



(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)



1-0052143

(51)^{2006.01} H04N 7/26

(13) B

(21) 1-2023-00339

(22) 19/09/2013

(62) 1-2015-00409

(86) PCT/JP2013/005541 19/09/2013

(87) WO 2014/050038 03/04/2014

(30) 61/705,846 26/09/2012 US; 61/711,892 10/10/2012 US

(45) 27/10/2025 451

(43) 27/11/2023 428A

(73) Sun Patent Trust (US)

437 Madison Avenue, 35th Floor, New York, NY 10022, United States of America

(72) ESENLİK, Semih (TR); NARROSCHE, Matthias (DE); WEDI, Thomas (DE).

(74) Công ty TNHH Quốc tế D&N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ MÃ HÓA ẢNH

(21) 1-2023-00339

(57) Sáng chế đề cập đến phương pháp và thiết bị mã hóa ảnh. Chỉ báo phụ thuộc được báo hiệu trong phần đầu của gói, nghĩa là, nằm sát phần đầu lát cần được phân tích cú pháp hoặc tập hợp thông số. Điều này đạt được, ví dụ, nhờ bao gồm chỉ báo phụ thuộc ở đầu của phần đầu lát, tốt hơn là sau phần tử cú pháp nhận dạng tập hợp thông số và trước địa chỉ lát, bằng cách bao gồm chỉ báo phụ thuộc trước địa chỉ lát, bằng cách cung cấp chỉ báo phụ thuộc cho phần đầu NALU bằng cách sử dụng bản tin riêng, hoặc bằng cách sử dụng loại NALU đặc biệt cho các NALU mang các lát phụ thuộc.

slice_header () {	1310 Phân mô tả
first_slice_in_pic_flag	u(1)
if(RapPicFlag)	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(!dependent_slice_flag) {	
slice_type	ue(v)
... ..	...

FIG. 13

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp mã hóa ảnh để mã hoá ảnh và phương pháp giải mã ảnh để giải mã ảnh.

Tình trạng kỹ thuật của sáng chế

Phần lớn các thuật toán mã hóa video được chuẩn hóa hiện nay đều dựa vào mã hóa video lai. Trong các phương pháp mã hóa video lai, một vài sơ đồ nén có tổn hao và không tổn hao khác nhau được sử dụng để đạt được độ lợi nén mong muốn. Mã hóa video lai cũng là cơ sở đối với các chuẩn ITU-T (các chuẩn H.26x chẳng hạn như H.261, H.263) cũng như các chuẩn ISO/IEC (các chuẩn MPEG-X chẳng hạn như MPEG-1, MPEG-2, và MPEG-4).

Chuẩn mã hóa video tiên tiến và mới nhất hiện là chuẩn được gọi là mã hóa video nâng cao (advanced video coding - AVC) H.264/MPEG-4. Đây là kết quả của các nỗ lực chuẩn hóa bởi nhóm video liên kết (joint video team - JVT), nhóm liên kết của các nhóm MPEG ITU-T và ISO/IEC.

Hơn nữa, chuẩn mã hóa video được gọi là mã hóa video hiệu quả cao (High-Efficiency Video Coding - HEVC) đang được xem xét bởi nhóm cộng tác liên kết về mã hóa video (Joint Collaborative Team on Video Coding - JCT-VC)), với mục đích cụ thể là nâng cao hiệu quả về mã hóa video độ phân giải cao.

Danh mục tài liệu trích dẫn

Tài liệu phi sáng chế

Tài liệu phi sáng chế 1: C. Gordon, và các tác giả khác, “Wavefront Parallel Processing for HEVC Encoding and Decoding”, JCTVC-F274-v2, từ hội nghị ở Torino, tháng bảy năm 2011, nguồn Internet <URL: <http://phenix.int-evry.fr>.>

Tài liệu phi sáng chế 2: A. Fuldseth, và các tác giả khác, ““Tiles”, JCTVC-F355-v1, từ hội nghị ở Torino, tháng bảy năm 2011, nguồn Internet <URL: <http://phenix.int-evry.fr>.>

Tài liệu phi sáng chế 3: JCTVC-J1003_d7, “High efficiency video coding (HEVC) text specification draft 8”, tháng bảy năm 2012, trang 73, “dependent_slice_flag”, nguồn

Internet <URL: <http://phenix.IT-sudparis.eu/jct/>>

Vấn đề kỹ thuật

Tuy nhiên, có vấn đề là phương pháp mã hóa ảnh, phương pháp giải mã ảnh, và các phương pháp tương tự không có đủ hiệu quả xử lý.

Bản chất kỹ thuật của sáng chế

Do đó, sáng chế đề xuất phương pháp mã hóa ảnh và phương pháp giải mã ảnh để có thể nâng cao hiệu quả xử lý.

Cách thức giải quyết vấn đề

Phương pháp mã hóa ảnh theo một khía cạnh của sáng chế là phương pháp mã hóa ảnh để thực hiện xử lý mã hóa bằng cách phân chia hình ảnh thành các lát, phương pháp mã hóa ảnh bao gồm bước truyền dòng bit gồm có: cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc trên đó việc xử lý mã hóa được thực hiện phụ thuộc vào kết quả của việc xử lý mã hóa trên lát khác với lát hiện thời hay không; địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời; và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Phương pháp giải mã ảnh theo một khía cạnh của sáng chế là phương pháp giải mã ảnh để thực hiện xử lý giải mã bằng cách phân chia hình ảnh thành các lát, phương pháp giải mã ảnh bao gồm trích, từ dòng bit được mã hóa, cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc trên đó xử lý giải mã được thực hiện phụ thuộc vào kết quả của việc xử lý giải mã trên lát khác với lát hiện thời hay không, địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời, và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Các khía cạnh chung và cụ thể được bộc lộ ở trên có thể được thực hiện bằng cách sử dụng hệ thống, phương pháp, và mạch tích hợp, chương trình máy tính, hoặc phương

tiện lưu trữ đọc được bởi máy tính chẳng hạn như CD-ROM, hoặc dạng kết hợp bất kỳ của các hệ thống, phương pháp, mạch tích hợp, các chương trình máy tính, hoặc các phương tiện lưu trữ đọc được bởi máy tính.

Hiệu quả có lợi của sáng chế

Phương pháp mã hóa ảnh và phương pháp giải mã ảnh theo sáng chế có thể nâng cao hiệu quả mã hóa.

Mô tả vắn tắt các hình vẽ

Các mục đích này và các mục đích khác, các dấu hiệu và các lợi ích của sáng chế sẽ trở nên rõ ràng từ phần mô tả dưới đây có dựa vào các hình vẽ kèm theo mà minh họa phương án cụ thể của sáng chế.

Fig.1 là sơ đồ khối thể hiện ví dụ về bộ mã hóa phù hợp với HEVC.

Fig.2 là sơ đồ khối thể hiện ví dụ về bộ giải mã phù hợp với HEVC.

Fig.3 là sơ đồ thể hiện ví dụ về cấu hình của ảnh trong xử lý song song mặt sóng (wavefront parallel processing - WPP).

Fig.4 là sơ đồ thể hiện ví dụ về mối tương quan giữa lát thông thường và lát phụ thuộc trong xử lý song song mặt sóng.

Fig.5 là sơ đồ thể hiện ví dụ về phần đầu gói.

Fig.6 là sơ đồ thể hiện ví dụ về phần đầu lát của lát entropy hoặc lát phụ thuộc.

Fig.7 là sơ đồ thể hiện các phần phụ thuộc và việc báo hiệu khi lát thông thường được sử dụng.

Fig.8 là hình vẽ dạng sơ đồ thể hiện các phần phụ thuộc và việc báo hiệu khi lát phụ thuộc và lát entropy được sử dụng.

Fig.9A là sơ đồ thể hiện ví dụ về cài đặt cú pháp của các phần phụ thuộc liên lớp, các phần phụ thuộc theo thời gian và các phần phụ thuộc liên lát trong HM8.0.

Fig.9B là sơ đồ để giải thích các bước phân tích cú pháp được thực hiện để phân tích cú pháp các phần phụ thuộc liên lớp trong HM8.0.

Fig.9C là sơ đồ để giải thích các bước phân tích cú pháp được thực hiện để phân tích cú pháp các phần phụ thuộc liên lớp trong HM8.0.

Fig.10 là sơ đồ thể hiện ví dụ về vị trí của `dependent_slice_flag` (cờ lát phụ thuộc).

Fig.11 là sơ đồ thể hiện ví dụ về cú pháp khi điều kiện phân tích cú pháp liên quan đến `dependent_slice_enabled_flag` (cờ cho phép lát phụ thuộc) trên Fig.10 được loại bỏ.

Fig.12 là sơ đồ thể hiện ví dụ về cú pháp khi `dependent_slice_flag` được dịch chuyển trước `first_slice_in_pic_flag` (lát thứ nhất trong cờ ảnh).

Fig.13 là sơ đồ thể hiện ví dụ về cú pháp khi `dependent_slice_flag` được dịch chuyển trước phần tử cú pháp `slice_address` (địa chỉ lát).

Fig.14 là sơ đồ thể hiện ví dụ về cú pháp khi `dependent_slice_flag` được dịch chuyển bên trong phần đầu NAL.

Fig.15 là sơ đồ thể hiện ví dụ về cú pháp của phần đầu lát của lát phụ thuộc khi loại mới được bổ sung vào các loại đơn vị NAL được sử dụng cho các lát phụ thuộc.

Fig.16 là sơ đồ thể hiện ví dụ về cú pháp của phần đầu lát và của phần đầu đơn vị NAL khi được giả thiết rằng `dependent_slice_flag` được đặt bằng 1 cho các loại NALU nhất định.

Fig.17 thể hiện cấu hình tổng thể của hệ thống cung cấp nội dung để thực hiện các dịch vụ phân phối nội dung.

Fig.18 thể hiện cấu hình tổng thể của hệ thống phát quảng bá số.

Fig.19 thể hiện sơ đồ khối minh họa ví dụ về cấu hình của máy thu hình.

Fig.20 thể hiện sơ đồ khối minh họa ví dụ về cấu hình của bộ ghi/tái tạo thông tin để đọc thông tin từ và ghi thông tin lên phương tiện ghi là đĩa quang.

Fig.21 thể hiện ví dụ về cấu hình của phương tiện ghi là đĩa quang.

Fig.22A thể hiện ví dụ về điện thoại di động.

Fig.22B là sơ đồ khối thể hiện ví dụ về cấu hình của điện thoại di động.

Fig.23 minh họa cấu trúc của dữ liệu được ghép kênh.

Fig.24 thể hiện dưới dạng sơ đồ cách thức mỗi dòng được ghép kênh trong dữ liệu được ghép kênh.

Fig.25 thể hiện cách thức dòng video được lưu trữ trong dòng gồm các gói PES một cách chi tiết hơn.

Fig.26 thể hiện cấu trúc của các gói TS và các gói nguồn trong dữ liệu được ghép kênh.

Fig.27 thể hiện cấu trúc dữ liệu của PMT.

Fig.28 thể hiện cấu trúc bên trong của thông tin dữ liệu được ghép kênh.

Fig.29 thể hiện cấu trúc bên trong của thông tin thuộc tính dòng.

Fig.30 thể hiện các bước để nhận dạng dữ liệu video.

Fig.31 thể hiện ví dụ về cấu hình của mạch tích hợp để thực hiện phương pháp mã hóa hình ảnh động và phương pháp giải mã hình ảnh động theo mỗi trong số các phương án.

Fig.32 thể hiện cấu hình để chuyển mạch giữa các tần số dẫn.

Fig.33 thể hiện các bước nhận dạng dữ liệu video và chuyển mạch giữa các tần số dẫn.

Fig.34 thể hiện ví dụ về bảng tìm kiếm trong đó các chuẩn dữ liệu video được kết hợp với các tần số dẫn.

Fig.35A là sơ đồ thể hiện ví dụ về cấu hình để dùng chung môđun của bộ xử lý tín hiệu.

Fig.35B là sơ đồ thể hiện ví dụ khác về cấu hình để dùng chung môđun của bộ xử lý tín hiệu.

Mô tả chi tiết sáng chế

Kiến thức cơ bản tạo nên nền tảng của sáng chế

Liên quan đến phương pháp mã hóa ảnh và phương pháp giải mã ảnh được mô tả trong phần tình trạng kỹ thuật của sáng chế, các tác giả đã thấy được vấn đề sau.

Thứ nhất, thiết bị mã hóa ảnh và thiết bị giải mã ảnh trong HEVC sẽ được mô tả.

Tín hiệu video được đưa vào thiết bị mã hóa ảnh là chuỗi các ảnh được gọi là các khung (các hình ảnh). Mỗi khung gồm có ma trận hai chiều của các điểm ảnh. Tất cả các chuẩn nêu trên dựa vào mã hóa video lai bao gồm phân chia mỗi khung video đơn lẻ thành các khối nhỏ hơn bao gồm các điểm ảnh. Kích cỡ của các khối có thể thay đổi, chẳng hạn, phù hợp với nội dung của ảnh. Phương pháp mã hóa có thể được thay đổi cụ thể trên cơ sở

mỗi khối. Kích cỡ lớn nhất có thể dùng cho khối như vậy, chẳng hạn trong HEVC, là 64 x 64 điểm ảnh. Nó được gọi là đơn vị mã hóa lớn nhất (largest coding unit - LCU). LCU có thể được phân chia theo cách đệ quy thành 4 CU.

Trong H.264/MPEG-4 AVC, khối macro (thường chỉ khối gồm 16 x 16 điểm ảnh) là phần tử ảnh cơ sở, mà việc mã hóa được thực hiện. Khối macro có thể còn được chia thành các khối con nhỏ hơn. Các bước mã hóa được bao gồm trong phương pháp mã hóa và/hoặc các bước giải mã được bao gồm trong phương pháp giải mã được thực hiện trên mỗi cơ sở khối con.

1-1. Mã hóa video lai

Phần dưới đây mô tả đơn giản kỹ thuật mã hóa video lai.

Diễn hình là, các bước mã hóa của kỹ thuật mã hóa video lai bao gồm dự đoán không gian và/hoặc thời gian (dự đoán không gian và/hoặc dự đoán thời gian). Theo đó, mỗi khối được mã hóa trước tiên được dự đoán nhờ sử dụng các khối trong vùng lân cận theo không gian hoặc các khối trong vùng lân cận theo thời gian của nó, nghĩa là, từ các khung video được mã hóa trước đó. Khối dư là chênh lệch giữa khối cần được mã hóa và kết quả dự đoán của nó sau đó được tính toán. Tiếp theo, khối dư được biến đổi từ miền không gian (điểm ảnh) thành miền tần số. Sự biến đổi hướng đến việc làm giảm sự tương quan của khối đầu vào.

Hơn nữa, các hệ số biến đổi thu được từ việc biến đổi được lượng tử hóa. Việc lượng tử hóa này là nén có tổn hao (không thể đảo ngược). Thông thường, các giá trị hệ số biến đổi đã nén còn được nén không tổn hao nhờ mã hóa entropy. Ngoài ra, thông tin hỗ trợ cần thiết cho việc tái tạo tín hiệu video mã hóa được mã hóa và được bố trí cùng với tín hiệu video mã hóa. Đây là ví dụ về thông tin về dự đoán không gian, dự đoán thời gian, và/hoặc lượng tử hóa.

1-2. Cấu hình của thiết bị mã hóa ảnh

Fig.1 là ví dụ cụ thể về thiết bị mã hóa ảnh H.264/MPEG-4 AVC và/hoặc HEVC (bộ mã hóa 100).

Như được thể hiện trên Fig.1, bộ mã hóa 100 gồm có bộ trừ 105, bộ biến đổi 110, bộ lượng tử hóa 120, bộ biến đổi ngược 130, bộ cộng 140, bộ lọc giải khối 150, bộ lọc vòng lặp thích ứng 160, bộ nhớ khung 170, bộ dự đoán 180, và bộ mã hóa entropy 190.

Bộ dự đoán 180 suy ra tín hiệu dự đoán s2 nhờ dự đoán thời gian hoặc dự đoán không gian. Loại dự đoán được sử dụng trong bộ dự đoán 180 có thể được thay đổi trên cơ sở mỗi khung hoặc trên cơ sở mỗi khối. Dự đoán thời gian được gọi là dự đoán liên khung, và dự đoán không gian được gọi là dự đoán nội khung. Việc mã hóa sử dụng tín hiệu dự đoán s2 nhờ dự đoán thời gian được gọi là mã hóa liên khung, và việc mã hóa sử dụng tín hiệu dự đoán s2 nhờ dự đoán không gian được gọi là mã hóa nội khung. Trong việc suy ra tín hiệu dự đoán sử dụng dự đoán thời gian, các ảnh được mã hóa lưu trữ trong bộ nhớ được sử dụng. Trong việc suy ra tín hiệu dự đoán sử dụng dự đoán không gian, giá trị điểm ảnh ở biên của khối lân cận được mã hóa hoặc được giải mã lưu trữ trong bộ nhớ được sử dụng. Số lượng chiều dự đoán trong dự đoán nội khung tùy thuộc vào kích cỡ của đơn vị mã hóa (coding unit - CU). Cần lưu ý rằng chi tiết về việc dự đoán sẽ được mô tả dưới đây.

Bộ trừ 105 trước tiên xác định sự chênh lệch (tín hiệu lỗi dự đoán e) giữa khối hiện thời cần được mã hóa của ảnh đầu vào (= tín hiệu đầu vào s1) và khối dự đoán tương ứng (= tín hiệu dự đoán s2). Chênh lệch này được sử dụng để dự đoán khối hiện thời cần được mã hóa. Cần lưu ý rằng tín hiệu lỗi dự đoán e cũng được gọi là tín hiệu dư dự đoán.

Bộ biến đổi 110 biến đổi tín hiệu lỗi dự đoán e thành các hệ số. Nói chung, bộ biến đổi 110 sử dụng phép biến đổi trực giao chẳng hạn như biến đổi cosin rời rạc (discrete cosine transformation - DCT) hai chiều hoặc kiểu số nguyên của nó. Biến đổi trực giao có thể làm giảm sự tương quan của tín hiệu đầu vào s1 (tín hiệu video trước khi mã hóa) một cách hữu hiệu. Sau khi biến đổi, các thành phần tần số thấp thường quan trọng hơn đối với chất lượng ảnh so với các thành phần tần số cao sao cho nhiều bit hơn có thể được dùng cho mã hóa các thành phần tần số thấp so với các thành phần tần số cao.

Bộ lượng tử hóa 120 lượng tử hóa các hệ số và suy ra các hệ số lượng tử hóa.

Bộ mã hóa entropy 190 thực hiện mã hóa entropy trên các hệ số lượng tử hóa. Các hệ số lượng tử hóa được nén không tổn hao nhờ mã hóa entropy. Hơn nữa, nhờ mã hóa entropy, khối lượng dữ liệu được lưu trữ trong bộ nhớ và khối lượng dữ liệu (dòng bit) được truyền có thể được giảm thêm. Mã hóa entropy được thực hiện nhờ chủ yếu ứng dụng việc mã hóa sử dụng từ mã độ dài thay đổi. Độ dài của từ mã được chọn dựa vào xác suất xuất hiện của nó.

Bộ mã hóa entropy 190 biến đổi ma trận hai chiều của các hệ số lượng tử hóa thành mảng một chiều. Điển hình là, bộ mã hóa entropy 190 thực hiện sự chuyển đổi này nhờ

việc quét gọi là quét theo hình chữ chi. Việc quét theo hình chữ chi bắt đầu với hệ số DC ở góc trên bên trái của mảng hai chiều và quét mảng hai chiều theo trình tự định trước kết thúc với hệ số AC ở góc dưới bên phải. Năng lượng được tập trung điển hình ở phần trên bên trái của ma trận hai chiều của các hệ số. Nói chung, khi các hệ số được bố trí ở góc trên bên trái, chúng là các hệ số thành phần tần số thấp. Khi các hệ số được bố trí ở góc dưới bên phải, chúng là các hệ số thành phần tần số cao. Do đó, việc quét theo hình chữ chi tạo ra mảng mà thường các giá trị cuối cùng là các số một hoặc các số không liên tiếp. Điều này cho phép mã hóa hiệu quả nhờ sử dụng các mã độ dài chạy trong khi/trước khi mã hóa entropy thực tế.

H.264/MPEG-4 AVC và HEVC sử dụng các loại mã hóa entropy khác nhau. Mặc dù một vài phần tử cú pháp được mã hóa với độ dài cố định, phần lớn phần tử cú pháp đều được mã hóa với các mã độ dài thay đổi. Cụ thể, trong số các cú pháp, các mã độ dài thay đổi thích ứng ngữ cảnh (context adaptive variable length code - CABAC) được sử dụng để mã hóa các tín hiệu lỗi dự đoán (các tín hiệu dư dự đoán). Nói chung, các mã số nguyên khác nhau khác với các mã độ dài thay đổi thích ứng ngữ cảnh được sử dụng để mã hóa các phần tử cú pháp khác. Tuy nhiên, mã hóa số học nhị phân thích ứng ngữ cảnh có thể được sử dụng.

Các mã độ dài thay đổi cho phép nén không tổn hao dòng bit được mã hóa. Tuy nhiên, vì các từ mã có độ dài thay đổi, nên việc giải mã phải được thực hiện theo thứ tự trên các từ mã. Nói cách khác, không thể mã hóa hoặc giải mã các từ mã trước khi mã hóa hoặc giải mã các từ mã trước đó mà không khởi động lại (khởi chạy) mã hóa entropy hoặc không chỉ báo tách biệt vị trí của từ mã (điểm bắt đầu) để bắt đầu với khi giải mã.

Mã hóa số học mã hóa chuỗi các bit thành từ mã đơn dựa vào mẫu xác suất định trước. Mẫu xác suất định trước được xác định theo nội dung của chuỗi video trong trường hợp CABAC. Mã hóa số học, và do đó cũng là CABAC, là hữu hiệu hơn khi độ dài của dòng bit được mã hóa là lớn hơn. Nói cách khác, CABAC được ứng dụng cho các chuỗi bit là hữu hiệu đối với các khối lớn hơn. Ở đầu mỗi chuỗi, CABAC được khởi động lại. Nói cách khác, ở đầu mỗi chuỗi video, mẫu xác suất của nó được khởi tạo với một vài giá trị định trước hoặc xác định trước.

Bộ mã hóa entropy 109 truyền, tới phía bộ giải mã, dòng bit bao gồm các hệ số lượng tử hóa được mã hóa (các tín hiệu video mã hóa) và thông tin hỗ trợ được mã hóa.

H.264/MPEG-4 và H.264/MPEG-4 AVC cũng như HEVC bao gồm hai lớp chức năng, lớp mã hóa video (Video Coding Layer - VCL) và lớp trừu tượng hóa mạng (Network Abstraction Layer - NAL). VCL cung cấp chức năng mã hóa như được nêu trên. NAL đóng gói các thành phần thông tin thành các đơn vị được chuẩn hóa được gọi là các đơn vị NAL theo ứng dụng bổ sung của chúng chẳng hạn như truyền qua kênh hoặc lưu trữ trong thiết bị lưu trữ. Các thành phần thông tin được đóng gói bởi NAL là, chẳng hạn, (1) tín hiệu lỗi dự đoán được mã hóa (dữ liệu video được nén) hoặc (2) thông tin khác cần thiết cho việc giải mã tín hiệu video chẳng hạn như loại dự đoán, thông số lượng tử hóa, các vector chuyển động, v.v.. Có các đơn vị NAL VCL chứa dữ liệu video được nén và thông tin liên quan, cũng như các đơn vị không phải VCL đóng gói dữ liệu bổ sung chẳng hạn như tập hợp thông số liên quan đến toàn bộ chuỗi video, hoặc thông tin nâng cao bổ sung (Supplemental Enhancement Information - SEI) cung cấp thông tin bổ sung mà có thể được sử dụng để nâng cao hiệu suất giải mã.

Một vài đơn vị NAL không phải VCL bao gồm, chẳng hạn, các tập hợp thông số. Tập hợp thông số là tập hợp các thông số liên quan đến việc mã hóa và giải mã phần nhất định của chuỗi video. Chẳng hạn, có tập hợp thông số chuỗi (sequence parameter set - SPS) gồm có thông số liên quan đến việc mã hóa và giải mã toàn bộ chuỗi các hình ảnh. Cụ thể, tập hợp thông số chuỗi là cấu trúc cú pháp bao gồm các phần tử cú pháp. Cụ thể, các phần tử cú pháp được ứng dụng cho không hoặc nhiều toàn bộ các chuỗi video được mã hóa như được xác định bởi nội dung của `seq_parameter_set_id` (nhận dạng tập hợp thông số chuỗi). `Seq_parameter_set_id` là phần tử cú pháp được bao gồm trong tập hợp thông số hình ảnh (được mô tả dưới đây) được tham chiếu đến bởi `pic_parameter_set_id`. `Pic_parameter_set_id` là phần tử cú pháp được bao gồm trong mỗi phần đầu lát.

Tập hợp thông số hình ảnh (picture parameter set - PPS) là tập hợp thông số mà định rõ các thông số được ứng dụng để mã hóa và giải mã hình ảnh trong chuỗi hình ảnh (chuỗi video). Cụ thể, PPS là cấu trúc cú pháp bao gồm các phần tử cú pháp. Các phần tử cú pháp được ứng dụng cho không hoặc nhiều toàn bộ các hình ảnh được mã hóa như được xác định bởi `pic_parameter_set_id` mà là phần tử cú pháp được thấy trong mỗi phần đầu lát.

Theo đó, đơn giản hơn khi duy trì theo dõi SPS so với PPS. Điều này là do PPS thay đổi đối với mỗi hình ảnh, trong khi đó SPS là không đổi đối với toàn bộ chuỗi video mà có thể thậm chí dài nhiều phút hoặc nhiều giờ.

Bộ mã hóa 100 gồm có bộ tái tạo (cũng được gọi là bộ giải mã) mà suy ra tín hiệu được tái tạo (cũng được gọi là tín hiệu được giải mã) s3. Nhờ bộ tái tạo, hình ảnh được tái tạo thu được bằng cách tái tạo (giải mã) ảnh được mã hóa được tạo ra và được lưu trữ trong bộ nhớ khung 170.

Bộ tái tạo gồm có bộ biến đổi ngược 130, bộ cộng 140, bộ lọc giải khối 150, và bộ lọc vòng lặp thích ứng 160.

Bộ biến đổi ngược 130, theo các bước mã hóa nêu trên, thực hiện lượng tử hóa ngược và biến đổi ngược. Cần lưu ý là tín hiệu lỗi dự đoán e' được suy ra từ bộ biến đổi ngược 130 là khác với tín hiệu lỗi dự đoán e do lỗi lượng tử hóa, cũng được gọi là nhiễu lượng tử hóa.

Bộ cộng 140 suy ra tín hiệu được tái tạo s' nhờ bổ sung tín hiệu lỗi dự đoán được tái tạo e' được tái tạo bởi bộ biến đổi ngược 130 vào tín hiệu dự đoán s2.

Bộ lọc giải khối 150 thực hiện xử lý lọc giải khối để làm giảm nhiễu lượng tử hóa mà chồng lên tín hiệu được tái tạo s' do lượng tử hóa. Ở đây, vì các bước mã hóa nêu trên được thực hiện trên cơ sở mỗi khối, có trường hợp ở đó biên khối là nhìn thấy được khi nhiễu được chồng lên (các đặc tính tạo khối của nhiễu). Nhiễu được chồng lên được gọi là nhiễu tạo khối. Cụ thể, khi lượng tử hóa mạnh được thực hiện bởi bộ lượng tử hóa 120, có nhiễu biên khối nhìn thấy được trong ảnh được tái tạo (ảnh được giải mã). Nhiễu tạo khối như vậy có ảnh hưởng tiêu cực đối với cảm nhận bằng mắt người, điều này có nghĩa là con người cảm thấy rằng chất lượng ảnh bị suy giảm. Để làm giảm nhiễu tạo khối, bộ lọc giải khối 150 thực hiện xử lý lọc giải khối trên mọi tín hiệu được tái tạo s' (khối được tái tạo).

Chẳng hạn, trong xử lý lọc giải khối của H.264/MPEG-4 AVC, đối với mỗi vùng, xử lý lọc thích hợp cho vùng được lựa chọn. Trong trường hợp mức nhiễu tạo khối cao, bộ lọc thông thấp (băng hẹp) mạnh được áp dụng, trong khi đó đối với mức độ nhiễu tạo khối thấp, bộ lọc thông thấp (băng rộng) yếu hơn được áp dụng. Cường độ của bộ lọc thông thấp được xác định bởi tín hiệu dự đoán e2 và bởi tín hiệu lỗi dự đoán e'. Xử lý lọc giải khối thường làm trơn các mép khối. Điều này dẫn đến việc nâng cao chất lượng ảnh chủ quan của các tín hiệu được giải mã. Ảnh đã lọc được sử dụng cho dự đoán bù chuyển động của ảnh tiếp theo. Vì xử lý lọc cũng làm giảm các lỗi dự đoán, hiệu quả mã hóa có thể được nâng cao.

Bộ lọc vòng lặp thích ứng 160 ứng dụng xử lý độ lệch thích ứng mẫu (sample

adaptive offset - SAO) và/hoặc xử lý lọc vòng lặp thích ứng (adaptive loop filter - ALF) cho hình ảnh được tái tạo s'' sau khi xử lý lọc giải khối trong bộ lọc giải khối 150, để suy ra tín hiệu được tái tạo (tín hiệu được giải mã) s_3 .

Xử lý lọc giải khối trong bộ lọc giải khối 150 hướng đến mục đích nâng cao chất lượng chủ quan. Trong khi đó, xử lý ALF và xử lý SAO trong bộ lọc vòng lặp thích ứng 160 hướng đến mục đích nâng cao độ trung thực theo điểm ảnh (chất lượng “khách quan”). Xử lý SAO được sử dụng để bổ sung giá trị độ lệch vào giá trị điểm ảnh đối với mỗi điểm ảnh bằng cách sử dụng giá trị điểm ảnh của điểm ảnh ngay sát. Xử lý ALF được sử dụng để bù biến dạng ảnh gây ra bởi việc nén. Điển hình là, bộ lọc được sử dụng trong xử lý ALF là bộ lọc Wiener với các hệ số lọc được xác định sao cho sai số bình phương trung bình (mean square error - MSE) giữa tín hiệu được tái tạo s' và tín hiệu đầu vào s_1 được giảm thiểu. Các hệ số lọc của xử lý ALF được tính toán và được truyền trên cơ sở từng khung chẳng hạn. Xử lý ALF có thể được áp dụng cho toàn bộ khung (ảnh) hoặc tới các vùng cục bộ (các khối). Thông tin hỗ trợ chỉ báo các vùng nào được lọc có thể được truyền trên cơ sở từng khối, cơ sở từng khung, hoặc cơ sở từng cây tứ phân.

Bộ nhớ khung (bộ đệm khung) 170 lưu trữ một phần của ảnh được mã hóa và được tái tạo (được giải mã) (tín hiệu được tái tạo s_3). Hình ảnh được tái tạo đã lưu trữ được sử dụng để giải mã khối được mã hóa liên khung.

Bộ dự đoán 180 suy ra tín hiệu dự đoán s_2 sử dụng (cùng) tín hiệu mà có thể được sử dụng ở cả phía bộ mã hóa và phía bộ giải mã, để duy trì khả năng tương thích giữa phía bộ mã hóa và phía bộ giải mã. Tín hiệu mà có thể được sử dụng ở cả phía bộ mã hóa và phía bộ giải mã là tín hiệu được tái tạo s_3 (tín hiệu video sau khi xử lý lọc bởi bộ lọc vòng lặp thích ứng 160) ở phía bộ mã hóa mà được mã hóa và sau đó được tái tạo (được giải mã), và tín hiệu được tái tạo s_4 (tín hiệu video sau khi xử lý lọc bởi bộ lọc vòng lặp thích ứng trên Fig.2) ở phía bộ giải mã mà được giải mã từ dòng bit.

Bộ dự đoán 180, khi tạo ra tín hiệu dự đoán s_2 nhờ mã hóa liên khung, dự đoán sử dụng dự đoán bù chuyển động. Bộ ước lượng chuyển động của bộ dự đoán 180 (không được minh họa) tìm kiếm khối tương hợp nhất dùng cho khối hiện thời từ các khối nằm trong các khung video được mã hóa và được tái tạo trước đó. Khối tương hợp nhất sau đó trở thành tín hiệu dự đoán. Sự dịch chuyển (chuyển động) tương ứng giữa khối hiện thời và khối tương hợp nhất của nó sau đó được báo hiệu dưới dạng dữ liệu chuyển động được

bao gồm trong thông tin hỗ trợ ở dạng các vector chuyển động ba chiều. Tín hiệu được truyền cùng với dữ liệu video được mã hóa. Vector chuyển động ba chiều gồm có vector chuyển động hai chiều theo không gian và vector chuyển động một chiều theo thời gian. Để tối ưu hóa độ chính xác dự đoán, các vector chuyển động có thể được xác định với độ phân giải dưới điểm ảnh theo không gian, ví dụ, độ phân giải nửa điểm ảnh hoặc một phần tư điểm ảnh. Vector chuyển động với độ phân giải dưới điểm ảnh theo không gian có thể trở vào vị trí không gian nằm trong khung đã được tái tạo ở đó không có giá trị điểm ảnh thực, nghĩa là, vị trí dưới điểm ảnh. Vì vậy, phép nội suy không gian của các giá trị điểm ảnh như vậy được cần đến để thực hiện dự đoán bù chuyển động. Điều này có thể đạt được bởi bộ lọc nội suy (được tích hợp trong bộ dự đoán 180 trên Fig.1).

1-3. Cấu hình của thiết bị giải mã ảnh

Cấu hình của bộ giải mã (thiết bị giải mã ảnh) sẽ được mô tả dựa vào Fig.2.

