



(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)



1-0052145

(51)<sup>2021.01</sup> H04N 19/60; H04N 19/176; H04N 19/18; H04N 19/96; H04N 19/70; H04N 19/132; H04N 19/186 (13) B

(21) 1-2022-04762

(22) 11/01/2021

(86) PCT/KR2021/000339 11/01/2021

(87) WO 2021/141472 15/07/2021

(30) 62/959,814 10/01/2020 US

(45) 27/10/2025 451

(43) 25/10/2022 415A

(73) LG ELECTRONICS INC. (KR)

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

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

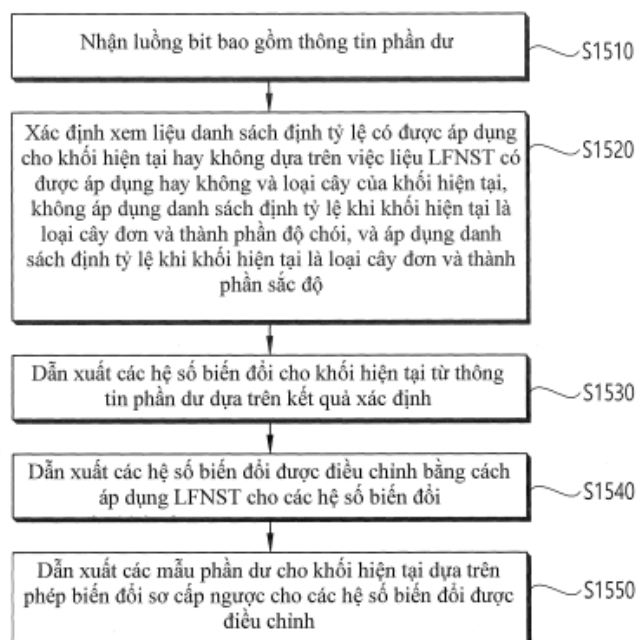
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ ẢNH, PHƯƠNG PHÁP MÃ HÓA ẢNH VÀ PHƯƠNG PHÁP TRUYỀN DỮ LIỆU CHO THÔNG TIN ẢNH

(21) 1-2022-04762

(57) Sáng chế này liên quan đến phương pháp giải mã ảnh, phương pháp mã hóa ảnh, phương pháp truyền dữ liệu cho thông tin ảnh và phương tiện lưu trữ kỹ thuật số đọc được bởi máy tính. Phương pháp giải mã ảnh theo sáng chế này bao gồm các bước là: áp dụng phép biến đổi không phân tách được tần số thấp (low-frequency non-separable transform, LFNST) cho các hệ số biến đổi để dẫn xuất các hệ số biến đổi được điều chỉnh; và dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên phép biến đổi sơ cấp ngược cho các hệ số biến đổi được điều chỉnh, trong đó bước dẫn xuất các hệ số biến đổi bao gồm: xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại dựa trên loại cây của khối hiện tại hay không và liệu LFNST có được áp dụng hay không; và dẫn xuất các hệ số biến đổi cho khối hiện tại từ thông tin phần dư dựa trên kết quả xác định, và khi loại cây của khối hiện tại là cây đơn và thành phần sắc độ, thì danh sách định tỷ lệ có thể được áp dụng.

FIG. 15



### **Lĩnh vực kỹ thuật được đề cập**

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

### **Tình trạng kỹ thuật của sáng chế**

Ngày nay, nhu cầu về các ảnh/các video độ phân giải cao và chất lượng cao như các ảnh/các video 4K, 8K hoặc độ phân giải siêu cao (Ultra High Definition, UHD) khác đang gia tăng trong nhiều lĩnh vực khác nhau. Bởi vì dữ liệu ảnh/video ngày càng có độ phân giải cao hơn và chất lượng cao hơn, nên lượng thông tin được truyền hoặc lượng bit tăng lên so với dữ liệu ảnh thông thường. Do đó, khi dữ liệu ảnh/video được truyền sử dụng các phương tiện như băng rộng dùng dây/không dây thông thường, hoặc dữ liệu ảnh/video được lưu trữ bằng phương tiện lưu trữ hiện có, thì chi phí truyền và chi phí lưu trữ bị tăng lên.

Thêm nữa, ngày nay, sự quan tâm và nhu cầu cho phương tiện nhúng như nội dung thực tế ảo (Virtual reality, VR), nội dung thực tế nhân tạo (Artificial reality, AR) hoặc ảnh toàn ký, hoặc tương tự đang gia tăng, và việc quảng bá cho các ảnh/các video có các đặc điểm ảnh khác so với các đặc điểm của các ảnh thực tế, như ảnh trò chơi đang gia tăng.

Theo đó, có nhu cầu cho kỹ thuật nén ảnh/video chất lượng cao để nén và truyền hiệu quả hoặc lưu trữ, và tái tạo thông tin của các ảnh/các video 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**

Mục đích kỹ thuật của sáng chế là để cung cấp phương pháp và thiết bị để tăng hiệu quả lập mã ảnh.

Khía cạnh kỹ thuật khác của sáng chế là để cung cấp phương pháp và thiết bị để tăng hiệu quả lượng tử hóa.

Khía cạnh kỹ thuật khác nữa của sáng chế này là để cung cấp phương pháp và thiết bị để tăng hiệu quả trong việc lượng tử hóa thành phần sắc độ trong loại cây đơn.

Theo phương án của sáng chế này, có phương pháp giải mã ảnh được thực hiện bởi thiết bị giải mã được cung cấp. Phương pháp có thể bao gồm bước dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư nhận được từ luồng bit, bước dẫn xuất các hệ số biến đổi được điều chỉnh bằng cách áp dụng phép biến đổi không phân tách được tần số thấp (low-frequency non-separable transform, LFNST) cho các hệ số biến đổi, và bước dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên phép biến đổi sơ cấp ngược cho các hệ số biến đổi được điều chỉnh, trong đó bước dẫn xuất các hệ số biến đổi có thể bao gồm việc xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu LFNST có được áp dụng hay không và loại cây của khối hiện tại, và bước dẫn xuất các hệ số biến đổi cho khối hiện tại từ thông tin phần dư dựa trên kết quả xác định, và khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ, thì danh sách định tỷ lệ có thể được áp dụng.

LFNST có thể không được áp dụng cho thành phần sắc độ của khối hiện tại.

Khi loại cây của khối hiện tại là cây đơn và LFNST được thực hiện trên khối hiện tại, thì danh sách định tỷ lệ có thể không được áp dụng cho thành phần độ chói của khối hiện tại.

Phương pháp có thể còn bao gồm bước nhận được thông tin cờ chỉ báo liệu danh sách định tỷ lệ là sẵn có khi LFNST được thực hiện hay không.

Khi thông tin cờ chỉ ra rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0, danh sách định tỷ lệ có thể không được áp dụng cho thành phần độ chói.

Khi thông tin cờ chỉ ra rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0, nếu loại cây của khối hiện tại là sắc độ kép, danh sách định tỷ lệ có thể không được áp dụng cho thành phần sắc độ.

Khi thông tin cờ chỉ ra rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0, nếu loại cây của khối hiện tại là độ chói kép, danh sách định tỷ lệ có thể không được áp dụng cho thành phần độ chói.

Theo phương án khác của sáng chế, có phương pháp mã hoá ảnh được thực hiện bởi thiết bị mã hoá được cung cấp. Phương pháp có thể bao gồm bước dẫn xuất các hệ số biến đổi cho khối hiện tại từ các mẫu phần dư cho khối hiện tại dựa trên quy trình biến đổi, bước xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu LFNST có được thực hiện trong quy trình biến đổi hay không và loại cây của khối hiện tại, và bước lượng tử hóa các hệ số biến đổi dựa trên việc xác định, trong đó khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ, thì danh sách định tỷ lệ có thể được áp dụng.

Thế nhưng, theo phương án khác của sáng chế này, có thể có phương tiện lưu trữ kỹ thuật số được cung cấp mà lưu trữ dữ liệu ảnh bao gồm thông tin ảnh được mã hóa và luồng bit được tạo ra theo phương pháp mã hóa ảnh được thực hiện bằng thiết bị mã hóa.

Theo phương án khác nữa của sáng chế này, có thể có phương tiện lưu trữ kỹ thuật số được cung cấp mà lưu trữ dữ liệu ảnh bao gồm thông tin ảnh được mã hóa và luồng bit để khiến cho thiết bị giải mã để thực hiện phương pháp giải mã ảnh.

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

Theo sáng chế, có thể tăng hiệu quả lượng tử hóa.

Theo sáng chế này là có thể tăng hiệu quả của việc lượng tử hóa thành phần sắc độ trong loại cây đơn.

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

### **Mô tả vắn tắt các hình vẽ**

Fig.1 là hình vẽ minh họa theo cách sơ lược ví dụ về hệ thống lập mã video/ảnh mà sáng chế có thể áp dụng với.

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

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

Fig.4 là hình vẽ minh họa theo cách giản lược sơ đồ đa biến đổi theo một phương án của sáng chế.

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

Fig.6 là sơ đồ để 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 phép biến đổi sơ cấp xuôi thành vectơ một chiều theo một ví dụ.

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

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

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

Fig.11 là sơ đồ minh họa sự sắp xếp dữ liệu xuất ra của LFNST xuôi theo một ví dụ.

Fig.12 là sơ đồ minh họa số dữ liệu đầu ra cho LFNST xuôi được giới hạn đến tối đa là 16 theo một ví dụ.

Fig.13 là sơ đồ minh họa việc đặt bằng không (0) trong khối mà LFNST 4x4 được áp dụng với theo một ví dụ.

Fig.14 là sơ đồ minh họa việc thiết đặt bằng không (0) trong khối mà LFNST 8x8 được áp dụng với theo một ví dụ.

Fig.15 là sơ đồ minh họa phương pháp giải mã ảnh theo ví dụ.

Fig.16 là sơ đồ minh họa phương pháp mã hóa ảnh theo ví dụ.

Fig.17 là hình vẽ minh họa cấu trúc của hệ thống tạo luồng nội dung mà sáng chế được áp dụng với.

### **Mô tả chi tiết sáng chế**

Trong khi sáng chế có thể là dễ dàng đối với các biến đổi khác nhau và bao gồm các phương án khác nhau, các phương án cụ thể của chúng đã được thể hiện trên các hình vẽ theo cách ví dụ và sẽ được mô tả chi tiết ngay. Tuy nhiên, điều này không được nhằm để giới hạn sáng chế đến các phương án cụ thể được bộc lộ ở đây. Thuật ngữ được sử dụng ở đây là cho mục đích mô tả chỉ các phương án cụ thể, và không được nhằm để giới hạn ý tưởng kỹ thuật của sáng chế. Các dạng số ít có thể bao gồm

cả dạng số nhiều, trừ khi ngữ cảnh nêu rõ ngược lại. Thuật ngữ như “chứa” và “có” nhằm để chỉ ra rằng các dấu hiệu, các số, các bước, các thao tác, các phần tử, các thành phần, hoặc các tổ hợp của chúng được sử dụng phần mô tả sau đây sẽ tồn tại và sẽ được hiểu là có khả năng tồn tại hoặc việc thêm vào một hoặc nhiều dấu hiệu, các số, các bước, các thao tác, các phần tử, các thành phần, hoặc các tổ hợp của chúng là không bị loại trừ từ trước.

Trong khi đó, mỗi thành phần trên các hình vẽ được mô tả ở đây được minh họa một cách độc lập để tạo thuận tiện cho việc mô tả để phân biệt các chức năng khác nhau, và, tuy nhiên, nó không có nghĩa là từng thành phần được hiện thực hóa bởi phần cứng hoặc phần mềm riêng biệt. Ví dụ, bất kỳ một hoặc nhiều thành phần trong số các thành phần có thể được kết hợp để tạo thành thành phần đơn, và bất kỳ thành phần đơn nào 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 vào phạm vi của quyền sáng chế của sáng chế này với điều kiện chúng không tách khỏi nguyên lý của tài liệu này.

Ở dưới đây, các phương án 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 dấu hiệu tham chiếu giống nhau được sử dụng cho các thành phần giống nhau trên các hình vẽ, và các phần mô tả lặp lại cho các thành phần giống nhau sẽ được lược bỏ.

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

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

Trong tài liệu này, video có thể nghĩa là chuỗi ảnh theo thời gian. Nhìn chung, hình ảnh nghĩa là đơn vị biểu diễn ảnh ở vùng thời gian cụ thể, và lát/ngói là đơn vị cấu thành phần của hình ảnh. Lát/ngói có thể bao gồm một hoặc nhiều đơn vị cây lập mã (Coding tree unit, CTU). Một hình ảnh có thể được cấu thành bằng một hoặc nhiều lát/ngói. Một hình ảnh có thể được cấu thành bằng một hoặc nhiều nhóm ngôi. Một nhóm ngôi có thể bao gồm một hoặc nhiều ngôi.

Điểm ảnh (pixel) hoặc pel có thể nghĩa là đơn vị nhỏ nhất cấu thành một ảnh (hoặc hình ảnh). Cũng vậy, “mẫu” có thể được sử dụng như là thuật ngữ tương ứng với điểm ảnh. Mẫu có thể biểu diễn chung cho điểm ảnh hoặc giá trị của điểm ảnh, và có thể biểu diễn chỉ điểm ảnh/giá trị điểm ảnh của thành phần độ chói hoặc chỉ điểm ảnh/giá trị điểm ảnh của thành phần sắc độ. Thay vào đó, mẫu có thể đề cập đến giá trị điểm ảnh trong miền không gian, hoặc khi giá trị điểm ảnh này được chuyển đổi sang miền tần số, thì nó có thể đề cập đến hệ số biến đổi trong miền tần số.

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

Trong tài liệu này, thuật ngữ “/” và “,” nên được diễn dịch để thể hiện “và/hoặc”. Ví dụ, cách diễn đạt “A/B” có thể nghĩa là “A và/hoặc B”. Thêm nữa, “A, B” có thể nghĩa là “A và/hoặc B”. Thêm nữa, “A/B/C” có thể nghĩa là “ít nhất một trong số A,

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

Thêm nữa, trong sáng chế này, thuật ngữ “hoặc” nên được diễn dịch để thể hiện “và/hoặc”. Ví dụ, biểu diễn “A hoặc B” có thể bao gồm 1) chỉ có A, 2) chỉ có B, và/hoặc 3) cả hai A và B. Nói cách khác, thuật ngữ “hoặc” trong tài liệu này nên được dẫn giải là để chỉ thị “việc thêm vào hoặc thay vào đó”.

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

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

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

Các đặc điểm kỹ thuật được mô tả riêng trên một hình vẽ trong sáng chế này có thể được thực hiện riêng hoặc có thể được thực hiện đồng thời.

Fig.1 là hình vẽ minh họa theo cách sơ lược ví dụ về hệ thống lập mã video/ảnh mà sáng chế có thể áp dụng với.

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

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

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

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

Bộ truyền có thể truyền thông tin hoặc dữ liệu video/ảnh được mã hóa được xuất ra dạng luồng bit đến bộ thu của thiết bị thu thông qua phương tiện lưu trữ kỹ

thuật số hoặc mạng dưới dạng tập tin hoặc tạo luồng. Phương tiện lưu trữ kỹ thuật số có thể bao gồm phương tiện lưu trữ khác nhau như USB, SD, CD, DVD, Blu-ray, HDD, SSD, và tương tự. Bộ truyền có thể bao gồm phần tử để tạo ra tập tin phương tiện thông qua định dạng tập tin định trước và có thể bao gồm phần tử để truyền qua mạng quảng bá/truyền thông. Bộ thu có thể nhận/tách luồng bit và truyền luồng bit nhận được đến thiết bị giải mã.

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

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

Fig.2 là sơ đồ minh họa theo cách sơ lược cấu hình của thiết bị mã hóa video/ảnh mà sáng chế có thể áp dụng với. Dưới đây, điều được đề cập đến như là thiết bị mã hóa video có thể bao gồm thiết bị mã hóa ảnh.

Tham chiếu đến Fig.2, thiết bị mã hóa 200 có thể bao gồm bộ phân vùng ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropy 240, bộ cộng 250, bộ lọc 260, và bộ nhớ 270. Bộ dự đoán 220 có thể bao gồm bộ liên dự đoán 221 và bộ nội dự đoán 222. Bộ xử lý phần dư 230 có thể bao gồm bộ biến đổi 232, bộ lượng tử hóa 233, bộ giải lượng tử hóa 234, bộ biến đổi ngược 235. Bộ xử lý phần dư 230 có thể còn bao gồm bộ trừ 231. Bộ cộng 250 có thể được gọi là bộ tái dựng hoặc bộ tạo ra khối được tái dựng. Bộ phân vùng hình ảnh 210, bộ dự đoán 220, bộ xử lý phần dư 230, bộ mã hóa entropy 240, bộ cộng 250, và bộ lọc 260, đã được mô tả bên trên, có thể được cấu thành bởi một hoặc nhiều thành phần phần cứng (ví dụ, các bộ chip hoặc các bộ xử lý bộ mã hóa) theo một phương án thực hiện. Ngoài ra, bộ nhớ 270 có thể bao gồm bộ đệm hình ảnh được giải mã (Decoded picture buffer, DPB) có thể được

cấu thành bằng phương tiện lưu trữ kỹ thuật số. Thành phần phần cứng có thể còn bao gồm bộ nhớ 270 như thành phần bên trong/bên ngoài.

Bộ phân vùng ảnh 210 có thể phân vùng ảnh đầu vào (hoặc hình ảnh hoặc khung) được nhập vào thiết bị mã hóa 200 thành một hoặc nhiều đơn vị xử lý. Như một ví dụ, đơn vị xử lý có thể được gọi là đơn vị lập mã (coding unit, CU). Trong trường hợp này, bắt đầu với đơn vị cây lập mã (Coding tree unit, CTU) hoặc đơn vị lập mã lớn nhất (Largest coding unit, LCU), đơn vị lập mã có thể được phân vùng theo cách đệ quy theo cấu trúc cây tứ phân nhị phân tam phân (Quad-tree binary-tree ternary-tree, QTBTBT). Ví dụ, một đơn vị lập mã có thể được phân vùng thành nhiều đơn vị lập mã có độ sâu sâu hơn dựa trên cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc tam phân. Trong trường hợp này, ví dụ, cấu trúc cây tứ phân có thể được áp dụng đầu tiên và cấu trúc cây nhị phân và/hoặc cấu trúc tam phân có thể được áp dụng sau. Thay vào đó, cấu trúc cây nhị phân có thể được áp dụng trước. Thủ tục lập mã theo sáng chế này có thể được thực hiện dựa trên đơn vị lập mã cuối cùng mà không được phân vùng thêm. Trong trường hợp này, đơn vị lập mã lớn nhất có thể được sử dụng trực tiếp như đơn vị lập mã cuối cùng dựa trên hiệu quả lập mã theo đặc tính ảnh. Thay vào đó, đơn vị lập mã có thể được phân vùng theo cách đệ quy thành các đơn vị lập mã có chiều sâu sâu hơn như được yêu cầu sao cho đơn vị lập mã có kích thước tối ưu có thể được sử dụng như đơn vị lập mã cuối cùng. Ở đây, thủ tục lập mã có thể bao gồm các thủ tục như dự đoán, biến đổi, và tái dựng, mà sẽ được mô tả sau. Như là một ví dụ khác, đơn vị xử lý có thể còn bao gồm đơn vị dự đoán (Prediction unit, PU) hoặc đơn vị biến đổi (Transform unit, TU). Trong trường hợp này, đơn vị dự đoán và đơn vị biến đổi có thể được chia hoặc phân chia từ đơn vị lập mã cuối cùng được mô tả đến ở trên. Đơn vị dự đoán có thể là đơn vị của việc dự đoán mẫu, và đơn vị biến đổi có thể là đơn vị để dẫn xuất hệ số biến đổi và/hoặc đơn vị để dẫn xuất tín hiệu phần dư từ hệ số biến đổi.

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

Bộ trừ 231 trừ tín hiệu dự đoán (khối được dự đoán, mảng mẫu dự đoán) được xuất từ bộ dự đoán 220 từ tín hiệu ảnh đầu vào (khối gốc, mảng mẫu gốc) để tạo ra tín hiệu phần dư (khối phần dư, mảng mẫu phần dư), và tín hiệu phần dư được tạo ra được truyền đến bộ biến đổi 232. Bộ dự đoán có thể thực hiện việc dự đoán trên khối mục tiêu cần xử lý (ở dưới đây, được đề cập tới như là “khối hiện tại”) và có thể tạo ra khối được dự đoán bao gồm các mẫu dự đoán cho khối hiện tại. Bộ dự đoán 220 có thể xác định liệu việc nội dự đoán hay liên dự đoán được áp dụng trên cơ sở khối hiện tại hay CU. Như được thảo luận sau trong phần mô tả về mỗi chế độ dự đoán, bộ dự đoán có thể tạo ra thông tin khác nhau liên quan đến dự đoán, như thông tin chế độ dự đoán, và truyền thông tin được tạo ra đến bộ mã hóa entropy 240. Thông tin về việc dự đoán có thể được mã hóa trong bộ mã hóa entropy 240 và được xuất ra dưới dạng luồng bit.

Bộ nội dự đoán 222 có thể dự đoán khối hiện tại bằng cách tham chiếu đến các mẫu trong hình ảnh hiện tại. Các mẫu được tham chiếu có thể được đặt ở vùng lân cận hoặc xa khỏi khối hiện tại theo chế độ dự đoán. Trong nội dự đoán, các chế độ dự đoán có thể bao gồm nhiều chế độ không có hướng và nhiều chế độ có hướng. Các chế độ không hướng có thể bao gồm, ví dụ, chế độ DC và chế độ phẳng. Chế độ có hướng có thể bao gồm, ví dụ, 33 chế độ dự đoán có hướng hoặc 65 chế độ dự đoán có hướng theo cấp độ chi tiết của hướng dự đoán. Tuy nhiên, điều này chỉ đơn thuần là ví dụ, và nhiều hoặc ít chế độ dự đoán có hướng hơn có thể được sử dụng thứ cấp

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

Bộ liên dự đoán 221 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên hình ảnh tham chiếu. Lúc 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 trên cơ sở khối, khối con hoặc mẫu dựa trên sự tương quan về thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng liên dự đoán (dự đoán L0, dự đoán L1, song dự đoán, v.v.). Trong trường hợp của việc liên dự đoán, khối lân cận có thể bao gồm khối lân cận không gian hiện có trong hình ảnh hiện tại và khối lân cận thời gian hiện có trong hình ảnh tham chiếu. Hình ảnh tham chiếu bao gồm khối tham chiếu và ảnh tham chiếu bao gồm khối lân cận thời gian có thể là giống nhau hoặc khác với nhau. Trong trường hợp của việc liên dự đoán, khối lân cận có thể bao gồm khối lân cận không gian hiện có trong hình ảnh hiện tại và khối lân cận thời gian hiện có trong hình ảnh tham chiếu. Ví dụ, bộ liên dự đoán 221 có thể tạo cấu trúc danh sách ứng viên thông tin chuyển động dựa trên các khối lân cận và tạo ra thông tin chỉ báo ứng viên nào được sử dụng để dẫn xuất vector chuyển động và/hoặc chỉ số hình ảnh tham chiếu của khối hiện tại. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau. Ví dụ, trong trường hợp chế độ bỏ cóc và chế độ hợp nhất, bộ liên dự đoán 221 có thể sử dụng thông tin chuyển động của khối lân cận như thông tin chuyển động của khối hiện tại. Trong chế độ bỏ cóc, không giống chế độ hợp nhất, tín hiệu phân dư có thể không được truyền. Trong trường hợp chế độ dự đoán thông tin chuyển động (motion vector prediction, MVP), vector chuyển động của khối lân cận có thể được sử dụng

như bộ dự đoán vector chuyển động và vector chuyển động của khối hiện tại có thể được chỉ báo bằng cách báo hiệu mức chênh lệch vector chuyển động.

Bộ dự đoán 220 có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc liên dự đoán cho việc dự đoán trên một khối, và cũng như vậy, có thể áp dụng việc nội dự đoán và việc liên dự đoán cùng lúc. Điều này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (Combined inter and intra prediction, CIIP). Hơn nữa, bộ dự đoán có thể căn cứ vào chế độ dự đoán sao chép nội khối (intra block copy, IBC) hoặc chế độ bảng màu (palette) để thực hiện việc dự đoán trên khối. Chế độ dự đoán IBC hoặc chế độ bảng màu có thể được sử dụng cho việc lập mã ảnh/video nội dung của trò chơi hoặc dạng tương tự, ví dụ lập mã nội dung màn (screen content coding, SCC). Mặc dù IBC về cơ bản thực hiện việc dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện tương tự với việc liên dự đoán trong đó nó dẫn xuất khối tham chiếu trong khối hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong các kỹ thuật liên dự đoán được mô tả trong sáng chế.

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

thước hoặc có thể được áp dụng cho các khối có kích thước thay đổi chứ không phải kích thước vuông.

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

bit có thể được truyền qua mạng hoặc được lưu trữ trong phương tiện lưu trữ kỹ thuật số. Ở đây, mạng có thể bao gồm mạng quảng bá, mạng truyền thông và/hoặc tương tự, và phương tiện lưu trữ dạng số có thể bao gồm các phương tiện lưu trữ khác nhau như USB, SD, CD, DVD, Blu-ray, HDD, SSD và tương tự. Bộ truyền (không được thể hiện) truyền tín hiệu được xuất ra từ bộ mã hóa entropy 240 và/hoặc bộ lưu trữ (không được thể hiện) mà lưu trữ tín hiệu này có thể được tạo cấu hình làm phần tử bên trong/bên ngoài của thiết bị mã hóa 200, hoặc bộ truyền cũng có thể được bao gồm trong bộ mã hóa entropy 240.

