặc thông tin chuyển động của khối lân cận thời gian. Bộ nhớ 360 có thể lưu trữ các mẫu được dựng lại của các khối được dựng lại trong ảnh hiện tại, và truyền chúng đến bộ nội dự đoán 331.

Trong sáng chế, 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 cho bộ dự đoán 220, bộ giải lượng tử hóa 234, bộ bộ biến đổi ngược 235, và bộ lọc 260 của thiết bị mã hóa 200, tương ứng.

Như được mô tả ở trên, sự dự đoán được thực hiện để tăng hiệu quả nén trong việc thực hiện tạo mã viđeo. Qua cách này, khối được dự đoán gồm các mẫu dự

đoán cho khối hiện tại, mà là khối đích tạo mã, có thể được tạo ra. Ở đây, khối được dự đoán gồm các mẫu dự đoán trong miền không gian (hoặc miền pixel). Khối được dự đoán có thể được suy ra một cách đồng nhất trong thiết bị mã hóa và thiết bị giải mã, và thiết bị mã hóa có thể tăng hiệu quả tạo mã hình ảnh bằng cách tín hiệu hóa cho thiết bị giải mã không phải giá trị mẫu ban đầu của chính khối ban đầu mà là thông tin về phần dư (thông tin phần dư) giữa khối ban đầu và khối dự đoán. Thiết bị giải mã có thể suy ra khối phần dư gồm các mẫu phần dư dựa trên thông tin phần dư, tạo ra khối được dựng lại gồm các mẫu được dựng lại bằng cách thêm khối phần dư vào khối dự đoán, và tạo ra ảnh được dựng lại gồm các khối được dựng lại.

Thông tin phần dư có thể được tạo ra thông qua các quy trình biến đổi và lượng tử hóa. Ví dụ, thiết bị mã hóa có thể suy ra khối phần dư giữa khối ban đầu và khối dự đoán, suy ra các hệ số biến đổi bằng cách thực hiện quy trình biến đổi trên các mẫu phần dư (mảng mẫu phần dư) được gồm trong khối phần dư, và suy ra các hệ số biến đổi được lượng tử hóa bằng cách thực hiện quy trình lượng tử hóa trên các hệ số biến đổi, sao cho nó có thể tín hiệu hóa thông tin phần dư được liên kết đến thiết bị giải mã (qua dòng bit). Ở đây, thông tin phần dư có thể gồm thông tin giá trị, thông tin vị trí, kỹ thuật biến đổi, nhân biến đổi, thông số lượng tử hóa hoặc thông tin tương tự của các hệ số biến đổi được lượng tử hóa. Thiết bị giải mã có thể thực hiện quy trình lượng tử hóa/giải lượng tử hóa và suy ra các mẫu phần dư (hoặc khối mẫu phần dư), dựa trên thông tin phần dư. Thiết bị giải mã có thể tạo ra khối được dựng lại dựa trên khối được dự đoán và khối phần dư. Thiết bị mã hóa có thể suy ra khối phần dư bằng cách giải lượng tử hóa/biến đổi ngược các hệ số biến đổi được lượng tử hóa cho sự tham chiếu cho sự liên dự đoán của ảnh tiếp theo, và có thể tạo ra ảnh được dựng lại dựa trên điều này.

Fig.4 minh họa dạng giản đồ kỹ thuật biến đổi nhiều lần theo một phương án của sáng chế.

Tham chiếu đến Fig.4, bộ biến đổi có thể tương ứng với bộ biến đổi trong thiết bị mã hóa của Fig.2 đã nói ở trên, và bộ biến đổi ngược có thể tương ứng với bộ biến đổi ngược trong thiết bị mã hóa của Fig.2 đã nói ở trên, hoặc với bộ biến đổi ngược trong thiết bị giải mã của Fig.3.

Bộ biến đổi có thể suy ra các hệ số biến đổi (sơ cấp) bằng cách thực hiện biến đổi sơ cấp dựa trên các mẫu phần dư (mảng mẫu phần dư) trong khối phần dư (S410). Biến đổi sơ cấp này có thể được gọi là biến đổi lõi. Ở đây, biến đổi sơ cấp có thể dựa trên lựa chọn biến đổi nhiều lần (multiple transform selection, MTS), và khi biến đổi nhiều lần được áp dụng như biến đổi sơ cấp, nó có thể được tham chiếu là biến đổi lõi nhiều lần.

Biến đổi lõi nhiều lần có thể biểu diễn phương pháp biến đổi bổ sung sử dụng biến đổi cosin rời rạc (discrete cosine transform, DCT) kiểu 2 và biến đổi sin rời rạc (discrete sine transform, DST) kiểu 7, DCT kiểu 8, và/hoặc DST kiểu 1. Nghĩa là, biến đổi lõi 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 lựa chọn trong số DCT kiểu 2, DST kiểu 7, DCT kiểu 8 và DST kiểu 1. Ở đây, hệ số biến đổi sơ cấp có thể được gọi là hệ số biến đổi trung gian từ quan điểm của bộ biến đổi.

Nói cách khác, khi phương pháp biến đổi thông thường được áp dụng, thì các hệ số biến đổi có thể được tạo ra bằng cách áp dụng biến đổi từ miền không gian sang miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT kiểu 2. Không giống như điều này, khi biến đổi lõi nhiều lần được áp dụng, thì các hệ số biến đổi (hoặc các hệ số biến đổi sơ cấp) có thể được tạo ra bằng cách áp dụng biến đổi từ miền không gian sang miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT kiểu 2, DST kiểu 7, DCT kiểu 8, và/hoặc DST kiểu 1. Ở đây, DCT kiểu 2, DST kiểu 7, DCT kiểu 8, và DST kiểu 1 có thể được gọi là kiểu biến đổi,

nhân biến đổi hoặc lõi biến đổi. Các kiểu biến đổi DCT/DST này có thể được xác định dựa trên các hàm cơ sở.

Khi biến đổi lõi nhiều lần được thực hiện, thì nhân biến đổi dọc và nhân biến đổi ngang cho khối đích có thể được lựa chọn trong số các nhân biến đổi, biến đổi dọc có thể được thực hiện trên khối đích dựa trên nhân biến đổi dọc và một phép ngang biến đổi có thể được thực hiện trên khối đích dựa trên nhân biến đổi ngang. Ở đây, biến đổi ngang có thể chỉ ra biến đổi trên các thành phần ngang của khối đích và biến đổi dọc có thể chỉ ra biến đổi trên các thành phần dọc của khối đích. Nhân biến đổi dọc/nhân biến đổi ngang có thể được xác định một cách thích ứng dựa trên chế độ dự đoán và/hoặc chỉ số biến đổi cho khối đích (CU hoặc khối con) gồm khối phần dư.

Hơn nữa, theo một ví dụ, nếu biến đổi sơ cấp được thực hiện bằng cách áp dụng MTS, thì mối quan hệ ánh xạ cho các nhân biến đổi có thể được cài đặt bằng cách cài đặt các hàm cơ sở cụ thể thành các giá trị được xác định trước và kết hợp các hàm cơ sở cần được áp dụng trong biến đổi dọc hoặc biến đổi ngang. Ví dụ, khi nhân biến đổi ngang được diễn tả là trTypeHor và nhân biến đổi hướng dọc được diễn tả là trTypeVer, thì giá trị trTypeHor hoặc trTypeVer là 0 có thể được cài đặt thành DCT2, giá trị trTypeHor hoặc trTypeVer là 1 có thể được cài đặt thành DST7, và giá trị trTypeHor hoặc trTypeVer là 2 có thể được đặt thành DCT8.

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

Trong 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 suy ra các hệ số biến đổi (thứ cấp) được sửa đổi bằng cách thực hiện biến đổi thứ cấp dựa trên các hệ số biến đổi (sơ cấp) (S420). Biến đổi sơ cấp là biến đổi từ miền không gian sang miền tần số, và biến đổi thứ cấp đề cập đến quá trình biến đổi thành phương trình nén hơn nữa bằng cách sử dụng mối tương quan có mặt giữa các hệ số biến đổi (sơ cấp). Biến đổi thứ cấp có thể gồm biến đổi không tách được. Trong trường hợp này, biến đổi thứ cấp có thể được gọi là biến đổi thứ cấp không tách được (non-separable secondary transform, NSST), hoặc biến đổi thứ cấp không tách được phụ thuộc chế độ (mode-dependent non-separable secondary transform, MDNSST). Biến đổi thứ cấp không tách được có thể biểu diễn biến đổi mà tạo ra các hệ số biến đổi được sửa 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 được, các hệ số biến đổi (sơ cấp) được suy ra từ biến đổi sơ cấp. Tại thời điểm này, biến đổi dọc và biến đổi ngang có thể không được áp dụng riêng biệt (hoặc các biến đổi ngang và dọc có thể không được áp dụng độc lập) cho các hệ số biến đổi (sơ cấp), nhưng các biến đổi có thể được áp dụng cùng một lúc dựa trên ma trận biến đổi không tách được. Nói cách khác, biến đổi thứ cấp không tách được có thể biểu diễn phương pháp biến đổi mà trong đó các thành phần dọc và ngang của hệ số biến đổi (sơ cấp) không được tách ra, và ví dụ, các tín hiệu hai chiều (các hệ số biến đổi) được sắp xếp lại thành tín hiệu một chiều qua hướng được xác định nhất định (ví dụ, hướng theo hàng thứ nhất hoặc hướng theo cột thứ nhất),

và sau đó các hệ số biến đổi được sửa đổi (hoặc hệ số biến đổi thứ cấp) được tạo ra dựa trên ma trận biến đổi không tách được. Ví dụ, theo thứ tự theo hàng thứ nhất, M x N khối được bố trí trên dòng theo thứ tự của hàng thứ nhất, hàng thứ hai, …, và hàng thứ N. Theo thứ tự thứ nhất theo cột, M x N khối được bố trí theo dòng theo thứ tự của cột thứ nhất, cột thứ hai, …, và cột thứ N. Biến đổi thứ cấp không tách được có thể được áp dụng cho vùng trên cùng bên trái của khối được tạo cấu hình với hệ số biến đổi (sơ cấp) (sau đây, có thể được tham chiếu là là khối hệ số biến đổi). Ví dụ, nếu chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi đều bằng hoặc lớn hơn 8, thì biến đổi thứ cấp không tách được 8x8 có thể được áp dụng cho vùng 8x8 trên cùng bên trái của khối hệ số biến đổi. Hơn nữa, nếu chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi đều bằng hoặc lớn hơn 4 và chiều rộng (W) hoặc chiều cao (H) của khối hệ số biến đổi nhỏ hơn 8, thì biến đổi thứ cấp không tách được 4x4 có thể được áp dụng cho vùng min(8, W) x min(8, H) trên cùng bên trái của khối hệ số biến đổi. Tuy nhiên, phương án không bị giới hạn với điều này, và ví dụ, ngay cả khi chỉ thỏa mãn điều kiện rằng chiều rộng (W) hoặc chiều cao (H) của khối hệ số biến đổi bằng hoặc lớn hơn 4, thì biến đổi thứ cấp không tách được 4x4 có thể được áp dụng cho vùng min(8, W)×min(8, H) trên cùng bên trái của khối hệ số biến đổi.

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

Khối đầu vào X 4x4 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 dưới dạng véc-tơ, thì véc-tơ $\vec{X}$ có thể được biểu diễn như dưới đây.

[Phương trình 2]

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

Trong phương trình 2, véc-tơ $\vec{X}$ là véc-tơ một chiều được thu nhận bằng cách sắp xếp khối hai chiều X của phương trình 1 theo thứ tự theo hàng thứ nhất.

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

[Phương trình 3]

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

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

Qua phương trình 3 đã nói ở trên, véc-tơ hệ số biến đổi $\vec{F}$ 16x1 có thể được suy ra, và $\vec{F}$ có thể được tổ chức lại thành khối 4x4 qua thứ tự quét (chiều ngang, chiều dọc, đường chéo và tương tự). Tuy nhiên, sự tính toán được mô tả ở trên là một ví dụ, và biến đổi hypercube-Givens (HyGT) hoặc kiểu tương tự có thể được sử dụng cho sự suy ra của biến đổi thứ cấp không tách được nhằm giảm độ phức tạp tính toán của biến đổi thứ cấp không tách được.

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

Như được mô tả ở trên, biến đổi thứ cấp không tách được có thể được thực hiện dựa trên biến đổi 8x8 hoặc biến đổi 4x4 được xác định dựa trên chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi. Biến đổi 8x8 đề cập đến biến đổi áp dụng cho vùng 8x8 được gồm trong khối hệ số biến đổi khi cả W và H đều bằng hoặc lớn

hơn 8, và vùng 8x8 có thể là vùng 8x8 trên cùng bên trái trong khối hệ số biến đổi. Tương tự, biến đổi 4x4 đề cập đến biến đổi áp dụng cho vùng 4x4 được gồm trong khối hệ số biến đổi khi cả W và H bằng hoặc lớn hơn 4, và vùng 4x4 có thể là vùng 4x4 trên cùng bên trái trong khối hệ số biến đổi. Ví dụ, ma trận nhân biến đổi 8x8 có thể là ma trận 64x64/16x64, và ma trận nhân biến đổi 4x4 có thể là ma trận 16x16/8x16.

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

Tuy nhiên, vì kích thước của biến đổi, nghĩa là, kích thước của vùng mà biến đổi được áp dụng, chẳng hạn, có thể là kích thước khác 8 x 8 hoặc 4 x 4, nên số lượng các tập hợp có thể là n, và số lượng nhân biến đổi trong mỗi tập hợp có thể là k.

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

Để tham khảo, ví dụ, chế độ nội dự đoán có thể gồm hai chế độ nội dự đoán không định hướng (hoặc không góc) và 65 chế độ nội dự đoán định hướng (hoặc có góc). Các chế độ nội dự đoán không định hướng có thể 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 định hướng có thể gồm 65 chế độ nội dự đoán từ số 2 đến số 66. Tuy nhiên, đây là một ví dụ, và sáng chế có thể được áp dụng ngay cả khi số lượng các chế độ nội dự đoán khác nhau. Trong khi đó, trong một số trường hợp, chế độ nội dự đoán số 67 có thể được sử dụng thêm, và chế độ nội dự đoán số 67 có thể biểu diễn chế độ mô hình tuyến tính (linear model, LM).

Fig.5 thể hiện ví dụ các chế độ định hướng nội ảnh của 65 hướng dự đoán.

Tham chiếu đến Fig.5, trên cơ sở chế độ nội dự đoán 34 có hướng dự đoán chéo lên bên trái, các chế độ nội dự đoán có thể được chia thành các chế độ nội dự đoán có hướng ngang và chế độ nội dự đoán có hướng dọc. Trên Fig.5, H và V lần lượt biểu thị hướng ngang và hướng dọc, và các chữ số từ -32 đến 32 chỉ ra các dịch chuyển trong 1/32 đơn vị trên vị trí lưới mẫu. Các chữ số này có thể biểu diễn phần bù cho giá trị chỉ số chế độ. Các chế độ nội dự đoán từ 2 đến 33 có hướng ngang, và các chế độ nội dự đoán từ 34 đến 66 có hướng dọc. Nói đúng ra, chế độ nội dự đoán 34 có thể được coi là không theo chiều ngang cũng như chiều dọc, nhưng có thể được phân loại là thuộc về hướng ngang trong việc xác định tập hợp biến đổi của biến đổi thứ cấp. Điều này là do dữ liệu đầu vào được chuyển vị để được sử dụng cho chế độ hướng dọc đối xứng trên cơ sở của chế độ nội dự đoán 34 và phương pháp căn chỉnh dữ liệu đầu vào cho chế độ ngang được sử dụng cho chế độ nội dự đoán 34. Chuyển vị dữ liệu đầu vào có nghĩa là các hàng và các cột của dữ liệu khối M x N hai chiều được chuyển thành dữ liệu N x M. Chế độ nội dự đoán 18 và chế độ nội dự đoán 50 có thể biểu diễn chế độ nội dự đoán ngang và chế độ nội dự đoán dọc, và chế độ nội dự đoán 2 có thể được tham chiếu là chế độ nội dự đoán đường chéo hướng lên phía phải vì chế độ nội dự đoán 2 có pixel tham chiếu bên trái và thực hiện dự đoán theo hướng lên phía phải. Tương tự như vậy, chế độ nội dự đoán 34 có thể được gọi là chế độ nội dự đoán theo đường chéo hướng xuống phía phải

và chế độ nội dự đoán 66 có thể được gọi là chế độ nội dự đoán theo đường chéo hướng xuống phía trái.

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

[Bảng 2]

IfnstPredModeIntra	IfnstTrSetIdx
IfnstPredModeIntra < 0	1
0 <= IfnstPredModeIntra <= 1	0
2 <= IfnstPredModeIntra <= 12	1
13 <= IfnstPredModeIntra <= 23	2
24 <= IfnstPredModeIntra <= 44	3
45 <= IfnstPredModeIntra <= 55	2
56 <= IfnstPredModeIntra <= 80	1
81 <= IfnstPredModeIntra <= 83	0

Như được thể hiện trong bảng 2, bất kỳ một trong bốn bộ biến đổi, nghĩa là, lfnstTrSetIdx, có thể được ánh xạ tới bất kỳ một trong bốn chỉ số, nghĩa là, từ 0 đến 3, theo chế độ nội dự đoán.

Khi được xác định được rằng một tập hợp cụ thể được sử dụng cho biến đổi không tách được, một trong k nhân biến đổi trong tập hợp cụ thể đó có thể được lựa chọn thông qua chỉ số biến đổi thứ cấp không tách được. Thiết bị mã hóa có thể suy ra chỉ số biến đổi thứ cấp không tách được chỉ ra một nhân biến đổi cụ thể dựa trên kiểm tra tỷ lệ-biến dạng (rate-distortion, RD) và có thể tín hiệu hóa chỉ số biến đổi thứ cấp không tách được cho thiết bị giải mã. Thiết bị giải mã có thể lựa chọn một trong k nhân biến đổi trong tập cụ thể dựa trên chỉ số biến đổi thứ cấp không tách được. Ví dụ, giá trị chỉ số lfnst 0 có thể tham chiếu đến nhân biến đổi thứ cấp không tách được thứ nhất, giá trị chỉ số lfnst 1 có thể tham chiếu đến nhân biến đổi thứ cấp không tách được thứ hai và giá trị chỉ số lfnst 2 có thể tham chiếu đến nhân biến đổi thứ cấp không tách được thứ ba. Ngoài ra, giá trị chỉ số lfnst 0 có thể chỉ ra rằng biến đổi thứ cấp không tách được thứ nhất không được áp dụng cho khối đích, và các giá trị chỉ số lfnst từ 1 đến 3 có thể chỉ ra ba nhân biến đổi.

Bộ biến đổi có thể thực hiện biến đổi thứ cấp không tách được dựa trên các nhân biến đổi được lựa chọn, và có thể thu nhận các hệ số biến đổi (thứ cấp) được sửa đổi. Như được mô tả ở trên, các hệ số biến đổi được sửa đổi có thể được suy ra là các hệ số biến đổi được lượng tử hóa thông qua bộ lượng tử hóa, và có thể được mã hóa và được tín hiệu hóa đến thiết bị giải mã và được chuyển đến bộ biến đổi ngược/bộ giải lượng tử hóa 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), mà là đầu ra của biến đổi sơ cấp (tách được), có thể được suy ra 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 tín hiệu hóa đến thiết bị giải mã và được chuyển đến bộ giải lượng tử hóa/bộ biến đổi ngược trong thiết bị mã hóa.

Bộ biến đổi ngược có thể thực hiện một loạt các quy trình theo thứ tự ngược mà trong đó chúng đã được thực hiện trong bộ biến đổi được mô tả ở trên. Bộ biến đổi ngược có thể nhận các hệ số biến đổi (được giải lượng tử hóa), và suy ra các hệ số biến đổi (sơ cấp) bằng cách thực hiện biến đổi (ngược) thứ cấp (S450) và có thể thu nhận khối phần dư (các mẫu phần dư) bằng cách thực hiện biến đổi (ngược) sơ cấp trên các hệ số biến đổi (sơ cấp) (S460). Trong mối quan hệ này, các hệ số biến đổi sơ cấp có thể được gọi là hệ số biến đổi được sửa đổi theo quan điểm của bộ biến đổi ngược. Như được mô tả ở trên, thiết bị mã hóa và thiết bị giải mã có thể tạo ra khối được dựng lại dựa trên khối phần dư và khối được dự đoán, và có thể tạo ra ảnh được dựng lại dựa trên khối được dựng lại.

Bộ giải mã có thể còn gồm bộ xác định áp dụng biến đổi ngược thứ cấp (hoặc phần tử để xác định liệu có áp dụng biến đổi ngược thứ cấp hay không) và bộ xác định biến đổi ngược thứ cấp (hoặc phần tử để xác định biến đổi ngược thứ cấp). Bộ xác định áp dụng biến đổi ngược thứ cấp có thể xác định liệu có áp dụng biến đổi ngược thứ cấp hay không. Ví dụ, biến đổi ngược thứ cấp có thể là NSST, RST, hoặc LFNST và bộ xác định áp dụng biến đổi ngược thứ cấp có thể xác định liệu có

áp dụng biến đổi ngược thứ cấp hay không dựa trên cờ biến đổi thứ cấp được thu nhận bằng cách phân tích cú pháp dòng bit. Trong một ví dụ khác, bộ xác định áp dụng biến đổi ngược thứ cấp có thể xác định liệu có áp dụng biến đổi ngược thứ cấp hay không dựa trên hệ số biến đổi của khối phần dư.

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

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

Trong khi đó, theo sáng chế, biến đổi thứ cấp rút gọn (RST) mà trong đó kích thước của ma trận (nhân) biến đổi được rút gọn 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 được.

Trong khi đó, nhân biến đổi, ma trận biến đổi, và hệ số cấu thành ma trận nhân biến đổi, nghĩa là, hệ số nhân hoặc hệ số ma trận, được mô tả trong sáng chế có thể được diễn tả bằng 8 bit. Đây có thể là điều kiện cho cách thực hiện trong thiết bị giải mã và thiết bị mã hóa, và có thể làm giảm dung lượng bộ nhớ cần thiết để

lưu trữ nhân biến đổi với sự suy giảm hiệu suất mà có thể được cung cấp hợp lý so với 9 bit hoặc 10 bit hiện có. Ngoài ra, việc biểu thị của ma trận nhân bằng 8 bit có thể cho phép hệ số nhân nhỏ được sử dụng, và có thể phù hợp hơn cho các lệnh dữ liệu nhiều lệnh đơn lẻ (single instruction multiple data, SIMD) được sử dụng cho cách thực hiện phần mềm tối ưu.

Trong sáng chế, thuật ngữ "RST" có thể có nghĩa là một biến đổi mà được thực hiện trên các mẫu phần dư cho khối đích dựa trên ma trận biến đổi có kích thước được rút gọn theo nhân tử rút gọn. Trong trường hợp thực hiện biến đổi rút gọn, lượng tính toán cần thiết cho biến đổi có thể được giảm xuống do sự giảm kích thước của ma trận biến đổi. Nghĩa là, RST có thể được sử dụng để giải quyết vấn đề phức tạp suy ra xảy ra ở biến đổi không tách được hoặc biến đổi của khối có kích thước lớn.

RST có thể được tham chiếu là các thuật ngữ khác nhau, chẳng hạn như biến đổi rút gọn, biến đổi thứ cấp rút gọn, biến đổi rút gọn, biến đổi đơn giản hóa, biến đổi đơn giản, và các thuật ngữ tương tự, và tên mà RST có thể được tham chiếu không bị giới hạn với các ví dụ được liệt kê. Ngoài ra, vì RST chủ yếu được thực hiện trong vùng tần số thấp gồm hệ số khác không trong khối biến đổi, nên nó có thể được tham chiếu là biến đổi không tách được tần số thấp (Low-Frequency Non-Separable Transform, LFNST). Chỉ số biến đổi có thể được gọi là chỉ số LFNST.

Trong khi đó, khi biến đổi ngược thứ cấp được thực hiện dựa trên RST, bộ biến đổi ngược 235 của thiết bị mã hóa 200 và bộ biến đổi ngược 322 của thiết bị giải mã 300 có thể gồm bộ biến đổi thứ cấp rút gọn ngược mà suy ra các hệ số biến đổi được sửa đổi dựa trên RST ngược của các hệ số biến đổi, và bộ biến đổi sơ cấp ngược mà suy ra các mẫu phần dư cho khối đích dựa trên biến đổi sơ cấp ngược của các hệ số biến đổi được sửa đổi. Biến đổi sơ cấp ngược đề cập đến biến đổi ngược của biến đổi sơ cấp được áp dụng cho phần dư. Trong sáng chế, suy ra hệ số biến

đổi dựa trên biến đổi có thể đề cập đến sự suy ra các hệ số biến đổi bằng cách áp dụng biến đổi.

Fig.6 là sơ đồ minh họa biến đổi thứ cấp rút gọn (reduced secondary transform, RST) theo một phương án của sáng chế.

Trong sáng chế, "khối đích" có thể đề cập đến khối hiện tại được tạo mã, khối phần dư, hoặc khối biến đổi.

