



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11)  
CỤC SỞ HỮU TRÍ TUỆ



1-0049036

(51)<sup>2020.01</sup> H04N 19/18; H04N 19/48; H04N 19/176 (13) B

---

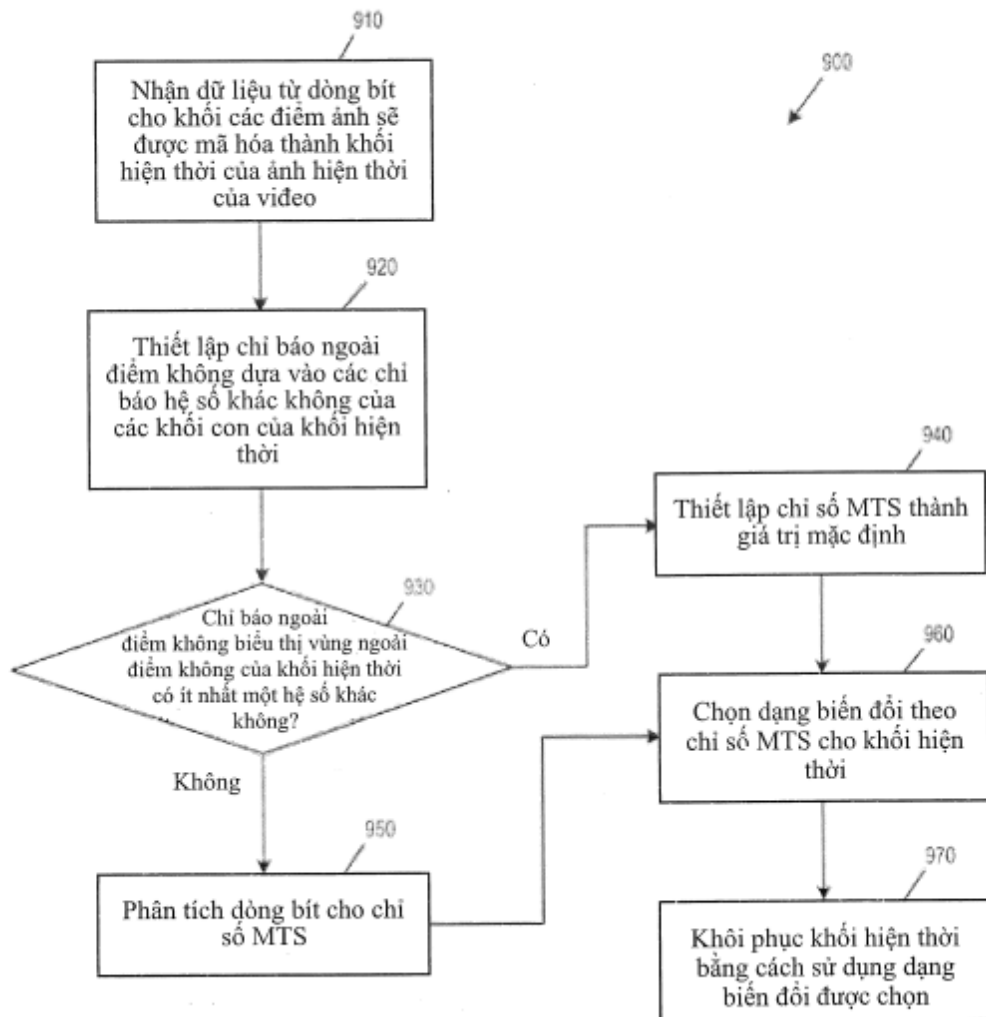
- (21) 1-2022-02870 (22) 27/11/2020  
(86) PCT/CN2020/132374 27/11/2020 (87) WO2021/104473 03/06/2021  
(30) 62/940,955 27/11/2019 US; 17/105,573 26/11/2020 US  
(45) 25/07/2025 448 (43) 25/08/2022 413A  
(73) HFI INNOVATION INC. (TW)  
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan  
(72) CHIANG, Man-Shu (TW); HSU, Chih-Wei (TW); CHUANG, Tzu-Der (TW);  
CHEN, Ching-Yeh (TW).  
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)
- 

(54) PHƯƠNG PHÁP GIẢI MÃ VIDEO, PHƯƠNG PHÁP MÃ HÓA VIDEO, VÀ  
THIẾT BỊ ĐIỆN TỬ

(21) 1-2022-02870

(57) Sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video, thiết bị điện tử. Bộ giải mã video nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video. Bộ giải mã thiết lập chỉ báo ngoài điểm không dựa vào một hoặc nhiều chỉ báo hệ số khác không của các khối con của khối hiện thời. Khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, bộ giải mã phân tích dòng bit cho chỉ số MTS. Khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không có ít nhất một hệ số khác không, bộ giải mã thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit. Bộ giải mã chọn dạng biến đổi theo chỉ số MTS cho khối hiện thời. Bộ giải mã khôi phục khối hiện thời bằng cách sử dụng dạng biến đổi được chọn.

FIG. 9



### Lĩnh vực kỹ thuật được đề cập

Sáng chế nhìn chung đề cập đến lập mã video. Cụ thể hơn là, sáng chế đề cập đến phương pháp giải mã video, phương pháp mã hóa video, và thiết bị điện tử.

### Tình trạng kỹ thuật của sáng chế

Trừ khi được chỉ rõ mang nghĩa khác ở bản mô tả này, nếu không thì những khía cạnh được mô tả ở phần tình trạng kỹ thuật của sáng chế này không phải là giải pháp kỹ thuật ưu tiên cho các điểm yêu cầu bảo hộ được liệt kê ở dưới đây cũng như không được coi là giải pháp kỹ thuật ưu tiên thuộc phần này.

Chuẩn lập mã video hiệu suất cao HEVC (High-Efficiency Video Coding - HEVC) là một chuẩn lập video quốc tế mới nhất được phát triển bởi nhóm hợp tác chung về mã hóa video JCT-VC (Joint Collaborative Team on Video Coding - JCT-VC). HEVC dựa trên cấu trúc mã hóa biến đổi tương tự với phép biến đổi DCT bù chuyển động dựa vào khối mạch tích hợp. Đơn vị cơ bản để nén, về mặt thuật ngữ được gọi là đơn vị lập mã CU (Coding Unit - CU), là khối vuông có kích thước  $2N \times 2N$ , và mỗi CU có thể được chia đệ quy thành bốn CU nhỏ hơn cho đến khi kích cỡ nhỏ nhất định trước đạt được. Mỗi CU bao gồm một hoặc nhiều đơn vị dự đoán PU (Prediction Unit - PU).

Để đạt được hiệu suất lập mã tốt nhất cho kiến trúc lập mã tích hợp trong HEVC, dự đoán nội ảnh và dự đoán liên ảnh có thể được sử dụng cho mỗi PU. Đối với các cơ chế dự đoán nội ảnh, các điểm ảnh được tái tạo liên kề theo không gian có thể được sử dụng để tạo ra các dự đoán có hướng. Có tới 35 hướng trong HEVC. Đối với các cơ chế dự đoán liên ảnh, các khung hình tham chiếu được khôi phục theo thời gian có thể được sử dụng để tạo ra các dự đoán được bù chuyển động. Có ba loại cơ chế dự đoán liên ảnh, bao gồm cơ chế rẽ nhánh (Skip), cơ chế trộn (Merge) và cơ chế dự đoán vectơ tiên tiến liên ảnh AMVP (Inter Advanced Motion Vector Prediction - AMVP).

Sau khi dự đoán, các phần dư dự đoán cho một CU được chia thành các đơn vị biến đổi TU (Transform Unit - TU) và được lập mã bằng phép biến đổi và lượng tử hóa. HEVC sử dụng phép biến đổi cosin rời rạc dạng II (DCT-II) là phép đa biến đổi

(biến đổi chính) bởi vì nó có đặc tính nén năng lượng mạnh. Phép biến đổi sin rời rạc DST (Discrete Sine Transform - DST) cũng được đề xuất để được sử dụng thay phép DCT đối với những phương vị nội ảnh theo phương chéo. Đối với phần dư được dự đoán liên ảnh, DCT-II là phép biến đổi duy nhất được sử dụng ở phiên bản HEVC hiện thời. Phép biến đổi sin rời rạc dạng VII (DST-VII) và phép biến đổi côsin rời rạc dạng VIII (DCT-VIII) cũng được sử dụng trong một số trường hợp.

### **Bản chất kỹ thuật của sáng chế**

Phần bản chất kỹ thuật dưới đây chỉ nhằm mục đích minh họa chứ không nhằm giới hạn sáng chế theo bất kỳ cách thức nào. Tức là, phần bản chất kỹ thuật dưới đây được trình bày nhằm đề xuất những ý tưởng, những điểm nổi bật, những lợi ích và những hiệu quả có lợi của những kỹ thuật mới và chưa từng minh được mô tả ở bản mô tả này. Việc chọn một số chứ không phải toàn bộ các phương án được mô tả rõ hơn dưới đây ở phần mô tả chi tiết sáng chế. Do đó, phần bản chất kỹ thuật của sáng chế dưới đây không nhằm định rõ các dấu hiệu cơ bản của các đối tượng được bảo hộ, cũng như không nhằm xác định phạm vi của các đối tượng được bảo hộ.

Một số phương án đề xuất phương pháp chọn phép đa biến đổi MTS (Multiple Transform Selection - MTS). Bộ giải mã video nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video. Bộ giải mã thiết lập chỉ báo ngoài điểm không dựa vào các chỉ báo hệ số khác không của các khối con của khối hiện thời. Khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, bộ giải mã phân tích dòng bit cho chỉ số MTS. Hoặc, khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không có ít nhất một hệ số khác không, bộ giải mã thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit. Bộ giải mã chọn dạng biến đổi từ nhiều nhân biến đổi theo chỉ số MTS cho khối hiện thời. Bộ giải mã khôi phục khối hiện thời bằng cách sử dụng dạng biến đổi được chọn.

Dạng biến đổi xác định các biến đổi ngang và đứng của các khối biến đổi thành phần sáng (thành phần luma) liên quan ở khối hiện thời. Khi khối hiện thời là khối lập mã chứa một hoặc nhiều khối biến đổi, vùng ngoài điểm không của khối hiện thời là vùng của khối hiện thời được định nghĩa như sau. Theo một ví dụ, vùng ngoài điểm không là vùng nằm ngoài vùng 16x16 trên cùng-bên trái của khối hiện thời. Theo ví dụ

khác, vùng ngoài điểm không được định bằng một hoặc nhiều vị trí khối con vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang. Vị trí ngưỡng theo phương ngang và vị trí ngưỡng theo phương đứng là các vị trí  $(x_S, y_S)$  của các nhóm lập mã trong khối biến đổi hiện thời (hoặc khối con nhất định tại vị trí  $(x_S, y_S)$  nằm trong khối biến đổi hiện thời). Ví dụ, khối con bao gồm 16 hệ số biến đổi. Nói cách khác, khối con bao gồm các hệ số biến đổi  $4 \times 4$ .

Theo một số phương án, chỉ báo ngoài điểm không được sử dụng tiếp để biểu thị liệu khối hiện thời có một hay nhiều hệ số khác không ở vùng ngoài điểm không của khối hiện thời. Vùng ngoài điểm không của khối hiện thời là vùng của khối hiện thời được định bằng các vị trí khối con vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang. Chỉ báo ngoài điểm không được định dựa vào một hoặc nhiều chỉ báo hệ số khác không của các khối con, được sử dụng để biểu thị liệu các khối con của khối hiện thời, được xác định nằm ở vùng ngoài điểm không của khối hiện thời, có một hay nhiều hệ số khác không.

Theo một số phương án, giá trị mặc định của dạng biến đổi cho MTS là phép biến đổi cosin rời rạc dạng II (DCT-2) cho cả biến đổi ngang và đứng. Theo một số phương án, biến đổi không phân tách tần số thấp LFNST (Low-Frequency Non-Separable Transform - LFNST) được vô hiệu hóa cho khối hiện thời khi MTS được sử dụng. Theo một số phương án, MTS được vô hiệu hóa khi LFNST được sử dụng.

### **Mô tả vắn tắt các hình vẽ**

Các hình vẽ đính kèm được sử dụng nhằm giúp hiểu rõ hơn sáng chế, và được đưa vào và cấu thành nên một phần của sáng chế. Các hình vẽ minh họa các phương án thực hiện của sáng chế và, cùng với phần mô tả, đóng vai trò giải thích các nguyên lý của sáng chế. Có thể thấy là các hình vẽ không nhất thiết phải đúng về mặt tỷ lệ vì một số bộ phận có thể được thể hiện không đúng với tỷ lệ kích thước trong thực tế nhằm minh họa rõ ý tưởng của sáng chế.

FIG. 1 minh họa phép biến đổi không phân tách tần số thấp (LFNST);

FIG. 2 minh họa khối biến đổi  $32 \times 32$  có các hệ số không và khác không;

FIG. 3 minh họa cách sử dụng các chỉ báo hệ số khác không của các khối con để xác định liệu có phân tích chỉ số chọn phép đa biến đổi MTS (MTS);

FIG.4 minh họa bộ mã hóa video được lấy làm ví dụ;

FIG. 5 minh họa những bộ phận của bộ mã hóa thực hiện phép MTS bằng cách tùy chọn các phép biến đổi cho phép;

FIG. 6 minh họa về mặt khái niệm quy trình sử dụng phép chọn đa biến đổi để mã hóa khối các điểm ảnh;

FIG.7 minh họa bộ giải mã video được lấy làm ví dụ;

FIG. 8 minh họa những bộ phận của bộ giải mã thực hiện phép MTS bằng cách tùy chọn các phép biến đổi cho phép;

FIG. 9 minh họa về mặt khái niệm quy trình sử dụng phép chọn đa biến đổi để giải mã khối các điểm ảnh; và

FIG.10 minh họa về mặt ý tưởng hệ thống điện tử qua đó một số phương án của sáng chế được thực hiện.

### **Mô tả chi tiết sáng chế**

Ở phần mô tả chi tiết sáng chế dưới đây, nhiều phương án chi tiết cụ thể được trình bày dưới dạng các ví dụ nhằm giúp hiểu toàn diện các nội dung sáng chế liên quan. Bất kỳ mọi sửa đổi, mọi suy dẫn và/hoặc mọi mở rộng dựa trên nội dung sáng chế được mô tả ở bản mô tả này đều nằm trong phạm vi bảo hộ của sáng chế. Trong một số trường hợp, các phương pháp, các quy trình, các bộ phận, và/hoặc hệ vi mạch vốn đã biết rõ mà liên quan đến một hoặc nhiều phương án thực hiện được lấy làm ví dụ được bộc lộ ở bản mô tả này có thể được mô tả ở mức độ vừa phải chứ không mô tả chi tiết quá, để tránh làm rối các khía cạnh nội dung của sáng chế.

#### **I. Phép chọn đa biến đổi (MTS)**

Để nâng cao khả năng nén dữ liệu điểm ảnh thông qua phép biến đổi, theo một số phương án của sáng chế, mô hình phép chọn đa biến đổi (MTS) được sử dụng để lập mã phần dư cho các khối được lập mã nội ảnh và/hoặc liên ảnh. Mô hình MTS sử dụng nhiều phép biến đổi được chọn từ nhóm các phép DCT/DST, bao gồm phép biến đổi DCT-8 mới nhất được giới thiệu. Cụ thể là, theo một số phương án, ngoài DCT-II thì mô hình MTS được sử dụng để lập mã phần dư cho các khối được lập mã nội ảnh, ví dụ, bằng cách sử dụng DCT-8 và/hoặc DST-7. Bảng 1 dưới đây thể hiện các hàm cơ bản của DST/DCT được chọn.

Bảng 1: Các hàm cơ bản biến đổi của DCT-II/VIII và DSTVII cho đầu vào N điểm

Dạng biến đổi	Hàm cơ bản $T_i(j)$ , $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ <p>trong đó, <math>\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} &amp; i = 0 \\ 1 &amp; i \neq 0 \end{cases}</math></p>
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

Để giữ tính trực giao của ma trận biến đổi, các ma trận biến đổi được lượng tử hóa chính xác hơn các ma trận biến đổi trong HEVC. Theo một số phương án, để giữ các giá trị trung gian của các hệ số biến đổi nằm trong phạm vi 16 bit, sau khi biến đổi ngang và sau khi biến đổi đứng, thì tất cả các hệ số đều phải có 10 bit.