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

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

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

Hình ảnh được tái dựng được điều chỉnh mà được truyền tới bộ nhớ 270 có thể được sử dụng làm hình ảnh tham chiếu trong bộ liên dự đoán 221. Thông qua điều này, thiết bị mã hóa có thể tránh việc không khớp dự đoán trong thiết bị mã hóa 100 và thiết bị giải mã khi việc liên dự đoán được áp dụng, và có thể cũng nâng cao hiệu quả lập mã.

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

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

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

Khi luồng bit bao gồm thông tin video/ảnh được nhập vào, thì thiết bị giải mã 300 có thể tái dựng ảnh tương ứng với quy trình mà thông tin video/ảnh được xử lý bởi thiết bị mã hóa của Fig.2. Ví dụ, thiết bị giải mã 300 có thể dẫn xuất các đơn vị/các khối dựa trên thông tin liên quan đến sự phân vùng khối thu được từ luồng bit. Thiết bị giải mã 300 có thể thực hiện việc giải mã bằng cách sử dụng đơn vị xử lý trong thiết bị mã hóa. Do đó, đơn vị xử lý việc giải mã có thể là, ví dụ, đơn vị lập mã, có thể được phân vùng theo cấu trúc cây tứ phân, cấu trúc cây nhị phân, và/hoặc cấu trúc cây tam phân từ đơn vị cây lập mã hoặc đơn vị lập mã lớn nhất. Một hoặc nhiều đơn vị biến đổi có thể được dẫn xuất từ đơn vị lập mã này. Và, tín hiệu ảnh được tái dựng được giải mã và được xuất ra thông qua thiết bị giải mã 300 có thể được tái tạo qua bộ tái tạo.

Thiết bị giải mã 300 có thể thu tín hiệu mà được xuất ra từ thiết bị mã hóa trên Fig.2 dưới dạng luồng bit, và tín hiệu thu được có thể được giải mã thông qua bộ giải mã entropy 310. Ví dụ, bộ giải mã entropy 310 có thể phân tích cú pháp luồng bit để

dẫn xuất thông tin (ví dụ, thông tin video/ảnh) cần thiết cho việc tái dựng ảnh (hoặc tái dựng hình ảnh). Thông tin video/ảnh có thể còn bao gồm thông tin về các tập tham số khác nhau như tập tham số thích ứng (APS), tập tham số hình ảnh (PPS), tập tham số trình tự (SPS), tập tham số video (VPS) hoặc tương tự. Ngoài ra, thông tin video/ảnh có thể còn bao gồm thông tin ràng buộc chung. Thiết bị giải mã có thể giải mã hình ảnh còn dựa trên thông tin về tập tham số và/hoặc thông tin ràng buộc chung. Trong phần bộc lộ này, thông tin được báo hiệu/thu được và/hoặc các phần tử cú pháp, mà sẽ được mô tả sau, có thể được giải mã thông qua thủ tục giải mã và được thu được từ luồng bit. Ví dụ, bộ giải mã entropy 310 có thể giải mã thông tin trong luồng bit dựa trên phương pháp mã hoá như mã hoá Golomb hàm số mũ, CAVLC, CABAC, hoặc tương tự, và có thể xuất ra giá trị của phần tử cú pháp cần thiết cho việc tái dựng ảnh và các giá trị được lượng tử hoá của hệ số biến đổi liên quan đến phần dư. Cụ thể hơn, phương pháp giải mã entropy CABAC có thể nhận nhị phân (bin) tương ứng với mỗi phần tử cú pháp trong luồng bit, xác định mô hình ngữ cảnh nhờ sử dụng thông tin về phần tử cú pháp mục tiêu giải mã và thông tin giải mã của các khối lân cận và các khối mục tiêu giải mã hoặc thông tin về ký hiệu/bin được giải mã ở bước trước đó, dự đoán xác suất tạo bin theo mô hình ngữ cảnh xác định được và thực hiện việc giải mã số học đối với bin này để tạo ra ký hiệu tương ứng với mỗi giá trị phần tử cú pháp. Ở đây, phương pháp giải mã entropy CABAC có thể cập nhật mô hình ngữ cảnh sử dụng thông tin về ký hiệu/bin được giải mã cho mô hình ngữ cảnh của ký hiệu/bin tiếp theo sau khi xác định mô hình ngữ cảnh này. Thông tin về việc dự đoán trong số thông tin được giải mã ở bộ giải mã entropy 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 dư, tức là các hệ số biến đổi đã được lượng tử hoá, mà việc giải mã entropy đã được thực hiện trong bộ giải mã entropy 310 và thông tin tham số liên quan có thể được nhập vào bộ xử lý phần dư 320. Bộ xử lý phần dư 320 có thể dẫn xuất tín hiệu dư (khối phần dư, các mẫu phần dư, mảng mẫu phần dư). Thêm nữa, thông tin về việc lọc trong số thông

tin được giải mã bởi bộ giải mã entropy 310 có thể được cấp đến bộ lọc 350. Trong khi đó, bộ thu (không được thể hiện) mà thu tín hiệu được xuất ra từ thiết bị mã hóa có thể còn cấu thành thiết bị giải mã 300 như thành phần bên trong/bên ngoài, và bộ thu có thể là thành phần của bộ giải mã entropy 310. Trong khi đó, thiết bị giải mã theo phần bộc lộ này có thể được gọi là thiết bị lập mã video/ảnh/hình ảnh, và thiết bị giải mã cũng có thể được phân loại thành bộ giải mã thông tin (bộ giải mã thông tin video/ảnh/hình ảnh) và bộ giải mã mẫu (bộ giải mã mẫu video/ảnh/hình ảnh). Bộ giải mã thông tin có thể bao gồm bộ giải mã entropy 310, và bộ giải mã mẫu có thể bao gồm ít nhất một trong bộ giải lượng tử hóa 321, bộ biến đổi ngược 322, bộ cộng 340, bộ lọc 350, bộ nhớ 360, bộ liên dự đoán 332, và bộ nội dự đoán 331.

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

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

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

Bộ dự đoán có thể tạo ra tín hiệu dự đoán dựa trên các phương pháp dự đoán khác nhau. Ví dụ, bộ dự đoán có thể áp dụng việc nội dự đoán hoặc việc liên dự đoán cho việc dự đoán trên một khối, và cũng như vậy, có thể áp dụng việc nội dự đoán và việc liên dự đoán cùng lúc. Điều này có thể được gọi là liên dự đoán và nội dự đoán kết hợp (CIIP). Ngoài ra, bộ dự đoán có thể thực hiện việc sao chép nội khối (IBC) cho việc dự đoán trên khối. Việc sao chép nội khối có thể được sử dụng cho việc lập mã ảnh/video nội dung của trò chơi hoặc tương tự, như lập mã nội dung màn (SCC). Mặc dù IBC về cơ bản thực hiện việc dự đoán trong khối hiện tại, nhưng nó có thể được thực hiện tương tự với việc liên dự đoán trong đó nó dẫn xuất khối tham chiếu trong khối hiện tại. Tức là, IBC có thể sử dụng ít nhất một trong các kỹ thuật liên dự đoán được mô tả trong sáng chế.

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

Bộ liên dự đoán 332 có thể dẫn xuất khối được dự đoán cho khối hiện tại dựa trên khối tham chiếu (mảng mẫu tham chiếu) được quy định bởi vector chuyển động trên hình ảnh tham chiếu. Lúc 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 trên cơ sở khối, khối con hoặc mẫu dựa trên sự tương quan về thông tin chuyển động giữa khối lân cận và khối hiện tại. Thông tin chuyển động có thể bao gồm vector chuyển động và chỉ số hình ảnh tham chiếu. Thông tin chuyển động có thể còn bao gồm thông tin hướng liên dự đoán (dự đoán L0, dự đoán L1, song dự đoán, v.v.). Trong trường hợp của việc liên dự đoán, khối lân cận có thể bao gồm khối lân cận không gian hiện có trong hình ảnh hiện tại và khối lân cận thời gian hiện có trong hình 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 dẫn xuất vector chuyển động và/hoặc chỉ số hình ảnh tham chiếu của khối hiện tại dựa trên thông tin chọn ứng viên nhận được. Việc liên dự đoán có thể được thực hiện dựa trên các chế độ dự đoán khác nhau, và thông tin về việc dự đoán có thể bao gồm thông tin chỉ báo chế độ liên dự đoán cho khối hiện tại.

Bộ cộng 340 có thể tạo ra tín hiệu được tái dựng (hình ảnh được tái dựng, khối được tái dựng, mảng mẫu được tái dựng) bằng cộng tín hiệu dư thu được vào tín hiệu dự đoán (khối được dự đoán, mảng mẫu được dự đoán) được xuất ra từ bộ dự đoán 330). Khi không có tín hiệu phần dư cho khối mục tiêu xử lý như trong trường hợp chế độ nhảy cóc được áp dụng, thì khối được dự đoán có thể được sử dụng như khối được tái dựng.

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

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

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

Hình ảnh được tái dựng (được điều chỉnh) mà đã được lưu trữ trong DPB của bộ nhớ 360 có thể được sử dụng như hình ảnh tham chiếu trong bộ liên dự đoán 332. Bộ nhớ 360 có thể lưu trữ thông tin chuyển động của khối trong hình ảnh hiện tại mà thông tin chuyển động đã được dẫn xuất (hoặc được giải mã) và/hoặc thông tin chuyển động của các khối trong hình ảnh đã được tái dựng. Thông tin chuyển động được lưu trữ có thể được truyền tới bộ liên dự đoán 260 để được tận dụng 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 tái dựng của các khối được tái dựng trong hình ảnh hiện tại, và truyền chúng đến bộ nội dự đoán 331.

Trong bản mô tả này, các ví dụ được mô tả trong bộ dự đoán 330, bộ giải lượng tử hóa 321, bộ biến đổi ngược 322, và bộ lọc 350 của thiết bị giải mã 300 có thể được áp dụng tương tự hoặc tương ứng lần lượt cho bộ dự đoán 220, bộ giải lượng tử hóa 234, bộ biến đổi ngược 235, và bộ lọc 260 của thiết bị mã hóa 200.

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

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

Fig.4 là hình vẽ minh họa theo cách sơ lược kỹ thuật đa biến đổi theo 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 đề cập đến ở trên, và bộ biến đổi ngược có thể tương ứng với bộ biến đổi ngược trong thiết bị mã hóa của Fig.2 nêu trên, hoặc với bộ biến đổi ngược trong thiết bị giải mã của Fig.3.

Bộ biến đổi này có thể dẫn xuất các hệ số biến đổi (sơ cấp) bằng cách thực hiện phép biến đổi sơ cấp dựa trên các mẫu phần dư (mảng mẫu phần dư) trong khối phần dư (S410). Phép biến đổi sơ cấp này có thể được đề cập đến như phép biến đổi lõi. Ở đây, phép biến đổi sơ cấp có thể được dựa trên việc chọn đa biến đổi (multiple

transform selection, MTS), và khi đa biến đổi được áp dụng như phép biến đổi sơ cấp, thì nó có thể được đề cập đến như phép biến đổi đa lõi.

Phép biến đổi đa lõi này có thể biểu thị phương pháp biến đổi sử dụng thêm phép biến đổi cosin rời rạc (DCT) loại 2 và phép biến đổi sin rời rạc (DST) loại 7, DCT loại 8, và/hoặc DST loại 1. Tức là, phép biến đổi đa lõi này có thể biểu thị phương pháp biến đổi để biến đổi tín hiệu phần dư (hoặc khối phần dư) của miền không gian thành các hệ số biến đổi (hoặc các hệ số biến đổi sơ cấp) của miền tần số dựa trên nhiều nhân biến đổi được chọn từ trong số DCT loại 2, DST loại 7, DCT loại 8 và DST loại 1. Ở đây, các hệ số biến đổi sơ cấp có thể được gọi các hệ số biến đổi tạm thời từ góc độ 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 phép biến đổi từ miền không gian sang miền tần số cho tín hiệu phần dư (hoặc khối phần dư) dựa trên loại DCT 2. Không giống việc này, khi phép biến đổi đa lõi 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 phép biến đổi từ miền không gian sang miền tần số đối với tín hiệu phần dư (hoặc khối phần dư) dựa trên DCT loại 2, DST loại 7, DCT loại 8, và/hoặc DST loại 1. Ở đây, DCT loại 2, DST loại 7, DCT loại 8, và DST loại 1 có thể được gọi là loại biến đổi, nhân biến đổi hoặc lõi biến đổi. Các loại biến đổi DCT/DST này có thể được xác định dựa trên các hàm cơ sở.

Khi đa biến đổi lõi được thực hiện, thì nhân biến đổi thẳng và nhân biến đổi ngang cho khối mục tiêu có thể được chọn từ trong số các nhân biến đổi, việc biến đổi thẳng có thể được thực hiện trên khối mục tiêu dựa trên nhân biến đổi thẳng, và việc biến đổi ngang có thể được thực hiện trên khối mục tiêu dựa trên nhân biến đổi ngang. Ở đây, việc biến đổi ngang có thể thể hiện việc biến đổi trên các thành phần ngang của khối mục tiêu, và việc biến đổi thẳng có thể thể hiện việc biến đổi trên các

thành phần thẳng của khối mục tiêu. Nhân biến đổi thẳng/nhân biến đổi ngang có thể được xác định theo cách thích ứng dựa trên chế độ dự đoán và/hoặc chỉ số biến đổi cho khối mục tiêu (CU hoặc khối con) bao gồm khối phần dư.

Thêm nữa, theo ví dụ, nếu phép 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 thiết đặt bằng cách thiết đặt các hàm cơ sở cụ thể thành các giá trị định trước và kết hợp các hàm cơ sở cần được áp dụng trong phép biến đổi thẳng hoặc phép biến đổi ngang. Ví dụ, khi nhân biến đổi ngang được biểu diễn như trTypeHor và nhân biến đổi hướng thẳng được biểu diễn như trTypeVer, thì giá trị trTypeHor hoặc trTypeVer là 0 có thể được thiết đặt là DCT2, giá trị trTypeHor hoặc trTypeVer là 1 có thể được thiết đặt là DST7, và giá trị trTypeHor hoặc trTypeVer là 2 có thể được thiết đặt là DCT8.

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

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

[Bảng 1]

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

Bộ biến đổi có thể dẫn xuất các hệ số biến đổi (thứ cấp) được điều chỉnh bằng cách thực hiện phép biến đổi thứ cấp dựa trên các hệ số biến đổi (sơ cấp) (S420). Phép biến đổi sơ cấp là phép biến đổi từ miền không gian sang miền tần số, và phép biến đổi thứ cấp đề cập đến phép biến đổi thành biểu thức gọn hơn qua việc sử dụng mối tương quan tồn tại giữa các hệ số biến đổi (sơ cấp). Phép biến đổi thứ cấp có thể bao gồm phép biến đổi không phân tách được. Trong trường hợp này, phép biến đổi thứ cấp này có thể được gọi là phép biến đổi thứ cấp không phân tách được (non-separable secondary transform, NSST), hoặc phép biến đổi thứ cấp không phân tách được phụ thuộc chế độ (mode-dependent non-separable secondary transform, MDNSST). Phép biến đổi thứ cấp không phân tách được có thể biểu thị phép biến đổi mà tạo ra các hệ số biến đổi được điều chỉnh (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 phân tách được, các hệ số biến đổi (sơ cấp) mà được dẫn xuất thông qua phép biến đổi sơ cấp. Lúc này, phép biến đổi thẳng và phép biến đổi ngang có thể là không được áp dụng một cách riêng rẽ (phép biến đổi ngang và phép biến đổi thẳng có thể là không được áp dụng một cách độc lập) cho các hệ số biến đổi (sơ cấp), nhưng các phép biến đổi này có thể được áp dụng cùng một lúc dựa trên ma trận biến đổi không phân tách được. Nói cách khác, phép biến đổi thứ cấp không phân tách được có thể biểu diễn phương pháp biến đổi trong đó không được áp dụng riêng theo hướng thẳng và hướng ngang cho các hệ số biến đổi (sơ cấp), và ví dụ, các tín hiệu hai chiều (các hệ số biến đổi) được sắp xếp lại thành tín hiệu một chiều thông qua hướng nhất định được xác định (ví dụ, hướng hàng-thứ nhất hoặc hướng cột-thứ nhất), và sau đó các hệ số biến đổi được điều chỉnh (hoặc các hệ số biến đổi thứ cấp) được tạo ra dựa trên ma trận biến đổi không phân tách được. Ví dụ, theo thứ tự hàng-thứ nhất, các khối  $M \times N$  được sắp xếp theo tuyến với thứ tự là hàng thứ nhất, hàng thứ hai, ..., và hàng thứ  $N$ . Theo thứ tự cột-thứ nhất, các khối  $M \times N$  được sắp xếp trong tuyến với thứ tự là cột thứ nhất, cột thứ hai, ..., và cột thứ  $N$ . Phép biến đổi thứ cấp không phân tách

được có thể được áp dụng cho vùng trên cùng-bên trái của khối được tạo cấu hình với các hệ số biến đổi (sơ cấp) (sau đây có thể được gọi là khối hệ số biến đổi). Ví dụ, nếu chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi đều bằng hoặc lớn hơn 8, thì phép biến đổi thứ cấp không phân tách được 8x8 có thể được áp dụng cho vùng phía trên cùng-bên trái 8x8 của khối hệ số biến đổi. Thêm 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 là nhỏ hơn 8, thì sau đó phép biến đổi thứ cấp không phân tách được 4x4 có thể được áp dụng cho vùng phía trên cùng-bên trái  $\min(8, W) \times \min(8, H)$  của khối hệ số biến đổi. Tuy nhiên, phương án không được giới hạn ở điều này, và ví dụ, thậm chí nếu chỉ điều kiện mà chiều rộng (W) hoặc chiều cao (H) của khối hệ số biến đổi là bằng hoặc lớn hơn 4 được thỏa mãn, thì phép biến đổi thứ cấp không phân tách được 4x4 có thể được áp dụng cho vùng phía trên cùng-bên trái  $\min(8, W) \times \min(8, H)$  của khối hệ số biến đổi.

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

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

[Phương trình 1]

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

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

[Phương trình 2]

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

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

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

[Phương trình 3]

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

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

Thông qua Phương trình 3 nêu trên, vector hệ số biến đổi 16x1  $\vec{F}$  có thể được dẫn xuất, và  $\vec{F}$  có thể được tổ chức lại thành khối 4x4 thông qua thứ tự quét (ngang, thẳng, đường chéo và tương tự). Tuy nhiên, phép tính nêu trên là một ví dụ, và phép biến đổi hypercube-Givens (Hypercube-Givens Transform, HyGT) hoặc tương tự cũng có thể được dùng để tính toán phép biến đổi thứ cấp không phân tách được để giảm sự phức tạp tính toán của phép biến đổi thứ cấp không phân tách được.

Trong khi đó, theo phép biến đổi thứ cấp không phân tách được, thì nhân biến đổi (hoặc lỗi biến đổi, loại biến đổi) là có thể được chọn để là chế độ phụ thuộc. Trong trường hợp này, chế độ có thể bao gồm chế độ nội dự đoán và/hoặc chế độ liên dự đoán.

Như đã mô tả trên đây, phép biến đổi thứ cấp không phân tách được là có thể được thực hiện dựa trên phép biến đổi 8 x 8 hoặc phép biến đổi 4 x 4 mà xác định được dựa trên chiều rộng (W) và chiều cao (H) của khối hệ số biến đổi. Phép biến đổi 8x8 đề cập đến phép biến đổi mà có thể áp dụng được cho vùng 8x8 được bao gồm trong khối hệ số biến đổi khi cả W và H là 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, phép biến đổi 4x4

đề cập đến phép biến đổi mà có thể áp dụng được cho vùng  $4 \times 4$  được bao gồm trong khối hệ số biến đổi khi cả  $W$  và  $H$  là bằng hoặc lớn hơn 4, và vùng  $4 \times 4$  có thể là vùng  $4 \times 4$  trên cùng-bên trái trong khối hệ số biến đổi. Ví dụ, ma trận nhân biến đổi  $8 \times 8$  có thể là ma trận  $64 \times 64 / 16 \times 64$ , và ma trận nhân biến đổi  $4 \times 4$  có thể là ma trận  $16 \times 16 / 8 \times 16$ .

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

Tuy nhiên, bởi vì kích thước biến đổi, tức là, kích thước của vùng mà việc biến đổi được áp dụng với, có thể là, ví dụ, kích thước ngoài  $8 \times 8$  hoặc  $4 \times 4$ , số lượng tập có thể là  $n$ , và số lượng nhân biến đổi trong mỗi tập có thể là  $k$ .

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

Để tham khảo, ví dụ, chế độ nội dự đoán có thể bao gồm hai chế độ nội dự đoán không có hướng (hoặc không có góc) và 65 chế độ nội dự đoán có hướng (hoặc có góc). Các chế độ nội dự đoán không có hướng có thể bao gồm chế độ nội dự đoán phẳng là số 0 và chế độ nội dự đoán DC là số 1, và các chế độ nội dự đoán có hướng

có thể bao gồm 65 chế độ nội dự đoán là từ số 2 đến số 66. Tuy nhiên, đây là một ví dụ, và sáng chế này có thể được áp dụng thậm chí khi số lượng chế độ nội dự đoán là khác. Trong khi đó, trong một số trường hợp, chế độ nội dự đoán số 67 có thể còn được sử dụng, và chế độ nội dự đoán số 67 có thể biểu diễn chế độ mô hình tuyến tính (linear model, LM).

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

Tham chiếu đến Fig.5, dựa trên chế độ nội dự đoán 34 có hướng dự đoán theo đường chéo bên trái hướng lên trên, các chế độ nội dự đoán có thể được chia thành các chế độ nội dự đoán có hướng tính ngang và các chế độ nội dự đoán có hướng tính thẳng. Trên Fig.5, H và V lần lượt biểu thị hướng tính ngang và hướng tính thẳng, và các số từ -32 đến 32 thể hiện các độ dịch trong các đơn vị  $1/32$  trên vị trí lưới mẫu. Các số này có thể biểu diễn độ bù trừ cho giá trị chỉ số chế độ. Các chế độ nội dự đoán từ 2 đến 33 có hướng tính ngang và các chế độ nội dự đoán từ 34 đến 66 có hướng tính thẳng. Nói đúng ra, thì chế độ nội dự đoán 34 có thể được xem xét như không ngang cũng không thẳng, nhưng có thể được phân loại như thuộc về hướng tính ngang khi xác định tập biến đổi của phép biến đổi thứ cấp. Điều này là bởi vì dữ liệu đầu vào được chuyển vị để được sử dụng cho chế độ hướng thẳng 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. Việc chuyển vị dữ liệu đầu vào nghĩa là các hàng và các cột của dữ liệu khối hai chiều  $M \times N$  được chuyển dịch thành dữ liệu  $N \times M$ . Chế độ nội dự đoán 18 và chế độ nội dự đoán 50 có thể lần lượt biểu diễn chế độ nội dự đoán ngang và chế độ nội dự đoán thẳng và chế độ nội dự đoán 2 có thể được đề cập đến như chế độ nội dự đoán theo đường chéo bên phải hướng lên trên bởi vì chế độ nội dự đoán 2 có điểm ảnh tham chiếu bên trái và thực hiện việc dự đoán theo hướng bên phải hướng lên trên. Cũng như vậy, chế độ nội dự đoán 34 có thể được đề cập đến như chế độ nội dự đoán theo đường chéo bên phải hướng

xuống dưới, và chế độ nội dự đoán 66 có thể được đề cập đến như chế độ nội dự đoán theo đường chéo bên trái hướng xuống dưới.

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

[Bảng 2]

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

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

Khi xác định được là tập cụ thể được sử dụng cho việc biến đổi không phân tách được, thì một trong các nhân biến đổi *k* trong tập cụ thể có thể được chọn thông qua chỉ số biến đổi thứ cấp không thể phân tách được. Thiết bị mã hoá có thể dẫn xuất chỉ số biến đổi thứ cấp không phân tách được mà chỉ thị nhân biến đổi cụ thể dựa trên việc kiểm tra sự méo-tốc độ (Rate-Distortion, RD), và có thể báo hiệu chỉ số biến đổi thứ cấp không phân tách được này cho thiết bị giải mã. Thiết bị giải mã có thể chọn một trong số các nhân biến đổi *k* trong tập cụ thể này dựa trên chỉ số biến đổi thứ cấp không phân tách được này. Ví dụ, giá trị chỉ số *lnst* 0 có thể đề cập đến nhân biến đổi thứ cấp không phân tách được thứ nhất, giá trị chỉ số *lnst* 1 có thể đề cập đến nhân biến đổi thứ cấp không phân tách được thứ hai, và giá trị chỉ số *lnst* 2 có thể đề cập đến nhân biến đổi thứ cấp không phân tách được thứ ba. Theo cách khác, giá trị chỉ số *lnst* 0 có thể thể hiện phép biến đổi thứ cấp không phân tách được thứ nhất là

không được áp dụng cho khối mục tiêu, và các giá trị chỉ số lnst từ 1 đến 3 có thể thể hiện ba nhân biến đổi này.

Bộ biến đổi có thể thực hiện phép biến đổi thứ cấp không phân tách được dựa trên các nhân biến đổi được chọn, và có thể thu thập các hệ số biến đổi (thứ cấp) được điều chỉnh. Như đã mô tả trên đây, các hệ số biến đổi được điều chỉnh có thể được dẫn xuất dưới dạng các hệ số biến đổi mà được lượng tử hoá thông qua bộ lượng tử hoá, và có thể được mã hoá và được báo hiệu cho thiết bị giải mã và được chuyển đến bộ giải lượng tử/bộ biến đổi ngược trong thiết bị mã hoá.

Trong khi đó, nếu phép biến đổi thứ cấp được lược bỏ, thì các hệ số biến đổi (sơ cấp), mà là xuất ra của phép biến đổi (phân tách được) sơ cấp, có thể được dẫn xuất dưới dạng các hệ số biến đổi mà được lượng tử hoá thông qua bộ lượng tử hoá như đã mô tả trên đây, và có thể được mã hoá và được báo hiệu cho thiết bị giải mã và được chuyển đến bộ giải lượng tử/bộ biến đổi ngược trong thiết bị mã hoá.

