



(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)



1-0043214

(51)^{2020.01} H04N 19/80; H04N 19/124

(13) B

(21) 1-2021-00504

(22) 02/07/2019

(86) PCT/RU2019/050101 02/07/2019

(87) WO 2020/009618 09/01/2020

(30) 62/693,441 02/07/2018 US; 62/725,845 31/08/2018 US; 62/731,967 16/09/2018 US;
62/731,972 17/09/2018 US; 62/735,722 24/09/2018 US; 62/757,732 08/11/2018 US;
62/793,866 17/01/2019 US

(45) 25/02/2025 443

(43) 26/04/2021 397

(73) HUAWEI TECHNOLOGIES CO., LTD. (CN)

Huawei Administration Building, Bantian, Longgang District Shenzhen, Guangdong
518129, P. R. China

(72) IKONIN, Sergey Yurievich (RU); STEPIN, Victor Alexeevich (RU); KURYSHEV,
Dmitry (RU); CHEN, Jianle (CN); CHERNYAK, Roman Igorevich (RU).

(74) Công ty TNHH một thành viên Sở hữu trí tuệ VCCI (VCCI-IP CO.,LTD)

(54) PHƯƠNG PHÁP XỬ LÝ KHỐI, BỘ MÃ HÓA VÀ BỘ GIẢI MÃ

(21) 1-2021-00504

(57) Sáng chế đề cập đến phương pháp xử lý khối, bộ mã hóa và bộ giải mã. Bộ lọc được tạo cấu hình để xử lý khối dùng để tạo ra khối được lọc, và trong đó khối bao gồm các điểm ảnh. Bộ lọc bao gồm một hoặc nhiều bộ xử lý được tạo cấu hình để: thu nhận tham số lượng tử hóa (quantization parameter, QP) của khối; thu nhận ngưỡng (threshold, THR) dựa vào QP; và thu nhận bảng tra cứu dựa vào QP, để tạo ra khối được lọc dựa vào ngưỡng và bảng tra cứu. Bộ lọc được đề xuất cho phép nâng cao hiệu quả lập mã video.

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến lĩnh vực lập mã video. Cụ thể hơn là, sáng chế đề cập đến bộ lọc dùng để lập mã video, phương pháp lọc các khung video được cấu trúc lại, và phương pháp lọc các khối video cũng như máy mã hóa và máy giải mã bao gồm bộ lọc như vậy để lập mã video.

Tình trạng kỹ thuật của sáng chế

Video kỹ thuật số đã được sử dụng rộng rãi từ khi giới thiệu các đĩa DVD. Trước khi truyền video được mã hóa và được truyền bằng cách sử dụng phương tiện truyền. Người xem thu video và sử dụng thiết bị xem để giải mã và hiển thị video. Trong những năm qua chất lượng của video đã được nâng cao, ví dụ, do các độ phân giải, các độ sâu màu sắc và các tỷ lệ khung cao hơn. Điều này đã dẫn đến việc các dòng dữ liệu lớn hơn mà ngày nay thường được vận chuyển qua Internet và các mạng truyền thông di động.

Các video độ phân giải cao, tuy nhiên, thường yêu cầu độ rộng dải nhiều hơn khi chúng có nhiều thông tin hơn. Để làm giảm các yêu cầu độ rộng dải các chuẩn lập mã video bao gồm việc nén video đã được giới thiệu. Khi video được mã hóa các yêu cầu độ rộng dải (hoặc các yêu cầu bộ nhớ tương ứng trong trường hợp lưu trữ) được giảm. Thông thường việc giảm này đi kèm với chi phí chất lượng. Vì vậy, các chuẩn lập mã video cố gắng tìm sự cân bằng giữa các yêu cầu độ rộng dải và chất lượng.

Khi có nhu cầu liên tục để nâng cao chất lượng và làm giảm các yêu cầu độ rộng dải, các giải pháp mà duy trì chất lượng với các yêu cầu độ rộng dải được giảm hoặc nâng cao chất lượng trong khi duy trì yêu cầu độ rộng dải tiếp tục được nghiên cứu. Ngoài ra, đôi khi các thỏa hiệp có thể chấp nhận được. Ví dụ, có thể chấp nhận việc làm tăng các yêu cầu độ rộng dải nếu việc nâng cao chất lượng là quan trọng.

Lập mã video hiệu quả cao (High Efficiency Video Coding, HEVC) là ví dụ về chuẩn lập mã video mà thường được biết đến với những người có trình độ trung bình trong lĩnh vực kỹ thuật tương ứng. Trong HEVC, để phân chia đơn vị lập mã (coding unit, CU) thành các đơn vị dự báo (prediction unit, PU) hoặc các đơn vị biến đổi (transform unit, TU). Chuẩn thế hệ tiếp theo lập mã video đa

năng (Versatile Video Coding, VVC) là dự án video liên kết gần đây nhất của các tổ chức chuẩn hóa nhóm các chuyên gia lập mã video (Video Coding Experts Group, VCEG) ITU-T và nhóm các chuyên gia hình ảnh động (Moving Picture Experts Group, MPEG) ISO/IEC, làm việc cùng nhau trong mối quan hệ được biết đến là nhóm khám phá video liên kết (Joint Video Exploration Team, JVET). VVC cũng được gọi là chuẩn ITU-T H.266/lập mã video thế hệ tiếp theo (Next Generation Video Coding, NGVC). Trong VVC, nó loại bỏ các khái niệm về nhiều loại phân vùng, nghĩa là, nó loại bỏ việc tách riêng các khái niệm CU, PU và TU trừ khi cần cho các CU mà có kích thước quá lớn so với độ dài biên đổi lớn nhất, và hỗ trợ sự linh hoạt hơn đối với các hình dạng phân chia CU.

Việc lọc ảnh thường được sử dụng để làm nổi bật các dấu hiệu nhất định của ảnh hoặc để nâng cao mục đích hoặc chất lượng giác quan của ảnh được lọc. Việc lọc ảnh phải giải quyết các nguồn nhiễu khác nhau. Theo đó, các cách tiếp cận khác nhau nhằm nâng cao chất lượng đã được đề xuất và hiện nay đang được sử dụng. Ví dụ, theo phương pháp lọc vòng lặp thích nghi (adaptive Loop filter, ALF), mỗi khung được cấu trúc lại được chia thành tập hợp của các khối nhỏ (các siêu điểm ảnh) và mỗi khối được lọc bởi bộ lọc vòng lặp thích nghi trong đó mỗi điểm ảnh của khung được cấu trúc lại được lọc là tổng trọng số của nhiều điểm ảnh trong vùng được kết nối của điểm ảnh từ khung được cấu trúc lại xung quanh vị trí tạo ra điểm ảnh được lọc. Các hệ số trọng số (cũng được gọi là các hệ số lọc) có đặc tính đối xứng trung tâm và được truyền từ bộ mã hóa tới phía bộ giải mã. Các sườn thông thường có kích thước lớn và do đó số lượng của các hệ số trọng số được truyền có thể trở nên quá lớn so với việc xử lý hiệu quả. Số lượng lớn của các hệ số trọng số yêu cầu tối ưu hóa tỷ lệ biến dạng phức tạp (complex rate-distortion optimization, RDO) ở phía bộ mã hóa để làm giảm số lượng của các hệ số trọng số dùng cho việc truyền. Ở phía bộ giải mã ALF yêu cầu thực hiện các số nhân phổ quát và các số nhân này sẽ được tải lại cho mỗi khối điểm ảnh 2x2.

Vì vậy, có nhu cầu đối với bộ lọc được nâng cao và phương pháp cho phép nâng cao chất lượng dự báo với độ phức tạp thấp và, vì vậy, làm tăng hiệu quả lập mã video.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất bộ lọc được nâng cao và phương pháp cho phép nâng cao hiệu quả lọc với độ phức tạp hạn chế và, vì vậy, làm tăng hiệu quả lập mã video.

Các mục đích nêu trên và các mục đích khác đạt được bởi nội dung chủ yếu của các điểm yêu cầu bảo hộ độc lập. Các hình thức thực hiện khác rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Theo khía cạnh thứ nhất sáng chế đề cập đến bộ lọc dùng để lập mã video, trong đó bộ lọc được tạo cấu hình để xử lý khối dùng cho việc tạo ra khối được lọc, trong đó khối bao gồm các điểm ảnh. Bộ lọc bao gồm bộ phận lưu trữ bộ nhớ bao gồm các lệnh; và một hoặc nhiều bộ xử lý truyền thông với bộ nhớ. Một hoặc nhiều bộ xử lý thực hiện các lệnh để: tải điểm ảnh hiện tại và các điểm ảnh lân cận của nó tới bộ đệm tuyến tính theo mẫu quét được xác định trước; thu nhận các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu nhận các thành phần phổ được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc; thu nhận các điểm ảnh được lọc bằng cách thực hiện biến đổi 1D ngược cho các thành phần phổ được lọc; và tạo ra khối được lọc dựa vào các điểm ảnh được lọc.

Theo ví dụ, khối (hoặc khung) có thể là khối được dự báo, và khối được lọc là khối được dự báo được lọc. Theo ví dụ khác, khối (hoặc khung) có thể là khối được cấu trúc lại, và khối được lọc là khối được cấu trúc lại được lọc.

Theo ví dụ, hệ số khuếch đại là chức năng của thành phần phổ tương ứng và tham số lọc. Tham số lọc có thể được dẫn ra từ tham số lượng tử hóa (quantization parameter, QP) bộ mã hóa-giải mã.

Ví dụ khác, thành phần phổ thứ nhất được tránh qua mà không lọc khi hệ số khuếch đại $G(i, \sigma)$ dùng cho thành phần phổ thứ nhất bằng một. Thành phần phổ thứ nhất tương ứng với tổng hoặc giá trị trung bình của các mẫu trong bộ đệm tuyến tính, và thành phần phổ thứ nhất có thể tương ứng với DC.

Theo ví dụ khác, trong đó một hoặc nhiều bộ xử lý thực hiện các lệnh để: giảm N các bit từ các giá trị bảng của LUT, N là số nguyên. N có thể phụ thuộc vào giá trị QP, hoặc là giá trị cố định.

Theo ví dụ, mẫu quét được xác định trước được định rõ là tập hợp của các dịch vị mảnh hoặc không gian tương ứng với vị trí của điểm ảnh hiện tại bên trong khối được cấu trúc lại. Các dịch vị chỉ tới các điểm ảnh lân cận bên trong khối được cấu trúc lại. Ít nhất một điểm ảnh được lọc có thể được đặt ở vị trí gốc của nó theo mẫu quét được xác định trước. Khi tất cả các điểm ảnh được lọc được bổ sung vào bộ đệm tích lũy theo mẫu quét được xác định trước, và bộ đệm tích lũy có thể được khởi tạo bằng không trước khi thu nhận các thành phần phổ được lọc. Khi các điểm ảnh được lọc cuối cùng được thu nhận như các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng của các điểm ảnh bổ sung vào vị trí hiện tại của bộ đệm tích lũy; một hoặc nhiều bộ xử lý được tạo cấu hình để tạo ra khối được cấu trúc lại được lọc dựa vào các điểm ảnh được lọc cuối cùng.

Một cách tùy chọn, các sự chênh lệch giữa tất cả các điểm ảnh được lọc và không được lọc tương ứng được bổ sung vào bộ đệm tích lũy theo mẫu quét được xác định trước, và bộ đệm tích lũy có thể được khởi tạo bởi các điểm ảnh không được lọc được nhân với số lượng lớn nhất của các trị số điểm ảnh để được bổ sung trong khối. Các điểm ảnh được lọc cuối cùng được thu nhận như các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng lớn nhất của các trị số điểm ảnh để được bổ sung trong khối.

Theo khía cạnh thứ hai sáng chế đề cập đến phương pháp lọc tương ứng để xử lý khối dùng cho việc tạo ra khối được lọc, trong đó khối bao gồm các điểm ảnh. Mỗi điểm ảnh được kết hợp với trị số điểm ảnh. Phương pháp lọc bao gồm các bước: tải điểm ảnh hiện tại và các điểm ảnh lân cận của nó tới bộ đệm tuyến tính theo mẫu quét được xác định trước; thu nhận các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu nhận các thành phần phổ được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc; thu nhận các điểm ảnh được lọc bằng cách thực hiện biến đổi 1D ngược cho các thành phần phổ được lọc; và tạo ra khối được lọc dựa vào các điểm ảnh được lọc.

Theo ví dụ, khối (hoặc khung) có thể là khối được dự báo, và khối được lọc là khối được dự báo được lọc. Theo ví dụ khác, khối (hoặc khung) có thể là khối được cấu trúc lại, và khối được lọc là khối được cấu trúc lại được lọc.

Theo ví dụ, hệ số khuếch đại là chức năng của thành phần phổ tương ứng và tham số lọc. Tham số lọc có thể được dẫn ra từ tham số lượng tử hóa (quantization parameter, QP) bộ mã hóa-giải mã.

Ví dụ khác, thành phần phổ thứ nhất được tránh qua mà không lọc khi hệ số khuếch đại $G(i, \sigma)$ dùng cho thành phần phổ thứ nhất bằng một. Thành phần phổ thứ nhất có thể tương ứng với trị số DC.

Theo ví dụ khác, việc lọc của các thành phần phổ dựa vào bảng tra cứu (look up table, viết tắt là LUT). Việc tạo ra LUT có thể dựa vào chức năng phụ cho ít nhất một vài tham số lượng tử hóa (QP). Chức năng phụ có thể là phương trình đường thẳng đi qua các điểm (i, THR) và $(a, 0)$, trong đó $a > 0$ và a tùy thuộc vào tham số lọc σ hoặc giá trị QP. Ví dụ, đối với QP cuối cùng trong tập hợp của các QP, a bằng 11; hoặc đối với QP cuối cùng thứ hai trong tập hợp của các QP, a bằng 9.

Theo ví dụ khác, phương pháp còn bao gồm bước: giảm N các bit từ các giá trị bảng của LUT, N là số nguyên. N có thể phụ thuộc vào giá trị QP, hoặc là giá trị cố định. Khi N được lựa chọn ít hơn đối với QP thấp hơn khi so sánh với QP cao hơn từ tập hợp của các QP, ví dụ, đối với QP thứ nhất trong tập hợp của các QP, N bằng 2; hoặc đối với các QP còn lại từ tập hợp của các QP, N bằng 3. Theo cách khác, khi N được lựa chọn cao hơn đối với QP cao hơn khi so sánh với QP thấp hơn từ tập hợp của các QP trong số các QP, ví dụ, đối với QP cuối cùng hoặc đối với hai QP cuối cùng trong tập hợp của các QP, N bằng 4; hoặc đối với các QP còn lại từ tập hợp của các QP, N bằng 3. Theo cách khác, khi N được lựa chọn ít hơn đối với QP thấp hơn và cao hơn đối với QP cao hơn khi so sánh với QP còn lại từ tập hợp của các QP, ví dụ, đối với QP thứ nhất trong tập hợp của các QP, N bằng 2; đối với QP cuối cùng hoặc đối với hai QP cuối cùng trong tập hợp của các QP, N bằng 4; hoặc, đối với các QP còn lại từ tập hợp của các QP, N bằng 3.

Theo ví dụ, mẫu quét được xác định trước được định rõ là tập hợp của các dịch vị mảnh hoặc không gian tương ứng với vị trí của điểm ảnh hiện tại bên trong khối được cấu trúc lại. Các dịch vị chỉ tới các điểm ảnh lân cận bên trong khối được cấu trúc lại. Ít nhất một điểm ảnh được lọc có thể được đặt ở vị trí gốc của nó theo mẫu quét được xác định trước. Khi tất cả các điểm ảnh được lọc được bổ sung vào bộ đệm tích lũy theo mẫu quét được xác định trước, và bộ đệm tích lũy

có thể được khởi tạo bằng không trước khi thu nhận các thành phần phổ được lọc. Khi các điểm ảnh được lọc cuối cùng được thu nhận như các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng của các điểm ảnh bổ sung vào vị trí hiện tại của bộ đệm tích lũy; một hoặc nhiều bộ xử lý được tạo cấu hình để tạo ra khối được cấu trúc lại được lọc dựa vào các điểm ảnh được lọc cuối cùng.

Một cách tùy chọn, các sự chênh lệch giữa tất cả các điểm ảnh được lọc và không được lọc tương ứng được bổ sung vào bộ đệm tích lũy theo mẫu quét được xác định trước, và bộ đệm tích lũy có thể được khởi tạo bởi các điểm ảnh không được lọc được nhân với số lượng lớn nhất của các trị số điểm ảnh để được bổ sung trong khối. Các điểm ảnh được lọc cuối cùng được thu nhận như các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng lớn nhất của các trị số điểm ảnh để được bổ sung trong khối.

Theo khía cạnh thứ ba sáng chế đề cập đến máy mã hóa để mã hóa khung hiện tại từ dòng video đầu vào, trong đó máy mã hóa bao gồm bộ lọc theo khía cạnh thứ nhất của sáng chế.

Theo khía cạnh thứ tư sáng chế đề cập đến máy giải mã để giải mã khung hiện tại từ dòng bit thu được, trong đó máy giải mã bao gồm bộ lọc theo khía cạnh thứ nhất của sáng chế.

Theo khía cạnh thứ năm sáng chế đề cập đến chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh thứ hai được thực hiện trên máy tính.

Vì vậy, bộ lọc được đề xuất cho phép nâng cao hiệu quả lập mã video. Cụ thể hơn là, bộ lọc được nâng cao theo các phương án của sáng chế đánh giá các tham số lọc từ chính khung mà không truyền tín hiệu các tham số lọc và, do đó, yêu cầu việc truyền tín hiệu ít hơn đáng kể so với các bộ lọc thông thường, mà truyền tín hiệu các hệ số trọng số để lọc trong miền ảnh.

Mô tả vắn tắt các hình vẽ

Các phương án của sáng chế sẽ được mô tả dựa vào các hình vẽ sau đây, trong đó:

Fig.1A thể hiện hình vẽ giản lược minh họa máy mã hóa theo phương án bao gồm bộ lọc theo một phương án;

Fig.1B thể hiện hình vẽ giản lược minh họa máy mã hóa theo phương án bao gồm bộ lọc theo một phương án;

Fig.2A thể hiện hình vẽ giản lược minh họa máy giải mã theo phương án bao gồm bộ lọc theo một phương án;

Fig.2B thể hiện hình vẽ giản lược minh họa máy giải mã theo phương án bao gồm bộ lọc theo một phương án;

Fig.3A thể hiện hình vẽ giản lược minh họa các khía cạnh của quy trình lọc được thực hiện trong bộ lọc theo một phương án;

Fig.3B thể hiện hình vẽ giản lược minh họa các khía cạnh của quy trình lọc được thực hiện trong bộ lọc theo một phương án;

Fig.3C thể hiện hình vẽ giản lược minh họa các khía cạnh của quy trình lọc được thực hiện trong bộ lọc theo khía cạnh khác;

Fig.4A minh họa các mẫu cho vị trí điểm ảnh khác nhau bên trong khối được cấu trúc lại hình vuông;

Fig.4B đã minh họa hình dạng bộ lọc tương đương cho một điểm ảnh;

Fig.4C đưa ra ví dụ về việc đệm;

Fig.4D đưa ra ví dụ khác về việc đệm;

Fig.5A thể hiện sơ đồ trình tự minh họa các bước của phương pháp lọc theo phương án;

Fig.5B thể hiện sơ đồ trình tự minh họa các bước của phương pháp lọc theo phương án;

Fig.6 là thiết kế bộ lọc phần cứng ví dụ dựa vào SRAM;

Fig.7 là thiết kế phần cứng ví dụ của việc lọc nhóm 2x2 dựa vào các mạch lật (flip-flop);

Fig.8 thể hiện ví dụ về việc kết hợp các kết quả của việc lọc bốn nhóm 2x2 với việc sử dụng lại các kết quả của cùng bộ lọc cuối cùng nhóm không gian;

Fig.9 theo ví dụ, thể hiện kết quả tối ưu hóa LUT;

Fig.10 theo ví dụ, thể hiện khoảng trống không mong muốn trong chức năng chuyển đổi bộ lọc;

Fig.11 theo ví dụ, biểu diễn cùng bảng với các nhập mục bảng là từng biểu đồ;

Fig.12 theo ví dụ, minh họa phương pháp về cách thức khoảng trống có thể được loại bỏ bằng cách sử dụng chức năng phụ;

Fig.13 minh họa ví dụ về việc loại bỏ khoảng trống bằng cách lấy trị số cao nhất từ hai trị số trong khi tạo ra LUT;

Fig.14, theo ví dụ, minh họa chức năng chuyển đổi bộ lọc sau khi áp dụng phương pháp nêu trên;

Fig.15 minh họa ví dụ về các chức năng chuyển đổi bộ lọc tùy thuộc vào năm QP trong tập hợp;

Fig.16 minh họa ví dụ về các chức năng chuyển đổi bộ lọc cho năm QP trong tập hợp dựa vào các bảng tương ứng;

Fig.17 minh họa ví dụ về các chức năng chuyển đổi bộ lọc cho năm QP trong tập hợp dựa vào các bảng tương ứng; và

Fig.18 là hình vẽ giản lược minh họa cấu trúc ví dụ của máy theo một phương án.

Trên các hình vẽ khác nhau, các ký hiệu chỉ dẫn giống nhau sẽ được sử dụng cho các dấu hiệu giống nhau hoặc tương đương về mặt chức năng.

Mô tả chi tiết sáng chế

Trong phần mô tả dưới đây, tham chiếu được thực hiện tới các hình vẽ kèm theo, mà tạo nên một phần của sáng chế, và trong đó được thể hiện, bằng cách minh họa, các khía cạnh cụ thể trong đó sáng chế có thể được đặt. Hiểu rằng các khía cạnh khác có thể được sử dụng và các sự thay đổi về cấu trúc và logic có thể được thực hiện mà không trệch khỏi phạm vi của sáng chế. Phần mô tả chi tiết sau đây, do đó, không được coi là hạn chế, do phạm vi của sáng chế được xác định bởi các điểm yêu cầu bảo hộ kèm theo.

Ví dụ, hiểu rằng sáng chế kết hợp với phương pháp được mô tả cũng có thể đúng đối với thiết bị hoặc hệ thống tương ứng được tạo cấu hình để thực hiện phương pháp và ngược lại. Ví dụ, nếu bước về phương pháp cụ thể được mô

tả, thiết bị tương ứng có thể bao gồm bộ phận để thực hiện bước về phương pháp được mô tả, thậm chí nếu bộ phận như vậy không được mô tả hoặc được minh họa rõ ràng trên các hình vẽ. Hơn nữa, hiểu rằng các dấu hiệu của các khía cạnh ví dụ khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi được lưu ý cụ thể khác.

Fig.1A thể hiện máy mã hóa 100 theo phương án bao gồm bộ lọc 120 theo một phương án. Máy mã hóa 100 được tạo cấu hình để mã hóa khối của khung của tín hiệu video bao gồm các khung (cũng được gọi là các hình ảnh hoặc các ảnh ở đây), trong đó mỗi khung có thể chia thành các khối và mỗi khối bao gồm các điểm ảnh. Theo phương án, các khối có thể là các khối rất lớn, các đơn vị cây lập mã, các đơn vị lập mã, các đơn vị dự báo và/hoặc các khối dự báo.

Thuật ngữ “khối” theo sáng chế được sử dụng cho khối loại bất kỳ hoặc cho khối độ sâu bất kỳ, ví dụ, thuật ngữ “khối” được bao gồm nhưng không giới hạn ở khối gốc, khối, khối con, nút lá, và v.v. Các khối được lập mã không cần có cùng kích thước. Một hình ảnh có thể bao gồm các khối có các kích thước khác nhau và các mảnh khối có các hình ảnh khác nhau của chuỗi video cũng có thể khác nhau.

Theo phương án ví dụ được thể hiện trên Fig.1A, máy mã hóa 100 được thực hiện dưới dạng của bộ mã hóa lập mã video lai. Thông thường, khung thứ nhất của tín hiệu video là nội khung, mà được mã hóa bằng cách sử dụng chỉ dự báo nội khung. Vì vậy, phương án của máy mã hóa 100 được thể hiện trên Fig.1A bao gồm bộ phận dự báo nội khung 154 dùng cho việc dự báo nội khung. Nội khung có thể được giải mã mà không có thông tin từ các khung khác. Bộ phận dự báo nội khung 154 có thể thực hiện việc dự báo nội khung của khối trên cơ sở của thông tin được cung cấp bởi bộ phận đánh giá nội khung 152.

Các khối của các khung tiếp theo tiếp theo nội khung thứ nhất có thể được lập mã bằng cách sử dụng việc liên dự báo hoặc dự báo nội khung, như được lựa chọn bởi bộ phận lựa chọn chế độ 160, Vì vậy, máy mã hóa 100 được thể hiện trên Fig.1A còn bao gồm bộ phận liên dự báo 144. Nói chung, bộ phận liên dự báo 144 có thể được tạo cấu hình để thực hiện việc bù chuyển động của khối dựa vào việc đánh giá chuyển động được cung cấp bởi bộ phận liên đánh giá 142.