Để điều khiển mô hình MTS, các cờ kích hoạt riêng được xác định tại phân mức SPS cho khối nội ảnh và liên ảnh, một cách tương ứng. Theo một số phương án, khi MTS được sử dụng tại SPS, cờ phân mức CU được báo hiệu để biểu thị liệu MTS được áp dụng hay không. Ở đây, MTS chỉ được áp dụng cho luma. Cờ phân mức MTS CU ( $mts\_idx$ ) có thể được báo hiệu khi cả chiều rộng và chiều cao nhỏ hơn hoặc bằng 32 và/hoặc cờ CBF cho khối hiện thời (phân mức CU) bằng 1. Nếu cờ MTS CU bằng 0, thì DCT-2 được áp dụng theo cả hai phương (biến đổi ngang và biến đổi đứng). Tuy nhiên, nếu cờ MTS CU bằng 1, thì hai cờ khác được báo hiệu thêm để biểu thị dạng biến đổi nào cho phương ngang và phương đứng, một cách tương ứng. Biến đổi và báo hiệu bằng ánh xạ dựa vào cờ MTS CU như được thể hiện trên Bảng 2.

Bảng 2: Ánh xạ cho các phép biến đổi theo mô hình MTS

MTS_index	MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	Nội ảnh/Liên ảnh	
				Ngang	Đứng
0	0			DCT2	
1	1	0	0	DST7	DST7
2		0	1	DCT8	DST7
3		1	0	DST7	DCT8
4		1	1	DCT8	DCT8

Ở bảng trên, các tổ hợp khác nhau giữa MTS\_CU\_flag, MTS\_Hor\_flag, và MTS\_Ver\_flag được ánh xạ thành các giá trị khác nhau của chỉ số MTS, có thể được sử dụng để chọn tổ hợp giữa các biến đổi phương ngang và đứng. Ví dụ, chỉ số MTS bằng 0 biểu thị DCT2 được sử dụng cho cả hai phương. Theo ví dụ khác, chỉ số MTS bằng 1 biểu thị DST7 được sử dụng cho cả hai phương. Theo ví dụ khác, chỉ số MTS bằng 2 biểu thị DCT8 được sử dụng cho biến đổi phương ngang và DST7 được sử dụng cho biến đổi phương đứng. Theo ví dụ khác, chỉ số MTS bằng 3 biểu thị DST7 được sử dụng cho biến đổi phương ngang và DCT8 được sử dụng cho biến đổi phương đứng. Theo ví dụ khác, chỉ số MTS bằng 4 biểu thị DCT8 được sử dụng cho cả hai phương.

Theo một số phương án, để thực hiện chính xác ma trận biến đổi, các nhân biến đổi chính 8 bit được sử dụng, và toàn bộ các nhân biến đổi HEVC được sử dụng, bao gồm DCT-2 4 điểm và DST-7 4 điểm, DCT-2 8 điểm, 16 điểm và 32 điểm. Ngoài ra, các nhân biến đổi khác, chẳng hạn DCT-2 64 điểm, DCT-8 4 điểm, DST-7 và DCT-8 8 điểm, 16 điểm và 32 điểm, sử dụng các nhân biến đổi chính 8 bit. Theo một số phương án, để giảm độ phức tạp của DST-7 và DCT-8 kích cỡ lớn, các hệ số biến đổi tần số cao được đặt bằng không đối với các khối DST-7 và DCT-8 có kích cỡ (chiều rộng hoặc chiều cao, hoặc cả chiều rộng và chiều cao) bằng 32. Chỉ các hệ số thuộc



vùng tần số thấp 16x16 được sử dụng.

#### A. Biến đổi không phân tách tần số thấp (LFNST)

Theo một số phương án, biến đổi không phân tách tần số thấp (LFNST), còn được hiểu là biến đổi phụ suy giảm, được áp dụng giữa phép lượng tử hóa và biến đổi chính thuận (tại bộ mã hóa) và giữa phép giải lượng tử và biến đổi chính ngược (tại bộ giải mã). FIG. 1 minh họa phép biến đổi không phân tách tần số thấp (LFNST). Theo LFNST, phép biến đổi không phân tách 4x4 hoặc phép biến đổi không phân tách 8x8 được áp dụng theo kích cỡ khối. Ví dụ, LFNST 4x4 được áp dụng cho các khối nhỏ (tức là,  $\min(\text{width}, \text{height}) < 8$ ) và LFNST 8x8 được áp dụng cho các khối lớn hơn (tức là,  $\min(\text{width}, \text{height}) > 4$ ). Việc áp dụng phép biến đổi không phân tách, mà được sử dụng trong LFNST, được mô tả dưới đây bằng cách sử dụng đầu vào là một ví dụ. Để áp dụng LFNST 4x4, khối X đầu vào 4x4

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

trước tiên được biểu diễn dưới dạng vectơ  $\vec{X}$ :

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

Biến đổi không phân tách được tính là  $\vec{F} = T \cdot \vec{X}$ , trong đó  $\vec{F}$  biểu thị vectơ hệ số biến đổi, và T là ma trận biến đổi 16x16. Vectơ  $\vec{F}$  hệ số 16x1 được tái tạo lại thành khối 4x4 nhờ sử dụng lệnh quét (ngang, đứng hoặc chéo) đối với khối đó. Các hệ số có chỉ số nhỏ hơn sẽ được thay bằng chỉ số quét nhỏ hơn trong khối hệ số 4x4.

LFNST dựa theo phương pháp nhân ma trận trực tiếp để áp dụng phép biến đổi không phân tách sao cho nó được thực hiện trong một lần duy nhất mà không lặp lại nhiều lần. Kích thước ma trận biến đổi không phân tách được giảm để giảm thiểu mức độ phức tạp tính toán và không gian bộ nhớ để lưu các hệ số biến đổi. Vì vậy, phương pháp biến đổi không phân tách suy giảm (hoặc RST) được sử dụng trong LFNST. Theo một số phương án, phương pháp biến đổi không phân tách suy giảm ánh xạ N (N thường bằng 64 đối với NSST 8x8) vectơ chiều thành R vectơ chiều ở không gian khác, trong đó N/R ( $R < N$ ) là hệ số suy giảm. Vì vậy, thay cho ma trận NxN, ma trận RST trở thành ma trận R×N sau:

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

trong đó R hàng của ma trận biến đổi là R của ma trận biến đổi là R chiều của N không gian chiều. Ma trận biến đổi ngược cho RT là chuyển vị của phép biến đổi thuận. Đối với LFNST 8x8, hệ số suy giảm bằng 4 được áp dụng, và ma trận trực tiếp 64x64, là kích cỡ ma trận biến đổi không phân tách 8x8 thông thường, được giảm xuống thành ma trận trực tiếp 16x48. Vì vậy, ma trận RST ngược 48x16 được sử dụng tại bộ giải mã để tạo ra các hệ số đa biến đổi (chính) ở các vùng trên cùng-bên trái 8x8. Khi các ma trận 16x48 được áp dụng thay cho 16x64 với cùng cấu hình thiết lập biến đổi, mỗi chúng lấy 48 dữ liệu đầu vào từ ba khối 4x4 ở khối 8x8 trên cùng-bên trái không chứa khối 4x4 bên phải-dưới cùng. Nhờ kích thước được giảm, nên việc sử dụng bộ nhớ để chứa toàn bộ các ma trận LFNST được giảm từ 10KB xuống 8KB với độ sụt hiệu suất chấp nhận được.

Theo một số phương án, để giảm độ phức tạp tính toán, LFNST được áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên là không có nghĩa. Nói cách khác, toàn bộ các hệ số chỉ theo biến đổi chính bằng 0 khi LFNST được áp dụng. Điều này cho phép việc báo hiệu chỉ số LFNST sẽ được đặt điều kiện theo vị trí có nghĩa cuối cùng, và nhờ đó sẽ tránh được việc quét hệ số phụ thêm trong thiết kế LFNST (nếu không thì, điều đó là việc cần thiết để kiểm tra các hệ số có nghĩa chỉ ở các vị trí cụ thể).

Cách xử lý trường hợp xấu nhất của LFNST (khi thực hiện nhiều phép nhân trên điểm ảnh) sẽ giới hạn các biến đổi không phân tách cho các khối 4x4 và 8x8 thành các biến đổi 8x16 và 8x48, một cách tương ứng. Đối với những trường hợp này, theo một số phương án, vị trí quét có nghĩa cuối cùng được ràng buộc sao cho nhỏ hơn 8 khi LFNST được áp dụng, cho các kích cỡ khác nhỏ hơn 16. Đối với các khối có hình dạng 4xN và Nx4 và N > 8, việc giới hạn theo đề xuất ở trên có nghĩa là LFNST được áp dụng chỉ một lần, và chỉ áp dụng cho vùng 4x4 trên cùng-bên trái. Vì toàn bộ các hệ số chỉ theo biến đổi chính bằng 0 khi LFNST được áp dụng, nên số lượng các thuật toán dùng cho các biến đổi chính được giảm ở những trường hợp vừa nêu. Xét từ bộ

mã hóa, việc lượng tử hóa các hệ số được đơn giản hóa khi các biến đổi LFNST được kiểm tra. Phép lượng tử tối ưu độ méo được tính cực đại cho 16 hệ số đầu tiên (theo thứ tự quét), các hệ số còn lại được thiết đặt bằng 0.

Theo một số phương án, 4 thiết lập biến đổi và 2 ma trận (nhân) biến đổi không phân tách cho một tập biến đổi được sử dụng ở LFNST. Ánh xạ từ cơ chế dự đoán nội ảnh thành tập biến đổi được thiết lập trước như được thể hiện ở Bảng 3:

Bảng 3: Bảng chọn biến đổi cho LFNST

IntraPredMode	Tr. setindex
$\text{IntraPredMode} < 0$	1
$0 \leq \text{IntraPredMode} \leq 1$	0
$2 \leq \text{IntraPredMode} \leq 12$	1
$13 \leq \text{IntraPredMode} \leq 23$	2
$24 \leq \text{IntraPredMode} \leq 44$	3
$45 \leq \text{IntraPredMode} \leq 55$	2
$56 \leq \text{IntraPredMode} \leq 80$	1
$81 \leq \text{IntraPredMode} \leq 83$	0

Nếu một trong số ba cơ chế CCLM (INTRA\_LT\_CCLM, INTRA\_T\_CCLM hoặc INTRA\_L\_CCLM) được sử dụng cho khối hiện thời ( $81 \leq \text{predModeIntra} \leq 83$ ), thì tập biến đổi 0 được chọn cho khối màu (chroma) hiện thời. Đối với mỗi tập biến đổi, ứng viên biến đổi phụ không phân tách được xác định tiếp bằng chỉ số LFNST được báo hiệu hiện. Chỉ số này được báo hiệu trong dòng bit một lần cho CU nội ảnh sau các hệ số biến đổi.

Vì LFNST bị giới hạn về khả năng áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên là không có nghĩa, theo một số phương án, nên lập mã chỉ số LFNST phụ thuộc vào vị trí của hệ số có nghĩa cuối cùng. Ngoài ra, chỉ số LFNST

được lập mã thuộc tính nhưng không phụ thuộc vào cơ chế dự đoán nội ảnh, và chỉ vùng trống đầu tiên được lập mã thuộc tính. Ngoài ra, LFNST được áp dụng cho CU nội ảnh ở cả mảng (slice) nội ảnh và liên ảnh, và cho cả Luma và Chroma. Nếu cây lưỡng được sử dụng, thì các tham số LFNST cho Luma và Chroma được báo hiệu riêng. Đối với slice liên ảnh (cây lưỡng được vô hiệu hóa), một chỉ số LFNST duy nhất được báo hiệu và được sử dụng cho cả Luma và Chroma.

Giả sử rằng CU lớn (ví dụ, lớn hơn 64x64) bị phân chia không tương minh (phân lát TU) do giới hạn kích cỡ biến đổi lớn nhất (kích cỡ 64x64 hoặc được thiết lập theo cấu hình, việc tìm kiếm chỉ số LFNST có thể làm tăng bộ đệm dữ liệu lên bốn lần cho số lượng các trạng thái truyền giải mã nhất định. Vì vậy, kích cỡ cho phép lớn nhất cho LFNST là 64x64 hoặc kích cỡ biến đổi lớn nhất. Theo một số phương án, MTS được sử dụng chỉ khi LFNST không được dùng.

#### B. Báo hiệu chỉ số MTS

Theo một số phương án, chỉ số MTS xác định các nhân biến đổi nào được áp dụng theo phương ngang và đứng của các khối biến đổi luma liên quan ở khối hiện thời của các điểm ảnh (ví dụ, CU). Chỉ số MTS được báo hiệu tại phân mức CU theo cú pháp dưới đây của LFNST (sau khi báo hiệu/phân tích toàn bộ các khối biến đổi ở khối hiện thời). Ví dụ về bảng cú pháp CU tương ứng được thể hiện ở Bảng 4.

Bảng 4: Bảng cú pháp CU bao gồm chỉ số MTS

coding_unit ( x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType ) {...	Biến mô tả
if( cu_cbf ) {...	
LfnstDcOnly = 1	
LfnstZeroOutSigCoeffFlag = 1	
MtsZeroOutSigCoeffFlag = 1	
transform_tree( x0, y0, cbWidth, cbHeight, treeType, ch	

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

<pre> Max( cbWidth, cbHeight ) &lt;= 32 &amp;&amp;     IntraSubPartitionsSplit[ x0 ][ y0 ] == ISP_NO_SPLIT &amp;&amp; cu_sbt_flag == 0 &amp;&amp;     MtsZeroOutSigCoeffFlag == 1 &amp;&amp; tu_cbf_luma[ x0 ][ y0 ] ) { </pre>	
<pre>     if( ( ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTER &amp;&amp;         sps_explicit_mts_inter_enabled_flag )            ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &amp;&amp;         sps_explicit_mts_intra_enabled_flag ) ) ) </pre>	
<pre>         mts_idx </pre>	ae(v)
<pre>     } </pre>	
<pre> } </pre>	

Tiêu chuẩn báo hiệu MTS là khi toàn bộ các điều kiện dưới đây được thỏa mãn, chỉ số MTS được báo hiệu; nếu không thì, chỉ số MTS không được báo hiệu (và được xem là 0 (giá trị mặc định)).

- CB hiện thời là cho luma
- Cbf của CB hiện thời lớn hơn 0
- Chiều rộng CB  $\leq 32$  và chiều cao CB  $\leq 32$
- CU hiện thời không là ISP (ISP có quy tắc chọn ngầm riêng cho MTS)
- CU hiện thời không là SBT (SBT có quy tắc chọn ngầm riêng cho MTS)
- Transform\_skip\_flag cho khối biến đổi trong CB hiện thời = 0
- MtsZeroOutSigCoeffFlag = 1 (được khởi tạo bằng 1 và được cập nhật cho từng TU/TB trong CU/CB hiện thời như được thể hiện ở Bảng 5)

Bảng 5: Bảng cú pháp lập mã phân dư

residual_coding( x0, y0, log2TbWidth, log2TbHeight, cIdx )	Biến mô tả
<pre>{   if( ( ( sps_mts_enabled_flag &amp;&amp; cu_sbt_flag &amp;&amp;     log2TbWidth &lt; 6 &amp;&amp; log2TbHeight &lt; 6 ) )     &amp;&amp; cIdx == 0 &amp;&amp; log2TbWidth &gt; 4 )</pre>	
<pre>    log2ZoTbWidth = 4</pre>	
<pre>  else</pre>	
<pre>    log2ZoTbWidth = Min( log2TbWidth, 5 )</pre>	
<pre>  if( ( sps_mts_enabled_flag &amp;&amp; cu_sbt_flag &amp;&amp;     log2TbWidth &lt; 6 &amp;&amp; log2TbHeight &lt; 6 ) )     &amp;&amp; cIdx == 0 &amp;&amp; log2TbHeight &gt; 4 )</pre>	
<pre>    log2ZoTbHeight = 4</pre>	
<pre>  else</pre>	
<pre>    log2ZoTbHeight = Min( log2TbHeight, 5 ) ...</pre>	
<pre>    numSbCoeff = 1 &lt;&lt; ( log2SbW + log2SbH )</pre>	
<pre>    lastScanPos = numSbCoeff</pre>	
<pre>    lastSubBlock = ( 1 &lt;&lt; ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1</pre>	
<pre>    do {</pre>	
<pre>      if( lastScanPos == 0 ) {</pre>	

lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos-- ...	
} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
if( lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[ x0 ][ y0 ][ cIdx ] && lastScanPos > 0 )	
LfstDcOnly = 0	
if( ( lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2 )    ( lastScanPos > 7 && ( log2TbWidth == 2    log2TbWidth == 3 ) && log2TbWidth == log2TbHeight ) )	
LfstZeroOutSigCoeffFlag = 0	
if( ( LastSignificantCoeffX > 15    LastSignificantCoeffY > 15 ) && cIdx == 0 )	
MtsZeroOutSigCoeffFlag = 0	
QState = 0	
for( i = lastSubBlock; i >= 0; i-- ) {	
startQStateSb = QState	



$xS =$ $\text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}]$ $[i][0]$	
$yS =$ $\text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}]$ $[i][1]$	
$\text{inferSbDcSigCoeffFlag} = 0$	
$\text{if}(i < \text{lastSubBlock} \ \&\& \ i > 0) \{$	
$\text{coded\_sub\_block\_flag}[xS][yS]$	$\text{ae}(v)$
$\text{inferSbDcSigCoeffFlag} = 1$	
$\} \dots \}$	

Theo một số phương án, khi bộ giải mã nhận các hệ số biến đổi cho từng TU/TB ở CU/CB hiện thời, cho từng TB (khối biến đổi), thì bộ giải mã trước tiên sẽ phân tích vị trí hệ số (khác không) có nghĩa cuối cùng và sau đó sẽ phân tích cú pháp cho từng nhóm lập mã (CG) hoặc khối con (ví dụ, CG = khối con 4x4). FIG. 2 minh họa khối biến đổi 32x32 200 có các hệ số không và khác không. Ở hình vẽ này, mỗi vị trí lưới ô vuông biểu diễn một CG 4x4 và chữ số cho mỗi vị trí lưới ô vuông biểu diễn thứ tự quét (chéo) cho từng CG trong TB. Các CG theo đó được xác định tại các lưới ô vuông được đánh số từ 0 đến 63 theo thứ tự quét chéo. Theo một số phương án, cú pháp cho mỗi CG bao gồm cờ có nghĩa ( $\text{coded\_sub\_block\_flag}$  hoặc  $\text{sb\_coded\_flag}$ ) để biểu thị liệu có bất kỳ các hệ số khác không nào nằm trong CG đó, tức là, liệu các hệ số của khối con được lập mã hay không. Cụ thể là,  $\text{sb\_coded\_flag}[xS][yS]$  xác định những điều sau cho khối con tại vị trí  $(xS, yS)$  nằm trong khối biến đổi hiện thời, trong đó khối con là mảng gồm các mức hệ số biến đổi:

Khi  $\text{sb\_coded\_flag}[xS][yS]$  bằng 0, tất cả các mức hệ số biến đổi của khối con

tại vị trí ( xS, yS ) được xem là bằng 0.