Fig.2 là sơ đồ khối thể hiện ví dụ về bộ giải mã 200 theo chuẩn mã hóa video H.264/MPEG-4 AVC hoặc HEVC.

Như được thể hiện trên Fig.2, bộ giải mã 200 gồm có bộ giải mã entropy 290, bộ biến đổi ngược 230, bộ cộng 240, bộ lọc giải khối 250, bộ lọc vòng lặp thích ứng 260, bộ nhớ khung 270, và bộ dự đoán 280.

Dòng bit được đưa vào bộ giải mã 200 (tín hiệu video mã hóa) trước tiên được truyền tới bộ giải mã entropy 290.

Bộ giải mã entropy 290 trích các hệ số lượng tử hóa được mã hóa từ dòng bit và thông tin hỗ trợ được mã hóa, và giải mã các hệ số lượng tử hóa được mã hóa và thông tin hỗ trợ được mã hóa. Thông tin hỗ trợ, như được mô tả ở trên, là thông tin cần thiết để giải mã chẳng hạn như dữ liệu chuyển động (vector chuyển động) và chế độ dự đoán (loại dự đoán).

Bộ giải mã entropy 290 biến đổi các hệ số lượng tử hóa được giải mã trong mảng một chiều thành các hệ số trong mảng hai chiều nhờ quét ngược. Bộ giải mã entropy 290 đưa vào, tới bộ biến đổi ngược 230, các hệ số lượng tử hóa sau khi được biến đổi thành các hệ số trong mảng hai chiều.

Bộ biến đổi ngược 230 thực hiện lượng tử hóa ngược và biến đổi ngược trên các hệ số lượng tử hóa được biến đổi thành các hệ số trong mảng hai chiều, để suy ra tín hiệu lỗi

dự đoán e' . Tín hiệu lỗi dự đoán e' tương ứng với các chênh lệch thu được bằng cách trừ tín hiệu dự đoán từ tín hiệu được đưa vào bộ mã hóa trong trường hợp không có nhiều lượng tử hóa được đưa vào và không có lỗi xuất hiện.

Bộ dự đoán 280 suy ra tín hiệu dự đoán s_2 nhờ dự đoán thời gian hoặc dự đoán không gian. Thông tin chẳng hạn như loại dự đoán được bao gồm trong thông tin hỗ trợ được sử dụng trong trường hợp dự đoán nội khung (dự đoán không gian). Hơn nữa, thông tin chẳng hạn như dữ liệu chuyển động được bao gồm trong thông tin hỗ trợ được sử dụng trong trường hợp dự đoán bù chuyển động (dự đoán liên khung, dự đoán thời gian).

Bộ cộng 240 cộng tín hiệu lỗi dự đoán e' thu được từ bộ biến đổi ngược 230 và tín hiệu dự đoán e_2 thu được từ bộ dự đoán 280, để suy ra tín hiệu được tái tạo s' .

Bộ lọc giải khối 250 thực hiện xử lý lọc giải khối trên tín hiệu được tái tạo s' . Bộ lọc vòng lặp thích ứng 260 áp dụng xử lý SAO và xử lý ALF cho tín hiệu được tái tạo s' mà ở đó xử lý lọc giải khối được áp dụng bởi bộ lọc giải khối 250. Tín hiệu được giải mã S_4 thu được từ việc áp dụng xử lý SAO và xử lý ALF trong bộ lọc vòng lặp thích ứng 260 được lưu trữ trong bộ nhớ khung 270. Tín hiệu được giải mã S_4 lưu trữ trong bộ nhớ khung 270 được sử dụng, trong bộ dự đoán 280, để dự đoán khối hiện thời tiếp theo sẽ được giải mã hoặc ảnh hiện thời sẽ được giải mã.

1-4. Hiệu quả xử lý

Nói chung, việc xử lý song song được xét đến để nâng cao hiệu quả xử lý của xử lý mã hóa và xử lý giải mã.

So với H.264/MPEG-4 AVC, HEVC có chức năng để hỗ trợ việc xử lý song song mức cao (xử lý song song) trong việc mã hóa và giải mã. Trong HEVC, có thể chia khung thành các lát, tương tự với H.264/MPEG-4 AVC. Ở đây, các lát là các nhóm của các LCU theo thứ tự quét. Trong H.264/MPEG-4 AVC, các lát có thể giải mã được theo cách độc lập, và không có dự đoán không gian được áp dụng giữa các lát. Do đó, việc xử lý song song có thể được thực hiện trên cơ sở từng lát.

Tuy nhiên, vì các lát có các phần đầu lớn đáng kể và thiếu sự phụ thuộc giữa các lát, nên hiệu quả của việc nén bị giảm. Hơn nữa, mã hóa CABAC làm giảm hiệu quả khi được áp dụng cho các khối dữ liệu nhỏ.

Để cho phép xử lý song song hữu hiệu hơn, sáng chế đề xuất xử lý song song mặt

sóng (wavefront parallel processing - WPP). WPP giữ mức phụ thuộc không đổi, khác với xử lý song song trong đó mỗi lát trong số các lát là độc lập.

Phần mô tả dưới đây sẽ được thực hiện dựa vào trường hợp ở đó hình ảnh bao gồm các LCU trong đó các hình ảnh được bố trí trong ma trận và mỗi hàng LCU bao gồm một lát (tham khảo Fig.3). Trong WPP, trong số các LCU bao gồm hàng LCU hiện thời 32, như mẫu xác suất CABAC dùng để thiết lập lại trạng thái CABAC của LCU thứ nhất (đầu LCU), mẫu xác suất CABAC ngay sau khi xử lý trên LCU thứ hai của hàng LCU trước đó 31 kết thúc được sử dụng. Tất cả các phần phụ thuộc liên khối được duy trì. Điều này cho phép giải mã song song các hàng LCU. Thời điểm để bắt đầu mỗi quy trình xử lý hàng LCU được làm trễ bởi hai LCU đối với LCU trước đó. Thông tin về các điểm bắt đầu để bắt đầu giải mã hàng LCU được bao gồm trong phần đầu lát. WPP được mô tả chi tiết trong Tài liệu phi sáng chế 1.

Cách khác để nâng cao việc song song hóa được gọi là các ô. Theo đó, khung (hình ảnh) được phân chia thành các ô. Các ô là các nhóm hình chữ nhật của các LCU. Các biên giữa các ô được thiết lập sao cho toàn bộ hình ảnh được phân chia trong ma trận. Các ô được xử lý theo thứ tự quét màn hình.

Tất cả các phần phụ thuộc bị vỡ ở các biên ô. Mã hóa entropy chẳng hạn như CABAC cũng được thiết lập lại ở đầu mỗi ô. Chỉ xử lý lọc giải khối và xử lý độ lệch thích ứng mẫu có thể được áp dụng trên các biên ô. Do đó, các ô có thể được mã hóa và được giải mã song song. Các ô được mô tả chi tiết trong Tài liệu phi sáng chế 2 và Tài liệu phi sáng chế 3.

Hơn nữa, để nâng cao khái niệm của các lát và làm cho nó thích hợp hơn đối với song song hóa hơn là khả năng phục hồi lỗi mà là mục đích ban đầu của các lát trong H.264/MPEG-4 AVC, sáng chế đề xuất khái niệm về các lát phụ thuộc và các lát entropy.

Nói cách khác, trong HEVC, có ba loại lát được hỗ trợ: (1) các lát thông thường; (2) các lát entropy; và (3) các lát phụ thuộc.

Các lát thông thường thể hiện các lát đã được biết từ H.264/MPEG-4 AVC. Không có dự đoán không gian được cho phép giữa các lát thông thường. Nói cách khác, việc dự đoán trên các biên lát không được cho phép. Điều này có nghĩa là lát thông thường được mã hóa mà không tham chiếu đến bất kỳ lát nào khác. Để cho phép giải mã độc lập các lát như vậy, CABAC được khởi động lại ở đầu mỗi lát.

Khi lát cần được xử lý là lát thông thường, việc khởi động lại CABAC gồm có việc kết thúc xử lý (chấm dứt xử lý) xử lý mã hóa số học hoặc xử lý giải mã số học ở cuối lát trước, và quy trình xử lý khởi tạo bảng ngữ cảnh (bảng xác suất) bằng giá trị mặc định ở đầu lát thông thường.

Các lát thông thường được sử dụng ở đầu mỗi khung. Nói cách khác, mọi khung phải bắt đầu với lát thông thường. Lát thông thường có phần đầu bao gồm các thông số cần thiết để giải mã dữ liệu lát.

Thuật ngữ “lát entropy” ký hiệu lát trong đó dự đoán không gian được cho phép giữa lát cha và lát entropy. Việc phân tích cú pháp lát cha và lát entropy được thực hiện độc lập.

Tuy nhiên, lát cha là, chẳng hạn, lát thông thường ngay trước lát entropy. Lát cha được yêu cầu cho việc tái tạo các giá trị điểm ảnh của lát entropy. Để cho phép phân tích cú pháp độc lập các lát entropy, CABAC cũng được khởi động lại ở đầu lát. Đối với phần đầu lát của các lát entropy, có thể sử dụng phần đầu lát mà ngắn hơn so với phần đầu lát của lát thông thường. Phần đầu lát của các lát entropy gồm có tập hợp con của các thông số mã hóa đối với thông tin được truyền nằm trong phần đầu của lát thông thường. Các thành phần thiếu trong phần đầu của lát entropy được sao chép từ phần đầu của lát cha.

Khi lát cần được xử lý là lát entropy, việc khởi động lại CABAC, như tương tự với lát thông thường, gồm có bước kết thúc xử lý (chấm dứt xử lý) ở cuối lát trước, và quy trình xử lý khởi tạo bảng ngữ cảnh (bảng xác suất) bằng giá trị mặc định ở đầu lát hiện thời.

(3) Lát phụ thuộc là tương tự với lát entropy, nhưng khác nhau một phần trong quy trình xử lý trong đó CABAC được khởi động lại.

Khi lát cần được xử lý là lát phụ thuộc và WPP là không hiệu quả, việc khởi động lại CABAC gồm có bước kết thúc xử lý (chấm dứt xử lý) ở lát trước và quy trình xử lý khởi tạo bảng ngữ cảnh bằng giá trị trạng thái ở cuối của lát trước. Khi lát cần được xử lý là lát phụ thuộc và WPP là không hiệu quả, việc khởi động lại CABAC gồm có kết thúc xử lý (chấm dứt xử lý) ở lát trước và quy trình xử lý khởi tạo bảng ngữ cảnh bằng giá trị trạng thái sau khi xử lý LCU mà thuộc lát trước và là thứ hai tính từ đầu bên trái ở đầu lát hiện thời.

Như được nêu trên, việc khởi động lại CABAC luôn gồm có xử lý kết thúc. Ngược lại, trong khi khởi động lại CABAC, trạng thái của CABAC là thường được chuyển qua.

Các lát phụ thuộc không thể được phân tích cú pháp mà không có lát cha. Do đó, các lát phụ thuộc không thể được giải mã khi không nhận được lát cha. Lát cha thường là lát trước của các lát phụ thuộc theo thứ tự mã hóa, và lát bao gồm phần đầu lát đầy đủ. Điều này tương tự đối với lát cha của lát entropy.

Như được nêu trên, các lát phụ thuộc và các lát entropy sử dụng phần đầu lát (cụ thể, thông tin của phần đầu lát mà thiếu trong phần đầu của lát phụ thuộc) của lát ngay trước theo thứ tự mã hóa của các lát. Quy tắc này được ứng dụng theo cách đệ quy. Lát cha mà lát phụ thuộc hiện thời phụ thuộc vào được ghi nhận là có sẵn để tham chiếu. Việc tham chiếu bao gồm việc sử dụng dự đoán không gian giữa các lát, dùng chung các trạng thái CABAC, và tương tự. Lát phụ thuộc sử dụng các bảng ngữ cảnh CABAC được tạo ra ở cuối của lát ngay trước đó. Do đó, lát phụ thuộc không khởi tạo các bảng CABAC bằng các giá trị mặc định, mà, thay vào đó vẫn sử dụng các bảng ngữ cảnh đã được phát triển. Các phần mô tả chi tiết hơn liên quan đến các lát entropy và các lát phụ thuộc có thể được thấy trong Tài liệu phi sáng chế 3.

HEVC cung cấp một vài biên dạng. Biên dạng gồm có một vài thiết lập của thiết bị mã hóa ảnh và thiết bị giải mã ảnh thích hợp cho ứng dụng cụ thể. Chẳng hạn, “biên dạng chính” chỉ gồm có các lát thông thường và phụ thuộc, nhưng không gồm có các lát entropy.

Như được nêu trên, các lát được mã hóa được đóng gói thêm thành các đơn vị NAL, mà được đóng gói thêm, chẳng hạn, thành các gói giao thức thời gian thực (Real Time Protocol - RTP) và cuối cùng thành các gói giao thức Internet (Internet Protocol - IP). Hoặc các ngăn xếp giao thức này hay các ngăn xếp giao thức khác, cho phép truyền video được mã hóa trong các mạng định hướng gói, chẳng hạn như Internet hoặc một vài mạng độc quyền khác.

Các mạng thường bao gồm ít nhất một hoặc nhiều các bộ định tuyến, mà ứng dụng phần cứng đặc biệt thao tác rất nhanh. Chức năng của bộ định tuyến là để nhận các gói IP, phân tích các phần đầu gói IP và, theo đó, để chuyển tiếp các gói IP tới các đích tương ứng của chúng. Vì các bộ định tuyến cần xử lý lưu lượng dữ liệu từ nhiều nguồn, logic xử lý gói cần càng đơn giản càng tốt. Yêu cầu tối thiểu đối với bộ định tuyến là kiểm tra trường địa chỉ đích trong phần đầu IP để xác định lộ trình nào để thực hiện việc chuyển tiếp chúng.

Để hỗ trợ thêm cho chất lượng dịch vụ (QoS), các bộ định tuyến thông minh (nhận biết phương tiện) kiểm tra thêm các trường đặc biệt trong các phần đầu giao thức mạng, chẳng hạn như phần đầu IP, phần đầu RTP, và ngay cả phần đầu của NALU.

Như có thể được thấy từ phần mô tả nêu trên về mã hóa video, các loại lát khác nhau được định rõ cho mục đích xử lý song song, chẳng hạn như các lát phụ thuộc và các lát entropy, là có tầm quan trọng khác nhau đối với sự suy giảm chất lượng khi có sự hư hỏng của chúng. Cụ thể, các lát phụ thuộc không thể được phân tích cú pháp và được giải mã mà không có lát cha. Điều này là do ở đầu lát phụ thuộc, bộ mã hóa entropy hoặc bộ giải mã entropy không thể được khởi động lại. Theo đó, lát cha là quan trọng hơn đối với việc tái tạo hình ảnh hoặc video.

Trong HEVC, các lát phụ thuộc và các lát entropy đưa vào chiều phụ thuộc bổ sung, chẳng hạn như, phần phụ thuộc liên lát (phần phụ thuộc trong khung). Loại phụ thuộc này không được xét đến bởi các bộ định tuyến.

Các tính phụ thuộc nêu trên và, cụ thể, phụ thuộc liên lát không được xét đến ở mức mạng. Tuy nhiên, có thể mong muốn đưa phần phụ thuộc nêu trên vào xem xét ở mức mạng để cung cấp hỗ trợ tốt hơn cho chất lượng dịch vụ. Theo đó, cần nâng cao tính linh hoạt xử lý gói ở mức mạng bằng cách xem xét tính phụ thuộc của các lát.

Mô tả chi tiết vấn đề

1-5. WPP và lát phụ thuộc

Các lát phụ thuộc có thể được sử dụng cùng với các công cụ xử lý song song chẳng hạn như xử lý song song mặt sóng (WPP) và các ô. Cụ thể, các lát phụ thuộc khiến mặt sóng (dòng phụ) có khả năng giảm độ trễ cuộc truyền mà không gây nên tổn hao mã hóa.

Hơn nữa, các lát phụ thuộc dùng làm các điểm bắt đầu đối với các dòng phụ CABAC vì CABAC không được khởi động lại ở các lát phụ thuộc. Hơn nữa, thông tin chỉ báo các điểm bắt đầu có thể được truyền trong dòng bit để bố trí các điểm bắt đầu để có thể phân tích cú pháp độc lập. Cụ thể, nếu nhiều hơn hai dòng phụ CABAC được đóng gói ở lát thông thường hoặc phụ thuộc, các điểm bắt đầu được báo hiệu rõ ràng ở dạng số lượng byte trên mỗi dòng phụ. Ở đây, dòng phụ chỉ báo một phần của dòng mà có thể phân tích cú pháp độc lập dựa vào các điểm bắt đầu. Ngoài ra, các lát phụ thuộc có thể được sử dụng như “các dấu mốc” điểm bắt đầu, vì mọi lát phụ thuộc cần có phần đầu đơn vị NAL. Điều

này có nghĩa là các điểm bắt đầu có thể được báo hiệu đối với các dấu mốc như vậy.

Hai phương pháp tiếp cận, cụ thể là việc báo hiệu điểm bắt đầu rõ ràng và việc đánh dấu các điểm bắt đầu qua các lát phụ thuộc được sử dụng kết hợp.

Theo quy tắc, điểm bắt đầu của mỗi đơn vị NAL (bắt đầu của mỗi phần đầu NAL) phải có thể nhận dạng được. Không có yêu cầu về thao tác nhận dạng chính xác. Ví dụ, hai phương pháp dưới đây có thể được áp dụng.

Phương pháp thứ nhất là phương pháp đặt mã khởi động (chẳng hạn, dài 3 byte) ở đầu mỗi phần đầu NAL. Phương pháp thứ hai là phương pháp đưa mỗi đơn vị NAL vào trong gói riêng. Do phân phụ thuộc của các lát, kích thước phần đầu lát có thể giảm.

Liên quan đến các lát entropy, phương pháp cho phép phân tích cú pháp CABAC song song. Điều này là do CABAC được khởi động lại thực tế ở đầu các lát entropy. Trong trường hợp xử lý song song của CABAC, CABAC là điểm nghẽn mà có thể được khắc phục bởi phân tích cú pháp CABAC song song được theo sau bởi các thao tác tái tạo điểm ảnh theo trình tự. Cụ thể, công cụ song song hóa WPP cho phép giải mã mỗi hàng LCU bởi một lõi xử lý (mã sở hữu trí tuệ (mã IP), khối chức năng). Cần lưu ý rằng việc gán các hàng LCU cho các lõi có thể khác nhau. Chẳng hạn, hai hàng có thể được gán cho một lõi, và một hàng có thể được gán cho hai lõi.

Fig.3 là sơ đồ thể hiện ví dụ về cấu hình của hình ảnh 300. Trên Fig.3, hình ảnh 300 được chia nhỏ thành từ 31 đến 3m (m là số thứ tự của LCU) hàng của các đơn vị mã hóa lớn nhất (LCU). Mỗi trong số các hàng LCU 3i (I = 1 đến m) bao gồm các LCU từ 3i1 đến 3in (n là số thứ tự của cột LCU) mà được bố trí trong một hàng. Hàng LCU 3i tương ứng với “Mặt sóng i”. Xử lý song song có thể được thực hiện đối với các mặt sóng. Mũi tên của trạng thái CABAC trên Fig.3 chỉ báo mối tương quan giữa LCU đề cập đến trạng thái CABAC và đích tham chiếu.

Cụ thể là, trên Fig.3, trước tiên, trong số các LCU được bao gồm trong hàng LCU 31, quy trình xử lý (mã hóa hoặc giải mã) bắt đầu đối với LCU đầu 311. Quy trình xử lý trên các LCU được thực hiện theo thứ tự từ LCU 311 đến 31n. Sau khi quy trình xử lý trên hai LCU thứ nhất 311 và 312 ở hàng LCU 31 được thực hiện, quy trình xử lý trên hàng LCU 32 được bắt đầu. Trong quy trình xử lý của hàng LCU thứ nhất 321 của cột LCU 32, như được thể hiện ở mũi tên về trạng thái CABAC trên Fig.3, trạng thái CABAC ngay sau khi quy trình xử lý trên LCU 312 ở hàng LCU 31 trong hàng thứ nhất được sử dụng làm

trạng thái ban đầu của trạng thái CABAC. Nói cách khác, có độ trễ bằng hai LCU giữa hai quy trình xử lý song song.

Fig.4 là sơ đồ thể hiện ví dụ về trường hợp ở đó lát phụ thuộc sử dụng WPP được sử dụng. Các hàng LCU từ 41 đến 43 lần lượt tương ứng với “Mặt sóng 1”, “Mặt sóng 2” và Mặt sóng 3”. Các hàng LCU từ 41 đến 43 được xử lý bởi các lõi độc lập tương ứng của chúng. Trên Fig.4, hàng LCU 41 là lát thông thường, và các hàng LCU từ 42 đến 4m là các lát phụ thuộc.

Các lát phụ thuộc khiến WPP có khả năng giảm trễ. Các lát phụ thuộc không có phần đầu lát đầy đủ. Hơn nữa, các lát phụ thuộc có thể được giải mã độc lập so với các lát khác miễn là các điểm bắt đầu (hoặc điểm bắt đầu của các lát phụ thuộc, mà được biết là quy tắc như được nêu trên) được biết đến. Cụ thể, các lát phụ thuộc có thể khiến WPP cũng thích hợp đối với các ứng dụng trễ thấp mà không xảy ra tổn hao mã hóa.

Trong trường hợp thông thường về đóng gói các dòng phụ (các hàng LCU) thành các lát, bắt buộc chèn các điểm bắt đầu rõ ràng vào trong phần đầu lát để đảm bảo mã hóa và giải mã entropy song song. Kết quả là lát là sẵn sàng cho việc truyền chỉ sau khi dòng phụ cuối cùng của lát được mã hóa hoàn toàn. Phần đầu lát được kết thúc chỉ sau khi việc mã hóa của tất cả của các dòng phụ ở lát được hoàn thành. Điều này có nghĩa là việc truyền phần đầu lát không thể được bắt đầu qua phân đoạn gói ở lớp RTP/IP cho đến khi toàn bộ lát được kết thúc.

Tuy nhiên vì các lát phụ thuộc có thể được sử dụng làm các dấu mốc điểm bắt đầu, việc báo hiệu điểm bắt đầu rõ ràng không được yêu cầu. Do đó, có thể phân chia lát thông thường thành nhiều lát phụ thuộc mà không có tổn hao mã hóa. Các lát phụ thuộc có thể được truyền ngay khi việc mã hóa của dòng phụ được đóng gói được hoàn thành (hoặc thậm chí sớm hơn trong trường hợp phân đoạn gói).

Các lát phụ thuộc không phá vỡ phần phụ thuộc dự đoán không gian. Các lát phụ thuộc thậm chí không phá vỡ phần phụ thuộc phân tích cú pháp. Điều này là do thông thường việc phân tích cú pháp lát phụ thuộc hiện thời yêu cầu các trạng thái CABAC từ lát trước đó.

Khi các lát phụ thuộc không được cho phép, thì mỗi hàng LCU có thể được tạo cấu hình là lát. Cấu hình như vậy làm giảm độ trễ truyền, nhưng đồng thời, dẫn tới tổn hao mã hóa khá cao như được mô tả trong phần tình trạng kỹ thuật của sáng chế nêu trên.

Theo cách khác, toàn bộ khung (hình ảnh) được đóng gói thành lát đơn. Trong trường hợp này, các điểm bắt đầu đối với các dòng phụ (các hàng LCU) cần được báo hiệu trong phần đầu lát để cho phép phân tích cú pháp song song. Kết quả là, có độ trễ truyền ở mức khung. Nói cách khác, phần đầu cần được sửa đổi sau khi toàn bộ khung được mã hóa. Việc toàn bộ hình ảnh được đóng gói trong lát đơn không tự làm tăng độ trễ truyền. Ví dụ, việc truyền một vài phần của lát có thể đã bắt đầu trước khi toàn bộ việc mã hóa kết thúc. Tuy nhiên, nếu WPP được sử dụng, thì phần đầu lát cần được sửa đổi sau để ghi các điểm bắt đầu. Do đó, toàn bộ lát cần được làm trễ để truyền.

Việc sử dụng của các lát phụ thuộc nhờ đó cho phép giảm độ trễ. Như được thể hiện trên Fig.4, hình ảnh 400 được chia thành hàng LCU 41 là lát thông thường, và các hàng LCU từ 42 đến 4m là các lát phụ thuộc. Khi mỗi hàng LCU là một lát phụ thuộc, độ trễ truyền bằng một hàng LCU có thể đạt được mà không có bất kỳ tổn hao mã hóa nào. Điều này là do thực tế là các lát phụ thuộc không phá vỡ bất kỳ phần phụ thuộc theo không gian nào và không khởi động lại công cụ CABAC.

1-6. Cấu hình của gói

Như được nêu trên, các bộ định tuyến mạng phải phân tích các phần đầu của các gói để cho phép cung cấp chất lượng dịch vụ. Chất lượng dịch vụ là khác nhau theo loại ứng dụng và/hoặc quyền ưu tiên của dịch vụ và/hoặc của sự liên quan của gói đối với sự biến dạng gây ra bởi tổn hao gói của nó.

Fig.5 là sơ đồ thể hiện ví dụ về việc đóng gói (tạo gói) dòng bit.

Nói chung, giao thức thời gian thực (RTP) được sử dụng cho việc tạo gói. RTP thường được sử dụng cho việc truyền phương tiện thời gian thực. Các độ dài phần đầu của các giao thức liên quan tương ứng về cơ bản được cố định. Các phần đầu giao thức có các trường mở rộng. Các trường mở rộng có thể mở rộng độ dài của các phần đầu 4 byte. Ví dụ, phần đầu IP có thể được mở rộng lên đến 20 byte. Các phần tử cú pháp trong các phần đầu của IP, giao thức gói dữ liệu người dùng (UDP) và RTP cũng được cố định theo độ dài của chúng.

Fig.5 thể hiện phần đầu gói 500 được bao gồm trong gói IP. Phần đầu gói được thể hiện trên Fig.5 gồm có phần đầu IP 510, phần đầu UDP 530, phần đầu RTP 540, phần đầu tải tin RTP H264 560, và phần đầu NAL 570. Phần đầu IP 510 là phần đầu với độ dài 20 byte với trường mở rộng 520 là 4 byte. Tải tin của gói IP là gói UDP. Gói UDP gồm có

phần đầu UDP 530 với độ dài 8 byte và tải tin UDP. Tải tin UDP được tạo nên bởi gói RTP. Gói RTP gồm có phần đầu RTP 540 với độ dài phần đầu là 12 byte và trường mở rộng 550 là 4 byte. Gói RTP có thể được mở rộng có chọn lọc bởi trường mở rộng. Tải tin của gói RTP gồm có phần đầu tải tin RTP H264 đặc biệt 560 với độ dài từ 0 đến 3 byte được theo sau bởi phần đầu NAL 570 của HEVC có độ dài 2 byte. Tải tin của NALU bao gồm gói video được mã hóa theo sau các phần đầu gói 500 (không được thể hiện trên Fig.5).

Các bộ định tuyến có khả năng cung cấp chất lượng dịch vụ nâng cao được gọi là các phần tử mạng nhận biết phương tiện (Media Aware Network Element - MANE). Các phần tử mạng nhận biết phương tiện kiểm tra một vài trường của các phần đầu gói được thể hiện trên Fig.5. Ví dụ, MANE được gọi là “temporal_id (id theo thời gian)” và được bao gồm trong phần đầu NAL 570 hoặc số thứ tự giải mã được bao gồm trong phần đầu RTP 540 có thể được kiểm tra để phát hiện các tổn hao và thứ tự trình diễn của các nội dung gói nhận được. Các bộ định tuyến (các phần tử mạng) xử lý các gói càng nhanh càng tốt để cho phép thông lượng cao trong mạng. Logic được yêu cầu để truy cập các trường trong các phần đầu gói nhanh và đơn giản, để giữ độ phức tạp của quá trình xử lý phần tử mạng ở mức thấp.

NALU được đóng gói bởi phần đầu 500. NALU có thể bao gồm dữ liệu lát khi phần đầu lát có mặt.

Fig.6 là sơ đồ thể hiện ví dụ về cú pháp phần đầu lát 600. Phần tử cú pháp `dependent_slice_flag` 601 là phần tử cú pháp chỉ báo xem lát là lát phụ thuộc hay không. Phần tử cú pháp này có thể được sử dụng để nhận dạng phần phụ thuộc liên lát. Tuy nhiên, phần đầu lát là nội dung của NALU. Việc phân tích cú pháp các phần tử cú pháp trước `dependent_slice_flag` 601 yêu cầu logic khá phức tạp. Đây là mức mà không thể được coi là hiệu quả bởi các bộ định tuyến thông thường như sẽ được thể hiện dưới đây.

Như được nêu trên, NALU gồm có thông tin chung dùng cho các lát chẳng hạn như các tập hợp thông số, hoặc gồm có các lát được mã hóa trực tiếp với thông tin cần thiết để giải mã được bao gồm trong phần đầu lát. Cú pháp của phần đầu lát được sử dụng cho lát entropy hoặc lát phụ thuộc được minh họa trên Fig.6. Fig.6 thể hiện bảng với cấu trúc phần đầu lát. Khi phần tử cú pháp “`dependent_slice_flag`” được đặt bằng 1, tất cả các lát đến lát thông thường thứ nhất (lát mà không phải lát entropy và không phải lát phụ thuộc) đứng trước lát hiện thời theo thứ tự giải mã được yêu cầu. Khi các lát không được giải mã, nói

chung, lát phụ thuộc hiện thời không thể được giải mã. Trong một vài trường hợp đặc biệt, ví dụ, lát phụ thuộc có thể giải mã được khi một vài thông tin phụ khác được báo hiệu hoặc suy ra là sẵn có. Phần tử cú pháp `dependent_slice_flag` 601 được bao gồm gần như ở giữa của phần đầu lát. Hơn nữa, phần đầu lát gồm có số lượng dòng phụ CABAC nằm trong lát hiện thời được báo hiệu bởi phần tử thông tin `num_entry_point_offsets` 602 và số lượng byte trong dòng phụ được báo hiệu bởi phần tử cú pháp `entry_point_offset[i]` 603. Ở đây, phần tử thông tin `num_entry_point_offsets` 602 tương ứng với số lượng các điểm vào. Hơn nữa, i là số nguyên và chỉ số chỉ báo các điểm vào cụ thể (các độ lệch của các điểm vào). Số lượng byte trong dòng phụ được thể hiện bởi `entry_point_offset[i]` 603 cho phép điều hướng dễ dàng trong dòng bit.

1-7. Phần phụ thuộc của hình ảnh

Như được nêu trên, có vài loại phần phụ thuộc có được từ phương pháp mã hóa HEVC.

Fig.7 là sơ đồ thể hiện các phần phụ thuộc và việc báo hiệu của chúng trong trường hợp trong đó chỉ các lát thông thường, nghĩa là, không có lát phụ thuộc và không có lát entropy, được sử dụng. Fig.7 thể hiện ba hình ảnh 710, 720, và 730.

Hình ảnh 710 là hình ảnh lớp cơ sở được mang trong hai NALU VCL, chẳng hạn đơn vị NAL VCL 1 và đơn vị NAL VCL 2. POC chỉ báo thứ tự trong đó các hình ảnh được kết xuất. NALU VCL gồm có phần tử cú pháp chỉ báo xem hình ảnh thuộc lớp cơ sở hay lớp nâng cao, và phần tử cú pháp `temporal_id`. Phần tử cú pháp chỉ báo xem hình ảnh thuộc lớp cơ sở hay lớp nâng cao được truyền dưới trạng thái là nằm trong phần đầu NAL 570 của phần đầu gói 500 được thể hiện trên Fig.5. Phần tử cú pháp “`temporal_id`” cũng được truyền dưới trạng thái là nằm trong phần đầu NAL 570. Phần tử cú pháp “`temporal_id`” chỉ báo mức độ phụ thuộc vào các hình ảnh khác. Chẳng hạn, các hình ảnh hoặc các lát được mã hóa với `temporal_id=0` có thể được giải mã độc lập với các hình ảnh/các lát khác mà có `temporal_id` cao hơn. Cần lưu ý rằng trong HEVC, `temporal_id` được báo hiệu trong phần đầu NAL dưới dạng `nuh_temporal_id_plus1` (tham khảo Fig.9A). Cụ thể, biểu thức 1 dưới đây có thể được ứng dụng cho mối tương quan giữa `temporal_id` được sử dụng trong các ví dụ này và phần tử cú pháp `nuh_temporal_id_plus1`.

Biểu thức toán học 1

$$\text{temporal_id} = \text{nuh_temporal_id_plus1} - 1$$

(Biểu thức 1)

Các lát với $\text{temporal_id}=1$ phụ thuộc vào các lát của temporal_id với giá trị thấp hơn. Nói cách khác, giá trị của temporal_id trong trường hợp này là 0. Cụ thể, phần tử cú pháp temporal_id chỉ cấu trúc dự đoán của hình ảnh. Nói chung, các lát với giá trị cụ thể của temporal_id chỉ phụ thuộc vào các lát với giá trị bằng hoặc thấp hơn của temporal_id .

Theo đó, hình ảnh 710 trên Fig.7 có thể được giải mã trước tiên.

Hình ảnh 720 là lớp nâng cao so với lớp cơ sở của hình ảnh 710. Do đó, có sự phụ thuộc yêu cầu hình ảnh 720 được giải mã sau khi giải mã hình ảnh 710. Hình ảnh 720 gồm có hai NALU, chẳng hạn đơn vị NAL VCL 3 và đơn vị NAL VCL 4. Cả các hình ảnh 710 và 720 đều có giá trị POC là 0. Điều này có nghĩa là các hình ảnh 710 và 720 thuộc cùng ảnh được hiển thị cùng lúc. Các ảnh bao gồm lớp cơ sở và lớp nâng cao.

