



(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)



1-0044760

(51)^{2020.01} H04N 19/635

(13) B

(21) 1-2021-00514

(22) 02/07/2019

(86) PCT/RU2019/050100 02/07/2019

(87) WO2020/009617 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) 25/04/2025 445

(43) 26/04/2021 397A

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

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, China

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

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP ĐỂ XỬ LÝ KHÓI, BỘ MÃ HÓA, BỘ GIẢI MÃ VÀ PHƯƠNG
TIỆN LƯU TRỮ ĐƯỢC ĐỌC ĐƯỢC BẰNG MÁY TÍNH

(21) 1-2021-00514

(57) Sáng chế đề cập đến phương pháp để xử lý khối và thiết bị để xử lý khối. Cụ thể, sáng chế đề cập đến bộ lọc để tạo mã video, trong đó bộ lọc được tạo cấu hình để xử lý khối để tạo khối được lọc, và trong đó khối bao gồm nhiều điểm ảnh. Bộ lọc gồm một hoặc nhiều bộ xử lý được tạo cấu hình để: quét điểm ảnh hiện tại của khối và các điểm ảnh lân cận của nó của điểm ảnh hiện tại theo mẫu quét định trước; thu các thành phần phổ bằng cách thực hiện biến đổi cho điểm ảnh hiện tại và các điểm ảnh lân cận của nó; thu các thành phần phổ đã được lọc được dựa trên tham số lọc và các thành phần phổ; thu các điểm ảnh được lọc bằng cách thực hiện biến đổi ngược cho các thành phần phổ đã được lọc; và tạo khối được lọc dựa trên các điểm ảnh được lọc. Bộ lọc được cung cấp để cho phép cải thiện hiệu quả tạo mã video.

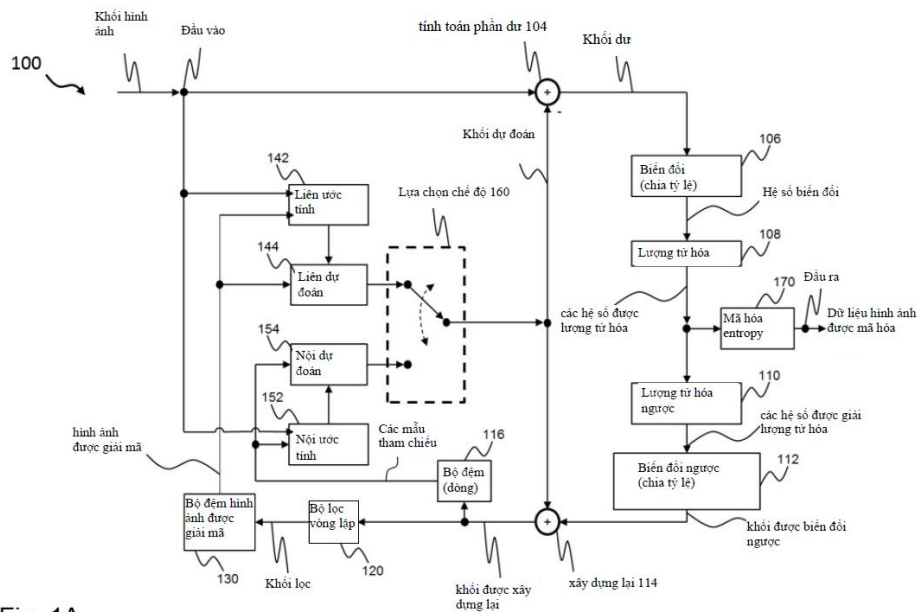


Fig. 1A

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế liên quan đến lĩnh vực mã hóa video. Cụ thể hơn, sáng chế đề cập đến bộ lọc để tạo mã video, phương pháp lọc các khung video được xây dựng lại, và phương pháp lọc các khối video cũng như thiết bị mã hóa và thiết bị giải mã bao gồm bộ lọc này để tạo mã video.

Tính trạng kỹ thuật của sáng chế

Video kỹ thuật số đã được sử dụng rộng rãi kể từ khi đĩa DVD ra đời. Trước khi truyền, video được mã hóa và truyền bằng phương tiện truyền dẫn. Người xem nhận video và sử dụng thiết bị xem để giải mã và hiển thị video. Trong những năm qua, chất lượng video đã được cải thiện, chẳng hạn như do độ phân giải, độ sâu màu và tốc độ khung hình cao hơn. Điều này đã dẫn đến các luồng dữ liệu lớn hơn ngày nay thường được vận chuyển qua mạng internet và mạng thông tin di động.

Tuy nhiên, video có độ phân giải cao hơn thường yêu cầu nhiều băng thông hơn vì chúng có nhiều thông tin hơn. Để giảm yêu cầu băng thông, tiêu chuẩn tạo mã video liên quan đến nén video đã được giới thiệu. Khi video được mã hóa, yêu cầu băng thông (hoặc yêu cầu bộ nhớ tương ứng trong trường hợp lưu trữ) sẽ giảm xuống. Thường thì việc giảm này đi kèm với chi phí chất lượng. Do đó, các tiêu chuẩn tạo mã video cố gắng tìm ra sự cân bằng giữa yêu cầu băng thông và chất lượng.

Vì nhu cầu liên tục cải thiện chất lượng và giảm yêu cầu băng thông, các giải pháp duy trì chất lượng với yêu cầu băng thông giảm hoặc cải thiện chất lượng trong khi duy trì yêu cầu băng thông được liên tục tìm kiếm. Hơn nữa, đôi khi các thỏa hiệp có thể được chấp nhận. Ví dụ, có thể chấp nhận việc tăng yêu cầu băng thông nếu việc cải thiện chất lượng là đáng kể.

Tạo mã video hiệu quả cao (HEVC, High Efficiency Video Coding) là một ví dụ về tiêu chuẩn tạo mã video thường được những người có hiểu biết trung bình trong lĩnh vực này biết đến. Trong HEVC, để tách một đơn vị tạo mã (CU, coding unit) thành các đơn vị dự đoán (PU, prediction unit) hoặc đơn vị biến đổi (TU, transform unit). Tiêu

chuẩn thế hệ tiếp theo Tạo mã video đa năng (VVC, Versatile Video Coding) là dự án video chung gần đây nhất của Nhóm chuyên gia tạo mã video ITU-T (VCEG, Video Coding Experts Group) và tổ chức tiêu chuẩn hóa nhóm chuyên gia hình ảnh chuyển động ISO / IEC (MPEG, Moving Picture Experts Group), hợp tác cùng nhau trong mối quan hệ đối tác được biết đến với tư cách là Nhóm khám phá video chung (JVET, Joint Video Exploration Team). VVC còn được gọi là tiêu chuẩn ITU-T H.266/Tạo mã video thế hệ mới (NGVC, Next Generation Video Coding). Trong VVC, nó loại bỏ các khái niệm về nhiều loại phân vùng, tức là nó loại bỏ sự tách biệt của các khái niệm CU, PU và TU ngoại trừ khi cần thiết cho các CU có kích thước quá lớn so với chiều dài biến đổi tối đa và hỗ trợ linh hoạt hơn cho các hình dạng phân vùng CU.

Lọc hình ảnh thường được sử dụng để nhấn mạnh các tính năng nhất định của hình ảnh hoặc để nâng cao chất lượng mục tiêu hoặc cảm quan của hình ảnh được lọc. Lọc hình ảnh phải đối phó với nhiều nguồn nhiễu khác nhau. Theo đó, nhiều cách tiếp cận khác nhau để nâng cao chất lượng đã được đề xuất và hiện đang được sử dụng. Ví dụ: trong phương pháp bộ lọc vòng lặp thích ứng (ALF, adaptive Loop filter), mỗi khung được xây dựng lại được chia thành một tập hợp các khối nhỏ (siêu điểm ảnh) và mỗi khối được lọc bởi bộ lọc vòng lặp thích ứng trong đó mỗi điểm ảnh của khung được xây dựng lại đã lọc là tổng trọng số của một số điểm ảnh trong vùng được kết nối của điểm ảnh từ khung được xây dựng lại xung quanh vị trí của điểm ảnh được lọc tạo. Hệ số trọng số (còn được gọi là hệ số bộ lọc) có thuộc tính đối xứng trung tâm và được truyền từ bộ mã hóa đến phía bộ giải mã. Các cạnh thường có kích thước lớn và do đó số lượng hệ số trọng số truyền có thể trở nên quá lớn để xử lý hiệu quả. Số lượng lớn các hệ số trọng số yêu cầu tối ưu hóa tỷ lệ biến dạng phức tạp (RDO, rate-distortion optimization) ở phía bộ mã hóa để giảm số lượng hệ số trọng số để truyền. Về phía bộ giải mã, ALF yêu cầu triển khai các hệ số phổ quát và các hệ số này phải được tải lại cho mỗi khối điểm ảnh 2x2

Do đó, cần có một bộ lọc và phương pháp cải tiến cho phép cải thiện chất lượng dự đoán với độ phức tạp thấp và do đó, tăng hiệu quả tạo mã video.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất bộ lọc và phương pháp cải tiến cho phép nâng

cao hiệu quả lọc với độ phức tạp giới hạn và do đó, tăng hiệu quả tạo mã video.

Các đối tượng nêu trên và các đối tượng khác đạt được là do đối tượng của các điểm yêu cầu bảo hộ độc lập. Các hình thức triển khai khác rõ ràng từ các 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 để tạo mã video, trong đó bộ lọc được tạo cấu hình để xử lý khối để tạo khối được lọc, trong đó khối bao gồm nhiều điểm ảnh. Bộ lọc gồm bộ nhớ lưu trữ bao gồm các lệnh; và một hoặc nhiều bộ xử lý trong việc truyền thông với bộ nhớ. Một hoặc nhiều bộ xử lý thực thi các lệnh để: tải điểm ảnh hiện tại và các điểm ảnh lân cận của nó vào bộ đệm tuyến tính theo mẫu quét định trước; thu các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu các thành phần phổ đã được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc; thu các điểm ảnh được lọc bằng cách thực hiện phép biến đổi 1D ngược cho các thành phần phổ đã được lọc; và tạo khối được lọc được dựa trên các điểm ảnh được lọc.

Ví dụ, khối (hoặc khung) có thể là khối được dự đoán, và khối được lọc là khối được dự đoán được lọc. Ví dụ khác, khối (hoặc khung) có thể là khối được xây dựng lại, và khối được lọc là khối được xây dựng lại được lọc.

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 suy ra từ tham số lượng tử hóa (QP, quantization parameter) bộ mã hóa/giải mã (codec).

Ví dụ khác, thành phần phổ thứ nhất bị bỏ qua mà không lọc khi hệ số khuếch đại $G(i, \sigma)$ cho thành phần phổ thứ nhất bằng 1. 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.

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

Ví dụ, mẫu quét định trước được định nghĩa là tập hợp các độ lệch không gian

hoặc mảnh liên quan tới vị trí của điểm ảnh hiện tại trong khối được xây dựng lại. Các độ lệch trở tới các điểm ảnh lân cận trong khối được xây dựng lại. Ít nhất một điểm ảnh được lọc có thể được đặt về vị trí ban đầu của nó theo mẫu quét định trước. Khi tất cả các điểm ảnh được lọc được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và bộ đệm tích lũy có thể được khởi tạo bằng 0 trước khi thu các thành phần phổ đã được lọc. Khi các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia cho số các điểm ảnh thêm vào vị trí hiện tại của bộ đệm tích lũy; một hoặc nhiều bộ xử lý được tạo cấu hình để tạo khối được xây dựng lại được lọc được dựa trên các điểm ảnh được lọc cuối cùng.

Tùy chọn, các sự chênh lệch giữa tất cả các điểm ảnh được lọc và không được lọc tương ứng được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và bộ đệm tích lũy có thể được khởi tạo bởi các điểm ảnh không được lọc được nhân bởi số lượng các giá trị điểm ảnh tối đa được thêm vào trong khối. Các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng các giá trị điểm ảnh tối đa được thêm vào trong khối.

Theo khía cạnh thứ hai sáng chế đề xuất đến phương pháp lọc tương ứng để xử lý khối để tạo khối được lọc, trong đó khối bao gồm nhiều điểm ảnh. Mỗi điểm ảnh được liên kết với giá trị 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 định trước; thu các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu các thành phần phổ đã được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc; thu các điểm ảnh được lọc bằng cách thực hiện phép biến đổi 1D ngược cho các thành phần phổ đã được lọc; và tạo khối được lọc được dựa trên các điểm ảnh được lọc.

Ví dụ, khối (hoặc khung) có thể là khối được dự đoán, và khối được lọc là khối được dự đoán được lọc. Ví dụ khác, khối (hoặc khung) có thể là khối được xây dựng lại, và khối được lọc là khối được xây dựng lại được lọc.

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 suy ra từ tham số lượng tử hóa (QP) bộ mã hóa/giải mã.

Ví dụ khác, thành phần phổ thứ nhất bị bỏ qua mà không lọc khi hệ số khuếch

đại $G(i, \sigma)$ cho thành phần phổ thứ nhất bằng 1. Thành phần phổ thứ nhất có thể tương ứng với giá trị DC.

Ví dụ khác, lọc các thành phần phổ được dựa trên bảng tra cứu (look up table, LUT). Việc tạo LUT có thể được dựa trên hàm phụ cho ít nhất một số các tham số lượng tử hóa (quantization parameters, QPs). Hàm phụ có thể là phương trình đường thẳng đi qua các điểm (i, THR) và $(a, 0)$, trong đó $a > 0$ và a phụ thuộc vào tham số lọc σ hoặc giá trị QP. Ví dụ, đối với QP cuối cùng trong tập các QP, a bằng 11; hoặc đối với QP cuối cùng thứ hai trong tập các QP, a bằng 9.

Ví dụ khác, phương pháp còn gồm bước: loại bỏ N bit từ bảng các giá trị 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 cho QP thấp hơn so với QP cao hơn từ tập các QP, ví dụ, đối với QP thứ nhất trong tập các QP, N bằng 2; hoặc đối với các QP còn lại từ tập các QP, N bằng 3. Ngoài ra, khi N được lựa chọn cao hơn đối với QP cao hơn so với QP thấp hơn từ tập các QP của các QP, ví dụ, đối với QP cuối cùng hoặc cho hai QP cuối cùng trong tập các QP, N bằng 4; hoặc đối với các QP còn lại từ tập các QP, N bằng 3. Ngoài ra, khi N được lựa chọn ít hơn cho QP thấp hơn và cao hơn đối với QP cao hơn so với QP còn lại từ tập các QP, ví dụ, đối với QP thứ nhất trong tập các QP, N bằng 2; đối với QP cuối cùng hoặc cho hai QP cuối cùng trong tập các QP, N bằng 4; hoặc, đối với các QP còn lại từ tập các QP, N bằng 3.

Ví dụ, mẫu quét định trước được định nghĩa là tập hợp các độ lệch không gian hoặc mảnh liên quan tới vị trí của điểm ảnh hiện tại trong khối được xây dựng lại. Các độ lệch trở tới các điểm ảnh lân cận trong khối được xây dựng lại. Ít nhất một điểm ảnh được lọc có thể được đặt về vị trí ban đầu của nó theo mẫu quét định trước. Khi tất cả các điểm ảnh được lọc được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và bộ đệm tích lũy có thể được khởi tạo bằng 0 trước khi thu các thành phần phổ đã được lọc. Khi các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia cho số các điểm ảnh thêm vào vị trí hiện tại của bộ đệm tích lũy; một hoặc nhiều bộ xử lý được tạo cấu hình để tạo khối được xây dựng lại được lọc được dựa trên các điểm ảnh được lọc cuối cùng.

Tùy chọn, các sự chênh lệch giữa tất cả các điểm ảnh được lọc và không được

lọc tương ứng được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và bộ đệm tích lũy có thể được khởi tạo bởi các điểm ảnh không được lọc được nhân bởi số lượng các giá trị điểm ảnh tối đa được thêm vào trong khối. Các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia bởi số lượng các giá trị điểm ảnh tối đa được thêm vào trong khối.

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

Theo khía cạnh thứ tư sáng chế đề xuất đến thiết bị giải mã để giải mã khung hiện tại từ luồng bit nhận được, trong đó thiết bị giải mã bao gồm bộ lọc theo khía cạnh thứ nhất của sáng chế.

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

Do đó, bộ lọc được cung cấp để cho phép cải thiện hiệu quả tạo mã video. Cụ thể hơn, bộ lọc được cải tiến theo các phương án của sáng chế ước tính các tham số lọc từ khung của nó mà không báo hiệu các tham số lọc và, do đó, yêu cầu tín hiệu ít hơn đáng kể so với các bộ lọc thông thường, mà các hệ số trọng số tín hiệu để lọc trong miền hình ảnh.

Mô tả vắn tắt các hình vẽ

Các phương án thêm nữa của sáng chế sẽ được mô tả liên quan đến các hình vẽ sau, trong đó:

Fig. 1A thể hiện sơ đồ minh họa thiết bị mã hóa theo phương án bao gồm bộ lọc theo phương án;

Fig. 1B thể hiện sơ đồ minh họa thiết bị mã hóa theo phương án bao gồm bộ lọc theo phương án;

Fig. 2A thể hiện sơ đồ minh họa thiết bị giải mã theo phương án bao gồm bộ lọc theo phương án;

Fig. 2B thể hiện sơ đồ minh họa thiết bị giải mã theo phương án bao gồm bộ lọc theo phương án;

Fig. 3A thể hiện sơ đồ minh họa các khía cạnh của thủ tục lọc được thực hiện trong bộ lọc theo phương án;

Fig. 3B thể hiện sơ đồ minh họa các khía cạnh của thủ tục lọc được thực hiện trong bộ lọc theo phương án;

Fig. 3C thể hiện sơ đồ minh họa các khía cạnh của thủ tục lọc được thực hiện trong bộ lọc theo phương án khác;

Fig. 4A minh họa các mẫu cho vị trí điểm ảnh khác nhau trong khối được xây dựng lại vuông;

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

Fig. 4C là ví dụ về việc đệm;

Fig. 4D là ví dụ khác về việc đệm;

Fig. 5A là lưu đồ minh họa các bước của phương pháp lọc theo phương án;

Fig. 5B là lưu đồ minh họa các bước của phương pháp lọc theo phương án;

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

Fig. 7 là thiết kế phản ứng điển hình của lọc nhóm 2x2 được dựa trên các flip-flop (mạch lật);

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

Fig. 9 ví dụ, hiển thị kết quả của việc tối ưu hóa LUT;

Fig. 10 ví dụ, minh họa khoảng trống không mong muốn trong hàm truyền bộ lọc;

Fig. 11 ví dụ, đại diện cho cùng một bảng với các mục trong bảng được vẽ từng cái một;

Fig. 12 ví dụ, minh họa phương pháp làm thế nào để loại bỏ khoảng cách bằng cách sử dụng hàm phụ;

Fig. 13 minh họa ví dụ về loại bỏ khoảng cách bằng cách lấy tối đa từ hai giá trị trong khi tạo LUT;

Fig. 14, ví dụ, minh họa hàm truyền bộ lọc sau khi áp dụng phương pháp mô tả ở trên;

Fig. 15 minh họa ví dụ các hàm truyền bộ lọc phụ thuộc vào năm QP trong tập hợp;

Fig. 16 minh họa ví dụ các hàm truyền bộ lọc cho năm QP trong tập hợp được dựa trên các bảng tương ứng;

Fig. 17 minh họa ví dụ các hàm truyền bộ lọc cho năm QP trong tập hợp được dựa trên các bảng tương ứng; và

Fig. 18 là một sơ đồ minh họa cấu trúc mẫu của thiết bị theo phương án.

Trong các hình khác nhau, các dấu hiệu tham chiếu giống nhau sẽ được sử dụng cho các tính năng giống nhau hoặc tương đương về chức năng.

Mô tả chi tiết sáng chế

Trong phần mô tả sau đây, tham chiếu đến các hình vẽ kèm theo, là một phần của sáng chế, và trong đó được thể hiện bằng cách minh họa, các khía cạnh cụ thể mà sáng chế có thể được đưa vào. Điều này được hiểu rằng các khía cạnh khác có thể được sử dụng và các thay đổi về cấu trúc hoặc logic có thể được thực hiện mà không vượt quá phạm vi của sáng chế. Do đó, mô tả chi tiết sau đây không được hiểu theo nghĩa giới hạn sáng chế, vì phạm vi của sáng chế được xác định bởi các yêu cầu bảo hộ phụ thuộc.

Ví dụ, cần hiểu rằng bậc lệ liên quan đến 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 đó và ngược lại. Ví dụ, nếu bước phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm đơn vị để thực hiện bước phương pháp được mô tả, ngay cả khi đơn vị đó không được mô tả hoặc minh họa rõ ràng trong các hình. Hơn nữa, cần hiểu rằng các đặc điểm của các khía cạnh mẫu khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi có ghi chú cụ thể khác.

Fig. 1A thể hiện thiết bị mã hóa 100 theo phương án bao gồm bộ lọc 120 theo

phương án. Thiết bị mã hóa 100 được tạo cấu hình để mã hóa khối của khung của tín hiệu video bao gồm nhiều khung (có thể được gọi là ảnh hoặc hình ảnh), trong đó mỗi khung có thể chia thành nhiều khối và mỗi khối bao gồm nhiều các điểm ảnh. Theo một phương án, các khối có thể là các khối macro, các đơn vị cây tạo mã, các đơn vị tạo mã, các đơn vị dự đoán và/hoặc các khối dự đoán.

Thuật ngữ “khối” trong phần mô tả này được sử dụng cho bất kỳ khối loại nào hoặc cho bất kỳ khối độ sâu nào, ví dụ, thuật ngữ “khối” được bao gồm nhưng không giới hạn ở khối gốc, khối, khối phụ, nút lá, và vân vân. Các khối được tạo mã không nhất thiết phải có cùng kích thước. Một ảnh có thể bao gồm các khối có kích thước khác nhau và các mảnh khối ảnh khác nhau của chuỗi video cũng có thể khác nhau.

