



(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)



1-0048207

(51)<sup>2020.01</sup> H04N 19/117; H04N 19/176

(13) B

(21) 1-2021-07754

(22) 30/04/2020

(86) PCT/CN2020/088389 30/04/2020

(87) WO2020/224545 12/11/2020

(30) 62/843,431 04/05/2019 US

(45) 25/07/2025 448

(43) 25/02/2022 407A

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

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

(72) KOTRA, Anand Meher (IN); ESENLİK, Semih (TR); CHEN, Jianle (CN); GAO,  
Han (CN); WANG, Biao (CN).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP TẠO MÃ, BỘ GIẢI MÃ, BỘ MÃ HÓA VÀ PHƯƠNG TIỆN  
LƯU TRỮ KHÔNG TẠM THỜI

(21) 1-2021-07754

(57) Sáng chế đề cập đến phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước: thu nhận dòng bit trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (adaptive loop filter, viết tắt là ALF); phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại. Ở đây mã độ dài cố định nghĩa là tất cả các trị số khả thi của thành phần cú pháp được báo hiệu nhờ sử dụng số các bit như nhau. Sáng chế cũng đề cập đến bộ giải mã, bộ mã hóa và phương tiện lưu trữ không tạm thời.

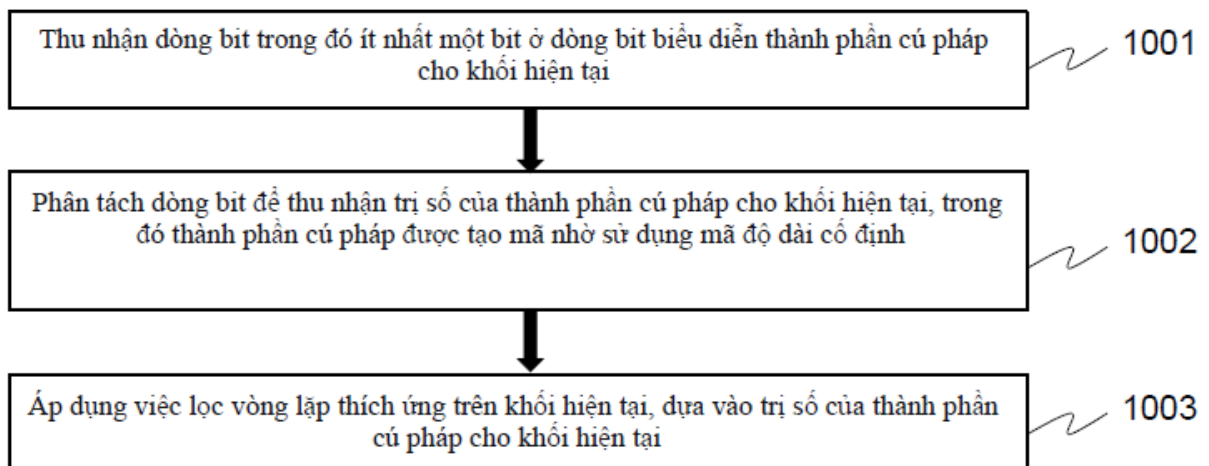


FIG. 10

### **Lĩnh vực kỹ thuật được đề cập**

Các phương án của sáng chế nói chung đề cập đến lĩnh vực về việc xử lý ảnh và cụ thể hơn là đến việc lọc các mẫu của các khối trong ảnh.

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

Tạo mã video (mã hóa và giải mã video) được sử dụng trong phạm vi rộng của các ứng dụng video số, ví dụ TV số phát rộng, truyền video qua Internet và các mạng di động, các ứng dụng đàm thoại thời gian thực chẳng hạn như trò chuyện video, hội nghị video, các đĩa DVD và Blu-ray, các hệ thống thu nhận và biên tập nội dung video, và các máy quay video của các ứng dụng bảo mật.

Lượng dữ liệu video được cần để mô tả thậm chí ngay cả video tương đối ngắn có thể là đáng kể, mà có thể dẫn đến những khó khăn khi dữ liệu được tạo dòng hoặc được truyền thông theo cách khác dọc theo mạng truyền thông với dung lượng độ rộng dải giới hạn. Do vậy, dữ liệu video thường được nén trước khi được truyền thông dọc theo các mạng viễn thông hiện nay. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ bởi vì các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để tạo mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó làm giảm lượng dữ liệu được cần để biểu diễn các hình ảnh video số. Dữ liệu được nén sau đó được thu ở đích bởi thiết bị giải nén video mà giải mã dữ liệu video. Với các tài nguyên mạng hạn chế và các nhu cầu ngày càng tăng về chất lượng video cao hơn, các kỹ thuật nén và giải nén được nâng cao mà nó nâng cao tỷ lệ nén với độ suy giảm bằng không hoặc không đáng kể về chất lượng ảnh là điều mong muốn.

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

Các phương án của sáng chế đề xuất các máy và các phương pháp để mã hóa và giải mã theo các điểm yêu cầu bảo hộ độc lập.

Các mục đích nêu trên và các mục đích khác đạt được bởi đối tượng của các điểm yêu cầu bảo hộ độc lập. Các dạng thực hiện thêm là rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phân mô tả và các hình vẽ.

Theo khía cạnh thứ nhất, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (adaptive loop filter, viết tắt là ALF); phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại. Ở đây mã độ dài cố định nghĩa là tất cả các trị số khả thi của thành phần cú pháp được báo hiệu nhờ sử dụng số các bit như nhau. Điều này đưa ra phương pháp đơn giản hơn của việc báo hiệu các thông số cắt bớt. Hơn nữa, hiệu quả tạo mã được nâng cao.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ nhất như vậy, trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, ít nhất một bit là hai bit.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, trong đó ít nhất một bit ở dòng bit biểu diễn trị số của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, trong đó thành phần cú pháp cho bộ lọc vòng lặp thích ứng chroma (sắc độ) hoặc bộ lọc vòng lặp thích ứng luma (độ chói).

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, trong đó trị số cắt bớt được sử dụng để xác định khoảng cắt bớt được sử dụng để giới hạn (hoặc cắt bớt) độ chênh lệch giữa trị số mẫu đích và trị số mẫu ở gần, và độ chênh lệch trị số mẫu được giới hạn (hoặc độ chênh lệch trị số mẫu được cắt bớt) được sử dụng để điều chỉnh trị số mẫu đích trong quy trình xử lý của ALF.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, trong đó bước áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp bao gồm thu nhận trị số cắt bớt

dựa vào trị số của thành phần cú pháp; sử dụng trị số cắt bớt để giới hạn (hoặc cắt bớt) độ chênh lệch giữa trị số mẫu đích của khối hiện tại và trị số mẫu ở gần; nhân độ chênh lệch trị số mẫu được giới hạn (hoặc độ chênh lệch trị số mẫu được cắt bớt) với hệ số của bộ lọc vòng lặp thích ứng (ALF); sử dụng kết quả của phép nhân để điều chỉnh trị số mẫu đích.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, trong đó trị số cắt bớt được xác định nhờ sử dụng chỉ số cắt bớt được quy định bởi thành phần cú pháp và việc ánh xạ giữa các chỉ số cắt bớt và các trị số cắt bớt.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit. Nói cách khác, ít nhất một bit là sự biểu diễn nhị phân của trị số của thành phần cú pháp và trị số của thành phần cú pháp là số nguyên không dấu.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện bất kỳ nêu trên hoặc khía cạnh thứ nhất như vậy, thành phần cú pháp ở mức lát.

Theo khía cạnh thứ hai, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) và/hoặc thông số hệ số ALF; phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy, thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện nêu trên, mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit. Nói cách khác, ít nhất một bit là sự biểu diễn nhị phân của trị số của thành phần cú pháp và trị số của thành phần cú pháp là số nguyên không dấu.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, chính thành phần cú pháp định rõ trị số của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó ít nhất một bit ở dòng bit biểu diễn trị số của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó chỉ số trị số cắt bớt ALF định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF).

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó thông số hệ số ALF được sử dụng để thu nhận hệ số ALF.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp nghĩa là trị số của thành phần cú pháp được định rõ bởi chính thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp ở mức lát.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thông số hệ số ALF được sử dụng để xác định hệ số ALF.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ nhất hoặc khía cạnh thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp là chỉ số trị số cắt bớt ALF, và ít nhất một bit biểu diễn thành phần cú pháp là hai bit.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện nêu trên, chỉ số trị số cắt bớt ALF nhận dạng một trị số cắt bớt trong số bốn trị số cắt bớt.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ nhất hoặc thứ hai như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trị số của chỉ số trị số cắt bớt ALF được sử dụng để xác định khoảng cắt bớt, khoảng cắt bớt được sử dụng trong quy trình lọc vòng lặp thích ứng.

Theo khía cạnh thứ ba, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF); tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy, ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trị số của thành phần cú pháp tương ứng với độ chênh lệch nhỏ nhất (ví dụ, sai số toàn phương trung bình hoặc chi phí tốc độ biến dạng) giữa khối được tái cấu trúc (hoặc khối được lọc) của khối hiện tại và tín hiệu gốc của khối hiện tại, và khối được tái cấu trúc (hoặc khối được lọc) là kết quả sử dụng trị số của thành phần cú pháp, và độ chênh lệch nhỏ nhất thấp hơn độ chênh lệch khác bất kỳ tương ứng với trị số khả thi khác bất kỳ của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó ít nhất một bit ở dòng bit biểu diễn trị số của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó trị số cắt bớt được sử dụng để xác định khoảng cắt bớt được sử dụng để giới hạn (hoặc cắt bớt) độ chênh lệch giữa trị số mẫu đích và trị số mẫu ở gần, và độ chênh lệch trị số mẫu được giới hạn (hoặc độ chênh

lệch trị số mẫu được cắt bớt) được sử dụng để điều chỉnh trị số mẫu đích trong quy trình xử lý của ALF.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit. Nói cách khác, ít nhất một bit là sự biểu diễn nhị phân của trị số của thành phần cú pháp và trị số của thành phần cú pháp là số nguyên không dấu.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp ở mức lát.

Theo khía cạnh thứ tư, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) và/hoặc thông số hệ số bộ lọc ALF; tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện nêu trên, mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit. Nói cách khác, ít nhất một bit là sự biểu diễn nhị phân của trị số của thành phần cú pháp và trị số của thành phần cú pháp là số nguyên không dấu.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trong đó ít nhất một bit ở dòng bit biểu diễn trị số của thành phần cú pháp.



Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp ở mức lát.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thông số hệ số ALF được sử dụng để xác định hệ số ALF.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba hoặc khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, thành phần cú pháp là chỉ số trị số cắt bớt ALF, và ít nhất một bit biểu diễn thành phần cú pháp là hai bit.

Theo dạng thực hiện khả thi của phương pháp theo cách thực hiện nêu trên của khía cạnh thứ tư, chỉ số trị số cắt bớt ALF nhận dạng một trị số cắt bớt trong số bốn trị số cắt bớt.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ ba hoặc khía cạnh thứ tư như vậy hoặc cách thực hiện bất kỳ nêu trên của chúng, trị số của chỉ số trị số cắt bớt ALF được sử dụng để xác định khoảng cắt bớt, khoảng cắt bớt được sử dụng trong quy trình lọc vòng lặp thích ứng.

Theo khía cạnh thứ năm, sáng chế đề xuất bộ giải mã bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ nhất hoặc thứ hai hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ sáu, sáng chế đề xuất bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ ba hoặc thứ tư hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ bảy, sáng chế đề xuất sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh bất kỳ trong số khía cạnh thứ nhất đến khía cạnh thứ tư hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ tám, sáng chế đề xuất phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp của khía cạnh bất kỳ trong số khía cạnh thứ nhất đến khía cạnh thứ tư hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ chín, sáng chế đề xuất bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo khía cạnh thứ nhất hoặc thứ hai hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ mười, sáng chế đề xuất bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp theo khía cạnh thứ ba hoặc thứ tư hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ mười một, sáng chế đề xuất bộ giải mã, bao gồm:

bộ phận giải mã entrôpi, được tạo cấu hình để thu nhận dòng bit trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF);

bộ phận giải mã entrôpi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định; và bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Theo khía cạnh thứ mười hai, sáng chế đề xuất bộ giải mã, bao gồm:

bộ phận giải mã entrôpi, được tạo cấu hình để thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số ALF;

bộ phận giải mã entrôpi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp; và bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Theo khía cạnh thứ mười ba, sáng chế đề xuất bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF); bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

Theo khía cạnh thứ mười bốn, sáng chế đề xuất bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF hoặc thông số hệ số bộ lọc của bộ lọc vòng lặp thích ứng (ALF); bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại.

Theo khía cạnh thứ mười lăm, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó  $n$  bit ở dòng bit biểu diễn thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0;

phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp là sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ mười lăm như vậy, thành phần cú pháp có thể là thành phần cú pháp mức lát.

Theo khía cạnh thứ mười sáu, sáng chế đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0; tạo ra dòng bit bao gồm  $n$  bit, dựa vào trị số của thành phần cú pháp, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp.

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ mười sáu như vậy, thành phần cú pháp có thể là thành phần cú pháp mức lát.

Theo khía cạnh thứ mười bảy, sáng chế đề xuất bộ giải mã, bao gồm:

bộ phận giải mã entropi, được tạo cấu hình để thu nhận dòng bit, trong đó  $n$  bit ở dòng bit biểu diễn thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0; bộ phận giải mã entropi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp là sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit; bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Theo khía cạnh thứ mười tám, sáng chế đề xuất bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0; bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit bao gồm  $n$  bit, dựa vào trị số của thành phần cú pháp, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp.

Theo khía cạnh thứ mười chín, sáng chế đề xuất bộ giải mã bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ mười lăm hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi, sáng chế đề xuất bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ sáu hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi một, sáng chế đề xuất sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh thứ mười lăm hoặc khía cạnh thứ sáu hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi hai, sáng chế đề xuất phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo khía cạnh bất kỳ trong số khía cạnh thứ mười lăm hoặc khía cạnh thứ sáu hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi ba, sáng chế đề xuất bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo khía cạnh thứ mười lăm hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi tư, sáng chế đề xuất bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp theo khía cạnh thứ sáu hoặc cách thực hiện bất kỳ của chúng.

Theo khía cạnh thứ hai mươi năm, sáng chế đề xuất phương tiện lưu trữ không tạm thời bao gồm dòng bit bao gồm  $n$  bit, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp, và thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0.

Theo khía cạnh thứ hai mươi sáu, sáng chế đề xuất phương tiện lưu trữ không tạm thời bao gồm dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định và định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF).

Theo dạng thực hiện khả thi của phương pháp theo khía cạnh thứ hai mươi sáu như vậy, trong đó chính thành phần cú pháp định rõ trị số của thành phần cú pháp.

Theo khía cạnh thứ hai mươi bảy, sáng chế đề xuất phương tiện lưu trữ không tạm thời bao gồm dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số bộ lọc ALF, và ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp.

Theo khía cạnh thứ hai mươi tám, sáng chế đề xuất phương tiện lưu trữ không tạm thời bao gồm dòng bit được mã hóa bởi phương pháp theo khía cạnh bất kỳ hoặc cách thực hiện bất kỳ của chúng.

Chi tiết về một hoặc nhiều phương án được đưa ra trên các hình vẽ kèm theo và phần mô tả dưới đây. Các dấu hiệu, các đối tượng, và các ưu điểm khác sẽ rõ ràng từ phần mô tả, các hình vẽ, và các điểm yêu cầu bảo hộ.

### **Mô tả vắn tắt các hình vẽ**

Trong phần sau đây các phương án của sáng chế được mô tả chi tiết hơn dựa vào các hình vẽ và các hình vẽ minh họa kèm theo, trong đó:

Fig.1A là sơ đồ khối thể hiện ví dụ về hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.1B là sơ đồ khối thể hiện ví dụ khác về hệ thống tạo mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.2 là sơ đồ khối thể hiện ví dụ về bộ mã hóa video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.3 là sơ đồ khối thể hiện cấu trúc ví dụ của bộ giải mã video được tạo cấu hình để thực hiện các phương án của sáng chế;

Fig.4 là sơ đồ khối minh họa ví dụ về máy mã hóa hoặc máy giải mã;

Fig.5 là sơ đồ khối minh họa ví dụ khác về máy mã hóa hoặc máy giải mã;

Fig.6 minh họa các hình dạng bộ lọc ALF, kim cương 5 x 5 chroma, kim cương 7 x 7 luma;

Fig.7 minh họa việc phân loại khối ALF được lấy mẫu con;

Fig.8 minh họa việc báo hiệu các thông số cắt bớt luma và chroma ALF VTM-5.0;

Fig.9 minh họa việc báo hiệu các thông số cắt bớt luma và chroma ALF VTM-5.0 được điều chỉnh, trong đó các thông số cắt bớt được báo hiệu nhờ sử dụng mã độ dài cố định của 2 bit;

Fig.10 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ nhất của sáng chế;

Fig.11 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ hai của sáng chế;

Fig.12 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ ba của sáng chế;

Fig.13 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ tư của sáng chế;

Fig.14 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ năm của sáng chế;

Fig.15 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ sáu của sáng chế;

Fig.16 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ chín của sáng chế;

Fig.17 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười của sáng chế;

Fig.18 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ mười một của sáng chế;

Fig.19 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ mười hai của sáng chế;

Fig.20 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười ba của sáng chế;

Fig.21 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười bốn của sáng chế;

Fig.22 là sơ đồ khối thể hiện cấu trúc ví dụ của hệ thống cung cấp nội dung 3100 mà thực hiện dịch vụ phân phối nội dung; và

Fig.23 là sơ đồ khối thể hiện cấu trúc của ví dụ về thiết bị đầu cuối.

Trong phần sau đây các ký hiệu tham chiếu giống nhau đề cập đến các dấu hiệu giống nhau hoặc ít nhất tương đương về mặt chức năng nếu không được quy định rõ ràng theo cách khác.

### **Mô tả chi tiết sáng chế**

Trong phần mô tả sau đây, tham chiếu được thực hiện tới các hình vẽ kèm theo, mà tạo nên một phần của sáng chế, và thể hiện, bằng cách minh họa, các khía cạnh cụ thể của các phương án của sáng chế hoặc các khía cạnh cụ thể mà trong đó các phương án của sáng chế có thể được sử dụng. Cần hiểu rằng các phương án của sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các sự thay đổi cấu trúc hoặc logic không được mô tả trên các hình vẽ. Phần mô tả chi tiết sau đây, do đó, không được coi là giới hạn, và phạm vi của sáng chế được định rõ bởi các điểm yêu cầu bảo hộ kèm theo.

Chẳng hạn, cần 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 một hoặc các bước về phương pháp cụ thể được mô tả, thì thiết bị tương ứng có thể bao gồm một hoặc các bộ phận, ví dụ, các bộ phận chức năng, để thực hiện một hoặc các bước về phương pháp được mô tả (ví dụ, một bộ phận thực hiện một hoặc các bước, hoặc các bộ phận đều thực hiện một hoặc nhiều trong số các bước), ngay cả khi một hoặc nhiều bộ phận như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Mặt khác, ví dụ, nếu máy cụ thể được mô

tả dựa vào một hoặc các bộ phận, ví dụ, các bộ phận chức năng, thì phương pháp tương ứng có thể bao gồm một bước để thực hiện tính năng của một hoặc các bộ phận (ví dụ, một bước thực hiện tính năng của một hoặc các bộ phận, hoặc các bước đều thực hiện tính năng của một hoặc nhiều trong số các bộ phận), ngay cả khi một hoặc các bước như vậy không được mô tả hoặc được minh họa một cách rõ ràng trên các hình vẽ. Hơn nữa, cần hiểu rằng các dấu hiệu của các phương án và/hoặc các khía cạnh làm ví dụ khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi được lưu ý cụ thể theo cách khác.

Tạo mã video thường đề cập đến việc xử lý của chuỗi các ảnh, mà tạo nên video hoặc chuỗi video. Thay vì thuật ngữ “ảnh” thuật ngữ “khung” hoặc “hình ảnh” có thể được sử dụng như các từ đồng nghĩa trong lĩnh vực về tạo mã video. Tạo mã video (hoặc tạo mã nói chung) bao gồm hai phần mã hóa video và giải mã video. Mã hóa video được thực hiện ở phía nguồn, thường bao gồm việc xử lý (ví dụ, bằng cách nén) các ảnh video gốc để làm giảm lượng dữ liệu được yêu cầu để biểu diễn các ảnh video (đối với việc lưu trữ và/hoặc truyền hiệu quả hơn). Giải mã video được thực hiện ở phía đích và thường bao gồm việc xử lý ngược so với bộ mã hóa để tái cấu trúc các ảnh video. Các phương án đề cập đến việc “tạo mã” của các ảnh video (hoặc các ảnh nói chung) sẽ được hiểu liên quan đến việc “mã hóa” hoặc “giải mã” của các ảnh video hoặc các chuỗi video tương ứng. Sự kết hợp của phần mã hóa và phần giải mã cũng được gọi là CODEC (bộ tạo mã/giải mã) (tạo mã và giải mã).

Trong trường hợp tạo mã video mất mát, các ảnh video gốc có thể được tái cấu trúc, nghĩa là, các ảnh video được tái cấu trúc có cùng chất lượng như các ảnh video gốc (giả sử không có sự mất mát truyền hoặc sự mất mát dữ liệu khác trong suốt thời gian lưu trữ hoặc truyền). Trong trường hợp tạo mã video mất mát, việc nén thêm, ví dụ, bởi việc lượng tử hóa, được thực hiện, để làm giảm lượng dữ liệu biểu diễn các ảnh video, mà không thể được tái cấu trúc hoàn toàn ở bộ giải mã, nghĩa là, chất lượng của các ảnh video được tái cấu trúc thấp hơn hoặc kém hơn so với chất lượng của các ảnh video gốc.

Nhiều chuẩn tạo mã video thuộc về nhóm của “các bộ mã hóa/giải mã video lai mất mát” (nghĩa là, kết hợp việc dự đoán không gian và thời gian trong miền mẫu và việc tạo mã biến đổi 2D để áp dụng việc lượng tử hóa trong miền biến đổi). Mỗi ảnh của chuỗi video thường được phân chia thành tập hợp của các khối không chồng lấn, và



việc tạo mã thường được thực hiện trên mức khối. Nói cách khác, ở bộ mã hóa video thường được xử lý, nghĩa là, được mã hóa, trên mức khối (khối video), ví dụ, nhờ sử dụng việc dự đoán không gian (ảnh bên trong) và/hoặc việc dự đoán thời gian (ảnh liên kết) để tạo ra khối dự đoán, trừ khối dự đoán khối khối hiện tại (khối được xử lý hiện tại/cần được xử lý) để thu nhận khối dư, biến đổi khối dư và lượng tử hóa khối dư trong miền biến đổi để làm giảm lượng dữ liệu được truyền (nén), trong khi ở bộ giải mã quy trình xử lý ngược so với bộ mã hóa được áp dụng cho khối được mã hóa hoặc được nén để tái cấu trúc khối hiện tại nhằm biểu diễn. Hơn nữa, bộ mã hóa nhân đôi vòng lặp xử lý bộ giải mã sao cho cả hai sẽ tạo ra các việc dự đoán giống nhau (ví dụ, các việc nội và liên dự đoán) và/hoặc các việc tái cấu trúc để xử lý, ví dụ, tạo mã, các khối tiếp theo.

Trong phần sau đây các phương án về hệ thống tạo mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa vào Fig.1 đến Fig.3.

Fig.1A là sơ đồ khối giản lược minh họa hệ thống tạo mã 10 ví dụ, ví dụ, hệ thống tạo mã video 10 (hoặc gọi tắt là hệ thống tạo mã 10) mà có thể sử dụng các kỹ thuật theo sáng chế. Bộ mã hóa video 20 (hoặc gọi tắt là bộ mã hóa 20) và bộ giải mã video 30 (hoặc gọi tắt là bộ giải mã 30) của hệ thống tạo mã video 10 biểu diễn các ví dụ về các thiết bị mà có thể được tạo cấu hình để thực hiện các kỹ thuật phù hợp với các ví dụ khác nhau được mô tả theo sáng chế.

Như được thể hiện trên Fig.1A, hệ thống tạo mã 10 bao gồm thiết bị nguồn 12 được tạo cấu hình để cung cấp dữ liệu ảnh được mã hóa 21 ví dụ, tới thiết bị đích 14 để giải mã dữ liệu ảnh được mã hóa 13.

Thiết bị nguồn 12 bao gồm bộ mã hóa 20, và có thể bổ sung, nghĩa là, một cách tùy chọn, bao gồm nguồn ảnh 16, bộ xử lý trước (hoặc bộ phận xử lý trước) 18, ví dụ, bộ phận xử lý trước ảnh 18, và giao diện truyền thông hoặc bộ phận truyền thông 22.

Nguồn ảnh 16 có thể bao gồm hoặc là loại thiết bị chụp ảnh bất kỳ, ví dụ camera để chụp ảnh thế giới thực, và/hoặc loại thiết bị tạo ảnh bất kỳ, ví dụ bộ xử lý đồ họa máy tính để tạo ra ảnh hoạt hình máy tính, hoặc loại thiết bị khác bất kỳ để thu nhận và/hoặc cung cấp ảnh thế giới thực, ảnh được tạo ra bởi máy tính (ví dụ, nội dung màn hình, ảnh thực tế ảo (virtual reality, viết tắt là VR)) và/hoặc sự kết hợp bất kỳ của chúng (ví dụ, ảnh thực tế tăng cường (augmented reality, viết tắt là AR)). Nguồn ảnh có thể là loại bộ nhớ bất kỳ hoặc thiết bị lưu trữ bất kỳ trong số các ảnh nêu trên.

Phân biệt với bộ xử lý trước 18 và quy trình xử lý được thực hiện bởi bộ phận xử lý trước 18, ảnh hoặc dữ liệu ảnh 17 cũng có thể được gọi là ảnh thô hoặc dữ liệu ảnh thô 17.

Bộ xử lý trước 18 được tạo cấu hình để thu dữ liệu ảnh (thô) 17 và để thực hiện việc xử lý trước trên dữ liệu ảnh 17 để thu nhận ảnh được xử lý trước 19 hoặc dữ liệu ảnh được xử lý trước 19. Việc xử lý trước được thực hiện bởi bộ xử lý trước 18 có thể, ví dụ, bao gồm việc sắp xếp lại, chuyển đổi định dạng màu (ví dụ, từ RGB đến YCbCr), điều chỉnh màu sắc, hoặc khử nhiễu. Có thể hiểu rằng bộ phận xử lý trước 18 có thể là bộ phận tùy chọn.

Bộ mã hóa video 20 được tạo cấu hình để thu dữ liệu ảnh được xử lý trước 19 và cung cấp dữ liệu ảnh được mã hóa 21 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 và để truyền dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng) qua kênh truyền thông 13 tới thiết bị khác, ví dụ, thiết bị đích 14 hoặc thiết bị khác bất kỳ, nhằm lưu trữ hoặc tái cấu trúc trực tiếp.

Thiết bị đích 14 bao gồm bộ giải mã 30 (ví dụ, bộ giải mã video 30), và có thể bao gồm bổ sung, nghĩa là, một cách tùy chọn, giao diện truyền thông hoặc bộ phận truyền thông 28, bộ xử lý sau 32 (hoặc bộ phận xử lý sau 32) và thiết bị hiển thị 34.

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm bất kỳ của chúng), ví dụ, trực tiếp từ thiết bị nguồn 12 hoặc từ nguồn khác bất kỳ, ví dụ, thiết bị lưu trữ, ví dụ, thiết bị lưu trữ dữ liệu ảnh được mã hóa, và cung cấp dữ liệu ảnh được mã hóa 21 tới bộ giải mã 30.

Giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình để truyền hoặc thu dữ liệu ảnh được mã hóa 21 hoặc dữ liệu được mã hóa 13 qua liên kết truyền thông trực tiếp giữa thiết bị nguồn 12 và thiết bị đích 14, ví dụ, sự kết nối nối dây hoặc không dây trực tiếp, hoặc qua loại mạng bất kỳ, ví dụ, mạng nối dây hoặc không dây hoặc sự kết hợp bất kỳ của chúng, hoặc loại mạng riêng và công cộng bất kỳ, hoặc loại kết hợp bất kỳ của chúng.

Giao diện truyền thông 22 có thể, ví dụ, được tạo cấu hình để đóng gói dữ liệu ảnh được mã hóa 21 thành định dạng thích hợp, ví dụ, các gói, và/hoặc xử lý dữ liệu ảnh được mã hóa nhờ sử dụng loại mã hóa hoặc xử lý truyền bất kỳ đối với việc truyền qua liên kết truyền thông hoặc mạng truyền thông.

Giao diện truyền thông 28, tạo nên bản sao của giao diện truyền thông 22, có thể, ví dụ, được tạo cấu hình để thu dữ liệu được truyền và xử lý dữ liệu truyền nhờ sử dụng loại giải mã hoặc xử lý và/hoặc mở gói truyền tương ứng bất kỳ để thu nhận dữ liệu ảnh được mã hóa 21.

Cả hai, giao diện truyền thông 22 và giao diện truyền thông 28 có thể được tạo cấu hình là các giao diện truyền thông một chiều như được chỉ báo bởi mũi tên cho kênh truyền thông 13 trên Fig.1A chỉ từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai chiều, và có thể được tạo cấu hình, ví dụ, để gửi và thu các tin nhắn, ví dụ, để thiết lập sự kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc việc truyền dữ liệu, ví dụ, việc truyền dữ liệu ảnh được mã hóa.

Bộ giải mã 30 được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 và cung cấp dữ liệu ảnh được giải mã 31 hoặc ảnh được giải mã 31 (chi tiết thêm sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.3 hoặc Fig.5).

Bộ xử lý sau 32 của thiết bị đích 14 được tạo cấu hình để xử lý sau dữ liệu ảnh được giải mã 31 (cũng được gọi là dữ liệu ảnh được tái cấu trúc), ví dụ, ảnh được giải mã 31, để thu nhận dữ liệu ảnh được xử lý sau 33, ví dụ, ảnh được xử lý sau 33. Quy trình xử lý sau được thực hiện bởi bộ phận xử lý sau 32 có thể bao gồm, ví dụ, việc chuyển đổi định dạng màu (ví dụ, từ YCbCr đến RGB), điều chỉnh màu sắc, sắp xếp lại, hoặc lấy mẫu lại, hoặc quy trình xử lý khác bất kỳ, ví dụ, để chuẩn bị dữ liệu ảnh được giải mã 31 nhằm hiển thị, ví dụ, bởi thiết bị hiển thị 34.

Thiết bị hiển thị 34 của thiết bị đích 14 được tạo cấu hình để thu dữ liệu ảnh được xử lý sau 33 để hiển thị ảnh, ví dụ, tới người dùng hoặc người xem. Thiết bị hiển thị 34 có thể là hoặc bao gồm loại màn hình bất kỳ để biểu diễn ảnh được tái cấu trúc, ví dụ, màn hình hoặc màn hiển thị bên ngoài hoặc được tích hợp. Các màn hình có thể, ví dụ, bao gồm các màn hình tinh thể lỏng (liquid crystal display, viết tắt là LCD), các màn hình điốt phát sáng hữu cơ (organic light emitting diodes, viết tắt là OLED), các màn

hình plasma, các máy chiếu, các màn hình LED nhỏ, tinh thể lỏng trên silicon (liquid crystal on silicon, viết tắt là LCoS), bộ xử lý ánh sáng kỹ thuật số (digital light processor, viết tắt là DLP) hoặc loại màn hình khác bất kỳ.

Mặc dù Fig.1A mô tả thiết bị nguồn 12 và thiết bị đích 14 là các thiết bị riêng biệt, nhưng các phương án về các thiết bị cũng có thể bao gồm cả hai hoặc cả hai tính năng, thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng. Theo các phương án như vậy thiết bị nguồn 12 hoặc tính năng tương ứng và thiết bị đích 14 hoặc tính năng tương ứng có thể được thực hiện nhờ sử dụng cùng phần cứng và/hoặc phần mềm hoặc bởi phần cứng và/hoặc phần mềm riêng biệt hoặc sự kết hợp bất kỳ của chúng.