Bộ biến đổi ngược có thể thực hiện một loạt thủ tục theo thứ tự ngược lại so với thứ tự mà các thủ tục đã được thực hiện ở bộ biến đổi nêu trên. Bộ biến đổi ngược có thể nhận các hệ số biến đổi (được giải lượng tử), và dẫn xuất các hệ số biến đổi (sơ cấp) bằng cách thực hiện phép biến đổi (ngược) thứ cấp (S450), và có thể thu được khối phần dư (các mẫu phần dư) bằng cách thực hiện phép biến đổi (ngược) sơ cấp trên các hệ số biến đổi (sơ cấp) (S460). Về mặt này, các hệ số biến đổi sơ cấp này có thể được gọi là các hệ số biến đổi được điều chỉnh từ góc độ của bộ biến đổi ngược. Như đã mô tả trên đây, thiết bị mã hoá và thiết bị giải mã có thể tạo ra khối được tái dựng dựa trên khối phần dư và khối được dự đoán, và có thể tạo ra hình ảnh được tái dựng dựa trên khối được tái dựng này.

Thiết bị giải mã có thể còn bao gồm bộ xác định áp dụng phép biến đổi ngược thứ cấp (hoặc phần tử để xác định xem liệu có áp dụng phép biến đổi ngược thứ cấp hay không) và bộ xác định phép biến đổi ngược thứ cấp (hoặc phần tử để xác định

phép biến đổi ngược thứ cấp). Bộ xác định áp dụng phép biến đổi ngược thứ cấp có thể xác định xem liệu có áp dụng phép biến đổi ngược thứ cấp hay không. Ví dụ, phép biến đổi ngược thứ cấp có thể là NSST, RST hoặc LFNST và bộ xác định áp dụng việc biến đổi ngược thứ cấp có thể xác định xem liệu có áp dụng việc biến đổi ngược thứ cấp dựa trên cờ biến đổi thứ cấp thu được bằng cách phân tích cú pháp luồng bit hay không. Ở ví dụ khác, bộ xác định áp dụng phép biến đổi ngược thứ cấp có thể xác định xem liệu có áp dụng phép biến đổi ngược thứ cấp dựa trên hệ số biến đổi của khối phần dư hay không.

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

Trong khi đó, như đã mô tả trên đây, nếu phép biến đổi (ngược) thứ cấp được lược bỏ, thì các hệ số biến đổi (được giải lượng tử) là có thể được thu, phép biến đổi ngược (phân tách được) sơ cấp có thể được thực hiện, và khối phần dư (các mẫu phần dư) là có thể thu được. Như đã mô tả trên đây, thiết bị mã hoá và thiết bị giải mã có thể tạo ra khối được tái dựng dựa trên khối phần dư và khối được dự đoán, và có thể tạo ra hình ảnh được tái dựng dựa trên khối được tái dựng này.

Trong khi đó, trong phần bộc lộ này, phép biến đổi thứ cấp giảm (reduced secondary transform, RST) trong đó kích thước của ma trận (nhân) biến đổi được

giảm có thể được áp dụng theo khái niệm của NSST để giảm lượng phép tính và bộ nhớ được đòi hỏi cho phép biến đổi thứ cấp không phân tách được.

Trong khi đó, nhân biến đổi, ma trận biến đổi, và hệ số cấu thành ma trận nhân biến đổi, tức là, hệ số nhân hoặc hệ số ma trận, được mô tả trong sáng chế này có thể được biểu diễn thành 8 bit. Điều này có thể là điều kiện để triển khai trong thiết bị giải mã và thiết bị mã hóa, và có thể giảm lượng bộ nhớ được đòi hỏi để lưu trữ nhân biến đổi có sự giảm về hiệu quả hoạt động mà có thể được bao gồm hợp lý so với 9 bit hoặc 10 bit hiện có. Ngoài ra, việc biểu diễn ma trận nhân thành 8 bit có thể cho phép số nhân nhỏ được sử dụng, và có thể là thích hợp hơn cho các lệnh dạng đơn lệnh đa dữ liệu (Single instruction multiple data, SIMD) được sử dụng cho việc triển khai phần mềm tối ưu.

Trong bản mô tả này, thuật ngữ “RST” có thể có nghĩa là phép biến đổi mà được thực hiện trên các mẫu phần dư cho khối mục tiêu dựa trên ma trận biến đổi mà có kích thước được giảm theo nhân tố giảm. Trong trường hợp thực hiện phép biến đổi giảm, lượng phép tính toán được đòi hỏi cho phép biến đổi có thể được giảm theo sự giảm về kích thước của ma trận biến đổi. Tức là, RST này có thể được sử dụng để giải quyết vấn đề về sự phức tạp tính toán xuất hiện ở phép biến đổi không phân tách được hoặc phép biến đổi khối có kích thước lớn.

RST có thể được đề cập đến như các thuật ngữ khác nhau, như phép biến đổi giảm, phép biến đổi thứ cấp giảm, phép biến đổi giảm, phép biến đổi được giảm, phép biến đổi đơn giản, và tương tự, và tên mà RST có thể được đề cập đến là không bị giới hạn ở các ví dụ được liệt kê. Thay vào đó, bởi vì RST được thực hiện chính trong vùng tần số thấp bao gồm hệ số khác không trong khối biến đổi, nên nó có thể được đề cập đến như phép biến đổi không phân tách được tần số thấp (Low-Frequency Non-Separable Transform, LFNST). Chỉ số biến đổi có thể được đề cập đến như chỉ số LFNST.

Trong khi đó, khi phép biến đổi ngược thứ cấp được thực hiện 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 bao gồm bộ biến đổi thứ cấp giảm ngược mà dẫn xuất các hệ số biến đổi được điều chỉnh dựa trên RST ngược của các hệ số biến đổi, và bộ biến đổi sơ cấp ngược mà dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên phép biến đổi sơ cấp ngược của các hệ số biến đổi được điều chỉnh. Phép biến đổi sơ cấp ngược đề cập tới việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Trong sáng chế này, việc dẫn xuất hệ số biến đổi dựa trên phép biến đổi có thể đề cập đến việc dẫn xuất hệ số biến đổi bằng cách áp dụng phép biến đổi.

Fig.6 là sơ đồ minh họa RST theo phương án của sáng chế.

Trong sáng chế này, “khối mục tiêu” có thể đề cập đến khối hiện tại cần được lập mã, khối phần dư, hoặc khối biến đổi.

Trong RST theo một ví dụ, vector có hướng N có thể được ánh xạ đến vector có hướng R được đặt ở không gian khác, sao cho ma trận biến đổi giảm có thể được xác định, trong đó R là nhỏ hơn N. N có thể nghĩa là bình phương chiều dài của cạnh của khối mà phép biến đổi được áp dụng với, hoặc tổng số các hệ số biến đổi tương ứng với khối mà phép biến đổi được áp dụng với, và nhân tố giảm có thể nghĩa là giá trị  $R/N$ . Nhân tố giảm có thể được đề cập đến như nhân tố được giảm (reduced factor), nhân tố giảm (reduction factor), nhân tố rút gọn (simplified factor), nhân tố đơn giản (simple factor), và các thuật ngữ khác nhau. Trong khi đó, R có thể được đề cập đến như hệ số giảm, nhưng theo các tình huống, nhân tố giảm có thể nghĩa là R. Thêm nữa, theo các tình huống, nhân tố giảm có thể nghĩa là giá trị  $N/R$ .

Trong ví dụ, nhân tố giảm hoặc hệ số giảm có thể được báo hiệu qua luồng bit, nhưng ví dụ không được giới hạn ở điều này. Ví dụ, giá trị được xác định trước cho nhân tố giảm hoặc hệ số giảm có thể được lưu trữ trong mỗi thiết bị mã hóa 200

và thiết bị giải mã 300, và trong trường hợp này, nhân tố giảm hoặc hệ số giảm có thể không được báo hiệu riêng biệt.

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

[Phương trình 4]

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

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

Ở một ví dụ, nếu kích thước của khối mà phép biến đổi được áp dụng với là  $8 \times 8$  và  $R=16$  (tức là,  $R/N = 16/64 = 1/4$ ), thì sau đó RST theo Fig.6(a) có thể được biểu diễn như phép toán ma trận như được thể hiện trong Phương trình 5 dưới đây. Trong trường hợp này, việc tính toán bộ nhớ và phép nhân có thể được giảm xuống xấp xỉ  $1/4$  bằng nhân tố giảm.

Trong sáng chế này, phép toán ma trận có thể được hiểu như phép toán nhân vector cột với ma trận, được bố trí trên bên trái của vector cột, để thu được vector cột.

[Phương trình 5]

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

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

[Phương trình 6]

Đối với  $i$  từ đến  $R$ :

$$c_i = 0$$

Đối với  $j$  từ 1 đến  $N$

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

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

Khi được xem xét từ góc độ về kích thước của ma trận biến đổi, kích thước của ma trận biến đổi thông thường là  $64 \times 64$  ( $N \times N$ ), nhưng kích thước của ma trận biến đổi giảm được giảm xuống  $16 \times 64$  ( $R \times N$ ), vì vậy việc sử dụng bộ nhớ trong trường hợp thực hiện RST có thể được giảm bằng tỉ số  $R/N$  khi được so sánh với trường hợp thực hiện phép biến đổi thông thường. Ngoài ra, khi được so sánh với số

lượng phép tính nhân  $N \times N$  trong trường hợp sử dụng ma trận biến đổi thông thường, thì việc sử dụng ma trận biến đổi giảm có thể giảm số lượng các phép tính nhân bằng tỉ số  $R/N$  ( $R \times N$ ).

Ở một ví dụ, bộ biến đổi 232 của thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi cho khối mục tiêu bằng cách thực hiện phép biến đổi sơ cấp và phép biến đổi thứ cấp dựa trên RST trên các mẫu phần dư cho khối mục tiêu. Các hệ số biến đổi này có thể được chuyển đến bộ biến đổi ngược của thiết bị giải mã 300, và bộ biến đổi ngược 322 của thiết bị giải mã 300 có thể dẫn xuất các hệ số biến đổi được điều chỉnh dựa trên phép biến đổi thứ cấp rút gọn (RST) ngược cho các hệ số biến đổi, và có thể dẫn xuất các mẫu dư cho khối mục tiêu dựa trên phép biến đổi sơ cấp ngược cho các hệ số biến đổi được điều chỉnh.

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

Ma trận  $Tt$  trong Reduced Inv. Khối biến đổi được thể hiện ở Fig.6(b) có thể nghĩa là ma trận RST ngược  $TR \times NT$  (chỉ số trên  $T$  nghĩa là sự chuyển vị). Ma trận RST ngược  $TR \times NT$  được nhân với các hệ số biến đổi cho khối mục tiêu như được thể hiện trên Fig.6(b), các hệ số biến đổi được điều chỉnh cho khối mục tiêu hoặc các mẫu phần dư cho khối hiện tại có thể được dẫn xuất. Ma trận RST ngược  $TR \times NT$  có thể được biểu diễn như là  $(TR \times NT)N \times R$ .

Cụ thể hơn, khi RST ngược được áp dụng như là việc biến đổi ngược thứ cấp, thì các hệ số biến đổi đã được biến đổi cho khối mục tiêu có thể được dẫn xuất khi ma trận RST ngược  $TR \times NT$  được nhân cho các hệ số biến đổi cho khối mục tiêu. Trong khi đó, RST ngược có thể được áp dụng như phép biến đổi sơ cấp ngược, và trong trường hợp này, các mẫu phần dư cho khối đích có thể được dẫn xuất khi ma trận RST ngược  $TR \times NT$  được nhân với các hệ số biến đổi cho khối mục tiêu.

Ở ví dụ, nếu kích thước của khối mà phép biến đổi ngược được áp dụng với là  $8 \times 8$  và  $R=16$  (tức là,  $R/N = 16/64 = 1/4$ ), thì sau đó RST theo Fig.6(b) có thể được biểu diễn như phép toán ma trận như được thể hiện trong Phương trình 7 dưới đây.

[Phương trình 7]

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

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

[Phương trình 8]

Đối với  $i$  từ 1 đến  $N$

$$r_i = 0$$

Đối với  $j$  từ 1 đến  $R$

$$r_i += t_{ji} * c_j$$

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

$x \times N$  trong trường hợp sử dụng ma trận biến đổi ngược thông thường, thì việc sử dụng ma trận biến đổi ngược giảm có thể giảm số lượng các phép tính nhân bằng tỉ số  $R/N$  ( $N \times R$ ).

Cấu hình tập biến đổi được thể hiện ở Bảng 2 có thể cũng được áp dụng cho RST  $8 \times 8$ . Tức là, RST  $8 \times 8$  có thể được áp dụng theo tập biến đổi trong Bảng 2. Bởi vì một tập biến đổi bao gồm hai hoặc ba phép biến đổi (các nhân) theo chế độ nội dự đoán, thì nó có thể được tạo cấu hình để chọn một trong đến bốn phép biến đổi bao gồm trong trường hợp mà không có phép biến đổi thứ cấp được áp dụng. Ở phép biến đổi trong đó không có phép biến đổi thứ cấp được áp dụng, nó có thể được xem xét để áp dụng ma trận đơn vị. Giả sử rằng các chỉ số 0, 1, 2, và 3 lần lượt được gán cho bốn phép biến đổi (ví dụ, chỉ số 0 có thể được phân bổ cho trường hợp trong đó ma trận đơn vị được áp dụng, tức là, trường hợp trong đó không có phép biến đổi thứ cấp được áp dụng), chỉ số biến đổi hoặc chỉ số lfnst như phần tử cú pháp có thể được báo hiệu cho mỗi khối hệ số biến đổi, nhờ đó chỉ định phép biến đổi cần được áp dụng. Tức là, đối với khối phía trên cùng-bên trái  $8 \times 8$ , thông qua chỉ số biến đổi, có thể chỉ định RST  $8 \times 8$  trong cấu hình RST, hoặc chỉ định lfnst  $8 \times 8$  khi LFNST được áp dụng. lfnst  $8 \times 8$  và RST  $8 \times 8$  tham chiếu đến các biến đổi có thể áp dụng được với vùng  $8 \times 8$  được bao gồm trong khối hệ số biến đổi khi cả W và H của khối mục tiêu cần được biến đổi là bằng hoặc lớn hơn 8, và vùng  $8 \times 8$  có thể là vùng phía trên cùng-bên trái  $8 \times 8$  trong khối hệ số biến đổi. Tương tự, lfnst  $4 \times 4$  và RST  $4 \times 4$  đề cập đến các biến đổi có thể áp dụng được với vùng  $4 \times 4$  được bao gồm trong khối hệ số biến đổi khi cả W và H của khối mục tiêu là bằng hoặc lớn hơn 4, và vùng  $4 \times 4$  có thể là vùng  $4 \times 4$  phía trên cùng-bên trái trong khối hệ số biến đổi.

Theo phương án của sáng chế, đối với việc biến đổi trong quy trình mã hóa, chỉ 48 mảnh dữ liệu có thể được chọn và ma trận nhân biến đổi tối đa là  $16 \times 48$  có thể được áp dụng cho chúng, chứ không phải áp dụng ma trận nhân biến đổi  $16 \times 64$  cho 64 mảnh dữ liệu tạo thành vùng  $8 \times 8$ . Ở đây, “lớn nhất” nghĩa là phương pháp

có giá trị tối đa là 16 trong ma trận nhân biến đổi  $m \times 48$  để tạo ra  $m$  hệ số. Tức là, khi RST được thực hiện bằng cách áp dụng ma trận nhân biến đổi  $m \times 48$  ( $m \leq 16$ ) cho vùng  $8 \times 8$ , thì 48 mảnh dữ liệu được nhập vào và  $m$  hệ số được tạo ra. Khi  $m$  là 16, thì 48 mảnh dữ liệu được nhập vào và 16 hệ số được tạo ra. Tức là, giả sử là 48 mảnh dữ liệu tạo thành vector  $48 \times 1$ , ma trận  $16 \times 48$  và vector  $48 \times 1$  được nhân tuần tự, nhờ đó tạo ra vector  $16 \times 1$ . Ở đây, 48 mảnh dữ liệu tạo thành vùng  $8 \times 8$  có thể được sắp xếp theo cách thích hợp, nhờ đó tạo thành vector  $48 \times 1$ . Ví dụ, vector  $48 \times 1$  có thể được dựng dựa trên 48 mảnh dữ liệu cấu thành vùng ngoại trừ vùng phía dưới cùng-bên phải  $4 \times 4$  trong số các vùng  $8 \times 8$ . Ở đây, khi phép toán ma trận được thực hiện bằng cách áp dụng ma trận nhân biến đổi tối đa  $16 \times 48$ , thì 16 hệ số biến đổi được điều chỉnh được tạo ra, và 16 hệ số biến đổi được điều chỉnh có thể được sắp xếp trong vùng phía trên cùng-bên trái  $4 \times 4$  theo thứ tự quét, và vùng phía trên cùng-bên phải  $4 \times 4$  và vùng phía dưới cùng-bên trái  $4 \times 4$  có thể được điền bằng các số không (0).

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

Tóm lại, trong quy trình biến đổi, khi RST hoặc LFNST được áp dụng cho vùng  $8 \times 8$ , thì phép toán ma trận của 48 hệ số biến đổi trong các vùng phía trên cùng-bên trái, phía trên cùng-bên phải, và phía dưới cùng-bên trái của vùng  $8 \times 8$  ngoại trừ vùng phía dưới cùng-bên phải trong số các hệ số biến đổi trong vùng  $8 \times 8$  và ma trận nhân biến đổi  $16 \times 48$ . Đối với phép toán ma trận, 48 hệ số biến đổi được nhập vào

mảng một chiều. Khi phép toán ma trận được thực hiện, thì 16 hệ số biến đổi được điều chỉnh được dẫn xuất, và các hệ số biến đổi được điều chỉnh có thể được sắp xếp trong vùng phía trên cùng-bên trái của vùng  $8 \times 8$ .

Trái lại, trong quy trình biến đổi ngược, khi RST hoặc LFNST ngược được áp dụng cho vùng  $8 \times 8$ , 16 hệ số biến đổi tương ứng với vùng phía trên-bên trái của vùng  $8 \times 8$  trong số các hệ số biến đổi trong vùng  $8 \times 8$  có thể được nhập vào mảng một chiều theo thứ tự quét và có thể được đưa vào phép toán ma trận với ma trận nhân biến đổi  $48 \times 16$ . Tức là, phép toán ma trận có thể được biểu diễn như (ma trận  $48 \times 16$ ) \* (vector hệ số biến đổi  $16 \times 1$ ) = (vector hệ số biến đổi được điều chỉnh  $48 \times 1$ ). Ở đây, vector  $n \times 1$  có thể được diễn dịch để có cùng nghĩa như ma trận  $n \times 1$  và do đó có thể được biểu diễn như vector cột  $n \times 1$ . Thêm nữa, \* biểu thị phép nhân ma trận. Khi phép toán ma trận được thực hiện, thì 48 hệ số biến đổi được điều chỉnh có thể được dẫn xuất, và 48 hệ số biến đổi được điều chỉnh có thể được sắp xếp trong các vùng phía trên cùng-bên trái, phía trên cùng-bên phải, và phía dưới cùng-bên trái của vùng  $8 \times 8$  ngoại trừ vùng phía dưới cùng-bên phải.

Khi việc biến đổi ngược thứ cấp được 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 bao gồm bộ biến đổi thứ cấp giảm ngược để dẫn xuất các hệ số biến đổi được điều chỉnh dựa trên RST ngược trên các hệ số biến đổi và bộ biến đổi sơ cấp ngược để dẫn xuất các mẫu phần dư cho khối mục tiêu dựa trên việc biến đổi sơ cấp ngược trên các hệ số biến đổi được điều chỉnh. Phép biến đổi sơ cấp ngược đề cập đến việc biến đổi ngược của việc biến đổi sơ cấp được áp dụng cho phần dư. Trong sáng chế này, việc dẫn xuất hệ số biến đổi dựa trên việc biến đổi có thể đề cập đến việc dẫn xuất hệ số biến đổi bằng cách áp dụng việc biến đổi.

Phép biến đổi không phân tách được, LFNST, sẽ được mô tả chi tiết như sau. LFNST có thể bao gồm phép biến đổi xuôi bởi thiết bị mã hóa và phép biến đổi ngược bởi thiết bị giải mã.

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

[Phương trình 9]

$$y = G^T$$

Trong Phương trình 9,  $x$  và  $y$  lần lượt là các đầu vào và các xuất ra của phép biến đổi thứ cấp, và  $G$  là ma trận biểu diễn phép biến đổi thứ cấp, và các vector dựa trên biến đổi bao gồm các vector cột. Trong trường hợp của LFNST ngược, khi chiều của ma trận biến đổi  $G$  được biểu diễn như là [ số hàng x số cột ], trong trường hợp của LFNST tiến, việc chuyển vị của ma trận  $G$  trở thành chiều của  $GT$ .

Đối với LFNST ngược, các kích thước của ma trận  $G$  là [ 48 x 16 ], [ 48 x 8 ], [ 16 x 16 ], [ 16 x 8 ], và ma trận [ 48 x 8 ] và ma trận [ 16 x 8 ] là các ma trận theo phần mà 8 vector dựa trên biến đổi được lấy mẫu từ bên trái lần lượt của ma trận [ 48 x 16 ] và ma trận [ 16 x 16 ].

Mặt khác, cho LFNST xuôi, các kích thước của ma trận  $GT$  là [ 16 x 48 ], [ 8 x 48 ], [ 16 x 16 ], [ 8 x 16 ], và ma trận [ 8 x 48 ] và ma trận [ 8 x 16 ] là các ma trận riêng phần thu được bằng cách lấy mẫu 8 vector cơ sở biến đổi từ phía trên cùng lần lượt của ma trận [ 16 x 48 ] và ma trận [ 16 x 16 ].

Do đó, trong trường hợp của LFNST xuôi, vector [ 48 x 1 ] hoặc vector [ 16 x 1 ] là có thể để làm đầu vào  $x$ , và vector [ 16 x 1 ] hoặc vector [ 8 x 1 ] là có thể để làm đầu ra  $y$ . Trong việc lập mã và giải mã video, đầu ra của việc biến đổi sơ cấp tiến là dữ liệu hai chiều (two-dimensional, 2D), để xây dựng nên vector [ 48 x 1 ] hoặc vector

[  $16 \times 1$  ] làm đầu vào  $x$ , vector một chiều phải được xây dựng bằng cách sắp xếp một cách phù hợp dữ liệu 2D mà là đầu ra của phép biến đổi xuôi.

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

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

Theo một ví dụ, thứ tự sắp xếp khác so với các thứ tự sắp xếp của (a) và (b) của Fig.7 có thể được sắp xếp, và để dẫn xuất cùng một kết quả (vector  $y$ ) như khi các thứ tự sắp xếp của (a) và (b) của Fig.7 được áp dụng, thì các vector cột của ma trận  $G$  có thể được sắp xếp lại theo thứ tự sắp xếp lại. Tức là, có thể sắp xếp lại các vector cột của  $G$  sao cho mỗi thành phần cấu thành vector  $x$  là luôn được nhân với cùng một vector trên cơ sở biến đổi.

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

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

Trong trường hợp của LFNST, các giá trị xuất ra có thể được sắp xếp trong khối 2D theo thứ tự quét định trước. (a) của Fig.8 thể hiện rằng khi  $y$  xuất ra là vector  $[16 \times 1]$ , 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  $[8 \times 1]$ , 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 bằng các không (0). X trên (b) của Fig.8 thể hiện rằng X được điền bằng 0.

Theo ví dụ khác, bởi vì thứ tự trong đó vector đầu ra  $y$  được xử lý trong khi thực hiện việc lượng tử hóa hoặc lập mã phần dư có thể được thiết đặt trước, vector đầu ra  $y$  có thể không được sắp xếp trong khối 2D như được thể hiện trên Fig.8. Tuy nhiên, trong trường hợp của lập mã phần dư có thể được thực hiện trong các đơn vị khối 2D (ví dụ,  $4 \times 4$ ) như nhóm hệ số (Coefficient Group, CG), và trong trường hợp này, dữ liệu được sắp xếp theo 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 vector đầu vào một chiều  $y$  bằng cách sắp xếp dữ liệu hai chiều được xuất ra thông qua quy trình giải lượng tử hóa hoặc tương tự theo thứ tự quét được thiết đặt trước cho việc biến đổi ngược. Vector đầu vào  $y$  có thể được xuất ra như vector đầu ra  $x$  bằng phương trình sau đây.

[Phương trình 10]

$$x = Gy$$

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

Vector đầu ra  $x$  được sắp xếp trong khối hai chiều theo thứ tự được thể hiện trên Fig.7 và được sắp xếp như 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 phép biến đổi sơ cấp ngược.

Theo đó, toàn bộ phép biến đổi thứ cấp ngược là đối ngược của quy trình biến đổi thứ cấp xuôi, và trong trường hợp của phép biến đổi ngược, không giống trong hướng xuôi, phép biến đổi thứ cấp ngược được áp dụng đầu tiên, và sau đó phép biến đổi sơ cấp ngược được áp dụng.

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

Ngoài ra, 8 ma trận có thể được dẫn xuất từ bốn tập biến đổi như được thể hiện trong Bảng 2 nêu trên, và mỗi tập biến đổi có thể bao gồm hai ma trận. Tập biến đổi nào để sử dụng trong số 4 tập biến đổi được xác định theo chế độ nội dự đoán, và cụ thể hơn, tập biến đổi được xác định dựa trên giá trị của chế độ nội dự đoán được mở rộng bằng việc xem xét việc nội dự đoán góc rộng (Wide Angle Intra Prediction, WAIP). Ma trận nào để chọn trong số hai ma trận cấu thành tập biến đổi được chọn được dẫn xuất thông qua việc báo hiệu chỉ số. Cụ thể hơn, 0, 1, và 2 là có thể như giá trị chỉ số được truyền, 0 có thể chỉ báo rằng LFNST không được áp dụng, và 1 và 2 có thể chỉ báo một ma trận bất kỳ trong số hai ma trận biến đổi cấu thành tập biến đổi được 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 phương án của sáng chế.