Khi `sb_coded_flag[ xS ][ yS ]` không xuất hiện, nó được xem là bằng 1. Đối với khối con mà có vị trí hệ số có nghĩa cuối cùng của TB hiện thời được xác định vị trí, `sb_coded_flag[ xS ][ yS ]` được xem là 1.

Theo một số phương án, liệu có phân tích chỉ số MTS hay không phụ thuộc vào thành phần cú pháp `MtsZeroOutSigCoeffFlag` (1 = phân tích chỉ số MTS cho TB/CU và 0 = bỏ qua phân tích chỉ số MTS cho TB/CU), mà được khởi tạo bằng 1 và được cập nhật cho từng TB luma ở CU/CB hiện thời. Theo bảng cú pháp được thể hiện ở Bảng 5, nếu `LastSignificantCoeffX > 15` hoặc `LastSignificantCoeffY > 15`, trị số của `MtsZeroOutSigCoeffFlag` được thiết lập bằng 0, và chỉ số MTS được bỏ qua (không được báo hiệu trong dòng bit hoặc không được phân tích từ dòng bit). Theo ví dụ trên FIG. 2, nếu `LastSignificantCoeffX ≤ 15` và `LastSignificantCoeffY ≤ 15`, thì hệ số có nghĩa cuối cùng là vùng được gạch chéo 210 (các CG tại các vị trí lưới ô vuông từ 0 đến 9, từ 11 đến 13, từ 17 đến 18, và 24 theo thứ tự quét chéo). Mặt khác, nếu `LastSignificantCoeffX > 15` hoặc `LastSignificantCoeffY > 15`, thì vị trí hệ số có nghĩa cuối cùng cho TB này được xác định nằm ở vùng màu đen (các CG tại các vị trí lưới ô vuông > 24 theo thứ tự quét chéo), `MtsZeroOutSigCoeffFlag` được thiết lập bằng 0, chỉ số MTS được bỏ qua.

Tuy nhiên, có sự dư thừa giữa báo hiệu phần dư và chỉ số MTS nếu bộ mã hóa/bộ giải mã video chỉ dựa vào `LastSignificantCoeffX` và/hoặc `LastSignificantCoeffY` để xác định liệu có bỏ qua chỉ số MTS cho khối hiện thời hay không. Ví dụ, đối với TB 32x32 200, vì chỉ các hệ số thuộc vùng tần số thấp 16x16 (vùng được áp dụng MTS 210, hoặc các CG tại các vị trí lưới ô vuông từ 0 đến 9, từ 11 đến 13, từ 17 đến 18, và 24) được tiếp tục dùng cho mỗi TB được áp dụng MTS, nên chỉ số MTS bằng 0 khi tồn tại bất kỳ hệ số khác không nào nằm ngoài vùng 16x16. Tuy nhiên, nếu chỉ `MtsZeroOutSigCoeffFlag`, mà được suy dẫn từ vị trí hệ số có nghĩa cuối cùng, được sử dụng, thì dư thừa xuất hiện khi vị trí hệ số có nghĩa cuối cùng thỏa mãn `LastSignificantCoeffX ≤ 15` và `LastSignificantCoeffY ≤ 15` (ví dụ được xác định tại CG tại vị trí lưới ô vuông 24), và một số hệ số khác không có thể tồn tại ở vùng màu trắng tại các vị trí lưới ô vuông ≤ 24 (các CG tại các vị trí lưới ô vuông 10, từ 14 đến 16, và từ 19 đến 23). Ở những trường hợp này, `MtsZeroOutSigCoeffFlag` sẽ được thiết lập bằng 1 và chỉ số MTS vẫn được phân tích nhưng luôn bằng 0, điều mà có thể tạo

du thừa.

Theo một số phương án của sáng chế, chỉ số MTS được phân tích chỉ khi không có hệ số nào xuất hiện nằm ngoài vùng được áp dụng MTS (ví dụ, vùng 210). Trị số của `MtsZeroOutSigCoeffFlag` được xác định dựa vào liệu có bất kỳ hệ số khác không nào nằm ngoài vùng được áp dụng MTS hay không. Nếu có, `MtsZeroOutSigCoeffFlag` được thiết lập bằng 0 để cho chỉ số MTS không được phân tích và xem là bằng 0. Nếu không thì `MtsZeroOutSigCoeffFlag` được thiết lập bằng 1 và nếu các điều kiện báo hiệu MTS khác được thỏa mãn, thì chỉ số MTS được phân tích từ dòng bit và được sử dụng để chọn dạng biến đổi MTS.

Theo một số phương án, thành phần cú pháp `coded_sub_block_flag` có thể được sử dụng để biết liệu có bất kỳ hệ số khác không cho các khối con hoặc các CG nằm ngoài vùng 16x16 (vùng được áp dụng MTS) hay không. Sau khi trị số `coded_sub_block_flag` của khối con hoặc CG được xác định (được phân tích/được suy diễn), `MtsZeroOutSigCoeffFlag` có thể được gán theo trị số `coded_sub_block_flag` của khối con. Cụ thể là, nếu `coded_sub_block_flag` của bất kỳ khối con nằm ngoài vùng được áp dụng MTS bằng 1, thì `MtsZeroOutSigCoeffFlag` được thiết lập bằng 0.

Cụ thể là, đối với khối con tại vị trí  $(xS, yS)$  nằm trong khối biến đổi hiện thời, `coded_sub_block_flag[xS][yS]` xác định điều sau:

- Nếu `coded_sub_block_flag[xS][yS]` không xuất hiện, thì nó được xem là bằng 1.
- Nếu `coded_sub_block_flag[xS][yS]` bằng 0, thì 16 mức hệ số biến đổi của khối con tại vị trí  $(xS, yS)$  được xem là bằng 0.
- Nếu `coded_sub_block_flag[xS][yS]` bằng 1, thì áp dụng điều kiện sau:
  - Nếu  $(xS, yS)$  bằng  $(0, 0)$  và  $(LastSignificantCoeffX, LastSignificantCoeffY)$  không bằng  $(0, 0)$ , thì ít nhất một trong số 16 thành phần cú pháp `sig_coeff_flag` xuất hiện cho khối con tại vị trí  $(xS, yS)$ .
  - Nếu không thì, ít nhất một trong số 16 mức hệ số biến đổi của khối con tại vị trí  $(xS, yS)$  có trị số khác không.

Ví dụ, khi  $(xS \geq 4$  hoặc  $yS \geq 4)$  và `coded_sub_block_flag[xS][yS] == true`,

MtsZeroOutSigCoeffFlag được thiết lập bằng 0. Theo một số phương án, quy tắc được định nghĩa dựa vào coded\_sub\_block\_flag chứ không phải vị trí hệ số có nghĩa cuối cùng có thể được sử dụng để xác định trị số của MtsZeroOutSigCoeffFlag.

Bảng cú pháp tương ứng dùng để lập mã phần dư được thể hiện ở Bảng 6 dưới đây.

Bảng 6: Bảng cú pháp lập mã phần dư bằng coded\_sub\_block\_flag được sử dụng để xác định trị số của MtsZeroOutSigCoeffFlag

residual_coding( x0, y0, log2TbWidth, log2TbHeight, cIdx ) {	Biên mô tả
if( ( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbWidth > 4 )	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min( log2TbWidth, 5 )	
if( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbHeight > 4 )	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min( log2TbHeight, 5 ) ...	
numSbCoeff = 1 << ( log2SbW + log2SbH )	
lastScanPos = numSbCoeff	
lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1	

do {	
if( lastScanPos == 0 ) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos-- ...	
} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
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	
QState = 0	
for( i = lastSubBlock; i >= 0; i-- ) {	
startQStateSb = QState	

$xS =$ DiagScanOrder[ log2TbWidth – log2SbW ][ log2TbHeight – lo g2SbH ] [ i ][ 0 ]	
$yS =$ DiagScanOrder[ log2TbWidth – log2SbW ][ log2TbHeight – lo g2SbH ] [ i ][ 1 ]	
inferSbDcSigCoeffFlag = 0	
if( i < lastSubBlock && i > 0 ) {	
coded_sub_block_flag[ xS ][ yS ]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
if( (xS >= 4    yS >=4) && coded_sub_block_flag[ xS ][ yS ] && cIdx == 0 )	
MtsZeroOutSigCoeffFlag = 0	
...}	

Theo Bảng 6, nếu bất kỳ khối con hoặc CG tại vị trí ngang hoặc đứng lớn hơn hoặc bằng 4 ( $xS \geq 4$  hoặc  $yS \geq 4$ ) có các hệ số khác không (được biểu thị bằng coded\_sub\_block\_flag[xS][yS]), MtsZeroOutSigCoeffFlag được thiết lập bằng 0, và việc phân tích chỉ số MTS được bỏ qua. Nói cách khác, bảng cú pháp tương ứng có thể còn là như sau.

```
if( sb_coded_flag[ xS ][ yS ] && ( xS > 3 || yS > 3 ) && cIdx == 0 )
```

MtsZeroOutSigCoeffFlag = 0

FIG. 3 minh họa cách sử dụng các chỉ báo hệ số khác không của các khối con (coded\_sub\_block\_flag) để xác định liệu có phân tích Chỉ số MTS. Như được minh họa, các khối con tại vị trí  $x_S \leq 3$  và  $y_S \leq 3$  của khối 200 (vùng được áp dụng MTS 210) có thể có các hệ số khác không trong khi chỉ số MTS được phân tích hoặc được báo hiệu. Tuy nhiên, nếu bất kỳ khối con hoặc CG được xác định nằm trong vùng 300 tại vị trí  $x_S \geq 4$  hoặc  $y_S \geq 4$  có các hệ số khác không (được biểu thị bằng coded\_sub\_block\_flag[xS][yS]), MtsZeroOutSigCoeffFlag được thiết lập bằng 0, và việc phân tích chỉ số MTS được bỏ qua. Vùng 300 của khối hiện thời trong đó  $x_S > 3$  hoặc  $y_S > 3$  theo đó còn được gọi là vùng ngoài điểm không MTS hoặc vùng ngoài điểm không của khối hiện thời.

Theo một số phương án, coded\_sub\_block\_flag có thể được sử dụng cùng với vị trí hệ số có nghĩa cuối cùng để xác định liệu có phân tích chỉ số MTS hay không. Bảng cú pháp tương ứng dùng để lập mã phân dư được thể hiện ở Bảng 7 dưới đây.

Bảng 7:

residual_coding( x0, y0, log2TbWidth, log2TbHeight, cIdx ) {	Biến mô tả
if( ( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbWidth > 4 )	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min( log2TbWidth, 5 )	
if( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbHeight > 4 )	

log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min( log2TbHeight, 5 ) ...	
numSbCoeff = 1 << ( log2SbW + log2SbH )	
lastScanPos = numSbCoeff	
lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1	
do {	
if( lastScanPos == 0 ) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos-- ...	
} 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 )	



<pre>( lastScanPos &gt; 7 &amp;&amp; ( log2TbWidth == 2    log2TbWidth == 3 ) &amp;&amp; log2TbWidth == log2TbHeight ) )</pre>	
<pre>LfstZeroOutSigCoeffFlag = 0</pre>	
<pre>if( ( LastSignificantCoeffX &gt; 15    LastSignificantCoeffY &gt; 15 ) &amp;&amp; cIdx == 0 )</pre>	
<pre>MtsZeroOutSigCoeffFlag = 0</pre>	
<pre>QState = 0</pre>	
<pre>for( i = lastSubBlock; i &gt;= 0; i-- ) {</pre>	
<pre>startQStateSb = QState</pre>	
<pre>xS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - lo g2SbH ] [ i ][ 0 ]</pre>	
<pre>yS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - lo g2SbH ] [ i ][ 1 ]</pre>	
<pre>inferSbDcSigCoeffFlag = 0</pre>	
<pre>if( i &lt; lastSubBlock &amp;&amp; i &gt; 0 ) {</pre>	
<pre>coded_sub_block_flag[ xS ][ yS ]</pre>	ae(v)
<pre>inferSbDcSigCoeffFlag = 1</pre>	
<pre>}</pre>	

<pre> if( (xS &gt;= 4    yS &gt;=4) &amp;&amp; coded_sub_block_flag[ xS ][ yS ] &amp;&amp; cIdx == 0 ) </pre>	
<pre> MtsZeroOutSigCoeffFlag = 0 </pre>	
<pre> ...} </pre>	

Theo một số phương án, MtsZeroOutSigCoeffFlag được thiết lập ban đầu (ví dụ, được chuyển thành 0 khi MtsZeroOutSigCoeffFlag không bằng 0) dựa vào LastSignificantCoeffX và LastSignificantCoeffY, và MtsZeroOutSigCoeffFlag đến lượt nó được sử dụng để xác định liệu có kiểm tra coded\_sub\_block\_flag flag của các khối con hay không. Bảng cú pháp tương ứng dùng để lập mã phần dư được thể hiện ở Bảng 8 dưới đây.