Như sẽ rõ ràng với người có hiểu biết dựa vào phần mô tả, sự tồn tại và sự phân tách (chính xác) các tính năng của các bộ phận khác nhau hoặc các tính năng nằm trong thiết bị nguồn 12 và/hoặc thiết bị đích 14 như được thể hiện trên Fig.1A có thể thay đổi tùy thuộc vào thiết bị và ứng dụng thực tế.

Bộ mã hóa 20 (ví dụ, bộ mã hóa video 20) và bộ giải mã 30 (ví dụ, bộ giải mã video 30) hoặc cả bộ mã hóa 20 và bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý như được thể hiện trên Fig.1B, chẳng hạn như một hoặc nhiều bộ vi xử lý, các bộ xử lý tín hiệu kỹ thuật số (digital signal processor, viết tắt là DSP), các mạch tích hợp chuyên dụng (application-specific integrated circuit, viết tắt là ASIC), các mảng cổng lập trình được dạng trường (field-programmable gate array, viết tắt là FPGA), logic rời rạc, phần cứng, hoặc các sự kết hợp bất kỳ của chúng. Bộ mã hóa 20 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống con bộ mã hóa khác bất kỳ được mô tả ở đây. Bộ giải mã 30 có thể được thực hiện qua hệ mạch xử lý 46 để bao gồm các môđun khác nhau như được thảo luận dựa vào bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống con bộ giải mã khác bất kỳ được mô tả ở đây. Hệ mạch xử lý có thể được tạo cấu hình để thực hiện các thao tác khác nhau như được thảo luận dưới đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được thực hiện một phần ở phần mềm, thì thiết bị có thể lưu trữ các lệnh cho phần mềm trong phương tiện lưu trữ có thể đọc được bởi máy tính không tạm thời, thích hợp và có thể thực hiện các lệnh ở phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật theo sáng chế. Một trong số

bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp như phần của bộ mã hóa/bộ giải mã (CODEC) được kết hợp trong thiết bị đơn, ví dụ, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm bất kỳ trong số khoảng rộng của các thiết bị, bao gồm loại bất kỳ của các thiết bị cầm tay hoặc văn phòng, ví dụ, các máy tính cầm tay hoặc xách tay, các điện thoại di động, các điện thoại thông minh, các máy tính bảng (tablet hoặc tablet computer), các camera, các máy tính để bàn, các thiết bị giải mã tín hiệu (set-top box), các máy thu hình, các thiết bị hiển thị, các trình phát phương tiện kỹ thuật số, các máy chơi trò chơi video, các thiết bị tạo dòng video (chẳng hạn như các máy chủ dịch vụ nội dung hoặc các máy chủ phân bổ nội dung), thiết bị thu phát sóng, thiết bị truyền phát sóng, hoặc tương tự và có thể sử dụng không hoặc loại hệ điều hành bất kỳ. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị cho việc truyền thông không dây. Vì vậy, thiết bị nguồn 12 và thiết bị đích 14 có thể là các thiết bị truyền thông không dây.

Trong một số trường hợp, hệ thống tạo mã video 10 được minh họa trên Fig.1A chỉ là ví dụ và các kỹ thuật theo sáng chế có thể áp dụng cho các sự thiết đặt tạo mã video (ví dụ, mã hóa video hoặc giải mã video) mà không cần thiết bao gồm việc truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Theo các ví dụ khác, dữ liệu được truy xuất từ bộ nhớ cục bộ, được tạo dòng qua mạng, hoặc tương tự. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu tới bộ nhớ, và/hoặc thiết bị giải mã video có thể truy xuất và giải mã dữ liệu từ bộ nhớ. Theo một số ví dụ, việc mã hóa và giải mã được thực hiện bởi các thiết bị mà không truyền thông với thiết bị khác, mà mã hóa một cách đơn giản dữ liệu tới bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Nhằm thuận tiện cho việc mô tả, các phương án của sáng chế được mô tả ở đây, ví dụ, bằng cách tham chiếu tới việc tạo mã video hiệu quả cao (High-Efficiency Video Coding, viết tắt là HEVC) hoặc phần mềm tham chiếu của việc tạo mã video đa năng (Versatile Video Coding, viết tắt là VVC), chuẩn tạo mã video thế hệ tiếp theo được phát triển bởi nhóm hợp tác tạo mã video liên kết (Joint Collaboration Team on Video Coding, viết tắt là JCT-VC) của nhóm các chuyên gia tạo mã video (Video Coding Experts Group, viết tắt là VCEG) ITU-T và nhóm các chuyên gia ảnh động (Motion Picture Experts Group, viết tắt là MPEG) ISO/IEC. Người có hiểu biết trung bình trong

lĩnh vực kỹ thuật tương ứng sẽ hiểu rằng các phương án của sáng chế không giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

Fig.2 thể hiện sơ đồ khối giản lược của bộ mã hóa video 20 ví dụ mà được tạo cấu hình để thực hiện các kỹ thuật theo sáng chế. Theo ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đầu vào 201 (hoặc giao diện đầu vào 201), bộ phận tính toán phần dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lượng tử hóa ngược 210, và bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ phận lọc vòng lặp 220, bộ đệm ảnh được giải mã (decoded picture buffer, viết tắt là DPB) 230, bộ phận lựa chọn chế độ 260, bộ phận mã hóa entropi 270 và đầu ra 272 (hoặc giao diện đầu ra 272). Bộ phận lựa chọn chế độ 260 có thể bao gồm bộ phận liên dự đoán 244, bộ phận nội dự đoán 254 và bộ phận phân chia 262. Bộ phận liên dự đoán 244 có thể bao gồm bộ phận đánh giá chuyển động và bộ phận bù chuyển động (không được thể hiện). Bộ mã hóa video 20 như được thể hiện trên Fig.2 cũng có thể được gọi là bộ mã hóa video lai hoặc bộ mã hóa video theo bộ mã hóa/giải mã video lai.

Bộ phận tính toán phần dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận lựa chọn chế độ 260 có thể được đề cập đến là tạo nên đường báo hiệu xuôi của bộ mã hóa 20, trong khi bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ đệm 216, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254 có thể được đề cập đến là tạo nên đường báo hiệu ngược của bộ mã hóa video 20, trong đó đường báo hiệu ngược của bộ mã hóa video 20 tương ứng với đường báo hiệu của bộ giải mã (xem bộ giải mã video 30 trên Fig.3). Bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254 cũng được đề cập là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20.

Các ảnh & việc phân chia ảnh (các ảnh & các khối)

Bộ mã hóa 20 có thể được tạo cấu hình để thu, ví dụ, qua đầu vào 201, ảnh 17 (hoặc dữ liệu ảnh 17), ví dụ, ảnh của chuỗi các ảnh tạo nên video hoặc chuỗi video. Ảnh hoặc dữ liệu ảnh thu được cũng có thể là ảnh được xử lý trước 19 (hoặc dữ liệu ảnh được xử lý trước 19). Nhằm đơn giản hóa phần mô tả sau đây đề cập đến ảnh 17. Ảnh 17 cũng

có thể được gọi là ảnh hiện tại hoặc ảnh được tạo mã (cụ thể trong việc tạo mã video để phân biệt ảnh hiện tại với các ảnh khác, ví dụ, các ảnh được mã hóa và/hoặc được giải mã trước đó của cùng chuỗi video, nghĩa là, chuỗi video mà cũng bao gồm ảnh hiện tại).

Ảnh (kỹ thuật số) là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ. Mẫu trong mảng cũng có thể được gọi là điểm ảnh (dạng ngắn của thành phần ảnh) hoặc pel. Số lượng các mẫu theo hướng chiều ngang và chiều dọc (hoặc trục) của mảng hoặc ảnh định rõ kích thước và/hoặc độ phân giải của ảnh. Nhằm biểu diễn màu sắc, thường ba thành phần màu sắc được sử dụng, nghĩa là, ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Theo định dạng RGB hoặc không gian màu ảnh bao gồm mảng mẫu đỏ, xanh lá cây và xanh lam tương ứng. Tuy nhiên, trong tạo mã video mỗi điểm ảnh thường được biểu diễn theo định dạng độ chói và sắc độ hoặc không gian màu, ví dụ, YCbCr, mà bao gồm thành phần độ chói được chỉ báo bởi Y (thỉnh thoảng L cũng được sử dụng thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần độ chói (hoặc viết tắt là luma) Y biểu diễn độ sáng hoặc cường độ mức xám (ví dụ, như trong ảnh thước xám), trong khi hai thành phần sắc độ (hoặc viết tắt là chroma) Cb và Cr biểu diễn sắc độ hoặc các thành phần thông tin màu sắc. Theo đó, ảnh theo định dạng YCbCr bao gồm mảng mẫu độ chói của các trị số mẫu độ chói (Y), và hai mảng mẫu sắc độ của các trị số sắc độ (Cb và Cr). Các ảnh theo định dạng RGB có thể được chuyển đổi hoặc được biến đổi thành định dạng YCbCr và ngược lại, quy trình xử lý cũng được biết đến là biến đổi hoặc chuyển đổi màu sắc. Nếu ảnh là đơn sắc, thì ảnh có thể bao gồm chỉ mảng mẫu độ chói. Theo đó, ảnh có thể là, ví dụ, mảng của các mẫu luma theo định dạng đơn sắc hoặc mảng của các mẫu luma và hai mảng tương ứng của các mẫu chroma theo định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Các phương án về bộ mã hóa video 20 có thể bao gồm bộ phận phân chia ảnh (không được mô tả trên Fig.2) được tạo cấu hình để phân chia ảnh 17 thành các khối ảnh 203 (thường không chẵn lẻ). Các khối này cũng có thể được gọi là các khối gốc, các khối lớn (H.264/AVC) hoặc các khối cây tạo mã (coding tree block, viết tắt là CTB) hoặc các bộ phận cây tạo mã (coding tree unit, viết tắt là CTU) (H.265/HEVC và VVC). Bộ phận phân chia ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối cho tất cả các ảnh của chuỗi video và mạng lưới tương ứng định rõ kích thước khối, hoặc để

thay đổi kích thước khối giữa các ảnh hoặc các tập hợp con hoặc các nhóm của các ảnh, và phân chia mỗi ảnh thành các khối tương ứng.

Theo các phương án khác, bộ mã hóa video có thể được tạo cấu hình để thu trực tiếp khối 203 của ảnh 17, ví dụ, một, nhiều hoặc tất cả các khối tạo nên ảnh 17. Khối ảnh 203 cũng có thể được gọi là khối ảnh hiện tại hoặc khối ảnh được tạo mã.

Giống như ảnh 17, khối ảnh 203 lại là hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các trị số cường độ (các trị số mẫu), mặc dù kích thước nhỏ hơn ảnh 17. Nói cách khác, khối 203 có thể bao gồm, ví dụ, một mảng mẫu (ví dụ, mảng luma trong trường hợp ảnh đơn sắc 17, hoặc mảng luma hoặc chroma trong trường hợp ảnh màu) hoặc ba mảng mẫu (ví dụ, luma và hai mảng chroma trong trường hợp ảnh màu 17) hoặc số lượng và/hoặc loại khác bất kỳ của các mảng tùy thuộc vào định dạng màu được áp dụng. Số lượng các mẫu theo hướng chiều ngang và chiều dọc (hoặc trục) của khối 203 định rõ kích thước của khối 203. Theo đó, khối có thể là, ví dụ, mảng  $M \times N$  (cột  $M$  nhân hàng  $N$ ) của các mẫu, hoặc mảng  $M \times N$  của các hệ số biến đổi.

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể được tạo cấu hình để mã hóa ảnh 17 theo từng khối, ví dụ, việc mã hóa và dự đoán được thực hiện trên mỗi khối 203.

#### Tính toán phần dư

Bộ phận tính toán phần dư 204 có thể được tạo cấu hình để tính toán khối dư 205 (cũng được gọi là phần dư 205) dựa vào khối ảnh 203 và khối dự đoán 265 (chi tiết thêm về khối dự đoán 265 được đưa ra dưới đây), ví dụ, bằng cách trừ các trị số mẫu của khối dự đoán 265 khỏi các trị số mẫu của khối ảnh 203, theo từng mẫu (theo từng điểm ảnh) để thu nhận khối dư 205 trong miền mẫu.

#### Biến đổi

Bộ phận xử lý biến đổi 206 được tạo cấu hình để áp dụng việc biến đổi, ví dụ, biến đổi côsin rời rạc (discrete cosine transform, viết tắt là DCT) hoặc biến đổi sin rời rạc (discrete sine transform, viết tắt là DST), trên các trị số mẫu của khối dư 205 để thu nhận các hệ số biến đổi 207 trong miền biến đổi. Các hệ số biến đổi 207 cũng có thể được gọi là các hệ số dư biến đổi và biểu diễn khối dư 205 trong miền biến đổi.

Bộ phận xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng các phép xấp xỉ số nguyên của DCT/DST, chẳng hạn như các sự biến đổi được quy định cho HEVC/H.265.



So với việc biến đổi DCT trực giao, các phép xấp xỉ số nguyên như vậy thường được định tỷ lệ bởi hệ số nhất định. Để giữ lại định mức của khối dư mà được xử lý bởi các sự biến đổi xuôi và ngược, các hệ số định tỷ lệ bổ sung được áp dụng như một phần của quy trình biến đổi. Các hệ số định tỷ lệ thường được chọn dựa vào các sự ràng buộc nhất định như các hệ số định tỷ lệ là lũy thừa của hai đối với các phép toán dịch chuyển, độ sâu bit của các hệ số biến đổi, sự cân bằng giữa độ chính xác và các chi phí thực hiện, v.v. Các hệ số định tỷ lệ cụ thể, ví dụ, được quy định đối với việc biến đổi ngược, ví dụ, bởi bộ phận xử lý biến đổi ngược 212 (và việc biến đổi ngược tương ứng, ví dụ, bởi bộ phận xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các hệ số định tỷ lệ tương ứng đối với việc biến đổi xuôi, ví dụ, bởi bộ phận xử lý biến đổi 206, ở bộ mã hóa 20 có thể được quy định theo đó.

Các phương án về bộ mã hóa video 20 (bộ phận xử lý biến đổi 206 tương ứng) có thể được tạo cấu hình để đưa ra các thông số biến đổi, ví dụ, loại biến đổi hoặc các phép biến đổi, ví dụ, trực tiếp hoặc được mã hóa hoặc được nén qua bộ phận mã hóa entrôpi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số biến đổi để giải mã.

#### Lượng tử hóa

Bộ phận lượng tử hóa 208 có thể được tạo cấu hình để lượng tử hóa các hệ số biến đổi 207 để thu nhận các hệ số được lượng tử hóa 209, ví dụ, bằng cách áp dụng việc lượng tử hóa vô hướng hoặc việc lượng tử hóa vectơ. Các hệ số được lượng tử hóa 209 cũng có thể được gọi là các hệ số biến đổi được lượng tử hóa 209 hoặc các hệ số dư được lượng tử hóa 209.

Quy trình lượng tử hóa có thể làm giảm độ sâu bit được kết hợp với một số hoặc tất cả các hệ số biến đổi 207. Ví dụ, hệ số biến đổi  $n$  bit có thể được làm tròn xuống tới hệ số biến đổi  $m$  bit trong suốt thời gian lượng tử hóa, trong đó  $n$  lớn hơn  $m$ . Mức lượng tử hóa có thể được điều chỉnh bằng cách điều chỉnh thông số lượng tử hóa (quantization parameter, viết tắt là QP). Ví dụ đối với việc lượng tử hóa vô hướng, việc định tỷ lệ khác có thể được áp dụng để đạt được việc lượng tử hóa mịn hơn hoặc thô hơn. Các kích thước bước lượng tử hóa nhỏ hơn tương ứng với việc lượng tử hóa mịn hơn, trong khi các kích thước bước lượng tử hóa lớn hơn tương ứng với việc lượng tử hóa thô hơn. Kích thước bước lượng tử hóa có thể áp dụng được có thể được chỉ báo bởi thông số

lượng tử hóa (QP). Thông số lượng tử hóa có thể ví dụ là chỉ số tới tập hợp được định rõ trước của các kích thước bước lượng tử hóa có thể áp dụng được. Ví dụ, các thông số lượng tử hóa nhỏ có thể tương ứng với việc lượng tử hóa mịn (các kích thước bước lượng tử hóa nhỏ) và các thông số lượng tử hóa lớn có thể tương ứng với việc lượng tử hóa thô (các kích thước bước lượng tử hóa lớn) hoặc ngược lại. Lượng tử hóa có thể bao gồm phép chia cho kích thước bước lượng tử hóa và tương ứng và/hoặc việc giải lượng tử hóa ngược, ví dụ, bởi bộ phận lượng tử hóa ngược 210, có thể bao gồm phép nhân với kích thước bước lượng tử hóa. Các phương án theo một số chuẩn, ví dụ, HEVC, có thể được tạo cấu hình để sử dụng thông số lượng tử hóa để xác định kích thước bước lượng tử hóa. Nói chung, kích thước bước lượng tử hóa có thể được tính toán dựa vào thông số lượng tử hóa nhờ sử dụng phép xấp xỉ điểm cố định của phương trình bao gồm phép chia. Các hệ số định tỷ lệ bổ sung có thể được giới thiệu cho việc lượng tử hóa và giải lượng tử hóa để khôi phục tiêu chuẩn của khối dư, mà có thể được điều chỉnh bởi vì việc định tỷ lệ được sử dụng trong phép xấp xỉ điểm cố định của phương trình đối với kích thước bước lượng tử hóa và thông số lượng tử hóa. Theo một cách thực hiện ví dụ, việc định tỷ lệ của việc biến đổi ngược và giải lượng tử hóa có thể được kết hợp. Theo cách khác, các bảng lượng tử hóa được tùy chỉnh có thể được sử dụng và được báo hiệu từ bộ mã hóa tới bộ giải mã, ví dụ, ở dòng bit. Lượng tử hóa là phép toán mất mát, trong đó sự mất mát tăng lên với các kích thước bước lượng tử hóa tăng.

Các phương án về bộ mã hóa video 20 (bộ phận lượng tử hóa 208 tương ứng) có thể được tạo cấu hình để đưa ra các thông số lượng tử hóa (QP), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể thu và áp dụng các thông số lượng tử hóa để giải mã.

#### Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 210 được tạo cấu hình để áp dụng việc lượng tử hóa ngược của bộ phận lượng tử hóa 208 trên các hệ số được lượng tử hóa để thu nhận các hệ số được giải lượng tử hóa 211, ví dụ, bằng cách áp dụng ngược sơ đồ lượng tử hóa được áp dụng bởi bộ phận lượng tử hóa 208 dựa vào hoặc nhờ sử dụng kích thước bước lượng tử hóa giống như bộ phận lượng tử hóa 208. Các hệ số được giải lượng tử hóa 211 cũng có thể được gọi là các hệ số dư được giải lượng tử hóa 211 và tương ứng -

mặc dù thường không giống với các hệ số biến đổi do sự mất mát bởi việc lượng tử hóa – với các hệ số biến đổi 207.

#### Biến đổi ngược

Bộ phận xử lý biến đổi ngược 212 được tạo cấu hình để áp dụng việc biến đổi ngược của việc biến đổi được áp dụng bởi bộ phận xử lý biến đổi 206, ví dụ, biến đổi cosin rời rạc (DCT) ngược hoặc biến đổi sin rời rạc (DST) ngược hoặc các phép biến đổi ngược khác, để thu nhận khối dư được tái cấu trúc 213 (hoặc các hệ số được giải lượng tử hóa 213 tương ứng) trong miền mẫu. Khối dư được tái cấu trúc 213 cũng có thể được gọi là khối biến đổi 213.

#### Tái cấu trúc

Bộ phận tái cấu trúc 214 (ví dụ, bộ bổ sung hoặc bộ cộng 214) được tạo cấu hình để bổ sung khối biến đổi 213 (nghĩa là, khối dư được tái cấu trúc 213) vào khối dự đoán 265 để thu nhận khối được tái cấu trúc 215 trong miền mẫu, ví dụ, bằng cách bổ sung – theo từng mẫu - các trị số mẫu của khối dư được tái cấu trúc 213 và các trị số mẫu của khối dự đoán 265.

#### Lọc

Bộ phận lọc vòng lặp 220 (hoặc viết tắt là “bộ lọc vòng lặp” 220), được tạo cấu hình để lọc khối được tái cấu trúc 215 để thu nhận khối được lọc 221, hoặc nói chung, để lọc các mẫu được tái cấu trúc để thu nhận các mẫu được lọc. Bộ phận lọc vòng lặp, ví dụ, được tạo cấu hình để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận lọc vòng lặp 220 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc giải khối, bộ lọc dịch vị thích nghi mẫu (sample-adaptive offset, viết tắt là SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc hai chiều, bộ lọc vòng lặp thích ứng (ALF), làm sắc, các bộ lọc làm mịn hoặc các bộ lọc phối hợp, hoặc sự kết hợp bất kỳ của chúng. Mặc dù bộ phận lọc vòng lặp 220 được thể hiện trên Fig.2 là bộ lọc trong vòng lặp, theo các cấu hình khác, bộ phận lọc vòng lặp 220 có thể được thực hiện như bộ lọc sau vòng lặp. Khối được lọc 221 cũng có thể được gọi là khối được tái cấu trúc được lọc 221.

Các phương án về bộ mã hóa video 20 (bộ phận lọc vòng lặp 220 tương ứng) có thể được tạo cấu hình để đưa ra các thông số bộ lọc vòng lặp (chẳng hạn như thông tin dịch vị thích nghi mẫu), ví dụ, trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270,

sao cho, ví dụ, bộ giải mã 30 có thể thu và áp dụng các thông số bộ lọc vòng lặp như nhau hoặc các bộ lọc vòng lặp tương ứng để giải mã.

Bộ đệm ảnh được giải mã

Bộ đệm ảnh được giải mã (DPB) 230 có thể là bộ nhớ mà lưu trữ các ảnh tham chiếu, hoặc dữ liệu ảnh tham chiếu nói chung, để mã hóa dữ liệu video bởi bộ mã hóa video 20. DPB 230 có thể được tạo nên bởi bất kỳ trong số các thiết bị nhớ, chẳng hạn như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory, viết tắt là DRAM), bao gồm DRAM đồng bộ (synchronous DRAM, viết tắt là SDRAM), RAM điện từ (magnetoresistive RAM, viết tắt là MRAM), RAM điện trở (resistive RAM, viết tắt là RRAM), hoặc các loại khác của các thiết bị nhớ. Bộ đệm ảnh được giải mã (DPB) 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. Bộ đệm ảnh được giải mã 230 còn có thể được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, ví dụ, các khối được lọc và được tái cấu trúc trước đó 221, của cùng ảnh hiện tại hoặc của các ảnh khác nhau, ví dụ, các ảnh được tái cấu trúc trước đó, và có thể cung cấp các ảnh được tái cấu trúc, nghĩa là, được giải mã trước đó hoàn toàn (và các khối và các mẫu tham chiếu tương ứng) và/hoặc một phần ảnh hiện tại được tái cấu trúc (và các khối và các mẫu tham chiếu tương ứng), ví dụ đối với việc liên dự đoán. Bộ đệm ảnh được giải mã (DPB) 230 cũng có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái cấu trúc không được lọc 215, hoặc các mẫu được tái cấu trúc không được lọc nói chung, ví dụ, nếu khối được tái cấu trúc 215 không được lọc bởi bộ phận lọc vòng lặp 220, hoặc phiên bản được xử lý thêm khác bất kỳ của các khối hoặc các mẫu được tái cấu trúc.

Lựa chọn chế độ (phân chia & dự đoán)

Bộ phận lựa chọn chế độ 260 bao gồm bộ phận phân chia 262, bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254, và được tạo cấu hình để thu hoặc thu nhận dữ liệu ảnh gốc, ví dụ, khối gốc 203 (khối hiện tại 203 của ảnh hiện tại 17), và dữ liệu ảnh được tái cấu trúc, ví dụ, các mẫu hoặc các khối được tái cấu trúc được lọc và/hoặc không được lọc của cùng ảnh (hiện tại) và/hoặc từ một hoặc nhiều ảnh được giải mã trước đó, ví dụ, từ bộ đệm ảnh được giải mã 230 hoặc các bộ đệm khác (ví dụ, bộ đệm dòng, không được thể hiện). Dữ liệu ảnh được tái cấu trúc được sử dụng như dữ liệu ảnh tham chiếu

cho việc dự đoán, ví dụ, liên dự đoán hoặc nội dự đoán, để thu nhận khối dự đoán 265 hoặc bộ dự đoán 265.

Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc lựa chọn việc phân chia cho chế độ dự đoán khối hiện tại (bao gồm việc không phân chia) và chế độ dự đoán (ví dụ, chế độ liên hoặc nội dự đoán) và tạo ra khối dự đoán 265 tương ứng, mà được sử dụng cho việc tính toán của khối dư 205 và cho việc tái cấu trúc của khối được tái cấu trúc 215.

Các phương án về bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để lựa chọn việc phân chia và chế độ dự đoán (ví dụ, từ các việc phân chia và chế độ dự đoán được hỗ trợ bởi hoặc khả dụng đối với bộ phận lựa chọn chế độ 260), mà đưa ra sự trùng khớp nhất hoặc nói cách khác dư nhỏ nhất (dư nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc phí tổn báo hiệu nhỏ nhất (phí tổn báo hiệu nhỏ nhất nghĩa là việc nén tốt hơn đối với việc truyền hoặc lưu trữ), hoặc xem xét hoặc cân bằng cả hai. Bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để xác định việc phân chia và chế độ dự đoán dựa vào sự tối ưu hóa tốc độ biến dạng (rate distortion optimization, viết tắt là RDO), nghĩa là, lựa chọn chế độ dự đoán mà đưa ra tốc độ biến dạng nhỏ nhất. Các thuật ngữ như “tốt nhất”, “nhỏ nhất”, “tối ưu” v.v. theo ngữ cảnh này không cần thiết đề cập đến tổng thể “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v. mà cũng có thể đề cập đến việc hoàn tất của tiêu chuẩn chấm dứt hoặc lựa chọn như trị số vượt quá hoặc giảm dưới ngưỡng hoặc các sự hạn chế khác có khả năng dẫn tới “sự lựa chọn dưới điều kiện tốt nhất” nhưng làm giảm độ phức tạp và thời gian xử lý.

Nói cách khác, bộ phận phân chia 262 có thể được tạo cấu hình để phân chia khối 203 thành các phần chia khối nhỏ hơn hoặc các khối con (mà tạo nên lại các khối), ví dụ, nhờ sử dụng lặp lại việc phân chia cây tứ phân (quad-tree, viết tắt là QT), việc phân chia cây nhị phân (binary-tree, viết tắt là BT), việc phân chia cây tam phân (triple-tree, viết tắt là TT) hoặc sự kết hợp bất kỳ của chúng, và để thực hiện, ví dụ, việc dự đoán cho mỗi trong số các phần chia khối hoặc các khối con, trong đó việc lựa chọn chế độ bao gồm việc lựa chọn cấu trúc cây của khối được phân chia 203 và các chế độ dự đoán được áp dụng cho mỗi trong số các phần chia khối hoặc các khối con.

Trong phần sau đây việc phân chia (ví dụ, bởi bộ phận phân chia 260) và việc xử lý dự đoán (bởi bộ phận liên dự đoán 244 và bộ phận nội dự đoán 254) được thực hiện bởi bộ mã hóa video 20 ví dụ sẽ được giải thích chi tiết hơn.

#### Phân chia

Bộ phận phân chia 262 có thể phân chia (hoặc phân tách) khối hiện tại 203 thành các phần chia nhỏ hơn, ví dụ, các khối nhỏ hơn có kích thước hình vuông hoặc hình chữ nhật. Các khối nhỏ hơn này (mà cũng có thể được gọi là các khối con) còn có thể được phân chia thậm chí thành các phần chia nhỏ hơn. Điều này cũng được đề cập đến là việc phân chia cây hoặc việc phân chia cây theo thứ bậc, trong đó khối gốc, ví dụ, ở mức cây gốc 0 (mức thứ bậc 0, độ sâu 0), có thể được phân chia đệ quy, ví dụ, được phân chia thành hai hoặc nhiều khối của mức cây thấp hơn tiếp theo, ví dụ, các nút ở mức cây 1 (mức thứ bậc 1, độ sâu 1), trong đó các khối này có thể được phân chia lại thành hai hoặc nhiều khối của mức thấp hơn tiếp theo, ví dụ, mức cây 2 (mức thứ bậc 2, độ sâu 2), v.v. cho đến khi việc phân chia được chấm dứt, ví dụ, bởi vì tiêu chuẩn chấm dứt được hoàn tất, ví dụ, độ sâu cây lớn nhất hoặc kích thước khối nhỏ nhất đạt được. Các khối mà không còn được phân chia cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng việc phân chia thành hai phần chia được gọi là cây nhị phân (BT), cây sử dụng việc phân chia thành ba phần chia được gọi là cây tam phân (TT), và cây sử dụng việc phân chia thành bốn phần chia được gọi là cây tứ phân (QT).

Như được nêu trên, thuật ngữ “khối” như được sử dụng ở đây có thể là phần, cụ thể là phần hình vuông hoặc hình chữ nhật, của ảnh. Ví dụ, dựa vào HEVC và VVC, khối có thể là hoặc tương ứng với bộ phận cây tạo mã (CTU), bộ phận tạo mã (coding unit, viết tắt là CU), bộ phận dự đoán (prediction unit, viết tắt là PU), và bộ phận biến đổi (transform unit, viết tắt là TU) và/hoặc với các khối tương ứng, ví dụ, khối cây tạo mã (CTB), khối tạo mã (coding block, viết tắt là CB), khối biến đổi (transform block, viết tắt là TB) hoặc khối dự đoán (prediction block, viết tắt là PB).

Ví dụ, bộ phận cây tạo mã (CTU) có thể là hoặc bao gồm CTB của các mẫu luma, hai CTB tương ứng của các mẫu chroma của ảnh mà có ba mảng mẫu, hoặc CTB của các mẫu của ảnh đơn sắc hoặc ảnh mà được tạo mã nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Một cách tương ứng, khối cây tạo mã (CTB) có thể là khối  $N \times N$  của các mẫu đối với một số trị số của

N sao cho việc chia bộ phận thành các CTB là phân chia. Bộ phận tạo mã (CU) có thể là hoặc bao gồm khối tạo mã của các mẫu luma, hai khối tạo mã tương ứng của các mẫu chroma của ảnh mà có ba mảng mẫu, hoặc khối tạo mã của các mẫu của ảnh đơn sắc hoặc ảnh mà được tạo mã nhờ sử dụng ba mặt phẳng màu sắc riêng biệt và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Một cách tương ứng khối tạo mã (CB) có thể là khối  $M \times N$  của các mẫu đối với một số trị số của  $M$  và  $N$  sao cho việc chia CTB thành các khối tạo mã là phân chia.

Theo các phương án, ví dụ, theo HEVC, bộ phận cây tạo mã (CTU) có thể được phân tách thành các CU nhờ sử dụng cấu trúc cây tứ phân được biểu thị là cây tạo mã. Việc quyết định xem có tạo mã vùng ảnh nhờ sử dụng việc liên dự đoán ảnh (thời gian) hay nội dự đoán ảnh (không gian) được thực hiện ở mức CU. Mỗi CU còn có thể được phân tách thành một, hai hoặc bốn PU theo loại phân tách PU. Bên trong một PU, quy trình dự đoán giống nhau được áp dụng và thông tin thích hợp được truyền tới bộ giải mã trên cơ sở PU. Sau khi thu nhận khối dư bằng cách áp dụng quy trình dự đoán dựa vào loại phân tách PU, CU có thể được phân chia thành các bộ phận biến đổi (TU) theo cấu trúc cây tứ phân khác giống với cây tạo mã cho CU.