Giá trị chế độ nội dự đoán chung có thể có các giá trị từ 0 đến 66 và từ 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 ở trên. Các giá trị từ 81 đến 83 thể hiện chế độ mô hình tuyến tính thành phần chéo (Cross Component Linear Model, CCLM), và các giá trị từ -14

đến -1 và các giá trị từ 67 đến 80 thể hiện chế độ nội dự đoán được mở rộng do áp dụng WAIP.

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

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

Bảng 3 dưới đây thể hiện quy trình dẫn xuất nội chế độ được điều chỉnh bằng cách ánh xạ lại chế độ nội dự đoán đến chế độ nội dự đoán góc rộng.

[Bảng 3]

Inputs to this process are:

- a variable `predModeIntra` specifying the intra prediction mode,
  - a variable `nTbW` specifying the transform block width,
  - a variable `nTbH` specifying the transform block height,
  - a variable `cIdx` specifying the colour component of the current block.
- Output of this process is the modified intra prediction mode `predModeIntra`.  
The variables `nW` and `nH` are derived as follows:

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

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

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

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

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

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

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

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

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

Trong Bảng 3, giá trị chế độ nội dự đoán được mở rộng được lưu trữ cuối cùng trong biến `predModeIntra`, và `ISP_NO_SPLIT` thể hiện rằng khối CU không được chia thành các phân vùng con bởi các kỹ thuật nội phân vùng con (Intra Sub Partition, ISP) hiện được chấp thuận trong chuẩn VVC, và các giá trị của biến `cIdx` là 0, 1, và 2 lần lượt thể hiện các thành phần độ chói, Cb, và Cr. Hàm `Log2` được thể hiện trong Bảng 3 trả lại giá trị log (loga) có cơ số là 2, và hàm `Abs` trả lại giá trị tuyệt đối.

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

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

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

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

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

Bảng 2 nêu trên thể hiện tập biến đổi được chọn như thế nào 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 từ 14 đến 33 và các chế độ 35 đến 80 là đối xứng so với hướng dự đoán xung quanh chế độ 34. Ví dụ, chế độ 14 và chế độ 54 là đối xứng so với hướng

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

Trong khi đó, giả sử rằng dữ liệu đầu vào LFNST xuôi cho chế độ 54 là đối xứng với dữ liệu đầu vào LFNST xuôi 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 trên (a) của Fig.7 và (b) của Fig.7. Ngoài ra, có thể thấy rằng các kiểu theo thứ tự được thể hiện trên (a) của Fig.7 và (b) của Fig.7 là đối xứng so với hướng (hướng đường chéo) được thể hiện bởi chế độ 34.

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

Fig.10 là sơ đồ minh họa hình dạng khối mà LFNST được áp dụng với. (a) của Fig.10 thể hiện các khối  $4 \times 4$ , (b) thể hiện các khối  $4 \times 8$  và  $8 \times 4$ , (c) thể hiện các khối  $4 \times N$  hoặc  $N \times 4$  trong đó  $N$  là 16 hoặc lớn hơn, (d) thể hiện các khối  $8 \times 8$ , (e) thể hiện các khối  $M \times N$  trong đó  $M \geq 8$ ,  $N \geq 8$ , và  $N > 8$  hoặc  $M > 8$ .

Ở Fig.10, các khối có các đường viền dày chỉ báo các vùng mà LFNST được áp dụng với. Đối với các khối của Fig.10 (a) và (b), LFNST được áp dụng cho vùng phía trên cùng-bên trái  $4 \times 4$ , và đối với khối của Fig.10 (c), LFNST được áp dụng riêng lẻ cho hai vùng phía trên cùng-bên trái  $4 \times 4$  được sắp xếp liên tục. Trên (a), (b), và (c) của Fig.10, bởi vì LFNST được áp dụng trong các đơn vị là các vùng  $4 \times 4$ , nên LFNST này sẽ được đề cập đến như “LFNST  $4 \times 4$ ” ở dưới đây. Như ma trận biến đổi tương ứng, ma trận  $[16 \times 16]$  hoặc  $[16 \times 8]$  có thể được áp dụng dựa trên kích thước ma trận cho G trong các Phương trình 9 và 10.

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

của Fig.10. Điều này là để điều chỉnh độ phức tạp tính toán đối với trường hợp xấu nhất là 8 phép nhân trên mẫu.

Đối với (d) và (e) của Fig.10, LFNST được áp dụng cho vùng phía trên cùng-bên trái  $8 \times 8$ , và LFNST này ở dưới đây được đề cập đến như “LFNST  $8 \times 8$ ”. Như ma trận biến đổi tương ứng, ma trận  $[48 \times 16]$  hoặc  $[48 \times 8]$  có thể được áp dụng. Trong trường hợp của LFNST xuôi, bởi vì vector  $[48 \times 1]$  (vector  $x$  trong Phương trình 9) được nhập vào như dữ liệu đầu vào, nên tất cả giá trị mẫu của vùng phía trên cùng-bên trái  $8 \times 8$  không được sử dụng như các giá trị đầu vào của LFNST xuôi. Tức là, như có thể được nhìn thấy theo thứ tự bên trái của Fig.7 (a) hoặc thứ tự bên trái của Fig.7 (b), vector  $[48 \times 1]$  có thể được dựng dựa trên các mẫu thuộc về 3 khối  $4 \times 4$  còn lại trong khi để khối phía dưới cùng-bên phải  $4 \times 4$  như chính nó.

Ma trận  $[48 \times 8]$  có thể được áp dụng cho khối  $8 \times 8$  (TU  $8 \times 8$  hoặc CU  $8 \times 8$ ) trên Fig.10 (d), và ma trận  $[48 \times 16]$  có thể được áp dụng với khối  $8 \times 8$  trên Fig.10(e). Điều này cũng là để điều chỉnh độ phức tạp tính toán cho trường hợp xấu nhất là 8 phép tính nhân trên mẫu.

Tùy thuộc vào hình dạng khối, khi LFNST xuôi tương ứng (LFNST  $4 \times 4$  hoặc LFNST  $8 \times 8$ ) được áp dụng, thì 8 hoặc 16 dữ liệu đầu ra (vector  $y$  trong Phương trình 9, vector  $[8 \times 1]$  hoặc  $[16 \times 1]$ ) được tạo ra. Trong LFNST tiến, số dữ liệu đầu ra là bằng với hoặc nhỏ hơn số dữ liệu đầu vào do các đặc tính của ma trận GT.

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

Vùng được đổ bóng tại phía trên cùng-bên trái của khối được thể hiện trên Fig.11 tương ứng với khu vực trong đó dữ liệu đầu ra của LFNST xuôi được đặt, các vị trí được đánh dấu bằng 0 chỉ báo các mẫu được điền bằng các giá trị 0, và khu vực còn lại biểu diễn các vùng mà không được thay đổi bởi LFNST xuôi. Trong khu vực

không được thay đổi bởi LFNST, dữ liệu đầu ra của phép biến đổi sơ cấp xuôi duy trì không đổi.

Như được mô tả ở trên, bởi vì kích thước của ma trận biến đổi được áp dụng thay đổi theo hình dạng của khối, nên số lượng dữ liệu đầu ra cũng thay đổi. Như Fig.11, dữ liệu đầu ra của LFNST xuôi có thể không điền hoàn toàn khối phía trên cùng-bên trái  $4 \times 4$ . Trong trường hợp của (a) và (d) của Fig.11, ma trận  $[16 \times 8]$  và ma trận  $[48 \times 8]$  được áp dụng cho khối được chỉ báo lần lượt bằng đường dày hoặc vùng một phần bên trong khối, và vector  $[8 \times 1]$  như đầu ra của LFNST xuôi được tạo ra. Tức là, theo thứ tự quét được thể hiện trên (b) của Fig.8, chỉ 8 dữ liệu đầu ra có thể được điền như được thể hiện trên (a) và (d) của Fig.11, và 0 có thể được điền trong 8 vị trí còn lại. Trong trường hợp của khối được áp dụng LFNST của Fig.10 (d), như được thể hiện trên Fig.11(d), hai khối  $4 \times 4$  ở phía trên cùng-bên trái và phía dưới cùng-bên trái gần kề với khối phía trên cùng-bên trái  $4 \times 4$  cũng được điền bằng các giá trị 0.

Như được mô tả ở trên, về cơ bản, bằng cách báo hiệu chỉ số LFNST, việc liệu có áp dụng LFNST hay không và ma trận biến đổi cần được áp dụng được đặc tả. Như được thể hiện trên Fig.11, khi LFNST được áp dụng, bởi vì số lượng dữ liệu đầu ra của LFNST xuôi có thể là bằng hoặc thấp hơn số lượng dữ liệu đầu vào, nên vùng được điền bằng giá trị không (0) xuất hiện như sau.

1) Như được thể hiện trên (a) của Fig.11, các mẫu từ thứ 8 và các vị trí sau theo thứ tự quét trong khối phía trên cùng-bên trái  $4 \times 4$ , tức là, các mẫu từ thứ 9 đến thứ 16.

2) Như được thể hiện trên (d) và (e) của Fig.11, khi ma trận  $[16 \times 48]$  hoặc ma trận  $[8 \times 48]$  được áp dụng, thì hai khối  $4 \times 4$  gần kề khối phía trên cùng-bên trái  $4 \times 4$  hoặc các khối thứ hai và thứ ba  $4 \times 4$  theo thứ tự quét.

Do đó, nếu dữ liệu khác không tồn tại bằng cách kiểm tra các khu vực 1) và 2), thì chắc chắn là LFNST không được áp dụng, sao cho việc báo hiệu chỉ số LFNST tương ứng có thể được lược bỏ.

Trong khi đó, đối với LFNST được chấp thuận, các phương pháp rút gọn sau đây có thể được áp dụng.

(i) Theo một ví dụ, số dữ liệu đầu ra cho LFNST xuôi có thể được giới hạn đến lớn nhất là 16.

Trong trường hợp của (c) của Fig.10, LFNST 4x4 có thể được áp dụng lần lượt cho hai vùng 4x4 gần kề với phía trên cùng-bên trái và trong trường hợp này, tối đa là 32 dữ liệu đầu ra LFNST có thể được tạo ra. Khi số lượng dữ liệu đầu ra cho LFNST xuôi được giới hạn đến tối đa là 16, trong trường hợp của các khối 4xN/Nx4 ( $N \geq 16$ ) (TU hoặc CU), thì LFNST 4x4 chỉ được áp dụng cho một vùng 4x4 ở phía trên cùng-bên trái, LFNST có thể được áp dụng chỉ một lần cho tất cả các khối của Fig.10. Thông qua điều này, các phương án thực hiện việc lập mã ảnh có thể được rút gọn.

Fig.12 là hình vẽ thể hiện số lượng dữ liệu đầu ra cho LFNST xuôi được giới hạn đến tối đa là 16 theo một ví dụ. Như Fig.12, khi LFNST được áp dụng cho vùng phía trên cùng-bên trái ngoài cùng 4x4 trong khối 4xN hoặc Nx4 trong đó N là 16 hoặc hơn, thì dữ liệu đầu ra của LFNST xuôi trở thành 16 mảnh.

(ii) Theo một ví dụ, việc đặt bằng không (0) có thể được áp dụng thêm cho vùng mà LFNST không được áp dụng với. Trong tài liệu này, việc đặt bằng không có thể nghĩa là các giá trị điền vào tất cả các vị trí thuộc về vùng cụ thể có giá trị là 0. Tức là, việc đặt bằng không có thể được áp dụng cho vùng mà không được thay đổi do LFNST và duy trì kết quả của phép biến đổi sơ cấp xuôi. Như được mô tả ở trên, bởi vì LFNST được chia thành LFNST 4x4 và LFNST 8x8, thì việc đặt bằng không có thể được chia thành hai loại ((ii)-(A) và (ii)-(B)) như sau.

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

Như được thể hiện trên Fig.13, đối với khối mà LFNST  $4 \times 4$  được áp dụng, tức là, đối với tất cả các khối trên (a), (b) và (c) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng với có thể được điền bằng các không (0).

Nói cách khác, (d) của Fig.13 thể hiện rằng khi giá trị lớn nhất của số lượng dữ liệu đầu ra của LFNST xuôi được giới hạn là 16 như được thể hiện trên Fig.12, việc đặt bằng không được thực hiện trên các khối còn lại mà LFNST  $4 \times 4$  không được áp dụng.

(ii)-(B) Khi LFNST  $8 \times 8$  được áp dụng, thì vùng mà LFNST  $8 \times 8$  không được áp dụng với có thể được đặt bằng không. Fig.14 là sơ đồ minh họa việc đặt bằng không (0) trong khối mà LFNST  $8 \times 8$  được áp dụng với theo một ví dụ.

Như được thể hiện trên Fig.14, đối với khối mà LFNST  $8 \times 8$  được áp dụng với, tức là, đối với tất cả các khối trong (d) và (e) của Fig.11, toàn bộ vùng mà LFNST không được áp dụng với có thể được điền bằng các không (0).

(iii) Do việc đặt bằng không được biểu diễn trong (ii) nêu trên, khu vực được điền bằng các không (0) có thể là không giống nhau khi LFNST được áp dụng. Theo đó, có thể kiểm tra liệu dữ liệu khác không có tồn tại hay không theo việc đặt bằng không được đề xuất trong (ii) trên khu vực rộng hơn so với trường hợp của LFNST của Fig.11.

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

Tất nhiên, thậm chí nếu việc đặt bằng không được đề xuất trong (ii) được áp dụng, thì có thể kiểm tra xem liệu dữ liệu khác không có tồn tại theo cách giống như việc báo hiệu chỉ số LFNST hiện có hay không. Tức là, sau khi kiểm tra xem liệu dữ liệu khác không có tồn tại trong khối được điền bằng các không (0) trên Fig.11 hay không, việc báo hiệu chỉ số LFNST có thể được áp dụng. Trong trường hợp này, thiết bị mã hóa chỉ thực hiện việc đặt bằng không và thiết bị giải mã không cho rằng việc đặt bằng không, tức là, việc kiểm tra chỉ liệu dữ liệu khác không có tồn tại chỉ trong khu vực được đánh dấu rõ ràng như 0 trên Fig.11 hay không, có thể thực hiện việc phân tích cú pháp chỉ số LFNST.

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

#### Phương án

- Giới hạn số dữ liệu đầu ra cho LFNST xuôi đến lớn nhất 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, thì tất cả các khu vực mà LFNST 8x8 không được áp dụng với được thiết đặt bằng không → (ii)-(B)
- Sau khi kiểm tra xem liệu dữ liệu khác không cũng có tồn tại hay không khu vực hiện có được điền với giá trị không và khu vực được điền với các không do các lần thiết đặt bằng không bổ sung ((ii)-(A), (ii)-(B)), thì chỉ số LFNST được báo hiệu chỉ khi dữ liệu khác không không tồn tại → (iii)

Trong trường hợp của phương án, khi LFNST được áp dụng, thì khu vực trong đó dữ liệu xuất ra khác không có thể tồn tại được giới hạn bên trong của khu vực phía

trên cùng-bên trái  $4 \times 4$ . Cụ thể 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 trong đó dữ liệu khác không có thể tồn tại. Trong trường hợp của Fig.13 (b) và (c) và Fig.14 (b), thì vị trí thứ 16 theo thứ tự quét (tức là, vị trí của mép phía dưới cùng-bên phải của khối phía trên cùng-bên trái  $4 \times 4$ ) là vị trí cuối cùng trong đó dữ liệu ngoài 0 có thể tồn tại.

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

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

Thứ nhất, như được mô tả ở trên, lượng tính toán được đòi hỏi để thực hiện toàn bộ quy trình biến đổi được giảm.

Cụ thể, khi (ii)-(B) được áp dụng, thì lượng phép tính cho trường hợp xấu nhất được giảm sao cho quy trình biến đổi có thể được giảm nhẹ. Nói cách khác, nhìn chung, lượng phép tính lớn được đòi hỏi để thực hiện việc biến đổi sơ cấp kích thước lớn. Bằng cách áp dụng (ii)-(B), thì số lượng dữ liệu được dẫn xuất như kết quả thực hiện LFNST xuôi có thể được giảm đến 16 hoặc thấp hơn. Ngoài ra, khi kích thước

của toàn bộ khối (TU hoặc CU) tăng, thì hiệu quả giảm lượng hoạt động vận hành biến đổi được tăng thêm nữa.

Thứ hai, lượng phép tính được đòi hỏi cho toàn bộ quy trình biến đổi có thể được giảm, nhờ đó giảm sự tiêu thụ nguồn được đòi hỏi để thực hiện phép biến đổi.

Thứ ba, độ trễ được liên quan trong quy trình biến đổi được giảm.

Việc biến đổi thứ cấp như LFNST theo lượng tính toán cho việc biến đổi sơ cấp hiện có, do đó tăng tổng thể thời gian trễ liên quan trong khi thực hiện việc biến đổi. Cụ thể, trong trường hợp của việc nội dự đoán, bởi vì dữ liệu được tái dựng của các khối lân cận được sử dụng trong quy trình dự đoán, trong khi mã hóa, sự gia tăng độ trễ do phép biến đổi thứ hai dẫn đến sự gia tăng về độ trễ cho đến khi tái dựng. Điều này có thể dẫn đến sự gia tăng về độ trễ tổng thể của việc mã hóa nội dự đoán.

Tuy nhiên, nếu việc đặt bằng không được gợi ý trong (ii) được áp dụng, thì thời gian trễ để thực hiện phép biến đổi sơ cấp có thể được giảm phần lớn khi LFNST được áp dụng, thời gian trễ cho toàn bộ phép biến đổi được giữ nguyên hoặc giảm, sao cho thiết bị mã hóa có thể được thực hiện đơn giản hơn.

Trong khi đó, trong nội dự đoán thông thường, khối mục tiêu lập mã được coi như là một đơn vị lập mã, và việc lập mã được thực hiện mà không phân vùng từ đó. Tuy nhiên, việc lập mã ISP (các nội phân vùng con) đề cập đến việc thực hiện lập mã nội dự đoán với khối mục tiêu lập mã được phân vùng theo hướng ngang hoặc hướng thẳng. Trong trường hợp này, khối được tái dựng có thể được tạo ra bằng cách thực hiện việc mã hóa/giải mã trong các đơn vị là các khối được phân vùng, và khối được tái dựng có thể được sử dụng như khối tham chiếu của khối được phân vùng tiếp theo. Theo một ví dụ, trong khi lập mã ISP, một khối lập mã có thể được phân vùng thành hai hoặc bốn khối con và được lập mã, và trong ISP, việc nội dự đoán được thực hiện trên một khối con bằng cách tham chiếu đến giá trị điểm ảnh được tái dựng của khối con được đặt gần phía trái hoặc trên cùng của nó. Ở dưới đây, thuật ngữ “lập mã” có

thể được sử dụng như khái niệm bao gồm cả lập mã được thực hiện bởi thiết bị mã hóa và giải mã được thực hiện bởi thiết bị giải mã.

Sau đây, phương án áp dụng LFNST chỉ cho thành phần độ chói trong cây đơn được mô tả.

Bảng cú pháp đơn vị lập mã liên quan đến việc báo hiệu của chỉ số LFNST và chỉ số MTS theo ví dụ được thể hiện như dưới đây.

[Bảng 4]

|                                                                                                                                                                                                                                                                                                                                                                                                                      |                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <code>LfnstDcOnly = 1</code>                                                                                                                                                                                                                                                                                                                                                                                         |                    |
| <code>LfnstZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                            |                    |
| <code>MtsZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                              |                    |
| <code>transform_tree( x0, y0, cbWidth, cbHeight, treeType, chType )</code>                                                                                                                                                                                                                                                                                                                                           |                    |
| <code>lfnstWidth = ( treeType == DUAL_TREE_CHROMA ) ? cbWidth / SubWidthC<br/>: ( ( IntraSubPartitionsSplitType == ISP_VER_SPLIT ) ? cbWidth /<br/>NumIntraSubPartitions : cbWidth )</code>                                                                                                                                                                                                                          |                    |
| <code>lfnstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC<br/>: ( ( IntraSubPartitionsSplitType == ISP_HOR_SPLIT ) ? cbHeight /<br/>NumIntraSubPartitions : cbHeight )</code>                                                                                                                                                                                                                     |                    |
| <code>if( Min( lfnstWidth, lfnstHeight ) &gt;= 4 &amp;&amp; sps_lfnst_enabled_flag == 1<br/>&amp;&amp;<br/>CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &amp;&amp;<br/>transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 &amp;&amp;<br/>( treeType != DUAL_TREE_CHROMA    !intra_mip_flag[ x0 ][ y0 ]   <br/>Min( lfnstWidth, lfnstHeight ) &gt;= 16 ) &amp;&amp;<br/>Max( cbWidth, cbHeight ) &lt;= MaxTbSizeY ) {</code> |                    |
| <code>if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfnstDcOnly ==<br/>0 ) &amp;&amp;<br/>LfnstZeroOutSigCoeffFlag == 1 )</code>                                                                                                                                                                                                                                                                              |                    |
| <code>lfnst_idx</code>                                                                                                                                                                                                                                                                                                                                                                                               | <code>ae(v)</code> |
| <code>}</code>                                                                                                                                                                                                                                                                                                                                                                                                       |                    |

Các ý nghĩa của các biến số chính trong bảng nêu trên là như sau đây.

1. `cbWidth, cbHeight`: Độ rộng và độ cao của khối lập mã hiện tại
2. `log2TbWidth, log2TbHeight`: Các giá trị logarit cơ số 2 của độ rộng và độ cao của khối biến đổi hiện tại. Kích thước của khối biến đổi hiện tại có thể được giảm đến vùng phía trên cùng-bên trái trong đó hệ số khác không có thể tồn tại bằng việc áp dụng việc đặt bằng không.
3. `sps_lfnst_enabled_flag`: Cờ chỉ báo liệu LFNST có được cho phép hay

không. Giá trị của cờ bằng 0 chỉ báo rằng LFNST không được cho phép, và giá trị của cờ bằng 1 chỉ báo rằng LFNST được cho phép. Cờ được xác định trong tập tham số trình tự (SPS).

4. CuPredMode[ chType ][ x0 ][ y0 ]: Chế độ dự đoán tương ứng với biến chType và vị trí (x0, y0). chType có thể có các giá trị là 0 và 1, trong đó 0 biểu diễn thành phần độ chói và 1 biểu diễn thành phần sắc độ. Vị trí của (x0, y0) chỉ ra vị trí trên hình ảnh, và MODE\_INTRA (việc nội dự đoán) và MODE\_INTER (việc liên dự đoán) là có thể với giá trị là CuPredMode[ chType ][ x0 ][ y0 ].

5. IntraSubPartitionsSplitType: Chỉ báo ISP nào được áp dụng cho đơn vị lập mã hiện tại, và ISP\_NO\_SPLIT chỉ báo rằng đơn vị lập mã không được phân chia thành các khối phân vùng.

6. intra\_mip\_flag[ x0 ][ y0 ]: Vị trí của (x0, y0) được mô tả như trên trong 4. intra\_mip\_flag là cờ chỉ báo liệu chế độ nội dự đoán trên cơ sở ma trận (MIP) có được áp dụng hay không. Giá trị của cờ bằng 0 chỉ báo rằng MIP là không thể áp dụng, và giá trị của cờ bằng 1 chỉ báo rằng MIP được áp dụng.

7. cIdx: Giá trị 0 chỉ báo độ chói, và các giá trị là 1 và 2 lần lượt chỉ báo các thành phần sắc độ Cb và Cr.

8. treeType: thể hiện cây đơn, cây kép, hoặc tương tự. (SINGLE\_TREE: Cây kép, DUAL\_TREE\_LUMA: Cây kép cho thành phần độ chói, DUAL\_TREE\_CHROMA: Cây kép cho thành phần sắc độ)

9. lfnst\_idx[ x0 ][ y0 ]: Phần tử cú pháp chỉ số LFNST cần được phân tích cú pháp. Nếu không được phân tích cú pháp, phần tử này được suy ra là có giá trị là 0. Tức là, giá trị mặc định được thiết đặt là 0, mà chỉ báo rằng không có LFNST nào được áp dụng.

Phân mô tả của các phần tử cú pháp nêu trên có thể được áp dụng cho các

phần tử cú pháp được thể hiện trong các bảng dưới đây.

Trong Bảng 4, `transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0` là một điều kiện để xác định xem chỉ số `lfnst` có được báo hiệu đối với thành phần độ chói hay không theo việc liệu việc biến đổi có được bỏ qua hay không.