Ngoài ra, tín hiệu dự báo theo phương án bộ mã hóa lai được thể hiện trên Fig.1B thu nhận từ việc liên dự báo hoặc dự báo nội khung còn được lọc bởi bộ lọc 145.

Ngoài ra, theo phương án bộ mã hóa lai được thể hiện trên Fig.1A và Fig.1B bộ phận tính toán phần dư 104 xác định sự chênh lệch giữa khối gốc và việc dự báo của nó, nghĩa là, khối dư định rõ lỗi dự báo của việc dự báo liên hình ảnh/hình ảnh nội khung. Khối dư này được biến đổi bởi bộ phận biến đổi 106 (ví dụ bằng cách sử dụng DCT) và các hệ số biến đổi được lượng tử hóa bởi bộ phận lượng tử hóa 108. Đầu ra của bộ phận lượng tử hóa 108 cũng như thông tin phụ hoặc mã hóa được cung cấp, ví dụ, bởi bộ phận dự báo nội khung 154, bộ phận liên dự báo 144 và bộ lọc 120 còn được mã hóa bởi bộ phận mã hóa entropi 170,

Bộ mã hóa video lai thường sao chép quy trình xử lý bộ giải mã sao cho cả hai sẽ tạo ra cùng các dự báo. Vì vậy, theo phương án được thể hiện trên Fig.1 bộ phận lượng tử hóa ngược 110 và bộ phận biến đổi ngược thực hiện các thao tác ngược của bộ phận biến đổi 106 và bộ phận lượng tử hóa 108 và sao chép tính xấp xỉ được giải mã của khối dư. Dữ liệu khối dư được giải mã sau đó được bổ sung vào các kết quả dự báo, nghĩa là, khối dự báo, bởi bộ phận cấu trúc lại 114. Sau đó, đầu ra của bộ phận cấu trúc lại 114 có thể được cung cấp tới bộ đệm dòng 116 để được sử dụng cho việc dự báo nội khung và còn được xử lý bởi bộ lọc 120, mà sẽ được mô tả chi tiết hơn dưới đây. Hình ảnh cuối cùng được lưu trữ trong bộ đệm hình ảnh được giải mã 130 và có thể được sử dụng cho việc dự báo liên khung của các khung tiếp theo.

Fig.2A thể hiện máy giải mã 200 theo phương án bao gồm bộ lọc 220 theo một phương án. Máy giải mã 200 được tạo cấu hình để giải mã khối của khung của tín hiệu video được mã hóa. Theo phương án được thể hiện trên Fig.2A máy giải mã 200 được thực hiện như bộ giải mã lai. Bộ phận giải mã entropi 204 thực hiện việc giải mã entropi của dữ liệu hình ảnh được mã hóa, mà nói chung có thể bao gồm các lỗi dự báo (nghĩa là, các khối dư), dữ liệu chuyển động và thông tin phụ khác, mà cần thiết, cụ thể là, cho bộ phận dự báo nội khung 254 và bộ phận liên dự báo 244 cũng như các bộ phận khác của máy giải mã 200, chẳng hạn như bộ lọc 220. Nói chung, bộ phận dự báo nội khung 254 và bộ phận liên dự báo 244 của máy giải mã 200 được thể hiện trên Fig.2A được lựa chọn bởi bộ phận lựa chọn chế độ 260 và có chức năng giống như cách bộ phận dự báo nội

khung 154 và bộ phận liên dự báo 144 của máy mã hóa 100 được thể hiện trên Fig.1, sao cho các việc dự báo giống nhau có thể được tạo ra bởi máy mã hóa 100 và máy giải mã 200, bộ phận cấu trúc lại 214 của máy giải mã 200 được tạo cấu hình để cấu trúc lại khối trên cơ sở của khối được dự báo được lọc và khối dự được cung cấp bởi bộ phận lượng tử hóa ngược 210 và bộ phận biến đổi ngược 212. Như trong trường hợp của máy mã hóa 100, khối được cấu trúc lại có thể được cung cấp tới bộ đệm dòng 216 được sử dụng cho việc dự báo nội khung và khối/khung được lọc có thể được cung cấp tới bộ đệm hình ảnh được giải mã 230 bởi bộ lọc 220 dùng cho việc liên dự báo.

Như được thể hiện trên Fig.2B, tín hiệu dự báo theo phương án bộ giải mã lai được thể hiện trên Fig.2B thu nhận từ việc liên dự báo hoặc dự báo nội khung còn được lọc bởi bộ lọc 264.

Như được mô tả ở trên, bộ lọc 120, 220 có thể được sử dụng ở mức khung, ví dụ, bộ lọc 120, 220 có thể được tạo cấu hình để xử lý khung được cấu trúc lại từ dòng video được cấu trúc lại được giải mã để tạo ra khung được cấu trúc lại được lọc, trong đó khung được cấu trúc lại bao gồm các khối. Bộ lọc 120, 220 cũng có thể được sử dụng ở mức khối ngay sau khi cấu trúc lại khối mà không chờ toàn bộ khung, ví dụ, bộ lọc 120, 220 có thể được tạo cấu hình để xử lý khối được cấu trúc lại để tạo ra khối được cấu trúc lại được lọc, trong đó khối được cấu trúc lại bao gồm các điểm ảnh.

Bộ lọc 120, 220, 145 hoặc 264 bao gồm một hoặc nhiều bộ xử lý (hoặc một hoặc nhiều bộ phận xử lý). Như sẽ được giải thích chi tiết hơn dưới đây, một hoặc nhiều bộ xử lý (hoặc một hoặc nhiều bộ phận xử lý) được tạo cấu hình để: tải điểm ảnh hiện tại và các điểm ảnh lân cận của nó tới bộ đệm tuyến tính theo mẫu quét được xác định trước (nói cách khác, thứ tự quét, hoặc mẫu quét); thu nhận các thành phần phổ bằng cách thực hiện biến đổi 1D cho mỗi điểm ảnh trong bộ đệm tuyến tính; thu nhận các bộ phận phổ được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc; thu nhận các điểm ảnh được lọc bằng cách thực hiện biến đổi 1D ngược cho phổ được lọc; và tạo ra khối được cấu trúc lại được lọc dựa vào các điểm ảnh được lọc được đánh giá trên các bước xử lý trước đó. Theo ví dụ, hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc. Theo ví dụ khác, hệ số khuếch đại tùy thuộc vào một hoặc nhiều

tham số lọc và một hoặc nhiều bộ phận phổ tương ứng. Theo ví dụ khác, hệ số khuếch đại tương ứng có thể tùy thuộc vào một hoặc nhiều tham số lọc và thành phần phổ tương ứng cũng như các bộ phận phổ lân cận tới phía bên trái và tới phía bên phải của bộ phận phổ.

Sáng chế mô tả bộ lọc trong vòng lặp đối với bộ mã hóa-giải mã video bị mất mà thực hiện việc lọc cục bộ và/hoặc không cục bộ của khối được cấu trúc lại từ khung được cấu trúc lại. Theo ví dụ, khung được cấu trúc lại được chia thành tập hợp của các khối hình chữ nhật không được xếp chồng nhỏ (các khối CU). Ở bước tiếp theo mỗi khối được cấu trúc lại (khối CU được cấu trúc lại) được lọc trong miền tần số độc lập với các khối được cấu trúc lại khác. Bộ lọc cũng có thể được áp dụng sau khi biến đổi và cấu trúc lại, và kết quả được lọc được sử dụng cả hai cho đầu ra cũng như cho việc dự báo không gian và thời gian.

Theo ví dụ khác, sáng chế mô tả bộ lọc dự báo cho bộ mã hóa-giải mã video bị mất mà thực hiện việc lọc cục bộ và/hoặc không cục bộ của khối dự báo của khung được cấu trúc lại.

Ở bước xử lý thứ nhất tất cả các điểm ảnh bên trong khối được cấu trúc lại có thể được xử lý độc lập với nhau. Đối với việc xử lý của điểm ảnh $r(0)$, các điểm ảnh lân cận được sử dụng. Ví dụ, như được minh họa trên Fig.3A, các điểm ảnh từ $r(1)$ đến $r(7)$ được sử dụng, và các điểm ảnh từ $r(0)$ đến $r(7)$ tạo nên một nhóm xử lý.

Fig.3A hoặc 3B thể hiện hình vẽ giản lược 300, 300' minh họa các khía cạnh của quy trình lọc được thực hiện trong bộ lọc theo một phương án. Ở bước 302, 302', điểm ảnh hiện tại và các điểm ảnh lân cận của nó từ khối được tải tới bộ đệm tuyến tính theo mẫu quét được xác định trước. Theo ví dụ, khối có thể là khối được dự báo. Theo ví dụ khác, khối có thể là khối được cấu trúc lại.

Ở bước 304, 304', việc biến đổi 1D được thực hiện đối với điểm ảnh $r(0)$ và các điểm ảnh lân cận từ $r(1)$ đến $r(7)$ của nó trong bộ đệm tuyến tính để

$$R = 1D_Transform(r)$$

thu nhận các thành phần phổ R:

Theo ví dụ, việc biến đổi 1D có thể là biến đổi Hadamard.

Cần hiểu rằng xem có thực hiện biến đổi 1D trên 4 điểm ảnh trong hàng (ví dụ, các điểm ảnh A, B, C, D như ở ví dụ trên Fig.3B) hay thực hiện việc

biến đổi 2D tới các điểm ảnh A, B, C, D được bố trí về mặt không gian liên quan tới việc thực hiện cụ thể. Việc áp dụng việc biến đổi 2D trên 4 điểm ảnh A, B, C, D được bố trí trong khối 2x2 có thể dẫn tới cùng kết quả như việc áp dụng việc biến đổi 1D tới 4 điểm ảnh A, B, C, D được lấy ra như hàng. Ở bước 306, 306', việc lọc được thực hiện trong miền tần số dựa vào phép nhân của mỗi thành phần phổ $R(i)$ bởi hệ số khuếch đại tương ứng $G(i, \sigma)$ để thu nhận các thành phần phổ được lọc $F(i)$:

$$\text{Tập hợp của các hệ số khuếch đại cho tất cả các thành phần phổ là}$$

$$F(i) = R(i) * G(i, \sigma) \quad (1)$$

phản hồi xung tần số của bộ lọc.

Theo ví dụ, hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc. Theo ví dụ khác, hệ số khuếch đại tùy thuộc vào một hoặc nhiều tham số lọc và một hoặc nhiều thành phần phổ tương ứng. Theo ví dụ khác, hệ số khuếch đại tương ứng có thể tùy thuộc vào một hoặc nhiều tham số lọc và thành phần phổ tương ứng cũng như các bộ phận phổ lân cận tới phía bên trái và tới phía bên phải của bộ phận phổ. Nếu mỗi hệ số khuếch đại là chức năng của thành phần phổ của khối được cấu trúc lại và tham số lọc, hoặc là chức năng của

$$G(i, \sigma) = \frac{R(i)^2}{R(i)^2 + m * \sigma^2} \quad (2)$$

thành phần phổ của khối được dự báo và tham số lọc, hệ số khuếch đại $G(i, \sigma)$ có thể được mô tả bởi công thức sau đây theo ví dụ:

trong đó (i) là chỉ số của bộ phận phổ, $R(i)$ là thành phần phổ tương ứng với chỉ số (i), $G(i, \sigma)$ là hệ số khuếch đại tương ứng với $R(i)$, σ là tham số lọc, m là hằng số chuẩn hóa bằng với số lượng các thành phần phổ. Ví dụ, m là 1, 2, 3, 4 Các thành phần phổ khác nhau có thể có cùng hệ số khuếch đại, hoặc có thể có các hệ số khuếch đại khác nhau.

Đối với các sự biến đổi đó mà có các thành phần phổ tương ứng với số trung bình (FFT, DCT, DST v.v.) hoặc tổng (Hadamard) của các mẫu đầu vào của khối biến đổi (thường bộ phận thứ nhất tương ứng với trị số DC) điều có thể

có lợi là có hệ số lọc bằng 1 để tránh khỏi việc thay đổi của độ sáng trung bình của khối được lọc. Điều đó có nghĩa là tránh qua (không lọc) cho thành phần phổ thứ nhất tương ứng với trị số DC.

σ là tham số lọc, có thể dẫn ra từ tham số lượng tử hóa (QP) bộ mã hóa-giải mã ở các phía bộ mã hóa và bộ giải mã, ví dụ, bằng cách sử dụng công thức sau đây:

$$\sigma = k * 2^{(n*(QP-s))}$$

, trong đó k, n và s là các hằng số có các trị số như ví dụ: $k = 2,64, n = 0,1296, s = 11$.

Các thành phần phổ khác nhau có thể có cùng tham số lọc, hoặc có thể có các tham số lọc khác nhau.

Các thông số k, n và s có thể được lựa chọn theo cách như vậy để tạo ra xicma phụ thuộc vào kích thước bước lượng tử hóa mà tăng gấp đôi mỗi lần giá trị QP tăng thêm 6 trong các chuẩn lập mã video mới nhất. Theo ví dụ có các thông số $k = 0,5, n = \frac{1}{6}$ và $s = 0$ các thông số được dẫn ra như sau:

$$\sigma = 0,5 * 2^{\left(\frac{QP}{6}\right)}$$

Các ma trận chia tỷ lệ lượng tử hóa được sử dụng rộng rãi để nâng cao chất lượng nén video. Theo phương pháp này kích thước bước lượng tử hóa được dẫn ra dựa vào QP được nhân với hệ số chia tỷ lệ được truyền trong dòng bit. Đối với phương pháp như vậy việc dẫn ra các thông số σ có thể dựa vào kích thước bước lượng tử hóa được chia tỷ lệ thực tế được sử dụng cho QP nhất định:

$$\sigma = k * \text{Kích thước bước lượng tử hóa (QP - s)}$$

Các hằng số k, n và s có thể có các giá trị cố định dùng cho việc tính toán σ , hoặc có thể có các giá trị khác nhau tùy thuộc vào QP, kích thước và hình dạng khối, loại dự báo (liên dự báo/dự báo nội khung) cho khối hiện tại. Ví dụ đối với các khối vuông được dự báo nội khung có kích thước 32x32 hoặc lớn hơn

thông số s có thể được tính toán là $s = 11 + 8 = 19$. Với thông số bộ lọc tương đương σ có giá trị nhỏ hơn mà dẫn đến bộ lọc mềm hơn mà thích hợp hơn đối với các khối được dự báo nội khung vuông lớn thường tương ứng với các vùng phẳng. Theo ví dụ khác, k có thể được điều chỉnh dựa vào độ sâu bit của các điểm ảnh $k_{mod} = k * (1 \ll (\text{độ sâu bit} - 8))$

Theo phương pháp 300, 300', hệ số khuếch đại cho mỗi tần số được dẫn ra từ thành phần phổ của các điểm ảnh được cấu trúc lại hoặc các điểm ảnh được dự báo. Do đó, phương pháp 300, 300' không cần truyền các tham số lọc và có thể được áp dụng cho khối được cấu trúc lại hoặc khối được dự báo bất kỳ mà không truyền tín hiệu bổ sung.

Các chi tiết LUT được thảo luận.

Có thể lưu ý rằng việc lọc hàm ý phép nhân của thành phần phổ $R(i)$ trên hệ số chia tỷ lệ luôn nhỏ hơn 1. Cũng có thể được quan sát rằng ở các giá trị cao của $R(i)$ hệ số chia tỷ lệ gần bằng 1. Dựa vào các sự quan sát này việc lọc phổ được thực hiện bằng cách sử dụng bảng tra cứu mà cho phép loại trừ các phép nhân và phép chia khỏi các thao tác lọc:

Hệ số khuếch đại phổ nhỏ hơn 1, vì vậy việc lọc có thể được thực hiện dựa vào việc đọc bảng tra cứu (look up table, viết tắt là LUT) ngắn theo các công thức sau đây:

$$F(i, \sigma) = \begin{cases} R(i), & \text{Abs}(R(i)) \geq THR \\ LUT(R(i), \sigma), & R(i) > 0 \\ -LUT(-R(i), \sigma), & R(i) \leq 0 \end{cases}$$

Trong đó $LUT(R(i), \sigma) = \frac{R(i)^3}{R(i)^2 + m * \sigma^2}$. (i) là chỉ số của bộ phận phổ, $R(i)$ là thành phần phổ tương ứng với chỉ số (i) , σ là tham số lọc, và THR là ngưỡng, m là hằng số chuẩn hóa bằng với số lượng các thành phần phổ. Ví dụ, m là 1, 2, 3, 4

Theo ví dụ, THR có thể được tính toán từ công thức sau đây,

$$\frac{THR^2}{THR^2 + m \cdot \sigma^2} = C$$

trong đó C là giá trị gần bằng 1, ví dụ, 0,8 hoặc 0,9. Để làm giảm ngưỡng kích thước LUT THR có thể phụ thuộc từ giá trị QP.

Đối với việc làm giảm thêm kích thước LUT ngưỡng thứ hai có thể được giới thiệu để thay thế các giá trị được lọc nhỏ bằng không. Trong trường hợp đó thành phần phổ được lọc là $F(i, \sigma)$ còn được dẫn ra là:

$$F(i, \sigma) = \begin{cases} R(i), Abs(R(i)) > THR \\ 0, Abs(R(i)) < THR2 \\ LUT(R(i), \sigma), R(i) > 0 \\ -LUT(-R(i), \sigma), R(i) \leq 0 \end{cases}$$

Trong đó THR2 định rõ ngưỡng dưới đây mà thành phần phổ được lọc được coi là không. THR2 thứ hai cũng có thể được định rõ tùy thuộc vào giá trị QP.

Sau khi lọc trong miền tần số, việc biến đổi 1D ngược được thực hiện cho thành phần phổ được lọc ở bước 308 để thu nhận các điểm ảnh được lọc f :

$$f = \text{Biến đổi ngược 1D}(F)$$

Ở bước 310, 310', kết quả của việc biến đổi 1D ngược được đặt ở bộ đệm tuyến tính của các điểm ảnh được cấu trúc lại được lọc hoặc các điểm ảnh được lọc.

Ở bước 312, 312' (không được thể hiện trên Fig.3A hoặc 3B), khối được lọc được tạo ra dựa vào các điểm ảnh được lọc được đánh giá trên các bước xử lý trước đó. Theo ví dụ, khối được lọc có thể là khối được dự báo được lọc. Theo ví dụ khác, khối được lọc có thể là khối được cấu trúc lại được lọc.

Như được thể hiện trên Fig.3A theo phương án, sau bước lọc 306, điểm ảnh được lọc $f(0)$ được đặt ở vị trí gốc của nó theo mẫu quét được xác định trước. Các mẫu được lọc khác $f(1) - f(7)$ không được sử dụng. Ở khía cạnh khác, nhiều hơn một điểm ảnh được lọc, ví dụ, tất cả các điểm ảnh được lọc từ bộ đệm

tuyến tính của các mẫu được lọc được bổ sung vào bộ đệm tích lũy theo mẫu quét được xác định trước được sử dụng ở bước 302 trên Fig.3A. Bộ đệm tích lũy nên được khởi tạo bằng không trước bước lọc. Ở bước chuẩn hóa cuối cùng, các điểm ảnh được lọc cuối cùng được thu nhận như các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng của các điểm ảnh được bổ sung vào vị trí hiện tại của bộ đệm tích lũy, nói cách khác, số lượng các giá trị điểm ảnh được bổ sung vào vị trí hiện tại của bộ đệm tích lũy trên các bước xử lý trước đó. Sau đó khối được cấu trúc lại được lọc hoặc khối được dự báo được tạo ra dựa vào các điểm ảnh được lọc cuối cùng.

Theo khía cạnh khác, bộ lọc có cùng cách thực hiện cho cả việc lọc đơn vị lập mã (CU) liên khung và nội khung.

Bộ lọc miền biến đổi Hadamard luôn được áp dụng tới các khối được cấu trúc lại độ sáng (luma) với các hệ số biến đổi không là không, loại trừ các khối 4x4 và nếu tham số lượng tử hóa lớn hơn 17. Các thông số bộ lọc được dẫn ra rõ ràng từ thông tin được lập mã. Bộ lọc được đề xuất, nếu được áp dụng, được thực hiện trên các mẫu được giải mã ngay sau khi cấu trúc lại khối. Kết quả được lọc được sử dụng cả hai cho đầu ra cũng như cho việc dự báo không gian và thời gian.

Quy trình lọc được thảo luận, như được biểu diễn giản lược trên Fig.3C.

Đối với mỗi điểm ảnh từ khối được cấu trúc lại quy trình xử lý điểm ảnh bao gồm các bước sau đây:

Quét 4 điểm ảnh lân cận xung quanh điểm ảnh xử lý bao gồm điểm ảnh xử lý hiện tại theo mẫu quét.

Biến đổi Hadamard 4 điểm của các điểm ảnh được đọc.

Lọc phổ dựa vào công thức (1) và (2).

Thành phần phổ thứ nhất tương ứng với trị số DC được tránh qua mà không lọc.

Biến đổi Hadamard 4 điểm ngược của phổ được lọc.

Sau bước lọc các điểm ảnh được lọc được đặt ở các vị trí gốc của nó vào bộ đệm tích lũy.

Sau khi hoàn tất việc lọc của các điểm ảnh các giá trị được tích lũy được chuẩn hóa bởi số lượng của các nhóm xử lý được sử dụng cho việc lọc mỗi điểm ảnh. Do sử dụng phần đệm của một mẫu xung quanh số lượng khối của các nhóm xử lý bằng 4 đối với mỗi điểm ảnh trong khối và việc chuẩn hóa được thực hiện bằng cách dịch chuyển bên phải trên 2 bit.

Có thể nhận thấy rằng tất cả các điểm ảnh trong khối có thể được xử lý độc lập trong trường hợp của tính song song lớn nhất được yêu cầu.

Theo các phương án này, ngưỡng THR được thiết đặt tới trị số định trước, ví dụ, 128, mà trong trường hợp của cách thực hiện đơn giản yêu cầu lưu trữ 128 ($1 \ll 7$) nhập mục của các trị số 7 bit trên mỗi QP.

Kích thước của LUT ảnh hưởng đến các yêu cầu về lượng của bộ nhớ trên chip và chi phí thực hiện phần cứng của bộ lọc. Để làm giảm lượng lưu trữ trên chip LUT nếu được tính toán chỉ cho tập hợp giới hạn của các QP bắt đầu từ QP 20 có khoảng không đổi là 8. Tổng năm LUT định trước (đối với nhóm năm QP) được lưu trữ. Đối với việc lọc của QP của khối CU hiện tại được làm tròn tới QP sát nhất từ bảng.

Đối với việc làm giảm thêm kích thước LUT các bit thấp nhất N được giảm (hoặc được bỏ qua) trong suốt thời gian tạo ra LUT. Điều đó cho phép có sự biểu diễn bảng thừa.

Đối với cách thực hiện ví dụ A, N bằng 2, mà dẫn tới $7 - 2 = 5$ bit của độ sâu bảng (32 nhập mục của các trị số 7 bit);

Đối với cách thực hiện ví dụ B, N bằng 3, mà dẫn tới $7 - 3 = 4$ bit của độ sâu bảng (16 nhập mục của các trị số 7 bit).

Vì vậy tổng kích thước bộ nhớ được yêu cầu cho toàn bộ sự lưu trữ LUT:

Đối với cách thực hiện ví dụ A: $5 \times 32 \times 7 \text{ bit} = 1120 \text{ bit} = 140 \text{ byte}$;

Đối với cách thực hiện ví dụ B: $5 \times 16 \times 7 \text{ bit} = 560 \text{ bit} = 70 \text{ byte}$;

Cách thực hiện ví dụ B đang nhằm tới 16 byte có kích thước LUT để cho phép truy cập song song theo cách thực hiện phần mềm do khả năng lưu trữ toàn bộ LUT trong một thanh ghi SSE 16 byte do đó cấu hình này được gợi ý.

