



(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)



1-0050818

(51)^{2021.01} H04N 19/00

(13) B

(21) 1-2022-06189

(22) 20/02/2021

(86) PCT/CN2021/077000 20/02/2021

(87) WO2021/169859 02/09/2021

(30) PCT/EP2020/055220 28/02/2020 EP

(45) 25/08/2025 449

(43) 25/11/2022 416A

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

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

(72) ESENLİK, Semih (TR); WANG, Biao (CN); KOTRA, Anand Meher (IN);
ALSHINA, Elena Alexandrovna (RU).

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP LẬP MÃ, BỘ GIẢI MÃ, BỘ MÃ HÓA, PHƯƠNG TIỆN LƯU
TRỮ KHÔNG TẠM THỜI, THIẾT BỊ VÀ MÁY GIẢI MÃ VIDEO, VÀ THIẾT BỊ
MÃ HÓA VIDEO

(21) 1-2022-06189

(57) Sáng chế đề cập đến phương pháp lập mã được thực hiện bởi thiết bị giải mã và bộ giải mã, trong đó phương pháp này bao gồm các bước: thu nhận dòng bit đối với ảnh hiện tại; thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; thu nhận trị số của địa chỉ lát hiện tại, lát hiện tại được bao gồm trong ảnh hiện tại; thu nhận trị số của tileX theo trị số của địa chỉ lát hiện tại; phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất. Bộ mã hóa, phương tiện lưu trữ không tạm thời, thiết bị và máy giải mã video, và thiết bị mã hóa video cũng được bộc lộ.

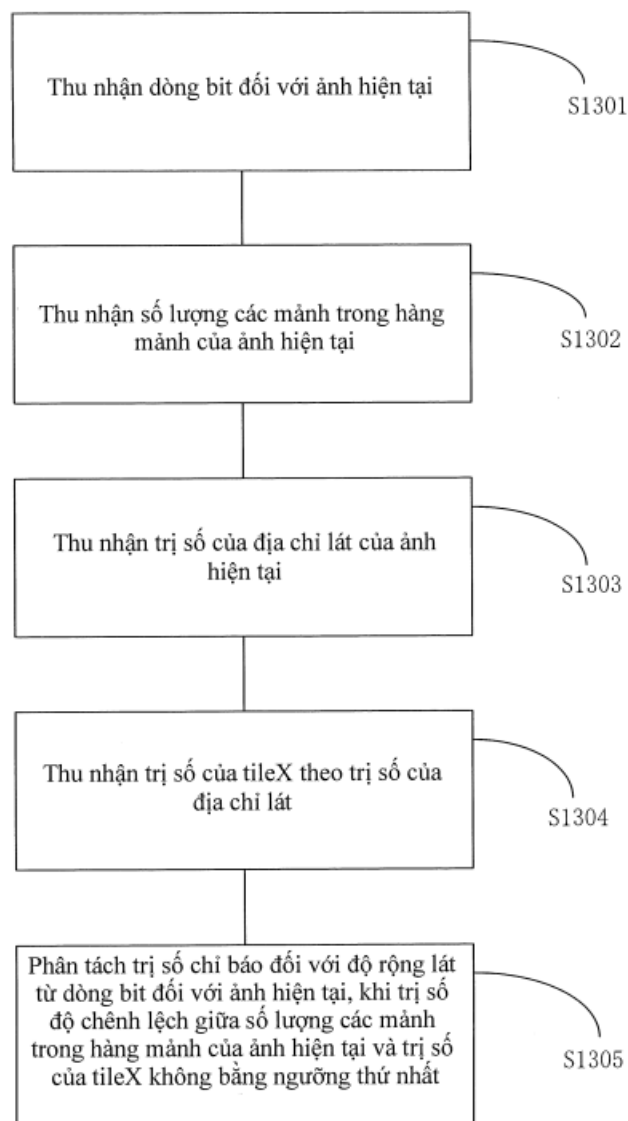


FIG. 13

Lĩnh vực kỹ thuật được đề cập

Các phương án của sáng chế nói chung đề cập đến lĩnh vực xử lý ảnh và cụ thể hơn là đến thông tin phân chia ảnh tín hiệu đối với các lát.

Tình trạng kỹ thuật của sáng chế

Lập mã video (mã hóa và giải mã video) được sử dụng trong phạm vi rộng của các ứng dụng video số, ví dụ máy thu hình kỹ thuật số phát rộng, truyền video qua Internet và các mạng di động, các ứng dụng hội 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 phim kết hợp của các ứng dụng an ninh.

Lượng dữ liệu video được cần để mô tả thậm chí video tương đối ngắn có thể là đáng kể, mà có thể dẫn đến những khó khăn khi dữ liệu cần được tạo dòng hoặc theo cách khác được truyền thông qua mạng truyền thông với dung lượng độ rộng dải giới hạn. Do đó, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông hiện nay. Kích thước của video có thể cũng 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 lập 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 giới hạn và các nhu cầu ngày càng tăng về chất lượng video cao hơn, các kỹ thuật nén và giải nén được cải tiến mà nâng cao tỷ lệ nén với chất lượng hình ảnh suy giảm ít hoặc gần như không suy giảm là điều được mong muốn.

Bản chất kỹ thuật của sáng chế

Các phương án của sáng chế đề xuất các máy và các phương pháp 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 khác là rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Khía cạnh thứ nhất của sáng chế đề xuất phương pháp lập mã được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước: thu nhận dòng bit đối với ảnh hiện tại; thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; thu nhận trị số của địa chỉ lát của ảnh hiện tại; thu nhận trị số của tileX theo trị số của địa chỉ lát; phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

Theo các phương án của sáng chế, sơ đồ báo hiệu được bộc lộ, chỉ có ít nhất hai cột mảnh, thì thành phần cú pháp đối với độ rộng lát được báo hiệu. Vì vậy, hiệu quả giải mã và ứng dụng dòng bit đã được nâng cao.

Ngưỡng thứ nhất là trị số nguyên, ngưỡng thứ nhất có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong hàng mảnh có nghĩa là giống như số các cột mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX là bằng ngưỡng thứ nhất. Trị số mặc định là trị số nguyên, theo ví dụ, trị số mặc định có thể là 0, hoặc 1.

Theo một cách thực hiện, trong đó trị số mặc định chỉ báo rằng độ rộng của lát bao gồm 1 cột mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ nhất”. Vì vậy, trị số của tileX ban đầu không được tính nhưng được biểu diễn theo kết quả của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó trị số của tileX chỉ báo trị số của tọa độ theo chiều ngang. Theo ví dụ, trị số của tileX chỉ báo địa chỉ theo chiều ngang trong chỉ số mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; thu nhận trị số của tileY theo trị số của địa chỉ lát; phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của

tileY không bằng ngưỡng thứ hai. Ngưỡng thứ hai là trị số nguyên, ngưỡng thứ hai có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong cột mảnh có nghĩa là giống như số các hàng mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY là bằng ngưỡng thứ hai.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Theo một cách thực hiện, trong đó trị số của tileY chỉ báo trị số của tọa độ theo chiều dọc. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong cột mảnh của ảnh hiện tại không bằng ngưỡng thứ hai”. Vì vậy, trị số của tileY ban đầu không được tính nhưng được biểu diễn theo kết quả của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Theo ví dụ, trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh.

Khía cạnh thứ hai của sáng chế đề xuất phương pháp lập mã được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước: thu nhận dòng bit đối với ảnh hiện tại; thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; thu nhận trị số

của địa chỉ lát của ảnh hiện tại; thu nhận trị số của tileY theo trị số của địa chỉ lát; phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

Ngưỡng thứ hai là trị số nguyên, ngưỡng thứ hai có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong cột mảnh có nghĩa là giống như số các hàng mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY là bằng ngưỡng thứ hai. Trị số mặc định là trị số nguyên, theo ví dụ, trị số mặc định có thể là 0, hoặc 1.

Theo một cách thực hiện, trong đó trị số mặc định chỉ báo rằng độ cao của lát bao gồm 1 hàng mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của tileY chỉ báo trị số của tọa độ theo chiều dọc. Theo ví dụ, trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; thu nhận trị số của tileX theo trị số của địa chỉ lát; phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất. Ngưỡng thứ nhất là trị số nguyên, ngưỡng thứ nhất có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong hàng mảnh có nghĩa là giống như số các cột mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX là bằng ngưỡng thứ nhất. Trị số mặc định là trị số nguyên, theo ví dụ, trị số mặc định có thể là 0, hoặc 1.

Theo một cách thực hiện, trong đó trị số mặc định chỉ báo rằng độ rộng của lát bao gồm 1 cột mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ nhất”. Vì vậy, trị số của tileX ban đầu không được tính nhưng được biểu diễn theo kết quả của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó trị số của tileX chỉ báo trị số của tọa độ theo chiều ngang.

Khía cạnh thứ ba của sáng chế đề xuất phương pháp lập mã được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước: thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; thu nhận trị số của địa chỉ lát của ảnh hiện tại; thu nhận trị số của tileX theo trị số của địa chỉ lát; lập mã trị số chỉ báo đối với độ rộng lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

Theo các phương án của sáng chế, sơ đồ báo hiệu được bộc lộ, chỉ có ít nhất hai cột mảnh, thì thành phần cú pháp đối với độ rộng lát được báo hiệu. Vì vậy, hiệu quả giải mã và ứng dụng dòng bit đã được nâng cao.

Ngưỡng thứ nhất là trị số nguyên, ngưỡng thứ nhất có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong hàng mảnh có nghĩa là giống như số các cột mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ nhất”. Vì vậy, trị số của tileX ban đầu không được tính nhưng được biểu diễn theo kết quả của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó trị số của tileX chỉ báo trị số của tọa độ theo chiều ngang.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; thu nhận trị số của tileY theo trị số của địa chỉ lát; lập mã trị số chỉ báo đối với độ cao lát trong dòng bit đối với ảnh hiện tại, khi trị

số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai. Ngưỡng thứ hai là trị số nguyên, ngưỡng thứ hai có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong cột mảnh có nghĩa là giống như số các hàng mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Theo một cách thực hiện, trong đó trị số của tileY chỉ báo trị số của tọa độ theo chiều dọc. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ hai”. Vì vậy, trị số của tileY ban đầu không được tính nhưng được biểu diễn theo kết quả của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong cột mảnh của ảnh hiện tại.

Khía cạnh thứ tư của sáng chế đề xuất phương pháp lập mã được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước: thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; thu nhận trị số của địa chỉ lát của ảnh hiện tại; thu nhận trị số của tileY theo trị số của địa chỉ lát; lập mã trị số chỉ báo đối với độ cao lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

Ngưỡng thứ hai là trị số nguyên, ngưỡng thứ hai có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong cột mảnh có nghĩa là giống như số các hàng mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm:

thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của tileY chỉ báo trị số của tọa độ theo chiều dọc.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó phương pháp còn bao gồm: thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; thu nhận trị số của tileX theo trị số của địa chỉ lát; lập mã trị số chỉ báo đối với độ rộng lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất. Ngưỡng thứ nhất là trị số nguyên, ngưỡng thứ nhất có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong hàng mảnh có nghĩa là giống như số các cột mảnh.

Theo một cách thực hiện, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo một cách thực hiện, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ nhất”. Vì vậy, trị số của tileX ban đầu không được tính nhưng được biểu diễn theo kết quả của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo một cách thực hiện, trong đó trị số của tileX chỉ báo trị số của tọa độ theo chiều ngang.

khía cạnh thứ năm của sáng chế đề xuất máy giải mã video, máy bao gồm:

môđun thu nhận, mà được tạo cấu hình để thu nhận dòng bit đối với ảnh hiện tại; và môđun thu nhận được tạo cấu hình để thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại và thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, mà được tạo cấu hình để thu nhận trị số của tileX theo trị số của địa chỉ lát;

môđun phân tách, mà được tạo cấu hình để phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

Chi tiết đối với mỗi môđun theo khía cạnh thứ năm của sáng chế có thể tham chiếu tới các phương án và các cách thực hiện theo khía cạnh thứ nhất nêu trên của sáng chế.

Khía cạnh thứ sáu của sáng chế đề xuất thiết bị giải mã video, thiết bị giải mã bao gồm:

môđun thu nhận, mà được tạo cấu hình để thu nhận dòng bit đối với ảnh hiện tại; môđun thu nhận được tạo cấu hình để thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại, và thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, mà được tạo cấu hình để thu nhận trị số của tileY theo trị số của địa chỉ lát;

môđun phân tách, mà được tạo cấu hình để phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

Chi tiết đối với mỗi môđun theo khía cạnh thứ sáu của sáng chế có thể tham chiếu tới các phương án và các cách thực hiện theo khía cạnh thứ hai nêu trên của sáng chế.

Khía cạnh thứ bảy của sáng chế đề xuất thiết bị mã hóa video, thiết bị mã hóa bao gồm:

môđun thu nhận, mà được tạo cấu hình để thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; môđun thu nhận được tạo cấu hình để thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, mà được tạo cấu hình để thu nhận trị số của tileX theo trị số của địa chỉ lát;

môđun lập mã, mà được tạo cấu hình để lập mã trị số chỉ báo đối với độ rộng lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

Chi tiết đối với mỗi môđun theo khía cạnh thứ bảy của sáng chế có thể tham chiếu tới các phương án và các cách thực hiện theo khía cạnh thứ ba nêu trên của sáng chế.

Khía cạnh thứ tám của sáng chế đề xuất thiết bị mã hóa video, thiết bị mã hóa video bao gồm: môđun thu nhận, mà được tạo cấu hình để thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; môđun thu nhận được tạo cấu hình để thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, mà được tạo cấu hình để thu nhận trị số của tileY theo trị số của địa chỉ lát;

môđun lập mã, mà được tạo cấu hình để lập mã trị số chỉ báo đối với độ cao lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

Chi tiết đối với mỗi môđun theo khía cạnh thứ tám của sáng chế có thể tham chiếu tới các phương án và các cách thực hiện theo khía cạnh thứ tư nêu trên của sáng chế.

Khía cạnh thứ chín của sáng chế đề xuất bộ giải mã bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ nhất, khía cạnh thứ hai và bất kỳ một trong số cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ hai.

Khía cạnh thứ mười của sáng chế đề xuất bộ giải mã bao gồm hệ mạch xử lý để thực hiện phương pháp theo khía cạnh thứ ba, khía cạnh thứ tư và bất kỳ một trong số cách thực hiện của khía cạnh thứ ba, khía cạnh thứ tư.

Khía cạnh thứ mười của sáng chế đề xuất sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo khía cạnh thứ nhất, khía cạnh thứ hai, khía cạnh thứ ba, khía cạnh thứ tư và bất kỳ một trong số cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ hai, khía cạnh thứ ba và khía cạnh thứ tư.

Khía cạnh thứ mười một của 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 thứ nhất, khía cạnh thứ hai, khía cạnh thứ ba, khía cạnh thứ tư và bất kỳ một trong số cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ hai, khía cạnh thứ ba và khía cạnh thứ tư.

Khía cạnh thứ mười hai của 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 cho việc 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, khía cạnh thứ hai và bất kỳ một trong số cách thực hiện của khía cạnh thứ nhất, khía cạnh thứ hai.

Khía cạnh thứ mười ba của 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 cho việc 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 ba, khía cạnh thứ tư và bất kỳ một trong số cách thực hiện của khía cạnh thứ ba, khía cạnh thứ tư.

Mô tả vắn tắt các hình vẽ

Trong phần dưới đây, các phương án của sáng chế được mô tả chi tiết dựa vào các dấu hiệu và hình vẽ kèm theo, trong đó:

Fig.1A là sơ đồ khối thể hiện ví dụ của hệ thống lập 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 của hệ thống lập 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 về 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 là ví dụ về ảnh được chia thành các CTU;

Fig.7 là ví dụ về thứ tự quét mảnh trên cơ sở mảnh của các CTU;

Fig.8 là ví dụ về việc phân chia ảnh;

Fig.9 là ví dụ về thông tin địa chỉ CTU trong ảnh;

Fig.10 là ví dụ khác về việc phân chia ảnh;

Fig.11 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;

Fig.12 là sơ đồ khối thể hiện cấu trúc của ví dụ về thiết bị đầu cuối.

Fig.13 là lưu đồ thể hiện phương án về phương pháp.

Fig.14 là sơ đồ khối thể hiện phương án về máy.

Trong phần dưới đây, các ký hiệu tham chiếu giống hệt nhau đề cập đến các đặc điểm giống hệt nhau hoặc ít nhất tương đương về mặt chức năng nếu không được định rõ theo cách khác.

Mô tả chi tiết sáng chế

Trong phần mô tả dưới đây, tham chiếu được thực hiện tới các hình vẽ kèm theo, mà tạo nên một phần 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 trong đó các phương án của sáng chế có thể được sử dụng. Được 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 trong các hình vẽ. Do đó, phần mô tả chi tiết dưới đây 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, được hiểu rằng phần bộc lộ kết hợp với phương pháp được mô tả có thể cũng giữ 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 nhiều bước phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, để thực hiện một hoặc nhiều bước phương pháp được mô tả (ví dụ, một bộ phận thực hiện một hoặc nhiều bước, hoặc nhiều bộ phận đều thực hiện một hoặc nhiều trong số nhiều bước), ngay cả khi một hoặc nhiều bộ phận như vậy không được mô tả hoặc được minh họa rõ ràng trong các hình vẽ. Mặt khác, ví dụ, nếu máy cụ thể được mô tả dựa vào một hoặc nhiều bộ phận, ví dụ, các bộ phận chức năng, phương pháp tương ứng có thể bao gồm một bước để thực hiện chức năng của một hoặc nhiều bộ phận (ví dụ, một bước thực hiện chức năng của một hoặc nhiều bộ phận, hoặc nhiều bước đều thực hiện chức năng của một hoặc nhiều trong số nhiều bộ phận), ngay cả khi một hoặc nhiều bước như vậy không được mô tả hoặc được minh họa rõ ràng trong các hình vẽ. Hơn nữa, được hiểu rằng các đặc điểm của các phương án và/hoặc các khía cạnh lấy làm

ví dụ khác nhau được mô tả ở đây có thể được kết hợp với phương án và/hoặc khía cạnh khác, trừ khi được lưu ý cụ thể theo cách khác.

Lập mã video đề cập cụ thể đến việc xử lý của chuỗi của chuỗi của các ảnh, mà nó tạo nên video hoặc chuỗi video. Thay vì thuật ngữ “ảnh (picture)” thuật ngữ “khung (frame)” hoặc “hình ảnh (image)” có thể được sử dụng như từ đồng nghĩa trong lĩnh vực lập mã video. Lập mã video (hoặc lập mã nói chung) bao gồm hai phần là mã hóa video và giải mã video. Mã hóa video được thực hiện ở phía nguồn, cụ thể là 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 (cho việc lưu trữ và/hoặc truyền hữu hiệu hơn). Giải mã video được thực hiện ở phía đích và cụ thể là bao gồm việc xử lý ngược so với bộ mã hóa để khôi phục các ảnh video. Các phương án đề cập đến “việc lập mã” của các ảnh video (hoặc các ảnh nói chung) sẽ được hiểu là liên quan đến “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 đề cập đến là bộ lập mã -giải mã (Coding and Decoding, viết tắt là CODEC).

Trong trường hợp lập mã video không tổn thất, các ảnh video gốc có thể được khôi phục, nghĩa là, các ảnh video được khôi phục có chất lượng giống như các ảnh video gốc (giả sử không có tổn thất truyền hoặc tổn thất dữ liệu khác trong suốt thời gian lưu trữ hoặc truyền). Trong trường hợp lập mã video tổn thất, việc nén thêm, ví dụ, bằng cách lượng tử hóa, được thực hiện, để làm giảm lượng dữ liệu biểu diễn các ảnh video, mà không thể được khôi phục hoàn toàn ở bộ giải mã, nghĩa là, chất lượng của các ảnh video được khôi phục là thấp hơn hoặc tồi hơn so với chất lượng của các ảnh video gốc.

Một vài chuẩn lập mã video thuộc về nhóm của “các bộ lập mã -giải mã video lai tổn thất” (nghĩa là, kết hợp việc dự đoán theo không gian và thời gian trong miền mẫu và việc lập mã biến đổi 2D để áp dụng việc lượng tử hóa trong miền biến đổi). Mỗi ảnh của chuỗi của chuỗi video cụ thể được phân chia thành tập hợp của các khối không

chồng lẩn và việc lập mã cụ thể được thực hiện ở mức khối. Nói cách khác, ở bộ mã hóa video cụ thể được xử lý, nghĩa là, được mã hóa, ở mức khối (khối video), ví dụ, nhờ sử dụng việc dự đoán theo không gian (ảnh bên trong) và/hoặc việc dự đoán theo thời gian (ảnh liên kết) để tạo ra khối dự đoán, trừ khối dự đoán từ khối hiện tại (khối hiện đang được xử lý/sẽ 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 việc xử lý ngược so với bộ mã hóa được áp dụng tới khối được mã hóa hoặc được nén để khôi phục khối hiện tại cho việc biểu diễn. Hơn nữa, bộ mã hóa nhân đôi vòng lặp xử lý bộ giải mã sao cho cả hai sẽ tạo ra các sự dự đoán giống hệt nhau (ví dụ, các sự dự đoán bên trong và liên kết) và/hoặc các sự khôi phục đối với việc xử lý, nghĩa là, lập mã, các khối tiếp theo.

Trong phần dưới đây, các phương án của hệ thống lập mã video 10, bộ mã hóa video 20 và bộ giải mã video 30 được mô tả dựa vào các hình vẽ từ Fig.1 đến Fig.3.

Fig.1A là sơ đồ khối giản lược minh họa hệ thống lập mã ví dụ 10, ví dụ, hệ thống lập mã video 10 (hoặc ngắn gọn là hệ thống lập mã 10) mà có thể ứng dụng các kỹ thuật của sáng chế này. Bộ mã hóa video 20 (hoặc ngắn gọn là bộ mã hóa 20) và bộ giải mã video 30 (hoặc ngắn gọn là bộ giải mã 30) của hệ thống lập 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 trong sáng chế.

Như được thể hiện trên Fig.1A, hệ thống lập 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 theo cách bổ sung, nghĩa là, một cách tùy chọn, bao gồm nguồn ảnh 16, bộ tiền xử lý (hoặc bộ phận tiền xử lý) 18, ví dụ, bộ tiền xử lý ảnh 18, và giao diện truyền thông hoặc bộ phận truyền thông 22.

Nguồn ảnh 16 có thể bao gồm hoặc là loại bất kỳ của thiết bị chụp ảnh, ví dụ camera dùng để chụp ảnh thể giới thực, và/hoặc loại bất kỳ của thiết bị tạo ảnh, ví dụ bộ xử lý đồ họa máy tính để tạo ra ảnh động máy tính, hoặc loại bất kỳ của thiết bị khác để thu nhận và/hoặc cung cấp ảnh thể giới thực, ảnh được tạo 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 nó (ví dụ, ảnh thực tế tăng cường (augmented reality, viết tắt là AR)). Nguồn ảnh có thể là loại bất kỳ của bộ nhớ hoặc bộ phận lưu trữ lưu trữ ảnh bất kỳ trong số các ảnh nêu trên.

Phân biệt với bộ tiền xử lý 18 và việc xử lý được thực hiện bởi bộ phận tiền xử lý 18, ảnh hoặc dữ liệu ảnh 17 có thể cũng được đề cập đến là ảnh thô hoặc dữ liệu ảnh thô 17.

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

Bộ mã hóa video 20 được tạo cấu hình để thu dữ liệu ảnh được tiền xử lý 19 và cung cấp dữ liệu ảnh được mã hóa 21 (chi tiết hơn 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 nó) qua kênh truyền thông 13 tới thiết bị khác, ví dụ, thiết bị đích 14 hoặc thiết bị bất kỳ khác, cho việc lưu trữ hoặc việc khôi phục trực tiếp.

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

Giao diện truyền thông 28 của thiết bị đích 14 được tạo cấu hình thu dữ liệu ảnh được mã hóa 21 (hoặc hoặc phiên bản được xử lý thêm bất kỳ của nó), ví dụ, một cách trực tiếp từ thiết bị nguồn 12 hoặc từ nguồn bất kỳ khác, 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 không dây hoặc nối dây trực tiếp, hoặc qua loại bất kỳ của mạng, ví dụ, mạng không dây hoặc nối dây hoặc sự kết hợp bất kỳ của nó, hoặc loại bất kỳ của mạng cá nhân và công cộng, hoặc loại bất kỳ của sự kết hợp của nó.

Giao diện truyền thông 22 có thể, ví dụ, được tạo cấu hình để gói dữ liệu ảnh được mã hóa 21 thành định dạng thích hợp, ví dụ, gói, và/hoặc xử lý dữ liệu ảnh được mã hóa nhờ sử dụng loại bất kỳ của việc xử lý hoặc mã hóa truyền 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 bất kỳ của việc xử lý và/hoặc giải gói hoặc giải mã truyền tương ứng để thu nhận dữ liệu ảnh được mã hóa 21.

Cả giao diện truyền thông 22 và giao diện truyền thông 28 đều có thể được tạo cấu hình như các giao diện truyền thông một chiều như được chỉ báo bởi mũi tên đối với kênh truyền thông 13 trên Fig.1A trở từ thiết bị nguồn 12 tới 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 đặt sự kết nối, để báo nhận và trao đổi thông tin bất kỳ khác 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 hơn sẽ được mô tả dưới đây, ví dụ, dựa vào Fig.3 hoặc Fig.5).

Bộ hậu xử lý 32 của thiết bị đích 14 được tạo cấu hình để hậu xử lý dữ liệu ảnh được giải mã 31 (cũng được gọi là dữ liệu ảnh được khôi phục), ví dụ, ảnh được giải mã 31, để thu nhận dữ liệu ảnh được hậu xử lý 33, ví dụ, ảnh được hậu xử lý 33. Việc hậu xử lý được thực hiện bởi bộ phận hậu xử lý 32 có thể bao gồm, ví dụ, chuyển đổi định dạng màu sắc (ví dụ, từ YCbCr tới RGB), chỉnh sửa màu sắc, tinh chỉnh, hoặc lấy mẫu lại, hoặc việc xử lý bất kỳ khác, ví dụ, để chuẩn bị dữ liệu ảnh được giải mã 31 cho việc 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 hậu xử lý 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 bất kỳ của màn hình để biểu diễn ảnh được khôi phục, ví dụ, màn hiển thị hoặc màn hình 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 diode, viết tắt là OLED), các màn hình plasma, các máy chiếu, các màn hình vi LED, tinh thể lỏng trên silicon (liquid crystal on silicon, viết tắt là LCoS), bộ xử lý ánh sáng số (digital light processor, viết tắt là DLP) hoặc loại bất kỳ của màn hình khác.

Mặc dù Fig.1A thể hiện thiết bị nguồn 12 và thiết bị đích 14 dưới dạng các thiết bị tách biệt, các phương án về các thiết bị có thể cũng bao gồm cả hai hoặc cả hai chức năng, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng

tương ứng. Theo các phương án như vậy, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức 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 tách biệt hoặc sự kết hợp bất kỳ của nó.

Như sẽ rõ ràng đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng dựa vào phần mô tả, sự tồn tại và sự phân tách (chính xác) của các chức năng của các bộ phận khác nhau hoặc các chức năng 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 ứng dụng và thiết bị thực tế.