Theo đó, theo một ví dụ, bảng cú pháp đơn vị lập mã dưới đây được đề xuất để loại bỏ sự phụ thuộc giữa việc báo hiệu của cờ bỏ qua biến đổi cho thành phần độ chói và việc báo hiệu của chỉ số LFNST cho thành phần sắc độ

[Bảng 5]

|                                                                                                                                                                                                                                                                                                                                                                                                          |                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <code>LfnstDcOnly = 1</code>                                                                                                                                                                                                                                                                                                                                                                             |                    |
| <code>LfnstZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                |                    |
| <code>MtsZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                  |                    |
| <code>transform_tree( x0, y0, cbWidth, cbHeight, treeType, chType )</code>                                                                                                                                                                                                                                                                                                                               |                    |
| <code>lfnstWidth = ( treeType == DUAL_TREE_CHROMA ) ? cbWidth / SubWidthC<br/>: ( ( IntraSubPartitionsSplitType == ISP_VER_SPLIT ) ? cbWidth<br/>/<br/>NumIntraSubPartitions : cbWidth )</code>                                                                                                                                                                                                          |                    |
| <code>lfnstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC<br/>: ( ( IntraSubPartitionsSplitType ==<br/>ISP_HOR_SPLIT ) ? cbHeight /<br/>NumIntraSubPartitions : cbHeight )</code>                                                                                                                                                                                                     |                    |
| <code>LfnstTransformNotSkipFlag = ( treeType != DUAL_TREE_CHROMA ?<br/>transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 :<br/>( transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0   <br/>transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 ) )</code>                                                                                                                                                                           |                    |
| <code>if( Min( lfnstWidth, lfnstHeight ) &gt;= 4 &amp;&amp; sps_lfnst_enabled_flag == 1<br/>&amp;&amp;<br/>CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &amp;&amp;<br/>LfnstTransformNotSkipFlag &amp;&amp;<br/>( treeType != DUAL_TREE_CHROMA    ( !intra_mip_flag[ x0 ][ y0 ]   <br/>Min( lfnstWidth, lfnstHeight ) &gt;= 16 ) ) &amp;&amp;<br/>Max( cbWidth, cbHeight ) &lt;= MaxTbSizeY ) {</code> |                    |
| <code>if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfnstDcOnly ==<br/>0 ) &amp;&amp;<br/>LfnstZeroOutSigCoeffFlag == 1 )</code>                                                                                                                                                                                                                                                                  |                    |
| <code>lfnst_idx</code>                                                                                                                                                                                                                                                                                                                                                                                   | <code>ae(v)</code> |
| <code>}</code>                                                                                                                                                                                                                                                                                                                                                                                           |                    |

Trong phương án được thể hiện trong Bảng 5, việc báo hiệu chỉ số LFNST cho thành phần độ chói phụ thuộc chỉ vào cờ bỏ qua biến đổi cho thành phần độ chói cho cả loại cây kép và chế độ phân vùng cây đơn. Trong chế độ cây kép, chỉ số LFNST cho thành phần sắc độ có thể được báo hiệu phụ thuộc chỉ vào cờ bỏ qua biến đổi cho

thành phần sắc độ. Trong chế độ phân vùng cây đơn, LFNST không được áp dụng cho thành phần sắc độ để làm giảm độ trễ cho trường hợp xấu nhất.

Biến số LfnstTransformNotSkipFlag được thể hiện trong Bảng 5 được thiết đặt theo loại cây của khối hiện tại và giá trị cờ bỏ qua biến đổi cho thành phần màu, và chỉ khi giá trị của biến số là 1, chỉ số LFNST có thể được báo hiệu.

Khi loại cây không phải là sắc độ cây kép (`treeType != DUAL_TREE_CHROMA`), tức là, khi loại cây là cây đơn hoặc độ chói cây kép, nếu giá trị cờ bỏ qua biến đổi cho thành phần độ chói là 0, biến số LfnstTransformNotSkipFlag có thể được thiết đặt là 1 (`treeType != DUAL_TREE_CHROMA ? transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 : (transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)`), và khi loại cây là sắc độ cây kép, nếu giá trị cờ bỏ qua biến đổi (`transform_skip_flag[ x0 ][ y0 ][ 1 ]`) cho thành phần sắc độ Cr là 0 hoặc giá trị cờ bỏ qua biến đổi (`transform_skip_flag[ x0 ][ y0 ][ 1 ]`) cho thành phần sắc độ Cr là 0, biến số LfnstTransformNotSkipFlag có thể được thiết đặt là 1 (`treeType != DUAL_TREE_CHROMA ? transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 : (transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 || transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)`)).

Trong sáng chế, toán tử "`x ? y : z`" chỉ ra rằng `x` là `y` nếu `x` là TRUE (ĐÚNG) và `x` là `z` nếu ngược lại (nếu `x` là TRUE, biểu thị giá trị của `y`; trái lại, biểu thị giá trị của `z`).

Văn bản đặc tả cho quy trình biến đổi xem xét đến Bảng 5 là như sau.

[Bảng 6]

## 8.7.4 Transformation process for scaled transform coefficients

### 8.7.4.1 General

Inputs to this process are:

- a luma location (  $xTbY$ ,  $yTbY$  ) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable  $nTbW$  specifying the width of the current transform block,
- a variable  $nTbH$  specifying the height of the current transform block,
- a variable  $cIdx$  specifying the colour component of the current block,
- an  $(nTbW) \times (nTbH)$  array  $d[x][y]$  of scaled transform coefficients with  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ . Output of this process is the  $(nTbW) \times (nTbH)$  array  $res[x][y]$  of residual samples with  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ .

The variable  $applyLfnstFlag$  is derived as follows:

- If  $treeType$  is equal to `SINGLE_TREE` and  $lfnst\_idx$  is not equal to 0 and  $transform\_skip\_flag[xTbY][yTbY][cIdx]$  is equal to 0 and  $cIdx$  is equal to 0 and both  $nTbW$  and  $nTbH$  are greater than or equal to 4,  $applyLfnstFlag$  is set to 1.
- Otherwise, if  $treeType$  is not equal to `SINGLE_TREE` and  $lfnst\_idx$  is not equal to 0 and  $transform\_skip\_flag[xTbY][yTbY][cIdx]$  is equal to 0 and both  $nTbW$  and  $nTbH$  are greater than or equal to 4,  $applyLfnstFlag$  is set to 1.
- Otherwise,  $applyLfnstFlag$  is set to 0.

When  $applyLfnstFlag$  is equal to 1, the following applies:

- The variables  $predModeIntra$ ,  $nLfnstOutSize$ ,  $log2LfnstSize$ ,  $nLfnstSize$ , and  $nonZeroSize$  are derived as follows:

$$predModeIntra = (cIdx == 0) ? IntraPredModeY[xTbY][yTbY] : IntraPredModeC[xTbY][yTbY] \quad (1149)$$

$$nLfnstOutSize = (nTbW \geq 8 \ \&\& \ nTbH \geq 8) ? 48 : 16 \quad (1150)$$

$$log2LfnstSize = (nTbW \geq 8 \ \&\& \ nTbH \geq 8) ? 3 : 2 \quad (1151)$$

$$nLfnstSize = 1 \ll log2LfnstSize \quad (1152)$$

$$nonZeroSize = ((nTbW == 4 \ \&\& \ nTbH == 4) || (nTbW == 8 \ \&\& \ nTbH == 8)) ? 8 : 16 \quad (1153)$$

Biến số  $LfnstZeroOutSigCoeffFlag$  trong Bảng 4 là 0 nếu có hệ số có nghĩa ở vị trí đặt bằng không khi LFNST được áp dụng, và là 1 nếu ngược lại. Biến số  $LfnstZeroOutSigCoeffFlag$  có thể được thiết đặt theo nhiều điều kiện được thể hiện dưới đây trong Bảng 11.

Biến số  $LfnstZeroOutSigCoeffFlag$  chỉ ra xem hệ số đáng kể có tồn tại trong vùng thứ hai ngoài vùng thứ nhất trên cùng bên trái của khối hiện tại hay không. Giá trị của biến được thiết đặt ban đầu là 1, và có thể được thay đổi thành 0 khi hệ số đáng

kể tồn tại trong vùng thứ hai. Chỉ số LFNST có thể được phân tích cú pháp chỉ khi giá trị được thiết đặt ban đầu của biến `LnstZeroOutSigCoeffFlag` được duy trì. Khi xác định và dẫn xuất xem giá trị của biến `LnstZeroOutSigCoeffFlag` có phải là 1 hay không, LFNST có thể được áp dụng cho thành phần độ chói hoặc tất cả thành phần sắc độ của khối hiện tại, và do đó chỉ số màu sắc của khối hiện tại không được xác định.

Theo một ví dụ, biến số `LnstDcOnly` trong Bảng 4 là 1 khi tất cả các hệ số có nghĩa cuối cùng cho các khối biến đổi mà cờ khối được lập mã tương ứng (CBF, mà là 1 khi có ít nhất một hệ số có nghĩa trong khối tương ứng, và là 0 nếu ngược lại) là 1 là ở các vị trí DC (các vị trí trên cùng bên trái), và trái lại là 0. Cụ thể là, vị trí của hệ số có nghĩa cuối cùng được kiểm tra đối với một khối biến đổi độ chói trong độ chói cây kép, và vị trí của hệ số có nghĩa cuối cùng được kiểm tra đối với cả khối biến đổi cho Cb và khối biến đổi cho Cr trong sắc độ cây kép. Trong cây đơn, vị trí của hệ số có nghĩa cuối cùng có thể được kiểm tra đối với các khối biến đổi cho độ chói, Cb, và Cr.

Bảng cú pháp của đơn vị lập mã báo hiệu chỉ số LFNST theo một ví dụ khác là như sau.

[Bảng 7]

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <code>coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {</code>                                                                                                                                                                                                                                                                                                                                                                                                                           |                    |
| <code>.....</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                    |
| <code>    LfnstDcOnly = 1</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                    |
| <code>    LfnstZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                    |
| <code>    MtsZeroOutSigCoeffFlag = 1</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                    |
| <code>    transform_tree( x0, y0, cbWidth, cbHeight, treeType, chType )</code>                                                                                                                                                                                                                                                                                                                                                                                                                                  |                    |
| <code>    lfnstWidth = ( treeType == DUAL_TREE_CHROMA ) ? cbWidth / SubWidthC</code><br><code>                  : ( ( IntraSubPartitionsSplitType == ISP_VER_SPLIT ) ? cbWidth /</code><br><code>                      NumIntraSubPartitions : cbWidth )</code>                                                                                                                                                                                                                                                 |                    |
| <code>    lfnstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC</code><br><code>                  : ( ( IntraSubPartitionsSplitType == ISP_HOR_SPLIT ) ? cbHeight /</code><br><code>                      NumIntraSubPartitions : cbHeight )</code>                                                                                                                                                                                                                                            |                    |
| <code>    LfnstNotSkipFlag = ( treeType == SINGLE_TREE ) ?</code><br><code>    transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 &amp;&amp;</code><br><code>    transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 &amp;&amp; transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0</code><br><code>    : ( treeType == DUAL_TREE_LUMA ) ? transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0</code><br><code>    : ( transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 &amp;&amp;</code><br><code>        transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 )</code> |                    |
| <code>    if( Min( lfnstWidth, lfnstHeight ) &gt;= 4 &amp;&amp; sps_lfnst_enabled_flag == 1 &amp;&amp;</code><br><code>        CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &amp;&amp;</code><br><code>        LfnstNotSkipFlag &amp;&amp;</code><br><code>        ( treeType != DUAL_TREE_CHROMA    ( !intra_mip_flag[ x0 ][ y0 ]   </code><br><code>            Min( lfnstWidth, lfnstHeight ) &gt;= 16 ) ) &amp;&amp;</code><br><code>        Max( cbWidth, cbHeight ) &lt;= MaxTbSizeY ) {</code>         |                    |
| <code>        if( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfnstDcOnly == 0 ) &amp;&amp;</code><br><code>            LfnstZeroOutSigCoeffFlag == 1 )</code>                                                                                                                                                                                                                                                                                                                                               |                    |
| <code>        lfnst_idx</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <code>ae(v)</code> |
| <code>    }</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                    |
| <code>.....</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                    |

Trong Bảng 7, biến số `transform_skip_flag[x0][y0][cIdx]` inn liệu việc bỏ qua biến đổi có được áp dụng cho khối lập mã cho thành phần được chỉ báo bởi `cIdx` hay không. `cIdx` có thể có các giá trị là 0, 1, và 2, trong đó 0 chỉ báo thành phần độ chói, và 1 và 2 lần lượt chỉ báo thành phần Cb và thành phần Cr. Giá trị của `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` bằng 1 chỉ báo rằng việc bỏ qua biến đổi được áp dụng, và 0 chỉ báo rằng việc bỏ qua biến đổi không được áp dụng.

Trong Bảng 7, biến số `LfnstNotSkipFlag` có thể được thiết đặt là 1 chỉ khi việc bỏ qua biến đổi không được áp dụng cho tất cả các thành phần (tất cả các khối lập mã) tạo thành đơn vị lập mã hiện tại, và có thể được thiết đặt là 0 trong các trường hợp khác, và chỉ số LFNST (`lfnst_idx` trong Bảng 7) có thể được báo hiệu chỉ khi `LfnstNotSkipFlag` là 1.

Khi đơn vị lập mã hiện tại được lập mã trong cấu trúc cây đơn (trong đó

treeType là SINGLE\_TREE trong Bảng 7), tất cả các thành phần gồm Y, Cb, và Cr, khi đơn vị lập mã hiện tại được lập mã trong cấu trúc cây riêng biệt cho độ chói (khi treeType là DUAL\_TREE\_LUMA trong Bảng 7), tất cả các thành phần gồm chỉ Y, và khi đơn vị lập mã hiện tại được lập mã trong cấu trúc cây riêng biệt cho sắc độ (khi treeType là DUAL\_TREE\_CHROMA trong Bảng 7), tất cả các thành phần gồm Cb và Cr.

Nói cách khác, khi thậm chí một trong các thành phần bao gồm đơn vị lập mã hiện tại được lập mã bằng việc bỏ qua biến đổi, chỉ số LFNST không được báo hiệu, và giá trị của chỉ số LFNST được suy ra là 0. Nghĩa là, không có LFNST nào được áp dụng.

Trong cấu trúc giới hạn LFNST không được áp dụng khi thậm chí một trong các thành phần (Y, Cb, và Cr) được lập mã bằng việc bỏ qua biến đổi như trong Bảng 7, khi nhiều thành phần (Y, Cb, và Cr) được lập mã liên tiếp như trong cây đơn và thành phần được xác định là được tiến hành việc bỏ qua biến đổi (ví dụ, khi việc bỏ qua biến đổi được xác định cho Cb và Cr) trong suốt quá trình phân tích cú pháp, các hệ số biến đổi tương ứng có thể được tạo cấu hình không được đệm bổ sung đối với các thành phần tương ứng được lập mã bởi việc bỏ qua biến đổi cho đến khi giá trị của chỉ số LFNST được phân tích cú pháp.

Ví dụ, khi một thành phần bất kỳ được phát hiện là được lập mã bởi việc bỏ qua biến đổi, thì đảm bảo rằng không có LFNST nào được áp dụng, và do đó việc lượng tử hóa ngược, việc biến đổi ngược, và tương tự có thể được thực hiện ngay lập tức sau đó.

Thay vì Bảng 7, bảng cú pháp có thể được mô tả ngắn gọn như trong Bảng 8.

[Bảng 8]

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |
| .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |       |
| if( Min( lfstWidth, lfstHeight ) >= 4 && sps_lfst_enabled_flag == 1 &&<br>CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&<br>(treeType == DUAL_TREE_CHROMA    transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0)<br>&&<br>(treeType == DUAL_TREE_LUMA    (transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 &&<br>transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)) &&<br>(treeType == DUAL_TREE_CHROMA    !intra_mip_flag[ x0 ][ y0 ]   <br>Min( lfstWidth, lfstHeight ) >= 16) &&<br>Max( cbWidth, cbHeight ) <= MaxTbSizeY) { |       |
| if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfstDcOnly == 0 ) &&<br>LfstZeroOutSigCoeffFlag == 1 )                                                                                                                                                                                                                                                                                                                                                                                                  |       |
| lfst_idx                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | ae(v) |
| }                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |
| .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |       |

Khi xác định liệu có báo hiệu chỉ số LFNST hay không bằng việc kiểm tra chỉ liệu việc bỏ qua biến đổi có được áp dụng cho thành phần độ chói trong cây đơn hay không, bảng cú pháp cho đơn vị lập mã có thể được tạo cấu hình như dưới đây.

[Bảng 9]

|                                                                                                                                                                                                                                                                                                                   |       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {                                                                                                                                                                                                                                          |       |
| .....                                                                                                                                                                                                                                                                                                             |       |
| LfstDcOnly = 1                                                                                                                                                                                                                                                                                                    |       |
| LfstZeroOutSigCoeffFlag = 1                                                                                                                                                                                                                                                                                       |       |
| MtsZeroOutSigCoeffFlag = 1                                                                                                                                                                                                                                                                                        |       |
| transform_tree( x0, y0, cbWidth, cbHeight, treeType, chType )                                                                                                                                                                                                                                                     |       |
| lfstWidth = ( treeType == DUAL_TREE_CHROMA ) ? cbWidth / SubWidthC<br>: ( ( IntraSubPartitionsSplitType == ISP_VER_SPLIT ) ? cbWidth /<br>NumIntraSubPartitions : cbWidth )                                                                                                                                       |       |
| lfstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC<br>: ( ( IntraSubPartitionsSplitType == ISP_HOR_SPLIT ) ? cbHeight /<br>NumIntraSubPartitions : cbHeight )                                                                                                                                  |       |
| LfstNotSkipFlag = ( treeType == DUAL_TREE_CHROMA ) ?<br>( transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 && transform_skip_flag[ x0 ][ y0 ][ 2 ]<br>== 0 ) : ( transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0 )                                                                                                             |       |
| if( Min( lfstWidth, lfstHeight ) >= 4 && sps_lfst_enabled_flag == 1 &&<br>CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&<br>LfstNotSkipFlag &&<br>( treeType != DUAL_TREE_CHROMA    ( !intra_mip_flag[ x0 ][ y0 ]   <br>Min( lfstWidth, lfstHeight ) >= 16 ) ) &&<br>Max( cbWidth, cbHeight ) <= MaxTbSizeY) { |       |
| if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfstDcOnly == 0 )<br>&&<br>LfstZeroOutSigCoeffFlag == 1 )                                                                                                                                                                                                    |       |
| lfst_idx                                                                                                                                                                                                                                                                                                          | ae(v) |
| }                                                                                                                                                                                                                                                                                                                 |       |
| .....                                                                                                                                                                                                                                                                                                             |       |

Trong Bảng 9, LfnstNotSkipFlag được xác định giống như trong Bảng 7 hoặc Bảng 8 trong cây riêng biệt (nghĩa là, trong cây riêng biệt độ chói, LfnstNotSkipFlag được thiết đặt là 1 khi việc bỏ qua biến đổi không được áp dụng cho thành phần độ chói, và trái lại được thiết đặt là 0, và trong cây riêng biệt sắc độ, LfnstNotSkipFlag được thiết đặt là 1 khi không có việc bỏ qua biến đổi nào được áp dụng cho cả thành phần Cb và thành phần Cr, và trái lại được thiết đặt là 0). Trong cây đơn, LfnstNotSkipFlag được thiết đặt là 1 khi không có việc bỏ qua biến đổi nào được áp dụng chỉ cho thành phần độ chói, và trái lại được thiết đặt là 0. Thay vì Bảng 8, bảng cú pháp được thể hiện trong Bảng 9 có thể được áp dụng.

[Bảng 10]

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |       |
| .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |
| iff( Min( lfnstWidth, lfnstHeight ) >= 4 && sps_lfnst_enabled_flag == 1 &&<br>CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&<br>(treeType == DUAL_TREE_CHROMA    transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0)<br>&&<br>(treeType == DUAL_TREE_LUMA    treeType == SINGLE_TREE   <br>( transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 &&<br>transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 )) &&<br>( treeType == DUAL_TREE_CHROMA    !intra_mip_flag[ x0 ][ y0 ]   <br>Min( lfnstWidth, lfnstHeight ) >= 16 ) &&<br>Max( cbWidth, cbHeight ) <= MaxTbSizeY ) { |       |
| iff( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT    LfnstDcOnly == 0 ) &&<br>LfnstZeroOutSigCoeffFlag == 1 )                                                                                                                                                                                                                                                                                                                                                                                                                                        |       |
| lfnst_idx                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | ae(v) |
| }                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |
| .....                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |

Phương án xác định điều kiện báo hiệu chỉ số LFNST khi áp dụng LFNST chỉ cho thành phần độ chói trong cây đơn được mô tả dưới đây.

Trong các bảng cú pháp đơn vị lập mã của Bảng 4 và Bảng 5, Bảng 7, Bảng 8, Bảng 9, và Bảng 10, biến số LfnstDcOnly và biến số LfnstZeroOutSigCoeffFlag được sử dụng như các điều kiện để báo hiệu chỉ số LFNST. Về cơ bản, cả biến số LfnstDcOnly và biến số LfnstZeroOutSigCoeffFlag được bắt đầu ở 1 như được thể hiện trong Bảng 7, và các giá trị của hai biến có thể được cập nhật là 0 trong bảng cú

pháp cho việc lập mã phần dư như được thể hiện trong Bảng 11. Để tham chiếu, khi thành phần (mà có thể là Y, Cb, hoặc Cr) được lập mã với việc bỏ qua biến đổi, bảng cú pháp khác (transform\_ts\_coding) được nhập thay vì nhập việc lập mã phần dư trong Bảng 11, và do đó biến số LfnstDcOnly và biến số LfnstZeroOutSigCoeffFlag không được cập nhật khi chỉ số LFNST cho thành phần được phân tích cú pháp.

[Bảng 11]

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

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

Trong Bảng 11, lastSubBlock chỉ ra vị trí của khối con (nhóm hệ số (CG)) trong đó hệ số có nghĩa cuối cùng (khác không) được đặt trong thứ tự quét. 0 chỉ báo khối con gồm thành phần DC, và giá trị lớn hơn 0 chỉ ra khối con không gồm thành phần DC.

lastScanPos chỉ báo vị trí của hệ số có nghĩa cuối cùng trong một khối con trong thứ tự quét. Khi một khối con gồm 16 vị trí, các giá trị từ 0 đến 15 là có thể.

LastSignificantCoeffX và LastSignificantCoeffY chỉ báo tọa độ x và tọa độ y của hệ số có nghĩa cuối cùng được đặt trong khối biến đổi. Các tọa độ x bắt đầu từ 0 và tăng từ trái sang phải, và các tọa độ y bắt đầu từ 0 và tăng từ phía trên cùng đến phía dưới cùng. Các giá trị của hai biến số bằng 0 có nghĩa là hệ số có nghĩa cuối

cùng được định vị ở DC.

Về cơ bản, Bảng 11 được áp dụng cho phương án của Bảng 5 để xác định các giá trị của biến số `LfnstDcOnly` và biến số `LfnstZeroOutSigCoeffFlag`. Khi Bảng 11 được áp dụng và khối lập mã được lập mã trong cây đơn, thì việc lập mã phần dư được biểu diễn trong Bảng 11 có thể được nhập cho tất cả các thành phần. Ví dụ, khi tất cả trong số các thành phần `Y`, `Cb`, và `Cr` không được lập mã với việc bỏ qua biến đổi, việc lập mã phần dư có thể được thực hiện trên mọi thành phần.

Do đó, khi Bảng 11 được áp dụng và khối lập mã hiện tại được lập mã bởi cây đơn, nếu hệ số khác không cuối cùng của thậm chí một thành phần được định vị ở vị trí ngoài vị trí DC (vị trí trên cùng bên trái của khối biến đổi), giá trị của biến `LfnstDcOnly` có thể được cập nhật là 0, và khi vị trí của hệ số khác không cuối cùng của thậm chí một thành phần được định vị trong vùng trong đó hệ số biến đổi trong trường hợp trong đó LFNST được áp dụng không thể được định vị (nghĩa là, được định vị trong vùng ngoài các vị trí thứ nhất đến thứ tám theo thứ tự quét hệ số biến đổi xuôi trong khối biến đổi  $4 \times 4$  hoặc khối biến đổi  $8 \times 8$  hoặc được định vị trong vùng ngoài vùng  $4 \times 4$  trên cùng bên trái trong khối biến đổi mà cho nó LFNST có thể áp dụng trong tiêu chuẩn VVC hiện tại), giá trị của biến `LfnstZeroOutSigCoeffFlag` có thể được cập nhật là 0. Như được thể hiện trong bảng cú pháp đơn vị lập mã của Bảng 5, Bảng 7, Bảng 8, Bảng 9, và Bảng 10, chỉ số LFNST có thể được báo hiệu chỉ khi giá trị của biến số `LfnstZeroOutSigCoeffFlag` là 1, và trong chế độ ngoài chế độ ISP, chỉ số LFNST có thể được báo hiệu chỉ khi giá trị của biến số `LfnstDcOnly` là 0.

Tuy nhiên, khi LFNST được áp dụng chỉ cho thành phần độ chói trong cây đơn, thì biến số `LfnstDcOnly` và biến số `LfnstZeroOutSigCoeffFlag` có thể không được cho phép được cập nhật khi việc lập mã phần dư được thực hiện trên các thành phần mà LFNST không được áp dụng với (các thành phần sắc độ `Cb` hoặc `Cr`). Đó là

bởi vì không thích hợp về logic rằng việc sắp xếp hoặc việc bố trí của các hệ số biến đổi cho thành phần mà LFNST không được áp dụng với xác định liệu chỉ số LFNST có được báo hiệu hay không, nghĩa là, liệu LFNST có được áp dụng hay không.

Bảng 12 giới hạn biến số `LfnstDcOnly` và biến số `LfnstZeroOutSigCoeffFlag` để được cập nhật chỉ cho thành phần độ chói trong cây đơn. Trong trường hợp ngoài cây đơn, biến số `LfnstDcOnly` và biến số `LfnstZeroOutSigCoeffFlag` có thể được cập nhật cho tất cả các thành phần như trong Bảng 11.