Theo phương án được lấy làm ví dụ thể hiện trên Fig. 1A, thiết bị mã hóa 100 được triển khai dưới dạng bộ mã hóa tạo mã video lai. Thông thường, khung của tín hiệu video thứ nhất là nội khung, mà được giải mã chỉ sử dụng nội dự đoán. Cuối cùng, phương án thiết bị mã hóa 100 được thể hiện trên Fig. 1A bao gồm đơn vị nội dự đoán 154 để nội dự đoán. Nội khung có thể được giải mã mà không có thông tin từ các khung khác. Đơn vị nội dự đoán 154 có thể thực hiện nội dự đoán của khối trên cơ sở thông tin được cung cấp bởi đơn vị nội ước tính 152.

Các khối của các khung tiếp theo sau nội khung thứ nhất có thể được tạo mã sử dụng liên dự đoán hoặc nội dự đoán, như được lựa chọn bởi đơn vị lựa chọn chế độ 160. Cuối cùng, thiết bị mã hóa 100 được thể hiện trên Fig. 1A còn bao gồm đơn vị liên dự đoán 144. Nói chung, đơn vị liên dự đoán 144 có thể được tạo cấu hình để thực hiện việc bù chuyển động của khối được dựa trên việc ước tính chuyển động được cung cấp bởi đơn vị liên ước tính 142.

Hơn nữa, tín hiệu dự đoán trong phương án bộ mã hóa lai được thể hiện trên Fig. 1B thu từ nội dự đoán hoặc liên dự đoán còn được lọc bởi bộ lọc 145.

Hơn nữa, theo phương án bộ mã hóa lai được thể hiện trên Fig. 1A và Fig. 1B đơn vị tính toán dư 104 xác định sự chênh lệch giữa khối gốc và khối dự đoán của nó, tức là khối dư định nghĩa lỗi dự đoán của dự đoán nội ảnh/liên ảnh. Khối dư này được biến đổi bởi đơn vị biến đổi 106 (cho các trường hợp sử dụng DCT) và các hệ số biến đổi được lượng tử hóa bởi đơn vị lượng tử hóa 108. Đầu ra của đơn vị lượng tử hóa

108 cũng như là thông tin mã hóa hoặc thông tin phụ được cung cấp, chẳng hạn, bởi đơn vị nội dự đoán 154, đơn vị liên dự đoán 144 và bộ lọc 120 còn được mã hóa bởi đơn vị mã hóa entropy 170.

Bộ mã hóa video lại thường sao chép việc xử lý của bộ giải mã để cả hai cùng tạo ra các dự đoán giống nhau. Do đó, theo phương án được thể hiện trên Fig. 1 đơn vị lượng tử hóa ngược 110 và đơn vị biến đổi ngược thực hiện các phép toán ngược của đơn vị biến đổi 106 và đơn vị lượng tử hóa 108 và nhân đôi giá trị gần đúng được giải mã của khối dư. Dữ liệu khối dư được giải mã sau đó được bổ sung vào các kết quả của việc dự đoán, tức là khối dự đoán, bởi đơn vị xây dựng lại 114. Sau đó, đầu ra của đơn vị xây dựng lại 114 có thể được cung cấp cho bộ đệm dòng 116 được sử dụng cho nội dự đoán và còn được xử lý bởi bộ lọc 120, mà sẽ được mô tả chi tiết hơn dưới đây. Ảnh cuối cùng được lưu trữ trong bộ đệm ảnh được giải mã 130 và có thể được sử dụng để liên dự đoán các khung tiếp theo.

Fig. 2A minh họa thiết bị giải mã 200 theo phương án bao gồm bộ lọc 220 theo phương án. Thiết bị giải mã 200 được tạo cấu hình để giải mã khối khung của tín hiệu video được mã hóa. Theo phương án được thể hiện trên Fig. 2A thiết bị giải mã 200 được thực thi dưới dạng bộ giải mã lai. Đơn vị giải mã entropy 204 thực hiện giải mã entropy của dữ liệu ảnh được mã hóa, mà thường có thể bao gồm các lỗi dự đoán (nghĩa là các khối dư), dữ liệu chuyển động và thông tin phụ khác, mà cần thiết, cụ thể hơn, đối với đơn vị nội dự đoán 254 và đơn vị liên dự đoán 244 cũng như là các thành phần khác của thiết bị giải mã 200, như là bộ lọc 220. Nói chung, đơn vị nội dự đoán 254 và đơn vị liên dự đoán 244 của thiết bị giải mã 200 được thể hiện trên Fig. 2A được lựa chọn bởi đơn vị lựa chọn chế độ 260 và chức năng theo cách tương tự đơn vị nội dự đoán 154 và đơn vị liên dự đoán 144 của thiết bị mã hóa 100 được thể hiện trên Fig. 1, mà các dự đoán đồng nhất có thể được tạo ra bởi thiết bị mã hóa 100 và thiết bị giải mã 200. Đơn vị xây dựng lại 214 của thiết bị giải mã 200 được tạo cấu hình để xây dựng lại khối trên cơ sở của khối được dự đoán được lọc và khối dư được cung cấp bởi đơn vị lượng tử hóa ngược 210 và đơn vị biến đổi ngược 212. Như trong trường hợp của thiết bị mã hóa 100, khối được xây dựng lại có thể được cung cấp cho bộ đệm dòng 216 được sử dụng để nội dự đoán và khối/khung được lọc có thể được cung cấp cho bộ đệm ảnh được giải

mã 230 bởi bộ lọc 220 để liên dự đoán.

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

Như đã được mô tả ở trên, bộ lọc 120, 220 có thể được sử dụng ở mức độ khung, ví dụ, bộ lọc 120, 220 có thể được tạo cấu hình để xử lý khung được xây dựng lại từ luồng video xây dựng lại được giải mã để tạo khung được xây dựng lại được lọc, trong đó khung được xây dựng lại gồm nhiều khối. Bộ lọc 120, 220 có thể cũng được sử dụng ở cấp độ khối ngay sau việc xây dựng lại khối mà không cần đợi cả khung, ví dụ, bộ lọc 120, 220 có thể được tạo cấu hình để xử lý khối được xây dựng lại để tạo khối được xây dựng lại được lọc, trong đó khối được xây dựng lại gồm nhiều các điểm ảnh.

Bộ lọc 120, 220, 145 hoặc 264 bao gồm một hoặc nhiều bộ xử lý (hoặc một hoặc nhiều đơn vị xử lý). Như sẽ được giải thích chi tiết hơn dưới đây, một hoặc nhiều bộ xử lý (hoặc một hoặc nhiều đơn vị 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 định trước (cách khác, thứ tự quét, hoặc mẫu quét); thu các thành phần phổ bằng cách thực hiện biến đổi 1D cho mỗi điểm ảnh trong bộ đệm tuyến tính; thu các thành phần phổ đã được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc; thu các điểm ảnh được lọc bằng cách thực hiện phép biến đổi 1D ngược cho phổ được lọc; và tạo khối được xây dựng lại được lọc được dựa trên các điểm ảnh được lọc được ước tính trên các bước xử lý trước. Ví dụ, hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc. Ví dụ khác, hệ số khuếch đại phụ thuộc vào một hoặc nhiều các tham số lọc và một hoặc nhiều các thành phần phổ tương ứng. Ví dụ khác, hệ số khuếch đại tương ứng có thể phụ thuộc vào một hoặc nhiều các tham số lọc và thành phần phổ tương ứng cũng như là các thành phần phổ lân cận ở bên trái và bên phải của thành phần phổ.

Phần bộc lộ mô tả bộ lọc trong vòng cho bộ mã hóa/giải mã video bị tổn hao mà thực hiện lọc nội vùng và/hoặc không nội vùng của khối được xây dựng lại từ khung được xây dựng lại. Theo ví dụ, khung được xây dựng lại được chia thành tập các khối hình chữ nhật không chồng lên nhau nhỏ (các khối CU). Ở bước tiếp theo mỗi khối được

xây dựng lại (khối CU được xây dựng lại) được lọc trong miền tần số độc lập với các khối được xây dựng lại khác. Bộ lọc có thể cũng được áp dụng sau khi biến đổi và xây dựng lại, và kết quả được lọc được sử dụng cho cả đầu ra cũng như là cho dự đoán không gian và thời gian.

Ví dụ khác, phần bộc lộ mô tả bộ lọc dự đoán cho bộ mã hóa/giải mã video bị tổn hao mà thực hiện lọc nội vùng và/hoặc không nội vùng của khối dự đoán của khung được xây dựng lại.

Ở bước thứ nhất của quá trình xử lý tất cả các điểm ảnh trong khối được xây dựng lại có thể được xử lý độc lập với nhau. Để xử lý điểm ảnh $r(0)$, các điểm ảnh lân cận được sử dụng. Ví dụ, như được minh họa trên Fig. 3A, các điểm ảnh $r(1)$ tới $r(7)$ được sử dụng, và các điểm ảnh $r(0)$ tới $r(7)$ tạo thành một nhóm xử lý.

Fig. 3A hoặc Fig. 3B thể hiện sơ đồ 300, 300' minh họa các khía cạnh của thủ tục lọc được thực hiện trong bộ lọc theo phương án. Ở bước 302, 302', điểm ảnh hiện tại và các điểm ảnh lân cận của nó từ khối được tải vào bộ đệm tuyến tính theo mẫu quét định trước. Ví dụ, khối có thể là khối được dự đoán. Ví dụ khác, khối có thể là khối được xây dựng lại.

Ở bước 304, 304', biến đổi 1D được thực hiện đối với điểm ảnh $r(0)$ và các điểm ảnh lân cận của nó $r(1)$ tới $r(7)$ trong bộ đệm tuyến tính để thu các thành phần phổ R :

$$R = 1D_Transform(r)$$

Ví dụ, biến đổi 1D có thể là biến đổi Hadamard.

Cần được hiểu rằng nên thực hiện biến đổi 1D trên 4 điểm ảnh trong hàng (ví dụ: các điểm ảnh A,B,C,D như trong ví dụ trên Fig. 3B) hoặc thực hiện biến đổi 2D thành các điểm ảnh được đặt trong không gian A,B,C,D liên quan đến việc thực hiện cụ thể. Áp dụng biến đổi 2D trên 4 điểm ảnh A,B,C,D được đặt trong khối 2x2 có thể dẫn đến kết quả tương tự như áp dụng biến đổi 1D thành 4 điểm ảnh A,B,C,D được đặt thành hàng. Ở bước 306, 306', việc lọc được thực hiện trong miền tần số được dựa trên phép nhân từng thành phần phổ $R(i)$ với hệ số khuếch đại $G(i, \sigma)$ tương ứng để thu các thành phần phổ đã được lọc $F(i)$:

Tập các hệ số khuếch đại cho tất cả các thành phần phổ là đáp ứng xung tần số

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

của bộ lọc.

Ví dụ, hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc. Ví dụ khác, hệ số khuếch đại phụ thuộc vào một hoặc nhiều các tham số lọc và một hoặc nhiều các thành phần phổ tương ứng. Ví dụ khác, hệ số khuếch đại tương ứng có thể phụ thuộc vào một hoặc nhiều các tham số lọc và thành phần phổ tương ứng cũng như là các thành phần phổ lân cận bên trái và bên phải của thành phần phổ. Nếu mỗi hệ số khuếch đại là hàm số của thành phần phổ của khối được xây dựng lại và tham số lọc, hoặc là hàm số thành phần phổ của khối được dự đoán và tham số lọc, hệ số khuếch đại $G(i, \sigma)$ có thể được mô tả bằng công thức sau làm ví dụ:

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

trong đó (i) là chỉ số của thành phần 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 số lượng thành phần phổ. Ví dụ, m là 1, 2, 3, 4 Các thành phần phổ khác có thể có cùng hệ số khuếch đại, hoặc có thể có các hệ số khuếch đại khác nhau.

Đối với những biến đổi mà có các thành phần phổ tương ứng với trung bình (FFT, DCT, DST v.v.) hoặc tổng (Hadamard) của các mẫu đầu vào của khối biến đổi (thường thành phần thứ nhất tương ứng với giá trị DC) có thể có lợi khi có hệ số lọc bằng 1 để tránh sự thay đổi của độ sáng trung bình của khối được lọc. Điều đó có nghĩa là bỏ qua (không lọc) đối với thành phần phổ thứ nhất tương ứng với giá trị DC.

σ là tham số lọc, có thể suy ra từ tham số lượng tử hóa (QP) bộ mã hóa/giải mã ở phía bộ mã hóa và bộ giải mã, ví dụ, sử dụng biểu thức sau:

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

trong đó k , n và s là các hằng số có các giá trị như ví dụ: $k = 2.64$, $n = 0.1296$, $s = 11$.

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

Các tham số k, n và s có thể được lựa chọn theo cách làm cho sigma (σ) phụ thuộc vào kích thước bước lượng tử hóa mà tăng gấp đôi mỗi khi giá trị QP tăng thêm 6 trong các tiêu chuẩn tạo mã video mới nhất. Ví dụ các tham số $k=0,5, n=1/6$ và $s=0$ ($k = 0,5, n = \frac{1}{6}$ and $s = 0$), các tham số σ được suy ra từ:

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

Các ma trận tỷ lệ lượng tử hóa được sử dụng rộng rãi để cải thiện chất lượng nén video. Trong phương pháp này kích thước bước lượng tử hóa được suy ra được dựa trên QP được nhân với hệ số tỷ lệ được truyền trong luồng bit. Đối với phương pháp này phép tính các tham số σ có thể được dựa trên kích thước bước lượng tử hóa được chia tỷ lệ thực tế được sử dụng để cho QP nhất định:

$$\sigma = k * \text{Quantization_step_size}(QP - s)$$

Các hằng số k, n và s có thể có các giá trị cố định để tính toán σ , hoặc có thể có các giá trị khác nhau phụ thuộc vào QP, kích thước và hình dạng khối, loại dự đoán (liên/nội) cho khối hiện tại. Ví dụ đối với các khối vuông nội dự đoán với kích thước 32x32 hoặc tham số s có thể được tính toán như $s = 11 + 8 = 19$. Vì tham số lọc tương đương σ có giá trị nhỏ hơn dẫn đến bộ lọc mịn hơn mà phù hợp hơn với khối nội dự đoán hình vuông lớn thường tương ứng với các vùng phẳng. Ví dụ khác, k có thể được sửa đổi được dựa trên độ sâu bit của các điểm ảnh $k_{mod} = k * (1 \ll (\text{bit_depth} - 8))$

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

Các chi tiết về LUT được thảo luận.

Có thể lưu ý rằng việc lọc ngụ ý nhân thành phần phổ $R(i)$ với hệ số tỷ lệ mà luôn nhỏ hơn 1. Cũng có thể thấy rằng tại các giá trị cao của $R(i)$ hệ số tỷ lệ gần bằng 1. Được dựa trên lọc phổ quan trắc được thực thi sử dụng bảng tra cứu mà cho phép loại trừ phép nhân và phép chia từ hoạt động lọc:

Hệ số khuếch đại phổ nhỏ hơn 1, nên việc lọc được thực thi được dựa trên việc đọc bảng tra cứu (look up table , LUT) ngắn theo các biểu thức sau:

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

Trong đó $LUT(R(i), \sigma) = \frac{R(i)^3}{R(i)^2 + m \cdot \sigma^2}$. (i) là chỉ số của thành phần 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 số lượng thành phần phổ. Ví dụ, m là 1, 2, 3, 4

Ví dụ, THR có thể được tính toán từ biểu thức sau,

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

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

Để giảm hơn nữa kích thước của LUT ngưỡng thứ hai có thể được đưa ra để thay thế các giá trị được lọc nhỏ bằng 0. Trong trường hợp này thành phần phổ được lọc là $F(i, \sigma)$ còn được suy ra là:

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

Trong đó THR2 xác định ngưỡng bên dưới mà thành phần phổ được lọc được coi bằng 0. THR2 thứ hai có thể cũng được xác định phụ thuộc vào giá trị QP.

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

$$f = 1D_Inverse_Transform(F)$$

Ở bước 310, 310', kết quả của biến đổi 1D ngược được đặt vào bộ đệm tuyến tính của các điểm ảnh được xây dựng 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 được dựa trên các điểm ảnh được lọc ước tính trên các bước xử lý trước đó. Ví dụ, khối được lọc có thể là khối được dự đoán được lọc. Ví dụ khác, khối được lọc có thể là khối được xây dựng lại được lọc.

Như được thể hiện trên Fig. 3A như phương án, sau bước lọc 306, điểm ảnh được lọc $f(0)$ được đặt về vị trí ban đầu của nó theo mẫu quét định trước. Các mẫu lọc khác $f(1) - f(7)$ không được sử dụng. Theo phương án 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 thêm vào bộ đệm tích lũy theo mẫu quét định trước được sử dụng ở bước 302 của Fig. 3A. Bộ đệm tích lũy còn được khởi tạo bằng 0 trước bước lọc. Ở bước chuẩn hóa cuối cùng, các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia cho số các điểm ảnh được bổ sung vào vị trí hiện tại của bộ đệm tích lũy, cách khác, số các giá trị của các điểm ảnh được bổ sung vào vị trí hiện tại của bộ đệm tích lũy trên các bước xử lý trước. Sau khi khối được xây dựng lại được lọc hoặc khối được dự đoán được tạo được dựa trên các điểm ảnh được lọc cuối cùng.

Như phương án khác, bộ lọc có cùng cách triển khai cho cả việc lọc đơn vị nội tạo mã và liên tạo mã (CU, coding unit).

Bộ lọc miền biến đổi Hadamard luôn được áp dụng cho các khối được xây dựng lại sắc độ với các hệ số biến đổi khác 0, loại từ các khối 4x4 và nếu tham số lượng tử hóa lớn hơn 17. Các tham số lọc rõ ràng được suy ra từ thông tin được tạo mã. Bộ lọc được đề xuất, nếu được áp dụng, được thực hiện trên các mẫu được giải mã ngay sau khi xây dựng lại khối. Kết quả được lọc được sử dụng cho cả đầu ra cũng như là để dự đoán không gian và thời gian.

Quá trình lọc được thảo luận, như được trình bày theo sơ đồ trên Fig. 3C.

Đối với mỗi điểm ảnh từ việc xử lý điểm ảnh khối được xây dựng lại bao gồm

các bước sau:

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

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

Việc lọc phổ được dựa trên biểu thức (1) và (2).

Thành phần phổ thứ nhất tương ứng với giá trị DC bị bỏ qua mà không lọc.

Biến đổi Hadamard 4 điểm ngược của phổ được lọc.

Sau bước lọc các điểm ảnh được lọc được đặt về vị trí ban đầu của nó trong bộ đệm tích lũy.

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

Có thể thấy rằng tất cả các điểm ảnh trong khối có thể được xử lý độc lập trong trường hợp yêu cầu song song tối đa.

Trong các phương án này, ngưỡng THR được đặt thành giá trị xác định trước, ví dụ, 128, mà trong trường hợp triển khai đơn giản, yêu cầu lưu trữ 128 ($1 \ll 7$) các mục nhập của các giá trị 7 bit cho mỗi QP.

Kích thước của LUT tác động đến lượng yêu cầu bộ nhớ trên chip và chi phí triển khai phần cứng của bộ lọc. Để giảm dung lượng lưu trữ trên chip LUT nên chỉ được tính toán cho một tập được giới hạn các QP bắt đầu từ QP 20 với khoảng thời gian không đổi 8. Tổng cộng năm LUT định trước (cho năm nhóm QP) được lưu trữ. Để lọc khối hiện tại QP của CU được làm thành khối gần nhất so với bảng.

Để giảm thêm kích thước LUT, N bit thấp nhất sẽ bị loại bỏ (hoặc bỏ qua) trong quá trình tạo LUT. Điều đó cho phép có biểu diễn bảng thưa.

Đối với triển khai mẫu A, N bằng 2, mà dẫn đến $7 - 2 = 5$ bit của chiều sâu bảng (32 các mục nhập của các giá trị 7 bit);

Đối với triển khai mẫu B, N bằng 3, mà dẫn đến $7 - 3 = 4$ bit của chiều sâu bảng

(16 các mục nhập của các giá trị 7 bit).

Do đó tổng kích thước bộ nhớ được yêu cầu cho bộ nhớ LUT thực thể:

Đối với triển khai mẫu A: $5 \times 32 \times 7 \text{ bit} = 1120 \text{ bit} = 140 \text{ byte}$;

Đối với triển khai mẫu B: $5 \times 16 \times 7 \text{ bit} = 560 \text{ bit} = 70 \text{ byte}$;

Triển khai mẫu B nhằm mục tiêu đến 16 byte của kích thước LUT để cho phép truy nhập song song trong triển khai phần mềm do khả năng lưu trữ toàn bộ LUT trong một thanh ghi SSE 16 byte do đó cấu hình này được đề xuất.