Theo các phương án, ví dụ, theo chuẩn tạo mã video mới nhất hiện nay đang được phát triển, mà được gọi là tạo mã video đa năng (VVC), việc phân chia cây tứ phân và cây nhị phân (Quad-tree and binary tree, viết tắt là QTBT) được sử dụng để phân chia khối tạo mã. Trong cấu trúc khối QTBT, CU có thể có hoặc dạng hình vuông hoặc hình chữ nhật. Ví dụ, bộ phận cây tạo mã (CTU) đầu tiên được phân chia bởi cấu trúc cây tứ phân. Các nút lá tứ phân còn được phân chia bởi cấu trúc cây nhị phân hoặc bậc ba (tam phân (triple)). Các nút lá cây phân chia được gọi là các bộ phận tạo mã (các CU), và việc phân đoạn đó được sử dụng cho việc dự đoán và xử lý biến đổi mà không phân chia thêm bất kỳ. Điều này nghĩa là CU, PU và TU có cùng kích thước khối trong cấu trúc khối tạo mã QTBT. Song song, nhiều phân chia, ví dụ, phân chia cây tam phân cũng đã được đề xuất được sử dụng cùng với cấu trúc khối QTBT.

Theo một ví dụ, bộ phận lựa chọn chế độ 260 của bộ mã hóa video 20 có thể được tạo cấu hình để thực hiện sự kết hợp bất kỳ của các kỹ thuật phân chia được mô tả ở đây.

Như được mô tả ở trên, bộ mã hóa video 20 được tạo cấu hình để xác định hoặc lựa chọn chế độ dự đoán tốt nhất hoặc tối ưu từ tập hợp của các chế độ dự đoán (định trước). Tập hợp của các chế độ dự đoán có thể bao gồm, ví dụ, các chế độ nội dự đoán và/hoặc các chế độ liên dự đoán.

#### Nội dự đoán

Tập hợp của các chế độ nội dự đoán có thể bao gồm 35 chế độ nội dự đoán khác nhau, ví dụ, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ có hướng, ví dụ, như được định rõ trong HEVC, hoặc có thể bao gồm 67 chế độ nội dự đoán khác nhau, ví dụ, các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ hai chiều, hoặc các chế độ có hướng, ví dụ, như được định rõ đối với VVC.

Bộ phận nội dự đoán 254 được tạo cấu hình để sử dụng các mẫu được tái cấu trúc của các khối lân cận của cùng ảnh hiện tại để tạo ra khối nội dự đoán 265 theo chế độ nội dự đoán của tập hợp của các chế độ nội dự đoán.

Bộ phận nội dự đoán 254 (hoặc bộ phận lựa chọn chế độ 260 nói chung) còn được tạo cấu hình để đưa ra các thông số nội dự đoán (hoặc thông tin biểu thị của chế độ nội dự đoán được lựa chọn cho khối nói chung) tới bộ phận mã hóa entropi 270 dưới dạng của các thành phần cú pháp 266 cho việc đưa vào dữ liệu ảnh được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số dự đoán để giải mã.

#### Liên dự đoán

Tập hợp của các chế độ liên dự đoán (hoặc khả thi) tùy thuộc vào các ảnh tham chiếu khả dụng (nghĩa là, các ảnh được giải mã ít nhất một phần trước đó, ví dụ, được lưu trữ trong DBP 230) và các thông số liên dự đoán khác, ví dụ, xem toàn bộ ảnh tham chiếu hay chỉ một phần, ví dụ, vùng cửa sổ tìm kiếm xung quanh vùng của khối hiện tại, của ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu phù hợp nhất, và/hoặc ví dụ, xem phép nội suy điểm ảnh được áp dụng hay không, ví dụ, phép nội suy nửa điểm ảnh và/hoặc một phần tư điểm ảnh.

Ngoài các chế độ dự đoán nêu trên, chế độ bỏ qua và/hoặc chế độ trực tiếp có thể được áp dụng.

Bộ phận liên dự đoán 244 có thể bao gồm bộ phận đánh giá chuyển động (motion estimation, viết tắt là ME) và bộ phận bù chuyển động (motion compensation, viết tắt



là MC) (cả hai không được thể hiện trên Fig.2). Bộ phận đánh giá chuyển động có thể được tạo cấu hình để thu hoặc thu nhận khối ảnh 203 (khối ảnh hiện tại 203 của ảnh hiện tại 17) và ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái cấu trúc trước đó, ví dụ, các khối được tái cấu trúc của một hoặc nhiều ảnh được giải mã trước đó khác/khác nhau 231, dùng cho việc đánh giá chuyển động. Ví dụ, chuỗi video có thể bao gồm ảnh hiện tại và các ảnh được giải mã trước đó 231, hoặc nói cách khác, ảnh hiện tại và các ảnh được giải mã trước đó 231 có thể là phần của hoặc tạo nên chuỗi các ảnh tạo nên chuỗi video.

Bộ mã hóa 20 có thể, ví dụ, được tạo cấu hình để lựa chọn khối tham chiếu từ các khối tham chiếu của cùng hoặc các ảnh khác nhau của các ảnh khác và cung cấp ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu) và/hoặc dịch vị (dịch vị không gian) giữa vị trí (các tọa độ  $x, y$ ) của khối tham chiếu và vị trí của khối hiện tại là các thông số liên dự đoán tới bộ phận đánh giá chuyển động. Dịch vị này cũng được gọi là vector chuyển động (motion vector, viết tắt là MV).

Bộ phận bù chuyển động được tạo cấu hình để thu nhận, ví dụ, thu, thông số liên dự đoán và để thực hiện việc liên dự đoán dựa vào hoặc nhờ sử dụng thông số liên dự đoán để thu nhận khối liên dự đoán 265. Việc bù chuyển động, được thực hiện bởi bộ phận bù chuyển động, có thể bao gồm việc nạp hoặc tạo ra khối dự đoán dựa vào vector chuyển động/khối được xác định bởi việc đánh giá chuyển động, có thể thực hiện các phép nội suy với độ chính xác điểm ảnh con. Việc lọc nội suy có thể tạo ra các mẫu điểm ảnh bổ sung từ các mẫu điểm ảnh đã biết, vì vậy có khả năng làm tăng số lượng các khối dự đoán ứng viên mà có thể được sử dụng để tạo mã khối ảnh. Khi thu vector chuyển động cho PU của khối ảnh hiện tại, bộ phận bù chuyển động có thể bố trí khối dự đoán trong đó vector chuyển động chỉ trong một trong số các danh mục ảnh tham chiếu.

Bộ phận bù chuyển động cũng có thể tạo ra các thành phần cú pháp được kết hợp với các khối và lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối ảnh của lát video.

Tạo mã entropi

Bộ phận mã hóa entropi 270 được tạo cấu hình để áp dụng, ví dụ, thuật toán hoặc sơ đồ mã hóa entropi (ví dụ, sơ đồ tạo mã độ dài biến đổi (variable length coding, viết

tất là VLC), sơ đồ VLC thích nghi ngữ cảnh (context adaptive VLC, viết tắt là CAVLC), sơ đồ tạo mã thuật toán, nhị phân hóa, tạo mã thuật toán nhị phân thích nghi ngữ cảnh (context adaptive binary arithmetic coding, viết tắt là CABAC), tạo mã thuật toán nhị phân thích nghi ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, viết tắt là SBAC), tạo mã entropi phân chia khoảng xác suất (probability interval partitioning entropy, viết tắt là PIPE) hoặc phương pháp hoặc kỹ thuật mã hóa entropi khác) hoặc tránh qua (không nén) trên các hệ số được lượng tử hóa 209, các thông số liên dự đoán, các thông số nội dự đoán, các thông số bộ lọc vòng lặp và/hoặc các thành phần cú pháp khác để thu nhận dữ liệu ảnh được mã hóa 21 mà có thể được đưa ra qua đầu ra 272, ví dụ, dưới dạng của dòng bit được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số để giải mã. Dòng bit được mã hóa 21 có thể được truyền tới bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ dùng cho việc truyền hoặc truy xuất sau đó bởi bộ giải mã video 30.

Các biến thể cấu trúc khác của bộ mã hóa video 20 có thể được sử dụng để mã hóa dòng video. Ví dụ, bộ mã hóa dựa trên việc không biến đổi 20 có thể lượng tử hóa trực tiếp tín hiệu dư mà không có bộ phận xử lý biến đổi 206 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ mã hóa 20 có thể có bộ phận lượng tử hóa 208 và bộ phận lượng tử hóa ngược 210 được kết hợp thành bộ phận đơn.

#### Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện ví dụ về bộ giải mã video 30 mà được tạo cấu hình để thực hiện các kỹ thuật theo sáng chế. Bộ giải mã video 30 được tạo cấu hình để thu dữ liệu ảnh được mã hóa 21 (ví dụ, dòng bit được mã hóa 21), ví dụ, được mã hóa bởi bộ mã hóa 20, để thu nhận ảnh được giải mã 331. Dữ liệu ảnh được mã hóa hoặc dòng bit bao gồm thông tin để giải mã dữ liệu ảnh được mã hóa, ví dụ, dữ liệu mà biểu diễn các khối ảnh của lát video được mã hóa và các thành phần cú pháp được kết hợp.

Theo ví dụ trên Fig.3, bộ giải mã 30 bao gồm bộ phận giải mã entropi 304, bộ phận lượng tử hóa ngược 310, bộ phận xử lý biến đổi ngược 312, bộ phận tái cấu trúc 314 (ví dụ, bộ cộng 314), bộ lọc vòng lặp 320, bộ đệm ảnh được giải mã (DBP) 330, bộ phận liên dự đoán 344 và bộ phận nội dự đoán 354. Bộ phận liên dự đoán 344 có thể là hoặc bao gồm bộ phận bù chuyển động. Bộ giải mã video 30 có thể, theo một số ví dụ, thực

hiện quy trình giải mã thường ngược với quy trình mã hóa được mô tả dựa vào bộ mã hóa video 100 từ Fig.2.

Như được giải thích liên quan đến bộ mã hóa 20, bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 214 bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận liên dự đoán 344 và bộ phận nội dự đoán 354 cũng được đề cập đến là tạo nên “bộ giải mã tích hợp” của bộ mã hóa video 20. Theo đó, bộ phận lượng tử hóa ngược 310 có thể giống nhau về chức năng với bộ phận lượng tử hóa ngược 110, bộ phận xử lý biến đổi ngược 312 có thể giống nhau về chức năng với bộ phận xử lý biến đổi ngược 212, bộ phận tái cấu trúc 314 có thể giống nhau về chức năng với bộ phận tái cấu trúc 214, bộ lọc vòng lặp 320 có thể giống nhau về chức năng với bộ lọc vòng lặp 220, và bộ đệm ảnh được giải mã 330 có thể giống nhau về chức năng với bộ đệm ảnh được giải mã 230. Do đó, các phần giải thích được đưa ra đối với các bộ phận và các chức năng tương ứng của bộ mã hóa video 20 áp dụng một cách tương ứng với các bộ phận và các chức năng tương ứng của bộ giải mã video 30.

#### Giải mã entropi

Bộ phận giải mã entropi 304 được tạo cấu hình để tách dòng bit 21 (hoặc dữ liệu ảnh được mã hóa 21 nói chung) và thực hiện, ví dụ, việc giải mã entropi tới dữ liệu ảnh được mã hóa 21 để thu nhận, ví dụ, các hệ số được lượng tử hóa 309 và/hoặc các thông số tạo mã được giải mã (không được thể hiện trên Fig.3), ví dụ, bất kỳ hoặc tất cả các thông số liên dự đoán (ví dụ, chỉ số ảnh tham chiếu và vectơ chuyển động), thông số nội dự đoán (ví dụ, chế độ hoặc chỉ số nội dự đoán), các thông số biến đổi, các thông số lượng tử hóa, các thông số bộ lọc vòng lặp, và/hoặc các thành phần cú pháp khác. Bộ phận giải mã entropi 304 có thể được tạo cấu hình để áp dụng các thuật toán hoặc các sơ đồ giải mã tương ứng với các sơ đồ mã hóa như được mô tả liên quan đến bộ phận mã hóa entropi 270 của bộ mã hóa 20. Bộ phận giải mã entropi 304 còn có thể được tạo cấu hình để cung cấp các thông số liên dự đoán, thông số nội dự đoán và/hoặc các thành phần cú pháp khác tới bộ phận lựa chọn chế độ 360 và các thông số khác tới các bộ phận khác của bộ giải mã 30. Bộ giải mã video 30 có thể thu các thành phần cú pháp ở mức lát video và/hoặc mức khối video.

#### Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 310 có thể được tạo cấu hình để thu các thông số lượng tử hóa (QP) (hoặc thông tin liên quan đến việc lượng tử hóa ngược nói chung) và các hệ số được lượng tử hóa từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) và để áp dụng dựa vào các thông số lượng tử hóa việc lượng tử hóa ngược trên các hệ số được lượng tử hóa được giải mã 309 để thu nhận các hệ số được giải lượng tử hóa 311, mà cũng có thể được gọi là các hệ số biến đổi 311. Quy trình lượng tử hóa ngược có thể bao gồm việc sử dụng của thông số lượng tử hóa được xác định bởi bộ mã hóa video 20 cho mỗi khối video trong lát video để xác định mức lượng tử hóa và, tương tự, mức lượng tử hóa ngược mà sẽ được áp dụng.

#### Biến đổi ngược

Bộ phận xử lý biến đổi ngược 312 có thể được tạo cấu hình để thu các hệ số được giải lượng tử hóa 311, cũng được gọi là các hệ số biến đổi 311, và để áp dụng việc biến đổi tới các hệ số được giải lượng tử hóa 311 để thu nhận các khối dư được tái cấu trúc 213 trong miền mẫu. Các khối dư được tái cấu trúc 213 cũng có thể được gọi là các khối biến đổi 313. Việc biến đổi có thể là biến đổi ngược, ví dụ, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc quy trình biến đổi ngược giống nhau về mặt khái niệm. Bộ phận xử lý biến đổi ngược 312 còn có thể được tạo cấu hình để thu các thông số biến đổi hoặc thông tin tương ứng từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) để xác định việc biến đổi được áp dụng cho các hệ số được giải lượng tử hóa 311.

#### Tái cấu trúc

Bộ phận tái cấu trúc 314 (ví dụ, bộ bổ sung hoặc bộ cộng 314) có thể được tạo cấu hình để bổ sung khối dư được tái cấu trúc 313, vào khối dự đoán 365 để thu nhận khối được tái cấu trúc 315 trong miền mẫu, ví dụ, bằng cách bổ sung các trị số mẫu của khối dư được tái cấu trúc 313 và các trị số mẫu của khối dự đoán 365.

#### Lọc

Bộ phận lọc vòng lặp 320 (hoặc trong vòng lặp tạo mã hoặc sau vòng lặp tạo mã) được tạo cấu hình để lọc khối được tái cấu trúc 315 để thu nhận khối được lọc 321, ví dụ, để làm mịn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận lọc vòng lặp 320 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng

hạn như bộ lọc giải khối, bộ lọc dịch vị thích nghi mẫu (SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc hai chiều, bộ lọc vòng lặp thích ứng (ALF), làm sắc, các bộ lọc làm mịn hoặc các bộ lọc phối hợp, hoặc sự kết hợp bất kỳ của chúng. Mặc dù bộ phận lọc vòng lặp 320 được thể hiện trên Fig.3 là bộ lọc trong vòng lặp, nhưng theo các cấu hình khác, bộ phận lọc vòng lặp 320 có thể được thực hiện như bộ lọc sau vòng lặp.

Bộ đệm ảnh được giải mã

Các khối video được giải mã 321 của ảnh sau đó được lưu trữ trong bộ đệm ảnh được giải mã 330, mà lưu trữ các ảnh được giải mã 331 như các ảnh tham chiếu cho việc bù chuyển động tiếp theo cho các ảnh khác và/hoặc cho việc đưa ra màn hình tương ứng.

Bộ giải mã 30 được tạo cấu hình để đưa ra ảnh được giải mã 311, ví dụ, qua đầu ra 312, nhằm biểu diễn hoặc xem tới người dùng.

Dự đoán

Bộ phận liên dự đoán 344 có thể giống với bộ phận liên dự đoán 244 (cụ thể là bộ phận bù chuyển động) và bộ phận nội dự đoán 354 có thể giống với bộ phận nội dự đoán 254 về chức năng, và thực hiện các quyết định phân tách hoặc phân chia và dự đoán dựa vào các thông số dự đoán và/hoặc phân chia hoặc thông tin tương ứng thu được từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304). Bộ phận lựa chọn chế độ 360 có thể được tạo cấu hình để thực hiện việc dự đoán (liên hoặc nội dự đoán) trên mỗi khối dựa vào các ảnh được tái cấu trúc, các khối hoặc các mẫu tương ứng (được lọc hoặc không được lọc) để thu nhận khối dự đoán 365.

Khi lát video được tạo mã là lát (I) được tạo mã bên trong, bộ phận nội dự đoán 354 của bộ phận lựa chọn chế độ 360 được tạo cấu hình để tạo ra khối dự đoán 365 cho khối ảnh của lát video hiện tại dựa vào chế độ nội dự đoán được báo hiệu và dữ liệu từ các khối được giải mã trước đó của ảnh hiện tại. Khi ảnh video được tạo mã là lát (nghĩa là, B hoặc P) được tạo mã liên kết, bộ phận liên dự đoán 344 (ví dụ, bộ phận bù chuyển động) của bộ phận lựa chọn chế độ 360 được tạo cấu hình để tạo ra các khối dự đoán 365 cho khối video của lát video hiện tại dựa vào các vectơ chuyển động và các thành phần cú pháp khác thu được từ bộ phận giải mã entropi 304. Đối với việc liên dự đoán, các khối dự đoán có thể được tạo ra từ một trong số các ảnh tham chiếu nằm trong một

trong số các danh mục ảnh tham chiếu. Bộ giải mã video 30 có thể cấu trúc các danh mục khung tham chiếu, danh mục 0 và danh mục 1, nhờ sử dụng các kỹ thuật cấu trúc mặc định dựa vào các ảnh tham chiếu được lưu trữ trong DPB 330.

Bộ phận lựa chọn chế độ 360 được tạo cấu hình để xác định thông tin dự đoán cho khối video của lát video hiện tại bằng cách tách các vectơ chuyển động và các thành phần cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán cho khối video hiện tại được giải mã. Ví dụ, bộ phận lựa chọn chế độ 360 sử dụng một vài trong số các thành phần cú pháp thu được để xác định chế độ dự đoán (ví dụ, liên hoặc nội dự đoán) được sử dụng để tạo mã các khối video của lát video, loại lát liên dự đoán (ví dụ, lát B, lát P, hoặc lát GPB), thông tin cấu trúc cho một hoặc nhiều trong số các danh mục ảnh tham chiếu cho lát, các vectơ chuyển động cho mỗi khối video được mã hóa liên kết của lát, trạng thái liên dự đoán cho mỗi khối video được tạo mã liên kết của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại.

Các biến thể khác của bộ giải mã video 30 có thể được sử dụng để giải mã dữ liệu ảnh được mã hóa 21. Ví dụ, bộ giải mã 30 có thể tạo ra dòng video đầu ra mà không có bộ phận lọc vòng lặp 320. Ví dụ, bộ giải mã dựa trên việc không biến đổi 30 có thể lượng tử hóa ngược trực tiếp tín hiệu dư mà không có bộ phận xử lý biến đổi ngược 312 cho các khối hoặc các khung nhất định. Theo cách thực hiện khác, bộ giải mã video 30 có thể có bộ phận lượng tử hóa ngược 310 và bộ phận xử lý biến đổi ngược 312 được kết hợp thành bộ phận đơn.

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại còn có thể được xử lý và sau đó được đưa ra tới bước tiếp theo. Ví dụ, sau khi lọc nội suy, việc dẫn ra vectơ chuyển động hoặc lọc vòng lặp, phép toán khác, chẳng hạn như cắt bớt hoặc dịch chuyển, có thể được thực hiện trên kết quả xử lý của việc lọc nội suy, việc dẫn ra vectơ chuyển động hoặc lọc vòng lặp.

Cần lưu ý rằng các phép toán khác có thể được áp dụng cho các vectơ chuyển động được dẫn ra của khối hiện tại (bao gồm nhưng không giới hạn ở các vectơ chuyển động điểm điều khiển của chế độ afin, các vectơ chuyển động khối con ở các chế độ afin, hai chiều, ATMVP, các vectơ chuyển động thời gian, và v.v.). Ví dụ, trị số của vectơ chuyển động bị ràng buộc ở khoảng được định rõ trước theo bit biểu diễn của nó. Nếu bit biểu diễn của vectơ chuyển động là bitDepth, thì sau đó khoảng là  $-2^{(\text{bitDepth}-1)} \sim$

$2^{(bitDepth-1)-1}$ , trong đó “^” nghĩa là lũy thừa. Ví dụ, nếu bitDepth được thiết đặt bằng 16, thì khoảng là -32768 ~ 32767; nếu bitDepth được thiết đặt bằng 18, thì khoảng là -131072~131071. Ví dụ, trị số của vectơ chuyển động được dẫn ra (ví dụ, các MV của bốn khối con 4x4 nằm trong một khối 8x8) bị ràng buộc sao cho độ chênh lệch lớn nhất giữa các phân số nguyên của bốn MV khối con 4x4 không nhiều hơn N điểm ảnh, chẳng hạn như không nhiều hơn 1 điểm ảnh. Ở đây đề xuất hai phương pháp ràng buộc vectơ chuyển động theo bitDepth.

Phương pháp 1: loại bỏ MSB tràn (bit quan trọng nhất) bởi các phép toán sau đây

$$ux = (mvx + 2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

$$uy = (mvy + 2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

$$mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

trong đó mvx là thành phần theo chiều ngang của vectơ chuyển động của khối ảnh hoặc khối con, mvy là thành phần theo chiều dọc của vectơ chuyển động của khối ảnh hoặc khối con, và ux và uy chỉ báo trị số trung gian;

Ví dụ, nếu trị số của mvx là -32769, sau khi áp dụng công thức (1) và (2), trị số kết quả là 32767. Trong hệ thống máy tính, các số thập phân được lưu trữ là bù hai. Bù hai của -32769 là 1,0111,1111,1111,1111 (17 bit), sau đó MSB được loại bỏ, vì vậy bù hai kết quả là 0111,1111,1111,1111 (số thập phân là 32767), mà giống như đầu ra bằng cách áp dụng công thức (1) và (2).

$$ux = (mvp_x + mvd_x + 2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

$$mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (8)$$

Các phép toán có thể được áp dụng trong suốt thời gian tổng của.mvp và mvd, như được thể hiện trong công thức (5) đến (8).

Phương pháp 2: loại bỏ MSB tràn bằng cách cắt bớt trị số

$$vx = \text{Clip3}(-2^{bitDepth-1}, 2^{bitDepth-1} - 1, vx)$$

$$vy = \text{Clip3}(-2^{bitDepth-1}, 2^{bitDepth-1} - 1, vy)$$

trong đó vx là thành phần theo chiều ngang của vectơ chuyển động của khối ảnh hoặc khối con, vy là thành phần theo chiều dọc của vectơ chuyển động của khối ảnh hoặc khối con; x, y và z lần lượt tương ứng với ba trị số đầu vào của quy trình cắt bớt MV, và định nghĩa của phương trình Clip3 như sau:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{theo cách khác} \end{cases}$$

Fig.4 là hình vẽ giản lược của thiết bị tạo mã video 400 theo phương án của sáng chế. Thiết bị tạo mã video 400 thích hợp cho việc thực hiện các phương án được bộc lộ như được mô tả ở đây. Theo phương án, thiết bị tạo mã video 400 có thể là bộ giải mã chẳng hạn như bộ giải mã video 30 trên Fig.1A hoặc bộ mã hóa chẳng hạn như bộ mã hóa video 20 trên Fig.1A.

Thiết bị tạo mã video 400 bao gồm các cổng vào 410 (hoặc các cổng đầu vào 410) và các bộ phận bộ thu (Rx) 420 để thu dữ liệu; bộ xử lý, bộ phận logic, hoặc bộ xử lý trung tâm (central processing unit, viết tắt là CPU) 430 để xử lý dữ liệu; các bộ phận bộ truyền (Tx) 440 và các cổng ra 450 (hoặc các cổng đầu ra 450) để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị tạo mã video 400 cũng có thể bao gồm các thành phần quang-điện (optical-to-electrical, viết tắt là OE) và các thành phần điện-quang (electrical-to-optical, viết tắt là EO) được ghép nối với các cổng vào 410, các bộ phận bộ thu 420, các bộ phận bộ truyền 440, và các cổng ra 450 dùng cho việc ra hoặc vào của các tín hiệu quang hoặc điện.

Bộ xử lý 430 được thực hiện bởi phần cứng và phần mềm. Bộ xử lý 430 có thể được thực hiện như một hoặc nhiều chip CPU, các lõi (ví dụ, như bộ xử lý nhiều lõi), các FPGA, các ASIC, và các DSP. Bộ xử lý 430 truyền thông với các cổng vào 410, các bộ phận bộ thu 420, các bộ phận bộ truyền 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun tạo mã 470. Môđun tạo mã 470 thực hiện các phương án được bộc lộ nêu trên. Chẳng hạn, môđun tạo mã 470 thực hiện, xử lý, chuẩn bị, hoặc đưa ra các thao tác tạo mã khác nhau. Việc bao gồm môđun tạo mã 470 do đó cung cấp sự nâng cao đáng kể tới tính năng của thiết bị tạo mã video 400 và ảnh hưởng việc biến đổi của thiết bị tạo mã video 400 tới trạng thái khác. Theo cách khác, môđun tạo mã 470 được thực hiện như các lệnh được lưu trữ trong bộ nhớ 460 và được thực hiện bởi bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, nhiều ổ đĩa băng, và nhiều ổ đĩa trạng thái rắn và có thể được sử dụng như thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình như vậy được lựa chọn cho việc thực hiện, và để lưu trữ các lệnh và dữ liệu mà được đọc trong suốt thời gian thực hiện chương trình. Bộ nhớ



460 có thể, ví dụ, khả biến và/hoặc bất khả biến và có thể là bộ nhớ chỉ đọc (read-only memory, viết tắt là ROM), bộ nhớ truy cập ngẫu nhiên (random access memory, viết tắt là RAM), bộ nhớ có thể lập địa chỉ nội dung bậc ba (ternary content-addressable memory, viết tắt là TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (static random-access memory, viết tắt là SRAM).

Fig.5 là sơ đồ khối được đơn giản hóa của máy 500 mà có thể được sử dụng như một trong hai hoặc cả thiết bị nguồn 12 và thiết bị đích 14 từ Fig.1 theo phương án làm ví dụ.

Bộ xử lý 502 trong máy 500 có thể là bộ xử lý trung tâm. Theo cách khác, bộ xử lý 502 có thể là loại thiết bị khác bất kỳ, hoặc nhiều thiết bị, có khả năng điều khiển hoặc xử lý thông tin tồn tại hiện thời hoặc được phát triển sau này. Mặc dù cách thực hiện được bộc lộ có thể được thực hiện với bộ xử lý đơn như được thể hiện, ví dụ, bộ xử lý 502, nhưng các ưu điểm về tốc độ và hiệu quả có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong máy 500 có thể là thiết bị bộ nhớ chỉ đọc (ROM) hoặc thiết bị bộ nhớ truy cập ngẫu nhiên (RAM) theo cách thực hiện. Loại thiết bị lưu trữ thích hợp khác bất kỳ có thể được sử dụng như bộ nhớ 504. Bộ nhớ 504 có thể bao gồm mã và dữ liệu 506 mà được truy cập bởi bộ xử lý 502 nhờ sử dụng bus 512. Bộ nhớ 504 còn có thể bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình mà cho phép bộ xử lý 502 thực hiện các phương pháp được mô tả ở đây. Ví dụ, các chương trình ứng dụng 510 có thể bao gồm ứng dụng 1 qua N, mà còn bao gồm ứng dụng tạo mã video mà thực hiện các phương pháp được mô tả ở đây.

Máy 500 cũng có thể bao gồm một hoặc nhiều thiết bị đầu ra, chẳng hạn như màn hình 518. Màn hình 518 có thể là, theo một ví dụ, màn hình nhạy chạm mà kết hợp màn hình với thành phần nhạy chạm mà có thể thao tác được để cảm biến các đầu vào chạm. Màn hình 518 có thể được ghép nối với bộ xử lý 502 qua bus 512.

Mặc dù được mô tả ở đây là bus đơn, nhưng bus 512 của máy 500 có thể bao gồm nhiều bus. Hơn nữa, thiết bị lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các bộ phận còn lại của máy 500 hoặc có thể được truy cập qua mạng và có thể bao gồm bộ

phân tích hợp đơn chẳng hạn như thẻ nhớ hoặc nhiều bộ phận chẳng hạn như nhiều thẻ nhớ. Máy 500 vì vậy có thể được thực hiện theo nhiều cấu hình khác nhau.

### **Bộ lọc trong vòng lặp**

Có tổng cộng ba bộ lọc trong vòng lặp trong VTM3. Ngoài bộ lọc tách khối và SAO (hai bộ lọc vòng lặp trong HEVC), bộ lọc vòng lặp thích ứng (ALF) được áp dụng trong VTM3. Thứ tự của quy trình lọc trong VTM3 là bộ lọc tách khối, SAO và ALF.

### **ALF**

Trong VTM5, bộ lọc vòng lặp thích ứng (ALF) với việc thích ứng bộ lọc dựa trên khối được áp dụng. Đối với thành phần luma, một trong số 25 bộ lọc được lựa chọn cho mỗi khối  $4 \times 4$ , dựa vào hướng và hoạt động của các gradient cục bộ.

### **Hình dạng bộ lọc:**

Trong JEM, hai hình dạng bộ lọc kim cương (như được thể hiện trên Fig.6) được sử dụng cho thành phần luma. Hình dạng kim cương  $7 \times 7$  được áp dụng cho thành phần luma và hình dạng kim cương  $5 \times 5$  được áp dụng cho thành phần chroma.

### **Phân loại khối:**

Đối với thành phần luma, mỗi khối  $4 \times 4$  được phân loại thành một trong số 25 lớp. Chỉ số phân loại  $C$  được dẫn ra dựa vào hướng của nó  $D$  và trị số hoạt động được lượng tử hóa  $\hat{A}$ , như sau:

$$C = 5D + \hat{A} \quad (9)$$

Để tính toán  $D$  và  $\hat{A}$ , các gradient của hướng theo chiều ngang, theo chiều dọc và theo hai đường chéo đầu tiên được tính toán nhờ sử dụng 1-D Laplacian:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad (10)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad (11)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (12)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (13)$$

Trong đó các chỉ số  $i$  và  $j$  đề cập đến các tọa độ của mẫu trên cùng bên trái nằm trong khối  $4 \times 4$  và  $R(i, j)$  chỉ báo mẫu được tái cấu trúc ở tọa độ  $(i, j)$ .

Để làm giảm độ phức tạp của việc phân loại khối, việc tính toán 1-D Laplacian được lấy mẫu phụ được áp dụng. Như được thể hiện trên Fig.7, các vị trí được lấy mẫu phụ giống nhau được sử dụng cho việc tính toán gradient của tất cả các hướng.

Sau đó các trị số lớn nhất và nhỏ nhất  $D$  của các gradient của các hướng theo chiều ngang và theo chiều dọc được thiết đặt là:

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v) \quad (14)$$

Các trị số lớn nhất và nhỏ nhất của gradient của hai hướng theo đường chéo được thiết đặt là:

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}) \quad (15)$$

Để dẫn ra trị số của hướng  $D$ , các trị số này được so với nhau và với hai ngưỡng  $t_1$  và  $t_2$ :

**Bước 1.** Nếu cả  $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$  và  $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$  là đúng, thì  $D$  được thiết đặt bằng 0.

**Bước 2.** Nếu  $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$ , tiếp tục từ bước 3; theo cách khác tiếp tục từ bước 4.

**Bước 3.** Nếu  $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ ,  $D$  được thiết đặt bằng 2; theo cách khác  $D$  được thiết đặt bằng 1.