Bảng 8:

residual_coding( x0, y0, log2TbWidth, log2TbHeight, cIdx ) {	Biểu mô tả
<pre> if( ( ( sps_mts_enabled_flag &amp;&amp; cu_sbt_flag &amp;&amp; log2TbWidth &lt; 6 &amp;&amp; log2TbHeight &lt; 6 ) ) &amp;&amp; cIdx == 0 &amp;&amp; log2TbWidth &gt; 4 ) </pre>	
<pre> log2ZoTbWidth = 4 </pre>	
<pre> else </pre>	
<pre> log2ZoTbWidth = Min( log2TbWidth, 5 ) </pre>	
<pre> if( ( sps_mts_enabled_flag &amp;&amp; cu_sbt_flag &amp;&amp; log2TbWidth &lt; 6 &amp;&amp; log2TbHeight &lt; 6 ) ) &amp;&amp; cIdx == 0 &amp;&amp; log2TbHeight &gt; 4 ) </pre>	
<pre> log2ZoTbHeight = 4 </pre>	
<pre> else </pre>	

$\log_2 Z_{oTbHeight} = \text{Min}(\log_2 TbHeight, 5) \dots$	
$\text{numSbCoeff} = 1 \ll (\log_2 SbW + \log_2 SbH)$	
$\text{lastScanPos} = \text{numSbCoeff}$	
$\text{lastSubBlock} = (1 \ll (\log_2 TbWidth + \log_2 TbHeight - (\log_2 SbW + \log_2 SbH))) - 1$	
do {	
if( lastScanPos == 0 ) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos-- ...	
} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
if( lastSubBlock == 0 && log <sub>2</sub> TbWidth >= 2 && log <sub>2</sub> TbHeight >= 2 && !transform_skip_flag[ x0 ][ y0 ][ cIdx ] && lastScanPos > 0 )	
LfstDcOnly = 0	
if( ( lastSubBlock > 0 && log <sub>2</sub> TbWidth >= 2 && log <sub>2</sub> TbHeight >= 2 )    ( lastScanPos > 7 && ( log <sub>2</sub> TbWidth == 2    log <sub>2</sub> TbWidth == 3 ) && log <sub>2</sub> TbWidth == log <sub>2</sub> TbHeight ) )	

LnstZeroOutSigCoeffFlag = 0	
if( ( LastSignificantCoeffX > 15    LastSignificantCoeffY > 15 ) && cIdx == 0 )	
MtsZeroOutSigCoeffFlag = 0	
QState = 0	
for( i = lastSubBlock; i >= 0; i-- ) {	
startQStateSb = QState	
xS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ] [ i ][ 0 ]	
yS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ] [ i ][ 1 ]	
inferSbDcSigCoeffFlag = 0	
if( i < lastSubBlock && i > 0 ) {	
coded_sub_block_flag[ xS ][ yS ]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
if( MtsZeroOutSigCoeffFlag != 0 && (xS >= 4    yS >=4) && coded_sub_block_flag[ xS ][ yS ] && cIdx == 0 )	
MtsZeroOutSigCoeffFlag = 0	

...}	
------	--

Bảng cú pháp tương ứng được lấy làm ví dụ khác dùng để lập mã phần dư được thể hiện ở Bảng 9 dưới đây.

Bảng 9:

residual_coding( x0, y0, log2TbWidth, log2TbHeight, cIdx ) {	Biến mô tả
if( ( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbWidth > 4 )	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min( log2TbWidth, 5 )	
if( ( sps_mts_enabled_flag && cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cIdx == 0 && log2TbHeight > 4 )	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min( log2TbHeight, 5 ) ...	
numSbCoeff = 1 << ( log2SbW + log2SbH )	
lastScanPos = numSbCoeff	
lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1	

do {	
if( lastScanPos == 0 ) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos-- ...	
} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
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	
QState = 0	
for( i = lastSubBlock; i >= 0; i-- ) {	
startQStateSb = QState	

$xS =$ $\text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}]$ $[i][0]$	
$yS =$ $\text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}]$ $[i][1]$	
$\text{inferSbDcSigCoeffFlag} = 0$	
$\text{if}(i < \text{lastSubBlock} \ \&\& \ i > 0) \{$	
$\text{coded\_sub\_block\_flag}[xS][yS]$	$\text{ae}(v)$
$\text{inferSbDcSigCoeffFlag} = 1$	
$\}$	
$\text{if}(\text{MtsZeroOutSigCoeffFlag} \neq 0 \ \&\& \ (xS \geq 4 \    \ yS \geq 4) \ \&\& \ \text{coded\_sub\_block\_flag}[xS][yS] \ \&\& \ \text{cIdx} == 0)$	
$\text{MtsZeroOutSigCoeffFlag} = 0$	
$\dots\}$	

Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được kết hợp. Bất kỳ các biến thể của các phương pháp trên có thể được thực hiện không tương minh bằng chiều rộng khối hoặc chiều cao khối hoặc diện tích khối, hoặc được thực hiện tương minh bằng cờ được báo hiệu tại phân mức CU, CTU, mảng ảnh, lát ảnh (tile), nhóm lát ảnh (nhóm tile), SPS, hoặc PPS. “Khối” ở sáng chế này có thể có nghĩa là TU/TB/CU/CB/PU/PB. Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực thi ở các bộ mã hóa và/hoặc các bộ giải mã. Ví dụ, bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực thi ở môđun lập mã biến đổi/nội

ảnh/liên ảnh của bộ mã hóa, môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã. Ngoài ra, Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực thi dưới dạng mạch được liên kết với môđun lập mã biến đổi/nội ảnh/liên ảnh của bộ mã hóa và/hoặc môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã.

## II. Bộ mã hóa video được lấy làm ví dụ

FIG. 4 minh họa bộ mã hóa video được lấy làm ví dụ 400. Như được minh họa, bộ mã hóa video 400 nhận tín hiệu video đầu vào từ nguồn video 405 và mã hóa tín hiệu vừa nêu thành dòng bit 495. Bộ mã hóa video 400 có một số bộ phận hoặc môđun dùng để mã hóa tín hiệu từ nguồn video 405, ít nhất bao gồm một số bộ phận được chọn từ môđun biến đổi 410, môđun lượng tử hóa 411, môđun lượng tử hóa ngược 414, môđun biến đổi ngược 415, môđun ước lượng nội ảnh 420, môđun dự đoán nội ảnh 425, môđun bù chuyển động 430, môđun ước lượng chuyển động 435, bộ lọc vòng lặp kín 445, bộ đệm hình ảnh tái tạo 450, bộ đệm MV 465, và môđun dự đoán MV 475, và bộ mã hóa entropy 490. Môđun bù chuyển động 430 và môđun ước lượng chuyển động 435 là một phần của môđun dự đoán liên ảnh 440.

Theo một số phương án, các môđun từ 410 đến 490 là các môđun có các lệnh phần mềm được xử lý bởi một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán hoặc thiết bị điện tử. Theo một số phương án, các môđun từ 410 đến 490 là các môđun của các mạch phần cứng được thực hiện bởi một hoặc nhiều mạch tích hợp IC (Integrated Circuit - IC) của thiết bị điện tử. Mặc dù các môđun từ 410 đến 490 được minh họa dưới dạng các môđun riêng biệt, song một số trong số các môđun đó có thể được kết hợp thành một môđun duy nhất.

Nguồn video 405 cấp tín hiệu video thô thể hiện dữ liệu điểm ảnh của từng khung hình video chưa nén mà không. Bộ trừ 408 tính sai lệch giữa dữ liệu điểm ảnh video thô của nguồn video 405 và dữ liệu điểm ảnh được dự đoán 413 từ môđun bù chuyển động 430 hoặc môđun dự đoán nội ảnh 425. Môđun biến đổi 410 biến sai lệch (hoặc phần dư dữ liệu điểm ảnh hoặc tín hiệu dư thừa 409) thành các hệ số biến đổi (ví dụ, bằng phép biến đổi côsin rời rạc, hay gọi là phép DCT) các hệ số biến đổi. Môđun lượng tử hóa 411 lượng tử hóa các hệ số biến đổi thành dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) các hệ số biến đổi 412, được mã hóa thành dòng bit



495 bởi bộ mã hóa entropy 490.

Môđun lượng tử hóa ngược 414 giải lượng tử dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) 412 để thu được các hệ số biến đổi các hệ số biến đổi, và môđun biến đổi ngược 415 thực hiện phép biến đổi ngược đối với các hệ số biến đổi để tạo ra phần dư tái tạo 419. Phần dư tái tạo 419 được cộng với dữ liệu điểm ảnh được dự đoán 413 để tạo ra dữ liệu điểm ảnh được tái tạo 417. Theo một số phương án, dữ liệu điểm ảnh được tái tạo 417 được lưu tạm thời ở bộ đệm dòng (không được minh họa) để dự đoán nội ảnh và dự đoán MV không gian. Các điểm ảnh được tái tạo được lọc bởi bộ lọc vòng lặp kín 445 và được lưu ở bộ đệm hình ảnh tái tạo 450. Theo một số phương án, bộ đệm hình ảnh tái tạo 450 là bộ nhớ nằm ngoài bộ mã hóa video 400. Theo một số phương án, bộ đệm hình ảnh tái tạo 450 là bộ nhớ nằm trong bộ mã hóa video 400.

Môđun ước lượng nội ảnh 420 thực hiện dự đoán nội ảnh dựa vào dữ liệu điểm ảnh được tái tạo 417 để tạo ra dữ liệu dự đoán nội ảnh. Dữ liệu dự đoán nội ảnh được cấp đến bộ mã hóa entropy 490 để được mã hóa thành dòng bit 495. Dữ liệu dự đoán nội ảnh còn được sử dụng bởi môđun dự đoán nội ảnh 425 để tạo ra dữ liệu điểm ảnh được dự đoán 413.

Môđun ước lượng chuyển động 435 thực hiện dự đoán liên ảnh bằng cách tạo ra các MV để tham chiếu dữ liệu điểm ảnh của các khung hình được giải mã trước đó được lưu ở bộ đệm hình ảnh tái tạo 450. Những MV này được cấp đến môđun bù chuyển động 430 để tạo ra dữ liệu điểm ảnh được dự đoán.

Thay vì mã hóa các MV thực hoàn toàn trong dòng bit, bộ mã hóa video 400 sử dụng dự đoán MV để tạo ra các MV được dự đoán, và sai lệch giữa các MV được sử dụng để bù chuyển động và các MV được dự đoán được mã hóa dưới dạng dữ liệu chuyển động dư thừa và được lưu trong dòng bit 495.

Môđun dự đoán MV 475 tạo ra các MV được dự đoán dựa vào các MV tham chiếu mà được tạo ra để lập mã các khung hình video trước đó, tức là, các MV bù chuyển động được sử dụng để thực hiện bù chuyển động. Môđun dự đoán MV 475 gọi các MV tham chiếu từ các khung hình video trước đó từ bộ đệm MV 465. Bộ mã hóa video 400 lưu các MV được tạo ra cho khung hình video hiện thời ở bộ đệm MV 465 dưới dạng các MV tham chiếu để tạo ra các MV được dự đoán.

Môđun dự đoán MV 475 sử dụng các MV tham chiếu để tạo ra các MV được dự đoán. Các MV được dự đoán có thể được tính theo dự đoán dự đoán MV không gian hoặc dự đoán dự đoán MV thời gian. Sai lệch giữa các MV được dự đoán và các MV bù chuyển động (các MC MV) của khung hình hiện thời (dữ liệu chuyển động dư thừa) được mã hóa thành dòng bit 495 bởi bộ mã hóa entropy 490.

Bộ mã hóa entropy 490 mã hóa nhiều thông số và dữ liệu khác nhau thành dòng bit 495 bằng cách sử dụng các phương pháp lập mã entropy chẳng hạn mã hóa số học nhị phân thích ứng ngữ cảnh CABAC (Context-adaptive Binary Arithmetic Coding - CABAC) hoặc mã hóa Huffman. Bộ mã hóa entropy 490 mã hóa nhiều thành phần tiêu đề, nhiều cờ, cùng với các hệ số chuyển đổi đã lượng tử hóa các hệ số biến đổi 412, và dữ liệu chuyển động dư thừa dưới dạng các thành phần cú pháp thành dòng bit 495. Dòng bit 495 sau đó được lưu trong thiết bị lưu trữ hoặc được truyền đến bộ giải mã thông qua các phương tiện truyền thông chẳng hạn mạng.

Bộ lọc vòng lặp kín 445 thực hiện lọc hoặc làm mượt những phần tử trên dữ liệu điểm ảnh được tái tạo 417 để giảm độ giả tạo của lập mã, đặc biệt là tại các biên giới của các khối điểm ảnh. Theo một số phương án, phép lọc được thực hiện gồm có phép giải khối hoặc phép bù thích ứng mẫu SAO (Sample Adaptive Offset - SAO). Theo một số phương án, những phép lọc bao gồm phép lọc vòng lặp thích ứng ALF (Adaptive Loop Filter - ALF).

FIG. 5 minh họa những bộ phận của bộ mã hóa 400 thực hiện phép chọn đa biến đổi (MTS) bằng cách tùy chọn các phép biến đổi cho phép. Như được minh họa, đối với từng khối của các điểm ảnh, bộ mã hóa 400 xác định liệu có thực hiện dạng biến đổi phụ cụ thể nào hay không (ví dụ, LFNST), liệu có thực hiện dạng biến đổi chính cụ thể nào hay không (ví dụ, DCT-II), liệu có sử dụng MTS nào hay không, và/hoặc các biến đổi nào để sử dụng dưới dạng biến đổi ngang và biến đổi đứng cho MTS. Bộ mã hóa còn xác định báo hiệu cho MTS trong dòng bit 495.

Như được minh họa, môđun biến đổi 410 thực hiện biến đổi chính và/hoặc biến đổi phụ đối với tín hiệu dư 409, và môđun biến đổi ngược 415 thực hiện phép biến đổi chính ngược và/hoặc biến đổi phụ ngược tương ứng. Bộ mã hóa 400 chọn biến đổi chính và/hoặc biến đổi phụ cho môđun biến đổi 410 và môđun biến đổi ngược 415. Biến đổi chính được chọn có thể điều khiển MTS theo biến đổi ngang và biến đổi

đứng.

Bộ mã hóa 400 bao gồm môđun điều khiển lập mã 500. Môđun điều khiển lập mã 500 xác định việc chọn biến đổi chính và/hoặc biến đổi phụ, và nếu MTS sẽ được sử dụng, thì xác định các biến đổi nào sẽ được sử dụng làm các biến đổi ngang và đứng. Môđun điều khiển lập mã 500 có thể thực hiện điều vừa nêu dựa vào tập hợp thông tin khối hiện thời 510 từ môđun biến đổi 409, mà có thể biểu thị các khối con nào của khối hiện thời có các hệ số khác không (hoặc các hệ số có nghĩa lớn hơn giá trị ngưỡng).

Dựa vào thông tin khối hiện thời 510, môđun điều khiển lập mã 500 tạo ra các chỉ báo hệ số khác không tương ứng cho các khối con của khối hiện thời để biểu thị các khối con nào có ít nhất một hệ số khác không (ví dụ, `coded_sub_block_flag`). Môđun điều khiển lập mã 500 có thể còn xác định vị trí của hệ số khác không cuối cùng trong khối hiện thời. Môđun điều khiển lập mã 500 có thể còn tạo ra chỉ số MTS trong đó xác định biến đổi ngang và biến đổi đứng đang được sử dụng cho MTS. Những tín hiệu vừa nêu được cấp đến bộ mã hóa entropy 490 để được chứa trong dòng bit 495.

Bộ mã hóa entropy 490 đến lượt nó sẽ mã hóa các chỉ báo hệ số khác không của các khối con khác nhau thành dòng bit (ví dụ, dưới dạng `coded_sub_block_flag` của các khối con khác nhau). Bộ mã hóa entropy 490 có thể còn báo hiệu vị trí của hệ số khác không cuối cùng (ví dụ, `LastSignificantCoeffX` và `LastSignificantCoeffY`) và chỉ số MTS thành dòng bit 495. Theo một số phương án, bộ mã hóa entropy 490 có thể bỏ qua hoặc không thực hiện việc báo hiệu chỉ số MTS trong dòng bit nếu các chỉ báo hệ số khác không biểu thị rằng có các hệ số khác không ở vùng ngoài điểm không được định bằng các vị trí khối con vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang (ví dụ, vùng ngoài điểm không 300 trên FIG. 3 được định bằng  $xS>3$  hoặc  $yS>3$ ), hoặc nếu vị trí của hệ số khác không cuối cùng ở khối hiện thời biểu thị rằng có hệ số khác không vượt quá khối con nhất định theo thứ tự quét chéo (ví dụ, `LastSignificantCoeffX>15` hoặc `LastSignificantCoeffY>15`).

FIG. 6 minh họa về mặt khái niệm quy trình 600 dùng để sử dụng phép chọn đa biến đổi nhằm mã hóa khối các điểm ảnh. Theo một số phương án, một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán điều khiển bộ mã hóa 400 thực hiện

quy trình 600 bằng cách xử lý các lệnh được lưu ở phương tiên đọc được bằng máy tính. Theo một số phương án, thiết bị điện tử điều khiển bộ mã hóa 400 thực hiện quy trình 600.

Bộ mã hóa nhận (tại khối 610) dữ liệu cho khối các điểm ảnh sẽ được mã hóa dưới dạng khối hiện thời của ảnh hiện thời của video thành dòng bit.

Bộ mã hóa thiết lập (tại khối 620) chỉ báo ngoài điểm không (ví dụ, `MtsZeroOutSigCoeffFlag`) dựa vào các chỉ báo hệ số khác không của các khối con (ví dụ, `coded_sub_block_flag`) của khối hiện thời (ví dụ, vùng ngoài điểm không 300 trên FIG. 3 được định bằng  $xS>3$  hoặc  $yS>3$ ). Theo một số phương án, vùng ngoài điểm không của khối hiện thời là vùng của mỗi khối biến đổi ở khối hiện thời mà vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang (ví dụ,  $xS>3$  hoặc  $yS>3$ ). (Vị trí ngưỡng theo phương ngang và vị trí ngưỡng theo phương đứng là các vị trí của các nhóm lập mã mà mỗi chúng có thể bao gồm 16 hệ số biến đổi). Theo một số phương án, bộ mã hóa kiểm tra các chỉ báo hệ số khác không của các khối con ở vùng ngoài điểm không để xem liệu có bất kỳ các khối con nào ở vùng ngoài điểm không có hệ số khác không hay không. Nếu có, thì bộ mã hóa thiết lập chỉ báo ngoài điểm không để biểu thị có hệ số khác không ở vùng ngoài điểm không hay không. Quy trình này chuyển sang 640.