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

// quét các điểm ảnh được xây dựng lại/được dự đoán

const int x0 = pIn[p0];

const int x1 = pIn[p1];

const int x2 = pIn[p2];

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

// biến đổi Hadamard chuyển tiếp 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 lùi
```

```
const int iy0 = z0 + z2;
```

```
const int iy1 = z1 + z3;
```

```
const int iy2 = z0 - z2;
```

```
const int iy3 = z1 - z3;
```

```
// đưa ra điểm ảnh được lọc
```

```
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, thì đoạn mã giả sau đây mô tả quy trình lọc của trường hợp này:

```
// quét các điểm ảnh được xây dựng lại/được dự đoán
```

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

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

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

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

// biến đổi Hadamard chuyển tiếp 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 lùi

```
const int iy0 = z0 + z2;
```

```
const int iy1 = z1 + z3;
```

```
const int iy2 = z0 - z2;
```

```
const int iy3 = z1 - z3;
```

Theo phương án thay thế, bộ đệm tích lũy cần được khởi tạo bởi các giá trị điểm ảnh chưa được lọc được nhân với số lượng các giá trị điểm ảnh tối đa sẽ được thêm vào trong khối. Số lượng các giá trị điểm ảnh tối đa được thêm vào trong khối được xác định dựa trên mẫu quét. Mẫu quét thực tế xác định lượng các giá trị điểm ảnh được bổ sung cho mỗi vị trí. Được dựa trên, số lớn nhất từ tất cả các vị trí trong khối có thể được lựa chọn và được sử dụng trong suốt quá trình khởi tạo bộ đệm tích lũy. Sau đó, trong mỗi

bước tích lũy, giá trị điểm ảnh chưa được lọc bị trừ từ giá trị được lọc tương ứng và được bổ sung vào bộ đệm tích lũy:

// việc tích lũy các điểm ảnh được lọc

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

pOut[p1] += iy0 - iy1

pOut[p2] += iy2 + iy3

pOut[p3] += iy2 - iy3

Để giảm độ sâu bit của các giá trị điểm ảnh được tích lũy trước khi đặt vào trong bộ đệm tích lũy kết quả của biến đổi lùì 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 đối với kích thước biến đổi là 4.

Phương án này cho phép hệ thống tránh lưu trữ số các điểm ảnh được bổ sung vào vị trí hiện tại và cho phép việc thay thế phép chia và phép nhân bằng cách chuyển ở bước chuẩn hóa cuối cùng và bước khởi tạo bộ đệm tích lũy tương ứng nếu số lượng các giá trị điểm ảnh tối đa được bổ sung vào nguồn điện là ví dụ: 2, 4, 8 v.v.

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

// việc chuẩn hóa

pFiltered[p0] = CLIP3(0, (1 << BIT_DEPTH) - 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 trong bộ đệm tích lũy, ví dụ: đối với lượng 4 HTDF_CNT_SCALE là bằng 2, và

HTDF_CNT_SCALE_RND bằng $(1 \ll (\text{HTDF_CNT_SCALE} - 1))$. CLIP3 là hàm số cắt mà đảm bảo mẫu được lọc trong phạm vi cho phép giữa giá trị mẫu nhỏ nhất và lớn nhất.

Như được đề cập ở trên để tránh sự thay đổi độ sáng trung bình của khối được lọc có thể có lợi nếu bỏ qua lọc thành phần phổ thứ nhất (tương ứng với giá trị DC). Điều đó cho phép đơn giản hóa việc triển khai 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 trong khối được xây dựng lại hoặc khối được dự đoán, mẫu quét được chọn được dựa trên vị trí của điểm ảnh lọc trong khối được xây dựng lại hoặc khối được dự đoán đối với các bước 302 và 310. Mẫu quét được chọn để đảm bảo tất cả các điểm ảnh trong CU được xây dựng lại hoặc CU được dự đoán và đặt gần để xử lý điểm ảnh. Thứ tự quét bất kỳ có thể được sử dụng trong mẫu. Ví dụ, mẫu quét định trước được định nghĩa là tập hợp các độ lệch không gian hoặc mảnh liên quan tới vị trí của điểm ảnh hiện tại trong khối được xây dựng lại hoặc khối được dự đoán, trong đó các độ lệch chỉ tới các điểm ảnh lân cận trong khối được xây dựng lại hoặc khối được dự đoán. Dưới đây là ví dụ về mẫu quét:

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

Fig. 4A minh họa ví dụ của các mẫu cho vị trí điểm ảnh khác nhau trong khối hình vuông (Ví dụ, khối CU vuông được dự đoán, hoặc khối CU vuông xây dựng lại). Theo hình này các điểm ảnh ranh giới có thể được lọc được dựa trên 4 biến đổi điểm và các điểm ảnh trung tâm có thể được lọc được dựa trên 8 biến đổi điểm.

Đối với các khối được xây dựng lại hoặc các khối được dự đoán hình chữ nhật, trong đó kích thước của một cạnh lớn hơn kích thước của cạnh khác, quá trình quét nên được thực hiện dọc theo chiều dài. Ví dụ đối với khối chữ nhật nằm ngang thứ tự quét

sau có thể được sử dụng

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

trong đó mỗi cặp (y, x) x là độ lệch ngang và y là độ lệch dọc liên quan tới vị trí của điểm ảnh lọc trong khối được xây dựng lại hoặc khối được dự đoán lọc.

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

- Đối với các khối được xây dựng lại hoặc các khối được dự đoán với tín hiệu dư khác 0;
- phụ thuộc vào kích thước khối ví dụ: đối với khối được xây dựng lại hoặc khối được dự đoán nhỏ (kích thước tối thiểu ít hơn ngưỡng);
- phụ thuộc vào tỷ lệ co của khối được xây dựng lại hoặc khối được dự đoán;
- phụ thuộc vào chế độ dự đoán (nội hoặc liên) của khối được xây dựng lại hoặc khối được dự đoán, ví dụ:; bằng cách áp dụng bộ lọc chỉ cho các khối được liên dự đoán; hoặc
- đối với bất kỳ sự kết hợp nào của các điều kiện được mô tả ở trên.

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

Ví dụ khác bộ lọc chỉ được áp dụng chỉ các khối mà có tín hiệu dư khác 0. Điều này có lợi nếu việc lượng tử hóa hoặc các phần dư được sử dụng khi bộ lọc nhằm để cải thiện lỗi lượng tử hóa. Nếu không còn dư thì có thể là một dấu hiệu cho thấy dự đoán là tốt và không cần lọc thêm.

Ví dụ khác, do nội dự đoán thường kém hơn liên dự đoán, bộ lọc có thể được áp dụng cho các khối nội dự đoán độc lập với sự hiện diện của phần dư khác 0 và được áp dụng chỉ cho các khối liên dự đoán nếu khối có tín hiệu dư khác 0.

Tham số lọc sigma và mẫu quét có thể phụ thuộc nhiều 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

khối hiện tại đối với mẫu quét mẫu $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. Đối với việc lọc điểm ảnh hiện tại vùng hình vuông 3×3 các điểm ảnh được sử dụng (điểm ảnh hiện tại được đánh dấu bằng màu xám đậm ở giữa hình vuông 3×3). Điểm ảnh được lọc thu được bằng cách kết hợp các mẫu được lọc miền biên đổi từ bốn nhóm 2×2 . Cần được hiểu rằng nếu điểm ảnh hiện tại được đặt trong đường viền khối (ví dụ: đường viền trên cùng) các nhóm trên cùng bên trái và trên cùng bên phải là không khả dụng và chỉ hai nhóm 2×2 (dưới cùng bên trái và dưới cùng bên phải) có thể được sử dụng để lọc. Hơn nữa nếu điểm ảnh hiện tại được đặt trong góc khối (ví dụ: góc trên cùng bên trái) chỉ một nhóm 2×2 (dưới cùng bên phải) có thể được sử dụng để lọc.

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

Để thống nhất hơn nữa quy trình lọc cho tất cả các điểm ảnh trong khối (bốn nhóm 2×2 được sử dụng để lọc tất cả các điểm ảnh trong khối hiện tại), ngoài phần đệm khối hiện tại trên cùng bên trái còn mở rộng bởi phần đệm dưới cùng bên phải như được minh họa trên Fig. 4D. Việc hợp nhất lọc có lợi do đơn giản hóa việc triển khai bằng cách loại trừ các trường hợp xử lý đặc biệt đối với các điểm ảnh góc và/hoặc các điểm ảnh đường viền.

Các mẫu đệm tốt hơn là được lấy từ các mẫu lân cận đã được điều chỉnh từ các khối được xây dựng lại. Trong các bộ mã hóa/giải mã video hiện tại, các khối được xây dựng lại có thể được đặt ở phía trên bên trái hoặc phía dưới bên phải phụ thuộc vào thứ tự việc xây dựng lại khối. Sử dụng thêm thông tin từ các mẫu điều chỉnh, nó cải thiện chất lượng lọc và giúp quá trình chuyển đổi giữa các khối diễn ra suôn sẻ hơn.

Truy xuất các mẫu được xây dựng lại từ các khối đã điều chỉnh hoặc các khối được xây dựng lại trước đó có thể yêu cầu tải bộ nhớ bổ sung để triển khai phần cứng hoặc phần mềm. Để giảm thiểu hoặc loại trừ bộ nhớ bổ sung, sẽ có lợi khi sử dụng các mẫu dành cho nội dự đoán của khối hiện tại mà thường được lấy từ một, hai hoặc nhiều hàng và cột từ các khối lân cận được điều chỉnh theo đường viền khối hiện tại. Các mẫu này thường được lưu trữ trong bộ nhớ nhanh (còn được gọi là bộ đệm “dòng”) để dễ

dàng truy cập cho nội dự đoán và được gọi là mẫu tham chiếu của nội dự đoán.

Cần lưu ý thêm rằng trong một số cách triển khai, trước khi thực hiện nội dự đoán, mẫu tham chiếu (các mẫu nội tham chiếu) được xử lý trước trước khi dự đoán, ví dụ: bằng cách làm mịn, làm sắc nét, khử chuông hoặc lọc hai bên. Trong trường hợp này, có thể có lợi khi sử dụng các mẫu đã xử lý trước để đệm cho khối hiện tại.

Nếu một số mẫu trong vùng đệm không có sẵn, do thứ tự việc xây dựng 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 đệm như được minh họa trên Fig. 4D.

Fig. 5A là lưu đồ minh họa các bước của phương pháp lọc trong vòng tương ứng 500 theo phương án. Khối được xây dựng lại bao gồm nhiều các điểm ảnh. Phương pháp 500 bao gồm các bước sau: tải (502) điểm ảnh hiện tại và các điểm ảnh lân cận của nó tới bộ đệm tuyến tính theo mẫu quét định trước; thu (504) các thành phần phổ bằng cách thực hiện biến đổi 1D cho các điểm ảnh trong bộ đệm tuyến tính; thu (506) các thành phần phổ đã được lọc bằng cách nhân mỗi thành phần phổ với hệ số khuếch đại, trong đó hệ số khuếch đại phụ thuộc vào thành phần phổ tương ứng và tham số lọc; thu (508) các điểm ảnh được lọc bằng cách thực hiện phép biến đổi 1D ngược cho các thành phần phổ đã được lọc; và tạo (510) khối được xây dựng lại được lọc được dựa trên các điểm ảnh được lọc ước tính trên các bước xử lý trước đó. Phương pháp 500 có thể được thực hiện bởi thiết bị mã hóa như được thể hiện trên Fig.1 và thiết bị giải mã khi được thể hiện trên Fig.2. Thông tin chi tiết 300 của Fig.3A hoặc thông tin 300' của Fig.3B cũng được áp dụng vào phương pháp 500 như được thể hiện trên Fig.5A.

Tương tự Fig. 5A, Fig.5B là lưu đồ minh họa các bước của phương pháp lọc trong vòng tương ứng 500' theo phương án khác. Ở ví dụ này, khối (hoặc khung) là khối được dự đoán, và khối được lọc là khối được dự đoán được lọc. Phần mô tả chi tiết của Fig. 5B tương tự với Fig. 5A.

Việc triển khai phần cứng được thảo luận.

Bộ lọc miền biến đổi Hadamard được đặt ngay sau việc xây dựng lại khối và các mẫu xử lý có thể tham gia vào việc xây dựng lại các khối tiếp theo đặc biệt như các mẫu tham chiếu nội dự đoán. Do đó, thời gian chờ bởi bộ lọc cần được giảm thiểu để đảm

bảo toàn bộ đường xây dựng lại không bị ảnh hưởng nhiều.

Biến đổi Hadamard được coi là tương đối đơn giản để thực hiện phần cứng. Chỉ các phép cộng được yêu cầu để thực hiện trừ phép nhân. Như có thể thấy bên dưới Mã giả 1, phép biến đổi tiến và lùi gồm 4 phép cộng có thể được thực hiện song song hoặc bằng cách 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 = RdTbl1(t1, tbl1, thr);
        const int z2 = RdTbl1(t2, tbl1, thr);
        const int z3 = RdTbl1(t3, tbl1, 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 tiến và lùi có thể được thực thi trong phần cứng bằng cách sử dụng logic tổ hợp. Cần chú ý nhiều hơn để truy nhập nhanh và song song tới LUT.

SRAM được dựa trên LUT được thảo luận.

Trong triển khai ví dụ này, LUT được lưu trữ trong RAM tĩnh đơn cổng trên chip

(Fig. 6).

Khi dữ liệu từ bước xử lý trước đó có sẵn trong bộ đệm bởi sườn lên của xung đồng hồ được truy cập bởi logic tổ hợp thực hiện biến đổi Hadamard tiến (chứa hai phép cộng tuần tự). Sau khi hoàn thành địa chỉ logic tổ hợp là có sẵn cho mỗi LUT. Sử dụng bộ đảo và sườn xuống của xung đồng hồ, dữ liệu được truy cập từ SRAM. Logic tổ hợp thứ hai thực hiện biến đổi Hadamard lùi và việc chuẩn hóa được bắt đầu ngay sau dữ liệu từ LUT là có sẵn. Đầu ra các mẫu được lọc trở nên có sẵn ở cuối chu kỳ định thời hiện tại và sẵn sàng để xử lý bởi thuật toán tiếp theo tại sườn lên tiếp theo của xung đồng hồ.

LUT dựa trên flip-flop (mạch lật) được thảo luận.

Xem xét một bảng cho quá trình lọc được giới hạn bởi 16 các mục nhập có vẻ hiệu quả hơn để triển khai LUT được dựa trên các flip-flop (mạch lật). Thiết kế như vậy không yêu cầu một số LUT để xử lý song song và cạnh xung đồng hồ để truy cập dữ liệu. Truy nhập song song được cung cấp bởi bộ ghép kênh như được mô tả trong Fig. 7 minh họa thiết kế ví dụ để xử lý một nhóm 2x2. Trong thiết kế đề xuất 16 các flip-flop (mạch lật) của 7 bit được yêu cầu để cung cấp truy nhập song song trong suốt quy trình lọc. LUT cụ thể của QP có thể được tải vào flip-flop khi QP cho CU hiện tại khả dụng.

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

Với phân tích ở trên cho thấy rằng bộ lọc được đề xuất có thể được thực thi trong phần cứng trong một xung đồng hồ sử dụng hoặc SRAM hoặc flip-flop (mạch lật) dựa trên triển khai LUT.

Phân tích độ phức tạp được thảo luận.

Phép đo tác động lên Tốc độ bit/PSNR liên quan tới (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 các bảng dưới đây).

Thử nghiệm #	hình dạng bộ lọc	Độ phức tạp tính	Độ chính	Song song	Số vòng	Yêu cầu	Cách tính	Kích thước
-----------------	------------------	------------------	----------	-----------	---------	---------	-----------	------------

		toán (# của các phép nhân/ Các phép cộng/phép dịch/kiểm tra*) cho mỗi mẫu	xác của phép nhân	(mỗi mẫu có thể được lọc độc lập với các mẫu khác nhau)	xung đồng hồ	bộ nhớ	hệ số bộ lọc	CU tối thiểu để áp dụng cho bộ lọc
14.3a	3x3	0/20+4 1- thêm bit cho vòng/5/6	n/a	đúng	< 1 xung đồng hồ	140 byte (32 giá trị 7 bit cho mỗi nhóm qp)	LUT được tính toán trước	4x8, 8x4
14.3b	3x3	0/20+4 1- thêm bit cho vòng /5/6	n/a	đúng	< 1 xung đồng hồ	70 byte (16 các giá trị 7 bit cho mỗi nhóm qp)	Được tính toán trước trong LUT	4x8, 8x4

* max/min/abs được đếm khi kiểm tra

Bảng 1. Tổng hợp phân tích độ phức tạp CE14-3

Kết quả thí nghiệm được thảo luận.

Các kết quả khách quan được thảo luận.

Mục tiêu hiệu suất được hiển thị trong các bảng sau:

Tất cả phân nội cơ sở 10					
Trên BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Lớp A1	-0,19%	0,29%	0,24%	110%	115%
Lớp A2	-0,25%	0,21%	0,22%	109%	112%
Lớp B	-0,42%	0,29%	0,22%	109%	113%
Lớp C	-0,83%	0,23%	0,42%	110%	110%
Lớp E	-0,54%	0,28%	0,19%	109%	114%

Tổng	-0,48%	0,28%	0,21%	109%	112%
Lớp D	-0,61%	0,47%	0,01%	109%	108%
Lớp F (tùy chọn)	-0,82%	0,07%	0,04%	108%	107%

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

	Độ trễ thấp B Cơ sở 10				
	Trên BMS-2.0,1 với cấu hình VTM				
	Y	U	V	EncT	DecT
Lớp A1					
Lớp A2					
Lớp B	-0,58%	0,52%	0,54%	104%	104%
Lớp C	-0,74%	0,23%	0,52%	104%	104%
Lớp E	-0,75%	0,21%	1,09%	102%	101%
Tổng	-0,68%	0,40%	0,67%	104%	103%
Lớp D	-0,90%	0,75%	0,28%	104%	104%
Lớp F (tùy chọn)	-0,73%	0,20%	-0,09%	103%	102%

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

	Tất cả nội cơ sở 10				
	Trên BMS-2.0.1 với cấu hình VTM				
	Y	U	V	EncT	DecT
Lớp A1	-0,17%	0,28%	0,28%	110%	115%
Lớp A2	-0,23%	0,27%	0,29%	109%	112%
Lớp B	-0,40%	0,21%	0,29%	109%	113%
Lớp C	-0,82%	0,20%	0,22%	109%	110%
Lớp E	-0,54%	0,24%	0,25%	109%	113%
Tổng	-0,47%	0,28%	0,22%	109%	112%
Lớp D	-0,65%	0,43%	0,22%	109%	108%
Lớp F (tùy chọn)	-0,81%	0,01%	0,12%	108%	107%

Truy cập ngẫu nhiên cơ sở 10

Trên BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Lớp A1	-0,41%	-0,26%	-0,24%	105%	105%
Lớp A2	-0,92%	-0,49%	-0,27%	105%	103%
Lớp B	-0,69%	-0,12%	0,12%	106%	104%
Lớp C	-0,75%	0,14%	0,10%	105%	104%
Lớp E					
Tổng	-0,70%	-0,17%	-0,06%	105%	104%
Lớp D	-0,69%	-0,20%	0,49%	104%	103%
Lớp F (tùy chọn)	-0,82%	0,11%	0,11%	104%	102%

Độ trễ thấp B Cơ sở 10					
Trên BMS-2.0.1 với cấu hình VTM					
	Y	U	V	EncT	DecT
Lớp A1					
Lớp A2					
Lớp B	-0,57%	0,49%	0,69%	104%	104%
Lớp C	-0,76%	0,21%	0,42%	104%	106%
Lớp E	-0,67%	-0,66%	-0,48%	102%	102%
Tổng	-0,66%	0,14%	0,21%	103%	104%
Lớp D	-0,88%	0,92%	0,21%	103%	105%
Lớp F (tùy chọn)	-0,74%	0,28%	-0,15%	103%	102%

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

LUT 70 byte được đề xuất với 16 byte trên mỗi QP, cho phép triển khai HW 1 xung đồng hồ. V.v.. Đề xuất áp dụng bộ lọc miền chuyển đổi Hadamard cho phiên bản tiếp theo của VTM.

Các tài liệu tham khảo sau đây được đưa vào đây với tư cách là tài liệu tham khảo nếu được sao chép toàn bộ:

Tài liệu JVET-K0068 của Nhóm chuyên gia video chung (JVET, Joint Video Experts Team).

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

Như một ví dụ 1, tập các tham số lượng tử hóa (việc lượng tử hóa các tham số, QPs) được chọn để tạo bảng tra cứu (LUT), trong đó tập các QP gồm QP thứ nhất tương ứng với chỉ số (i) và QP thứ hai tương ứng với chỉ số (i+1), và QP thứ nhất và QP thứ hai có một khoảng thời gian không đổi. Ví dụ, khoảng thời gian có thể bằng 8, 10 hoặc

16.

Ví dụ, lấy khoảng không đổi là 8 ví dụ, bảng thừa bằng cách có LUT cho $qp = \{20, 28, 36, 44, 52\}$. Khoảng cách giữa gp thứ nhất 20 và gp thứ hai 28 là 8. Tương tự, khoảng cách giữa gp thứ hai 28 và gp thứ ba 36 là 8. Trong quá trình lọc bảng có QP gần nhất được chọn.

Ví dụ khác, lấy khoảng không đổi là 8 làm ví dụ, bảng thừa bằng cách có LUT cho $qp = \{18, 26, 34, 42, 50\}$. Khoảng cách giữa gp thứ nhất 18 và gp thứ hai 26 là 8. Tương tự, khoảng cách giữa gp thứ hai 26 và gp thứ ba 34 là 8. Trong quá trình lọc bảng có QP gần nhất được chọn.

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

Dưới đây là mã giả 2 phản ánh các QP được chọn để tạo 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 đại diện cho khoảng thời gian không đổi. Có khoảng thời gian là lũy thừa của hai là thuận lợi cho phép thực hiện phép toán chia để tính chỉ số khi dịch chuyển. Cần lưu ý rằng các giá trị khác nhau của khoảng thời gian không đổi có thể được chọn ví dụ: 2, 4, 10, 15 hoặc 16 v.v.. Hơn nữa, theo phương án thay thế khoảng thời gian có thể là tùy ý và LUT được tính cho tập QP tùy ý.