**Bước 4.** Nếu  $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ ,  $D$  được thiết đặt bằng 4; theo cách khác  $D$  được thiết đặt bằng 3.

Trị số hoạt động  $A$  được tính toán là:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \quad (16)$$

$A$  còn được lượng tử hóa tới khoảng từ 0 đến 4, và trị số được lượng tử hóa được biểu thị là  $\hat{A}$ .

Đối với các thành phần chroma trong ảnh, không có phương pháp phân loại nào được áp dụng, nghĩa là, tập hợp đơn của các hệ số ALF được áp dụng cho mỗi thành phần chroma.

### Các phép biến đổi hình học của các hệ số bộ lọc

Trước khi lọc mỗi khối luma  $4 \times 4$ , các phép biến đổi hình học chẳng hạn như quay hoặc lật theo đường chéo hoặc theo chiều dọc được áp dụng cho các hệ số bộ lọc  $f(k, l)$  tùy thuộc vào các trị số gradient được tính toán cho khối đó. Điều này tương đương với việc áp dụng các phép biến đổi này cho các mẫu trong vùng hỗ trợ bộ lọc. Ý tưởng là làm cho các khối khác nhau mà trong đó ALF được áp dụng giống nhau hơn bằng cách căn chỉnh hướng của chúng.

Ba phép biến đổi hình học, bao gồm lật theo đường chéo, theo chiều dọc và quay được giới thiệu:

$$\text{Theo đường chéo: } f_D(k, l) = f(l, k), \quad (17)$$

$$\text{Lật theo chiều dọc: } f_V(k, l) = f(k, K - l - 1) \quad (18)$$

$$\text{Quay: } f_R(k, l) = f(K - l - 1, k) \quad (19)$$

trong đó  $K$  là kích thước của bộ lọc và  $0 \leq k, l \leq K - 1$  là các tọa độ các hệ số, sao cho vị trí  $(0,0)$  ở góc trên cùng bên trái và vị trí  $(K - 1, K - 1)$  ở góc dưới cùng bên phải. Các phép biến đổi được áp dụng cho các hệ số bộ lọc  $f(k, l)$  tùy thuộc vào các trị số gradient được tính toán cho khối đó. Mỗi tương quan giữa phép biến đổi và bốn gradient của bốn hướng được tóm tắt trong bảng sau đây.

**Bảng 1 Việc ánh xạ của gradient được tính toán cho một khối và các phép biến đổi**

Các trị số gradient	Phép biến đổi
$g_{d2} < g_{d1}$ và $g_h < g_v$	Không có phép biến đổi
$g_{d2} < g_{d1}$ và $g_v < g_h$	Theo đường chéo
$g_{d1} < g_{d2}$ và $g_h < g_v$	Lật theo chiều dọc
$g_{d1} < g_{d2}$ và $g_v < g_h$	Quay

### Báo hiệu các thông số bộ lọc

Trong VTM3, các thông số bộ lọc ALF được báo hiệu ở đoạn đầu lát. Lên đến 25 tập hợp của các hệ số bộ lọc luma có thể được báo hiệu. Để làm giảm phí tổn các bit, các hệ số bộ lọc của việc phân loại khác có thể được hợp nhất.

Quy trình lọc có thể được điều khiển ở mức CTB. Cờ luôn được báo hiệu để chỉ báo xem ALF được áp dụng cho CTB luma hay không. Đối với mỗi CTB chroma, cờ có thể được báo hiệu để chỉ báo xem ALF được áp dụng cho CTB chroma hay không tùy thuộc vào trị số của `alf_chroma_ctb_present_flag`.

Các hệ số bộ lọc được lượng tử hóa với định mức bằng 128. Để hạn chế thêm độ phức tạp phép nhân, sự phù hợp dòng bit được áp dụng mà trị số hệ số của vị trí trung tâm sẽ trong khoảng từ 0 đến  $2^8$  và các trị số hệ số của các vị trí còn lại sẽ trong khoảng từ  $-2^7$  đến  $2^7 - 1$ .

### Quy trình lọc

Ở phía bộ giải mã, khi ALF được cho phép cho CTB, mỗi mẫu  $R(i, j)$  nằm trong CU được lọc, dẫn đến trị số mẫu  $R'(i, j)$  như được thể hiện dưới đây, trong đó  $L$  biểu thị độ dài bộ lọc,  $f_{m,n}$  biểu diễn hệ số bộ lọc, và  $f(k, l)$  biểu thị các hệ số bộ lọc được giải mã.

$$R'(i, j) = \left( \sum_{k=-\frac{L}{2}}^{\frac{L}{2}} \sum_{l=-\frac{L}{2}}^{\frac{L}{2}} f(k, l) \times R(i+k, j+l) + 64 \right) \gg 7 \quad (20)$$

$$\text{Hoặc việc lọc cũng có thể được thể hiện là } O(x, y) = \sum_{(i,j)} w(i, j) \cdot I(x+i, y+j) \quad (21)$$

trong đó các mẫu  $I(x+i, y+j)$  là các mẫu đầu vào,  $O(x, y)$  là mẫu đầu ra được lọc (nghĩa là, kết quả bộ lọc), và  $w(i, j)$  biểu thị các hệ số bộ lọc trong đó các mẫu  $I(x+i, y+j)$  là các mẫu đầu vào,  $O(x, y)$  là mẫu đầu ra được lọc (nghĩa là, kết quả bộ lọc), và  $w(i, j)$  biểu thị các hệ số bộ lọc. Trong thực tế, trong VTM nó được thực hiện nhờ sử dụng số học số nguyên cho các phép tính độ chính xác điểm cố định:

$$O(x, y) = \left( \sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i, j) \cdot I(x+i, y+j) + 64 \right) \gg 7 \quad (22)$$

trong đó  $L$  biểu thị độ dài bộ lọc, và trong đó  $w(i, j)$  là các hệ số bộ lọc về độ chính xác điểm cố định.

Từ VTM5 trở đi (ITU JVET-N0242), ALF được thực hiện theo cách phi tuyến tính. Phương trình 21 có thể được đưa vào công thức như sau:

$$O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x + i, y + j) - I(x, y)) \quad (23)$$

Trong đó  $w(i, j)$  là các hệ số bộ lọc giống như ở phương trình (22) [ $w(0, 0)$  được ngoài trừ mà bằng 1 theo phương trình (23) trong khi nó bằng  $1 - \sum_{(i,j) \neq (0,0)} w(i, j)$  theo phương trình (21)].

Bộ lọc được điều chỉnh thêm bằng cách giới thiệu tính phi tuyến tính để làm cho ALF hiệu quả hơn nhờ sử dụng chức năng cắt bớt để làm giảm tác động của các trị số mẫu lân cận ( $I(x + i, y + j)$ ) khi chúng quá khác với trị số mẫu hiện tại ( $I(x, y)$ ) được lọc.

Trong VTM5, bộ lọc ALF được điều chỉnh như sau:

$$O'(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot K(I(x + i, y + j) - I(x, y), k(i, j)) \quad (24)$$

trong đó  $K(d, b) = \min(b, \max(-b, d))$  là chức năng cắt bớt, và  $k(i, j)$  là các thông số cắt bớt, mà tùy thuộc vào hệ số bộ lọc  $(i, j)$ . Các thông số cắt bớt  $k(i, j)$  được quy định cho mỗi bộ lọc ALF, một trị số cắt bớt được báo hiệu trên mỗi hệ số bộ lọc. Nghĩa là lên đến 12 trị số cắt bớt có thể được báo hiệu ở dòng bit trên mỗi bộ lọc Luma và lên đến 6 trị số cắt bớt cho bộ lọc Chroma.

Để giới hạn chi phí báo hiệu và độ phức tạp bộ mã hóa, việc đánh giá của các trị số cắt bớt được giảm tới tập hợp nhỏ của các trị số khả thi. Trong VTM5, chỉ 4 trị số cố định khả thi được sử dụng mà giống nhau đối với các nhóm miếng bên trong và liên kết.

Bởi vì phương sai của các độ chênh lệch cục bộ thường cao hơn đối với Luma so với đối với Chroma, nên hai tập hợp khác nhau của các trị số cắt bớt đối với các bộ lọc Luma và Chroma được sử dụng. Các trị số cắt bớt cũng bao gồm trị số mẫu lớn nhất (ở đây 1024 đối với độ sâu bit 10 bit) trong mỗi tập hợp, sao cho việc cắt bớt có thể bị vô hiệu hóa nếu không cần thiết.

Các tập hợp của các trị số cắt bớt được sử dụng trong VTM5 được đưa ra trong bảng 2. 4 trị số đã được lựa chọn bằng cách tách gần như bằng nhau, trong miền logarit,

khoảng đầy đủ của các trị số mẫu (được tạo mã trên 10 bit) đối với Luma, và khoảng từ 4 đến 1024 đối với Chroma.

Một cách chính xác hơn, bảng Luma của các trị số cắt bớt đã được thu nhận bởi công thức sau đây:

$$\text{AlfClip}_L = \left\{ \text{round} \left( \left( (M)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\}, \text{ với } M=2^{10} \text{ và } N=4.$$

Tương tự như vậy, các bảng Chroma của các trị số cắt bớt được thu nhận theo công thức sau đây:

$$\text{AlfClip}_C = \left\{ \text{round} \left( A \cdot \left( \left( \frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\}, \text{ với } M=2^{10}, N=4 \text{ và } A=4.$$

A=4.

	Nhóm miếng bên trong/liên kết
LUMA	{ 1024, 181, 32, 6 }
CHROMA	{ 1024, 161, 25, 4 }

**Bảng 2: Các trị số cắt bớt được cho phép**

### **Bộ lọc trong vòng lặp khác**

Có tổng cộng ba bộ lọc trong vòng lặp trong VVC. Ngoài bộ lọc tách khối và SAO (hai bộ lọc vòng lặp trong HEVC), bộ lọc vòng lặp thích ứng (ALF) được áp dụng. ALF bao gồm ALF luma, ALF chroma và ALF thành phần chéo (cross-component ALF, viết tắt là CC-ALF). Quy trình lọc ALF được thiết kế sao cho ALF luma, ALF chroma và CC-ALF có thể được thực hiện song song. Thứ tự của quy trình lọc trong VVC là bộ lọc tách khối, SAO và ALF. SAO trong VVC giống như SAO trong HEVC.

Trong VVC, quy trình xử lý mới được gọi là việc ánh xạ luma với định tỷ lệ chroma đã được bổ sung (quy trình xử lý này đã được biết đến trước đó là bộ định hình lại trong vòng lặp thích ứng). LMCS điều chỉnh các trị số mẫu trước khi mã hóa và sau khi tái cấu trúc bằng cách phân phối lại các từ mã dọc theo toàn bộ khoảng động. Quy trình xử lý mới này được thực hiện trước khi tách khối.

### **Bộ lọc vòng lặp thích ứng**

Trong VVC, bộ lọc vòng lặp thích ứng (ALF) với việc thích ứng bộ lọc dựa trên khối được áp dụng. Đối với thành phần luma, một trong số 25 bộ lọc được lựa chọn cho

mỗi khối  $4 \times 4$ , dựa vào hướng và hoạt động của các gradien cục bộ.

### Hình dạng bộ lọc:

Hai hình dạng bộ lọc kim cương (như được thể hiện trên Fig.6) được sử dụng. Hình dạng kim cương  $7 \times 7$  được áp dụng cho thành phần luma và hình dạng kim cương  $5 \times 5$  được áp dụng cho các thành phần chroma.

### Phân loại khối:

Đối với thành phần luma, mỗi khối  $4 \times 4$  được phân loại thành một trong số 25 lớp. Chỉ số phân loại  $C$  được dẫn ra dựa vào hướng của nó  $D$  và trị số hoạt động được lượng tử hóa  $\hat{A}$ , như sau:

$$C = 5D + \hat{A} \quad (9)$$

Để tính toán  $D$  và  $\hat{A}$ , các gradien của hướng theo chiều ngang, theo chiều dọc và theo hai đường chéo đầu tiên được tính toán sử dụng 1-D Laplacian:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad (10)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad (11)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (12)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (13)$$

Trong đó các chỉ số  $i$  và  $j$  đề cập đến các tọa độ của mẫu trên cùng bên trái nằm trong khối  $4 \times 4$  và  $R(i,j)$  chỉ báo mẫu được tái cấu trúc ở tọa độ  $(i,j)$ .

Để làm giảm độ phức tạp của việc phân loại khối, việc tính toán 1-D Laplacian được lấy mẫu phụ được áp dụng. Như được thể hiện trên Fig.7, các vị trí được lấy mẫu phụ giống nhau được sử dụng cho việc tính toán gradien của tất cả các hướng.

Sau đó các trị số lớn nhất và nhỏ nhất  $D$  của các gradien của các hướng theo chiều ngang và theo chiều dọc được thiết đặt là:

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v) \quad (14)$$

Các trị số lớn nhất và nhỏ nhất của gradien của hai hướng theo đường chéo được thiết đặt là:



$$g_{d_0, d_1}^{max} = \max(g_{d_0}, g_{d_1}), \quad g_{d_0, d_1}^{min} = \min(g_{d_0}, g_{d_1}) \quad (15)$$

Để dẫn ra trị số của có hướng  $D$ , các trị số này được so với nhau và với hai ngưỡng  $t_1$  và  $t_2$ :

**Bước 1.** Nếu cả  $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$  và  $g_{d_0, d_1}^{max} \leq t_1 \cdot g_{d_0, d_1}^{min}$  là đúng, thì  $D$  được thiết đặt bằng 0.

**Bước 2.** Nếu  $g_{h,v}^{max}/g_{h,v}^{min} > g_{d_0, d_1}^{max}/g_{d_0, d_1}^{min}$ , tiếp tục từ bước 3; theo cách khác tiếp tục từ bước 4.

**Bước 3.** Nếu  $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ ,  $D$  được thiết đặt bằng 2; theo cách khác  $D$  được thiết đặt bằng 1.

**Bước 4.** Nếu  $g_{d_0, d_1}^{max} > t_2 \cdot g_{d_0, d_1}^{min}$ ,  $D$  được thiết đặt bằng 4; theo cách khác  $D$  được thiết đặt bằng 3.

Trị số hoạt động  $A$  được tính toán là:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \quad (16)$$

$A$  còn được lượng tử hóa tới khoảng từ 0 đến 4, và trị số được lượng tử hóa được biểu thị là  $\hat{A}$ .

Đối với các thành phần chroma trong ảnh, không có phương pháp phân loại nào được áp dụng, nghĩa là, tập hợp đơn của các hệ số ALF được áp dụng cho mỗi thành phần chroma.

### Các phép biến đổi hình học của các hệ số bộ lọc và các trị số cắt bớt

Trước khi lọc mỗi khối luma  $4 \times 4$ , các phép biến đổi hình học chẳng hạn như quay hoặc lật theo đường chéo hoặc theo chiều dọc được áp dụng cho các hệ số bộ lọc  $f(k, l)$  và cho các trị số cắt bớt bộ lọc tương ứng  $c(k, l)$  tùy thuộc vào các trị số gradient được tính toán cho khối đó. Điều này tương đương với việc áp dụng các phép biến đổi này cho các mẫu trong vùng hỗ trợ bộ lọc. Ý tưởng là làm cho các khối khác nhau mà trong đó ALF được áp dụng giống nhau hơn bằng cách căn chỉnh có hướng của chúng.

Ba phép biến đổi hình học, bao gồm lật theo đường chéo, theo chiều dọc và quay được giới thiệu:

$$\text{Theo đường chéo: } f_D(k, l) = f(l, k), \quad c_D(k, l) = c(l, k), \quad (25)$$

$$\text{Lật theo chiều dọc: } f_V(k, l) = f(k, K - l - 1), \quad c_V(k, l) = c(k, K - l - 1) \quad (26)$$

$$\text{Quay: } f_R(k, l) = f(K - l - 1, k), \quad c_R(k, l) = c(K - l - 1, k) \quad (27)$$

trong đó  $K$  là kích thước của bộ lọc và  $0 \leq k, l \leq K - 1$  là các tọa độ các hệ số, sao cho vị trí  $(0,0)$  ở góc trên cùng bên trái và vị trí  $(K - 1, K - 1)$  ở góc dưới cùng bên phải. Các phép biến đổi được áp dụng cho các hệ số bộ lọc  $f(k, l)$  và cho các trị số cắt bớt  $c(k, l)$  tùy thuộc vào các trị số gradient được tính toán cho khối đó. Mối tương quan giữa phép biến đổi và bốn gradient của bốn hướng được tóm tắt trong bảng sau đây.

**Bảng 1 Việc ánh xạ của gradient được tính toán cho một khối và các phép biến đổi**

Các trị số gradient	Phép biến đổi
$g_{d2} < g_{d1}$ và $g_h < g_v$	Không có phép biến đổi
$g_{d2} < g_{d1}$ và $g_v < g_h$	Theo đường chéo
$g_{d1} < g_{d2}$ và $g_h < g_v$	Lật theo chiều dọc
$g_{d1} < g_{d2}$ và $g_v < g_h$	Quay

### **Báo hiệu các thông số bộ lọc**

Các thông số bộ lọc ALF được báo hiệu trong tập hợp thông số thích ứng (Adaptation Parameter Set, viết tắt là APS). Trong một APS, lên đến 25 tập hợp của các hệ số bộ lọc luma và các chỉ số trị số cắt bớt, và lên đến tám tập hợp của các hệ số bộ lọc chroma và các chỉ số trị số cắt bớt có thể được báo hiệu. Để làm giảm phí tổn các bit, các hệ số bộ lọc của việc phân loại khác đối với thành phần luma có thể được hợp nhất. Ở đoạn đầu lát, các chỉ số của các APS được sử dụng cho lát hiện tại được báo hiệu.

Các chỉ số trị số cắt bớt, mà được giải mã từ APS, cho phép xác định các trị số cắt bớt nhờ sử dụng bảng của các trị số cắt bớt cho cả hai thành phần luma và chroma. Các trị số cắt bớt này phụ thuộc vào độ sâu bit bên trong. Một cách chính xác hơn, các trị số cắt bớt được thu nhận bởi công thức sau đây:

$$\text{AlfClip} = \{\text{round}(2^{B-\alpha*n}) \text{ đối với } n \in [0..N-1]\} \quad (28)$$

với B bằng độ sâu bit bên trong,  $\alpha$  là trị số không đổi được định rõ trước bằng 2,35, và N bằng 4 mà là số lượng của các trị số cắt bớt được cho phép trong VVC. AlfClip sau đó được làm tròn tới trị số gần nhất với định dạng của bội số của 2.

Ở đoạn đầu lát, lên đến 7 chỉ số APS có thể được báo hiệu để xác định các tập hợp bộ lọc luma mà được sử dụng cho lát hiện tại. Quy trình lọc còn có thể được điều khiển ở mức CTB. Cờ luôn được báo hiệu để chỉ báo xem ALF được áp dụng cho CTB luma hay không. CTB luma có thể lựa chọn tập hợp bộ lọc trong số 16 tập hợp bộ lọc cố định và các tập hợp bộ lọc từ các APS. Chỉ số tập hợp bộ lọc được báo hiệu cho CTB luma để chỉ báo tập hợp bộ lọc nào được áp dụng. 16 tập hợp bộ lọc cố định được định rõ trước và được tạo mã cứng trong cả bộ mã hóa và bộ giải mã.

Đối với thành phần chroma, chỉ số APS được báo hiệu trong đoạn đầu lát để chỉ báo các tập hợp bộ lọc chroma được sử dụng cho lát hiện tại. Ở mức CTB, chỉ số bộ lọc được báo hiệu cho mỗi CTB chroma nếu có nhiều hơn một tập hợp bộ lọc chroma trong APS.

Các hệ số bộ lọc được lượng tử hóa với định mức bằng 128. Để hạn chế độ phức tạp phép nhân, sự phù hợp dòng bit được áp dụng sao cho trị số hệ số của vị trí không trung tâm sẽ trong khoảng từ  $-2^7$  đến  $2^7 - 1$ . Hệ số vị trí trung tâm không được báo hiệu ở dòng bit và được xét đến là bằng 128.

### Quy trình lọc

Ở phía bộ giải mã, khi ALF được cho phép cho CTB, mỗi mẫu  $R(i, j)$  nằm trong CU được lọc, dẫn đến trị số mẫu  $R'(i, j)$  như được thể hiện dưới đây,

$$R'(i, j) = R(i, j) + \left( \left( \sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i+k, j+l) - R(i, j), c(k, l)) + 64 \right) \gg 7 \right) \quad (29)$$

trong đó  $f(k, l)$  biểu thị các hệ số bộ lọc được giải mã,  $K(x, y)$  là chức năng cắt bớt và  $c(k, l)$  biểu thị các thông số cắt bớt được giải mã. Biến k và l thay đổi giữa  $-\frac{L}{2}$  và  $\frac{L}{2}$  trong đó L biểu thị độ dài bộ lọc. Chức năng cắt bớt  $K(x, y) =$

$\min(y, \max(-y, x))$  mà tương ứng với chức năng  $Clip3(-y, y, x)$ . Thao tác cắt bớt giới thiệu tính phi tuyến tính để làm cho ALF hiệu quả hơn bằng cách làm giảm tác động của các trị số mẫu lân cận mà quá khác với trị số mẫu hiện tại.

Các trị số cắt bớt được lựa chọn được tạo mã trong thành phần cú pháp “alf\_data” nhờ sử dụng sơ đồ mã hóa Golomb tương ứng với chỉ số của trị số cắt bớt trong bảng 1 ở trên. Sơ đồ mã hóa này giống với sơ đồ mã hóa cho chỉ số bộ lọc. alf\_data có thể trong adaptation\_parameter\_set\_rbsp(), và adaptation\_parameter\_set\_rbsp() có thể được đề cập đến bởi đoạn đầu lát.

Các chi tiết cú pháp được thể hiện trong bảng dưới đây.

alf_data( ) {	Ký hiệu mô tả
<b>alf_chroma_idc</b>	tu(v)
<b>alf_luma_clip</b>	u(1)
if( alf_chroma_idc )	
<b>alf_chroma_clip</b>	u(1)
<b>alf_luma_num_filters_signalled_minus1</b>	tb(v)
if( alf_luma_num_filters_signalled_minus1 > 0 ) {	
for( filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++ )	
<b>alf_luma_coeff_delta_idx[ filtIdx ]</b>	tb(v)
}	
<b>alf_luma_coeff_delta_flag</b>	u(1)
if ( !alf_luma_coeff_delta_flag && alf_luma_num_filters_signalled_minus1 > 0 )	
<b>alf_luma_coeff_delta_prediction_flag</b>	u(1)
<b>alf_luma_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 3; i++ )	
<b>alf_luma_eg_order_increase_flag[ i ]</b>	u(1)
if ( alf_luma_coeff_delta_flag ) {	
for( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ )	
<b>alf_luma_coeff_flag[ sigFiltIdx ]</b>	u(1)
}	
for( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ ) {	
if ( alf_luma_coeff_flag[ sigFiltIdx ] ) {	

for ( j = 0; j < 12; j++ ) {	
<b>alf_luma_coeff_delta_abs</b> [ sigFiltIdx ][ j ]	uek(v)
if( alf_luma_coeff_delta_abs[ sigFiltIdx ][ j ] )	
<b>alf_luma_coeff_delta_sign</b> [ sigFiltIdx ][ j ]	u(1)
}	
}	
}	
if( alf_luma_clip ) {	
<b>alf_luma_clip_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 3; i++ )	
<b>alf_luma_clip_eg_order_increase_flag</b> [ i ]	u(1)
for ( sigFiltIdx = 0; sigFiltIdx <=	
alf_luma_num_filters_signalled_minus1; sigFiltIdx++ ) {	
if ( alf_luma_coeff_flag[ sigFiltIdx ] ) {	
for ( j = 0; j < 12; j++ ) {	
if( filterCoefficients[ sigFiltIdx ][ j ] )	
<b>alf_luma_clip_idx</b> [ sigFiltIdx ][ j ]	uek(v)
}	
}	
}	
}	
}	
if ( alf_chroma_idc > 0 ) {	
<b>alf_chroma_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 2; i++ )	
<b>alf_chroma_eg_order_increase_flag</b> [ i ]	u(1)
for( j = 0; j < 6; j++ ) {	
<b>alf_chroma_coeff_abs</b> [ j ]	uek(v)
if( alf_chroma_coeff_abs[ j ] > 0 )	
<b>alf_chroma_coeff_sign</b> [ j ]	u(1)
}	
}	
if ( alf_chroma_idc > 0 && alf_chroma_clip ) {	
<b>alf_chroma_clip_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 2; i++ )	
<b>alf_chroma_clip_eg_order_increase_flag</b> [ i ]	u(1)
for( j = 0; j < 6; j++ ) {	
if( alf_chroma_coeff_abs[ j ] )	

<b>alf_chroma_clip_idx[ j ]</b>	<b>uek(v)</b>
}	
}	
}	

Ngữ nghĩa của các thành phần cú pháp mới được giới thiệu như sau:

**alf\_luma\_clip** bằng 0 định rõ rằng bộ lọc vòng lặp thích ứng tuyến tính được áp dụng trên thành phần luma. **alf\_luma\_clip** bằng 1 định rõ rằng bộ lọc vòng lặp thích ứng phi tuyến tính có thể được áp dụng trên thành phần luma.

**alf\_chroma\_clip** bằng 0 định rõ rằng bộ lọc vòng lặp thích ứng tuyến tính được áp dụng trên các thành phần chroma; **alf\_chroma\_clip** bằng 1 định rõ rằng bộ lọc vòng lặp thích ứng phi tuyến tính được áp dụng trên thành phần chroma. Nếu không có mặt **alf\_chroma\_clip** được suy luận bằng 0.

**alf\_luma\_clip\_min\_eg\_order\_minus1** cộng 1 định rõ thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt luma. Trị số của **alf\_luma\_clip\_min\_eg\_order\_minus1** sẽ trong khoảng từ 0 đến 6.

**alf\_luma\_clip\_eg\_order\_increase\_flag[ i ]** bằng 1 định rõ rằng thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt luma được tăng lên thêm 1. **alf\_luma\_clip\_eg\_order\_increase\_flag[ i ]** bằng 0 định rõ rằng thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt luma không được tăng lên thêm 1.

Thứ tự **expGoOrderYClip[ i ]** của mã exp-Golomb được sử dụng để giải mã các trị số của **alf\_luma\_clip\_idx[ sigFiltIdx ][ j ]** được dẫn ra như sau:

$$\text{expGoOrderYClip}[ i ] = \text{alf\_luma\_clip\_min\_eg\_order\_minus1} + 1 + \text{alf\_luma\_clip\_eg\_order\_increase\_flag}[ i ]$$

**alf\_luma\_clip\_idx[ sigFiltIdx ][ j ]** định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số thứ j của bộ lọc luma được báo hiệu được chỉ báo bởi **sigFiltIdx**. Khi **alf\_luma\_clip\_idx[ sigFiltIdx ][ j ]** không có mặt, nó được suy luận bằng 0 (không cắt bớt).

Thứ tự k của **uek(v)** nhị phân hóa exp-Golomb được dẫn ra như sau:

$$\text{golombOrderIdxYClip}[ ] = \{ 0, 0, 1, 0, 0, 1, 2, 1, 0, 0, 1, 2 \}$$

$k = \text{expGoOrderYClip}[\text{golombOrderIdxYClip}[j]]$

Biến  $\text{filterClips}[\text{sigFiltIdx}][j]$  với  $\text{sigFiltIdx} = 0..\text{alf\_luma\_num\_filters\_signalled\_minus1}$ ,  $j = 0..11$  được khởi tạo như sau: Biến  $\text{NumYClipValue}$  được thiết đặt bằng 4.

Đối với  $i = 0..\text{NumYClipValue} - 1$

$\text{alf\_luma\_clipping\_value}[i] = \text{Round} ( 2^{(\text{BitDepthY} * (\text{NumYClipValue} - i)) / \text{NumYClipValue} )$

$\text{filterClips}[\text{sigFiltIdx}][j] = \text{alf\_luma\_clipping\_value}[\text{alf\_luma\_clip\_idx}[\text{sigFiltIdx}][j]]$

Các trị số cắt bớt bộ lọc luma  $\text{AlfClip}_L$  với các thành phần  $\text{AlfClip}_L[\text{filtIdx}][j]$ , với  $\text{filtIdx} = 0..\text{NumAlfFilters} - 1$  và  $j = 0..11$  được dẫn ra như sau

$\text{AlfClip}_L[\text{filtIdx}][j] = \text{filterClips}[\text{alf\_luma\_coeff\_delta\_idx}[\text{filtIdx}][j]]$

**alf\_chroma\_clip\_min\_eg\_order\_minus1** cộng 1 định rõ thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt chroma. Trị số của  $\text{alf\_chroma\_clip\_min\_eg\_order\_minus1}$  sẽ trong khoảng từ 0 đến 6.

**alf\_chroma\_clip\_eg\_order\_increase\_flag**[i] bằng 1 định rõ rằng thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt chroma được tăng lên thêm 1.  $\text{alf\_chroma\_clip\_eg\_order\_increase\_flag}[i]$  bằng 0 định rõ rằng thứ tự nhỏ nhất của mã exp-Golomb để báo hiệu các chỉ số cắt bớt chroma không được tăng lên thêm 1.

Thứ tự  $\text{expGoOrderC}[i]$  của mã exp-Golomb được sử dụng để giải mã các trị số của  $\text{alf\_chroma\_clip\_idx}[j]$  được dẫn ra như sau:

$\text{expGoOrderC}[i] = \text{alf\_chroma\_clip\_min\_eg\_order\_minus1} + 1 + \text{alf\_chroma\_clip\_eg\_order\_increase\_flag}[i]$

**alf\_chroma\_clip\_idx**[j] định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số thứ j của bộ lọc chroma. Khi  $\text{alf\_chroma\_clip\_idx}[j]$  không có mặt, nó được suy luận bằng 0 (không cắt bớt).

Thứ tự k của uek(v) nhị phân hóa exp-Golomb được dẫn ra như sau:

$\text{golombOrderIdxC}[] = \{ 0, 0, 1, 0, 0, 1 \}$

$k = \text{expGoOrderC}[\text{golombOrderIdxC}[j]]$

Các trị số cắt bớt bộ lọc chroma  $AlfClip_c$  với các thành phần  $AlfClip_c [ j ]$ , với  $j = 0..5$  được dẫn ra như sau:

Biến  $NumCClipValue$  được thiết đặt bằng 4.

Đối với  $i = 0..NumCClipValue - 1$

$$alf\_chroma\_clipping\_value[i] = Round ( 2^{(BitDepthC - 8)} * 2^{(8 * (NumCClipValue - i - 1)/(NumCClipValue - 1))})$$

$$AlfClip_c [ j ] = alf\_chroma\_clipping\_value [ alf\_chroma\_clip\_idx [ j ] ]$$

Đặc tả cú pháp ALF theo đặc tả VVC

### Quy trình bộ lọc vòng lặp thích ứng

#### 1.1 Khái quát

Các đầu vào của quy trình xử lý này là các mảng mẫu ảnh được tái cấu trúc trước bộ lọc vòng lặp thích ứng  $recPictureL$ ,  $recPictureCb$  và  $recPictureCr$ .

Các đầu ra của quy trình xử lý này là các mảng mẫu ảnh được tái cấu trúc được điều chỉnh sau bộ lọc vòng lặp thích ứng  $alfPictureL$ ,  $alfPictureCb$  và  $alfPictureCr$ .

Các trị số mẫu trong các mảng mẫu ảnh được tái cấu trúc được điều chỉnh sau bộ lọc vòng lặp thích ứng  $alfPictureL$ ,  $alfPictureCb$  và  $alfPictureCr$ , được thiết đặt ban đầu lần lượt bằng các trị số mẫu trong các mảng mẫu ảnh được tái cấu trúc trước bộ lọc vòng lặp thích ứng  $recPictureL$ ,  $recPictureCb$  và  $recPictureCr$ .