[Bảng 12]

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

|                                                                                    |  |
|------------------------------------------------------------------------------------|--|
| numSbCoeff = 1 << ( log2SbW + log2SbH )                                            |  |
| lastScanPos = numSbCoeff                                                           |  |
| lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1 |  |
| do {                                                                               |  |
| if( lastScanPos == 0 ) {                                                           |  |
| lastScanPos = numSbCoeff                                                           |  |
| lastSubBlock--                                                                     |  |
| }                                                                                  |  |
| lastScanPos--                                                                      |  |
| xS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ]              |  |
| [ lastSubBlock ][ 0 ]                                                              |  |
| yS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ]              |  |
| [ lastSubBlock ][ 1 ]                                                              |  |
| xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ lastScanPos ][ 0 ]   |  |
| yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ lastScanPos ][ 1 ]   |  |
| } while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )      |  |
| if( treeType != SINGLE_TREE    cIdx == 0 ) {                                       |  |
| if( lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2                     |  |
| && !transform_skip_flag[ x0 ][ y0 ][ cIdx ] && lastScanPos > 0 )                   |  |
| LfnstDcOnly = 0                                                                    |  |
| if( ( lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2 )                  |  |
| ( lastScanPos > 7 && ( log2TbWidth == 2    log2TbWidth == 3 ) &&                   |  |
| log2TbWidth == log2TbHeight )                                                      |  |
| LfnstZeroOutSigCoeffFlag = 0                                                       |  |
| }                                                                                  |  |
| .....                                                                              |  |

Trong việc lập mã phần dư được biểu diễn trong Bảng 12, do treeType được bổ sung như tham số được so sánh với Bảng 11, bảng cú pháp cho đơn vị biến đổi có thể được điều chỉnh như được thể hiện trong Bảng 13.

[Bảng 13]

| transform_unit( x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType ) {                                                                                                                            | Descriptor |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| .....                                                                                                                                                                                                  |            |
| if( tu_cbf_luma[ x0 ][ y0 ] && treeType != DUAL_TREE_CHROMA ) {                                                                                                                                        |            |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 0 ] &&<br>tbWidth <= MaxTsSize && tbHeight <= MaxTsSize &&<br>( IntraSubPartitionsSplit[ x0 ][ y0 ] == ISP_NO_SPLIT ) && !cu_sbt_flag ) |            |
| transform_skip_flag[ x0 ][ y0 ][ 0 ]                                                                                                                                                                   | ae(v)      |
| if( !transform_skip_flag[ x0 ][ y0 ][ 0 ] )                                                                                                                                                            |            |
| residual_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0, treeType )                                                                                                                              |            |
| Else                                                                                                                                                                                                   |            |
| residual_ts_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0 )                                                                                                                                     |            |
| }                                                                                                                                                                                                      |            |
| if( tu_cbf_cb[ xC ][ yC ] && treeType != DUAL_TREE_LUMA ) {                                                                                                                                            |            |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 1 ] &&<br>wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag )                                                                          |            |
| transform_skip_flag[ xC ][ yC ][ 1 ]                                                                                                                                                                   | ae(v)      |
| if( !transform_skip_flag[ xC ][ yC ][ 1 ] )                                                                                                                                                            |            |
| residual_coding( xC, yC, Log2( wC ), Log2( hC ), 1, treeType )                                                                                                                                         |            |
| Else                                                                                                                                                                                                   |            |
| residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 1 )                                                                                                                                                |            |
| }                                                                                                                                                                                                      |            |
| if( tu_cbf_cr[ xC ][ yC ] && treeType != DUAL_TREE_LUMA &&<br>!( tu_cbf_cb[ xC ][ yC ] && tu_joint_cbr_residual_flag[ xC ][ yC ] ) ) {                                                                 |            |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 2 ] &&<br>wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag )                                                                          |            |
| transform_skip_flag[ xC ][ yC ][ 2 ]                                                                                                                                                                   | ae(v)      |
| if( !transform_skip_flag[ xC ][ yC ][ 2 ] )                                                                                                                                                            |            |
| residual_coding( xC, yC, Log2( wC ), Log2( hC ), 2, treeType )                                                                                                                                         |            |
| Else                                                                                                                                                                                                   |            |
| residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 2 )                                                                                                                                                |            |
| }                                                                                                                                                                                                      |            |
| }                                                                                                                                                                                                      |            |

Dựa vào các chi tiết của Bảng 5, một số chi tiết có thể được thay thế với các phương án của Bảng 7 đến Bảng 10, hoặc các chi tiết của Bảng 11 hoặc Bảng 12 có thể được áp dụng. Các cách kết hợp có thể dưới đây có thể được tạo cấu hình dựa trên Bảng 7 đến Bảng 10, Bảng 11, hoặc Bảng 12.

1. Bảng 7 (hoặc Bảng 8) + Bảng 11
2. Bảng 7 (hoặc Bảng 8) + Bảng 12
3. Bảng 9 (hoặc Bảng 10) + Bảng 11
4. Bảng 9 (hoặc Bảng 10) + Bảng 12

Sau đây, phương pháp để áp dụng danh sách định tỷ lệ cho các thành phần sắc độ khi LFNST được áp dụng chỉ cho thành phần độ chói trong cây đơn được mô tả.

Hiện nay, trong WD VVC, phần tử cú pháp được gọi là `scaling_matrix_for_lfnst_disabled_flag` được xác định. Khi `scaling_matrix_for_lfnst_disabled_flag` là 1, thì danh sách định tỷ lệ không được áp dụng khi LFNST được áp dụng, và khi `scaling_matrix_for_lfnst_disabled_flag` là 0, thì danh sách định tỷ lệ có thể được áp dụng khi LFNST được áp dụng.

Ở đây, danh sách định tỷ lệ là ma trận để định rõ trọng số cụ thể (giá trị được gán trọng số) cho mỗi vị trí hệ số biến đổi trong khối biến đổi và cho phép việc khử lượng tử hóa hoặc việc lượng tử hóa bằng việc nhân với trọng số cho mỗi hệ số biến đổi, do đó cho phép việc lượng tử hóa hoặc khử lượng tử hóa vi phân được áp dụng theo tầm quan trọng của hệ số biến đổi.

Trong cây đơn, LFNST có thể được áp dụng chỉ cho thành phần độ chói như trong phương án của Bảng 5, và khi giá trị của `scaling_matrix_for_lfnst_disabled_flag` là 1 và LFNST được áp dụng trong khi khối lập mã được lập mã trong cây đơn, thì danh sách định tỷ lệ không được áp dụng cho thành phần độ chói. Ở đây, danh sách định tỷ lệ có thể được áp dụng cho thành phần sắc độ mà LFNST không được áp dụng với.

Bảng 14 thể hiện ví dụ về quy trình giải lượng tử hóa (quy trình định tỷ lệ) để thực hiện trường hợp nêu trên.

[Bảng 14]

### 8.7.3 Quy trình định tỷ lệ cho các hệ số biến đổi

Các đầu vào cho quy trình này là:

- vị trí độ chói (  $xTbY$ ,  $yTbY$  ) đặc tả mẫu phía trên-bên trái của khối biến đổi độ chói hiện tại liên quan đến mẫu độ chói phía trên-bên trái của hình ảnh hiện tại,
- biến số  $nTbW$  đặc tả chiều rộng khối biến đổi,

- biến số nTbH đặc tả chiều cao khối biến đổi,
- biến số predMode đặc tả chế độ dự đoán của đơn vị lập mã,
- biến số cIdx đặc tả thành phần màu của khối hiện tại.
- biến số treeType đặc tả liệu cây đơn (SINGLE\_TREE) hay cây kép được sử dụng để phân vùng nút cây lập mã và, khi cây kép được sử dụng, thì liệu các thành phần độ chói (DUAL\_TREE\_LUMA) hay sắc độ (DUAL\_TREE\_CHROMA) được xử lý hiện tại.

Đầu ra của quy trình này là mảng  $d[nTbW] \times [nTbH]$  của các hệ số biến đổi được định tỷ lệ với các phần tử  $d[x][y]$ .

Tham số lượng tử hóa qP được dẫn xuất như sau:

- Nếu cIdx là bằng 0, thì áp dụng như sau:

$$qP = Qp'_Y \quad (1129)$$

- Trái lại, nếu TuCResMode[ xTbY ][ yTbY ] là bằng 2, thì áp dụng như sau:

$$qP = Qp'_{CbCr} \quad (1130)$$

- Trái lại, nếu cIdx là bằng 1, thì áp dụng như sau:

$$qP = Qp'_{Cb} \quad (1131)$$

- Trái lại (cIdx là bằng 2), thì áp dụng như sau:

$$qP = Qp'_{Cr} \quad (1132)$$

Tham số lượng tử hóa qP được điều chỉnh và các biến số rectNonTsFlag và bdShift được dẫn xuất như sau:

- Nếu transform\_skip\_flag[ xTbY ][ yTbY ][ cIdx ] là bằng 0, thì áp dụng như sau:

$$qP = qP - (cu\_act\_enabled\_flag[ xTbY ][ yTbY ] ? 5 : 0) \quad (1133)$$

$$rectNonTsFlag = ((Log2(nTbW) + Log2(nTbH)) \& 1) == 1 ? 1 : 0 \quad (1134)$$

$$bdShift = BitDepth + rectNonTsFlag + ((Log2(nTbW) + Log2(nTbH)) / 2) - 5 + pic\_dep\_quant\_enabled\_flag \quad (1135)$$

- Trái lại (transform\_skip\_flag[ xTbY ][ yTbY ][ cIdx ] là bằng 1), thì áp dụng như sau:

$$qP = \text{Max}(QpPrimeTsMin, qP) - (cu\_act\_enabled\_flag[ xTbY ][ yTbY ] ? 5 : 0) \quad (1136)$$

$$rectNonTsFlag = 0 \quad (1137)$$

$$bdShift = 10 \quad (1138)$$

Biến số bdOffset được dẫn xuất như sau:

$$bdOffset = (1 \ll bdShift) \gg 1 \quad (1139)$$

Danh sách levelScale[ ][ ] được đặc tả như levelScale[ j ][ k ] = { { 40, 45, 51, 57, 64, 72 }, { 57, 64, 72, 80, 90, 102 } } với j = 0..1, k = 0..5.

Mảng dz[nTbW]x[nTbH] được thiết đặt bằng mảng TransCoeffLevel[ xTbY ][ yTbY ][ cIdx ][nTbW]x[nTbH].

Đối với việc dẫn xuất các hệ số biến đổi được định tỷ lệ  $d[x][y]$  với  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ , thì áp dụng như sau:

- Hệ số định tỷ lệ trung gian  $m[x][y]$  được dẫn xuất như sau:

- Nếu một hoặc nhiều điều kiện như sau là đúng, thì  $m[x][y]$  được thiết đặt bằng 16:
  - `sps_scaling_list_enabled_flag` là bằng 0.
  - `pic_scaling_list_present_flag` là bằng 0.
  - `transform_skip_flag[xTbY][yTbY][cIdx]` là bằng 1.
  - `scaling_matrix_for_lfnst_disabled_flag` là bằng 1 và `lfnst_idx[xTbY][yTbY]` không là bằng 0 và một trong các điều kiện sau đây là đúng.
    - `treeType` là bằng `SINGLE_TREE` và `cIdx` là bằng 0.
    - `treeType` là bằng `DUAL_TREE_LUMA` hoặc `DUAL_TREE_CHROMA`.
- Trái lại thì áp dụng như sau:
  - Biến số `id` được dẫn xuất dựa trên `predMode`, `cIdx`, `nTbW`, và `nTbH` như được đặc tả trong Bảng 36 và biến số `log2MatrixSize` được dẫn xuất như sau:
 
$$\text{log2MatrixSize} = (\text{id} < 2) ? 1 : (\text{id} < 8) ? 2 : 3 \quad (1140)$$
  - Hệ số định tỷ lệ  $m[x][y]$  được dẫn xuất như sau:
 
$$m[x][y] = \text{ScalingMatrixRec}[id][i][j]$$
 with  $i = (x << \text{log2MatrixSize}) >> \text{Log2}(nTbW)$ ,  
 $j = (y << \text{log2MatrixSize}) >> \text{Log2}(nTbH) \quad (1141)$
  - Nếu `id` là lớn hơn 13 và cả `x` và `y` là bằng 0, thì  $m[0][0]$  còn được điều chỉnh như sau:
 
$$m[0][0] = \text{ScalingMatrixDCRec}[id - 14] \quad (1142)$$

.....

Trong Bảng 14, `treeType` chỉ ra loại cây của đơn vị lập mã mà khối biến đổi được xử lý hiện tại thuộc về đơn vị này, và `SINGLE_TREE`, `DUAL_TREE_LUMA`, và `DUAL_TREE_CHROMA` lần lượt chỉ báo cây đơn, cây riêng biệt cho độ chói, và cây riêng biệt cho sắc độ (sắc độ cây kép).

Trong phương án này, do LFNST có thể được áp dụng chỉ cho thành phần độ chói trong cây đơn, khi giá trị của `scaling_matrix_for_lfnst_disabled_flag` là 1 và LFNST được áp dụng (giá trị `lfnst_idx[xTbY][yTbY]` là lớn hơn 0), danh sách định cỡ không được áp dụng cho thành phần độ chói (khi giá trị `cIdx` là 0).

Tuy nhiên, đối với thành phần sắc độ (khi giá trị `cIdx` là lớn hơn 0), thì việc liệu có áp dụng danh sách định tỷ lệ hay không có thể được xác định bằng việc kiểm tra thêm điều kiện khác (ví dụ, kiểm tra `transform_skip_flag[xTbY][yTbY][cIdx]`).

Trong cây riêng biệt, như trong trường hợp của thành phần độ chói trong cây

đơn, khi giá trị của `scaling_matrix_for_lfnst_disabled_flag` là 1 và LFNST được áp dụng (giá trị `lfnst_idx[xTbY][yTbY]` là lớn hơn 0), danh sách định tỷ lệ không được áp dụng cho các thành phần sắc độ và độ chói.

Ngoài ra, trong cây riêng biệt, như trong trường hợp của thành phần sắc độ trong cây đơn, việc liệu có áp dụng danh sách định tỷ lệ hay không có thể được xác định bằng việc kiểm tra thêm điều kiện khác (ví dụ, kiểm tra `transform_skip_flag[xTbY][yTbY][cIdx]`).

Do đó, khi giá trị của `scaling_matrix_for_lfnst_disabled_flag` là 1 và LFNST có thể được áp dụng chỉ cho thành phần độ chói trong cây đơn, thì danh sách định tỷ lệ có thể không được áp dụng cho thành phần độ chói và có thể được áp dụng cho thành phần sắc độ.

Theo một ví dụ, sự kết hợp của Bảng 14 và các phương án nêu trên (sự kết hợp bằng cách thay thế một số chi tiết với các phương án của Bảng 7 với Bảng 10 hoặc áp dụng các chi tiết của Bảng 11 hoặc Bảng 12 dựa trên các chi tiết của Bảng 5) có thể được áp dụng.

Trong trường hợp này, như trong văn bản đặc tả của "Quy trình biến đổi cho các hệ số biến đổi được định cỡ" trong Bảng 6, LFNST có thể được tạo cấu hình được áp dụng chỉ cho thành phần độ chói trong cây đơn.

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

Fig.15 là sơ đồ tiến trình minh họa hoạt động vận hành của thiết bị giải mã video theo phương án của sáng chế.

Mỗi quy trình được bộc lộ trên Fig.15 được dựa trên một số chi tiết được mô tả tham chiếu đến Fig.4 đến Fig.14. Do đó, phần mô tả về các chi tiết cụ thể chồng lấn với các chi tiết được mô tả tham chiếu đến Fig.3 đến Fig.14 sẽ được lược bỏ hoặc sẽ được đưa ra theo cách giản lược.

Thiết bị giải mã 300 theo một phương án có thể nhận thông tin cờ chỉ ra xem danh sách định tỷ lệ có sẵn có khi LFNST được thực hiện hay không, chỉ số LFNST cho khối hiện tại, và thông tin phần dư từ luồng bit (S1510).

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

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

Chỉ số LFNST là giá trị để định rõ ma trận LFNST khi LFNST được áp dụng

như việc biến đổi không thể phân tách thứ cấp ngược và có thể có giá trị nằm trong khoảng từ 0 đến 2. Ví dụ, giá trị chỉ số LFNST là 0 có thể chỉ ra rằng không có LFNST nào được áp dụng cho khối hiện tại, giá trị chỉ số LFNST là 1 có thể chỉ ra ma trận LFNST thứ nhất, và giá trị chỉ số LFNST là 2 có thể chỉ ra ma trận LFNST thứ hai.

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

Thông tin cờ chỉ ra xem danh sách định cỡ có sẵn có khi LFNST được thực hiện hay không, mà được nhận bởi thiết bị giải mã, có thể được biểu diễn bởi `scaling_matrix_for_lfnst_disabled_flag` hoặc `sps_scaling_matrix_for_lfnst_disabled_flag` và có thể được báo hiệu trong tập tham số trình tự. Giá trị của cờ này bằng 1 chỉ báo rằng danh sách định tỷ lệ không được áp dụng khi LFNST được áp dụng, và giá trị của cờ bằng 0 chỉ báo rằng danh sách định tỷ lệ là có thể áp dụng khi LFNST được áp dụng. Danh sách định tỷ lệ là ma trận để định rõ trọng số cụ thể (giá trị được gán trọng số) cho mỗi vị trí hệ số biến đổi trong khối biến đổi và cho phép việc giải lượng tử hóa hoặc việc lượng tử hóa bằng việc nhân với trọng số cho mỗi hệ số biến đổi, do đó cho phép việc lượng tử hóa hoặc giải lượng tử hóa khác được áp dụng theo tầm quan trọng của hệ số biến đổi.

Thiết bị giải mã 300 có thể xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu LFNST có được áp dụng hay không và loại cây của khối hiện tại để giải lượng tử hóa các hệ số biến đổi cho khối hiện tại (S1520).

Việc liệu danh sách định tỷ lệ có được áp dụng hay không có thể được xác định dựa trên thông tin cờ và giá trị của chỉ số LFNST.

Khi loại cây của khối hiện tại là cây đơn, thì các thành phần màu sắc của khối hiện tại có thể gồm thành phần độ chói, thành phần sắc độ thứ nhất chỉ ra sắc độ Cb, và thành phần sắc độ thứ hai chỉ ra sắc độ Cr, và khi loại cây của khối hiện tại là độ chói cây kép, thì khối hiện tại có thể gồm thành phần độ chói. Khi loại cây của khối

hiện tại là sắc độ cây kép, thì các thành phần màu sắc của khối hiện tại có thể gồm thành phần sắc độ thứ nhất và thành phần sắc độ thứ hai.

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

Theo một ví dụ, khi loại của khối hiện tại là cây đơn, thì LFNST có thể được áp dụng chỉ cho thành phần độ chói, và khi khối hiện tại được lập mã trong cây đơn, giá trị của `scaling_matrix_for_lfnst_disabled_flag` là 1, và LFNST được áp dụng, thì danh sách định cỡ không được áp dụng cho thành phần độ chói. Tuy nhiên, danh sách định tỷ lệ có thể được áp dụng cho thành phần sắc độ mà không có LFNST nào được áp dụng với nó.

Tóm lại, trong trường hợp trong đó thông tin cờ trên danh sách định tỷ lệ chỉ báo rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0 (nghĩa là, LFNST được áp dụng), khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần độ chói, thì danh sách định tỷ lệ có thể không được áp dụng, và khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ, thì danh sách định tỷ lệ có thể được áp dụng.

Theo một ví dụ, trong trường hợp trong đó thông tin cờ trên danh sách định tỷ lệ chỉ báo rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0, khi loại cây của khối hiện tại là sắc độ cây kép, thì LFNST có thể áp dụng được cho khối hiện tại, và do đó danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ.

Theo một ví dụ, trong trường hợp trong đó thông tin cờ trên danh sách định tỷ lệ chỉ báo rằng danh sách định tỷ lệ là không sẵn có và chỉ số LFNST là lớn hơn 0, khi loại cây của khối hiện tại là độ chói cây kép, LFNST có thể áp dụng được cho khối hiện tại, và do đó danh sách định tỷ lệ không được áp dụng cho thành phần độ chói.

Sau đó, thiết bị giải mã dẫn xuất các hệ số biến đổi cho khối hiện tại từ thông tin phân dư dựa trên kết quả xác định (S1530).

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

Thiết bị giải mã có thể dẫn xuất các hệ số biến đổi được điều chỉnh từ các hệ số biến đổi dựa trên chỉ số LFNST và ma trận LFNST cho LFNST, tức là, bằng việc áp dụng LFNST (S1540).

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

Thiết bị giải mã có thể dẫn xuất các biến khác nhau để áp dụng LFNST và có thể xác định xem có áp dụng LFNST hay không dựa trên loại cây và kích thước của khối hiện tại.

Thiết bị giải mã có thể dẫn xuất biến số thứ nhất (biến số `LfnstDcOnly`) chỉ báo liệu hệ số có nghĩa có tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối

hiện tại hay không và biến số thứ hai (biến số `LfnstZeroOutSigCoeffFlag`) chỉ báo liệu hệ số biến đổi có tồn tại trong vùng thứ hai ngoài vùng thứ nhất trên cùng bên trái của khối hiện tại hay không.

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

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

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

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

Tuy nhiên, cho thành phần sắc độ mà chế độ ISP không được áp dụng cho, giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 theo giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cb, `transform_skip_flag[ x0 ][ y0 ][ 1 ]`, và giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cr, `transform_skip_flag[ x0 ][ y0 ][ 2 ]`. Tức là, khi giá trị của `cIdx` trong `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` là 1, thì giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 1 ]` là 0, và khi giá trị của `cIdx` là 2, thì giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 2 ]` là 0. Khi giá trị của biến số `LfnstDcOnly` là 0, thiết bị giải mã có thể phân tích cú pháp chỉ số LFNST, và nếu không, chỉ số LFNST có thể được suy ra là

0 mà không được báo hiệu.

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

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

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

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

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

đổi phân vùng con mà khối hiện tại được phân chia thành.

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

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

Thiết bị giải mã có thể dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên biến đổi ngược sơ cấp cho các hệ số biến đổi được điều chỉnh (S1550).

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

Sau đó, thiết bị giải mã 300 có thể tạo ra các mẫu được tái dựng dựa trên các mẫu phần dư cho khối hiện tại và các mẫu dự đoán cho khối hiện tại.

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

Fig.16 là sơ đồ tiến trình minh họa hoạt động vận hành của thiết bị mã hóa video theo phương án của sáng chế.

Mỗi quy trình được bộc lộ trên Fig.16 được dựa trên một số chi tiết được mô tả tham chiếu đến Fig.4 đến Fig.14. Do đó, phần mô tả về các chi tiết cụ thể chồng

lần với các chi tiết được mô tả tham chiếu đến Fig.2 và Fig.4 đến Fig.14 sẽ được lược bỏ hoặc sẽ được đưa ra theo cách giản lược.

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

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

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

Cùng chế độ nội dự đoán có thể được áp dụng cho các khối biến đổi phân vùng con mà khối hiện tại được phân chia thành, và thiết bị mã hóa có thể dẫn xuất mẫu dự đoán cho mỗi khối biến đổi phân vùng con. Nghĩa là, thiết bị mã hóa thực hiện tuần tự việc nội dự đoán, ví dụ, theo chiều ngang hoặc dọc, hoặc từ trái sang phải hoặc từ phía trên cùng đến phía dưới cùng, theo dạng được phân chia của các khối biến đổi phân vùng con. Đối với khối con ngoài cùng bên trái hoặc trên cùng, điểm ảnh được tái dựng của khối lập mã đã được lập mã được tham chiếu đến như trong chế độ nội dự đoán thông thường. Ngoài ra, đối với mỗi phía của khối biến đổi nội phân vùng con tiếp theo, mà không liên kề với khối biến đổi phân vùng con trước đó, để dẫn xuất các điểm ảnh tham chiếu liền kề với phía này, thì điểm ảnh được tái dựng của khối lập mã liền kề đã được lập mã được tham chiếu đến như trong chế độ nội dự đoán thông thường.

Thiết bị mã hóa 200 có thể dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên các mẫu dự đoán (S1610).

Thiết bị mã hóa 200 có thể dẫn xuất các hệ số biến đổi cho khối hiện tại bằng việc áp dụng ít nhất một trong số LFNST hoặc MTS cho các mẫu phần dư và có thể

sắp xếp các hệ số biến đổi theo thứ tự quét được xác định trước.

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên quy trình biến đổi như phép biến đổi sơ cấp và/hoặc phép biến đổi thứ cấp trên các mẫu phần dư, có thể áp dụng LFNST khi khối hiện tại là loại cây đơn và thành phần độ chói, và có thể không áp dụng LFNST khi khối hiện tại là loại cây đơn và thành phần sắc độ (S1620).

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

Thiết bị mã hóa 200 có thể xác định có thực hiện phép biến đổi thứ cấp hoặc phép biến đổi không thể phân tách, cụ thể là LFNST, trên các hệ số biến đổi cho khối hiện tại hay không và có thể dẫn xuất các hệ số biến đổi được điều chỉnh bằng việc áp dụng LFNST cho các hệ số biến đổi.

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

Thiết bị mã hóa có thể dẫn xuất các biến số khác nhau để áp dụng LFNST và có thể xác định liệu có áp dụng LFNST hay không dựa trên loại cây và kích thước của khối hiện tại.

Thiết bị mã hóa có thể dẫn xuất biến số thứ nhất (biến số `LfnstDcOnly`) chỉ báo liệu hệ số có nghĩa có tồn tại ở vị trí ngoài vị trí của thành phần DC trong khối hiện tại hay không và biến số thứ hai (biến số `LfnstZeroOutSigCoeffFlag`) chỉ báo liệu hệ số biến đổi có tồn tại trong vùng thứ hai ngoài vùng thứ nhất phía trên cùng

bên trái của khối hiện tại hay không.

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

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

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

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

Tuy nhiên, cho thành phần sắc độ mà chế độ ISP không được áp dụng với, giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 theo giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cb, `transform_skip_flag[ x0 ][ y0 ][ 1 ]`, và giá trị của cờ bỏ qua biến đổi cho thành phần sắc độ Cr, `transform_skip_flag[ x0 ][ y0 ][ 2 ]`. Tức là, khi giá trị của `cIdx` trong `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` là 1, thì giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 1 ]` là 0, và khi giá trị của `cIdx` là 2, thì giá trị của biến số `LfnstDcOnly` có thể được thiết đặt là 0 chỉ khi giá trị của `transform_skip_flag[ x0 ][ y0 ][ 2 ]` là 0. Khi giá trị của biến số `LfnstDcOnly` là 0, thì thiết bị mã hóa có thể áp dụng LFNST, và trái lại, thiết bị mã hóa có thể không áp dụng LFNST.