Nếu biến đổi Hadamard được sử dụng, và điểm ảnh được lọc được đặt ở vị trí gốc của nó theo mẫu quét được xác định trước, sau đó mã giả sau đây mô tả quy trình lọc của phương pháp 300:

// quét các điểm ảnh được cấu trúc lại/được dự báo

```
const int x0 = pIn[p0];
```

```
const int x1 = pIn[p1];
```

```
const int x2 = pIn[p2];
```

```
const int x3 = pIn[p3]; // p0-p3 định rõ mẫu quét
```

// biến đổi Hadamard về phía trước 1D

```
const int y0 = x0 + x2;
```

```
const int y1 = x1 + x3;
```

```
const int y2 = x0 - x2;
```

```
const int y3 = x1 - x3;
```

```
const int t0 = y0 + y1;
```

```
const int t1 = y0 - y1;
```

```
const int t2 = y2 + y3;
```

```
const int t3 = y2 - y3;
```

// lọc miền tần số

```
const int z0 = pTbl[t0];
```

```
const int z1 = pTbl[t1];
```

```

const int z2 = pTbl[t2];
const int z3 = pTbl[t3];
// biến đổi Hadamard về phía sau
const int iy0 = z0 + z2;
const int iy1 = z1 + z3;
const int iy2 = z0 - z2;
const int iy3 = z1 - z3;

// điểm ảnh được lọc được đưa ra
pOut[p0_out] = iy0 + iy1;

```

Nếu biến đổi Hadamard được sử dụng, và nhiều hơn một điểm ảnh được lọc từ bộ đệm tuyến tính của các mẫu được lọc được bổ sung vào bộ đệm tích lũy, sau đó mã giả sau đây mô tả quy trình lọc của kịch bản này:

```

// quét các điểm ảnh được cấu trúc lại/được dự báo
const int x0 = pIn[p0];
const int x1 = pIn[p1];
const int x2 = pIn[p2];
const int x3 = pIn[p3]; // p0-p3 định rõ mẫu quét
// biến đổi Hadamard về phía trước 1D
const int y0 = x0 + x2;
const int y1 = x1 + x3;
const int y2 = x0 - x2;
const int y3 = x1 - x3;
const int t0 = y0 + y1;
const int t1 = y0 - y1;

```

```

const int t2 = y2 + y3;

const int t3 = y2 - y3;

// lọc miền tần số

const int z0 = pTbl[t0];

const int z1 = pTbl[t1];

const int z2 = pTbl[t2];

const int z3 = pTbl[t3];

// biến đổi Hadamard về phía sau

const int iy0 = z0 + z2;

const int iy1 = z1 + z3;

const int iy2 = z0 - z2;

const int iy3 = z1 - z3;

```

Như theo phương án khác bộ đệm tích lũy nên được khởi tạo bởi các trị số điểm ảnh không được lọc được nhân với số lượng lớn nhất của các trị số điểm ảnh để được bổ sung trong khối. Số lượng lớn nhất của các trị số điểm ảnh được bổ sung trong khối được định rõ dựa vào mẫu quét. Thực vậy mẫu quét định rõ số lượng của các trị số điểm ảnh được bổ sung cho mỗi vị trí. Dựa vào đó, số lượng lớn nhất từ tất cả các vị trí trong khối có thể được lựa chọn và được sử dụng trong suốt thời gian khởi tạo bộ đệm tích lũy. Sau đó, trong suốt thời gian mỗi bước tích lũy, trị số điểm ảnh không được lọc được trừ từ trị số được lọc tương ứng và được bổ sung vào bộ đệm tích lũy:

```

// tích lũy các điểm ảnh được lọc

pOut[p0] += iy0 + iy1 // p0-p3 định rõ mẫu quét

pOut[p1] += iy0 - iy1

pOut[p2] += iy2 + iy3

pOut[p3] += iy2 - iy3

```

Để làm giảm độ sâu bit của các trị số điểm ảnh được tích lũy trước khi đặt vào bộ đệm tích lũy kết quả biến đổi về phía sau có thể được chuẩn hóa trên kích thước biến đổi (m) :

```
pOut[p0] += ((iy0 + iy1) >> HTDF_BIT_RND4);
```

```
pOut[p1] += ((iy0 - iy1) >> HTDF_BIT_RND4);
```

```
pOut[p2] += ((iy2 + iy3) >> HTDF_BIT_RND4);
```

```
pOut[p3] += ((iy2 - iy3) >> HTDF_BIT_RND4);
```

trong đó HTDF_BIT_RND4 bằng 2 cho kích thước biến đổi là 4.

Phương án này cho phép hệ thống tránh khỏi việc lưu trữ số lượng của các điểm ảnh được bổ sung vào vị trí hiện tại và cho phép việc phân và chia thay thế bởi phép tính dịch chuyển ở bước chuẩn hóa cuối cùng và khởi tạo bước đệm tích lũy tương ứng nếu số lượng lớn nhất của các trị số điểm ảnh được bổ sung là công suất của ví dụ, 2, 4, 8 v.v.

Để giữ độ chính xác của giai đoạn chuẩn hóa có thể được thực hiện theo cách sau đây:

```
// chuẩn hóa
```

```
pFiltered[p0] = CLIP3(0, (1 << ĐỘ SÂU BIT) - 1, (pOut[p0] + HTDF_CNT_SCALE_RND) >> HTDF_CNT_SCALE);
```

trong đó HTDF_CNT_SCALE là Log2 của lượng của các điểm ảnh được đặt vào bộ đệm tích lũy, ví dụ, đối với lượng của 4 HTDF_CNT_SCALE bằng 2, và HTDF_CNT_SCALE_RND bằng $(1 << (HTDF_CNT_SCALE - 1))$. CLIP3 là chức năng cắt xén mà đảm bảo mẫu được lọc trong khoảng được cho phép giữa trị số mẫu lớn nhất và nhỏ nhất.

Như đã được nêu trên để tránh khỏi việc thay đổi của độ sáng trung bình của khối được lọc điều có lợi là bỏ qua việc lọc của thành phần phổ thứ nhất (tương ứng với trị số DC). Điều đó còn cho phép đơn giản hóa việc thực hiện bộ lọc. Trong trường hợp này bước lọc như sau:

```
// lọc miền tần số
```

```
const int z0 = t0;
```

```
const int z1 = pTbl[t1];
```

```
const int z2 = pTbl[t2];
```

```
const int z3 = pTbl[t3];
```

Đối với mỗi điểm ảnh bên trong của khối được cấu trúc lại hoặc khối được dự báo, mẫu quét được chọn dựa vào vị trí của điểm ảnh lọc bên trong khối được cấu trúc lại hoặc khối được dự báo đối với các bước 302 và 310. Mẫu quét được chọn để đảm bảo tất cả các điểm ảnh bên trong CU được cấu trúc lại hoặc CU được dự báo và đặt gần bằng điểm ảnh xử lý. Thứ tự quét tùy ý có thể được sử dụng bên trong mẫu. Ví dụ, mẫu quét được xác định trước được định rõ là tập hợp của các dịch vị mảnh hoặc không gian tương ứng với vị trí của điểm ảnh hiện tại bên trong khối được cấu trúc lại hoặc khối được dự báo, trong đó các dịch vị chỉ tới các điểm lân cận bên trong khối được cấu trúc lại hoặc khối được dự báo. Dưới đây là ví dụ về mẫu quét:

(0,0), (0,1), (1,0), (1,1)

Fig.4A minh họa ví dụ về các mẫu cho vị trí điểm ảnh khác nhau bên trong khối vuông (Ví dụ, khối được dự báo CU vuông, hoặc khối được cấu trúc lại CU vuông). Theo theo hình ảnh các điểm ảnh biên có thể được lọc dựa vào việc biến đổi 4 điểm và các điểm ảnh trung tâm có thể được lọc dựa vào việc biến đổi 8 điểm.

Đối với các khối được cấu trúc lại hoặc các khối được dự báo hình chữ nhật, trong đó kích thước của một phía nhiều hơn kích thước của phía khác việc quét sẽ được thực hiện dọc phía dài. Ví dụ đối với khối hình chữ nhật theo chiều ngang thứ tự quét say đây có thể được sử dụng

(0,-3), (0,-2), (0,-1), (0,0), (0,1), (0,2), (0,3), (0,4),

trong đó trong mỗi cặp (y,x) x là dịch vị theo chiều ngang và y là dịch vị theo chiều thẳng đứng đối với vị trí của điểm ảnh lọc bên trong khối được cấu trúc lại hoặc khối được dự báo lọc.

Bộ lọc được đề xuất có thể được áp dụng có lựa chọn tùy thuộc vào các điều kiện:

- đối với các khối được cấu trúc lại hoặc các khối được dự báo có tín hiệu dư không là không;
- tùy thuộc vào kích thước khối, ví dụ, đối với khối được cấu trúc lại hoặc khối được dự báo nhỏ (kích thước tối thiểu nhỏ hơn so với ngưỡng);
- tùy thuộc vào tỷ lệ khung ảnh của khối được cấu trúc lại hoặc khối được dự báo;
- tùy thuộc vào chế độ dự báo (liên dự báo hoặc dự báo nội khung) của khối được cấu trúc lại hoặc khối được dự báo, ví dụ, bằng cách chỉ áp dụng bộ lọc tới các khối được liên dự báo; hoặc
- đối với sự kết hợp bất kỳ của các điều kiện nêu trên.

Ví dụ, để tránh khỏi việc xử lý của các khối nhỏ bộ lọc có thể được tránh qua (không được áp dụng) nếu kích thước khối nhỏ hơn hoặc bằng các điểm ảnh 4x4. Điều đó làm giảm độ phức tạp trường hợp xấu nhất mà thường tương ứng với việc xử lý khối nhỏ nhất.

Theo ví dụ khác bộ lọc được áp dụng chỉ tới các khối mà có tín hiệu dư không là không. Điều đó có lợi nếu việc lượng tử hóa hoặc các phần dư đã được sử dụng do bộ lọc được nhằm nâng cao lỗi lượng tử hóa. Nếu khối không có phần dư mà có thể chỉ báo rằng việc dự báo là tốt và không việc lọc thêm được yêu cầu.

Theo ví dụ khác, do dự báo nội khung thường kém hơn so với liên dự báo, nên bộ lọc có thể được áp dụng tới các khối được dự báo nội khung độc lập với sự có mặt của phần dư không là không và được áp dụng tới các khối được liên dự báo chỉ khi khối có tín hiệu dư không là không.

Xicma tham số bộ lọc và mẫu quét có thể thay đổi tùy thuộc vào các điều kiện được liệt kê ở trên.

Fig.4B minh họa hình dạng bộ lọc tương đương xem xét một điểm ảnh bên trong của khối hiện tại dùng cho mẫu quét ví dụ (0,0), (0,1), (1,0), (1,1

). Đối với việc lọc của điểm ảnh hiện tại vùng hình vuông của các điểm ảnh 3×3 được sử dụng (điểm ảnh hiện tại được đánh dấu bởi màu xám đen ở trung tâm của hình vuông 3×3). Điểm ảnh được lọc được thu nhận bằng cách kết hợp các mẫu được lọc miền biến đổi từ bốn nhóm 2×2 . Có thể hiểu rằng nếu điểm ảnh hiện tại được bố trí trong đường viền khối (ví dụ, đường viền trên cùng) các nhóm 2×2 trên cùng bên trái và trên cùng bên phải không khả dụng và chỉ hai nhóm 2×2 (dưới cùng bên trái và dưới cùng bên phải) có thể được sử dụng cho việc lọc. Hơn nữa nếu điểm ảnh hiện tại được bố trí ở góc khối (ví dụ, góc trên cùng bên trái) chỉ một nhóm 2×2 (dưới cùng bên phải) có thể được sử dụng cho việc lọc.

Để làm tăng chất lượng lọc bằng cách sử dụng nhiều bốn nhóm 2×2 hơn cho đường viền và các điểm ảnh góc, khối hiện tại có thể được đệm bởi các mẫu bổ sung. Fig.4C đưa ra ví dụ về việc đệm ở các phía bên trái và trên cùng. Các mẫu đệm có thể được lấy ra từ các khối được cấu trúc lại rồi.

Đối với việc thống nhất thêm của quy trình lọc cho tất cả các điểm ảnh trong khối (bốn nhóm 2×2 được sử dụng cho việc lọc của tất cả các điểm ảnh trong khối hiện tại), ngoài việc đệm phía trên cùng bên trái khối hiện tại cũng có thể được kéo dài bằng cách đệm phía dưới cùng bên phải như được minh họa trên Fig.4D. Việc thống nhất việc lọc là có lợi do đơn giản hóa cách thực hiện bằng cách loại trừ các trường hợp xử lý đặc biệt cho các điểm ảnh góc và/hoặc các điểm ảnh đường viền.

Các mẫu đệm tốt hơn là được lấy từ các mẫu lân cận được điều chỉnh từ các khối được cấu trúc lại rồi. Theo các bộ mã hóa-giải mã video trong lĩnh vực kỹ thuật đã biết các khối được cấu trúc lại rồi đó có thể được bố trí hoặc ở phía bên trái hoặc trên cùng từ khối hiện tại hoặc ở phía bên phải hoặc dưới cùng tùy thuộc vào thứ tự cấu trúc lại khối. Bằng cách sử dụng nhiều thông tin hơn từ các mẫu điều chỉnh, nó nâng cao chất lượng lọc và thực hiện sự dịch chuyển giữa các khối trơn tru hơn.

Việc truy xuất các mẫu được cấu trúc lại từ các khối được điều chỉnh hoặc các khối được cấu trúc lại trước đó có thể yêu cầu tải bộ nhớ bổ sung cho

cách thực hiện phần cứng hoặc phần mềm. Để giảm thiểu hoặc loại trừ bộ nhớ bổ sung, điều có lợi là sử dụng các mẫu được dự định cho việc dự báo nội khung của khối hiện tại mà được lấy chung từ một, hai hoặc nhiều hàng và cột từ các khối lân cận được điều chỉnh tới các đường viền khối hiện tại. Các mẫu này thường được lưu trữ trong bộ nhớ nhanh (cũng được biết đến là bộ đệm “dòng”) để truy cập dễ dàng cho việc dự báo nội khung và được gọi là các mẫu tham chiếu của việc dự báo nội khung.

Cần lưu ý thêm rằng theo một số cách thực hiện, trước khi thực hiện việc dự báo nội khung, mẫu tham chiếu (các mẫu tham chiếu nội khung) được xử lý trước trước khi dự báo chẳng hạn, bằng cách làm mịn, làm sắc, khử dao động hoặc lọc hai chiều. Trong trường hợp này điều có thể có lợi là sử dụng các mẫu được xử lý cho việc đệm của khối hiện tại.

Nếu một số mẫu trong vùng được đệm không khả dụng, do thứ tự của việc cấu trúc lại khối được điều chỉnh, các mẫu được yêu cầu có thể được đệm từ khối hiện tại mở rộng các điểm ảnh đường viền tới vùng được đệm như được minh họa trên Fig.4D.

Fig.5A thể hiện sơ đồ trình tự minh họa các bước của phương pháp lọc trong vòng lặp 500 tương ứng theo một phương án. Khối được cấu trúc lại bao gồm các điểm ảnh. Phương pháp 500 bao gồm các bước sau đây: tải (502) điểm ảnh hiện tại và các điểm ảnh lân cận của nó tới bộ đệm tuyến tính theo mẫu quét được xác định trước; thu nhận (504) các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu nhận (506) các thành phần phổ được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại tùy thuộc vào thành phần phổ tương ứng và tham số lọc; thu nhận (508) các điểm ảnh được lọc bằng cách thực hiện biến đổi 1D ngược cho các thành phần phổ được lọc; và tạo ra (510) khối được cấu trúc lại được lọc dựa vào các điểm ảnh được lọc được đánh giá trên các bước xử lý trước đó. Phương pháp 500 có thể được thực hiện bởi máy mã hóa như được thể hiện trên Fig.1 và máy giải mã như được thể hiện trên Fig.2. Thông tin chi tiết 300 trên Fig.3A hoặc

thông tin 300' trên Fig.3B cũng được áp dụng tới phương pháp 500 như được thể hiện trên Fig.5A.

Tương tự Fig.5A, Fig.5B thể hiện sơ đồ trình tự minh họa các bước của phương pháp lọc trong vòng lặp 500' tương ứng theo khía cạnh khác. Ở ví dụ này, khối (hoặc khung) là khối được dự báo, và khối được lọc là khối được dự báo được lọc. Phần mô tả chi tiết trên Fig.5B tương tự Fig.5A.

Cách thực hiện phần cứng được thảo luận.

Bộ lọc miền biến đổi Hadamard được đặt chỉ sau khi cấu trúc lại khối và xử lý các mẫu mà có thể được bao gồm trong việc cấu trúc lại các khối tiếp theo cụ thể là các mẫu tham chiếu của việc dự báo nội khung. Vì vậy độ trễ được giới thiệu bởi bộ lọc cần giảm thiểu để đảm bảo toàn bộ ống dẫn cấu trúc lại không bị ảnh hưởng nhiều.

Biến đổi Hadamard được coi là tương đối đơn giản đối với cách thực hiện phần cứng. Chỉ các phép cộng được yêu cầu cho cách thực hiện của nó mà không phải các phép nhân. Như có thể nhận thấy từ mã giả 1 dưới đây, việc biến đổi về phía trước và về phía sau chứa 4 phép cộng mà có thể được thực hiện song song hoặc sử dụng lại các kết quả trung gian với hai phép cộng liên tiếp.

Mã giả 1

```

for (int r = 0; r < height - 1; ++r)
{
    Pel *pIn = &block[r*strideBlk];
    Pel *pAcc = &accBlock[r*strideAcc];

    for (int c = 0; c < width - 1; ++c, pIn++, pOut++)
    {
        const int x0 = pIn[p0];
        const int x1 = pIn[p1];
        const int x2 = pIn[p2];
        const int x3 = pIn[p3];

        // forward transform
        const int y0 = x0 + x2;
        const int y1 = x1 + x3;
        const int y2 = x0 - x2;
        const int y3 = x1 - x3;

        const int t0 = y0 + y1;
        const int t1 = y0 - y1;
        const int t2 = y2 + y3;
        const int t3 = y2 - y3;

        // filtering
        const int z0 = t0; // skip DC
        const int z1 = RdTbl(t1, tbl, thr);
        const int z2 = RdTbl(t2, tbl, thr);
        const int z3 = RdTbl(t3, tbl, thr);

        // backward transform
        const int iy0 = z0 + z2;
        const int iy1 = z1 + z3;
        const int iy2 = z0 - z2;
        const int iy3 = z1 - z3;

        pAcc[p0_out] += ((iy0 + iy1) >> HTDF_BIT_RND4);
        pAcc[p1_out] += ((iy0 - iy1) >> HTDF_BIT_RND4);
        pAcc[p2_out] += ((iy2 + iy3) >> HTDF_BIT_RND4);
        pAcc[p3_out] += ((iy2 - iy3) >> HTDF_BIT_RND4);

        // normalization
        pIn[p0] = ClipPel((pOut[p0_out] + HTDF_CNT_SCALE_RND) >> HTDF_CNT_SCALE, clipRng);
    }
}

```

Biến đổi Hadamard về phía trước và về phía sau có thể được thực hiện ở phần cứng bằng cách sử dụng logic kết hợp. Cần chú ý nhiều hơn để truy cập nhanh và song song tới LUT.

LUT dựa vào SRAM được thảo luận.

Theo cách thực hiện ví dụ này, LUT được lưu trữ trong RAM tĩnh cổng đơn trên chip (Fig.6).

Khi dữ liệu từ bước xử lý trước đó khả dụng trong bộ đệm bởi sườn lên của đồng hồ nó được truy cập bởi logic kết hợp thực hiện biến đổi Hadamard về phía trước (chứa hai phép cộng liên tiếp). Sau khi hoàn tất logic kết hợp địa chỉ là khả dụng cho mỗi LUT. Bằng cách sử dụng **biến tần** và sườn xuống của đồng hồ dữ liệu được truy cập từ SRAM. Logic kết hợp thứ hai thực hiện biến đổi

Hadamard về phía sau và việc chuẩn hóa được bắt đầu ngay sau khi dữ liệu từ LUT khả dụng. Các mẫu được lọc đầu ra đã trở nên khả dụng ở cuối của chu kỳ đồng hồ hiện tại và sẵn sàng để xử lý bởi thuật toán tiếp theo ở sườn lên tiếp theo của đồng hồ.

LUT dựa vào mạch lật được thảo luận.

Xem xét rằng một bảng cho quy trình lọc được giới hạn bởi 16 nhập mục nó trông hiệu quả hơn khi thực hiện LUT dựa vào các mạch lật. Thiết kế như vậy không yêu cầu nhiều LUT cho việc xử lý song song và sườn đồng hồ cho việc truy cập dữ liệu. Truy cập song song được bố trí bởi bộ đa hợp như được mô tả trên Fig.7 minh họa thiết kế ví dụ để xử lý một nhóm 2x2. Theo thiết kế được gợi ý 16 mạch lật của 7 bit được yêu cầu để bố trí việc truy cập song song trong suốt quy trình lọc. LUT QP cụ thể có thể được tải tới các mạch lật khi QP dùng cho CU hiện tại khả dụng.

Việc kết hợp các kết quả của việc lọc bốn nhóm 2x2 với việc sử dụng lại các kết quả của cùng nhóm không gian đầu ra bộ lọc cuối cùng được tạo ra như được mô tả trên Fig.8.

Phân tích nêu trên cho phép đi đến kết luận rằng bộ lọc được đề xuất có thể được thực hiện ở phần cứng nằm trong một khối bằng cách sử dụng hoặc SRAM hoặc cách thực hiện LUT dựa vào mạch lật.

Phân tích độ phức tạp được thảo luận.

Phép đo tác động tới Bitrate/PSNR tương ứng với (các) neo.

Phân tích độ phức tạp (ví dụ, phép đo thời gian mã hóa và giải mã, phân tích độ phức tạp bằng cách điền bảng dưới đây).

Thử nghiệm#	hình dạng bộ lọc	Độ phức tạp tính toán (# của các phép nhân/các phép cộng/các sự dịch chuyển/các kiểm tra*) trên mỗi mẫu	Độ chính xác của phép nhân	Thân thiện song song (mỗi mẫu có thể được lọc độc lập từ các mẫu khác)	Số lượng của các chu kỳ đồng hồ	yêu cầu bộ nhớ	Cách thức để dẫn ra các hệ số bộ lọc	Kích thước CU nhỏ nhất để áp dụng các bộ lọc
14.3a	3x3	0/20+4 bit I bổ sung cho vòng/5/6	không có dữ liệu	đúng	< 1 đồng hồ	140 byte (32 tri số 7 bit trên mỗi nhóm qp)	Được tính trước trong LUT	4x8, 8x4
14.3b	3x3	0/20+4 bit I bổ sung cho vòng/5/6	không có dữ liệu	đúng	< 1 đồng hồ	70 byte (16 giá trị 7 bit trên mỗi nhóm qp)	Được tính trước trong LUT	4x8, 8x4

* max/min/abs được đếm khi kiểm tra

Bảng 1. CE14-3 tóm tắt phân tích độ phức tạp

Các kết quả thí nghiệm được thảo luận.

Các kết quả khách quan được thảo luận.

Hiệu suất khách quan được thể hiện ở các bảng sau đây:

Toàn bộ Bo mạch chủ (Main) 10 bên trong					
Qua BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Loại A1	-0,19%	0,29%	0,24%	110%	115%
Loại A2	-0,35%	0,31%	0,32%	109%	112%
Loại B	-0,42%	0,29%	0,32%	109%	113%
Loại C	-0,83%	0,23%	0,42%	110%	110%
Loại E	-0,54%	0,28%	0,19%	109%	114%
Tổng cộng	-0,48%	0,28%	0,31%	109%	112%
Loại D	-0,61%	0,47%	0,01%	109%	108%
Loại F (tùy chọn)	-0,82%	0,07%	0,04%	108%	107%

Main 10 truy cập ngẫu nhiên					
Qua BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Loại A1	-0,43%	-0,23%	-0,25%	105%	104%
Loại A2	-0,92%	-0,57%	-0,43%	105%	103%
Loại B	-0,69%	-0,02%	-0,27%	106%	104%
Loại C	-0,75%	0,10%	0,14%	105%	104%
Loại E					
Tổng cộng	-0,70%	-0,14%	-0,19%	105%	104%
Loại D	-0,67%	-0,14%	0,53%	104%	103%
Loại F (tùy chọn)	-0,77%	0,10%	0,13%	104%	102%

Main 10 B độ trễ thấp					
Qua BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Loại A1					
Loại A2					
Loại B	-0,58%	0,52%	0,54%	104%	104%
Loại C	-0,74%	0,33%	0,52%	104%	104%
Loại E	-0,75%	0,31%	1,09%	102%	101%
Tổng cộng	-0,68%	0,40%	0,67%	104%	103%
Loại D	-0,90%	0,75%	0,28%	104%	104%
Loại F (tùy chọn)	-0,73%	0,20%	-0,09%	103%	102%

Bảng 2. Hiệu suất lập mã của thử nghiệm 14-3a

	Toàn bộ Main 10 bên trong				
	Qua BMS-2.0.1 với cấu hình VTM				
	Y	U	V	EncT	DecT
Loại A1	-0,17%	0,28%	0,28%	110%	115%
Loại A2	-0,33%	0,27%	0,29%	109%	112%
Loại B	-0,40%	0,31%	0,39%	109%	113%
Loại C	-0,82%	0,20%	0,32%	109%	110%
Loại E	-0,54%	0,34%	0,25%	109%	113%
Tổng cộng	-0,47%	0,28%	0,32%	109%	112%
Loại D	-0,65%	0,43%	0,22%	109%	108%
Loại F (tùy chọn)	-0,81%	0,01%	0,12%	108%	107%

	Main 10 truy cập ngẫu nhiên				
	Qua BMS-2.0.1 với cấu hình VTM				
	Y	U	V	EncT	DecT
Loại A1	-0,41%	-0,36%	-0,24%	105%	105%
Loại A2	-0,92%	-0,49%	-0,37%	105%	103%
Loại B	-0,69%	-0,12%	0,12%	106%	104%
Loại C	-0,75%	0,14%	0,10%	105%	104%
Loại E					
Tổng cộng	-0,70%	-0,17%	-0,06%	105%	104%
Loại D	-0,69%	-0,30%	0,49%	104%	103%
Loại F (tùy chọn)	-0,82%	0,11%	0,11%	104%	102%

	Main 10 B độ trễ thấp				
	Qua BMS-2.0.1 với cấu hình VTM				
	Y	U	V	EncT	DecT
Loại A1					
Loại A2					
Loại B	-0,57%	0,49%	0,69%	104%	104%
Loại C	-0,76%	0,31%	0,42%	104%	106%
Loại E	-0,67%	-0,66%	-0,48%	102%	102%
Tổng cộng	-0,66%	0,14%	0,31%	103%	104%
Loại D	-0,88%	0,92%	0,31%	103%	105%
Loại F (tùy chọn)	-0,74%	0,38%	-0,15%	103%	102%

Bảng 3. Hiệu suất lập mã của thử nghiệm 14-3b

70 byte LUT được gợi ý với 16 byte trên mỗi QP, cho phép cách thực hiện HW 1 đồng hồ, v.v. được đề xuất chấp nhận bộ lọc miền biến đổi Hadamard tới phiên bản tiếp theo của VTM.