Trong suốt quy trình lọc chỉ số tương ứng LUT cho QP nhất định được tính như sau:

```

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

```

hoặc cách khác với độ chính xác kém hơn:

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

Nếu khoảng thời gian không đổi là lũy thừa của 2, ví dụ: sau đó chỉ số của LUT có thể được tính toán thuận lợi sử dụng hoạt động dịch chuyển thay vì phân chia:

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

Ví dụ khác 2, bộ giải mã hoặc bộ mã hóa loại bỏ N bit từ các giá trị bảng, và N là số nguyên. Điều đó cho phép có biểu diễn LUT thưa chỉ lưu trữ các giá trị được chọn từ dải ô đã cho. Ví dụ, N là 3. Giá trị bảng lớn nhất là 127 (7 bit), loại bỏ 3 bit, kết quả là 4 bit, mà 16 mục nhập của giá trị 7 bit – khoảng 16 byte.

Dưới đây là mã giả 3 mô tả làm thế nào LUT được tạo ra theo giá trị qp đã cho.

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;
    }
}
```

Trong ví dụ cụ thể HTDF_TBL_SH xác định số bit để loại bỏ mà có thể là 1, 2, 3, 4 v.v.

Dưới đây là mã giả minh họa truy cập vào LUT thừa trong quá trình lọc:

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

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

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

Trong một số triển khai việc giữ kích thước LUT bị giới hạn và có THR để giữ cho phương trình từ đoạn [0077] không bị mâu thuẫn. Ở các giá trị QP cao (dẫn đến giá trị σ cao) việc giữ kích thước LUT bị giới hạn bởi ví dụ: 16 mục nhập có thể dẫn đến khoảng trống không mong muốn trong hàm truyền bộ lọc (mà được biểu diễn bởi LUT) quanh giá trị 120 như được mô tả trên Fig. 10 (cũng bao gồm phương pháp lấy mẫu phụ LUT bằng cách loại bỏ 3 bit quan trọng nhất).

Fig. 11 minh họa cho cùng một bảng với các mục nhập của bảng được vẽ từng cái một mà không xem xét đến hiệu ứng lấy mẫu phụ LUT minh họa khoảng cách giữa mục nhập LUT cuối cùng ở chỉ số 15 và điểm tiếp theo trong hàm truyền bộ lọc ở chỉ số 16.

Fig. 12 minh họa phương pháp làm thế nào để loại bỏ khoảng cách bằng cách sử

dùng hàm phụ đi qua giá trị THR tại đối số tương ứng với mục nhập LUT cuối cùng + 1, ví dụ: bằng cách sử dụng phương trình đường thẳng (được chỉ bởi màu xanh lá), đi qua điểm $THR = 128$ tại đối số 16 tương ứng với đối số của mục nhập LUT cuối cùng (là 15) tăng 1 hoặc cách khác kích thước LUT (là 16) và giao nhau trục x tại một số điểm (ví dụ ở giá trị 11 như trong ví dụ đã cho). Cần lưu ý rằng các loại hàm phụ khác có thể được sử dụng dựa trên nguyên tắc tương tự, ví dụ: hàm mũ, logarit, parabol, v.v. hoặc kết hợp của chúng.

Fig. 13 minh họa ví dụ về loại bỏ khoảng cách bằng cách lấy giá trị lớn nhất từ hai giá trị trong khi tạo LUT, trong đó giá trị đầu tiên là mục nhập LUT như mô tả ở trên $LUT(R_i, \sigma) = \frac{R_i^3}{R_i^2 + m * \sigma^2}$; và giá trị thứ hai là giá trị của AuxiliaryFunction (hàm phụ đường thẳng trong ví dụ này) tại cùng một 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 hàm phụ, và hàm phụ có giá trị bằng THR tại i tương ứng với đối số của LUT cuối cùng + 1.

Fig. 14 minh họa hàm truyền bộ lọc sau khi áp dụng phương pháp sử dụng được mô tả ở trên phương trình đường thẳng phụ và mẫu phụ LUT bằng cách loại bỏ 3 bit có ý nghĩa nhỏ nhất.

Như được mô tả trong đoạn [00144] một LUT có thể được sử dụng cho nhóm của các QP. Để bao phủ phạm vi QP khả dụng tập QP xác định trước được sử dụng và một LUT được tạo cho mỗi QP từ tập hợp. Fig.15 minh họa ví dụ các hàm truyền bộ lọc phụ thuộc vào năm QP trong tập hợp và các bảng tương ứng (từ 0 tới 4). Trong ví dụ này bảng 4 được tạo sử dụng phương pháp được mô tả trong các đoạn [00156]-[00157] với hàm phụ đường thẳng cắt qua trục x tại giá trị 11; và bảng 3 được tạo sử dụng cùng phương pháp với hàm phụ đường thẳng cắt qua trục x tại giá trị 9. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

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

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

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

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

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