Trong RST theo một ví dụ, véc-tơ N chiều có thể được ánh xạ tới véc-tơ R chiều được đặt trong một không gian khác, do đó, ma trận biến đổi rút gọn có thể được xác định, trong đó R nhỏ hơn N. N có thể có nghĩa là bình phương của độ dài cạnh của khối mà biến đổi được áp dụng, hoặc số lượng tổng của các hệ số biến đổi tương ứng với khối mà biến đổi được áp dụng, và nhân tử rút gọn có thể có nghĩa là giá trị R/N. Nhân tử rút gọn có thể được gọi là nhân tử rút gọn, nhân tử giảm, nhân tử đơn giản hóa, nhân tử đơn giản, hoặc nhiều thuật ngữ khác. Trong khi đó, R có thể được tham chiếu là hệ số rút gọn, nhưng tùy theo các tình huống, nhân tử rút gọn có thể có nghĩa là R. Hơn nữa, tùy theo các tình huống, nhân tử rút gọn có thể có nghĩa là giá trị N/R.

Trong một ví dụ, nhân tử rút gọn hoặc hệ số rút gọn có thể được tín hiệu hóa qua dòng bit, nhưng ví dụ không bị giới hạn với điều này. Ví dụ, giá trị được xác định trước cho nhân tử rút gọn hoặc hệ số rút gọn có thể được lưu trữ trong mỗi thiết bị mã hóa 200 và thiết bị giải mã 300, và trong trường hợp này, nhân tử rút gọn hoặc hệ số rút gọn có thể không được tín hiệu hóa riêng biệt.

Kích thước của ma trận biến đổi rút gọn theo một ví dụ có thể là RxN nhỏ hơn N x N, kích thước của ma trận biến đổi thông thường, và có thể được định nghĩa như trong phương trình 4 bên dưới.

[Phương trình 4]

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

Ma trận T trong khối biến đổi rút gọn được thể hiện trên Fig.6A có thể có nghĩa là ma trận T_{RxN} của phương trình 4. Như được thể hiện trên Fig.6A, khi ma trận biến đổi rút gọn T_{RxN} được nhân với các mẫu phần dư cho khối đích, thì các hệ số biến đổi cho khối đích có thể được suy ra.

Trong một ví dụ, nếu kích thước của khối mà biến đổi được áp dụng là 8x8 và R = 16 (tức là, R/N = 16/64 = 1/4), thì RST theo Fig.6A có thể được diễn tả là phép toán ma trận như được thể hiện trong phương trình 5 bên dưới. Trong trường hợp này, bộ nhớ và phép tính nhân có thể được giảm xuống khoảng 1/4 bởi nhân tử rút gọn.

Theo sáng chế, phép toán ma trận có thể được hiểu là phép toán nhân của véc-tơ cột với ma trận, được bố trí ở bên trái véc-tơ cột, để thu nhận véc-tơ 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 tạo ra bằng cách áp dụng biến đổi sơ cấp. Kết quả phép tính của phương trình 5, các hệ số biến đổi ci cho khối đích có thể được suy ra, và quá trình suy ra ci có thể như trong phương trình 6.

[Phương trình 6]

```
for i from to R:
    ci=0
    for j from 1 to N
        ci += tij * rj
```

Kết quả phép tính của phương trình 6, các hệ số biến đổi c_1 đến c_R cho khối đích có thể được suy ra. Nghĩa là, khi R = 16, thì các hệ số biến đổi c_1 đến c_{16} cho khối đích có thể được suy ra. Nếu, thay vì RST, biến đổi thường được áp dụng và ma trận biến đổi có kích thước 64x64 (N x N) được nhân với các mẫu phần dư có kích thước 64x1 (Nx1), thì chỉ có 16 (R) hệ số biến đổi được suy ra cho khối đích vì RST đã được áp dụng, mặc dù 64 (N) hệ số biến đổi được suy ra cho khối đích. Vì số lượng tổng của các hệ số biến đổi cho khối đích được giảm từ N xuống R, lượng dữ liệu được truyền bởi thiết bị mã hóa 200 đến thiết bị giải mã 300 giảm, do đó hiệu suất truyền giữa thiết bị mã hóa 200 và thiết bị giải mã 300 có thể được cải thiện.

Khi xem xét từ quan điểm về kích thước của ma trận biến đổi, thì kích thước của ma trận biến đổi thường là 64x64 (N x N), nhưng kích thước của ma trận biến đổi rút gọn được giảm xuống 16x64 (RxN), do đó mức sử dụng bộ nhớ trong trường hợp thực hiện RST có thể được giảm xuống với tỷ lệ R/N khi so sánh với trường hợp thực hiện biến đổi thường. Ngoài ra, khi so sánh với số lượng các phép tính nhân N x N trong trường hợp sử dụng ma trận biến đổi thường, việc sử dụng ma trận biến đổi rút gọn có thể làm giảm số lượng các phép tính nhân với tỷ lệ R/N (RxN).

Trong một ví dụ, bộ biến đổi 232 của thiết bị mã hóa 200 có thể suy ra các hệ số biến đổi cho khối đích bằng cách thực hiện biến đổi sơ cấp và biến đổi thứ cấp dựa trên RST 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 suy ra các hệ số biến đổi được sửa đổi dựa trên biến đổi thứ cấp rút gọn (reduced secondary transform, RST) ngược cho các hệ số biến đổi, và có thể suy ra các mẫu phần dư cho khối đích dựa trên biến đổi sơ cấp ngược cho các hệ số biến đổi được sửa đổi.

Kích thước của ma trận RST ngược T_{NxR} theo ví dụ là NxR nhỏ hơn kích thước N x N của ma trận biến đổi ngược thường, và có mối quan hệ chuyển vị với ma trận biến đổi rút gọn T_{RxN} được thể hiện trong phương trình 4.

Ma trận T^t trong Inv rút gọn. Khối biến đổi được thể hiện trên Fig.6(b) có thể có nghĩa là ma trận RST ngược T_{RxN}^T (chỉ số trên T có nghĩa là chuyển vị). Khi ma trận RST ngược T_{RxN}^T được nhân với các hệ số biến đổi cho khối đích như được thể hiện trên Fig.6 (b), thì các hệ số biến đổi được sửa đổi cho khối đích hoặc các mẫu phần dư cho khối hiện tại có thể được suy ra. Ma trận RST ngược T_{RxN}^T có thể được biểu diễn dưới dạng $(T_{RxN}^T)_{NxR}$.

Cụ thể hơn, khi RST ngược được áp dụng làm biến đổi ngược thứ cấp, thì các hệ số biến đổi được sửa đổi cho khối đích có thể được suy ra khi ma trận RST ngược T_{RxN}^T được nhân với các hệ số biến đổi cho khối đích. Trong khi đó, RST ngược có thể được áp dụng làm biến đổi sơ cấp ngược, và trong trường hợp này, các mẫu phần dư cho khối đích có thể được suy ra khi ma trận RST ngược T_{RxN}^T được nhân với các hệ số biến đổi cho khối đích.

Trong một ví dụ, nếu kích thước của khối mà biến đổi ngược được áp dụng là 8x8 và R = 16 (tức là, R/N = 16/64 = 1/4), thì RST theo Fig.6B có thể được diễn tả là phép toán ma trận như được thể hiện trong phương trình 7 dưới đây.

[Phương trình 7]

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

Trong phương trình 7, c_1 đến c_{16} có thể biểu diễn các hệ số biến đổi cho khối đích. Kết quả phép toán của phương trình 7, r_i biểu diễn các hệ số biến đổi được sửa đổi cho khối đích hoặc các mẫu phần dư cho khối đích có thể được suy ra, và quá trình suy ra r_i có thể như trong phương trình 8.

[Phương trình 8]

```
For i from 1 to N
    ri=0
     for j from 1 to R
          ri += tji * cj
```

Kết quả phép tính của phương trình 8, r_1 đến r_N biểu diễn các hệ số biến đổi được sửa đổi cho khối đích hoặc các mẫu phần dư cho khối đích có thể được suy ra. Khi xem xét từ quan điểm về kích thước của ma trận biến đổi ngược, thì kích thước của ma trận biến đổi ngược thường là 64x64 (NxN), nhưng kích thước của ma trận biến đổi ngược rút gọn được giảm thành 64x16 (RxN), do đó mức sử dụng bộ nhớ trong trường hợp thực hiện RST ngược có thể được giảm với tỷ lệ R/N khi so sánh với trường hợp thực hiện biến đổi ngược thường. Ngoài ra, khi so sánh với số lượng các phép tính nhân N x N trong trường hợp sử dụng ma trận biến đổi ngược thường, sự sử dụng ma trận biến đổi ngược rút gọn có thể làm giảm số lượng các phép tính nhân với tỷ lệ R/N (NxR).

Cấu hình tập hợp biến đổi được thể hiện trong bảng 2 cũng có thể được áp dụng cho RST 8 x 8. Nghĩa là, RST 8 x 8 có thể được áp dụng theo tập hợp biến đổi

trong bảng 2. Vì một tập hợp biến đổi gồm hai hoặc ba biến đổi (nhân) theo chế độ nội dự đoán, nên nó có thể được tạo cấu hình để lựa chọn một trong tối đa bốn biến đổi gồm biến đổi mà trong trường hợp đó không có biến đổi thứ cấp nào được áp dụng. Trong biến đổi mà không có biến đổi thứ cấp nào được áp dụng, nó có thể được coi là áp dụng ma trận nhận dạng. Giả sử rằng các chỉ số 0, 1, 2, và 3 tương ứng được gán cho bốn biến đổi (ví dụ, chỉ số 0 có thể được phân bổ cho trường hợp mà ma trận nhận dạng được áp dụng, nghĩa là, trường hợp mà không có biến đổi thứ cấp nào được áp dụng), thì chỉ số lfnst như phần tử cú pháp có thể được tín hiệu hóa cho mỗi khối hệ số biến đổi, bằng cách đó chỉ định biến đổi được áp dụng. Nghĩa là, đối với khối 8 x 8 trên cùng bên trái, thông qua chỉ số biến đổi, có thể chỉ định RST 8 x 8 trong cấu hình RST, hoặc chỉ định lfnst 8 x 8 khi LFNST được áp dụng. 8 x 8 lfnst và 8 x 8 RST đề cập đến các biến đổi áp dụng cho vùng 8 x 8 được gồm trong khối hệ số biến đổi khi cả W và H của khối đích được biến đổi đều bằng hoặc lớn hơn 8, và vùng 8 x 8 có thể là vùng 8 x 8 trên cùng bên trái trong khối hệ số biến đổi. Tương tự, lfnst 4 x 4 và RST 4 x 4 đề cập đến các biến đổi áp dụng cho vùng 4 x 4 được gồm trong khối hệ số biến đổi khi cả W và H của khối đích bằng hoặc lớn hơn 4 và vùng 4 x 4 có thể là vùng 4 x 4 trên cùng bên trái trong khối hệ số biến đổi.

Theo một phương án của sáng chế, đối với biến đổi trong quá trình mã hóa, chỉ có 48 phần dữ liệu có thể được lựa chọn và ma trận nhân biến đổi 16 x 48 cực đại có thể được áp dụng cho nó, thay vì áp dụng ma trận nhân biến đổi 16 x 64 cho 64 phần dữ liệu tạo thành vùng 8 x 8. Ở đây, "cực đại" có nghĩa là m có giá trị cực đại là 16 trong ma trận nhân biến đổi m x 48 để tạo ra m hệ số. Nghĩa là, khi RST được thực hiện bằng cách áp dụng ma trận nhân biến đổi m x 48 ($m \geq 16$) cho vùng 8 x 8, thì 48 phần dữ liệu được đưa vào và m hệ số được tạo ra. Khi m là 16, thì 48 phần dữ liệu được đưa vào và 16 hệ số được tạo ra. Nghĩa là, giả sử rằng 48 phần dữ liệu tạo thành véc-tơ 48×1, ma trận 16×48 và véc-tơ 48×1 được nhân tuần tự,

bằng cách đó tạo ra véc-tơ 16×1. Ở đây, 48 phần dữ liệu tạo thành vùng 8 x 8 có thể được sắp xếp đúng cách, bằng cách đó tạo thành véc-tơ 48 x 1. Ví dụ, một véc-tơ 48 x 1 có thể được xây dựng dựa trên 48 phần dữ liệu cấu thành vùng không gồm vùng 4 x 4 dưới cùng bên phải trong số các vùng 8 x 8. Ở đây, khi phép toán ma trận được thực hiện bằng cách áp dụng ma trận nhân biến đổi 16 x 48 cực đại, thì 16 hệ số biến đổi được sửa đổi được tạo ra, và 16 hệ số biến đổi được sửa đổi có thể được sắp xếp trong vùng 4 x 4 trên cùng bên trái theo thứ tự quét, và vùng 4 x 4 trên cùng bên phải và vùng 4 x 4 dưới cùng bên trái có thể được điền với các số không.

Đối với biến đổi ngược trong quá trình giải mã, ma trận chuyển vị của ma trận nhân biến đổi đã nói ở trên có thể được sử dụng. Nghĩa là, khi RST hoặc LFNST ngược được thực hiện trong quá trình biến đổi ngược được thực hiện bởi thiết bị giải mã, thì dữ liệu hệ số đầu vào mà RST ngược được áp dụng được tạo cấu hình trong véc-tơ một chiều theo thứ tự sắp xếp được xác định trước, và véc-tơ hệ số được sửa đổi được thu nhận bằng cách nhân véc-tơ một chiều và ma trận RST ngược tương ứng ở bên trái của véc-tơ một chiều có thể được sắp xếp trong khối hai chiều theo thứ tự sắp xếp được xác định trước.

Tóm lại, trong quá trình biến đổi, khi RST hoặc LFNST được áp dụng cho vùng 8 x 8, thì phép toán ma trận gồm 48 hệ số biến đổi ở các vùng trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của vùng 8 x 8 trừ ra vùng dưới cùng bên phải trong số các hệ số biến đổi trong vùng 8 x 8 và ma trận nhân biến đổi 16 x 48. Đối với phép toán ma trận, 48 hệ số biến đổi được đưa vào mảng một chiều. Khi phép toán ma trận được thực hiện, thì 16 hệ số biến đổi được sửa đổi được suy ra, và các hệ số biến đổi được sửa đổi có thể được sắp xếp ở vùng trên cùng bên trái của vùng 8 x 8.

Ngược lại, trong quá trình biến đổi ngược, khi RST hoặc LFNST ngược được áp dụng cho vùng 8 x 8, thì 16 hệ số biến đổi tương ứng với vùng trên cùng bên trái của vùng 8 x 8 trong số các hệ số biến đổi trong vùng 8 x 8 có thể được đưa

vào mảng một chiều theo thứ tự quét và có thể chịu phép toán ma trận với ma trận nhân biến đổi 48 x 16. Tức là, phép toán ma trận có thể được diễn tả là (ma trận 48 x 16) * (véc-tơ hệ số biến đổi 16 x 1) = (véc-tơ hệ số biến đổi được sửa đổi 48 x 1). Ở đây, véc-tơ n x 1 có thể được hiểu là có cùng ý nghĩa với một ma trận n x 1 và do đó có thể được diễn tả là véc-tơ cột nx1. Hơn nữa, * biểu thị phép nhân ma trận. Khi phép toán ma trận được thực hiện, thì 48 hệ số biến đổi được sửa đổi có thể được suy ra, và 48 hệ số biến đổi được sửa đổi có thể được sắp xếp ở các vùng trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của vùng 8x8 trừ ra vùng dưới cùng bên phải.

Khi biến đổi ngược thứ cấp dựa trên RST, thì bộ biến đổi ngược 235 của thiết bị mã hóa 200 và bộ biến đổi ngược 322 của thiết bị giải mã 300 có thể gồm bộ biến đổi thứ cấp rút gọn ngược để suy ra các hệ số biến đổi được sửa đổi dựa trên RST ngược trên các hệ số biến đổi và bộ biến đổi sơ cấp ngược để suy ra các mẫu phần dư cho khối đích dựa trên biến đổi sơ cấp ngược trên các hệ số biến đổi được sửa đổi. Biến đổi sơ cấp ngược đề cập đến biến đổi ngược của biến đổi sơ cấp được áp dụng cho phần dư. Theo sáng chế, việc suy ra hệ số biến đổi dựa trên biến đổi có thể đề cập đến việc suy ra hệ số biến đổi bằng cách áp dụng biến đổi.

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

Thiết bị mã hóa nhận một kết quả (hoặc một phần của kết quả) được suy ra sau khi áp dụng một biến đổi sơ cấp (lõi) làm đầu vào, và áp dụng biến đổi thứ cấp thuận (biến đổi thứ cấp).

[Phương trình 9]

$$y = G^T x$$

Trong phương trình 9, x và y lần lượt là các đầu vào và các đầu ra của biến đổi thứ cấp, và G là ma trận biểu diễn biến đổi thứ cấp và các véc-tơ cơ sở của biến

đổi được gồm là các véc-tơ cột. Trong trường hợp của LFNST ngược, khi kích thước của ma trận biến đổi G được biểu thị là [số lượng hàng x số lượng cột], thì trong trường hợp của LFNST thuận, chuyển vị của ma trận G trở thành kích thước của G^T.

Đối với LFNST ngược, 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 từng phần mà 8 véc-tơ cơ sở biến đổi được lấy mẫu từ bên trái của ma trận [48 x 16] và ma trận [16 x 16] tương ứng.

Mặt khác, đối với LFNST thuận, 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 từng phần được thu nhận bằng cách lấy mẫu 8 véc-tơ cơ sở biến đổi từ đỉnh của ma trận [16 x 48] và ma trận [16 x 16] tương ứng.

Do đó, trong trường hợp LFNST thuận, véc-tơ [48 x 1] hoặc véc-tơ [16 x 1] có thể là đầu vào x, và véc-tơ [16 x 1] hoặc véc-tơ [8 x 1] có thể là đầu ra y. Trong mã hóa và giải mã viđeo, đầu ra của biến đổi sơ cấp thuận là dữ liệu hai chiều (two-dimensional, 2D), do đó để xây dựng véc-tơ [48 x 1] hoặc véc-tơ [16 x 1] làm đầu vào x, véc-tơ một chiều phải được xây dựng bằng cách sắp xếp hợp lý dữ liệu 2D là đầu ra của biến đổi thuận.

Fig.7 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của biến đổi sơ cấp thuận thành véc-tơ một chiều theo một ví dụ. Các sơ đồ bên trái của (a) và (b) của Fig.7 thể hiện trình tự xây dựng véc-tơ [48 x 1], và các sơ đồ bên phải của (a) và (b) của Fig.7 thể hiện trình tự xây dựng véc-tơ [16 x 1]. Trong trường hợp của LFNST, véc-tơ một chiều x có thể thu được bằng cách sắp xếp tuần tự dữ liệu 2D theo thứ tự như trong (a) và (b) của Fig.7.

Hướng sắp xếp dữ liệu đầu ra của biến đổi sơ cấp thuận 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 theo hướng ngang đối với hướng chéo, thì dữ liệu đầu ra của biến đổi

sơ cấp thuận có thể được sắp xếp theo thứ tự (a) của Fig.7, và khi chế độ nội dự đoán của khối hiện tại theo hướng dọc đối với hướng chéo, thì dữ liệu đầu ra của biến đổi sơ cấp thuận có thể được sắp xếp theo thứ tự (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 suy ra cùng một kết quả (véc-tơ y) như khi các thứ tự sắp xếp (a) và (b) Fig.7 được áp dụng, các véc-tơ cột của ma trận G có thể được sắp xếp lại theo thứ tự sắp xếp. Nghĩa là, có thể sắp xếp lại các véc-tơ cột của G để mỗi phần tử tạo thành véc-tơ x luôn được nhân với cùng một véc-tơ cơ sở biến đổi.

Vì đầu ra y được suy ra từ phương trình 9 là véc-tơ một chiều, khi dữ liệu hai chiều được yêu cầu như dữ liệu đầu vào trong quá trình sử dụng kết quả của biến đổi thứ cấp thuận làm đầu vào, ví dụ, trong quá trình thực hiện lượng tử hóa hoặc tạo mã phần dư, véc-tơ y đầu ra của phương trình 9 phải được sắp xếp lại hợp lệ dưới dạng dữ liệu 2D.

Fig.8 là sơ đồ minh họa trình tự sắp xếp dữ liệu đầu ra của biến đổi thứ cấp thuận thành véc-tơ hai chiều theo một ví dụ.

Trong trường hợp của LFNST, các giá trị đầu ra có thể được sắp xếp trong khối 2D theo thứ tự quét được xác định trước. (a) của Fig.8 thể hiện rằng khi đầu ra y là vector [16 x 1], thì các giá trị 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 vector [8 x 1], thì các giá trị 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 ra rằng nó được điền với số không.

Theo một ví dụ khác, vì thứ tự mà véc-tơ đầu ra y được xử lý khi thực hiện lượng tử hóa hoặc tạo mã phần dư có thể được đặt trước, nên véc-tơ đầu ra y có thể không được sắp xếp trong khối 2D như được thể hiện trên Fig.8. Tuy nhiên, trong trường hợp tạo mã phần dư, tạo mã dữ liệu có thể được thực hiện trong các đơn vị khối 2D (ví dụ, 4x4) chẳng hạn như CG (Coefficient Group, nhóm hệ số), và trong

trường hợp này, dữ liệu được sắp xếp theo một thứ tự cụ thể như theo thứ tự quét đường chéo của Fig.8.

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

[Phương trình 10]

$$x = Gy$$

Trong trường hợp của LFNST ngược, véc-tơ đầu ra x có thể được suy ra bằng cách nhân véc-tơ đầu vào y, mà là véc-tơ [16 x 1] hoặc véc-tơ [8 x 1], với ma trận G. Đối với LFNST ngược, véc-tơ đầu ra x có thể là véc-tơ [48 x 1] hoặc véc-tơ [16 x 1].

Véc-tơ đầu ra x được sắp xếp trong một khối hai chiều theo thứ tự được thể hiện trên Fig.7 và được sắp xếp dưới dạng dữ liệu hai chiều, và dữ liệu hai chiều này trở thành dữ liệu đầu vào (hoặc một phần của dữ liệu đầu vào) của biến đổi sơ cấp ngược.

Theo đó, biến đổi thứ cấp ngược ngược lại với toàn bộ quá trình biến đổi thứ cấp thuận, và trong trường hợp của biến đổi ngược, không giống như theo chiều thuận, biến đổi thứ cấp ngược được áp dụng trước tiên, sau đó áp dụng biến đổi sơ cấp ngược.

Trong LFNST ngược, một trong 8 ma trận [48 x 16] và 8 ma trận [16 x 16] có thể được lựa chọn làm ma trận biến đổi G. Việc áp dụng ma trận [48 x 16] hay ma trận [16 x 16] tùy thuộc về kích thước và hình dạng của khối.

Ngoài ra, 8 ma trận có thể được suy ra từ bốn tập hợp biến đổi như được thể hiện trong bảng 2 ở trên, và mỗi tập hợp biến đổi có thể gồm hai ma trận. Bộ biến đổi nào sẽ sử dụng trong số 4 bộ biến đổi được xác định theo chế độ nội dự đoán, và cụ thể hơn, bộ biến đổi được xác định dựa trên giá trị của chế độ nội dự đoán được

mở rộng bằng cách xem xét nội dự đoán góc rộng (Wide Angle Intra Prediction, WAIP). Ma trận nào để lựa chọn trong số hai ma trận cấu thành tập hợp biến đổi được lựa chọn được suy ra qua sự tín hiệu hóa chỉ số. Cụ thể hơn, 0, 1, và 2 có thể là giá trị chỉ số được truyền, 0 có thể chỉ ra rằng LFNST không được áp dụng, và 1 và 2 có thể chỉ ra bất kỳ một trong hai ma trận biến đổi tạo thành tập hợp biến đổi được lựa chọn dựa trên giá trị chế độ nội dự đoán.

Fig.9 là sơ đồ minh họa các chế độ nội dự đoán góc rộng theo một phương án của sáng chế.

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

Khi chiều rộng của khối hiện tại dự đoán lớn hơn chiều cao, thì các pixel tham chiếu phía trên thường gần hơn với các vị trí bên trong khối cần được dự đoán. Do đó, dự đoán theo hướng dưới cùng bên trái có thể chính xác hơn là theo hướng trên cùng bên phải. Ngược lại, khi chiều cao của khối lớn hơn chiều rộng, thì các pixel tham chiếu bên trái thường gần với các vị trí bên trong khối cần được dự đoán. Do đó, dự đoán theo hướng trên cùng bên phải có thể chính xác hơn so với hướng dưới cùng bên trái. Do đó, có thể có lợi khi áp dụng sự ánh xạ lại, tức là sửa đổi chỉ số chế độ, cho chỉ số của chế độ nội dự đoán góc rộng.

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

Bảng 3 dưới đây thể hiện quá trình suy ra chế độ nội ảnh được sửa đổi bằng cách ánh xạ lại chế độ nội dự đoán thành chế độ nội dự đoán góc rộng.

[Bảng 3]

Các đầu vào cho quá trình là:
- Biến preModeIntra chỉ định chế độ nội dự đoán.
- Biến nTbW chỉ định chiều rộng khối biến đổi.
- Biến nTbH chỉ định chiều cao khối biến đổi,
- Biến cIdx chỉ định thành phần màu của khối hiện tại.

Đầu ra của quá trình này là chế độ nội dự đoán predModeIntra được sửa đổi.