Các phần tham chiếu sau đây được kết hợp ở đây nhằm viện dẫn như thể được sao chép toàn bộ:

Tài liệu nhóm chuyên gia video liên kết (Joint Video Experts Team, viết tắt là JVET) JVET-K0068.

Phần dưới đây minh họa các ví dụ để tối ưu hóa LUT.

Theo một ví dụ 1, tập hợp của các tham số lượng tử hóa (quantization parameter, viết tắt là QP) được chọn để tạo ra bảng tra cứu (LUT), trong đó tập hợp của các QP bao gồm QP thứ nhất tương ứng với chỉ số (i) và QP thứ hai tương ứng với chỉ mục (i+1), và QP thứ nhất và QP thứ hai này có khoảng cách không đổi. Ví dụ, khoảng có thể bằng 8, 10 hoặc 16.

Ví dụ, lấy khoảng không đổi là 8 theo ví dụ, bảng thừa bằng cách có LUT cho $qp = \{20, 28, 36, 44, 52\}$. Khoảng giữa gp thứ nhất 20 và gp thứ hai 28 là 8. Ngoài ra, khoảng giữa gp thứ hai 28 và gp thứ ba 36 là 8. Trong suốt thời gian lọc bằng với QP sát nhất được chọn.

Theo ví dụ khác, lấy khoảng không đổi là 8 theo ví dụ, bảng thừa bằng cách có LUT cho $qp = \{18, 26, 34, 42, 50\}$. Khoảng giữa gp thứ nhất 18 và gp thứ hai 26 là 8. Ngoài ra, khoảng giữa gp thứ hai 26 và gp thứ ba 34 là 8. Trong suốt thời gian lọc bằng với QP sát nhất được chọn.

Kích thước LUT: $5 \times 128 = 640$ byte

Dưới đây là mã giả 2 phản ánh các QP nào được chọn để tạo ra các bảng tra cứu (LUT).

mã giả 2

```

#define HTDF_MIN_QP                                20
#define HTDF_MAX_QP                                (MAX_QP+1)
#define HTDF_QP_ROUND                               8
for (int qp_in = HTDF_MIN_QP; qp_in < HTDF_MAX_QP; qp_in++)
{
    int idx = ( (qp_in - HTDF_MIN_QP) + (HTDF_QP_ROUND>>1) ) / HTDF_QP_ROUND;
    int qp = idx*HTDF_QP_ROUND + HTDF_MIN_QP;

    if (idxPrev != idx)
    {
        tblGen(HTDFTable[idx], qp);
        idxPrev = idx;
    }
}

```

Ở mã giả, HTDF_QP_ROUND biểu diễn khoảng không đổi. Có khoảng là công suất của hai thuận lợi để cho phép thực hiện phép tính chia đối với việc tính toán chỉ số như sự dịch chuyển. Cần lưu ý rằng các trị số khác nhau của khoảng không đổi có thể được chọn ví dụ, 2, 4, 10, 15 hoặc 16 v.v. Hơn nữa như theo phương án khác khoảng có thể là tùy chọn và LUT nào được tính toán cho tập hợp tùy chọn của các QP.

Trong suốt quy trình lọc chỉ số tương ứng LUT cho QP được đưa ra được tính toán là:

```
int idx = ((qp - HTDF_MIN_QP) + (HTDF_QP_ROUND >> 1)) / HTDF_QP_ROUND;
```

hoặc theo cách khác với độ chính xác ít hơn:

```
int idx = (qp - HTDF_MIN_QP) / HTDF_QP_ROUND;
```

Nếu khoảng không đổi là công suất của 2, ví dụ, sau đó chỉ số của LUT có thể được tính toán thuận lợi bằng cách sử dụng phép tính dịch chuyển thay vì chia:

```
int idx = (qp - HTDF_MIN_QP) >> HTDF_QP_ROUND_LOG2 = (qp - HTDF_MIN_QP) >> 3;
```

Theo ví dụ khác 2, bộ giải mã hoặc bộ mã hóa giảm N các bit từ các giá trị bảng, và N là số nguyên. Điều đó cho phép có sự biểu diễn LUT thưa lưu trữ chỉ các trị số được lựa chọn từ khoảng được đưa ra. Ví dụ, N là 3. Trị số bảng lớn nhất là 127 (7 bit), giảm 3 bit, kết quả là 4 bit, mà là 16 nhập mục của khoảng các trị số 7 bit - 16 byte.

Dưới đây là mã giả 3 mô tả cách thức LUT được tạo ra theo giá trị QP được đưa ra.

mã giả 3

```
#define HTDF_TBL_SH          3
#define HTDF_TBL_RND        ((1<<HTDF_TBL_SH)>>1)

void HTDFilter::tblGen(HTDF_TBL_TYPE *tbl, int qp)
{
    float sigma = (2.64f)*powf(2.0f, (0.1269f)*(qp - 11));
    int sigma_l = (int)(sigma * HTDF_SIGMA_SCALE );

    sigma_l *= sigma_l;
    sigma_l *= 4;

    for (int k_in = 0; k_in < HTDF_SHORT_TBL_THR; k_in++)
    {
        const int Idx = (k_in + HTDF_TBL_RND)>>HTDF_TBL_SH;
        const int k = Idx << HTDF_TBL_SH;

        unsigned int num = (k*k*k) << 10;
        int den = ((k*k) << 10) + sigma_l;
        int value_long = num / den;
        HTDF_TBL_TYPE value_int = (HTDF_TBL_TYPE)value_long;

        tbl[Idx] = value_int;
    }
}
```

Theo ví dụ được đưa ra HTDF_TBL_SH định rõ số lượng của bit để giảm mà có thể là 1, 2, 3, 4 v.v.

Dưới đây là mã giả minh họa việc truy cập tới LUT thưa trong suốt thời gian lọc:

```
tbl[(z + HTDF_TBL_RND)>> HTDF_TBL_SH]
```

Khi kết hợp các ví dụ 1 và 2 nêu trên, Fig.9 theo ví dụ, thể hiện kết quả tối ưu hóa LUT. Như được thể hiện trên Fig.9, kết hợp các ví dụ 1 và 2: 5 bảng x 16 nhập mục x 7 bit = 560 bit = 70 byte.

Cần lưu ý rằng lượng của các nhập mục LUT được định rõ bởi HTDF_SHORT_TBL_THR (ngưỡng giá trị từ đoạn [0076]) và số lượng HTDF_TBL_SH của các bit được giảm. Xem xét ngưỡng bằng 128 (nghĩa là $1 \ll 7$) và các bit được giảm 3 đưa ra số lượng các bảng các nhập mục bằng $1 \ll (7 - 3) = 1 \ll 4 = 16$. Như mô tả ở trên ngưỡng LUT tốt hơn là được chọn để tạo ra kết quả của phương trình từ đoạn [0077] gần bằng 1 và như cũng được mô tả ở đó ngưỡng có thể khác nhau tùy thuộc vào QP. Vì vậy đối với việc tạo ra LUT cho các giá trị QP cao hơn điều có thể có lợi là làm tăng ngưỡng từ 128 ($1 \ll 7$) tới ví dụ, 256 ($1 \ll 8$). Trong trường hợp đó việc giữ cùng độ chính xác của các nhập mục LUT (ví dụ, với 3 bit được giảm) sẽ yêu cầu 32 nhập mục ($32 = 1 \ll (8 - 3) = 1 \ll 5$). Theo cách khác để giữ cùng kích thước LUT đối với các QP cao hơn như đối với các QP thấp hơn độ chính xác có thể còn được giảm tới 4 để giữ 16 nhập mục của bảng $16 = 1 \ll (8 - 4) = 1 \ll 4$.

Theo một vài cách thực hiện việc giữ kích thước LUT được giới hạn và có THR để giữ phương trình từ đoạn [0077] có thể mâu thuẫn. Thực vậy các giá trị số QP cao (mà dẫn tới giá trị số σ cao) giữ kích thước LUT được giới hạn bởi ví dụ, 16 nhập mục có thể dẫn tới khoảng trống không mong muốn trong chức năng chuyển đổi bộ lọc (mà là được biểu diễn bởi LUT) xung quanh giá trị 120 như được mô tả trên Fig.10 (mà cũng bao gồm phương pháp lấy mẫu phụ LUT bằng cách giảm hết 3 bit ít quan trọng nhất).

Fig.11 biểu diễn cùng bảng với các nhập mục bảng là từng biểu đồ mà không xem xét khoảng trống minh họa hiệu quả lấy mẫu phụ LUT giữa nhập mục LUT cuối cùng ở chỉ mục 15 và điểm tiếp theo trong chức năng chuyển đổi bộ lọc ở chỉ mục 16.

Fig.12 minh họa phương pháp về cách thức khoảng trống có thể được loại bỏ bằng cách sử dụng chức năng phụ đi qua giá trị THR ở đối số của tương ứng với nhập mục LUT + 1 cuối cùng, ví dụ, bằng cách sử dụng phương trình đường thẳng (được chỉ ra bởi màu xanh), đi qua điểm THR=128 ở đối số 16 mà tương ứng với đối số của nhập mục LUT cuối cùng (mà là 15) được tăng thêm 1

hoặc nói cách khác kích thước LUT (mà là 16) và đi qua trục x ở một số điểm (ví dụ, ở giá trị 11 giống như theo ví dụ được đưa ra). Cần lưu ý rằng các loại khác nhau của các chức năng phụ có thể được sử dụng dựa vào cùng nguyên tắc, ví dụ, hàm mũ, logarit, parabol v.v. hoặc sự kết hợp của chúng.

Fig.13 minh họa ví dụ về việc loại bỏ khoảng trống bằng cách lấy trị số cao nhất từ hai trị số trong khi tạo ra LUT, trong đó giá trị thứ nhất là nhập mục LUT như được nêu trên $LUT(R_i, \sigma) = \frac{R_i^3}{R_i^2 + m * \sigma^2}$; và giá trị thứ hai là giá trị của chức năng phụ (đường thẳng theo ví dụ này) ở cùng đối số i:

$$LUT(R_i, \sigma) = \max\left(\frac{R_i^3}{R_i^2 + m * \sigma^2}, AuxiliaryFunc_{\sigma}(i)\right);$$

trong đó $AuxiliaryFunc_{\sigma}(i)$ biểu diễn chức năng phụ, và chức năng phụ có giá trị bằng THR ở i tương ứng với đối số của LUT + 1 cuối cùng.

Fig.14 minh họa chức năng chuyển đổi bộ lọc sau khi áp dụng phương pháp nêu trên của bằng cách sử dụng phương trình đường thẳng phụ và lấy mẫu phụ LUT bằng cách giảm 3 bit ít quan trọng nhất.

Như được mô tả trong đoạn [00144] một LUT có thể được sử dụng cho nhóm của các QP. Để bao hàm khoảng QP có thể tập hợp QP định trước được sử dụng và một LUT được tạo ra cho mỗi QP từ tập hợp. Fig.15 minh họa ví dụ về các chức năng chuyển đổi bộ lọc tùy thuộc vào năm QP trong tập hợp và các bảng tương ứng (từ 0 tới 4). Theo ví dụ này bảng 4 được tạo ra bằng cách sử dụng phương pháp được mô tả trong các đoạn [00156]-[00157] với chức năng phụ đường thẳng đi qua trục x ở giá trị là 11; và bảng 3 được tạo ra bằng cách sử dụng cùng phương pháp với chức năng phụ đường thẳng đi qua trục x ở giá trị là 9. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

$$\text{bảng0} = \{ 0, 2, 10, 19, 28, 36, 45, 53, 61, 70, 78, 86, 94, 102, 110, 118, \},$$

$$\text{bảng1} = \{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, \},$$

$$\text{bảng2} = \{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$$

$$\text{bảng3} = \{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, \},$$

$$\text{bảng4} = \{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, \},$$

Như được nêu trên trong các đoạn [00150]-[00153] phương pháp lấy mẫu phụ bảng có thể giảm N các bit từ các giá trị bảng để làm giảm kích thước bảng. Cũng như được nêu trong các đoạn này N có thể khác nhau tùy thuộc vào QP được sử dụng để tạo ra bảng nhất định và giá trị THR được lựa chọn cho bảng này. Ví dụ đối với các giá trị QP thấp hơn tham số lọc σ tương đối thấp hơn so với đối với các QP cao hơn. Vì vậy giá trị tuyệt đối của THR có thể được giảm mà không mất hiệu suất. Hơn nữa để giữ kích thước bảng giống nhau cho tất cả các QP trong tập hợp (điều đó có lợi cho việc đơn giản hóa cách thực hiện) và làm giảm tốc độ lấy mẫu phụ đối với các QP thấp hơn (mà tương ứng đối với mức nén ít hơn và chất lượng hình ảnh được cấu trúc lại tốt hơn) điều có thể có lợi là làm giảm lượng N các bit được giảm khi so sánh với các bảng QP khác, ví dụ, bằng cách thiết đặt N cho QP thấp hơn bằng 2 và thiết đặt THR tới 64. Fig.16 minh họa ví dụ về các chức năng chuyển đổi bộ lọc cho năm QP trong tập hợp dựa vào các bảng tương ứng (từ 0 tới 4) với N được thiết đặt tới 2 đối với bảng thứ nhất (tương ứng với khoảng QP thấp hơn) và N được thiết đặt tới 3 đối với các bảng khác. Ví dụ này cũng bao gồm phương pháp sử dụng *Chức năng phụ* đối với việc tạo ra bảng 3 và bảng 4 như được mô tả trong các đoạn [00159]-[00160]. Đối với bảng 4 chức năng phụ đường thẳng đi qua trục x ở giá trị là 11. Đối với bảng 4 chức năng phụ đường thẳng đi qua trục x ở giá trị là 9. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

$$\text{bảng0} = \{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, \},$$

$$\text{bảng1} = \{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, \},$$

$$\text{bảng2} = \{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$$

$$\text{bảng3} = \{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, \},$$

$$\text{bảng4} = \{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, \},$$

Như được nêu trên trong các đoạn [00150]-[00153] phương pháp lấy mẫu phụ bảng có thể giảm N các bit từ các giá trị bảng để làm giảm kích thước bảng. Cũng như được nêu trong các đoạn này N có thể khác nhau tùy thuộc vào QP được sử dụng để tạo ra bảng nhất định và giá trị THR được lựa chọn cho bảng này. Ví dụ đối với các giá trị QP cao hơn tham số lọc σ tương đối cao hơn so với đối với các QP thấp hơn mà có thể yêu cầu làm tăng giá trị THR để giữ phương trình từ đoạn [0077] sát hơn với 1. Đồng thời để giữ kích thước LUT giống nhau cho tất cả các QP trong tập hợp (mà có lợi do đơn giản hóa cách thực hiện) và cũng xem xét rằng đối với các giá trị QP cao hơn hình ảnh được cấu trúc lại có nhiều biến dạng hơn và việc làm tăng việc lấy mẫu phụ của LUT là chấp nhận được do ảnh hưởng chủ quan không thể nhận thấy của việc lấy mẫu phụ LUT với sự có mặt của các đối tượng nén mạnh giá trị N của các bit ít quan trọng nhất được giảm có thể được làm tăng tới 4 chẳng hạn, đối với bảng cuối cùng và đối với bảng cuối cùng thứ hai trong tập hợp. Fig.17 minh họa ví dụ về các chức năng chuyển đổi bộ lọc cho năm QP trong tập hợp dựa vào các bảng tương ứng (từ 0 tới 4) với N được thiết đặt tới 2 đối với bảng thứ nhất (bảng 0 tương ứng với khoảng QP thấp hơn), N được thiết đặt tới 4 đối với bảng cuối cùng và bảng cuối cùng thứ hai (bảng 3 và bảng 4) và N được thiết đặt tới 3 đối với các bảng khác. Theo ví dụ này TRH được thiết đặt tới 64 dùng để tạo ra bảng thứ nhất, được thiết đặt tới 256 đối với bảng cuối cùng thứ hai cuối cùng và được thiết đặt tới 128 đối với các bảng còn lại. Ví dụ này cũng bao gồm phương pháp sử dụng *Chức năng phụ* đối với việc tạo ra bảng 3 và bảng 4 như được mô tả trong các đoạn [00159]-[00160]. Đối với bảng 4 chức năng phụ đường thẳng đi qua trục x ở giá trị là 6. Đối với bảng 4 chức năng phụ đường thẳng đi qua trục x ở giá trị là 8. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

$$\text{bảng0} = \{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, \},$$

$$\text{bảng1} = \{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, \},$$

$$\text{bảng2} = \{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$$

$$\text{bảng3} = \{ 0, 0, 3, 9, 19, 32, 47, 64, 81, 99, 117, 135, 154, 179, 205, 230, \},$$

bảng4 = { 0, 0, 0, 2, 6, 11, 18, 27, 38, 51, 64, 96, 128, 160, 192, 224, },

Fiig.18 là sơ đồ khối của máy 600 mà có thể được sử dụng để thực hiện các phương án khác nhau. Máy 600 có thể là máy mã hóa như được thể hiện trên Fig.1 và máy giải mã như được thể hiện trên Fig.2. Ngoài ra, máy 600 có thể chứa một hoặc nhiều trong số các thành phần được mô tả. Theo một số phương án, máy 600 được trang bị một hoặc nhiều thiết bị đầu vào/đầu ra, chẳng hạn như loa, micro, chuột, màn hình cảm ứng, bàn phím nhỏ, bàn phím, máy in, màn hiển thị, và tương tự. Máy 600 có thể bao gồm một hoặc nhiều bộ xử lý trung tâm (central processing unit, CPU) 610, bộ nhớ 620, bộ phận lưu trữ cỡ lớn 630, bộ chuyển đổi video 640, và giao diện I/O 660 được kết nối với kênh truyền. Kênh truyền là một hoặc nhiều trong số loại bất kỳ của nhiều loại kiến trúc kênh truyền bao gồm kênh truyền bộ nhớ hoặc bộ điều khiển bộ nhớ, kênh truyền ngoại vi, kênh truyền video, hoặc tương tự.

CPU 610 có thể có loại bộ xử lý dữ liệu điện tử bất kỳ. Bộ nhớ 620 có thể có, hoặc là, loại bộ nhớ hệ thống bất kỳ chẳng hạn như bộ nhớ truy cập ngẫu nhiên tĩnh (static random access memory, SRAM), bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory, DRAM), DRAM đồng bộ (synchronous DRAM, SDRAM), bộ nhớ chỉ đọc (read-only memory, ROM), sự kết hợp của chúng, hoặc tương tự. Theo phương án, bộ nhớ 620 có thể bao gồm ROM để sử dụng khi khởi động, và DRAM dùng cho việc lưu trữ chương trình và dữ liệu để sử dụng trong khi thực hiện các chương trình. Theo các phương án, bộ nhớ 620 là không chuyển tiếp. Bộ phận lưu trữ cỡ lớn 630 bao gồm loại thiết bị lưu trữ bất kỳ mà lưu trữ dữ liệu, các chương trình, và thông tin khác và làm cho dữ liệu, các chương trình, và thông tin khác có thể truy cập qua kênh truyền. Bộ phận lưu trữ cỡ lớn 630 bao gồm, ví dụ, một hoặc nhiều trong số ổ đĩa trạng thái rắn, ổ đĩa cứng, ổ đĩa từ, ổ đĩa quang, hoặc tương tự.

Bộ chuyển đổi video 640 và giao diện I/O 660 cung cấp các giao diện để ghép nối các thiết bị đầu vào và đầu ra bên ngoài với máy 600, Ví dụ, máy 600 có thể cung cấp giao diện lệnh SQL tới các khách. Như được minh họa, các ví dụ

về các thiết bị đầu vào và đầu ra bao gồm màn hiển thị 690 được ghép nối với bộ chuyển đổi video 640 và sự kết hợp bất kỳ của chuột/bàn phím/máy in 670 được ghép nối với giao diện I/O 660. Các thiết bị khác có thể được ghép nối với máy 600, và các thẻ giao diện bổ sung hoặc ít hơn có thể được sử dụng. Ví dụ, thẻ giao diện nối tiếp (không được thể hiện) có thể được sử dụng để cung cấp giao diện nối tiếp cho máy in.

Máy 600 cũng bao gồm một hoặc nhiều giao diện mạng 650, mà bao gồm các liên kết nối dây, chẳng hạn như cáp Ethernet hoặc tương tự, và/hoặc các liên kết không dây để truy cập các nút hoặc một hoặc nhiều mạng 680, giao diện mạng 650 cho phép máy 600 truyền thông với các bộ phận từ xa qua mạng 680. Ví dụ, giao diện mạng 650 có thể cung cấp việc truyền thông tới cơ sở dữ liệu. Theo phương án, máy 600 được ghép nối với mạng vùng cục bộ hoặc mạng vùng rộng dùng cho việc xử lý dữ liệu và các việc truyền thông với các thiết bị từ xa, chẳng hạn như các bộ xử lý khác, Internet, các cơ sở lưu trữ từ xa, hoặc tương tự.

Thiết kế của bộ lọc trong vòng lặp hoặc bộ lọc dự báo được đề xuất có các ưu điểm sau đây đối với các phương pháp lọc thích nghi thông thường chẳng hạn như ALF:

- Bộ lọc miền tần số được đề xuất dẫn ra các tham số lọc (các hệ số khuếch đại miền tần số) từ khung được cấu trúc lại hoặc khối được dự báo ở phía bộ giải mã và vì vậy các tham số lọc không được yêu cầu được chuyển đổi từ bộ mã hóa tới phía bộ giải mã.
- ALF yêu cầu tối ưu hóa tốc độ biến dạng (rate distortion optimization, RDO) phức tạp ở phía bộ mã hóa để làm giảm số lượng các hệ số trọng số dùng cho việc truyền. Phương pháp được đề xuất không yêu cầu RDO phức tạp ở phía bộ mã hóa (không chuyển đổi các thông số) và được áp dụng cho tất cả các khối mà đáp ứng các điều kiện định trước
- ALF là bộ lọc tuyến tính trong miền điểm ảnh. Bộ lọc được đề xuất là không tuyến tính do hệ số khuếch đại cho mỗi thành phần phổ 1D tùy thuộc vào

giá trị thành phần phổ này. Nó cho phép đạt được độ khuếch đại mã hóa bổ sung từ việc xử lý không tuyến tính.

- ALF yêu cầu các số nhân phổ quát ở phía bộ giải mã. Theo phương pháp được đề xuất việc lọc có thể được thực hiện như bảng tra cứu, do độ khuếch đại cho mỗi hệ số phổ nhỏ hơn một. Do đó phương pháp được đề xuất có thể được thực hiện mà không có phép nhân bất kỳ.

Vì vậy, bộ lọc được đề xuất cho phép nâng cao hiệu quả lập mã video với độ phức tạp thấp.