Như được mô tả trong các đoạn [00150]-[00153] phương pháp của bảng mẫu phụ có thể loại bỏ N bit từ các giá trị bảng để giảm kích thước bảng. Như đã được đề cập trong các đoạn này N có thể khác nhau phụ thuộc vào QP được sử dụng để tạo bảng hiện tại 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 các QP cao hơn. Do đó, giá trị tuyệt đối của THR có thể được giảm xuống mà không làm giảm hiệu suất. Hơn nữa để kích thước bảng giống như đối với tất cả các QP trong tập hợp (điều này có lợi để đơn giản hóa việc triển khai) và giảm tốc độ mẫu phụ cho các QP thấp hơn (mà tương ứng với mức độ nén ít hơn và chất lượng ảnh xây dựng lại tốt hơn) có thể có lợi khi giảm lượng bit bị loại bỏ N so với các bảng QP khác, ví dụ: bằng cách thiết lập N cho QP thấp hơn bằng 2 và thiết lập THR bằng 64. Fig. 16 minh họa ví dụ các hàm truyền bộ lọc cho năm QP trong tập hợp được dựa trên các bảng tương ứng (từ 0 tới 4) với N được đặt là 2 cho bảng thứ nhất (tương ứng với khoảng QP thấp hơn) và N được đặt bằng 3 đối với các bảng khác. Ví dụ này cũng gồm phương pháp sử dụng *AuxiliaryFunction* cho việc tạo bảng 3 và bảng 4 như được mô tả trong các đoạn [00159]-[00160]. Đối với bảng 4 hàm phụ đường thẳng cắt trục x tại giá trị 11. Đối với bảng 4 hàm phụ đường thẳng cắt trục x tại giá trị 9. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

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

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

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

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

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

Như được mô tả trong các đoạn [00150]-[00153] phương pháp của bảng mẫu phụ có thể loại bỏ N bit từ các giá trị bảng để giảm kích thước bảng. Như đã được đề cập trong các đoạn này N có thể khác nhau phụ thuộc vào QP được sử dụng để tạo bảng hiện tại 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 các QP thấp hơn mà có thể yêu cầu tăng giá trị THR để giữ phương trình từ đoạn [0077] gần bằng 1. Đồng thời để giữ nguyên kích thước LUT giống như đối với tất cả các QP trong tập hợp (điều này có lợi do đơn giản hóa việc triển khai) và cũng xem xét rằng đối với các giá trị QP cao hơn hình ảnh được xây dựng lại có nhiều biến dạng hơn và tăng mẫu phụ của LUT được cho phép do tác động chủ quan không thể nhận thấy của mẫu phụ LUT hiện có khi nén mạnh giá trị N của các bit ít quan trọng nhất được loại bỏ có thể được tăng thành 4 ví dụ: bảng cuối cùng và bảng cuối cùng thứ hai trong tập hợp. Fig. 17 minh họa ví dụ các hàm truyền bộ lọc cho năm QP trong tập hợp được dựa trên các bảng tương ứng (từ 0 đến 4) với N được đặt là 2 cho bảng thứ nhất (bảng 0 tương ứng với khoảng QP thấp hơn), N đặt là 4 cho bảng cuối cùng và bảng cuối cùng thứ hai (bảng 3 và bảng 4) và N được đặt là 3 đối với các bảng khác. Trong ví dụ này TRH được đặt là 64 cho việc tạo bảng thứ nhất, được đặt là 256 cho bảng cuối cùng và bảng cuối cùng thứ hai và đặt là 128 cho các bảng còn lại. Ví dụ này cũng gồm phương pháp sử dụng *AuxiliaryFunction* đối với việc tạo bảng 3 và bảng 4 như được mô tả trong các đoạn [00159]-[00160]. Đối với bảng 4 hàm phụ đường thẳng cắt trục x tại giá trị 6. Đối với bảng 4 hàm phụ đường thẳng cắt trục x tại giá trị 8. Các bảng được sử dụng trong ví dụ này có các giá trị theo:

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

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

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

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, },

Fig. 18 là sơ đồ khối của thiết bị 600 mà có thể được sử dụng để thực hiện các phương án khác nhau. Thiết bị 600 có thể là thiết bị mã hóa như được thể hiện trên Fig. 1 và thiết bị giải mã như được thể hiện trên Fig. 2. Ngoài ra, thiết bị 600 có thể chứa một hoặc nhiều các phần tử được mô tả. Trong một số các phương án, thiết bị 600 được trang bị một hoặc nhiều thiết bị đầu vào/đầu ra, như là loa, micrô, chuột, màn hình chạm, bàn phím, bàn phím phụ, máy in, màn hiển thị, và tương tự. Thiết bị 600 có thể gồm một hoặc nhiều đơn vị xử lý trung tâm (CPUs, central processing units) 610, bộ nhớ 620,

khối lưu trữ 630, bộ điều hợp video 640, và giao diện I/O 660 được kết nối với buýt. Buýt là một hoặc nhiều loại kiến trúc buýt bất kỳ gồm buýt bộ nhớ hoặc bộ điều khiển bộ nhớ, buýt ngoại vi, buýt video, hoặc tương tự.