Các biến nW và nH được dẫn xuất như sau:
- Nếu IntraSubPartitionsSplitType bằng \ ISP_NO_SPLIT hoặc cIdx không bằng 0
  điều sau đây áp dụng:

$$nW = nTbW \quad (8\text{-}97)$$

$$nH = nTbH \quad (8\text{-}98)$$

- Nếu không (IntraSubPartitionsSplitType không bằng ISP_NO_SPLIT hoặc cIdx bằng 0).
  điều sau đây áp dụng:

$$nW = nCbW \quad (8\text{-}99)$$

$$nH = nCbH \quad (8\text{-}100)$$

Biến whRatio được đặt bằng Abs(Log2(nW / nH)).
Đối với các khối không vuông (nW không bằng nH), chế độ nội dự đoán predModeIntra được sửa đổi như sau:
- Nếu tất cả các điều kiện sau đây là đúng, thì predModeIntra được đặt bằng (predModeIntra + 65).
  - nW lớn hơn nH
  - predModeIntra lớn hơn hoặc bằng 2
  - predModeIntra nhỏ hơn (whRatio > 1) ? (8 + 2 * whRatio) : 8

- Nếu không, nếu tất cả các điều kiện sau đây là đúng, thì predModeIntra được đặt bằng (predModeIntra - 67).
  - nH lớn hơn nW
  - predModeIntra lớn hơn hoặc bằng 66
  - predModeIntra nhỏ hơn (whRatio > 1) ? (60 - 2 * whRatio) : 60

Trong bảng 3, giá trị chế độ nội dự đoán mở rộng được lưu trữ cuối cùng trong biến predModeIntra, và ISP_NO_SPLIT chỉ ra rằng khối CU không được chia thành các phân vùng con (Intra Sub Partitions, ISP) bằng kỹ thuật nội phân chia con hiện được áp dụng trong tiêu chuẩn VVC, và các giá trị biến cIdx 0, 1, và 2 chỉ ra trường hợp của các thành phần luma, Cb, và Cr tương ứng. Hàm Log2 được thể hiện trong bảng 3 trả về giá trị log cơ số là 2 và hàm Abs trả về giá trị tuyệt đối.

Biến predModeIntra chỉ ra chế độ nội dự đoán và chiều cao và chiều rộng của khối biến đổi, v.v. được sử dụng làm các giá trị đầu vào của quá trình ánh xạ chế độ nội dự đoán góc rộng,và giá trị đầu ra là chế độ dự đoán bên trong được sửa đổi predModeIntra. Chiều cao và chiều rộng của khối biến đổi hoặc khối tạo mã có thể là chiều cao và chiều rộng của khối hiện tại cho sự ánh xạ lại của chế độ nội dự đoán. Tại thời điểm này, biến whRatio phản ánh tỷ số giữa chiều rộng với chiều rộng có thể được đặt thành Abs(Log2(nW / nH)).

Đối với khối không vuông, chế độ nội dự đoán có thể được chia thành hai trường hợp và được sửa đổi.

Thứ nhất, nếu tất cả các điều kiện (1) ~ (3) được thỏa mãn, (1) chiều rộng của khối hiện tại lớn hơn chiều cao, (2) chế độ nội dự đoán trước khi sửa đổi bằng hoặc lớn hơn 2, (3) chế độ nội dự đoán nhỏ hơn giá trị được suy ra từ (8 + 2 * whRatio) khi biến whRatio lớn hơn 1 và nhỏ hơn 8 khi biến whRatio nhỏ hơn hoặc bằng 1 [predModeIntra nhỏ hơn (whRatio > 1)? (8 + 2 * whRatio): 8], thì chế độ nội dự đoán được đặt thành giá trị lớn hơn 65 so với chế độ nội dự đoán [predModeIntra được đặt bằng (predModeIntra + 65)].

Nếu khác với điều trên, nghĩa là, theo các điều kiện (1) ~ (3) được thỏa mãn, (1) chiều cao của khối hiện tại lớn hơn chiều rộng, (2) chế độ nội dự đoán trước khi sửa đổi nhỏ hơn hoặc bằng 66, (3) chế độ nội dự đoán lớn hơn giá trị được suy ra từ (60 - 2 * whRatio) khi biến whRatio lớn hơn 1, và lớn hơn 60 khi biến whRatio nhỏ hơn hoặc bằng 1 [predModeIntra lớn hơn (whRatio> 1)? (60 - 2 * whRatio): 60], thì chế độ nội dự đoán được đặt thành giá trị nhỏ hơn 67 so với chế độ nội dự đoán [predModeIntra được đặt bằng (predModeIntra - 67)].

Bảng 2 ở trên thể hiện cách một tập hợp biến đổi được lựa chọn dựa trên giá trị 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 đối xứng với hướng dự đoán xung quanh chế độ 34. Ví dụ, chế độ 14 và chế độ 54 là đối xứng theo hướng

tương ứng với chế độ 34. Do đó, cùng biến đổi được áp dụng cho các chế được đặt theo các hướng đối xứng lẫn nhau, và sự đối xứng này cũng được phản ánh trong bảng 2.

Trong khi đó, giả định rằng dữ liệu đầu vào LFNST thuận cho chế độ 54 là đối xứng với dữ liệu đầu vào LFNST thuận cho chế độ 14. Ví dụ, đối 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, tương ứng. Ngoài ra, có thể thấy rằng các mẫu theo thứ tự được thể hiện trong (a) của Fig.7 và (b) của Fig.7 đối xứng với hướng (hướng chéo) được chỉ ra bởi chế độ 34.

Trong khi đó, như được mô tả ở trên, ma trận biến đổi nào trong các ma trận [48 x 16] và ma trận [16 x 16] được áp dụng cho LFNST được xác định bởi kích thước và hình dạng của khối đích biến đổi.

Fig.10 là một sơ đồ minh họa hình dạng khối mà LFNST được áp dụng. (a) của FIG.10 thể hiện các khối 4 x 4, (b) thể hiện các khối 4 x 8 và 8 x 4, (c) thể hiện các khối 4 x N hoặc N x 4 trong đó N là 16 trở lên, (d) thể hiện các khối 8 x 8, (e) thể hiện các khối M x N trong đó $M \leq 8$, $N \leq 8$, và $N > 8$ hoặc $M > 8$.

Trên FIG.10, các khối có các đường bao dày chỉ ra các vùng mà LFNST được áp dụng. Đối với các khối của các hình Fig.10 (a) và Fig.10(b), LFNST được áp dụng cho vùng 4x4 trên cùng bên trái, và đối với khối của Fig.10 (c), LFNST được áp dụng riêng lẻ hai vùng 4x4 trên cùng bên trái được sắp xếp liên tục. Trong (a), (b), và (c) của Fig.10, vì LFNST được áp dụng cho các đơn vị của các vùng 4x4, LFNST này sau đây sẽ được tham chiếu là "LFNST 4x4". Dựa trên kích thước ma trận cho G, ma trận [16 x 16] hoặc [16 x 8] có thể được áp dụng.

Cụ thể hơn, ma trận [16 x 8] được áp dụng cho khối 4x4 (4x4 TU hoặc 4x4 CU) của Fig.10 (a) và ma trận [16 x 16] được áp dụng cho các khối trong (b) và (c) của Fig.10. Điều này là để điều chỉnh độ phức tạp suy ra trong trường hợp xấu nhất thành 8 phép nhân trên mỗi mẫu.

Đối với (d) và (e) của Fig.10, LFNST được áp dụng cho vùng 8x8 trên cùng bên trái, và LFNST này sau đây được tham chiếu là "LFNST 8x8". Là ma trận biến đổi tương ứng, ma trận [48 x 16] hoặc [48 x 8] có thể được áp dụng. Trong trường hợp của LFNST thuận, vì véc-tơ [48 x 1] (véc-tơ x trong phương trình 9) được đưa vào làm dữ liệu đầu vào, nên tất cả các giá trị mẫu của vùng 8x8 trên cùng bên trái không được sử dụng làm các giá trị đầu vào của LFNST thuận. Nghĩa là, như có thể thấy trong thứ tự bên trái của Fig.7 (a) hoặc thứ tự bên trái của Fig.7 (b), véc-tơ [48 x 1] có thể được xây dựng dựa trên các mẫu thuộc 3 khối 4x4 còn lại trong khi vẫn giữ nguyên khối 4x4 dưới cùng bên phải.

Ma trận [48 x 8] có thể được áp dụng cho khối 8×8 (8×8 TU hoặc 8×8 CU) trên Fig.10 (d), và ma trận [48 × 16] có thể được áp dụng cho khối 8×8 trên Fig.10 (e). Điều này cũng để điều chỉnh độ phức tạp suy ra trong trường hợp xấu nhất thành 8 phép nhân trên mỗi mẫu.

Tùy thuộc vào hình dạng khối, khi áp dụng LFNST thuận tương ứng (LFNST 4x4 hoặc LFNST 8x8), thì dữ liệu đầu ra 8 hoặc 16 (véc-tơ y trong phương trình 9, véc-tơ [8 x 1] hoặc [16 x 1]) được tạo ra. Trong LFNST thuận, số lượng dữ liệu đầu ra bằng hoặc ít hơn số lượng dữ liệu đầu vào do các đặc điểm của ma trận GT.

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

Khu vực bị che ở trên cùng bên trái của khối được thể hiện trên Fig.11 tương ứng với khu vực mà dữ liệu đầu ra của LFNST thuận được đặt, các vị trí được đánh dấu với 0 chỉ ra các mẫu được điền với các giá trị 0, và vùng còn lại biểu diễn các vùng không bị thay đổi bởi LFNST thuận. Trong vùng không bị thay đổi bởi LFNST, dữ liệu đầu ra của biến đổi sơ cấp thuận vẫn không thay đổi.

Như được mô tả ở trên, vì kích thước của ma trận biến đổi được áp dụng thay đổi tùy theo hình dạng của khối, nên số lượng dữ liệu đầu ra cũng khác nhau. Như Fig.11, dữ liệu đầu ra của LFNST thuận có thể không điền hoàn toàn khối 4x4 trên cùng bên trái. Trong trường hợp (a) và (d) của Fig.11, ma trận [16 x 8] và ma trận [48 x 8] được áp dụng cho khối được chỉ ra bởi đường dày hoặc vùng bên trong khối, tương ứng, và véc-tơ [8 x 1] là đầu ra của LFNST thuận được tạo ra. Nghĩa là, theo thứ tự quét được thể hiện trong (b) của Fig.8, chỉ có 8 dữ liệu đầu ra có thể được điền như được thể hiện trong (a) và (d) của Fig.11, và 0 có thể được điền vào 8 vị trí còn lại. Trong trường hợp khối được áp dụng LFNST của Fig.10 (d), như được thể hiện trên Fig.11 (d), hai khối 4x4 ở trên cùng bên phải và dưới cùng bên trái liền kề với khối 4x4 trên cùng bên trái cũng được điền với các giá trị 0.

Như được mô tả ở trên, về cơ bản, bằng cách tín hiệu hóa chỉ số LFNST, việc áp dụng LFNST và ma trận biến đổi được áp dụng sẽ được chỉ định. Như được thể hiện Fig.11, khi LFNST được áp dụng, vì số lượng dữ liệu đầu ra của LFNST thuận có thể bằng hoặc ít hơn số lượng dữ liệu đầu vào, thì vùng được điền với giá trị 0 sẽ xảy ra như sau.

1) Như được thể hiện trong (a) của Fig.11, các mẫu từ vị trí thứ 8 trở lên theo thứ tự quét trong khối 4x4 trên cùng bên trái, tức là các mẫu từ thứ 9 đến ngày 16.

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

Do đó, nếu dữ liệu khác 0 có mặt bằng cách kiểm tra các vùng 1) và 2), chắc chắn rằng LFNST không được áp dụng, do đó có thể bỏ qua sự báo hiệu của chỉ số LFNST tương ứng.

Theo một ví dụ, ví dụ, trong trường hợp LFNST được thông qua trong tiêu chuẩn VVC, vì tín hiệu của chỉ số LFNST được thực hiện sau khi tạo mã phần dư, thiết bị mã hóa có thể biết liệu có dữ liệu khác 0 (các hệ số có nghĩa) cho tất cả các vị trí trong phạm vi khối TU hoặc CU qua tạo mã phần dư. Theo đó, thiết bị mã hóa có thể xác định liệu có thực hiện sự tín hiệu hóa trên chỉ số LFNST hay không dựa trên sự có mặt của dữ liệu khác 0, và thiết bị giải mã có thể xác định liệu chỉ số LFNST có được phân tích cú pháp hay không. Khi dữ liệu khác 0 không có mặt trong khu vực được chỉ định trong 1) và 2) ở trên, thì sự tín hiệu hóa chỉ số LFNST được thực hiện.

Vì mã đơn phân cụt được áp dụng làm phương pháp nhị phân hóa cho chỉ số LFNST, nên chỉ số LFNST gồm tối đa hai bin (nhị phân), và 0, 10, và 11 được gán là các mã nhị phân cho các giá trị chỉ số LFNST có thể có là 0, 1, và 2, tương ứng. Trong trường hợp LFNST hiện được áp dụng cho VVC, tạo mã CABAC dựa trên ngữ cảnh được áp dụng cho bin thứ nhất (mã hóa thường). và mã hóa bỏ qua được áp dụng cho bịn thứ hai. Tổng số lượng các ngữ cảnh cho bin thứ nhất là 2, khi (DCT-2, DCT-2) được áp dụng làm cặp biến đổi sơ cấp cho các hướng ngang và dọc, và thành phần luma và thành phần chroma được tạo mã trong kiểu cây kép, thì một ngữ cảnh được phân bổ và một ngữ cảnh khác áp dụng cho các trường hợp còn lại. Tạo mã của chỉ số LFNST được thể hiện trong bảng như sau.

[Bảng 4]

Phần tử cú pháp	binIdx					
	0	1	2	3	4	>= 5
lfnst_idx[ ][ ]	(tu_mts_idx[x0][y0] == 0 && treeType != SINGLE_TREE) ? 1 : 0	bypass	na	na	na	na

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

(i) Theo một ví dụ, số lượng dữ liệu đầu ra cho LFNST thuận có thể bị giới hạn ở giá trị tối đa là 16.

Trong trường hợp (c) của Fig.10, LFNST 4x4 có thể được áp dụng cho hai vùng 4x4 liền kề ở trên cùng bên trái, và trong trường hợp này, tối đa 32 dữ liệu đầu ra LFNST có thể được tạo ra. khi số lượng dữ liệu đầu ra cho LFNST thuận bị giới hạn ở mức tối đa là 16, thì trong trường hợp của các khối (TU hoặc CU) 4xN/Nx4 ($N \leq 16$), LFNST 4x4 chỉ được áp dụng cho một vùng 4x4 ở trên cùng bên trái, LFNST chỉ có thể chỉ được áp dụng một lần cho tất cả các khối của Fig.10. Qua đó, cách thực hiện của tạo mã hình ảnh có thể được đơn giản hóa.

Fig.12 thể hiện rằng số lượng dữ liệu đầu ra cho LFNST thuận được giới hạn ở giá trị tối đa là 16 theo một ví dụ. Như Fig.12, khi LFNST được áp dụng cho vùng 4x4 trên cùng bên trái nhất trong khối 4xN hoặc Nx4 trong đó N là 16 trở lên, thì dữ liệu đầu ra của LFNST thuận sẽ trở thành 16 phần.

(ii) Theo một ví dụ, sự đặt bằng không có thể được áp dụng bổ sung cho khu vực mà LFNST không được áp dụng. Trong sáng chế, đặt bằng không có thể có nghĩa là điền các giá trị của tất cả các vị trí thuộc một vùng cụ thể với giá trị 0. Nghĩa là, sự đặt bằng không có thể được áp dụng cho vùng mà không bị thay đổi do LFNST và duy trì kết quả của biến đổi sơ cấp thuận. Như được mô tả ở trên, vì LFNST được chia thành LFNST 4x4 và LFNST 8x8, nên sự đặt bằng không có thể được chia thành hai loại ((ii) - (A) và (ii) - (B)) như sau.

(ii) - (A) Khi LFNST4x4 được áp dụng, thì vùng mà LFNST 4x4 không được áp dụng có thể được đặt bằng không (0). Fig.13 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 4x4 được áp dụng theo một ví dụ.

Như được thể hiện trên Fig.13, đối với khối mà LFNST 4x4 được áp dụng, nghĩa là, đối với tất cả các khối trong (a), (b), và (c) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng có thể được điền với các số không.

Mặt khác, (d) của Fig.13 thể hiện rằng khi giá trị lớn nhất của đầu ra của LFNST thuận được giới hạn ở 16 như được thể hiện trên Fig.12, sự đặt bằng không được thực hiện trên các khối còn lại mà LFNST 4x4 không được áp dụng.

(ii) - (B) Khi LFNST8x8 được áp dụng, thì vùng mà LFNST 8x8 không được áp dụng có thể bị xóa. Fig.14 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 8x8 được áp dụng theo một ví dụ.

Như được thể hiện trên Fig.14, đối với khối mà LFNST 8x8 được áp dụng, nghĩa là, đối với tất cả các khối trong (d) và (e) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng có thể được điền với các số không.

(iii) Do sự đặt bằng không được trình bày trong (ii) ở trên, khu vực được điền với các số không có thể không giống khi LFNST được áp dụng. Theo đó, có thể kiểm tra xem dữ liệu khác 0 có mặt hay không theo sự đặt bằng không được đề xuất trong (ii) trên khu vực rộng hơn so với trường hợp LFNST của Fig.11.

Ví dụ, khi (ii) - (B) được áp dụng, thì sau khi kiểm tra xem dữ liệu khác 0 có mặt hay không trong đó vùng được điền với các giá trị 0 trong (d) và (e) của Fig.11 ngoài vùng được điền với 0 bổ sung trên Fig.14, sự tín hiệu hóa cho chỉ số LFNST chỉ có thể được thực hiện khi dữ liệu khác 0 không có mặt.

Tất nhiên, ngay cả khi sự đặt bằng không được đề xuất trong (ii) được áp dụng, vẫn có thể kiểm tra xem dữ liệu khác 0 có mặt hay không theo cùng cách như tín hiệu hóa chỉ số LFNST có mặt. Nghĩa là, sau khi kiểm tra xem dữ liệu khác 0 có mặt trong khối được điền với các số không trên Fig.11, sự tín hiệu hóa chỉ số LFNST có thể được áp dụng. Trong trường hợp này, thiết bị mã hóa chỉ thực hiện sự đặt bằng không và thiết bị giải mã không giả định sự đặt bằng 0, nghĩa là, chỉ kiểm tra liệu dữ liệu khác 0 có chỉ có mặt trong khu vực được đánh dấu rõ ràng là 0 trên Fig.11 hay không, có thể thực hiện phân tích chỉ số LFNST.

Ngoài ra, theo một ví dụ khác, sự đặt bằng không có thể được thực hiện như trên Fig.15. Fig.15 là sơ đồ minh họa sự đặt bằng không trong khối mà LFNST 8x8 được áp dụng theo một ví dụ khác.

Như được thể hiện trên các hình Fig.13 và Fig.14, sự đặt bằng không có thể được áp dụng cho tất cả các vùng khác với vùng mà LFNST được áp dụng, hoặc sự

đặt bằng không có thể chỉ được áp dụng cho vùng cục bộ như được thể hiện trên Fig.15. Sự đặt bằng không chỉ được áp dụng cho các vùng khác với vùng 8x8 trên cùng bên trái của Fig.15, sự đặt bằng không có thể không được áp dụng cho khối 4x4 dưới cùng bên phải trong vùng 8x8 trên cùng bên trái.

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

Phương án 1

- Giới hạn số lượng dữ liệu đầu ra đối với LFNST thuận đến giá trị tối đa là 16 → (i)

- khi LFNST 4x4 được áp dụng, thì tất cả các khu vực mà LFNST 4x4 không được áp dụng được đặt bằng không → (ii)-(A)

- Khi LFNST 8x8 được áp dụng, tất cả các khu vực mà LFNST 8x8 không được áp dụng được đặt bằng không → (ii)-(B)

- Sau khi kiểm tra xem dữ liệu khác không có mặt hay không cũng như khu vực có mặt được điền với giá trị không và khu vực đó được điền với các số không hay không do các sự đặt bằng không bổ sung ((ii)-(A), (ii)-(B)), chỉ số LFNST chỉ được tín hiệu hóa khi dữ liệu khác không không có mặt → (iii)

Trong trường hợp của phương án 1, khi LFNST được áp dụng, vùng mà dữ liệu đầu ra khác 0 có thể có mặt được giới hạn ở bên trong vùng 4x4 trên cùng bên trái. Chi tiết hơn, trong trường hợp của Fig.13 (a) và Fig.14 (a), vị trí thứ 8 theo thứ tự quét là vị trí cuối cùng mà dữ liệu khác 0 có thể có mặt. Trong trường hợp của Fig.13 (b) và (c) và Fig.14 (b), vị trí thứ 16 theo thứ tự quét (tức là vị trí của cạnh dưới cùng bên phải của khối 4x4 trên cùng bên trái) là vị trí cuối cùng mà dữ liệu khác 0 có thể có mặt.

Do đó, khi LFNST được áp dụng, thì sau khi kiểm tra xem dữ liệu khác 0 có mặt hay không ở vị trí mà quá trình tạo mã phần dư không được phép (ở vị trí ngoài vị trí cuối cùng), có thể xác định liệu chỉ số LFNST có được tín hiệu hóa hay không.

Phần sau đây là bảng thể hiện quy trình LFNST theo phương án 1.

[Bảng 5]

7.3.7.11 Cú pháp tạo mã phần dư

...	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) ||	
(lastScanPos > 7 && (log2TbWidth == 2 || log2TbHeight == 3) && log2TbWidth == log2TbHeight)) {	
lfnstZeroOutSigCoeffFlag = 1	
}	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder[log2TbWidth − log2SbW][log2TbHeight − log2SbH][i][0]	
yS = DiagScanOrder[log2TbWidth − log2SbW][log2TbHeight − log2SbH][i][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = −1	
remBinsPass1 = ((log2SbW + log2SbH) < 4 ? 8 : 32)	
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]) {	
if(!transform_skip_flag[x0][y0]) {	
numSigCoeff++	
}	
...	

[Bảng 6]

7.3.7.5 Cú pháp đơn vị tạo mã

...	
numSigCoeff = 0	
lfnstZeroOutSigCoeffFlag = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && !intra_mip_flag[x0][y0]) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && lfnstZeroOutSigCoeffFlag == 0)	
lfnst_idx[x0][y0]	ae(v)
}	
}	

Các bảng 5 và 6 trình bày quá trình giải mã cho (ii) và (iii) của phương pháp đơn giản hóa LFNST.

Theo phần trên của bảng 5, khi chỉ số của khối con trong đó có mặt hệ số khác 0 lớn hơn 0, và chiều rộng và chiều cao của khối biến đổi lớn hơn 4 [(lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2)], hoặc vị trí cuối cùng của hệ số khác 0 trong phạm vi khối con mà hệ số khác 0 cuối cùng có mặt lớn hơn 7, và kích thước của khối biến đổi là 4x4 hoặc 8x8[(lastScanPos > 7 && (log2TbWidth == 2 | | log2TbHeight == 3) && log2TbWidth == log2TbHeight)], biến cờ lfnstZeroOutSigCoeffFlag cho sự đặt bằng không của LFNST được đặt thành 1.

Nghĩa là, điều kiện thứ nhất là điều kiện mà các hệ số khác 0 được suy ra từ các vùng khác với vùng trên cùng bên trái mà LFNST có thể được áp dụng trong khối biến đổi (nghĩa là, khi các hệ số có nghĩa trong các khối con khác với khối con (4x4) trên cùng bên trái được suy ra). Khi điều kiện thứ nhất được thỏa mãn, biến cờ lfnstZeroOutSigCoeffFlag cho sự đặt bằng không của LFNST được đặt thành 1. Trong trường hợp này, vì kích thước của khối biến đổi là 4x4 hoặc lớn hơn, nó chỉ

ra rằng sự đặt bằng không được thực hiện trong khối biến đổi mà LFNST 4x4 và LFNST 8x8 được áp dụng.

Vì điều kiện thứ hai là điều kiện mà khi LFNST được áp dụng cho khối 4x4 và khối 8x8, thì vị trí cuối cùng mà các hệ số khác 0 có thể có mặt là vị trí thứ 8 như được thể hiện trong (a) và (d) của Fig.11, khi hệ số khác 0 có mặt bên ngoài vị trí thứ 7 khi bắt đầu từ 0, thì biến cờ lfnstZeroOutSigCoeffFlag được đặt thành 1.

Do đó, khi biến cờ lfnstZeroOutSigCoeffFlag được đặt thành 1, như được thể hiện trong bảng 6, lfnst_idx được tín hiệu hóa ở cấp đơn vị tạo mã khi lfnstZeroOutSigCoeffFlag bằng 0, không được tín hiệu hóa.

Theo một ví dụ khác, khi điều kiện thứ nhất hoặc thứ hai của bảng 5 được thỏa mãn, thì biến cờ lfnstZeroOutSigCoeffFlag được đặt thành 0, và lfnst_idx có thể được tín hiệu hóa khi lfnstZeroOutSigCoeffFlag là 1.