Trong khi dấu hiệu hoặc khía cạnh cụ thể của sáng chế có thể đã được bộc lộ dựa vào chỉ một trong số nhiều cách thực hiện hoặc các phương án, dấu hiệu hoặc khía cạnh như vậy có thể được kết hợp với một hoặc nhiều dấu hiệu hoặc khía cạnh khác của các cách thực hiện hoặc các phương án khác khi có thể được mong muốn và có lợi cho ứng dụng cụ thể hoặc được đưa ra bất kỳ. Ngoài ra, trong phạm vi các thuật ngữ "bao gồm", "có", "với", hoặc các biến thể khác của chúng được sử dụng trong hoặc phần mô tả chi tiết hoặc các điểm yêu cầu bảo hộ, các thuật ngữ như vậy được dự định để bao gồm theo cách giống như thuật ngữ "bao gồm". Ngoài ra, các thuật ngữ "ví dụ (exemplary)", "ví dụ (for example)" và "ví dụ (e.g.)," chỉ có nghĩa là theo ví dụ, không là tốt nhất hoặc tối ưu. Các thuật ngữ "được ghép nối" và "được kết nối", cùng với các phát sinh có thể đã được sử dụng. Cần hiểu rằng các thuật ngữ này có thể đã được sử dụng để chỉ báo rằng hai thành phần kết hợp hoặc tương tác với nhau không quan tâm đến việc xem chúng tiếp xúc vật lý hay điện trực tiếp, hay chúng không tiếp xúc trực tiếp với nhau.

Mặc dù các khía cạnh cụ thể đã được minh họa và được mô tả ở đây, sẽ được đánh giá cao bởi những người có trình độ trung bình trung bình trong lĩnh vực kỹ thuật tương ứng mà hàng loạt các cách thực hiện thay thế và/hoặc tương đương có thể được thay thế cho các khía cạnh cụ thể được thể hiện và được mô tả mà không trệt khỏi phạm vi của sáng chế. Sáng chế nhằm bao gồm các sự thích ứng hoặc các sự thay đổi của các khía cạnh cụ thể được thảo luận ở đây.

Mặc dù các thành phần trong các điểm yêu cầu bảo hộ sau đây được đọc theo trình tự cụ thể với nhãn tương ứng, trừ khi các thảo luận yêu cầu bảo hộ khác hàm ý trình tự cụ thể để thực hiện một số hoặc tất cả các thành phần này, các thành phần đó không cần được dự định để được giới hạn ở việc được thực hiện theo trình tự cụ thể đó.

Nhiều sự thay thế, sự cải biến, và sự thay đổi sẽ rõ ràng với những người có trình độ trung bình trong lĩnh vực kỹ thuật tương ứng xét đến các phần nêu trên. Nhất định, những người có trình độ trung bình trong lĩnh vực kỹ thuật tương ứng dễ dàng nhận ra rằng có nhiều ứng dụng của sáng chế ngoài các phần được mô tả ở đây. Trong khi sáng chế đã được mô tả dựa vào một hoặc nhiều phương án cụ thể, những người có trình độ trung bình trong lĩnh vực kỹ thuật tương ứng nhận ra rằng nhiều sự thay đổi có thể được thực hiện ở đó mà không trệch khỏi phạm vi của sáng chế. Do đó được hiểu rằng nằm trong phạm vi của các điểm yêu cầu bảo hộ kèm theo và các sự tương đương của chúng, sáng chế có thể được thực hiện khác so với được mô tả cụ thể ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý khối, trong đó khối này bao gồm các điểm ảnh, trong đó phương pháp này bao gồm các bước:

thu nhận tham số lượng tử hóa (quantization parameter, QP) của khối; trong đó khối này là khối được cấu trúc lại hoặc khối được dự báo;

thu nhận ngưỡng (threshold, THR) dựa vào tham số lọc, trong đó tham số lọc này được dẫn ra dựa vào QP; và

thu nhận bảng tra cứu dựa vào QP, để tạo ra khối được lọc dựa vào ngưỡng và bảng tra cứu,

trong đó tạo ra khối được lọc dựa vào ngưỡng và bảng tra cứu bao gồm:

tạo ra thành phần phổ được lọc của khối được lọc dựa vào bảng tra cứu khi giá trị tuyệt đối của thành phần phổ tương ứng nhỏ hơn ngưỡng.

2. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm các bước:

quét điểm ảnh hiện tại của khối và các điểm ảnh lân cận của nó của điểm ảnh hiện tại theo mẫu quét được xác định trước;

thu nhận các thành phần phổ bằng cách thực hiện biến đổi đối với điểm ảnh hiện tại và các điểm ảnh lân cận của nó;

thu nhận các thành phần phổ được lọc bằng cách sử dụng bảng tra cứu và các thành phần phổ.

3. Phương pháp theo điểm 2, trong đó thành phần phổ được lọc $F(i, \sigma)$ được dẫn ra bởi:

$$F(i, \sigma) = \begin{cases} R(i), \text{Abs}(R(i)) \geq THR \\ LUT(R(i), \sigma), R(i) > 0 \\ -LUT(-R(i), \sigma), R(i) \leq 0 \end{cases}$$

trong đó (i) là chỉ số của bộ phận phổ, $R(i)$ là thành phần phổ tương ứng với chỉ số (i), σ là tham số lọc, $LUT(R(i), \sigma)$ là mảng trong bảng tra cứu, và THR là ngưỡng được dẫn ra từ QP.

4. Phương pháp theo điểm 3, trong đó $LUT(R(i), \sigma) = \frac{R(i)^3}{R(i)^2 + m \cdot \sigma^2}$, m là hằng số chuẩn hóa bằng với số lượng các thành phần phổ.

5. Phương pháp theo điểm 4, trong đó tham số lọc σ được dẫn ra bởi

$$\sigma = k * 2^{(n * (QP - s))}$$

, trong đó QP là tham số lượng tử hóa bộ mã hóa-giải mã, k, n và s là các hằng số.

6. Phương pháp theo điểm 5, trong đó $k = 2,64, n = 0,1296$ và $s = 11$.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 6, trong đó ngưỡng được dẫn ra bởi:

$$\frac{THR^2}{THR^2 + m \cdot \sigma^2} = C$$

trong đó C là giá trị gần bằng 1, σ là tham số lọc, và m là hằng số chuẩn hóa bằng với số lượng các thành phần phổ.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 7, trong đó phương pháp này còn bao gồm bước:

chọn tập hợp của các QP để thu nhận bảng tra cứu (LUT), trong đó tập hợp của các QP bao gồm QP thứ nhất tương ứng với chỉ số (i) và QP thứ hai tương ứng với chỉ mục (i+1), và trong đó QP thứ nhất và QP thứ hai có khoảng cách lớn hơn 1.

9. Phương pháp theo điểm 8, trong đó khoảng cách là không đổi và bằng 8, 10 hoặc 16.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 9, trong đó phương pháp này còn bao gồm bước:

giảm N các bit từ các giá trị bảng của LUT, N là số nguyên.

11. Phương pháp theo điểm 10, trong đó việc truy cập vào phần tử của LUT là như sau

$tbl[(z + HTDF_TBL_RND) \gg HTDF_TBL_SH],$

trong đó HTDF_TBL_SH là bí danh của N, và $HTDF_TBL_RND = 1 \ll (HTDF_TBL_SH - 1)$

12. Phương pháp theo điểm 10, trong đó việc truy cập vào phần tử của LUT là như sau

$$LUT[(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift]$$

đối với các giá trị dương của fHad[i], và

$$-LUT[(-(fHad[i] + (1 \ll (tblShift - 1))) \gg tblShift)]$$

đối với các giá trị âm của fHad[i],

trong đó tblShift là bí danh của N, và fHad[i] là các thành phần phổ sẽ được lọc.

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 10 đến 12, trong đó N là 3.

14. Phương pháp theo điểm bất kỳ trong số các điểm 10 đến 12, trong đó đối với QP thứ nhất trong tập hợp của các QP, N bằng 2, và trong đó tập hợp của các QP được sử dụng để thu nhận bảng tra cứu (LUT);

trong đó đối với QP cuối cùng hoặc đối với hai QP cuối cùng trong tập hợp của các QP, N bằng 4; và

trong đó đối với các QP còn lại từ tập hợp của các QP, N bằng 3.

15. Phương pháp theo điểm bất kỳ trong số các điểm từ 10 đến 12, trong đó N được xác định như sau;

$$tblShift = tblThrLog2[qpIdx] - 4$$

$$tblThrLog2[5] = \{6, 7, 7, 8, 8\}$$

trong đó tblShift là bí danh của N, và qpIdx được dẫn ra dựa vào QP.

16. Phương pháp theo điểm 15, trong đó qpIdx được dẫn ra dựa vào QP là như sau;

```

if(pred_mode_flag[ xCb ][ yCb ] == 0 && nCbW == nCbH && min(nCbW,
nCbH
)
>=
32)
qpIdx = Clip3(0, 4, (QpY - 28 + (1 << 2 ) ) >> 3 )
else
qpIdx = Clip3(0, 4, (QpY - 20 + (1 << 2 ) ) >> 3 )

```

trong đó QpY là QP khôi hiện tại, xCb, yCb định rõ vị trí của khối hiện tại trên hình ảnh (x, y), pred_mode_flag[xCb][yCb] định rõ chế độ dự báo của khối hiện tại, nếu bằng 0 việc dự báo là liên dự báo, nếu không sẽ là dự báo nội khung, nCbW, nCbH tương ứng là độ rộng và độ cao của khối hiện tại.

17. Phương pháp theo điểm bất kỳ trong số các điểm từ 10 đến 12, trong đó N tùy thuộc vào giá trị QP.

18. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 17, trong đó việc tạo ra LUT dựa vào chức năng phụ cho ít nhất một vài tham số lượng tử hóa (QP).

19. Phương pháp theo điểm 18, trong đó việc tạo ra LUT $LUT(R_i, \sigma)$ được dẫn ra bởi:

$$LUT(R_i, \sigma) = \max \left(\frac{R_i^3}{R_i^2 + m * \sigma^2}, AuxiliaryFunc_{\sigma}(i) \right),$$

trong đó $AuxiliaryFunc_{\sigma}(i)$ biểu diễn chức năng phụ, và chức năng phụ này có giá trị bằng THR ở i tương ứng với đối số của LUT + 1 cuối cùng.

20. Phương pháp theo điểm 18 hoặc 19, trong đó chức năng phụ là phương trình đường thẳng đi qua các điểm (i, THR) và (a, 0), trong đó $a > 0$ và a tùy thuộc vào tham số lọc σ hoặc giá trị QP.

21. Phương pháp theo điểm 20, trong đó LUT được xác định như sau

```

setOfLUT[ 5 ][ 16 ] =
{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, },
{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, },

```

$\{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$
 $\{ 0, 0, 3, 9, 19, 32, 47, 64, 81, 99, 117, 135, 154, 179, 205, 230, \},$
 $\{ 0, 0, 0, 2, 6, 11, 18, 27, 38, 51, 64, 96, 128, 160, 192, 224, \}.$

22. Phương pháp theo điểm 20, trong đó LUT được xác định như sau

$\text{setOfLUT}[5][16] =$
 $\{ 0, 0, 2, 6, 10, 14, 19, 23, 28, 32, 36, 41, 45, 49, 53, 57, \},$
 $\{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, \},$
 $\{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$
 $\{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, \},$
 $\{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, \}.$

23. Phương pháp theo điểm 20, trong đó LUT được xác định như sau

$\text{setOfLUT}[5][16] =$
 $\{ 0, 2, 10, 19, 28, 36, 45, 53, 61, 70, 78, 86, 94, 102, 110, 118, \},$
 $\{ 0, 0, 5, 12, 20, 29, 38, 47, 56, 65, 73, 82, 90, 98, 107, 115, \},$
 $\{ 0, 0, 1, 4, 9, 16, 24, 32, 41, 50, 59, 68, 77, 86, 94, 103, \},$
 $\{ 0, 0, 0, 1, 3, 5, 9, 14, 19, 25, 32, 40, 55, 73, 91, 110, \},$
 $\{ 0, 0, 0, 0, 0, 1, 2, 4, 6, 8, 11, 14, 26, 51, 77, 102, \}.$

24. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 23, trong đó biến đổi là biến đổi Hadamard.

25. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 24, trong đó biến đổi là biến đổi 1D.

26. Bộ giải mã bao gồm mạch xử lý để tiến hành phương pháp theo điểm bất kỳ trong số các điểm 1-25. 2

27. Bộ mã hóa bao gồm mạch xử lý để tiến hành phương pháp theo điểm bất kỳ trong số các điểm 1-25. 3

28. Bộ giải mã, bao gồm: 4

bộ phận lưu trữ bộ nhớ bao gồm các lệnh; và

một hoặc nhiều bộ xử lý truyền thông với bộ nhớ, trong đó một hoặc nhiều bộ xử lý thực hiện các lệnh để tiến hành phương pháp theo điểm bất kỳ trong số các điểm 1-25.

29. Bộ mã hóa, bao gồm:

5

bộ phận lưu trữ bộ nhớ bao gồm các lệnh; và

một hoặc nhiều bộ xử lý truyền thông với bộ nhớ, trong đó một hoặc nhiều bộ xử lý này thực hiện các lệnh để tiến hành phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 25.

30. Bộ phận lưu trữ bộ nhớ bao gồm sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 25 khi được thực hiện trên máy tính hoặc bộ xử lý.

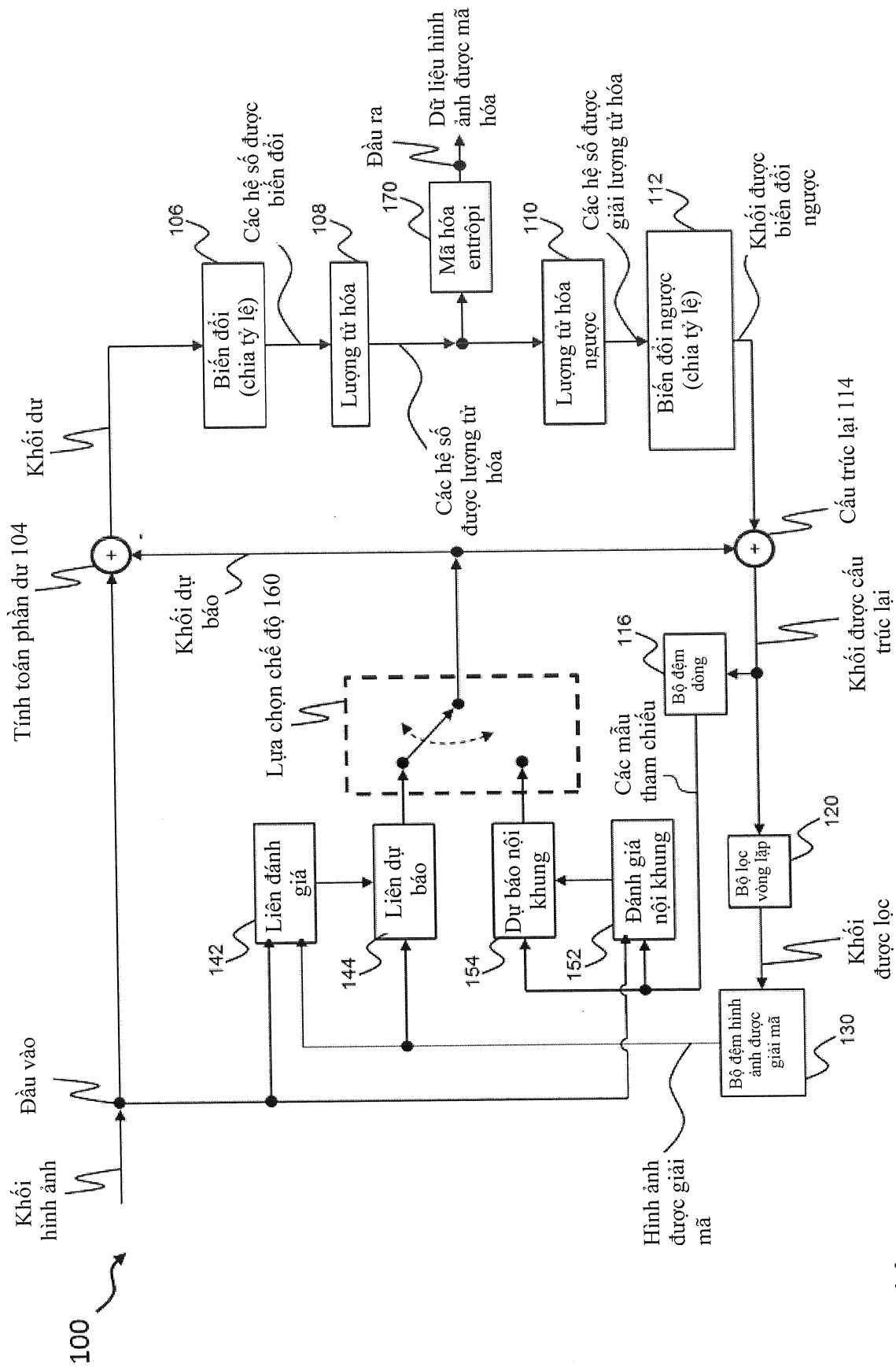


Fig. 1A

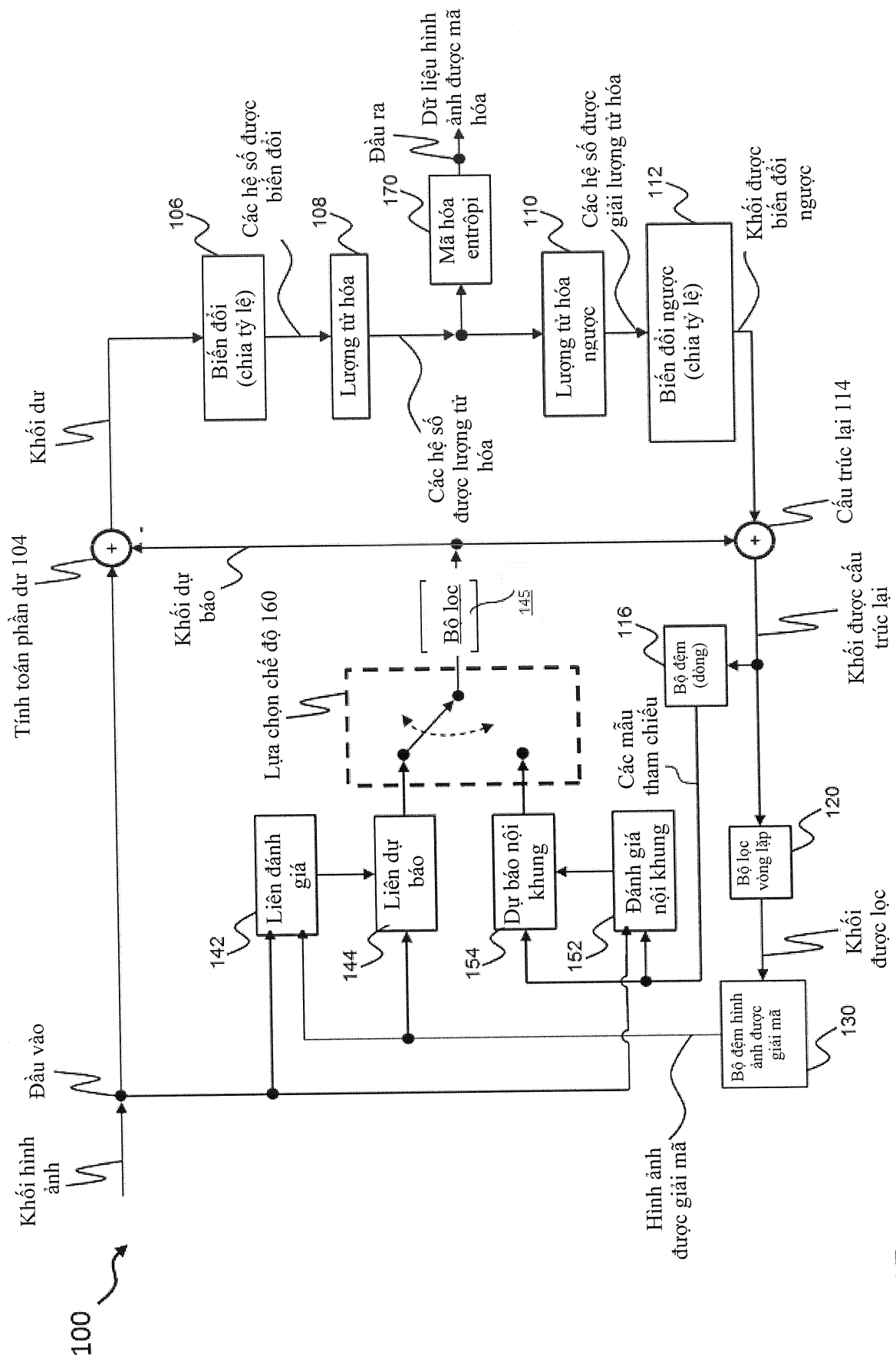
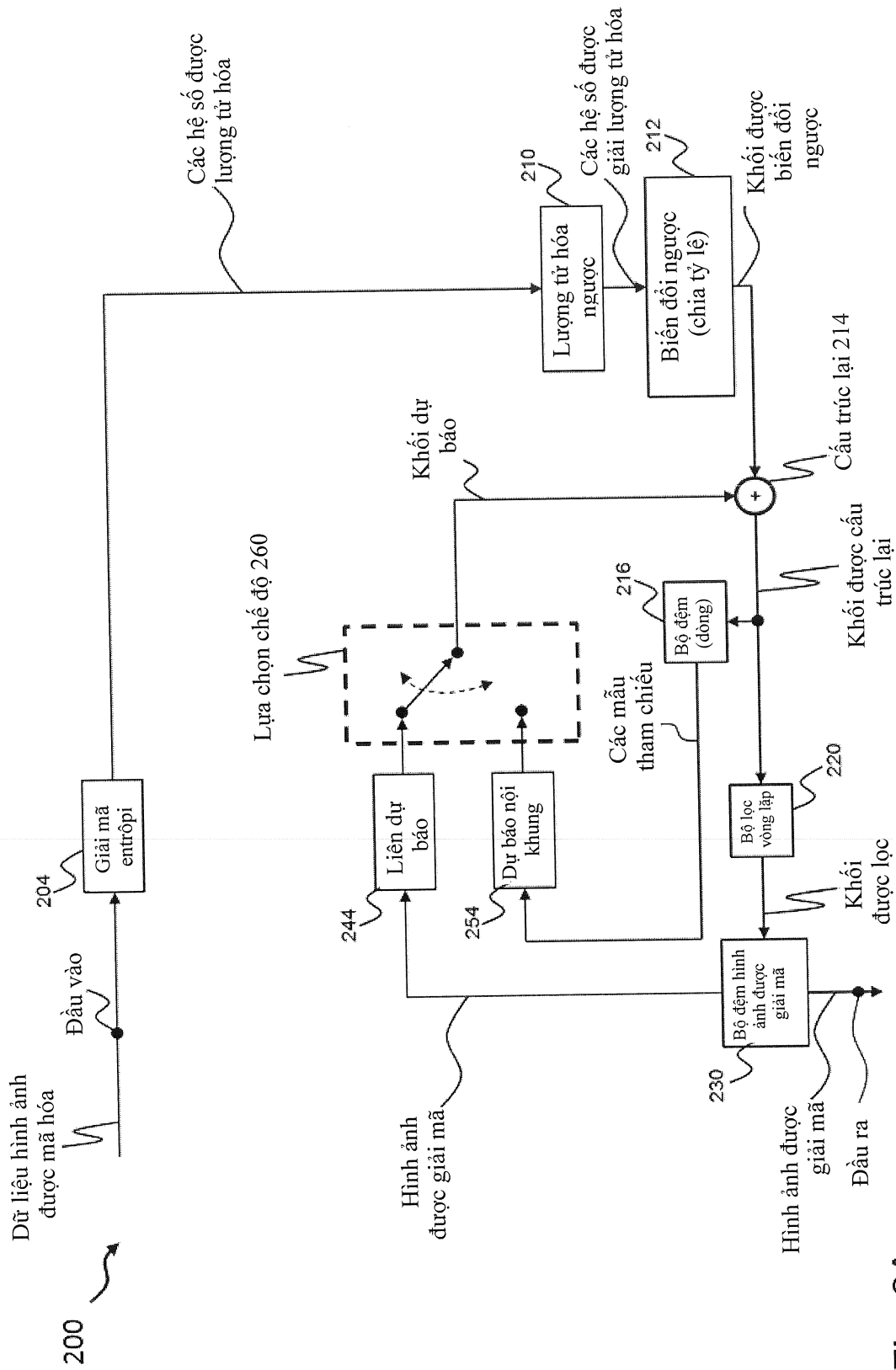