Bộ mã hóa xác định (tại khối 630) liệu chỉ báo ngoài điểm không biểu thị vùng ngoài điểm không của khối hiện thời có ít nhất một hệ số khác không hay không. Nếu có, thì quy trình vừa nêu chuyển sang 640. Nếu chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, thì bộ mã hóa báo hiệu (tại khối 650) chỉ số MTS thành dòng bit, và quy trình vừa nêu chuyển sang 660.

Tại khối 640, bộ mã hóa thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit. Bộ mã hóa chọn (tại khối 660) dạng biến đổi theo chỉ số MTS cho khối hiện thời. Theo một số phương án, dạng biến đổi xác định các biến đổi ngang và đứng của các khối biến đổi luma liên quan ở khối hiện thời. Theo một số phương án, giá trị mặc định của chỉ số MTS tương ứng với dạng biến đổi mà sử dụng Phép biến đổi cosin rời rạc dạng II (DCT-2) cho cả biến đổi ngang và đứng. Bộ mã hóa mã hóa (tại khối 670) khối hiện thời thành dòng bit bằng cách sử dụng dạng biến đổi được chọn.

### III. Bộ giải mã video được lấy làm ví dụ

FIG. 7 minh họa bộ giải mã video được lấy làm ví dụ 700. Như được minh họa, bộ giải mã video 700 là mạch giải mã ảnh hoặc giải mã video mà nhận dòng bit 795 và giải mã nội dung của dòng bit thành dữ liệu điểm ảnh của các khung hình video để hiển thị. Bộ giải mã video 700 có một số bộ phận hoặc các môđun dùng để giải mã dòng bit 795, bao gồm một số bộ phận được chọn từ môđun lượng tử hóa ngược 711, môđun biến đổi ngược 710, môđun dự đoán nội ảnh 725, môđun bù chuyển động 730, bộ lọc vòng lặp kín 745, bộ đệm hình ảnh được giải mã 750, bộ đệm MV 765, môđun dự đoán MV 775, và bộ phân tích 790. Môđun bù chuyển động 730 là một phần của môđun dự đoán liên ảnh 740.

Theo một số phương án, các môđun 710 đến 790 là các môđun có các lệnh phần mềm được xử lý bằng một hoặc nhiều đơn vị xử lý (ví dụ, bộ xử lý) của thiết bị tính toán. Theo một số phương án, các môđun từ 710 đến 790 là các môđun có các mạch phần cứng được thực hiện bởi một hoặc nhiều IC của thiết bị điện tử. Mặc dù các môđun 710 đến 790 được minh họa dưới dạng các môđun riêng biệt, song một số trong số các môđun đó có thể được kết hợp thành một môđun duy nhất.

Bộ phân tích 790 (hoặc bộ giải mã entropy) nhận dòng bit 795 và thực hiện phép phân tích ban đầu theo cú pháp được định nghĩa bởi tiêu chuẩn lập mã video hoặc tiêu chuẩn lập mã ảnh. Thành phần cú pháp được phân tích bao gồm nhiều các thành phần tiêu đề, các cờ, cũng như dữ liệu được lượng tử hóa (hoặc các hệ số lượng tử) 712. Bộ phân tích 790 phân tích nhiều các thành phần cú pháp bằng cách sử dụng các phương pháp lập mã entropy chẳng hạn mã hóa số học nhị phân thích ứng ngữ cảnh CABAC (Context-adaptive Binary Arithmetic Coding - CABAC) hoặc mã hóa Huffman.

Môđun lượng tử hóa ngược 711 giải lượng tử dữ liệu được lượng tử hóa (hoặc các hệ số được lượng tử hóa) 712 các hệ số biến đổi để thu được các hệ số biến đổi, và môđun biến đổi ngược 710 thực hiện phép biến đổi ngược đối với các hệ số biến đổi 716 để tạo ra tín hiệu dư được tái tạo 719. Tín hiệu dư tái tạo 719 được cộng với dữ liệu điểm ảnh được dự đoán 713 từ môđun dự đoán nội ảnh 725 hoặc môđun bù chuyển động 730 để tạo ra dữ liệu điểm ảnh được giải mã 717. Dữ liệu điểm ảnh được giải mã được lọc bởi bộ lọc vòng lặp kín 745 và được lưu ở bộ đệm hình ảnh được giải mã 750. Theo một số phương án, bộ đệm hình ảnh được giải mã 750 là bộ nhớ nằm

ngoài bộ giải mã video 700. Theo một số phương án, bộ đệm hình ảnh được giải mã 750 là bộ nhớ nằm trong bộ giải mã video 700.

Môđun dự đoán nội ảnh 725 nhận dữ liệu dự đoán nội ảnh từ dòng bit 795 và theo đó, tạo ra dữ liệu điểm ảnh được dự đoán 713 từ dữ liệu điểm ảnh được giải mã 717 được lưu ở bộ đệm hình ảnh được giải mã 750. Theo một số phương án, dữ liệu điểm ảnh được giải mã 717 còn được lưu ở bộ đệm dòng (không được minh họa) để dự đoán nội ảnh và dự đoán MV không gian.

Theo một số phương án, nội dung của bộ đệm hình ảnh được giải mã 750 được sử dụng để hiển thị. Thiết bị hiển thị 755 hoặc truy xuất nội dung của bộ đệm hình ảnh được giải mã 750 để hiển thị trực tiếp, hoặc gọi nội dung của bộ đệm hình ảnh được giải mã cho bộ đệm hiển thị. Theo một số phương án, thiết bị hiển thị nhận các giá trị điểm ảnh bộ đệm hình ảnh được giải mã 750 thông qua bộ truyền điểm ảnh.

Môđun bù chuyển động 730 tạo ra dữ liệu điểm ảnh được dự đoán 713 từ dữ liệu điểm ảnh được giải mã 717 được lưu ở bộ đệm hình ảnh được giải mã 750 theo các MV bù chuyển động (các MC MV). Các MV bù chuyển động này được giải mã bằng cách cộng dữ liệu chuyển động dư thừa được nhận từ dòng bit 795 với các MV được dự đoán được nhận từ môđun dự đoán MV 775.

Môđun dự đoán MV 775 tạo ra các MV được dự đoán dựa vào các MV tham chiếu mà được tạo ra để giải mã các khung hình video trước đó, ví dụ, các MV bù chuyển động được sử dụng để thực hiện bù chuyển động. Môđun dự đoán MV 775 gọi các MV tham chiếu của các khung hình video trước đó từ bộ đệm MV 765. Bộ giải mã video 700 lưu các MV bù chuyển động được tạo ra để giải mã khung hình video hiện thời ở bộ đệm MV 765 thành các MV tham chiếu để tạo ra các MV được dự đoán.

Bộ lọc vòng lặp kín 745 thực hiện các thuật toán lọc hoặc làm mượt dữ liệu điểm ảnh được giải mã 717 để giảm độ giả tạo của lập mã, đặc biệt là tại các biên giới của các khối điểm ảnh. Theo một số phương án, phép lọc được thực hiện gồm có phép giải khối hoặc phép bù thích ứng mẫu SAO (Sample Adaptive Offset - SAO). Theo một số phương án, những phép lọc bao gồm phép lọc vòng lặp thích ứng ALF (Adaptive Loop Filter - ALF).

FIG. 8 minh họa những bộ phận của bộ giải mã 700 thực hiện phép chọn đa biến đổi (MTS) bằng cách tùy chọn các phép biến đổi cho phép. Như được minh họa, đổi

với từng khối của các điểm ảnh, bộ giải mã 700 xác định liệu có thực hiện dạng biến đổi phụ cụ thể nào hay không, liệu có thực hiện dạng biến đổi chính cụ thể nào hay không (ví dụ, DCT-II), liệu có sử dụng MTS nào hay không, và/hoặc các biến đổi nào để sử dụng dưới dạng biến đổi ngang và biến đổi đứng cho MTS. Bộ giải mã còn xác định báo hiệu cho MTS trong dòng bit 795.

Như được minh họa, bộ giải mã 700 chọn biến đổi chính và/hoặc biến đổi phụ cho môđun biến đổi ngược 715, và môđun biến đổi ngược 715 thực hiện phép biến đổi chính ngược và/hoặc biến đổi phụ ngược tương ứng. Biến đổi chính được chọn có thể điều khiển MTS theo biến đổi ngang và biến đổi đứng.

Bộ giải mã 700 bao gồm môđun điều khiển lập mã 800. Môđun điều khiển lập mã 800 xác định việc chọn biến đổi chính và/hoặc biến đổi phụ, và nếu MTS sẽ được sử dụng, thì xác định các biến đổi nào sẽ được sử dụng làm các biến đổi ngang và đứng. Môđun điều khiển lập mã 800 có thể ánh xạ chỉ số MTS được cấp bởi bộ giải mã entropy 790 thành phép chọn các biến đổi ngang và đứng tương ứng.

Bộ giải mã entropy 790 phân tích dòng bit 795 để xác định trị số của chỉ số MTS. Dòng bit có thể bao gồm các thành phần cú pháp trong đó biểu thị các khối con nào của khối hiện thời có các hệ số khác không (ví dụ, `coded_sub_block_flag`), và/hoặc các thành phần cú pháp trong đó xác định vị trí của hệ số khác không cuối cùng ở khối hiện thời. Dựa vào các thành phần cú pháp này, bộ giải mã entropy có thể xác định liệu có phân tích dòng bit cho chỉ số MTS hay không. Cụ thể là, bộ giải mã entropy 790 có thể không thực hiện hoặc bỏ qua phân tích dòng bit cho chỉ số MTS nếu vị trí của hệ số khác không cuối cùng ở khối hiện thời biểu thị rằng có hệ số khác không vượt quá khối con nhất định theo thứ tự quét chéo (ví dụ, `LastSignificantCoeffX > 15` hoặc `LastSignificantCoeffY > 15`), hoặc nếu các chỉ báo hệ số khác không biểu thị rằng có các hệ số khác không ở vùng ngoài điểm không được định bằng các vị trí khối con vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang (ví dụ, vùng 300 trên FIG. 3 được định bằng  $xS > 3$  hoặc  $yS > 3$ ). Nếu bộ giải mã entropy 790 không phân tích dòng bit cho chỉ số MTS, thì bộ giải mã entropy 790 thiết lập trị số của chỉ số MTS thành 0 cho môđun điều khiển lập mã 800.

FIG. 9 minh họa về mặt khái niệm quy trình 900 để sử dụng phép chọn đa biến đổi nhằm giải mã khối các điểm ảnh. Theo một số phương án, một hoặc nhiều đơn vị

xử lý (ví dụ, bộ xử lý) của thiết bị tính toán điều khiển bộ giải mã 700 thực hiện quy trình 900 bằng cách xử lý các lệnh được lưu ở phương tiên đọc được bằng máy tính. Theo một số phương án, thiết bị điện tử điều khiển bộ giải mã 700 thực hiện quy trình 900.

Bộ giải mã nhận (tại khối 910) dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video.

Bộ giải mã thiết lập (tại khối 920) chỉ báo ngoài điểm không (ví dụ, `MtsZeroOutSigCoeffFlag`) dựa vào các chỉ báo hệ số khác không (ví dụ, `coded_sub_block_flag`) của các khối con của khối hiện thời (ví dụ, vùng 300 trên FIG. 3 được định bằng  $xS > 3$  hoặc  $yS > 3$ ). Theo một số phương án, vùng ngoài điểm không của khối hiện thời là vùng của mỗi khối biến đổi trong khối hiện thời mà vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang (ví dụ,  $xS > 3$  hoặc  $yS > 3$ ). Vị trí ngưỡng theo phương ngang và vị trí ngưỡng theo phương đứng là các vị trí ( $xS, yS$ ) của các nhóm lập mã trong khối biến đổi hiện thời (hoặc khối con đã nêu tại vị trí ( $xS, yS$ ) nằm trong khối biến đổi hiện thời). Ví dụ, khối con bao gồm 16 hệ số biến đổi. Nói cách khác, khối con bao gồm các hệ số biến đổi  $4 \times 4$ . Theo một số phương án, bộ giải mã kiểm tra các chỉ báo hệ số khác không của các khối con để xem nếu bất kỳ các khối con nào ở vùng ngoài điểm không của khối hiện thời có hệ số khác không. Nếu có, thì bộ giải mã thiết lập chỉ báo ngoài điểm không để biểu thị có ít nhất một hệ số khác không ở vùng ngoài điểm không của khối hiện thời. Quy trình này chuyển sang 940.

Bộ giải mã xác định (tại khối 930) liệu chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không của khối hiện thời có ít nhất một hệ số khác không hay không. Nếu có, quy trình này chuyển sang 940. Nếu chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không của khối hiện thời không có hệ số khác không (nếu MTS khác báo hiệu những điều kiện được thỏa mãn), thì bộ giải mã phân tích (tại khối 950) dòng bit cho chỉ số MTS và quy trình này chuyển sang 960.

Tại khối 940, bộ giải mã thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit. Bộ giải mã chọn (tại khối 960) dạng biến đổi theo chỉ số MTS cho khối hiện thời. Theo một số phương án, dạng biến đổi xác định các biến đổi ngang và đứng của các khối biến đổi luma liên quan ở khối hiện thời. Theo



một số phương án, giá trị mặc định của chỉ số MTS tương ứng với dạng biến đổi mà sử dụng Phép biến đổi côsin rời rạc dạng II (DCT-2) cho cả biến đổi ngang và đứng. Bộ giải mã khôi phục (tại khối 970) khôi hiện thời bằng cách sử dụng dạng biến đổi được chọn.

Theo một số phương án, bộ mã hóa có thể báo hiệu (hoặc tạo ra) một hoặc nhiều thành phần cú pháp trong dòng bit, sao cho bộ giải mã có thể phân tích một hoặc nhiều thành phần cú pháp vừa nêu từ dòng bit.

#### IV. Hệ thống điện tử được lấy làm ví dụ

Rất nhiều các dấu hiệu và ứng dụng được mô tả ở trên được thực hiện dưới dạng các xử lý phần mềm mà được chỉ dẫn dưới dạng tập hợp các lệnh được ghi vào phương tiện lưu trữ có thể đọc được bằng máy tính (còn được gọi là phương tiện có thể đọc được bằng máy tính). Khi các lệnh này được xử lý bởi một hoặc nhiều bộ xử lý hoặc bộ tính toán (ví dụ, một hoặc nhiều bộ xử lý, lõi của các bộ xử lý, hoặc các bộ xử lý khác), chúng giúp cho bộ xử lý (các bộ xử lý) thực hiện những chức năng được chỉ thị trong những câu lệnh. Những ví dụ về các phương tiện đọc được bằng máy tính bao gồm, nhưng không bị giới hạn chỉ ở, các đĩa CD-ROM, các ổ đĩa nhanh, các vi mạch bộ nhớ ngẫu nhiên RAM (Random-Access Memory - RAM), các ổ cứng, các bộ nhớ chỉ đọc chương trình có thể xóa được (Erasable Programmable Read Only Memory - EPROM), các bộ nhớ chỉ đọc chương trình có thể xóa được bằng điện (Electrically Erasable Programmable Read-Only Memories - EEPROM), .v.v. Các phương tiện đọc được bằng máy tính không bao gồm các sóng truyền thông và các tín hiệu điện truyền kết nối có dây hoặc không dây.

Ở bản mô tả này, thuật ngữ “phần mềm” được hiểu là bao gồm phần vi chương trình nằm trong bộ nhớ chỉ đọc hoặc các ứng dụng được lưu được lưu trong bộ nhớ từ mà có thể có thể được ghi vào bộ nhớ để xử lý bởi khối xử lý. Tương tự, theo một số phương án, nhiều phát minh phần mềm có thể còn được sử dụng dưới dạng các chương trình con của một chương trình lớn hơn trong khi giữ lại các kỹ thuật phần mềm riêng. Theo một số phương án, nhiều kỹ thuật phần mềm có thể cũng được sử dụng dưới dạng các chương trình riêng biệt. Cuối cùng, bất kỳ sự kết hợp nào giữa các chương trình riêng biệt mà cùng thực hiện kỹ thuật phần mềm được mô tả ở đây đều nằm trong phạm vi của sáng chế. Theo một số phương án, các chương trình phần mềm,

khi được cài đặt để vận hành một hoặc nhiều hệ thống điện tử, định ra một hoặc nhiều thiết bị thực thi chuyên biệt để xử lý và thực hiện các phép toán của các chương trình phần mềm.