[Bảng 7]

8.7.4 Quá trình biến đổi cho các hệ số biến đổi được tỷ lệ hóa

8.7.4.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí luma (xTbY, yTbY) chỉ định mẫu trên cùng bên trái của khối biến đổi luma hiện tại so với mẫu luma trên cùng của ảnh hiện tại,
- biến nTbW chỉ định chiều rộng của khối biến đổi hiện tại,
- biến nTbH chỉ định chiều cao của khối biến đổi hiện tại,
- biến cIdx chỉ định thành phần màu của khối hiện tại,
- mảng d[x][y] (nTbW)x(nTbH) của các hệ số biến đổi được tỷ lệ hóa với x = 0..nTbW - 1, y = 0..nTbH - 1.

Đầu ra của quá trình này là mảng r[x][y] (nTbW)x(nTbH)của các mẫu phần dư với x = 0..nTbW - 1, y = 0..nTbH - 1.

Khi lfnst_idx[xTbY][yTbY] không bằng 0 và cả nTbW và nTbH đều lớn hơn hoặc bằng 4, điều sau đây áp dụng:

- Các biến predModeIntra, nLfnstOutSize, log2LfnstSize, nLfnstSize, numLfnstX, numLfnstY, và nonZeroSize được suy ra như sau:

```
predModeIntra =
( cIdx = = 0 ) ? IntraPredModeY[ xTbY ][ yTbY ] : IntraPredModeC[ xTbY ][ yTbY ]
nLfnstOutSize = ( nTbW >= 8 && nTbH >= 8 ) ? 48 : 16
log2LfnstSize = ( nTbW >= 8 && nTbH >= 8 ) ? 3 : 2
nLfnstSize = 1 << log2LfnstSize
nonZeroSize = ( ( nTbW = = 4 && nTbH = = 4 ) || ( nTbW = = 8 && nTbH = = 8 )
) ? 8 : 16                                                    (8959)
```

- Quá trình ánh xạ chế độ nội dự đoán góc rộng được quy định trong điều 8.4.5.2.6 được dẫn ra với predModeIntra, nTbW, nTbH và cIdx làm các đầu vào, và predModeIntra được sửa đổi làm đầu ra.
- Các giá trị của danh sách u[x] với x = 0..nonZeroSize − 1 được suy ra như sau:

```
xC = DiagScanOrder[ log2LfnstSize ][ log2LfnstSize ][ x ][ 0 ]
```

```
yC = DiagScanOrder[ log2LfnstSize ][ log2LfnstSize ][ x ][ 1 ]
u[ x ] = d[ xC ][ yC ]
```

- Quá trình biến đổi không tách rời tần số thấp một chiều được quy định trong 8.7.4.2 được dẫn ra với chiều dài đầu vào của các hệ số biến đổi được tỷ lệ hóa nonSize, chiều dài đầu ra biến đổi nTrS được đặt bằng nLfntsOutSize, danh sách các hệ số biến đổi khách không được tỷ lệ hóa u[x] với x = 0..nonZeroSize - 1, chế độ nội dự đoán cho sự lựa chọn tập hợp LFNST predNodeItra, và chỉ số LFNST cho sựa lựa chọn biến đổi trong tập hợp LFNST được lựa chọn lfnst_idx[xTbY][yTbY] làm các đầu vào, và danh sách v[x] với x = 0....nLfnstOutSize - 1 làm đầu ra.

Mảng d[x][y] với x = 0....nLfnstSize - 1, y = 0....nLfnstSize - 1 được dẫn xuất như sau:

- Nếu predModeIntra nhỏ hơn hoặc bằng 34, hoặc bằng INTRA_LT_CCLM, INTRA_T_CCLM, hoặc INTRA_L_CCLM, thì điều sau đây áp dụng:

```
d[ x ][ y ] = ( y < 4 )  ?  v[ x + ( y << log2LfnstSize ) ]  :                    (8-963)
              ( ( x < 4 )  ?  v[ 32 + x + ( ( y − 4 ) << 2 ) ]  :  d[ x ][ y ] )
```

- Nếu không, thì điều sau đây áp dụng:

```
d[ x ][ y ] = ( x < 4 )  ?  v[ y + ( x << log2LfnstSize ) ]  :                    (8-964)
              ( ( y < 4 )  ?  v[ 32 + y + ( ( x − 4 ) << 2 ) ]  :  d[ x ][ y ] )
```

Biến implicitMtsEnabled được dẫn xuất như sau:

- Nếu sps_mts_enabled_flag bằng 1 và một trong các điều kiện sau đây là đúng, implicitMtsEnabled được đặt bằng 1:
  - InrtaSubPartitionsSplitType không bằng ISP_NO_SPLIT
  - cu_sbt_flag bằng 1 và Max(nTbW,nTbH) nhỏ hơn hoặc bằng 32
  - sps_explicit_mts_intra_enabled_flag và sps_explicit_mts_inter_enabled_flag đều bằng 0 và CuPredMode[xTbY][yTbY] bằng MODE_INTRA

- Nếu không, implicitMtsEnabled được đặt bằng 0.

Biến trTypeHor quy định nhân biến đổi ngang và biến trTypeVer quy định nhân biến đổi dọc được dẫn xuất như sau:

- Nếu cIdx lớn hơn 0, thì trTypyHor và trTyeVer được đặt bằng 0.
- Nếu không, nếu implicitMtsEnabled bằng 1, điều sau đây áp dụng:

  Nếu IntraSubPartitionSplitType không bằng ISP_NO_SPLIT hoặc cả sps_explicit_mts_intra_enabled_flag và sps_explicit_mts_inter_enabled_flag bằng 0 và CuPredMode[xTbY][yTbY] bằng MODE_INTRA, thì trTypyHor và trTyeVer được dẫn xuất như sau:

  trTypeHor = (nTbW >= 4 && nTbW <= 16) ? 1 : 0 (8-965)

  trTypeVer = (nTbH >= 4 && nTbH <= 16) ? 1 : 0 (8-966)

  - Nếu không, nếu cu_sbt_flag bằng 1, trTypyHor và trTyeVer được quy định trong Bảng 8-17 tùy thuộc vào cu_sbt_horizontal_flag và cu_sbt_pos_flag.
- Nếu không, trTypyHor và trTyeVer được quy định trong Bảng 8-16 tùy thuộc vào tu_mts_idx[xTbY][yTbY]

Các biến nonZeroW và nonZeroH được dẫn xuất như sau:

– Nếulfnst_idx[xTbY][yTbY] không bằng 0, điều sau đây áp dụng:

  nonZeroW = (nTbW == 4 || nTbH == 4) ? 4 : 8

  nonZeroH = (nTbW == 4 || nTbH == 4) ? 4 : 8

– Nếu không, điều sau đây áp dụng:

  nonZeroW = Min(nTbW, (trTypeHor > 0) ? 16 : 32) (8-967)

  nonZeroH = Min(nTbH, (trTypeVer > 0) ? 16 : 32) (8-968)

Tham khảo bảng 7, biến nonZeroSize chỉ ra kích thước hoặc số lượng của các hệ số khác 0 mà phép toán ma trận được thực hiện để áp dụng LFNST được đặt thành 8 hoặc 16. Khi chiều rộng và chiều cao của khối biến đổi là 4 hoặc 8, nghĩa là, như được thể hiện trên Fig.11, thì độ dài của dữ liệu đầu ra của LFNST thuận cho khối 4×4 và khối 8×8 hoặc dữ liệu đầu vào của LFNST ngược là 8. Đối với tất cả các khối khác, độ dài của dữ liệu đầu ra của LFNST thuận hoặc dữ liệu đầu vào của LFNST ngược là 16 [nonZeroSize = ((nTbW = = 4 && nTbH = = 4) ||

(nTbW = = 8 && nTbH = = 8)) ? 8 : 16]. Nghĩa là, khi LFNST thuận được áp dụng, thì số lượng tối đa của dữ liệu đầu ra được giới hạn ở 16.

Dữ liệu đầu vào của LFNST ngược này có thể được sắp xếp hai chiều theo sự quét đường chéo [xC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][0], yC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][1]]. Phần được mô tả ở trên thể hiện quá trình giải mã cho (i) của phương pháp đơn giản hóa LFNST.

Ngoài ra, các biến nonZeroW và nonZeroH chỉ ra chiều rộng và chiều cao của khối trên cùng bên trái mà trong đó các hệ số biến đổi khác 0 được đưa vào cho biến đổi sơ cấp ngược có thể có mặt, khi chỉ số lfnst khác 0, thì được suy ra là 4 nếu chiều rộng hoặc chiều rộng của khối biến đổi là 4, và nếu không được suy ra là 8 [nonZeroW = (nTbW == 4 | | nTbH == 4) ? 4 : 8, nonZeroH = (nTbW == 4 | | nTbH == 4) ? 4 : 8]. Nghĩa là, trong khối biến đổi, sự đặt bằng không không được thực hiện ở các vùng khác với vùng 4x4 và vùng 8x8 mà lfnst được áp dụng. Các khu vực này thể hiện quá trình giải mã cho (ii) của phương pháp đơn giản hóa LFNST.

Phương án 2

- Giới hạn số lượng dữ liệu đầu ra cho LFNST thuận ở giá trị tối đa là 16 → (i)

Trong phương án 2), vì sự đặt bằng không được đề xuất trong (ii) không được áp dụng, nên hương pháp tín hiệu hóa chỉ số LFNST theo phương pháp không kiểm tra xem dữ liệu khác 0 có mặt hay không được áp dụng. Nghĩa là, chỉ sự đặt bằng không được thể hiện trên Fig.11 được kiểm tra và chỉ số LFNST được tín hiệu hóa dựa trên điều này. Trường hợp của Fig.11 (c) khác với phương án 2 vì số lượng dữ liệu đầu ra là 32 (nghĩa là, trường hợp của phương án 2 như được thể hiện trên Fig.12) nhưng ngoại trừ việc dữ liệu đầu ra được đặt trong khối 4x4 trên cùng bên trái áp dụng tương tự không áp dụng cho việc đặt bằng không được đề xuất trong (ii).

Như Fig.12, khi LFNST 4x4 chỉ được áp dụng cho vùng 4x4 trên cùng bên trái và ma trận [16 x 16] được áp dụng cho vùng tương ứng, thì vùng được điền với các giá trị không sẽ không xảy ra. Vì khu vực để kiểm tra sự có mặt của dữ liệu khác 0 không thay đổi, nên phương pháp tín hiệu hóa chỉ số LFNST theo phương pháp không kiểm tra xem có mặt dữ liệu khác 0 hay không có thể được áp dụng.

[Bảng 8]

8.7.4 Quá trình biến đổi cho các hệ số biến đổi được tỷ lệ hóa

8.7.4.1 Tổng quát

Các đầu vào cho quá trình này là:

- vị trí luma (xTbY, yTbY) quy định mẫu trên cùng bên trái của khối biến đổi luma hiện tại so với mẫu luma trên cùng bên trái của ảnh hiện tại,
- biến nTbW quy định chiều rộng của khối biến đổi hiện tại,
- biến nTbH quy định chiều cao của khối biến đổi hiện tại,
- biến cIdx quy định thành phần màu của khối hiện tại,
- mảng d[x][y] (nTbW)x(nTbH) của các hệ số biến đổi được tỷ lệ hóa với x = 0..nTbW - 1.

Đầu ra của quá trình là mảng d[x][y] (nTbW)x(nTbH) của các mẫu phần dư với x = 0..nTbW - 1, y = 0..nTbH - 1.

Khi lfnst_idx[xTbY][yTbY] không bằng 0 và cả nTbW và nTbH đều lớn hơn hoặc bằng 4, thì điều sau đây áp dụng:

- Biến predModeIntra, nLfnstOutSize, log2LfnstSize, nLfnstSize,, numLfnstX, numLfnstY, và nonZeroSize được dẫn xuất như sau:

predModeIntra = (cIdx = = 0) ? IntraPredModeY[xTbY][yTbY] : IntraPredModeC[xTbY][yTbY] (8-953)

nLfnstOutSize = (nTbW >= 8 && nTbH >= 8) ? 48 : 16 (8-954)

log2LfnstSize = (nTbW >= 8 && nTbH >= 8) ? 3 : 2 (8-955)

nLfnstSize = 1 << log2LfnstSize (8-956)

nonZeroSize = ((nTbW = = 4 && nTbH = = 4) || (nTbW = = 8 && nTbH = = 8)) ? 8 : 16 (8-959)

- Quá trình ánh xạ chế độ nội dự đoán góc rộng được quy định trong điều 8.4.5.2.6 được dẫn ra với predModeIntra, nTbW, nTbH và cIdx làm đầu vào, và predModeIntra được sửa đổi làm đầu ra.

- Các giá trị của danh sách u[x] với x = 0..nonZeroSize - 1 được dẫn xuất như sau:

xC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][0] (8-960)

yC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][1] (8-961)

u[x] = d[xC][yC] (8-962)

- Quá trình biến đổi không tách rời tần số thấp một chiều như được quy định trong điều 8.7.4.1 được dẫn ra với chiều dài đầu vào của các hệ số biến đổi được tỷ lệ hóa nonZeroSize, chiều dài đầu ra biến đổi nTrS được đặt bằng nLfnstOutSize, danh sách của các hệ số biến đổi khác không được tỷ lệ hóa u[x] với x = 0..nonZeroSize - 1, chế độ nội dự đoán cho lựa chọn LFNST tập hợp LFNST predModeIntra, và chỉ siis LFNST cho lựa chọn biến đổi trong tập hợp LFNST được lựa chọn lfnst_idx[ơ xTbY][yTbY] làm các đầu vào, và danh sách v[x] với x = 0..nLfnstOutSize - 1 làm đầu ra.
- Mảng d[x][y] với x = 0..nLfnstSize - 1, y = 0..nLfnstSize - 1 được dẫn xuất như sau:
- Nếu predModeIntra nhỏ hơn hoặc bằng 34, hoặc bằng INTRA_LTLCCLM, INTRA_T_CCLM, hoặc INTRA_L_CCLM, điều sau đây áp dụng:

$$d[x][y] = (y < 4) \ ? \ v[x + (y << \log2LfnstSize)] \ : \ ((x < 4) \ ? \ v[32 + x + ((y - 4) << 2)] \ : \ d[x][y]) \quad (8\text{-}963)$$

- Nếu không, điều sau đây áp dụng:

$$d[x][y] = (x < 4) \ ? \ v[y + (x << \log2LfnstSize)] \ : \ ((y < 4) \ ? \ v[32 + y + ((x - 4) << 2)] \ : \ d[x][y]) \quad (8\text{-}964)$$

Tham chiếu đến bảng 8, biến nonZeroSize chỉ ra kích thước hoặc số lượng các biến khác không mà phép toán ma trận được thực hiện để áp dụng LFNST được đặt thành 8 hoặc 16. Khi chiều rộng và chiều cao của khối biến đổi là 4 hoặc 8, nghĩa là, chiều dài của dữ liệu đầu ra của LFNST thuận hoặc dữ liệu đầu vào của LFNST ngược của khối 4×4 và khối 8×8 như được thể hiện trên Fig.11 là 8. Đối với tất cả các khối khác, độ dài của dữ liệu đầu ra của LFNST thuận hoặc dữ liệu đầu vào của LFNST ngược là 16 [nonZeroSize = ((nTbW = = 4 && nTbH = = 4) || (nTbW = = 8 && nTbH = = 8)) ? 8 : 16]. Nghĩa là, khi LFNST thuận được áp dụng, thì số lượng tối đa của dữ liệu đầu ra được giới hạn ở 16.

Dữ liệu đầu vào của LFNST ngược này có thể được sắp xếp hai chiều theo sự quét đường chéo [xC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][0], yC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][1]]. Phần được mô tả với tham chiếu đến bảng 8 thể hiện quá trình giải mã cho (i) của phương pháp đơn giản hóa LFNST.

Phương án 3

- Khi LFNST 8x8 được áp dụng, thì tất cả các khu vực mà LFNST 8x8 không được áp dụng được đặt bằng không → (ii)-(B)

- Sau khi kiểm tra xem dữ liệu khác không có mặt hay không cũng như khu vực tồn tai có được điền với giá trị không hay không (khu vực đặt bằng không được thể hiện trên Fig.11) và khu vực này có được điền với các số không hay không do các sự đặt bằng không bổ sung ((ii)-(B)), chỉ số LFNST chỉ được tín hiệu hóa khi dữ liệu khác không không có mặt → (iii)

[Bảng 9]

Cú pháp tạo mã phần dư

if((lastSubBlock > 0 && log2TbWidth >= 3 && log2TbHeight >= 3) ||	
(lastScanPos > 7 && (log2TbWidth == 2 || log2TbHeight == 3) && log2TbWidth == log2TbHeight)) {	
lfnstZeroOutSigCoeffFlag = 1	
}	

Trong bảng 9, các điều kiện của phần trên của bảng 5 đã được thay đổi. Kích thước của khối biến đổi mà điều kiện thứ nhất trong đó biến cờ lfnstZeroOutSigCoeffFlag cho sự đặt bằng không của LFNST được đặt thành 1 được áp dụng sẽ thay đổi từ 4x4 thành 8x8. Sự thay đổi về kích thước của khối biến đổi chỉ ra rằng việc sự đặt bằng không chỉ được thực hiện trong khối biến đổi mà LFNST 8x8 được áp dụng.

Bảng 6 có thể được áp dụng cho phần tử cú pháp cho đơn vị tạo mã cho phương án 3.

[Bảng 10]

8.7.4 Quá trình biến đổi cho các hệ số biến đổi được tỷ lệ hóa

8.7.4.1 Tổng quát

...

- Các biến predModeIntra, nLfnstOutSize, log2LfnstSize, nLfnstSize,, numLfnstX, numLfnstY, và nonZeroSize được dẫn xuất như sau:

predModeIntra = (cIdx = = 0) ? IntraPredModeY[xTbY][yTbY] : IntraPredModeC[xTbY][yTbY] (8-953)

nLfnstOutSize = (nTbW >= 8 && nTbH >= 8) ? 48 : 16 (8-954)

log2LfnstSize = (nTbW >= 8 && nTbH >= 8) ? 3 : 2 (8-955)

nLfnstSize = 1 << log2LfnstSize (8-956)

numLfnstX = (nTbW > 8 && nTbH = = 4) ? 2 : 1 (8-957)

numLfnstY = (nTbW = = 4 && nTbH > 8) ? 2 : 1 (8-958)

nonZeroSize = ((nTbW = = 4 && nTbH = = 4) ||
(nTbW = = 8 && nTbH = = 8)) ? 8 : 16 (8-959)

- Quá trình ánh xạ chế độ nội dự đoán góc rộng được quy định trong điều 8.4.5.2.6 được dẫn ra với predModeIntra, nTbW, nTbH và cIdx làm đầu vào, và predModeIntra được sửa đổi làm đầu ra.

- Cho xSbIdx = 0..numLfnstX - 1 và ySbIdx = 0..numLfnstY - 1, điều sau đây áp dụng:

- Giá trị của danh sách u[x] với x = 0..nonZeroSize - 1 được dẫn xuất như sau:

xC = (xSbIdx << log2LfnstSize) +
DiagScanOrder[log2LfnstSize][log2LfnstSize][x][0] (8-960)

yC = (ySbIdx << log2LfnstSize) +
DiagScanOrder[log2LfnstSize][log2LfnstSize][x][1] (8-961)

u[x] = d[xC][yC] (8-962)

...

Các biến nonZeroW và nonZeroH được dẫn xuất như sau:

nonZeroW = Min((lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8 :
nTbW, (trTypeHor > 0) ? 16 : 32) (8-967)

nonZeroH = Min((lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8 :
nTbH, (trTypeVer > 0) ? 16 : 32) (8-968)

Theo phương án này, vì sự đặt bằng không của LFNST được thực hiện trong khối biến đổi có chiều rộng và chiều cao là 8 hoặc lớn hơn trong đó LFNST 8x8 được thực hiện. Như được thể hiện trong bảng 10, các biến nonZeroW và nonZeroH chỉ ra chiều rộng và chiều cao của khối gồm các hệ số biến đổi khác 0 được đưa vào

cho biến đổi sơ cấp ngược có thể được suy ra là 8 khi chỉ số lfnst lớn hơn 0 và chiều rộng và chiều cao của khối biến đổi là 8 hoặc lớn hơn [(lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8, (lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8]. Nghĩa là, trong khối biến đổi, sự đặt bằng không được thực hiện trong khu vực khác với khu vực 8x8 mà lfnst được áp dụng. Phần này thể hiện quá trình giải mã cho (ii) - (B) của phương pháp đơn giản hóa LFNST.

Phương án 4

- Khi LFNST 8x8 được áp dụng, thì tất cả các khu vực mà LFNST 8x8 không được áp dụng được đặt bằng không → (ii)-(B)

- Phương pháp tín hiệu hóa chỉ số LFNST theo phương pháp không kiểm tra xem dữ liệu khác 0 có mặt hay không có thể được áp dụng. Nghĩa là, sự đặt bằng không ở (ii) - (B) được thực hiện bởi thiết bị mã hóa, và thiết bị giải mã có thể thực hiện sự tín hiệu hóa chỉ số LFNST với giả định rằng dữ liệu khác 0 sự đặt bằng không có thể có mặt cho vùng đặt bằng không tương ứng. Ngay cả trong trường hợp này, chỉ chỉ đặt bằng không được trình bày trên Fig.11 được kiểm tra và chỉ số LFNST được tín hiệu hóa dựa trên điều này.

Phương án 5

- Giới hạn số lượng dữ liệu đầu ra đối với LFNST thuận ở 16 → (i)

- Khi LFNST 8x8 được áp dụng, thì tất cả các khu vực mà LFNST 8x8 không được áp dụng được đặt bằng không → (ii)-(B)

- Sau khi kiểm tra xem dữ liệu khác 0 có mặt hay không cũng là vùng có mặt (vùng 0 đặt bằng không còn lại trên Fig.11 không gồm Fig.11(c), khu vực mà Fig.12 được áp dụng thay vì Fig.11(c), nhưng sự đặt bằng không không được áp dụng) được điền với giá trị không và khu vực này được điền với các số không do các sự đặt bằng không bổ sung ((ii)-(B)), chỉ số LFNST chỉ được tín hiệu hóa khi giá trị các không không có mặt → (iii)

Trong trường hợp của phương án này, bảng 9 của phương án 3 có thể được áp dụng cho phần tủ cú pháp tạo mã phần dư, và bảng 6 của phương án 1 có thể được áp dụng cho cú pháp cú pháp cho đơn vị tạo tạo mã.

Theo bảng 9, kích thước của khối biến đổi mà điều kiện thứ nhất trong đó biến cờ lfnstZeroOutSigCoeffFlag cho sự đặt bằng không trong LFNST được đặt thành 1 được áp dụng là 8x8. Giới hạn kích thước của khối biến đổi như vậy chỉ ra rằng sự đặt bằng không chỉ được thực hiện trong khối biến đổi mà LFNST 8x8 được áp dụng.

[Bảng 11]

8.7.4 Quá trình biến đổi cho các hệ số biến đổi được tỷ lệ hóa
8.7.4.1 Tổng quát
...

– Các biến predModeIntra, nLfnstOutSize, log2LfnstSize, nLfnstSize, numLfnstX, numLfnstY, và nonZeroSize được suy ra như sau:

```
predModeIntra = ( cIdx = = 0 ) ? IntraPredModeY[ xTbY ][ yTbY ] : IntraPredModeC[ xTbY ][ yTbY ]
nLfnstOutSize = ( nTbW >= 8 && nTbH >= 8 ) ? 48 : 16
log2LfnstSize = ( nTbW >= 8 && nTbH >= 8 ) ? 3 : 2
nLfnstSize = 1 << log2LfnstSize
nonZeroSize = ( ( nTbW = = 4 && nTbH = = 4 ) || ( nTbW = = 8 && nTbH = = 8 ) ) ? 8 : 16 (8959)
```

– Quá trình ánh xạ chế độ nội dự đoán góc rộng như được quy định trong điều 8.4.5.2.6 được dẫn ra với predModeIntra, nTbW, nTbH và cIdx làm các đầu vào, và predModeIntra được sửa đổi làm đầu ra.

– Các giá trị của danh sách u[x] với x = 0..nonZeroSize − 1 được suy ra như sau:

```
xC = DiagScanOrder[ log2LfnstSize ][ log2LfnstSize ][ x ][ 0 ]
yC = DiagScanOrder[ log2LfnstSize ][ log2LfnstSize ][ x ][ 1 ]
u[ x ] = d[ xC ][ yC ]
```

...

– Mảng d[x][y] với x = 0..nLfnstSize − 1, y = 0..nLfnstSize − 1 được suy ra như sau:

– Nếu predModeIntra nhỏ hơn hoặc bằng 34 hoặc bằng INTRA_LT_CCLM, INTRA_T_CCLM, hoặc INTRA_L_CCLM, điều sau sẽ áp dụng:

```
d[ x ][ y ] = ( y < 4 ) ? v[ x + ( y << log2LfnstSize ) ] :
              ( ( x < 4 ) ? v[ 32 + x + ( ( y − 4 ) << 2 ) ] : d[ x ][ y ] )
```

– Nếu không, phần sau đây áp dụng:

```
d[ x ][ y ] = ( x < 4 ) ? v[ y + ( x << log2LfnstSize ) ] :
              ( ( y < 4 ) ? v[ 32 + y + ( ( x − 4 ) << 2 ) ] : d[ x ][ y ] )
```

...