Fig. 1B



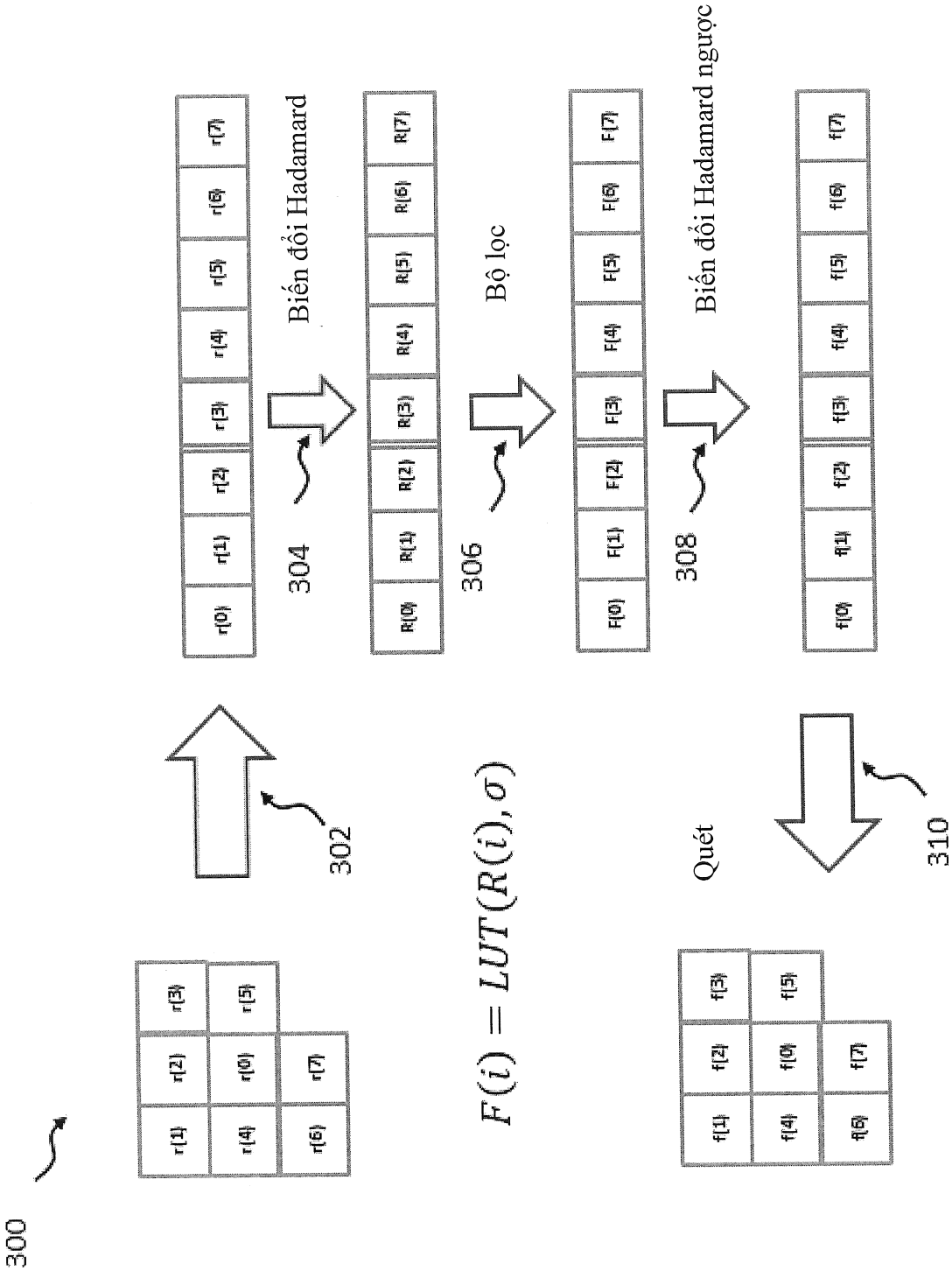


Fig. 3A

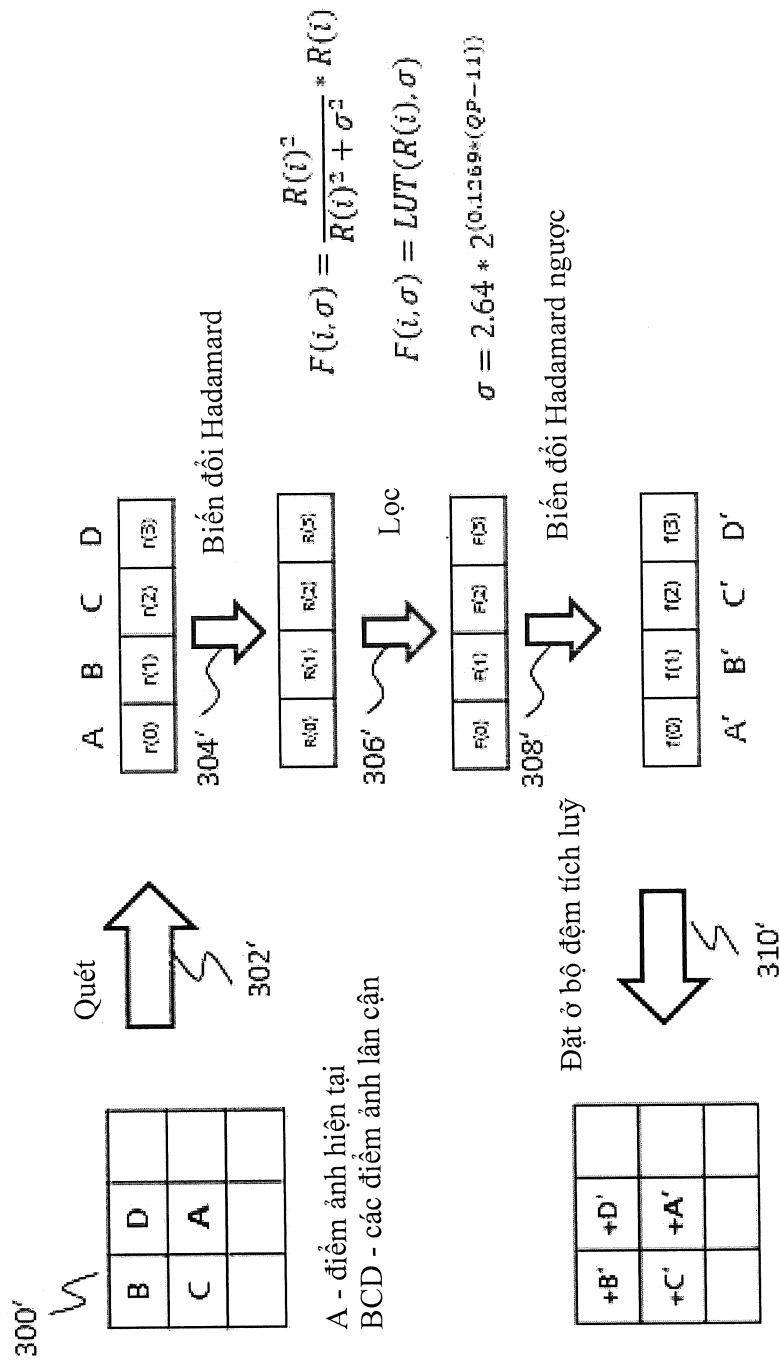


Fig. 3B

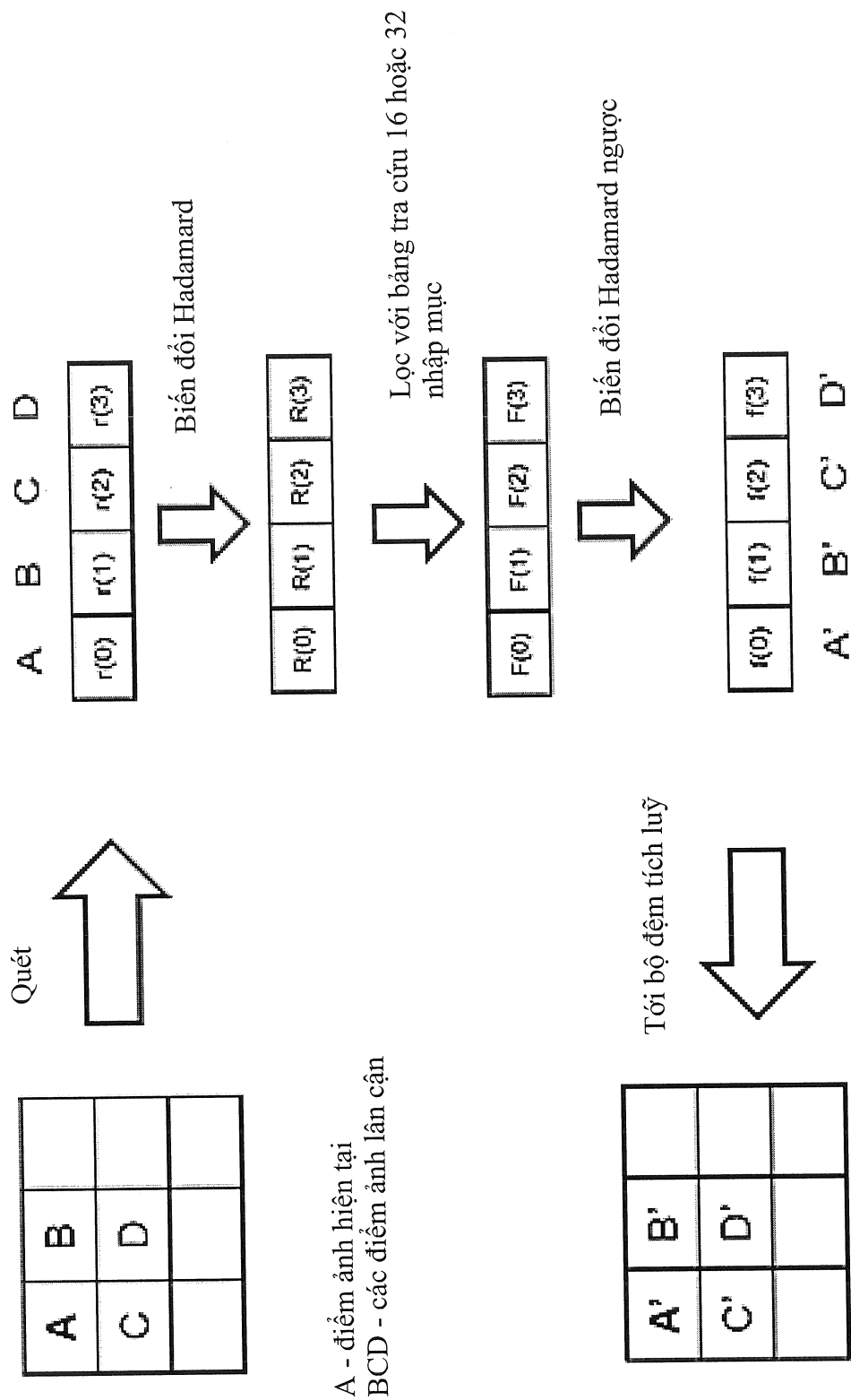


Fig. 3C

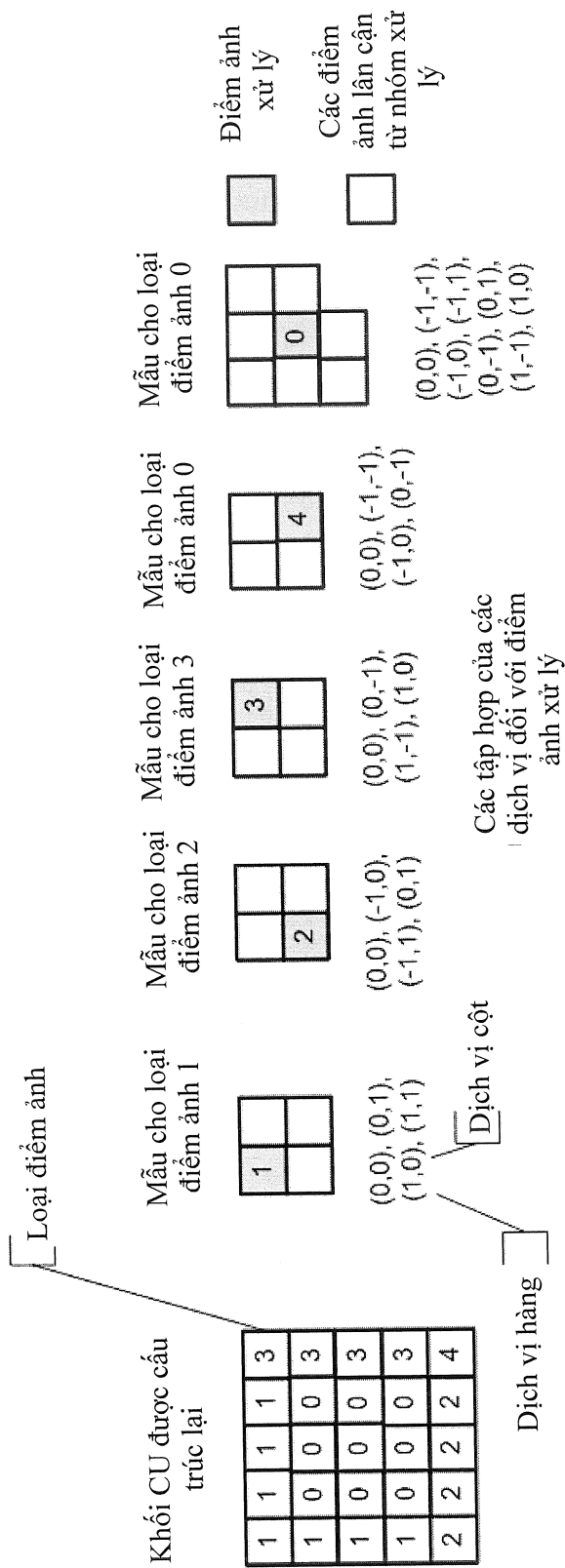


Fig. 4A

9/26

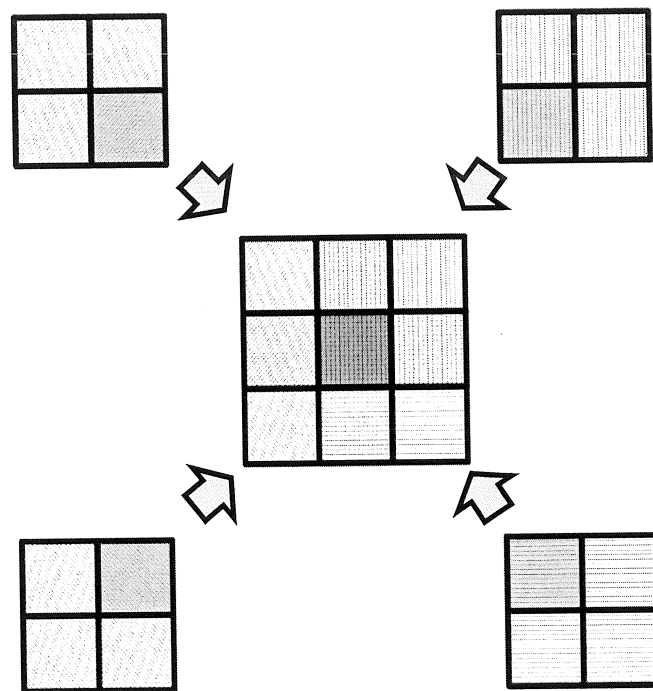


Fig. 4B

10/26

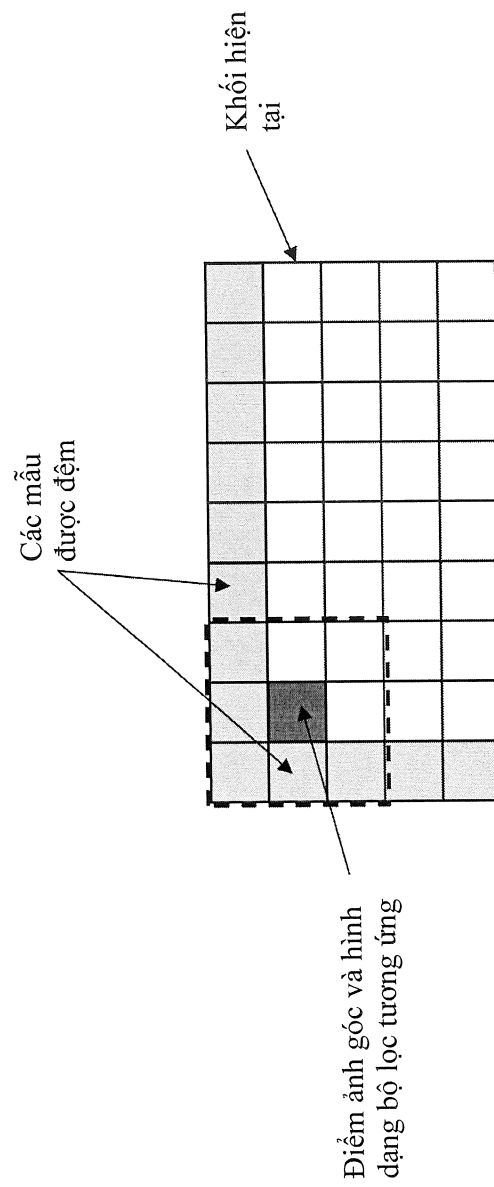


Fig. 4C

11/26

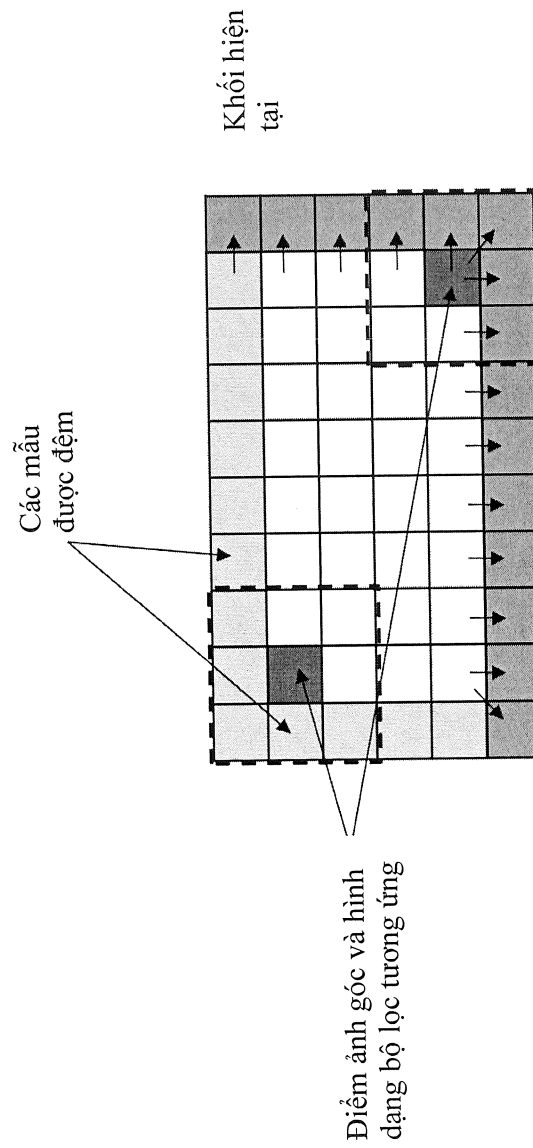


Fig. 4D

500