FIG.10 minh họa về mặt ý tưởng hệ thống điện tử 1000 qua đó một số phương án của sáng chế được thực hiện. Hệ thống điện tử 1000 có thể là máy tính (ví dụ, máy tính để bàn, máy tính cá nhân, máy tính bảng, .v.v), điện thoại, PDA, hoặc bất kỳ loại thiết bị điện tử nào. Hệ thống điện tử vừa nêu gồm có nhiều loại phương tiện có thể đọc được bằng máy tính và nhiều giao diện dùng cho nhiều loại phương tiện có thể đọc được bằng máy tính đó. Hệ thống điện tử 1000 bao gồm bus 1005, bộ xử lý (các bộ xử lý) 1010, bộ xử lý đồ họa GPU (Graphic Processing Unit - GPU) 1015, bộ nhớ hệ thống 1020, mạng 1025, bộ nhớ chỉ đọc 1030, thiết bị lưu trữ cố định 1035, các thiết bị nhập 1040, và các thiết bị xuất 1045.

Bus 1005 là thuật ngữ chỉ tập hợp toàn bộ hệ thống, thiết bị ngoại vi, và các bus vi mạch mà kết nối truyền thông nhiều thiết bị nội bộ của hệ thống điện tử 1000. Ví dụ, bus 1005 kết nối truyền thông bộ xử lý (các bộ xử lý) 1010 với GPU 1015, bộ nhớ chỉ đọc 1030, bộ nhớ hệ thống 1020, và thiết bị lưu trữ cố định 1035.

Từ những bộ nhớ vừa nêu, bộ xử lý (các bộ xử lý) 1010 gọi ra các lệnh xử lý và dữ liệu nhằm xử lý các quá trình xử lý của sáng chế. Bộ xử lý (các bộ xử lý) có thể là bộ xử lý đơn nhân hoặc bộ xử lý đa nhân theo các phương án khác nhau. Một số lệnh được phân tích và được xử lý bởi GPU 1015. GPU 1015 có thể nạp ngoại tuyến nhiều phép tính toán hoặc bổ sung phép xử lý ảnh được cấp bởi bộ xử lý (các bộ xử lý) 1010.

Bộ nhớ chỉ đọc ROM (Read Only Memory - ROM) 1030 lưu các dữ liệu cố định và các lệnh được sử dụng bởi bộ xử lý (các bộ xử lý) 1010 và các môđun khác của hệ thống điện tử. Mặt khác, thiết bị lưu trữ cố định 1035 mặt khác là thiết bị nhớ đọc và ghi được. Thiết bị này là bộ nhớ không khả biến để lưu các lệnh và dữ liệu ngay cả khi hệ thống điện tử 1000 không hoạt động. Một số phương án của sáng chế sử dụng bộ nhớ dung lượng cao (chẳng hạn đĩa từ hoặc đĩa quang và ổ đĩa tương ứng của nó) làm thiết bị lưu trữ cố định 1035.

Các phương án khác sử dụng thiết bị lưu di động (chẳng hạn đĩa mềm, thiết bị nhớ cực nhanh, .v.v., và ổ đĩa tương ứng của nó) làm thiết bị lưu trữ cố định. Tương tự thiết bị lưu trữ cố định 1035, bộ nhớ hệ thống 1020 là thiết bị nhớ đọc và ghi được.

Tuy nhiên, không giống thiết bị lưu trữ 1035, bộ nhớ hệ thống 1020 là bộ nhớ đọc và ghi khả biến, chẳng hạn bộ nhớ truy cập ngẫu nhiên. Bộ nhớ hệ thống 1020 lưu một số câu lệnh và dữ liệu mà bộ xử lý cần để khi vận hành. Theo một số phương án, những quy trình xử lý của sáng chế được lưu trong bộ nhớ hệ thống 1020, thiết bị lưu trữ cố định 1035, và/hoặc bộ nhớ chỉ đọc 1030. Ví dụ, nhiều bộ nhớ khác nhau chứa các lệnh để xử lý các đoạn phim đa phương tiện theo một số phương án. Từ những bộ nhớ vừa nêu, bộ xử lý (các bộ xử lý) 1010 gọi ra các lệnh xử lý và dữ liệu nhằm xử lý các quá trình xử lý của một số phương án.

Bus 1005 còn kết nối với các thiết bị nhập và xuất 1040 và 1045. Các thiết bị nhập 1040 cho phép người sử dụng truyền thông tin và chọn các lệnh cho hệ thống điện tử. Các thiết bị nhập 1040 bao gồm các bàn phím chữ-số và các thiết bị trỏ (còn được gọi “các thiết bị điều khiển con trỏ”), các máy quay (ví dụ, các webcam), các ống nói hoặc các thiết bị tương tự để nhận các lệnh âm thanh, v.v. Các thiết bị xuất 1045 hiển thị các hình ảnh được tạo ra bởi hệ thống điện tử hoặc nếu không thì hiển thị dữ liệu đầu ra. Các thiết bị xuất 1045 bao gồm các máy in và các thiết bị hiển thị, chẳng hạn các màn hình ống tia catốt CRT (Cathode Ray Tube - CRT) hoặc màn hình tinh thể lỏng (Liquid Crystal Display - LCD), cũng như các loại loa hoặc các thiết bị xuất âm thanh tương tự. Một số phương án bao gồm các thiết bị chẳng hạn màn hình cảm ứng có chức năng làm cả thiết bị nhập và thiết bị xuất.

Cuối cùng, như được thể hiện trên FIG.10, bus 1005 còn kết nối hệ thống điện tử 1000 vào mạng 1025 thông qua các mạng (không được thể hiện). Theo cách thức vừa nêu, máy tính có thể làm một thành phần của mạng máy tính (chẳng hạn mạng máy tính cục bộ LAN (“Local Area Network - LAN”), mạng diện rộng (“Wide Area Network - WAN”), hoặc mạng nội bộ Intranet, hoặc mạng của các mạng, chẳng hạn mạng Internet. Bất kỳ hoặc tất cả các bộ phận của hệ thống điện tử 1000 có thể được sử dụng theo sáng chế.

Một số phương án bao gồm các thiết bị điện tử, chẳng hạn các bộ vi xử lý, bộ lưu trữ và bộ nhớ để lưu các lệnh chương trình máy tính trong phương tiện có thể đọc được bằng máy hoặc có thể đọc được bằng máy tính (theo cách khác còn được gọi là các phương tiện có thể đọc được bằng máy tính, các phương tiện có thể đọc được bằng máy, hoặc các phương tiện lưu trữ có thể đọc được bằng máy). Một số ví dụ về phương tiện có thể đọc được bằng máy tính bao gồm RAM, ROM, các đĩa nén chỉ đọc

(CD-ROM), các đĩa nén có thể ghi (CD-R), các đĩa nén có thể ghi lại (CD-RW), các đĩa đa năng số DVD chỉ đọc (ví dụ, đĩa DVD-ROM, đĩa hai lớp DVD-ROM), nhiều loại đĩa DVD có thể ghi/ghi lại (ví dụ, đĩa DVD-RAM, đĩa DVD-RW, đĩa DVD+RW, v.v), bộ nhớ cực nhanh (ví dụ, các cạc SD, các cạc mini-SD, các cạc micro-SD, v.v), các ổ cứng từ và/hoặc các ổ cứng trạng thái rắn, các đĩa Blu-Ray® chỉ đọc và có thể ghi được, các đĩa quang siêu mật độ, bất kỳ phương tiện từ hoặc quang khác, và các đĩa mềm. Phương tiện có thể đọc được bằng máy tính có thể lưu chương trình máy tính mà có thể xử lý được bởi ít nhất một bộ xử lý và gồm có bộ các lệnh dùng để thực hiện nhiều phép tính khác nhau. Các ví dụ về các chương trình máy tính hoặc mã máy tính bao gồm mã máy, chẳng hạn được tạo ra bởi trình biên dịch, và các tệp tin bao gồm mã bậc cao được xử lý bởi máy tính, thiết bị điện tử, hoặc bộ vi xử lý bằng cách sử dụng trình biên dịch.

Mặc dù phần trình bày ở trên chủ yếu đề cập đến bộ vi xử lý hoặc các bộ vi xử lý đa nhân để xử lý phần mềm, song rất nhiều các dấu hiệu và ứng dụng được mô tả ở trên được thực hiện bởi một hoặc nhiều vi mạch tích hợp, chẳng hạn các vi mạch tích hợp chuyên dụng ASIC (Application Specific Integrated Circuit - ASIC) hoặc vi mạch tích hợp FPGA (Field Programmable Gate Array - FPGA). Theo một số phương án, các mạch tích hợp vừa nêu xử lý các lệnh được lưu trên chính mạch đó. Ngoài ra, một số phương án xử lý phần mềm được lưu trong các thiết bị logic lập trình được PLD (Programmable Logic Device - PLD), ROM, hoặc RAM.

Như được sử dụng ở bản mô tả này và bất kỳ điểm yêu cầu bảo hộ của sáng chế, các thuật ngữ “máy tính”, “máy chủ”, “bộ xử lý”, và “bộ nhớ” toàn bộ chúng đề cập đến các thiết bị điện tử hoặc các thiết bị công nghệ khác. Những thuật ngữ vừa nêu không bao gồm người hoặc nhóm người. Nhằm những mục đích của bản mô tả này, các thuật ngữ hiển thị hoặc biểu thị mang nghĩa là hiển thị trên thiết bị điện tử. Như được sử dụng ở bản mô tả này và bất kỳ điểm yêu cầu bảo hộ của sáng chế, các thuật ngữ “phương tiện đọc được bằng máy tính,” “các phương tiện đọc được bằng máy tính,” và “phương tiện đọc được bằng máy” tất cả đều được giới hạn là các đối tượng vật lý có tính hữu hình mà lưu thông tin dưới dạng có thể đọc được bằng máy tính. Những thuật ngữ này không bao gồm bất kỳ tín hiệu truyền không dây, tín hiệu truyền theo dây, và bất kỳ tín hiệu nào khác.

Mặc dù sáng chế vừa được mô tả theo nhiều phương án chi tiết cụ thể, song một

người trình độ trung bình trong lĩnh vực kỹ thuật này sẽ nhận ra rằng sáng chế có thể được thực hiện theo các cách thức cụ thể khác nhưng không trệch khỏi nội dung của sáng chế. Ngoài ra, số lượng cách hình vẽ (bao gồm FIG. 6 và FIG. 9) minh họa về mặt khái niệm các quy trình. Các phần tử cụ thể của những quy trình này có thể không được thực hiện được thực hiện theo thứ tự chính xác như được thể hiện và được mô tả. Những phần tử cụ thể đó có thể không được thực hiện theo chuỗi liên tiếp của các phần tử, và những phần tử cụ thể khác có thể được thực hiện in các phương án khác. Ngoài ra, quy trình vừa nêu có thể được thực hiện bằng cách sử dụng một số quy trình con, hoặc một phần của quy trình macrô lớn hơn. Do đó, một người có kiến thức trung bình trong lĩnh vực kỹ thuật này sẽ hiểu rằng sáng chế không bị giới hạn bởi phần mô tả chi tiết minh họa được trình bày ở trên, mà sáng chế được định rõ theo những điểm yêu cầu bảo hộ đính kèm.

#### Những chú ý phụ

Đối tượng được mô tả ở bản mô tả này đôi khi minh họa các thành phần khác chứa trong, hoặc hoặc được kết nối với, các thành phần khác nữa. Sẽ được hiểu rằng những cấu trúc được mô tả như vậy chỉ là các ví dụ, và rằng trong thực tế rất nhiều cấu trúc khác có thể được thực hiện để đạt được chức năng tương tự. Về mặt khái niệm, bất kỳ sự bố trí nào của các thành phần nhằm đạt được chức năng tương tự đều "được kết hợp" hiệu quả. Vì vậy, bất kỳ hai thành phần nào ở đây được phối hợp nhằm đạt được chức năng cụ thể đều có thể được xem là "được kết hợp" với nhau sao cho chức năng mong muốn đạt được, bất kể đến các cấu trúc hoặc các thành phần trung gian. Tương tự, bất kỳ hai thành phần nào được kết hợp như vậy có thể còn được coi là "được kết nối hoạt động được", hoặc "được liên kết hoạt động được", với nhau nhằm đạt được chức năng mong muốn, và bất kỳ hai thành phần nào có khả năng được kết hợp như vậy có thể còn được coi là "có thể liên kết hoạt động được", với nhau nhằm đạt được chức năng mong muốn. Các ví dụ cụ thể về liên kết hoạt động được bao gồm nhưng không bị giới hạn chỉ ở các thành phần có thể liên kết vật lý với nhau và/hoặc có thể tương tác vật lý với nhau và/hoặc các thành phần có thể tương tác không dây với nhau và/hoặc có thể liên kết không dây với nhau và/hoặc các thành phần có thể tương tác logic với nhau và/hoặc có thể liên kết logic với nhau.

Ngoài ra, liên quan đến việc sử dụng bất kỳ các thuật ngữ mang nghĩa số nhiều và/hoặc số ít ở bản mô tả này, những người có chuyên môn trong cùng lĩnh vực kỹ

thuật này có thể dịch từ số nhiều sang số ít và/hoặc từ số ít sang số nhiều miễn là phù hợp với ngữ cảnh và/hoặc ứng dụng. Nhiều phép hoán vị số ít/số nhiều khác nhau có thể được trình bày rõ ràng ở đây nhằm giúp cho hiểu rõ.

Ngoài ra, sẽ được hiểu bởi những chuyên gia trong cùng lĩnh vực kỹ thuật này rằng, nhìn chung, các thuật ngữ được sử dụng ở bản mô tả này, và đặc biệt là trong các điểm yêu cầu bảo hộ đính kèm, ví dụ, các phần chính của các điểm yêu cầu bảo hộ đính kèm, thường được hiểu là các thuật ngữ “mở”, ví dụ, thuật ngữ “bao gồm” nên được hiểu dưới dạng “bao gồm nhưng không bị giới hạn chỉ ở,” thuật ngữ “có” nên được hiểu dưới dạng “có ít nhất,” thuật ngữ “gồm có” nên được hiểu dưới dạng “gồm có nhưng không bị giới hạn chỉ có,” v.v. Sẽ được hiểu rõ hơn bởi những người làm trong cùng lĩnh vực kỹ thuật này rằng nếu có ý định miêu tả số lượng chính xác các điểm yêu cầu bảo hộ, thì ý định đó sẽ được nêu rõ trong phần yêu cầu bảo hộ, và nếu không có miêu vừa nêu thì không có ý định trên. Ví dụ, như một cách để hiểu, các điểm yêu cầu bảo hộ đính kèm dưới đây có thể chứa các cụm từ giới thiệu sau “ít nhất một” và “một hoặc nhiều” để miêu tả yêu cầu bảo hộ. Tuy nhiên, việc sử dụng các cụm từ như vậy sẽ không được hiểu là ám chỉ rằng việc miêu tả yêu cầu bảo hộ bằng mạo từ không xác định “một” sẽ giới hạn bất kỳ yêu cầu bảo hộ cụ thể nào đều chứa phần mô tả yêu cầu bảo hộ được giới thiệu đối với các phương án thực hiện chỉ chứa một miêu tả như vậy, ngay cả khi yêu cầu bảo hộ giống nhau gồm có các cụm từ giới thiệu “một hoặc nhiều” hoặc “ít nhất một” và mạo từ không xác định chẳng hạn “một” nên được hiểu là có nghĩa “ít nhất một” hoặc “một hoặc nhiều;” điều tương tự cũng đúng khi sử dụng các mạo từ xác định để mô tả các yêu cầu bảo hộ. Ngoài ra, thậm chí nếu số lượng chính xác yêu cầu bảo hộ được liệt kê rõ ràng thì những chuyên gia trong cùng lĩnh vực kỹ thuật này sẽ nhận ra rằng số lượng yêu cầu bảo hộ như vậy sẽ được hiểu là mang nghĩa là giá trị tối thiểu số được liệt kê, ví dụ, liệt kê nguyên bản là “hai liệt kê,” mà không có sửa đổi khác, có nghĩa là ít nhất hai liệt kê, hoặc hai hoặc nhiều hơn hai liệt kê. Ngoài ra, ở những trường hợp trong đó thuật ngữ quy ước “ít nhất một trong số A, B, và C, v.v.” được sử dụng, nhìn chung cách sử dụng thuật ngữ như vậy về mặt ý nghĩa sẽ được một người có kiến thức trong lĩnh vực kỹ thuật này hiểu thuật ngữ quy ước đó, ví dụ, “hệ thống có ít nhất một trong số A, B, và C” sẽ bao gồm nhưng không bị giới hạn chỉ ở các hệ thống có chỉ riêng A, chỉ riêng B, chỉ riêng C, A và B cùng nhau, A và C cùng nhau, B và C cùng nhau, và/hoặc A, B, và C cùng nhau,

v.v. Ở những trường hợp trong đó thuật ngữ quy ước “ít nhất một trong số A, B, hoặc C, v.v.” được sử dụng, nhìn chung cách sử dụng thuật ngữ như vậy về mặt ý nghĩa sẽ được một người có kiến thức trong lĩnh vực kỹ thuật này hiểu thuật ngữ quy ước đó, ví dụ, “hệ thống có ít nhất một trong số A, B, hoặc C” sẽ bao gồm nhưng không bị giới hạn chỉ ở các hệ thống có chỉ riêng A, chỉ riêng B, chỉ riêng C, A và B cùng nhau, A và C cùng nhau, B và C cùng nhau, và/hoặc A, B, và C cùng nhau, v.v. Sẽ được hiểu rõ hơn bởi những người làm trong cùng lĩnh vực kỹ thuật này rằng hầu như bất kỳ từ và/hoặc cụm từ phân biệt điển đạt hai hoặc nhiều hơn hai thuật ngữ có thể chọn khả năng, liệu trong phần mô tả, các điểm yêu cầu bảo hộ, hoặc các hình vẽ, sẽ được hiểu để suy ra những khả năng bao gồm một trong số các thuật ngữ, hoặc một trong các thuật ngữ đó, hay bao gồm cả hai thuật ngữ đó. Ví dụ, cụm từ “A hoặc B” sẽ được hiểu là bao gồm những khả năng sau “A” hoặc “B” hoặc “A và B”.