Khi trị số của  $tile\_group\_alf\_enabled\_flag$  bằng 1, đối với mọi bộ phận cây tạo mã với vị trí khối cây tạo mã luma ( $rx, ry$ ), trong đó  $rx = 0..PicWidthInCtbs - 1$  và  $ry = 0..PicHeightInCtbs - 1$ , quy trình xử lý sau đây được áp dụng:

Khi trị số của  $alf\_ctb\_flag [ 0 ] [ rx ] [ ry ]$  bằng 1, quy trình lọc khối cây tạo mã cho các mẫu luma như được quy định trong khoản 1.2 được truy xuất với  $recPictureL$ ,  $alfPictureL$ , và vị trí khối cây tạo mã luma ( $xCtb, yCtb$ ) được thiết đặt bằng ( $rx \ll CtbLog2SizeY, ry \ll CtbLog2SizeY$ ) là các đầu vào, và đầu ra là ảnh được lọc được điều chỉnh  $alfPictureL$ .

Khi trị số của  $alf\_ctb\_flag [ 1 ] [ rx ] [ ry ]$  bằng 1, quy trình lọc khối cây tạo mã cho các mẫu chroma như được quy định trong khoản 1.1 được truy xuất với  $recPicture$  được thiết đặt bằng  $recPictureCb$ ,  $alfPicture$  được thiết đặt bằng  $alfPictureCb$ , và vị trí khối cây tạo mã chroma ( $xCtbC, yCtbC$ ) được thiết đặt bằng



(  $rx \ll ( CtbLog2SizeY - 1 )$ ,  $ry \ll ( CtbLog2SizeY - 1 )$  ) là các đầu vào, và đầu ra là ảnh được lọc được điều chỉnh  $alfPictureCb$ .

Khi trị số của  $alf\_ctb\_flag[ 2 ][ rx ][ ry ]$  bằng 1, quy trình lọc khối cây tạo mã cho các mẫu chroma như được quy định trong khoản 1.4 được truy xuất với  $recPicture$  được thiết đặt bằng  $recPictureCr$ ,  $alfPicture$  được thiết đặt bằng  $alfPictureCr$ , và vị trí khối cây tạo mã chroma (  $xCtbC$ ,  $yCtbC$  ) được thiết đặt bằng (  $rx \ll ( CtbLog2SizeY - 1 )$ ,  $ry \ll ( CtbLog2SizeY - 1 )$  ) là các đầu vào, và đầu ra là ảnh được lọc được điều chỉnh  $alfPictureCr$ .

## 1.2 Quy trình lọc khối cây tạo mã cho các mẫu luma

Các đầu vào của quy trình xử lý này là:

mảng mẫu ảnh luma được tái cấu trúc  $recPictureL$  trước quy trình lọc vòng lặp thích ứng,

mảng mẫu ảnh luma được tái cấu trúc được lọc  $alfPictureL$ ,

vị trí luma (  $xCtb$ ,  $yCtb$  ) định rõ mẫu trên cùng bên trái của khối cây tạo mã luma hiện tại liên quan đến mẫu trên cùng bên trái của ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu ảnh luma được tái cấu trúc được lọc được điều chỉnh  $alfPictureL$ .

Quy trình dẫn ra cho khoản chỉ số bộ lọc 1.3 được truy xuất với vị trí (  $xCtb$ ,  $yCtb$  ) và mảng mẫu ảnh luma được tái cấu trúc  $recPictureL$  là các đầu vào, và  $filtIdx[ x ][ y ]$  và  $transposeIdx[ x ][ y ]$  với  $x, y = 0..CtbSizeY - 1$  là các đầu ra.

Đối với việc dẫn ra của các mẫu luma được tái cấu trúc được lọc  $alfPictureL[ x ][ y ]$ , mỗi mẫu luma được tái cấu trúc bên trong khối cây tạo mã luma hiện tại  $recPictureL[ x ][ y ]$  được lọc như sau với  $x, y = 0..CtbSizeY - 1$ :

Mảng của các hệ số bộ lọc luma  $f[ j ]$  tương ứng với bộ lọc được quy định bởi  $filtIdx[ x ][ y ]$  được dẫn ra như sau với  $j = 0..12$ :

$$f[ j ] = AlfCoeffL[ filtIdx[ x ][ y ] ][ j ].$$

– Mảng của các trị số cắt bớt bộ lọc luma  $c[ j ]$  tương ứng với bộ lọc được quy định bởi  $filtIdx[ x ][ y ]$  được dẫn ra như sau với  $j = 0..11$ :

$$c[ j ] = AlfClipL[ filtIdx[ x ][ y ] ][ j ]$$

Các hệ số bộ lọc bộ lọc luma  $filterCoeff$  được dẫn ra tùy thuộc vào  $transposeIdx[ x ][ y ]$  như sau:

Nếu  $\text{transposeIndex}[x][y] = 1$ ,

$\text{filterCoeff}[] = \{ f[9], f[4], f[10], f[8], f[1], f[5], f[11], f[7], f[3], f[0], f[2], f[6], f[12] \}$

$\text{filterClip}[] = \{ c[9], c[4], c[10], c[8], c[1], c[5], c[11], c[7], c[3], c[0], c[2], c[6] \}$

Theo cách khác, nếu  $\text{transposeIndex}[x][y] = 2$ ,

$\text{filterCoeff}[] = \{ f[0], f[3], f[2], f[1], f[8], f[7], f[6], f[5], f[4], f[9], f[10], f[11], f[12] \}$

$\text{filterClip}[] = \{ c[0], c[3], c[2], c[1], c[8], c[7], c[6], c[5], c[4], c[9], c[10], c[11] \}$

Theo cách khác, nếu  $\text{transposeIndex}[x][y] = 3$ ,

$\text{filterCoeff}[] = \{ f[9], f[8], f[10], f[4], f[3], f[7], f[11], f[5], f[1], f[0], f[2], f[6], f[12] \}$

$\text{filterClip}[] = \{ c[9], c[8], c[10], c[4], c[3], c[7], c[11], c[5], c[1], c[0], c[2], c[6] \}$

Theo cách khác,

$\text{filterCoeff}[] = \{ f[0], f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9], f[10], f[11], f[12] \}$

$\text{filterClip}[] = \{ c[0], c[1], c[2], c[3], c[4], c[5], c[6], c[7], c[8], c[9], c[10], c[11] \}$

Các vị trí (hx, vy) cho mỗi trong số các mẫu luma tương ứng (x, y) bên trong mảng đã cho `recPicture` của các mẫu luma được dẫn ra như sau:

$hx = \text{Clip3}(0, \text{pic\_width\_in\_luma\_samples} - 1, x\text{Ctb} + x)$

$vy = \text{Clip3}(0, \text{pic\_height\_in\_luma\_samples} - 1, y\text{Ctb} + y)$

Tổng biến được dẫn ra như sau:

– Tổng biến được dẫn ra như sau:

$\text{sum} = \text{filterCoeff}[0] * (\text{Clip3}(-\text{filterClip}[0], \text{filterClip}[0], \text{recPictureL}[hx, vy+3] - \text{curr}) +$

$\text{Clip3}(-\text{filterClip}[0], \text{filterClip}[0], \text{recPictureL}[hx, vy-3] - \text{curr})) +$

$\text{filterCoeff}[1] * (\text{Clip3}(-\text{filterClip}[1], \text{filterClip}[1], \text{recPictureL}[hx+1, vy+2] - \text{curr}) +$   
 $\text{Clip3}(-\text{filterClip}[1], \text{filterClip}[1], \text{recPictureL}[hx-1, vy-2] - \text{curr})) +$

$\text{filterCoeff}[2] * (\text{Clip3}(-\text{filterClip}[2], \text{filterClip}[2], \text{recPictureL}[hx, vy+2] - \text{curr}) +$   
 $\text{Clip3}(-\text{filterClip}[2], \text{filterClip}[2], \text{recPictureL}[hx, vy-2] - \text{curr})) +$

```

filterCoeff[ 3 ] * ( Clip3(-filterClip[3], filterClip[3], recPictureL[ hx-1, vy+2 ] -
curr) + Clip3(-filterClip[3], filterClip[3], recPictureL[ hx+1, vy-2 ] - curr) ) +
filterCoeff[ 4 ] * ( Clip3(-filterClip[4], filterClip[4], recPictureL[ hx+2, vy+1 ] -
curr) + Clip3(-filterClip[4], filterClip[4], recPictureL[ hx-2, vy-1 ] - curr) ) +
filterCoeff[ 5 ] * ( Clip3(-filterClip[5], filterClip[5], recPictureL[ hx+1, vy+1 ] -
curr) + Clip3(-filterClip[5], filterClip[5], recPictureL[ hx-1, vy-1 ] - curr) ) +
filterCoeff[ 6 ] * ( Clip3(-filterClip[6], filterClip[6], recPictureL[ hx, vy+1 ] -
curr) + Clip3(-filterClip[6], filterClip[6], recPictureL[ hx, vy-1 ] - curr) ) +
filterCoeff[ 7 ] * ( Clip3(-filterClip[7], filterClip[7], recPictureL[ hx-1, vy+1 ] -
curr) + Clip3(-filterClip[7], filterClip[7], recPictureL[ hx+1, vy-1 ] - curr) ) +
filterCoeff[ 8 ] * ( Clip3(-filterClip[8], filterClip[8], recPictureL[ hx-2, vy+1 ] -
curr) + Clip3(-filterClip[8], filterClip[8], recPictureL[ hx+2, vy-1 ] - curr) ) +
filterCoeff[ 9 ] * ( Clip3(-filterClip[9], filterClip[9], recPictureL[ hx+3, vy ] -
curr) + Clip3(-filterClip[9], filterClip[9], recPictureL[ hx-3, vy ] - curr) ) +
filterCoeff[ 10 ] * ( Clip3(-filterClip[10], filterClip[10], recPictureL[ hx+2, vy ] -
curr) + Clip3(-filterClip[10], filterClip[10], recPictureL[ hx-2, vy ] - curr) ) +
filterCoeff[ 11 ] * ( Clip3(-filterClip[11], filterClip[11], recPictureL[ hx+1, vy ] -
curr) + Clip3(-filterClip[11], filterClip[11], recPictureL[ hx-1, vy ] - curr) )
sum = curr + (( sum + 64 ) >> 7)

```

Mẫu ảnh luma được tái cấu trúc được lọc được điều chỉnh  $\text{alfPictureL}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y]$  được dẫn ra như sau:

$$\text{alfPictureL}[x_{\text{Ctb}} + x][y_{\text{Ctb}} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, \text{sum}).$$

### 1.3 Quy trình dẫn ra đối với việc chuyển vị ALF và chỉ số bộ lọc cho các mẫu luma

Các đầu vào của quy trình xử lý này là:

vị trí luma ( $x_{\text{Ctb}}, y_{\text{Ctb}}$ ) định rõ mẫu trên cùng bên trái của khối cây tạo mã luma hiện tại liên quan đến mẫu trên cùng bên trái của ảnh hiện tại,

mảng mẫu ảnh luma được tái cấu trúc  $\text{recPictureL}$  trước quy trình lọc vòng lặp thích ứng.

Các đầu ra của quy trình xử lý này là

mảng chỉ số bộ lọc phân loại  $\text{filtIdx}[x][y]$  với  $x, y = 0..C_{\text{tb}}\text{SizeY} - 1$ ,

mảng chỉ số chuyển vị  $\text{transposeIdx}[x][y]$  với  $x, y = 0..CtbSizeY - 1$ .

Các vị trí  $(hx, vy)$  cho mỗi trong số các mẫu luma tương ứng  $(x, y)$  bên trong mảng đã cho  $\text{recPicture}$  của các mẫu luma được dẫn ra như sau:

$$hx = \text{Clip3}(0, \text{pic\_width\_in\_luma\_samples} - 1, x)$$

$$vy = \text{Clip3}(0, \text{pic\_height\_in\_luma\_samples} - 1, y)$$

Mảng chỉ số bộ lọc phân loại  $\text{filtIdx}$  và mảng chỉ số chuyển vị  $\text{transposeIdx}$  được dẫn ra bởi các bước theo thứ tự sau đây:

Các biến  $\text{filtH}[x][y]$ ,  $\text{filtV}[x][y]$ ,  $\text{filtD0}[x][y]$  và  $\text{filtD1}[x][y]$  với  $x, y = -2..CtbSizeY + 1$  được dẫn ra như sau:

Nếu cả  $x$  và  $y$  là các số chẵn hoặc cả  $x$  và  $y$  là các số lẻ, được áp dụng như sau:

$$\text{filtH}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y] - \text{recPicture}[hxCtb+x+1, vyCtb+y])$$

$$\text{filtV}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x, vyCtb+y-1] - \text{recPicture}[hxCtb+x, vyCtb+y+1])$$

$$\text{filtD0}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x-1, vyCtb+y-1] - \text{recPicture}[hxCtb+x+1, vyCtb+y+1])$$

$$\text{filtD1}[x][y] = \text{Abs}(\text{recPicture}[hxCtb+x, vyCtb+y] \ll 1) - \text{recPicture}[hxCtb+x+1, vyCtb+y-1] - \text{recPicture}[hxCtb+x-1, vyCtb+y+1])$$

Theo cách khác,  $\text{filtH}[x][y]$ ,  $\text{filtV}[x][y]$ ,  $\text{filtD0}[x][y]$  và  $\text{filtD1}[x][y]$  được thiết đặt bằng 0.

Các biến  $\text{varTempH1}[x][y]$ ,  $\text{varTempV1}[x][y]$ ,  $\text{varTempD01}[x][y]$ ,  $\text{varTempD11}[x][y]$  và  $\text{varTemp}[x][y]$  với  $x, y = 0..(CtbSizeY - 1) \gg 2$  được dẫn ra như sau:

$$\text{sumH}[x][y] = \sum_i \sum_j \text{filtH}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumV}[x][y] = \sum_i \sum_j \text{filtV}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD0}[x][y] = \sum_i \sum_j \text{filtD0}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumD1}[x][y] = \sum_i \sum_j \text{filtD1}[(x \ll 2) + i][(y \ll 2) + j] \text{ với } i, j = -2..5$$

$$\text{sumOfHV}[x][y] = \text{sumH}[x][y] + \text{sumV}[x][y]$$

Các biến  $\text{dir1}[x][y]$ ,  $\text{dir2}[x][y]$  và  $\text{dirS}[x][y]$  với  $x, y = 0..CtbSizeY - 1$  được dẫn ra như sau:

Các biến  $\text{hv1}$ ,  $\text{hv0}$  và  $\text{dirHV}$  được dẫn ra như sau:

Nếu  $\text{sumV}[x \gg 2][y \gg 2]$  lớn hơn  $\text{sumH}[x \gg 2][y \gg 2]$ , được áp dụng như sau:

$$\text{hv1} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 1$$

Theo cách khác, được áp dụng như sau:

$$\text{hv1} = \text{sumH}[x \gg 2][y \gg 2]$$

$$\text{hv0} = \text{sumV}[x \gg 2][y \gg 2]$$

$$\text{dirHV} = 3$$

Các biến  $d1$ ,  $d0$  và  $\text{dirD}$  được dẫn ra như sau:

Nếu  $\text{sumD0}[x \gg 2][y \gg 2]$  lớn hơn  $\text{sumD1}[x \gg 2][y \gg 2]$ , được áp dụng như sau:

$$d1 = \text{sumD0}[x \gg 2][y \gg 2]$$

$$d0 = \text{sumD1}[x \gg 2][y \gg 2]$$

$$\text{dirD} = 0$$

Theo cách khác, được áp dụng như sau:

$$d1 = \text{sumD1}[x \gg 2][y \gg 2]$$

$$d0 = \text{sumD0}[x \gg 2][y \gg 2]$$

$$\text{dirD} = 2$$

Các biến  $\text{hvd1}$ ,  $\text{hvd0}$ , được dẫn ra như sau:

$$\text{hvd1} = (d1 * \text{hv0} > \text{hv1} * d0) ? d1 : \text{hv1}$$

$$\text{hvd0} = (d1 * \text{hv0} > \text{hv1} * d0) ? d0 : \text{hv0}$$

Các biến  $\text{dirS}[x][y]$ ,  $\text{dir1}[x][y]$  và  $\text{dir2}[x][y]$  được dẫn ra như sau:

$$\text{dir1}[x][y] = (d1 * \text{hv0} > \text{hv1} * d0) ? \text{dirD} : \text{dirHV}$$

$$\text{dir2}[x][y] = (d1 * \text{hv0} > \text{hv1} * d0) ? \text{dirHV} : \text{dirD}$$

$$\text{dirS}[x][y] = (\text{hvd1} > 2 * \text{hvd0}) ? 1 : ((\text{hvd1} * 2 > 9 * \text{hvd0}) ? 2 : 0)$$

Biến  $\text{avgVar}[x][y]$  với  $x, y = 0..CtbSizeY - 1$  được dẫn ra như sau:

$$\text{varTab}[] = \{ 0, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4 \}$$

$$\text{avgVar}[x][y] = \text{varTab}[\text{Clip3}(0, 15, (\text{sumOfHV}[x \gg 2][y \gg 2] * 64) \gg (3 + \text{BitDepthY}))]$$

Mảng chỉ số bộ lọc phân loại  $\text{filtIdx}[x][y]$  và mảng chỉ số chuyển vị

$\text{transposeIdx}[x][y]$  với  $x = y = 0..CtbSizeY - 1$  được dẫn ra như sau:

$$\text{transposeTable}[] = \{ 0, 1, 0, 2, 2, 3, 1, 3 \}$$

$$\text{transposeIdx}[x][y] = \text{transposeTable}[\text{dir1}[x][y] * 2 + (\text{dir2}[x][y] \gg 1)]$$

$$\text{filtIdx}[x][y] = \text{avgVar}[x][y]$$

Khi  $\text{dirS}[x][y]$  không bằng 0,  $\text{filtIdx}[x][y]$  được điều chỉnh như sau:

$$\text{filtIdx}[x][y] += (((\text{dir1}[x][y] \& 0x1) \ll 1) + \text{dirS}[x][y]) * 5.$$

#### 1.4 Quy trình lọc khối cây tạo mã cho các mẫu chroma

Các đầu vào của quy trình xử lý này là:

mảng mẫu ảnh chroma được tái cấu trúc  $\text{recPicture}$  trước quy trình lọc vòng lặp thích ứng,

mảng mẫu ảnh chroma được tái cấu trúc được lọc  $\text{alfPicture}$ ,

vị trí chroma ( $xCtbC, yCtbC$ ) định rõ mẫu trên cùng bên trái của khối cây tạo mã chroma hiện tại liên quan đến mẫu trên cùng bên trái của ảnh hiện tại.

Đầu ra của quy trình xử lý này là mảng mẫu ảnh chroma được tái cấu trúc được lọc được điều chỉnh  $\text{alfPicture}$ .

Kích thước của khối cây tạo mã chroma hiện tại  $\text{ctbSizeC}$  được dẫn ra như sau:

$$\text{ctbSizeC} = \text{CtbSizeY} / \text{SubWidthC}$$

Đối với việc dẫn ra của các mẫu chroma được tái cấu trúc được lọc  $\text{alfPicture}[x][y]$ , mỗi mẫu chroma được tái cấu trúc bên trong khối cây tạo mã chroma hiện tại  $\text{recPicture}[x][y]$  được lọc như sau với  $x, y = 0..ctbSizeC - 1$ :

Các vị trí ( $hx, vy$ ) cho mỗi trong số các mẫu chroma tương ứng ( $x, y$ ) bên trong mảng đã cho  $\text{recPicture}$  của các mẫu chroma được dẫn ra như sau:

$$hx = \text{Clip3}(0, \text{pic\_width\_in\_luma\_samples} / \text{SubWidthC} - 1, xCtbC + x)$$

$$vy = \text{Clip3}(0, \text{pic\_height\_in\_luma\_samples} / \text{SubHeightC} - 1, yCtbC + y)$$

Tổng biến được dẫn ra như sau:

$$\text{sum} = \text{AlfCoeffc}[0] * (\text{Clip3}(-\text{AlfClipc}[0], \text{AlfClipc}[0], \text{recPicture}[hx, vy+2] - \text{curr}) + \text{Clip3}(-\text{AlfClipc}[0], \text{AlfClipc}[0], \text{recPicture}[hx, vy-2] - \text{curr})) +$$

$$\text{AlfCoeffc}[1] * (\text{Clip3}(-\text{AlfClipc}[1], \text{AlfClipc}[1], \text{recPicture}[hx+1, vy+1] - \text{curr}) + \text{Clip3}(-\text{AlfClipc}[1], \text{AlfClipc}[1], \text{recPicture}[hx-1, vy-1] - \text{curr})) +$$

$$\text{AlfCoeffc}[2] * (\text{Clip3}(-\text{AlfClipc}[2], \text{AlfClipc}[2], \text{recPicture}[hx, vy+1] - \text{curr}) + \text{Clip3}(-\text{AlfClipc}[2], \text{AlfClipc}[2], \text{recPicture}[hx, vy-1] - \text{curr})) +$$

$$\begin{aligned} & \text{AlfCoeffc}[3] * (\text{Clip3}(-\text{AlfClipc}[3], \text{AlfClipc}[3], \text{recPicture}[h_x - 1, v_y + 1] - \\ & \text{curr}) + \text{Clip3}(-\text{AlfClipc}[3], \text{AlfClipc}[3], \text{recPicture}[h_x + 1, v_y - 1] - \text{curr}) + \\ & \text{AlfCoeffc}[4] * (\text{Clip3}(-\text{alfClipc}[4], \text{alfClipc}[4], \text{recPicture}[h_x + 2, v_y] - \\ & \text{curr}) + \text{Clip3}(-\text{AlfClipc}[4], \text{AlfClipc}[4], \text{recPicture}[h_x - 2, v_y] - \text{curr}) + \\ & \text{AlfCoeffc}[5] * (\text{Clip3}(-\text{AlfClipc}[5], \text{AlfClipc}[5], \text{recPicture}[h_x + 1, v_y] - \text{curr}) + \\ & \text{Clip3}(-\text{AlfClipc}[5], \text{AlfClipc}[5], \text{recPicture}[h_x - 1, v_y] - \text{curr})) \\ & \text{sum} = \text{curr} + ((\text{sum} + 64) \gg 7) \end{aligned}$$

Mẫu ảnh chroma được tái cấu trúc được lọc được điều chỉnh  $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y]$  được dẫn ra như sau:

$$\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, \text{sum}) .$$

Như được nêu trên và như được thể hiện trên Fig.8, các thông số cắt bớt Luma và Chroma ALF được truyền nhờ sử dụng các mã golomb theo cấp số nhân thứ tự thứ K giống với các hệ số bộ lọc ALF.

Việc sử dụng của các mã golomb theo cấp số nhân thứ tự thứ K đối với các thông số cắt bớt có thể không hữu hiệu về mặt hiệu quả tạo mã, như thông số cắt bớt được báo hiệu chỉ là chỉ số trong bảng của các trị số cắt bớt (xem bảng 2 nêu trên). Trị số của các khoảng chỉ số từ 0 đến 3.

Do đó việc báo hiệu các trị số chỉ số 0 đến 3 sử dụng các mã golomb theo cấp số nhân thứ tự thứ K theo cách giống như các hệ số bộ lọc ALF sử dụng các thành phần cú pháp bổ sung, **alf\_luma\_clip\_min\_eg\_order\_minus1**, **alf\_luma\_clip\_eg\_order\_increase\_flag[i]** để xác định trị số K (thứ tự của mã golomb theo cấp số nhân được sử dụng) và sau đó thành phần cú pháp **alf\_luma\_clip\_idx** được báo hiệu nhờ sử dụng mã golomb theo cấp số nhân thứ tự K. Do đó phương pháp báo hiệu này phức tạp và cũng không hữu hiệu về mặt hiệu quả tạo mã. Do đó phương pháp báo hiệu đơn giản hơn các thông số cắt bớt là điều mong muốn.

Theo một phương án của giải pháp được đề xuất (giải pháp 1), như được thể hiện trên Fig.9, các thông số cắt bớt là các mã độ dài cố định được sử dụng được báo hiệu và do đó các thành phần cú pháp **alf\_luma\_clip\_min\_eg\_order\_minus1**, **alf\_luma\_clip\_eg\_order\_increase\_flag[i]** không được sử dụng. Thành phần cú pháp **alf\_luma\_clip\_idx** được báo hiệu nhờ sử dụng mã độ dài cố định của 2 bit. Phương pháp này có ưu điểm rằng các thông số cắt bớt được báo hiệu theo cách rất đơn giản,

hiệu quả tạo mã được nâng cao khi ít trong số các thành phần cú pháp liên quan đến các mã golomb theo cấp số nhân thứ tự thứ K không được báo hiệu nữa.

Cú pháp `alf_data` được điều chỉnh như sau:

<code>alf_data() {</code>	Ký hiệu mô tả
<code>alf_chroma_idc</code>	tu(v)
<code>alf_luma_clip</code>	u(1)
<code>if( alf_chroma_idc )</code>	
<code>alf_chroma_clip</code>	u(1)
<code>alf_luma_num_filters_signalled_minus1</code>	tb(v)
<code>if( alf_luma_num_filters_signalled_minus1 &gt; 0 ) {</code>	
<code>for( filtIdx = 0; filtIdx &lt; NumAlfFilters; filtIdx++ )</code>	
<code>alf_luma_coeff_delta_idx[ filtIdx ]</code>	tb(v)
<code>}</code>	
<code>alf_luma_coeff_delta_flag</code>	u(1)
<code>if ( !alf_luma_coeff_delta_flag &amp;&amp; alf_luma_num_filters_signalled_minus1 &gt; 0 )</code>	
<code>alf_luma_coeff_delta_prediction_flag</code>	u(1)
<code>alf_luma_min_eg_order_minus1</code>	ue(v)
<code>for( i = 0; i &lt; 3; i++ )</code>	
<code>alf_luma_eg_order_increase_flag[ i ]</code>	u(1)



if ( alf_luma_coeff_delta_flag ) {	
for( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ )	
<b>alf_luma_coeff_flag</b> [ sigFiltIdx ]	u(1)
}	
for( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ ) {	
if ( alf_luma_coeff_flag[ sigFiltIdx ] ) {	
for ( j = 0; j < 12; j++ ) {	
<b>alf_luma_coeff_delta_abs</b> [ sigFiltIdx ][ j ]	uek(v)
if( alf_luma_coeff_delta_abs[ sigFiltIdx ][ j ] )	
<b>alf_luma_coeff_delta_sign</b> [ sigFiltIdx ][ j ]	u(1)
}	
}	
}	
if( alf_luma_clip ) {	
for ( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ ) {	
if ( alf_luma_coeff_flag[ sigFiltIdx ] ) {	
for ( j = 0; j < 12; j++ ) {	
if( filterCoefficients[ sigFiltIdx ][ j ] )	
<b>alf_luma_clip_idx</b> [ sigFiltIdx ][ j ]	u(2)
}	
}	
}	
if ( alf_chroma_idc > 0 ) {	
for( i = 0; i < 2; i++ )	
for( j = 0; j < 6; j++ ) {	
<b>alf_chroma_coeff_abs</b> [ j ]	uek(v)
if( alf_chroma_coeff_abs[ j ] > 0 )	
<b>alf_chroma_coeff_sign</b> [ j ]	u(1)
}	
}	
if ( alf_chroma_idc > 0 && alf_chroma_clip ) {	
for( j = 0; j < 6; j++ ) {	
if( alf_chroma_coeff_abs[ j ] )	
<b>alf_chroma_clip_idx</b> [ j ]	u(2)

}	
}	
}	

Với phương án của giải pháp thay thế (giải pháp 2), mã một ngôi được cắt ngắn cũng có thể được sử dụng để báo hiệu chỉ số thông số cắt bớt.

Với phương án của giải pháp thay thế (giải pháp 3), nếu số lượng của các thông số cắt bớt được thay đổi từ trị số cố định 4 tới số khác lớn hơn 4, mã độ dài cố định ( $v$ ) “trị số  $v$ ” được tăng lên một cách tương ứng. Ví dụ nếu số lượng của các thông số cắt bớt được tăng lên từ 4 đến 5 hoặc 6, sau đó mã độ dài cố định sử dụng 3 bit để báo hiệu thông số cắt bớt.

Với phương án của giải pháp thay thế (giải pháp 4), các hệ số bộ lọc ALF được báo hiệu nhờ sử dụng mã độ dài cố định thay vì mã golomb theo cấp số nhân thứ tự  $K$ .

Fig.10 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ nhất của sáng chế. Phương pháp bao gồm các bước thu nhận dòng bit trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, 1001, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số ALF; phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định, 1002; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại, 1003.

Fig.11 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ hai của sáng chế. Theo khía cạnh thứ hai, có đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm bước

thu nhận dòng bit, ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, 1101, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số ALF; phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp, 1102; áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại, 1103.

Fig.12 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ ba của sáng chế. Theo khía cạnh thứ ba, có đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước: xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số của bộ lọc vòng lặp thích ứng (ALF), 1201; tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định, 1202.

Fig.13 là sơ đồ khối minh họa phương pháp theo khía cạnh thứ tư của sáng chế. Theo khía cạnh thứ tư, có đề xuất phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước: xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số bộ lọc ALF, 1301; tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại, 1302.

Fig.14 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ năm của sáng chế. Theo khía cạnh thứ năm, sáng chế đề xuất bộ giải mã 1400 bao gồm hệ mạch xử lý 1401 để thực hiện phương pháp theo khía cạnh thứ nhất hoặc thứ hai hoặc cách thực hiện bất kỳ của chúng.

Fig.15 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ sáu của sáng chế. Theo khía cạnh thứ sáu, sáng chế đề xuất bộ mã hóa 1500 bao gồm hệ mạch xử lý 1501 để thực hiện phương pháp theo khía cạnh thứ ba hoặc thứ tư hoặc cách thực hiện bất kỳ của chúng.

Fig.16 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ chín của sáng chế. Theo khía cạnh thứ chín, sáng chế đề xuất bộ giải mã 1600, bao gồm: một hoặc nhiều bộ xử lý 1601; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời 1602 được ghép nối với các bộ xử lý 1601 và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý 1601, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý 1601, tạo cấu hình bộ giải mã 1600 để thực hiện phương pháp theo khía cạnh thứ nhất hoặc thứ hai hoặc cách thực hiện bất kỳ của chúng.

Fig.17 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười của sáng chế. Theo khía cạnh thứ mười, sáng chế đề xuất bộ mã hóa 1700, bao gồm: một hoặc nhiều bộ xử lý 1701; và phương tiện lưu trữ đọc được bởi máy tính không tạm thời 1702 được ghép nối với các bộ xử lý 1701 và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý 1701, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý 1701, tạo cấu hình bộ mã hóa 1700 để thực hiện phương pháp theo khía cạnh thứ ba hoặc thứ tư hoặc cách thực hiện bất kỳ của chúng.

Fig.18 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ mười một của sáng chế. Theo khía cạnh thứ mười một, sáng chế đề xuất bộ giải mã 1800, bao gồm: bộ phận giải mã entropi 1801 (có thể là bộ phận giải mã entropi 304), được tạo cấu hình để thu nhận dòng bit 1811 trong đó ít nhất một bit ở dòng bit 1811 biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số của bộ lọc vòng lặp thích ứng (ALF); bộ phận giải mã entropi 1801, còn được tạo cấu hình để phân tách dòng bit 1811 để thu nhận trị số 1812 của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định; và bộ phận lọc 1803 (có thể là bộ lọc vòng lặp 320), được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số 1812 của thành phần cú pháp cho khối hiện tại.

Fig.19 là sơ đồ khối minh họa bộ giải mã theo khía cạnh thứ mười hai của sáng chế. Theo khía cạnh thứ mười hai, sáng chế đề xuất bộ giải mã 1900, bao gồm: bộ phận giải mã entropi 1901 (có thể là bộ phận giải mã entropi 304), được tạo cấu hình để thu nhận dòng bit 1911, ít nhất một bit ở dòng bit 1911 biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF hoặc thông số hệ số ALF; bộ phận giải mã entropi 1801, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số 1912 của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp; và bộ phận lọc 1903 (có thể là bộ lọc vòng lặp 320), được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số 1912 của thành phần cú pháp cho khối hiện tại.