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

tồn tại trong vùng thứ hai.

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

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

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

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

Như được mô tả ở trên, tập biến đổi có thể được xác định theo chế độ nội dự đoán cho khối biến đổi cần được biến đổi, và LFNST có thể được thực hiện dựa trên ma trận nhân biến đổi, nghĩa là, bất kỳ trong số các ma trận LFNST, được gồm trong

tập biến đổi LFNST. Ma trận được áp dụng cho LFNST có thể được gọi là ma trận LFNST hoặc, và được đề cập đến bằng thuật ngữ bất kỳ miễn là ma trận là chuyển vị của ma trận được sử dụng cho LFNST ngược.

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

Thiết bị mã hóa có thể xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu LFNST có được thực hiện trong quy trình biến đổi hay không và loại cây của khối hiện tại (S1630).

Danh sách định tỷ lệ là ma trận để định rõ trọng số cụ thể (giá trị được gán trọng số) cho mỗi vị trí hệ số biến đổi trong khối biến đổi và cho phép việc giải lượng tử hóa hoặc việc lượng tử hóa bằng việc nhân với trọng số cho mỗi hệ số biến đổi, do đó cho phép việc lượng tử hóa hoặc giải lượng tử hóa khác được áp dụng theo tầm quan trọng của hệ số biến đổi.

Theo một ví dụ, khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần độ chói, thiết bị mã hóa có thể không áp dụng danh sách định tỷ lệ, và khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ, thiết bị mã hóa có thể áp dụng danh sách định tỷ lệ.

Khi loại cây của khối hiện tại là cây đơn, thì các thành phần màu sắc của khối hiện tại có thể gồm thành phần độ chói, thành phần sắc độ thứ nhất chỉ ra sắc độ Cb, và thành phần sắc độ thứ hai chỉ ra sắc độ Cr, và khi loại cây của khối hiện tại là độ chói cây kép, thì khối hiện tại có thể gồm thành phần độ chói. Khi loại cây của khối hiện tại là sắc độ cây kép, thì các thành phần màu sắc của khối hiện tại có thể gồm thành phần sắc độ thứ nhất và thành phần sắc độ thứ hai.

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

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

Theo một ví dụ, khi khối hiện tại là cây đơn, thiết bị mã hóa có thể áp dụng LFNST chỉ cho thành phần độ chói, và khi LFNST được áp dụng, thiết bị mã hóa không áp dụng danh sách định cỡ cho thành phần độ chói. Tuy nhiên, thiết bị mã hóa có thể áp dụng danh sách định tỷ lệ cho thành phần sắc độ mà không có LFNST nào được áp dụng cho nó.

Tóm lại, trong trường hợp trong đó chỉ số LFNST là lớn hơn 0 (nghĩa là, LFNST được áp dụng), khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần độ chói, thì thiết bị mã hóa có thể không áp dụng danh sách định tỷ lệ, và khi loại cây của khối hiện tại là cây đơn và khối hiện tại là thành phần sắc độ, thì thiết bị mã hóa có thể áp dụng danh sách định tỷ lệ.

Theo một ví dụ, trong trường hợp trong đó chỉ số LFNST là lớn hơn 0, khi loại cây của khối hiện tại là sắc độ cây kép, thì LFNST có thể áp dụng được cho khối hiện tại, và do đó thiết bị mã hóa không áp dụng danh sách định tỷ lệ.

Theo một ví dụ, trong trường hợp trong đó chỉ số LFNST là lớn hơn 0, khi loại cây của khối hiện tại là độ chói cây kép, thì LFNST có thể áp dụng được cho khối hiện tại, và do đó thiết bị mã hóa không áp dụng danh sách định tỷ lệ.

Thiết bị mã hóa có thể lượng tử hóa các hệ số biến đổi dựa trên việc xác định, nghĩa là, việc liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không (S1640).

Tức là, thiết bị mã hóa có thể lượng tử hóa các hệ số biến đổi cho khối biến đổi mà LFNST không được áp dụng cho nó sử dụng danh sách định tỷ lệ và có thể lượng tử hóa các hệ số biến đổi cho khối biến đổi mà LFNST được áp dụng cho mà

không sử dụng danh sách định tỷ lệ.

Thiết bị mã hóa có thể mã hóa và xuất ra thông tin phần dư và thông tin cờ chỉ báo liệu danh sách định tỷ lệ là sẵn có khi LFNST được thực hiện hay không (S1650).

Thông tin cờ chỉ báo liệu danh sách định tỷ lệ là sẵn có khi LFNST được thực hiện hay không có thể được biểu diễn bởi `scaling_matrix_for_lfnst_disabled_flag` hoặc `sps_scaling_matrix_for_lfnst_disabled_flag` và có thể được báo hiệu trong tập tham số trình tự. Giá trị của cờ này bằng 1 chỉ báo rằng danh sách định tỷ lệ không được áp dụng khi LFNST được áp dụng, và giá trị của cờ bằng 0 chỉ báo rằng danh sách định tỷ lệ là có thể áp dụng khi LFNST được áp dụng.

Khi chỉ số LFNST là lớn hơn 0 và khối hiện tại là cây đơn, LFNST có thể áp dụng được cho thành phần độ chói, và do đó thiết bị mã hóa có thể mã hóa giá trị của cờ thành 1.

Tuy nhiên, khi chỉ số LFNST là lớn hơn 0 và khối hiện tại là cây đơn, LFNST không được áp dụng cho thành phần sắc độ, và do đó thiết bị mã hóa có thể xây dựng thông tin hình ảnh sao cho danh sách định tỷ lệ có thể được áp dụng.

Khi chỉ số LFNST là lớn hơn 0 và loại cây của khối hiện tại là sắc độ cấp kép, thì LFNST có thể áp dụng được cho khối hiện tại, và do đó thiết bị mã hóa có thể mã hóa giá trị của cờ thành 1 sao cho danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ.

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

Thiết bị mã hóa có thể dẫn xuất các hệ số biến đổi được lượng tử hóa bằng

việc lượng tử hóa các hệ số biến đổi được điều chỉnh cho khối hiện tại và có thể mã hóa chỉ số LFNST.

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

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

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

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

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

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

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

Theo các phương án nêu trên, thì các phương pháp được giải thích dựa trên các lưu đồ bằng hàng loạt các bước hoặc các khối, nhưng sáng chế không chỉ giới hạn ở thứ tự đó của các bước, và bước nhất định có thể được thực hiện với thứ tự khác so với thứ tự được mô tả trên đây, hoặc xuất hiện đồng thời với bước khác. Hơn nữa, người có hiểu biết trung bình trong lĩnh vực kỹ thuật có thể hiểu rằng các bước được thể hiện trên lưu đồ là không có tính loại trừ, và rằng bước khác là có thể được tích hợp hoặc một hoặc nhiều bước của lưu đồ là có thể được loại bỏ mà không ảnh hưởng đến phạm vi bảo hộ của sáng chế này.

Các phương pháp nêu trên theo sáng chế là có thể được thực hiện dưới dạng dạng phần mềm, và thiết bị mã hoá và/hoặc thiết bị giải mã theo sáng chế là có thể được bao gồm trong thiết bị xử lý hình ảnh, chẳng hạn TV (TeleVision - ti vi), máy tính, điện thoại thông minh, hộp chuyển đổi tín hiệu cáp TV, thiết bị hiển thị hoặc tương tự.

Khi các phương án theo sáng chế được thực hiện bằng phần mềm, thì các phương pháp nêu trên có thể được thực hiện dưới dạng các mô đun (các tiến trình, các chức năng hoặc tương tự) để thực hiện các chức năng được mô tả trên đây. Các mô đun

có thể được lưu trong bộ nhớ và có thể được thực thi bởi bộ xử lý. Bộ nhớ có thể ở bên trong hoặc bên ngoài bộ xử lý và có thể được kết nối tới bộ xử lý theo nhiều cách đã biết rõ khác nhau. Bộ xử lý này có thể bao gồm mạch tích hợp chuyên dụng (Application-Specific Integrated Circuit, ASIC), con chip khác, mạch logic, và/hoặc thiết bị xử lý dữ liệu. Bộ nhớ này có thể bao gồm bộ nhớ chỉ đọc (Read-Only Memory, ROM), bộ nhớ truy cập ngẫu nhiên (Random Access Memory, RAM), bộ nhớ tốc độ nhanh (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 triển khai và được thực hiện trên bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc con chip. Ví dụ, các đơn vị chức năng được thể hiện trong mỗi hình vẽ có thể được triển khai và được thực hiện trên máy tính, bộ xử lý, bộ vi xử lý, bộ điều khiển hoặc con chip.

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

Thêm vào đó, phương pháp xử lý mà phần bộc lộ này được áp dụng vào đó là có thể được tạo ra dưới dạng chương trình được thực thi bởi máy tính, và được lưu

giữ trên phương tiện ghi đọc được bằng máy tính. Dữ liệu đa phương tiện mà có cấu trúc dữ liệu theo sáng chế thì cũng có thể được lưu trữ trên phương tiện ghi đọc được bằng máy tính. Phương tiện ghi đọc được bằng máy tính này bao gồm tất cả các loại thiết bị lưu trữ và các thiết bị lưu trữ phân tán mà dữ liệu đọc được bằng máy tính được lưu trữ trong đó. Phương tiện ghi đọc được bằng máy tính này có thể bao gồm, ví dụ, đĩa Blu-ray (Blu-ray Disc, BD), thiết bị bus nối tiếp vạn năng (Universal Serial Bus, USB), ROM, PROM, EPROM, EEPROM, RAM, đĩa CD-ROM, băng từ, đĩa mềm, và thiết bị lưu trữ dữ liệu quang học. Thêm nữa, phương tiện ghi đọc được bằng máy tính này bao gồm các phương tiện được thực hiện dưới dạng sóng mang (ví dụ, truyền qua mạng Internet). Ngoài ra, luồng bit được tạo ra bởi phương pháp mã hóa này là có thể được lưu giữ trên phương tiện ghi đọc được bằng máy tính hoặc được truyền qua mạng truyền thông dùng dây hoặc không dây. Ngoài ra, các phương án của sáng chế này có thể được bao gồm như sản phẩm chương trình máy tính bởi các mã chương trình, và các mã chương trình có thể được thực thi trên máy tính bởi các phương án của sáng chế này. Các mã chương trình này có thể được lưu trữ trên bộ mang đọc được bằng máy tính.

Fig.17 là hình vẽ minh họa cấu trúc của hệ thống tạo luồng nội dung mà sáng chế được áp dụng với.

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

Máy chủ mã hóa có chức năng nén thành các dữ liệu kỹ thuật số, các nội dung được nhập vào từ các thiết bị đầu vào đa phương tiện, chẳng hạn điện thoại thông minh, camera, máy ghi hình cầm tay, và dạng tương tự, để tạo ra luồng bit, và truyền nó đến máy chủ tạo luồng. Như là một ví dụ khác, trong trường hợp mà thiết bị đầu vào đa phương tiện, chẳng hạn điện thoại thông minh, camera, máy ghi hình cầm tay,

hoặc dạng tương tự, trực tiếp tạo ra luồng bit, thì máy chủ mã hóa có thể được bỏ qua. Luồng bit này có thể được tạo ra bằng phương pháp mã hoá hoặc phương pháp tạo luồng bit mà sáng chế được áp dụng vào đó. Và máy chủ tạo luồng có thể lưu trữ luồng bit tạm thời trong quá trình truyền hoặc nhận luồng bit.

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

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

Ví dụ, thiết bị người dùng có thể bao gồm điện thoại di động, điện thoại thông minh, máy tính xách tay, thiết bị đầu cuối phát rộng kỹ thuật số, máy hỗ trợ kỹ thuật số cá nhân (personal digital assistant, PDA), máy chơi đa phương tiện mang đi được (portable multimedia player, PMP), bộ điều hướng, PC dạng phẩy, PC dạng bảng, máy tính 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), thiết bị hiển thị được lắp đầu (head mounted display, HMD)), TV kỹ thuật số, máy tính để bàn, bảng chỉ dẫn kỹ thuật số hoặc tương tự. Mỗi trong số các máy chủ trong hệ thống tạo luồng nội

dung này có thể được vận hành dưới dạng máy chủ phân tán, và trong trường hợp này, thì dữ liệu mà mỗi máy chủ nhận được có thể được xử lý theo cách phân tán.

Các yêu cầu bảo hộ được bộc lộ ở đây có thể được kết hợp theo cách khác nhau. Thêm nữa, các dấu hiệu kỹ thuật của các điểm yêu cầu bảo hộ của sáng chế này có thể được kết hợp để được thực hiện ở thiết bị, và các dấu hiệu kỹ thuật của các điểm yêu cầu bảo hộ về thiết bị có thể được kết hợp để được thực hiện ở phương pháp. Thêm nữa, các dấu hiệu kỹ thuật của các điểm yêu cầu bảo hộ về phương pháp của sáng chế này có thể được kết hợp để được thực hiện ở thiết bị, và các dấu hiệu kỹ thuật ở các điểm yêu cầu bảo hộ về thiết bị có thể được kết hợp để được thực hiện ở phương pháp.

## YÊU CẦU BẢO HỘ

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

dẫn xuất các hệ số biến đổi cho khối hiện tại dựa trên thông tin phần dư thu được từ luồng bit; và

dẫn xuất các mẫu phần dư cho khối hiện tại dựa trên phép biến đổi ngược cho các hệ số biến đổi,

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

xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu phép biến đổi không phân tách được tần số thấp (low frequency non-separable transform, LFNST) có được áp dụng hay không và loại cây của khối hiện tại; và

dẫn xuất các hệ số biến đổi cho khối hiện tại từ thông tin phần dư dựa trên việc xác định, và

trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và thành phần màu sắc của khối hiện tại là thành phần sắc độ, danh sách định tỷ lệ được áp dụng cho thành phần sắc độ của khối hiện tại, và

trong đó dựa trên việc loại cây của khối hiện tại là sắc độ cây kép và LFNST được áp dụng cho thành phần sắc độ của khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ của khối hiện tại.

2. Phương pháp giải mã ảnh theo điểm 1, trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và thành phần màu sắc của khối hiện tại là thành phần sắc độ, LFNST không được áp dụng cho thành phần sắc độ của khối hiện tại.

3. Phương pháp giải mã ảnh theo điểm 1, trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và LFNST được áp dụng cho khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần độ chói của khối hiện tại.

4. Phương pháp giải mã ảnh theo điểm 1, ngoài ra còn bao gồm bước:

nhận thông tin cờ chỉ báo liệu danh sách định tỷ lệ là sẵn có hay không cho khối mà cho nó LFNST được áp dụng.

5. Phương pháp giải mã ảnh theo điểm 4, trong đó dựa trên việc thông tin cờ chỉ báo rằng danh sách định tỷ lệ là không sẵn có, chỉ số LFNST là lớn hơn 0, và loại cây của khối hiện tại là cây đơn, danh sách định tỷ lệ không được áp dụng cho thành phần độ chói của khối hiện tại.

6. Phương pháp giải mã ảnh theo điểm 4, trong đó dựa trên việc thông tin cờ chỉ báo rằng danh sách định tỷ lệ là không sẵn có, chỉ số LFNST là lớn hơn 0, và loại cây của khối hiện tại là độ chói cây kép, danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ của khối hiện tại.

7. Phương pháp giải mã ảnh theo điểm 4, trong đó dựa trên việc thông tin cờ chỉ báo rằng danh sách định tỷ lệ là không sẵn có, chỉ số LFNST là lớn hơn 0, và loại cây của khối hiện tại là độ chói cây kép, danh sách định tỷ lệ không được áp dụng cho thành phần độ chói của khối hiện tại.

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

đẫn xuất các hệ số biến đổi cho khối hiện tại từ các mẫu phần dư cho khối hiện tại dựa trên quy trình biến đổi;

xác định liệu danh sách định tỷ lệ có được áp dụng cho khối hiện tại hay không dựa trên việc liệu phép biến đổi không phân tách được tần số thấp (LFNST) có được thực hiện trong quy trình biến đổi hay không và loại cây của khối hiện tại; và

lượng tử hóa các hệ số biến đổi dựa trên việc xác định,

trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và thành phần màu sắc của khối hiện tại là thành phần sắc độ, danh sách định tỷ lệ được áp dụng cho thành phần sắc độ của khối hiện tại, và

trong đó dựa trên việc loại cây của khối hiện tại là sắc độ cây kép và LFNST được thực hiện trên khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ của khối hiện tại.

9. Phương pháp mã hóa ảnh theo điểm 8, trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và thành phần màu sắc của khối hiện tại là thành phần sắc độ, LFNST không được thực hiện trên thành phần sắc độ của khối hiện tại.

10. Phương pháp mã hóa ảnh theo điểm 8, trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và LFNST được thực hiện trên khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần độ chói của khối hiện tại.

11. Phương pháp mã hóa ảnh theo điểm 8, trong đó dựa trên việc loại cây của khối hiện tại là độ chói cây kép và LFNST được thực hiện trên khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần độ chói của khối hiện tại.

12. Phương pháp mã hóa ảnh theo điểm 8, ngoài ra còn bao gồm bước:

mã hóa và xuất ra thông tin cờ chỉ báo liệu danh sách định tỷ lệ là sẵn có hay không cho khối mà cho nó LFNST được áp dụng.

13. Phương pháp mã hóa ảnh theo điểm 8, trong đó khối hiện tại bao gồm khối biến đổi.

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

tạo ra luồng bit cho thông tin ảnh, trong đó luồng bit được tạo ra dựa trên việc dẫn xuất các hệ số biến đổi cho khối hiện tại từ các mẫu phần dư cho khối hiện tại dựa trên quy trình biến đổi, xác định liệu danh sách định tỷ lệ có được áp dụng cho

khối hiện tại hay không dựa trên việc liệu phép biến đổi không phân tách được tần số thấp (LFNST) có được thực hiện trong quy trình biến đổi hay không và loại cây của khối hiện tại, lượng tử hóa các hệ số biến đổi dựa trên việc xác định, và mã hóa thông tin phần dư liên quan đến các hệ số biến đổi được lượng tử hóa; và

truyền dữ liệu bao gồm luồng bit,

trong đó dựa trên việc loại cây của khối hiện tại là cây đơn và thành phần màu sắc của khối hiện tại là thành phần sắc độ, danh sách định tỷ lệ được áp dụng cho thành phần sắc độ của khối hiện tại, và

trong đó dựa trên việc loại cây của khối hiện tại là sắc độ cây kép và LFNST được thực hiện trên khối hiện tại, danh sách định tỷ lệ không được áp dụng cho thành phần sắc độ của khối hiện tại.