Từ những gì được trình bày ở trên, sẽ được hiểu rằng nhiều phương án thực hiện sáng chế được mô tả ở đây nhằm các mục đích minh họa, và rằng nhiều sửa đổi có thể được thực hiện mà không trệch khỏi phạm vi và nội dung của sáng chế. Theo đó, các phương án thực hiện khác nhau được bộc lộ ở bản mô tả này không nhằm giới hạn sáng chế, và phạm vi và nội dung sáng chế được định rõ bởi các điểm yêu cầu bảo hộ dưới đây.

## YÊU CẦU BẢO HỘ

### 1. Phương pháp giải mã video bao gồm:

nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video;

thiết lập chỉ báo ngoài điểm không dựa vào một hoặc nhiều chỉ báo hệ số khác không của các khối con của khối hiện thời;

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, phân tích dòng bit cho chỉ số chọn phép đa biến đổi (MTS);

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không có ít nhất một hệ số khác không, thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit;

chọn dạng biến đổi theo chỉ số MTS cho khối hiện thời; và

khôi phục khối hiện thời bằng cách sử dụng dạng biến đổi được chọn.

2. Phương pháp giải mã video theo điểm 1, trong đó một hoặc nhiều chỉ báo hệ số khác không của các khối con được sử dụng để biểu thị liệu các khối con của khối hiện thời, được xác định nằm ở vùng ngoài điểm không của khối hiện thời, có một hay nhiều hệ số khác không hay không.

3. Phương pháp giải mã video theo điểm 1, trong đó dạng biến đổi xác định các biến đổi ngang và đứng của các khối biến đổi luma liên quan ở khối hiện thời.

4. Phương pháp giải mã video theo điểm 1, trong đó vùng ngoài điểm không của khối hiện thời là vùng của khối hiện thời được định bằng các vị trí khối con vượt quá vị trí ngưỡng theo phương đứng hoặc vị trí ngưỡng theo phương ngang.

5. Phương pháp giải mã video theo điểm 4, trong đó vị trí ngưỡng theo phương ngang và vị trí ngưỡng theo phương đứng là các vị trí của các khối con.

6. Phương pháp giải mã video theo điểm 1, trong đó giá trị mặc định tương ứng với dạng biến đổi mà sử dụng phép biến đổi cosin rời rạc dạng II (DCT-2) cho cả biến đổi ngang và đứng.

7. Phương pháp giải mã video theo điểm 1, trong đó biến đổi không phân tách tần số thấp (LFNST) được vô hiệu hóa đối với khối hiện thời.



#### 8. Phương pháp mã hóa video bao gồm:

nhận dữ liệu cho khối các điểm ảnh sẽ được mã hóa dưới dạng khối hiện thời của ảnh hiện thời của video thành dòng bit;

thiết lập chỉ báo ngoài điểm không dựa vào một hoặc nhiều chỉ báo hệ số khác không của các khối con của khối hiện thời;

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, báo hiệu chỉ số phép chọn đa biến đổi (MTS) trong dòng bit;

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không có ít nhất một hệ số khác không, thiết lập chỉ số MTS thành giá trị mặc định mà không báo hiệu chỉ số MTS trong dòng bit;

chọn dạng biến đổi theo chỉ số MTS cho khối hiện thời; và

mã hóa khối hiện thời thành dòng bit bằng cách sử dụng dạng biến đổi được chọn.

#### 9. Thiết bị điện tử, bao gồm:

mạch giải mã video được cấu hình để thực hiện các chức năng sau:

nhận dữ liệu từ dòng bit cho khối các điểm ảnh sẽ được giải mã thành khối hiện thời của ảnh hiện thời của video;

thiết lập chỉ báo ngoài điểm không dựa vào một hoặc nhiều chỉ báo hệ số khác không của các khối con của khối hiện thời;

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không không có hệ số khác không, phân tích dòng bit cho chỉ số chọn phép đa biến đổi (MTS);

khi chỉ báo ngoài điểm không biểu thị rằng vùng ngoài điểm không có ít nhất một hệ số khác không, thiết lập chỉ số MTS thành giá trị mặc định mà không phân tích chỉ số MTS từ dòng bit;

chọn dạng biến đổi theo chỉ số MTS cho khối hiện thời; và

khôi phục khối hiện thời bằng cách sử dụng dạng biến đổi được chọn.

FIG. 1

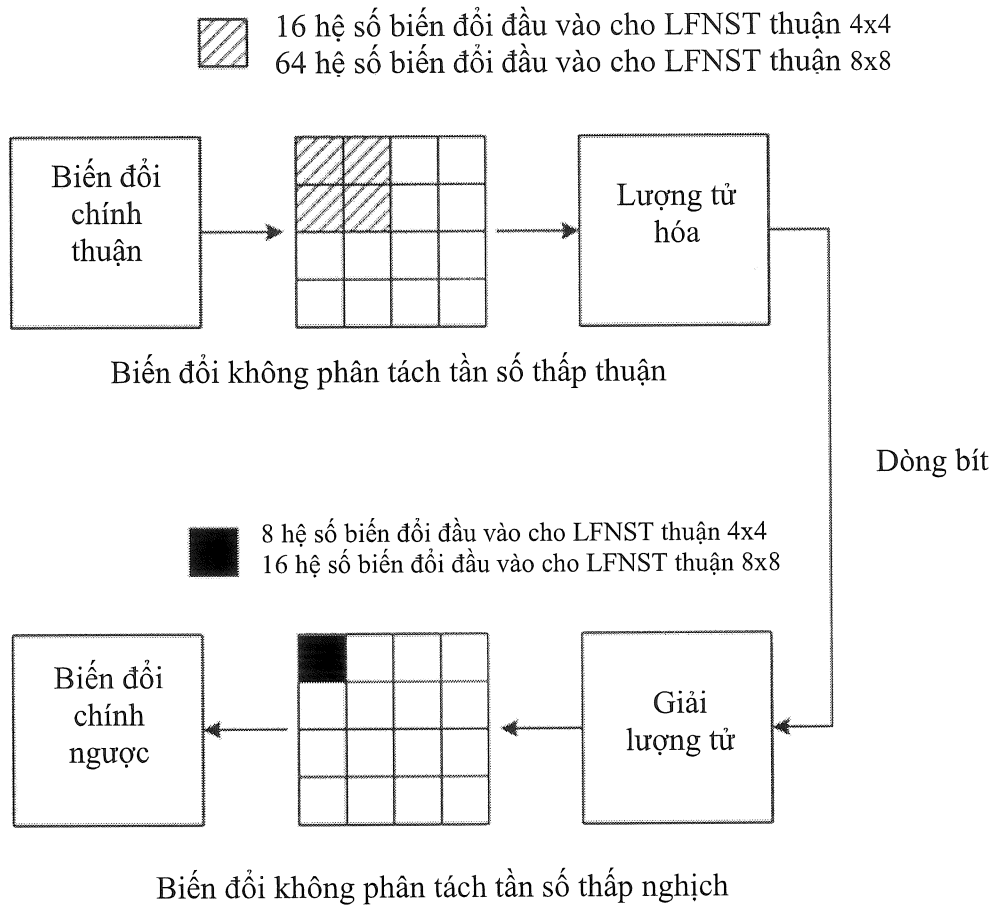


FIG. 2

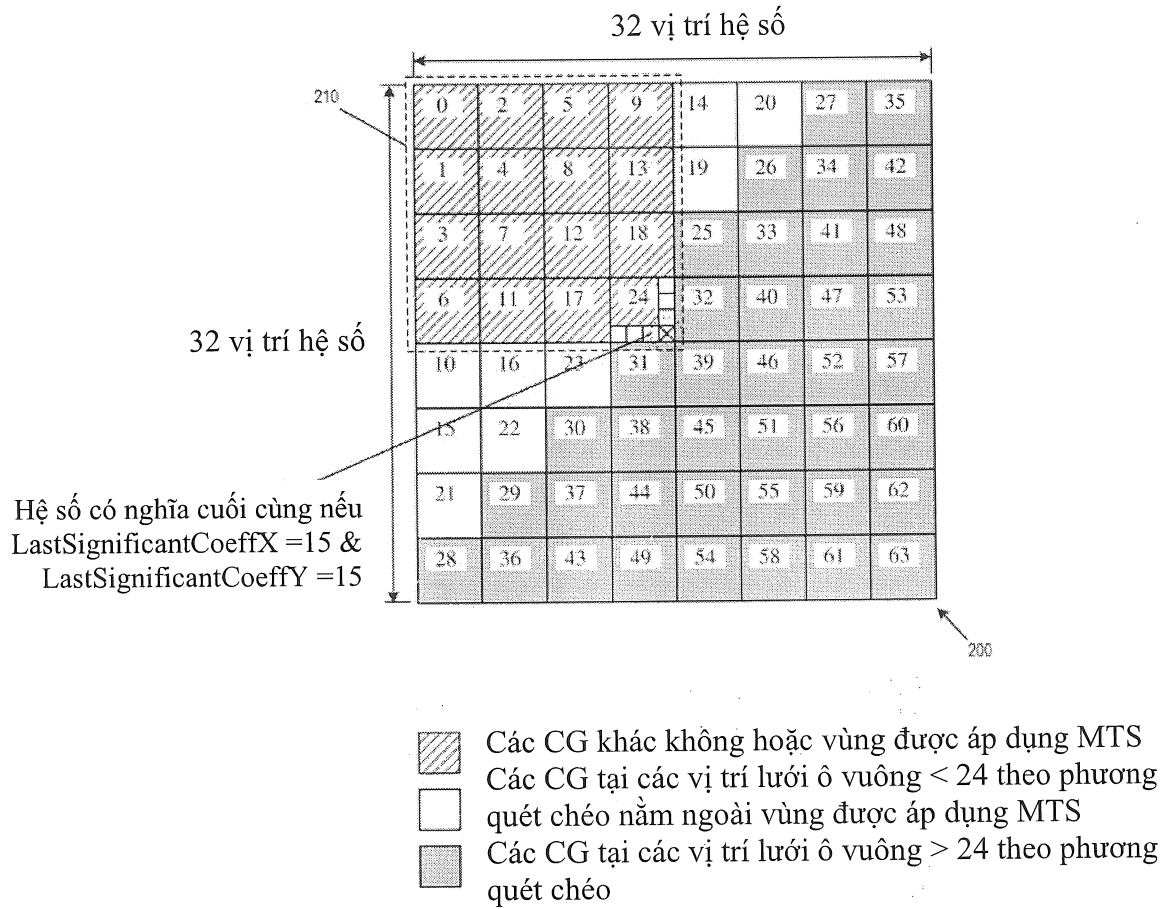
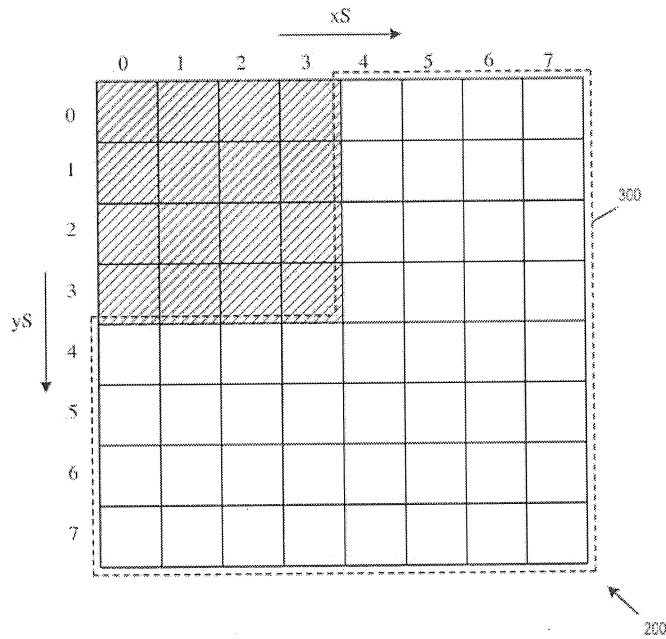


FIG. 3





-  Khối phụ (các hệ số 4x4) có  $xS \leq 3$  và  $yS \leq 3$
-  Khối phụ (các hệ số 4x4) ở vùng ngoài điểm không có  $xS > 3$  hoặc  $yS > 3$ ; `coded_sub_block_flag` được kiểm tra để xác định liệu có báo hiệu/phân tích chỉ số MTS