Các biến nonZeroW và nonZeroH được suy ra như sau:

```
nonZeroW = Min( ( lfnst_idx[ xTbY ][ yTbY ] > 0 && nTbW >= 8 && nTbH >= 8 ) ? 8 : nTbW, ( trTypeHor > 0 ) ? 16 : 32 )
nonZeroH = Min( ( lfnst_idx[ xTbY ][ yTbY ] > 0 && nTbW >= 8 && nTbH >= 8 ) ? 8 : nTbH, ( trTypeVer > 0 ) ? 16 : 32 ) (8968)
```

Tham khảo Bảng 11, biến nonZeroSize chỉ ra kích thước hoặc số lượng các biến khác 0 mà phép toán ma trận được thực hiện để áp dụng LFNST được đặt thành 8 hoặc 16. Khi chiều rộng và chiều cao của khối biến đổi là 4 hoặc 8, nghĩa là, chiều dài của dữ liệu đầu ra của LFNST thuận hoặc dữ liệu đầu vào của LFNST ngược của khối 4×4 và khối 8×8 như được thể hiện trên Fig.11 là 8. Đối với tất cả các khối khác, độ dài của dữ liệu đầu ra của LFNST thuận hoặc dữ liệu đầu vào của LFNST ngược là 16 [nonZeroSize = ((nTbW = = 4 && nTbH = = 4) || (nTbW = = 8 && nTbH = = 8)) ? 8 : 16]. Nghĩa là, khi LFNST thuận được áp dụng, thì số lượng tối đa của dữ liệu đầu ra được giới hạn ở 16.

Dữ liệu đầu vào của LFNST ngược này có thể được sắp xếp hai chiều theo sự quét đường chéo [xC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][0], yC = DiagScanOrder[log2LfnstSize][log2LfnstSize][x][1]]. Phần được mô tả ở trên thể hiện quá trình giải mã cho (i) của phương pháp đơn giản hóa LFNST.

Ngoài ra, theo phương án này, vì sự đặt bằng không của LFNST được thực hiện trong khối biến đổi có chiều rộng và chiều cao là 8 hoặc lớn hơn trong đó LFNST 8x8 được thực hiện. Như được thể hiện trong bảng 11, các biến nonZeroW và nonZeroH chỉ ra chiều rộng và chiều cao của khối gồm các hệ số biến đổi khác 0 được đưa vào cho biến đổi sơ cấp ngược có thể được suy ra là 8 khi chỉ số lfnst lớn hơn 0 và chiều rộng và chiều cao của khối biến đổi là 8 hoặc lớn hơn [(lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8, (lfnst_idx[xTbY][yTbY] > 0 && nTbW >= 8 && nTbH >= 8) ? 8]. Nghĩa là, trong khối biến đổi, sự đặt bằng không được thực hiện trong khu vực khác với khu vực 8x8 mà lfnst được áp dụng. Phần này thể hiện quá trình giải mã cho (ii) - (B) của phương pháp đơn giản hóa LFNST.

Phương án 6

- Giới hạn số lượng dữ liệu đầu ra đối với LFNST thuận ở 16 16 → (i)

- Khi LFNST 8x8 được áp dụng, thì tất cả các khu vực mà LFNST 8x8 không được áp dụng được đặt bằng không → (ii)-(B)

- Phương pháp tín hiệu hóa chỉ số LFNST theo phương pháp không kiểm tra xem dữ liệu khác 0 có mặt hay không có thể được áp dụng. Nghĩa là, sự đặt bằng không ở (ii)-(B) được thực hiện bởi thiết bị mã hóa, và thiết bị giải mã có thể thực hiện sự tín hiệu hóa chỉ số LFNST với giả định rằng dữ liệu khác 0 sự đặt bằng không có thể có mặt cho vùng đặt bằng không tương ứng. Ngay cả trong trường hợp này, khu vực đặt bằng không còn lại trên Fig.11 không gồm Fig.11 (c) chỉ được kiểm tra và chỉ số LFNST được tín hiệu hóa dựa trên điều này.

Trong trường hợp của các phương án 4 và 6, câu lệnh đặc tả cho chỉ số LFNST không bị thay đổi vì sự đặt bằng không chỉ được thực hiện trong thiết bị mã hóa, và thiết bị giải mã phân tích chỉ số lfnst mà không xem xét sự đặt bằng không.

Trong trường hợp của phương pháp đặt bằng không được đề xuất trong (ii), vì số lượng dữ liệu cuối cùng được tạo ra khi cả biến đổi sơ cấp và LFNST được áp dụng, nên lượng tính toán cần thiết để thực hiện toàn bộ quá trình biến đổi có thể được giảm xuống. Nghĩa là, khi LFNST được áp dụng, vì sự đặt bằng không được áp dụng cho dữ liệu đầu ra biến đổi sơ cấp thuận có mặt trong vùng mà LFNST không được áp dụng, nên không cần tạo dữ liệu cho vùng mà trở thành được đặt bằng không trong quá trình thực hiện biến đổi sơ cấp thuận. Theo đó, có thể giảm lượng tính toán cần thiết để tạo ra dữ liệu tương ứng. Các hiệu quả bổ sung của phương pháp đặt bằng không được đề xuất trong (ii) được tóm tắt như sau.

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

Đặc biệt, khi áp dụng (ii)-(B), thì lượng tính toán cho trường hợp xấu nhất được giảm xuống, sao cho quá trình biến đổi có thể được giảm xuống. Nói cách khác, nói chung, lượng tính toán lớn là cần thiết để thực hiện biến đổi sơ cấp kích thước lớn. Bằng cách áp dụng (ii)-(B), số lượng dữ liệu được suy ra khi thực hiện

LFNST thuận có thể giảm xuống 16 hoặc ít hơn. Ngoài ra, khi kích thước của toàn bộ khối (TU hoặc CU) tăng lên, thì hiệu quả của việc giảm số lượng phép toán biến đổi tăng hơn nữa.

Thứ hai, lượng tính toán cần thiết cho toàn bộ quá trình biến đổi có thể được giảm xuống, bằng cách đó giảm tiêu thụ điện năng cần thiết để thực hiện biến đổi.

Thứ ba, độ trễ liên quan đến quá trình biến đổi được giảm xuống.

Biến đổi thứ cấp chẳng hạn như LFNST bổ sung lượng tính toán vào biến đổi sơ cấp hiện có, do đó làm tăng thời gian trễ tổng thể liên quan đến việc thực hiện biến đổi. Đặc biệt, trong trường hợp của nội dự đoán, vì dữ liệu được dựng lại của các khối lân cận được sử dụng trong quá trình dự đoán, nên trong quá trình mã hóa, sự gia tăng độ trễ do biến đổi thứ cấp dẫn đến sự gia tăng độ trễ cho đến khi tái tạo. Điều này có thể dẫn đến sự gia tăng độ trễ tổng thể của mã hóa nội dự đoán.

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

Các hình vẽ sau được đề xuất để mô tả các ví dụ cụ thể của sáng chế. Vì các ký hiệu cụ thể của các thiết bị hoặc các ký hiệu của các tín hiệu/các thông điệp/các trường cụ thể được minh họa trên các hình vẽ được đề xuất để minh họa, nên các dấu hiệu kỹ thuật của sáng chế không bị giới hạn ở các ký hiệu cụ thể được sử dụng trong các hình vẽ sau.

Fig.16 là sơ đồ minh họa hoạt động của thiết bị giải mã viđeo theo một phương án của sáng chế.

Mỗi hoạt động được minh họa trên Fig.7 có thể được thực hiện bởi thiết bị giải mã 300 được minh họa trên Fig.3. Cụ thể hơn, S1610 và S1640 có thể được thực hiện bởi bộ giải mã entrôpi 310 được minh họa trên Fig.3, S1620 có thể được thực hiện bởi bộ giải lượng tử hóa 321 được minh họa trên Fig.3, S1630, S1650 và

S1660 có thể được thực hiện bởi bộ biến đổi ngược 322 được minh họa trên Fig.3 và S1670 có thể được thực hiện bởi bộ cộng 340 được minh họa trên Fig.3. Các hoạt động theo S1610 đến S1670 dựa trên một số chi tiết đã nói ở trên được giải thích với tham chiếu đến Fig.4 đến Fig.15. Do đó, mô tả của các chi tiết cụ thể trùng lặp với các chi tiết đã giải thích ở trên với tham chiếu đến Fig.3 đến Fig.15 sẽ được bỏ qua hoặc sẽ được thực hiện ngắn gọn.

Thiết bị giải mã 300 theo một phương án nhận dòng bit gồm thông tin phần dư, và có thể suy ra thông tin phần dư về khối hiện tại, nghĩa là, khối biến đổi cần được biến đổi, ví dụ, các hệ số biến đổi được lượng tử hóa, từ dòng bit (S1610).

Cụ thể hơn, thiết bị giải mã 300 có thể giải mã thông tin về các hệ số biến đổi được lượng tử hóa cho khối đích từ dòng bit và có thể suy ra các hệ số biến đổi được lượng tử hóa cho khối hiện tại 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 gồm trong tập hợp thông số trình tự (sequence parameter set, SPS) hoặc tiêu đề mảnh và có thể gồm ít nhất một thông tin về việc liệu biến đổi rút gọn (reduced transform, RST) có được áp dụng hay không, thông tin về nhân tử đơn giản hóa, thông tin về kích thước biến đổi tối thiểu mà biến đổi rút gọn được áp dụng, thông tin về kích thước biến đổi tối đa trong đó biến đổi rút gọn được áp dụng, kích thước biến rút gọn và thông tin về chỉ số biến đổi cho biết bất kỳ một trong các ma trận nhân biến đổi có trong một tập hợp biến đổi.

Thiết bị giải mã 300 có thể thực hiện giải lượng tử hóa trên các hệ số biến đổi được lượng tử hóa của khối hiện tại để suy ra các hệ số biến đổi (S1620).

Các hệ số biến đổi được suy ra có thể được sắp xếp hai chiều trong khối hiện tại,thiết bị giải mã có thể suy ra dữ liệu khác 0, tức là, thông tin về các hệ số có nghĩa khác 0 trong khối hiện tại, qua tạo mã phần dư. Nghĩa là, thiết bị giải mã có thể xác định thông tin vị trí cuối cùng của hệ số có nghĩa khác 0 trong khối hiện tại.

Các hệ số biến đổi được suy ra dựa trên thông tin phần dư của S1620 có thể là các hệ số biến đổi được giải lượng tử hóa như được mô tả ở trên, hoặc có thể là các hệ số biến đổi được lượng tử hóa. Nghĩa là, các hệ số biến đổi có thể là dữ liệu có khả năng kiểm tra dữ liệu khác 0 trong khối hiện tại có bất kể lượng tử hóa hay không.

Theo một ví dụ, thiết bị giải mã có thể xác định liệu hệ số có nghĩa có mặt hay không trong vùng thứ hai ngoại trừ vùng thứ nhất ở trên cùng bên trái của khối hiện tại (S1630).

Vùng thứ nhất có thể được được suy ra dựa trên kích thước của khối hiện tại.

Ví dụ, khi kích thước của khối hiện tại là 4x4 hoặc 8x8, thì khu vực thứ nhất có thể từ trên cùng bên trái của khối hiện tại đến vị trí mẫu thứ tám theo hướng quét.

Khi kích thước của khối hiện tại là 4x4 hoặc 8x8, vì 8 dữ liệu được xuất ra qua LFNST thuận, nên 8 hệ số biến đổi được nhận bởi thiết bị giải mã có thể được sắp xếp từ trên cùng bên trái của khối hiện tại đến vị trí mẫu thứ 8 theo hướng quét.

Ngoài ra, khi kích thước của khối hiện tại không phải là 4x4 hoặc 8x8, thì vùng thứ nhất có thể là vùng 4x4 ở trên cùng bên trái của khối hiện tại. Nếu kích thước của khối hiện tại không phải là 4x4 hoặc 8x8, vì 16 dữ liệu được xuất ra thông qua LFNST thuận, nên 16 hệ số biến đổi được nhận bởi thiết bị giải mã có thể được sắp xếp trong khu vực 4x4 trên cùng của khối hiện tại như được thể hiện trong các hình Fig.13 (b) đến (d) và Fig.14(b).

Trong khi đó, các hệ số biến đổi có thể được sắp xếp trong vùng thứ nhất có thể được sắp xếp dọc theo hướng quét đường chéo như được thể hiện trên Fig.8.

Ngoài ra, theo một ví dụ, số lượng các hệ số biến đổi cho khối mà LFNST được áp dụng có thể lên đến 16.

Khi xác định rằng hệ số có nghĩa không có mặt trong vùng thứ hai ngoại trừ vùng thứ nhất, nghĩa là, nếu hệ số có nghĩa không có mặt bằng cách kiểm tra hệ số có nghĩa cho đến vùng thứ hai của khối hiện tại, thì thiết bị giải mã có thể phân tích chỉ số LFNST từ dòng bit (S1640).

Như được mô tả ở trên, khi LFNST thuận được thực hiện bởi thiết bị mã hóa, thì có thể thực hiện sự đặt bằng không trong đó các vùng còn lại của khối hiện tại được coi à 0 có thể được thực hiện ngoại trừ các vùng trong đó các hệ số biến đổi theo LFNST có mặt.

Do đó, khi hệ số có nghĩa có mặt trong vùng thứ hai, vì LFNST không được áp dụng, nên chỉ số LFNST không được tín hiệu hóa và thiết bị giải mã không phân tích chỉ số LFNST.

Thông tin chỉ số LFNST được nhận dưới dạng thông tin cú pháp và thông tin cú pháp được nhận dưới dạng chuỗi bin được nhị phân hóa gồm 0 và 1.

Phần tử cú pháp của chỉ số LFNST theo phương án này có thể chỉ ra liệu một LFNST ngược hoặc một biến đổi ngược không tách được được áp dụng và bất kỳ một trong ma trận nhân biến đổi được gồm trong tập hợp biến đổi, và tập hợp biến đổi gồm hai ma trận nhân biến đổi. Trong trường hợp này, phần tử cú pháp của chỉ số biến đổi có thể có ba giá trị.

Nghĩa là, theo một phương án, giá trị phần tử cú pháp cho chỉ số LFNST có thể gồm 0 chỉ ra trường hợp trong đó LFNST ngược không được áp dụng cho khối đích, 1 chỉ ra ma trận nhân biến đổi thứ nhất trong số các ma trận nhân biến đổi, và 2 chỉ ra ma trận nhân biến đổi thứ hai trong số các ma trận nhân biến đổi.

Khi chỉ số LFNST được phân tích cú pháp, thì thiết bị giải mã có thể áp dụng ma trận LFNST cho các hệ số biến đổi của vùng thứ nhất để suy ra các hệ số biến đổi được sửa đổi (S1650).

Bộ biến đổi ngược 332 của thiết bị giải mã 300 có thể xác định tập hợp biến đổi dựa trên mối quan hệ ánh xạ theo chế độ nội dự đoán được áp dụng cho khối

hiện tại, và có thể thực hiện LFNST ngược, nghĩa là, biến đổi ngược không tách được dựa trên tập hợp biến đổi và các giá trị của các phần tử cú pháp cho chỉ số LFNST.

Như được mô tả ở trên, nhiều tập hợp biến đổi có thể được xác định theo chế độ nội dự đoán của khối biến đổi cần được biến đổi, và LFNST ngược có thể được thực hiện dựa trên bất kỳ một trong các ma trận nhân biến đổi, nghĩa là, các ma trận LFNST được gồm trong tập hợp biến đổi được chỉ ra bởi chỉ số LFNST. Ma trận được áp dụng cho LFNST ngược có thể được đặt tên là ma trận LFNST ngược hoặc ma trận LFNST, và tên của ma trận này không liên quan miễn là nó có quan hệ chuyển vị với ma trận được sử dụng cho LFNST thuận.

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

Trong khi đó, số lượng các hệ số biến đổi được sửa đổi được xác định trước có thể được suy ra dựa trên kích thước của khối hiện tại. Ví dụ, khi chiều cao và chiều rộng của khối hiện tại là 8 trở lên, thì 48 hệ số biến đổi được sửa đổi được suy ra như được thể hiện ở bên trái của Fig.7. Và khi chiều rộng và chiều cao của khối hiện tại không bằng hoặc không lớn hơn 8, nghĩa là khi chiều rộng và chiều cao của khối hiện tại lớn hơn hoặc bằng 4 và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, thì 16 hệ số biến đổi được sửa đổi có thể được suy ra như được thể hiện ở bên phải của Fig.7.

Như được thể hiện trên Fig.7, 48 hệ số biến đổi được sửa đổi có thể được sắp xếp ở các vùng 4x4 trên cùng bên trái, trên cùng bên phải, và dưới cùng bên trái của vùng 8x8 trên cùng bên trái của khối hiện tại, và 16 hệ số biến đổi được sửa đổi có thể được sắp xếp ở vùng 4x4 trên cùng -bên trái của khối hiện tại.

48 hệ số biến đổi được sửa đổi và 16 hệ số biến đổi được sửa đổi có thể được sắp xếp theo hướng dọc hoặc ngang tùy theo chế độ nội dự đoán của khối hiện tại. Ví dụ, khi chế độ nội dự đoán có hướng ngang (chế độ 2 đến 34 trên Fig.9) dựa

trên hướng chéo (chế độ 34 trên Fig.9), thì các hệ số biến đổi được sửa đổi có thể được sắp xếp theo hướng ngang, nghĩa là theo hướng theo hàng thứ nhất, như được thể hiện trong (a) của Fig.7. Khi chế độ nội dự đoán có hướng dọc (chế độ 35 đến 66 trên Fig.9) dựa trên hướng chéo, thì các hệ số biến đổi được sửa đổi có thể được sắp xếp theo hướng dọc, tức là theo hướng thứ nhất của cột, như được thể hiện trong (b) của Fig.7.

Theo một phương án, S1650 có thể gồm giải mã chỉ số biến đổi, xác định liệu nó có tương ứng với các điều kiện để áp dụng RST ngược hay không dựa trên chỉ số biến đổi, tức là chỉ số LFNST, lựa chọn ma trận nhân biến đổi và áp dụng LFNST ngược cho các hệ số biến đổi dựa trên ma trận nhân biến đổi đã chọn và/hoặc nhân tử đơn giản hóa khi các điều kiện để áp dụng LFNST ngược được thỏa mãn. Trong trường hợp này, kích thước của ma trận biến đổi ngược đơn giản hóa có thể được xác định dựa trên nhân tử đơn giản hóa.

Đề cập đến S1650, có thể khẳng định rằng các mẫu phần dư cho khối đích được suy ra dựa trên LFNST ngược của các hệ số biến đổi cho khối đích. Về kích thước của ma trận biến đổi ngược, kích thước của ma trận biến đổi ngược tổng quát là N x N, trong khi kích thước của ma trận LFNST ngược được giảm xuống N x R, do đó có thể giảm sự chiếm dụng bộ nhớ theo tỉ lệ R/N khi thực hiện LFNST ngược so với khi thực hiện biến đổi tổng quát. Hơn nữa, so với số lượng phép toán nhân, N x N, khi sử dụng ma trận biến đổi ngược tổng quát, có thể giảm số lượng phép toán nhân theo tỷ lệ R/N (thành N x R) khi ma trận LFNST ngược được sử dụng. Ngoài ra, vì chỉ cần giải mã các hệ số biến đổi R khi LFNST ngược được áp dụng, nên tổng số lượng các hệ số biến đổi cho khối đích có thể giảm từ N xuống R, so với khi biến đổi ngược tổng quát được áp dụng trong đó N hệ số biến đổi cần được giải mã, do đó tăng hiệu quả giải mã. Nghĩa là, theo S1650, hiệu suất biến đổi (ngược) và hiệu quả giải mã của thiết bị giải mã 300 có thể được tăng lên qua LFNST ngược.

Thiết bị giải mã 300 theo một phương án có thể suy ra các mẫu phần dư cho khối đích dựa trên biến đổi sơ cấp ngược của các hệ số biến đổi được sửa đổi (S1660).

Mặt khác, khi LFNST không được áp dụng, thì chỉ quy trình biến đổi ngược sơ cấp dựa trên MTS có thể được áp dụng trong quy trình biến đổi ngược như sau. Nghĩa là, thiết bị giải mã có thể xác định liệu liệu LFNST có được áp dụng cho khối hiện tại như trong phương án được mô tả ở trên hay không, và khi LFNST không được áp dụng, thì thiết bị giải mã có thể suy ra các mẫu phần dư từ các hệ số biến đổi qua một biến đổi ngược sơ cấp.

Như được thể hiện trên Fig.16, khi các hệ số có nghĩa có mặt trong vùng thứ hai khác với vùng thứ nhất ở trên cùng bên trái của khối hiện tại, thì xác định rằng LFNST không được áp dụng, và thiết bị giải mã suy ra các mẫu phần dư từ các hệ số biến đổi qua biến đổi ngược sơ cấp.

Quy trình biến đổi ngược sơ cấp có thể được tham chiếu là quy trình biến đổi sơ cấp ngược hoặc quy trình biến đổi MTS ngược. Quy trình biến đổi ngược sơ cấp dựa trên MTS như vậy cũng có thể được bỏ qua trong một số trường hợp.

Ngoài ra, biến đổi ngược đơn giản hóa có thể được áp dụng cho biến đổi sơ cấp ngược, hoặc biến đổi tách được thông thường có thể được sử dụng.

Thiết bị giải mã 300 theo một phương án có thể tạo ra các mẫu được dựng lại dựa trên các mẫu phần dư của khối hiện tại và các mẫu dự đoán của khối hiện tại (S1670).

Các hình vẽ sau được đề xuất để mô tả các ví dụ cụ thể của sáng chế. Vì các ký hiệu cụ thể của các thiết bị hoặc các ký hiệu của các tín hiệu/các thông điệp/các trường cụ thể được minh họa trên các hình vẽ được đề xuất để minh họa, nên các dấu hiệu kỹ thuật của sáng chế không bị giới hạn ở các ký hiệu cụ thể được sử dụng trong các hình vẽ sau.

Fig.17 là sơ đồ minh họa hoạt động của thiết bị mã hóa viđeo theo một phương án của sáng chế.

Mỗi hoạt động được minh họa trên Fig.17 có thể được thực hiện bởi thiết bị mã hóa 200 được minh họa trên Fig.2. Cụ thể, S1710 có thể được thực hiện bởi bộ dự đoán được minh họa trên Fig.2, S1720 có thể được thực hiện bởi bộ trừ 231 được minh họa trên Fig.2, S1730 đến S1750 có thể được thực hiện bởi bộ biến đổi 232 được minh họa trên Fig.2, và S1760 có thể được thực hiện bởi bộ lượng tử hóa 233 và bộ mã hóa entrôpi 240 được minh họa trên Fig.2. Các hoạt động theo S1710 đến S1760 dựa trên một số nội dung được mô tả trên Fig.4 đến Fig.15. Do đó, mô tả của các chi tiết cụ thể trùng lặp với các chi tiết đã giải thích ở trên với tham chiếu đến Fig.2 và Fig.4 đến Fig.15 sẽ được bỏ qua hoặc sẽ được thực hiện ngắn gọn.

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

Thiết bị mã hóa 200 theo một phương án có thể suy ra các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán (S1620).

Thiết bị mã hóa 200 theo một phương án có thể suy ra các hệ số biến đổi cho khối đích dựa trên biến đổi sơ cấp cho các mẫu phần dư (S1630).

Biến đổi sơ cấp có thể được thực hiện qua nhiều nhân biến đổi, và trong trường hợp này, nhân biến đổi có thể được lựa chọn dựa trên chế độ nội dự đoán.

Thiết bị mã hóa 200 có thể xác định liệu thực hiện biến đổi thứ cấp hay biến đổi không tách được, cụ thể là, LFNST, dựa trên các hệ số biến đổi cho khối hiện tại.

Khi xác định thực hiện LFNST, thì thiết bị mã hóa 200 có thể suy ra các hệ số biến đổi được sửa đổi cho khối hiện tại dựa trên các hệ số biến đổi của vùng thứ nhất ở trên cùng bên trái của khối hiện tại và ma trận LFNST được xác định trước (S1740).

Thiết bị mã hóa 200 có thể xác định bộ biến đổi dựa trên mối quan hệ ánh xạ theo chế độ nội dự đoán được áp dụng cho khối hiện tại, và có thể thực hiện LFNST, nghĩa là, biến đổi không tách được dựa trên một trong hai ma trận LFNST được gồm trong tập hợp biến đổi.

Như được mô tả ở trên, nhiều tập hợp biến đổi có thể được xác định theo chế độ nội dự đoán của khối biến đổi cần được biến đổi. Ma trận được áp dụng cho LFNST có mối quan hệ chuyển vị với ma trận được sử dụng cho LFNST ngược.

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

Vùng thứ nhất có thể được được suy ra dựa trên kích thước của khối hiện tại. Ví dụ, khi chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, thì vùng thứ nhất là vùng 4x4 trên cùng bên trái, và dưới cùng bên phải của khu vực 8x8 ở trên cùng bên trái của khối hiện tại như được thể hiện ở bên trái của Fig.7. Khi chiều cao và chiều rộng của khối hiện tại không bằng hoặc lớn hơn 8, thì regio thứ nhất có thể là vùng 4x4 ở trên cùng bên trái của khối hiện tại như được thể hiện ở bên phải của Fig.7.