Bộ mã hóa 20 (ví dụ, bộ mã hóa video 20) hoặc bộ giải mã 30 (ví dụ, bộ giải mã video 30) hoặc cả bộ mã hóa 20 và bộ giải mã 30 đều 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 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, lập mã video chuyên dụng hoặc các sự kết hợp bất kỳ của nó. 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 đối với bộ mã hóa 20 trên Fig.2 và/hoặc hệ thống hoặc hệ thống con bộ mã hóa bất kỳ khác 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 đối với bộ giải mã 30 trên Fig.3 và/hoặc hệ thống hoặc hệ thống con bộ giải mã bất kỳ khác 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 mô tả dưới đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được thực hiện một phần trong phần mềm, thiết bị có thể lưu trữ các lệnh đối với phần mềm trong phương tiện lưu trữ đọc được bởi máy tính không tạm thời, phù hợp và có thể thực hiện các lệnh trong phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của

sáng chế này. 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 lẻ, ví dụ, như được thể hiện trên Fig.1B.

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong số phạm vi rộng của các thiết bị, bao gồm loại bất kỳ của các thiết bị cố định hoặc cầm tay, ví dụ, các máy tính bảng cỡ nhỏ (notebook) hoặc các máy tính xách tay (laptop), các điện thoại di động, các điện thoại thông minh, các bảng (tablets) hoặc các máy tính bảng (tablet), các camera, các máy tính để bàn, các thiết bị giải mã tín hiệu truyền hình (set-top boxe), các máy thu hình, các thiết bị hiển thị, các máy chơi phương tiện số (digital media player), các bảng điều khiển trò chơi video, các thiết bị tạo dòng video (chẳng hạn như các máy chủ dịch vụ nội dung hoặc các máy chủ phân phối nội dung), thiết bị bộ thu phát rộng, thiết bị bộ truyền phát rộng, hoặc tương tự và có thể không sử dụng hoặc loại bất kỳ của hệ điều hành. 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. Do đó, 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 lập mã video 10 được minh họa trên Fig.1A chỉ là ví dụ và các kỹ thuật của sáng chế có thể áp dụng tới các việc thiết đặt lập mã video (ví dụ, mã hóa video hoặc giải mã video) mà không nhất 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 một thiết bị khác, nhưng dễ dàng mã hóa dữ liệu tới bộ nhớ và/hoặc truy xuất và giải mã dữ liệu từ bộ nhớ.

Nhằm thuận lợi cho việc mô tả, các phương án của sáng chế được mô tả ở đây, ví dụ, bằng cách tham chiếu tới lập mã video hiệu quả cao (High-Efficiency Video Coding,

viết tắt là HEVC) hoặc tới phần mềm tham chiếu của lập mã video đa năng (Versatile Video coding, viết tắt là VVC), chuẩn lập mã video thế hệ tiếp theo được phát triển bởi nhóm cộng tác chung về lập mã video (Joint Collaboration Team on Video Coding, viết tắt là JCT-VC) của nhóm các chuyên gia lập 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 là 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 ví dụ 20 mà được tạo cấu hình để thực hiện các kỹ thuật của 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 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 khôi phục 214, bộ phận bộ 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 dự đoán liên kết 244, bộ phận dự đoán bên trong 254 và bộ phận phân chia 262. Bộ phận dự đoán liên kết 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 có thể cũng được đề cập đến là bộ mã hóa video lai hoặc bộ mã hóa video theo bộ lập mã -giải mã video lai .

Bộ phận tính 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 như việc tạo nên đường tín hiệu thuận của bộ mã hóa 20, trong khi bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận khôi phục 214, bộ đệm 216, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận dự đoán liên kết 244 và bộ phận dự đoán bên trong

254 có thể được đề cập đến như việc tạo nên đường tín hiệu ngược của bộ mã hóa video 20, trong đó đường tín hiệu ngược của bộ mã hóa video 20 tương ứng với đường tín 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 khôi phục 214, bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận dự đoán liên kết 244 và bộ phận dự đoán bên trong 254 cũng được tham chiếu tới việc tạo nên “bộ giải mã lắp sẵn” 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 của các ảnh tạo nên video hoặc chuỗi video. Ảnh được thu hoặc dữ liệu ảnh có thể cũng là ảnh được tiền xử lý 19 (hoặc dữ liệu ảnh được tiền xử lý 19). Nhằm đơn giản, phần mô tả dưới đây đề cập đến ảnh 17. Ảnh 17 có thể cũng được đề cập đến là ảnh hiện tại hoặc ảnh cần được lập mã (cụ thể là trong lập mã video để phân biệt ảnh hiện tại với các ảnh khác, ví dụ, được mã hóa 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) được hoặc có thể được coi là mảng hai chiều hoặc ma trận của các mẫu với các trị số cường độ. Mẫu trong mảng có thể cũng được đề cập đến là điểm ảnh (dạng ngắn gọn của phần tử ảnh) hoặc ảnh điểm (pel). Số các mẫu theo chiều (trục) ngang và dọc của mảng hoặc ảnh định rõ kích thước và/hoặc độ phân giải của ảnh. Đối với sự biểu diễn của màu sắc, cụ thể là ba thành phần màu sắc được ứng dụng, nghĩa là, ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Theo định dạng RGB hoặc không gian màu sắc, ảnh bao gồm mảng mẫu màu đỏ, xanh lá cây và xanh da trời tương ứng. Tuy nhiên, trong lập mã video, mỗi điểm ảnh cụ thể được biểu diễn theo định dạng sắc độ và độ chói hoặc không gian màu sắc, ví dụ, YCbCr, mà bao gồm thành phần độ chói được chỉ báo bởi Y (đôi khi 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 ngắn gọn là luma) Y biểu diễn độ sáng chói hoặc cường độ mức xám (ví dụ, giống như trong ảnh mức xám), trong khi hai thành phần sắc độ (hoặc ngắn gọn là chroma) Cb và Cr biểu diễn độ màu 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 việc biến đổi hoặc chuyển đổi màu sắc. Nếu ảnh là đơn sắc, 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 nhiều khối ảnh (cụ thể là không chồng lấn) 203. Các khối này có thể cũng được đề cập đến là các khối gốc, các khối cỡ lớn (macro) (H.264/AVC) hoặc các khối cây lập mã (coding tree block, viết tắt là CTB) hoặc các bộ phận cây lập 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 kích thước khối như nhau đối với tất cả các ảnh của chuỗi của chuỗi video và 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 nữa, bộ mã hóa video có thể được tạo cấu hình để thu trực tiếp khối 203 của ảnh 17, ví dụ, một, một vài hoặc tất cả các khối tạo nên ảnh 17. Khối ảnh 203 có thể cũng được đề cập đến là khối ảnh hiện tại hoặc khối ảnh cần được lập mã.

Giống như ảnh 17, khối ảnh 203 cũng được hoặc có thể được coi là mảng hai chiều hoặc ma trận của các mẫu với các trị số cường độ (các trị số mẫu), mặc dù có kích cỡ

nhỏ hơn so với 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 và/hoặc loại bất kỳ khác của các mảng tùy thuộc vào định dạng màu sắc được áp dụng. Số các mẫu theo chiều (trục) ngang và dọc của khối 203 định rõ kích thước của khối 203. Theo đó, khối có thể, ví dụ, mảng $M \times N$ (M cột nhân N hàng) 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 trên cơ sở từng khối, ví dụ, việc mã hóa và dự đoán được thực hiện trên mỗi khối 203.

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân chia và/hoặc mã hóa ảnh nhờ sử dụng các lát (cũng được đề cập đến là các lát video), trong đó ảnh có thể được phân chia thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát (cụ thể là không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU).

Các phương án về bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân chia và/hoặc mã hóa ảnh nhờ sử dụng các nhóm mảnh (cũng được đề cập đến là các nhóm mảnh video) và/hoặc các mảnh (cũng được đề cập đến là các mảnh video), trong đó ảnh có thể được phân chia thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều nhóm mảnh (cụ thể là không chồng lấn), và mỗi nhóm mảnh có thể bao gồm, ví dụ, một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều mảnh, trong đó mỗi mảnh, ví dụ, có thể là ở dạng hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ, các khối hoàn chỉnh hoặc phân mảnh.

Tính phần dư

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

Biến đổi

Bộ phận xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng sự biến đổi, ví dụ, biến đổi cosin 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 có thể cũng được đề cập đến là các hệ số phần 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 định rõ đối với H.265/HEVC. So với biến đổi DCT trực giao, các phép xấp xỉ số nguyên như vậy cụ thể được mở rộng bởi yếu tố nhất định. Để duy trì định chuẩn của khối dư mà được xử lý bởi các việc biến đổi thuận và ngược, các yếu tố mở rộng bổ sung được áp dụng như phần của quy trình biến đổi. Các yếu tố mở rộng cụ thể được chọn lựa dựa vào các ràng buộc nhất định giống như các yếu tố mở rộng là lũy thừa của hai đối với các thao tác dịch vị, 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 yếu tố mở rộng cụ thể, ví dụ, được định rõ 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 yếu tố mở rộng tương ứng đối với việc biến đổi thuận, ví dụ, bởi bộ phận xử lý biến đổi 206, ở bộ mã hóa 20 có thể được định rõ 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 của sự biến đổi hoặc các sự biến đổi, ví dụ, một cách trực tiếp hoặc được mã hóa hoặc được nén qua bộ phận mã hóa entropi 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 sự lượng tử hóa vô hướng hoặc sự lượng tử hóa vector. Các hệ số được lượng tử hóa 209 có thể cũng được đề cập đến là các hệ số biến đổi được lượng tử hóa 209 hoặc các hệ số phần 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 của 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 thời gian lượng tử hóa, trong đó n lớn hơn m . Mức độ lượng tử hóa có thể được chỉnh sửa bằng cách điều chỉnh thông số lượng tử hóa (quantization parameter, viết tắt là QP). Ví dụ đối với sự lượng tử hóa vô hướng, việc mở rộng khác nhau 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 lượng tử hóa mịn hơn, trong khi các kích thước bước lượng tử hóa lớn hơn tương ứng với lượng tử hóa thô hơn. Kích thước bước lượng tử hóa có thể ứng 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 ứng dụng được. Ví dụ, các thông số lượng tử hóa nhỏ có thể tương ứng với 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 lượng tử hóa thô (kích thước bước lượng tử hóa lớn) hoặc ngược lại. Việc lượng tử hóa có thể bao gồm việc chia bởi 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 dựa vào thông số lượng tử hóa nhờ sử dụng phép xấp xỉ điểm cố định của phương trình bao gồm phép chia. Các yếu tố mở rộng bổ sung có thể được đưa vào đối với việc lượng tử hóa và giải lượng tử hóa để phục hồi định chuẩn của khối dư, mà có thể được chỉnh sửa bởi vì việc mở rộng 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 mở rộng của việc lượng tử hóa và biến đổi ngược có thể được kết hợp. Theo cách khác, các bảng lượng tử hóa được tùy biến có thể được sử dụng và được báo hiệu từ bộ mã hóa tới bộ giải mã, ví dụ, trong dòng bit. Việc lượng tử hóa là thao tác tổn thất, trong đó tổn thất tăng lên với việc tăng các kích thước bước lượng tử hóa.

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 lại của sơ đồ lượng tử hóa được áp dụng bởi bộ phận lượng tử hóa 208 dựa vào hoặc 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 có thể cũng được đề cập đến là các hệ số phần dư được giải lượng tử hóa 211 và tương ứng - mặc dù cụ thể là không giống hệt với các hệ số biến đổi do tổn thất 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 ngược (inverse discrete cosine transform, viết tắt là DCT) hoặc biến đổi sin rời rạc ngược (inverse discrete sine transform, viết tắt là DST) hoặc các việc biến đổi ngược khác, để thu nhận khối được khôi phục dư 213 (hoặc các hệ số được giải lượng tử hóa tương ứng 213) trong miền mẫu. Khối được khôi phục dư 213 có thể cũng được đề cập đến là khối biến đổi 213.

Khôi phục

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

Lọc

Bộ phận bộ lọc vòng lặp 220 (hoặc ngắn gọn là “bộ lọc vòng lặp” 220), được tạo cấu hình để lọc khối được khôi phục 215 để thu nhận khối được lọc 221, hoặc nói chung, để lọc các mẫu được khôi phục để thu nhận các mẫu được lọc. Bộ phận bộ lọc vòng lặp, ví dụ, được tạo cấu hình để làm nhẵn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận bộ lọc vòng lặp 220 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc tách khối, bộ lọc dịch vị thích ứng mẫu (sample-adaptive offset, viết tắt là SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc song phương, bộ lọc vòng lặp thích ứng (adaptive loop filter, viết tắt là ALF), làm sắc, các bộ lọc làm nhẵn hoặc các bộ lọc cộng tác, hoặc sự kết hợp bất kỳ của nó. Mặc dù bộ phận bộ lọc vòng lặp 220 được thể hiện trên Fig.2 là trong bộ lọc vòng lặp, theo các cấu hình

khác, bộ phận bộ lọc vòng lặp 220 có thể được thực hiện như bộ lọc vòng lặp sau. Khối được lọc 221 có thể cũng được đề cập đến là khối được khôi phục được lọc 221.

Các phương án về bộ mã hóa video 20 (bộ phận bộ 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 tích ứng 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 cùng các thông số bộ lọc vòng lặp 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 nói chung dữ liệu ảnh tham chiếu, cho việc mã hóa dữ liệu video bởi bộ mã hóa video 20. DPB 230 có thể được tạo nên bởi bất kỳ trong số các loại của của các thiết bị bộ 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 từ trở (magnetoresistive RAM, viết tắt là MRAM), RAM điện trở (resistive RAM, viết tắt là RRAM), hoặc các loại khác của các thiết bị bộ 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 có thể còn được tạo cấu hình để lưu trữ các khối được lọc khác trước đó, ví dụ, các khối được lọc và được khôi phục trước đó 221, của cùng ảnh hiện tại hoặc của các ảnh khác nhau, ví dụ, các ảnh được khôi phục trước đó, và có thể cung cấp các ảnh (và các mẫu hoặc các khối tham chiếu tương ứng) được khôi phục hoàn thiện trước đó, nghĩa là, được giải mã và/hoặc ảnh hiện tại được khôi phục một phần (và các mẫu hoặc các khối tham chiếu tương ứng), ví dụ cho việc dự đoán liên kết. Bộ đệm ảnh được giải mã (DPB) 230 có thể cũng được tạo cấu hình để lưu trữ một hoặc nhiều khối được khôi phục chưa được lọc 215, hoặc nói chung các mẫu được khôi phục chưa được lọc, ví dụ, nếu khối được khôi phục 215 không được lọc bởi

bộ phận bộ lọc vòng lặp 220, hoặc phiên bản được xử lý thâm bất kỳ khác của các mẫu hoặc các khối được khôi phục.

Lựa chọn chế độ (phân chia & dự đoán)

Bộ phận lựa chọn chế độ 260 bao gồm bộ phận phân chia 262, bộ phận dự đoán liên kết 244 và bộ phận dự đoán bên trong 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 khôi phục, ví dụ, các mẫu hoặc các khối được khôi phục được lọc và/hoặc chưa 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 tuyến tính, không được thể hiện).. Dữ liệu ảnh được khôi phục được sử dụng như dữ liệu ảnh tham chiếu cho việc dự đoán, ví dụ, dự đoán liên kết hoặc dự đoán bên trong, 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 phân chia đối với chế độ dự đoán khối hiện tại (bao gồm không phân chia) và chế độ dự đoán (ví dụ, chế độ dự đoán bên trong hoặc liên kết) và tạo ra khối dự đoán 265 tương ứng, mà được sử dụng cho việc tính của khối dư 205 và cho việc khôi phục của khối được khôi phục 215.

Các phương án về bộ phận lựa chọn chế độ 260 có thể được tạo cấu hình để lựa chọn phân chia và chế độ dự đoán (ví dụ, từ các chế độ dự đoán được hỗ trợ bởi hoặc sẵn có đối với bộ phận lựa chọn chế độ 260), mà cung cấp sự so khớp tốt nhất hoặc nói cách khác phần dư nhỏ nhất (phần dư nhỏ nhất nghĩa là việc nén tốt hơn cho việc truyền hoặc lưu trữ), hoặc phí tổn báo hiệu nhỏ nhất (phí tổn báo hiệu nhỏ nhất nghĩa là việc nén tốt hơn cho việc truyền hoặc lưu trữ), hoặc cân nhắc 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 việc tối ưu hóa méo tỷ lệ (rate distortion optimization, viết tắt là RDO), nghĩa là, lựa chọn chế độ dự đoán mà cung cấp sự méo tỷ lệ nhỏ nhất. Các thuật ngữ

như “tốt nhất”, “nhỏ nhất”, “tối ưu” v.v., trong ngữ cảnh này không nhất thiết đề cập đến toàn bộ “tốt nhất”, “nhỏ nhất”, “tối ưu”, v.v., nhưng có thể cũng đề cập đến sự hoàn chỉnh của tiêu chuẩn kết thúc hoặc lựa chọn giống như trị số vượt quá hoặc giảm xuống dưới ngưỡng hoặc các ràng buộc khác có khả năng dẫn đến “lựa chọn dưới mức tối ưu” 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 nó lại tạo nên các khối), ví dụ, sử dụng lặp lại việc phân chia cây tứ phân (quad-tree, viết tắt là QT), phân chia nhị phân (binary tree, viết tắt là BT) hoặc phân chia cây tam phân (triple-tree, viết tắt là TT) hoặc sự kết hợp bất kỳ của nó, và để thực hiện, ví dụ, dự đoán cho mỗi trong số các phần chia khối hoặc các khối con, trong đó lựa chọn chế độ bao gồm sự lựa chọn của cấu trúc cây của khối được phân chia 203 và các chế độ dự đoán được áp dụng tới mỗi trong số các phần chia khối hoặc các khối con.

Trong phần dưới đây, việc phân chia (ví dụ, bởi bộ phận phân chia 260) và việc xử lý dự đoán (bởi bộ phận dự đoán liên kết 244 và bộ phận dự đoán bên trong 254) được thực hiện bởi bộ mã hóa video ví dụ 20 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 có thể cũng được đề cập đến là các khối con) có thể còn được phân chia thậm chí thành các phần chia nhỏ hơn. Điều này cũng được đề cập đến việc phân chia cây hoặc việc phân chia cây phân cấp, trong đó khối gốc, ví dụ, ở mức cây gốc 0 (mức phân cấp 0, độ sâu 0), có thể được phân chia dưới dạng đệ quy, ví dụ, được phân chia thành hai hoặc nhiều hơn hai khối có mức cây thấp hơn tiếp theo, ví dụ, các nút ở mức cây 1 (mức phân cấp 1, độ sâu 1), trong đó các khối này có thể lại được phân chia thành hai hoặc nhiều hơn hai khối có mức thấp hơn tiếp theo, ví dụ, mức cây

2 (mức phân cấp 2, độ sâu 2), v.v., cho đến khi việc phân chia được kết thúc, ví dụ, bởi vì tiêu chuẩn kết thúc được thỏa mãn, ví dụ, độ sâu cây lớn nhất hoặc kích thước khối nhỏ nhất đạt được. Các khối mà không được phân chia thêm cũng được đề cập đến là các khối lá hoặc các nút lá của cây. Cây sử dụng phân chia thành hai phần chia được đề cập đến là cây nhị phân (binary-tree, viết tắt là BT), cây sử dụng phân chia thành ba phần chia được đề cập đến là cây tam phân (ternary-tree, viết tắt là TT), và cây sử dụng phân chia thành bốn phần chia được đề cập đến là cây tứ phân (quad-tree, viết tắt là QT).

Như được nêu trên, thuật ngữ “khối (block)” như được sử dụng ở đây có thể là phần, cụ thể là phần hình vuông hoặc hình chữ nhật, của ảnh. Ví dụ, tham chiếu tới HEVC và VVC, khối có thể là hoặc tương ứng với bộ phận cây lập mã (coding tree unit, viết tắt là CTU), bộ phận lập 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 tới các khối tương ứng, ví dụ, khối cây lập mã (coding tree block, viết tắt là CTB), khối lập 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 lập 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 lập mã nhờ sử dụng ba mặt phẳng màu riêng biệt và các cấu trúc cú pháp được sử dụng để lập mã các mẫu. Do đó, khối cây lập mã (CTB) có thể là khối $N \times N$ của các mẫu đối với một số trị số của N sao cho phép chia của thành phần thành các CTB là việc phân chia. Bộ phận lập mã (CU) có thể là hoặc bao gồm khối lập mã của các mẫu luma, hai khối lập mã tương ứng của các mẫu chroma của ảnh mà có ba mảng mẫu, hoặc khối lập mã của các mẫu của ảnh đơn sắc hoặc ảnh mà được lập mã nhờ sử dụng ba mặt phẳng màu riêng biệt và các cấu trúc cú pháp được sử dụng để lập mã các mẫu. Do đó, khối lập mã (CB) có thể là khối $M \times N$ của các mẫu

đối với một số trị số của M và N sao cho phép chia của CTB thành các khối lập mã là việc phân chia.

Theo các phương án, ví dụ, theo HEVC, bộ phận cây lập mã (CTU) có thể được phân tách thành các CU nhờ sử dụng cấu trúc cây tứ phân được biểu thị là cây lập mã. Việc quyết định xem có lập mã vùng ảnh nhờ sử dụng dự đoán ảnh liên kết (theo thời gian) hoặc ảnh bên trong (theo không gian) được thực hiện ở mức CU hay không. Mỗi CU có thể còn được phân tách thành một, hai hoặc bốn PU theo loại phân tách PU. Bên trong một PU, quy trình dự đoán như nhau được áp dụng và thông tin liên quan 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 (các TU) theo cấu trúc cây tứ phân khác tương tự với cây lập mã đối với CU.

Theo các phương án, ví dụ, theo chuẩn lập mã video gần nhất hiện đang phát triển, mà được đề cập đến là lập mã video đa năng (VVC), việc phân chia cây nhị phân và cây tứ phân (Quad-tree and binary tree, viết tắt là QTBT) được kết hợp là ví dụ được sử dụng để phân chia khối lập mã. Trong cấu trúc khối QTBT, CU có thể có dạng hình vuông hoặc hình chữ nhật. Ví dụ, bộ phận cây lập mã (CTU) trước tiên được phân chia bởi cấu trúc cây tứ phân. Các nút lá cây tứ phân còn được phân chia bởi cấu trúc cây nhị phân hoặc cây tam phân (hoặc ba phần). Các nút lá cây phân chia được gọi là các bộ phận lập mã (các CU), và việc phân nhỏ đó được sử dụng cho việc xử lý biến đổi và dự đoán mà không có bất kỳ việc phân chia nào. Điều này có nghĩa là CU, PU và TU có kích thước khối như nhau trong cấu trúc khối lập mã QTBT. Song song với đó, việc đa phân chia, ví dụ, việc phân chia cây tam phân có thể 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 nêu 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 (ví dụ, được xác định trước). Tập hợp của các chế độ dự đoán có thể bao gồm, ví dụ, các chế độ dự đoán bên trong và/hoặc các chế độ dự đoán liên kết.

Dự đoán bên trong

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

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

Bộ phận dự đoán bên trong 254 (hoặc nói chung là bộ phận lựa chọn chế độ 260) còn được tạo cấu hình để đưa ra các thông số dự đoán bên trong (hoặc nói chung là thông tin chỉ báo về chế độ dự đoán bên trong được lựa chọn cho khối) tới bộ phận mã hóa entropi 270 ở dạng các thành phần cú pháp 266 cho việc bao gồm trong 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 cho việc giải mã.

Dự đoán liên kết

Tập hợp của (hoặc có thể) các chế độ dự đoán liên kết phụ 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 dự đoán liên kết khác, ví dụ, toàn bộ ảnh tham chiếu hay chỉ một phần, ví dụ, vùng cửa sổ tìm kiếm quanh vùng của khối hiện tại, của

ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu trùng khớp nhất, và/hoặc ví dụ, việc nội suy điểm ảnh được áp dụng hay không, ví dụ, nội suy nửa/bán điểm ảnh, một phần tư điểm ảnh và/hoặc 1/16 điểm ảnh.

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

Bộ phận dự đoán liên kết 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 đều 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 (ảnh hiện tại khối 203 của ảnh hiện tại 17) và ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều được khôi phục trước đó, ví dụ, các khối được khôi phục của một hoặc nhiều ảnh được giải mã khác/khác nhau trước đó 231, 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 của 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 nhiều khối tham chiếu của các ảnh như nhau hoặc khác nhau của nhiều ảnh khác và cung cấp ảnh tham chiếu (hoặc chỉ số ảnh tham chiếu) và/hoặc dịch vị (dịch vị theo không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện tại như các thông số dự đoán liên kết tới bộ phận đánh giá chuyển động. Dịch vị này cũng được gọi là vectơ 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 dự đoán liên kết và để thực hiện việc dự đoán liên kết dựa vào hoặc sử dụng thông số dự đoán liên kết để thu nhận khối dự đoán liên kết 265. Việc bù chuyển động, được thực hiện bởi bộ phận bù chuyển động, có thể bao hàm việc truy xuất hoặc tạo ra khối dự đoán dựa vào vectơ khối/chuyển động được xác định nhờ việc đánh giá chuyển động,

có khả năng thực hiện các việc nội suy tới 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, do đó có khả năng làm tăng số các khối dự đoán ứng viên mà có thể được sử dụng để lập mã khối ảnh. Khi thu vector chuyển động đối với PU của khối ảnh hiện tại, bộ phận bù chuyển động có thể định vị khối dự đoán mà trong đó các điểm vector chuyển động trong một trong số các danh mục ảnh tham chiếu.

Bộ phận bù chuyển động có thể cũng tạo ra các thành phần cú pháp được kết hợp với các khối và các lát video để sử dụng bởi bộ giải mã video 30 trong việc giải mã các khối ảnh của lát video. Ngoài ra hoặc như sự thay thế cho các lát và các thành phần cú pháp tương ứng, các nhóm mảnh và/hoặc các mảnh và các thành phần cú pháp tương ứng có thể được tạo ra hoặc được sử dụng.