FIG. 5

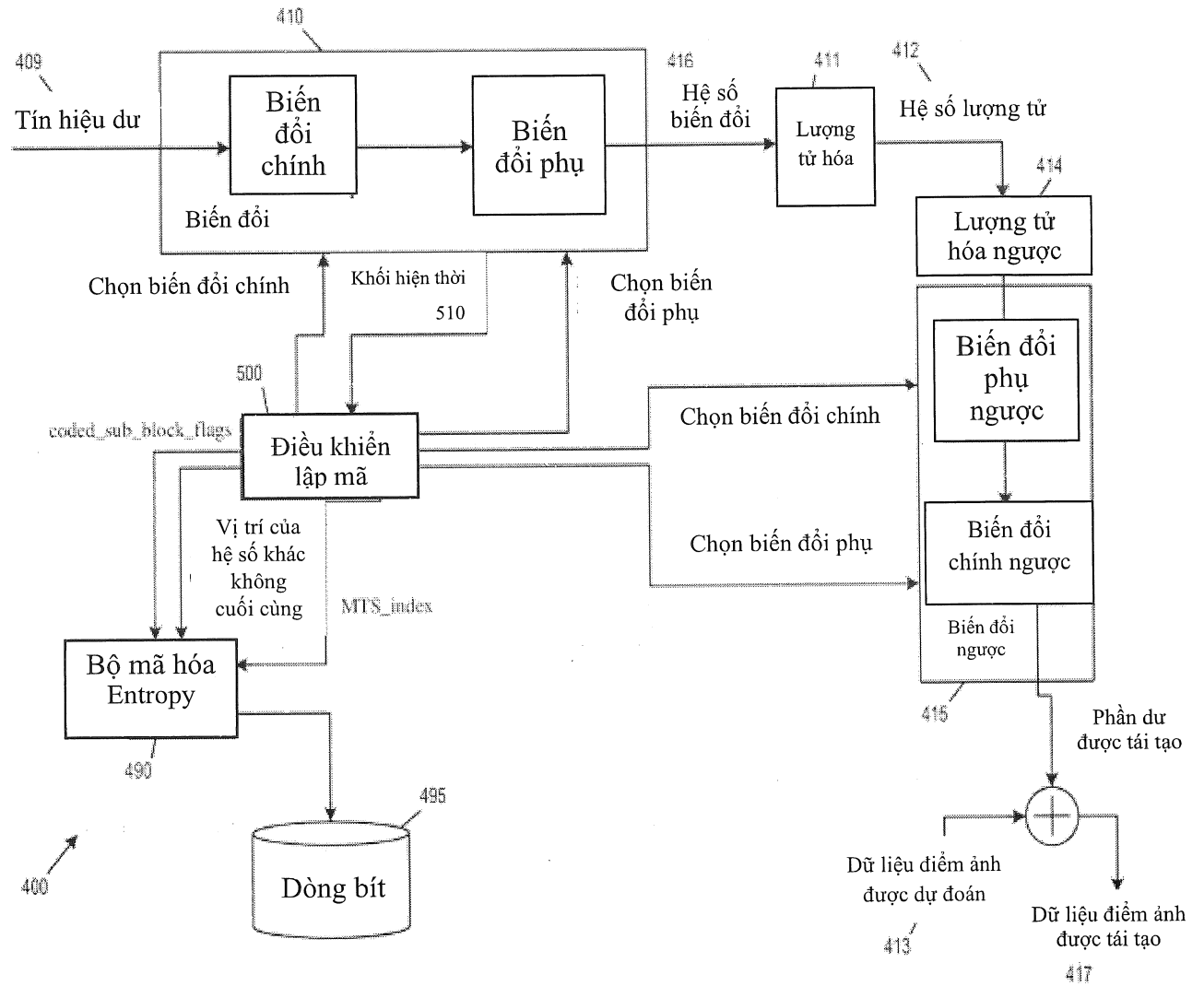


FIG. 6

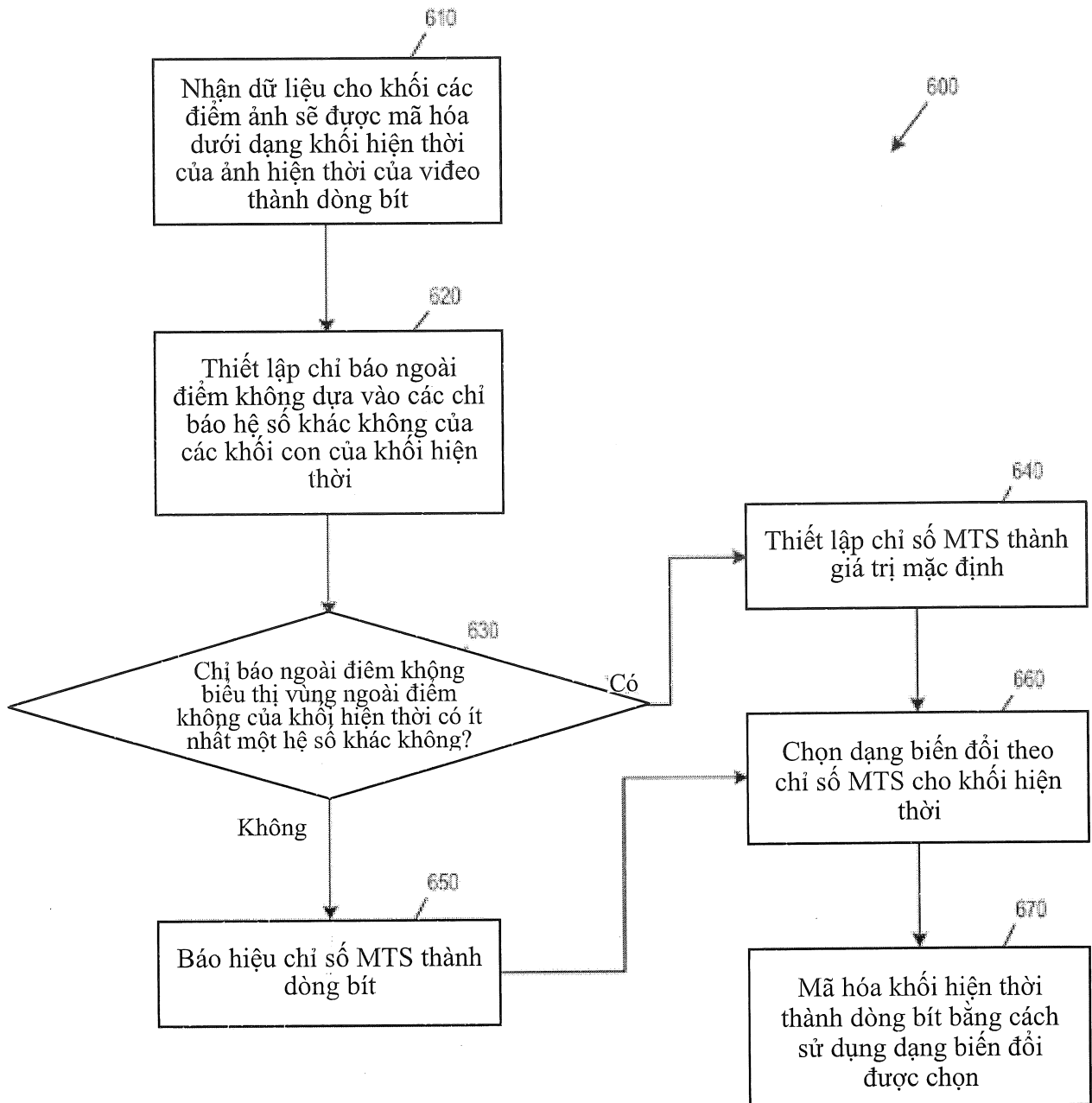


FIG. 7

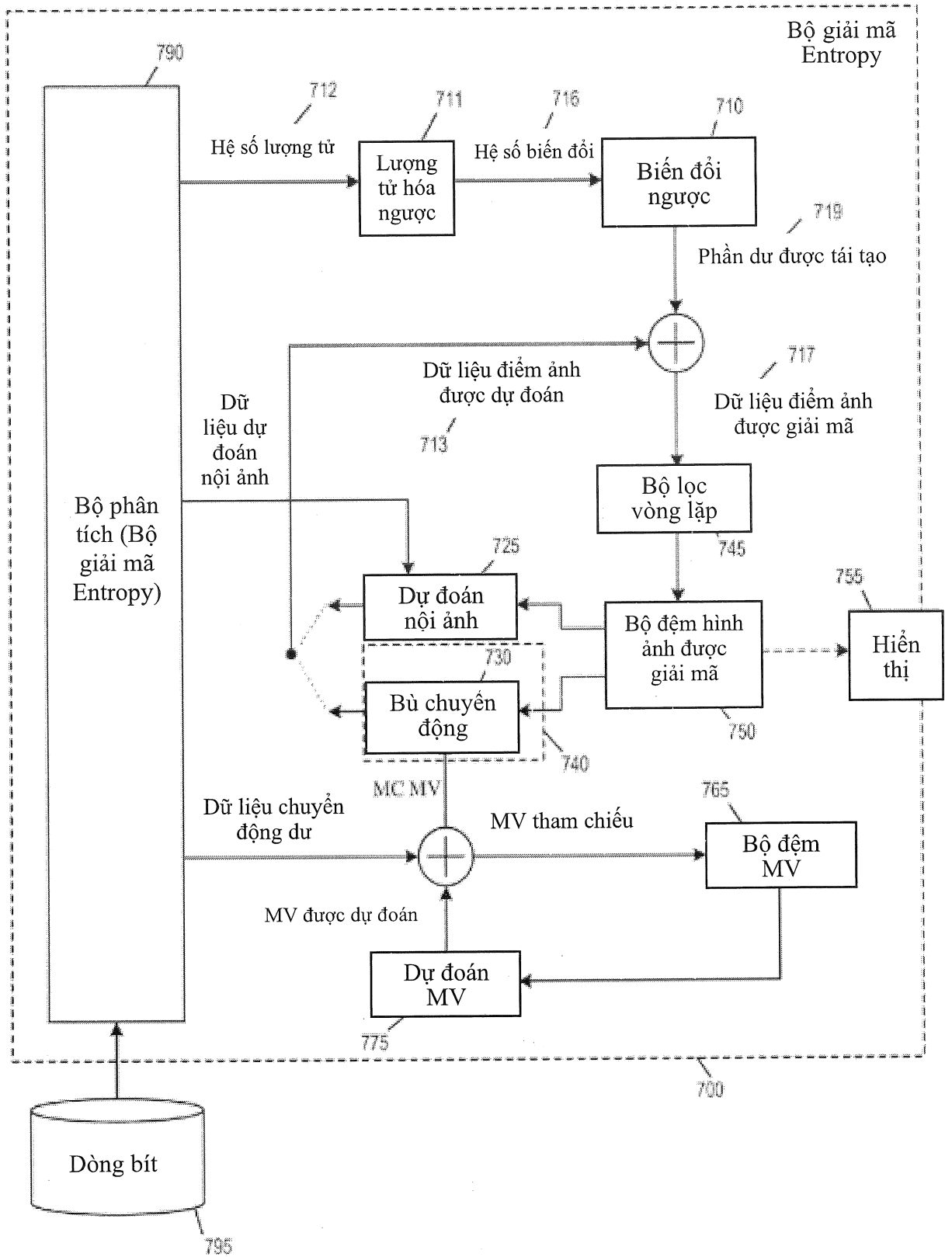




FIG. 8

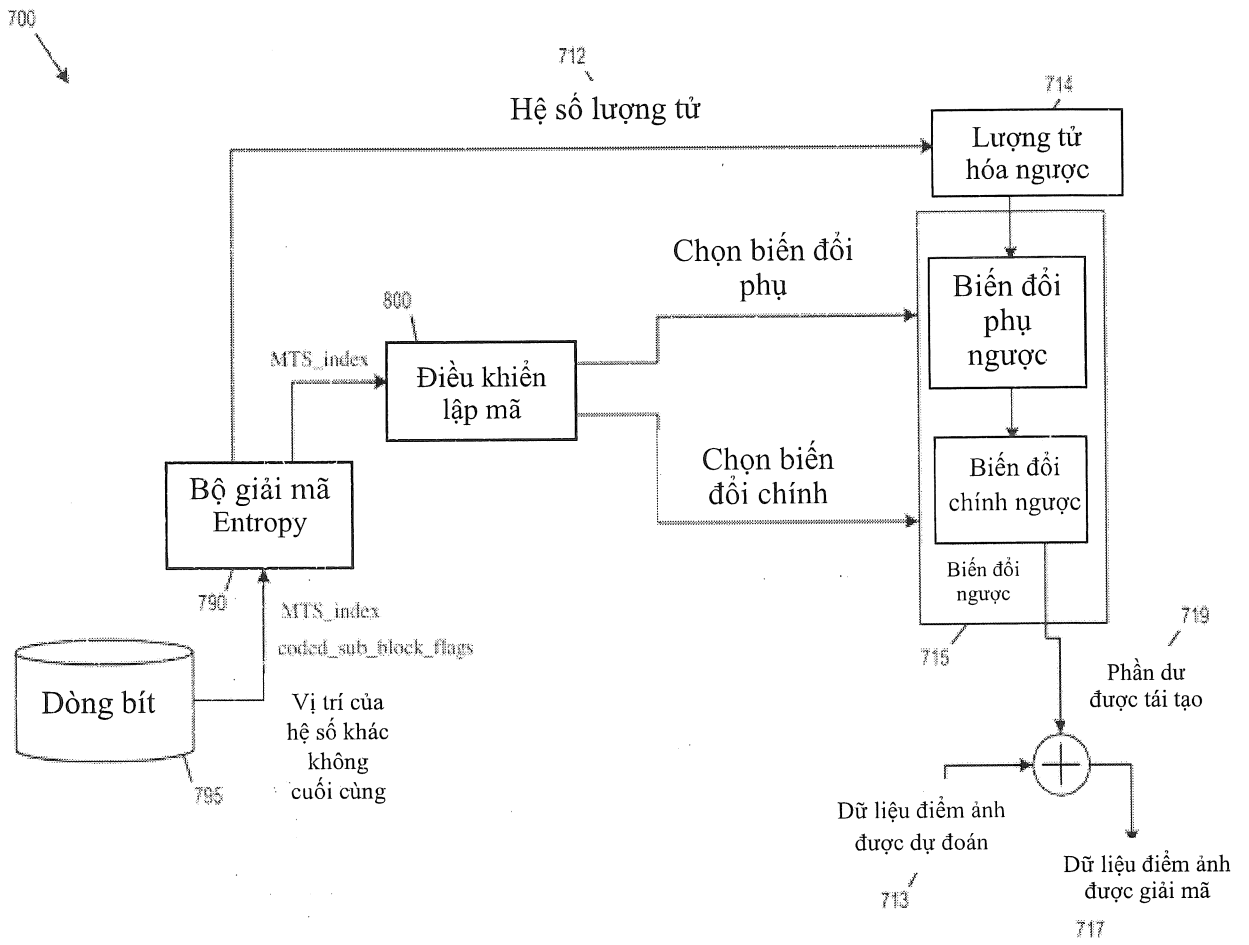


FIG. 9

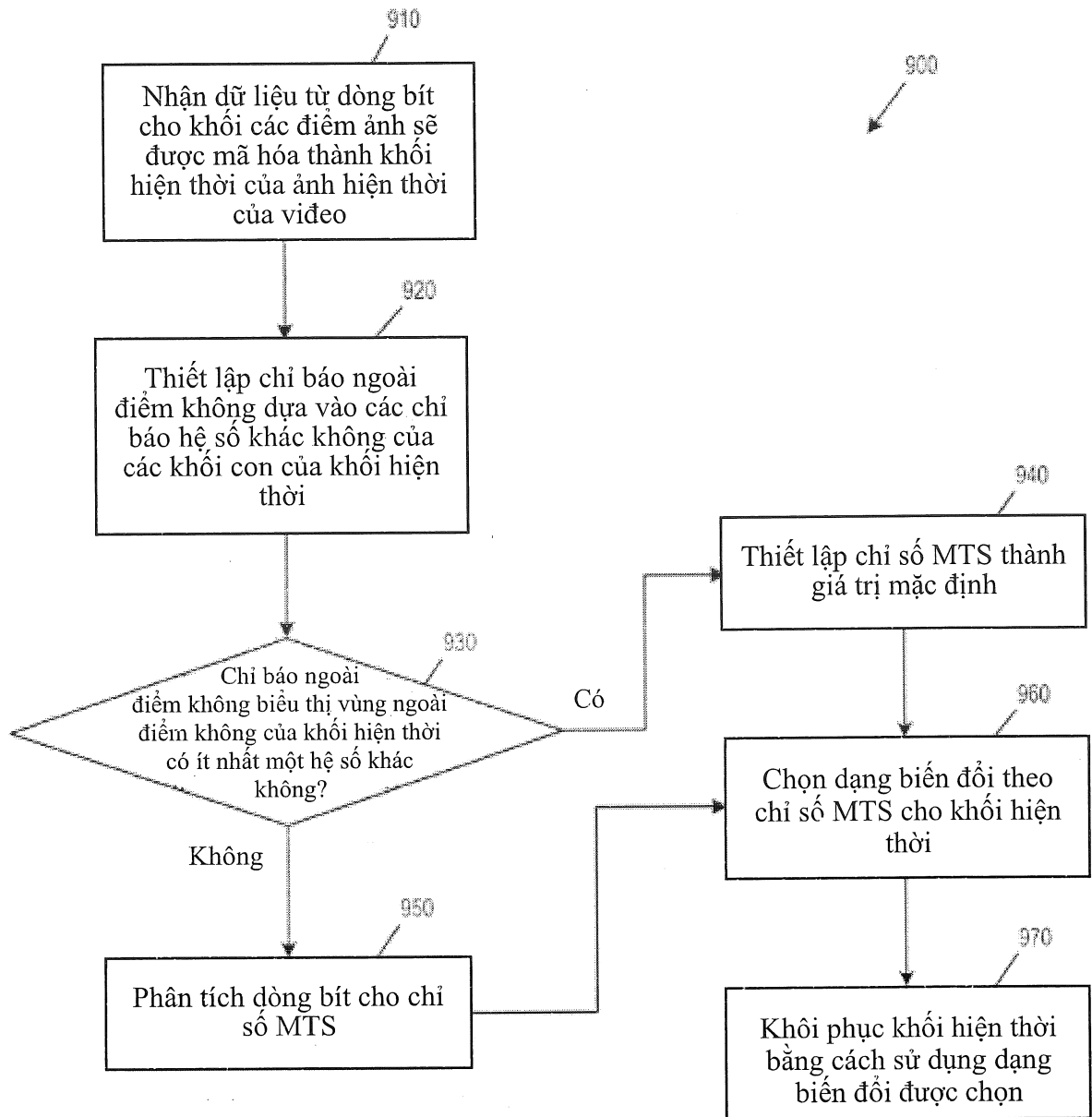


FIG. 10