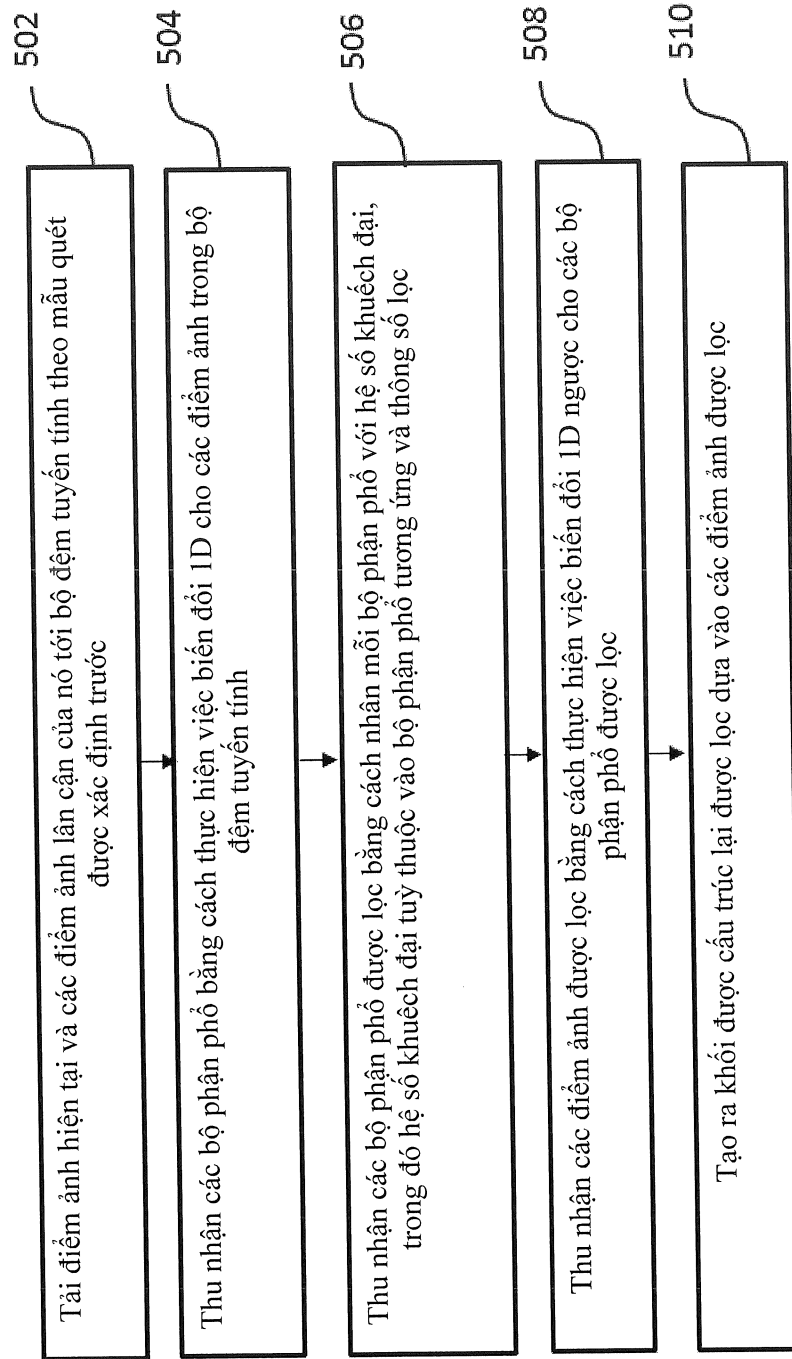


Fig. 5A

13/26

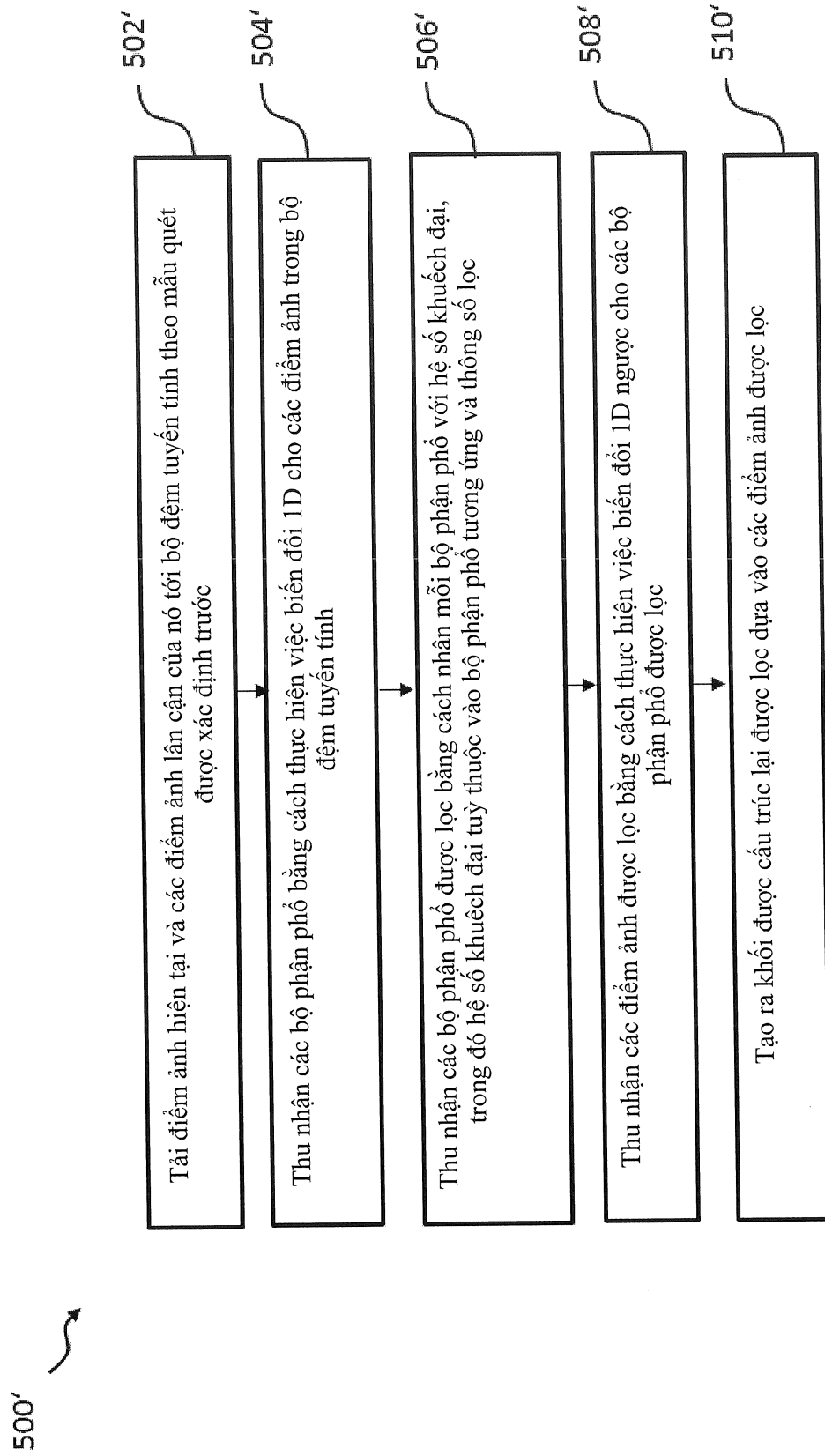


Fig. 5B

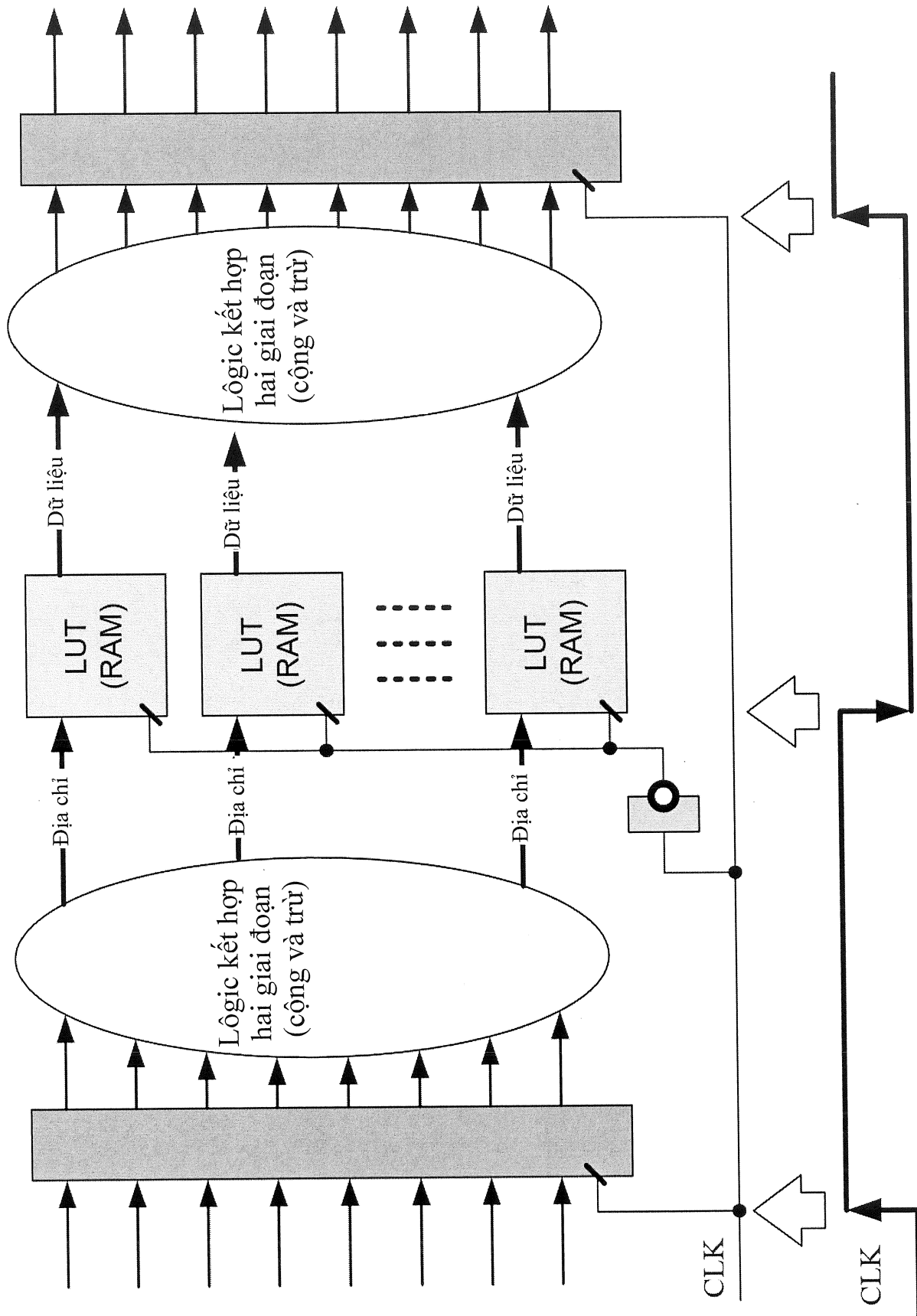


FIG. 6

15/26

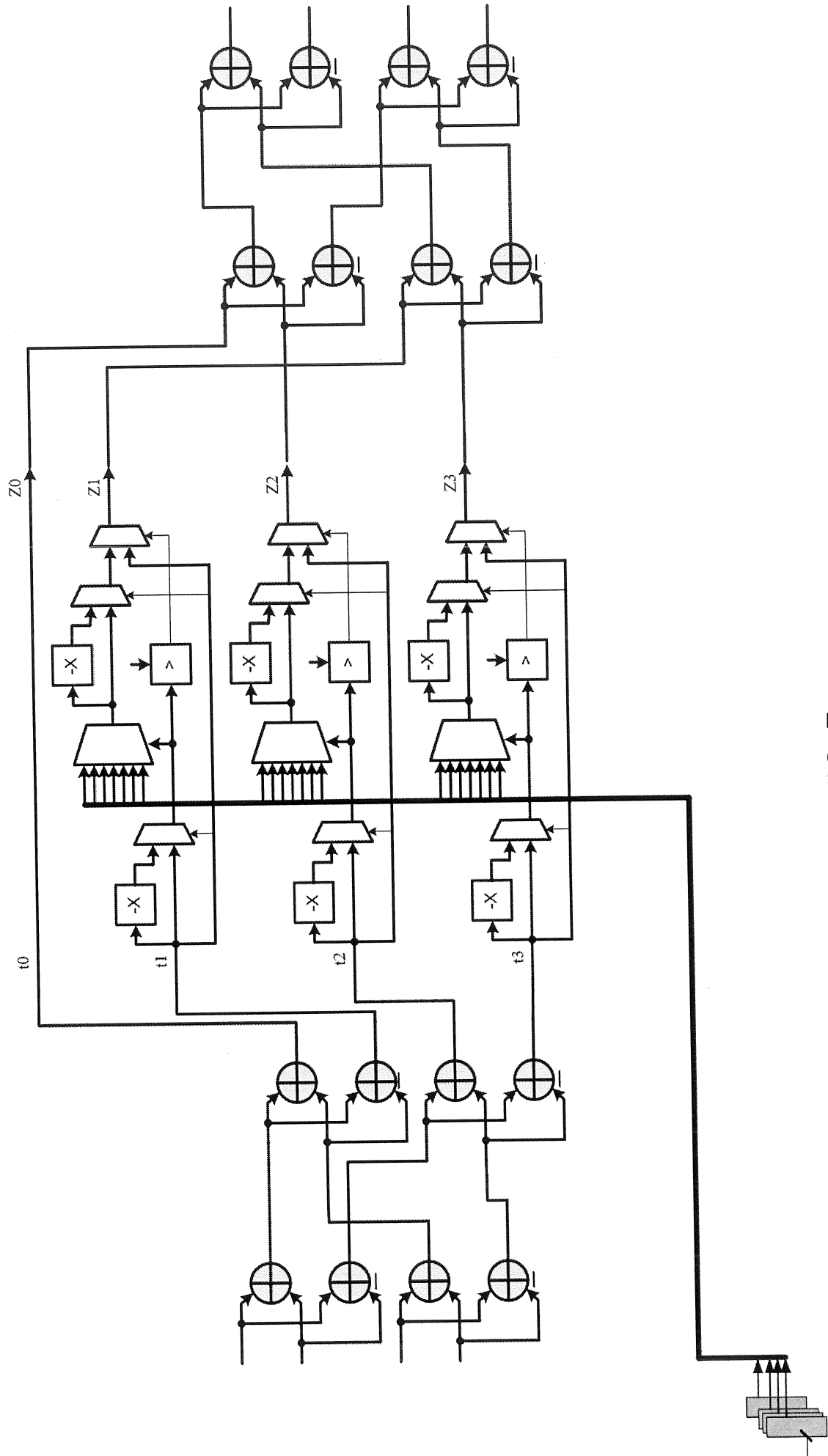


FIG. 7

16/26

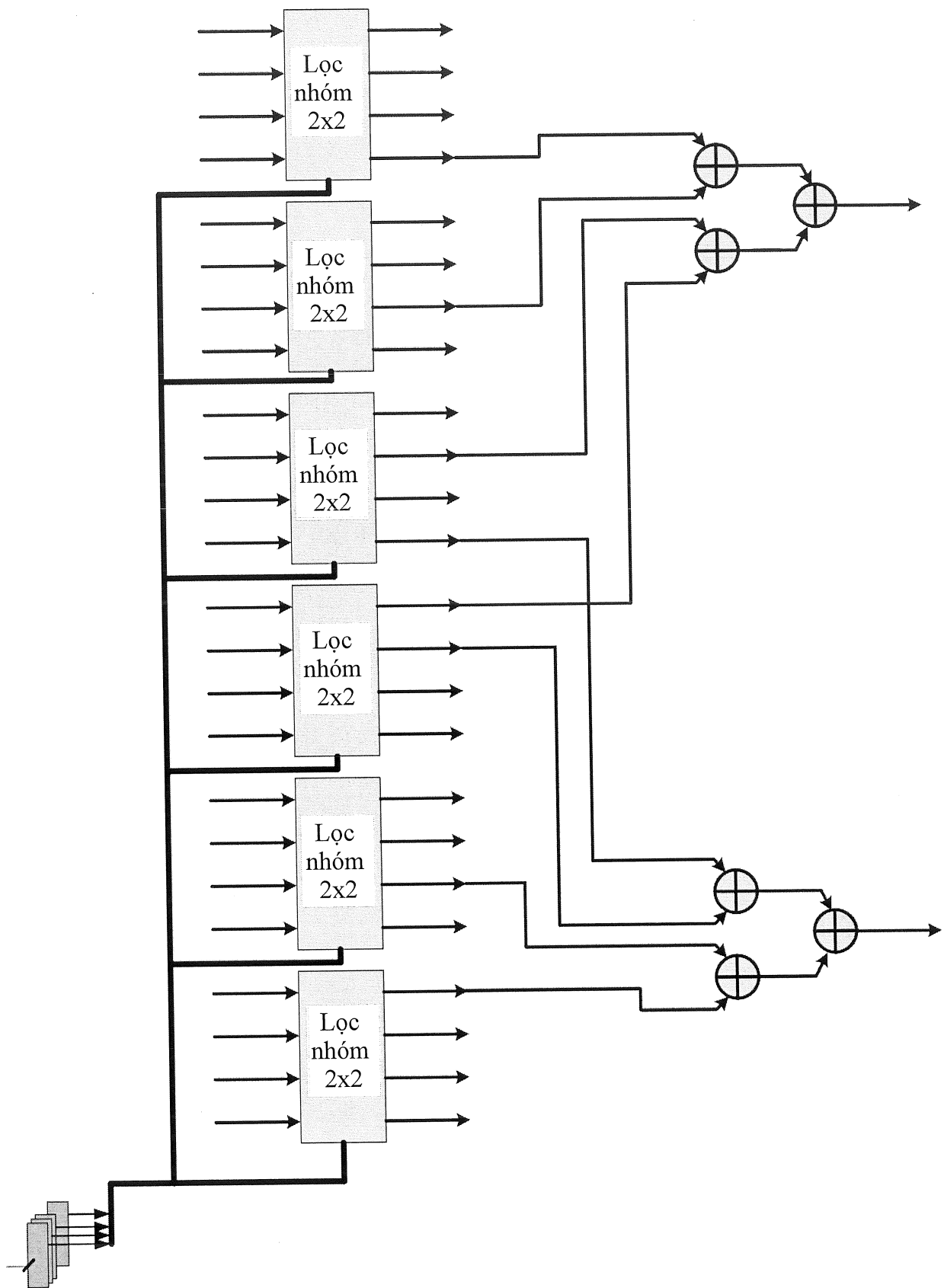


FIG. 8

17/26

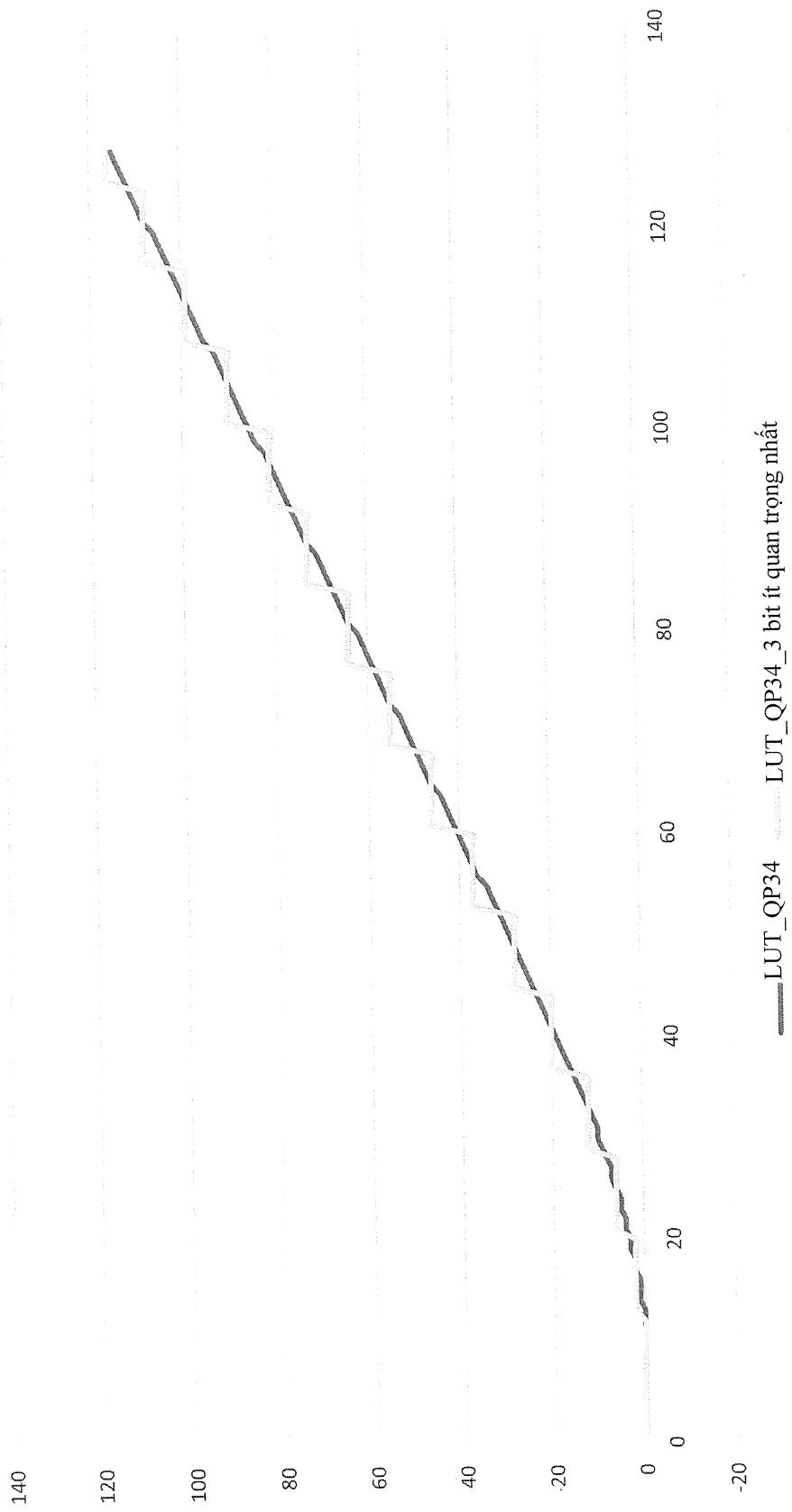


FIG. 9

18/26

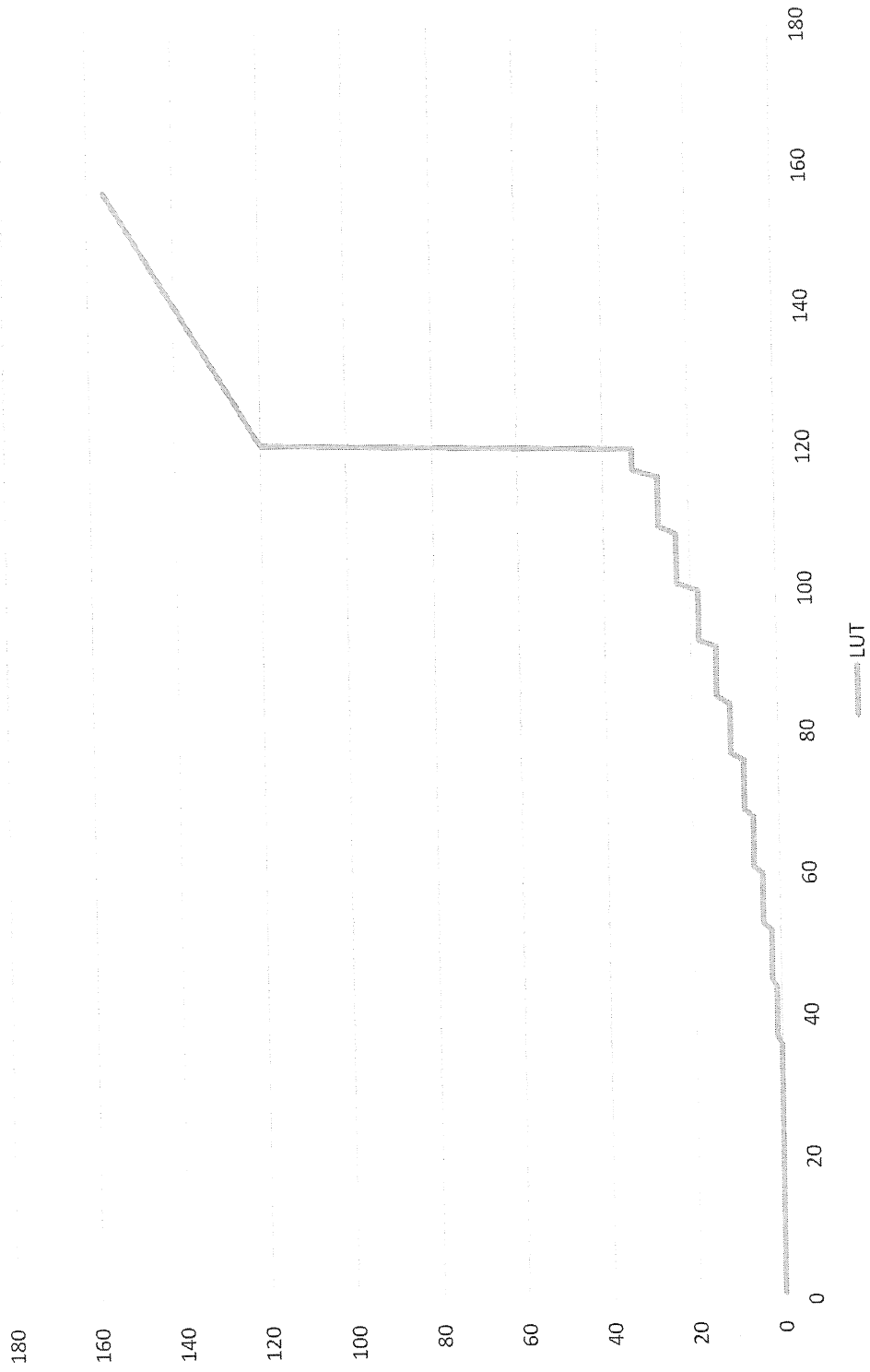


FIG. 10

19/26

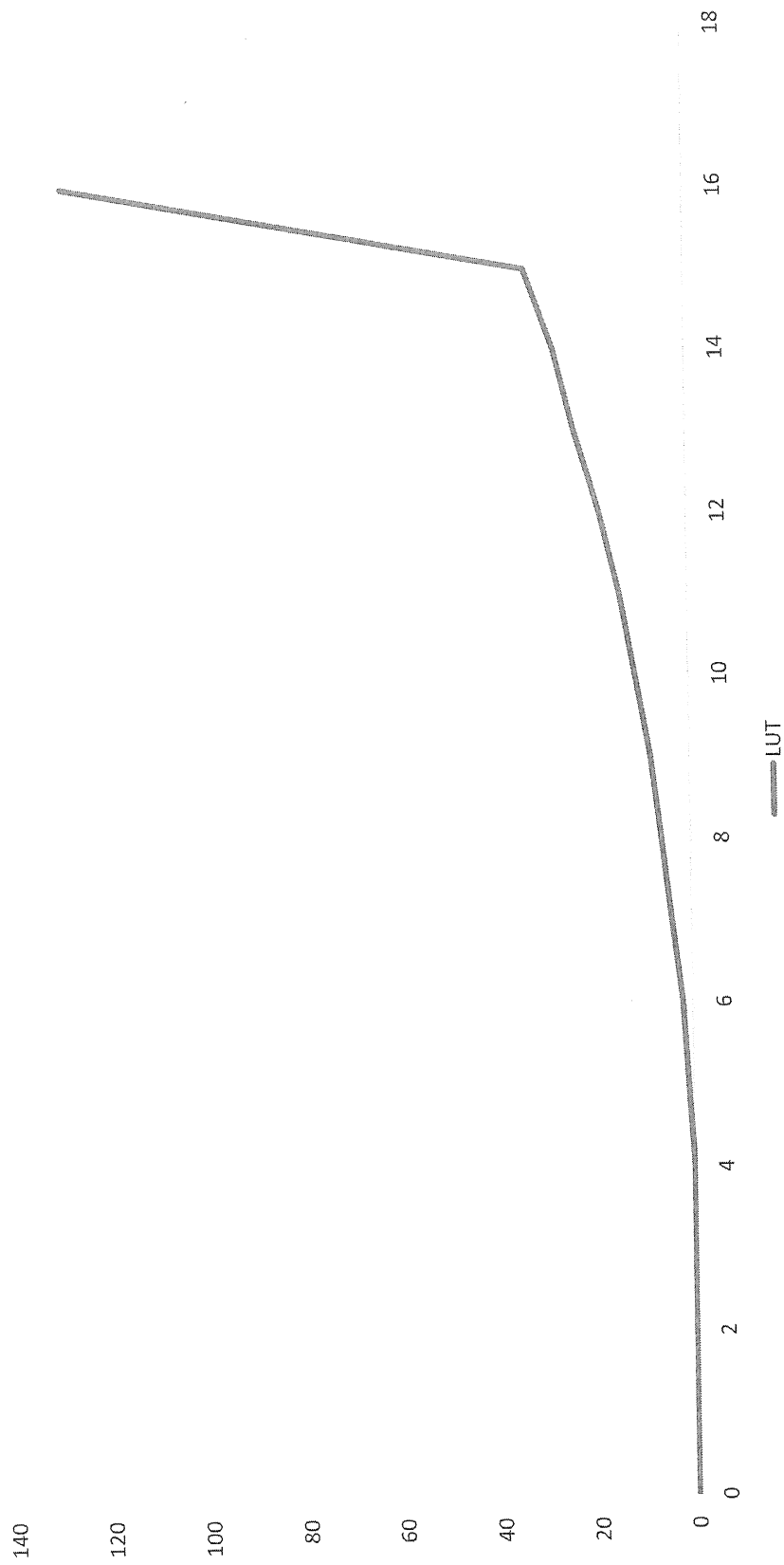