Fig.20 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười ba của sáng chế. Theo khía cạnh thứ mười ba, sáng chế đề xuất bộ mã hóa 2000, bao gồm: bộ phận xác định 2001 (có thể là bộ lọc vòng lặp 220), được tạo cấu hình để xác định trị số 2012 của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt để sử dụng trước khi nhân với hệ số của bộ lọc vòng lặp thích ứng (ALF); bộ phận mã hóa entropi 2002 (có thể là bộ phận mã hóa entropi 270), được tạo cấu hình để tạo ra dòng bit 2011 dựa vào trị số 2012 của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit 2011 biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

Fig.21 là sơ đồ khối minh họa bộ mã hóa theo khía cạnh thứ mười bốn của sáng chế. Theo khía cạnh thứ mười bốn, sáng chế đề xuất bộ mã hóa 2100, bao gồm: bộ phận xác định 2101 (có thể là bộ lọc vòng lặp 220), được tạo cấu hình để xác định trị số 2112 của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF hoặc thông số hệ số bộ lọc ALF; bộ phận mã hóa entropi 2102 (có thể là bộ phận mã hóa entropi 270), được tạo cấu hình để tạo ra dòng bit 2111 dựa vào trị số 2112 của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit 2111 biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại.

Sáng chế đề xuất các phương án khác sau đây:

Phương án 1. Phương pháp tạo mã được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, ít nhất một bit ở dòng bit tương ứng với thành phần cú pháp cho khối hiện tại (hoặc tập hợp của các khối, một khối trong tập hợp của các khối là khối hiện tại);

phân tách dòng bit, để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận chỉ đề cập đến ít nhất một bit;

lọc khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

Phương án 2. Phương pháp theo phương án 1, trong đó trị số của thành phần cú pháp được tạo mã theo mã độ dài cố định (mã độ dài cố định nghĩa là tất cả các trị số khả thi của thành phần cú pháp được báo hiệu nhờ sử dụng số các bit như nhau).

Phương án 3. Phương pháp theo phương án 1, trong đó trị số của thành phần cú pháp được tạo mã theo mã một ngôi được cắt ngắn (mã một ngôi được cắt ngắn nghĩa là trị số xảy ra thường xuyên nhất của thành phần cú pháp đã cho được báo hiệu nhờ sử dụng số lượng các bit ít nhất, và trị số xảy ra ít nhất của thành phần cú pháp được báo hiệu nhờ sử dụng số lượng các bit nhiều nhất).

Phương án 4. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 3, trong đó thành phần cú pháp là thông số chỉ số cắt bớt bộ lọc vòng lặp thích ứng.

Phương án 5. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 3, trong đó thành phần cú pháp là thông số hệ số bộ lọc vòng lặp thích ứng.

Phương án 6. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 5, trong đó trị số của thành phần cú pháp được sử dụng để xác định hệ số bộ lọc, hệ số bộ lọc được sử dụng trong quy trình lọc.

Phương án 7. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 5, trong đó trị số của thành phần cú pháp được sử dụng để xác định khoảng cắt bớt, khoảng cắt bớt được sử dụng trong quy trình lọc (khoảng cắt bớt được sử dụng để giới hạn lượng điều chỉnh được cho phép trên mẫu đã cho bởi mẫu lân cận của nó).

Phương án 8. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 7.

Phương án 9. Sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 7.

Phương án 10. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 7.

Phần sau đây là giải thích của các ứng dụng của phương pháp mã hóa cũng như phương pháp giải mã như được thể hiện theo các phương án nêu trên, và hệ thống sử dụng chúng.

Fig.22 là sơ đồ khối thể hiện hệ thống cung cấp nội dung 3100 để thực hiện dịch vụ phân phối nội dung. Hệ thống cung cấp nội dung 3100 này bao gồm thiết bị chụp 3102,

thiết bị đầu cuối 3106, và một cách tùy chọn bao gồm màn hình 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 qua liên kết truyền thông 3104. Liên kết truyền thông có thể bao gồm kênh truyền thông 13 nêu trên. Liên kết truyền thông 3104 bao gồm nhưng không giới hạn ở WIFI, Ethernet, cáp, (3G/4G/5G) không dây, USB, hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Thiết bị chụp 3102 tạo ra dữ liệu, và có thể mã hóa dữ liệu bởi phương pháp mã hóa như được thể hiện theo các phương án nêu trên. Theo cách khác, thiết bị chụp 3102 có thể phân phối dữ liệu tới máy chủ tạo dòng (không được thể hiện trên các hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa tới thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm nhưng không giới hạn ở camera, điện thoại thông minh hoặc Pad, máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, thiết bị được lắp trên phương tiện giao thông, hoặc sự kết hợp của bất kỳ trong số chúng, hoặc tương tự. Ví dụ, thiết bị chụp 3102 có thể bao gồm thiết bị nguồn 12 như được nêu trên. Khi dữ liệu bao gồm video, bộ mã hóa video 20 được bao gồm trong thiết bị chụp 3102 thực tế có thể thực hiện việc xử lý mã hóa video. Khi dữ liệu bao gồm audio (nghĩa là, giọng nói), bộ mã hóa audio được bao gồm trong thiết bị chụp 3102 thực tế có thể thực hiện việc xử lý mã hóa audio. Đối với một số kịch bản thực tế, thiết bị chụp 3102 phân phối dữ liệu video và audio được mã hóa bằng cách đa hợp chúng với nhau. Đối với các kịch bản thực tế khác, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được đa hợp. Thiết bị chụp 3102 phân phối dữ liệu audio được mã hóa và dữ liệu video được mã hóa tới thiết bị đầu cuối 3106 một cách tách biệt.

Trong hệ thống cung cấp nội dung 3100, thiết bị đầu cuối 310 thu và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng thu và khôi phục dữ liệu, chẳng hạn như điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, đầu ghi video mạng (network video recorder, viết tắt là NVR)/ đầu ghi video kỹ thuật số (digital video recorder, viết tắt là DVR) 3112, TV 3114, thiết bị giải mã tín hiệu (set top box, viết tắt là STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, thiết bị hỗ trợ kỹ thuật số cá nhân (personal digital assistant, viết tắt là PDA) 3122, thiết bị được lắp trên phương tiện giao thông 3124, hoặc sự kết hợp của bất kỳ trong số chúng, hoặc tương tự có khả năng giải mã dữ liệu được mã hóa nêu trên. Ví dụ, thiết bị đầu cuối 3106 có thể bao gồm thiết bị đích 14 như được nêu trên. Khi dữ

liệu được mã hóa bao gồm video, bộ giải mã video 30 được bao gồm trong thiết bị đầu cuối được quyền ưu tiên thực hiện việc giải mã video. Khi dữ liệu được mã hóa bao gồm audio, bộ giải mã audio được bao gồm trong thiết bị đầu cuối được quyền ưu tiên thực hiện việc xử lý giải mã audio.

Đối với thiết bị đầu cuối với màn hình của nó, ví dụ, điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay 3110, đầu ghi video mạng (NVR)/ đầu ghi video kỹ thuật số (DVR) 3112, TV 3114, thiết bị hỗ trợ kỹ thuật số cá nhân (PDA) 3122, hoặc thiết bị được lắp trên phương tiện giao thông 3124, thiết bị đầu cuối có thể nạp dữ liệu được giải mã vào màn hình của nó. Đối với thiết bị đầu cuối được trang bị không có màn hình, ví dụ, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, màn hình bên ngoài 3126 được tiếp xúc ở đó để thu và thể hiện dữ liệu được giải mã.

Khi mỗi thiết bị trong hệ thống này thực hiện việc mã hóa hoặc giải mã, thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh, như được thể hiện theo các phương án nêu trên, có thể được sử dụng.

Fig.23 là sơ đồ thể hiện cấu trúc của ví dụ về thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 thu dòng từ thiết bị chụp 3102, bộ phận triển khai giao thức 3202 phân tích giao thức truyền của dòng. Giao thức bao gồm nhưng không giới hạn ở giao thức tạo dòng thời gian thực (Real Time Streaming Protocol, viết tắt là RTSP), giao thức chuyển đổi siêu văn bản (Hyper Text Transfer Protocol, viết tắt là HTTP), giao thức tạo dòng trực tuyến HTTP (HTTP Live streaming, viết tắt là HLS), MPEG-DASH, giao thức vận chuyển thời gian thực (Real-time Transport protocol, viết tắt là RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, viết tắt là RTMP), hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Sau khi bộ phận triển khai giao thức 3202 xử lý dòng, tệp dòng được tạo ra. Tệp được đưa ra tới bộ phận giải đa hợp 3204. Bộ phận giải đa hợp 3204 có thể tách riêng dữ liệu được đa hợp thành dữ liệu audio được mã hóa và dữ liệu video được mã hóa. Như được nêu trên, đối với một số kịch bản thực tế, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được đa hợp. Trong trường hợp này, dữ liệu được mã hóa được truyền tới bộ giải mã video 3206 và bộ giải mã audio 3208 mà không thông qua bộ phận giải đa hợp 3204.



Qua việc xử lý giải đa hợp, dòng sơ cấp (elementary stream, viết tắt là ES) video, ES audio, và một cách tùy chọn phụ đề được tạo ra. Bộ giải mã video 3206, mà bao gồm bộ giải mã video 30 như được giải thích theo các phương án nêu trên, giải mã ES video bởi phương pháp giải mã như được thể hiện theo các phương án nêu trên để tạo ra khung video, và nạp dữ liệu này vào bộ phận đồng bộ 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo ra khung audio, và nạp dữ liệu này vào bộ phận đồng bộ 3212. Theo cách khác, khung video có thể lưu trữ ở bộ đệm (không được thể hiện trên Fig.23) trước khi nạp nó vào bộ phận đồng bộ 3212. Tương tự như vậy, khung audio có thể lưu trữ ở bộ đệm (không được thể hiện trên Fig.23) trước khi nạp nó vào bộ phận đồng bộ 3212.

bộ phận đồng bộ 3212 đồng bộ hóa khung video và khung audio, và cấp video/audio tới màn hình video/audio 3214. Ví dụ, bộ phận đồng bộ 3212 đồng bộ hóa sự biểu diễn của thông tin video và audio. Thông tin có thể tạo mã ở cú pháp nhờ sử dụng các dấu thời gian liên quan đến sự biểu diễn của dữ liệu trực quan và audio được tạo mã và các dấu thời gian liên quan đến việc phân phối của chính dòng dữ liệu.

Nếu phụ đề được bao gồm trong dòng, bộ giải mã phụ đề 3210 giải mã phụ đề, và đồng bộ hóa nó với khung video và khung audio, và cấp video/audio/phụ đề tới màn hình video/audio/phụ đề 3216.

Sáng chế không giới hạn ở hệ thống nêu trên, và hoặc thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh theo các phương án nêu trên có thể được kết hợp vào hệ thống khác, ví dụ, hệ thống ô tô.

Mặc dù các phương án của sáng chế đã được mô tả chủ yếu dựa vào việc tạo mã video, cần lưu ý rằng các phương án của hệ thống tạo mã 10, bộ mã hóa 20 và bộ giải mã 30 (và một cách tương ứng hệ thống 10) và các phương án còn lại được mô tả ở đây cũng có thể được tạo cấu hình để xử lý hoặc tạo mã ảnh tĩnh, nghĩa là, việc xử lý hoặc tạo mã của ảnh riêng độc lập với ảnh trước đó hoặc liên tiếp bất kỳ như trong việc tạo mã video. Nói chung chỉ các bộ phận liên dự đoán 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp xử lý ảnh tạo mã giới hạn ở ảnh đơn 17. Tất cả các tính năng khác (cũng được gọi là các công cụ hoặc các công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng như nhau để xử lý ảnh tĩnh, ví dụ, tính toán phần dư 204/304, biến đổi 206, lượng tử hóa 208, lượng tử hóa ngược

210/310, biến đổi (ngược) 212/312, phân chia 262/362, nội dự đoán 254/354, và/hoặc lọc vòng lặp 220, 320, và tạo mã entrôpi 270 và giải mã entrôpi 304.

Các phương án, ví dụ, của bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, ví dụ, dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được thực hiện ở phần cứng, phần mềm, phần sụn, hoặc sự kết hợp bất kỳ của chúng. Nếu được thực hiện ở phần mềm, các chức năng có thể được lưu trữ trên phương tiện có thể đọc được bởi máy tính hoặc được truyền qua các phương tiện truyền thông như một hoặc nhiều lệnh hoặc mã và được thực hiện bởi bộ phận xử lý dựa trên phần cứng. Các phương tiện đọc được bởi máy tính có thể bao gồm các phương tiện lưu trữ đọc được bởi máy tính, mà tương ứng với phương tiện hữu hình chẳng hạn như các phương tiện lưu trữ dữ liệu, hoặc các phương tiện truyền thông bao gồm phương tiện bất kỳ mà tạo điều kiện thuận lợi cho việc chuyển đổi của chương trình máy tính từ một địa điểm tới địa điểm khác, ví dụ, theo giao thức truyền thông. Theo cách thức này, các phương tiện đọc được bởi máy tính nói chung có thể tương ứng với (1) các phương tiện lưu trữ đọc được bởi máy tính hữu hình mà không tạm thời hoặc (2) phương tiện truyền thông chẳng hạn như tín hiệu hoặc sóng mang. Các phương tiện lưu trữ dữ liệu có thể là các phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy xuất các lệnh, mã và/hoặc các cấu trúc dữ liệu cho việc thực hiện của các kỹ thuật được mô tả theo sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện đọc được bởi máy tính.

Bằng ví dụ, và không giới hạn, các phương tiện lưu trữ đọc được bởi máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc thiết bị lưu trữ đĩa quang, thiết bị lưu trữ đĩa từ khác, hoặc các thiết bị lưu trữ từ, bộ nhớ tia chớp khác, hoặc phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn dưới dạng của các lệnh hoặc các cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Ngoài ra, sự kết hợp bất kỳ được gọi thích hợp là phương tiện đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ trang mạng, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường dây thuê bao kỹ thuật số (digital subscriber line, viết tắt là DSL), hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng, sau đó cáp đồng trục, cáp sợi quang, cáp xoắn đôi, DSL, hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng được

bao gồm theo định nghĩa về phương tiện. Tuy nhiên, cần hiểu rằng các phương tiện lưu trữ đọc được bởi máy tính và các phương tiện lưu trữ dữ liệu không bao gồm các sự kết nối, các sóng mang, các tín hiệu, hoặc các phương tiện nhất thời khác, nhưng thay vào đó được hướng đến các phương tiện lưu trữ hữu hình, không tạm thời. Đĩa (disk) và đĩa (disc), như được sử dụng ở đây, bao gồm đĩa compac (compact disc, viết tắt là CD), đĩa laze, đĩa quang, đĩa đa năng kỹ thuật số (digital versatile disc, viết tắt là DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa thường tái tạo dữ liệu từ, trong khi các đĩa tái tạo dữ liệu quang với các laze. Các sự kết hợp của bộ phận nêu trên cũng cần được bao gồm nằm trong phạm vi của các phương tiện đọc được bởi máy tính.

Các lệnh có thể được thực hiện bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu kỹ thuật số (DSP), bộ vi xử lý đa năng, các mạch tích hợp chuyên dụng (ASIC), các mảng cổng lập trình được dạng trường (FPGA), hoặc hệ mạch logic rời rạc hoặc được tích hợp tương đương khác. Theo đó, thuật ngữ “bộ xử lý,” như được sử dụng ở đây có thể đề cập đến bất kỳ trong số cấu trúc nêu trên hoặc cấu trúc khác bất kỳ thích hợp cho việc thực hiện của các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, tính năng được mô tả ở đây có thể được đưa ra nằm trong các môđun phần cứng và/hoặc phần mềm dành riêng được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp trong bộ mã hóa/giải mã được kết hợp. Ngoài ra, các kỹ thuật có thể được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc các thành phần logic.

Các kỹ thuật theo sáng chế có thể được thực hiện trong nhiều thiết bị hoặc nhiều máy, bao gồm thiết bị cầm tay không dây, mạch tích hợp (integrated circuit, viết tắt là IC) hoặc tập hợp của các IC (ví dụ, bộ chip). Các thành phần, các môđun, hoặc các bộ phận khác nhau được mô tả theo sáng chế nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không cần thiết yêu cầu thực hiện bởi các bộ phận phần cứng khác nhau. Hơn nữa, như được mô tả ở trên, các bộ phận khác nhau có thể được kết hợp trong bộ phận phần cứng bộ mã hóa/giải mã hoặc được cung cấp bởi tập hợp của các bộ phận phần cứng tương tác, bao gồm một hoặc nhiều bộ xử lý như được nêu trên, kết hợp với phần mềm và/hoặc phần sụn thích hợp.

## YÊU CẦU BẢO HỘ

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

thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (adaptive loop filter, viết tắt là ALF);

phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định;

áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

2. Phương pháp theo điểm 1, trong đó mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit.

3. Phương pháp theo điểm 1 hoặc 2, trong đó thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

4. Phương pháp theo điểm 1 hoặc 2, trong đó thành phần cú pháp ở mức lát.

5. Phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số ALF;

phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp;

áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

6. Phương pháp theo điểm 5, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

7. Phương pháp theo điểm 6, trong đó mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit.

8. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 7, trong đó thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

9. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 7, trong đó thành phần cú pháp ở mức lát.

10. Phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 9, trong đó thông số hệ số ALF được sử dụng để xác định hệ số ALF.

11. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 10, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF, và ít nhất một bit biểu diễn thành phần cú pháp là hai bit.

12. Phương pháp theo điểm 11, trong đó chỉ số trị số cắt bớt ALF nhận dạng một trị số cắt bớt trong số bốn trị số cắt bớt.

13. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 12, trong đó trị số của chỉ số trị số cắt bớt ALF được sử dụng để xác định khoảng cắt bớt, khoảng cắt bớt được sử dụng trong quy trình lọc vòng lặp thích ứng.

14. Phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit, trong đó  $n$  bit ở dòng bit biểu diễn thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0;

phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp là sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit;

áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

15. Phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF);

tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

16. Phương pháp theo điểm 15, trong đó mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit của thành phần cú pháp.

17. Phương pháp theo điểm 15 hoặc 16, trong đó thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

18. Phương pháp theo điểm 15 hoặc 16, trong đó thành phần cú pháp ở mức lát.

19. Phương pháp tạo mã, được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số bộ lọc ALF;

tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khối hiện tại.

20. Phương pháp theo điểm 19, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

21. Phương pháp theo điểm 20, trong đó mã độ dài cố định bao gồm sự biểu diễn nhị phân của số nguyên không dấu sử dụng ít nhất một bit của thành phần cú pháp.

22. Phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 21, trong đó thành phần cú pháp được áp dụng cho tập hợp của các khối và khối hiện tại là một khối trong tập hợp của các khối.

23. Phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 21, trong đó thành phần cú pháp ở mức lát.

24. Phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 23, trong đó thông số hệ số ALF được sử dụng để xác định hệ số ALF.

25. Phương pháp theo điểm bất kỳ trong số các điểm từ 15 đến 24, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF, và ít nhất một bit biểu diễn thành phần cú pháp là hai bit.

26. Phương pháp theo điểm 25, trong đó chỉ số trị số cắt bớt ALF nhận dạng một trị số cắt bớt trong số bốn trị số cắt bớt.

27. Phương pháp theo điểm bất kỳ trong số các điểm từ 15 đến 26, trong đó trị số của chỉ số trị số cắt bớt ALF được sử dụng để xác định khoảng cắt bớt, khoảng cắt bớt được sử dụng trong quy trình lọc vòng lặp thích ứng.

28. Phương pháp tạo mã, được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

xác định trị số của thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0;

tạo ra dòng bit bao gồm  $n$  bit, dựa vào trị số của thành phần cú pháp, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp.

29. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

30. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 13.

31. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm 14.

32. Bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 15 đến 18.

33. Bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 27.

34. Bộ mã hóa (20) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm 28.

35. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

36. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 13.

37. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm 14.

38. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 15 đến 18.

39. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp



theo điểm bất kỳ trong số các điểm từ 19 đến 27.

40. Phương tiện đọc được bởi máy tính không tạm thời mang mã chương trình mà, khi được thực hiện bởi thiết bị máy tính, khiến thiết bị máy tính thực hiện phương pháp theo điểm 28.

41. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4.

42. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 5 đến 13.

43. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo điểm 14.

44. Bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi

được thực hiện bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 15 đến 18.

45. Bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 27.

46. Bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bởi máy tính không tạm thời được ghép nối với các bộ xử lý và lưu trữ sự lập trình để thực hiện bởi các bộ xử lý, trong đó sự lập trình, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp theo điểm 28.

47. Bộ giải mã, bao gồm:

bộ phận giải mã entropi, được tạo cấu hình để thu nhận dòng bit trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF);

bộ phận giải mã entropi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định; và

bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

48. Bộ giải mã, bao gồm:

bộ phận giải mã entropi, được tạo cấu hình để thu nhận dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp cho khối hiện tại, trong đó thành phần

cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số ALF;

bộ phận giải mã entropi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp cho khối hiện tại được thu nhận nhờ sử dụng chỉ ít nhất một bit của thành phần cú pháp; và

bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

49. Bộ giải mã, bao gồm:

bộ phận giải mã entropi, được tạo cấu hình để thu nhận dòng bit, trong đó  $n$  bit ở dòng bit biểu diễn thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0;

bộ phận giải mã entropi, còn được tạo cấu hình để phân tách dòng bit để thu nhận trị số của thành phần cú pháp cho khối hiện tại, trong đó trị số của thành phần cú pháp là sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit;

bộ phận lọc, được tạo cấu hình để áp dụng việc lọc vòng lặp thích ứng trên khối hiện tại, dựa vào trị số của thành phần cú pháp cho khối hiện tại.

50. Bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp cho khối hiện tại, trong đó thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF);

bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định.

51. Bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp cho

khởi hiện tại, trong đó thành phần cú pháp là chỉ số trị số cắt bớt ALF hoặc thông số hệ số bộ lọc của bộ lọc vòng lặp thích ứng (ALF);

bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit dựa vào trị số của thành phần cú pháp, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó ít nhất một bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp cho khởi hiện tại.

52. Bộ mã hóa, bao gồm:

bộ phận xác định, được tạo cấu hình để xác định trị số của thành phần cú pháp mức lát định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0;

bộ phận mã hóa entropi, được tạo cấu hình để tạo ra dòng bit bao gồm  $n$  bit, dựa vào trị số của thành phần cú pháp, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp.

53. Phương tiện lưu trữ không tạm thời bao gồm dòng bit bao gồm  $n$  bit, trong đó sự biểu diễn nhị phân của số nguyên không dấu sử dụng  $n$  bit là trị số của thành phần cú pháp, và thành phần cú pháp định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF), trong đó  $n$  là số nguyên lớn hơn hoặc bằng 0.

54. Phương tiện lưu trữ không tạm thời bao gồm dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp được tạo mã nhờ sử dụng mã độ dài cố định và định rõ chỉ số cắt bớt của trị số cắt bớt cho bộ lọc vòng lặp thích ứng (ALF).

55. Phương tiện lưu trữ không tạm thời theo điểm 54, trong đó chính thành phần cú pháp định rõ trị số của thành phần cú pháp.

56. Phương tiện lưu trữ không tạm thời bao gồm dòng bit, trong đó ít nhất một bit ở dòng bit biểu diễn thành phần cú pháp, trong đó thành phần cú pháp là chỉ số trị số cắt bớt bộ lọc vòng lặp thích ứng (ALF) hoặc thông số hệ số bộ lọc ALF, và ít nhất một

bit của thành phần cú pháp được thu nhận nhờ sử dụng chỉ trị số của thành phần cú pháp.