Các hệ số biến đổi của vùng thứ nhất có thể được sắp xếp một chiều theo hướng dọc hoặc ngang tùy theo chế độ nội dự đoán của khối hiện tại cho phép toán nhân với ma trận LFNST.

48 hệ số biến đổi được sửa đổi hoặc 16 hệ số biến đổi được sửa đổi của vùng thứ nhất có thể được sắp xếp theo hướng dọc hoặc ngang tùy theo chế độ nội dự đoán của khối hiện tại. Ví dụ, nếu chế độ nội dự đoán là hướng ngang (chế độ 2 đến 34 trên Fig.9) dựa trên hướng chéo (chế độ 34 trên Fig.9), thì các hệ số biến đổi có thể được sắp xếp theo hướng ngang, tức là theo hướng theo hàng thứ nhất, như được thể hiện trong (a) của Fig.7. Khi chế độ nội dự đoán có hướng dọc (chế độ 35 đến 66 trên Fig.9) dựa trên hướng chéo, thì các hệ số biến đổi có thể được sắp xếp

theo hướng dọc, tức là theo hướng thứ nhất của cột, như được thể hiện trong (b) của Fig.7.

Trong một ví dụ, LFNST có thể được thực hiện dựa trên ma trận biến đổi đơn giản hóa hoặc ma trận nhân biến đổi, và ma trận biến đổi đơn giản hóa có thể là ma trận không vuông trong đó số lượng hàng ít hơn số lượng cột.

Theo một phương án, S1740 có thể gồm các bước xác định liệu các điều kiện áp dụng LFNST có được thỏa mãn hay không, tạo và mã hóa chỉ số LFNST dựa trên sự xác định, lựa chọn ma trận nhân biến đổi và áp dụng LFNST cho các mẫu phần dư dựa trên ma trận nhân biến đổi đã lựa chọn và/hoặc nhân tử đơn giản hóa khi các điều kiện áp dụng LFNST được thỏa mãn. Trong trường hợp này, kích thước của ma trận biến đổi đơn giản hóa có thể được xác định dựa trên nhân tử đơn giản hóa.

Đề cập đến S1740, có thể khẳng định rằng các hệ số biến đổi cho khối đích được suy ra dựa trên LFNST cho các mẫu phần dư. Về kích thước của ma trận nhân biến đổi, kích thước của ma trận nhân biến đổi chung là N x N, trong khi kích thước của ma trận biến đổi đơn giản được giảm xuống R x N, do đó có thể giảm chiếm dụng bộ nhớ theo tỷ lệ R/N khi thực hiện RST so với khi thực hiện biến đổi tổng quát. Hơn nữa, so với số lượng phép toán nhân, N x N, khi sử dụng ma trận nhân biến đổi tổng quát, có thể giảm số lượng phép toán nhân theo tỷ lệ R/N (thành R x N) khi ma trận nhân biến đổi đơn giản được sử dụng. Ngoài ra, vì chỉ hệ số biến đổi R được suy ra khi áp dụng RST, nên tổng số hệ số biến đổi cho khối đích có thể giảm từ N xuống R, so với khi áp dụng biến đổi chung trong đó N hệ số biến đổi được suy ra, do đó làm giảm lượng dữ liệu được truyền bởi thiết bị mã hóa 200 đến thiết bị giải mã 300. Nghĩa là, theo S1740, hiệu suất biến đổi và hiệu quả mã hóa của thiết bị mã hóa 200 có thể được tăng lên qua LFNST.

Trong khi đó, theo một ví dụ, thiết bị mã hóa có thể loại bỏ vùng thứ hai của khối hiện tại mà hệ số biến đổi đã sửa đổi không có mặt (S1750).

Như các Hình 13 và 14, tất cả các vùng còn lại của khối hiện tại mà hệ số biến đổi được sửa đổi không có mặt có thể được coi là các số không. Do sự đặt bằng không, lượng tính toán cần thiết để thực hiện toàn bộ quá trình biến đổi được giảm xuống, và lượng tính toán cần thiết cho toàn bộ quá trình biến đổi được giảm xuống, do đó giảm tiêu thụ điện năng cần thiết để thực hiện biến đổi. Ngoài ra, hiệu quả tạo mã hình ảnh có thể được tăng lên bằng cách giảm độ trễ liên quan đến quá trình biến đổi.

Mặt khác, khi LFNST không được áp dụng, thì chỉ có thể áp dụng quy trình biến đổi sơ cấp dựa trên MTS trong quy trình biến đổi như được mô tả ở trên. Nghĩa là, thiết bị mã hóa có thể xác định liệu LFNST có được áp dụng cho khối hiện tại như trong phương án được mô tả ở trên hay không, và khi LFNST không được áp dụng, thì thiết bị mã hóa có thể suy ra các hệ số biến đổi từ các mẫu phần dư thông qua biến đổi sơ cấp.

Quy trình biến đổi sơ cấp này có thể được tham chiếu là quy trình biến đổi sơ cấp hoặc quy trình biến đổi MTS. Quy trình biến đổi sơ cấp dựa trên MTS như vậy cũng có thể được bỏ qua trong một số trường hợp.

Thiết bị mã hóa 200 theo một phương án có thể suy ra các hệ số biến đổi được lượng tử hóa bằng cách thực hiện lượng tử hóa dựa trên hệ số biến đổi được sửa đổi cho khối đích, và có thể mã hóa thông tin về các hệ số biến đổi được lượng tử hóa và chỉ số LFNST (S1760). Nghĩa là, thiết bị mã hóa có thể tạo ra thông tin phần dư gồm thông tin về các hệ số biến đổi được lượng tử hóa. Thông tin phần dư có thể gồm phần tử thông tin/cú pháp liên quan đến biến đổi được mô tả ở trên. Thiết bị mã hóa có thể mã hóa thông tin hình ảnh/viđeo gồm thông tin phần dư và xuất ra thông tin hình ảnh/viđeo được mã hóa dưới dạng dòng bit.

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

Trong một ví dụ, thông tin về các hệ số biến đổi được lượng tử hóa có thể gồm ít nhất một thông tin về việc liệu LFNST có được áp dụng hay không, thông tin về nhân tử đơn giản hóa, thì thông tin về kích thước biến đổi tối thiểu mà LFNST được áp dụng và thông tin về kích thước biến đổi tối đa mà LFNST được áp dụng.

Ngoài ra, thiết bị mã hóa 200 có thể mã hóa thông tin về kích thước của khối được áp dụng biến đổi cực đại, ví dụ, thông tin cờ chẳng hạn như sps_max_luma_transform_size_64_flag, ở mức tập hợp thông số trình tự.

Theo sáng chế, ít nhất một trong phép lượng tử hóa/giải lượng tử hóa và/hoặc biến đổi/biến đổi ngược có thể được bỏ qua. Khi sự lượng tử hóa/giải lượng tử hóa được bỏ qua, thì hệ số biến đổi được lượng tử hóa có thể được tham chiếu là hệ số biến đổi. Khi biến đổi/biến đổi ngược được bỏ qua, thì hệ số biến đổi có thể được tham chiếu là hệ số hoặc hệ số phần dư, hoặc vẫn có thể được tham chiếu là hệ số biến đổi cho tính nhất quán của biểu thức.

Ngoài ra, theo sáng chế, hệ số biến đổi được lượng tử hóa và hệ số biến đổi có thể được tham chiếu là hệ số biến đổi và hệ số biến đổi được tỷ lệ hóa, tương ứng. Trong trường hợp này, thông tin phần dư có thể gồm thông tin về (các) hệ số biến đổi, và thông tin về (các) hệ số biến đổi có thể được tín hiệu hóa qua cú pháp tạo mã phần dư. Các hệ số biến đổi có thể được suy ra dựa trên thông tin phần dư (hoặc thông tin về (các) hệ số biến đổi), và các hệ số biến đổi được tỷ lệ hóa có thể được suy ra qua biến đổi ngược (tỷ lệ hóa) của các hệ số biến đổi. Các mẫu phần dư có thể được suy ra dựa trên biến đổi ngược (biến đổi) của các hệ số biến đổi được tỷ lệ hóa. Các chi tiết này cũng có thể được áp dụng/được diễn tả trong các phần khác của sáng chế.

Trong các phương án được mô tả ở trên, các phương pháp được giải thích trên cơ sở các lưu đồ bằng một loạt các bước hoặc các khối, nhưng sáng chế không bị giới hạn với thứ tự của các bước, và một bước nhất định có thể được thực hiện theo thứ tự hoặc bước khác với bước được mô tả ở trên, hoặc đồng thời với một

bước khác. Hơn nữa, người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể hiểu rằng các bước được thể hiện trong lưu đồ không phải là độc nhất, và rằng một bước khác có thể được kết hợp hoặc một hoặc nhiều bước của lưu đồ có thể bị loại bỏ mà không ảnh hưởng đến phạm vi của sáng chế.

Các phương pháp đã được mô tả ở trên theo sáng chế có thể được thực hiện ở dạng phần mềm, và thiết bị mã hóa và/hoặc thiết bị giải mã theo sáng chế có thể được gồm trong thiết bị xử lý hình ảnh, ví dụ, TV, máy tính, điện thoại thông minh, hộp đổi tín hiệu (set-top box), thiết bị hiển bị, hoặc loại tương tự.

Khi các phương án trong sáng chế được biểu hiện bằng phần mềm, thì các phương pháp có thể được biểu hiện bằng các mô-đun (các quá trình, các chức năng, hoặc loại tương tự) để thực hiện chức năng đã được mô tả ở trên. Các mô-đun có thể được lưu trữ 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 với bộ xử lý trong nhiều phương tiện đã phổ biến khác nhau. Bộ xử lý có thể gồm mạch tích hợp chuyên dụng (application-specific integrated circuit, ASIC), bộ chip khác, mạch lôgic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ có thể gồm bộ nhớ chỉ đọc (read-only memory, ROM), bộ nhớ truy cập ngẫu nhiên (random access memory, RAM), bộ nhớ flash, thẻ nhớ, phương tiện lưu trữ, và/hoặc thiết bị lưu trữ khác. Nghĩa là, các phương án được mô tả trong sáng chế có thể được thể hiện và được thực hiện trên bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc chip. Ví dụ, các đơn vị chức năng được thể hiện trong mỗi hình vẽ có thể được thể hiện và được thực hiện trên máy tính, bộ vi xử lý, bộ vi xử lý, bộ điều khiển hoặc chip.

Hơn nữa, thiết bị giải mã và thiết bị mã hóa mà các phương án của sáng chế được áp dụng, có thể được gồm trong bộ truyền nhận truyền phát đa phương tiện, thiết bị đầu cuối truyền thông di động, thiết bị viđeo chiếu phim gia đình, thiết bị viđeo chiếu phim kỹ thuật số, camera giám sát, thiết bị trò chuyện viđeo, thiết bị truyền thông thời gian thực chẳng hạn như thiết bị truyền thông viđeo, thiết bị tạo

dòng di động, phương tiện lưu trữ, máy quay phim, thiết bị cung cấp dịch vụ viđeo theo yêu cầu (video on demand, VoD), thiết bị viđeo OTT (over the top, OTT), thiết bị cung cấp dịch vụ tạo dòng Internet, thiết bị viđeo ba chiều (three-dimensional, 3D), thiết bị viđeo điện thoại viđeo, và thiết bị viđeo y tế, và được sử dụng để xử lý tín hiệu viđeo hoặc tín hiệu dữ liệu. Ví dụ, thiết bị viđeo OTT (over the top, OTT) có thể gồm bảng điều khiển trò chơi, bộ đọc Blu-ray, TV truy cập Internet, hệ thống rạp hát gia đình, điện thoại thông minh, máy tính bảng, đầu ghi viđeo kỹ thuật số (digital viđeo recorder, DVR) và loại tương tự.

Ngoài ra, phương pháp xử lý mà sáng chế được áp dụng, có thể được tạo ra ở dạng chương trình được thực thi bởi máy tính, và được lưu trữ trong phương tiện ghi có thể đọc được bằng máy tính. Dữ liệu đa phương tiện có cấu trúc dữ liệu theo sáng chế cũng có thể được lưu trữ trong phương tiện ghi có thể đọc được bằng máy tính. Phương tiện ghi có thể đọc được bằng máy tính gồm tất cả các loại của các thiết bị lưu trữ và các thiết bị lưu trữ phân tán mà trong đó dữ liệu có thể đọc được bằng máy tính có thể được lưu trữ. Phương tiện ghi có thể đọc được bằng máy tính có gồm, ví dụ, đĩa Blu-ray (Blu-ray Disc, BD), bus nối tiếp đa 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. Hơn nữa, phương tiện ghi có thể đọc được bằng máy cũng gồm phương tiện được thực hiện ở dạng sóng mang (ví dụ, truyền qua Internet). Ngoài ra, dòng bit được tạo ra bởi phương pháp mã hóa có thể được lưu trữ trong phương tiện ghi có thể đọc được bằng máy tính hoặc được truyền qua mạng truyền thông có dây hoặc không dây. Ngoài ra, các phương án của sáng chế có thể được biểu hiện dưới sản phẩm chương trình máy tính bằng các mã chương trình, và các mã chương trình có thể được thực thi trên máy tính bởi các phương án của sáng chế. Các mã chương trình có thể được lưu trữ trên bộ mang có thể đọc được bằng máy tính.

Fig.18 minh họa cấu trúc của hệ thống tạo dòng nội dung mà sáng chế được áp dụng.

Hơn nữa, hệ thống tạo dòng nội dung mà sáng chế được áp dụng phần lớn có thể gồm máy chủ mã hóa, máy chủ tạo dòng, máy chủ web, bộ lưu trữ phương tiện, thiết bị người dùng, và thiết bị đầu vào đa phương tiện.

Máy chủ mã hóa thực hiện chức năng để nén thành dữ liệu kỹ thuật số các nội dung được đưa vào từ các thiết bị đầu vào đa phương tiện, chẳng hạn như điện thoại thông minh, camera, và máy quay viđeo (camcoder) và thiết bị tương tự, để tạo ra dòng bit, và truyền dòng bit đến máy chủ tạo dòng. Một ví dụ khác, trong trường hợp các thiết bị đầu vào đa phương tiện, chẳng hạn như, điện thoại thông minh, camera, máy quay viđeo hoặc thiết bị tương tự, trực tiếp tạo ra dòng bit, máy chủ mã hóa có thể được bỏ qua. Dòng bit có thể được tạo bằng phương pháp mã hóa hoặc phương pháp tạo dòng bit mà sáng chế được áp dụng. Và máy chủ tạo dòng có thể lưu trữ dòng bit tạm thời trong quá trình truyền hoặc nhận dòng bit.

Máy chủ tạo dòng truyền dữ liệu đa phương tiện đến thiết bị người dùng trên cơ sở của yêu cầu người dùng qua máy chủ web, mà thực hiện chức năng như công cụ mà thông báo cho người dùng dịch vụ nào có. Khi người dùng yêu cầu dịch vụ mà người dùng muốn, thì máy chủ web chuyển yêu cầu đó đến máy chủ tạo dòng, và máy chủ tạo dòng sẽ truyền dữ liệu đa phương tiện tới người dùng. Về mặt này, hệ thống tạo dòng các nội dung có thể gồm máy chủ điều khiển riêng biệt, và trong trường hợp này, máy chủ điều khiển thực hiện chức năng để điều khiển các lệnh/các phản hồi giữa các thiết bị tương ứng trong hệ thống tạo dòng nội dung.

Máy chủ tạo dòng có thể nhận các nội dung từ bộ nhớ phương tiện và/hoặc máy chủ mã hóa. Ví dụ, trong trường hợp các nội dung được nhận từ máy chủ mã hóa, các nội dung có thể được nhận trong thời gian thực. Trong trường hợp này, máy chủ tạo dòng có thể lưu trữ dòng bit trong khoảng thời gian được xác định trước để cung cấp dịch vụ tạo dòng một cách trôi chảy.

Ví dụ, thiết bị người dùng có thể gồm điện thoại di động, điện thoại thông minh, máy tính xách tay, thiết bị đầu cuối phát rộng kỹ thuật số, trợ lý kỹ thuật số cá nhân (personal digital assistant, PDA), trình phát đa phương tiện di động (portable multimedia player, PMP), bộ điều hướng, máy tính bảng slate (slate PC), máy tính bảng tablet (tablet PC), siêu sách (ultrabook), thiết bị đeo được (ví dụ,thiết bị đầu cuối loại đồng hồ (đồng hồ thông minh), thiết bị đầu cuối loại kính (kính thông minh), màn hình gắn trên đầu (head mounted display, HMD)), TV kỹ thuật số, máy tính để bàn, biển báo kỹ thuật số hoặc loại tương tự. Các máy chủ tương ứng trong hệ thống tạo dòng các nội dung có thể được vận hành như máy chủ phân tán, và trong trường hợp này, dữ liệu được nhận 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 nhiều cách khác nhau. Ví dụ, các dấu hiệu kỹ thuật của các yêu cầu bảo hộ phương pháp của sáng chế có thể được kết hợp để được thực hiện hoặc được thực hiện trong một máy, và các dấu hiệu kỹ thuật của các yêu cầu bảo hộ máy có thể được kết hợp để được thực hiện hoặc được thực hiện trong một phương pháp. Hơn nữa, các dấu hiệu kỹ thuật của các yêu cầu bảo hộ phương pháp và các yêu cầu bảo hộ máy có thể được kết hợp để được thực hiện hoặc được thực hiện trong một máy, và các dấu hiệu kỹ thuật của các yêu cầu bảo hộ phương pháp và các yêu cầu bảo hộ máy có thể được kết hợp để được thực hiện hoặc được thực hiện trong một phương pháp.

YÊU CẦU BẢO HỘ

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

thu nhận thông tin phần dư từ dòng bit;

suy ra các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư;

xác định liệu hệ số có nghĩa có mặt trong vùng thứ hai khác với vùng thứ nhất ở trên cùng bên trái của khối hiện tại hay không;

phân tích chỉ số biến đổi không tách được tần số thấp (low frequency non-separable transform, LFNST) từ dòng bit dựa trên hệ số có nghĩa không có mặt trong vùng thứ hai;

suy ra các hệ số biến đổi được sửa đổi bằng cách áp dụng ma trận LFNST được suy ra dựa trên chỉ số LFNST cho các hệ số biến đổi trong vùng thứ nhất;

suy ra các mẫu phần dư cho khối hiện tại dựa trên biến đổi sơ cấp ngược cho các hệ số biến đổi được sửa đổi; và

tạo ra ảnh được dựng lại dựa trên các mẫu phần dư cho khối hiện tại,

trong đó vùng thứ nhất được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên kích thước của khối hiện tại là 4×4 hoặc 8×8, vùng thứ nhất là từ trên cùng bên trái của khối hiện tại đến vị trí mẫu thứ 8 theo thứ tự quét, và

trong đó dựa trên kích thước của khối hiện tại không phải là 4×4 và 8×8, vùng thứ nhất là khu vực 4×4 ở trên cùng bên trái của khối hiện tại.

2. Phương pháp giải mã hình ảnh theo điểm 1, trong đó thứ tự quét là thứ tự quét đường chéo.

3. Phương pháp giải mã hình ảnh theo điểm 1, trong đó số lượng được xác định trước của các hệ số biến đổi được sửa đổi được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, 48 hệ số biến đổi được sửa đổi được suy ra, và

trong đó dựa trên chiều rộng và chiều cao của khối hiện tại lớn hơn hoặc bằng 4 và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, 16 hệ số biến đổi được sửa đổi được suy ra.

4. Phương pháp giải mã hình ảnh theo điểm 3, trong đó 48 hệ số biến đổi được sửa đổi được sắp xếp ở khu vực 4x4 ở trên cùng bên trái, khu vực 4x4 ở trên cùng bên phải, và khu vực 4x4 ở dưới cùng bên trái trong khu vực 8x8 ở trên cùng bên trái của khối hiện tại.

5. Phương pháp giải mã hình ảnh theo điểm 3, trong đó 16 hệ số biến đổi được sửa đổi được sắp xếp ở khu vực 4x4 ở trên cùng bên trái của khối hiện tại.

6. Phương pháp giải mã hình ảnh theo điểm 4, trong đó 48 hệ số biến đổi được sửa đổi và 16 hệ số biến đổi được sửa đổi được sắp xếp theo hướng dọc hoặc ngang theo chế độ nội dự đoán của khối hiện tại.

7. Phương pháp mã hóa hình ảnh được thực hiện bởi thiết bị mã hóa hình ảnh, bao gồm:

suy ra các mẫu dự đoán cho khối hiện tại;

suy ra các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán;

suy ra các hệ số biến đổi cho khối hiện tại dựa trên biến đổi sơ cấp cho các mẫu phần dư;

suy ra các hệ số biến đổi được sửa đổi cho khối hiện tại dựa trên các hệ số biến đổi của vùng thứ nhất ở trên cùng bên trái của khối hiện tại và ma trận biến đổi

không tách được tần số thấp (low frequency non-separable transform, LFNST) được xác định trước; và

mã hóa thông tin phần dư được suy ra qua lượng tử hóa các hệ số biến đổi được sửa đổi và chỉ số LFNST chỉ ra ma trận LFNST,

trong đó vùng thứ nhất được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, vùng thứ nhất là khu vực 4x4 ở trên cùng bên trái, khu vực 4x4 ở trên cùng bên phải, và khu vực 4×4 ở dưới cùng bên trái trong khu vực 8×8 ở trên cùng bên trái của khối hiện tại, và

trong đó dựa trên chiều rộng và chiều cao của khối hiện tại lớn hơn hoặc bằng 4 và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, vùng thứ nhất là khu vực 4×4 ở trên cùng bên trái của khối hiện tại.

8. Phương pháp mã hóa hình ảnh theo điểm 7, trong đó các hệ số biến đổi trong vùng thứ nhất được sắp xếp một chiều theo hướng dọc hoặc ngang theo chế độ nội dự đoán của khối hiện tại cho phép toán nhân với ma trận LFNST.

9. Phương pháp mã hóa hình ảnh theo điểm 7, trong đó số lượng được xác định trước của các hệ số biến đổi được sửa đổi được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên kích thước của khối hiện tại là 4x4 hoặc 8x8, 8 hệ số biến đổi được sửa đổi được suy ra, và

trong đó dựa trên kích thước của khối hiện tại không phải là 4x4 và 8x8, 16 hệ số biến đổi được sửa đổi được suy ra.

10. Phương pháp mã hóa hình ảnh theo điểm 9, trong đó các hệ số biến đổi được sửa đổi được sắp xếp theo thứ tự quét đường chéo ở trên cùng bên trái của khối hiện tại.

11. Phương pháp mã hóa hình ảnh theo điểm 10, trong đó khi kích thước của khối hiện tại là 4x4 hoặc 8x8, các hệ số biến đổi được sửa đổi được sắp xếp từ trên cùng bên trái của khối hiện tại đến vị trí mẫu thứ 8 theo thứ tự quét đường chéo, và

trong đó khi kích thước của khối hiện tại không phải là 4x4 và 8x8, các hệ số biến đổi được sửa đổi được sắp xếp ở khu vực 4x4 ở trên cùng bên trái của khối hiện tại

12. Phương tiện lưu trữ số đọc được bằng máy tính mà lưu trữ dòng bit được tạo ra bởi phương pháp, phương pháp này bao gồm các bước:

suy ra các mẫu dự đoán cho khối hiện tại;

suy ra các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán;

suy ra các hệ số biến đổi cho khối hiện tại dựa trên biến đổi sơ cấp cho các mẫu phần dư;

suy ra các hệ số biến đổi được sửa đổi cho khối hiện tại dựa trên các hệ số biến đổi của vùng thứ nhất ở trên cùng bên trái của khối hiện tại và ma trận biến đổi không tách được tần số thấp (low frequency non-separable transform, LFNST) được xác định trước; và

mã hóa thông tin phần dư được suy ra qua lượng tử hóa các hệ số biến đổi được sửa đổi và chỉ số LFNST chỉ ra ma trận LFNST để tạo ra dòng bit,

trong đó vùng thứ nhất được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, vùng thứ nhất là khu vực 4x4 ở trên cùng bên trái, khu vực 4x4 ở trên cùng bên phải, và khu vực 4×4 ở dưới cùng bên trái trong khu vực 8×8 ở trên cùng bên trái của khối hiện tại, và

trong đó dựa trên chiều rộng và chiều cao của khối hiện tại lớn hơn hoặc bằng 4 và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, vùng thứ nhất là khu vực 4×4 ở trên cùng bên trái của khối hiện tại.