Lập mã entropi

Bộ phận mã hóa entropi 270 được tạo cấu hình để áp dụng, ví dụ, sơ đồ hoặc thuật toán mã hóa entropi (ví dụ, sơ đồ lập mã độ dài biến đổi (variable length coding, viết tắt là VLC), sơ đồ VLC thích ứng ngữ cảnh (context adaptive VLC, viết tắt là CAVLC), sơ đồ lập mã số học, nhị phân hóa, lập mã số học nhị phân thích ứng ngữ cảnh (context adaptive binary arithmetic coding, viết tắt là CABAC), lập mã số học nhị phân thích ứng ngữ cảnh dựa trên cú pháp (syntax-based context adaptive binary arithmetic coding, viết tắt là SBAC), lập mã entropi phân chia khoảng xác suất (probability interval partitioning entropy, viết tắt là PIPE) hoặc kỹ thuật hoặc phương pháp luận mã hóa entropi khác) hoặc bỏ qua (không nén) trên các hệ số được lượng tử hóa 209, các thông số dự đoán liên kết, các thông số dự đoán bên trong, 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 dòng bit được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể thu và sử dụng các thông số cho việc 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 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 trên cơ sở không biến đổi 20 có thể lượng tử hóa tín hiệu phần dư trực tiếp mà không bộ phận xử lý biến đổi 206 đối với 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 của sáng chế này. 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/hoặc các nhóm mảnh hoặc các mảnh) 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 khôi phục 314 (ví dụ, bộ lấy tổng 314), bộ lọc vòng lặp 320, bộ đệm ảnh được giải mã (DBP) 330, bộ phận ứng dụng chế độ 360, bộ phận dự đoán liên kết 344 và bộ phận dự đoán bên trong 354. Bộ phận dự đoán liên kết 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 tiến trình giải mã nói chung ngược lại với tiến trình mã hóa được mô tả đối với 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 khôi phục 214 bộ lọc vòng lặp 220, bộ đệm ảnh được giải mã (DPB) 230, bộ phận dự đoán liên kết 344 và bộ phận dự đoán bên

trong 354 cũng được đề cập đến như việc tạo nên “bộ giải mã lắp sẵn” của bộ mã hóa video 20. Theo đó, bộ phận lượng tử hóa ngược 310 có thể là giống hệt về chức năng với bộ phận lượng tử hóa ngược 110, bộ phận xử lý biến đổi ngược 312 có thể là giống hệt về chức năng với bộ phận xử lý biến đổi ngược 212, bộ phận khôi phục 314 có thể là giống hệt về chức năng với bộ phận khôi phục 214, bộ lọc vòng lặp 320 có thể là giống hệt về chức năng với bộ lọc vòng lặp 220, và bộ đệm ảnh được giải mã 330 có thể là giống hệt 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 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 phân tách dòng bit 21 (hoặc nói chung là dữ liệu ảnh được mã hóa 21) và thực hiện, ví dụ, 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 lập mã được giải mã (không được thể hiện trên Fig.3), ví dụ, bất kỳ hoặc tất cả trong số các thông số dự đoán liên kết (ví dụ, chỉ số ảnh tham chiếu và vectơ chuyển động), thông số dự đoán bên trong (ví dụ, chỉ số hoặc chế độ dự đoán bên trong), 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 sơ đồ hoặc các thuật toán 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 có thể còn được tạo cấu hình để cung cấp các thông số dự đoán liên kết, thông số dự đoán bên trong và/hoặc các thành phần cú pháp khác tới bộ phận ứng dụng 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. Ngoài ra hoặc như sự thay thế cho các lát và các thành phần cú pháp tương ứng, các

nhóm mảnh và/hoặc các mảnh và các thành phần cú pháp tương ứng có thể được thu và/hoặc được sử dụng.

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 nói chung là thông tin liên quan đến việc lượng tử hóa ngược) và các hệ số được lượng tử hóa từ dữ liệu ảnh được mã hóa 21 (ví dụ, bằng cách phân 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 có thể cũng được đề cập đến là các hệ số biến đổi 311. Quy trình lượng tử hóa ngược có thể bao gồm việc sử dụng thông số lượng tử hóa được xác định bởi bộ mã hóa video 20 đối với mỗi khối video trong lát video (hoặc mảnh hoặc nhóm mảnh) để xác định mức độ lượng tử hóa và, cũng vậy, mức độ 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 đề cập đến là các hệ số biến đổi 311, và để áp dụng sự biến đổi tới các hệ số được giải lượng tử hóa 311 để thu nhận các khối dự được khôi phục 213 trong miền mẫu. Các khối dự được khôi phục 213 có thể cũng được đề cập đến là các khối biến đổi 313. Việc biến đổi có thể là việc biến đổi ngược, ví dụ, DCT ngược, DST ngược, biến đổi số nguyên ngược, hoặc tương tự về mặt khái niệm với quy trình biến đổi ngược. Bộ phận xử lý biến đổi ngược 312 có thể còn 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 phân tách và/hoặc giải mã, ví dụ, bởi bộ phận giải mã entropi 304) để xác định việc biến đổi cần được áp dụng tới các hệ số được giải lượng tử hóa 311.

Khôi phục

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

Lọc

Bộ phận bộ lọc vòng lặp 320 (hoặc trong vòng lặp lặp mã hoặc sau vòng lặp lặp mã) được tạo cấu hình để lọc khối được khôi phục 315 để thu nhận khối được lọc 321, ví dụ, để làm nhẵn các sự chuyển tiếp điểm ảnh, hoặc theo cách khác nâng cao chất lượng video. Bộ phận bộ lọc vòng lặp 320 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp chẳng hạn như bộ lọc tách khối, bộ lọc dịch vị thích ứng mẫu (sample-adaptive offset, viết tắt là SAO) hoặc một hoặc nhiều bộ lọc khác, ví dụ, bộ lọc song phương, bộ lọc vòng lặp thích ứng (ALF), làm sắc, các bộ lọc làm nhẵn hoặc các bộ lọc cộng tác, hoặc sự kết hợp bất kỳ của nó. Mặc dù bộ phận bộ lọc vòng lặp 320 được thể hiện trên Fig.3 là trong bộ lọc vòng lặp, theo các cấu hình khác, bộ phận bộ lọc vòng lặp 320 có thể được thực hiện như bộ lọc vòng lặp sau.

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 đối với 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, cho việc biểu diễn hoặc xem tới người dùng.

Dự đoán

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

Khi lát video được lập mã là lát được lập mã bên trong (I), bộ phận dự đoán bên trong 354 của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo ra khối dự đoán 365 đối với khối ảnh của lát video hiện tại dựa vào chế độ dự đoán bên trong được báo hiệu và dữ liệu từ các khối được giải mã trước đó của ảnh hiện tại. Khi ảnh video được lập mã là lát được lập mã liên kết (nghĩa là, B, hoặc P), bộ phận dự đoán liên kết 344 (ví dụ, bộ phận bù chuyển động) của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo ra các khối dự đoán 365 đối với khối video của lát video hiện tại dựa vào các vector chuyển động và các thành phần cú pháp khác được thu từ bộ phận giải mã entropi 304. Đối với việc dự đoán liên kết, 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 xây dựng các danh mục khung tham chiếu, danh mục 0 (List 0) và danh mục 1 (List 1), nhờ sử dụng các kỹ thuật xây dựng mặc định dựa vào các ảnh tham chiếu được lưu trữ trong DPB 330. Điều giống nhau hoặc tương tự có thể được áp dụng đối với hoặc bởi các phương án sử dụng các nhóm mảnh (ví dụ, các nhóm mảnh video) và/hoặc các mảnh (ví dụ, các mảnh video) bổ sung hoặc thay thế cho các lát (ví dụ, các

lát video), ví dụ, video có thể được lập mã nhờ sử dụng các mảnh và/hoặc các nhóm mảnh I, P hoặc B.

Bộ phận ứng dụng chế độ 360 được tạo cấu hình để xác định thông tin dự đoán đối với khối video của lát video hiện tại bằng cách phân tách các vector chuyển động hoặc thông tin liên quan và các thành phần cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán đối với khối video hiện tại được giải mã. Ví dụ, bộ phận ứng dụng chế độ 360 sử dụng một số trong số các thành phần cú pháp được thu để xác định chế độ dự đoán (ví dụ, việc dự đoán bên trong hoặc liên kết) được sử dụng để lập mã các khối video của lát video, loại lát dự đoán liên kết (ví dụ, lát B, lát P, hoặc lát GPB), thông tin xây dựng cho một hoặc nhiều trong số các danh mục ảnh tham chiếu đối với lát, các vector chuyển động đối với mỗi khối video được mã hóa liên kết của lát, trạng thái dự đoán liên kết đối với mỗi khối video được lập mã liên kết của lát, và thông tin khác để giải mã các khối video trong lát video hiện tại. Điều giống như hoặc tương tự có thể được áp dụng đối với hoặc bởi các phương án sử dụng các nhóm mảnh (ví dụ, các nhóm mảnh video) và/hoặc các mảnh (ví dụ, các mảnh video) bổ sung hoặc thay thế cho các lát (ví dụ, các lát video), ví dụ, video có thể được lập mã nhờ sử dụng các mảnh và/hoặc các nhóm mảnh I, P hoặc B.

Các phương án về bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân chia và/hoặc giải mã ảnh nhờ sử dụng các lát (cũng được đề cập đến là các lát video), trong đó ảnh có thể được phân chia thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát (cụ thể là không chồng lấn), và mỗi lát có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU).

Các phương án về bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân chia và/hoặc giải mã ảnh nhờ sử dụng các nhóm mảnh (cũng được đề cập đến là các nhóm mảnh video) và/hoặc các mảnh (cũng được đề cập đến là các mảnh video), trong đó ảnh có thể được phân chia thành hoặc được giải mã nhờ sử dụng một

hoặc nhiều nhóm mảnh (cụ thể là không chồng lấn), và mỗi nhóm mảnh có thể bao gồm, ví dụ, một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều mảnh, trong đó mỗi mảnh, ví dụ, có thể là có dạng hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ, các khối hoàn chỉnh hoặc phân mảnh.

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 trên cơ sở không biến đổi 30 có thể lượng tử hóa ngược tín hiệu phần dư một cách trực tiếp mà không có bộ phận xử lý biến đổi ngược 312 đối với các khung hoặc các khối 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 trong bộ phận đơn.

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện tại có thể còn được xử lý và sau đó đưa ra tới bước tiếp theo. Ví dụ, sau khi lọc nội suy, dẫn ra vector chuyển động hoặc lọc vòng lặp, thao tác khác, chẳng hạn như ghim (Clip) hoặc dịch vị (shift), có thể được thực hiện trên kết quả xử lý của việc lọc nội suy, dẫn ra vector chuyển động hoặc lọc vòng lặp.

Cần lưu ý rằng các thao tác khác có thể được áp dụng tới các vector chuyển động được dẫn ra của khối hiện tại (bao gồm những không giới hạn ở các vector chuyển động điểm điều khiển của chế độ afin, các vector chuyển động khối con trong các chế độ afin, mặt phẳng, ATMVP, các vector chuyển động theo thời gian, và v.v.). Ví dụ, trị số của vector chuyển động bị ràng buộc tới khoảng được định rõ trước theo bit biểu diễn của nó. Nếu bit biểu diễn của vector chuyển động là bitDepth, thì khoảng là $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)}-1$, trong đó “ $\sim$ ” nghĩa là số mũ. Ví dụ, nếu bitDepth được thiết đặt bằng 16, khoảng là $-32768 \sim 32767$; nếu bitDepth được thiết đặt bằng 18, khoảng là $-131072 \sim 131071$. Ví dụ, trị số của vector 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 phần nguyên của bốn MV khối con 4x4 là không lớn hơn N điểm ảnh, chẳng hạn là không lớn hơn 1 điểm ảnh. Ở đây đưa ra hai phương pháp để ràng buộc vector chuyển động theo bitDepth.

Phương pháp 1: loại bỏ bit quan trọng nhất (most significant bit, viết tắt là MSB) bởi các phép toán sau

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 lược bỏ MV, và việc định rõ của hàm Clip3 là như sau:

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

Fig.4 là hình vẽ giản lược của thiết bị lập mã video 400 theo phương án của sáng chế. Thiết bị lập mã video 400 là thích hợp cho việc thực hiện các phương án được bộc lộ như được mô tả ở đây. Theo một phương án, thiết bị lập 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 lập 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 (transmitter unit, viết tắt là Tx) 440 và các cổng ra 450 (hoặc các cổng ra 450) để truyền dữ liệu; và bộ nhớ 460 để lưu trữ dữ liệu. Thiết bị lập mã video 400 có thể cũng bao gồm các thành phần quang thành điện (optical-to-electrical, viết tắt là OE) và các thành phần điện thành 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 để đưa vào hoặc đưa ra 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 đa 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 lập mã 470. Môđun lập mã 470 thực hiện các phương án được bộc lộ nêu trên. Chẳng hạn, môđun lập mã 470 thực hiện, xử lý, chuẩn bị, hoặc cung cấp các thao tác lập mã khác nhau. Sự bao gồm của môđun lập mã 470 do đó đưa ra sự cải tiến đáng kể đối với chức năng của thiết bị lập mã video 400 và đem lại sự biến đổi của thiết bị lập mã video 400 tới trạng thái khác nhau. Theo cách khác, môđun lập 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, các ổ băng, và các 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 lưu trữ các lệnh và dữ liệu mà được đọc trong khi thực hiện chương trình. Bộ nhớ 460 có thể là, 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 nội dung địa chỉ 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 giản lược của máy 500 mà có thể được sử dụng như một hoặc cả trong số thiết bị nguồn 12 và thiết bị đích 14 từ Fig.1 theo phương án lấy làm ví dụ.

Bộ xử lý 502 trong máy 500 có thể là bộ phận xử lý trung tâm. Theo cách khác, bộ xử lý 502 có thể là loại bất kỳ khác của thiết bị, hoặc nhiều thiết bị, có khả năng thao tác hoặc xử lý thông tin hiện có hoặc được phát triển về sau. Mặc dù các 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, các ưu điểm về tốc độ và hiệu quả có thể là 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 (read only memory, viết tắt là ROM) hoặc thiết bị bộ nhớ truy cập ngẫu nhiên (random access memory, viết tắt là RAM) theo cách thực hiện. Loại thích hợp bất kỳ khác của thiết bị lưu trữ 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 có thể còn 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 các ứng dụng từ 1 đến N, mà còn bao gồm ứng dụng lập mã video mà thực hiện các phương pháp được mô tả ở đây.

Máy 500 có thể cũng 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 cảm ứng chạm (touch sensitive display) mà kết hợp màn hình với thành phần cảm ứng chạm mà hoạt động được để cảm nhận 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, bus 512 của máy 500 có thể được bao gồm nhiều bus. Hơn nữa, bộ phận lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các thành phần khác của máy 500 hoặc có thể được truy cập qua mạng và có thể bao gồm bộ phận tích hợp đơn chẳng hạn như thẻ nhớ hoặc nhiều bộ phận chẳng hạn như nhiều thẻ nhớ. Máy 500 do đó có thể được thực hiện theo nhiều loại cấu hình.

Các tập hợp thông số

Các tập hợp thông số về cơ bản là tương tự và có chung các mục đích thiết kế cơ bản—chẳng hạn như hiệu quả tốc độ bit, khả năng phục hồi lỗi, và cung cấp các giao diện lớp hệ thống. Có sự phân cấp về các tập hợp thông số trong HEVC (ITU-T H.265), bao gồm tập hợp thông số video (Video Parameter Set, viết tắt là VPS), tập hợp thông số chuỗi (Sequence Parameter Set, viết tắt là SPS) và tập hợp thông số ảnh (Picture Parameter Set, viết tắt là PPS) mà là tương tự với các bản sao của chúng trong AVC và

VVC. Mỗi lát tham chiếu PPS, SPS và VPS hoạt động đơn lẻ để truy cập thông tin được sử dụng để giải mã lát. PPS chứa thông tin mà áp dụng tới tất cả các lát trong ảnh, và vì vậy tất cả các lát trong ảnh phải tham chiếu tới cùng PPS. Các lát trong các ảnh khác nhau cũng được phép tham chiếu tới cùng PPS. Tương tự, SPS chứa thông tin mà áp dụng tới tất cả các ảnh trong cùng chuỗi video được lập mã.

Trong khi PPS có thể khác nhau đối với các ảnh riêng biệt, nó là phổ biến đối với nhiều hoặc tất cả các ảnh trong chuỗi video được lập mã để tham chiếu tới cùng PPS. Việc tái sử dụng các tập hợp thông số là hiệu quả tốc độ bit bởi vì nó tránh khỏi việc cần phải gửi thông tin được chia sẻ nhiều lần. Đây cũng là tổn thất lớn bởi vì nó cho phép nội dung tập hợp thông số cần được mang bởi một số liên kết truyền thông bên ngoài tin cậy hơn hoặc cần được chuẩn bị thường xuyên bên trong dòng bit để đảm bảo rằng nó sẽ không bị tổn thất.

Tập hợp thông số chuỗi (SPS)

SPS chứa các thông số mà áp dụng tới một hoặc nhiều lớp của chuỗi của chuỗi video được lập mã và không thay đổi từ ảnh tới ảnh bên trong chuỗi video được lập mã. Cụ thể là, SPS bao gồm thông tin mà báo hiệu của ảnh con.

Một số phần của bảng dưới đây thể hiện sự lưu nhanh của phần của việc báo hiệu ảnh con trong SPS trong ITU JVET-Q2001-v11, với liên kết tải xuống là như sau:

<http://phenix.it->

sudparis.eu/jvet/doc_end_user/documents/17_Brussels/wg11/JVET-Q2001-v11.zip.

Trong phần còn lại của sáng chế, tài liệu này sẽ được đặt tên là phác thảo VCC 8 nhằm đơn giản.

seq_parameter_set_rbsp() {	Ký hiệu mô tả
sps_seq_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)

sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <=	
sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY)	
{	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 &&	
pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 &&	
pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)

loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
...	

Một số thành phần cú pháp trong thông tin vị trí tín hiệu SPS và các cờ điều khiển của mỗi ảnh con. Thông tin vị trí đối với ảnh con thứ i bao gồm:

- subpic_ctu_top_left_x[i], chỉ báo thành phần theo chiều ngang của tọa độ trên cùng bên trái của ảnh con i trong ảnh; hoặc
- subpic_ctu_top_left_y[i] chỉ báo thành phần theo chiều dọc của tọa độ trên cùng bên trái của ảnh con i trong ảnh; hoặc
- subpic_width_minus1[i] chỉ báo độ rộng của ảnh con i trong ảnh; hoặc
- subpic_height_minus1[i] chỉ báo độ cao của ảnh con i trong ảnh.

Một số thành phần cú pháp chỉ báo số các ảnh con trong ảnh, ví dụ, sps_num_subpics_minus1.

Phân chia ảnh thành các CTU, các lát, các mảnh và các ảnh con

Phân chia ảnh thành các CTU

Ảnh được chia thành chuỗi của các bộ phận cây lập mã (CTU). Thuật ngữ CTU đôi khi được sử dụng theo cách hoán đổi như khối cây lập mã (coding tree block, viết tắt là CTB). Theo ví dụ, thuật ngữ CTU là giống như việc định rõ CTU trong ITU-T H.265. Đối với ảnh mà có ba mảng mẫu, CTU bao gồm khối $N \times N$ của các mẫu luma, và hai khối tương ứng của các mẫu chroma. Fig.6 thể hiện ví dụ của ảnh được chia thành các CTU. Kích thước của các CTU có thể là như nhau ngoại trừ đối với các CTU được nằm ở các ranh giới ảnh (trong đó các CTU không hoàn chỉnh có thể có mặt).

Phân chia ảnh thành các mảnh

Theo một số ví dụ, khi các mảnh được cho phép, ảnh được chia thành các nhóm có dạng hình chữ nhật của các CTU được tách biệt bởi các ranh giới theo chiều dọc và/hoặc theo chiều ngang. Các ranh giới mảnh theo chiều dọc và chiều ngang giao với ảnh lần lượt từ trên cùng tới dưới cùng và từ ranh giới ảnh bên trái tới ranh giới ảnh bên phải. Thông tin chỉ báo liên quan đến vị trí của các ranh giới mảnh theo chiều ngang và các ranh giới mảnh theo chiều dọc được lập mã trong dòng bit.

Fig.7 minh họa ví dụ việc phân chia ảnh thành 9 mảnh. Theo ví dụ này, các ranh giới mảnh được đánh dấu bằng các đường đứt nét đậm. Thứ tự quét mảnh trên cơ sở mảnh của các CTU với chín mảnh có các kích thước khác nhau trong ảnh.

Khi có nhiều hơn 1 mảnh chia ảnh theo chiều dọc, thứ tự quét của các CTU được thay đổi đối với thứ tự quét mảnh của các CTU trong ảnh. Các CTU được quét theo quy tắc sau đây:

1. Các mảnh được quét từ bên trái đến bên phải và từ trên cùng xuống dưới cùng theo thứ tự quét mảnh, mà được gọi là thứ tự quét mảnh. Điều này có nghĩa là bắt đầu từ mảnh trên cùng bên trái, tất cả các mảnh trong cùng hàng mảnh được quét từ bên trái đến bên phải. Sau đó bắt đầu với mảnh thứ nhất trong hàng mảnh thứ hai (hàng mảnh mà là một hàng bên dưới), tất cả các mảnh trong hàng mảnh thứ hai được quét từ bên trái đến bên phải. Quy trình được lặp lại cho đến khi tất cả các mảnh được quét.

2. Đối với mảnh, các CTU trong mảnh này được quét theo thứ tự quét mảnh. Đối với mỗi hàng CTU, các CTU được quét từ bên trái đến bên phải và các hàng CTU được quét từ trên cùng xuống dưới cùng. Fig.7 minh họa ví dụ thứ tự quét của các CTU trong các mảnh, các số tương ứng với các CTU chỉ báo thứ tự quét.

Mảnh cung cấp việc phân chia của ảnh theo cách như vậy mà mỗi mảnh có thể được giải mã một cách độc lập từ các mảnh khác của cùng ảnh, trong đó việc giải mã đề cập

đến việc giải mã entropi, phần dư, và dự đoán. Hơn nữa, với các mảnh, có thể phân chia ảnh thành các vùng có kích thước tương tự. Do đó, có thể xử lý các mảnh của ảnh song song với nhau, mà tốt hơn là đối với các môi trường xử lý đa lõi trong đó mỗi lõi xử lý là giống hệt nhau.

Các thuật ngữ thứ tự xử lý và thứ tự quét được sử dụng như sau trong sáng chế:

Quy trình xử lý đề cập đến việc mã hóa hoặc giải mã của các CTU trong bộ mã hóa hoặc bộ giải mã. Thứ tự quét liên quan đến việc lấy chỉ số của phần chia cụ thể trong ảnh. Các CTU được lấy chỉ số theo thứ tự tăng dần trong ảnh theo thứ tự quét được định rõ. Thứ tự quét CTU trong mảnh có nghĩa là cách thức các CTU bên trong mảnh được lấy chỉ số, mà không phải là thứ tự như nhau trong đó chúng được xử lý (nghĩa là, thứ tự xử lý).

Phân chia ảnh thành các lát

Khái niệm lát cung cấp việc phân chia của ảnh theo cách như vậy mà mỗi lát có thể được giải mã một cách độc lập từ các lát khác của cùng ảnh, trong đó việc giải mã đề cập đến việc việc giải mã entropi, phần dư, và dự đoán. Độ chênh lệch tới các mảnh là ở chỗ các lát có thể có các hình dạng tùy ý mà không nhất thiết phải là hình chữ nhật (có thể linh hoạt hơn về các khả năng phân chia), và mục đích của việc phân chia lát là không cho việc xử lý song song nhưng cho việc so khớp kích thước gói trong các môi trường truyền và khả năng phục hồi lỗi.

Lát có thể bao gồm ảnh hoàn chỉnh hoặc bao gồm phần của ảnh. Trong ITU-T H.265, lát bao gồm các CTU liên tiếp của ảnh theo thứ tự xử lý. Lát được xác định theo địa chỉ CTU bắt đầu, địa chỉ CTU bắt đầu được báo hiệu trong đoạn đầu lát hoặc trong tập hợp thông số ảnh hoặc trong bộ phận khác. Theo ví dụ, lát có thể được xác định theo địa chỉ mảnh bắt đầu khi các lát được yêu cầu bao gồm số nguyên của các mảnh.

Trong phác thảo 8 của VVC, lát bao gồm số nguyên của các mảnh hoặc số nguyên của các hàng CTU liên tiếp bên trong mảnh của ảnh. Do đó, ranh giới lát theo chiều dọc cũng là ranh giới mảnh theo chiều dọc. Có thể là ranh giới theo chiều ngang của lát không phải là ranh giới mảnh, các ranh giới CTU theo chiều ngang có thể được bao gồm bên trong mảnh. Theo ví dụ, khi mảnh được phân tách thành nhiều lát hình chữ nhật, mỗi lát bao gồm số nguyên của các hàng CTU hoàn chỉnh liên tiếp bên trong lát.

Theo một số ví dụ, có hai chế độ lát, chế độ lát quét mảnh và chế độ lát hình chữ nhật. Trong chế độ lát quét mảnh, lát bao gồm chuỗi của các mảnh trong việc quét mảnh mảnh của ảnh. Trong chế độ lát hình chữ nhật, lát bao gồm số các mảnh mà tạo nên dưới dạng tập hợp vùng hình chữ nhật của ảnh, hoặc lát bao gồm số các hàng CTU liên tiếp của một mảnh mà tạo nên dưới dạng tập hợp vùng hình chữ nhật của ảnh. Các mảnh bên trong lát hình chữ nhật được quét theo thứ tự quét mảnh mảnh bên trong vùng hình chữ nhật tương ứng với lát đó.

Tất cả các lát của ảnh tạo nên dưới dạng tập hợp toàn bộ ảnh, tất cả các CTU của ảnh được bao gồm trong một trong số các lát của ảnh. Các quy tắc tương tự áp dụng đối với các mảnh và các ảnh con.

Phân chia ảnh thành các ảnh con

Ảnh con có thể là phân chia hình chữ nhật của ảnh. Ảnh con có thể là toàn bộ ảnh hoặc phần của ảnh. Ảnh con là việc phân chia của ảnh theo cách như vậy mà mỗi ảnh con có thể được giải mã một cách độc lập từ các ảnh con khác của toàn bộ chuỗi video. Trong phác thảo VCC 8, khi sự chỉ báo `subpic_treated_as_pic_flag[i]` là đúng (ví dụ, trị số của `subpic_treated_as_pic_flag[i]` là 1) đối với ảnh con *i*, ảnh con *i* có thể được giải mã một cách độc lập từ các ảnh con khác của toàn bộ chuỗi video.

Độ chênh lệch giữa ảnh con và các mảnh hoặc các lát là ở chỗ, các ảnh con đề cập đến chuỗi video có thể được giải mã một cách độc lập. Đối với các mảnh và các lát, việc giải mã độc lập được thực hiện trong ảnh đơn lẻ của chuỗi video.

Trong phác thảo VCC 8, ảnh con bao gồm một hoặc nhiều lát mà bao phủ một cách tập hợp vùng hình chữ nhật của ảnh. Do đó, mỗi ranh giới ảnh con luôn là ranh giới lát, và mỗi ranh giới ảnh con theo chiều dọc luôn là ranh giới mảnh theo chiều dọc.

Fig.8 đưa ra ví dụ về các mảnh, các lát và các ảnh con. Trên Fig.8, ví dụ của ảnh chứa 4 mảnh (2 cột mảnh và 2 hàng mảnh), 4 lát hình chữ nhật, và 3 ảnh con. Ảnh con 2 chứa hai lát, ảnh con 1 và 3 chứa mỗi 1 lát. Ảnh con 1 chứa 2 mảnh, ảnh con 2 và 3 chứa mỗi 1 mảnh. Ảnh con, lát và các ranh giới mảnh đều trùng khớp với các ranh giới CTU. Trong các ranh giới được thể hiện với dịch vị nhỏ với nhau để làm cho chúng có khả năng phân biệt.

Theo một ví dụ như được thể hiện trên Fig.8, ảnh được phân chia thành 216 CTU, 4 mảnh, 4 lát và 3 ảnh con. Trị số của `sps_num_subpics_minus1` là 2, và các thành phần cú pháp liên quan đến vị trí có các trị số sau đây:

Đối với ảnh con 1

- `subpic_ctu_top_left_x[ 0]`, không được báo hiệu nhưng được suy ra là 0;
- `subpic_ctu_top_left_y[ 0 ]`, không được báo hiệu nhưng được suy ra là 0;
- `subpic_width_minus1[ 0 ]`, trị số là 8;
- `subpic_height_minus1[ 0 ]`, trị số là 11.

Đối với ảnh con 2

- `subpic_ctu_top_left_x[ 1]`, trị số là 9;
- `subpic_ctu_top_left_y[ 1 ]`, trị số là 0;
- `subpic_width_minus1[ 1 ]`, trị số là 8;
- `subpic_height_minus1[ 1 ]`, trị số là 5.

Đối với ảnh con 3

- subpic_ctu_top_left_x[2], trị số là 9;
- subpic_ctu_top_left_y[2], trị số là 6;
- subpic_width_minus1[2], không được báo hiệu nhưng được suy ra là 8;
- subpic_height_minus1[2], không được báo hiệu nhưng được suy ra là 5.