FIG. 1

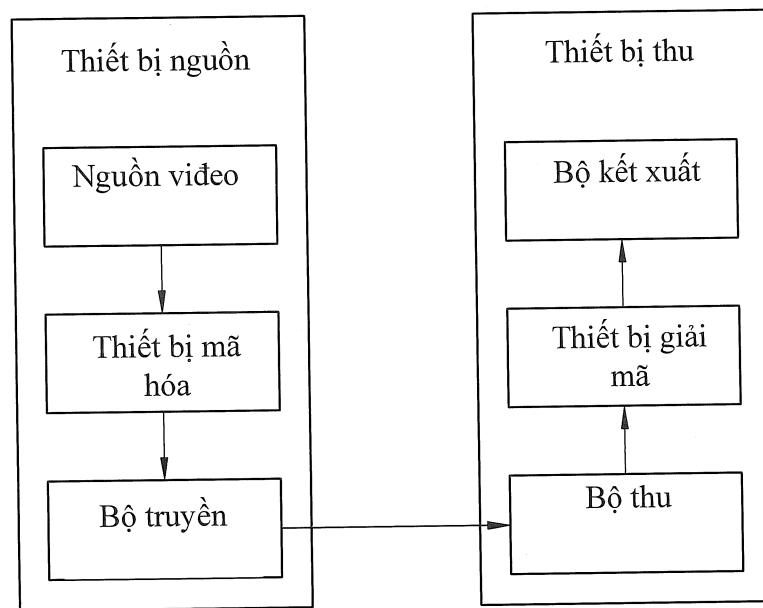


FIG. 2

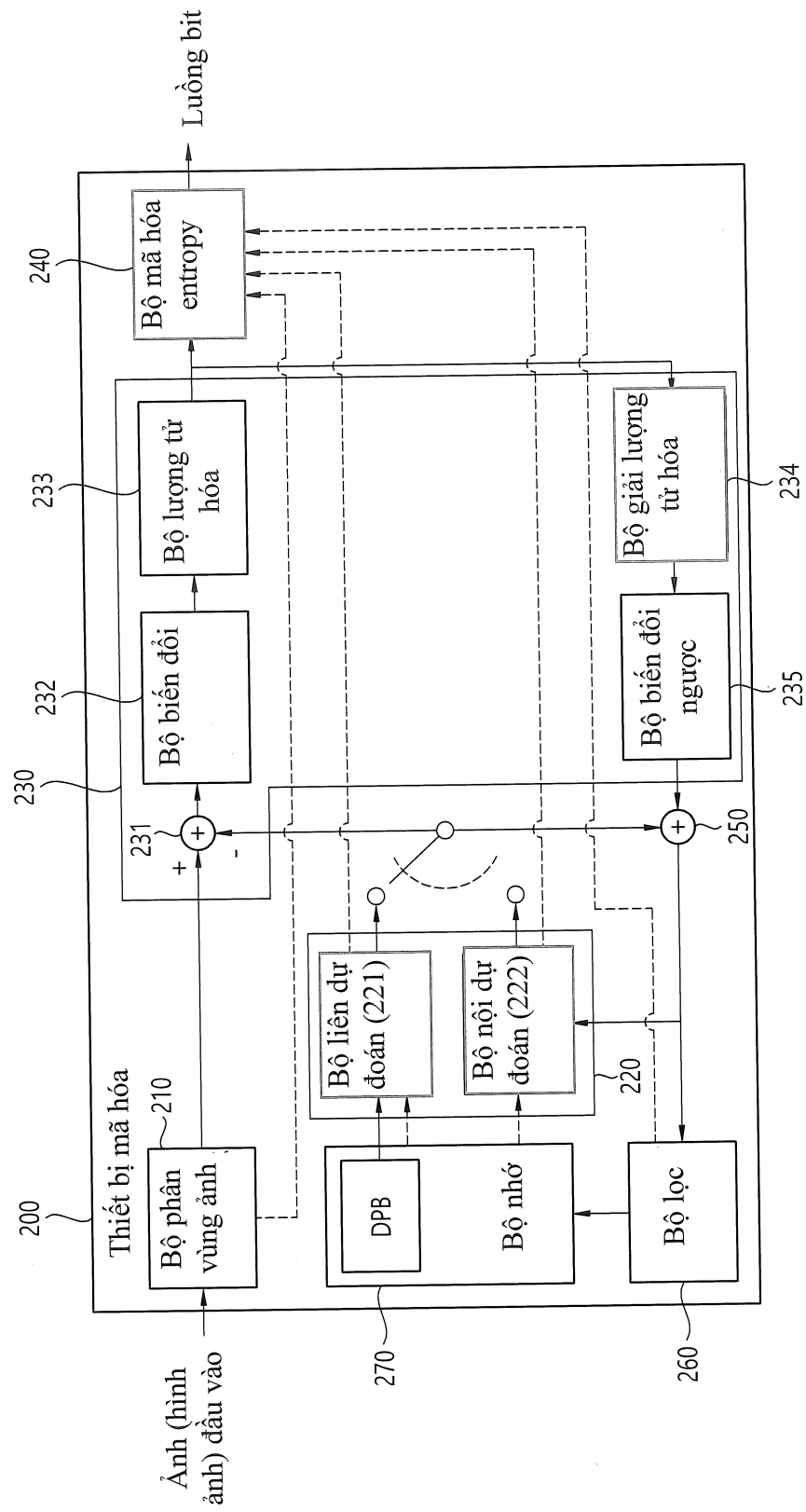


FIG. 3

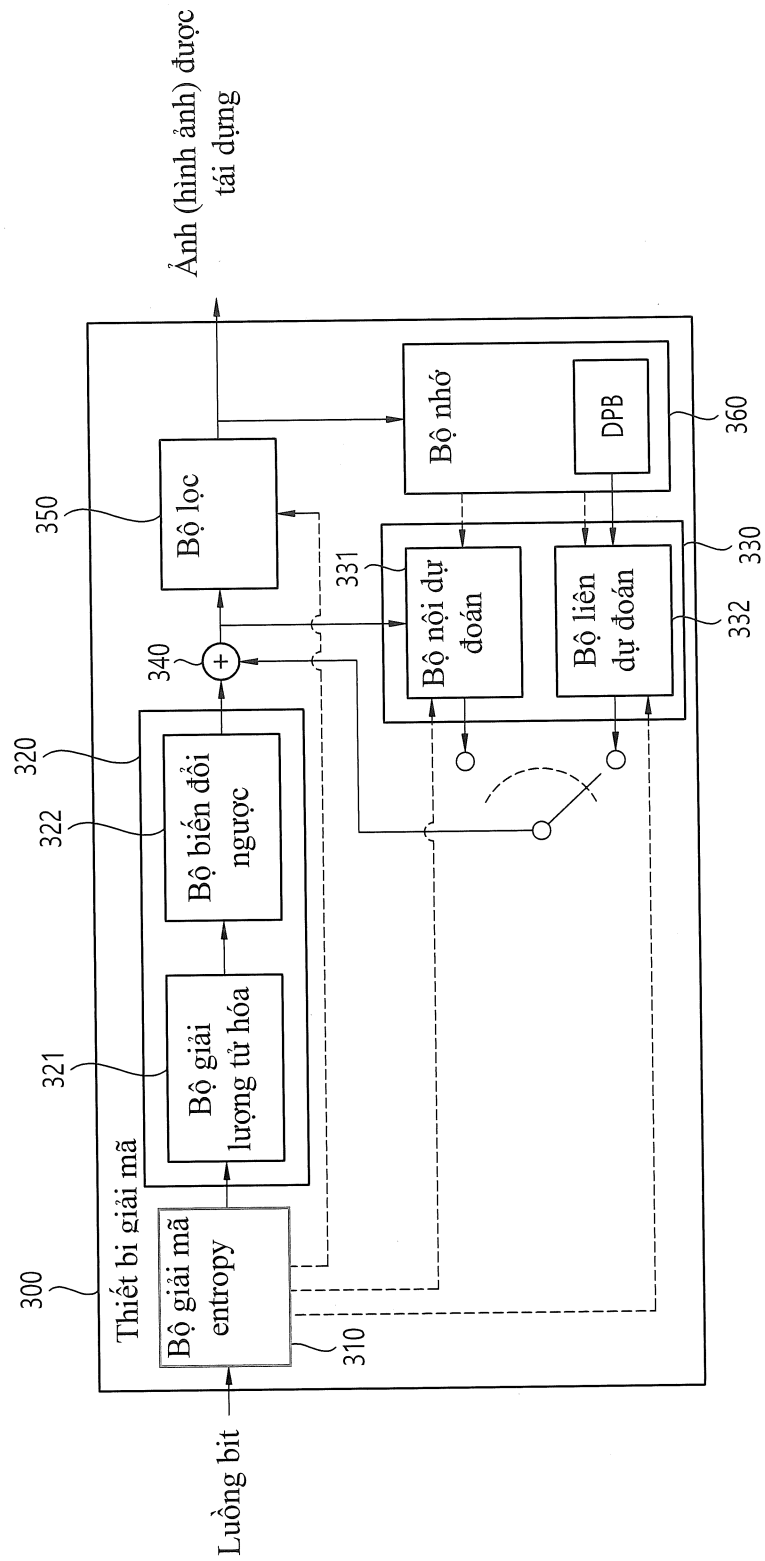


FIG. 4

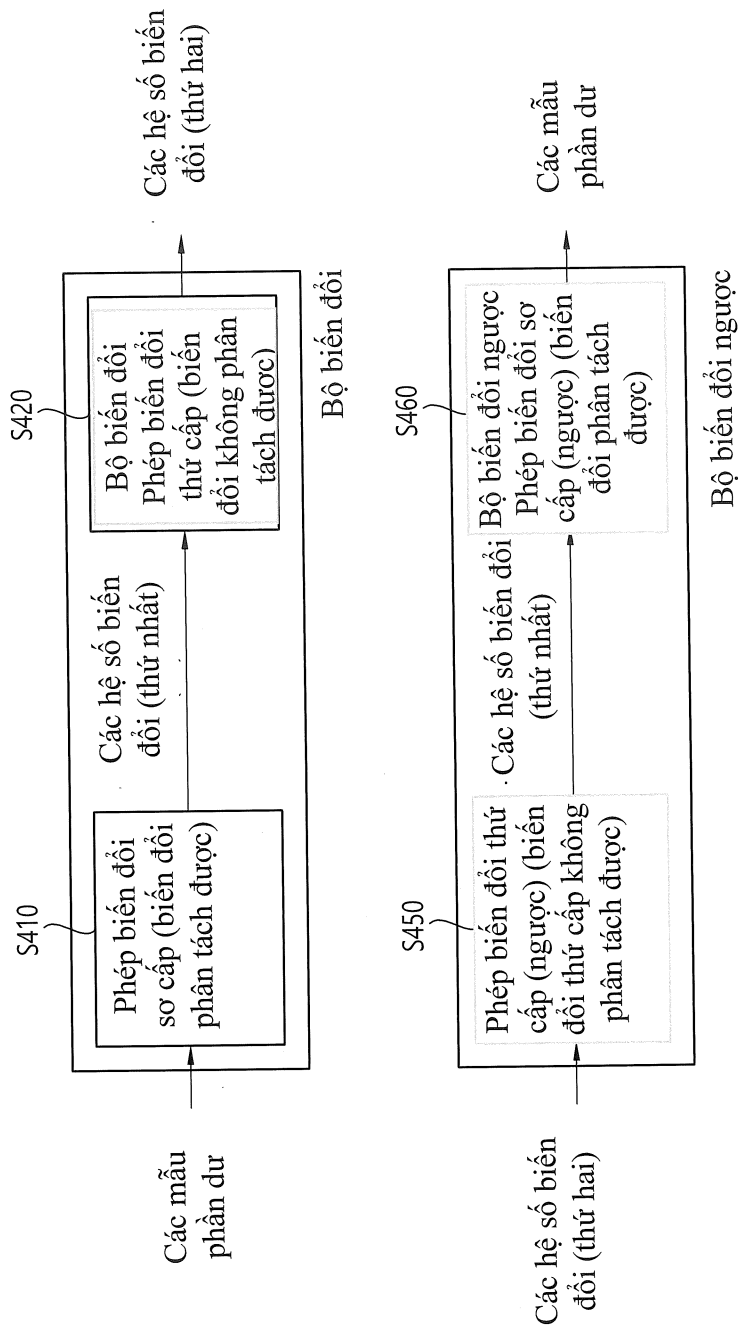


FIG. 5

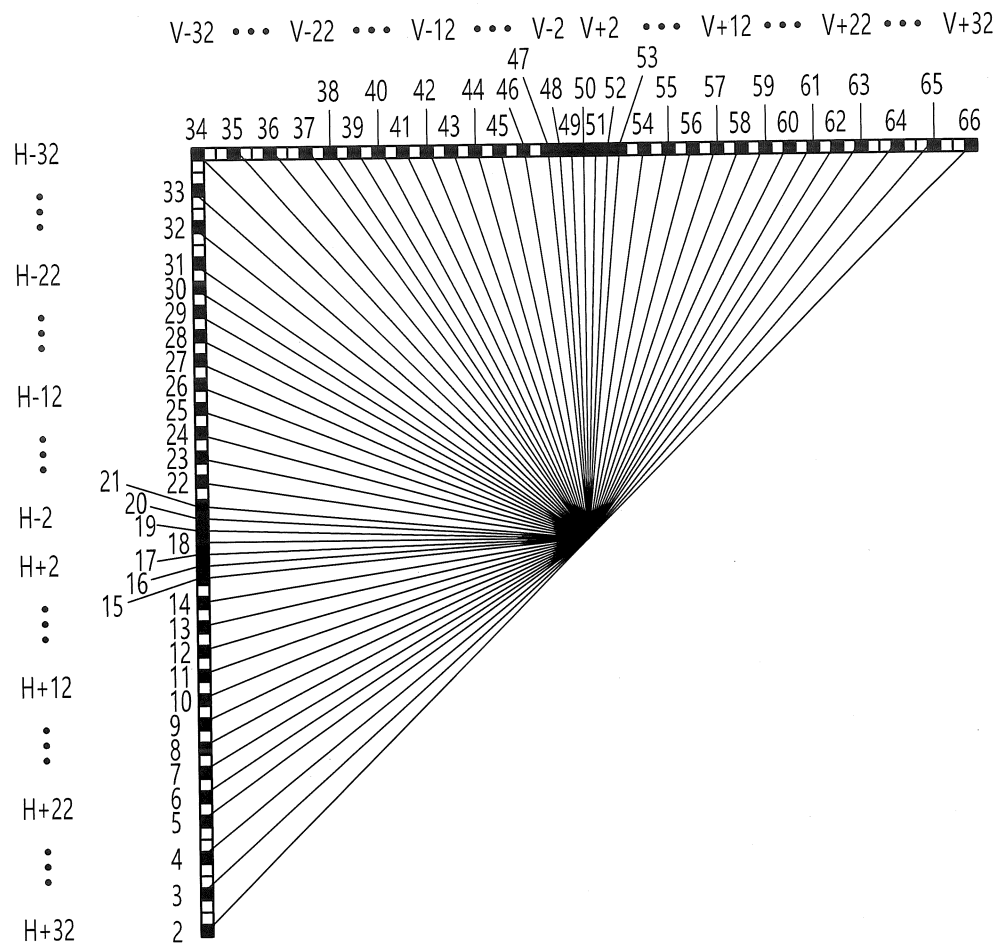


FIG. 6

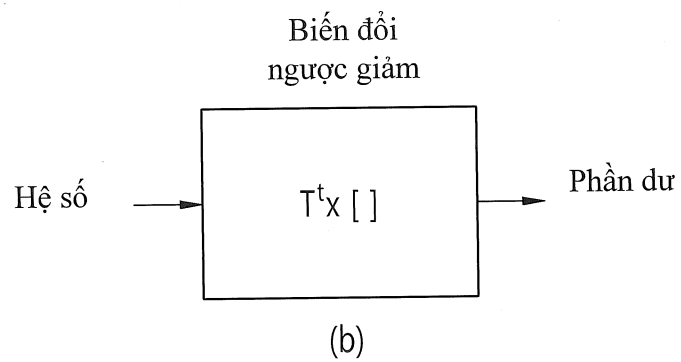
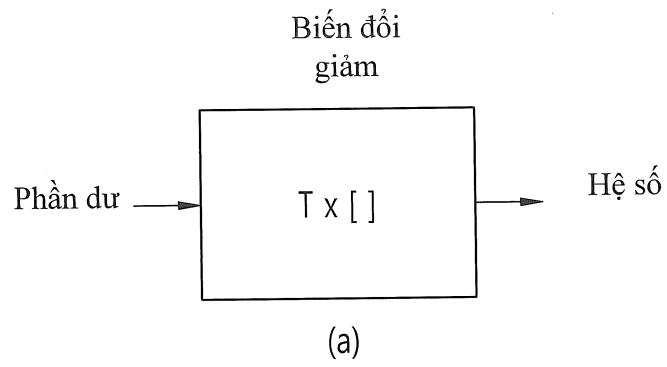


FIG. 7

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

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

(a)

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

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

(b)

FIG. 8

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

(a)

|   |                                                                                     |                                                                                     |                                                                                     |
|---|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1 | 3                                                                                   | 6                                                                                   |  |
| 2 | 5                                                                                   |  |  |
| 4 | 8                                                                                   |  |  |
| 7 |  |  |  |

(b)

FIG. 9

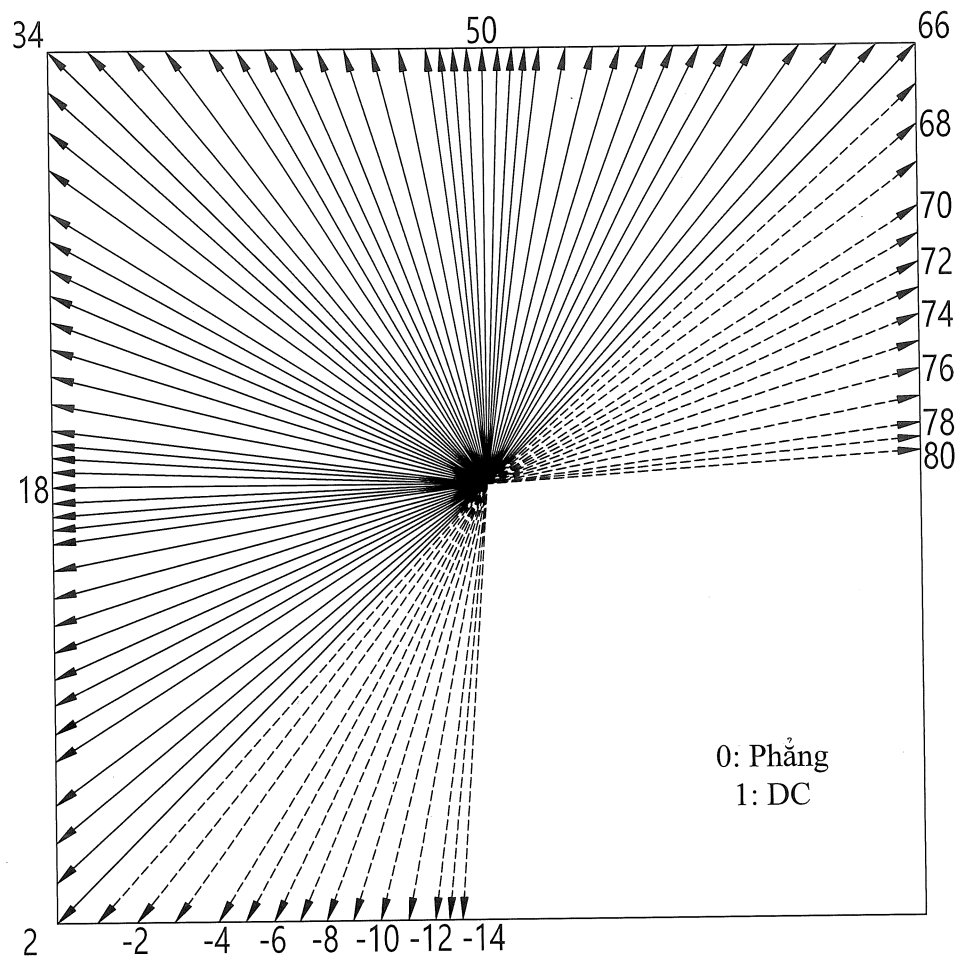
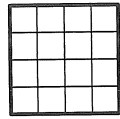
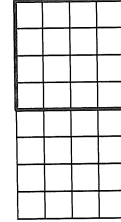
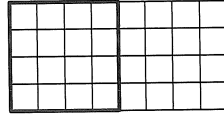


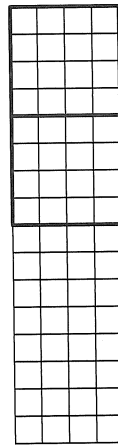
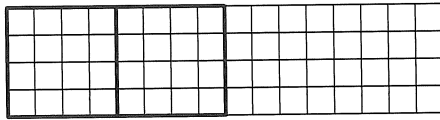
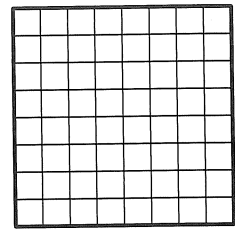
FIG. 10



(a) 4x4



(b) 8x4 / 4x8

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

(d) 8x8

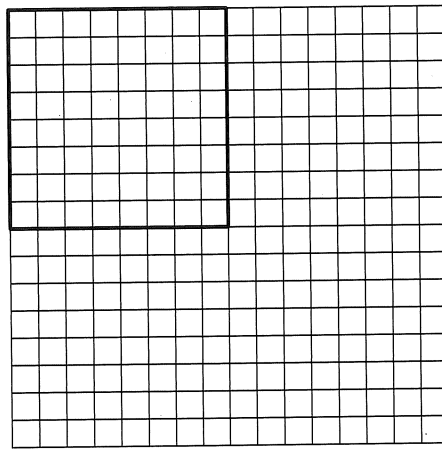
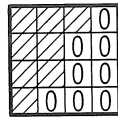
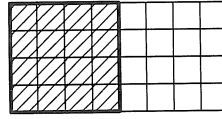
(e)  $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

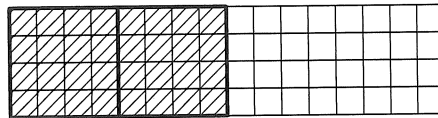
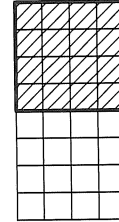
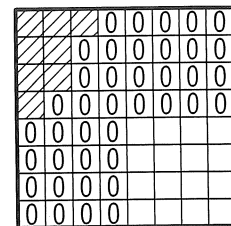
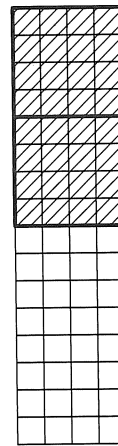
FIG. 11



(a) 4x4



(b) 8x4 / 4x8

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

(d) 8x8

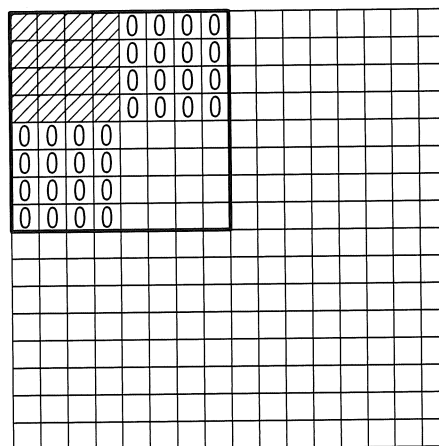
(e)  $M \times N (M \geq 8, N \geq 8, M > 8 \text{ hoặc } N > 8)$

FIG. 12

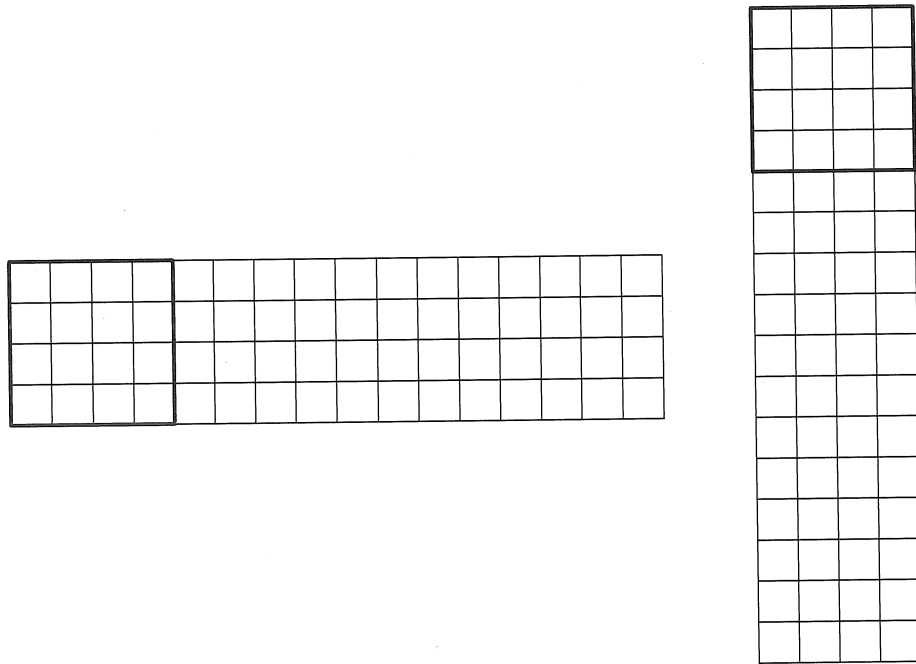
 $4 \times N / N \times 4, \text{ khi } N \geq 16$

FIG. 13

|  |   |   |   |
|--|---|---|---|
|  |   |   | 0 |
|  |   | 0 | 0 |
|  |   | 0 | 0 |
|  | 0 | 0 | 0 |

(a) 4x4

|  |  |  |  |   |   |   |   |
|--|--|--|--|---|---|---|---|
|  |  |  |  | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 |

|   |   |   |   |
|---|---|---|---|
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

(b) 8x4 / 4x8

|  |  |  |  |  |  |  |  |   |   |   |   |   |   |   |   |
|--|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   |   |   |   |
|---|---|---|---|
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

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

|  |  |  |  |   |   |   |   |   |   |   |   |   |   |   |   |
|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   |   |   |   |
|---|---|---|---|
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

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



FIG. 15

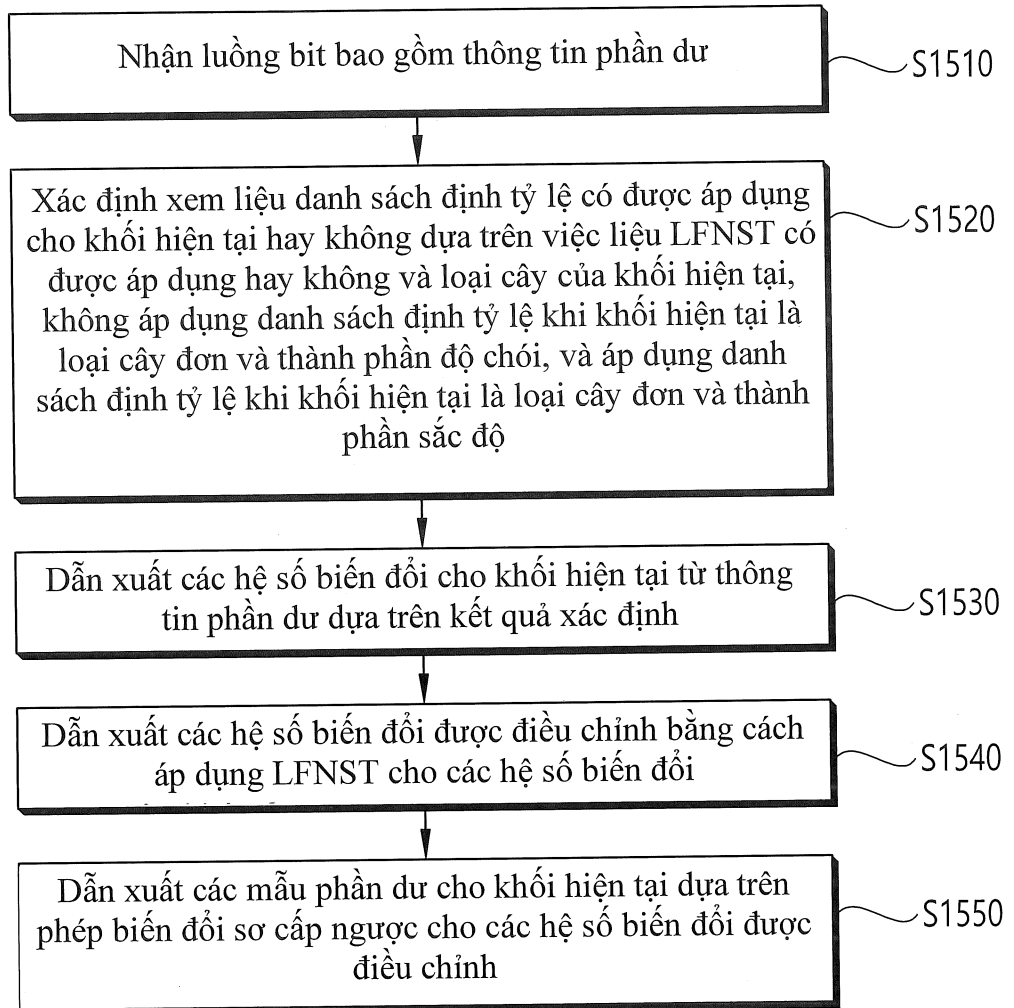


FIG. 16

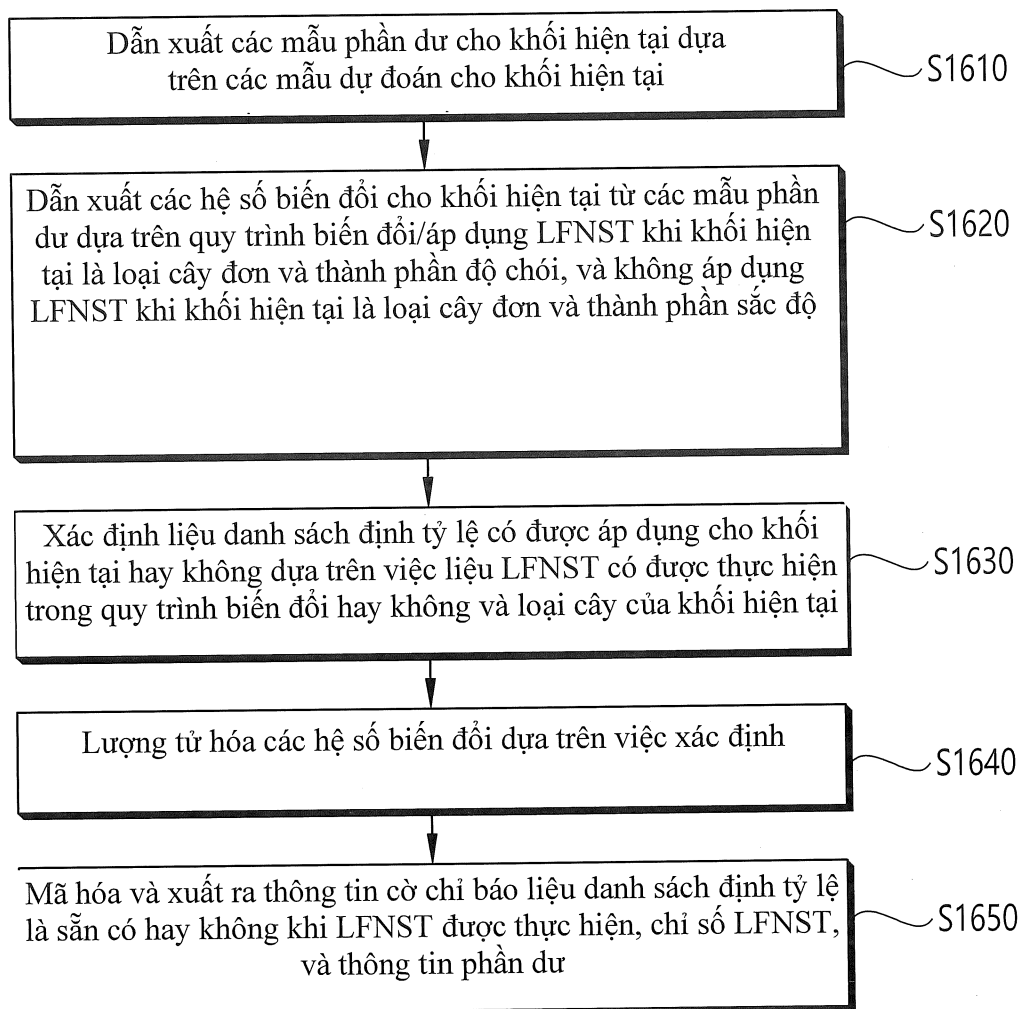


FIG. 17