13. Phương pháp truyền, phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó dòng bit được tạo ra bằng cách suy ra các mẫu dự đoán cho khối hiện tại, suy ra các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán, suy ra các hệ số biến đổi cho khối hiện tại dựa trên biến đổi sơ cấp cho các mẫu phần dư, suy ra các hệ số biến đổi được sửa đổi cho khối hiện tại dựa trên các hệ số biến đổi của vùng thứ nhất ở trên cùng bên trái của khối hiện tại và ma trận biến đổi không tách được tần số thấp (LFNST) được xác định trước, và mã hóa thông tin phần dư được suy ra qua lượng tử hóa các hệ số biến đổi được sửa đổi và chỉ số LFNST chỉ ra ma trận LFNST để tạo ra dòng bit; và

truyền dòng bit,

trong đó vùng thứ nhất được suy ra dựa trên kích thước của khối hiện tại,

trong đó dựa trên chiều cao và chiều rộng của khối hiện tại lớn hơn hoặc bằng 8, vùng thứ nhất là khu vực 4x4 ở trên cùng bên trái, khu vực 4x4 ở trên cùng bên phải, và khu vực 4×4 ở dưới cùng bên trái trong khu vực 8×8 ở trên cùng bên trái của khối hiện tại, và

trong đó dựa trên chiều rộng và chiều cao của khối hiện tại lớn hơn hoặc bằng 4 và chiều rộng hoặc chiều cao của khối hiện tại nhỏ hơn 8, vùng thứ nhất là khu vực 4×4 ở trên cùng bên trái của khối hiện tại.

Fig.1

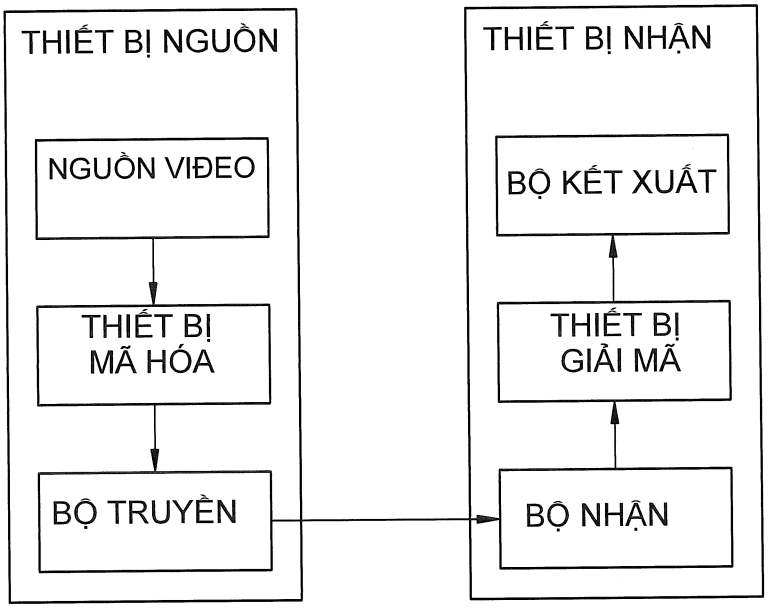

THIẾT BỊ NGUỒN
NGUỒN VIĐEO
THIẾT BỊ MÃ HÓA
BỘ TRUYỀN
THIẾT BỊ NHẬN
BỘ KẾT XUẤT
THIẾT BỊ GIẢI MÃ
BỘ NHẬN

Fig.2

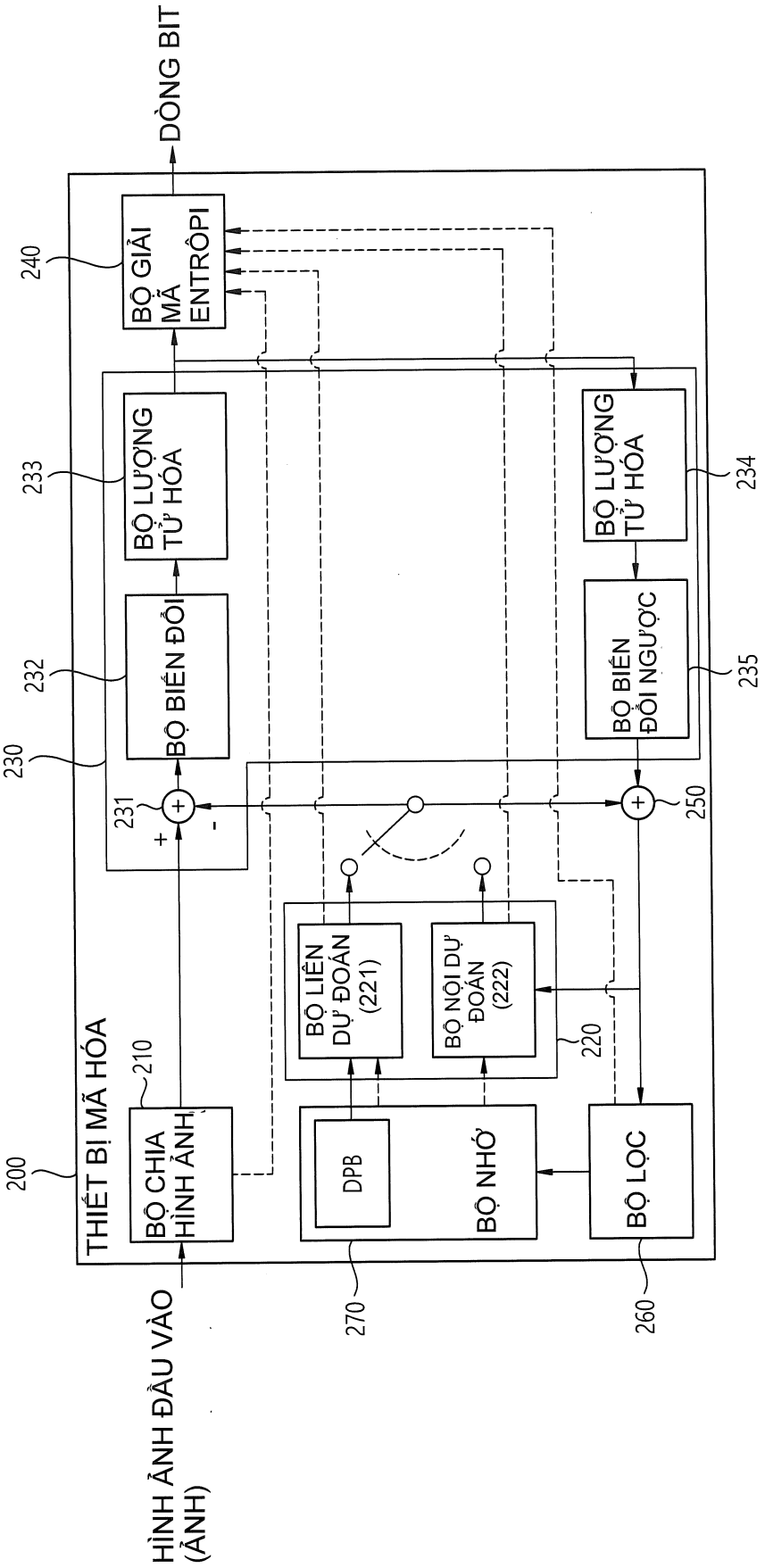

HÌNH ẢNH ĐẦU VÀO
(ẢNH)
THIẾT BỊ MÃ HÓA
200
BỘ CHIA HÌNH ẢNH
210
BỘ LIÊN DỰ ĐOÁN (221)
BỘ NỘI DỰ ĐOÁN (222)
220
DPB
BỘ NHỚ
270
BỘ LỌC
260
BỘ BIẾN ĐỔI
232
BỘ LƯỢNG TỬ HÓA
233
230
231
+
-
BỘ LƯỢNG TỬ HÓA
234
BỘ BIẾN ĐỔI NGƯỢC
235
250
BỘ GIẢI MÃ ENTRÔPI
240
DÒNG BIT

Fig.3

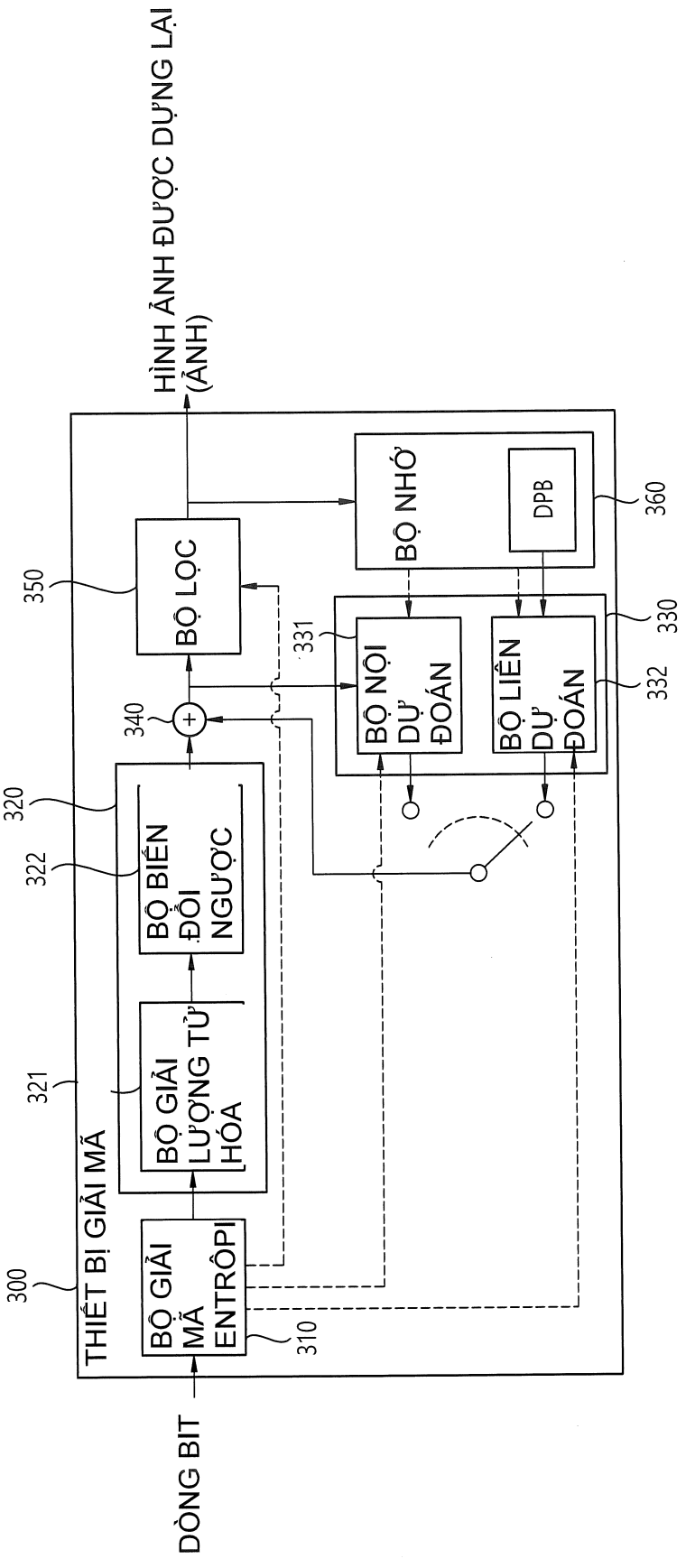

DÒNG BIT
THIẾT BỊ GIẢI MÃ
300
BỘ GIẢI MÃ ENTRÔPI
310
BỘ GIẢI LƯỢNG TỬ HÓA
321
BỘ BIẾN ĐỔI NGƯỢC
322
320
340
BỘ LỌC
350
HÌNH ẢNH ĐƯỢC DỰNG LẠI (ẢNH)
BỘ NHỚ
DPB
360
331
BỘ NỘI DỰ ĐOÁN
BỘ LIÊN DỰ ĐOÁN
332
330

Fig.4

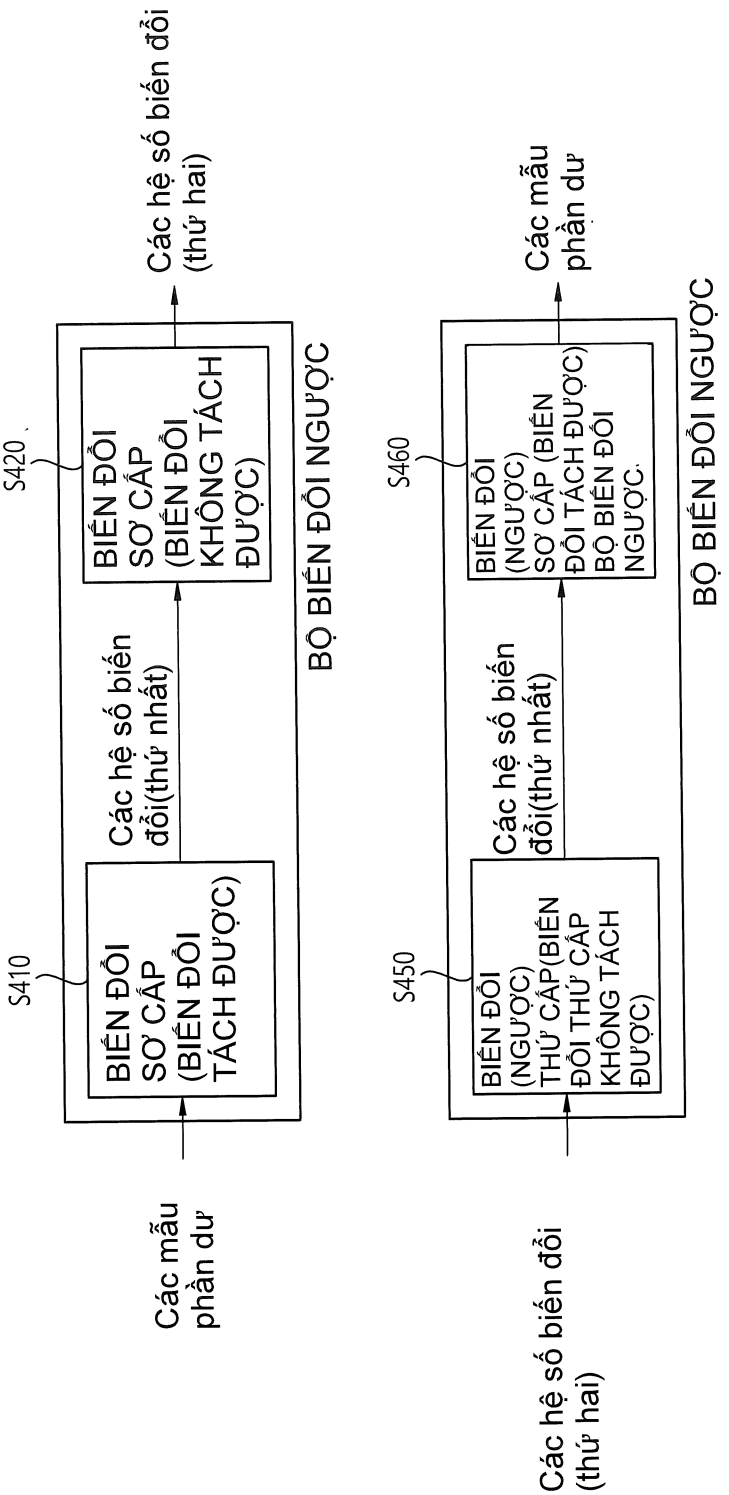

Các mẫu phần dư
S410
BIẾN ĐỔI SƠ CẤP (BIẾN ĐỔI TÁCH ĐƯỢC)
Các hệ số biến đổi(thứ nhất)
S420
BIẾN ĐỔI SƠ CẤP (BIẾN ĐỔI KHÔNG TÁCH ĐƯỢC)
Các hệ số biến đổi (thứ hai)
BỘ BIẾN ĐỔI NGƯỢC
Các hệ số biến đổi (thứ hai)
S450
BIẾN ĐỔI (NGƯỢC) THỨ CẤP(BIẾN ĐỔI THỨ CẤP KHÔNG TÁCH ĐƯỢC)
Các hệ số biến đổi(thứ nhất)
S460
BIẾN ĐỔI (NGƯỢC) SƠ CẤP (BIẾN ĐỔI TÁCH ĐƯỢC) BỘ BIẾN ĐỔI NGƯỢC
Các mẫu phần dư
BỘ BIẾN ĐỔI NGƯỢC

Fig.5

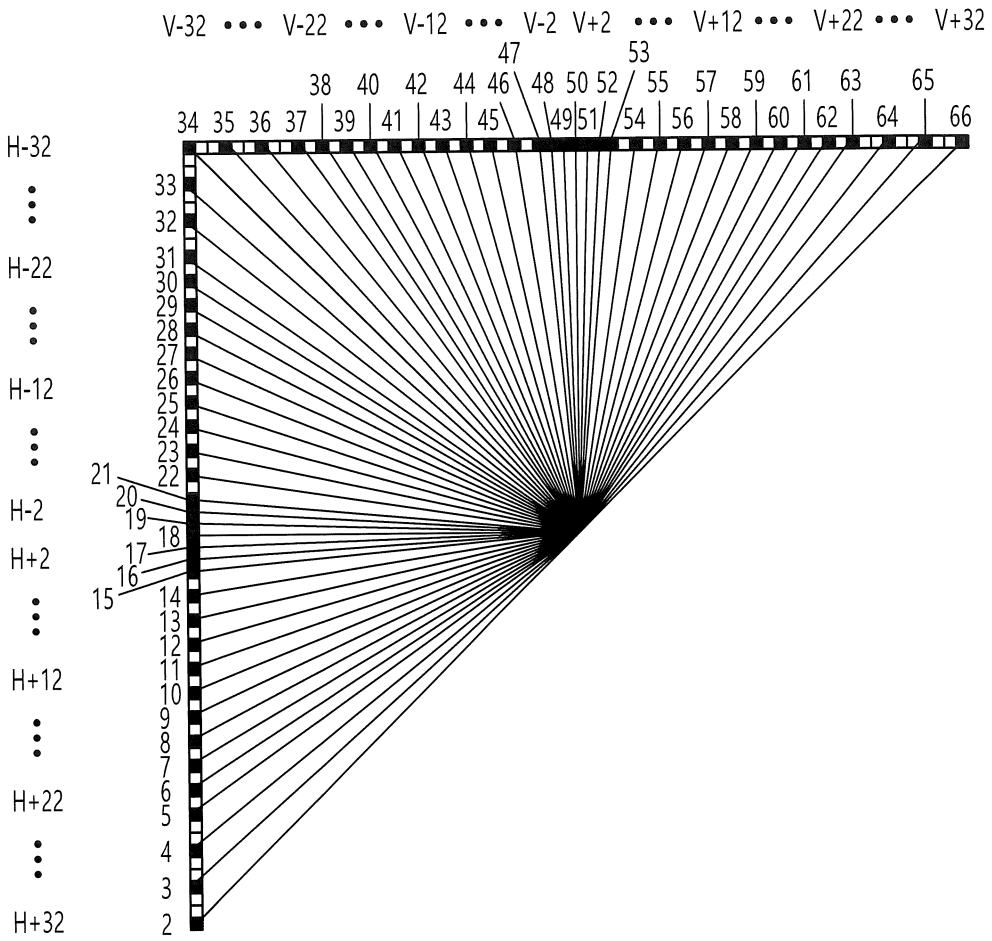
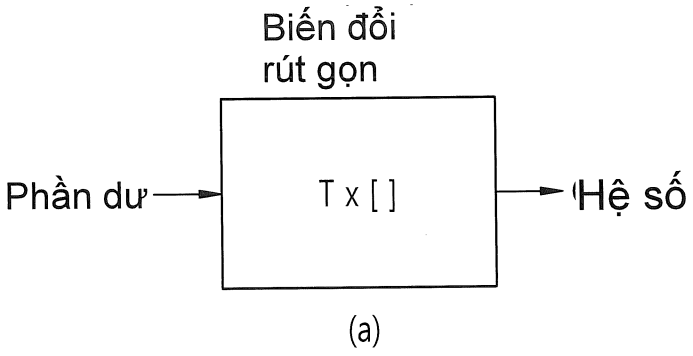
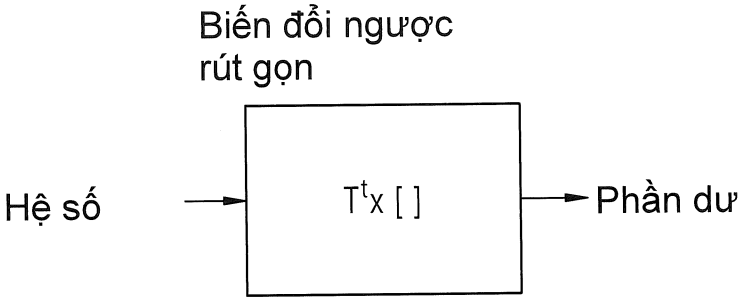

V-32 ••• V-22 ••• V-12 ••• V-2 V+2 ••• V+12 ••• V+22 ••• V+32
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
H-32
H-22
H-12
H-2
H+2
H+12
H+22
H+32
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Fig.6


Biến đổi
rút gọn
Phần dư
T x [ ]
Hệ số

(a)


Biến đổi ngược
rút gọn
Hệ số
T^t x [ ]
Phần dư

(b)

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

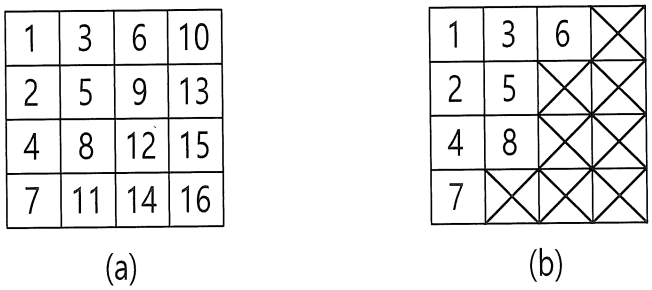


Fig.9

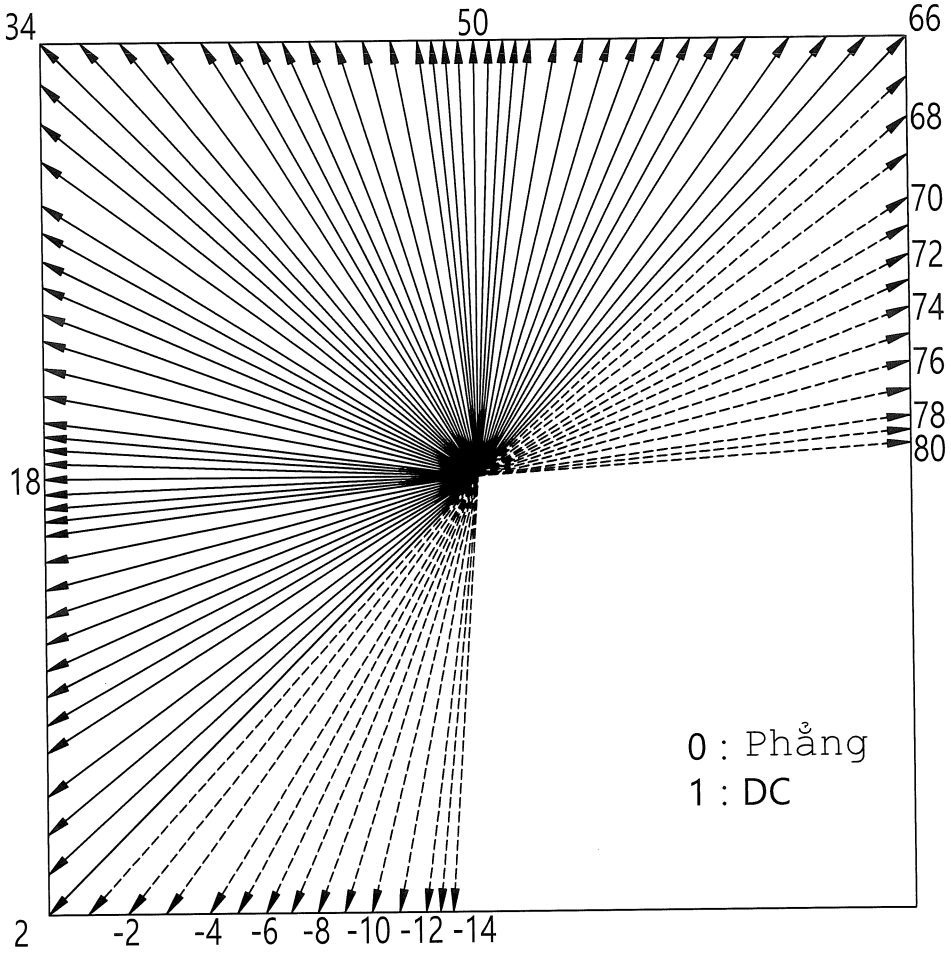
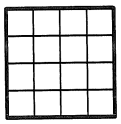
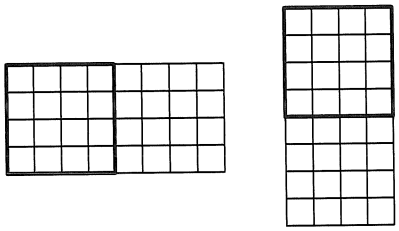