Báo hiệu các mảnh

Bảng dưới đây minh họa ví dụ việc báo hiệu của các kích thước mảnh và các tọa độ của các mảnh trong ảnh, mà được bọc lộ trong bảng cú pháp RBSP tập hợp thông số ảnh của phác thảo VCC 8.

no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	

Thông tin phân chia (ví dụ, địa chỉ mảnh và các kích cỡ mảnh) của mảnh thường được lập mã trong tập hợp thông số. Trong ví dụ nêu trên, trị số của sự chỉ báo no_pic_partition_flag được lập mã trong dòng bit, trị số của no_pic_partition_flag được sử dụng để chỉ báo xem ảnh có được phân chia thành các lát và các mảnh hay không.

Nếu `no_pic_partition_flag` là đúng (nghĩa là ảnh không được phân chia thành các lát hoặc các mảnh), được suy ra rằng ảnh được phân chia thành chỉ một lát và chỉ một mảnh, các ranh giới của nó được sắp thẳng với các ranh giới ảnh. Nếu không (`no_pic_partition_flag` là false), thông tin phân chia mảnh được lập mã trong dòng bit.

Trị số của thành phần cú pháp `tile_column_width_minus1[ i ]` chỉ báo độ rộng của các cột mảnh thứ *i*. Trị số của thành phần cú pháp `tile_row_height_minus1[ i ]` chỉ báo độ cao của hàng mảnh thứ *i*.

Độ cao hàng mảnh và các độ rộng cột mảnh có thể hoặc được báo hiệu rõ ràng trong dòng bit hoặc có thể được suy ra theo dòng bit. Các thành phần cú pháp `num_exp_tile_columns_minus1` và `num_exp_tile_rows_minus1` lần lượt chỉ báo số các cột mảnh và các hàng mảnh, các độ rộng và các độ cao của các mảnh này được báo hiệu rõ ràng trong dòng bit. Các độ rộng và các độ cao của các hàng và các cột mảnh khác được suy ra theo chức năng.

Việc lấy chỉ số các mảnh biểu diễn “thứ tự quét mảnh trong ảnh”. Các mảnh trong ảnh được lấy thứ tự (được quét) theo thứ tự quét mảnh, mảnh thứ nhất mà là ở góc trên cùng bên trái của ảnh là mảnh thứ 0, và chỉ số tăng lên từ bên trái đến bên phải trong mỗi hàng mảnh và sau khi mảnh cuối cùng trong hàng mảnh được quét, một mảnh tiếp theo với mảnh ngoài cùng bên trái của hàng mảnh tiếp theo (một hàng mảnh hiện tại bên dưới).

Các CTU trong mảnh (thứ tự quét CTU trong mảnh) được quét theo thứ tự quét mảnh.

Thứ tự quét trên cơ sở mảnh của các CTU trong ảnh chỉ báo rằng thứ tự quét mảnh trong ảnh và thứ tự quét CTU trong mỗi mảnh được theo sau đồng thời. Nói cách khác,

- các mảnh của ảnh được quét theo thứ tự quét mảnh trong ảnh;

- các CTU của một mảnh được quét hoàn toàn trước CTU bất kỳ trong số các CTU trong mảnh khác nhau;
- trong mỗi mảnh, các CTU được quét theo thứ tự quét mảnh (thứ tự quét CTU trong mảnh).

Việc báo hiệu của các lát

Bảng dưới đây minh họa ví dụ việc báo hiệu của các kích thước và các tọa độ của các lát được dạng hình chữ nhật trong ảnh, mà được bộc lộ trong bảng cú pháp RBSP tập hợp thông số ảnh của phác thảo VCC 8.

if(NumTilesInPic > 1)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 && (tile_idx_delta_present_flag tileIdx % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)

<pre> if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileRows] > 1) { </pre>	
<pre> num_exp_slices_in_tile[i] </pre>	ue(v)
<pre> numExpSlicesInTile = num_exp_slices_in_tile[i] </pre>	
<pre> for(j = 0; j < numExpSlicesInTile; j++) </pre>	
<pre> exp_slice_height_in_ctus_minus1[j] </pre>	ue(v)
<pre> i += NumSlicesInTile[i] </pre>	
<pre> } </pre>	
<pre> if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1) </pre>	
<pre> tile_idx_delta[i] </pre>	se(v)
<pre> } </pre>	
<pre> } </pre>	

Trong phác thảo VCC 8, lát chứa một hoặc nhiều mảnh hoàn chỉnh, hoặc mảnh chứa một hoặc nhiều lát hoàn chỉnh. Do đó các kích thước và các tọa độ lát được chỉ báo đối với việc phân chia mảnh. Trong phác thảo VCC 8, việc phân chia mảnh được báo hiệu trong tập hợp thông số ảnh. Lát thông tin phân chia được báo hiệu nhờ sử dụng thông tin ánh xạ mảnh.

Trong bảng nêu trên, trị số của thành phần cú pháp num_slices_in_pic_minus1 chỉ báo số các lát trong ảnh. Trị số của thành phần cú pháp tile_idx_delta[i] chỉ báo độ chênh

lệch giữa các chỉ số mảnh của mảnh thứ nhất của lát thứ $(i+1)$ và lát thứ i . Ví dụ, chỉ số mảnh của mảnh thứ nhất của lát thứ nhất trong ảnh là 0. Nếu chỉ số mảnh của mảnh thứ nhất của lát thứ hai là 5, trị số của `tile_idx_delta[0]` là bằng 5. Trong ngữ cảnh này, chỉ số mảnh có thể được sử dụng như địa chỉ lát, ví dụ, chỉ số của mảnh thứ nhất của lát là địa chỉ bắt đầu của lát.

Trị số của thành phần cú pháp `slice_width_in_tiles_minus1[i]` chỉ báo độ rộng lát thứ i trong ảnh theo số các mảnh, và trị số của thành phần cú pháp `slice_height_in_tiles_minus1[i]` chỉ báo độ cao của lát thứ i trong ảnh, theo số các mảnh (theo ví dụ, như được bộc lộ trong phác thảo VCC 8, `slice_width_in_tiles_minus1[i]` cộng 1 định rõ độ rộng của lát thứ i hình chữ nhật trong các bộ phận của các cột mảnh. `slice_height_in_tiles_minus1[i]` cộng 1 định rõ độ cao của lát thứ i hình chữ nhật trong các bộ phận của các hàng mảnh).

Trong bảng nêu trên, nếu trị số của `slice_width_in_tiles_minus1[i]` và trị số của `slice_height_in_tiles_minus1[i]` đều bằng 0 (chỉ báo rằng lát thứ i có kích cỡ của nhiều nhất một mảnh theo độ cao và một mảnh theo độ rộng), trị số của thành phần cú pháp `num_exp_slices_in_tile[i]` được lập mã trong dòng bit. Trị số thành phần cú pháp `num_exp_slices_in_tile[i]` chỉ báo số các lát trong mảnh.

Theo phác thảo VCC 8, lát có thể chứa nhiều mảnh hoàn chỉnh hoặc mảnh có thể chứa nhiều ảnh hoàn chỉnh. Như được thể hiện trong bảng cú pháp nêu trên, số các mảnh trong lát được chỉ báo bởi `slice_width_in_tiles_minus1[i]` và `slice_height_in_tiles_minus1[i]`. Và nếu số các mảnh trong lát là bằng 1, số các lát trong mảnh được chỉ báo bởi `num_exp_slices_in_tile[i]`. So nếu trị số của `slice_width_in_tiles_minus1[i]` và trị số của `slice_height_in_tiles_minus1[i]` đều bằng 1, kích thước của lát có thể là bằng hoặc nhỏ hơn kích thước của mảnh.

Thành phần cú pháp `single_slice_per_subpic_flag` là đúng (ví dụ, trị số của `single_slice_per_subpic_flag` là bằng 1) chỉ báo rằng có một và chỉ một lát trên mỗi ảnh

con đối với tất cả các ảnh con của ảnh (trong trường hợp này, ảnh con không thể được chia thành nhiều hơn một lát).

Theo một phương pháp báo hiệu thay thế, các địa chỉ bắt đầu lát và các kích thước lát được chỉ báo trong phác thảo VCC 8 là theo các bước sau:

1. Trước hết, thông tin phân chia mảnh được chỉ báo trong dòng bit, trong đó chỉ số (tileIdx) được sử dụng để lấy chỉ số các mảnh trong ảnh (theo thứ tự quét mảnh trong ảnh). Sau bước này, các trị số của các chỉ số, các tọa độ và các kích thước của mỗi mảnh được thu nhận.

2. Số các lát trong ảnh được báo hiệu. Theo một ví dụ, số các lát có thể được chỉ báo bởi trị số của thành phần cú pháp `num_slices_in_pic_minus1`.

3. Đối với lát thứ nhất trong ảnh, độ rộng và độ cao của lát theo số các mảnh được chỉ báo. Địa chỉ bắt đầu của lát thứ nhất không được báo hiệu rõ ràng, và được suy ra là tileIdx 0, mảnh thứ nhất trong ảnh là mảnh thứ nhất trong lát thứ nhất của ảnh.

4. Nếu kích thước của lát thứ nhất là bằng một mảnh theo độ rộng và một mảnh theo độ cao, và nếu có nhiều hơn một hàng CTU trong mảnh, trị số của thành phần cú pháp `num_exp_slices_in_tile[ 0 ]` được báo hiệu, trị số của thành phần cú pháp `num_exp_slices_in_tile[ 0 ]` chỉ báo số lượng các lát mà được bao gồm trong mảnh nêu trên. Trị số của `num_exp_slices_in_tile[ 0 ]` có thể chỉ báo số các lát trong mảnh mà các kích thước của nó được chỉ báo rõ ràng trong dòng bit, trong khi có thể có các lát trong mảnh nêu trên mà số lượng và các kích thước của nó được suy ra dựa vào kích thước của mảnh, và các kích thước của các lát mà các kích thước của nó được báo hiệu rõ ràng. Tổng số các lát trong mảnh do đó có thể được xác định theo `num_exp_slices_in_tile[ 0 ]` và kích thước của mảnh. Theo một ví dụ, số các lát trong mảnh có thể được biểu diễn bởi thông số `numSlicesInTile[0]`.

5. Đối với mỗi trong số lát thứ hai cho đến lát cuối cùng (bao gồm lát thứ hai nhưng không bao gồm lát cuối cùng) trong ảnh, độ rộng và độ cao của lát theo số các mảnh

được chỉ báo rõ ràng. Địa chỉ bắt đầu của lát có thể được chỉ báo rõ ràng bởi trị số của thành phần cú pháp `tile_idx_delta[ i ]`, trong đó `i` là chỉ số của lát. Nếu địa chỉ bắt đầu không được báo hiệu rõ ràng (ví dụ, nếu các lát được báo hiệu theo thứ tự mà làm cho nó có thể suy ra vị trí bắt đầu của lát tiếp theo, nhờ sử dụng vị trí bắt đầu và độ rộng và độ cao của lát hiện tại), thì địa chỉ bắt đầu của lát được suy ra qua chức năng. Với ví dụ, nếu các lát được báo hiệu trong dòng bit theo thứ tự từ bên trái đến bên phải trong ảnh, nếu tọa độ theo chiều ngang của địa chỉ bắt đầu của lát và độ rộng của lát được biết, tọa độ theo chiều ngang của tọa độ bắt đầu của lát tiếp theo được suy ra là tổng của độ rộng của các lát cộng địa chỉ bắt đầu của lát.

6. Nếu kích thước của lát thứ `n` (`n` nằm trong khoảng 2 và trị số các lát trong ảnh trừ 1, tính cả hai) là bằng 1 mảnh theo độ rộng và 1 mảnh theo độ cao, và nếu có nhiều hơn 1 hàng CTU trong mảnh mà được bao gồm bên trong lát thứ `n`, trị số của thành phần cú pháp `num_exp_slices_in_tile[ n ]` được báo hiệu, trị số của thành phần cú pháp `num_exp_slices_in_tile[ n ]` chỉ báo số lượng các lát mà được bao gồm trong mảnh nêu trên.

7. Đối với lát cuối cùng trong ảnh, độ rộng và độ cao của lát không được báo hiệu rõ ràng, nhưng được suy ra theo số các mảnh theo độ rộng ảnh, số các mảnh theo độ cao ảnh và địa chỉ bắt đầu của lát cuối cùng. Địa chỉ bắt đầu của lát cuối cùng có thể được chỉ báo rõ ràng hoặc được suy ra. Sự suy ra của độ rộng và độ cao của lát cuối cùng trong ảnh có thể được thực hiện theo hai phương trình sau đây, mà là từ phần 6.5.1 của phác thảo VCC 8.

$$\text{slice_width_in_tiles_minus1}[i] = \text{NumTileColumns} - 1 - \text{tileX}$$

$$\text{slice_height_in_tiles_minus1}[i] = \text{NumTileRows} - 1 - \text{tileY}$$

`slice_width_in_tiles_minus1[ i ]` cộng 1 định rõ độ rộng của lát thứ `i` hình chữ nhật trong các bộ phận của các cột mảnh.

`slice_height_in_tiles_minus1[ i ]` cộng 1 định rõ độ cao của lát thứ i hình chữ nhật trong các bộ phận của các hàng mảnh.

Trong các bước nêu trên, độ rộng và độ cao của lát cuối cùng không được báo hiệu, vì nó có thể được suy ra nếu địa chỉ bắt đầu của lát được biết.

Các biến `tileX`, `tileY`, `NumTileColumns`, `NumTileRows` trong phương trình nêu trên được giải thích dưới đây.

Phần 6.5.1 của phác thảo VCC 8

6.5.1 Các quy trình quét mảnh CTB, quét mảnh, và quét ảnh con

Đối với các lát hình chữ nhật, danh mục `NumCtusInSlice[ i ]` đối với i nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, định rõ số của CTU trong lát thứ i , danh mục `SliceTopLeftTileIdx[ i ]` đối với i nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, định rõ chỉ số của mảnh trên cùng bên trái của lát, và ma trận `CtbAddrInSlice[ i ][ j ]` đối với i nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, và j nằm trong khoảng từ 0 đến `NumCtusInSlice[ i ] - 1`, định rõ địa chỉ quét mảnh ảnh của CTB thứ j bên trong lát thứ i , được dẫn ra như sau:

```
if( single_slice_per_subpic_flag ) {
    for( i = 0; i <= sps_num_subpics_minus1; i++ )
        NumCtusInSlice[ i ] = 0
    for( i = 0; i < PicSizeInCtbsY; i++ ) {
        sliceIdx = subpic_info_present_flag ? CtbToSubpicIdx[ i ] : 0
        CtbAddrInSlice[ sliceIdx ][ NumCtusInSlice[ sliceIdx ] ] = i
        NumCtusInSlice[ sliceIdx ]++
    }
} else {
    tileIdx = 0
```

```

for( i = 0; i <= num_slices_in_pic_minus1; i++ )
    NumCtusInSlice[ i ] = 0
for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {
    SliceTopLeftTileIdx[ i ] = tileIdx
    tileX = tileIdx % NumTileColumns
    tileY = tileIdx / NumTileColumns
    if( i == num_slices_in_pic_minus1 ) {
        slice_width_in_tiles_minus1[ i ] = NumTileColumns - 1 - tileX
        slice_height_in_tiles_minus1[ i ] = NumTileRows - 1 - tileY
        NumSlicesInTile[ i ] = 1
    }
    if( slice_width_in_tiles_minus1[ i ] == 0 && slice_height_in_tiles_minus1[ i ]
        == 0 ) {
        ctbY = tileRowBd[ tileY ]
        for( j = 0; j < NumSlicesInTile[ i ] - 1; j++ ) {
            AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                           ctbY, ctbY + SliceHeightInCtusMinus1[ i ] + 1 )
            ctbY += SliceHeightInCtusMinus1[ i ] + 1
            i++
        }
        AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ], ctbY,
            tileRowBd[ tileY + 1 ] )
    } else
        for( j = 0; j <= slice_height_in_tiles_minus1[ i ]; j++ )
            for( k = 0; k <= slice_width_in_tiles_minus1[ i ]; k++ )
                AddCtbsToSlice( i, tileColBd[ tileX + k ], tileColBd[ tileX + k + 1 ],

```

```

        tileRowBd[ tileY + j ], tileRowBd[ tileY + j + 1 ] )

    if( tile_idx_delta_present_flag )
        tileIdx += tile_idx_delta[ i ]
    else {
        tileIdx += slice_width_in_tiles_minus1[ i ] + 1
        if( tileIdx % NumTileColumns == 0 )
            tileIdx += slice_height_in_tiles_minus1[ i ] * NumTileColumns
    }
}
}
}

```

Trong đó hàm số AddCtbsToSlice(sliceIdx, startX, stopX, startY, stopY) được định rõ như sau:

```

for( ctbY = startY; ctbY < stopY; ctbY++ )
    for( ctbX = startX; ctbX < stopX; ctbX++ ) {
        CtbAddrInSlice[ sliceIdx ][ NumCtbsInSlice[ sliceIdx ] ]
        =
        ctbY * PicWidthInCtbsY + ctbX
        (30)
        NumCtbsInSlice[ sliceIdx ]++
    }

```

Phần mô tả nêu trên của việc báo hiệu của các địa chỉ bắt đầu lát và các kích thước lát trong ảnh là một ví dụ được bộc lộ trong phác thảo VCC8. Cụ thể là, phần mô tả mô tả trường hợp trong đó các lát được tạo dạng hình chữ nhật được sử dụng, số các lát trong mỗi ảnh con không được chỉ báo là bằng 1 (trị số của thành phần cú pháp single_slice_per_subpic_flag không bằng 1; khi trị số của thành phần cú pháp single_slice_per_subpic_flag là bằng 1, mỗi ảnh con của ảnh được bao gồm một và chỉ một lát), nhiều hơn 1 mảnh có mặt trong ảnh, và số các hàng CTU trong mảnh là lớn hơn 1. Nếu một số trong số các thông số nêu trên được thay đổi, các chế độ khác của

việc báo hiệu của các địa chỉ bắt đầu lát và các kích thước lát có thể được sử dụng. Ví dụ, nếu chỉ có 1 lát trong mỗi ảnh con, độ rộng và các độ cao của các lát không được báo hiệu rõ ràng trong dòng bit, nhưng được suy ra lần lượt là bằng độ rộng và độ cao ảnh con tương ứng.

Mục nhỏ 6.5.1 của phác thảo VCC 8 cũng bộc lộ thứ tự quét của các CTU trong lát i , trong đó i là chỉ số lát. Ma trận $CtbAddrInSlice[i][n]$, mà là đầu ra của mục nhỏ này, định rõ thứ tự quét CTU trong lát i , trong đó n là chỉ số CTU mà là từ 0 đến số các CTU của lát i . Trị số của $CtbAddrInSlice[i][n]$ định rõ địa chỉ CTU thứ n trong lát i .

Thông thường, thứ tự quét CTU trong lát định rõ thứ tự trong đó các CTU của lát được quét. Các bước nêu trên của phần 6.5.1 của phác thảo VCC 8 minh họa ví dụ về cách thức thứ tự quét CTU trong lát được xác định, ví dụ, cách thức ma trận $CtbAddrInSlice[i][n]$ được xây dựng.

Như được bộc lộ trong phác thảo VCC 8, $CtbAddrInSlice[i][n]$ (mà định rõ thứ tự quét CTU trong lát i) cũng định rõ thứ tự xử lý của các CTU trong lát. Cụ thể là, các CTU của ảnh được xử lý theo thứ tự sau đây:

Các lát được lấy chỉ số bởi i và được xử lý tăng dần i , bắt đầu từ i mà là bằng 0 cho đến i nghĩa là bằng số các lát trong ảnh trừ 1. Mỗi CTU n trong lát i được xử lý theo thứ tự tăng dần, bắt đầu từ n mà là bằng 0 cho đến n nghĩa là bằng số các CTU trong lát i trừ 1. Trị số của $CtbAddrInSlice[i][n]$ định rõ địa chỉ CTU thứ n trong lát i .

Như được bộc lộ trong phác thảo VCC 8, phương pháp báo hiệu được sử dụng để chỉ báo số các lát trong mỗi ảnh con của ảnh. Trị số của `single_slice_per_subpic_flag` được sử dụng để chỉ báo xem tất cả các ảnh con của ảnh bao gồm một và chỉ một lát hay không. Nếu trị số của `single_slice_per_subpic_flag` bằng 1, tất cả các ảnh con của ảnh bao gồm một và chỉ một lát. Theo cách khác (trị số của `single_slice_per_subpic_flag` không bằng 1), ảnh con của ảnh có thể bao gồm nhiều hơn 1 lát.

Fig.9 thể hiện các địa chỉ CTU trong ảnh và một lát (lát 5, lát thứ năm trong ảnh) trong ảnh như ví dụ. Theo ví dụ này, các trị số của CtbAddrInSlice là như sau:

$$\text{CtbAddrInSlice}[4][0] = 27$$

$$\text{CtbAddrInSlice}[4][1] = 28$$

$$\text{CtbAddrInSlice}[4][2] = 29$$

$$\text{CtbAddrInSlice}[4][3] = 30$$

$$\text{CtbAddrInSlice}[4][4] = 37$$

$$\text{CtbAddrInSlice}[4][5] = 38$$

$$\text{CtbAddrInSlice}[4][6] = 39$$

$$\text{CtbAddrInSlice}[4][7] = 40.$$

Thứ tự quét đề cập đến việc lấy chỉ số, theo thứ tự trong đó chỉ số được tăng lên, của X trong Y, chẳng hạn như thứ tự quét CTU trong mảnh, trong đó X là CTU và Y là mảnh.

Việc xử lý nghĩa là việc giải mã hoặc mã hóa trong bộ mã hóa hoặc bộ giải mã. Do đó thứ tự xử lý nghĩa là thứ tự trong đó X (ví dụ, CTU) được xử lý trong bộ mã hóa hoặc bộ giải mã.

Trong phác thảo VCC 8, khi có nhiều hơn một mảnh được bao gồm trong ảnh, lát trong ảnh được báo hiệu như sau:

1. Trị số của thành phần cú pháp (sự chỉ báo rõ ràng trong dòng bit hoặc sự suy ra từ dòng bit) được sử dụng để xác định địa chỉ mảnh bắt đầu của lát, theo số các mảnh.

2. Đối với mỗi lát ngoại trừ lát cuối cùng, số lượng các mảnh được bao gồm trong lát được báo hiệu.

- a. Nếu lát chỉ bao gồm 1 mảnh, số lượng các lát được bao gồm trong mảnh được chỉ báo.

3. Đối với lát cuối cùng trong ảnh, số lượng các mảnh được bao gồm trong lát được suy ra nếu lát bao gồm ít nhất 1 mảnh hoàn chỉnh.

Trong phác thảo VCC 8, nếu kích thước của lát cuối cùng trong ảnh là lớn hơn hoặc bằng 1 mảnh (theo độ rộng và độ cao), thì trị số của kích thước của lát cuối cùng được suy ra từ dòng bit và không được lập mã trực tiếp trong dòng bit.

Như được nêu trên, trị số của `slice_width_in_tiles_minus1[ i ]` và trị số của `slice_height_in_tiles_minus1[ i ]` (mà lần lượt chỉ báo độ rộng và độ cao của lát thứ `i` theo số các mảnh) được lập mã trong dòng bit nếu `i` là nhỏ hơn `num_slices_in_pic_minus1` (do vòng lặp for “for(`i` = 0; `i` < `num_slices_in_pic_minus1`; `i`++)”). Trị số của độ rộng và trị số của độ cao của lát không được báo hiệu khi `i` bằng `num_slices_in_pic_minus1`, ví dụ, đối với lát cuối cùng.

Phương án 1

- Xác định xem ảnh được phân chia thành nhiều hơn một mảnh hay không.
- Khi được xác định rằng ảnh được phân chia thành nhiều hơn một mảnh (ví dụ, khi điều kiện `NumTilesInPic > 1` là đúng), thu nhận số lượng các mảnh theo độ rộng ảnh (ví dụ, số các mảnh trong hàng mảnh của ảnh), `NumTileColumns`.
- Thu nhận số lượng các hàng mảnh theo độ cao ảnh (ví dụ, số các mảnh trong cột mảnh của ảnh), `NumTileRows`.
- Thu nhận tọa độ trên cùng bên trái của lát trong chỉ số mảnh (`SliceTopLeftTileIdx[ i ]`). Trị số của chỉ số mảnh định rõ chỉ số trị số của mảnh trong ảnh, trong đó các mảnh được lập thứ tự (được lấy chỉ số) theo thứ tự quét mảnh trong ảnh.
- Thu nhận tọa độ theo chiều ngang của mảnh theo trị số của chỉ số mảnh, được chỉ báo bởi thông số `tileX`.
- Thu nhận tọa độ theo chiều dọc của mảnh theo trị số của chỉ số mảnh, được chỉ báo bởi thông số `tileY`.

- Đối với mỗi lát trong ảnh, trị số của sự chỉ báo đối với độ rộng của lát được lập mã trong dòng bit, nếu trị số độ chênh lệch giữa NumTileColumns và tileX là lớn hơn ngưỡng thứ nhất.

- Đối với mỗi lát trong ảnh, trị số của sự chỉ báo đối với độ cao của lát được lập mã trong dòng bit, nếu trị số độ chênh lệch giữa NumTileRows và tileY là lớn hơn ngưỡng thứ hai.

Ngưỡng thứ nhất và ngưỡng thứ hai có thể là các số nguyên mà lớn hơn 0. Theo một ví dụ cả hai trị số ngưỡng có thể bằng 1. Các ngưỡng chỉ báo ngưỡng kích thước theo số các mảnh theo các chiều ngang (ngưỡng thứ nhất) và chiều dọc (ngưỡng thứ hai). Khi ngưỡng thứ nhất chỉ báo rằng ngưỡng kích thước là bằng 1 mảnh, sự chỉ báo đối với độ rộng lát được bao gồm trong dòng bit nếu độ chênh lệch giữa tileX và số các mảnh theo độ rộng ảnh là lớn hơn 1 mảnh. Lưu ý rằng, tùy thuộc vào quy ước trị số bắt đầu của tileX (trị số của tileX đối với mảnh thứ nhất trong ảnh), ngưỡng có thể khác nhau. Ví dụ nếu bắt đầu trị số của tileX là 0, thì ngưỡng thứ nhất có thể là 1. Nếu bắt đầu trị số của tileX là 1, thì ngưỡng thứ nhất có thể là 0.

Theo một phương án, trị số của độ rộng của lát cuối cùng và trị số của độ cao của lát cuối cùng cũng không được lập mã trong dòng bit, nếu lát cuối cùng bao gồm ít nhất 1 mảnh hoàn chỉnh.

Theo một phương án, trị số của tileX được thu nhận theo phép toán môđun và theo trị số của NumTileColumns. Ví dụ, $\text{tileX} = \text{tileIdx} \% \text{NumTileColumns}$.

Theo một phương án, trị số của tileY được thu nhận theo phép toán chia và theo trị số của NumTileRows. Ví dụ, $\text{tileY} = \text{tileIdx} / \text{NumTileRows}$.

Theo một phương án, điều kiện “lớn hơn ngưỡng a” có thể được thực hiện là “không bằng ngưỡng b”. Cũng có thể thực hiện điều kiện là “lớn hơn hoặc bằng ngưỡng c”. Thông thường, ngưỡng a là bằng ngưỡng b, và ngưỡng thứ ba c là bằng ngưỡng a + 1.

Một cách thực hiện khả thi của sáng chế được mô tả trong bảng 3. Bảng 1, mà là phần của phác thảo VCC 8 được chỉnh sửa.