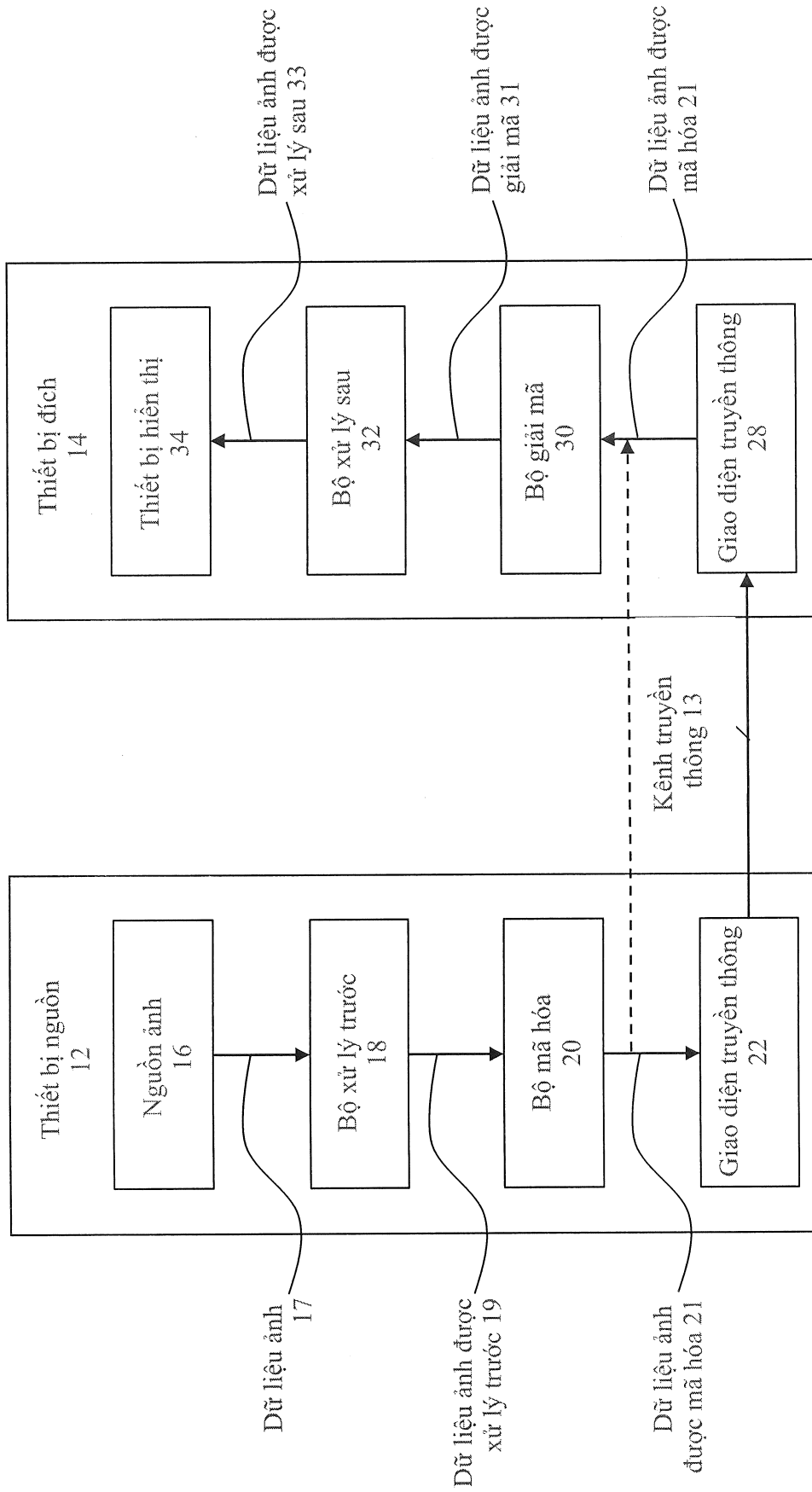


FIG. 1A

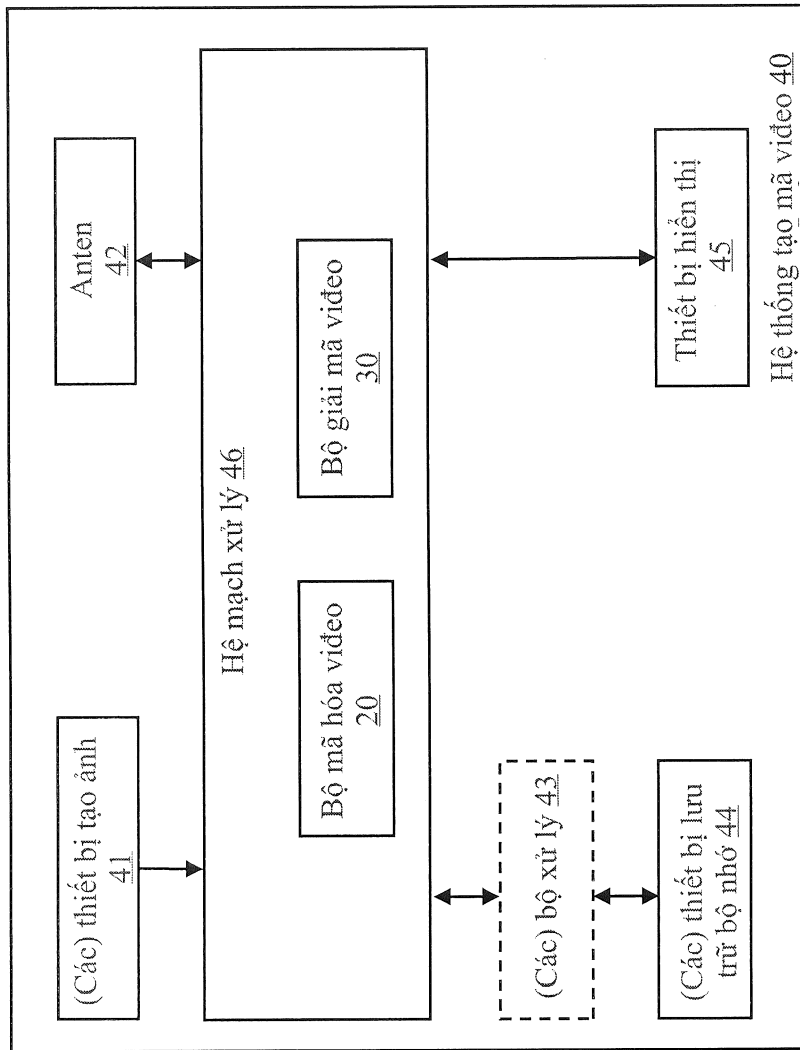


FIG. 1B

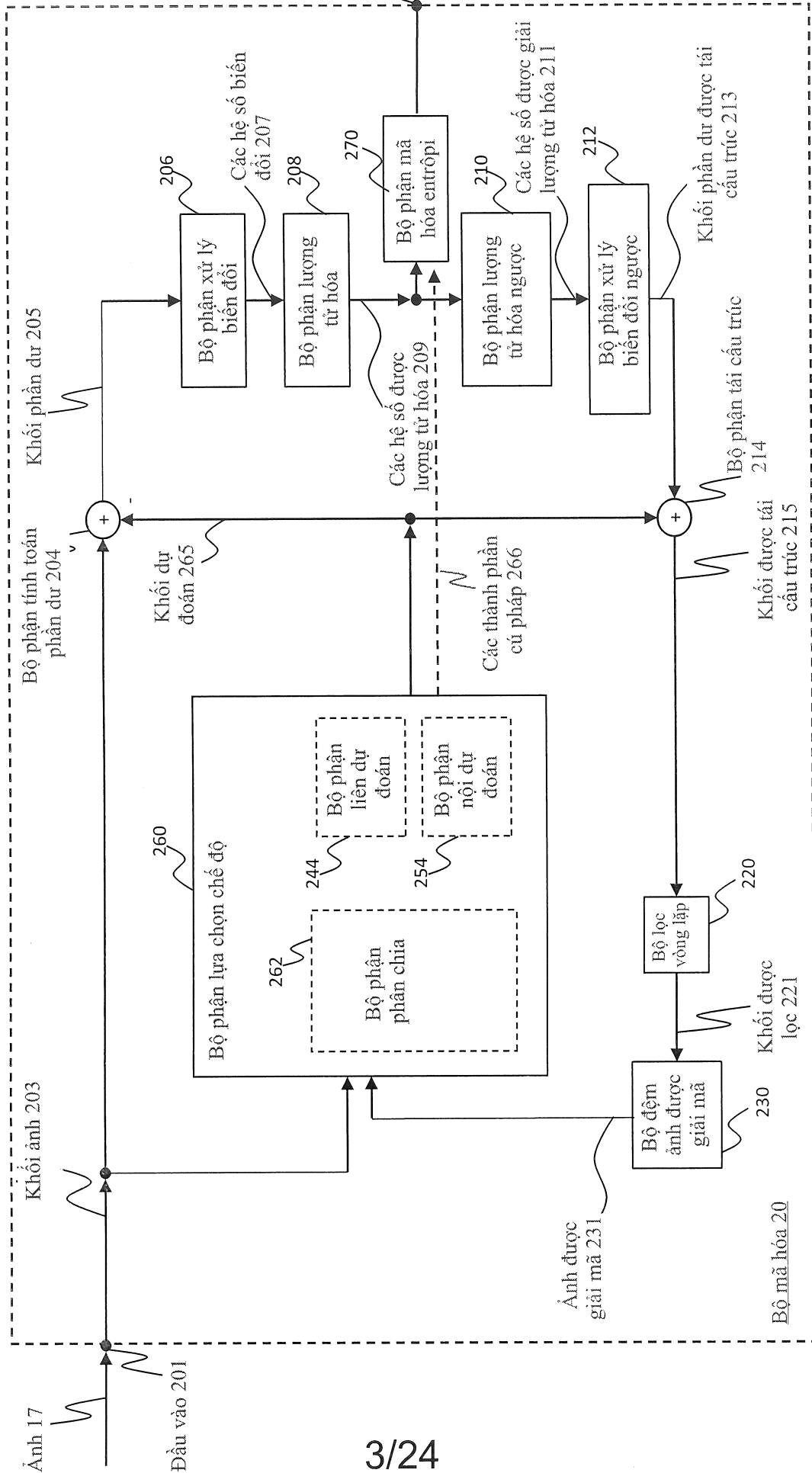


FIG. 2



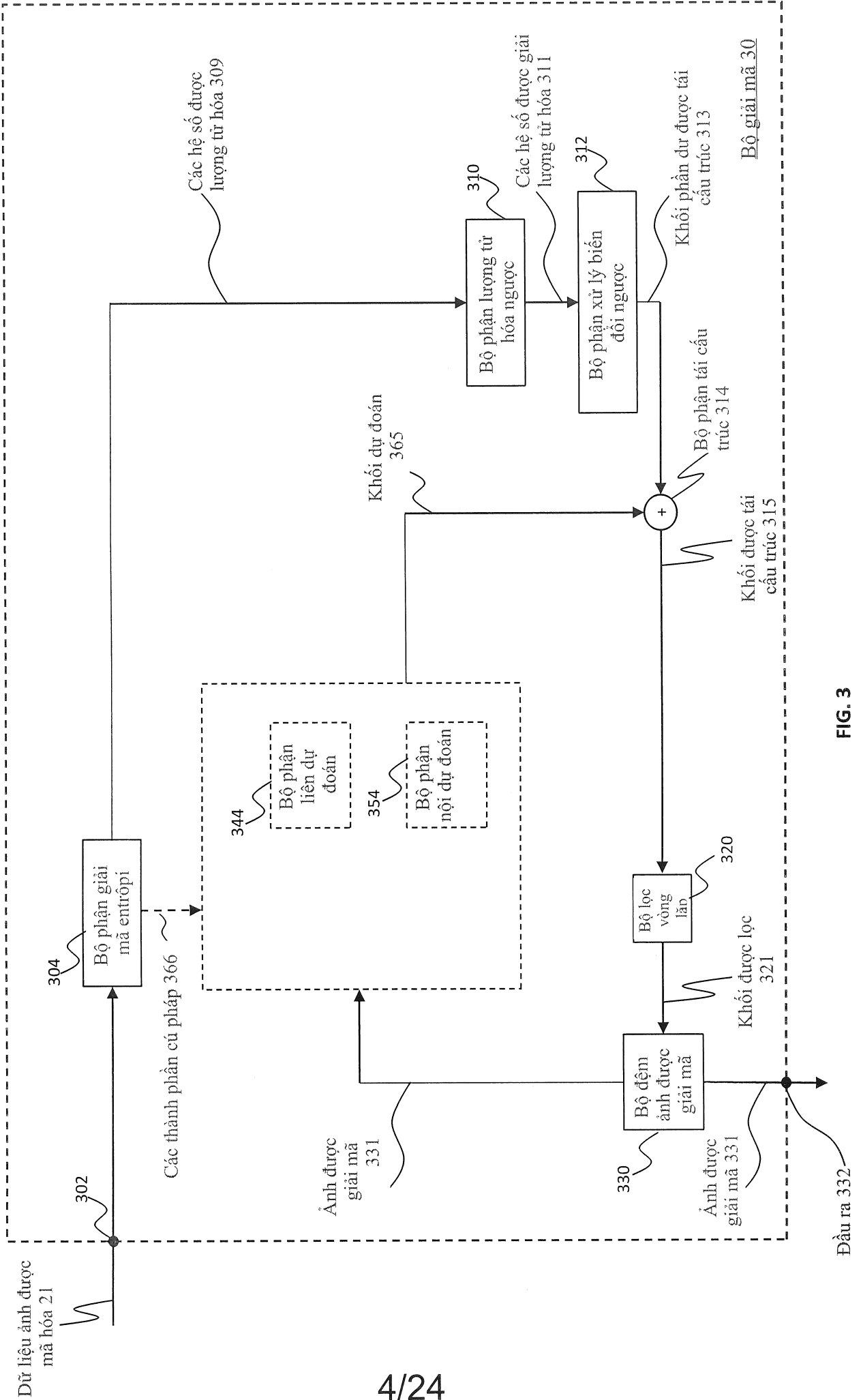


FIG. 3

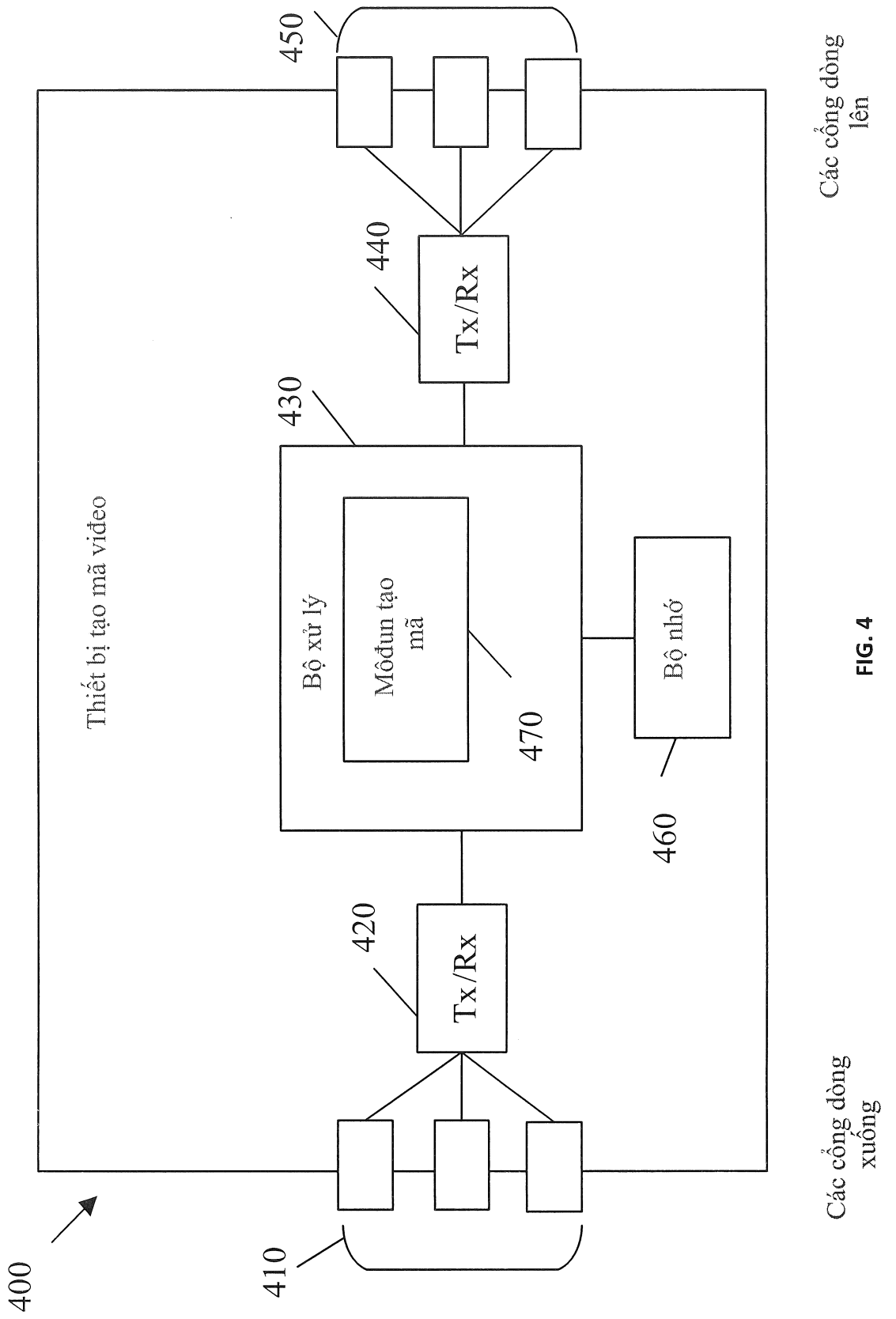


FIG. 4

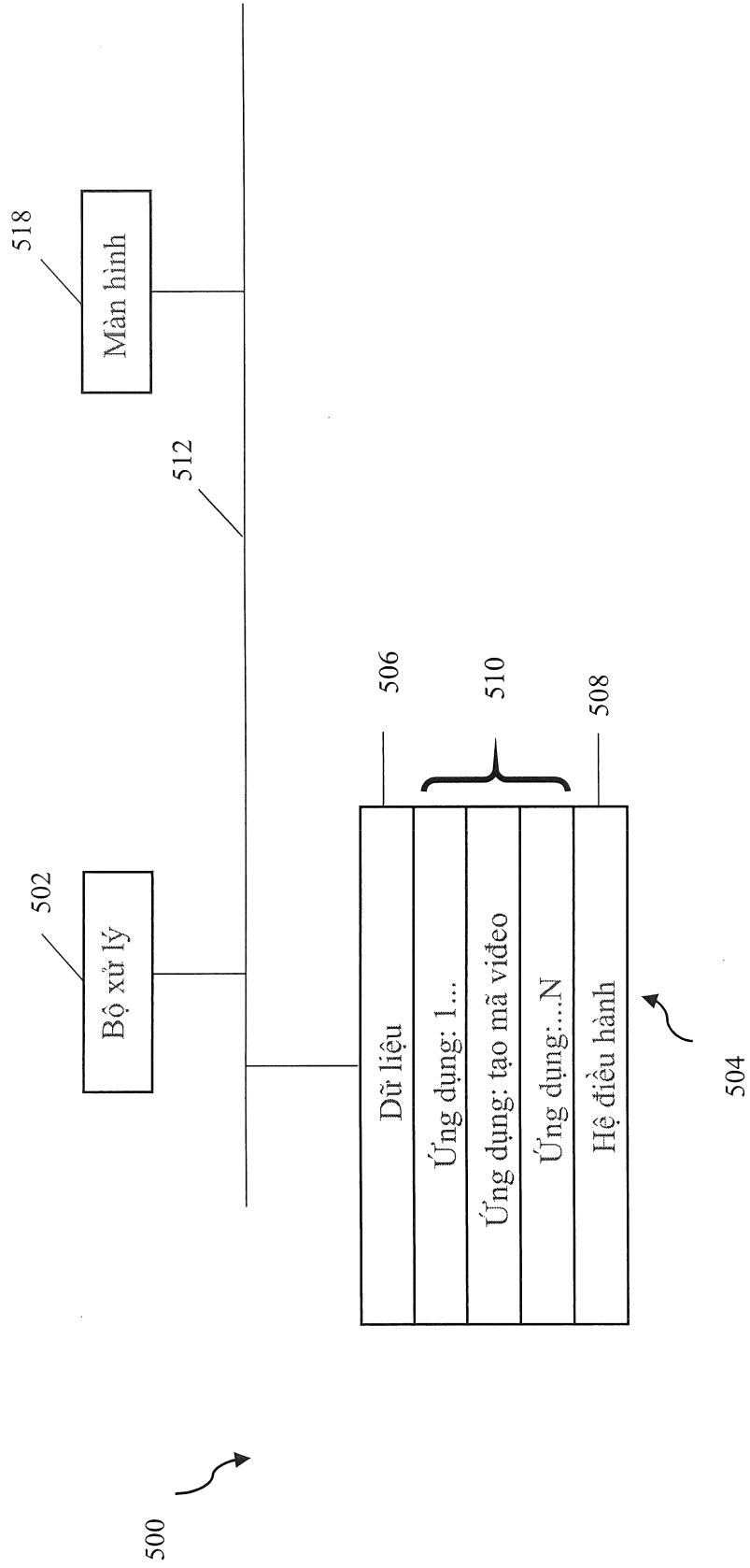


FIG. 5

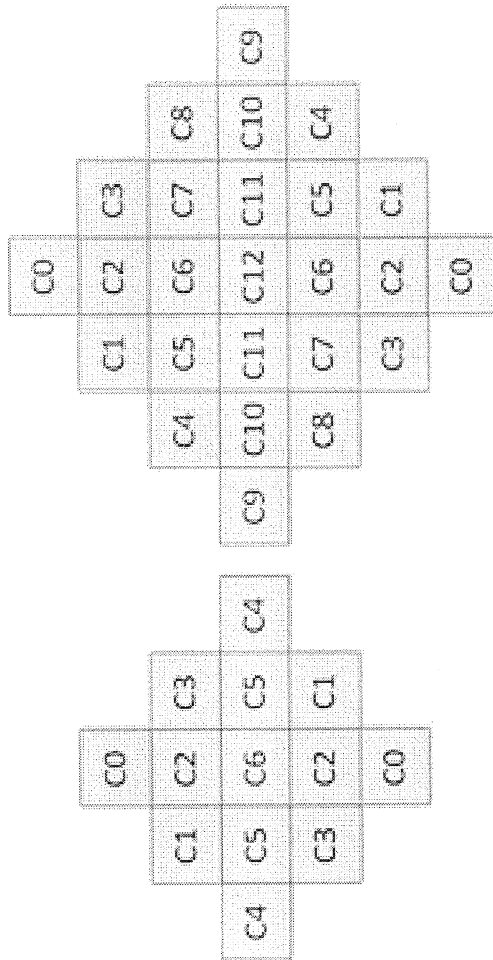


FIG. 6

V		V		V		V		V		V
	V		V		V		V		V	
V		V		V		V		V		V
	V		V		V		V		V	
V		V		V		V		V		V
	V		V		V		V		V	
V		V		V		V		V		V
	V		V		V		V		V	
V		V		V		V		V		V
	V		V		V		V		V	

(a) Các vị trí được lấy mẫu phụ cho gradient theo chiều dọc

H		H		H		H		H		H
	H		H		H		H		H	
H		H		H		H		H		H
	H		H		H		H		H	
H		H		H		H		H		H
	H		H		H		H		H	
H		H		H		H		H		H
	H		H		H		H		H	
H		H		H		H		H		H
	H		H		H		H		H	

(b) Các vị trí được lấy mẫu phụ cho gradient theo chiều ngang

D1		D1		D1		D1		D1		D1
	D1		D1		D1		D1		D1	
D1		D1		D1		D1		D1		D1
	D1		D1		D1		D1		D1	
D1		D1		D1		D1		D1		D1
	D1		D1		D1		D1		D1	
D1		D1		D1		D1		D1		D1
	D1		D1		D1		D1		D1	
D1		D1		D1		D1		D1		D1
	D1		D1		D1		D1		D1	

(c) Các vị trí được lấy mẫu phụ cho gradient theo đường chéo

D2		D2		D2		D2		D2		D2
	D2		D2		D2		D2		D2	
D2		D2		D2		D2		D2		D2
	D2		D2		D2		D2		D2	
D2		D2		D2		D2		D2		D2
	D2		D2		D2		D2		D2	
D2		D2		D2		D2		D2		D2
	D2		D2		D2		D2		D2	
D2		D2		D2		D2		D2		D2
	D2		D2		D2		D2		D2	

(d) Các vị trí được lấy mẫu phụ cho gradient theo đường chéo

**FIG. 7**

if( alf_luma_clip ) {	
<b>alf_luma_clip_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 3; i++)	
<b>alf_luma_clip_eg_order_increase_flag[ i ]</b>	u(1)
for ( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++) {	
if ( alf_luma_coeff_flag[ sigFiltIdx ] ) {	
for ( j = 0; j < 12; j++) {	
if( filterCoefficients[ sigFiltIdx ][ j ] )	
<b>alf_luma_clip_idx[ sigFiltIdx ][ j ]</b>	uek(v)
}	
}	
}	
}	
if ( alf_chroma_idc > 0 && alf_chroma_clip ) {	
<b>alf_chroma_clip_min_eg_order_minus1</b>	ue(v)
for( i = 0; i < 2; i++)	
<b>alf_chroma_clip_eg_order_increase_flag[ i ]</b>	u(1)
for( j = 0; j < 6; j++) {	
if( alf_chroma_coeff_abs[ j ] )	
<b>alf_chroma_clip_idx[ j ]</b>	uek(v)
}	
}	

FIG. 8

if( alf_luma_clip ) {	
for ( sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++ ) {	
if( alf_luma_coeff_flag[ sigFiltIdx ] ) {	
for( j = 0; j < 12; j++ ) {	
if( filterCoefficients[ sigFiltIdx ][ j ] )	
<b>alf_luma_clip_idx</b> [ sigFiltIdx ][ j ]	u(2)
}	
}	
}	
}	
}	
if( alf_chroma_idc > 0 && alf_chroma_clip ) {	
for( j = 0; j < 6; j++ ) {	
if( alf_chroma_coeff_abs[ j ] )	
<b>alf_chroma_clip_idx</b> [ j ]	u(2)
}	
}	

FIG. 9

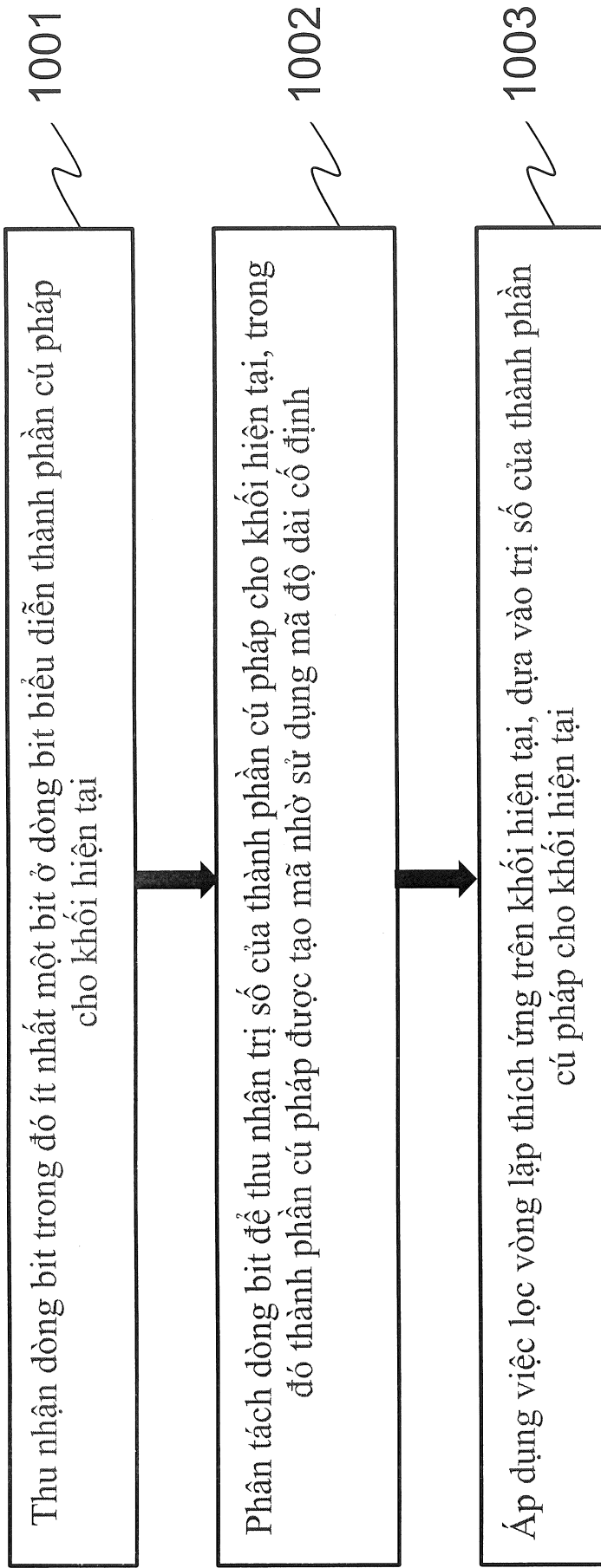
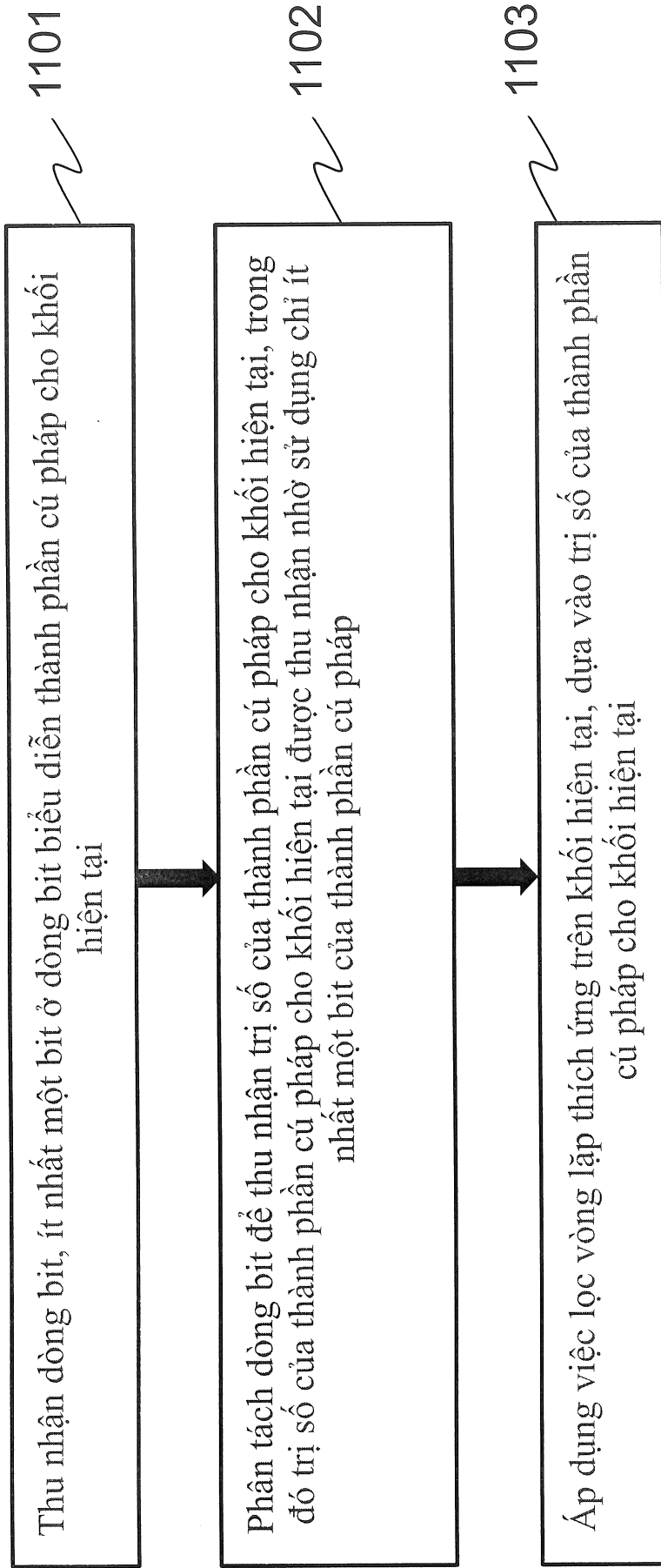