Hình ảnh 730 là lớp cơ sở mà gồm có hai NALU, chẳng hạn đơn vị NAL VCL 5 và đơn vị NAL VCL 6. Hình ảnh 730 có giá trị POC là 1. Điều này có nghĩa là (phần) hình ảnh 730 được hiển thị sau các hình ảnh 720 và 710. Hơn nữa, hình ảnh 730 có giá trị $\text{temporal_id} = 1$. Điều này có nghĩa là hình ảnh 730 phụ thuộc theo thời gian vào hình ảnh với $\text{temporal_id} = 0$. Theo đó, dựa vào phần phụ thuộc được báo hiệu trong phần đầu NAL, hình ảnh 730 phụ thuộc vào hình ảnh 710.

Fig.8 là sơ đồ thể hiện các phần phụ thuộc (mức độ phụ thuộc) và việc báo hiệu của chúng trong trường hợp trong đó không có lát phụ thuộc và lát entropy được sử dụng. Fig.8 thể hiện ba hình ảnh 810, 820, và 830. Fig.8 khác với Fig.7 được nêu trên ở chỗ các phần phụ thuộc của các lát phụ thuộc và các lát entropy được báo hiệu trong phần đầu lát được bổ sung.

Trên Fig.7, phần phụ thuộc liên lớp được thể hiện với ví dụ về các hình ảnh 710 và 720. Hơn nữa, phần phụ thuộc theo thời gian được thể hiện trong ví dụ về các hình ảnh 710 và 730. Các phần phụ thuộc này đều được báo hiệu trong phần đầu NAL.

Phần phụ thuộc liên lát như được thể hiện trên Fig.8 là sẵn có đối với các lát phụ thuộc và các lát entropy. Cụ thể, khung lớp cơ sở 810 và khung lớp nâng cao 820 đều có hai lát. Trong số hai lát, một lát là lát cha (lát thông thường) và lát còn lại là lát con (lát phụ thuộc). Trong khung 810, lát đơn vị NAL VCL 1 là lát cha của đơn vị NAL VCL 2. Trong

khung 820, lát đơn vị NAL VCL 3 là lát cha của đơn vị NAL VCL 4. Như được nêu trên, thuật ngữ “lát cha” của lát phụ thuộc dùng để chỉ lát mà lát phụ thuộc phụ thuộc vào, nghĩa là, lát có thông tin phần đầu lát được sử dụng bởi lát phụ thuộc. Đây là quy tắc mà lát đứng trước thứ nhất là lát có phần đầu hoàn chỉnh. Lát có phần đầu hoàn chỉnh là lát thông thường, không phải là lát phụ thuộc khác, chẳng hạn.

Cú pháp tương ứng của phần đầu đơn vị NAL và phần đầu lát như hiện được sử dụng trong HEVC và cụ thể trong HM8.0 sẽ được mô tả dựa vào Fig.9A.

Fig.9A là sơ đồ thể hiện cú pháp của phần đầu đơn vị NAL 910 và cú pháp của phần đầu lát 920. Cụ thể, các phần phụ thuộc liên lớp được lập kế hoạch (theo tiêu chuẩn hiện thời) được báo hiệu trong phần đầu đơn vị NAL sử dụng phần tử cú pháp `nuh_reserved_zero_6bits`. Các phần phụ thuộc theo thời gian được báo hiệu bằng cách sử dụng phần tử cú pháp `nuh_temporal_id_plus1`. Phần đầu lát 920 gồm có tín hiệu chỉ báo chỉ báo phụ thuộc liên lát. Chỉ báo phụ thuộc liên lát được chỉ báo bởi phần tử cú pháp `dependent_slice_flag`. Nói cách khác, phần phụ thuộc liên lát (ví dụ, phần phụ thuộc theo thời gian) được báo hiệu trong phần đầu lát, chỗ nào đó trong phần đầu lát.

Để phân tích cú pháp phần tử cú pháp này, tất cả các phần tử cú pháp đứng trước `dependent_slice_flag` phải được phân tích cú pháp cùng với các phần tử cú pháp tập hợp thông số cần thiết để phân tích cú pháp các phần tử cú pháp phần đầu lát đứng trước `dependent_slice_flag`.

1-8. Quá trình xử lý trong bộ định tuyến

Như được nêu trên, trong việc xác định định dạng lưu lượng, mong muốn tính đến các phần phụ thuộc được đưa vào bởi các lát phụ thuộc và các lát entropy, cùng với các phần phụ thuộc được báo hiệu trong phần đầu NAL. Chẳng hạn, bộ định tuyến có thể được thực hiện dưới dạng trạm gốc di động nhận biết phương tiện. Bảng thông trong đường xuống là rất hạn chế và cần được quản lý rất cẩn thận. Giả sử trường hợp ví dụ dưới đây. Giả sử rằng gói được loại bỏ ngẫu nhiên trong luồng dữ liệu lên bởi bộ định tuyến thông thường. Trong trường hợp này, các phần tử mạng nhận biết phương tiện (MANE) phát hiện mất gói bằng cách kiểm tra số lượng gói. Sau khi kiểm tra mất gói, MANE loại bỏ tất cả các gói phụ thuộc vào gói được loại bỏ và các gói theo sau. Đây là dấu hiệu được mong muốn đối với các phần tử mạng nhận biết phương tiện. Theo cách này, các gói có thể được loại bỏ một cách thông minh hơn. Khi bộ định tuyến được xác định để loại bỏ đơn vị NAL,

nó ngay lập tức suy ra rằng các lát phụ thuộc tiếp theo cũng cần được loại bỏ. Trong cú pháp hiện thời được đưa vào trên Fig.9A, việc truy cập `dependent_slice_flag` yêu cầu phân tích cú pháp lượng thông tin đáng kể. Điều này là không cần thiết đối với các thao tác định tuyến gói hoặc định dạng lưu lượng trong các bộ định tuyến. Tất cả thông tin cần thiết để phát hiện các mối tương quan liên lớp và liên thời gian đều có mặt trong tập hợp thông số video. Tập hợp thông số video là tập hợp cao nhất trong phân cấp tập hợp thông số.

Theo đó, thông tin nêu trên được báo hiệu trong phần đầu NAL 570. Tuy nhiên, trong trường hợp phần đầu NAL và phần đầu lát được thể hiện trên Fig.9A, việc truy cập thông tin phụ thuộc lát yêu cầu duy trì theo dõi các tập hợp thông số bổ sung chẳng hạn như PPS và SPS. Mặt khác, điều này tái sử dụng khả năng của các cổng hoặc các bộ định tuyến nhận biết phương tiện. Như được thấy từ Fig.9A, phần đầu lát 920 phải được phân tích cú pháp lên đến `dependent_slice_flag` và các thông số được phân tích cú pháp là không hiệu quả đối với hoạt động mạng.

Để có thể phân tích cú pháp địa chỉ lát mà đứng trước `dependent_slice_flag`, các phần tử cú pháp sau đây được yêu cầu từ các phần tử cú pháp được bao gồm trong SPS 930 như được thể hiện trên Fig.9B. Fig.9B là sơ đồ thể hiện ví dụ về cú pháp được bao gồm trong SPS.

- `pic_width_in_luma_samples` (ký hiệu tham chiếu 931 trên Fig.9B)
- `pic_height_in_luma_samples` (ký hiệu tham chiếu 932 trên Fig.9B)
- `log2_min_coding_block_size_minus 3` (ký hiệu tham chiếu 933 trên Fig.9B)
- `log2_diff_max_min_coding_block_size` (ký hiệu tham chiếu 934 trên Fig.9B)

Các thông số này được thể hiện ở bảng bên phải trên Fig.9B và là cần thiết để thu được thông số `slice_address`. Phần tử cú pháp `slice_address` được mã hóa độ dài thay đổi (như có thể được thấy khi nhìn ở độ dài “v” trong phần mô tả, cột thứ hai, của `slice_address` và phần đầu lát 920 trên Fig 9A). Để biết độ dài của thông số mã hóa độ dài thay đổi này, cần đến các phần tử cú pháp từ SPS. Thực tế, để có thể phân tích cú pháp `dependent_slice_flag`, giá trị thực của phần tử cú pháp `slice_address` là không cần thiết. Chỉ độ dài của phần tử cú pháp mà có thể thay đổi được là phải được biết sao cho quy trình phân tích cú pháp có thể tiếp tục.

Do đó, SPS cần được phân tích cú pháp lên đến điểm 935 từ các phần tử cú pháp

trong SPS 930 được thể hiện trên Fig.9B. Bốn phần tử cú pháp cần được lưu trữ. Chúng được sử dụng dưới đây trong công thức để tính toán độ dài của phần tử cú pháp `slice_address`.

Hơn nữa, để truy cập `dependent_slice_enabled_flag` cũng đứng trước `dependent_slice_flag`, PPS cần được phân tích cú pháp lên đến điểm 945 từ các phần tử cú pháp trong PPS được thể hiện trên Fig.9C. Fig.9C là sơ đồ thể hiện ví dụ về cú pháp được bao gồm trong PPS. Cần lưu ý rằng các phần tử cú pháp mà các phương pháp phân tích cú pháp của chúng được mô tả dựa vào các hình vẽ từ Fig.9A đến Fig.9C và được bố trí trong phần đầu lát và SPS và PPS là không cần thiết đối với các hoạt động bộ định tuyến chung. Hơn nữa, một vài phần tử cú pháp không thể dễ dàng được bỏ qua vì một vài phần tử cú pháp được mã hóa với các mã độ dài thay đổi. Theo đó, mặc dù nếu thao tác nhảy được thực hiện trong dòng bit bởi số lượng bit định trước, nhưng việc nhảy cho đến `dependent_slice_enabled_flag` là không khả thi.

Nói cách khác, để đọc `dependent_slice_flag` (chỉ báo phụ thuộc), MANE cần đi thêm nữa trong phần đầu lát (tham khảo phần đầu lát 920) mà việc phân tích cú pháp của chúng là khá phức tạp.

Cụ thể là, cờ `first_slice_in_pic_flag` phải được phân tích cú pháp. Cờ `first_slice_in_pic_flag` là cờ chỉ báo xem lát có là lát thứ nhất nằm trong hình ảnh hay không.

Sau đó, sự có mặt của `no_output_of_prior_pics_flag` là điều kiện về loại NALU phải được phân tích cú pháp.

Hơn nữa, `pic_parameter_set_id` được mã hóa độ dài thay đổi phải được giải mã. Phần tử cú pháp `pic_parameter_set_id` là phần tử cú pháp chỉ báo đâu là các tập hợp thông số được sử dụng (phần tử cú pháp nhận dạng tập hợp thông số). Nhờ phân tích cú pháp `pic_parameter_set_id`, tập hợp thông số được sử dụng có thể được nhận dạng.

Cuối cùng, phần tử cú pháp `slice_address` được yêu cầu. Phần tử cú pháp `slice_address` là phần tử cú pháp chỉ báo vị trí bắt đầu của lát. Phần tử cú pháp này còn yêu cầu phân tích cú pháp PPS và SPS cũng như tính toán bổ sung.

Với bước cuối cùng, giá trị của `dependent_slice_enabled_flag` (cờ cho phép lát phụ thuộc) phải thu được từ PPS, để biết xem `dependent_slice_flag` có mặt trong dòng bit hay

không. Khi `dependent_slice_enabled_flag = 0`, có nghĩa là lát hiện thời là lát thông thường vì các lát phụ thuộc không được phép. Để thu được giá trị của `dependent_slice_enabled_flag`, PPS cần được phân tích cú pháp lên đến gần phần giữa của nó.

Đáng tiếc là, các phần tử cú pháp trước `dependent_slice_flag` không thể được bỏ qua và cần được phân tích cú pháp không giống với trường hợp dữ liệu phần đầu NAL và RTP, trong đó vị trí của dữ liệu được xác định trước. Điều này là do thực tế là các phần tử cú pháp trong phần đầu lát được mã hóa độ dài thay đổi. Do đó, sự có mặt và độ dài của phần tử này cần được tính toán đối với mỗi đơn vị NAL VCL. Ngoài ra, dữ liệu phiên bổ sung cần được lưu trữ bởi vì chúng được cần đến sau đó (tham chiếu đến PPS và SPS). Hơn nữa, sự có mặt của một vài phần tử cú pháp phụ thuộc vào sự có mặt hoặc giá trị của các phần tử cú pháp khác có thể được bao gồm trong các cấu trúc thông số khác (các phần tử cú pháp được mã hóa có điều kiện).

Theo tiêu chuẩn hiện thời, đề xuất báo hiệu cấu trúc phần phụ thuộc của chuỗi video trong tập hợp thông số video (Video Parameter Set - VPS) mô tả bao nhiêu lớp được chứa trong dòng bit và các chỉ báo phụ thuộc để chỉ báo các phần phụ thuộc liên lớp khác nhau. VPS được báo hiệu ở ngay phần đầu của video, trước SPS thứ nhất. Nhiều SPS có thể tham chiếu đến một VPS. Điều này có nghĩa là VPS mang thông tin hợp lệ đối với nhiều chuỗi video. Mục đích chính của VPS là để thông báo cho bộ định tuyến hoặc bộ giải mã về nội dung của video bao gồm thông tin về bao nhiêu chuỗi video được bao gồm và chúng có tương quan như thế nào. SPS là hợp lệ chỉ nằm trong chuỗi video trong khi đó VPS mang thông tin liên quan đến nhiều chuỗi video.

Hơn nữa, đặc trưng của thông tin được mang trong VPS đặc biệt nhiều thông tin cho các bộ định tuyến. Ví dụ, VPS có thể mang thông tin cần thiết để thiết lập phiên phát trực tiếp vì thiết kế không được hoàn thiện. Bộ định tuyến phân tích cú pháp thông tin trong VPS. Bộ định tuyến, mà không cần tới các tập hợp thông số khác (nhờ nhìn ngay vào các phần đầu NAL), có thể xác định các gói dữ liệu nào để chuyển tiếp tới bộ giải mã và các gói nào để loại bỏ.

Tuy nhiên, để phát hiện VPS hoạt động hiện thời, các bước theo thứ tự sau cần được thực hiện:

phân tích cú pháp PPS_id trong phần đầu lát;

phân tích cú pháp SPS_id trong PPS hoạt động được xác định bởi PPS_id; và
phân tích cú pháp VPS_id trong SPS hoạt động được xác định bởi SPS_id.

Để giải quyết vấn đề nêu trên, phương pháp mã hóa ảnh theo một khía cạnh của sáng chế là phương pháp mã hóa ảnh để thực hiện xử lý mã hóa bằng cách phân chia hình ảnh thành các lát, phương pháp mã hóa ảnh bao gồm bước truyền dòng bit bao gồm: cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh gồm có lát phụ thuộc mà trên đó việc xử lý mã hóa được thực hiện phụ thuộc vào kết quả của việc xử lý mã hóa trên lát khác với lát hiện thời hay không; địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời; và chỉ báo phụ thuộc (dependent_slice_flag) chỉ báo xem lát hiện thời là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phân tử cú pháp (pic_parameter_set_id) nhận dạng tập hợp thông số.

Trong phương pháp mã hóa ảnh nêu trên, chỉ báo phụ thuộc của phần phụ thuộc liên lát được bố trí ở vị trí thích hợp cho việc phân tích cú pháp bởi bộ định tuyến. Với điều này, có thể mã hóa chỉ báo phụ thuộc bằng cú pháp một cách độc lập, nói cách khác, một cách không có điều kiện, với các phân tử cú pháp khác.

Ví dụ, chỉ báo phụ thuộc có thể được bao gồm trong dòng bit khi cờ cho phép lát phụ thuộc chỉ báo bao gồm lát phụ thuộc.

Ví dụ, cờ cho phép lát phụ thuộc có thể được bố trí ở phần bắt đầu của tập hợp thông số.

Ví dụ, mỗi trong số các lát có thể bao gồm các khối macro, và việc xử lý mã hóa trên lát hiện thời có thể được bắt đầu sau khi xử lý mã hóa được thực hiện trên hai trong số các khối macro được bao gồm trong lát ngay trước lát hiện thời.

Ví dụ, chỉ báo phụ thuộc có thể không được bao gồm trong phần đầu lát của lát mà được xử lý trước đối với hình ảnh, trong số các lát.

Để giải quyết vấn đề nêu trên, phương pháp giải mã ảnh theo một khía cạnh của sáng chế là phương pháp giải mã ảnh để thực hiện xử lý giải mã bằng cách phân chia hình ảnh thành các lát, phương pháp giải mã ảnh bao gồm bước trích, từ dòng bit được mã hóa, cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc mà trên đó việc

xử lý giải mã được thực hiện phụ thuộc vào kết quả của việc xử lý giải mã trên lát khác với lát hiện thời hay không, địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời, và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Ví dụ, chỉ báo phụ thuộc có thể được trích từ dòng bit khi cờ cho phép lát phụ thuộc chỉ báo việc bao gồm của lát phụ thuộc.

Ví dụ, cờ cho phép lát phụ thuộc có thể được bố trí ở phần bắt đầu của tập hợp thông số.

Ví dụ, mỗi trong số các lát có thể bao gồm các khối macro, và việc xử lý giải mã trên lát hiện thời có thể được bắt đầu sau khi xử lý giải mã được thực hiện ở hai trong số các khối macro được bao gồm trong lát ngay lát hiện thời trước.

Ví dụ, chỉ báo phụ thuộc có thể không được bao gồm trong phần đầu lát của lát được xử lý trước tiên đối với hình ảnh, trong số các lát.

Để giải quyết vấn đề, thiết bị mã hóa ảnh theo một khía cạnh của sáng chế là thiết bị mã hóa ảnh để thực hiện xử lý mã hóa bằng cách phân chia hình ảnh thành các lát, thiết bị mã hóa ảnh bao gồm bộ mã hóa để truyền dòng bit bao gồm: cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc mà trên đó việc xử lý mã hóa được thực hiện phụ thuộc vào kết quả của việc xử lý mã hóa trên lát khác với lát hiện thời hay không; địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời; và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Để giải quyết vấn đề, thiết bị giải mã ảnh theo một khía cạnh của sáng chế là thiết bị giải mã ảnh mà thực hiện việc xử lý giải mã bằng cách phân chia hình ảnh thành các lát, thiết bị giải mã ảnh bao gồm bộ giải mã để trích, từ dòng bit được mã hóa, cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh gồm có lát phụ thuộc mà trên đó việc xử lý giải mã được thực hiện phụ thuộc vào kết quả của việc xử lý giải mã trên lát khác với lát hiện thời hay không, địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời, và chỉ báo phụ thuộc chỉ báo xem

lát hiện thời có là lát phụ thuộc hay không, trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát, địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Để giải quyết vấn đề nêu trên, thiết bị mã hóa và giải mã ảnh theo một khía cạnh của sáng chế gồm có thiết bị mã hóa ảnh nêu trên và thiết bị giải mã ảnh nêu trên.

Theo phương pháp mã hóa ảnh, phương pháp giải mã ảnh, và phương pháp tương tự mà được cấu hình như trên, chỉ báo của phần phụ thuộc liên lát được bố trí trong cú pháp của dòng bit liên quan đến lát một cách độc lập với các thành phần khác. Chỉ báo phụ thuộc được bố trí tách biệt với các thành phần khác và không cần phân tích cú pháp các thành phần khác. Trong các ví dụ HEVC nêu trên, ký hiệu chỉ báo của phần phụ thuộc liên lát `dependent_slice_flag` được báo hiệu ở vị trí trong đó là không cần thiết để phân tích cú pháp các phần tử cú pháp không liên quan đối với hoạt động mạng.

Cụ thể là, sáng chế đề xuất thiết bị phân tích cú pháp dòng bit của chuỗi video của các ảnh được mã hóa ít nhất một phần với mã độ dài thay đổi và bao gồm các đơn vị dữ liệu mang các lát của chuỗi video được mã hóa. Thiết bị bao gồm bộ phân tích cú pháp để trích từ dòng bit chỉ báo phụ thuộc, là phần tử cú pháp chỉ báo đối với lát xem giải mã độ dài thay đổi hoặc phân tích cú pháp của lát phụ thuộc vào các lát khác hay không, trong đó chỉ báo phụ thuộc được trích từ dòng bit một cách độc lập và không cần trích các phần tử cú pháp khác trước đó.

Thiết bị như vậy có thể được bao gồm, chẳng hạn, trong bộ giải mã entropy 290 trên Fig.2. Khi đề cập đến việc trích từ dòng bit, ở đây nghĩa là việc trích và, nếu cần thiết cho việc trích, việc giải mã entropy. Mã hóa entropy là mã hóa độ dài thay đổi, chẳng hạn, mã hóa số học chẳng hạn như CABAC. Kỹ thuật này, trong HEVC, được áp dụng để mã hóa của dữ liệu ảnh. Các đơn vị dữ liệu ở đây đề cập đến, chẳng hạn, các đơn vị NAL hoặc các đơn vị truy cập. Cụm từ “mà không cần trích các phần tử cú pháp khác” chỉ tình trạng trong đó chỉ báo phụ thuộc được đứng trước chỉ bởi các phần tử mà độ dài đã được biết và sự có mặt của các thành phần này đã được biết hoặc tùy thuộc vào các thành phần đã được phân tích cú pháp hoặc đều không được mã hóa có điều kiện.

Sáng chế còn đề xuất thiết bị để tạo ra dòng bit của chuỗi video được mã hóa ít nhất một phần với mã độ dài thay đổi và bao gồm các đơn vị dữ liệu mang các lát được mã hóa

của các ảnh video. Thiết bị bao gồm bộ tạo dòng bit để nhúng vào dòng bit chỉ báo phụ thuộc mà là phần tử cú pháp chỉ báo đối với lát xem giải mã độ dài thay đổi của lát phụ thuộc vào các lát khác hay không, trong đó chỉ báo phụ thuộc được nhúng trong dòng bit một cách độc lập và không cần nhúng các phần tử cú pháp khác trước đó.

Thiết bị như vậy có thể được bao gồm, chẳng hạn, trong bộ mã hóa entropy 190 trên Fig.1.

Theo phương pháp mã hóa ảnh, phương pháp giải mã ảnh, và phương pháp tương tự mà được cấu hình nêu trên, dòng bit gồm có dữ liệu lát được mã hóa và dữ liệu phần đầu liên quan đến lát, và chỉ báo phụ thuộc được bố trí trong dòng bit ở đầu phần đầu lát. Điều này có nghĩa là phần đầu lát bắt đầu với các phần tử cú pháp chỉ báo phần phụ thuộc lát.

Cần lưu ý rằng chỉ báo phụ thuộc không nhất thiết được bố trí ở ngay đầu của phần đầu lát. Tuy nhiên, có ưu điểm khi không có phần tử cú pháp được mã hóa có điều kiện khác và/hoặc không có phần tử cú pháp được mã hóa độ dài thay đổi đứng trước ký hiệu chỉ báo phụ thuộc nằm trong phần đầu lát.

Chẳng hạn, vị trí hiện tại của `dependent_slice_flag` được thay đổi so với giải pháp kỹ thuật đã biết được nêu trên để được bố trí ở đầu của phần đầu lát. Với sự thay đổi này, có thể đạt được việc giảm lượng phần tử cú pháp cần được phân tích cú pháp. Tránh được các thao tác phân tích cú pháp phức tạp của các bộ định tuyến, chẳng hạn như giải mã độ dài thay đổi và phân tích cú pháp thông tin mà yêu cầu các tính toán bổ sung và/hoặc lưu trữ của các thông số bổ sung để sử dụng trong tương lai và/hoặc việc phân tích cú pháp các tập hợp thông số khác. Hơn nữa, số lượng của các tập hợp thông số cần theo dõi cũng giảm đi.

Dưới đây, các phương án được mô tả cụ thể dựa vào các hình vẽ. Mỗi trong số các phương án được mô tả dưới đây thể hiện ví dụ chung hoặc cụ thể. Các giá trị số, các hình dạng, các vật liệu, các thành phần cấu trúc, sự bố trí và sự kết nối của các thành phần cấu trúc, các bước, thứ tự xử lý của các bước, v.v. được thể hiện trong các phương án dưới đây chỉ là các ví dụ, và do đó không giới hạn phạm vi của sáng chế. Do đó, trong số các thành phần cấu trúc trong các phương án dưới đây, các thành phần cấu trúc mà không được trích dẫn trong điểm bất kỳ trong số các điểm yêu cầu bảo hộ độc lập được mô tả là các thành phần cấu trúc tùy ý.

Phương án 1

Fig.10 thể hiện ví dụ về cú pháp dòng bit theo phương án hiện thời. Phần đầu NAL 1010 được thể hiện trên Fig.10 là giống như phần đầu NAL 910 được thể hiện trên Fig.9A. Nói cách khác, không có thay đổi.

Tuy nhiên, cú pháp cấu trúc của phần đầu lát 1020 là khác với cú pháp cấu trúc của phần đầu lát 920 trên Fig.9A. Cụ thể, trong phần đầu lát 1020, `dependent_slice_flag` được dịch chuyển lên trong phần đầu lát theo cách sao cho không có phần tử cú pháp nào đứng trước `dependent_slice_flag`. `Dependent_slice_flag` được mã hóa có điều kiện, được mã hóa bằng cách sử dụng mã độ dài thay đổi, hoặc nhận việc phân tích cú pháp có yêu cầu tính toán bổ sung.

Các phần tử cú pháp `first_slice_in_pic_flag` và `dependent_slice_flag` thực tế đều xác định các phần phụ thuộc theo không gian. Các phần tử cú pháp được mã hóa ngay sau phần đầu NAL theo cách sao cho không có phần tử cú pháp nào khác cần được phân tích cú pháp. Vì `first_slice_in_pic_flag` cũng mang thông tin liên quan đến các phần phụ thuộc liên lát, nó có thể đứng trước `dependent_slice_flag`. Phần tử cú pháp `first_slice_in_pic_flag` là cờ được thiết lập theo quy tắc là mỗi khung phải bắt đầu với lát thông thường. Theo đó, khi cờ `first_slice_in_pic_flag` được thiết lập, có nghĩa là lát là lát thông thường và do đó độc lập. Do đó, `dependent_slice_flag` và `first_slice_in_pic_flag` có thể đều được xem chung như chỉ báo phần phụ thuộc liên lát.

Nói cách khác, chỉ báo phụ thuộc có thể được định rõ để bao gồm chỉ báo lát thứ nhất chỉ báo xem lát là lát thứ nhất trong hình ảnh hay không và cờ lát phụ thuộc chỉ báo xem giải mã độ dài thay đổi của lát có phụ thuộc vào các lát khác hay không. Lát thứ nhất trong hình ảnh luôn là lát mà việc giải mã độ dài thay đổi không phụ thuộc vào các lát khác.

Ưu điểm là, dòng bit gồm có cờ cho phép lát phụ thuộc chỉ báo xem các lát phụ thuộc có thể được bao gồm nằm trong dòng bit hay không. Chỉ báo phụ thuộc được bao gồm trong dòng bit chỉ khi cờ cho phép lát phụ thuộc chỉ báo rằng các lát phụ thuộc có thể được bao gồm trong dòng bit. Cờ cho phép lát phụ thuộc được bố trí trong dòng bit trong tập hợp thông số chung đối với các lát và được bố trí ở đầu tập hợp thông số. Tập hợp thông số có thể là, chẳng hạn, tập hợp thông số hình ảnh mà mang các thông số dùng cho hình ảnh đơn. Theo cách khác, cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chuỗi mà mang các thông số đối với toàn bộ chuỗi ảnh (video).

Tuy nhiên, theo sáng chế, `dependent_slice_flag` (chỉ báo phụ thuộc) được mã hóa không tùy thuộc vào phần tử cú pháp `dependent_slice_enabled_flag` (cờ cho phép lát phụ thuộc). Trong phương án hiện thời, vì tập hợp thông số hình ảnh id được bố trí sau chỉ báo phụ thuộc, có lợi khi tránh khỏi lỗi phân tích cú pháp có thể có trong trường hợp mà tập hợp thông số hình ảnh id được báo hiệu trong phần đầu lát.

Sự thay đổi này có thể cũng được thấy và/hoặc được đan xen nhờ thay đổi vị trí của các phần tử cú pháp được yêu cầu khác trong các tập hợp thông số hoặc các phần đầu để làm giảm lượng phần tử cú pháp mà cần được phân tích cú pháp nhằm xác định các phần phụ thuộc giữa các lát.

Chẳng hạn, phần tử cú pháp `dependent_slice_flag` trong phần đầu lát của cú pháp hiện tại là HM8.0 chỉ có mặt khi giá trị phần tử cú pháp “`dependent_slice_enabled_flag`” chỉ báo rằng việc sử dụng của các lát phụ thuộc nằm trong dòng bit được phép. Việc cho phép các lát phụ thuộc và do đó, phần tử cú pháp “`dependent_slice_enabled_flag`” cũng được bao gồm trong PPS như được thể hiện trên Fig.9C. Theo đó, phần tử cú pháp “`dependent_slice_enabled_flag`” trong PPS được dịch chuyển lên trong cú pháp của PPS để đơn giản hóa sự phân tích cú pháp cần thiết để phân tích cú pháp `dependent_slice_flag` (ví dụ, phần đầu của tập hợp thông số). Điều này có thể cũng hữu dụng khi `dependent_slice_flag` được mã hóa sau `pic_parameter_set_id` (phần tử cú pháp nhận dạng tập hợp thông số). Điều này là để, tránh được lỗi phân tích cú pháp ngay cả khi cờ cho phép lát phụ thuộc là tùy thuộc vào sự có mặt của chỉ báo phụ thuộc.

Thay vì việc dịch chuyển “`dependent_slice_enabled_flag`” lên trong PPS, “`dependent_slice_enabled_flag`” có thể được dịch chuyển từ PPS đến SPS và/hoặc VPS sao cho các tập hợp thông số mà thấp hơn theo thứ bậc phân cấp không cần được theo dõi.

Nói cách khác, theo phương án hiện thời, vị trí của các phần tử cú pháp cần thiết được thay đổi để làm giảm lượng tập hợp thông số cần được theo dõi. Điều này cũng làm giảm tính phức tạp của việc phân tích cú pháp. “Các thông số được yêu cầu” trong ngữ cảnh này có nghĩa là các thông số góp phần vào việc xác định xem lát có là lát phụ thuộc lát khác hay không. Khả năng thứ nhất có thể được áp dụng trực tiếp cho HEVC để cung cấp chỉ báo phụ thuộc ở đầu của phần đầu lát phụ thuộc và không tùy thuộc vào cờ cho phép lát phụ thuộc mà được bao gồm trong tập hợp thông số khác với phần đầu lát. Khả năng thứ hai có thể được áp dụng trực tiếp cho HEVC để cung cấp chỉ báo phụ thuộc ở

phần đầu lát phụ thuộc sau chỉ báo tập hợp thông số nhận dạng tập hợp thông số trong đó chờ cho phép lát phụ thuộc được bao gồm. Chỉ báo phụ thuộc có thể tùy thuộc vào chờ cho phép lát phụ thuộc. Việc dịch chuyển lên của chờ cho phép lát phụ thuộc trong PPS hoặc dịch chuyển chờ cho phép lát phụ thuộc vào SPS có thể có lợi đối với khả năng bất kỳ trong số các khả năng này. Cụ thể, điều này là có lợi đối với khả năng thứ hai, trong đó chờ cho phép lát phụ thuộc được cần đến để phân tích cú pháp chỉ báo phụ thuộc.

Như có thể được thấy trên Fig.10, phần đầu đơn vị NAL, cùng với phần liên quan của phần đầu lát, có 18 bit (14 bit của phần đầu NALU và 2 bit của phần đầu lát). Theo ví dụ này, các phần tử mạng nhận biết phương tiện có thể thao tác đối với gói lát hiện thời như sau. Nếu lát trước đó được loại bỏ là lát thông thường, lát entropy hoặc lát phụ thuộc, thì phần tử mạng kiểm tra hai bit đầu tiên của phần đầu lát hiện thời, là `first_slice_in_pic_flag` và (trong trường hợp mà các lát phụ thuộc được cho phép đối với dòng bit) `dependent_slice_flag`.

Khi loại đơn vị NAL là loại đơn vị NAL VCL và hai bit cuối cùng trong số 18 bit được kiểm tra là “01”, thì đơn vị NAL được loại bỏ. Cụ thể, khi bit đầu tiên của phần đầu lát là “1”, thì nó là lát thứ nhất trong hình ảnh mà (theo các quy tắc) không phải là lát phụ thuộc. Khi bit đầu tiên của phần đầu lát là “0” và bit tiếp theo của phần đầu lát cũng là “0”, thì lát là không phụ thuộc. Theo đó, chỉ khi hai bit đầu tiên của phần đầu lát là “01”, thì lát là phụ thuộc. Hơn nữa, lát cần được loại bỏ vì nó không thể được giải mã khi lát cha đã được loại bỏ. Theo đó, các cờ `first_slice_in_pic_flag` và `dependent_slice_flag` có thể được coi là phần mở rộng của phần đầu NAL, ngay cả khi chúng thuộc cú pháp phần đầu lát.

Theo đó, theo một trong số các khía cạnh, phương án hiện thời cũng đề xuất bộ định tuyến mạng để nhận, phân tích và chuyển tiếp các gói mạng tới các đích của chúng. Bộ định tuyến gồm có bộ thu để nhận gói mạng bao gồm địa chỉ đích gói và một phần dòng bit với dữ liệu video được mã hóa; bộ phân tích cú pháp bao gồm thiết bị phân tích cú pháp dòng bit của chuỗi video được mã hóa theo phương án bất kỳ trong số các phương án nêu trên đây và dưới đây, để xác định phần phụ thuộc của dữ liệu video được mã hóa từ các gói khác; và bộ phân tích gói để phân tích địa chỉ đích gói nhận được và phần phụ thuộc được xác định và để đánh giá cách thức xử lý gói mạng.