FIG. 11

20/26

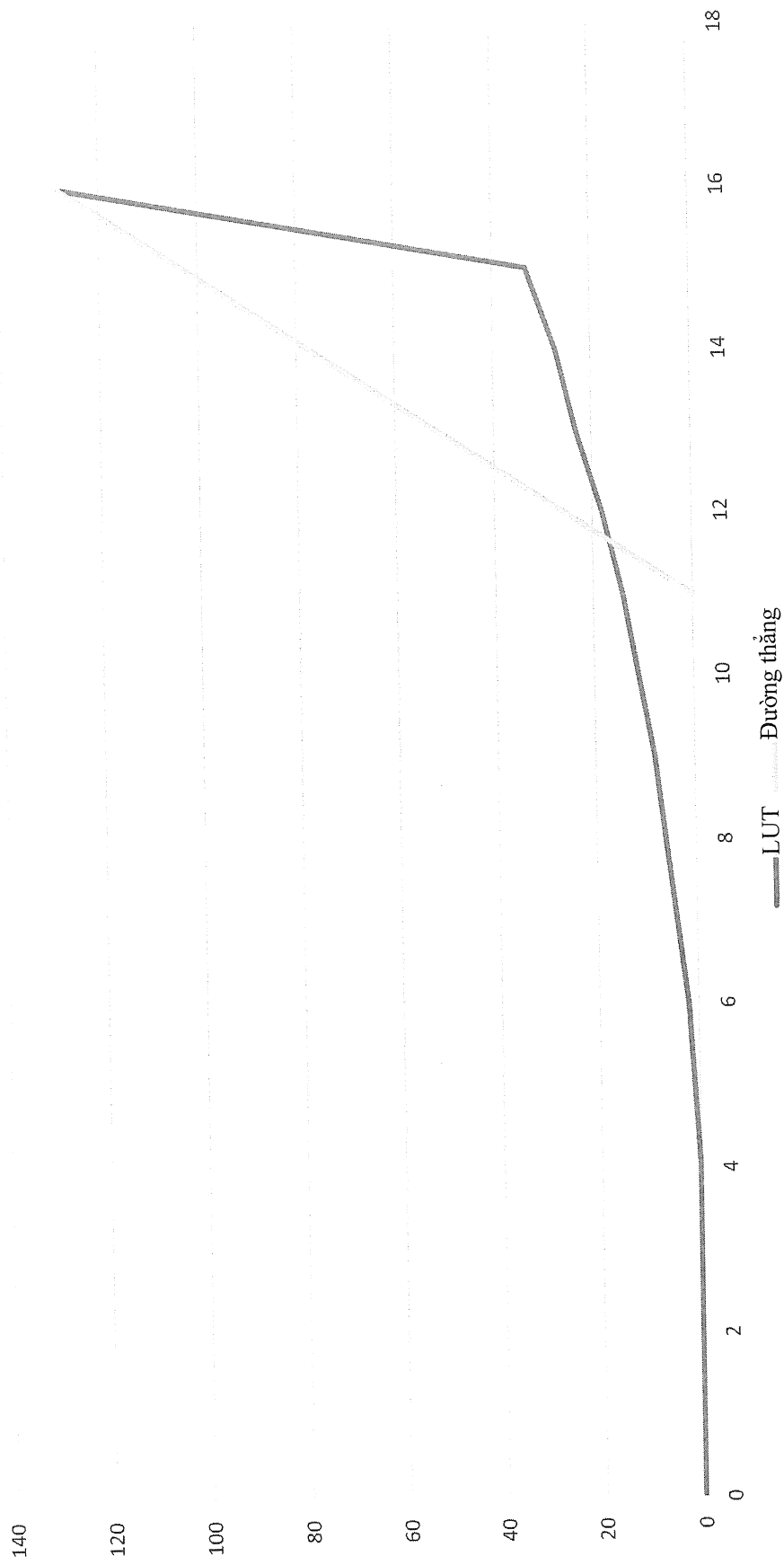


FIG. 12

21/26

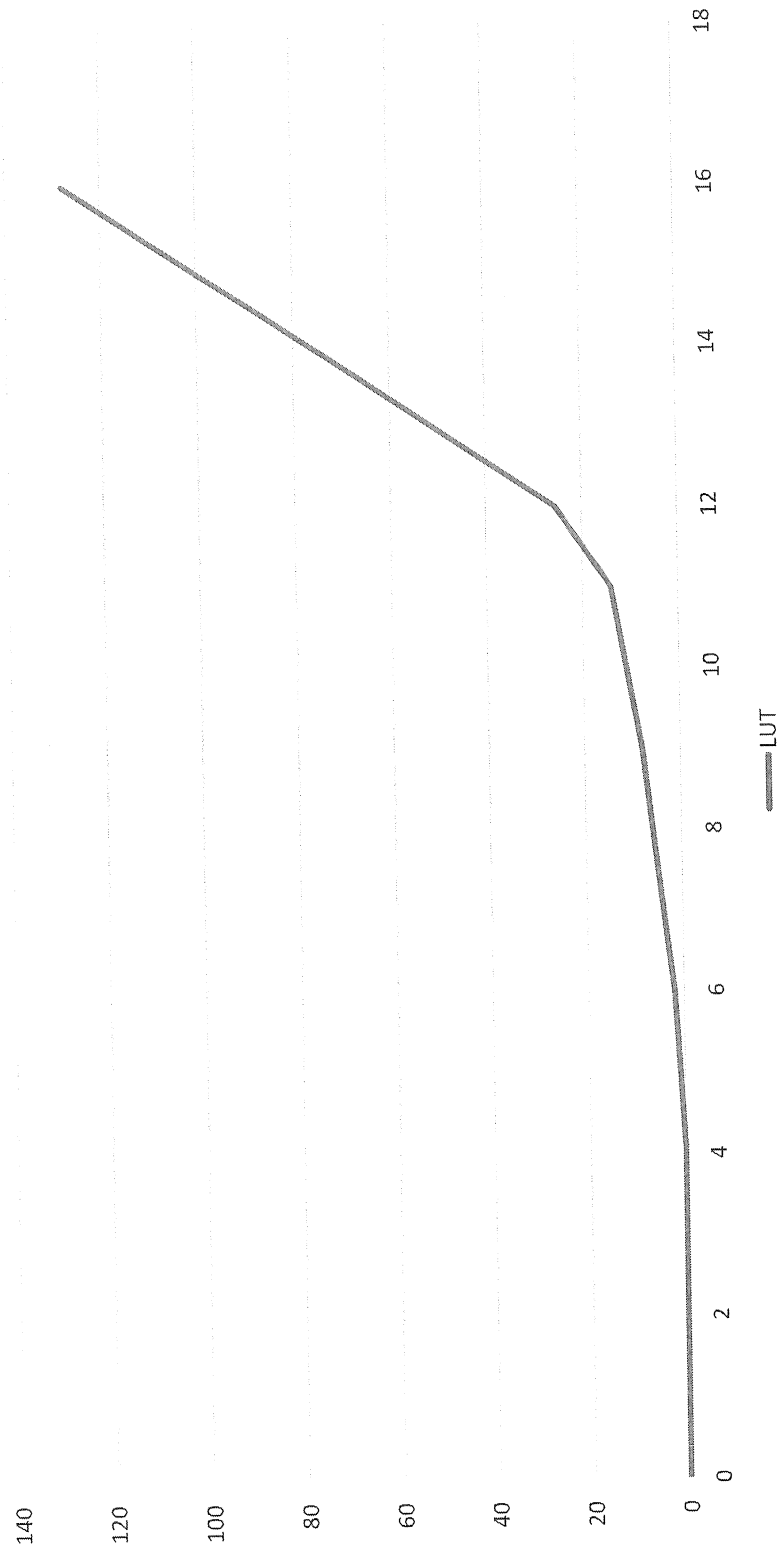


FIG. 13

22/26

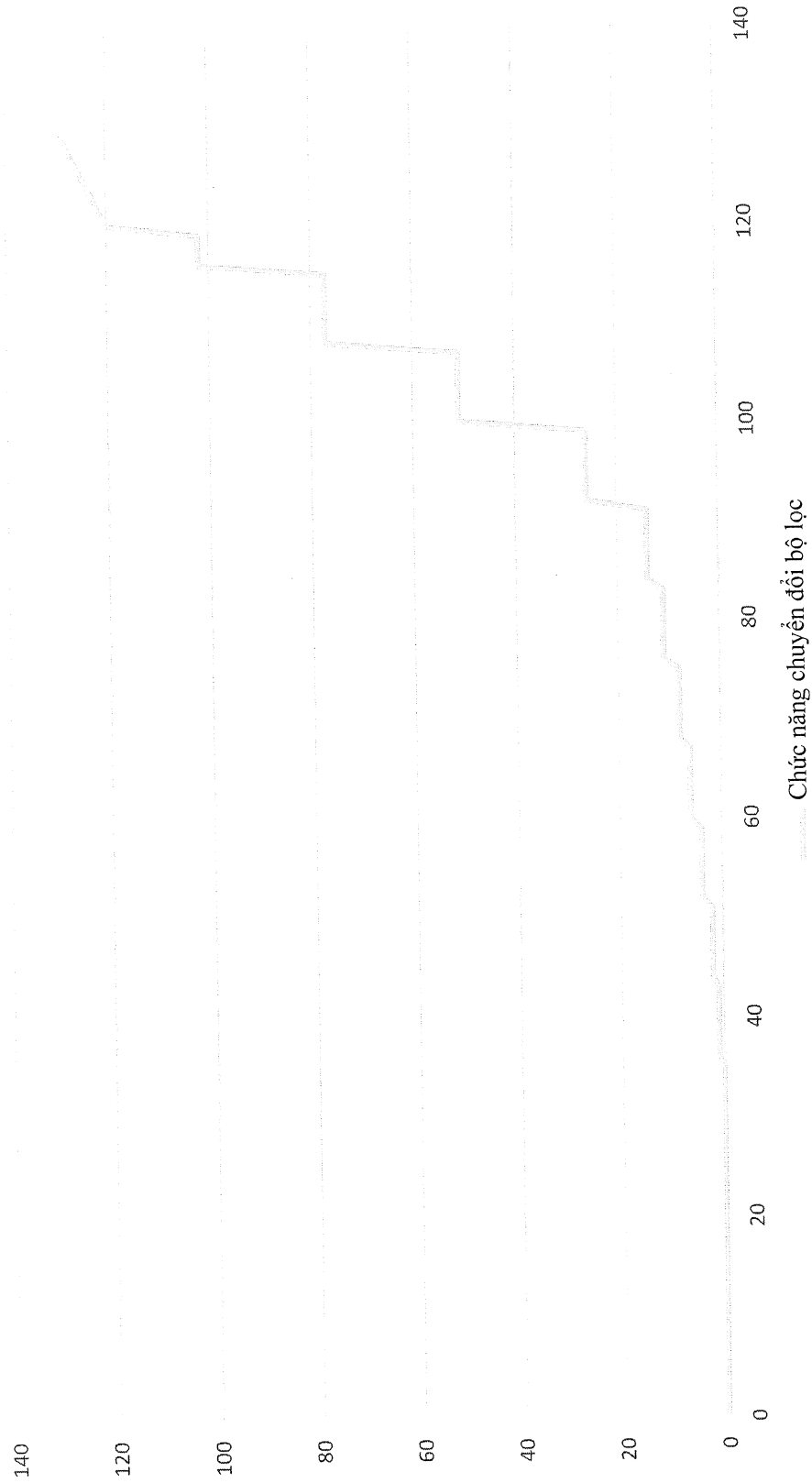


FIG. 14

23/26

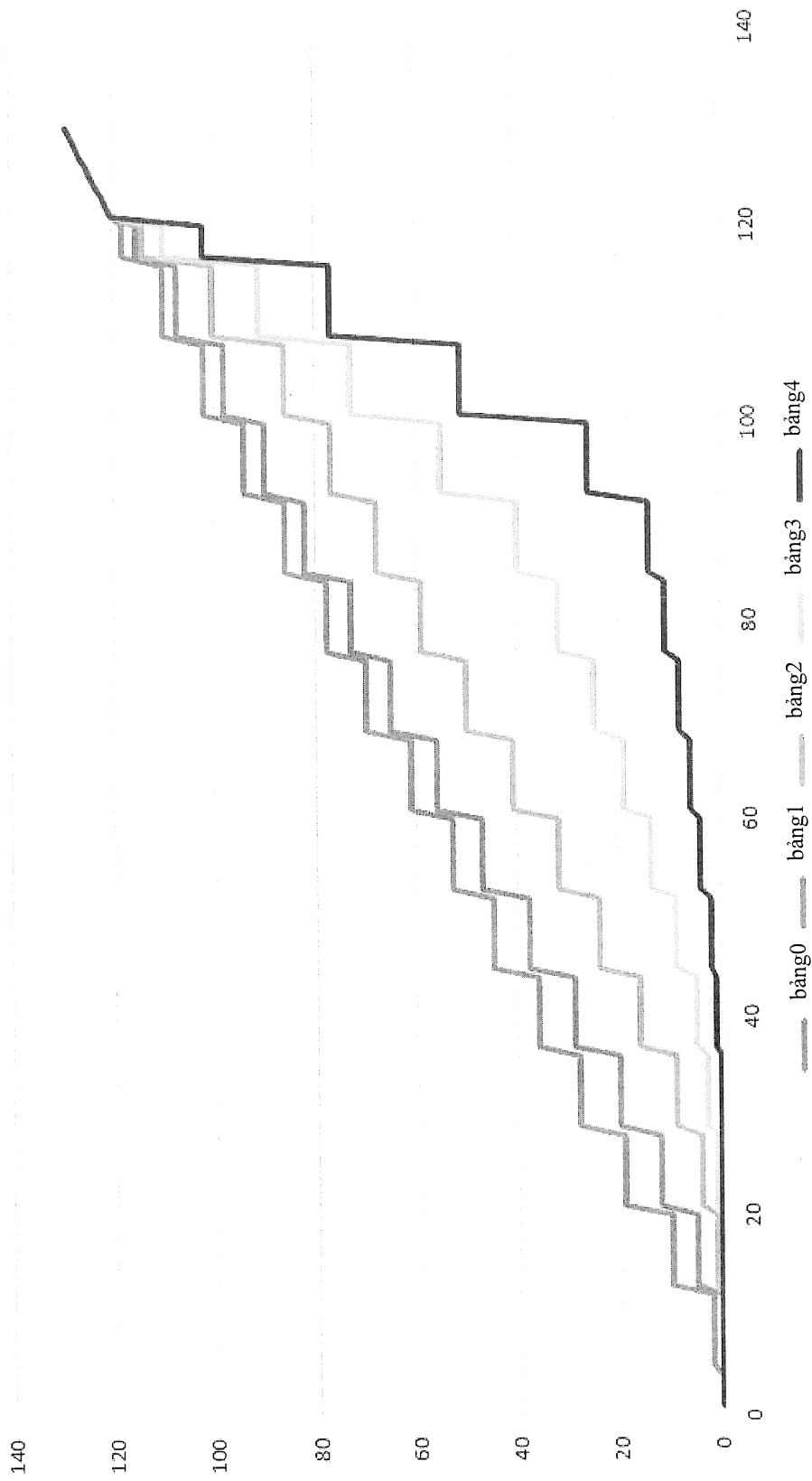


FIG. 15

24/26

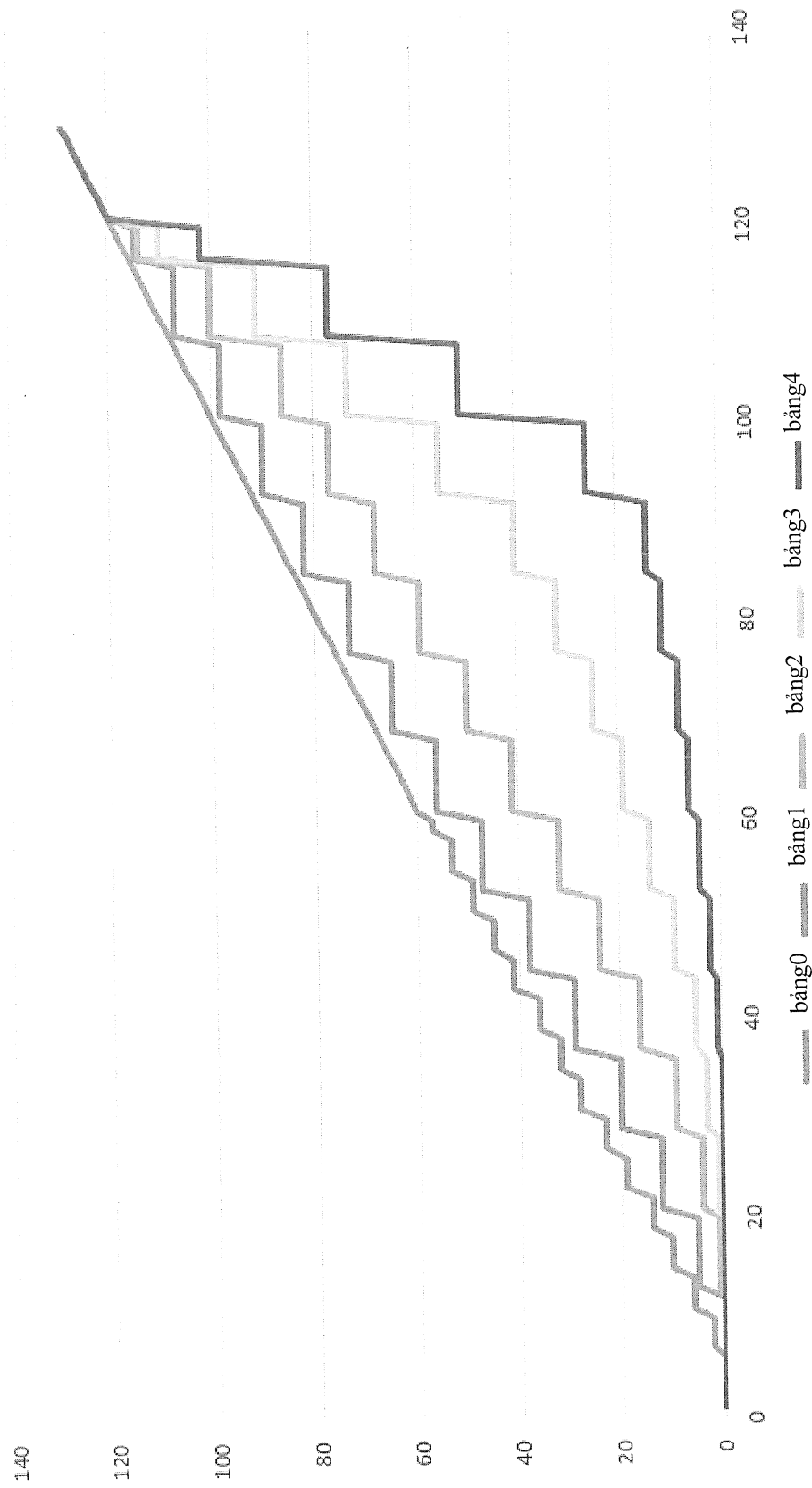


FIG. 16

25/26

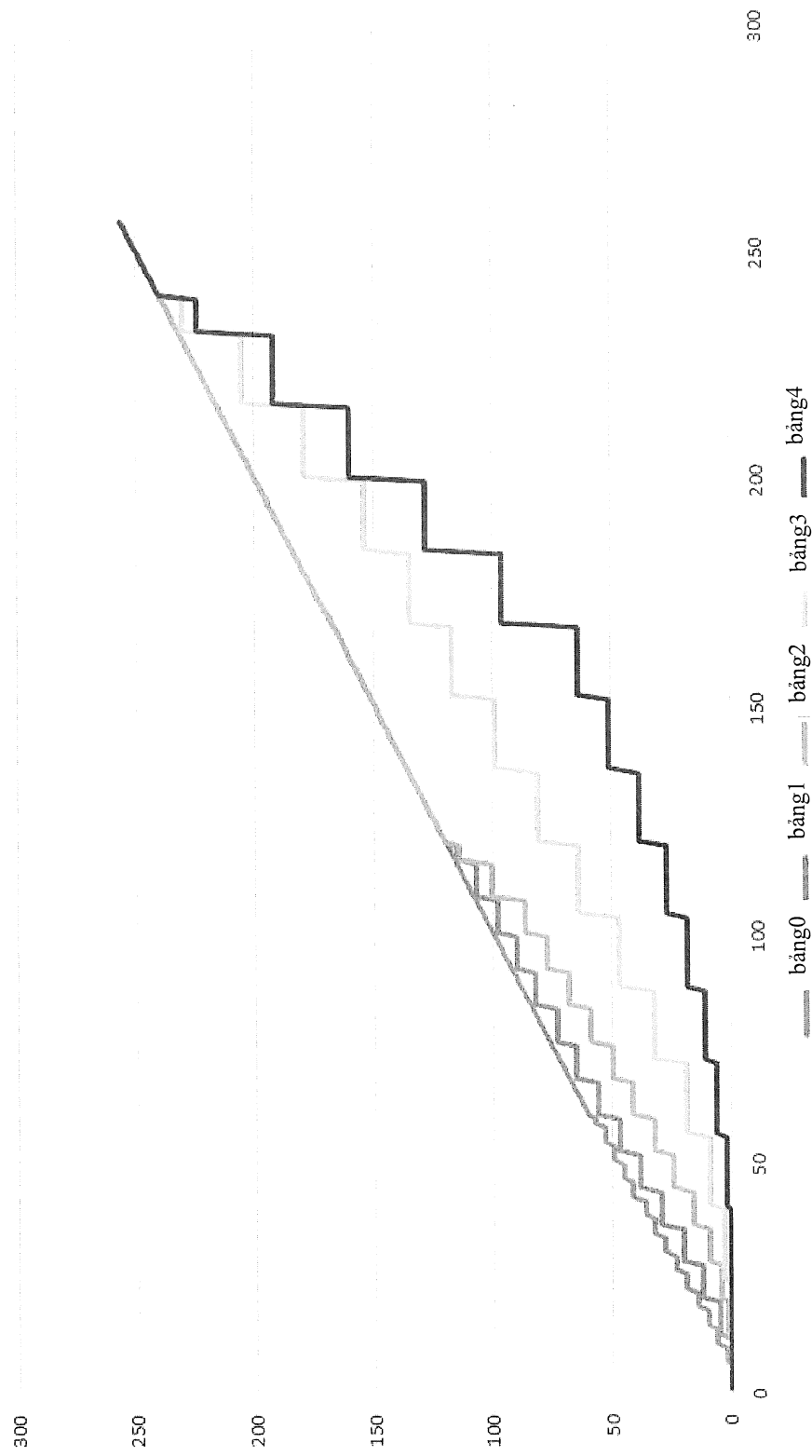


FIG. 17

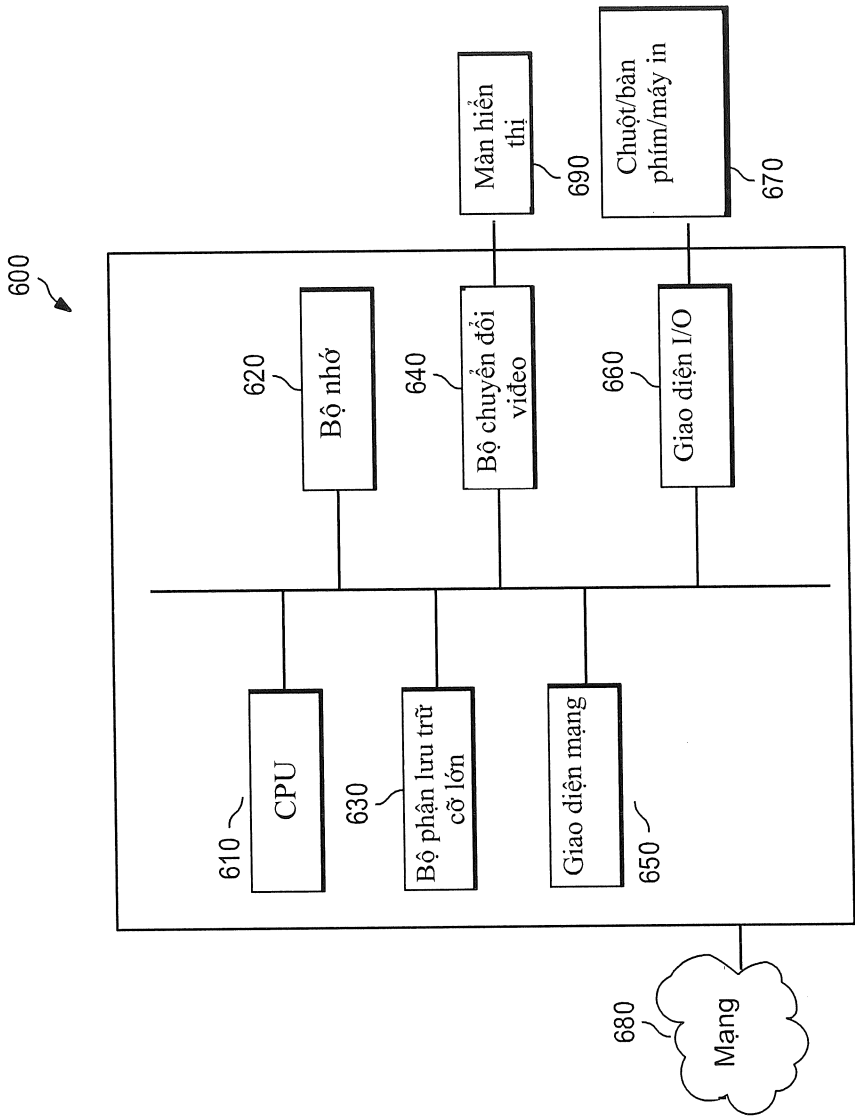


FIG. 18