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

Bộ điều hợp 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 thiết bị 600. Ví dụ: thiết bị 600 có thể cung cấp giao diện lệnh SQL cho máy khách. Như được minh họa, các ví dụ về thiết bị đầu vào và đầu ra gồm màn hiển thị 690 được kết hợp với bộ điều hợp video 640 và bất kỳ sự kết hợp nào của chuột / bàn phím / máy in 670 được kết hợp với giao diện I/O 660. Các thiết bị khác có thể được kết hợp với thiết bị 600 và bổ sung hoặc một ít thẻ giao diện hơn có thể được sử dụng. Ví dụ: thẻ giao diện nối tiếp (không hiển thị) có thể được sử dụng để cung cấp giao diện nối tiếp cho máy in.

Thiết bị 600 cũng bao gồm một hoặc nhiều giao diện mạng 650, bao gồm các liên kết có dây, chẳng hạn như cáp Ethernet hoặc loại 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 thiết bị 600 để truyền thông với các đơn vị ở xa qua mạng 680. Ví dụ, giao diện mạng 650 có thể đề xuất truyền thông với cơ sở dữ liệu. Theo một phương án, thiết bị 600 được kết hợp với mạng cục bộ hoặc mạng diện rộng để xử lý dữ liệu và truyền thông với các thiết bị từ xa, chẳng hạn như các thiết bị xử lý khác, Internet, phương tiện lưu

trừ từ xa hoặc tương tự.

Thiết kế đề xuất của bộ lọc trong vòng hoặc bộ lọc dự đoán có những ưu điểm sau so với các phương pháp lọc thích ứng thông thường như ALF:

- Bộ lọc miền tần số được đề xuất lấy các tham số lọc (hệ số khuếch đại miền tần số) từ khung được tái tạo hoặc khôi phục dự đoán ở phía bộ giải mã và do đó các thông số lọc không cần thiết phải chuyển từ bộ mã hóa sang phía bộ giải mã.
- ALF yêu cầu tối ưu hóa biến dạng tốc độ phức tạp (RDO) ở phía bộ mã hóa để giảm số lượng hệ số trọng số cho quá trình truyền. Phương pháp được đề xuất không yêu cầu RDO phức tạp ở phía bộ mã hóa (không truyền tham số) và được áp dụng cho tất cả các khối thỏa mãn các điều kiện xác định trước
- ALF là bộ lọc tuyến tính trong miền điểm ảnh. Bộ lọc được đề xuất là phi tuyến tính vì hệ số khuếch đại cho mỗi thành phần phổ 1D phụ thuộc vào giá trị thành phần phổ này. Nó cho phép đạt được độ lợi mã hóa bổ sung từ quá trình xử lý phi tuyến tính.
- ALF yêu cầu số nhân phổ quát ở phía bộ giải mã. Trong phương pháp đề xuất phương pháp lọc có thể được thực hiện dưới dạng bảng tra cứu, vì độ lợi cho mỗi hệ số phổ nhỏ hơn một. Do đó phương pháp đề xuất có thể được thực hiện mà không cần bất kỳ phép nhân nào.

Do đó, bộ lọc được cung cấp để cho phép cải thiện hiệu quả tạo mã video với độ phức tạp thấp.

Mặc dù tính năng hoặc khía cạnh cụ thể của sáng chế có thể đã được tiết lộ chỉ liên quan đến một trong số một số cách triển khai hoặc phương án, nhưng tính năng hoặc khía cạnh đó có thể được kết hợp với một hoặc nhiều tính năng hoặc khía cạnh khác của các triển khai hoặc phương án khác nếu có thể được mong muốn và có lợi cho bất kỳ ứng dụng cụ thể nào. Hơn nữa, trong phạm vi mà các thuật ngữ "bao gồm", "có", "với" hoặc các biến thể khác của chúng được sử dụng trong mô tả chi tiết hoặc các điểm yêu cầu bảo hộ, các thuật ngữ đó nhằm bao hàm theo cách tương tự như thuật ngữ "bao gồm". Ngoài ra, các thuật ngữ "ví dụ", "ví dụ" và "ví dụ:" chỉ có nghĩa là một ví dụ, chứ không phải là tốt nhất hoặc tối ưu. Các thuật ngữ "kết hợp" và "kết nối", cùng với các

dẫn xuất có thể đã được sử dụng. Cần hiểu rằng các thuật ngữ này có thể đã được sử dụng để chỉ ra rằng hai phần tử hợp tác hoặc tương tác với nhau bất kể chúng tiếp xúc trực tiếp vật lý hoặc điện, hoặc chúng không tiếp xúc trực tiếp với nhau.

Mặc dù các khía cạnh cụ thể đã được minh họa và mô tả ở đây, nhưng những người có hiểu biết trung bình trong lĩnh vực này sẽ đánh giá cao rằng nhiều cách triển khai thay thế và/hoặc tương đương có thể được thay thế cho các khía cạnh cụ thể được hiển thị và mô tả mà không vượt quá phạm vi của tiết lộ hiện tại. Sáng chế này nhằm mục đích bao gồm bất kỳ sự điều chỉnh hoặc biến thể nào của các khía cạnh cụ thể được thảo luận ở đây.

Mặc dù các phần tử trong các yêu cầu bảo hộ sau đây được nhắc lại theo một trình tự cụ thể với nhãn tương ứng, trừ khi yêu cầu bảo hộ nhắc lại ngụ ý khác về một trình tự cụ thể để triển khai một số hoặc tất cả các phần tử đó, nhưng các phần tử đó không nhất thiết phải được giới hạn để thực hiện trong trình tự cụ thể đó.