Phương án 2

Theo phương án 2, `dependent_slice_enabled_flag` được loại bỏ ra khỏi PPS. Cần

lưu ý rằng `dependent_slice_enabled_flag` có thể được dịch chuyển tới SPS, thay vì được loại bỏ.

Fig.11 thể hiện ví dụ trong đó `dependent_slice_enabled_flag` không cần được phân tích cú pháp trước khi truy cập `first_slice_in_pic_flag` và `dependent_slice_flag`.

Trong ví dụ này, `dependent_slice_enabled_flag` không được sử dụng bởi vì nó không tùy thuộc vào sự có mặt của chỉ báo phụ thuộc. Ví dụ này đưa ra khả năng là có chỉ báo phụ thuộc ở đầu của phần đầu lát mà không gây ra các vấn đề phân tích cú pháp do việc định danh chưa rõ của tập hợp PPS hiện thời.

Hiệu quả của phương án 2, v.v..

Trong Phương án 1, để phân tích cú pháp `dependent_slice_flag`, `dependent_slice_enabled_flag` phải được phân tích cú pháp. `Dependent_slice_enabled_flag` được báo hiệu trong PPS. Điều này có thể gây ra phí tổn phân tích cú pháp như được nêu trên, khi `dependent_slice_enabled_flag` được bố trí xa phần đầu PPS và các phần tử cú pháp đứng trước được mã hóa có điều kiện.

Hơn nữa, việc báo hiệu phần tử cú pháp `dependent_slice_flag` trước phần tử cú pháp `pic_parameter_set_id` trong PPS được phân tích cú pháp, có thể tạo ra các lỗi phân tích cú pháp như sau. Sự có mặt của `dependent_slice_flag` phụ thuộc vào `dependent_slice_enabled_flag` mà được báo hiệu trong PPS. Tuy nhiên, id của PPS hoạt động hiện thời được báo hiệu sau `dependent_slice_flag`. Do đó, không thể phân tích cú pháp `dependent_slice_flag` trước khi truy cập các phần tử trước đó.

Theo đó, có lợi khi loại bỏ điều kiện phân tích cú pháp trên `dependent_slice_enabled_flag`. Có thể có lợi hơn khi áp dụng phần hạn chế dưới đây. Chẳng hạn như, nếu `dependent_slice_enabled_flag` trong PPS bằng không, thì `dependent_slice_flag` phải bằng không.

Tuy nhiên, các phương án thực hiện có lợi này không giới hạn phạm vi của sáng chế.

Sửa đổi 1 của các phương án 1 và 2

Theo cách khác hoặc ngoài việc loại bỏ việc đặt điều kiện lên `dependent_slice_enabled_flag`, `dependent_slice_enabled_flag` có thể được dịch chuyển từ PPS tới SPS và/hoặc VPS.

Hơn nữa, thay vì chỉ dịch chuyển `dependent_slice_enabled_flag`, `dependent_slice_enabled_flag` có thể được nhân đôi trong SPS. Trong trường hợp này, chỉ báo trong SPS và PPS có thể được bắt buộc có giá trị như nhau. Hoặc, PPS có thể được cho phép ghi đè chỉ báo trong SPS.

Chẳng hạn, khi `sps_dependent_slice_enabled_flag` bằng 1, thì `pps_dependent_slice_enabled_flag` có thể bằng 0 hoặc 1. Sau đó, `sps_dependent_slice_enabled_flag` là chỉ báo về việc cho phép các lát phụ thuộc dùng cho chuỗi của các hình ảnh được báo hiệu trong SPS, và `pps_dependent_slice_enabled_flag` là chỉ báo về việc cho phép các lát phụ thuộc dùng cho hình ảnh được báo hiệu trong PPS. Tuy nhiên, khi giá trị của `dependent_slice_enabled_flag` có thể thay đổi trong PPS, điều này có nghĩa là việc phân tích cú pháp PPS vẫn được cần đến và ưu điểm là tần suất theo dõi thấp hơn và ngăn việc phân tích cú pháp của PPS.

Các sự sửa đổi này tạo ra ưu điểm là VPS và SPS mang các cấu trúc phụ thuộc. Việc mang các cấu trúc phụ thuộc bởi VPS và SPS cho phép các phần tử mạng định hình các dòng bit, nghĩa là, để xác định nhằm loại bỏ các gói phụ thuộc mà không thể được giải mã theo bất kỳ cách nào hoặc nhằm loại bỏ các lát phụ thuộc thay vì các lát độc lập. Do đó, `dependent_slice_enabled_flag` trong VPS sẽ kích hoạt bộ định tuyến để kiểm tra phần đầu lát dưới dạng bổ sung hay không.

Cần lưu ý rằng các sửa đổi này không làm giảm thêm tính phức tạp phân tích cú pháp nếu ví dụ trên Fig.10 và Fig.11 được ứng dụng. Tuy nhiên, nó tạo ra cấu trúc có lợi hơn của cú pháp để mang các cấu trúc phụ thuộc. Tóm lại, theo ví dụ này, chỉ báo để chỉ báo xem các lát phụ thuộc được phép dùng cho dòng bit hay không được báo hiệu trong tập hợp thông số video. Tập hợp thông số video là tập hợp thông số áp dụng cho nhiều hơn một lát trong nhiều hơn một hình ảnh.

Có hai ưu điểm khác nhau trong việc báo hiệu `dependent_slice_enabled_flag` là VPS và/hoặc SPS. Khi chỉ được dịch chuyển hoặc được nhân đôi, PPS không cần được phân tích cú pháp, làm giảm phí tổn phân tích cú pháp. Lợi ích khác là để các bộ định tuyến biết về cấu trúc dự đoán của chuỗi video. Ưu điểm này có ở mọi thời điểm. Thông thường, bộ định tuyến có thể kiểm tra nội dung của VPS/SPS để biết bộ định tuyến sẽ nhận được gì.

VPS là thông số cao nhất trong phân cấp. VPS có thể bao gồm thông tin về nhiều chuỗi video, trong khi đó SPS và PPS lần lượt dành riêng cho chuỗi video đơn và hình ảnh.

Thông tin trong VPS gồm có tốc độ bit, cấu trúc temporal_layering của các chuỗi video, và tương tự. Nó cũng gồm có thông tin về các phần phụ thuộc liên lớp (các phần phụ thuộc giữa các chuỗi video khác nhau). Theo đó, VPS có thể được coi là bộ chứa dùng cho nhiều chuỗi video, và nó đưa ra góc nhìn tổng quan về mỗi chuỗi.

Trong phiên bản HEVC hiện thời, phần phụ thuộc giữa các lát trong khung được thiết lập bởi cả `dependent_slice_flag` và `first_slice_in_pic_flag`. Theo các đặc điểm kỹ thuật hiện thời, các thực thể mạng không thể sử dụng các phần phụ thuộc liên lát mà không áp dụng phân tích cú pháp có tính phức tạp cao. Giải pháp không phức tạp sẽ là, nếu có tồn hao gói được phát hiện qua số gói thiếu, để loại bỏ tất cả các gói cho đến khi gặp `first_slice_in_pic_flag` bằng giá trị 1. Điều này là do lát thứ nhất trong hình ảnh luôn là lát thông thường.

Tuy nhiên, giải pháp này dẫn tới việc giảm hiệu quả mã hóa. Do đó, như được nêu trên, việc báo hiệu phần phụ thuộc liên lát cho phép phân tích cú pháp hữu hiệu có thể được sử dụng. Điều này đạt được nhờ báo hiệu `dependent_slice_flag` và `first_slice_in_pic_flag` trong phần đầu lát ngay sau phần đầu NAL.

Theo cách khác hoặc ngoài ra, các phần tử cú pháp liên quan đến các phần phụ thuộc liên lát được mã hóa một cách không có điều kiện, nghĩa là, độc lập với các phần tử cú pháp khác mà có thể là trong phần đầu lát hoặc trong PPS.

Sửa đổi 2 của các phương án 1 và 2

Fig.12 minh họa sửa đổi 2 theo cách khác với sửa đổi 1 được nêu trên. Cụ thể, phần đầu đơn vị NAL 1210 giống như phần đầu đơn vị NAL được thể hiện trên Fig.10 (phần đầu đơn vị NAL 910 được thể hiện trên Fig.9A). Tuy nhiên, phần đầu lát 1220 và phần đầu lát 1020 được thể hiện trên Fig.10 khác nhau ở chỗ các phần tử cú pháp phần đầu lát `dependent_slice_flag` và `first_slice_in_pic_flag` được đảo ngược thứ tự. Cụ thể, phần đầu lát 1220 gồm có `dependent_slice_flag` là phần tử cú pháp thứ nhất, và phần tử cú pháp `first_slice_in_pic_flag` là phần tử cú pháp thứ hai, tùy thuộc vào sự có mặt `dependent_slice_flag`.

Như có thể được thấy từ ví dụ này, chỉ báo lát thứ nhất chỉ báo xem lát là lát thứ nhất trong hình ảnh hay không được bao gồm trong cú pháp. Lát thứ nhất trong hình ảnh luôn là lát mà giải mã độ dài thay đổi không phụ thuộc vào các lát khác. Hơn nữa, cờ lát phụ thuộc được bao gồm trong dòng bit ở trước chỉ báo lát thứ nhất. Chỉ báo lát thứ nhất

được bao gồm trong dòng bit chỉ khi cờ lát phụ thuộc không chỉ báo lát phụ thuộc. Sự bố trí này tạo ra các ưu điểm giống như việc đặt điều kiện. Nói cách khác, phần phụ thuộc cờ tùy thuộc vào chỉ báo lát thứ nhất. Như có thể được thấy trên Fig.12, cả các thành phần có thể được hiểu như chỉ báo phụ thuộc và được bao gồm ở đầu của phần đầu lát.

Phương án 3

Trong phương án 3, so với các phương án 1 và 2, phương pháp bố trí các phần tử cú pháp được thay đổi để làm giảm việc phân tích cú pháp các phần tử cú pháp không cần thiết.

Trong các phương án nêu trên, `dependent_slice_flag` được mô tả trong trường hợp mà `first_slice_in_pic_flag` được bao gồm như điều kiện cho sự có mặt của `dependent_slice_flag`. Tuy nhiên, `first_slice_in_pic_flag` và `dependent_slice_flag` có thể đều được bao gồm trong dòng bit mà không tùy thuộc vào sự có mặt của phần tử cú pháp khác. Chẳng hạn, phương pháp mã hóa của `dependent_slice_flag` được thay đổi để độc lập với phần tử cú pháp `dependent_slice_enabled_flag` theo một trong số các sửa đổi được nêu trên.

Fig.13 là sơ đồ thể hiện ví dụ về phần đầu lát theo phương án hiện thời. Fig.13 minh họa trường hợp vẫn bao gồm việc đặt điều kiện về chỉ báo phụ thuộc trên cờ cho phép lát phụ thuộc.

Cụ thể là, trong phần đầu lát theo phương án hiện thời, `dependent_slice_flag` được bố trí trước `slice_address` so với phần đầu lát hiện có được thể hiện trên Fig.6. Hơn nữa, trong phần đầu lát theo phương án hiện thời, so với các ví dụ trên các hình vẽ từ Fig.10 đến Fig.12, `dependent_slice_flag` được bố trí sau `pic_parameter_set_id`.

Trong phương án hiện thời, vì `dependent_slice_flag` được bố trí trước `slice_address`, ít nhất SPS không cần được phân tích cú pháp để phân tích cú pháp `dependent_slice_flag`. Như được nêu trên, `slice_address` là phần tử cú pháp chỉ báo phần đầu của lát. Hơn nữa, `slice_address` có thể chỉ được phân tích cú pháp với sự trợ giúp của các phần tử cú pháp được báo hiệu trong SPS (`pic_parameter_set_id`).

Theo cách khác hoặc ngoài ra, `dependent_slice_enabled_flag` được dịch chuyển lên phía trên nằm trong PPS hoặc được dịch chuyển tới SPS và/hoặc VPS. Nếu cờ cho phép là trong VPS và/hoặc trong SPS, có thể không cần phân tích cú pháp và theo dõi PPS và SPS.

Sửa đổi của phương án 3, hiệu quả, và tương tự

(1) Ví dụ trên Fig.13 có thể dẫn đến việc bố trí thiết bị phân tích cú pháp dòng bit của chuỗi video được mã hóa ít nhất một phần với mã độ dài thay đổi và bao gồm các đơn vị dữ liệu mang các lát được mã hóa của các ảnh video. Trong trường hợp này, thiết bị được tạo cấu hình bao gồm bộ phân tích cú pháp trích từ dòng bit các phần tử cú pháp dưới đây:

chỉ báo phụ thuộc là phần tử cú pháp chỉ báo đối với lát trong phần đầu lát xem giải mã độ dài thay đổi của lát phụ thuộc vào các lát khác hay không;

cờ cho phép lát phụ thuộc được bao gồm trong tập hợp thông số dùng cho các lát và chỉ báo xem các lát phụ thuộc có thể được bao gồm trong dòng bit hay không; và

địa chỉ lát chỉ báo vị trí trong dòng bit mà tại đó lát bắt đầu.

(2) Hơn nữa, trong phương án hiện thời, chỉ báo phụ thuộc được báo hiệu trong phần đầu lát trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

Với phương án này, có thể không tạo ra các lỗi phân tích cú pháp để cấu hình là chỉ báo phụ thuộc được bao gồm trong dòng bit chỉ khi cờ cho phép lát phụ thuộc chỉ báo rằng các lát phụ thuộc có thể được bao gồm trong dòng bit.

(3) Trong phương án hiện thời, cờ cho phép lát phụ thuộc được bố trí trong dòng bit trong tập hợp thông số (PPS) chung đối với các lát tạo nên hình ảnh khung như nhau và được bố trí ở đầu tập hợp thông số. Tuy nhiên, nó không bị giới hạn ở đây.

Theo cách khác (hoặc ngoài ra), cờ cho phép lát phụ thuộc được bố trí trong dòng bit trong tập hợp thông số (SPS) chung đối với các lát tạo nên chuỗi của các hình ảnh như nhau. Theo cách khác nữa (hoặc ngoài ra), cờ cho phép lát phụ thuộc được bố trí trong dòng bit trong tập hợp thông số (VPS) chung đối với các lát tạo nên các chuỗi khung hình ảnh.

(4) Hơn nữa, trong phương án hiện thời, VPS_id và SPS_id có thể được báo hiệu rõ ràng trong bản tin SEI. Khi dependent_slice_enabled_flag được báo hiệu trong SPS, dependent_slice_flag phải vẫn theo sau pic_parameter_set_id.

Mặt khác, phần phụ thuộc phân tích cú pháp được đưa vào bởi vì SPS_id được báo hiệu trong PPS. Với việc báo hiệu nhận dạng của SPS hoặc VPS hiện thời có mang dependent_slice_enabled_flag, chỉ báo phụ thuộc có thể cũng được bao gồm trước pic_parameter_set_id bởi vì khi đó việc phân tích cú pháp tập hợp thông số hình ảnh là

không cần thiết. Hơn nữa, với bản tin SEI như vậy, việc mang VPS_id hoặc SPS_id là không cần thiết đối với thao tác giải mã vì các ID này cũng được xác định bằng cách phân tích cú pháp PPS. Bản tin SEI có thể do đó được loại bỏ mà không ảnh hưởng đến thao tác giải mã sau khi được các phần tử mạng sử dụng.

Phương án 4

Trong phương án 4, thông tin phụ thuộc liên lát được nhân đôi (bổ sung vào thông tin được báo hiệu trong phần đầu lát và/hoặc trong tập hợp thông số) trong đơn vị NAL khác chẳng hạn như bản tin SEI.

Chẳng hạn, bản tin SEI có thể được định rõ để vận chuyển thông tin phụ thuộc liên lát trong mỗi đơn vị truy cập hoặc trước mỗi lát phụ thuộc. Thuật ngữ “đơn vị truy cập” dùng để chỉ đơn vị dữ liệu được tạo nên từ tập hợp các đơn vị NAL. Đơn vị truy cập gồm có hình ảnh các lát được mã hóa, nghĩa là, các NALU VCL. Cụ thể, các đơn vị truy cập có thể định rõ các điểm dừng để truy cập ngẫu nhiên và có thể bao gồm các NALU của hình ảnh đơn. Tuy nhiên, đơn vị truy cập là không cần điểm truy cập ngẫu nhiên.

Theo các đặc tính kỹ thuật HEVC hiện thời, đơn vị truy cập được định rõ là tập hợp các đơn vị NAL liên tiếp theo thứ tự giải mã và chứa chỉ một hình ảnh được mã hóa. Cùng với các đơn vị NAL lát được mã hóa của hình ảnh được mã hóa, đơn vị truy cập có thể cũng chứa các đơn vị NAL khác không chứa các lát của hình ảnh được mã hóa. Việc giải mã đơn vị truy cập luôn tạo ra hình ảnh được giải mã. Tuy nhiên, ở phần mở rộng của HEVC sau này (giống như mã hóa đa khung hình (Multi-View Coding - MVC) hoặc mã hóa video có thể mở rộng (Scalable Video Coding - SVC)), việc xác định của đơn vị truy cập có thể được nói lỏng hoặc được sửa đổi. Phù hợp với các đặc điểm kỹ thuật hiện thời, đơn vị truy cập được tạo nên bởi dấu phân cách đơn vị truy cập, các bản tin SEI, và các NALU VCL.

Theo phương án hiện thời, chỉ báo phụ thuộc được bố trí trong dòng bit nằm ngoài phần đầu của lát mà chỉ báo phụ thuộc có liên quan. Hơn nữa, có thể có lợi khi chỉ báo phụ thuộc được bố trí trong dòng bit trong bản tin thông tin nâng cao bổ sung được bao gồm trong dòng bit trước lát phụ thuộc hoặc ngay trên mỗi đơn vị truy cập.

Phương án 5

Theo Phương án 5, thông tin phụ thuộc liên lát được báo hiệu trong phần đầu NAL

dưới dạng cờ hoặc ngầm định dưới dạng loại đơn vị NAL mà nó được kết hợp.

Theo quy tắc, việc phân tích cú pháp các phần tử cú pháp trong phần đầu NAL không phụ thuộc vào các phần tử cú pháp bất kỳ khác. Mỗi phần đầu đơn vị NAL có thể được phân tích cú pháp một cách độc lập. Phần đầu NAL là vị trí thông thường dùng để báo hiệu thông tin phụ thuộc. Theo đó, phù hợp với phương án hiện thời, phần phụ thuộc liên lát còn được báo hiệu trong đó.

Nói cách khác, thiết bị phân tích cú pháp có thể được thích ứng trong bộ định tuyến hoặc trong bộ giải mã. Thiết bị phân tích cú pháp còn bao gồm đơn vị lớp thích ứng mạng để bổ sung vào lát của dữ liệu video được mã hóa và vào phần đầu của lát lớp thích ứng mạng, và phần đầu NAL. Ưu điểm là, chỉ báo phụ thuộc được bố trí trong dòng bit trong phần đầu NAL và được mã hóa một cách độc lập với các phần tử cú pháp khác.

Chỉ báo phụ thuộc có thể được đặt trong phần đầu NAL vì phần đầu NAL trong các đặc tính kỹ thuật HEVC hiện thời dự tính một vài bit dự trữ để có thể được sử dụng từ đó. Một bit đơn sẽ là đủ để báo hiệu chỉ báo phụ thuộc.

Theo cách khác, chỉ báo phụ thuộc được chỉ báo bởi loại đơn vị NAL và loại đơn vị NAL định trước được dự trữ để mang thông tin phụ thuộc.

Phương án 6

Cần lưu ý rằng năm phương án nêu trên có thể được kết hợp tùy ý để cho phép phân tích cú pháp hiệu quả thông tin phụ thuộc trong các phần tử mạng. Ngay cả khi việc sử dụng của chúng là dư thừa, các phương án có thể được kết hợp. Theo đó, việc nhân đôi chỉ báo phụ thuộc có thể được áp dụng ngay cả khi chỉ báo phụ thuộc cũng được báo hiệu ở đầu của phần đầu lát.

Fig.14 thể hiện ví dụ về phần đầu đơn vị NAL 1410 trong đó phần đầu đơn vị NAL 910 được thể hiện trên Fig.9A được sửa đổi. Phần đầu đơn vị NAL 1410 bao gồm `dependent_slice_flag`.

Hơn nữa, để dịch chuyển `dependent_slice_flag` vào phần đầu NAL và để giữ kích cỡ của phần đầu NAL được cố định do khả năng tương thích ngược, một bit cần thiết đối với `dependent_slice_flag` được lấy từ phần tử cú pháp `nuh_reserved_zero_6bits` của phần đầu đơn vị NAL. Theo đó, phần tử cú pháp `nuh_reserved_zero_6bits` hiện chỉ có 5 bit. Phần tử cú pháp `nuh_reserved_zero_6bits` gồm có các bit được dự trữ dùng để sử dụng trong

tương lai sao cho việc giảm này không gây ra bất kỳ vấn đề nào và không cần thêm bất kỳ sửa đổi nào.

Nói chung, đơn vị NAL VCL hiện thời phụ thuộc vào đơn vị NAL VCL trước đó mà có cùng `temporal_layer_id`. Khi `dependent_slice_flag` được báo hiệu trong phần đầu NAL, một bit sẽ là lãng phí đối với cả các đơn vị NAL VCL và các đơn vị NAL không phải VCL vì mỗi đơn vị dữ liệu chẳng hạn như lát hình ảnh hoặc tập hợp thông số có cùng phần đầu NAL. Theo đó, mặc dù dường như là `dependent_slice_flag` cũng sẽ được báo hiệu đối với các tập hợp thông số hoặc đối với các bản tin SEI, điều này là không cần thiết. Hơn nữa, `dependent_slice_flag` luôn cần được báo hiệu ngay cả khi các lát phụ thuộc bị hủy kích hoạt trong tập hợp thông số chuỗi. Điều này dẫn đến phí tổn không cần thiết.

Trong tất cả các phương án trên, chỉ báo phụ thuộc có thể là cờ một bit.

Phương án 7

Theo Phương án 7, chỉ báo phụ thuộc được chỉ báo bởi loại đơn vị NAL và loại đơn vị NAL định trước được dự trữ để mang thông tin phụ thuộc.

Theo đó, loại VCL NAL mới (tách biệt) được định rõ với các ký hiệu tương tự như các đơn vị NAL VCL hiện có. Chẳng hạn, khi `NAL_unit_type` bằng 15 (hoặc loại định trước khác hoặc NALU mà không được dự trữ đối với loại NALU cụ thể khác), sau đó đơn vị NAL VCL hiện thời phụ thuộc vào đơn vị NAL VCL trước đó mà có cùng `temporal_layer_id`. Phần phụ thuộc này liên quan đến phần phụ thuộc của lát hiện thời trên phần đầu lát của lát đứng trước, như được nêu trên, nghĩa là, phần phụ thuộc trong phân tích cú pháp.

Có thể có lợi trong các trường hợp này khi bao gồm bit trong phần đầu NAL tới các loại đơn vị NAL bổ sung. Điều này có thể được sử dụng để chỉ báo xem lát hiện thời là lát phụ thuộc hay không.

Khi thông tin phụ thuộc được báo hiệu trong phần đầu lát ngoài phần đầu NAL, việc báo hiệu trong phần đầu NAL là tùy chọn. Cụ thể là, khi trường `NAL_unit_type` trong phần đầu NAL được tạo cấu hình để báo hiệu rằng lát hiện thời là lát phụ thuộc, thì không thể báo hiệu bất kỳ thông tin “loại” khác. Chẳng hạn, trong một vài trường hợp có thể có lợi hơn khi vận chuyển thông tin rằng lát hiện thời là “hình ảnh thứ nhất trong chuỗi” (`NAL_unit_type` bằng 10 hoặc 11). Khi thông tin phụ thuộc liên lát trong phần đầu NAL

là tùy chọn (vì nó được nhân đôi trong phần đầu lát), nó có thể được chọn để báo hiệu thông tin có giá trị hơn.

Hơn nữa, có thể có lợi khi bổ sung hai hoặc nhiều loại đơn vị NAL VCL, chẳng hạn như “hình ảnh RAP lát phụ thuộc” (cần cho phân tích cú pháp) hoặc “lát phụ thuộc không phải hình ảnh RAP”. “RAP” chỉ báo hình ảnh truy cập ngẫu nhiên. Hình ảnh truy cập ngẫu nhiên là hình ảnh được mã hóa một cách độc lập (xét về dự đoán) với các hình ảnh khác sao cho hình ảnh như vậy có thể được sử dụng như điểm bắt đầu cho việc mã hóa và giải mã. Với điều này, nó là thích hợp làm điểm truy cập ngẫu nhiên.

Trong phần đầu lát phụ thuộc, phần tử cú pháp *RapPicFlag* được sử dụng trong quy trình phân tích cú pháp. Cụ thể là, phần tử cú pháp *RapPicFlag* là chỉ báo cho biết xem hình ảnh hiện thời có là hình ảnh truy cập ngẫu nhiên hay không.

Giá trị của *RAPPicFlag* phụ thuộc vào loại đơn vị NAL giống như biểu thức 2 dưới đây.

Biểu thức toán học 2

$$RapPicFlag = (nal_unit_type \geq 7 \quad \&\& \quad nal_unit_type \leq 12)$$

(Biểu thức 2)

Nói cách khác, trong ví dụ được thể hiện trên Fig.15, các hình ảnh truy cập ngẫu nhiên được mang bởi các NALU với loại NALU trong khoảng giữa 7 và 12. Để cho phép phân tích cú pháp chính xác và cung cấp khả năng phụ thuộc lát cho các hình ảnh truy cập ngẫu nhiên, do đó, theo sáng chế, hai loại đơn vị NAL khác nhau được định rõ để đảm bảo phân tích cú pháp chính xác phần đầu lát.

Theo quy tắc chung, ngay cả khi loại đơn vị NAL VCL mới được định rõ, việc phân tích cú pháp phần đầu lát sẽ vẫn có thể thực hiện mà không có bất kỳ vấn đề nào. Một trong số các loại NAL được định rõ như trên hoặc phần đầu lát phụ thuộc được thay đổi theo cách sao cho không có vấn đề về phân tích cú pháp.

Khi loại đơn vị NAL VCL mới được định rõ để chỉ báo lát phụ thuộc, cấu trúc cú pháp phần đầu lát có thể được thay đổi như sau.

Trong ví dụ nêu trên, loại đơn vị NAL “DS_NUT” được sử dụng để chỉ báo rằng đơn vị NAL VCL hiện thời là lát phụ thuộc. So với cấu trúc cú pháp phần đầu lát theo giải pháp kỹ thuật đã biết mới nhất được mô tả trong tài liệu phi sáng chế 3, hai thay đổi dưới

đây được đưa vào phương án hiện thời.

(1) `no_output_of_prior_pics_flag` không được báo hiệu trong phần đầu lát phụ thuộc. Nói cách khác sự có mặt của `no_output_of_prior_pics_flag` là dựa vào điều kiện mà lát hiện thời là không phải lát phụ thuộc. (`no_output_of_prior_pics_flag` có thể có mặt trong phần đầu lát khi lát hiện thời không phải lát phụ thuộc).

(2) `first_slice_in_pic_flag` được báo hiệu phụ thuộc vào giá trị của `nal_unit_type`. Khi giá trị của `nal_unit_type` chỉ báo rằng lát hiện thời là lát phụ thuộc, phần tử cú pháp `first_slice_in_pic_flag` không được báo hiệu một cách rõ ràng và suy ra là 0. Điều này đảm bảo tốc độ bit ở chất lượng như nhau.

Theo ví dụ, `no_output_of_prior_pics_flag` không được báo hiệu khi lát hiện thời là lát phụ thuộc. Theo đó, giá trị của `RapPicFlag` không cần được đánh giá khi lát hiện thời là lát phụ thuộc. Do đó phần đầu lát của lát phụ thuộc có thể được phân tích cú pháp mà không có vấn đề gì. Cụ thể hơn là, phần đầu lát của lát phụ thuộc có thể được phân tích cú pháp mà không đề cập đến phần đầu đơn vị NAL của phần đầu đơn vị NAL đứng trước. Vấn đề xảy ra khi phần đầu đơn vị NAL đứng trước không có mặt ở thời điểm giải mã.

Thứ hai, `first_slice_in_pic_flag` được báo hiệu dựa vào giá trị của `NAL_unit_type`. Sự thay đổi này giống sự thay đổi trong ví dụ được mô tả trên Fig.12. Trên Fig.12, `first_slice_in_pic_flag` được báo hiệu trong phần đầu lát chỉ khi lát hiện thời không phải lát phụ thuộc (được chỉ báo bởi `dependent_slice_flag`). Tương tự trong ví dụ nêu trên `first_slice_in_pic_flag` được báo hiệu chỉ khi `nal_unit_type` không bằng “DS_NUT”, có nghĩa là lát hiện thời không phải lát phụ thuộc.

Hai thay đổi được nêu trên không cần thiết được thực hiện cùng nhau. Cũng có thể thực hiện chỉ một trong số các thay đổi trong phần đầu lát. Lợi ích của mỗi thay đổi gắn với chi phí kiểm tra xem lát có là lát phụ thuộc hay không. Tuy nhiên, khi hai thay đổi được thực hiện cùng nhau, các lợi ích của cả hai sự thay đổi có thể đều có được với cùng chi phí giống như lợi ích của mỗi thay đổi riêng lẻ trong trường hợp mà hai phần tử cú pháp `first_slice_in_pic_flag` và `no_output_of_prior_pics_flag` được mã hóa liên tiếp. Do đó, việc áp dụng cả hai sự thay đổi kết hợp với việc mã hóa liên tiếp hai phần tử cú pháp được nêu mang lại ưu điểm so với việc áp dụng trực tiếp mỗi thay đổi một cách riêng lẻ.

Trong toàn bộ phần giải thích trong các phương án, cũng có thể loại bỏ `dependent_slice_enabled_flag` ra khỏi dòng bit khi chỉ báo lát phụ thuộc không được mã

hóa có điều kiện trên nó. Nói cách khác, khi chẳng hạn loại đơn vị NAL mới được sử dụng để chỉ báo rằng lát hiện thời là lát phụ thuộc, thì `dependent_slice_enabled_flag` có thể được loại bỏ khỏi dòng bit.

Fig.15 thể hiện phần đầu đơn vị NAL 1510 giống như phần đầu đơn vị NAL 910 được thể hiện trên Fig.9A và phần đầu lát 1520 được thay đổi từ phần đầu lát 920 được thể hiện trên Fig.9A. Phần đầu lát 1520 gồm có phần kết thúc của giá trị `dependent_slice_flag` phù hợp với loại của NALU. Cụ thể, phần tử cú pháp `NAL_unit_type` với các giá trị 15 và 16 định rõ các lát phụ thuộc. Khi `NAL_unit_type` bằng 15, loại của lát là lát phụ thuộc của hình ảnh truy cập ngẫu nhiên. Nếu, mặt khác, `NAL_unit_type` bằng 16, thì lát là lát phụ thuộc của hình ảnh truy cập không ngẫu nhiên. Do đó, mối tương quan của Biểu thức 3 sau đây được thiết lập.

Biểu thức toán học 3

$$RapPicFlag = (nal_unit_type \geq 7 \&\& \quad nal_unit_type \leq 12 \parallel nal_unit_type == 15)$$

(Biểu thức 3)

Cần lưu ý rằng các giá trị 15 và 16 đã được lựa chọn chỉ là một ví dụ. Như được thấy rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật, bất kỳ các số định trước nào đều có thể được áp dụng mà không được sử dụng theo cách khác. Cụ thể là, loại thứ nhất của NALU được định rõ để nhận dạng nội dung lát phụ thuộc của hình ảnh truy cập ngẫu nhiên, và loại thứ hai của NALU được định rõ để nhận dạng nội dung lát phụ thuộc của hình ảnh truy cập không ngẫu nhiên.

Hơn nữa, sự hạn chế có thể được áp dụng mà các lát phụ thuộc chỉ được sử dụng cho các RAP hoặc chỉ được sử dụng cho các hình ảnh không phải RAP. Trong các trường hợp như vậy, chỉ một loại NALU mới là cần thiết.

Phương án 8

Fig.16 là sơ đồ thể hiện giải pháp theo cách khác. Phần đầu đơn vị NAL 1610 giống như phần đầu đơn vị NAL 910. Phần đầu lát 1620 giả sử việc xác định `NAL_unit_type` với các giá trị 15 và 16 báo hiệu các lát phụ thuộc như được nêu trên.

Tuy nhiên, loại đơn vị NAL không được sử dụng trong việc phân tích cú pháp cờ lát phụ thuộc. Điều này cho phép sử dụng `NAL_unit_type` tùy chọn đối với bộ mã hóa. Theo đó, ưu điểm của phương án hiện thời chỉ đạt được khi bộ mã hóa được xác định để

thích ứng các loại NALU mới.

Sau đó, bộ định tuyến chỉ cần tìm kiếm loại NALU. Tuy nhiên, khi bộ mã hóa không sử dụng các loại NALU mới, bộ định tuyến sẽ xử lý các lát phụ thuộc theo giải pháp kỹ thuật đã biết mới nhất.

Tóm lại, chỉ báo phụ thuộc có thể được chỉ báo bởi loại đơn vị NAL. Loại đơn vị NAL định trước có thể được dự trữ để mang các lát được mã hóa mà phần đầu lát của chúng phụ thuộc vào phần đầu lát của lát đứng trước. Ưu điểm là, loại đơn vị NAL riêng chỉ báo phụ thuộc được cung cấp cho các hình ảnh truy cập ngẫu nhiên và cho các hình ảnh truy cập không ngẫu nhiên.