Trong bảng 3, số hàng 10 và 11 mô tả cách thức thu nhận tileX và tileY. Trị số của SliceTopLeftTileIdx[i] định rõ trị số của chỉ số (chỉ số mảnh) của mảnh trên cùng bên trái trong lát thứ i. Trị số của chỉ số mảnh là chỉ số của mảnh trong ảnh, trong đó các mảnh được lập thứ tự trong ảnh theo thứ tự quét mảnh. Ví dụ được thể hiện trên Fig.7, mảnh bao gồm CTU 0 có chỉ số mảnh 0. Mảnh bao gồm CTU 16 có chỉ số mảnh 1, mảnh bao gồm CTU 24 có chỉ số mảnh 2, mảnh bao gồm CTU 44 có chỉ số mảnh 3, mảnh bao gồm CTU 52 có chỉ số mảnh 4, mảnh bao gồm CTU 56 có chỉ số mảnh 5, và v.v..

Trị số của tileX (tọa độ theo chiều ngang của mảnh trong ảnh) và trị số của tileY (tọa độ theo chiều dọc của mảnh trong ảnh) được thu nhận nhờ sử dụng các phương trình:

$$\text{tileX} = \text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns};$$

$$\text{tileY} = \text{SliceTopLeftTileIdx}[i] / \text{NumTileColumns}.$$

Theo một phương án, trong dòng 12, điều kiện “NumTileColumns – tileX > 1” sau đây được bổ sung tới các điều kiện khác. Nếu điều kiện không được thỏa mãn (NumTileColumns – tileX là không lớn hơn 1), trị số của thành phần cú pháp slice_width_in_tiles_minus1[i] không được lập mã trực tiếp trong dòng bit. Do đó tiết kiệm tốc độ bit và vì vậy hiệu quả nén có thể đạt được.

Tương tự, trong dòng 14, điều kiện “NumTileRows – tileY > 1” sau đây được bổ sung tới các điều kiện khác. Nếu điều kiện không được thỏa mãn (NumTileRows – tileY là không lớn hơn 1), trị số của thành phần cú pháp slice_height_in_tiles_minus1[i] không được lập mã trực tiếp trong dòng bit. Do đó tiết kiệm tốc độ bit và vì vậy hiệu quả nén có thể đạt được.

Theo ví dụ, ngưỡng trị số là 1, mà có thể là khác nhau theo các cách thực hiện khác nhau.

Bảng 3

if(NumTilesInPic > 1)		1
rect_slice_flag	u(1)	2
if(rect_slice_flag)		3
single_slice_per_subpic_flag	u(1)	4
if(rect_slice_flag && !single_slice_per_subpic_flag) {		5
num_slices_in_pic_minus1	ue(v)	6
if(num_slices_in_pic_minus1 > 0)		7
tile_idx_delta_present_flag	u(1)	8
for(i = 0; i < num_slices_in_pic_minus1; i++) {		9
tileX = SliceTopLeftTileIdx[i] % NumTileColumns		10
tileY = SliceTopLeftTileIdx[i] / NumTileRows		11
if(NumTileColumns > 1 && NumTileColumns – tileX > 1)		12
slice_width_in_tiles_minus1[i]	ue(v)	13
if(NumTileRows > 1 && NumTileRows – tileY > 1 && (tile_idx_delta_present_flag tileX == 0))		14
slice_height_in_tiles_minus1[i]	ue(v)	15

<pre> if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) { </pre>		16
<pre> num_exp_slices_in_tile[i] </pre>	ue(v)	17
<pre> numExpSlicesInTile = num_exp_slices_in_tile[i] </pre>		18
<pre> for(j = 0; j < numExpSlicesInTile; j++) </pre>		19
<pre> exp_slice_height_in_ctus_minus1[j] </pre>	ue(v)	20
<pre> i += NumSlicesInTile[i] </pre>		21
<pre> } </pre>		22
<pre> if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1) </pre>		23
<pre> tile_idx_delta[i] </pre>	se(v)	24
<pre> } </pre>		25
<pre> } </pre>		26

Theo một phương án, các trị số của `slice_width_in_tiles_minus1[ i ]` và `slice_height_in_tiles_minus1[ i ]` được suy ra khi chúng không có mặt trong dòng bit.

Dưới đây, cách thực hiện của sáng chế trong phần 6.5.1 của phác thảo VCC 8 được biểu diễn dưới dạng ví dụ.

Theo một ví dụ, các trị số được suy ra là bằng không. Điều này có thể đối với trường hợp được thực hiện trong phần 6.5.1 của phác thảo VCC 8 như sau:

```

if(NumTileColumns – tileX =< 1) {

    slice_width_in_tiles_minus1[ i ] = 0

```

```

    }

    if(NumTileRows – tileY ==< 1) {

        slice_height_in_tiles_minus1[ i ] = 0

    }

```

Trong các phép gán nêu trên, các trị số của slice_width_in_tiles_minus1[i] được thiết đặt bằng 0 khi độ chênh lệch giữa NumTileColumns và tileX là lớn hơn 1, mà là cùng ngưỡng mà xác định sự có mặt của slice_width_in_tiles_minus1[i] trong dòng bit. Nói cách khác, điều kiện “NumTileColumns – tileX <= thr” xác định xem slice_width_in_tiles_minus1[i] có mặt trong dòng bit hay không, và khi slice_width_in_tiles_minus1[i] không có mặt trong dòng bit, trị số của slice_width_in_tiles_minus1[i] được suy ra (được thiết đặt bằng 0). Phép gán tương tự được thực hiện đối với slice_height_in_tiles_minus1[i] khi slice_height_in_tiles_minus1[i] không có mặt trong dòng bit.

Các điều kiện “NumTileRows – tileY ==< 1” và “NumTileColumns – tileX ==< 1” trong phần 6.5.1 của phác thảo VCC 8 dưới đây là để kiểm tra sự có mặt của slice_width_in_tiles_minus1[i] và slice_height_in_tiles_minus1[i] trong dòng bit. Chúng có thể được biểu diễn cũng theo các cách khác, điểm chính là khi slice_width_in_tiles_minus1 và slice_height_in_tiles_minus1 không có mặt trong các hàng 13 và 15 của bảng 3 (điều kiện để xác định sự có mặt trong dòng bit đánh giá là sai), sau đó các trị số của slice_width_in_tiles_minus1 và slice_height_in_tiles_minus1 sẽ được suy ra là 0.

Trị số của slice_height_in_tiles_minus1[i] được suy ra là không chỉ báo độ cao của lát i là bằng 1 mảnh.

Khi trị số của slice_width_in_tiles_minus1[i] được suy ra là không, nó chỉ báo độ rộng của lát i là bằng 1 mảnh.

`slice_height_in_tiles_minus1` và `slice_width_in_tiles_minus1` là các thành phần cú pháp mà chỉ báo kích thước của lát theo mỗi chiều. Theo các phương án của sáng chế, nếu thành phần cú pháp liên quan đến độ rộng hoặc độ cao của lát theo số các mảnh không có mặt trong dòng bit, độ rộng hoặc độ cao của lát được suy ra là 1 mảnh.

Cách thực hiện của sáng chế trong phần 6.5.1 của phác thảo VCC 8

6.5.1 Các quy trình quét mảnh CTB, quét mảnh, và quét ảnh con

Đối với các lát hình chữ nhật, danh mục `NumCtusInSlice[ i ]` đối với `i` nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, định rõ số của CTU trong lát thứ `i`, danh mục `SliceTopLeftTileIdx[ i ]` đối với `i` nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, định rõ chỉ số của mảnh trên cùng bên trái của lát, và ma trận `CtbAddrInSlice[ i ][ j ]` đối với `i` nằm trong khoảng từ 0 đến `num_slices_in_pic_minus1`, và `j` nằm trong khoảng từ 0 đến `NumCtusInSlice[ i ] - 1`, định rõ địa chỉ quét mảnh ảnh của CTB thứ `j` bên trong lát thứ `i`, được dẫn ra như sau:

```
if( single_slice_per_subpic_flag ) {
    for( i = 0; i <= sps_num_subpics_minus1; i++ )
        NumCtusInSlice[ i ] = 0
    for( i = 0; i < PicSizeInCtbsY; i++ ) {
        sliceIdx = subpic_info_present_flag ? CtbToSubpicIdx[ i ] : 0
        CtbAddrInSlice[ sliceIdx ][ NumCtusInSlice[ sliceIdx ] ] = i
        NumCtusInSlice[ sliceIdx ]++
    }
} else {
    tileIdx = 0
    for( i = 0; i <= num_slices_in_pic_minus1; i++ )
        NumCtusInSlice[ i ] = 0
    for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {
```

```

SliceTopLeftTileIdx[ i ] = tileIdx
tileX = tileIdx % NumTileColumns
tileY = tileIdx / NumTileColumns
if( i == num_slices_in_pic_minus1 ) {
    slice_width_in_tiles_minus1[ i ] = NumTileColumns - 1 - tileX
    slice_height_in_tiles_minus1[ i ] = NumTileRows - 1 - tileY
    NumSlicesInTile[ i ] = 1
}

if( NumTileColumns - tileX <= 1 ) {
    slice_width_in_tiles_minus1[ i ] = 0
}

if( NumTileRows - tileY <= 1 ) {
    slice_height_in_tiles_minus1[ i ] = 0
}

if( slice_width_in_tiles_minus1[ i ] == 0 &&
slice_height_in_tiles_minus1[ i ] == 0 ) {
    ctbY = tileRowBd[ tileY ]
    for( j = 0; j < NumSlicesInTile[ i ] - 1; j++ ) {
        AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
            ctbY, ctbY + SliceHeightInCtusMinus1[ i ] + 1 )
        ctbY += SliceHeightInCtusMinus1[ i ] + 1
        i++
    }
    AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ], ctbY,
tileRowBd[ tileY + 1 ] )
} else

```

```

for( j = 0; j <= slice_height_in_tiles_minus1[ i ]; j++ )
    for( k = 0; k <= slice_width_in_tiles_minus1[ i ]; k++ )
        AddCtbsToSlice( i, tileColBd[ tileX + k ], tileColBd[ tileX + k + 1 ],
            tileRowBd[ tileY + j ], tileRowBd[ tileY + j + 1 ] )
if( tile_idx_delta_present_flag )
    tileIdx += tile_idx_delta[ i ]
else {
    tileIdx += slice_width_in_tiles_minus1[ i ] + 1
    if( tileIdx % NumTileColumns == 0 )
        tileIdx += slice_height_in_tiles_minus1[ i ] * NumTileColumns
    }
}
}
}

```

Trong đó hàm số AddCtbsToSlice(sliceIdx, startX, stopX, startY, stopY) được định rõ như sau:

```

for( ctbY = startY; ctbY < stopY; ctbY++ )
    for( ctbX = startX; ctbX < stopX; ctbX++ ) {
        CtbAddrInSlice[ sliceIdx ][ NumCtusInSlice[ sliceIdx ] ]
        =
        ctbY * PicWidthInCtbsY + ctbX
        NumCtusInSlice[ sliceIdx ]++
    }

```

(30)

Như được thể hiện trên Fig.13, phương pháp lập mã được thực hiện bởi thiết bị giải mã được bộc lộ, trong đó phương pháp này bao gồm các bước:

S1301: thu nhận dòng bit đối với ảnh hiện tại.

dòng bit có thể được thu nhận theo mạng không dây hoặc mạng nối dây. Dòng bit có thể được truyền từ trang mạng (website), máy chủ, hoặc nguồn từ xa khác sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn, đường dây thuê bao số (digital subscriber line, viết tắt là DSL), hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, vi sóng, WIFI, Bluetooth, LTE hoặc 5G.

Theo một phương án, dòng bit là chuỗi của các bit, ở dạng dòng bộ phận lớp trừu tượng mạng (network abstraction layer, viết tắt là NAL) hoặc dòng byte, mà tạo nên sự biểu diễn của chuỗi của chuỗi của các bộ phận truy cập (access unit, viết tắt là AU) tạo nên một hoặc nhiều chuỗi video được lập mã (coded video sequence, viết tắt là CVS).

Theo một số phương án, đối với quy trình giải mã, phía bộ giải mã đọc dòng bit và dẫn ra các ảnh được giải mã từ dòng bit; đối với quy trình mã hóa, phía bộ mã hóa tạo ra dòng bit.

Thông thường, dòng bit sẽ bao gồm các thành phần cú pháp mà được tạo nên bởi cấu trúc cú pháp. Thành phần cú pháp: thành phần của dữ liệu được biểu diễn trong dòng bit.

Cấu trúc cú pháp: Không hoặc nhiều hơn các thành phần cú pháp cùng có mặt trong dòng bit theo thứ tự được định rõ.

Theo ví dụ cụ thể, các định dạng dòng bit định rõ mối tương quan giữa dòng bộ phận lớp trừu tượng mạng (network abstraction layer, viết tắt là NAL) và dòng byte, một trong số chúng được đề cập đến là dòng bit.

Dòng bit có thể là trong một trong số hai định dạng: định dạng dòng bộ phận NAL hoặc định dạng dòng byte. Định dạng dòng bộ phận NAL về mặt khái niệm là loại "cơ bản" hơn. Định dạng dòng bộ phận NAL bao gồm chuỗi của các cấu trúc cú pháp được gọi là các bộ phận NAL. Trình tự này được lập thứ tự theo thứ tự giải mã. Có các ràng

buộc được đặt vào thứ tự giải mã (và các nội dung) của các bộ phận NAL trong dòng bộ phận NAL.

Định dạng dòng byte có thể được xây dựng từ định dạng dòng bộ phận NAL bằng cách lập thứ tự các bộ phận NAL theo thứ tự giải mã và tạo tiền tố mỗi bộ phận NAL với tiền tố mã bắt đầu và các byte được tạo trị số không hoặc lớn hơn không để tạo nên dòng của các byte. Định dạng dòng bộ phận NAL có thể được trích từ định dạng dòng byte bằng cách tìm kiếm vị trí của mẫu hình tiền tố mã bắt đầu duy nhất nằm trong dòng này của các byte.

Điều này định rõ mối tương quan giữa nguồn và các ảnh được giải mã mà được đưa ra qua dòng bit.

Nguồn video mà được biểu diễn bởi dòng bit là chuỗi của các ảnh theo thứ tự giải mã.

Nguồn và các ảnh được giải mã đều được bao gồm một hoặc nhiều mảng mẫu:

- Luma (Y) chỉ (đơn sắc).
- Luma và hai chroma (YCbCr hoặc YCgCo).
- Màu xanh lá cây, xanh da trời, đỏ (GBR, cũng được biết đến là RGB).
- Các mảng biểu diễn các việc lấy mẫu màu sắc ba màu cơ bản hoặc đơn sắc không được định rõ khác (ví dụ, YZX, cũng được biết đến là XYZ).

Các biến và các thuật ngữ được kết hợp với các mảng này được đề cập đến là luma (hoặc L hoặc Y) và chroma, trong đó hai mảng chroma được đề cập đến là Cb và Cr; mà không quan tâm đến phương pháp biểu diễn màu sắc thực trong khi sử dụng. Phương pháp biểu diễn màu sắc thực trong khi sử dụng có thể được chỉ báo trong cú pháp mà được định rõ trong các thông số VUI như được định rõ trong ITU-T H.SEI | ISO/IEC 23002-7.

S1302: thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Cần hiểu rằng, số lượng các mảnh trong hàng mảnh có nghĩa là giống như số các cột mảnh.

Theo ví dụ, số lượng các mảnh trong hàng mảnh của ảnh hiện tại được biểu diễn theo biến NumTileColumns, định rõ số các cột mảnh.

Theo ví dụ, số lượng các mảnh trong hàng mảnh của ảnh hiện tại được thu nhận theo các thông số được lập mã trong tập hợp thông số ảnh của dòng bit. Số lượng các mảnh trong hàng mảnh của ảnh hiện tại có thể được thu nhận theo thông số phân tách trực tiếp trong dòng bit hoặc được thu nhận theo việc dẫn ra trị số từ thông số.

S1303: thu nhận trị số của địa chỉ lát của ảnh hiện tại.

Theo ví dụ, trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

Theo ví dụ, địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Theo ví dụ, trị số của địa chỉ lát được biểu diễn theo biến SliceTopLeftTileIdx[i], “i” là chỉ số lát.

SliceTopLeftTileIdx[i] đối với i nằm trong khoảng từ 0 đến pps_num_slices_in_pic_minus1, định rõ chỉ số mảnh của mảnh chứa CTU thứ nhất trong lát. Pps_num_slices_in_pic_minus1 cộng 1 định rõ số các lát hình chữ nhật trong mỗi ảnh đề cập đến tập hợp thông số ảnh, PPS.

Theo ví dụ, trị số của địa chỉ lát được thu nhận theo thông số được lập mã trong tập hợp thông số ảnh của dòng bit. Số lượng các mảnh trong hàng mảnh của ảnh hiện tại có

thể được thu nhận theo thông số phân tách trực tiếp trong dòng bit hoặc được thu nhận theo việc dẫn ra trị số từ thông số.

S1304: thu nhận trị số của tileX theo trị số của địa chỉ lát.

Theo ví dụ, thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm: thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo ví dụ, trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo ví dụ, $\text{tileX} = \text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$.

Theo ví dụ, trị số của tileX chỉ báo trị số của tọa độ theo chiều ngang.

S1305: phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

ngưỡng thứ nhất là trị số nguyên, ngưỡng thứ nhất có thể là 0, 1, 2...

Theo ví dụ, trị số chỉ báo đối với độ rộng lát được biểu diễn là $\text{pps_slice_width_in_tiles_minus1}[i]$. $\text{pps_slice_width_in_tiles_minus1}[i]$ cộng 1 định rõ độ rộng của lát thứ i hình chữ nhật trong các bộ phận của các cột mảnh. Trị số của $\text{pps_slice_width_in_tiles_minus1}[i]$ có thể nằm trong khoảng từ 0 đến $\text{NumTileColumns} - 1$. Khi không có mặt, trị số của $\text{pps_slice_width_in_tiles_minus1}[i]$ được suy ra là bằng 0.

Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của phép toán môđun giữa trị số của địa chỉ lát và số

lượng các mảnh trong hàng mảnh của ảnh hiện tại không bằng ngưỡng thứ nhất”. Vì vậy, trị số của `tileX` ban đầu không được tính nhưng được biểu diễn theo kết quả của phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo các phương án của sáng chế, sơ đồ báo hiệu được bộc lộ, chỉ có ít nhất hai cột mảnh, thì thành phần cú pháp đối với độ rộng lát được báo hiệu. Vì vậy, hiệu quả giải mã và ứng dụng dòng bit đã được nâng cao.

Theo ví dụ, trong đó phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của `tileX` là bằng ngưỡng thứ nhất. Trị số mặc định là trị số nguyên, theo ví dụ, trị số mặc định có thể là 0, hoặc 1.

Theo ví dụ, trong đó trị số mặc định chỉ báo rằng độ rộng của lát bao gồm 1 cột mảnh.

Theo ví dụ, phương pháp còn bao gồm: thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại; thu nhận trị số của `tileY` theo trị số của địa chỉ lát; phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của `tileY` không bằng ngưỡng thứ hai. Ngưỡng thứ hai là trị số nguyên, ngưỡng thứ hai có thể là 0, 1, 2...

Cần hiểu rằng, số lượng các mảnh trong cột mảnh có nghĩa là giống như số các hàng mảnh.

Theo ví dụ, số lượng các mảnh trong cột mảnh của ảnh hiện tại được biểu diễn theo biến `NumTileRows`. Biến `NumTileRows`, định rõ số của hàng mảnh.

Theo ví dụ, trị số chỉ báo đối với độ cao lát được biểu diễn theo `pps_slice_height_in_tiles_minus1[i]`. `pps_slice_height_in_tiles_minus1[i]` cộng 1 định rõ độ cao của lát thứ `i` hình chữ nhật trong các bộ phận của các hàng mảnh khi

`pps_num_exp_slices_in_tile[ i ]` là bằng 0. Trị số của `pps_slice_height_in_tiles_minus1[ i ]` sẽ nằm trong khoảng từ 0 đến `NumTileRows - 1`.

Theo ví dụ, phương pháp còn bao gồm: thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của `tileY` là bằng ngưỡng thứ hai.

Theo ví dụ, trong đó bước thu nhận trị số của `tileY` theo trị số của địa chỉ lát bao gồm: thu nhận trị số của `tileY` theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Theo một cách thực hiện, trong đó trị số của `tileY` được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong cột mảnh của ảnh hiện tại. Ví dụ, `tileY=SliceTopLeftTileIdx[i]/NumTileColumns`.

Theo ví dụ, trong đó trị số của `tileY` chỉ báo trị số của tọa độ theo chiều dọc. Cần hiểu rằng, theo một số ví dụ, điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của `tileY` không bằng ngưỡng thứ hai” là bằng điều kiện “khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong cột mảnh của ảnh hiện tại không bằng ngưỡng thứ hai”. Vì vậy, trị số của `tileY` ban đầu không được tính nhưng được biểu diễn theo kết quả của phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong cột mảnh của ảnh hiện tại.

Theo ví dụ, bảng cú pháp đối với các phương án nêu trên là

<code>for(i = 0; i < pps_num_slices_in_pic_minus1; i++) {</code>	
<code>if(SliceTopLeftTileIdx[i] % NumTileColumns != NumTileColumns - 1)</code>	
<code>pps_slice_width_in_tiles_minus1[i]</code>	<code>ue(v)</code>

$\text{if}(\quad \text{SliceTopLeftTileIdx}[i] / \text{NumTileColumns} \quad \neq \quad$ $\text{NumTileRows} - 1 \quad \&\&$ $(\text{pps_tile_idx_delta_present_flag} \quad $ $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns} \quad == \quad 0))$	
$\text{pps_slice_height_in_tiles_minus1}[i]$	$\text{ue}(v)$

Như được thể hiện trên Fig.14, máy giải mã video 1400 được bộc lộ, máy 1400 bao gồm:

môđun thu nhận 1401, mà được tạo cấu hình để thu nhận dòng bit đối với ảnh hiện tại; và môđun thu nhận 1401 được tạo cấu hình để thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại và thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán 1402, mà được tạo cấu hình để thu nhận trị số của tileX theo trị số của địa chỉ lát;

môđun phân tách 1403, mà được tạo cấu hình để phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

chi tiết đối với mỗi môđun có thể tham chiếu tới các phương án và các cách thực hiện theo các phương án về phương pháp nêu trên.

Các lợi ích của sáng chế

Fig.10 minh họa ví dụ phân chia ảnh thành 96 CTU, 4 mảnh và 8 lát.

Theo giải pháp được bộc lộ trong tài liệu phác thảo VCC 8, các thành phần cú pháp sau đây cần được lập mã trong dòng bit theo bảng 1.

- num_slices_in_pic_minus1, trị số là 0.
- tile_idx_delta_present_flag, trị số là bằng 1

- đối với lát thứ nhất (nghĩa là i chỉ số là bằng 0)
 - `slice_width_in_tiles_minus1[ 0 ]`, trị số là bằng 0
 - `slice_height_in_tiles_minus1[ 0 ]`, trị số là bằng 0
 - `num_exp_slices_in_tile[ 0 ]`, trị số là bằng 2
 - `exp_slice_height_in_ctus_minus1[ 0 ]` trị số là bằng 0, mà chỉ báo độ cao của lát

thứ nhất theo số các hàng CTU.

- đối với lát thứ hai ($i=1$)
 - `exp_slice_height_in_ctus_minus1[ 1 ]` trị số là bằng 1,
 - `tile_idx_delta[ 1 ]`, trị số là bằng 1.
- đối với lát thứ ba ($i=2$)
 - `slice_width_in_tiles_minus1[ 2 ]`, trị số là bằng 0
 - `slice_height_in_tiles_minus1[ 2 ]`, trị số là bằng 0
 - `num_exp_slices_in_tile[ 2 ]`, trị số là bằng 2
 - `exp_slice_height_in_ctus_minus1[ 2 ]` trị số là bằng 0
- đối với lát thứ tư ($i=3$)
 - `exp_slice_height_in_ctus_minus1[ 3 ]` trị số là bằng 1,
 - `tile_idx_delta[ 3 ]`, trị số là bằng 1.
- đối với lát thứ năm ($i=4$)
 - `slice_width_in_tiles_minus1[ 4 ]`, trị số là bằng 0
 - `slice_height_in_tiles_minus1[ 4 ]`, trị số là bằng 0
 - `num_exp_slices_in_tile[ 4 ]`, trị số là bằng 0
 - `tile_idx_delta[ 4 ]`, trị số là bằng 1.
- đối với lát thứ sáu ($i=5$)
 - `slice_width_in_tiles_minus1[ 5 ]`, trị số là bằng 0
 - `slice_height_in_tiles_minus1[ 5 ]`, trị số là bằng 0
 - `num_exp_slices_in_tile[ 5 ]`, trị số là bằng 3

- exp_slice_height_in_ctus_minus1[5] trị số là bằng 1
- đối với lát thứ bảy (i=6)
- exp_slice_height_in_ctus_minus1[6] trị số là bằng 0,
- đối với lát thứ tám (i=7)
- exp_slice_height_in_ctus_minus1[7] trị số là bằng 2,

Theo phương án của sáng chế, như được thể hiện trong bảng 3, các thành phần cú pháp sau đây cần được lập mã trong dòng bit, để mô tả việc phân chia ảnh như nhau trên Fig.10.

- num_slices_in_pic_minus1, trị số là 0.
- tile_idx_delta_present_flag, trị số là bằng 1
- đối với lát thứ nhất (nghĩa là i chỉ số là bằng 0)
- slice_width_in_tiles_minus1[0], trị số là bằng 0
- slice_height_in_tiles_minus1[0], trị số là bằng 0
- num_exp_slices_in_tile[0], trị số là bằng 2
- exp_slice_height_in_ctus_minus1[0] trị số là bằng 0, mà chỉ báo độ cao của lát thứ nhất theo số các hàng CTU.

- đối với lát thứ hai (i=1)
- exp_slice_height_in_ctus_minus1[1] trị số là bằng 1,
- tile_idx_delta[1], trị số là bằng 1.
- đối với lát thứ ba (i=2)
- slice_height_in_tiles_minus1[2], trị số là bằng 0
- num_exp_slices_in_tile[2], trị số là bằng 2
- exp_slice_height_in_ctus_minus1[2] trị số là bằng 0
- đối với lát thứ tư (i=3)
- exp_slice_height_in_ctus_minus1[3] trị số là bằng 1,
- tile_idx_delta[3], trị số là bằng 1.