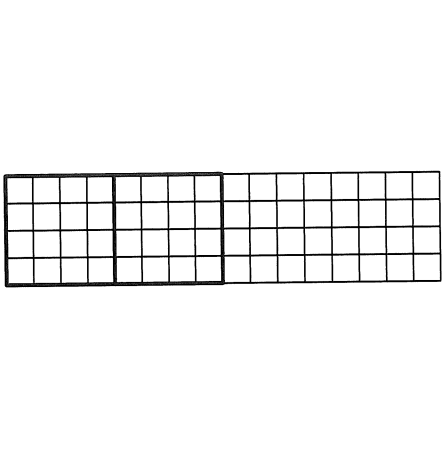
Fig.10

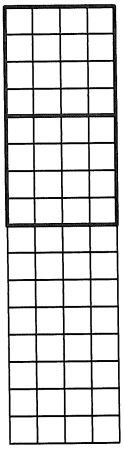


(a) 4x4

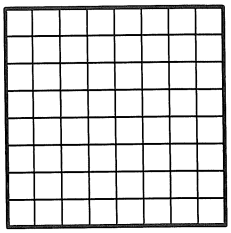


(b) 8x4 / 4x8

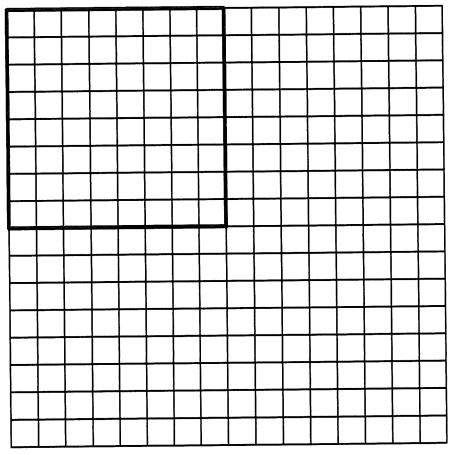




(c) 4xN / Nx4, khi N≥16

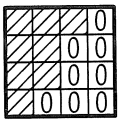


(d) 8x8

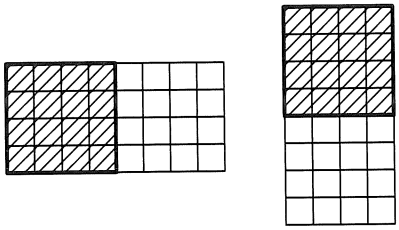


(e) MxN(M≥8, N≥8, M>8 hoặc N>8)

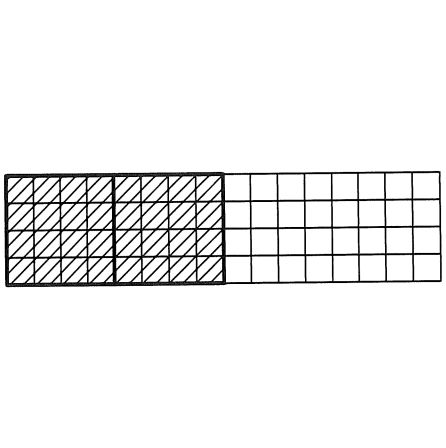
Fig.11

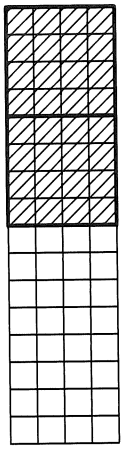


(a) 4x4

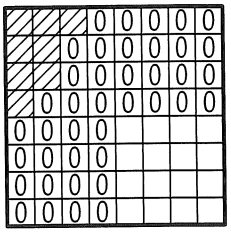


(b) 8x4 / 4x8

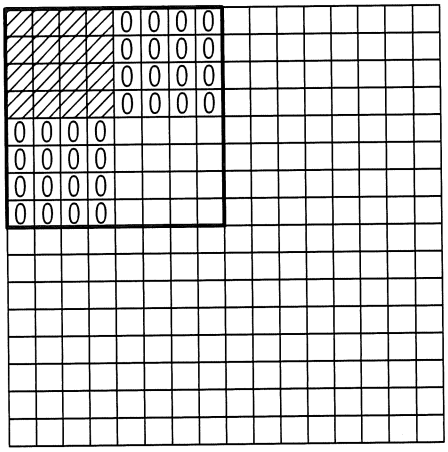




(c) 4xN / Nx4, khi N≥16

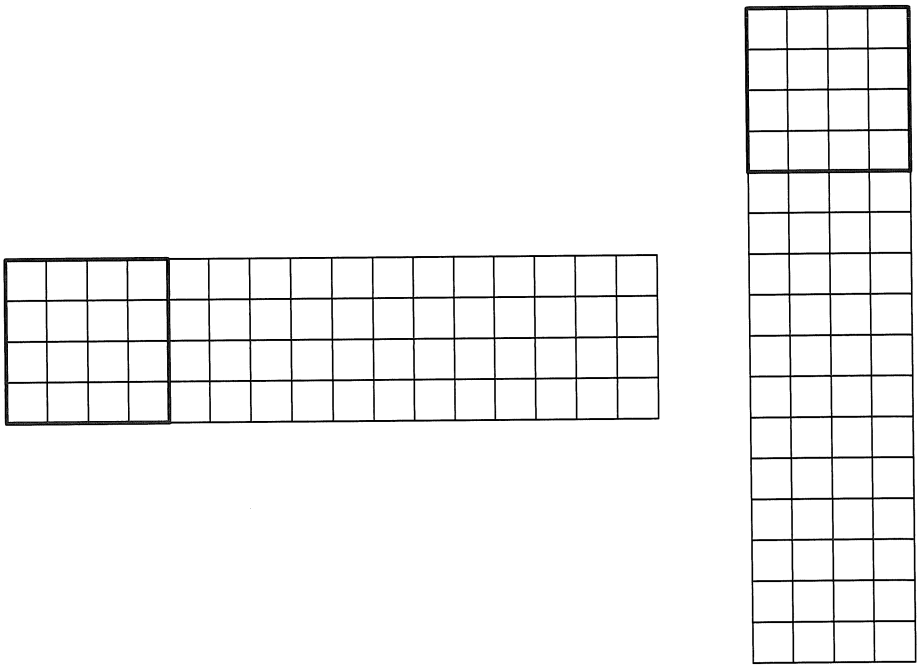


(d) 8x8



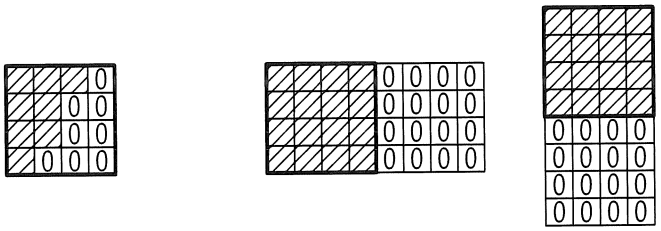
(e) MxN(M≥8, N≥8, M>8 hoặc N>8)

Fig.12

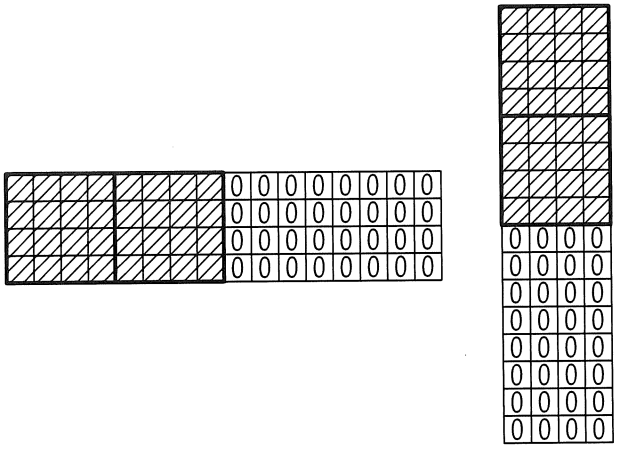


4xN / Nx4, khi N≥16

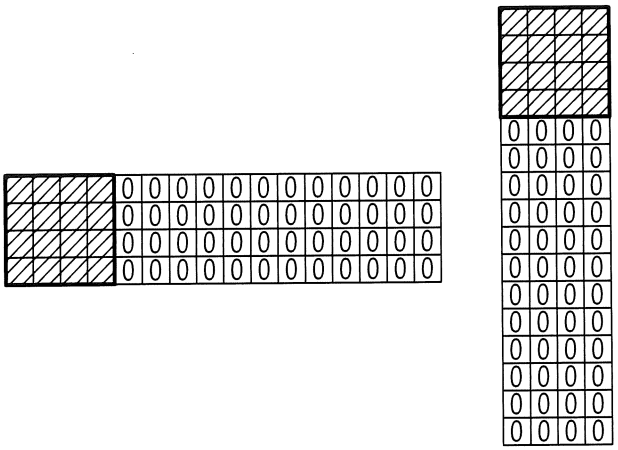
Fig.13



(a) 4x4 (b) 8x4 / 4x8

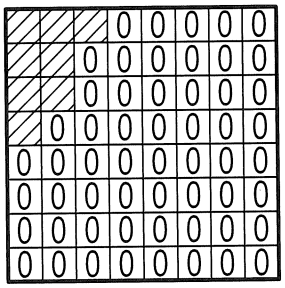


(c) 4xN / Nx4, khi N≥16

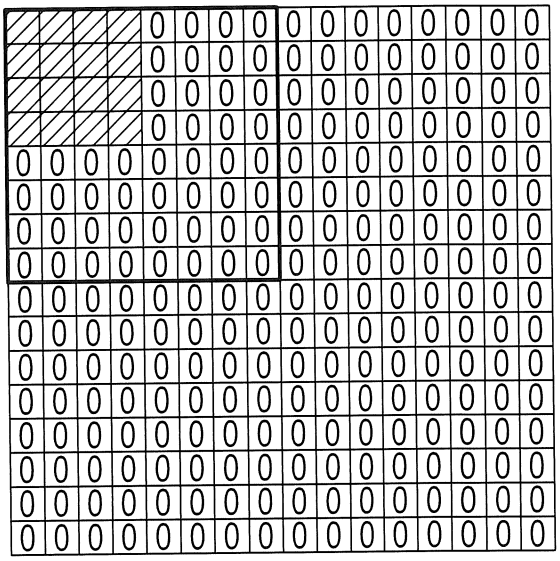


(d) 4xN / Nx4, khi N≥16

Fig.14



(a) 8x8



(b) MxN(M≥8, N≥8, M>8 hoặc N>8)

Fig.15

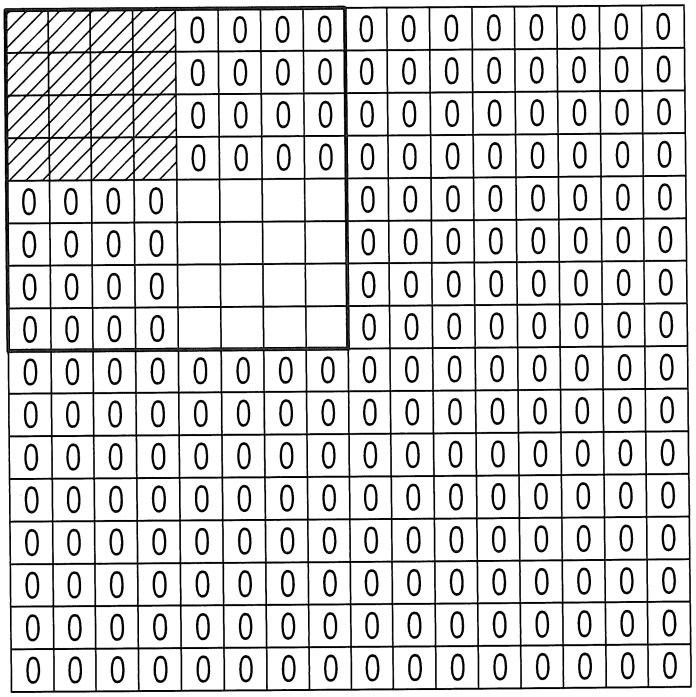


Fig.16

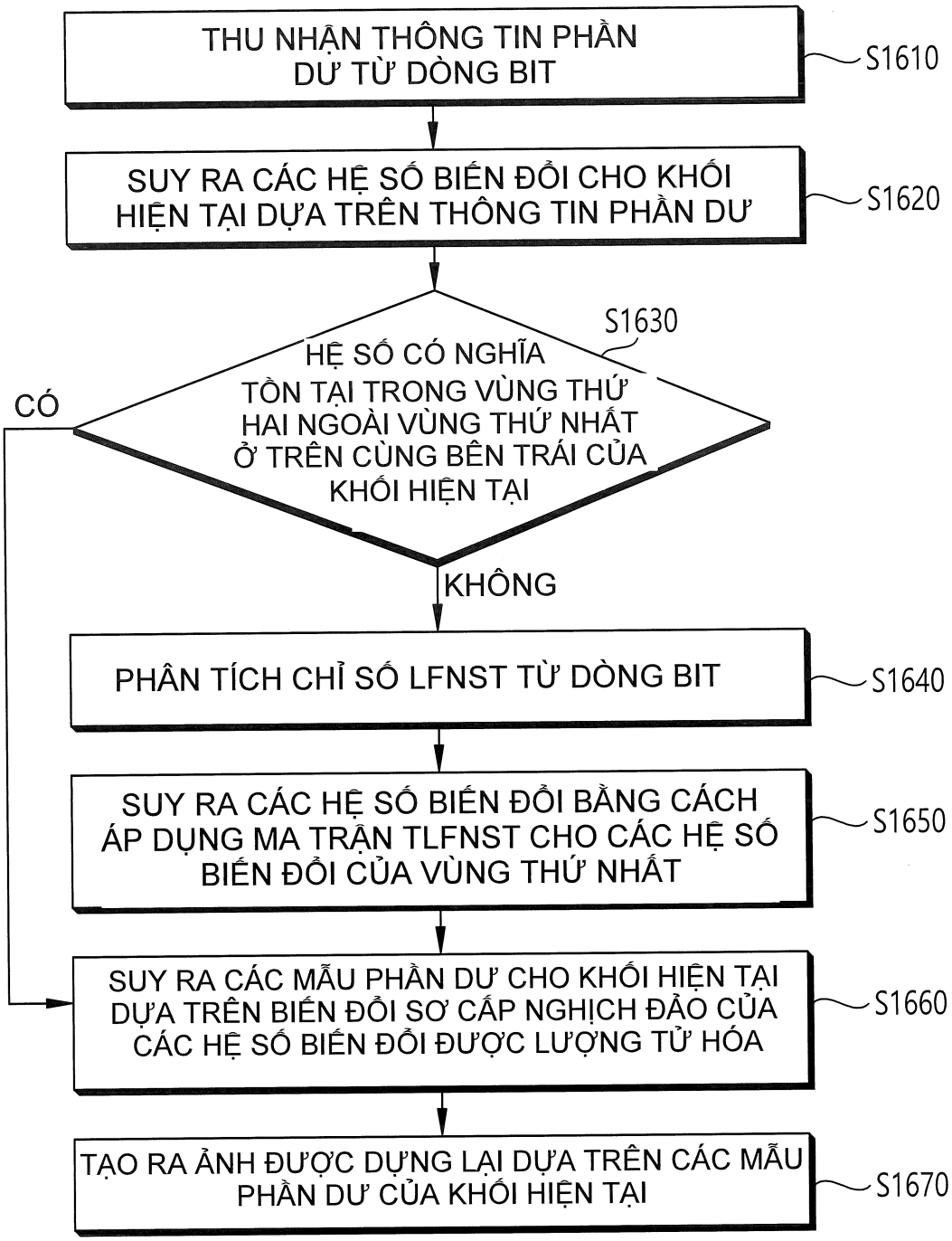

THU NHẬN THÔNG TIN PHẦN DƯ TỪ DÒNG BIT
S1610
SUY RA CÁC HỆ SỐ BIẾN ĐỔI CHO KHỐI HIỆN TẠI DỰA TRÊN THÔNG TIN PHẦN DƯ
S1620
S1630
HỆ SỐ CÓ NGHĨA TỒN TẠI TRONG VÙNG THỨ HAI NGOÀI VÙNG THỨ NHẤT Ở TRÊN CÙNG BÊN TRÁI CỦA KHỐI HIỆN TẠI
CÓ
KHÔNG
PHÂN TÍCH CHỈ SỐ LFNST TỪ DÒNG BIT
S1640
SUY RA CÁC HỆ SỐ BIẾN ĐỔI BẰNG CÁCH ÁP DỤNG MA TRẬN TLFNST CHO CÁC HỆ SỐ BIẾN ĐỔI CỦA VÙNG THỨ NHẤT
S1650
SUY RA CÁC MẪU PHẦN DƯ CHO KHỐI HIỆN TẠI DỰA TRÊN BIẾN ĐỔI SƠ CẤP NGHỊCH ĐẢO CỦA CÁC HỆ SỐ BIẾN ĐỔI ĐƯỢC LƯỢNG TỬ HÓA
S1660
TẠO RA ẢNH ĐƯỢC DỰNG LẠI DỰA TRÊN CÁC MẪU PHẦN DƯ CỦA KHỐI HIỆN TẠI
S1670

Fig.17

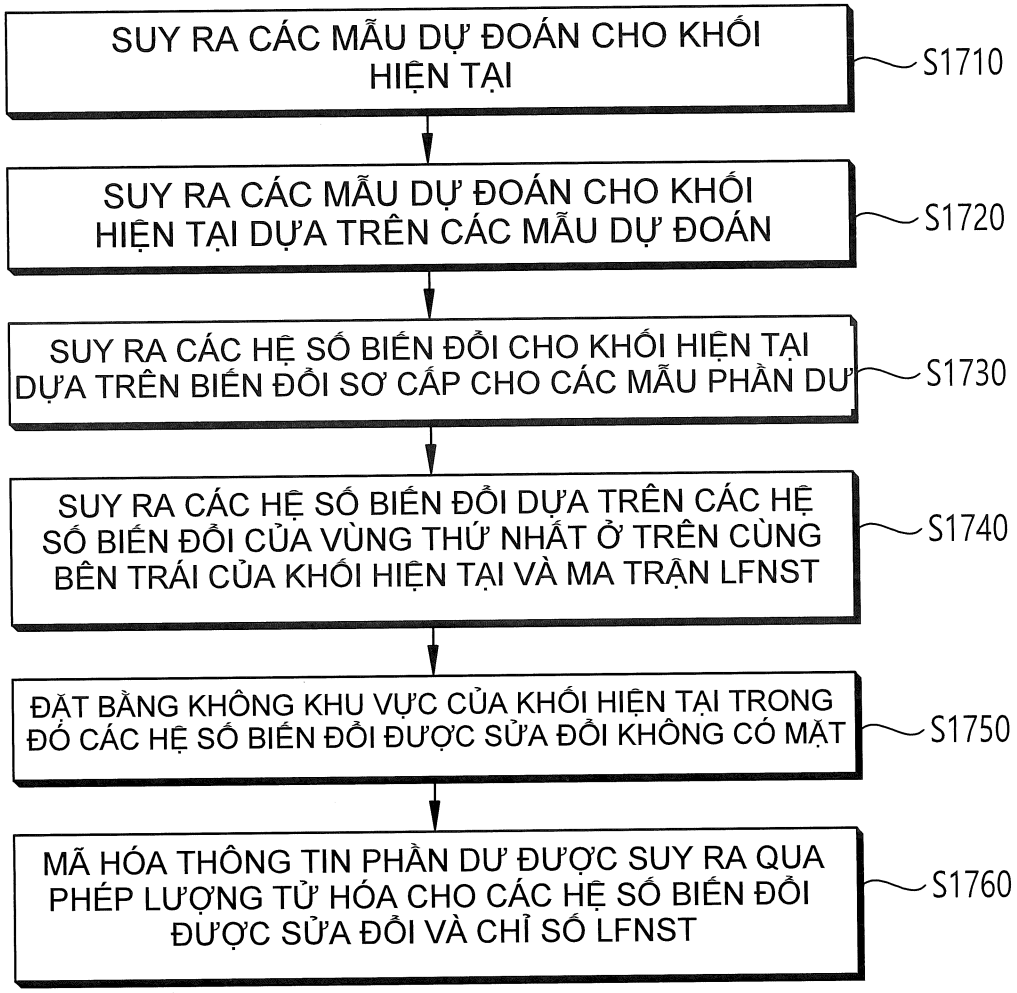
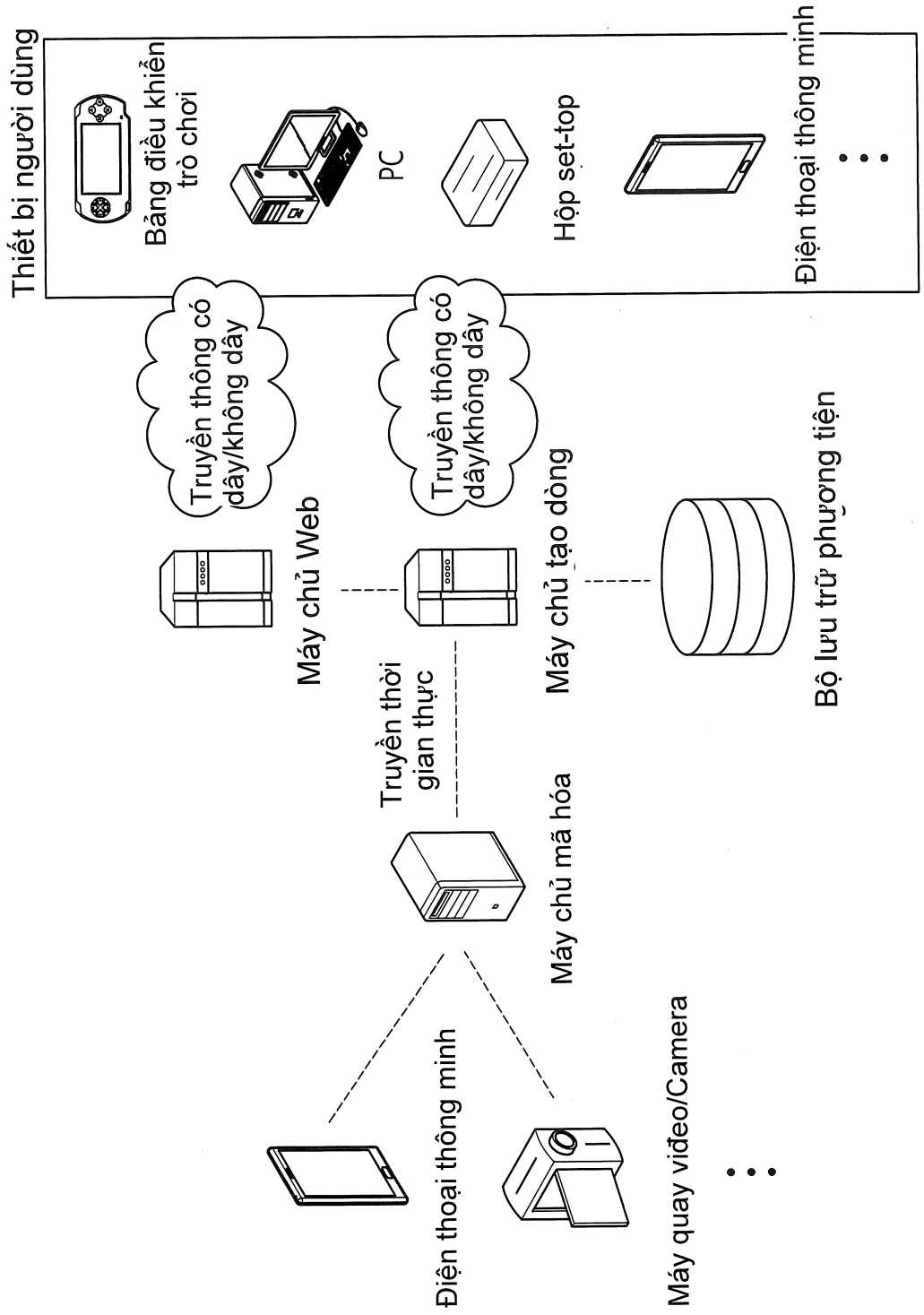

SUY RA CÁC MẪU DỰ ĐOÁN CHO KHỐI HIỆN TẠI
S1710
SUY RA CÁC MẪU DỰ ĐOÁN CHO KHỐI HIỆN TẠI DỰA TRÊN CÁC MẪU DỰ ĐOÁN
S1720
SUY RA CÁC HỆ SỐ BIẾN ĐỔI CHO KHỐI HIỆN TẠI DỰA TRÊN BIẾN ĐỔI SƠ CẤP CHO CÁC MẪU PHẦN DƯ
S1730
SUY RA CÁC HỆ SỐ BIẾN ĐỔI DỰA TRÊN CÁC HỆ SỐ BIẾN ĐỔI CỦA VÙNG THỨ NHẤT Ở TRÊN CÙNG BÊN TRÁI CỦA KHỐI HIỆN TẠI VÀ MA TRẬN LFNST
S1740
ĐẶT BẰNG KHÔNG KHU VỰC CỦA KHỐI HIỆN TẠI TRONG ĐÓ CÁC HỆ SỐ BIẾN ĐỔI ĐƯỢC SỬA ĐỔI KHÔNG CÓ MẶT
S1750
MÃ HÓA THÔNG TIN PHẦN DƯ ĐƯỢC SUY RA QUA PHÉP LƯỢNG TỬ HÓA CHO CÁC HỆ SỐ BIẾN ĐỔI ĐƯỢC SỬA ĐỔI VÀ CHỈ SỐ LFNST
S1760

Fig.18


Thiết bị người dùng
Bảng điều khiển trò chơi
PC
Hộp set-top
Điện thoại thông minh
Truyền thông có dây/không dây
Truyền thông có dây/không dây
Máy chủ Web
Máy chủ tạo dòng
Bộ lưu trữ phượng tiện
Truyền thời gian thực
Máy chủ mã hóa
Điện thoại thông minh
Máy quay video/Camera