Tóm lại, các phương án nêu trên liên quan đến cú pháp của dòng bit mang các chuỗi video được mã hóa. Cụ thể, các phương án nêu trên đề cập đến cú pháp liên quan đến các lát phụ thuộc và các lát entropy, trong đó phần đầu lát phụ thuộc vào phần đầu lát của lát đứng trước. Để cho phép phần tử mạng nhận biết phương tiện xem xét loại phụ thuộc này mà về bản chất là không làm tăng độ phức tạp và độ trễ do phân tích cú pháp, việc chỉ báo phụ thuộc được báo hiệu ở đầu các gói hoặc nói cách khác là ở gần các phần đầu hoặc các thông số được phân tích cú pháp. Điều này đạt được chẳng hạn, nhờ bao gồm chỉ báo phụ thuộc ở đầu của phần đầu lát (Fig.10 đến Fig.12), có thể là sau mã định danh tập hợp thông số và trước địa chỉ lát, hoặc nhờ bao gồm chỉ báo phụ thuộc trước địa chỉ lát (Fig.10 và Fig.11), hoặc nhờ cung cấp chỉ báo phụ thuộc trong phần đầu NALU (Fig.14), trong bản tin riêng hoặc bởi loại NALU đặc biệt dùng cho các NALU mang các lát phụ thuộc (Fig.15 và Fig.16).

Các sửa đổi của các phương án từ 1 đến 8, hiệu quả, và tương tự

Có thể thực hiện các thay đổi khác nhau mà không bị giới hạn bởi các phương án từ 1 đến 8, và hiển nhiên nằm trong phạm vi của sáng chế.

Mỗi trong số các thành phần cấu trúc ở mỗi trong số các phương án nêu trên có thể được tạo cấu hình ở dạng sản phẩm phần cứng duy nhất (mạch xử lý), hoặc có thể đạt được bằng cách thực thi chương trình phần mềm thích hợp dùng cho thành phần cấu trúc. Mỗi trong số các thành phần cấu trúc có thể đạt được nhờ đơn vị thực thi chương trình, chẳng hạn như CPU và bộ xử lý, đọc và thực thi chương trình phần mềm được ghi lên phương tiện lưu trữ chẳng hạn như đĩa cứng hoặc bộ nhớ bán dẫn.

Mặc dù trong các phương án từ 1 đến 8, phần mô tả giả sử mặt sóng, nhưng sáng chế không bị giới hạn như vậy.

Tuy nhiên, trong trường hợp mặt sóng, tất cả các dòng phụ không thể được bắt đầu đồng thời. Như được nêu trên, liên quan đến mỗi trong số các dòng phụ ngoại trừ đối với dòng phụ ở thời điểm bắt đầu, việc bắt đầu quy trình xử lý (mã hóa hoặc giải mã) được làm trễ bởi hai LCU từ dòng phụ đứng trước. Do đó, trong mặt sóng, rút ngắn thêm quy trình xử lý là cần thiết. Trong phương án hiện thời, nhờ định vị chỉ báo phụ thuộc (dependent_slice_flag) sau cú pháp mà nhận dạng PPS và trước địa chỉ lát, số lượng phần tử cú pháp cần được phân tích cú pháp có thể giảm đi và nhờ đó giảm việc xử lý.

Hơn nữa, trong các phương án từ 1 đến 8 nêu trên, nhờ bố trí chỉ báo phụ thuộc hướng về trong phần đầu lát (lưu ý là ở thời điểm bắt đầu), có thể, ví dụ, kiểm tra xem mỗi trong số các lát là lát phụ thuộc hay không ở giai đoạn sớm của quy trình xử lý hình ảnh.

Nói cách khác, ở thời điểm bắt đầu quy trình xử lý trên hình ảnh (mã hóa hoặc giải mã), khi bước kiểm tra xem mỗi trong số các lát là lát phụ thuộc hay không, có thể trích điểm bắt đầu của quá trình xử lý song song ở thời điểm bắt đầu quy trình xử lý trên hình ảnh. Nói cách khác, khi hình ảnh gồm có các lát thông thường, có thể trích điểm bắt đầu của quá trình xử lý song song ở thời điểm xử lý trên hình ảnh hoặc ở giai đoạn sớm của quy trình xử lý.

Ở đây, thông thường, khi chỉ báo phụ thuộc được bố trí sau địa chỉ lát, không thể kiểm tra xem lát là lát phụ thuộc hay lát thông thường cho đến khi việc phân tích cú pháp của địa chỉ lát hoàn tất. Trong trường hợp này, việc bắt đầu quy trình xử lý trên lát thông thường ở giữa hình ảnh sẽ bị trễ đáng kể so với việc bắt đầu quy trình xử lý lát thông thường ở đầu hình ảnh.

Ngược lại, trong các phương án từ 1 đến 8 nêu trên, vì có thể kiểm tra xem mỗi trong số các lát là lát phụ thuộc hay không ở giai đoạn sớm hơn của quy trình xử lý hình ảnh, có thể thúc đẩy phần đầu của quy trình xử lý lát thông thường ở giữa hình ảnh. Nói cách khác, có thể bắt đầu quy trình xử lý lát thông thường ở giữa hình ảnh đồng thời như lát thông thường ở đầu hình ảnh.

Phương án 9

Quy trình xử lý được mô tả ở mỗi trong số các phương án nêu trên có thể được thực

hiện đơn giản trong hệ thống máy tính độc lập, bằng cách ghi, trong phương tiện lưu trữ, chương trình để thực hiện các cấu hình của phương pháp mã hóa hình ảnh động (phương pháp mã hóa ảnh) và phương pháp giải mã hình ảnh động (phương pháp giải mã ảnh) được mô tả ở mỗi trong số các phương án. Phương tiện lưu trữ có thể là phương tiện lưu trữ bất kỳ miễn là chương trình có thể được ghi, chẳng hạn như đĩa từ, đĩa quang, đĩa từ quang, thẻ IC, và bộ nhớ bán dẫn.

Dưới đây, các ứng dụng của phương pháp mã hóa hình ảnh động (phương pháp mã hóa ảnh) và phương pháp giải mã hình ảnh động (phương pháp giải mã ảnh) được mô tả ở mỗi trong số các phương án và các hệ thống nhờ sử dụng các phương pháp này sẽ được mô tả. Hệ thống có dấu hiệu của việc có thiết bị mã hóa và giải mã ảnh gồm có thiết bị mã hóa ảnh sử dụng phương pháp mã hóa ảnh và thiết bị giải mã ảnh sử dụng phương pháp giải mã ảnh. Các cấu hình khác trong hệ thống có thể được thay đổi thích hợp tùy thuộc vào các trường hợp.

Fig.17 minh họa cấu hình tổng thể của hệ thống cung cấp nội dung ex100 để thực hiện các dịch vụ phân phối nội dung. Vùng để cung cấp các dịch vụ truyền thông được chia thành các ô có kích thước mong muốn, và các trạm gốc ex106, ex107, ex108, ex109, và ex110 là các trạm không dây cố định được đặt ở mỗi trong số các ô.

Hệ thống cung cấp nội dung ex100 lần lượt được nối với các thiết bị, chẳng hạn như máy tính ex111, máy hỗ trợ số cá nhân (personal digital assistant - PDA) ex112, camera ex113, điện thoại di động ex114 và máy chơi trò chơi ex115, qua Internet ex101, nhà cung cấp dịch vụ Internet ex102, mạng điện thoại ex104, cũng như các trạm gốc từ ex106 đến ex110.

Tuy nhiên, cấu hình của hệ thống cung cấp nội dung ex100 không giới hạn ở cấu hình được thể hiện trên Fig.17, và sự kết hợp trong đó các thành phần bất kỳ được kết nối có thể chấp nhận được. Ngoài ra, mỗi thiết bị có thể được nối trực tiếp với mạng điện thoại ex104, thay vì qua các trạm gốc từ ex106 đến ex110 mà là các trạm không dây cố định. Hơn nữa, các thiết bị có thể được liên kết với nhau qua truyền thông không dây tầm ngắn và các loại khác.

Camera ex113, chẳng hạn như camera video số, là thiết bị có khả năng ghi video. Camera ex116, chẳng hạn như camera số, là thiết bị có khả năng chụp cả ảnh tĩnh và ghi video. Hơn nữa, điện thoại di động ex114 có thể là một dạng đáp ứng chuẩn bất kỳ chẳng

hạn như hệ thống thông tin di động toàn cầu (Global System for Mobile Communications - GSM) (nhãn hiệu đã đăng ký), đa truy cập phân chia theo mã (Code Division Multiple Access - CDMA), đa truy cập phân chia theo mã băng rộng (Wideband-Code Division Multiple Access - W-CDMA), công nghệ tiến hóa dài hạn (Long Term Evolution - LTE), và truy cập gói tốc độ cao (High Speed Packet Access - HSPA). Theo cách khác, điện thoại di động ex114 có thể là hệ thống điện thoại cầm tay cá nhân (Personal Handyphone System - PHS).

Trong hệ thống cung cấp nội dung ex100, máy chủ truyền phát trực tiếp ex103 được nối với camera ex113 và các thiết bị khác qua mạng điện thoại ex104 và trạm gốc ex109, để cho phép phân phối các ảnh của buổi trình chiếu trực tiếp và các loại khác. Trong phân phối nội dung như vậy, nội dung (ví dụ, video của buổi trình chiếu âm nhạc trực tiếp) được ghi bởi người dùng sử dụng camera ex113 được mã hóa như được nêu trên ở mỗi trong số các phương án (nghĩa là, camera có chức năng như thiết bị mã hóa ảnh theo một khía cạnh của sáng chế), và nội dung được mã hóa được truyền tới máy chủ truyền phát trực tiếp ex103. Mặt khác, máy chủ truyền phát trực tiếp ex103 thực hiện phân phối dòng của dữ liệu nội dung được truyền tới các máy khách khi có yêu cầu. Các máy khách bao gồm máy tính ex111, PDA ex112, camera ex113, điện thoại di động ex114, và máy chơi trò chơi ex115 có khả năng giải mã dữ liệu được mã hóa nêu trên. Mỗi trong số các thiết bị nhận được dữ liệu được phân phối giải mã và tái tạo dữ liệu được mã hóa (nghĩa là, dùng làm thiết bị giải mã ảnh theo một khía cạnh của sáng chế).

Dữ liệu được chụp có thể được mã hóa bởi camera ex113 hoặc máy chủ truyền phát trực tiếp ex103 để truyền dữ liệu, hoặc các quy trình mã hóa có thể được dùng chung giữa camera ex113 và máy chủ truyền phát trực tiếp ex103. Tương tự, dữ liệu được phân phối có thể được giải mã bởi các máy khách hoặc máy chủ truyền phát trực tiếp ex103, hoặc quy trình giải mã có thể được dùng chung giữa các máy khách và máy chủ truyền phát trực tiếp ex103. Hơn nữa, dữ liệu gồm các ảnh tĩnh và video được ghi không chỉ bởi camera ex113 mà cả camera ex116 có thể được truyền tới máy chủ truyền phát trực tiếp ex103 qua máy tính ex111. Các quy trình mã hóa có thể được thực hiện bởi camera ex116, máy tính ex111, hoặc máy chủ truyền phát trực tiếp ex103, hoặc được dùng chung giữa các thiết bị này.

Hơn nữa, quy trình mã hóa và giải mã có thể được thực hiện bởi LSI ex500 nói chung được bao gồm trong mỗi trong số máy tính ex111 và các thiết bị. LSI ex500 có thể

được tạo cấu hình từ chip đơn hoặc các chip. Phần mềm dùng cho việc mã hóa và giải mã video có thể được tích hợp trong loại phương tiện lưu trữ nào đó (chẳng hạn như CD-ROM, đĩa mềm, và đĩa cứng) mà có thể đọc được bởi máy tính ex111 và các loại khác, và quy trình mã hóa và giải mã có thể được thực hiện bằng cách sử dụng phần mềm. Hơn nữa, khi điện thoại di động ex114 được trang bị camera, dữ liệu video thu được bởi camera có thể được truyền. Dữ liệu video là dữ liệu được mã hóa bởi LSI ex500 được bao gồm trong điện thoại di động ex114.

Hơn nữa, máy chủ truyền phát trực tiếp ex103 có thể bao gồm các máy chủ và các máy tính, và có thể phi tập trung hóa dữ liệu và xử lý dữ liệu phi tập trung, ghi, hoặc phân phối dữ liệu.

Như được nêu trên, các máy khách có thể nhận và tái tạo dữ liệu được mã hóa trong hệ thống cung cấp nội dung ex100. Nói cách khác, các máy khách có thể thu nhận và giải mã thông tin được truyền bởi người dùng, và tái tạo dữ liệu được giải mã theo thời gian thực trong hệ thống cung cấp nội dung ex100, sao cho người dùng mà không có bất kỳ quyền và thiết bị cụ thể nào đều có thể thực hiện phát quảng bá cá nhân.

Ngoài ví dụ về hệ thống cung cấp nội dung ex100, ít nhất một trong số thiết bị mã hóa hình ảnh động (thiết bị mã hóa ảnh) và thiết bị giải mã hình ảnh động (thiết bị giải mã ảnh) được mô tả ở mỗi trong số các phương án có thể được thực hiện trong hệ thống phát quảng bá số ex200 được minh họa trên Fig.18. Cụ thể hơn là, trạm phát quảng bá ex201 truyền thông hoặc truyền, qua các sóng vô tuyến tới vệ tinh phát quảng bá ex202, dữ liệu được ghép kênh thu được nhờ ghép kênh dữ liệu âm thanh và các loại khác trên dữ liệu video. Dữ liệu video là dữ liệu được mã hóa bởi phương pháp mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án (nghĩa là, dữ liệu được mã hóa bởi thiết bị mã hóa ảnh theo một khía cạnh của sáng chế). Khi nhận dữ liệu được ghép kênh, vệ tinh phát quảng bá ex202 truyền các sóng vô tuyến để phát quảng bá. Sau đó, anten gia đình ex204 với chức năng nhận tín hiệu vệ tinh thu các sóng vô tuyến. Tiếp theo, thiết bị chẳng hạn như máy thu hình (bộ thu) ex300 và đầu thu giải mã tín hiệu (set top box - STB) ex217 giải mã dữ liệu được ghép kênh nhận được, và tái tạo dữ liệu được giải mã (nghĩa là, dùng làm thiết bị giải mã ảnh theo một khía cạnh của sáng chế).

Hơn nữa, bộ đọc/bộ ghi ex218 (i) đọc và giải mã dữ liệu đã ghép kênh được ghi lên phương tiện lưu trữ ex215, chẳng hạn như DVD và BD, hoặc (i) mã hóa các tín hiệu video

trong phương tiện lưu trữ ex215, và trong một vài trường hợp, ghi dữ liệu thu được nhờ ghép kênh tín hiệu âm thanh trên dữ liệu được mã hóa. Bộ đọc/bộ ghi ex218 có thể bao gồm thiết bị giải mã hình ảnh động hoặc thiết bị mã hóa hình ảnh động như được thể hiện ở mỗi trong số các phương án. Trong trường hợp này, các tín hiệu video đã tái tạo được hiển thị trên màn hình ex219, và có thể được tái tạo bởi thiết bị hoặc hệ thống khác nhờ sử dụng phương tiện lưu trữ ex215 có dữ liệu đã ghép kênh được ghi trên đó. Cũng có thể cài đặt thiết bị giải mã hình ảnh động trong đầu thu giải mã tín hiệu ex217 được nối với cáp ex203 dùng cho truyền hình cáp hoặc với anten ex204 dùng cho phát quảng bá vệ tinh và/hoặc mặt đất, để hiển thị các tín hiệu video trên màn hình ex219 của máy thu hình ex300. Thiết bị giải mã hình ảnh động có thể không được lắp đặt trong đầu thu giải mã tín hiệu mà trong máy thu hình ex300.

Fig.19 minh họa máy thu hình (bộ thu) ex300 sử dụng phương pháp mã hóa hình ảnh động và phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án. Máy thu hình ex300 gồm có: bộ điều hướng ex301 để thu nhận hoặc cung cấp dữ liệu được ghép kênh thu được nhờ ghép kênh dữ liệu âm thanh trên dữ liệu video, qua anten ex204 hoặc cáp ex203, v.v., để nhận chương trình phát quảng bá; bộ điều chế/giải điều chế ex302 để giải điều chế dữ liệu được ghép kênh thu được hoặc điều chế dữ liệu thành dữ liệu được ghép kênh để cấp ra ngoài; và bộ ghép kênh/giải ghép kênh ex303 để giải ghép kênh dữ liệu đã ghép kênh được điều chế thành dữ liệu video và dữ liệu âm thanh, hoặc ghép kênh dữ liệu video và dữ liệu âm thanh được mã hóa bởi bộ xử lý tín hiệu ex306 thành dữ liệu.

Máy thu hình ex300 còn bao gồm: bộ xử lý tín hiệu ex306 bao gồm bộ xử lý tín hiệu âm thanh ex304 và bộ xử lý tín hiệu video ex305 để lần lượt giải mã dữ liệu âm thanh và dữ liệu video và mã hóa dữ liệu âm thanh và dữ liệu video, (mà dùng làm thiết bị mã hóa ảnh và thiết bị giải mã ảnh theo các khía cạnh của sáng chế); và bộ kết xuất ex309 bao gồm loa ex307 để cung cấp tín hiệu âm thanh được giải mã, và đơn vị hiển thị ex308 để hiển thị tín hiệu video được giải mã, chẳng hạn như màn hình. Hơn nữa, máy thu hình ex300 gồm có đơn vị giao diện ex317 bao gồm bộ đầu vào hoạt động ex312 để nhận dữ liệu đầu vào của thao tác người dùng. Hơn nữa, máy thu hình ex300 gồm có bộ điều khiển ex310 để điều khiển toàn bộ mỗi thành phần cấu thành của máy thu hình ex300, và bộ mạch cấp nguồn ex311 để cấp nguồn tới mỗi trong số các thành phần. Ngoài bộ đầu vào hoạt động ex312, bộ giao diện ex317 có thể bao gồm: cầu ex313 được kết nối với thiết bị bên ngoài,

chẳng hạn như bộ đọc/bộ ghi ex218; bộ khe cắm ex314 để cho phép lắp phương tiện lưu trữ ex216, chẳng hạn như thẻ SD; ổ đĩa ex315 được kết nối với phương tiện lưu trữ bên ngoài, chẳng hạn như đĩa cứng; và modem ex316 được kết nối với mạng điện thoại. Ở đây, phương tiện lưu trữ ex216 có thể ghi điện tử thông tin sử dụng phần tử nhớ bán dẫn khả biến/bất khả biến để lưu trữ. Các thành phần cấu thành của máy thu hình ex300 được kết nối với nhau qua bus đồng bộ.

Thứ nhất, cấu hình trong đó máy thu hình ex300 giải mã dữ liệu được ghép kênh thu được từ bên ngoài qua anten ex204 và các loại khác và tái tạo dữ liệu được giải mã sẽ được mô tả. Trong máy thu hình ex300, khi thao tác người dùng qua bộ điều khiển từ xa ex220 và các loại khác, bộ ghép kênh/giải ghép kênh ex303 giải ghép kênh dữ liệu đã ghép kênh được giải điều chế bởi bộ điều chế/giải điều chế ex302, dưới sự điều khiển của bộ điều khiển ex310 bao gồm CPU. Hơn nữa, bộ xử lý tín hiệu âm thanh ex304 giải mã dữ liệu âm thanh được giải ghép kênh, và bộ xử lý tín hiệu video ex305 giải mã dữ liệu video được giải ghép kênh, sử dụng phương pháp giải mã được mô tả ở mỗi trong số các phương án, trong máy thu hình ex300. Đơn vị đầu ra ex309 lần lượt cung cấp tín hiệu video và tín hiệu âm thanh được giải mã ra ngoài. Khi đơn vị đầu ra ex309 cung cấp tín hiệu video và tín hiệu âm thanh, các tín hiệu này có thể được lưu trữ tạm thời trong các bộ đệm ex318 và ex319, và các loại khác sao cho các tín hiệu được tái tạo đồng bộ với nhau. Hơn nữa, máy thu hình ex300 có thể đọc dữ liệu được ghép kênh không qua phát quảng bá và các loại khác mà từ phương tiện lưu trữ ex215 và ex216, chẳng hạn như đĩa từ, đĩa quang, và thẻ SD. Tiếp theo, cấu hình trong đó máy thu hình ex300 mã hóa tín hiệu âm thanh và tín hiệu video, và truyền dữ liệu ra ngoài hoặc ghi dữ liệu lên phương tiện lưu trữ sẽ được mô tả. Trong máy thu hình ex300, khi thao tác người dùng qua bộ điều khiển từ xa ex220 và các loại khác, bộ xử lý tín hiệu âm thanh ex304 mã hóa tín hiệu âm thanh, và bộ xử lý tín hiệu video ex305 mã hóa tín hiệu video, dưới sự điều khiển của bộ điều khiển ex310 sử dụng phương pháp mã hóa được mô tả ở mỗi trong số các phương án. Bộ ghép kênh/giải ghép kênh ex303 ghép kênh tín hiệu video và tín hiệu âm thanh đã mã hóa, và cung cấp tín hiệu nhận được ra ngoài. Khi bộ ghép kênh/giải ghép kênh ex303 ghép kênh tín hiệu video và tín hiệu âm thanh, các tín hiệu có thể được lưu trữ tạm thời trong các bộ đệm ex320 và ex321, và các loại khác sao cho các tín hiệu được tái tạo đồng bộ với nhau. Ở đây, các bộ đệm ex318, ex319, ex320, và ex321 có thể là số nhiều như được minh họa, hoặc ít nhất một bộ đệm có thể được dùng chung trong máy thu hình ex300. Hơn nữa, dữ liệu có thể

được lưu trữ trong bộ đệm sao cho hiện tượng tràn trên và tràn dưới hệ thống có thể được tránh khỏi giữa bộ điều chế/giải điều chế ex302 và bộ ghép kênh/giải ghép kênh ex303 chẳng hạn.

Hơn nữa, máy thu hình ex300 có thể bao gồm cấu hình để nhận đầu vào AV từ micro hoặc camera khác với cấu hình để thu được dữ liệu âm thanh và video từ phương tiện phát quang bá hoặc phương tiện lưu trữ, và có thể mã hóa dữ liệu thu được. Mặc dù máy thu hình ex300 có thể mã hóa, ghép kênh, và cung cấp ra ngoài dữ liệu trong phần mô tả, nó có thể có khả năng chỉ thu, giải mã, và cung cấp dữ liệu ra ngoài mà không phải mã hóa, ghép kênh, và cung cấp dữ liệu ra ngoài.

Hơn nữa, khi bộ đọc/bộ ghi ex218 đọc hoặc ghi dữ liệu được ghép kênh từ hoặc lên phương tiện lưu trữ, một trong số máy thu hình ex300 và bộ đọc/bộ ghi ex218 có thể giải mã hoặc mã hóa dữ liệu được ghép kênh, và máy thu hình ex300 và bộ đọc/bộ ghi ex218 có thể dùng chung dữ liệu giải mã hoặc mã hóa.

Theo một ví dụ, Fig.20 minh họa cấu hình của bộ ghi/tái tạo thông tin ex400 khi dữ liệu được đọc từ hoặc được ghi lên đĩa quang. Bộ ghi/tái tạo thông tin ex400 gồm có các thành phần cấu thành ex401, ex402, ex403, ex404, ex405, ex406, và ex407 được mô tả dưới đây. Đầu đọc quang ex401 chiếu vệt laze trên bề mặt ghi của phương tiện lưu trữ ex215 mà là đĩa quang để ghi thông tin, và phát hiện ánh sáng được phản xạ từ bề mặt ghi của phương tiện lưu trữ ex215 để đọc thông tin. Bộ ghi điều chế ex402 dẫn điện laze bán dẫn được bao gồm trong đầu đọc quang ex401, và điều chế ánh sáng laze theo dữ liệu được ghi. Bộ giải điều chế tái tạo ex403 khuếch đại tín hiệu tái tạo thu được nhờ phát hiện dưới dạng điện ánh sáng được phản xạ từ bề mặt ghi sử dụng bộ phát hiện quang được bao gồm trong đầu đọc quang ex401, và giải điều chế tín hiệu tái tạo nhờ tách thành phần tín hiệu được ghi lên phương tiện lưu trữ ex215 để tái tạo thông tin cần thiết. Bộ đệm ex404 lưu giữ tạm thời thông tin được ghi lên phương tiện lưu trữ ex215 và thông tin được tái tạo từ phương tiện lưu trữ ex215. Động cơ đĩa ex405 quay phương tiện lưu trữ ex215. Bộ điều khiển trợ lực ex406 dịch chuyển đầu đọc quang ex401 tới rãnh thông tin định trước trong khi điều khiển sự dẫn động chuyển động quay của động cơ đĩa ex405 để theo sát vệt laze. Bộ điều khiển hệ thống ex407 điều khiển toàn bộ bộ ghi/tái tạo thông tin ex400. Quy trình đọc và ghi có thể được thực hiện bởi bộ điều khiển hệ thống ex407 sử dụng thông tin khác nhau được lưu trữ trong bộ đệm ex404 và tạo ra và bổ sung thông tin mới khi cần, và bởi bộ ghi điều chế ex402, bộ giải điều chế tái tạo ex403, và bộ điều khiển trợ lực ex406 để

ghi và tái tạo thông tin qua đầu đọc quang ex401 trong khi được thao tác theo cách kết hợp. Bộ điều khiển hệ thống ex407 gồm có, ví dụ, bộ vi xử lý, và thực thi quy trình xử lý bằng cách khiến cho máy tính thực thi chương trình để đọc và ghi.

Mặc dù đầu đọc quang ex401 chiếu vệt laze như đã nêu trong phần mô tả, nó có thể thực hiện việc ghi mật độ cao sử dụng ánh sáng trường gần.

Fig.21 minh họa phương tiện lưu trữ ex215 là đĩa quang. Trên bề mặt ghi của phương tiện lưu trữ ex215, các rãnh dẫn được tạo nên dạng xoắn ốc, và rãnh thông tin ex230 ghi trước thông tin địa chỉ chỉ báo vị trí chính xác trên đĩa theo sự thay đổi theo hình dạng của các rãnh dẫn. Thông tin địa chỉ gồm có thông tin để xác định các vị trí để ghi các khối ex231 mà là đơn vị để ghi dữ liệu. Việc tái tạo rãnh thông tin ex230 và đọc thông tin địa chỉ trong thiết bị mà ghi và tái tạo dữ liệu có thể dẫn đến việc xác định các vị trí của các khối ghi. Hơn nữa, phương tiện lưu trữ ex215 gồm có vùng ghi dữ liệu ex233, vùng đường tròn trong ex232, và vùng đường tròn ngoài ex234. Vùng ghi dữ liệu ex233 là vùng để sử dụng trong việc ghi dữ liệu người dùng. Vùng đường tròn trong ex232 và vùng đường tròn ngoài ex234 lần lượt là bên trong và bên ngoài của vùng ghi dữ liệu ex233 là dành cho mục đích sử dụng riêng ngoại trừ để ghi dữ liệu người dùng. Bộ ghi/tái tạo thông tin 400 đọc và ghi âm thanh được mã hóa, dữ liệu video được mã hóa, hoặc dữ liệu được ghép kênh thu được nhờ ghép kênh dữ liệu âm thanh và video được mã hóa, từ và trên vùng ghi dữ liệu ex233 của phương tiện lưu trữ ex215.

Mặc dù đĩa quang có một lớp, chẳng hạn như DVD và BD được mô tả như ví dụ trong phần mô tả, đĩa quang không bị giới hạn ở đây, và có thể là đĩa quang có cấu trúc đa lớp và có khả năng được ghi lên một phần khác với bề mặt. Hơn nữa, đĩa quang có thể có cấu trúc để ghi/tái tạo đa chiều, chẳng hạn như ghi thông tin sử dụng ánh sáng màu với các bước sóng khác nhau trong cùng một phần của đĩa quang và để ghi thông tin có các lớp khác nhau từ các góc khác nhau.

Hơn nữa, xe ex210 có anten ex205 có thể nhận dữ liệu từ vệ tinh ex202 và các loại khác, và tái tạo video trên thiết bị hiển thị chẳng hạn như hệ thống điều hướng xe ex211 được đặt trong xe ex210, trong hệ thống phát quảng bá số ex200. Ở đây, cấu hình của hệ thống điều hướng xe ex211 sẽ là cấu hình, ví dụ, bao gồm bộ thu GPS từ cấu hình được minh họa trên Fig.19. Điều tương tự sẽ đúng với cấu hình của máy tính ex111, điện thoại di động ex114, và các loại khác.

Fig.22A minh họa điện thoại di động ex114 sử dụng phương pháp mã hóa hình ảnh động và phương pháp giải mã hình ảnh động được mô tả trong các phương án. Điện thoại di động ex114 gồm có: anten ex350 để truyền và nhận các sóng vô tuyến qua trạm gốc ex110; bộ camera ex365 có khả năng chụp các ảnh động và các ảnh tĩnh; và bộ hiển thị ex358 chẳng hạn như màn hình tinh thể lỏng để hiển thị dữ liệu chẳng hạn như được giải mã video được chụp bởi bộ camera ex365 hoặc nhận được bởi anten ex350. Điện thoại di động ex114 còn bao gồm: bộ thân chính bao gồm bộ phím thao tác ex366; bộ đầu ra âm thanh ex357 chẳng hạn như loa để phát ra âm thanh; bộ đầu vào âm thanh ex356 chẳng hạn như micro để nhập âm thanh vào; bộ nhớ ex367 để lưu trữ video hoặc các hình ảnh tĩnh được chụp, âm thanh được ghi, dữ liệu được mã hóa hoặc được giải mã của video thu được, các hình ảnh tĩnh, các thư điện tử, hoặc các loại khác; và bộ khe cắm ex364 là bộ giao diện để phương tiện lưu trữ lưu trữ dữ liệu theo cách tương tự như bộ nhớ ex367.

Tiếp theo, ví dụ về cấu hình của điện thoại di động ex114 sẽ được mô tả dựa vào Fig.22B. Trong điện thoại di động ex114, bộ điều khiển chính ex360 được thiết kế để điều khiển toàn bộ mỗi đơn vị của thân chính bao gồm bộ hiển thị ex358 cũng như bộ phím thao tác ex366 được kết nối chung, qua bus đồng bộ ex370, tới bộ mạch cấp nguồn ex361, bộ điều khiển đầu vào thao tác ex362, bộ xử lý tín hiệu video ex355, bộ giao diện camera ex363, bộ điều khiển màn hình tinh thể lỏng (LCD) ex359, bộ điều chế/giải điều chế ex352, bộ ghép kênh/giải ghép kênh ex353, bộ xử lý tín hiệu âm thanh ex354, bộ khe cắm ex364, và bộ nhớ ex367.

Khi phím kết thúc cuộc gọi hoặc phím nguồn được bật bởi thao tác của người dùng, bộ mạch cấp nguồn ex361 cấp cho các bộ phận tương ứng nguồn từ bộ pin để kích hoạt điện thoại di động ex114.

Trong điện thoại di động ex114, bộ xử lý tín hiệu âm thanh ex354 chuyển đổi các tín hiệu âm thanh được thu thập bởi bộ đầu vào âm thanh ex356 ở chế độ cuộc hội thoại thành các tín hiệu âm thanh số dưới sự điều khiển của bộ điều khiển chính ex360 bao gồm CPU, ROM, và RAM. Sau đó, bộ điều chế/giải điều chế ex352 thực hiện xử lý phổ trải rộng trên các tín hiệu âm thanh số, và bộ truyền và nhận ex351 thực hiện chuyển đổi số thành tương tự và chuyển đổi tần số trên dữ liệu, để truyền dữ liệu nhận được qua anten ex350. Ngoài ra, trong điện thoại di động ex114, bộ truyền và nhận ex351 khuếch đại dữ liệu nhận được bởi anten ex350 ở chế độ cuộc hội thoại và thực hiện chuyển đổi tần số và chuyển đổi tương tự thành số với dữ liệu. Sau đó, bộ điều chế/giải điều chế ex352 thực

hiện xử lý phổ trải rộng ngược trên dữ liệu, và bộ xử lý tín hiệu âm thanh ex354 chuyển đổi nó thành các tín hiệu âm thanh tương tự, để xuất chúng qua bộ đầu ra âm thanh ex357.

Hơn nữa, khi thư điện tử ở chế độ truyền thông dữ liệu được truyền, dữ liệu văn bản của thư điện tử được nhập vào bởi thao tác bộ phím thao tác ex366 và các loại khác của thân chính được gửi tới bộ điều khiển chính ex360 qua bộ điều khiển đầu vào thao tác ex362. Bộ điều khiển chính ex360 khiến cho bộ điều chế/giải điều chế ex352 thực hiện xử lý phổ trải rộng trên dữ liệu văn bản, và bộ truyền và nhận ex351 thực hiện chuyển đổi số thành tương tự và chuyển đổi tần số trên dữ liệu nhận được để truyền dữ liệu tới trạm gốc ex110 qua anten ex350. Khi nhận được thư điện tử, quy trình xử lý là gần như ngược lại với quy trình xử lý để truyền thư điện tử được thực hiện trên dữ liệu thu được, và dữ liệu nhận được được cấp tới bộ hiển thị ex358.

Khi video, các ảnh tĩnh, hoặc video và âm thanh ở chế độ truyền thông dữ liệu được truyền, bộ xử lý tín hiệu video ex355 nén và mã hóa các tín hiệu video được cấp từ bộ camera ex365 nhờ sử dụng phương pháp mã hóa hình ảnh động được thể hiện ở mỗi trong số các phương án (nghĩa là, dùng làm thiết bị mã hóa ảnh theo khía cạnh của sáng chế), và truyền dữ liệu video được mã hóa tới bộ ghép kênh/giải ghép kênh ex353. Ngược lại, trong khoảng thời gian khi bộ camera ex365 chụp video, các ảnh tĩnh, và các loại khác, bộ xử lý tín hiệu âm thanh ex354 mã hóa các tín hiệu âm thanh được thu thập bởi bộ đầu vào âm thanh ex356, và truyền dữ liệu âm thanh được mã hóa tới bộ ghép kênh/giải ghép kênh ex353.