- đối với lát thứ năm ($i=4$)
 - `slice_width_in_tiles_minus1[ 4 ]`, trị số là bằng 0
 - `num_exp_slices_in_tile[ 4 ]`, trị số là bằng 0
 - `tile_idx_delta[ 4 ]`, trị số là bằng 1.
- đối với lát thứ sáu ($i=5$)
 - `num_exp_slices_in_tile[ 5 ]`, trị số là bằng 3
 - `exp_slice_height_in_ctus_minus1[ 5 ]` trị số là bằng 1
- đối với lát thứ bảy ($i=6$)
 - `exp_slice_height_in_ctus_minus1[ 6 ]` trị số là bằng 0,
- đối với lát thứ tám ($i=7$)
 - `exp_slice_height_in_ctus_minus1[ 7 ]` trị số là bằng 2,

Tóm lại, các thành phần cú pháp sau đây không cần được lập mã trong dòng bit:

- `slice_width_in_tiles_minus1[ 2 ]`, trị số là bằng 0
- `slice_height_in_tiles_minus1[ 4 ]`, trị số là bằng 0
- `slice_width_in_tiles_minus1[ 5 ]`, trị số là bằng 0
- `slice_height_in_tiles_minus1[ 5 ]`, trị số là bằng 0

Các trị số của các thành phần cú pháp đó được suy ra là bằng không theo các điều kiện sau đây mà được bổ sung tới phần 6.5.1 của phác thảo VCC 8.

```

if( NumTileColumns - 1 - tileX <= 1 ) {

    slice_width_in_tiles_minus1[ i ] = 0

}

if( NumTileRows - 1 - tileY <= 1 ) {

    slice_height_in_tiles_minus1[ i ] = 0

}

```

if(NumTilesInPic > 1)		1
rect_slice_flag	u(1)	2
if(rect_slice_flag)		3
single_slice_per_subpic_flag	u(1)	4
if(rect_slice_flag && !single_slice_per_subpic_flag) {		5
num_slices_in_pic_minus1	ue(v)	6
if(num_slices_in_pic_minus1 > 0)		7
tile_idx_delta_present_flag	u(1)	8
for(i = 0; i < num_slices_in_pic_minus1; i++) {		9
tileX = SliceTopLeftTileIdx[i] % NumTileColumns		10
tileY = SliceTopLeftTileIdx[i] / NumTileColumns		11
if(NumTileColumns > 1 && NumTileColumns – tileX > 1)		12
slice_width_in_tiles_minus1[i]	ue(v)	13
if(NumTileRows > 1 && NumTileRows – tileY > 1 && (tile_idx_delta_present_flag tileX == 0))		14
slice_height_in_tiles_minus1[i]	ue(v)	15
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {		16
num_exp_slices_in_tile[i]	ue(v)	17
numExpSlicesInTile = num_exp_slices_in_tile[i]		18

for(j = 0; j < numExpSlicesInTile; j++)		19
exp_slice_height_in_ctus_minus1[j]	ue(v)	20
i += NumSlicesInTile[i]		21
}		22
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)		23
tile_idx_delta[i]	se(v)	24
}		25
}		26

Phần dưới đây là sự 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.

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

thu nhận dòng bit đối với ảnh (hoặc khung) hiện tại;

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns);

thu nhận trị số của địa chỉ lát hiện tại (ví dụ, SliceTopLeftTileIdx[i]), lát hiện tại được bao gồm trong ảnh hiện tại;

thu nhận trị số của tọa độ theo chiều ngang (theo ví dụ, tọa độ theo chiều ngang nghĩa là tọa độ được nằm ở trục tọa độ theo chiều ngang) (ví dụ, tileX) theo trị số của địa chỉ lát hiện tại (ví dụ, SliceTopLeftTileIdx[i]);

Phân tách trị số chỉ báo đối với độ rộng lát (nghĩa là, độ rộng của lát) từ dòng bit đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ rộng lát được lập mã trong dòng bit), khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns) và trị số của tọa độ theo chiều ngang (ví dụ, tileX) là lớn hơn ngưỡng thứ nhất (ví dụ, ngưỡng thứ nhất là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 2. Phương pháp theo ví dụ 1, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định (ví dụ, trị số mặc định có thể là bằng 0), khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns) và trị số của tọa độ theo chiều ngang (ví dụ, tileX) là nhỏ hơn hoặc bằng ngưỡng thứ nhất.

Ví dụ 3. Phương pháp theo ví dụ 2, trong đó trị số mặc định chỉ báo rằng độ rộng của lát hiện tại bao gồm 1 các cột mảnh.

Ví dụ 4. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 3, trong đó bước thu nhận trị số của tọa độ theo chiều ngang theo trị số của địa chỉ lát hiện tại bao gồm:

thu nhận trị số của tọa độ theo chiều ngang theo trị số của địa chỉ lát hiện tại và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Ví dụ 5. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 4, trong đó địa chỉ lát hiện tại là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát hiện tại, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Ví dụ 6. Phương pháp theo ví dụ 4 hoặc 5, trong đó trị số của tọa độ theo chiều ngang được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát hiện tại và số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$).

Ví dụ 7. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 6, trong đó trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`) định rõ địa chỉ của tọa độ trên cùng bên trái của lát hiện tại trong chỉ số mảnh.

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

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`);

thu nhận trị số của tọa độ theo chiều dọc (theo ví dụ, tọa độ theo chiều dọc nghĩa là tọa độ được nằm ở trục tọa độ theo chiều dọc) (ví dụ, `tileY`) theo trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`);

phân tách trị số chỉ báo đối với độ cao lát (nghĩa là, độ cao của lát) từ dòng bit đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ cao lát được lập mã trong dòng bit), khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`) và trị số của tọa độ theo chiều dọc (ví dụ, `tileY`) là lớn hơn ngưỡng thứ hai (ví dụ, ngưỡng thứ hai là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 9. Phương pháp theo ví dụ 8, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước (ví dụ, trị số được thiết đặt trước có thể là bằng 0), khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`) và trị số của tọa độ theo chiều dọc (ví dụ, `tileY`) là nhỏ hơn hoặc bằng ngưỡng thứ hai.

Ví dụ 10. Phương pháp theo ví dụ 8 hoặc 9, trong đó bước thu nhận trị số của tọa độ theo chiều dọc theo trị số của địa chỉ lát hiện tại bao gồm:

thu nhận trị số của tọa độ theo chiều dọc theo trị số của địa chỉ lát hiện tại và số lượng các mảnh trong cột mảnh của ảnh hiện tại.

Ví dụ 11. Phương pháp theo ví dụ 10, trong đó trị số của tọa độ theo chiều dọc được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát hiện tại và số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i] / \text{NumTileRows}$).

Ví dụ 12. Phương pháp lập mã được thực hiện bởi thiết bị giải mã, trong đó phương pháp này bao gồm các bước:

thu nhận dòng bit đối với ảnh (hoặc khung) hiện tại;

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của NumTileRows);

thu nhận trị số của địa chỉ lát hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i]$), lát hiện tại được bao gồm trong ảnh hiện tại;

thu nhận trị số của tọa độ theo chiều dọc (theo ví dụ, tọa độ theo chiều dọc nghĩa là tọa độ được nằm ở trục tọa độ theo chiều dọc) (ví dụ, tileY) theo trị số của địa chỉ lát hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i]$);

phân tách trị số chỉ báo đối với độ cao lát (nghĩa là, độ cao của lát) từ dòng bit đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ cao lát được lập mã trong dòng bit), khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của NumTileRows) và trị số của tọa độ theo chiều dọc (ví dụ, tileY) là lớn hơn ngưỡng thứ hai (ví dụ, ngưỡng thứ hai là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 13. Phương pháp theo ví dụ 12, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước (ví dụ, trị số được thiết đặt trước có thể là bằng 0), khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của NumTileRows) và trị số của tọa độ theo chiều dọc (ví dụ, tileY) là nhỏ hơn hoặc bằng ngưỡng thứ hai.

Ví dụ 14. Phương pháp theo ví dụ 13, trong đó

trị số mặc định chỉ báo rằng độ cao của lát hiện tại bao gồm 1 các hàng mảnh.

Ví dụ 15. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 12 đến 14, trong đó bước thu nhận trị số của tọa độ theo chiều dọc theo trị số của địa chỉ lát hiện tại bao gồm:

thu nhận trị số của tọa độ theo chiều dọc theo trị số của địa chỉ lát hiện tại và số lượng các mảnh trong cột mảnh của ảnh hiện tại.

Ví dụ 16. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 12 đến 15, trong đó địa chỉ lát hiện tại là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát hiện tại, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

Ví dụ 17. Phương pháp theo ví dụ 15 hoặc 16, trong đó trị số của tọa độ theo chiều dọc được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát hiện tại và số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i] / \text{NumTileRows}$).

Ví dụ 18. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 12 đến 17, trong đó trị số của địa chỉ lát hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i]$) định rõ địa chỉ của tọa độ trên cùng bên trái của lát hiện tại trong chỉ số mảnh.

Ví dụ 19. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 12 đến 18, trong đó phương pháp này còn bao gồm các bước:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns);

thu nhận trị số của tọa độ theo chiều ngang (theo ví dụ, tọa độ theo chiều ngang nghĩa là tọa độ được nằm ở trục tọa độ theo chiều ngang) (ví dụ, tileX) theo trị số của địa chỉ lát hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i]$);

phân tách trị số chỉ báo đối với độ rộng lát (nghĩa là, độ rộng của lát) từ dòng bit đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ rộng lát được lập mã trong dòng bit), khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns) và trị số của tọa độ theo chiều ngang (ví dụ, tileX) là lớn hơn ngưỡng thứ nhất (ví dụ, ngưỡng thứ nhất là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 20. Phương pháp theo ví dụ 19, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định (ví dụ, trị số mặc định có thể là bằng 0), khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns) và trị số của tọa độ theo chiều ngang (ví dụ, tileX) là nhỏ hơn hoặc bằng ngưỡng thứ nhất.

Ví dụ 21. Phương pháp theo ví dụ 19 hoặc 20, trong đó bước thu nhận trị số của tọa độ theo chiều ngang theo trị số của địa chỉ lát hiện tại bao gồm:

thu nhận trị số của tọa độ theo chiều ngang theo trị số của địa chỉ lát hiện tại và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

Ví dụ 22. Phương pháp theo ví dụ 21, trong đó trị số của tọa độ theo chiều ngang được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát hiện tại và số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i] \% \text{NumTileColumns}$).

Ví dụ 23. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của NumTileColumns);

thu nhận trị số của địa chỉ lát hiện tại (ví dụ, $\text{SliceTopLeftTileIdx}[i]$), lát hiện tại được bao gồm trong ảnh hiện tại;

thu nhận trị số của tọa độ theo chiều ngang (theo ví dụ, tọa độ theo chiều ngang nghĩa là tọa độ được nằm ở trục tọa độ theo chiều ngang) (ví dụ, `tileX`) theo trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`);

lập mã trị số chỉ báo đối với độ rộng lát (nghĩa là, độ rộng của lát) đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ rộng lát được lập mã trong dòng bit) để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileColumns`) và trị số của tọa độ theo chiều ngang (ví dụ, `tileX`) là lớn hơn ngưỡng thứ nhất (ví dụ, ngưỡng thứ nhất là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 24. Phương pháp theo ví dụ 23, trong đó phương pháp này còn bao gồm các bước:

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`);

thu nhận trị số của tọa độ theo chiều dọc (theo ví dụ, tọa độ theo chiều dọc nghĩa là tọa độ được nằm ở trục tọa độ theo chiều dọc) (ví dụ, `tileY`) theo trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`);

lập mã trị số chỉ báo đối với độ cao lát (nghĩa là, độ cao của lát) đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ cao lát được lập mã trong dòng bit) để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`) và trị số của tọa độ theo chiều dọc (ví dụ, `tileY`) là lớn hơn ngưỡng thứ hai (ví dụ, ngưỡng thứ hai là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 25. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`);

thu nhận trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`), lát hiện tại được bao gồm trong ảnh hiện tại;

thu nhận trị số của tọa độ theo chiều dọc (theo ví dụ, tọa độ theo chiều dọc nghĩa là tọa độ được nằm ở trục tọa độ theo chiều dọc) (ví dụ, `tileY`) theo trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`);

lập mã trị số chỉ báo đối với độ cao lát (nghĩa là, độ cao của lát) đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ cao lát được lập mã trong dòng bit) để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileRows`) và trị số của tọa độ theo chiều dọc (ví dụ, `tileY`) là lớn hơn ngưỡng thứ hai (ví dụ, ngưỡng thứ hai là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 26. Phương pháp theo ví dụ 25, trong đó phương pháp này còn bao gồm các bước:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileColumns`);

thu nhận trị số của tọa độ theo chiều ngang (theo ví dụ, tọa độ theo chiều ngang nghĩa là tọa độ được nằm ở trục tọa độ theo chiều ngang) (ví dụ, `tileX`) theo trị số của địa chỉ lát hiện tại (ví dụ, `SliceTopLeftTileIdx[ i ]`);

lập mã trị số chỉ báo đối với độ rộng lát (nghĩa là, độ rộng của lát) đối với ảnh hiện tại (nghĩa là, trị số chỉ báo đối với độ rộng lát được lập mã trong dòng bit) để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại (ví dụ, trị số của `NumTileColumns`) và trị số của tọa độ theo chiều ngang (ví dụ, `tileX`) là lớn hơn ngưỡng thứ nhất (ví dụ, ngưỡng thứ nhất là số nguyên, có thể là 0, 1, 2... và v.v.).

Ví dụ 27. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1 đến 22.

Ví dụ 28. Bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 23 đến 26.

Ví dụ 29. Sản phẩm chương trình máy tính bao gồm mã chương trình để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ nêu trên khi được thực hiện trên máy tính hoặc bộ xử lý.

Ví dụ 30. Bộ giải mã, trong đó bộ giải mã này 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 cho việc 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 ví dụ bất kỳ trong số các ví dụ nêu trên.

Ví dụ 31. Bộ mã hóa, trong đó bộ mã hóa này 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 cho việc 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 ví dụ bất kỳ trong số các ví dụ nêu trên.

Ví dụ 32. 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 ví dụ bất kỳ trong số các ví dụ nêu trên.

Fig.8 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 được nêu trên. Liên kết truyền thông 3104 bao gồm nhưng không giới hạn ở WIFI, Ethernet, Cable, (3G/4G/5G) không dây, USB, hoặc loại bất kỳ của sự kết hợp của nó, 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 trong 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 máy tính bảng (Pad), máy tính hoặc máy tính xách tay (laptop), hệ thống hội nghị video, PDA, thiết bị 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, âm thanh), 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 audio và video được mã hóa và 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 riêng 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 (laptop) 3110, bộ ghi video mạng (network video recorder, viết tắt là NVR)/bộ ghi video số (digital video recorder, viết tắt là DVR) 3112, TV 3114, thiết bị giải mã tín hiệu truyền hình (set top box, viết tắt là STB) 3116, hệ thống hội nghị video 3118, hệ

thống giám sát video 3120, thiết bị hỗ trợ số cá nhân (personal digital assistant, viết tắt là PDA) 3122, thiết bị 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 ưu tiên để thực hiện 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 ưu tiên để thực hiện việc xử lý giải mã audio.

Đối với thiết bị đầu cuối có màn hình của nó, ví dụ, điện thoại thông minh hoặc Pad 3108, máy tính hoặc máy tính xách tay (laptop) 3110, bộ ghi video mạng (NVR)/ bộ ghi video số (DVR) 3112, TV 3114, thiết bị hỗ trợ số cá nhân (PDA) 3122, hoặc thiết bị lắp trên phương tiện giao thông 3124, thiết bị đầu cuối có thể cấp dữ liệu được giải mã tới màn hình của nó. Đối với thiết bị đầu cuối không được trang bị 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 trong 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.9 là hình vẽ 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 tiền hành 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 truyền tải siêu văn bản (Hyper Text Transfer Protocol, viết tắt là HTTP), giao thức tạo dòng trực tiếp HTTP (HTTP Live streaming protocol, 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 bất kỳ của sự kết hợp của nó, hoặc tương tự.

Sau khi bộ phận tiến hành 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 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 tình huống này, dữ liệu được mã hóa được truyền tới bộ giải mã video 3206 và bộ giải mã audio 3208 mà không qua bộ phận giải đa hợp 3204.

Nhờ 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 cấp dữ liệu này tới bộ phận đồng bộ hóa 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo ra khung audio, và cấp dữ liệu này tới bộ phận đồng bộ hóa 3212. Theo cách khác, khung video có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.9) trước khi cấp nó tới bộ phận đồng bộ hóa 3212. Tương tự, khung audio có thể lưu trữ trong bộ đệm (không được thể hiện trên Fig.9) trước khi cấp nó tới bộ phận đồng bộ hóa 3212.

Bộ phận đồng bộ hóa 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 hóa 3212 đồng bộ hóa sự biểu diễn của thông tin video và audio. Thông tin có thể lập mã theo cú pháp nhờ sử dụng các dấu thời gian liên quan đến sự biểu diễn của audio được lập mã và dữ liệu trực quan và các dấu thời gian liên quan đến sự phân phối của chính dòng dữ liệu.

Nếu phụ đề được bao gồm trong dòng, phụ đề bộ giải mã 3210 giải mã phụ đề, và dòng 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 thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh theo các phương án nêu trên có thể được kết hợp trong hệ thống khác, ví dụ, hệ thống xe hơi.

Các toán tử toán học

Các toán tử toán học được sử dụng trong đơn này là tương tự với các toán tử toán học được sử dụng trong ngôn ngữ sự lập trình C. Tuy nhiên, các kết quả của phép chia số nguyên và các thao tác dịch vị số học được định rõ một cách chính xác hơn, và các thao tác bổ sung được định rõ, chẳng hạn như lũy thừa và phép chia trị số thực. Các quy ước đếm và đánh số thường bắt đầu từ 0, ví dụ, "thứ nhất" là tương đương với thứ 0, "thứ hai" tương đương với thứ 1, v.v..

Các toán tử số học

Các toán tử số học sau đây được định rõ như sau:

+ Cộng

− Trừ (như toán tử hai đối số) hoặc phủ định (như toán tử tiền số một ngôi)

* Nhân, bao gồm nhân ma trận

x^y Lũy thừa. Định rõ x có lũy thừa là y . Trong các ngữ cảnh khác, ký hiệu như vậy được sử dụng để viết lên trên không nhằm diễn giải là lũy thừa.

/ Chia số nguyên với sự cắt bớt của kết quả về phía không. Ví dụ, $7 / 4$ và $-7 / 4$ được cắt bớt tới 1 và $-7 / 4$ và $7 / -4$ được cắt bớt tới -1 .

$\div$ Được sử dụng để biểu thị phép chia trong các phương trình toán học trong đó việc cắt bớt hoặc làm tròn không được dự định.

$\frac{x}{y}$ Được sử dụng để biểu thị phép chia trong các phương trình toán học trong đó việc cắt bớt hoặc làm tròn không được dự định.

$\sum_{i=x}^y f(i)$ Tổng của $f(i)$ với i lấy tất cả các trị số nguyên từ x lên đến và bao gồm y .

$x \% y$ Môđun. Phần còn lại của x được chia cho y , được định rõ chỉ cho các số nguyên x và y với $x \geq 0$ và $y > 0$.

Các toán tử logic

Các toán tử logic sau đây được định rõ như sau:

$x \&\& y$ Logic Boolean "và (and)" của x và y

$x || y$ Logic Boolean "hoặc (or)" của x và y

$!$ Logic Boolean "không (not)"

$x ? y : z$ Nếu x là Đúng (TRUE) hoặc không bằng 0, đánh giá tới trị số của y ; mặt khác, đánh giá tới trị số của z .

Các toán tử quan hệ

Các toán tử quan hệ sau đây được định rõ như sau:

$>$ Lớn hơn

$>=$ Lớn hơn hoặc bằng

$<$ Nhỏ hơn

$<=$ Nhỏ hơn hoặc bằng

$==$ Bằng

$!=$ Không bằng

Khi toán tử quan hệ được áp dụng tới thành phần cú pháp hoặc biến mà đã được gán

trị số "na" (không áp dụng được), trị số "na" được xử lý như trị số khác biệt đối với thành phần cú pháp hoặc biến. Trị số "na" được coi là không bằng trị số bất kỳ khác.

Các toán tử thao tác bit (Bit-wise operator)

Các toán tử thao tác bit sau đây được định rõ như sau:

& Thao tác bit "và (and)". Khi thao tác trên các đối số nguyên, thao tác trên sự biểu diễn phần bù của hai của trị số nguyên. Khi thao tác trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

| Thao tác bit "hoặc (or)". Khi thao tác trên các đối số nguyên, thao tác trên sự biểu diễn phần bù của hai của trị số nguyên. Khi thao tác trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

^ Thao tác bit "hoặc loại trừ (exclusive or)". Khi thao tác trên các đối số nguyên, thao tác trên sự biểu diễn phần bù của hai của trị số nguyên. Khi thao tác trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit quan trọng hơn bằng 0.

x >> y Dịch vị phải số học của sự biểu diễn số nguyên phần bù của hai của x bởi y các chữ số nhị phân. Hàm này được định rõ chỉ đối với các trị số nguyên không âm của y. Các bit được dịch vị thành các bit quan trọng nhất (most significant bit, viết tắt là MSB) như kết quả của dịch vị phải có trị số bằng MSB của x trước thao tác dịch vị.

x << y Dịch vị trái số học của sự biểu diễn số nguyên phần bù của hai của x bởi y các chữ số nhị phân. Hàm này được định rõ chỉ đối với các trị số nguyên không âm của y. Các bit được dịch vị thành các bit ít quan trọng nhất (least significant bit, viết tắt là LSB) như kết quả của dịch vị trái có trị số bằng 0.

Các toán tử gán

Các toán tử số học sau đây được định rõ như sau:

= Toán tử gán

++ Tăng, nghĩa là, $x++$ là tương đương với $x = x + 1$; khi được sử dụng trong chỉ số mảng, đánh giá tới trị số của biến trước thao tác tăng.

-- Giảm, nghĩa là, $x--$ là tương đương với $x = x - 1$; khi được sử dụng trong chỉ số mảng, đánh giá tới trị số của biến trước thao tác giảm.

+= Sự tăng bởi lượng được định rõ, nghĩa là, $x += 3$ là tương đương với $x = x + 3$, và $x += (-3)$ là tương đương với $x = x + (-3)$.

-= Sự giảm bởi lượng được định rõ, nghĩa là, $x -= 3$ là tương đương với $x = x - 3$, và $x -= (-3)$ là tương đương với $x = x - (-3)$.

Ký hiệu khoảng

Ký hiệu dưới đây được sử dụng để định rõ khoảng của các trị số:

$x = y..z$ lấy các trị số nguyên bắt đầu từ y đến z , với x , y , và z là các số nguyên và z lớn hơn y .

Các hàm toán học

Các hàm toán học sau đây được định rõ:

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

$\text{Asin}(x)$ hàm sin nghịch đảo lượng giác, thao tác trên đối số x mà nằm trong khoảng từ $-1,0$ đến $1,0$, với trị số đầu ra trong khoảng từ $-\pi/2$ đến $\pi/2$, theo các đơn vị của các radian

$\text{Atan}(x)$ hàm tang nghịch đảo lượng giác, thao tác trên đối số x , với trị số đầu ra nằm trong khoảng từ $-\pi/2$ đến $\pi/2$, theo các đơn vị của các radian

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{khác} \end{cases}$$

$\text{Ceil}(x)$ số nguyên nhỏ nhất lớn hơn hoặc bằng x .

$$\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x)$$

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$$

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

$\text{Cos}(x)$ hàm cosin lượng giác thao tác trên đối số x theo các đơn vị của các radian.

$\text{Floor}(x)$ số nguyên lớn nhất nhỏ hơn hoặc bằng x .

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d/2 \\ c - d & ; \quad a - b > d/2 \\ c & ; \quad \text{khác} \end{cases}$$

$\text{Ln}(x)$ lôgarit tự nhiên của x (lôgarit cơ số e , trong đó e là hằng số cơ số lôgarit tự nhiên 2.718 281 828...).

$\text{Log2}(x)$ lôgarit cơ số 2 của x .

$\text{Log10}(x)$ lôgarit cơ số 10 của x .

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases}$$

$\text{Sin}(x)$ hàm sin lượng giác thao tác trên đối số x theo các đơn vị của các radian

$$\text{Sqrt}(x) = \sqrt{x}$$

$$\text{Swap}(x, y) = (y, x)$$

$\text{Tan}(x)$ hàm tang lượng giác thao tác trên đối số x theo các đơn vị của các radian

Thứ tự ưu tiên thao tác

Khi thứ tự ưu tiên trong biểu thức không được chỉ báo rõ ràng nhờ sử dụng dấu ngoặc đơn, các quy tắc sau đây áp dụng:

- Các thao tác của mức ưu tiên cao hơn được đánh giá trước thao tác bất kỳ mức ưu tiên thấp hơn.
- Các thao tác của cùng mức ưu tiên được đánh giá một cách liên tục từ trái sang phải.

Bảng dưới đây định rõ mức ưu tiên của các thao tác từ cao nhất tới thấp nhất; vị trí cao hơn trong bảng chỉ báo mức ưu tiên cao hơn.

Đối với các toán tử này mà cũng được sử dụng trong ngôn ngữ sự lập trình C, thứ tự của mức ưu tiên được sử dụng trong bản mô tả này là giống như được sử dụng trong ngôn ngữ sự lập trình C.

Bảng: Mức ưu tiên thao tác từ cao nhất (ở trên cùng của bảng) tới thấp nhất (ở dưới cùng của bảng)

Các thao tác (với toán hạng x, y, và z)
"x++", "x--"
"!x", "-x" (là toán hạng tiền tố một ngôi)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (là toán hạng hai đối số), $\sum_{i=x}^y f(i)$
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Mô tả bằng văn bản của các thao tác logic

Trong văn bản, câu lệnh của các thao tác logic như sẽ được mô tả dưới dạng toán học theo dạng sau đây:

```

    if( condition 0 )
        statement 0
else if( condition 1 )
    statement 1
...
else /* informative remark on remaining condition */
    statement n

```

có thể được mô tả theo cách sau đây:

... as follows / ... following applies:

- If condition 0, statement 0
- Otherwise, if condition 1, statement 1
- ...
- Otherwise (informative remark on remaining condition), statement n

Mỗi câu lệnh "If ... Otherwise, if ... Otherwise, ..." trong văn bản được đưa ra với "... as follows" hoặc "... following applies" được theo ngay sau bởi "If ... ". Điều kiện cuối cùng của "If ... Otherwise, if ... Otherwise, ..." luôn là "Otherwise, ...". Các câu lệnh "If ... Otherwise, if ... Otherwise, ..." được đan xen có thể được nhận dạng bằng cách so khớp "... as follows" hoặc "... following applies" với kết thúc "Otherwise, ...".