Nhiều lựa chọn thay thế, sửa đổi và biến thể sẽ hiển nhiên đối với những người có hiểu biết trung bình trong lĩnh vực nghệ thuật dựa trên những lời dạy trên. Tất nhiên, những người có kỹ năng trong lĩnh vực nghệ thuật này dễ dàng nhận ra rằng có rất nhiều ứng dụng của sáng chế ngoài những ứng dụng được mô tả ở đây. Mặc dù sáng chế được mô tả có tham chiếu đến một hoặc nhiều phương án cụ thể, nhưng những người có hiểu biết trung bình trong lĩnh vực này nhận ra rằng nhiều thay đổi có thể được thực hiện theo đó mà không nằm ngoài phạm vi của sáng chế. Do đó, cần phải hiểu rằng trong phạm vi của các yêu cầu bảo hộ được bổ sung và các điểm tương đương của chúng, sáng chế có thể được thực hành khác với những gì được mô tả cụ thể ở đây.

YÊU CẦU BẢO HỘ

1. Phương pháp để xử lý khối, trong đó khối bao gồm nhiều điểm ảnh, trong đó phương pháp này 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 định trước, trong đó khối là khối được xây dựng lại hoặc khối được dự đoán;

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

khi giá trị tuyệt đối của ít nhất một thành phần phổ trong số các thành phần phổ nhỏ hơn ngưỡng, thu các thành phần phổ đã được lọc cho mỗi thành phần phổ trong số ít nhất một thành phần phổ trong số các thành phần phổ dựa trên tham số lọc và thành phần phổ tương ứng;

thu các điểm ảnh được lọc bằng cách thực hiện biến đổi ngược cho các thành phần phổ đã được lọc; và

tạo khối được lọc dựa trên các điểm ảnh được lọc.

2. Phương pháp theo điểm 1, trong đó bước thu các thành phần phổ đã được lọc bằng cách sử dụng bảng tra cứu (look up table, LUT).

3. Phương pháp theo điểm 2, thành phần phổ đã được lọc $F(i, \sigma)$ được suy 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 thành phần 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.

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 số lượng thành phần phổ.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 4, trong đó tham số lọc được suy ra dựa trên tham số lượng tử hóa (quantization parameter, QP) bộ mã hóa/giải mã (codec).

6. Phương pháp theo điểm 5, trong đó tham số lọc σ được suy 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.

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

8. Phương pháp theo điểm 6 hoặc 7, trong đó k được sửa đổi dựa trên độ sâu bit của các điểm ảnh $k_{mod} = k * (1 \ll (\text{bit_depth} - 8))$.

9. Phương pháp theo điểm 5, trong đó tham số lọc σ bằng không (zero) nếu $QP \leq 17$, và khối bị bỏ qua mà không lọc.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 9, THR phụ thuộc vào giá trị QP.

11. Phương pháp theo điểm 10, THR được suy ra bởi:

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

trong đó C là giá trị gần với 1.

12. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 11, trong đó mẫu quét định trước được định nghĩa là tập hợp các độ lệch không gian hoặc mảnh liên quan đến vị trí của điểm ảnh hiện tại bên trong khối được xây dựng lại.

13. Phương pháp theo điểm 12, trong đó mẫu quét định trước được định nghĩa là định dạng của: $\text{scanTmpl}[4](y, x) = \{ (0,0), (0,1), (1,0), (1,1) \}$.

14. Phương pháp theo điểm 12, trong đó khi mẫu quét định trước được định nghĩa là tập hợp các độ lệch mảnh liên quan đến vị trí của điểm ảnh hiện tại bên trong khối được xây dựng lại, các vị trí độ lệch trở đến các điểm ảnh lân cận ở bên trong khối được xây dựng lại.

15. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 14, trong đó ít nhất một điểm ảnh được lọc được đặt đến vị trí ban đầu của nó theo mẫu quét định trước.

16. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 15, trong đó tất cả các điểm ảnh được lọc được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và trong đó bộ đệm tích lũy được khởi tạo bằng không (zero) trước khi thu các thành phần phổ

đã được lọc.

17. Phương pháp theo điểm 16, trong đó các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia cho số lượng điểm ảnh thêm vào vị trí hiện tại của bộ đệm tích lũy; và

trong đó bước tạo khối được lọc dựa trên các điểm ảnh được lọc bao gồm: tạo khối được lọc dựa trên các điểm ảnh được lọc cuối cùng.

18. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 17, trong đó các sự chênh lệch giữa tất cả các điểm ảnh được lọc và không được lọc tương ứng được thêm vào bộ đệm tích lũy theo mẫu quét định trước, và trong đó bộ đệm tích lũy được khởi tạo bởi các điểm ảnh không được lọc được nhân với giá trị tối đa của số lượng giá trị điểm ảnh cần được thêm vào trong khối.

19. Phương pháp theo điểm 18, trong đó các điểm ảnh được lọc cuối cùng thu được dưới dạng các giá trị được tích lũy trong bộ đệm tích lũy được chia cho giá trị tối đa của số lượng giá trị điểm ảnh cần được thêm vào trong khối.

20. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 19, trong đó phương pháp được áp dụng phụ thuộc vào các điều kiện, trong đó các điều kiện bao gồm ít nhất một trong số các điều kiện:

khối có tín hiệu dư khác không;

kích thước của khối;

tỷ lệ co của khối;

chế độ dự đoán của khối;

chế độ dự đoán của khối là nội dự đoán; hoặc

chế độ dự đoán của khối là liên dự đoán.

21. Phương pháp theo điểm 20, trong đó tham số lọc và/hoặc mẫu quét định trước phụ thuộc vào các điều kiện.

22. Phương pháp theo điểm 20, trong đó phương pháp được áp dụng nếu kích thước của khối thu được dưới dạng chiều rộng khối được nhân với chiều cao khối lớn hơn 4×4 .

23. Phương pháp theo điểm 20, trong đó phương pháp không được áp dụng nếu chế độ dự đoán của khối là liên dự đoán và chiều rộng khối và chiều cao khối lớn hơn hoặc

bảng 32.

24. Phương pháp theo điểm 20, trong đó phương pháp được áp dụng nếu chế độ dự đoán của khối là liên dự đoán và khối có tín hiệu dư khác không.

25. Phương pháp theo điểm 20, trong đó phương pháp được áp dụng nếu chế độ dự đoán của khối khối là nội dự đoán.

26. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 25, còn bao gồm bước quét điểm ảnh hiện tại của khối được xây dựng lại và các điểm ảnh lân cận của nó của điểm ảnh hiện tại theo mẫu quét định trước vào bộ đệm tích lũy; và

thu các thành phần phổ bằng cách thực hiện biến đổi cho các điểm ảnh trong bộ đệm tích lũy.

27. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 26, trong đó biến đổi là biến đổi Hadamard.

28. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 27, trong đó biến đổi là biến đổi 1D.

29. Bộ mã hóa bao gồm mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

30. Bộ giải mã bao gồm mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

31. Bộ mã hóa bao gồm bộ lọc để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

32. Bộ giải mã bao gồm bộ lọc để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

33. Bộ mã hóa, bộ mã hóa này bao gồm:

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 thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

34. Bộ giải mã, bộ giải mã này bao gồm:

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 thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 28.

35. Phương tiện lưu trữ đọc được bằng máy tính có chứa trên đó 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 28 khi được thực hiện trên máy tính hoặc bộ xử lý.

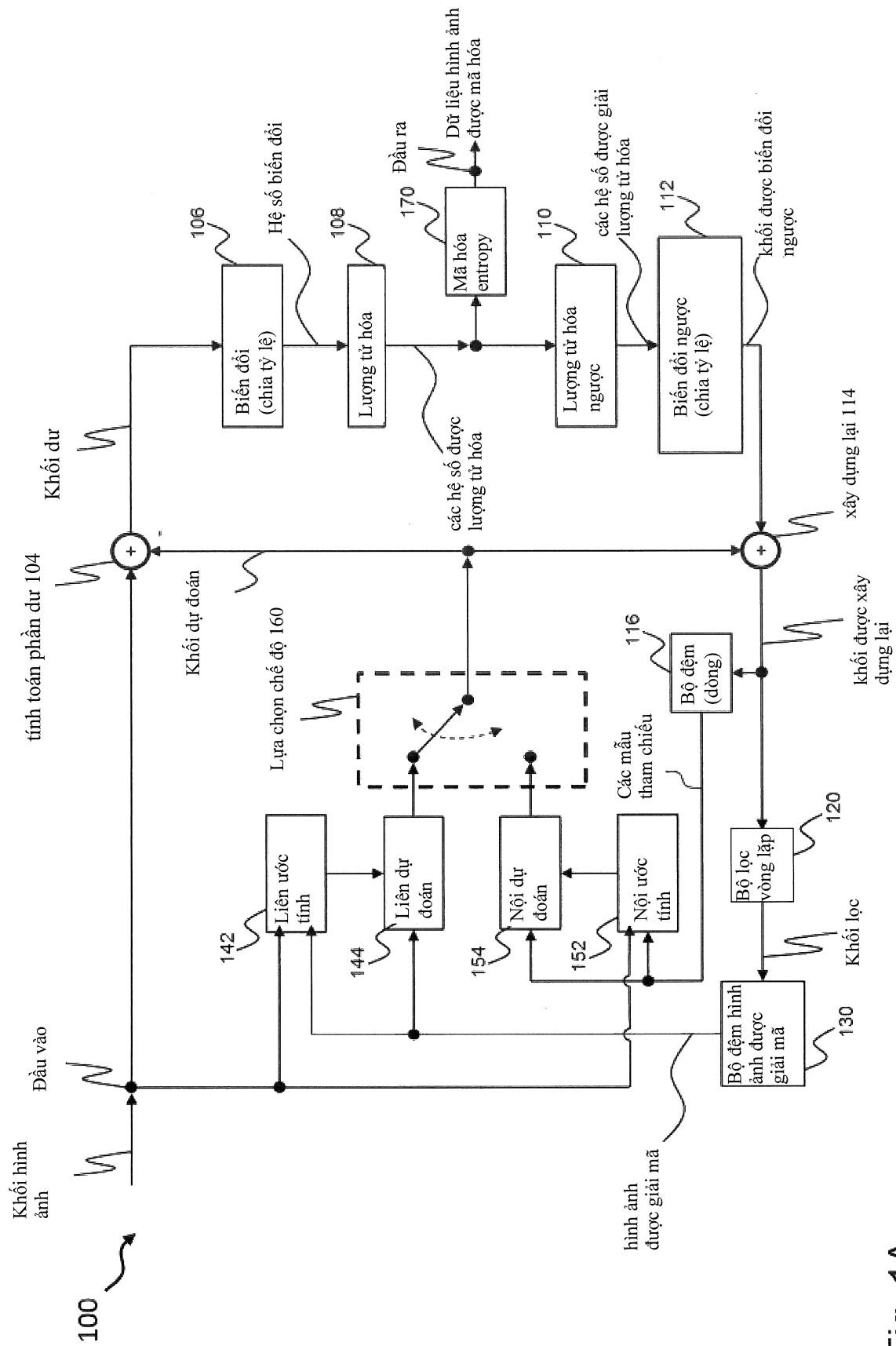


Fig. 1A

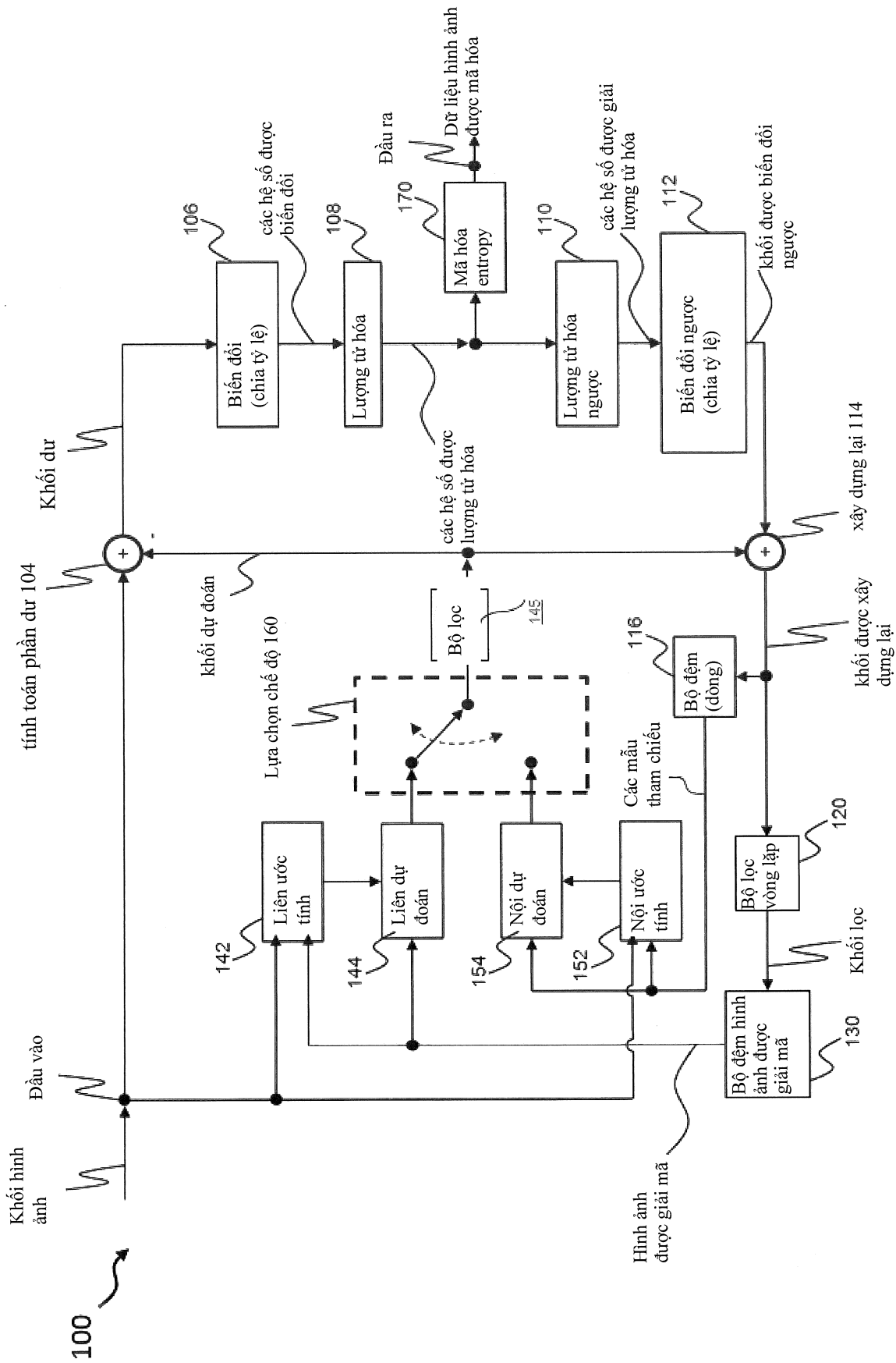


Fig. 1B

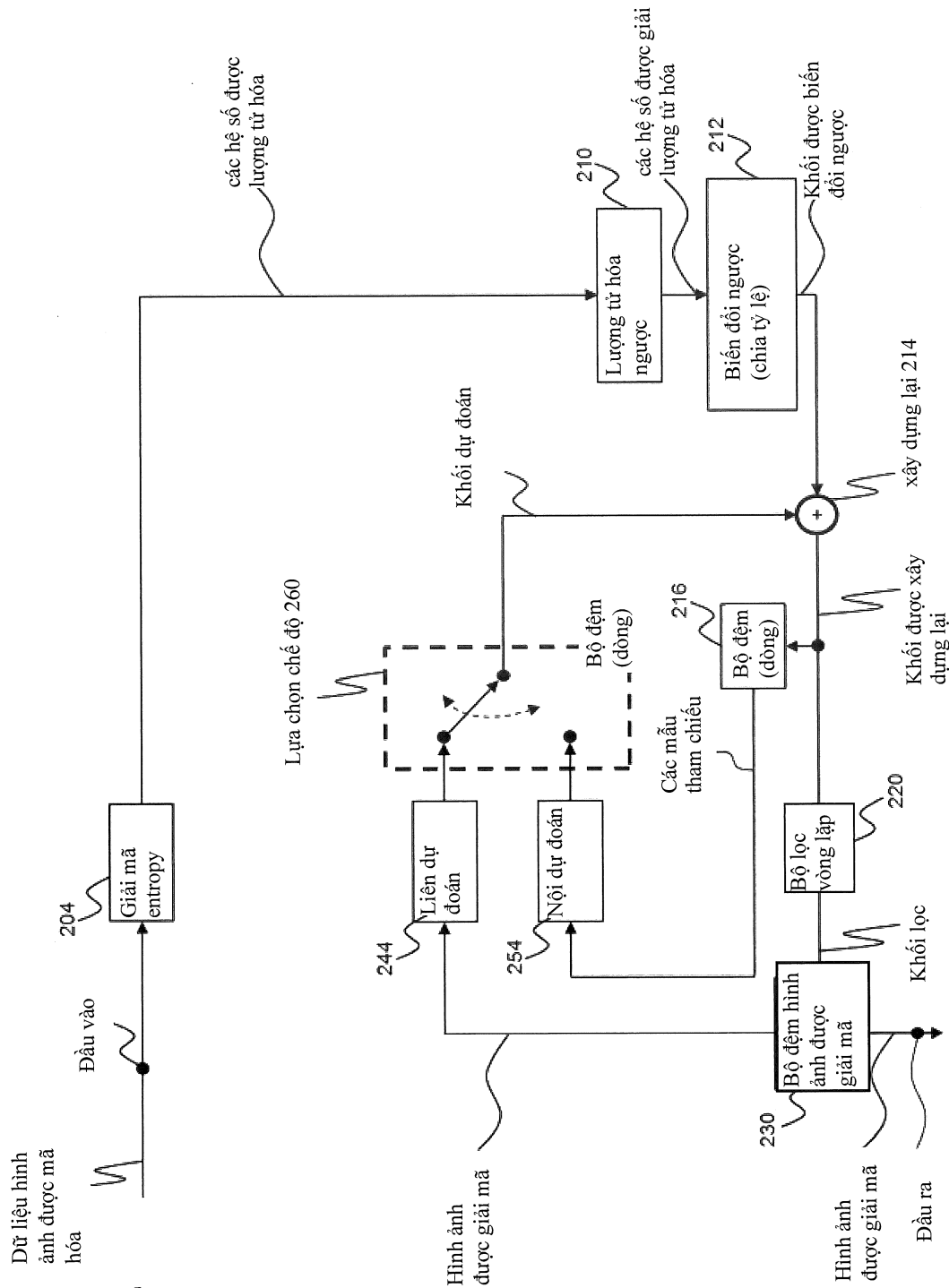


Fig. 2A

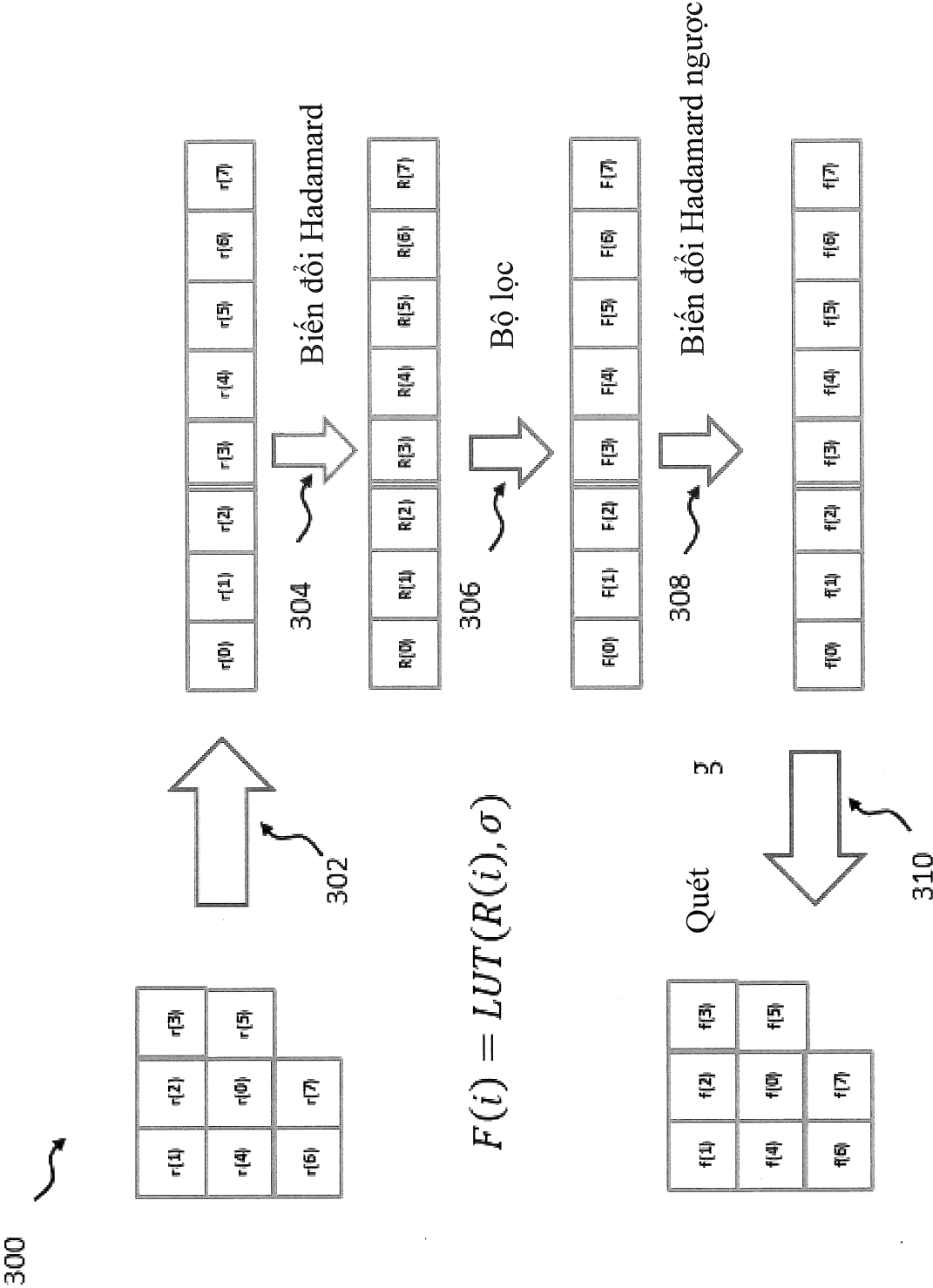


Fig. 3A

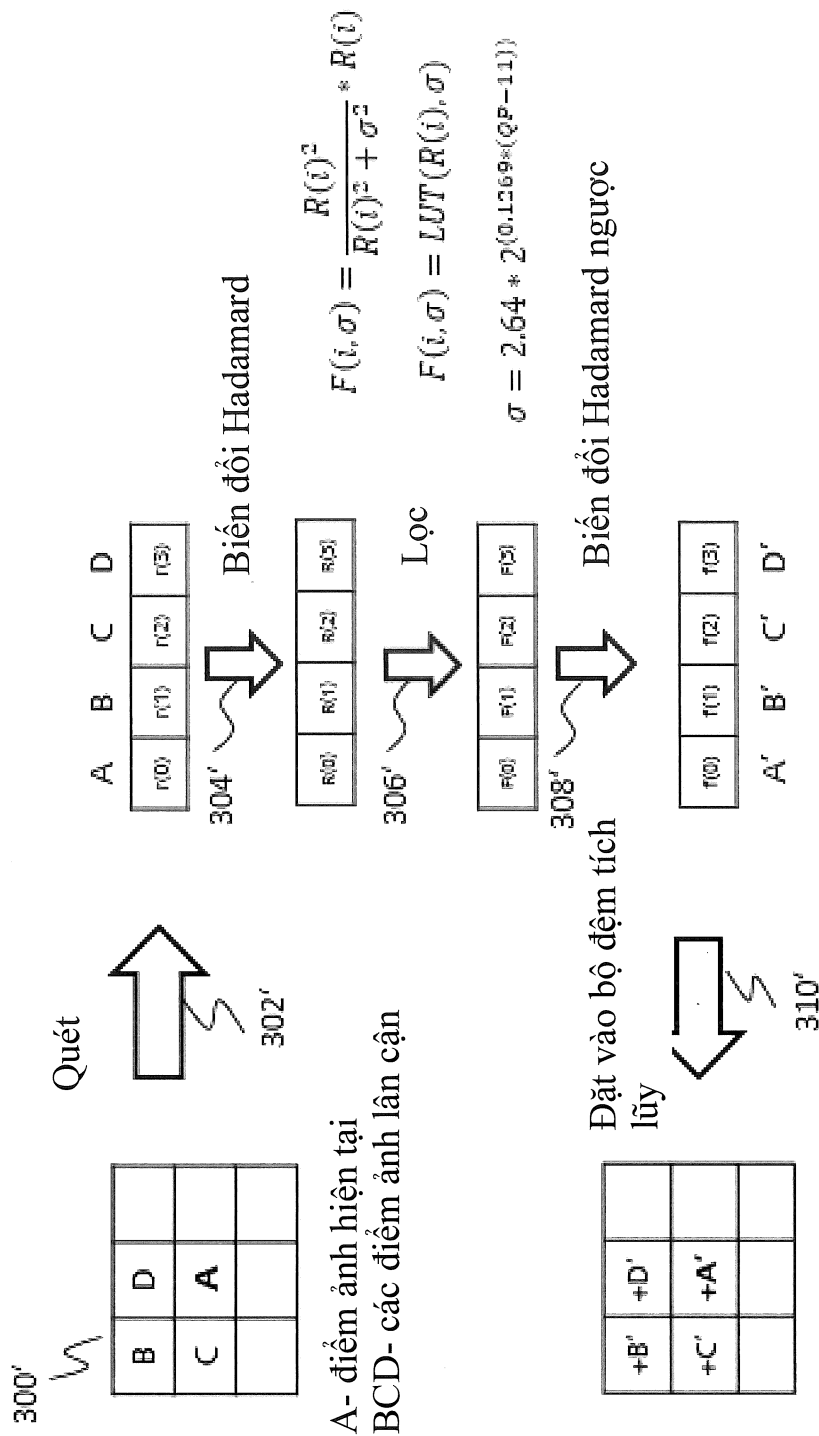


Fig. 3B

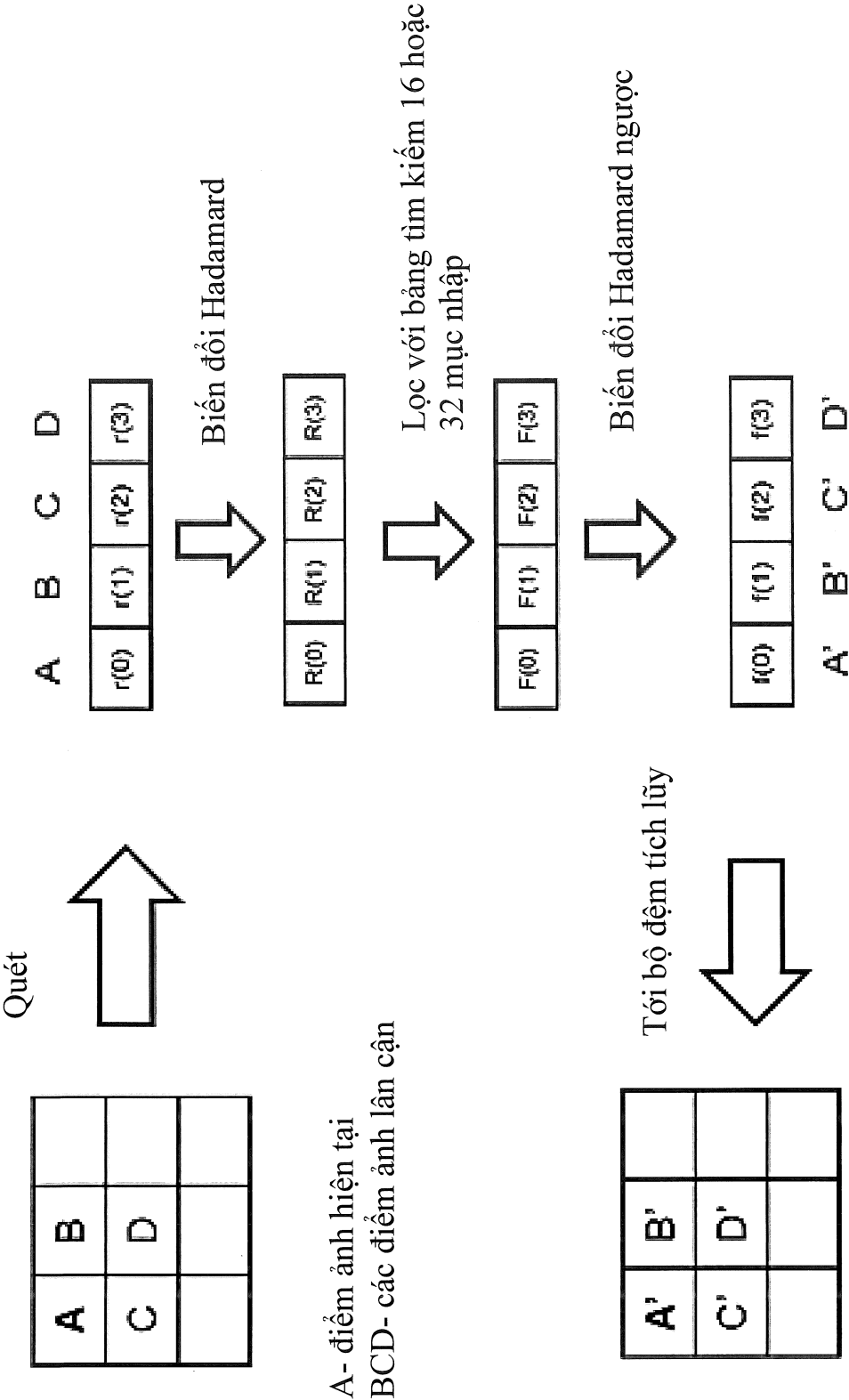


Fig. 3C

8/26

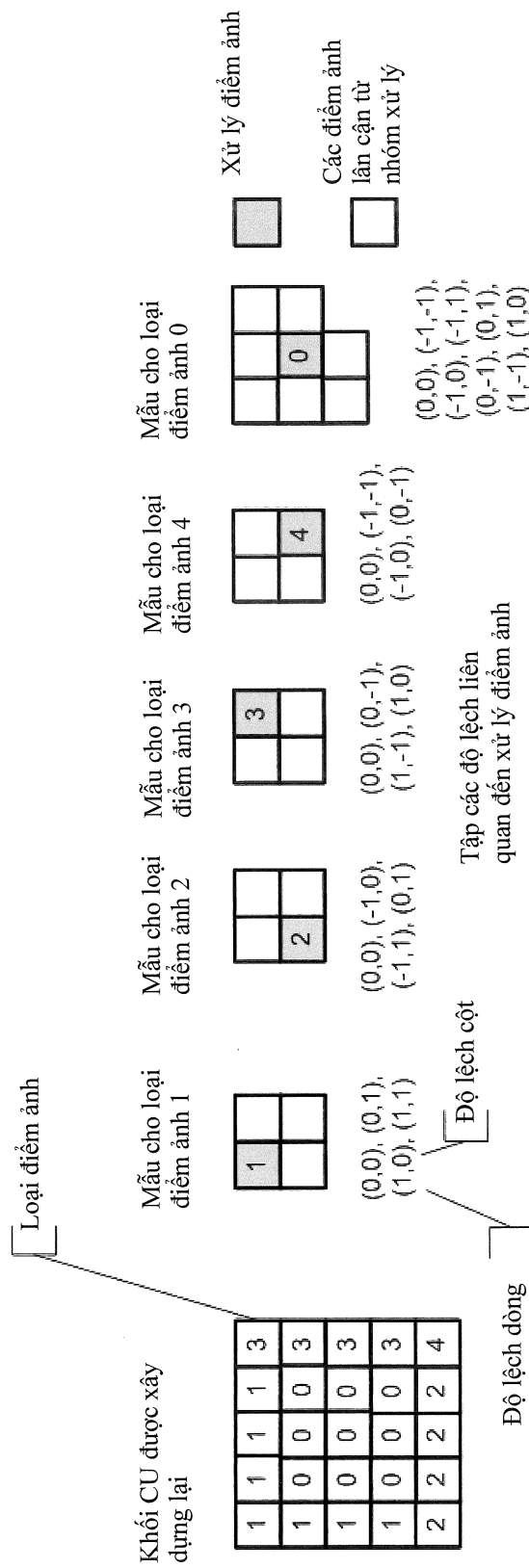


Fig. 4A

9/26

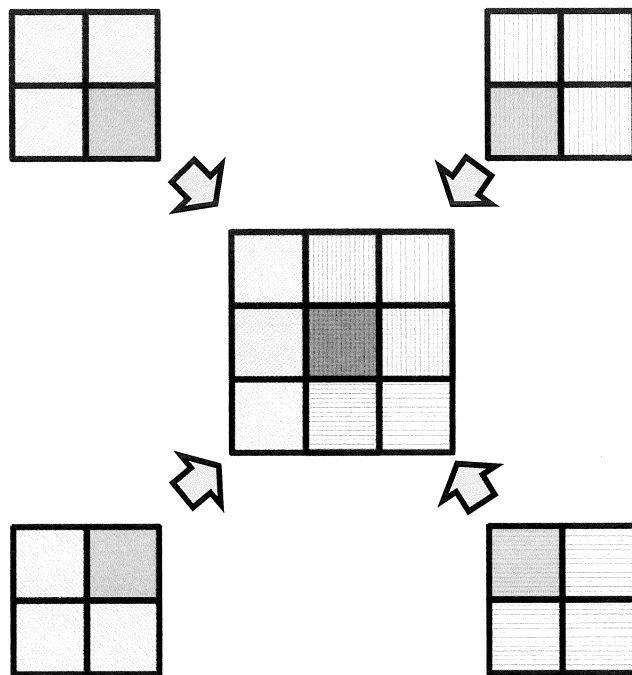


Fig. 4B

10/26

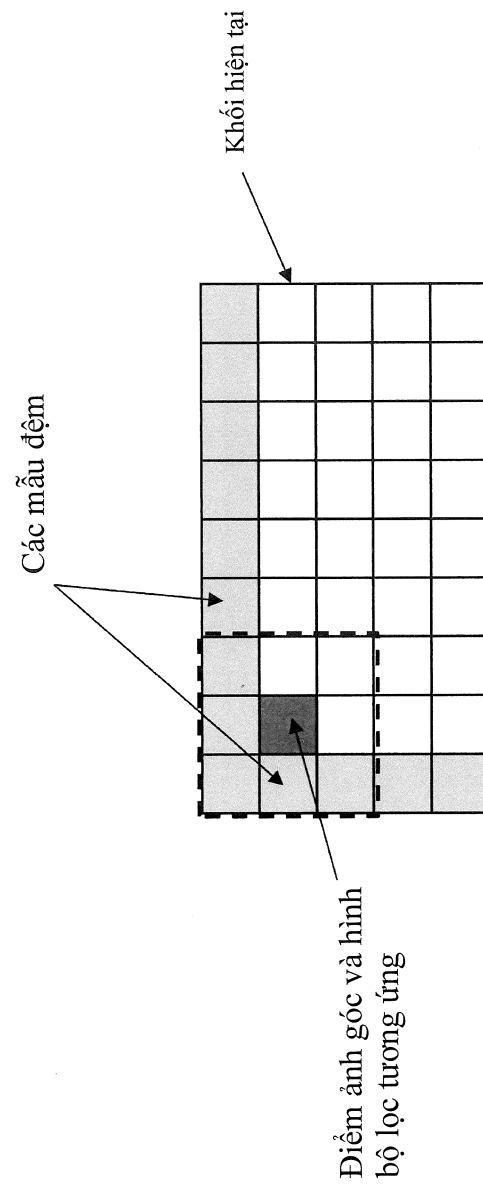


Fig. 4C

11/26

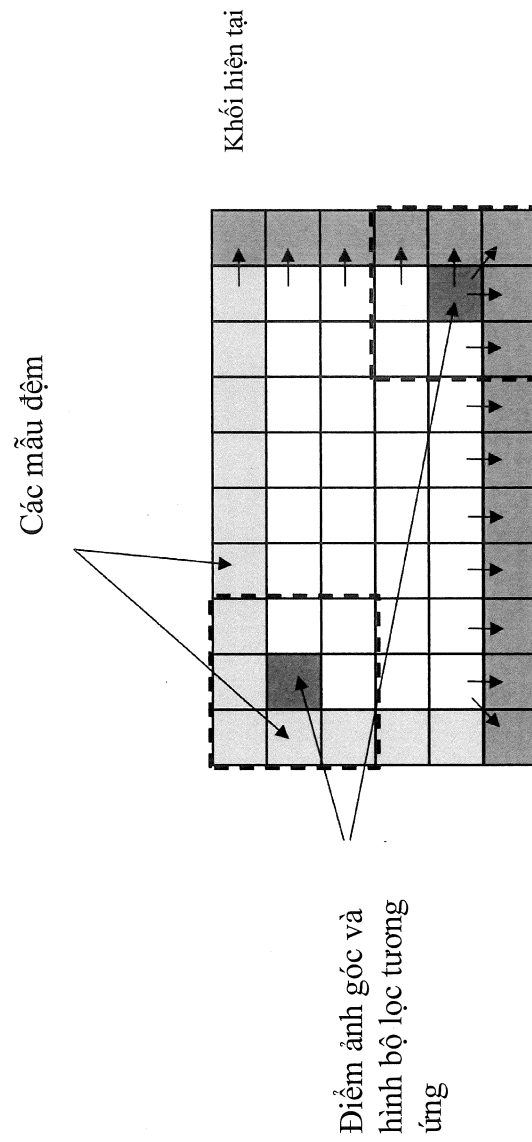


Fig. 4D

12/26

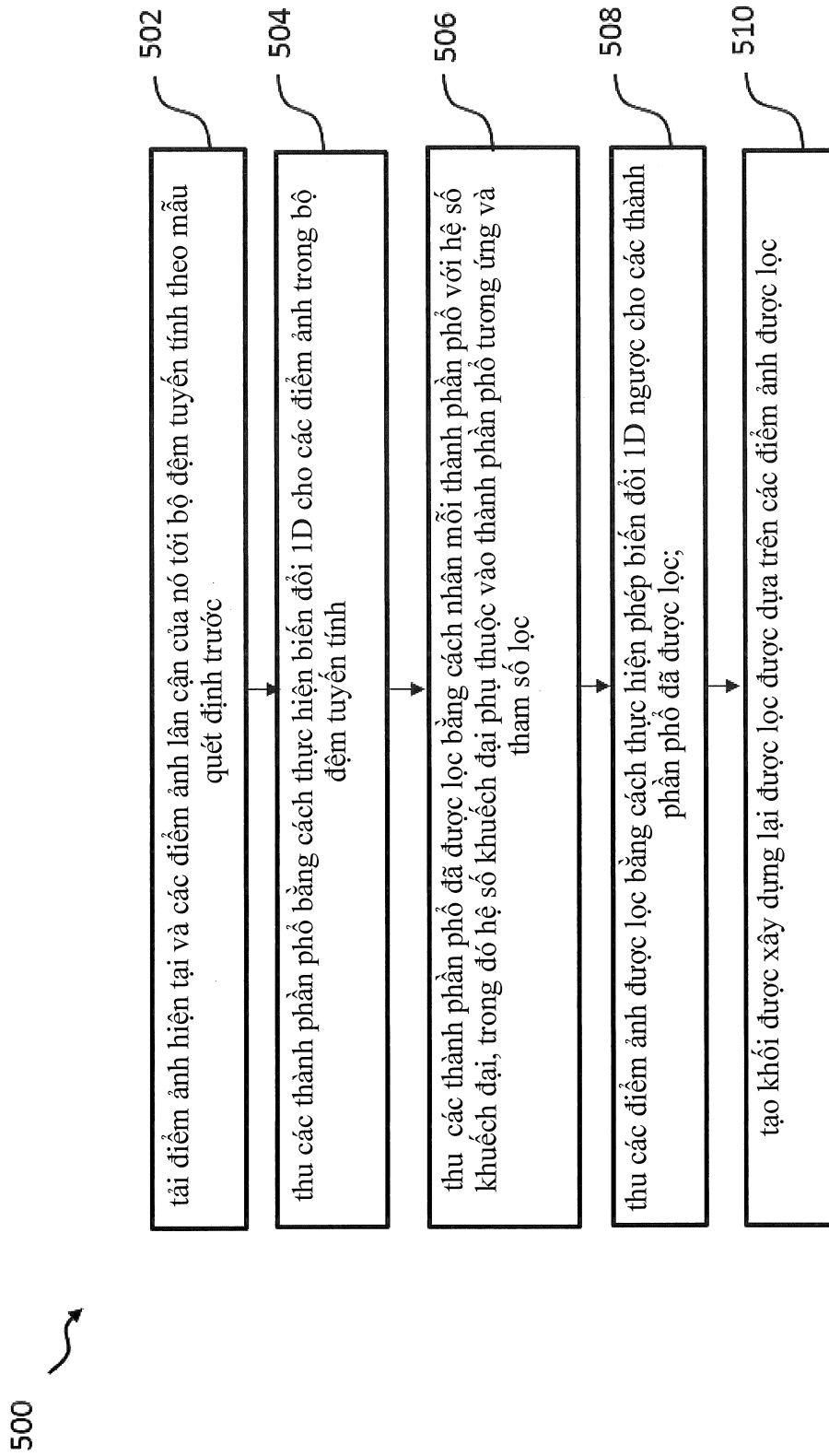


Fig. 5A

13/26

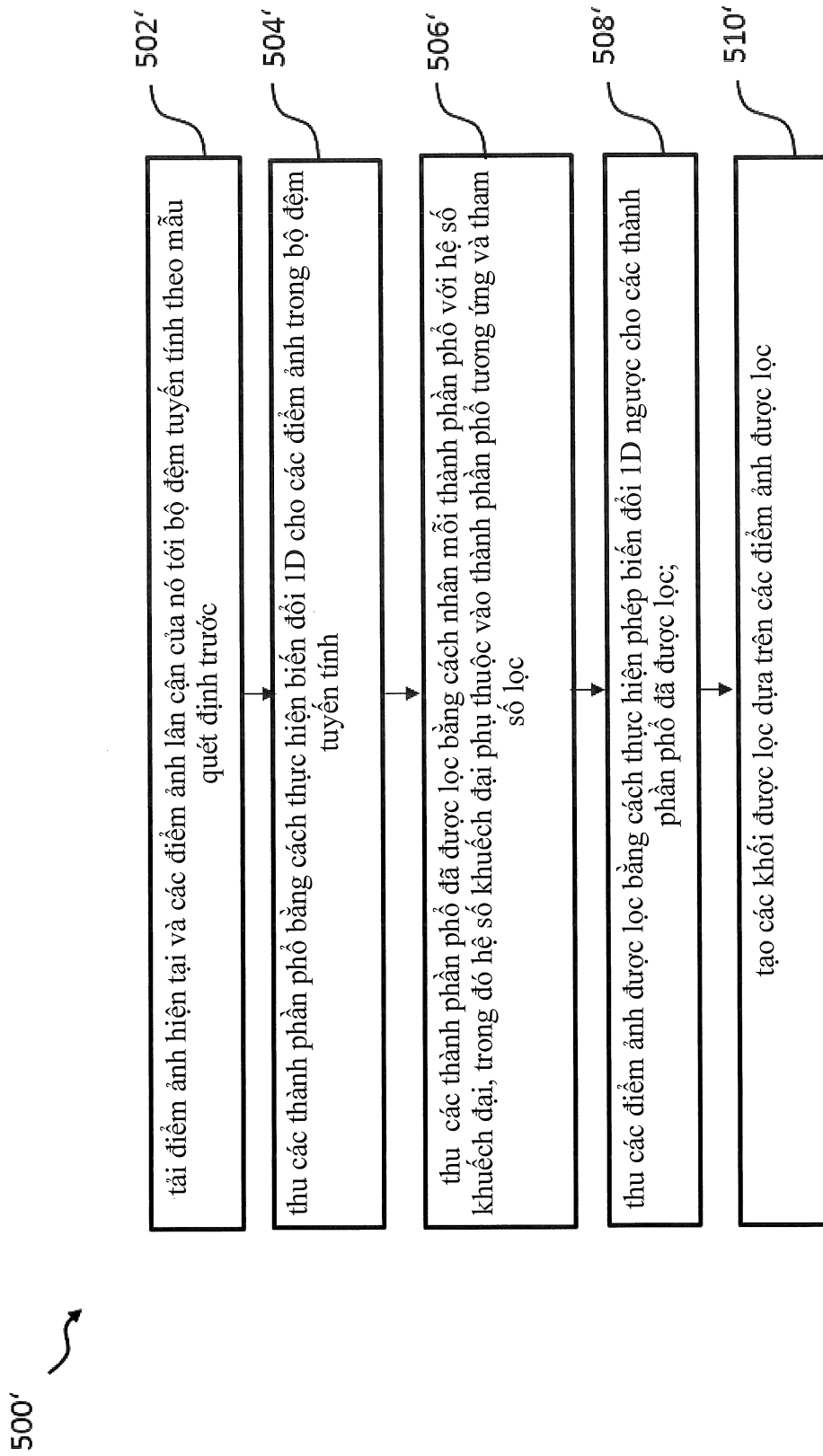


Fig. 5B

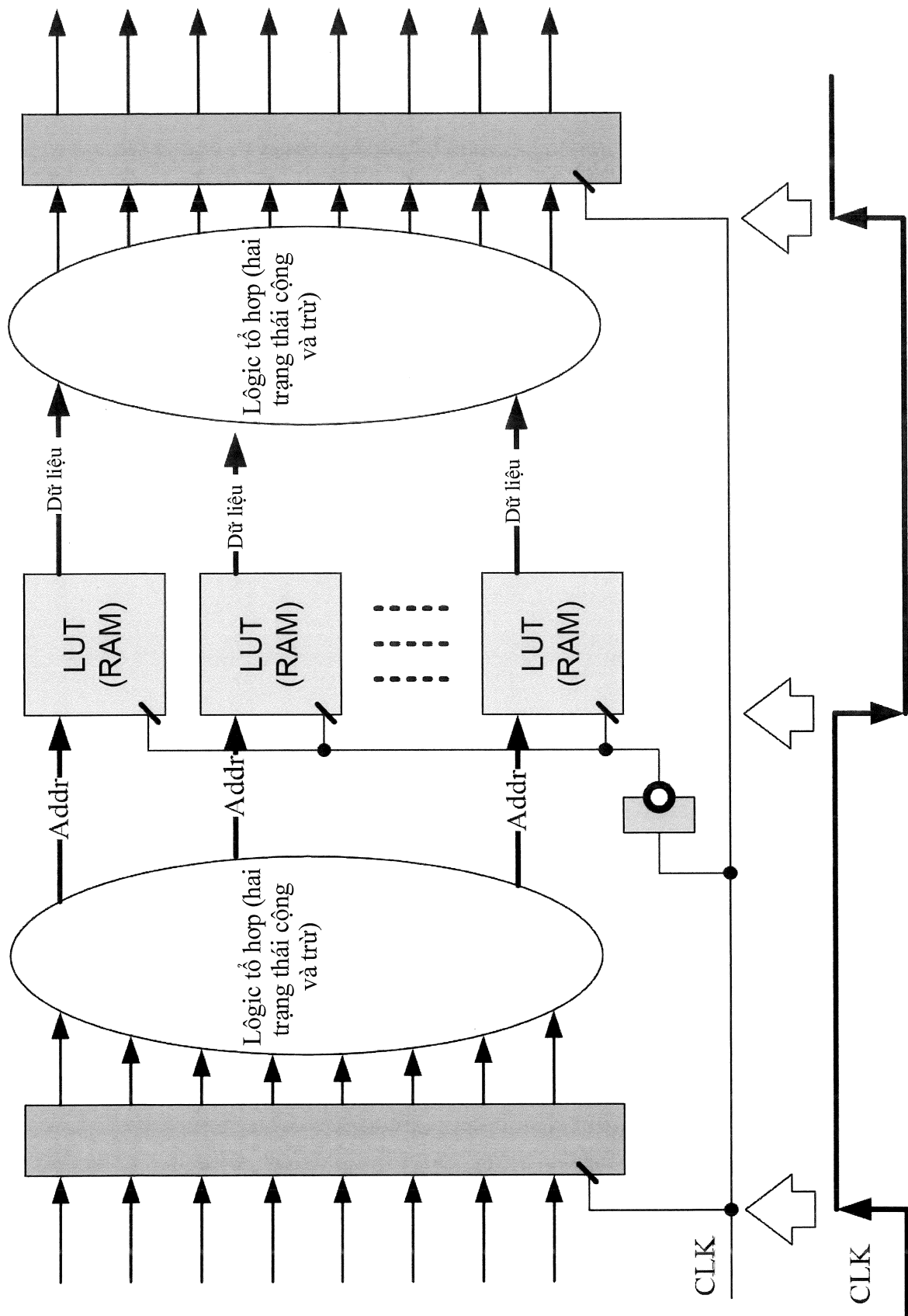
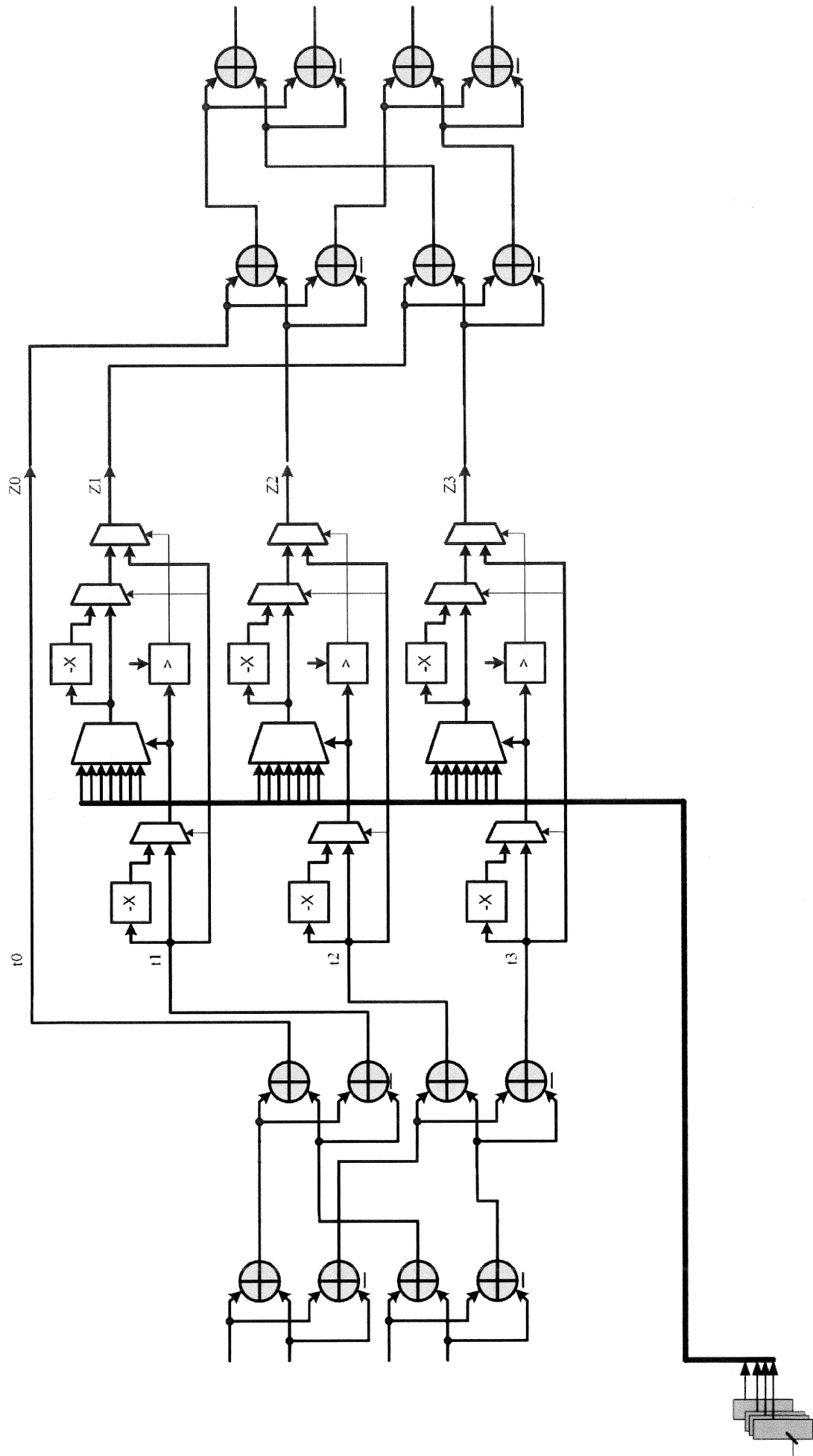


FIG. 6

15/26



16/26

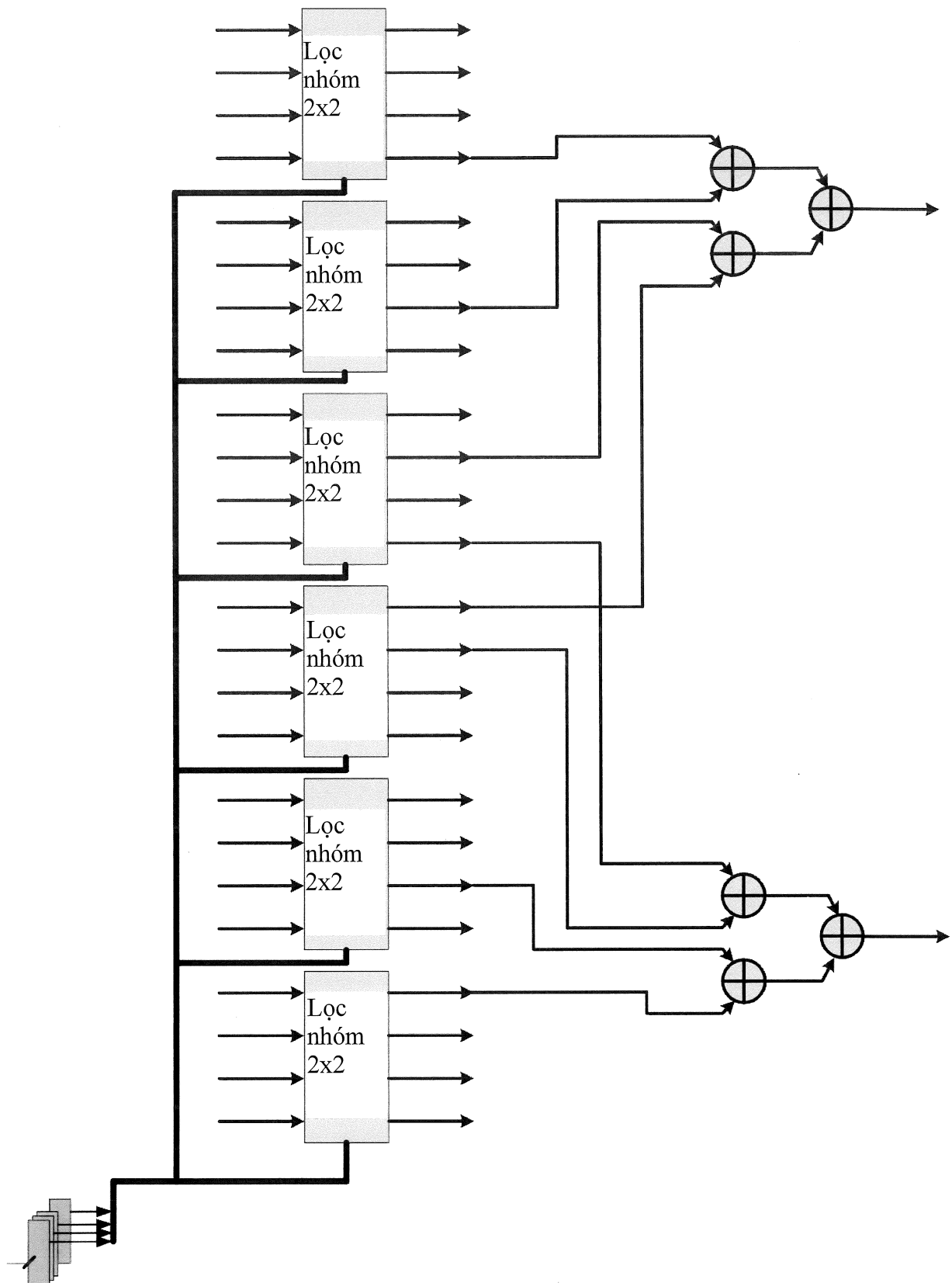


FIG. 8

17/26

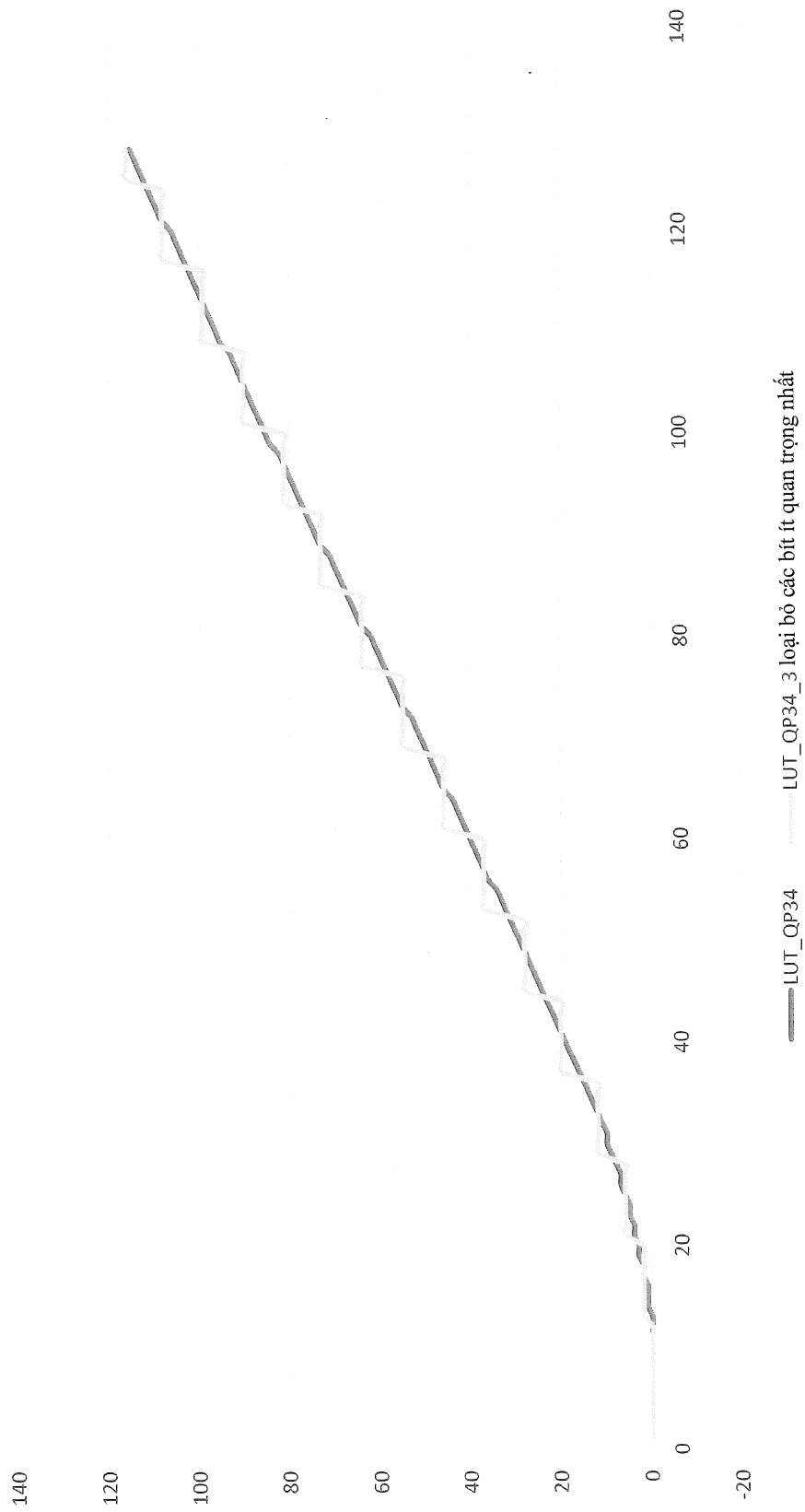


FIG. 9

18/26

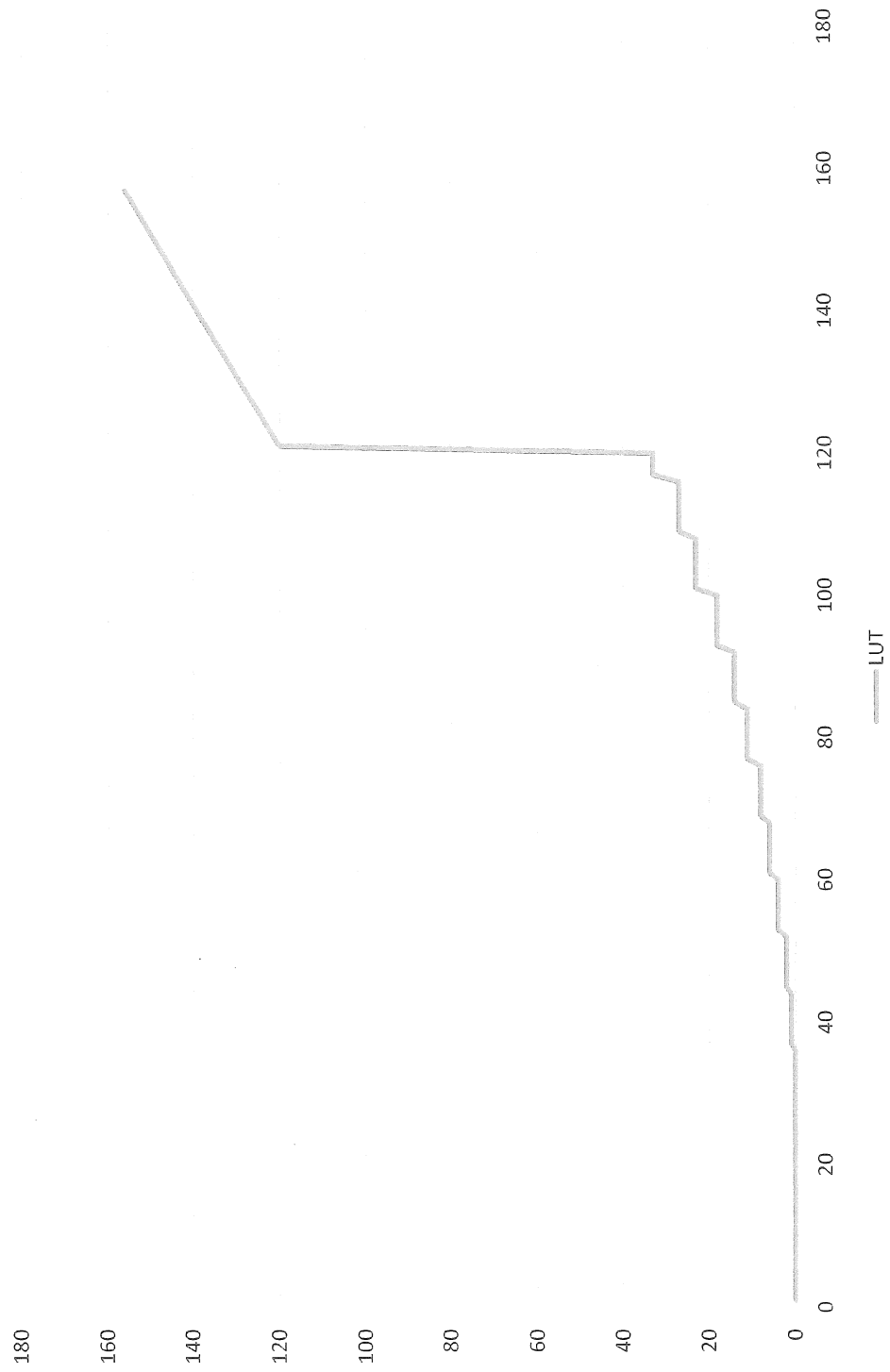


FIG. 10

19/26

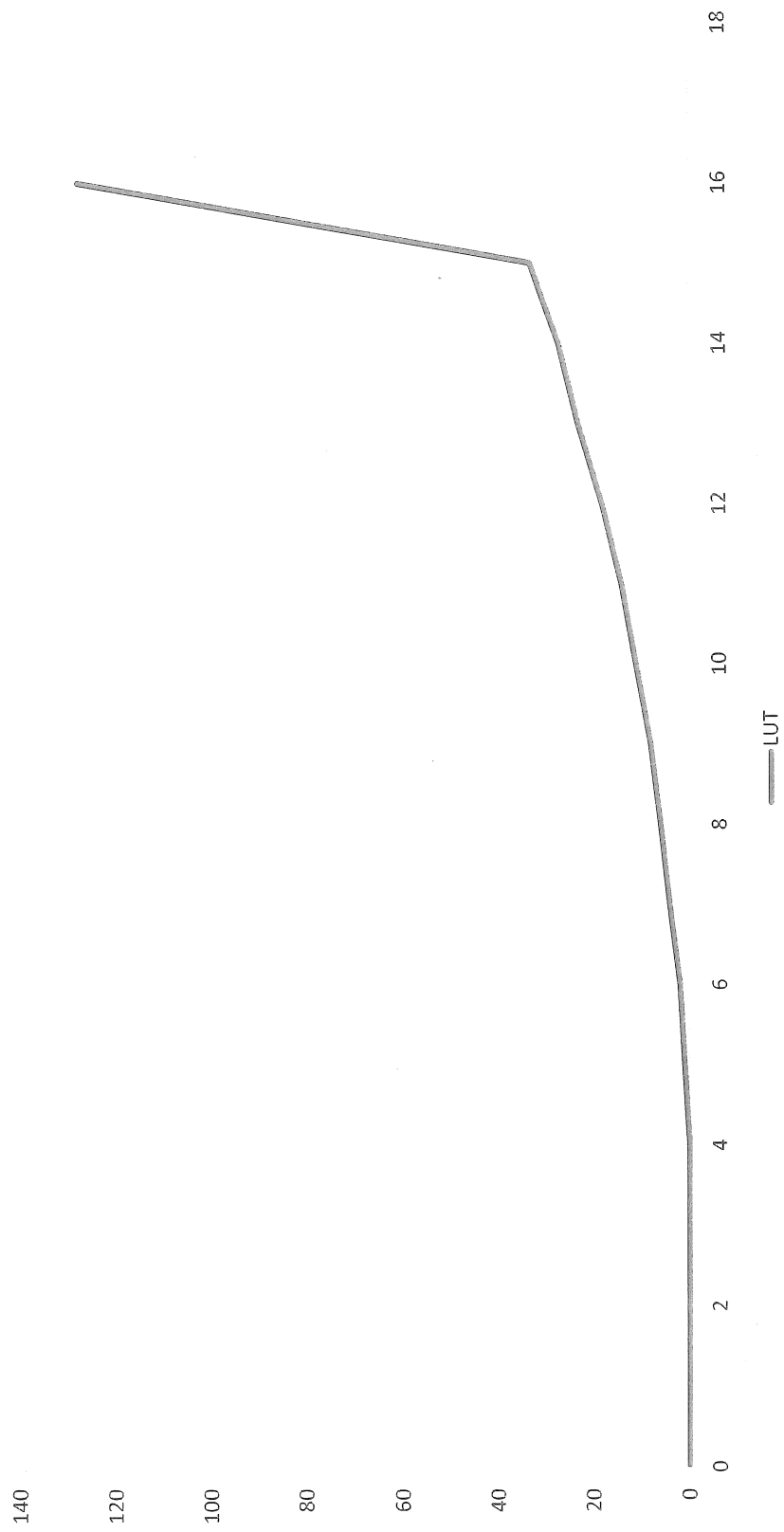


FIG. 11

20/26

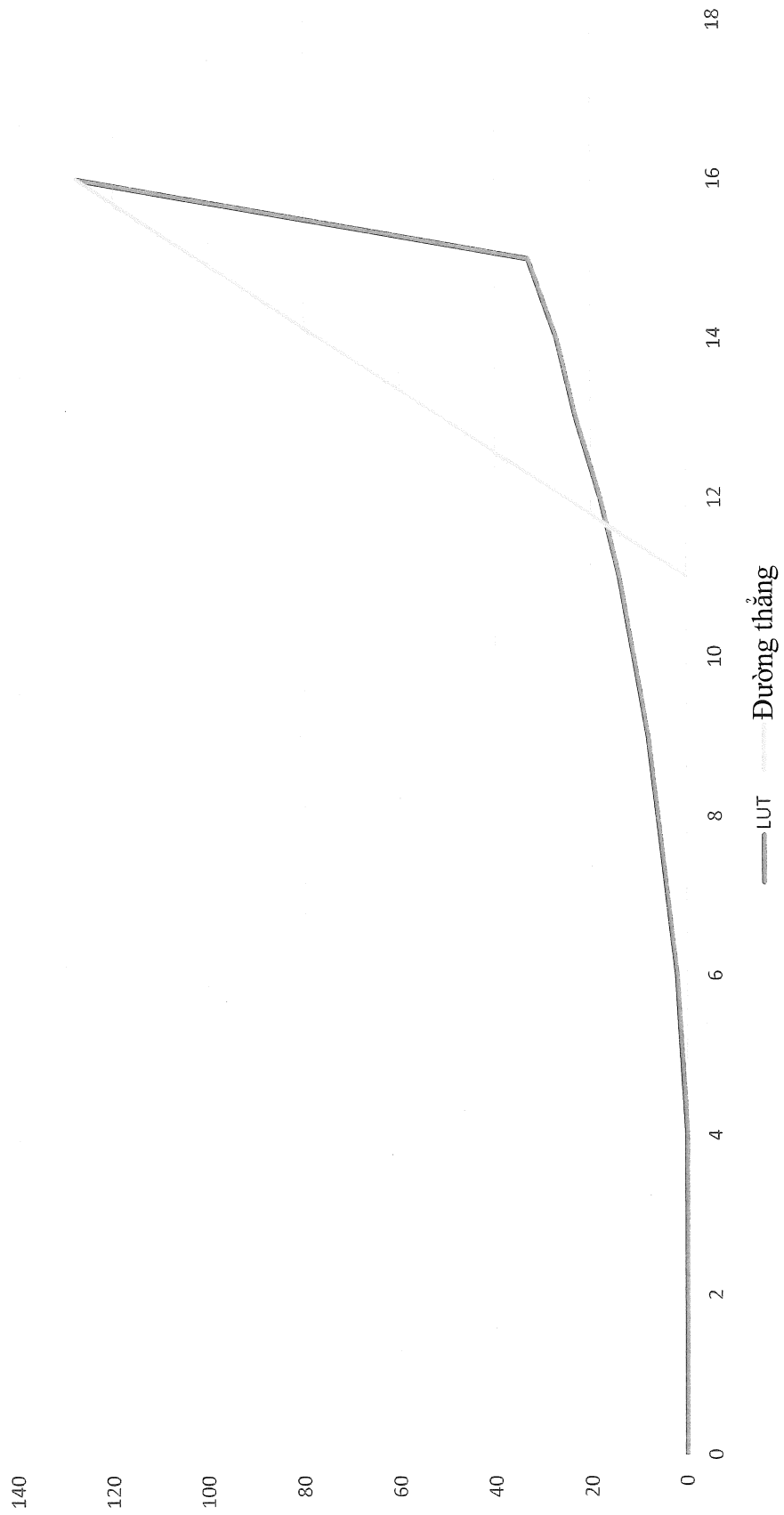


FIG. 12

21/26

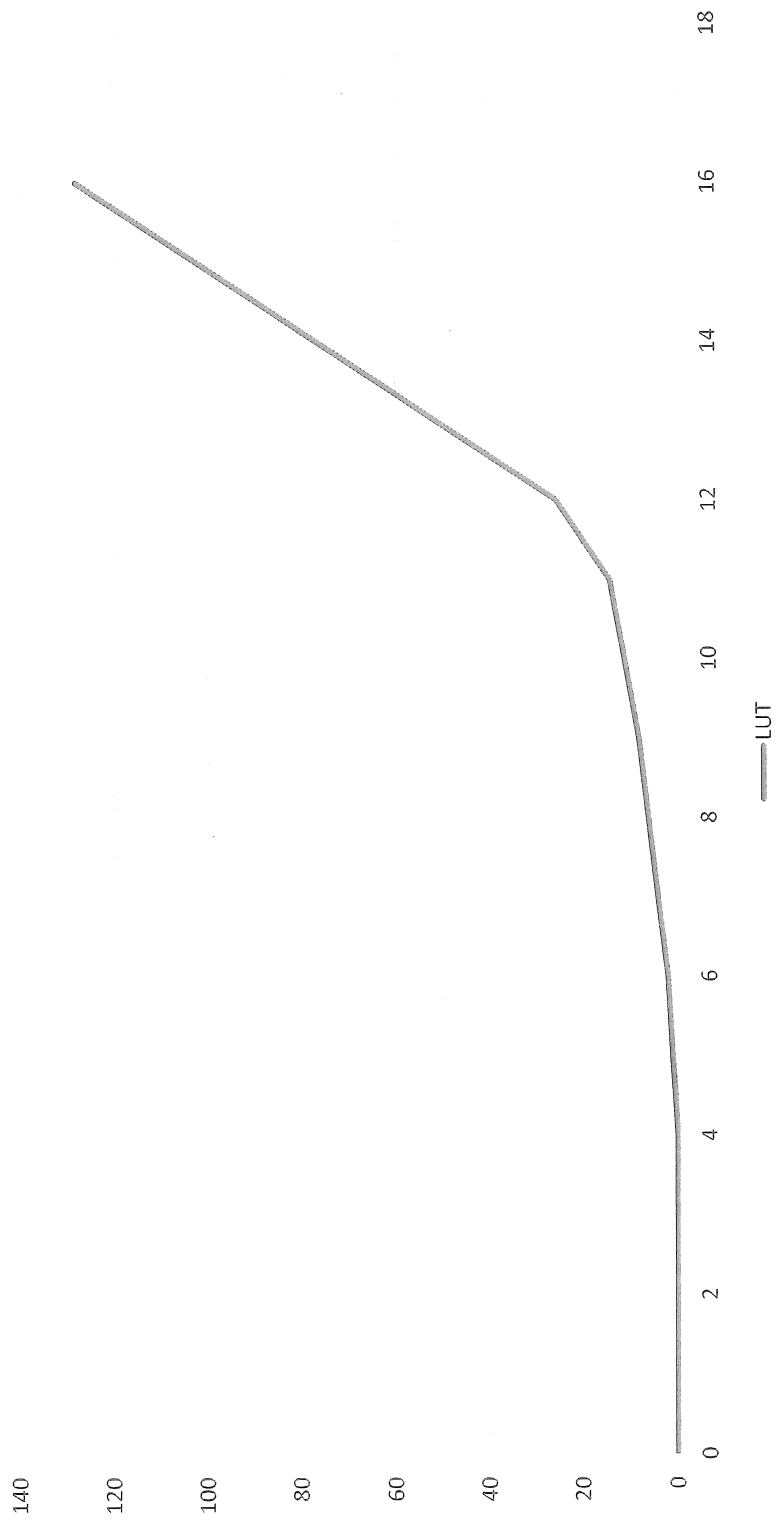


FIG. 13

22/26

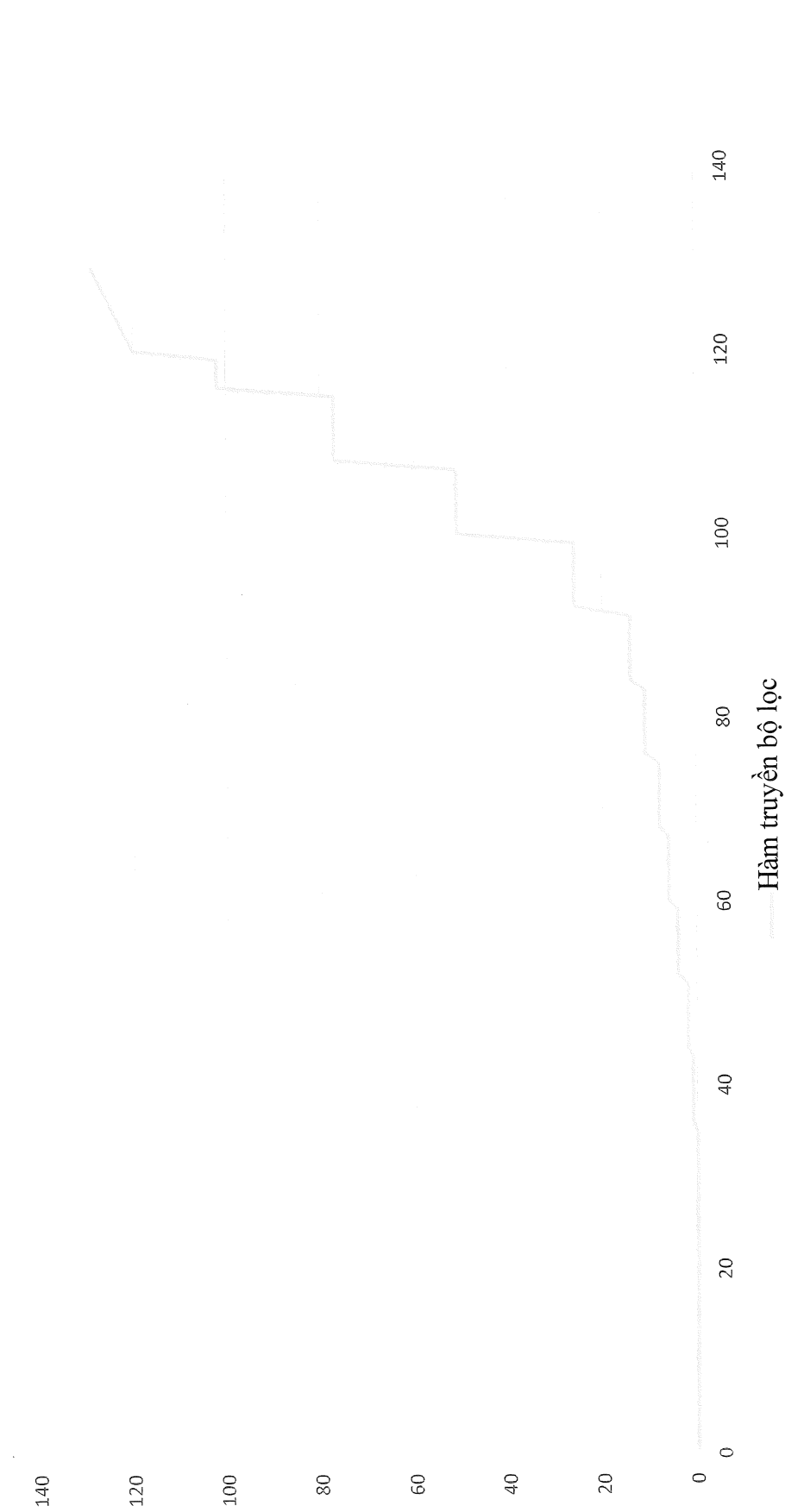


FIG. 14

23/26

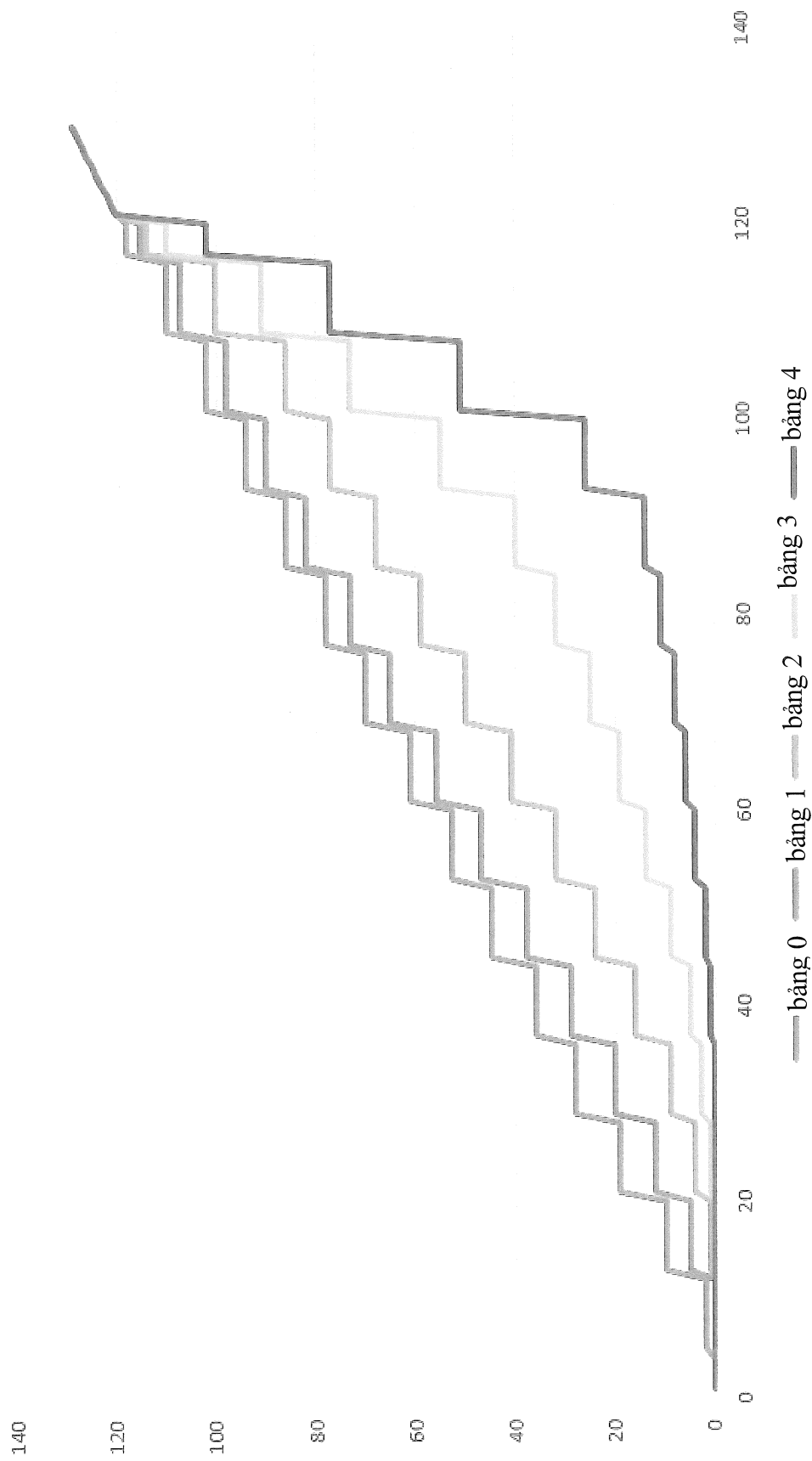


FIG. 15

24/26

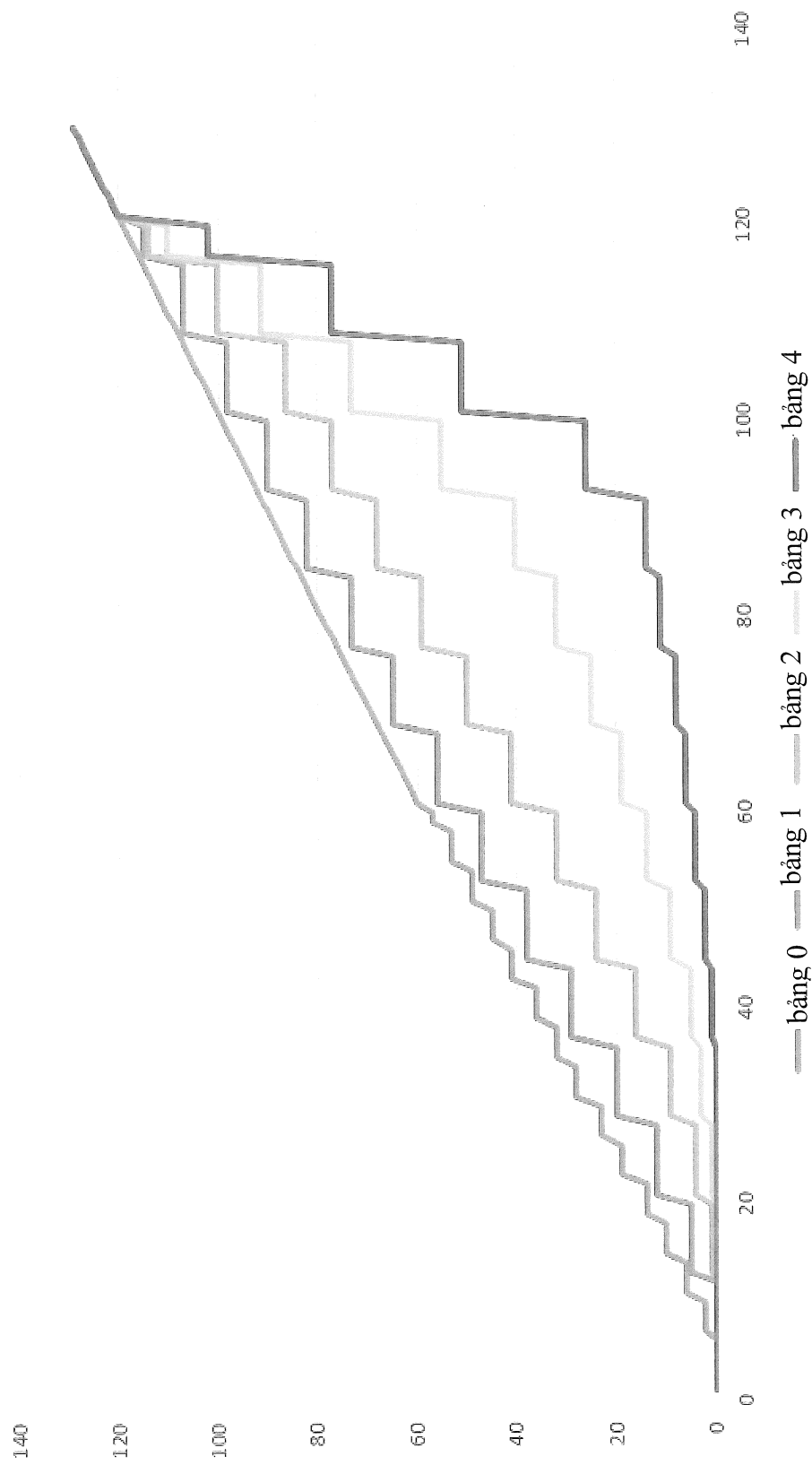


FIG. 16

25/26

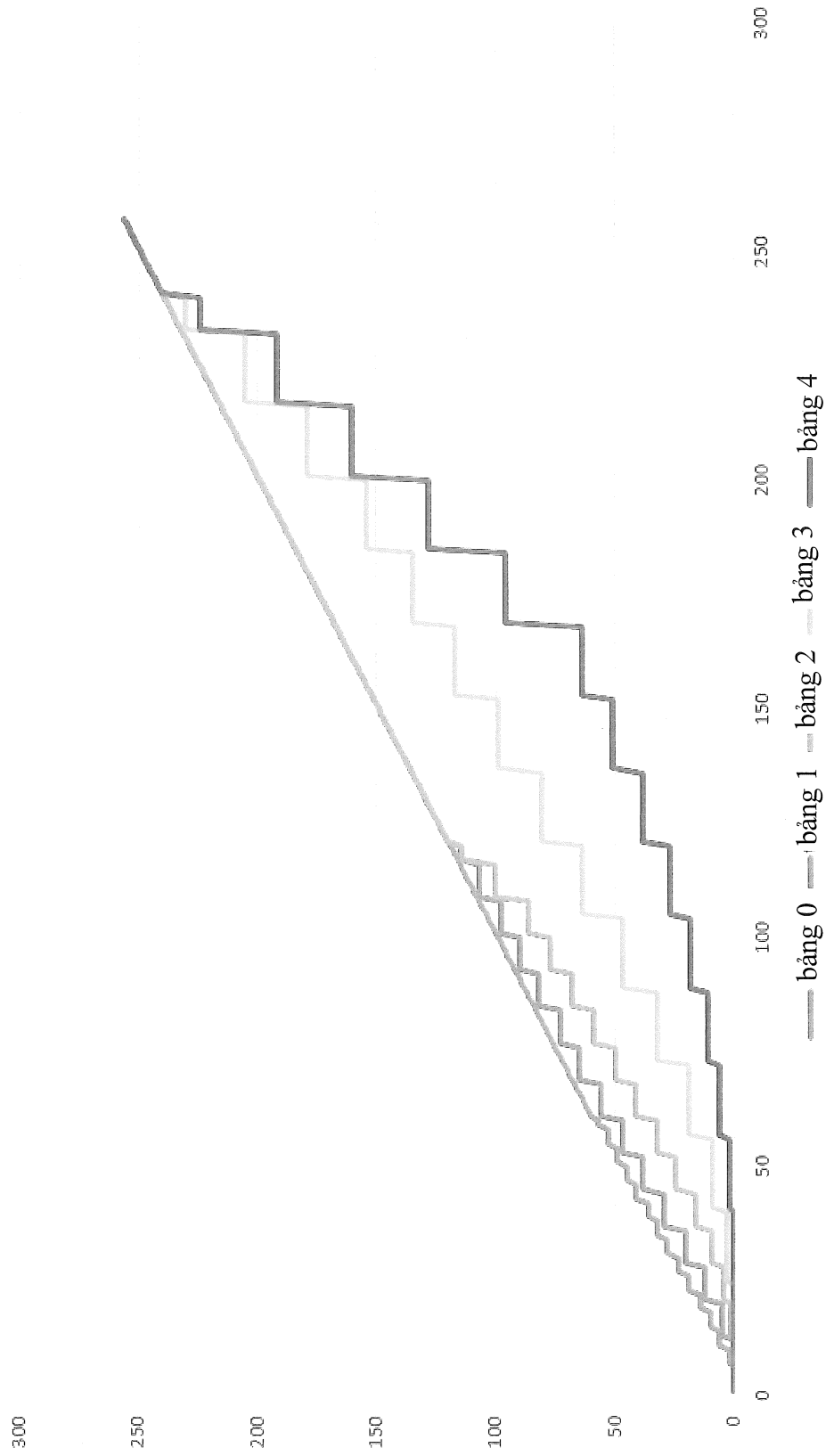


FIG. 17

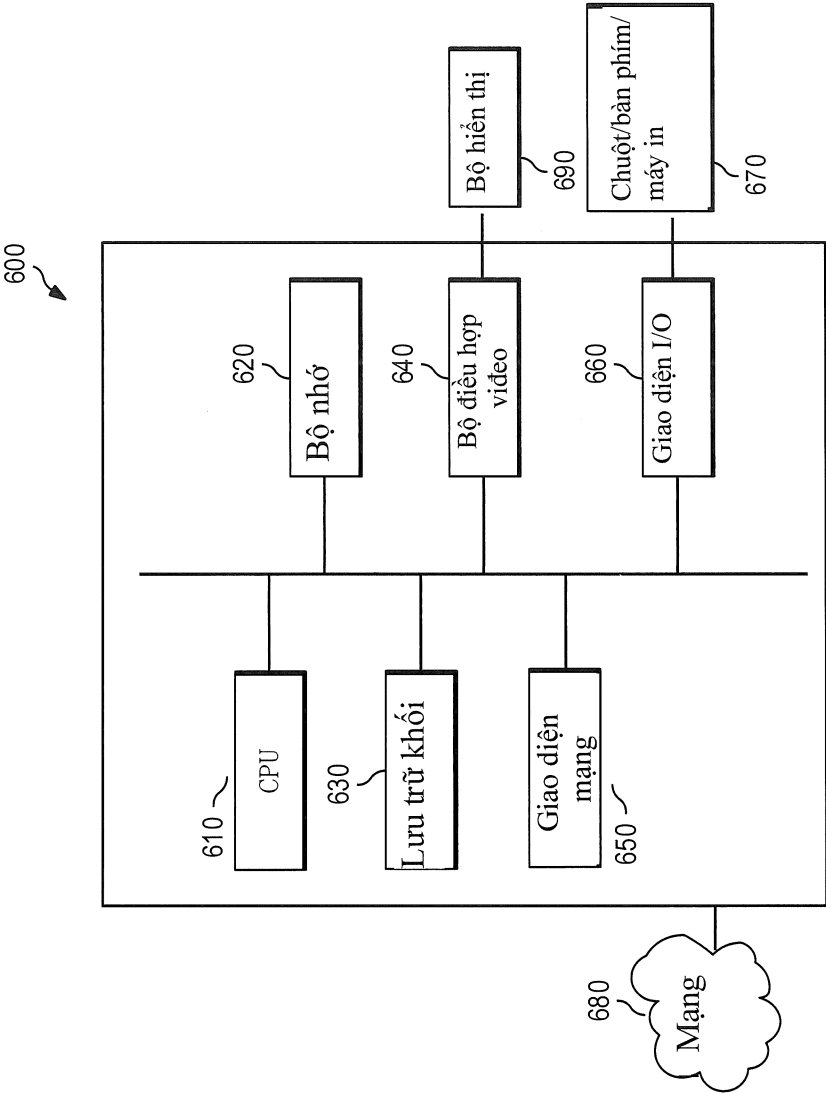


FIG. 18