Bộ ghép kênh/giải ghép kênh ex353 ghép kênh dữ liệu video đã mã hóa được cấp từ bộ xử lý tín hiệu video ex355 và dữ liệu âm thanh đã mã hóa được cấp từ bộ xử lý tín hiệu âm thanh ex354, bằng cách sử dụng phương pháp định trước. Sau đó, bộ điều chế/giải điều chế (bộ mạch điều chế/giải điều chế) ex352 thực hiện xử lý phổ trải rộng trên dữ liệu được ghép kênh, và bộ truyền và nhận ex351 thực hiện chuyển đổi số thành tương tự và chuyển đổi tần số trên dữ liệu để truyền dữ liệu nhận được qua anten ex350.

Khi nhận dữ liệu của tệp video được liên kết với trang Web và các loại khác ở chế độ truyền thông dữ liệu hoặc khi nhận thư điện tử có video và/hoặc âm thanh đính kèm, để giải mã dữ liệu được ghép kênh nhận được qua anten ex350, bộ ghép kênh/giải ghép kênh ex353 giải ghép kênh dữ liệu được ghép kênh thành dòng bit dữ liệu video và dòng bit dữ liệu âm thanh, và cấp cho bộ xử lý tín hiệu video ex355 dữ liệu video được mã hóa và cấp

cho bộ xử lý tín hiệu âm thanh ex354 dữ liệu âm thanh được mã hóa, qua bus đồng bộ ex370. Bộ xử lý tín hiệu video ex355 giải mã tín hiệu video bằng cách sử dụng phương pháp giải mã hình ảnh động tương ứng với phương pháp mã hóa hình ảnh động được thể hiện ở mỗi trong số các phương án (nghĩa là, dùng làm thiết bị giải mã ảnh theo khía cạnh của sáng chế), và sau đó bộ hiển thị ex358 hiển thị, chẳng hạn, video và các ảnh tĩnh được bao gồm trong tệp video được liên kết với trang Web qua bộ điều khiển LCD ex359. Hơn nữa, bộ xử lý tín hiệu âm thanh ex354 giải mã tín hiệu âm thanh, và bộ đầu ra âm thanh ex357 cung cấp âm thanh.

Hơn nữa, tương tự với máy thu hình ex300, có thể dùng cho thiết bị đầu cuối chẳng hạn như điện thoại di động ex114 để có 3 loại cấu hình thực hiện bao gồm không chỉ (i) thiết bị đầu cuối truyền và nhận bao gồm cả thiết bị mã hóa và thiết bị giải mã, mà còn (ii) thiết bị đầu cuối truyền chỉ bao gồm thiết bị mã hóa và (iii) thiết bị đầu cuối nhận chỉ bao gồm thiết bị giải mã. Mặc dù hệ thống phát quảng bá số ex200 nhận và truyền dữ liệu được ghép kênh thu được nhờ ghép kênh dữ liệu âm thanh trên dữ liệu video trong phần mô tả, dữ liệu được ghép kênh có thể là dữ liệu thu được nhờ ghép kênh không phải dữ liệu âm thanh mà là dữ liệu ký tự liên quan đến video trên dữ liệu video, và có thể không phải dữ liệu được ghép kênh mà chính là dữ liệu video.

Như vậy, phương pháp mã hóa hình ảnh động và phương pháp giải mã hình ảnh động ở mỗi trong số các phương án có thể được sử dụng trong các thiết bị và các hệ thống bất kỳ được mô tả. Do đó, có thể thu được các ưu điểm được mô tả ở mỗi trong số các phương án.

Hơn nữa, sáng chế không giới hạn ở các phương án, và các sửa đổi và các điều chỉnh khác nhau có thể được thực hiện mà không nằm ngoài phạm vi của sáng chế.

Phương án 10

Dữ liệu video có thể được tạo ra nhờ chuyển đổi, khi cần, giữa (i) phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động được thể hiện ở mỗi trong số các phương án và (ii) phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động phù hợp với chuẩn khác, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1.

Ở đây, khi các dữ liệu video phù hợp với các chuẩn khác nhau được tạo ra và sau đó được giải mã, các phương pháp giải mã cần được lựa chọn để phù hợp với các chuẩn khác nhau. Tuy nhiên, bởi vì không thể phát hiện được chuẩn mà mỗi trong số các dữ liệu

video được giải mã cần phù hợp, nên vấn đề là không thể chọn được phương pháp giải mã thích hợp.

Để giải quyết vấn đề này, dữ liệu được ghép kênh thu được bằng cách ghép kênh dữ liệu âm thanh và các loại khác trên dữ liệu video có cấu trúc bao gồm thông tin nhận dạng chỉ báo chuẩn nào mà dữ liệu video phù hợp. Cấu trúc cụ thể của dữ liệu được ghép kênh bao gồm dữ liệu video được tạo ra theo phương pháp mã hóa hình ảnh động và bởi thiết bị mã hóa hình ảnh động được thể hiện ở mỗi trong số các phương án sẽ được mô tả dưới đây. Dữ liệu được ghép kênh là dòng số theo định dạng dòng vận chuyển MPEG-2.

Fig.23 minh họa cấu trúc của dữ liệu được ghép kênh. Như được minh họa trên Fig.23, dữ liệu được ghép kênh có thể thu được nhờ ghép kênh ít nhất một trong số dòng video, dòng âm thanh, dòng đồ họa trình diễn (PG), và dòng đồ họa tương tác. Dòng video biểu diễn video sơ cấp và video thứ cấp của phim ảnh, dòng âm thanh (IG) biểu diễn phần âm thanh sơ cấp và phần âm thanh thứ cấp được trộn với phần âm thanh sơ cấp, và dòng đồ họa trình diễn biểu diễn phụ đề của bộ phim. Ở đây, video sơ cấp là video thông thường được hiển thị trên màn hình, và video thứ cấp là video được hiển thị trên cửa sổ nhỏ hơn trong video sơ cấp. Hơn nữa, dòng đồ họa tương tác biểu diễn màn hình tương tác được tạo ra bằng cách bố trí các thành phần GUI trên màn hình. Dòng video được mã hóa theo phương pháp mã hóa hình ảnh động hoặc bởi thiết bị mã hóa hình ảnh động được thể hiện ở mỗi trong số các phương án, hoặc theo phương pháp mã hóa hình ảnh động hoặc bởi thiết bị mã hóa hình ảnh động phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1. Dòng âm thanh được mã hóa phù hợp với chuẩn, chẳng hạn như Dolby-AC-3, Dolby Digital Plus, MLP, DTS, DTS-HD, và PCM tuyến tính.

Mỗi dòng được bao gồm trong dữ liệu được ghép kênh được định danh bởi PID. Ví dụ, 0x1011 được phân bổ cho dòng video được sử dụng cho video của bộ phim, 0x1100 đến 0x111F được phân bổ cho các dòng âm thanh, 0x1200 đến 0x121F được phân bổ cho các dòng đồ họa trình diễn, 0x1400 đến 0x141F được phân bổ cho các dòng đồ họa tương tác, 0x1B00 đến 0x1B1F được phân bổ cho các dòng video được sử dụng cho video thứ cấp của bộ phim, và 0x1A00 đến 0x1A1F được phân bổ cho các dòng âm thanh được sử dụng cho âm thanh thứ cấp được trộn với âm thanh sơ cấp.

Fig.24 minh họa dưới dạng sơ đồ cách dữ liệu được ghép kênh. Trước tiên, dòng video ex235 bao gồm các khung video và dòng âm thanh ex238 bao gồm âm thanh các

khung âm thanh được biến đổi thành dòng gồm các gói PES ex236 và dòng gồm các gói PES ex239, và còn thành các gói TS ex237 và các gói TS ex240 một cách tương ứng. Tương tự, dữ liệu của dòng đồ họa trình diễn ex241 và dữ liệu của dòng đồ họa tương tác ex244 được biến đổi thành dòng gồm các gói PES ex242 và dòng gồm các gói PES ex245, và còn thành các gói TS ex243 và các gói TS ex246 một cách tương ứng. Các gói TS này được ghép kênh thành dòng để nhận được dữ liệu được ghép kênh ex247.

Fig.25 minh họa cách thức dòng video được lưu trữ trong dòng gồm các gói PES một cách chi tiết hơn. Thanh thứ nhất trên Fig.25 thể hiện dòng khung video trong dòng video. Thanh thứ hai thể hiện dòng gồm các gói PES. Như được chỉ báo bởi các mũi tên được ký hiệu là yy1, yy2, yy3, và yy4 trên Fig.25, dòng video được chia thành các hình ảnh như các hình ảnh I, các hình ảnh B, và các hình ảnh P mà mỗi trong số đó là đơn vị trình diễn video, và các hình ảnh được lưu trữ trong tải tin của mỗi trong số các gói PES. Mỗi trong số các gói PES có phần đầu PES, và phần đầu PES lưu trữ dấu thời gian trình diễn (Presentation Time-Stamp - PTS) chỉ báo thời gian hiển thị của hình ảnh, và dấu thời gian giải mã (Decoding Time-Stamp - DTS) chỉ báo thời gian giải mã của hình ảnh.

Fig.26 minh họa định dạng của các gói TS cuối cùng được ghi trên dữ liệu được ghép kênh. Mỗi trong số các gói TS là gói có độ dài cố định 188 byte bao gồm phần đầu TS 4 byte có thông tin, chẳng hạn như PID để nhận dạng dòng và tải tin TS 184 byte dùng để lưu trữ dữ liệu. Các gói PES được chia, và được lưu trữ trong các tải tin TS một cách tương ứng. Khi BD ROM được sử dụng, mỗi trong số các gói TS được cung cấp TP_Extra_Header 4 byte, nhờ đó tạo ra các gói nguồn 192 byte. Các gói nguồn được ghi trên dữ liệu được ghép kênh. TP_Extra_Header lưu trữ thông tin chẳng hạn như dấu thời gian đến (Arrival_Time_Stamp - ATS). ATS thể hiện thời điểm bắt đầu truyền mà tại đó mỗi trong số các gói TS được truyền tới bộ lọc PID. Các gói nguồn được bố trí trong dữ liệu được ghép kênh như được thể hiện ở phần dưới cùng trên Fig.26. Các số tăng lên từ đầu của dữ liệu được ghép kênh được gọi là các số gói nguồn (source packet number - SPN).

Mỗi trong số các gói TS được bao gồm trong dữ liệu được ghép kênh gồm có không chỉ các dòng âm thanh, video, các phụ đề và các loại khác, mà còn bảng kết hợp chương trình (Program Association Table - PAT), bảng ánh xạ chương trình (Program Map Table - PMT), và chuẩn đồng hồ chương trình (Program Clock Reference - PCR). PAT thể hiện PID trong PMT được sử dụng trong dữ liệu được ghép kênh chỉ báo gì, và PID của chính

PAT được đăng ký là không. PMT lưu trữ các PID của các dòng video, âm thanh, các phụ đề và các loại khác được bao gồm trong dữ liệu được ghép kênh, và thông tin thuộc tính của các dòng tương ứng với các PID. PMT cũng có các phần mô tả khác nhau liên quan đến dữ liệu được ghép kênh. Các phần mô tả có thông tin chẳng hạn như thông tin điều khiển sao chép thể hiện xem việc sao chép của dữ liệu được ghép kênh được cho phép hay không. PCR lưu trữ thông tin thời gian STC tương ứng với ATS thể hiện khi gói PCR được truyền tới bộ giải mã, để đạt được sự đồng bộ giữa đồng hồ thời gian đến (Arrival Time Clock - ATC) mà là trục thời gian của các ATS, và đồng hồ thời gian hệ thống (System Time Clock - STC) mà là trục thời gian của các PTS và các DTS.

Fig.27 minh họa cấu trúc dữ liệu của PMT một cách chi tiết. Phần đầu PMT được bố trí ở trên cùng của PMT. Phần đầu PMT mô tả độ dài của dữ liệu được bao gồm trong PMT và các loại khác. Các phần mô tả liên quan đến dữ liệu đã ghép kênh được bố trí sau phần đầu PMT. Thông tin chẳng hạn như thông tin điều khiển sao chép được mô tả trong các phần mô tả. Sau các phần mô tả, các đoạn của thông tin dòng liên quan đến các dòng được bao gồm trong dữ liệu được ghép kênh được bố trí. Mỗi đoạn của thông tin dòng gồm có các phần mô tả dòng cho mỗi mô tả thông tin, chẳng hạn như loại dòng để nhận dạng bộ mã hóa-giải mã nén của dòng, PID dòng, và thông tin thuộc tính dòng (chẳng hạn như tốc độ khung hoặc tỉ lệ khung hình). Các phần mô tả dòng là bằng về số lượng với số lượng dòng trong dữ liệu được ghép kênh.

Khi dữ liệu được ghép kênh được ghi lên phương tiện lưu trữ và các loại khác, nó được ghi cùng với các tệp thông tin dữ liệu được ghép kênh.

Mỗi trong số các tệp thông tin dữ liệu được ghép kênh là thông tin quản lý của dữ liệu được ghép kênh như được thể hiện trên Fig.28. Các tệp thông tin dữ liệu được ghép kênh là tương ứng một-một với dữ liệu được ghép kênh, và mỗi trong số các tệp gồm có thông tin dữ liệu được ghép kênh, thông tin thuộc tính dòng, và ánh xạ mục nhập.

Như được minh họa trên Fig.28, thông tin dữ liệu được ghép kênh gồm có tốc độ hệ thống, thời gian bắt đầu tái tạo, và thời gian kết thúc tái tạo. Tốc độ hệ thống chỉ báo tốc độ truyền lớn nhất mà tại đó bộ giải mã đích hệ thống được mô tả dưới đây truyền dữ liệu được ghép kênh tới bộ lọc PID. Các khoảng thời gian ATS được bao gồm trong dữ liệu đã ghép kênh được thiết lập không cao hơn so với tốc độ hệ thống. Thời gian bắt đầu tái tạo chỉ báo PTS trong khung video ở đầu của dữ liệu được ghép kênh. Khoảng thời gian của

một khung được bổ sung vào PTS trong khung video ở cuối của dữ liệu được ghép kênh, và PTS được đặt bằng thời gian kết thúc tái tạo.

Như được thể hiện trên Fig.29, đoạn thông tin thuộc tính được đăng ký trong thông tin thuộc tính dòng, dùng cho mỗi PID của mỗi dòng được bao gồm trong dữ liệu được ghép kênh. Mỗi đoạn thông tin thuộc tính có thông tin khác nhau phụ thuộc vào việc dòng tương ứng là dòng video, dòng âm thanh, dòng đồ họa trình diễn, hay dòng đồ họa tương tác. Mỗi đoạn của thông tin thuộc tính dòng video mang thông tin bao gồm loại bộ mã hóa-giải mã nén nào được sử dụng để nén dòng video, và độ phân giải, tỉ lệ khung hình và tốc độ khung của các đoạn của dữ liệu hình ảnh được bao gồm trong dòng video. Mỗi đoạn của thông tin thuộc tính dòng âm thanh mang thông tin bao gồm loại bộ mã hóa-giải mã nén nào được sử dụng để nén dòng âm thanh, bao nhiêu kênh được bao gồm trong dòng âm thanh, ngôn ngữ nào mà dòng âm thanh hỗ trợ, và tần số lấy mẫu cao bao nhiêu. Thông tin thuộc tính dòng video và thông tin thuộc tính dòng âm thanh được sử dụng để thiết lập của bộ giải mã trước khi trình phát phương tiện phát lại thông tin.

Trong phương án hiện thời, dữ liệu được ghép kênh được sử dụng là loại dòng được bao gồm trong PMT. Hơn nữa, khi dữ liệu được ghép kênh được ghi lên phương tiện lưu trữ, thông tin thuộc tính dòng video được bao gồm trong thông tin dữ liệu đã ghép kênh được sử dụng. Cụ thể hơn là, phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án gồm có bước hoặc bộ phận để gán thông tin duy nhất chỉ báo dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động ở mỗi trong số các phương án, cho loại dòng được bao gồm trong PMT hoặc thông tin thuộc tính dòng video. Với cấu hình này, dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án có thể được phân biệt với dữ liệu video để phù hợp với chuẩn khác.

Hơn nữa, Fig.30 minh họa các bước của phương pháp giải mã hình ảnh động theo phương án hiện thời. Ở bước exS100, loại dòng được bao gồm trong PMT hoặc thông tin thuộc tính dòng video được bao gồm trong thông tin dữ liệu được ghép kênh thu được từ dữ liệu được ghép kênh. Tiếp theo, ở bước exS101, xác định xem loại dòng hoặc thông tin thuộc tính dòng video chỉ báo rằng dữ liệu được ghép kênh được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động ở mỗi trong số các phương án hay không. Khi xác định được rằng loại dòng hoặc thông tin thuộc tính dòng video chỉ báo

rằng dữ liệu đã ghép kênh được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động ở mỗi trong số các phương án, ở bước exS102, bước giải mã được thực hiện bởi phương pháp giải mã hình ảnh động ở mỗi trong số các phương án. Hơn nữa, khi loại dòng hoặc thông tin thuộc tính dòng video chỉ báo sự phù hợp với các chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1, ở bước exS103, bước giải mã được thực hiện bởi phương pháp giải mã hình ảnh động phù hợp với các chuẩn thông thường.

Như vậy, việc gán giá trị duy nhất mới cho loại dòng hoặc thông tin thuộc tính dòng video cho phép xác định xem phương pháp giải mã hình ảnh động hoặc thiết bị giải mã hình ảnh động được mô tả ở mỗi trong số các phương án có thể thực hiện việc giải mã hay không. Ngay cả khi dữ liệu được ghép kênh phù hợp với chuẩn khác được nhập vào, phương pháp hoặc thiết bị giải mã thích hợp có thể được lựa chọn. Do đó, có thể giải mã thông tin mà không có bất kỳ lỗi nào. Hơn nữa, phương pháp hoặc thiết bị mã hóa hình ảnh động, hoặc phương pháp hoặc thiết bị giải mã hình ảnh động trong phương án hiện thời có thể được sử dụng trong các thiết bị và các hệ thống được nêu trên.

Phương án 11

Mỗi trong số phương pháp mã hóa hình ảnh động, thiết bị mã hóa hình ảnh động, phương pháp giải mã hình ảnh động, và thiết bị giải mã hình ảnh động ở mỗi trong số các phương án thông thường có thể đạt được ở dạng mạch tích hợp hoặc mạch tích hợp quy mô lớn (Large Scale Integrated - LSI). Là ví dụ về LSI, Fig.31 minh họa cấu hình của LSI ex500 được làm thành một chip. LSI ex500 gồm có các thành phần ex501, ex502, ex503, ex504, ex505, ex506, ex507, ex508, và ex509 được mô tả dưới đây, và các thành phần này được nối với nhau qua bus ex510. Bộ mạch cấp nguồn ex505 được kích hoạt bằng cách cấp nguồn cho mỗi trong số các thành phần khi bộ mạch cấp nguồn ex505 được bật.

Ví dụ, khi việc mã hóa được thực hiện, LSI ex500 nhận tín hiệu AV từ micro ex117, camera ex113, và các loại khác qua IO AV ex509 dưới sự điều khiển của bộ điều khiển ex501 bao gồm CPU ex502, bộ điều khiển bộ nhớ ex503, bộ điều khiển dòng ex504, và bộ điều khiển tần số dẫn ex512. Tín hiệu AV thu được được lưu trữ tạm thời trong bộ nhớ ngoài ex511, chẳng hạn như SDRAM. Dưới sự điều khiển của bộ điều khiển ex501, dữ liệu đã lưu trữ được phân đoạn thành các phần dữ liệu theo lượng và tốc độ xử lý được truyền tới bộ xử lý tín hiệu ex507. Sau đó, bộ xử lý tín hiệu ex507 mã hóa tín hiệu âm thanh

và/hoặc tín hiệu video. Ở đây, việc mã hóa tín hiệu video là mã hóa được mô tả ở mỗi trong số các phương án. Hơn nữa, bộ xử lý tín hiệu ex507 đôi khi ghép kênh dữ liệu âm thanh được mã hóa và dữ liệu video được mã hóa, và IO dòng ex506 cung cấp dữ liệu được ghép kênh ra ngoài. Dữ liệu đã ghép kênh được cung cấp được truyền tới trạm gốc ex107, hoặc được ghi lên phương tiện lưu trữ ex215. Khi các tập hợp dữ liệu được ghép kênh, dữ liệu sẽ được lưu trữ tạm thời trong bộ đệm ex508 sao cho các tập hợp dữ liệu được đồng bộ hóa với nhau.

Mặc dù bộ nhớ ex511 là thành phần bên ngoài LSI ex500, nó có thể được bao gồm trong LSI ex500. Bộ đệm ex508 không giới hạn ở một bộ đệm, mà có thể bao gồm các bộ đệm. Hơn nữa, LSI ex500 có thể được làm thành một chip hoặc các chip.

Hơn nữa, mặc dù bộ điều khiển ex501 gồm có CPU ex502, bộ điều khiển bộ nhớ ex503, bộ điều khiển dòng ex504, bộ điều khiển tần số dẫn ex512, nhưng cấu hình của bộ điều khiển ex501 không bị giới hạn ở đây. Ví dụ, bộ xử lý tín hiệu ex507 có thể còn bao gồm CPU. Việc bao gồm CPU khác trong bộ xử lý tín hiệu ex507 có thể nâng cao tốc độ xử lý. Hơn nữa, theo ví dụ khác, CPU ex502 có thể dùng làm hoặc là một phần của bộ xử lý tín hiệu ex507, và, ví dụ, có thể bao gồm bộ xử lý tín hiệu âm thanh. Trong trường hợp như vậy, bộ điều khiển ex501 gồm có bộ xử lý tín hiệu ex507 hoặc CPU ex502 bao gồm một phần của bộ xử lý tín hiệu ex507.

Tên được sử dụng ở đây là LSI, nhưng nó có thể cũng được gọi là IC, LSI hệ thống, siêu LSI, hoặc LSI siêu cao phụ thuộc vào mức độ tích hợp.

Hơn nữa, các cách đạt được sự tích hợp không giới hạn ở LSI, và mạch chuyên dụng hoặc bộ xử lý đa dụng, v.v. cũng có thể được tích hợp. Mảng cổng lập trình được theo trường (Field Programmable Gate Array - FPGA) có thể được lập trình sau khi sản xuất các LSI hoặc bộ xử lý có thể cấu hình lại cho phép tạo cấu hình lại kết nối hoặc cấu hình của LSI có thể được sử dụng đối với cùng mục đích. Thiết bị logic lập trình được như vậy thông thường có thể thực hiện phương pháp mã hóa hình ảnh động và/hoặc phương pháp giải mã hình ảnh động theo phương án bất kỳ trong số các phương án trên, nhờ tải hoặc đọc từ bộ nhớ hoặc tương tự một hoặc nhiều chương trình được bao gồm trong phần mềm hoặc firmware.

Trong tương lai, với sự cải tiến trong công nghệ bán dẫn, công nghệ hoàn toàn mới có thể thay thế LSI. Các khối chức năng có thể được tích hợp sử dụng công nghệ như vậy.

Có khả năng để sáng chế được ứng dụng cho công nghệ sinh học.

Phương án 12

Khi dữ liệu video được tạo ra trong phương pháp mã hóa hình ảnh động hoặc bởi thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án được giải mã, lượng xử lý có thể tăng lên so với khi dữ liệu video phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1 được giải mã. Do đó, LSI ex500 cần được thiết lập bằng tần số dẫn cao hơn so với tần số của CPU ex502 được sử dụng khi dữ liệu video phù hợp với chuẩn thông thường được giải mã. Tuy nhiên, khi tần số dẫn được thiết lập cao hơn, có vấn đề là tăng công suất tiêu thụ.

Để giải quyết vấn đề này, thiết bị giải mã hình ảnh động, chẳng hạn như máy thu hình ex300 và LSI ex500 được cấu hình để xác định chuẩn nào mà dữ liệu video phù hợp, và chuyển mạch giữa các tần số dẫn theo chuẩn được xác định. Fig.32 minh họa cấu hình ex800 theo phương án hiện thời. Bộ chuyển mạch tần số dẫn ex803 thiết lập tần số dẫn tới tần số dẫn cao hơn khi dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án. Sau đó, bộ chuyển mạch tần số dẫn ex803 lệnh cho bộ xử lý giải mã ex801 thực hiện phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án để giải mã dữ liệu video. Khi dữ liệu video phù hợp với chuẩn thông thường, bộ chuyển mạch tần số dẫn ex803 thiết lập tần số dẫn bằng tần số dẫn thấp hơn so với tần số dẫn của dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động hoặc thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án. Sau đó, bộ chuyển mạch tần số dẫn ex803 lệnh cho bộ xử lý giải mã ex802 mà phù hợp với chuẩn thông thường để giải mã dữ liệu video.

Cụ thể hơn là, bộ chuyển mạch tần số dẫn ex803 gồm có CPU ex502 và bộ điều khiển tần số dẫn ex512 trên Fig.31. Ở đây, mỗi trong số bộ xử lý giải mã ex801 thực hiện phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án và bộ xử lý giải mã ex802 mà phù hợp với chuẩn thông thường tương ứng với bộ xử lý tín hiệu ex507 trên Fig.31. CPU ex502 xác định chuẩn nào mà dữ liệu video phù hợp. Sau đó, bộ điều khiển tần số dẫn ex512 xác định tần số dẫn dựa vào tín hiệu từ CPU ex502. Hơn nữa, bộ xử lý tín hiệu ex507 giải mã dữ liệu video dựa vào tín hiệu từ CPU ex502. Ví dụ, có thể là thông tin định danh được mô tả trong phương án 10 được sử dụng để nhận dạng dữ liệu video. Thông tin định danh không giới hạn ở thông tin được mô tả trong phương án 10 mà

có thể là thông tin bất kỳ miễn là thông tin chỉ báo chuẩn nào mà dữ liệu video phù hợp. Ví dụ, khi chuẩn nào mà dữ liệu video phù hợp có thể được xác định dựa vào tín hiệu bên ngoài để xác định rằng dữ liệu video được sử dụng cho máy thu hình hoặc đĩa, v.v., việc xác định có thể được thực hiện dựa vào tín hiệu bên ngoài như vậy. Hơn nữa, CPU ex502 lựa chọn tần số dẫn dựa vào, ví dụ, bảng tìm kiếm trong đó các chuẩn của dữ liệu video được kết hợp với các tần số dẫn như được thể hiện trên Fig.34. Tần số dẫn có thể được lựa chọn bằng cách lưu trữ bảng tìm kiếm trong bộ đệm ex508 và trong bộ nhớ trong của LSI, và dựa vào bảng tìm kiếm bởi CPU ex502.

Fig.33 minh họa các bước để thực hiện phương pháp trong phương án hiện thời. Thứ nhất, ở bước exS200, bộ xử lý tín hiệu ex507 thu nhận thông tin định danh từ dữ liệu được ghép kênh. Tiếp theo, ở bước exS201, CPU ex502 xác định xem dữ liệu video được tạo ra bởi phương pháp mã hóa và thiết bị mã hóa được mô tả ở mỗi trong số các phương án hay không, dựa vào thông tin định danh. Khi dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án, ở bước exS202, CPU ex502 truyền tín hiệu để thiết lập tần số dẫn bằng tần số dẫn cao hơn cho bộ điều khiển tần số dẫn ex512. Sau đó, bộ điều khiển tần số dẫn ex512 thiết lập tần số dẫn bằng tần số dẫn cao hơn. Mặt khác, khi thông tin định danh chỉ báo rằng dữ liệu video phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1, ở bước exS203, CPU ex502 truyền tín hiệu để thiết lập tần số dẫn bằng tần số dẫn thấp hơn cho bộ điều khiển tần số dẫn ex512. Sau đó, bộ điều khiển tần số dẫn ex512 thiết lập tần số dẫn bằng tần số dẫn thấp hơn so với tần số trong trường hợp dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả trong mỗi trong số các phương án.

Hơn nữa, cùng với việc chuyển mạch các tần số dẫn, hiệu quả bảo toàn công suất có thể được nâng cao nhờ thay đổi điện áp được áp dụng cho LSI ex500 hoặc thiết bị bao gồm LSI ex500. Ví dụ, khi tần số dẫn được thiết lập thấp hơn, có thể điện áp được áp dụng cho LSI ex500 hoặc thiết bị bao gồm LSI ex500 được thiết lập bằng điện áp thấp hơn so với điện áp trong trường hợp tần số dẫn được thiết lập cao hơn.

Hơn nữa, khi lượng xử lý để giải mã là lớn hơn, tần số dẫn có thể được thiết lập cao hơn, và khi lượng xử lý để giải mã là nhỏ hơn, tần số dẫn có thể được thiết lập thấp hơn như phương pháp để thiết lập tần số dẫn. Do đó, phương pháp thiết lập là không giới hạn ở các phương pháp được nêu trên. Ví dụ, khi lượng xử lý để giải mã dữ liệu video phù hợp

với MPEG-4 AVC là lớn hơn so với lượng xử lý để giải mã dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án, có thể là tần số dẫn được thiết lập theo thứ tự ngược lại với việc thiết lập được nêu trên.

Hơn nữa, phương pháp để thiết lập tần số dẫn không giới hạn ở phương pháp để thiết lập tần số dẫn thấp hơn. Ví dụ, khi thông tin định danh chỉ báo rằng dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án, có thể điện áp được áp dụng cho LSI ex500 hoặc thiết bị bao gồm LSI ex500 được thiết lập cao hơn. Khi thông tin định danh chỉ báo rằng dữ liệu video phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1, có thể là điện áp được áp dụng cho LSI ex500 hoặc thiết bị bao gồm LSI ex500 được thiết lập thấp hơn. Theo ví dụ khác, có thể là, khi thông tin định danh chỉ báo rằng dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án, việc điều khiển CPU ex502 không bị treo, và khi thông tin định danh chỉ báo rằng dữ liệu video phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1, việc điều khiển của CPU ex502 bị treo ở thời điểm cho trước bởi vì CPU ex502 có khả năng xử lý bổ sung. Có thể là, ngay cả khi thông tin định danh chỉ báo rằng dữ liệu video được tạo ra bởi phương pháp mã hóa hình ảnh động và thiết bị mã hóa hình ảnh động được mô tả ở mỗi trong số các phương án, trong trường hợp mà CPU ex502 có khả năng xử lý bổ sung, việc điều khiển của CPU ex502 bị treo ở thời điểm cho trước. Trong trường hợp như vậy, có thể là thời gian treo được thiết lập ngắn hơn so với thời gian treo trong trường hợp thông tin định danh chỉ báo rằng dữ liệu video phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1.

Theo đó, hiệu quả bảo toàn công suất có thể được nâng cao bằng cách chuyển mạch giữa các tần số dẫn theo chuẩn mà dữ liệu video phù hợp. Hơn nữa, khi LSI ex500 hoặc thiết bị bao gồm LSI ex500 được điều khiển bằng cách sử dụng pin, tuổi thọ pin có thể được kéo dài với hiệu quả bảo toàn công suất.

Phương án 13

Có các trường hợp ở đó các dữ liệu video phù hợp với các chuẩn khác nhau được cung cấp tới các thiết bị và các hệ thống, chẳng hạn như máy thu hình và điện thoại di

động. Để cho phép giải mã các dữ liệu video phù hợp với các chuẩn khác nhau, bộ xử lý tín hiệu ex507 của LSI ex500 cần phù hợp với các chuẩn khác nhau. Tuy nhiên, các vấn đề về tăng cỡ mạch của LSI ex500 và tăng chi phí phát sinh cùng với việc sử dụng riêng các bộ xử lý tín hiệu ex507 phù hợp với các chuẩn tương ứng.

Để giải quyết vấn đề này, cần có cấu hình trong đó bộ xử lý giải mã để thực hiện phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án và bộ xử lý giải mã phù hợp với chuẩn thông thường, chẳng hạn như MPEG-2, MPEG-4 AVC, và VC-1 được dùng chung một phần. Ex900 trên Fig.35A thể hiện ví dụ về cấu hình này. Ví dụ, phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án và phương pháp giải mã hình ảnh động phù hợp với MPEG-4 AVC có, một phần là chung, chi tiết về quy trình xử lý, chẳng hạn như mã hóa entropy, lượng tử hóa ngược, lọc giải khối, và dự đoán bù chuyển động. Bộ xử lý giải mã ex902 phù hợp với MPEG-4 AVC có thể được dùng chung bởi các thao tác xử lý chung, và bộ xử lý giải mã chuyên dụng ex901 có thể được sử dụng cho quy trình xử lý riêng theo khía cạnh của sáng chế và không phù hợp với MPEG-4 AVC. Bộ xử lý giải mã ex902 phù hợp với MPEG-4 AVC có thể được dùng chung bởi các thao tác xử lý chung, và bộ xử lý giải mã chuyên dụng ex901 có thể được sử dụng cho quy trình xử lý riêng theo khía cạnh của sáng chế và không phù hợp với MPEG-4 AVC. Bộ xử lý giải mã để thực hiện phương pháp giải mã hình ảnh động được mô tả ở mỗi trong số các phương án có thể được dùng chung đối với quy trình xử lý được dùng chung, và bộ xử lý giải mã chuyên dụng có thể được sử dụng cho quy trình xử lý riêng với quy trình xử lý của MPEG-4 AVC.

Hơn nữa, ex1000 trên Fig.35B thể hiện ví dụ khác trong đó quy trình xử lý được dùng chung một phần. Ví dụ này sử dụng cấu hình bao gồm bộ xử lý giải mã chuyên dụng ex1001 để hỗ trợ quy trình xử lý riêng theo khía cạnh của sáng chế, bộ xử lý giải mã chuyên dụng ex1002 để hỗ trợ quy trình xử lý riêng theo chuẩn thông thường khác, và bộ xử lý giải mã ex1003 để hỗ trợ quy trình xử lý được dùng chung giữa phương pháp giải mã hình ảnh động theo khía cạnh của sáng chế và phương pháp giải mã hình ảnh động thông thường. Ở đây, các bộ xử lý giải mã chuyên dụng ex1001 và ex1002 không nhất thiết lần lượt chuyên dụng cho quy trình xử lý theo khía cạnh của sáng chế và quy trình xử lý của chuẩn thông thường, và có thể là các bộ xử lý giải mã có khả năng thực hiện việc xử lý chung. Hơn nữa, cấu hình theo phương án hiện thời có thể được thực hiện bởi LSI ex500.