Trong văn bản, câu lệnh của các thao tác logic như sẽ được mô tả dưới dạng toán học theo dạng sau đây:

```

    if( condition 0a  &&  condition 0b )
        statement 0
else if( condition 1a  ||  condition 1b )
    statement 1

```

...

else

statement n

có thể được mô tả theo cách sau đây:

... as follows / ... following applies:

- If all of following conditions are true, statement 0:
- condition 0a
- condition 0b
- Otherwise, if one or more of following conditions are true, statement 1:
- condition 1a
- condition 1b
- ...
- Otherwise, statement n

Trong văn bản, câu lệnh của các thao tác logic như sẽ được mô tả dưới dạng toán học theo dạng sau đây:

if(condition 0)

statement 0

if(condition 1)

statement 1

có thể được mô tả theo cách sau đây:

When condition 0, statement 0

When condition 1, statement 1

Mặc dù các phương án của sáng chế đã được mô tả chủ yếu dựa vào lập mã video, cần lưu ý rằng các phương án của hệ thống lập mã 10, bộ mã hóa 20 và bộ giải mã 30 (và tương ứng là hệ thống 10) và các phương án được mô tả khác ở đây có thể cũng được tạo cấu hình cho việc lập mã hoặc xử lý ảnh tĩnh, nghĩa là xử lý hoặc lập mã của ảnh riêng độc lập với bất kỳ ảnh đứng trước hoặc tiếp theo như trong lập mã video. Nói chung chỉ các bộ phận dự đoán liên kết 244 (bộ mã hóa) và 344 (bộ giải mã) có thể là không khả dụng trong trường hợp lập mã xử lý ảnh bị giới hạn ở ảnh đơn 17. Tất cả các chức năng khác (cũng được đề cập đến là các công cụ hoặc các công nghệ) của bộ mã hóa video 20 và bộ giải mã video 30 có thể đều được sử dụng cho việc xử lý ảnh tĩnh, ví dụ, tính 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, dự đoán bên trong 254/354, và/hoặc lọc vòng lặp 220, 320, và lập mã entropi 270 và giải mã entropi 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 trong phần cứng, phần mềm, phần sụn, hoặc sự kết hợp bất kỳ của nó. Nếu được thực hiện trong phần mềm, các chức năng có thể được lưu trữ trên phương tiện đọc được bởi máy tính hoặc được truyền qua các phương tiện truyền thông dưới dạng một hoặc nhiều lệnh hoặc mã và được thực hiện bởi bộ 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 thuận lợi cho việc truyền của chương trình máy tính từ một vị trí này đến vị trí khác, ví dụ, theo giao thức truyền thông. Theo cách 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 phải 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, các cấu trúc dữ liệu và/hoặc mã cho việc thực hiện của các kỹ thuật được mô tả trong sáng chế này. 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 phương tiện lưu trữ dạng đĩa quang khác, phương tiện lưu trữ dạng đĩa từ, hoặc các thiết bị lưu trữ từ khác, bộ nhớ tia chớp, hoặc phương tiện bất kỳ khác mà có thể được sử dụng để lưu trữ mã chương trình mong muốn dưới dạng các cấu trúc dữ liệu hoặc các lệnh và có thể được truy cập bởi máy tính. Ngoài ra, sự kết nối bất kỳ được gọi một cách thích hợp là phương tiện đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ trang mạng (website), máy chủ, hoặc nguồn từ xa khác sử dụng cáp đồng trục, cáp sợi quang, cáp cặp xoắn, đường dây thuê bao số (digital subscriber line, viết tắt là DSL), hoặc các công nghệ không dây chẳng hạn như hồng ngoại, radio, và vi sóng, thì cáp đồng trục, cáp sợi quang, cáp cặp xoắn, 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 tạm 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 số (digital versatile disc, viết tắt là DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa (disk) thường tái tạo dữ liệu dưới dạng từ, trong khi đĩa (disc) tái tạo dữ liệu dưới dạng quang học bằng các tia laze. Các sự kết hợp của chúng cũng sẽ được bao gồm 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 số (DSP), các bộ vi xử lý đa năng, các mạch tích hợp chuyên

dụng (application specific integrated circuit, viết tắt là ASIC), các mảng logic lập trình được dạng trường (field programmable logic array, viết tắt là 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 của cấu trúc nêu trên hoặc cấu trúc bất kỳ khác thích hợp cho việc thực hiện của các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, chức năng được mô tả ở đây có thể được đưa ra trong các môđun phần cứng và/hoặc phần mềm chuyên dụng được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp trong bộ lập mã -giải mã được kết hợp. Ngoài ra, các kỹ thuật sẽ được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc các phân tử logic.

Các kỹ thuật của sáng chế có thể được thực hiện trong phạm vi rộng của các thiết bị hoặc các 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 trong sáng chế để nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, mà không nhất thiết phải thực hiện bởi các bộ phận phần cứng khác nhau. Đúng hơn, như được nêu trên, các bộ phận khác nhau có thể được kết hợp trong bộ phận phần cứng bộ lập mã -giải mã hoặc được cung cấp bởi sự tập hợp của các bộ phận phần cứng tương tác, bao gồm một hoặc nhiều bộ xử lý như được nêu trên, cùng với phần mềm và/hoặc phần sụn thích hợp.

YÊU CẦU BẢO HỘ

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

thu nhận dòng bit đối với ảnh hiện tại;

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại;

thu nhận trị số của địa chỉ lát của ảnh hiện tại;

thu nhận trị số của tileX theo trị số của địa chỉ lát, trong đó trị số của tileX chỉ báo địa chỉ theo chiều ngang trong chỉ số mảnh;

phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

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

thiết đặt trị số chỉ báo đối với độ rộng lát tới trị số mặc định, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX là bằng ngưỡng thứ nhất.

3. Phương pháp theo điểm 2, trong đó trị số mặc định chỉ báo rằng độ rộng của lát bao gồm 1 cột mảnh.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm:

thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 4, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

6. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

7. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 6, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

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

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát, trong đó trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh;

phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

9. Phương pháp theo điểm 8, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY là bằng ngưỡng thứ hai.

10. Phương pháp theo điểm 8 hoặc 9, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm:

thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

11. Phương pháp theo điểm bất kỳ trong số các điểm từ 8 đến 10, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

12. Phương pháp lập mã được thực hiện bởi thiết bị giải mã, phương pháp này bao gồm các bước:

thu nhận dòng bit đối với ảnh hiện tại;

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại;

thu nhận trị số của địa chỉ lát của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát, trong đó trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh;

phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

13. Phương pháp theo điểm 12, trong đó phương pháp này còn bao gồm bước:

thiết đặt trị số chỉ báo đối với độ cao lát tới trị số được thiết đặt trước, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY là bằng ngưỡng thứ hai.

14. Phương pháp theo điểm 13, trong đó trị số mặc định chỉ báo rằng độ cao của lát bao gồm 1 hàng mảnh.

15. Phương pháp theo điểm bất kỳ trong số các điểm từ 12 đến 14, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

16. Phương pháp theo điểm 15, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

17. Phương pháp theo điểm bất kỳ trong số các điểm từ 12 đến 16, trong đó địa chỉ lát là chỉ số của mảnh ngoài cùng bên trái của hàng mảnh trên cùng của lát, trong đó các mảnh được bao gồm trong ảnh hiện tại được lấy chỉ số theo thứ tự quét mảnh.

18. Phương pháp theo điểm bất kỳ trong số các điểm từ 12 đến 17, trong đó trị số của địa chỉ lát định rõ địa chỉ của tọa độ trên cùng bên trái của lát trong chỉ số mảnh.

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

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại;

thu nhận trị số của địa chỉ lát của ảnh hiện tại;

thu nhận trị số của tileX theo trị số của địa chỉ lát, trong đó trị số của tileX chỉ báo địa chỉ theo chiều ngang trong chỉ số mảnh;

lập mã trị số chỉ báo đối với độ rộng lát đối với ảnh hiện tại để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

20. Phương pháp theo điểm 19, trong đó phương pháp này còn bao gồm các bước:

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát, trong đó trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh; lập mã trị số chỉ báo đối với độ cao lát đối với ảnh hiện tại thành dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

21. Phương pháp theo điểm 19 hoặc 20, trong đó bước thu nhận trị số của tileX theo trị số của địa chỉ lát bao gồm:

thu nhận trị số của tileX theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

22. Phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 21, trong đó trị số của tileX được thu nhận theo phép toán môđun giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

23. Phương pháp theo điểm 20 hoặc 22, trong đó bước thu nhận trị số của tileY theo trị số của địa chỉ lát bao gồm:

thu nhận trị số của tileY theo trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

24. Phương pháp theo điểm bất kỳ trong số các điểm từ 20 đến 23, trong đó trị số của tileY được thu nhận theo phép chia số nguyên giữa trị số của địa chỉ lát và số lượng các mảnh trong hàng mảnh của ảnh hiện tại.

25. Phương pháp lập mã được thực hiện bởi thiết bị mã hóa, phương pháp này bao gồm các bước:

thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại;

thu nhận trị số của địa chỉ lát của ảnh hiện tại;

thu nhận trị số của tileY theo trị số của địa chỉ lát, trong đó trị số của tileY chỉ báo địa chỉ theo chiều dọc trong chỉ số mảnh;

lập mã trị số chỉ báo đối với độ cao lát đối với ảnh hiện tại để thu nhận dòng bit, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

26. 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 11.

27. Bộ giải mã (30) bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 12 đến 18.

28. Bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 24.

29. Bộ mã hóa bao gồm hệ mạch xử lý để thực hiện phương pháp theo điểm 25.

30. 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 cho việc 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 11.

31. 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 cho việc 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 12 đến 18.

32. 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 cho việc 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 19 đến 24.

33. 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 cho việc 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 25.

34. 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 11.

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 12 đến 18.

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 19 đến 24.

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 25.

38. Thiết bị giải mã dữ liệu video, bao gồm:

bộ lưu trữ bộ nhớ không tạm thời, được tạo cấu hình để lưu trữ dữ liệu video ở dạng dòng bit; và

bộ giải mã video, được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 11.

39. Thiết bị giải mã dữ liệu video, bao gồm:

bộ lưu trữ bộ nhớ không tạm thời, được tạo cấu hình để lưu trữ dữ liệu video ở dạng dòng bit; và

bộ giải mã video, được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 12 đến 18.

40. Thiết bị mã hóa dữ liệu video, bao gồm:

bộ lưu trữ bộ nhớ không tạm thời, được tạo cấu hình để lưu trữ dữ liệu video ở dạng dòng bit; và

bộ mã hóa video, được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 19 đến 24.

41. Thiết bị mã hóa dữ liệu video, bao gồm:

bộ lưu trữ bộ nhớ không tạm thời, được tạo cấu hình để lưu trữ dữ liệu video ở dạng dòng bit; và

bộ mã hóa video, được tạo cấu hình để thực hiện phương pháp theo điểm 25.

42. Phương tiện lưu trữ không tạm thời bao gồm dòng bit được giải mã bằng cách thực hiện các bước:

thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại;

thu nhận trị số của địa chỉ lát của ảnh hiện tại;

thu nhận trị số của tileX theo trị số của địa chỉ lát, trong đó trị số của tileX chỉ báo địa chỉ theo chiều ngang trong chỉ số mảnh;

phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

43. Máy giải mã video, bao gồm:

môđun thu nhận, được tạo cấu hình để thu nhận dòng bit đối với ảnh hiện tại;

môđun thu nhận, còn được tạo cấu hình để thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại và thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, mà được tạo cấu hình để thu nhận trị số của tileX theo trị số của địa chỉ lát;

môđun phân tách, mà được tạo cấu hình để phân tách trị số chỉ báo đối với độ rộng lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

44. Máy giải mã video theo điểm 43, trong đó môđun thu nhận, môđun tính toán và môđun phân tách còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 2 đến 11.

45. Thiết bị giải mã video, bao gồm:

môđun thu nhận, được tạo cấu hình để thu nhận dòng bit đối với ảnh hiện tại;

môđun thu nhận, còn được tạo cấu hình để thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại và thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, được tạo cấu hình để thu nhận trị số của tileY theo trị số của địa chỉ lát;

môđun phân tách, mà được tạo cấu hình để phân tách trị số chỉ báo đối với độ cao lát từ dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

46. Thiết bị giải mã video theo điểm 45, trong đó môđun thu nhận, môđun tính toán và môđun phân tách còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 13 đến 18.

47. Thiết bị mã hóa video, bao gồm:

môđun thu nhận, được tạo cấu hình để thu nhận số lượng các mảnh trong hàng mảnh của ảnh hiện tại; môđun thu nhận, còn được tạo cấu hình để thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, được tạo cấu hình để thu nhận trị số của tileX theo trị số của địa chỉ lát;

môđun lập mã, được tạo cấu hình để lập mã trị số chỉ báo đối với độ rộng lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong hàng mảnh của ảnh hiện tại và trị số của tileX không bằng ngưỡng thứ nhất.

48. Thiết bị mã hóa video theo điểm 47, trong đó môđun thu nhận, môđun tính toán và môđun lập mã còn được tạo cấu hình để thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 20 đến 24.

49. Thiết bị mã hóa video, bao gồm:

môđun thu nhận, được tạo cấu hình để thu nhận số lượng các mảnh trong cột mảnh của ảnh hiện tại;

môđun thu nhận, còn được tạo cấu hình để thu nhận trị số của địa chỉ lát của ảnh hiện tại;

môđun tính toán, được tạo cấu hình để thu nhận trị số của tileY theo trị số của địa chỉ lát;

môđun lập mã, được tạo cấu hình để lập mã trị số chỉ báo đối với độ cao lát trong dòng bit đối với ảnh hiện tại, khi trị số độ chênh lệch giữa số lượng các mảnh trong cột mảnh của ảnh hiện tại và trị số của tileY không bằng ngưỡng thứ hai.

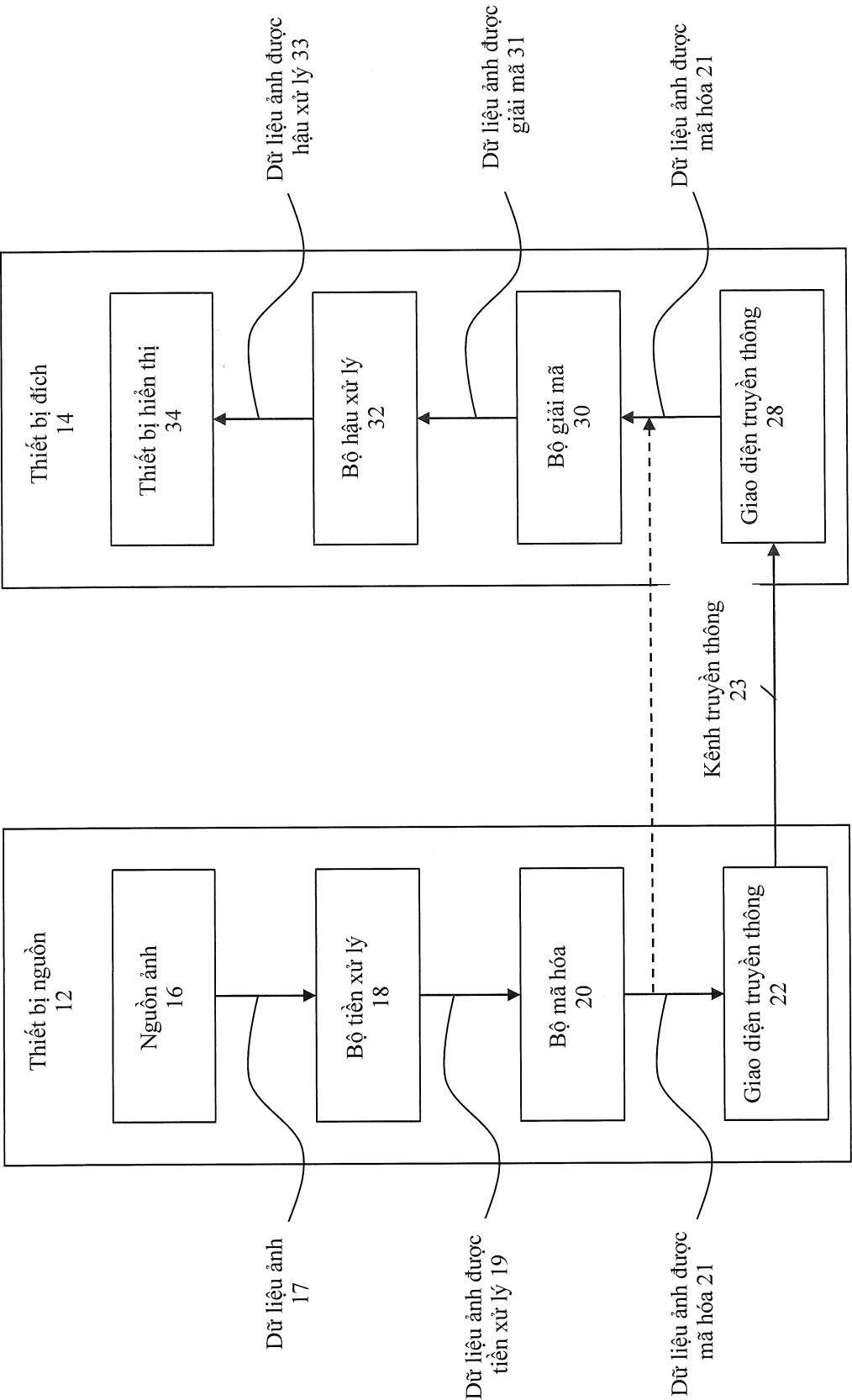


FIG. 1A

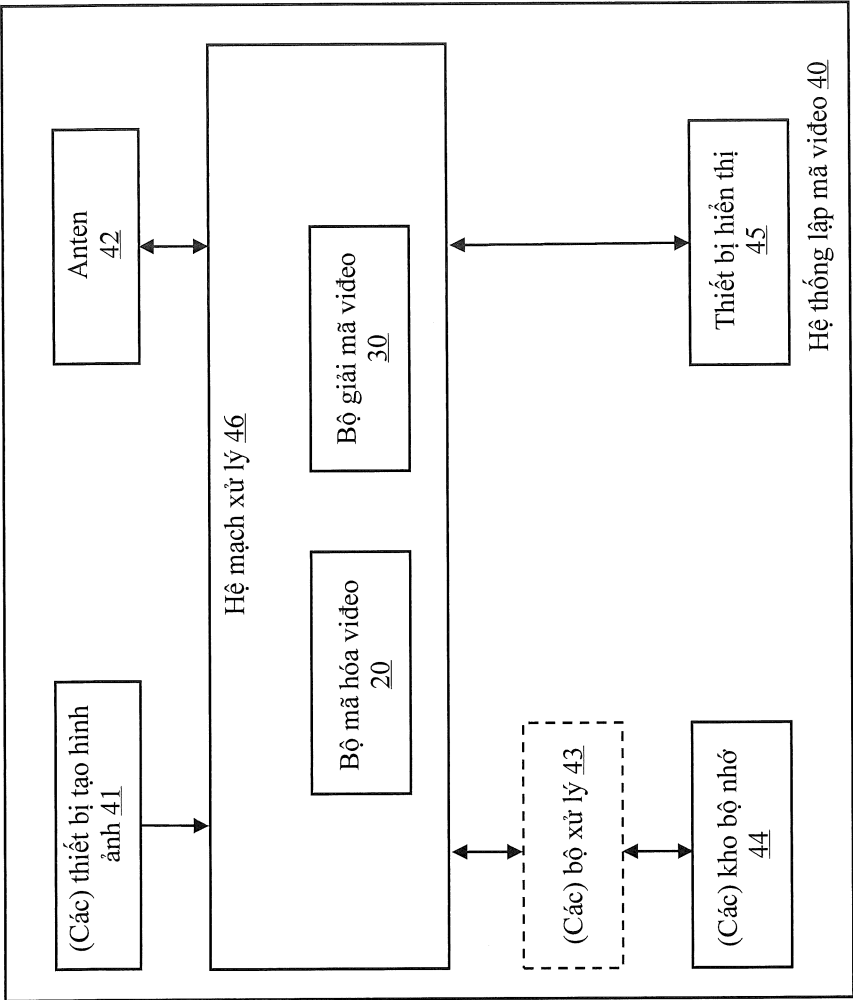


FIG. 1B

3/15

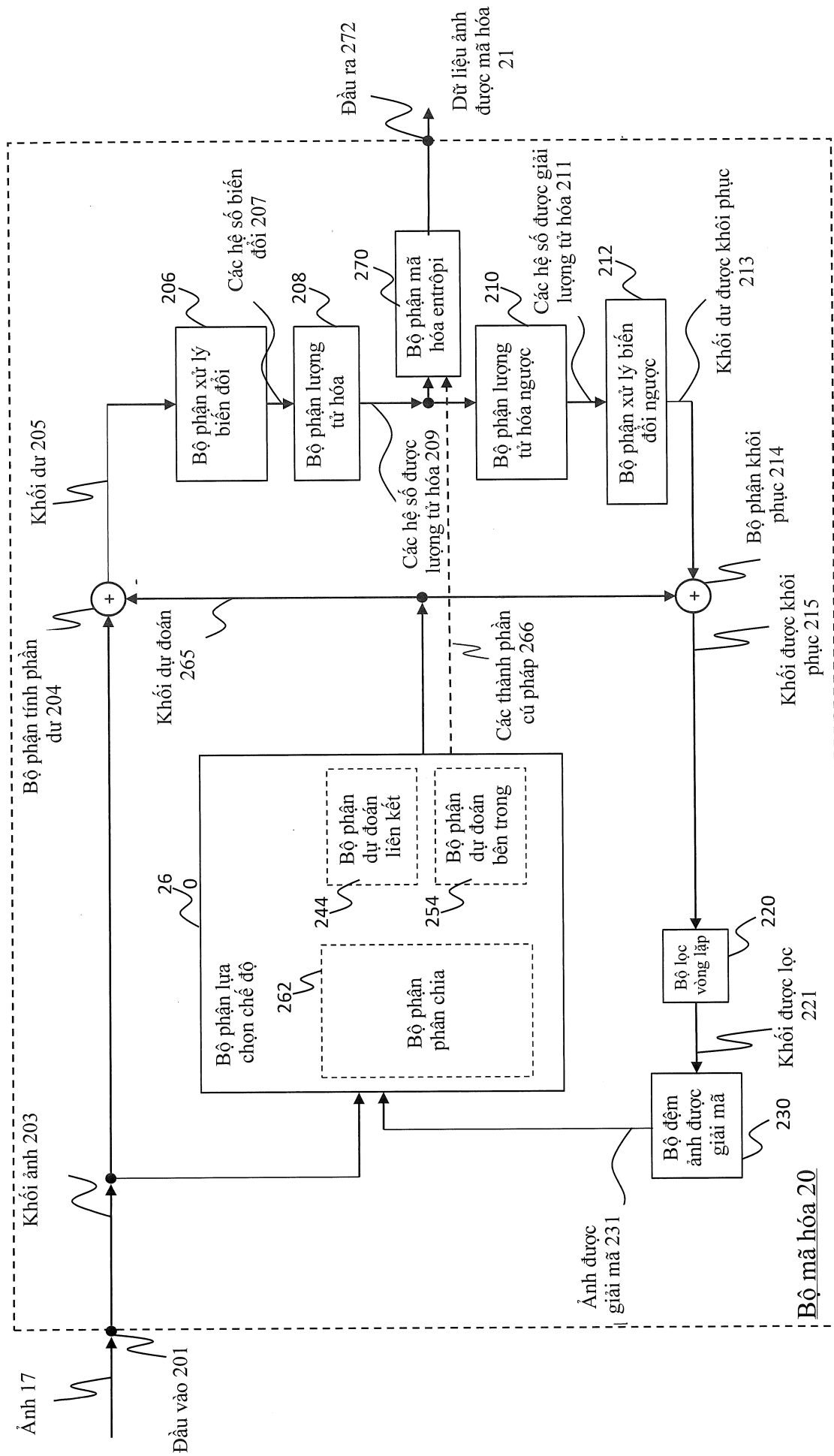


FIG. 2

4/15

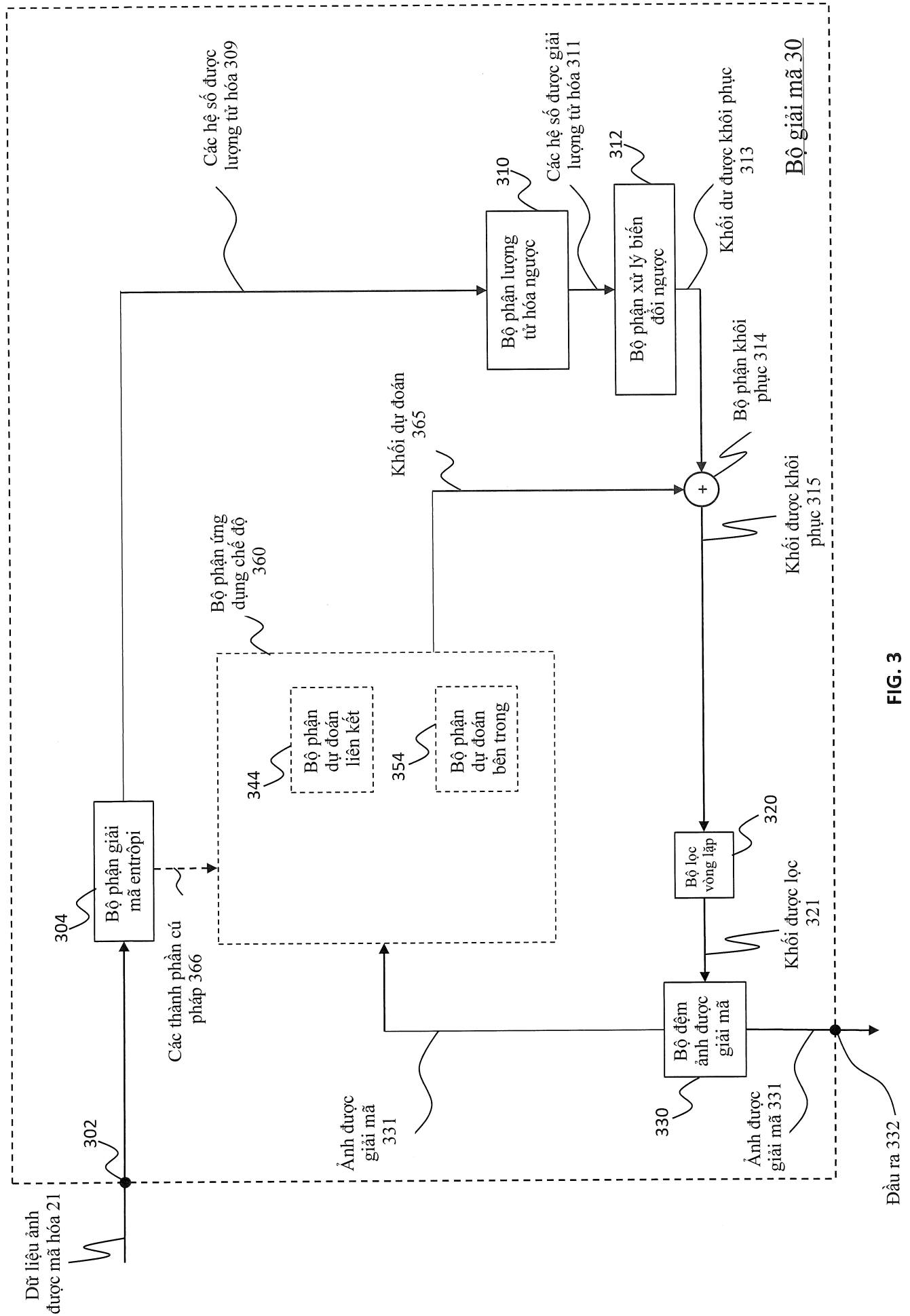
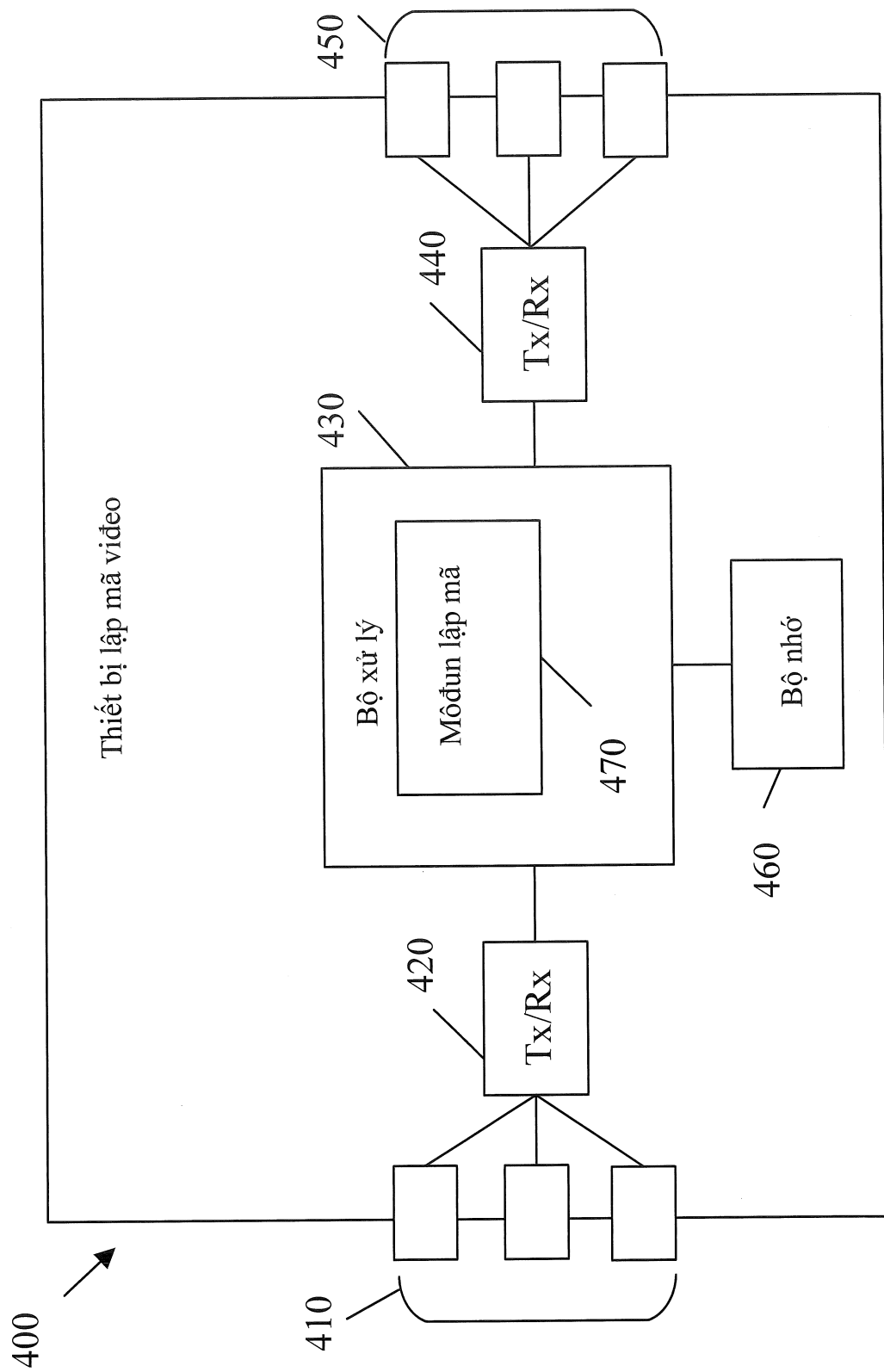