FIG. 10



**FIG. 11**

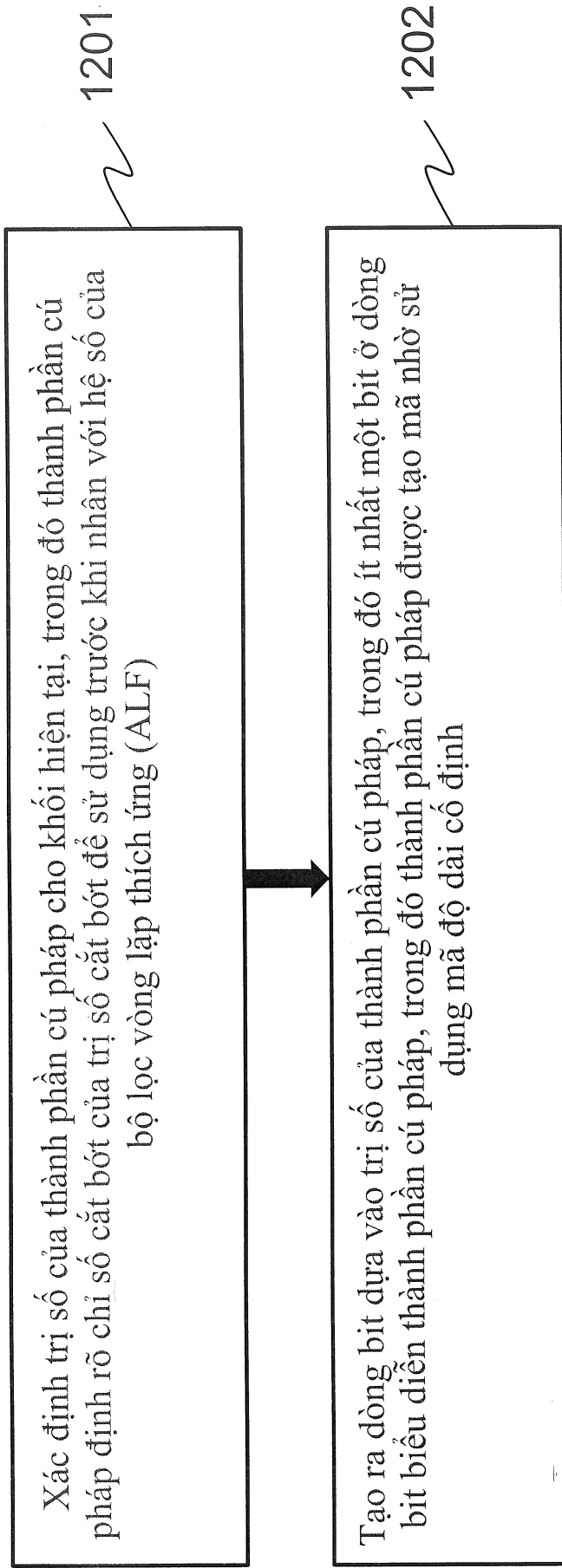


FIG. 12

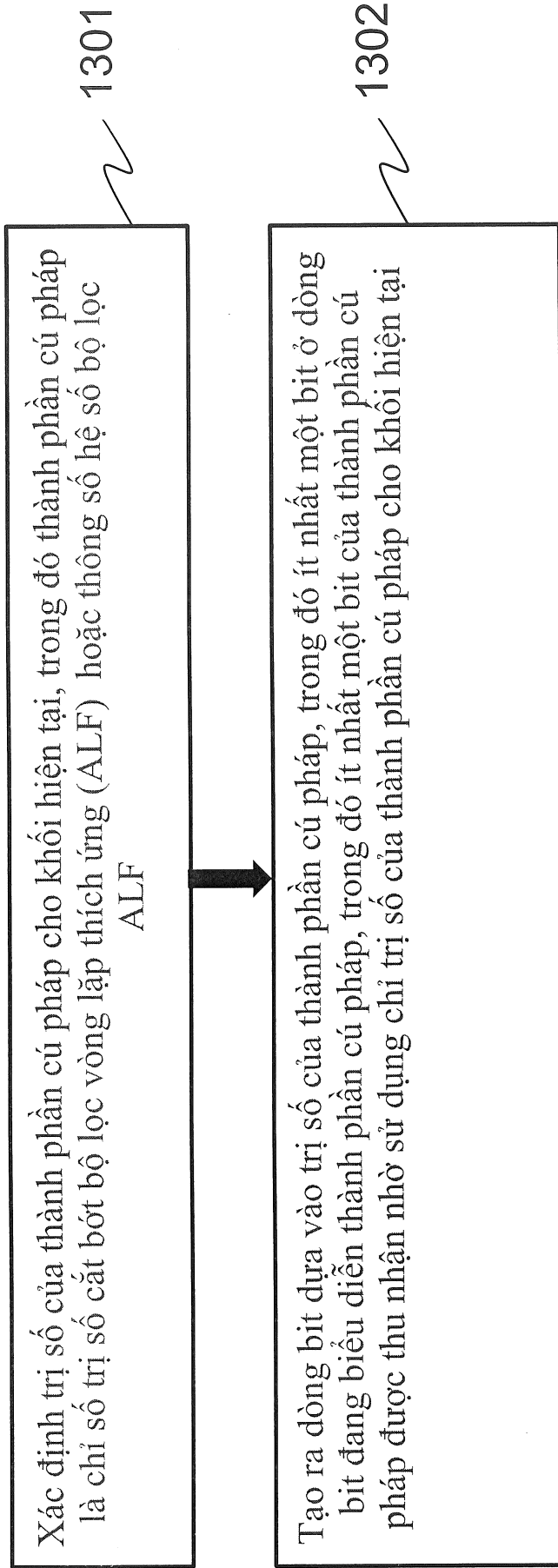


FIG. 13

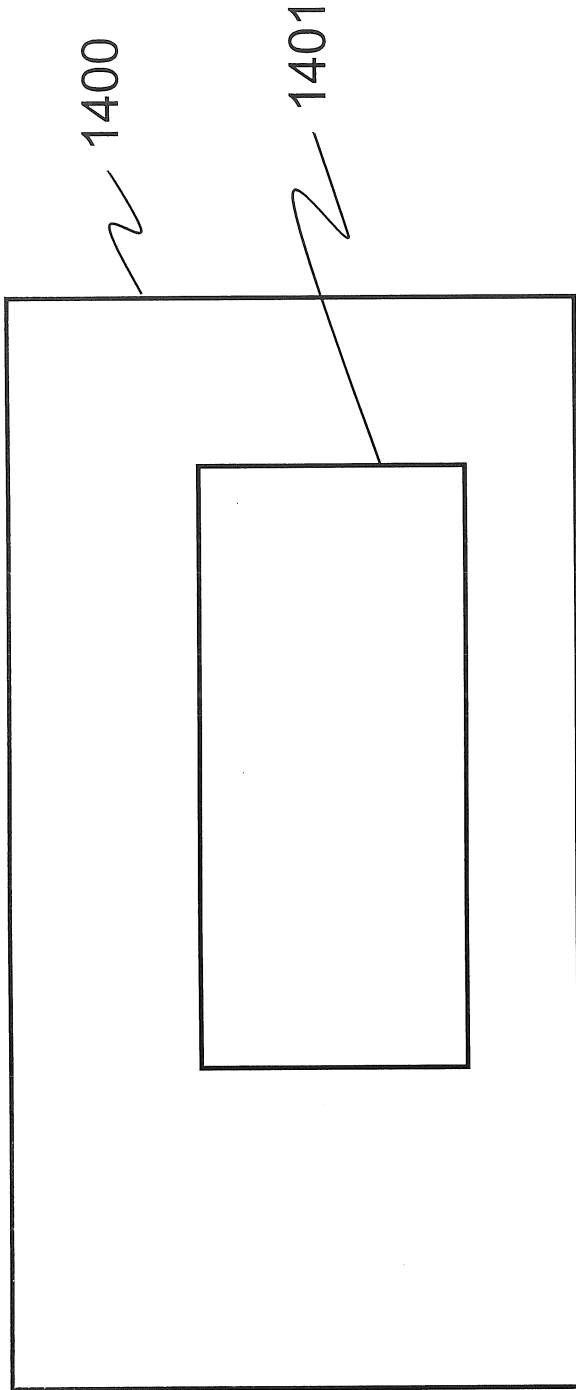


FIG. 14

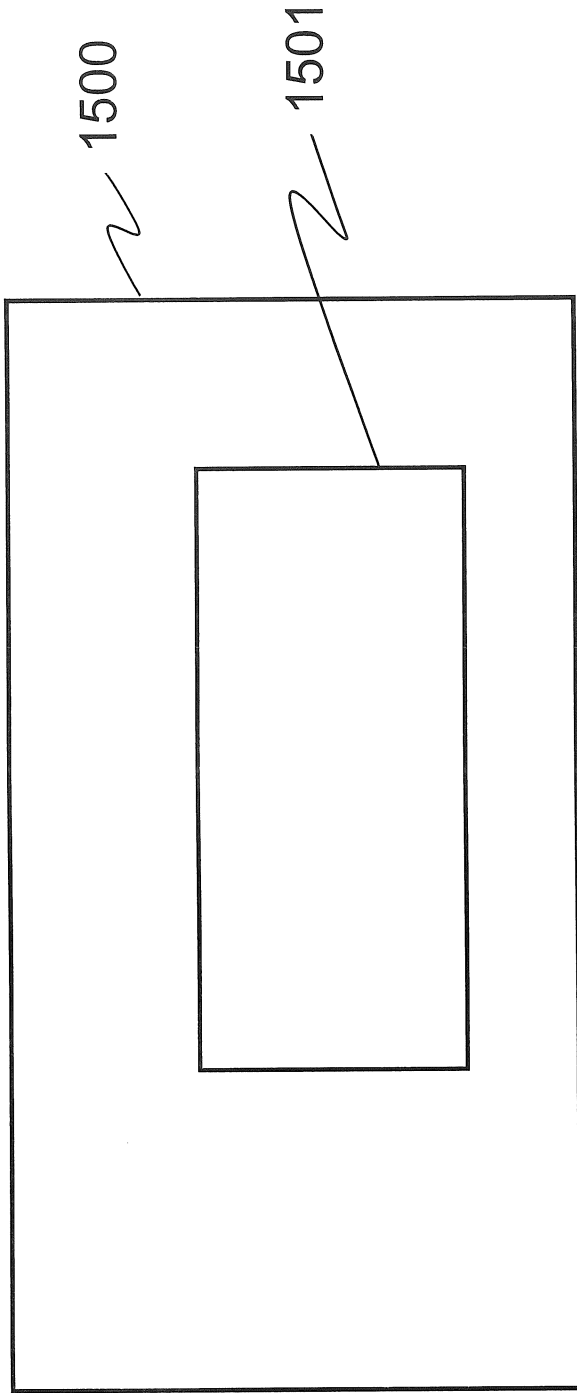


FIG. 15

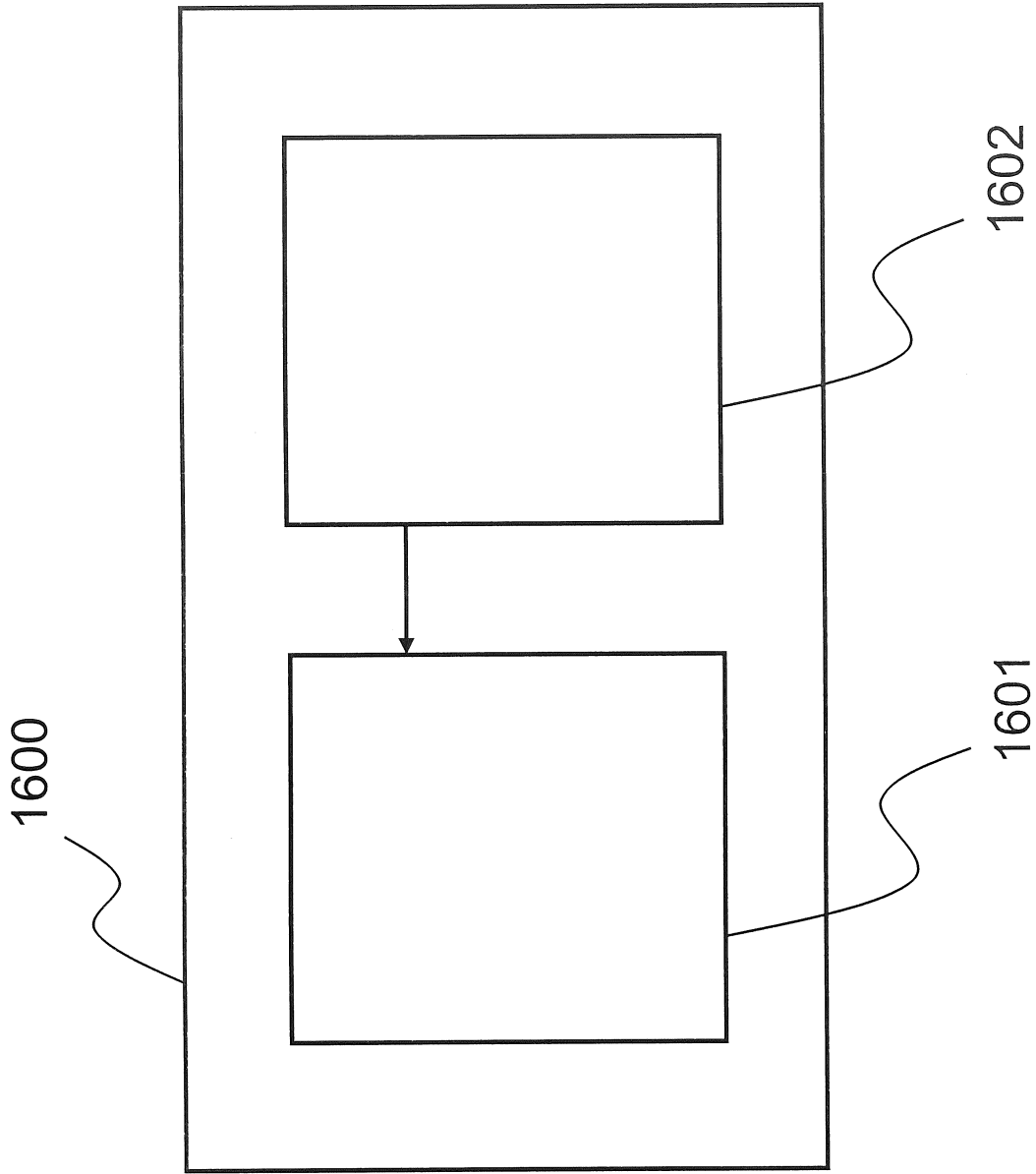


FIG. 16

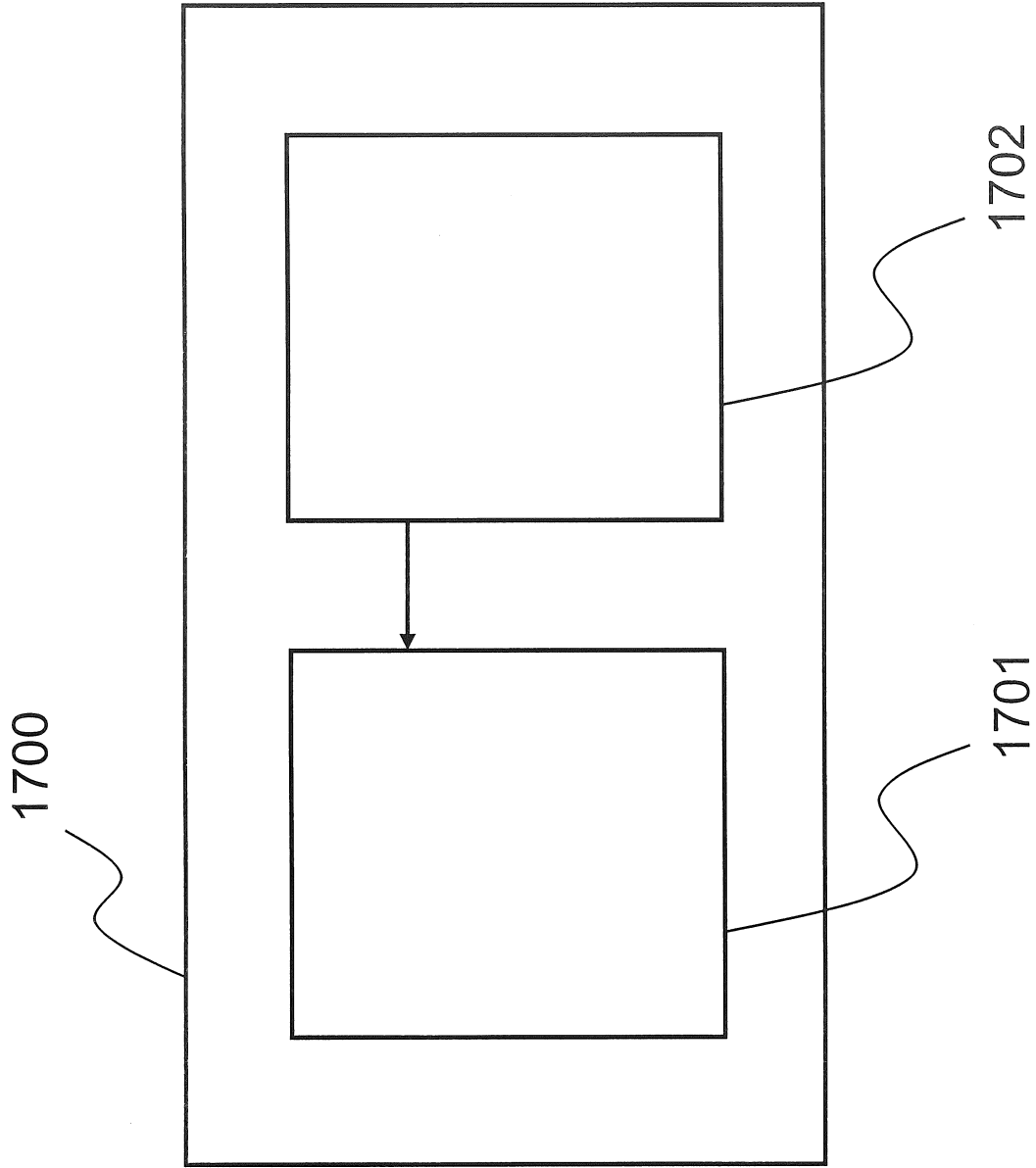


FIG. 17

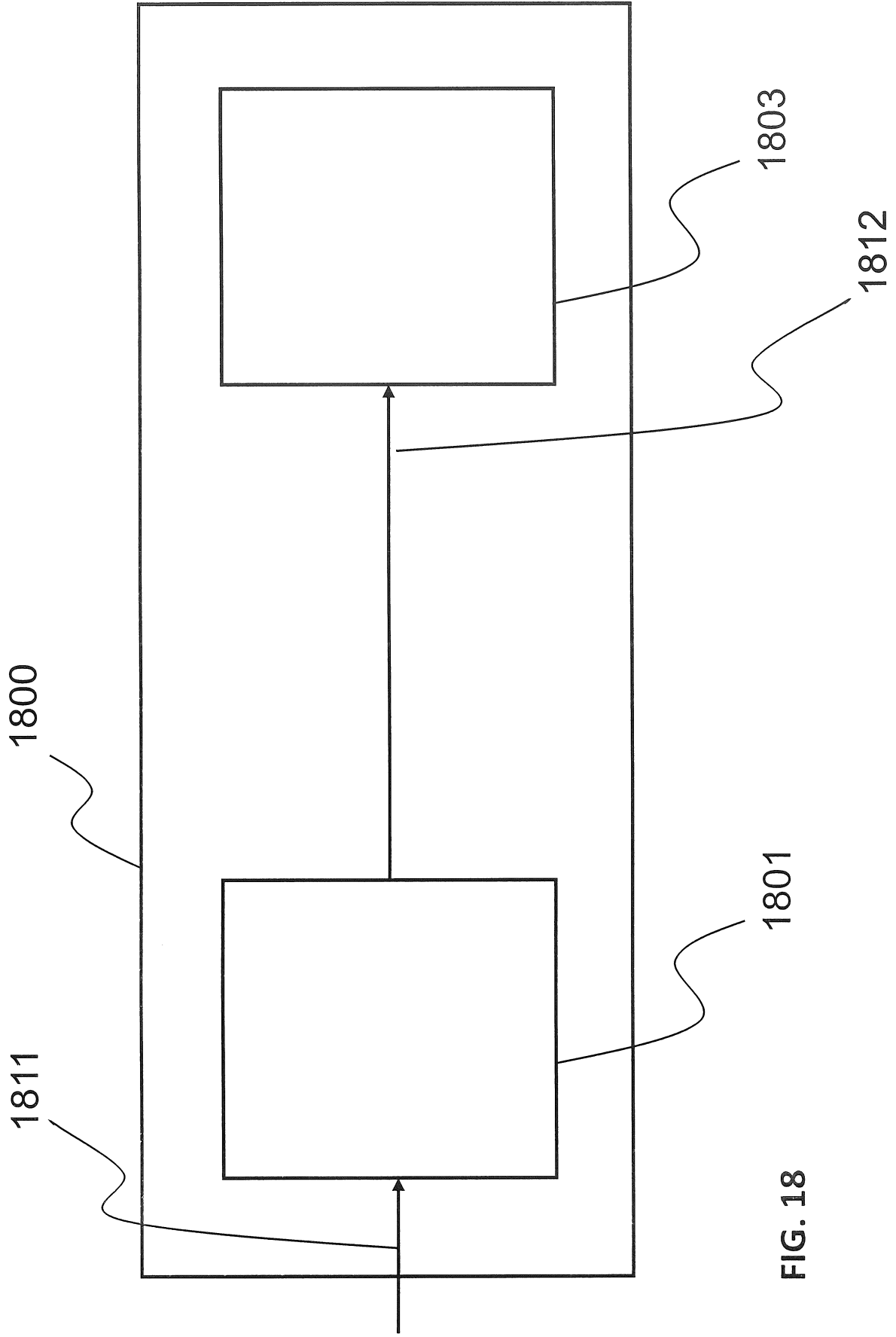


FIG. 18



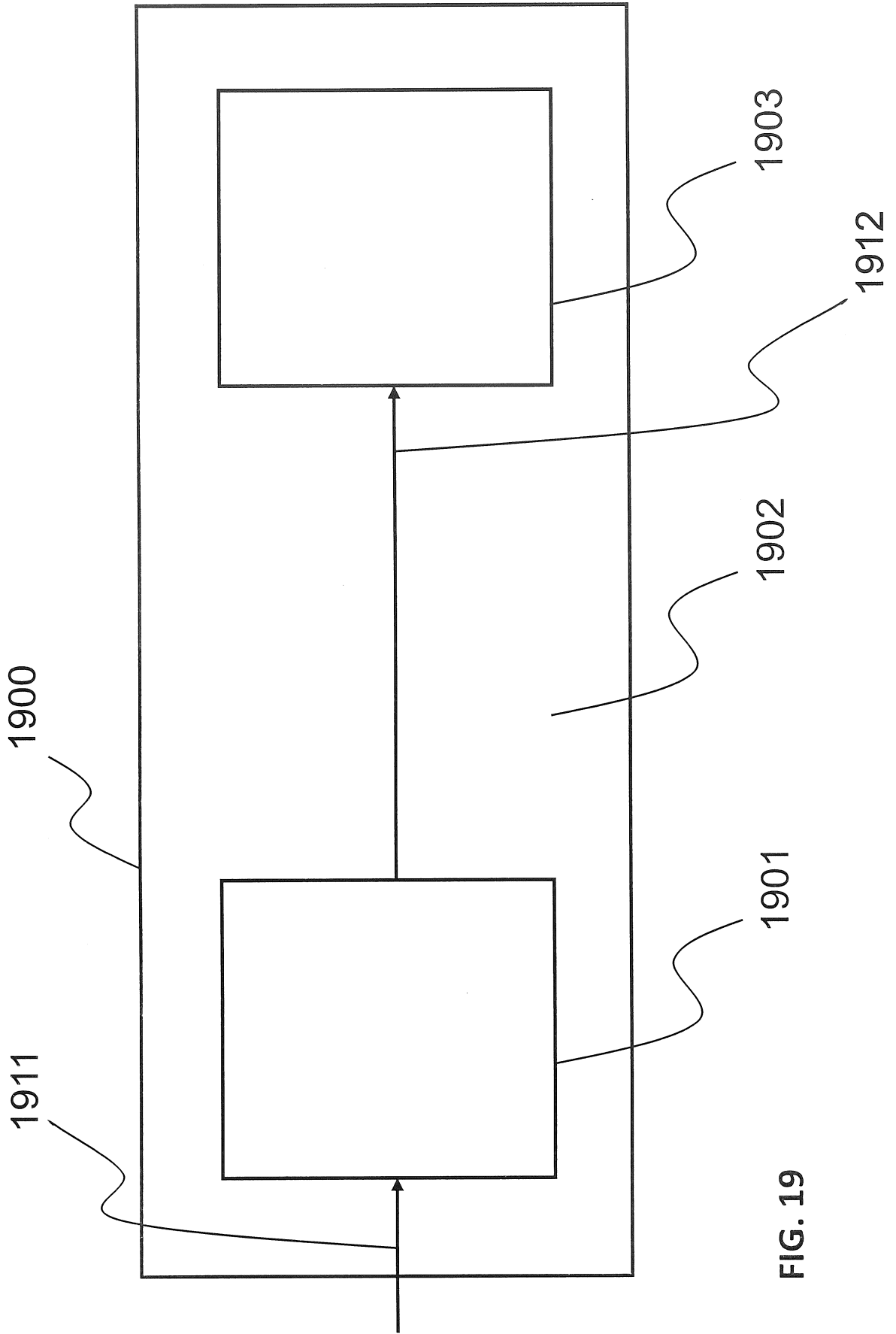


FIG. 19

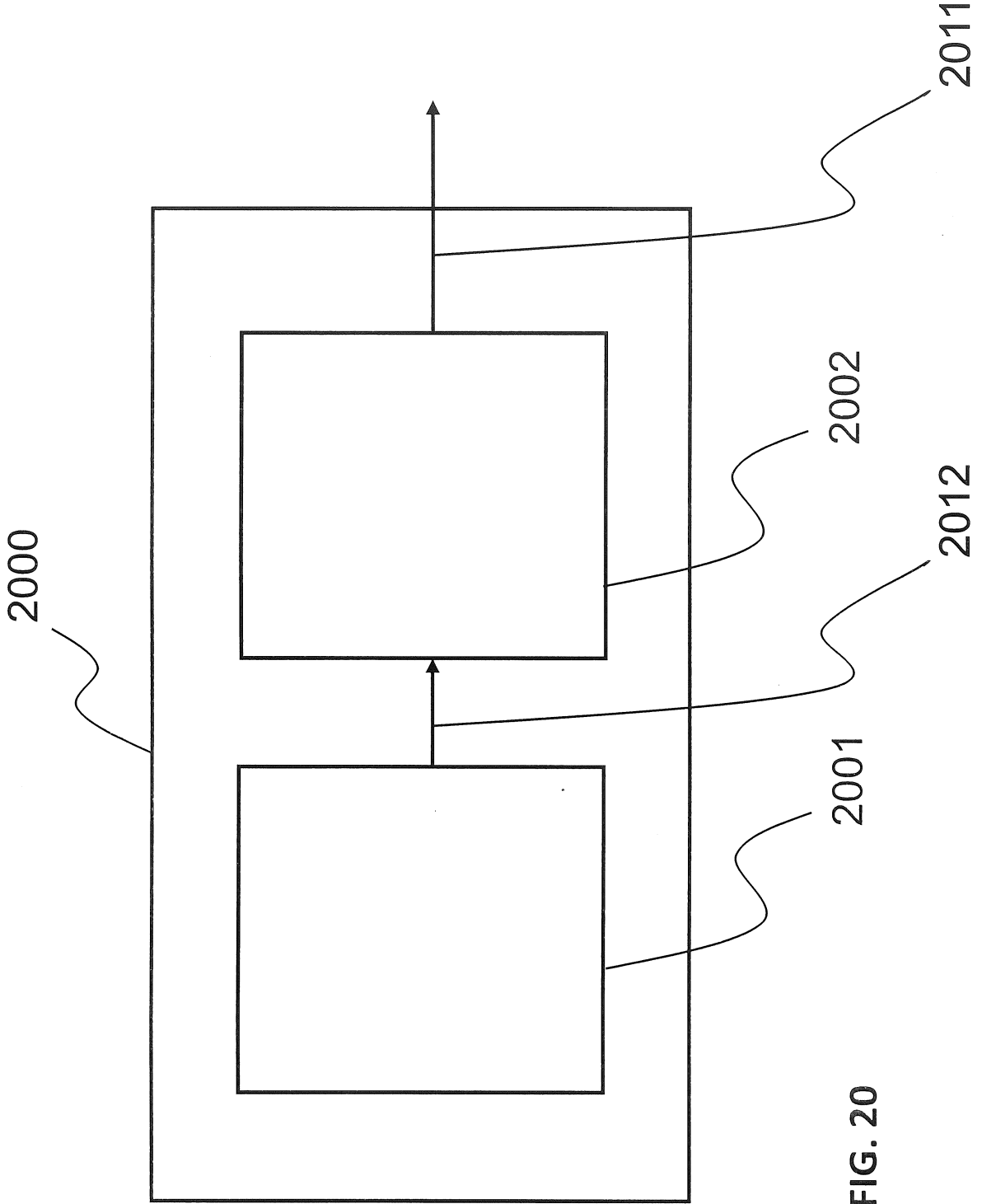


FIG. 20

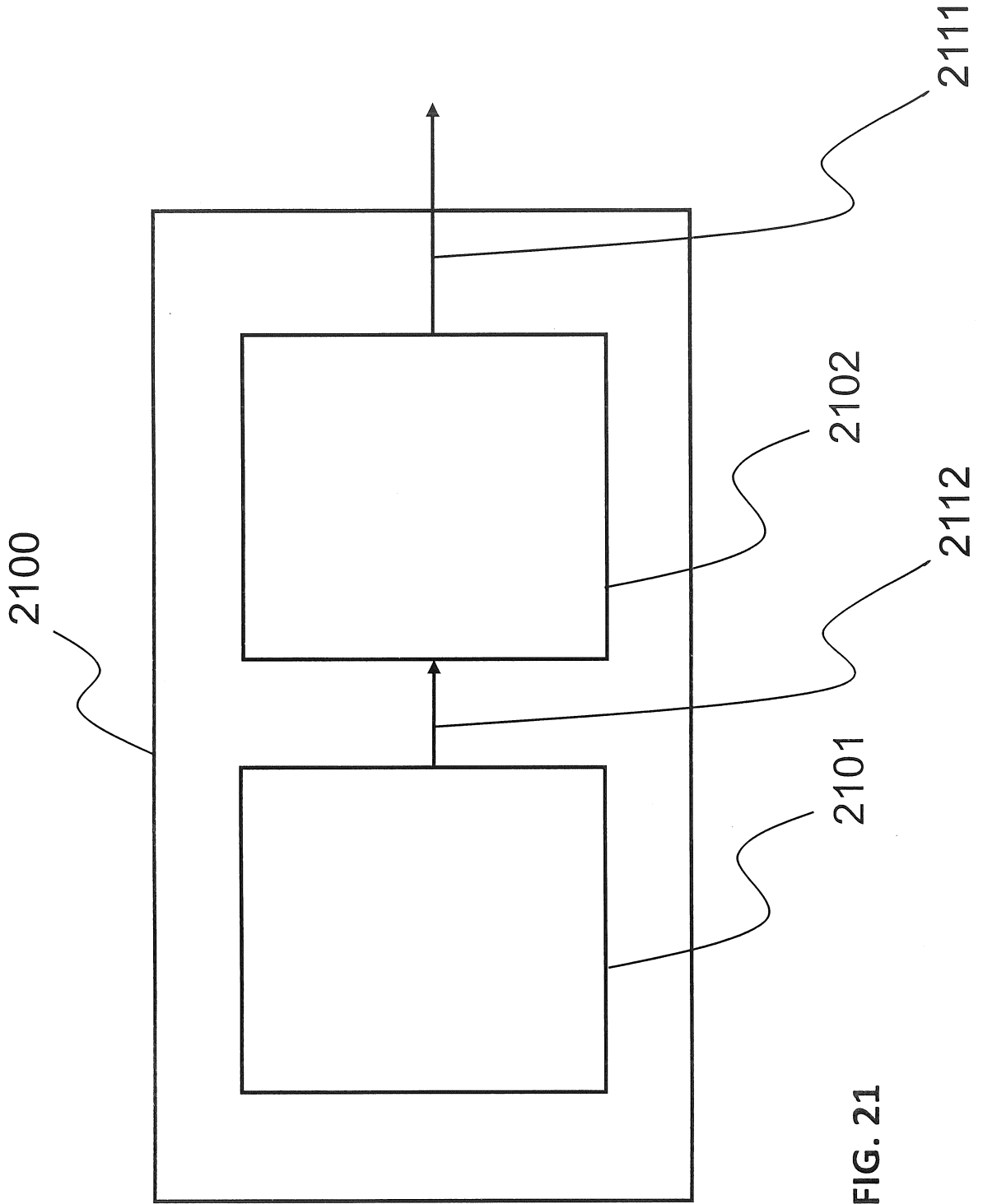


FIG. 21

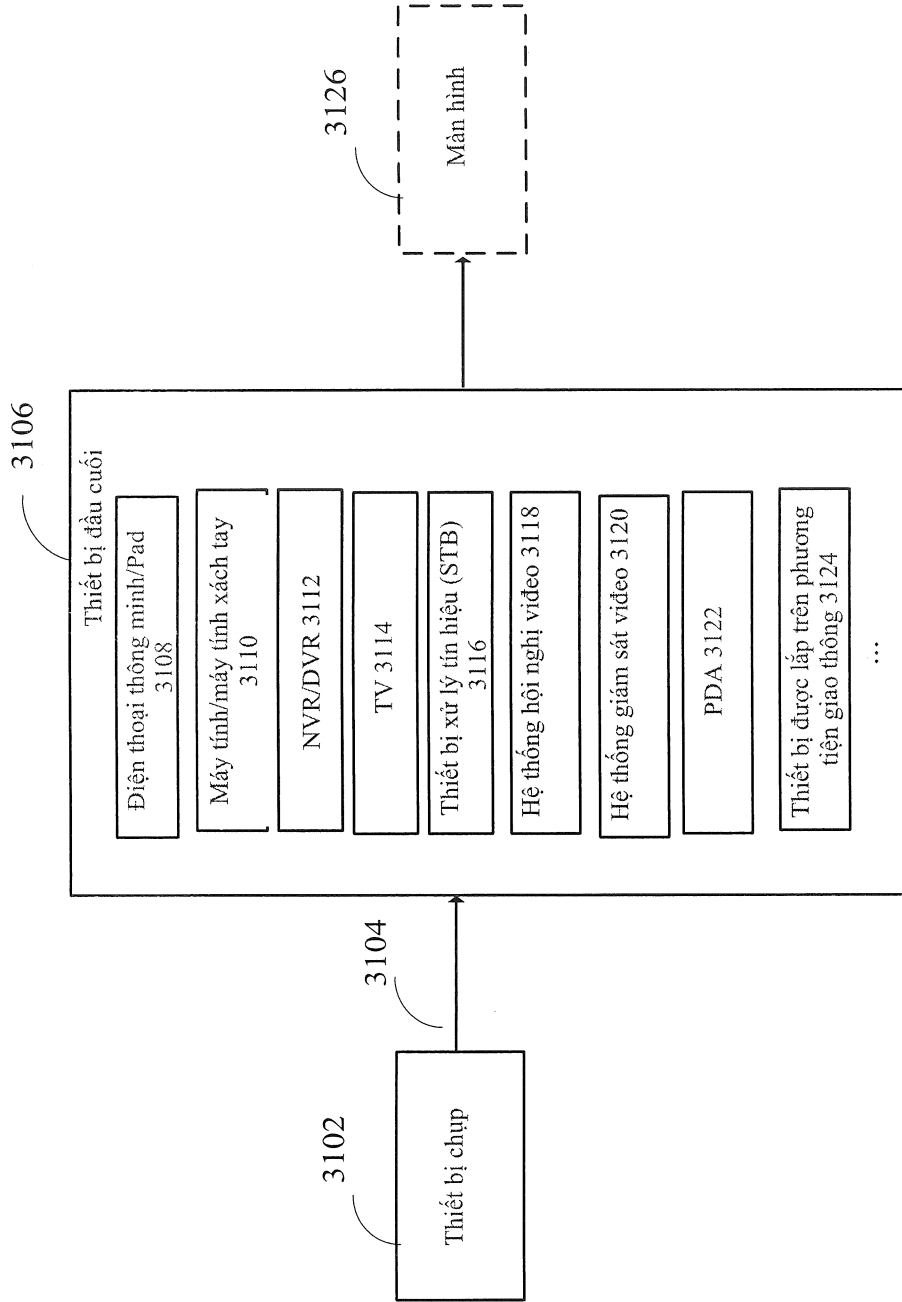


FIG. 22

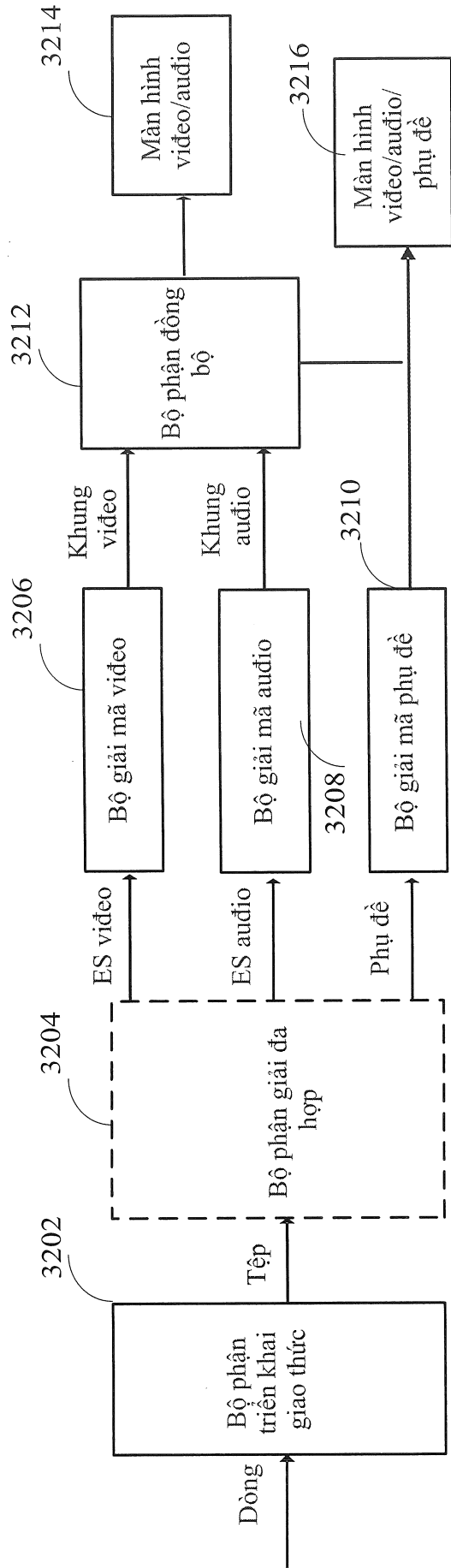


FIG. 23