Như vậy, có thể làm giảm cỡ mạch của LSI và làm giảm chi phí bằng cách dùng

chung bộ xử lý giải mã dùng cho quy trình xử lý được dùng chung giữa phương pháp giải mã hình ảnh động theo khía cạnh của sáng chế và phương pháp giải mã hình ảnh động phù hợp với chuẩn thông thường.

Khả năng áp dụng công nghiệp

Phương pháp mã hóa ảnh và phương pháp giải mã ảnh theo sáng chế có thể được ứng dụng cho dữ liệu đa phương tiện khác nhau. Phương pháp mã hóa ảnh và phương pháp giải mã ảnh theo sáng chế là hữu hiệu khi dùng làm phương pháp mã hóa ảnh và phương pháp giải mã ảnh trong việc lưu trữ, truyền, truyền thông, và tương tự sử dụng điện thoại di động, thiết bị DVD, máy tính cá nhân, và thiết bị tương tự.

Danh mục ký hiệu tham chiếu

100 bộ mã hóa

105 bộ trừ

110 bộ biến đổi

120 bộ lượng tử hóa

130, 230 bộ biến đổi ngược

140, 240 bộ cộng

150, 250 bộ lọc giải khối

160, 260 bộ lọc vòng lặp thích ứng

170, 270 bộ nhớ khung

180, 280 bộ dự đoán

190 bộ mã hóa entropy

200 bộ giải mã

290 bộ giải mã entropy

300, 400, 710 hình ảnh

31, 32, 3i, 41, 42 hàng LCU

311, 312, 3i1, 321 LCU

500 phần đầu gói

510 phần đầu IP

520, 550 trường mở rộng

530 phần đầu UDP

540 phần đầu RTP

560 phần đầu tải tin

570 phần đầu NAL

s1 tín hiệu đầu vào

s2 tín hiệu dự đoán

e, e' tín hiệu lỗi dự đoán

s', s'', s3 tín hiệu được tái tạo

YÊU CẦU BẢO HỘ

1. Phương pháp mã hóa ảnh để thực hiện xử lý mã hóa bằng cách phân chia hình ảnh thành các lát, phương pháp mã hóa ảnh bao gồm các bước:

truyền, nhờ sử dụng bộ mã hóa, dòng bit gồm có: cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc trên đó việc xử lý mã hóa được thực hiện phụ thuộc vào kết quả của việc xử lý mã hóa trên lát khác với lát hiện thời hay không; địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời; và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không,

trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát,

địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và

chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

2. Thiết bị mã hóa ảnh để thực hiện xử lý mã hóa bằng cách phân chia hình ảnh thành các lát, thiết bị mã hóa ảnh bao gồm:

bộ mã hóa để truyền dòng bit gồm có: cờ cho phép lát phụ thuộc chỉ báo xem hình ảnh có bao gồm lát phụ thuộc trên đó việc xử lý mã hóa được thực hiện phụ thuộc vào kết quả của việc xử lý mã hóa trên lát khác với lát hiện thời hay không; địa chỉ lát chỉ báo vị trí bắt đầu của lát hiện thời; và chỉ báo phụ thuộc chỉ báo xem lát hiện thời có là lát phụ thuộc hay không,

trong đó cờ cho phép lát phụ thuộc được bố trí trong tập hợp thông số chung cho các lát,

địa chỉ lát được bố trí trong phần đầu lát của lát hiện thời, và

chỉ báo phụ thuộc được bố trí trong phần đầu lát, và được bố trí trước địa chỉ lát và sau phần tử cú pháp nhận dạng tập hợp thông số.

FIG. 1

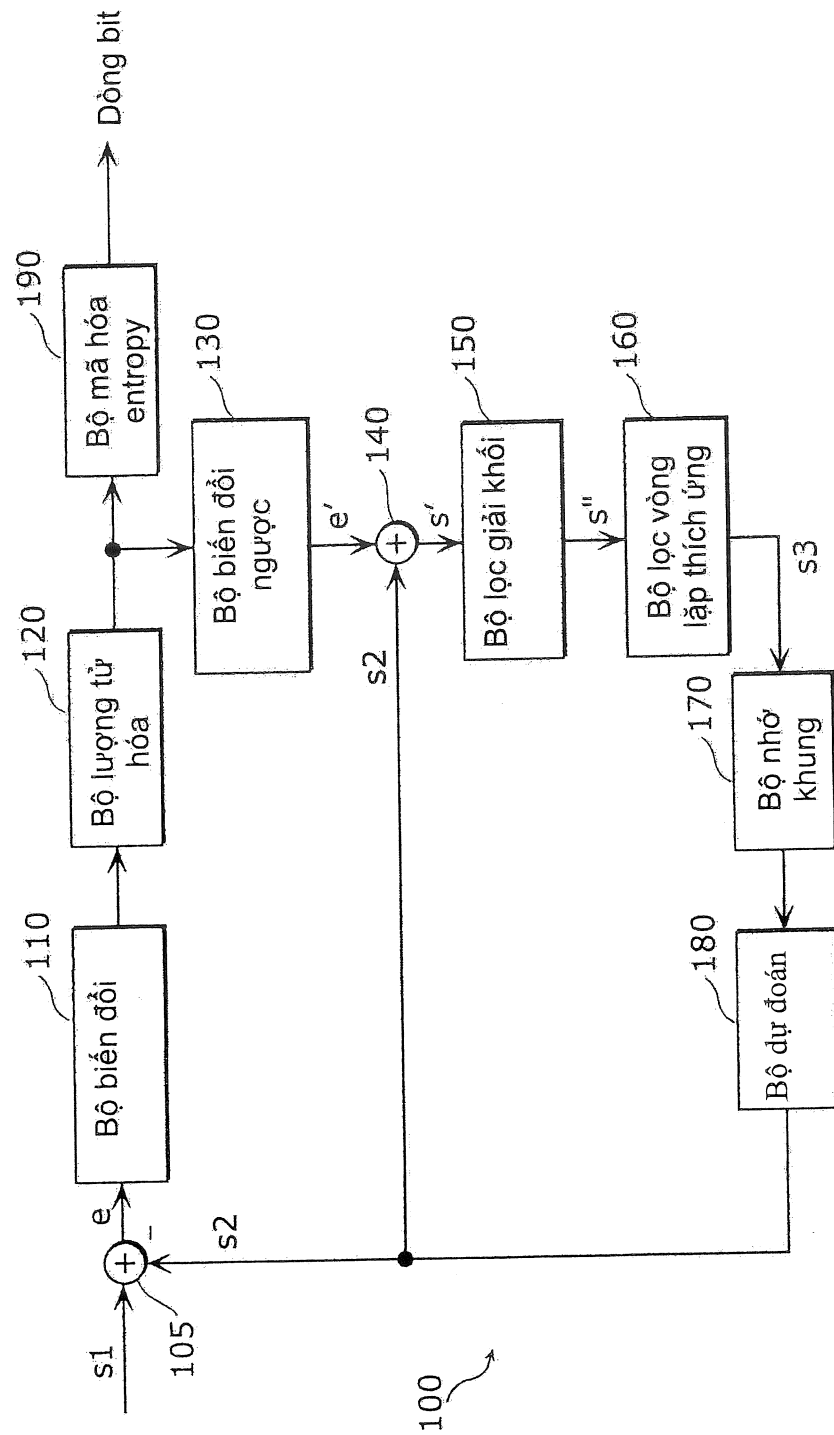


FIG. 2

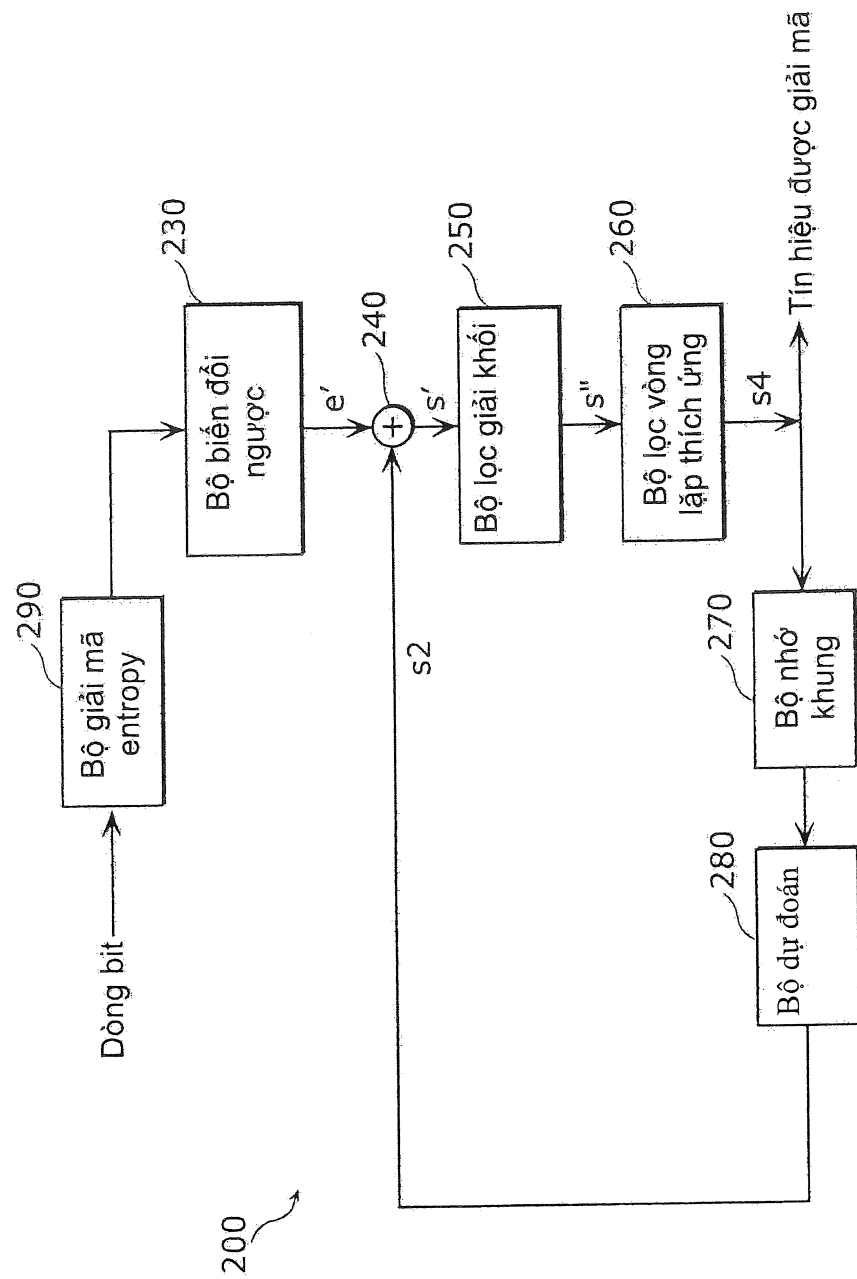


FIG. 3

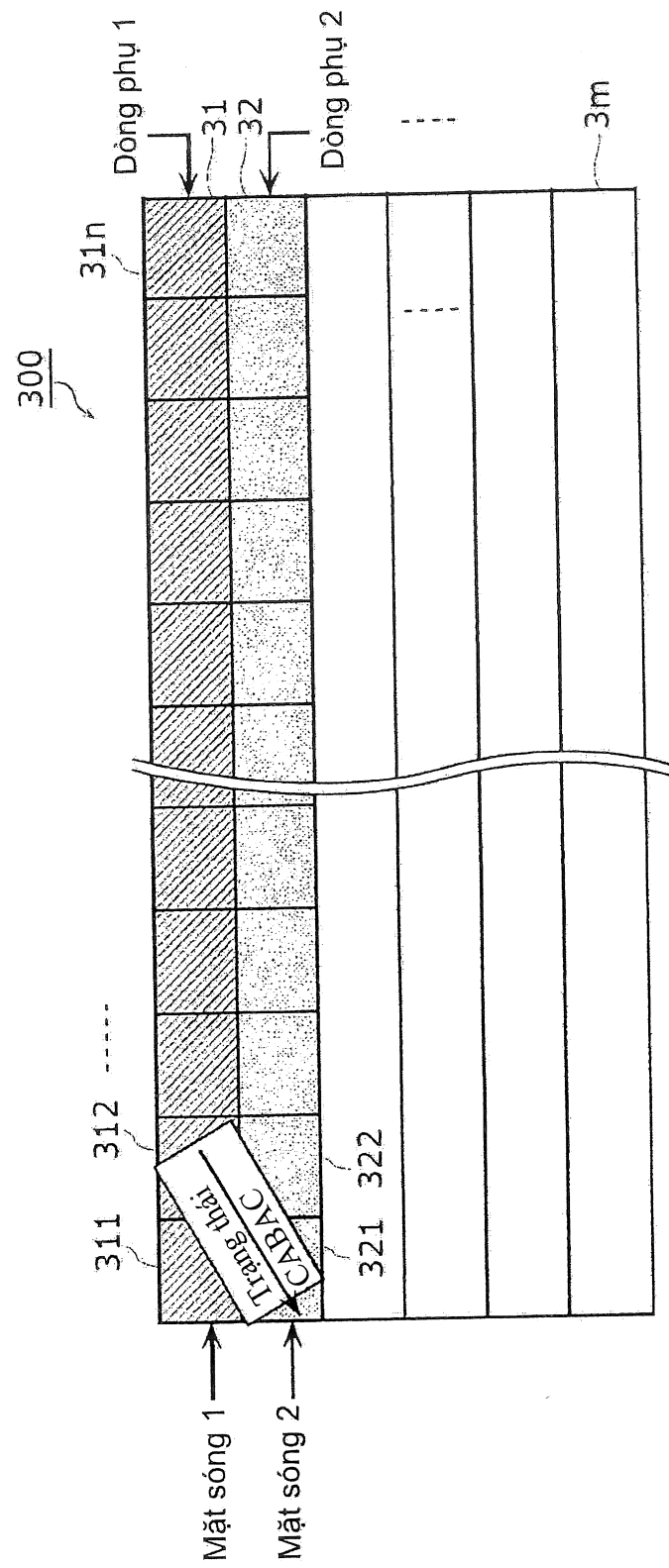


FIG. 4

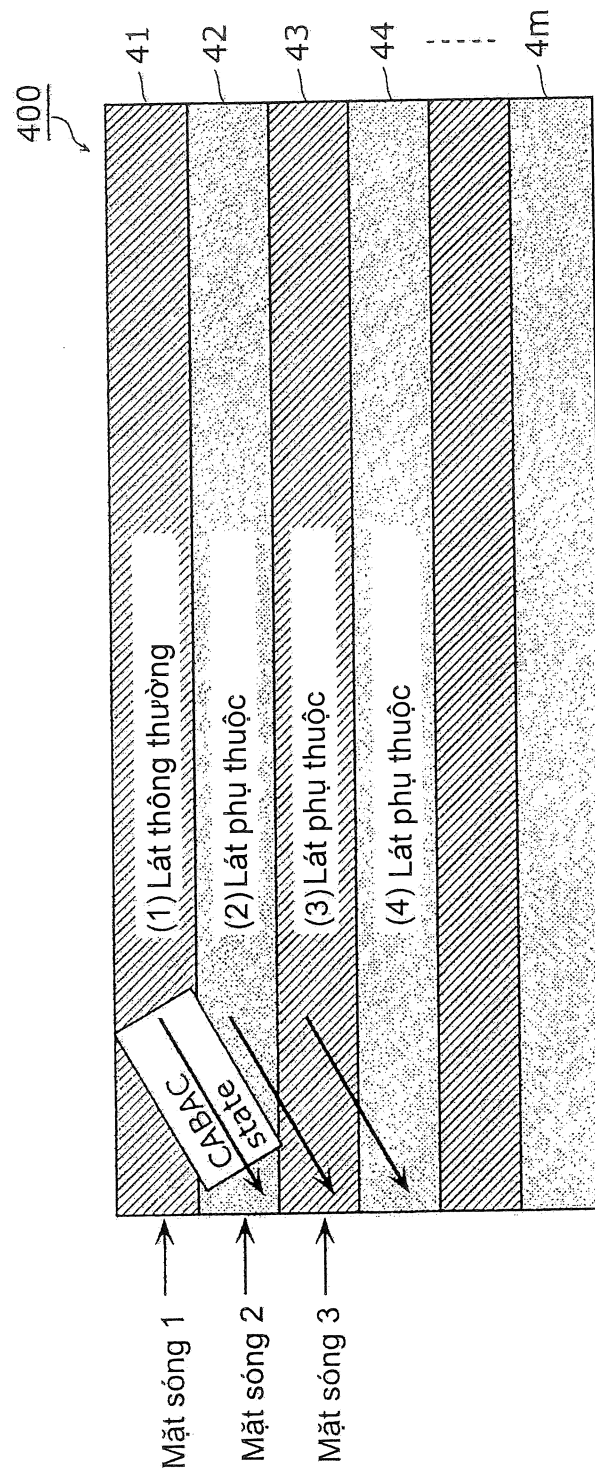


FIG. 5

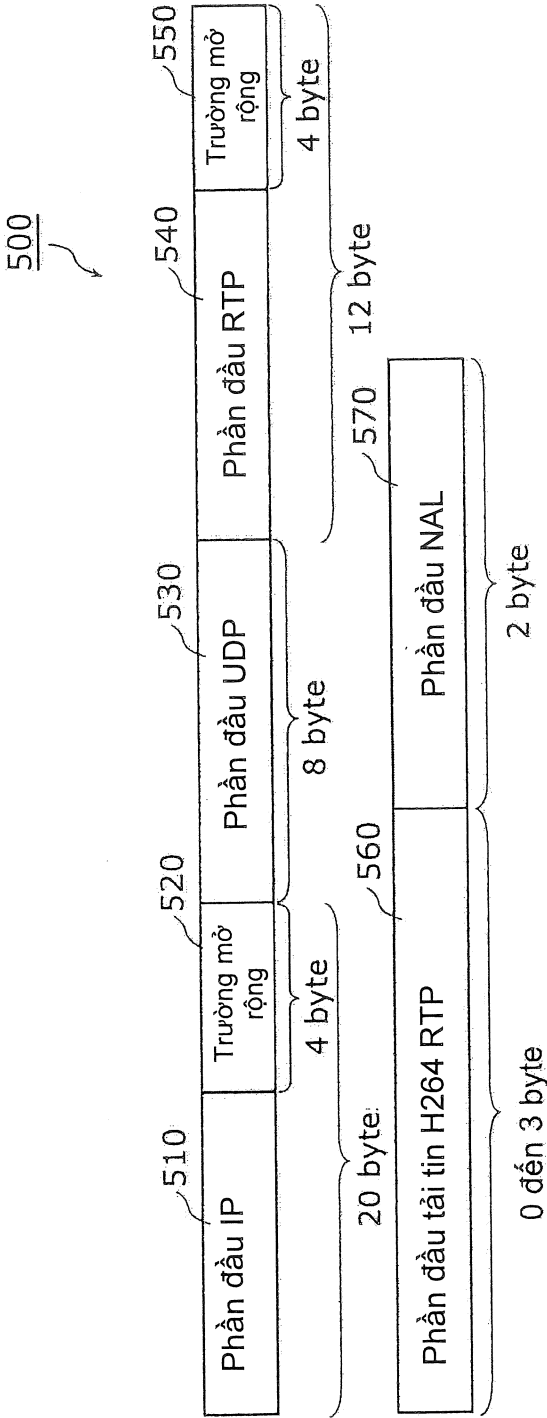


FIG. 6

600

slice_header() {	Phần mô tả
first_slice_in_pic_flag	u(1)
if(RapPicFlag)	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(dependent_slice_flag == 0) slice_header_info	
if(tiles_or_entropy_coding_sync_idc == 1	
tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
}	

Cờ lát phụ thuộc

601

Số lượng dòng CABAC trong lát

602

Số lượng byte của dòng phụ (điều hướng để hơn trong dòng bit)

603

FIG. 7

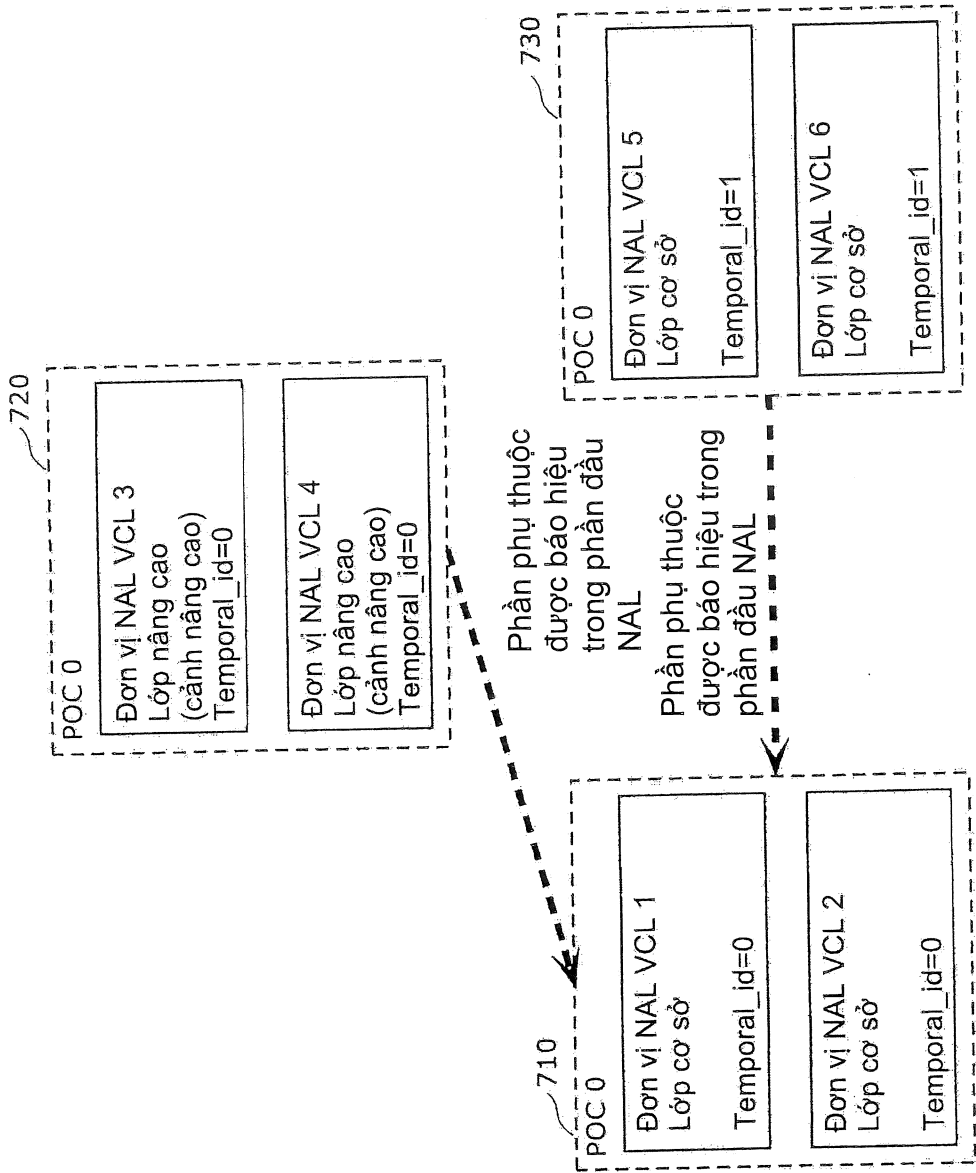
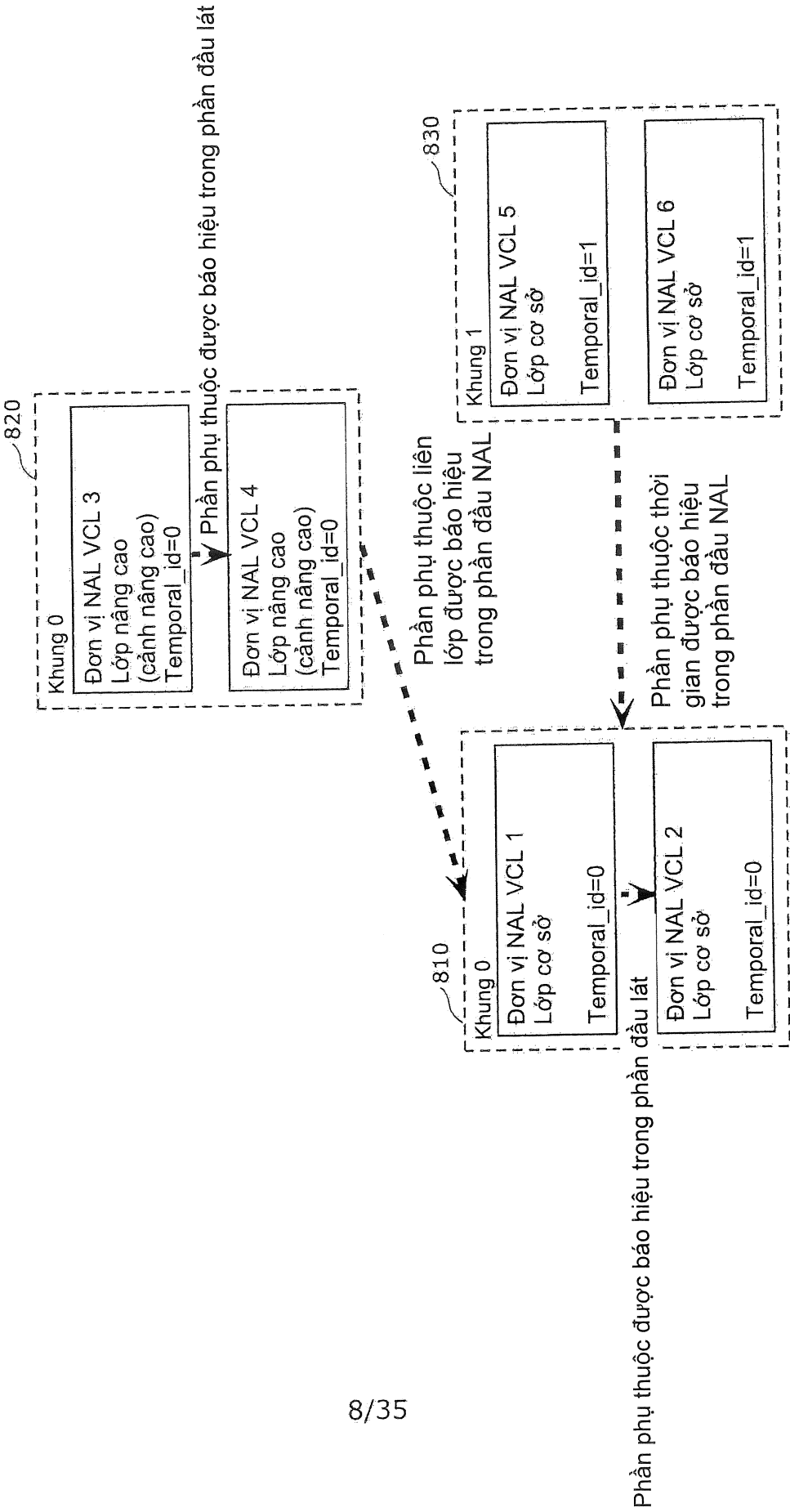


FIG. 8



910

	Phần mô tả
<code>nal_unit_header() {</code>	
<code> forbidden_zero_bit</code>	<code>f(1)</code>
<code> nal_unit_type</code>	<code>u(6)</code>
<code> nuh_reserved_zero_6bits</code>	<code>u(6)</code>
<code> nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>}</code>	

Các phần phụ thuộc lớp

Các phần phụ thuộc thời gian

920

	Phần mô tả
<code>slice_header() {</code>	
<code> first_slice_in_pic_flag</code>	<code>u(1)</code>
<code> if(RapPicFlag)</code>	
<code> no_output_of_prior_pics_flag</code>	<code>u(1)</code>
<code> pic_parameter_set_id</code>	<code>ue(v)</code>
<code> if(!first_slice_in_pic_flag)</code>	
<code> slice_address</code>	<code>u(v)</code>
<code> if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)</code>	
<code> dependent_slice_flag</code>	<code>u(1)</code>
<code> if(!dependent_slice_flag) {</code>	
<code> slice_type</code>	<code>ue(v)</code>
<code> </code>	<code>...</code>

Lát phụ thuộc/entropy bất hoặc tắt

FIG. 9B

seq_parameter_set_rbsp() {	Phân mô tả
video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_bit	u(1)
profile_and_level(1, sps_max_sub_layers_minus1)	
seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples ~931	ue(v)
pic_height_in_luma_samples ~932	ue(v)
pic_cropping_flag	u(1)
if(pic_cropping_flag) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
for(i = 0; i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pie_buffering[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase[i]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
if(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3 ~933	ue(v)
log2_diff_max_min_coding_block_size ~934	ue(v)
...	
...	
...	
}	

930

935

920

slice_header() {
first_slice_in_pic_flag
if(RapPicFlag)
no_output_of_prior_pics_flag
pic_parameter_set_id
if(!first_slice_in_pic_flag)
slice_address
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)
dependent_slice_flag
if(!dependent_slice_flag) {
slice_type
... ..

Để phân tích cú pháp slice_address, cần các phần tử cú pháp từ SPS. Do đó, SPS cần được phân tích cú pháp đến điểm này.

FIG. 9C

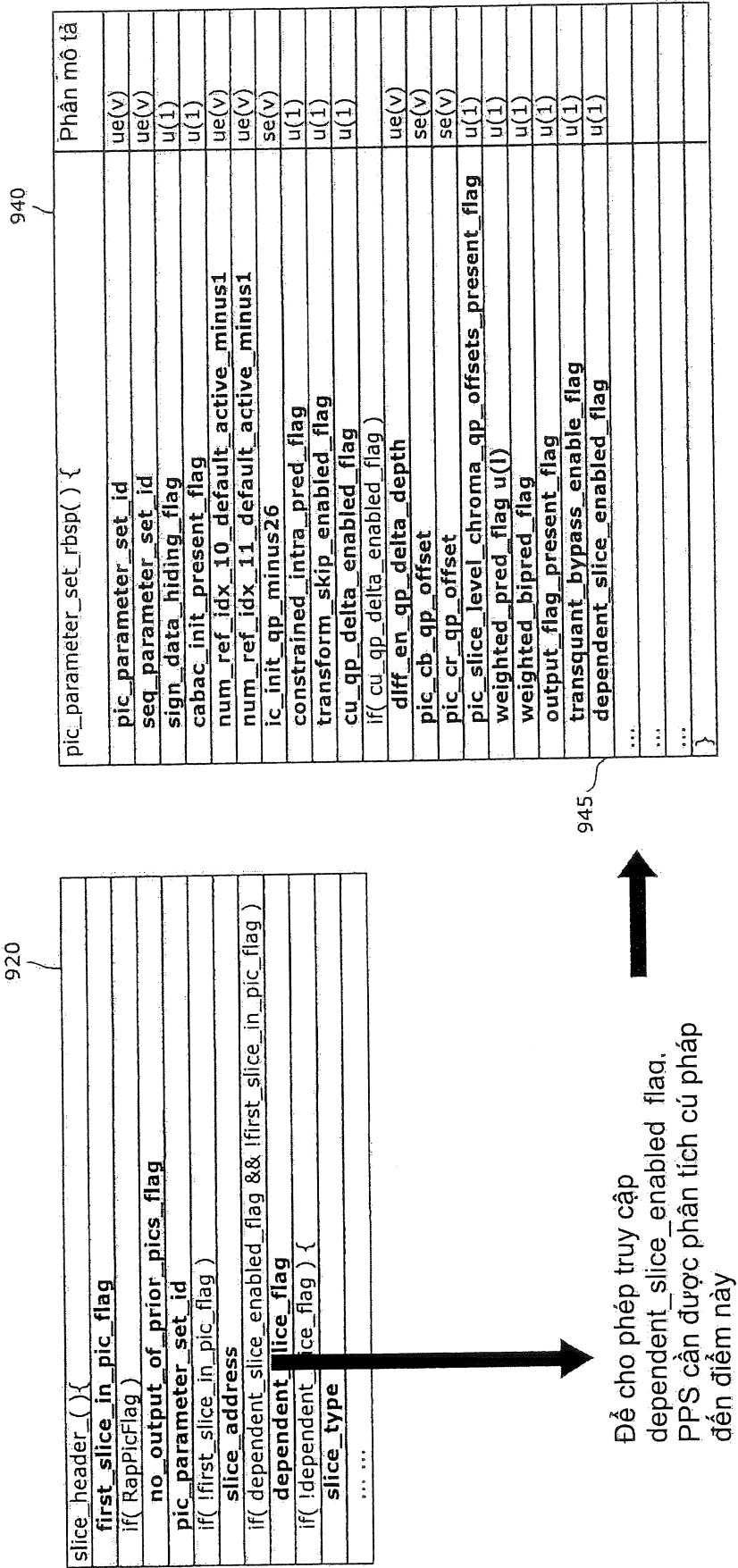


FIG. 10

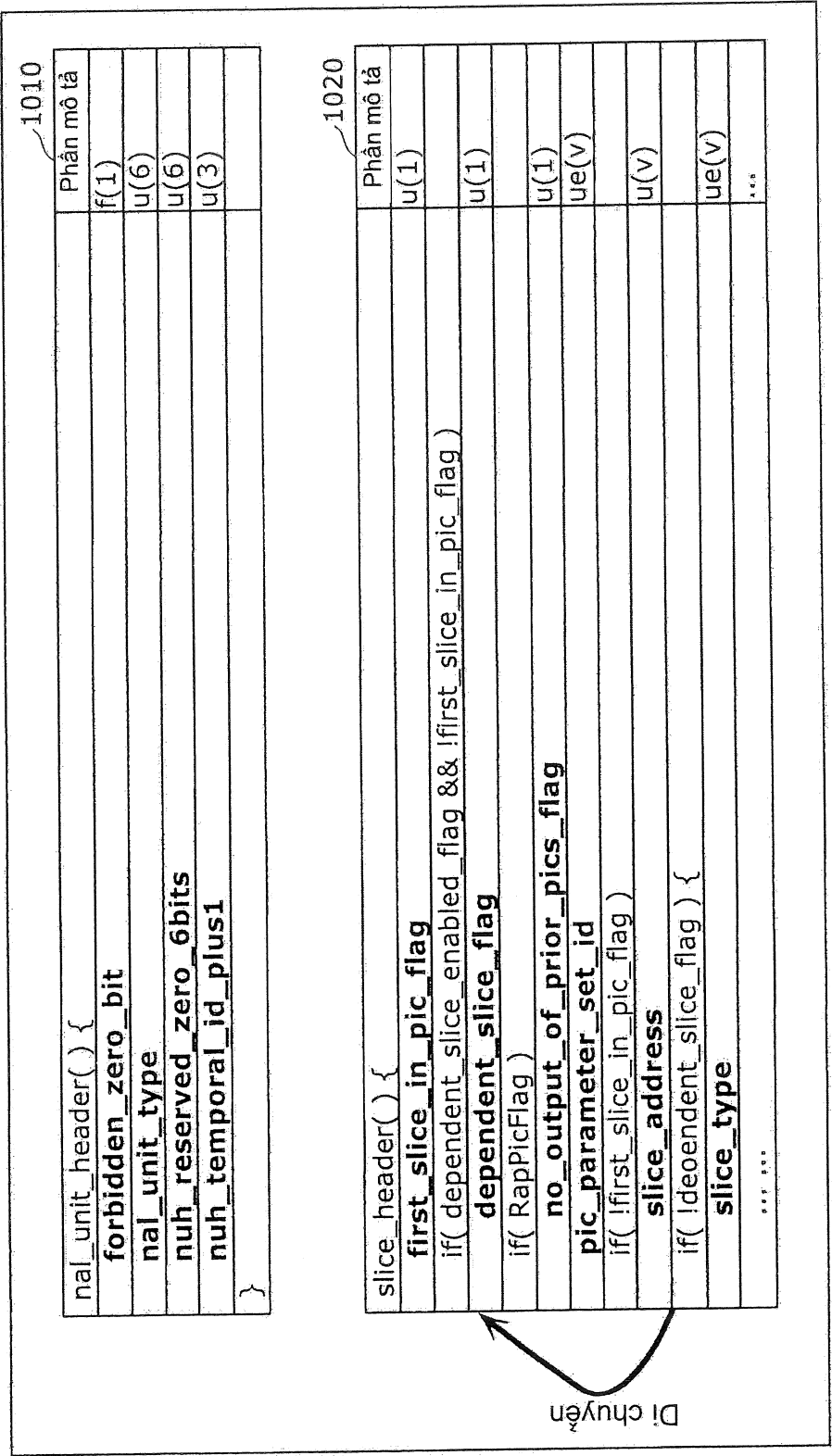


FIG. 11

1110

slice_header() {	Phân mô tả
first_slice_in_pic_flag	u(1)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(RapPicFlag)	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(!dependent_slice_flag) {	
slice_type	ue(v)
... ..	...