FIG. 3



Các cổng xuôi dòng

Các cổng ngược dòng

FIG. 4

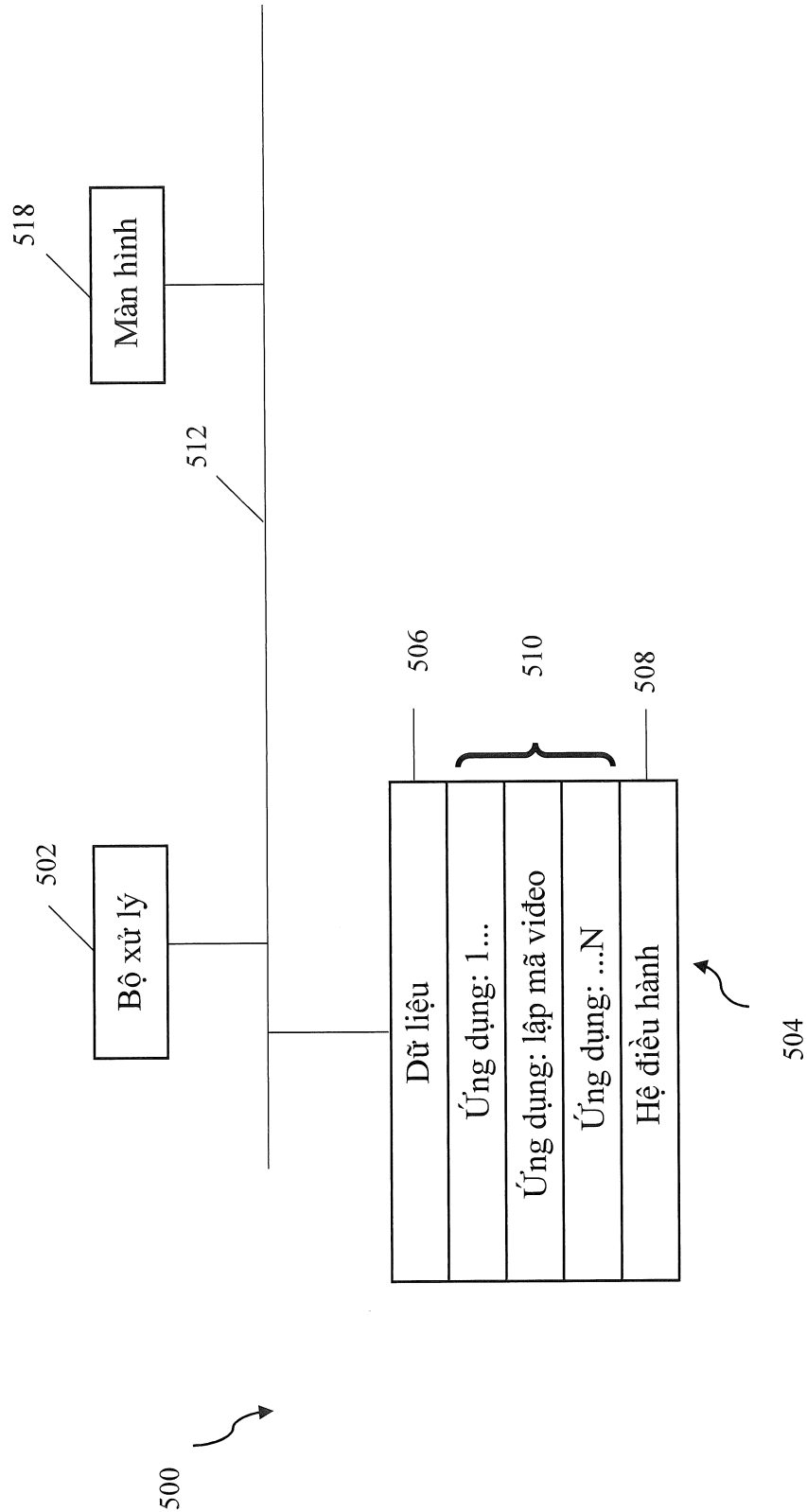


FIG. 5

7/15

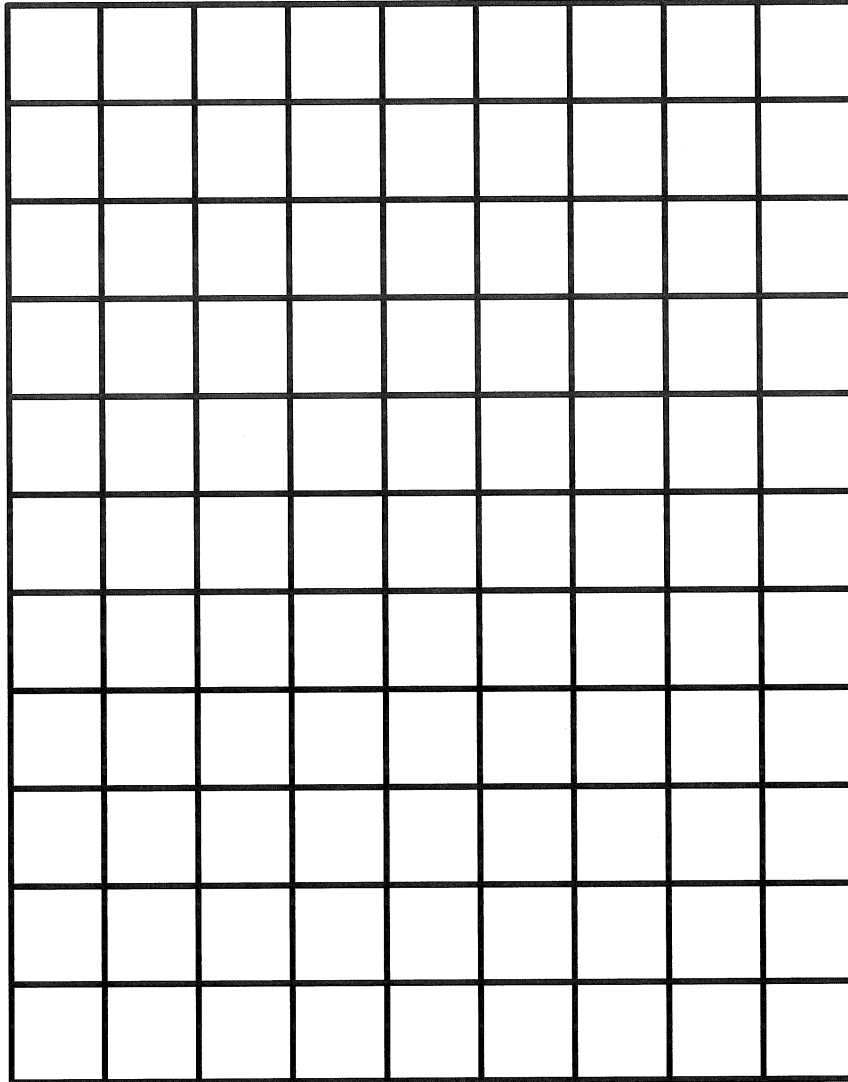


FIG. 6

0	1	2	3	16	17	24	25	26	27	28
4	5	6	7	18	19	29	30	31	32	33
8	9	10	11	20	21	34	35	36	37	38
12	13	14	15	22	23	39	40	41	42	43
44	45	46	47	52	53	56	57	58	59	60
48	49	50	51	54	55	61	62	63	64	65
66	67	68	69	78	79	84	85	86	87	88
70	71	72	73	80	81	89	90	91	92	93
74	75	76	77	82	83	94	95	96	97	98

FIG. 7

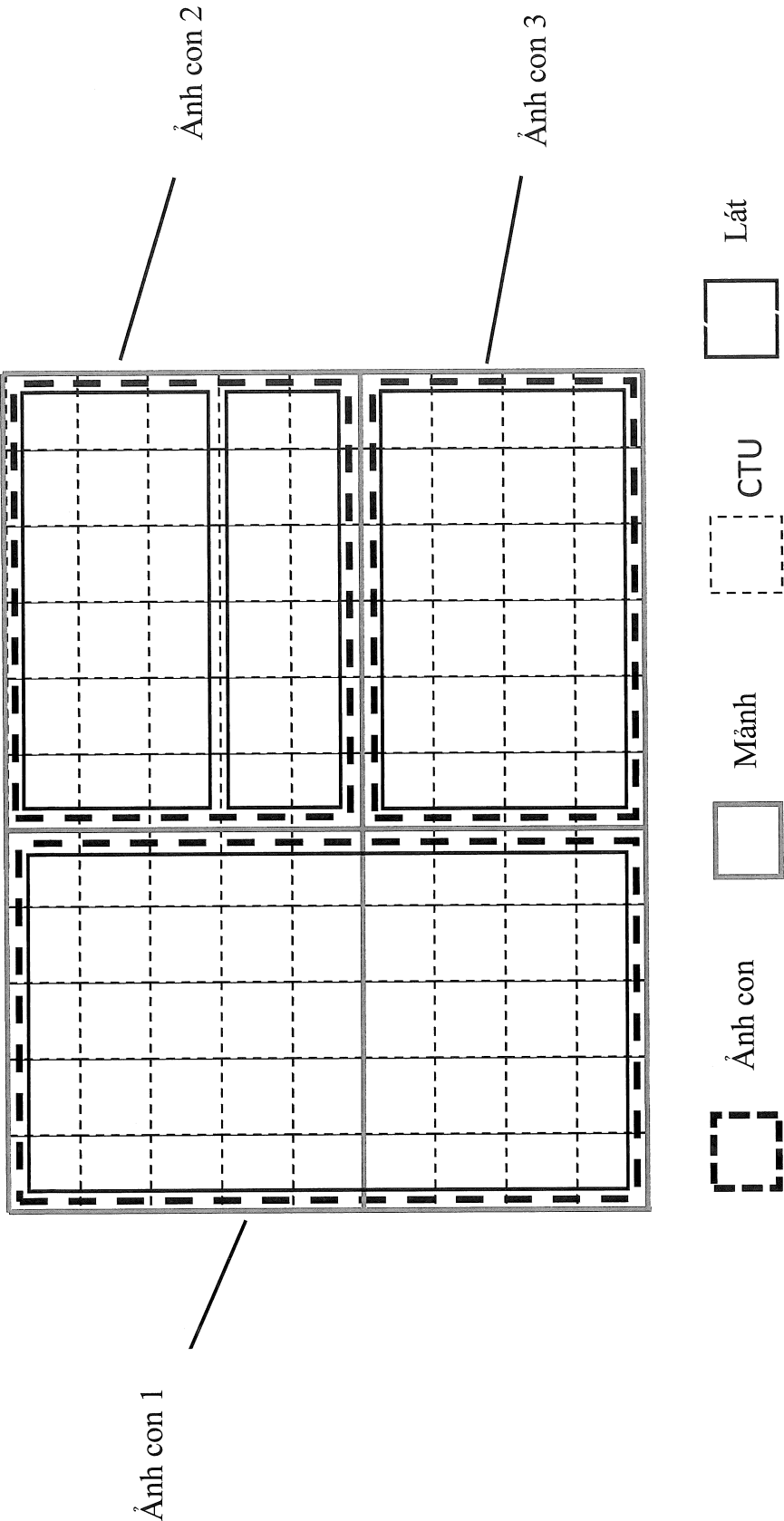


FIG. 8

1	2	3	4	5	6	7	8	9	10	CTU 10
11	12	13	14	15	16	17	18	19	20	Lát 5
21	22	23	24	25	26	27	28	29	30	
31	32	33	34	35	36	37	38	39	40	

FIG. 9

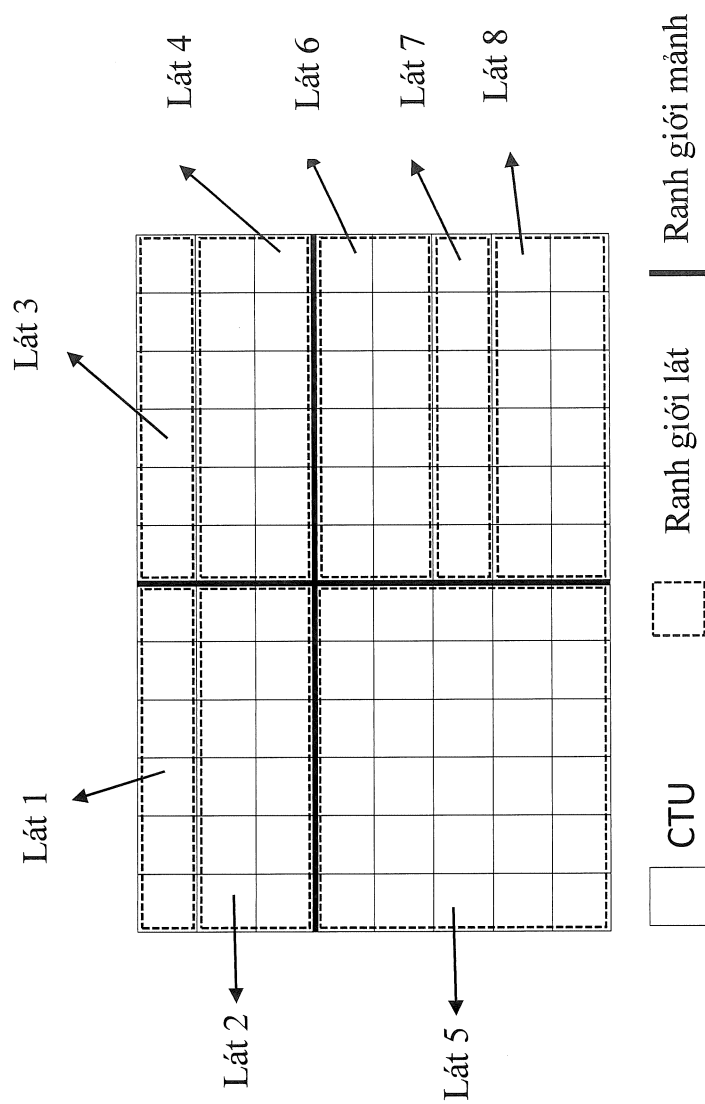


FIG. 10

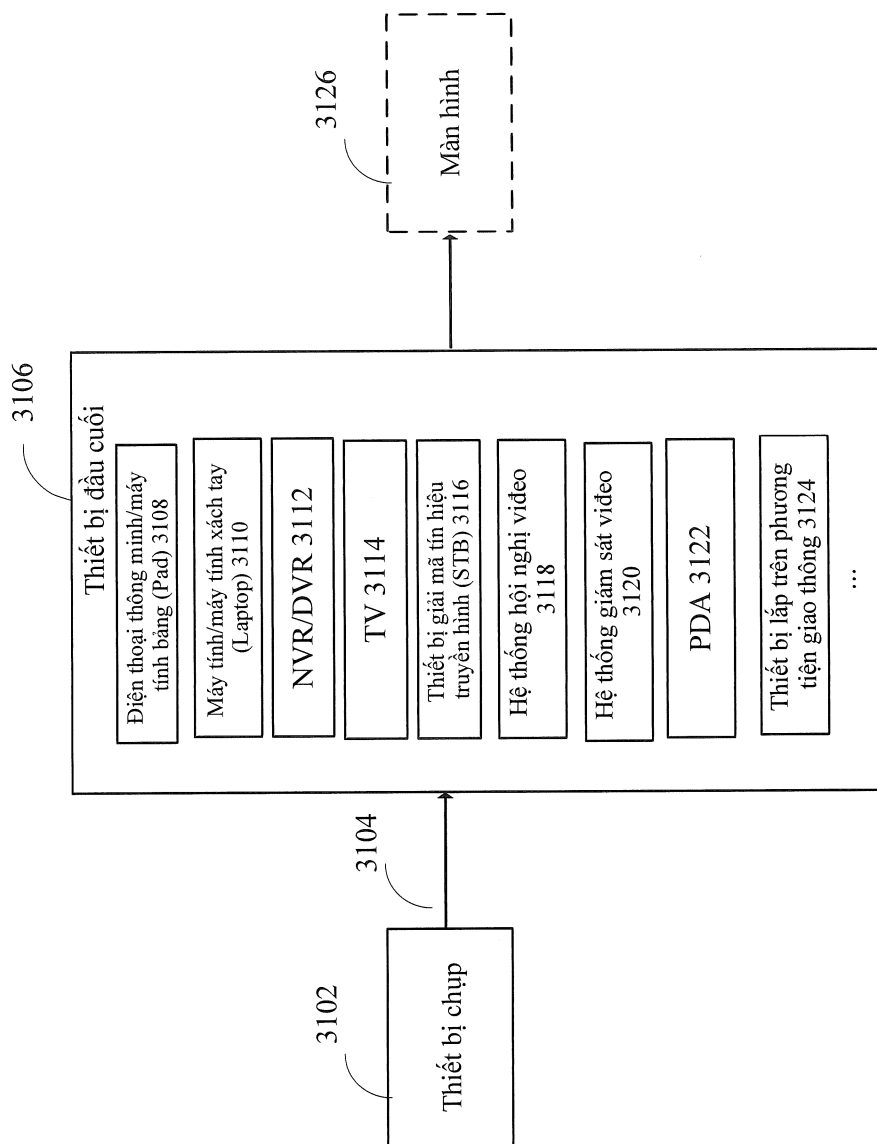


FIG. 11

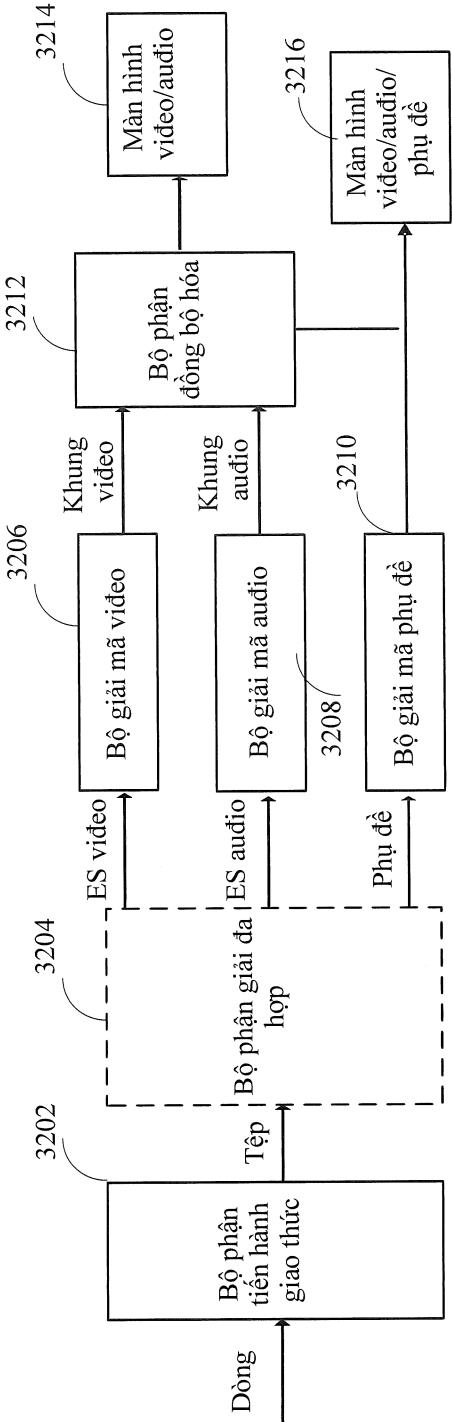


FIG. 12

14/15

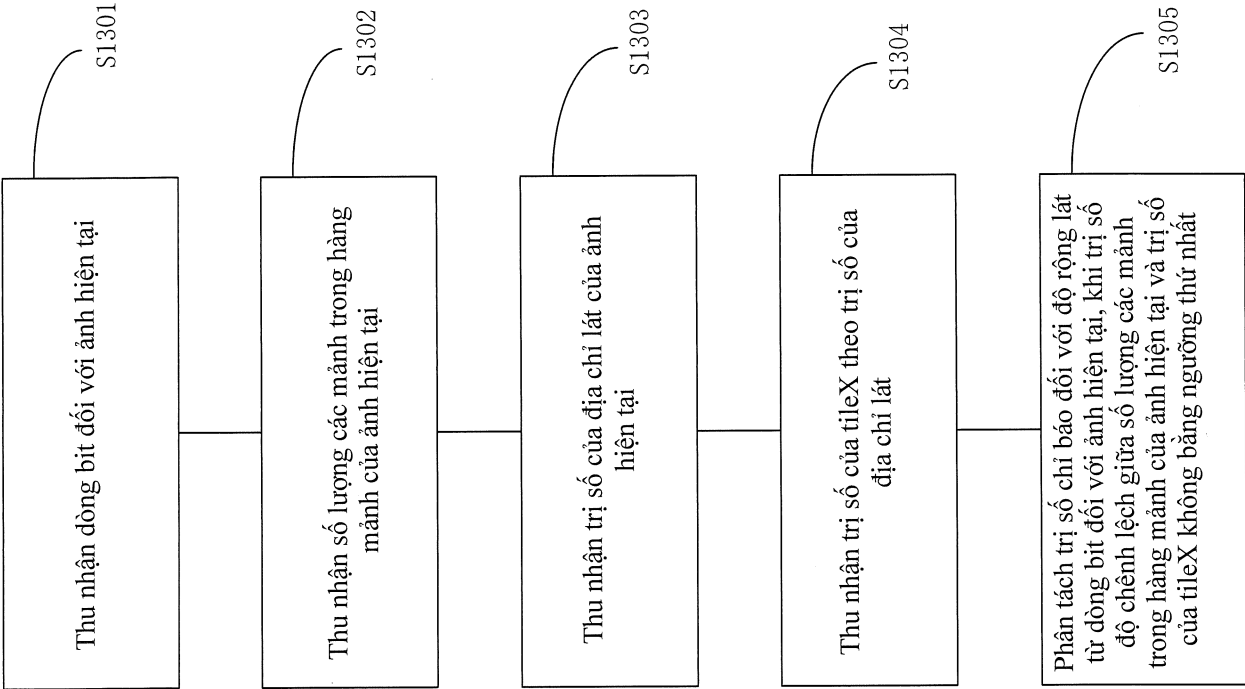


FIG. 13

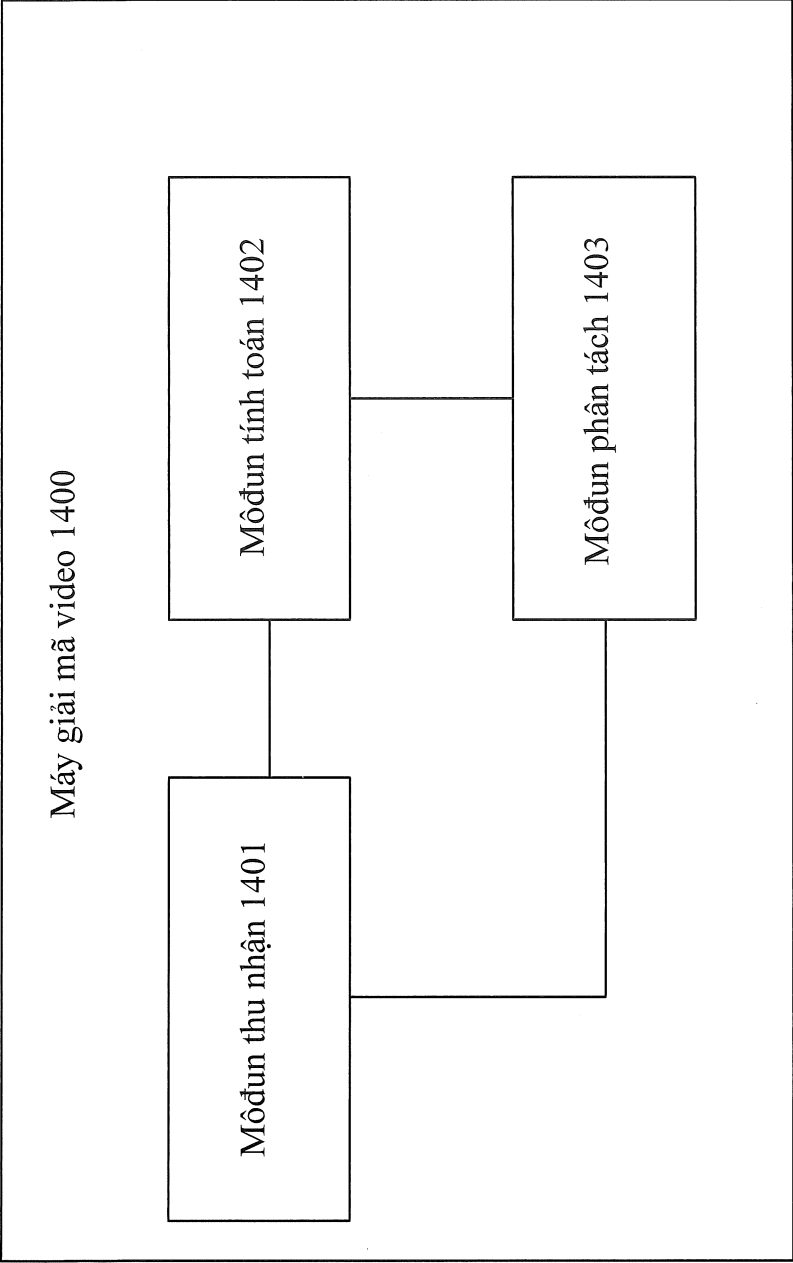


FIG. 14