FIG. 12

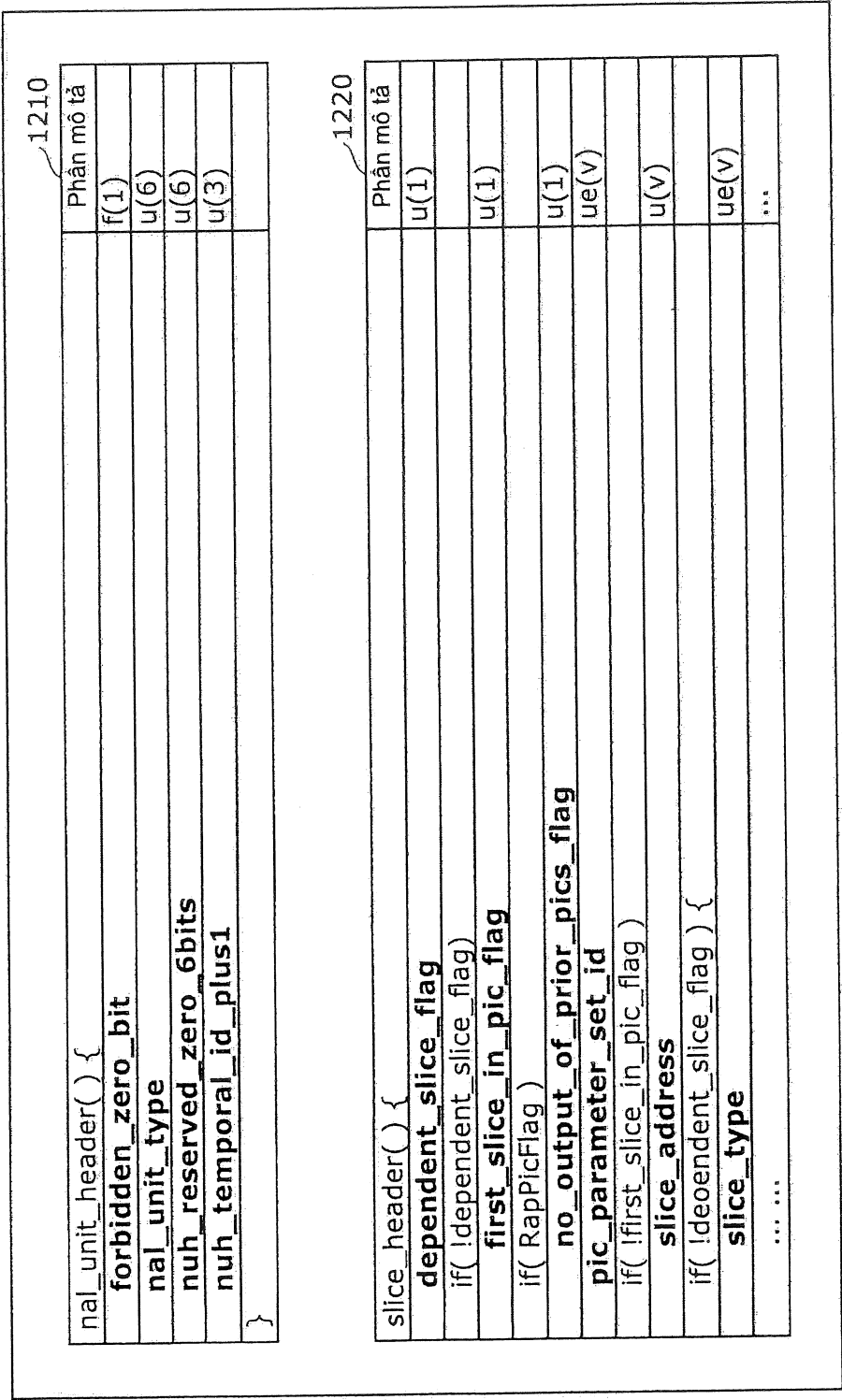


FIG. 13

1310

slice_header () {	Phần mô tả
first_slice_in_pic_flag	u(1)
if(RapPicFlag)	
no_output_of_prior_pics_flag	u(1)
pic_parameter_set_id	ue(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(!dependent_slice_flag) {	
slice_type	ue(v)
... ..	...

FIG. 14

nal_unit_header() {	1410
forbidden_zero_bit	Phân mô tả f(1)
nal_unit_type	u(6)
nuh_reserved_zero_6bits	u(5)
nuh_temporal_id_plus1	u(3)
dependent_slice_flag	u(1)
}	

FIG. 15

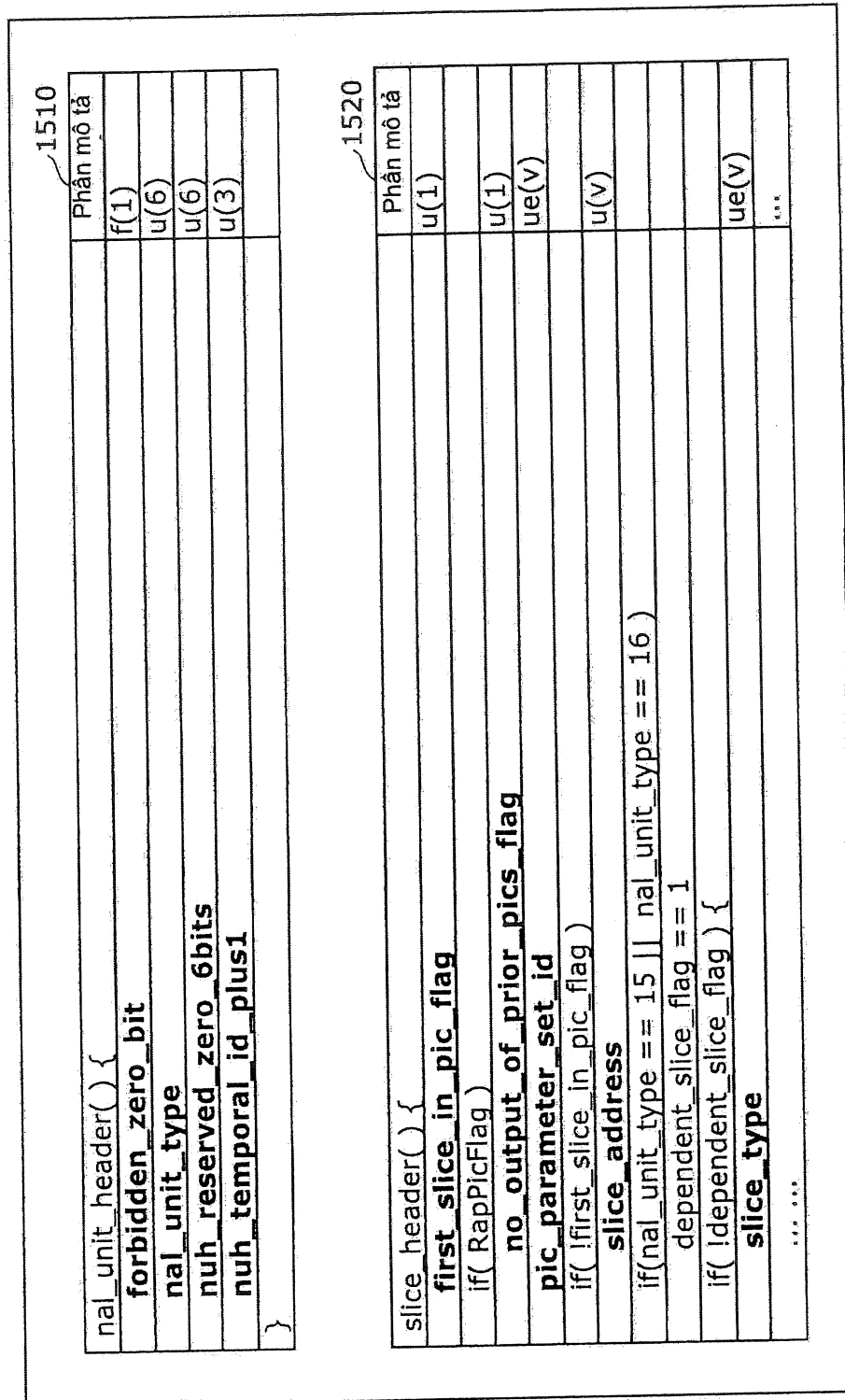


FIG. 16

1610	
<code>nal_unit_header() {</code>	Phân mô tả
<code>forbidden_zero_bit</code>	<code>f(1)</code>
<code>nal_unit_type</code>	<code>u(6)</code>
<code>nuh_reserved_zero_6bits</code>	<code>u(6)</code>
<code>nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>}</code>	
1620	
<code>slice_header() {</code>	Phân mô tả
<code>first_slice_in_pic_flag</code>	<code>u(1)</code>
<code>if(RapPicFlag)</code>	
<code>no_output_of_prior_pics_flag</code>	<code>u(1)</code>
<code>pic_parameter_set_id</code>	<code>ue(v)</code>
<code>if(!first_slice_in_pic_flag)</code>	
<code>slice_address</code>	<code>u(v)</code>
<code>if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)</code>	
<code>dependent_slice_flag</code>	<code>u(1)</code>
<code>if(!dependent_slice_flag) {</code>	
<code>slice_type</code>	<code>ue(v)</code>
<code>... ;</code>	<code>...</code>

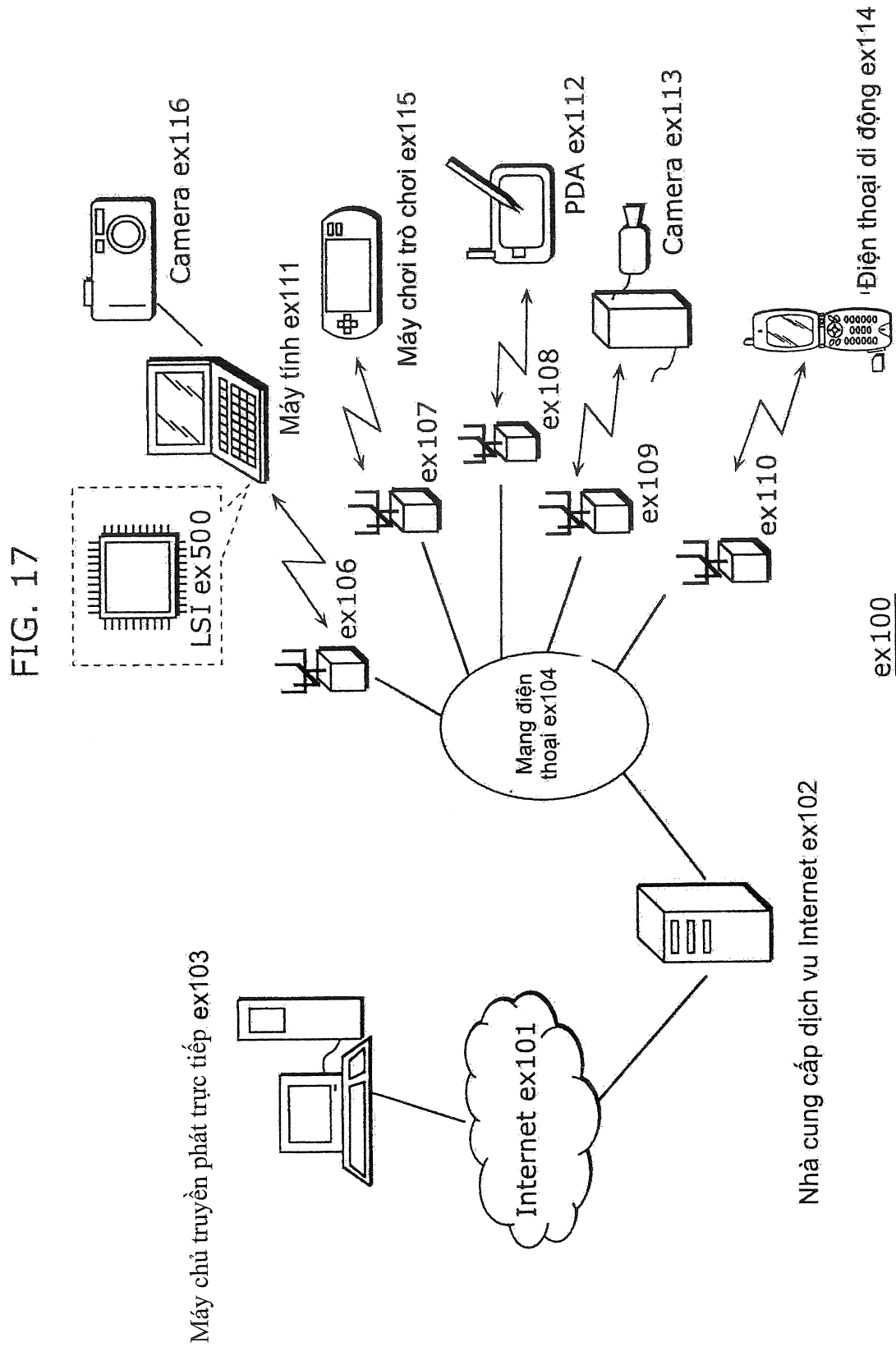


FIG. 18

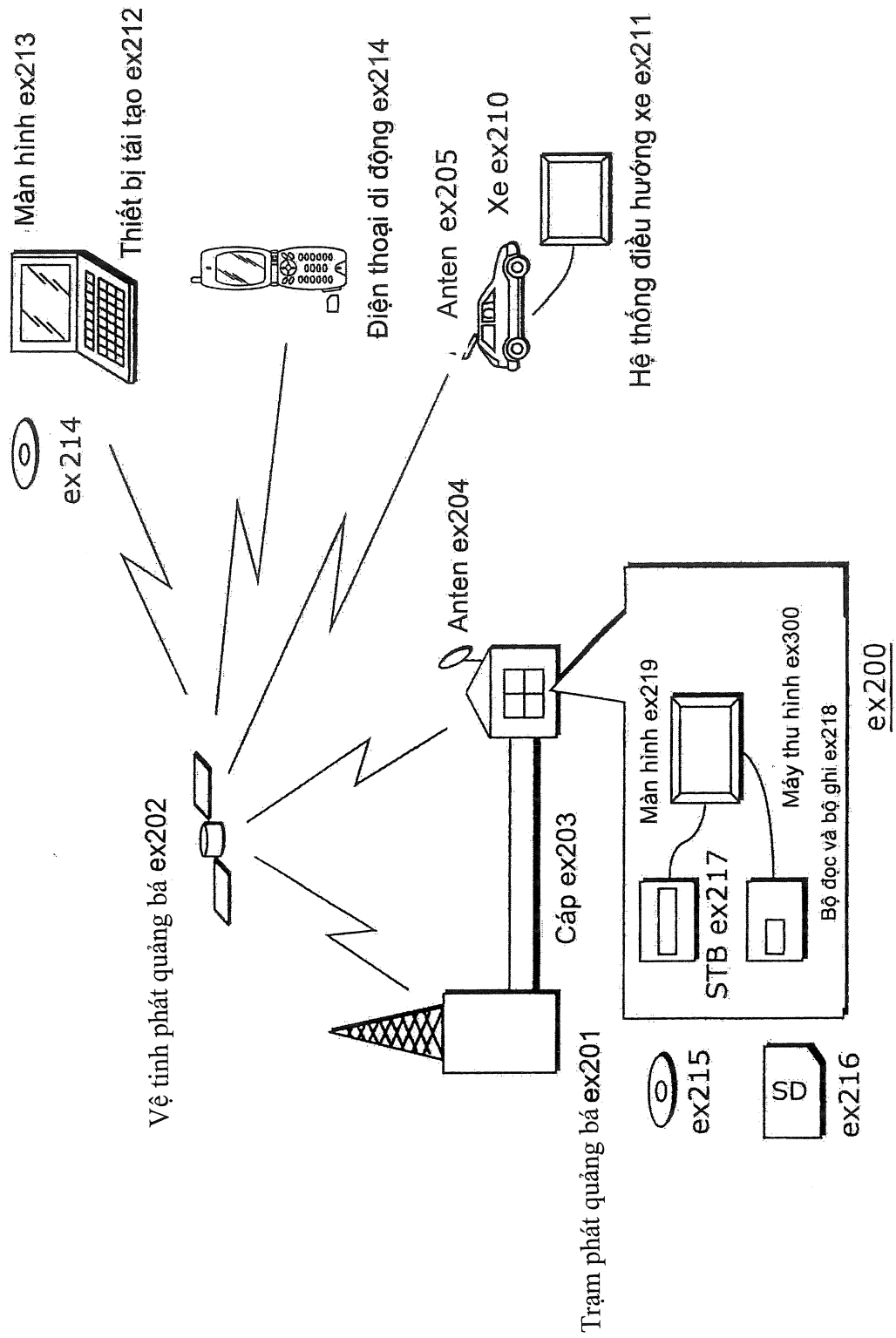


FIG. 19

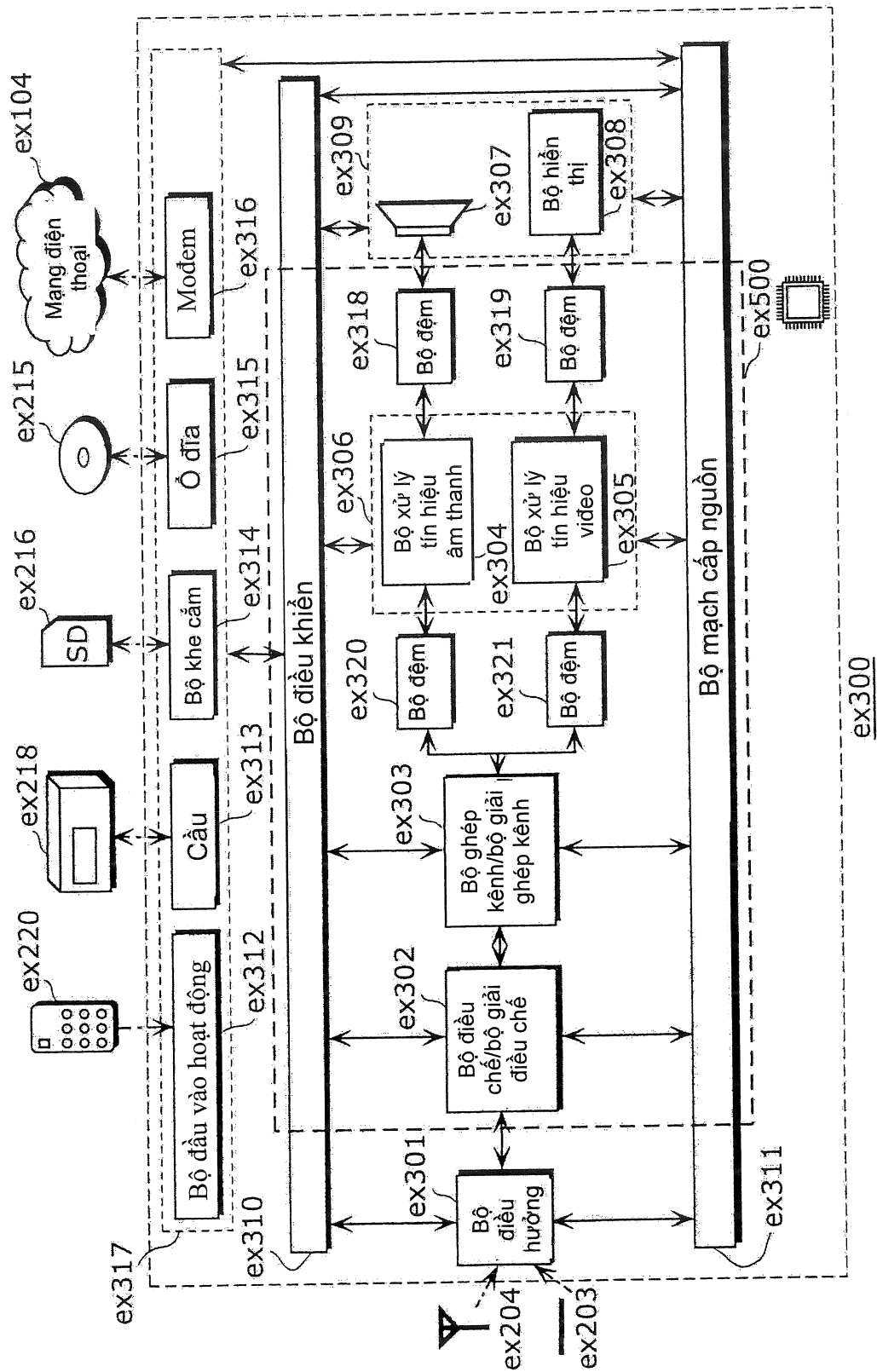


FIG. 20

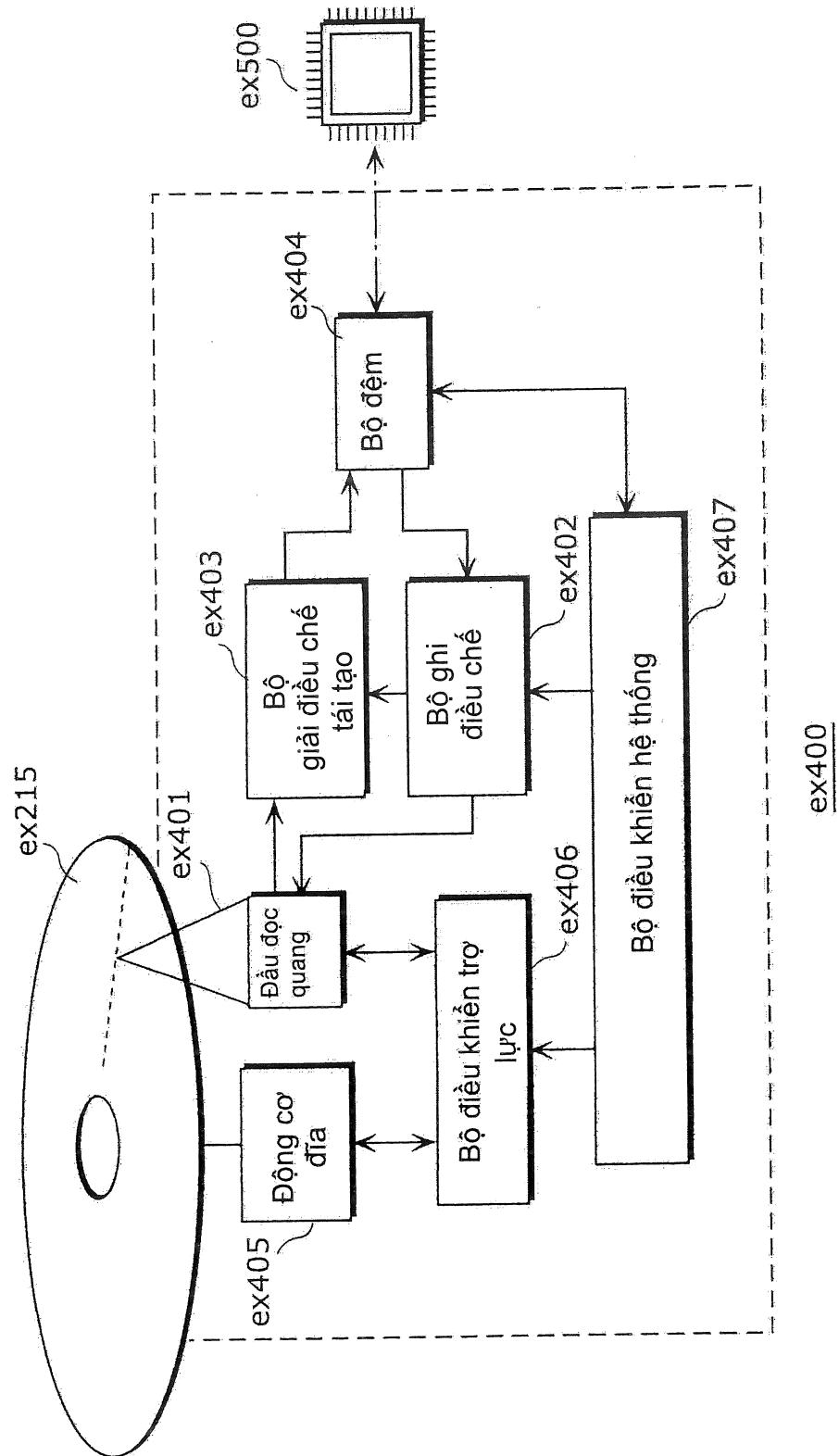


FIG. 21

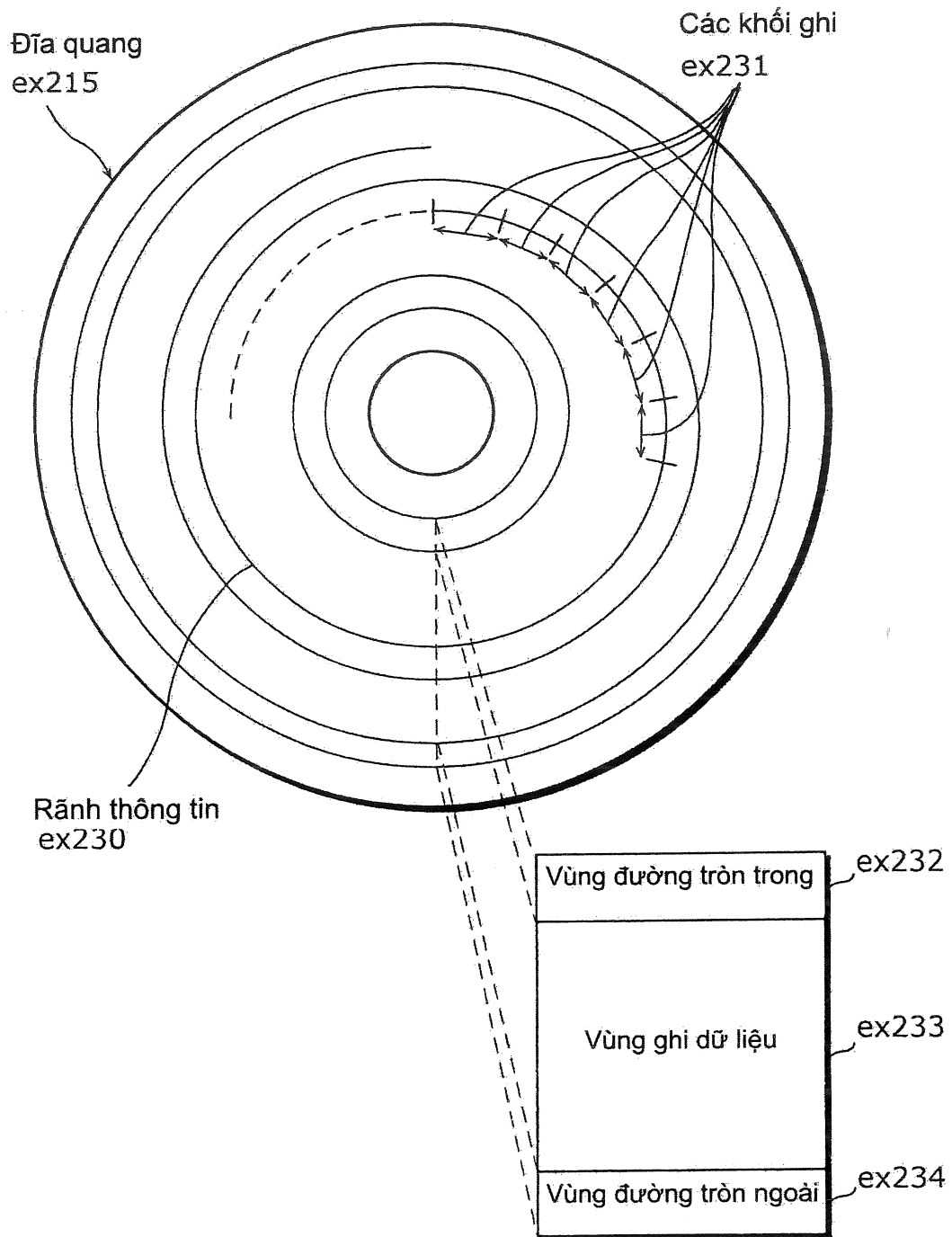


FIG. 22A

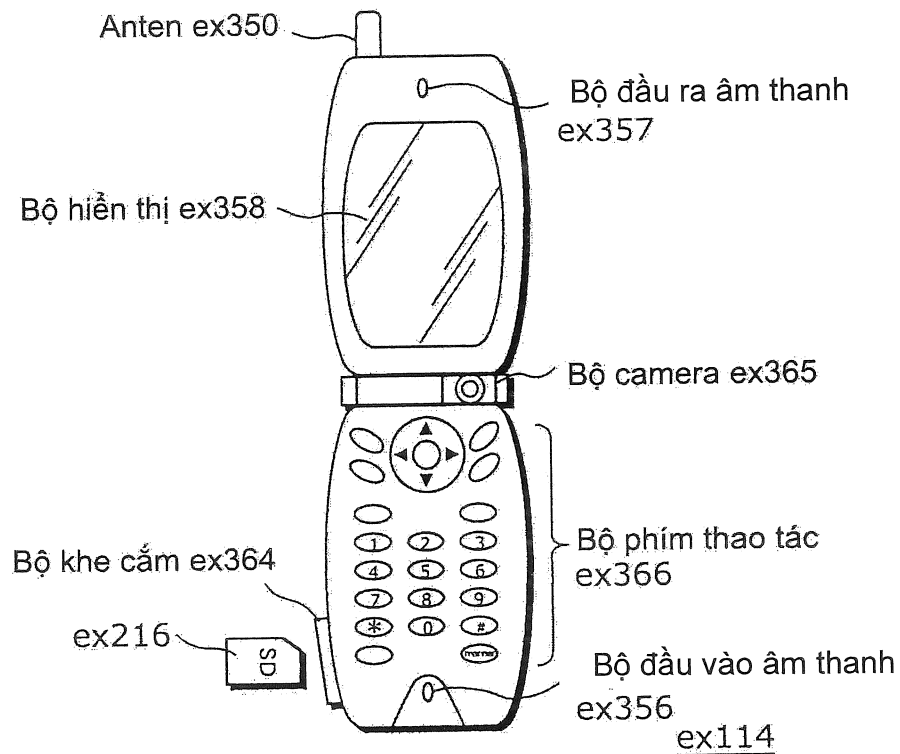


FIG. 22B

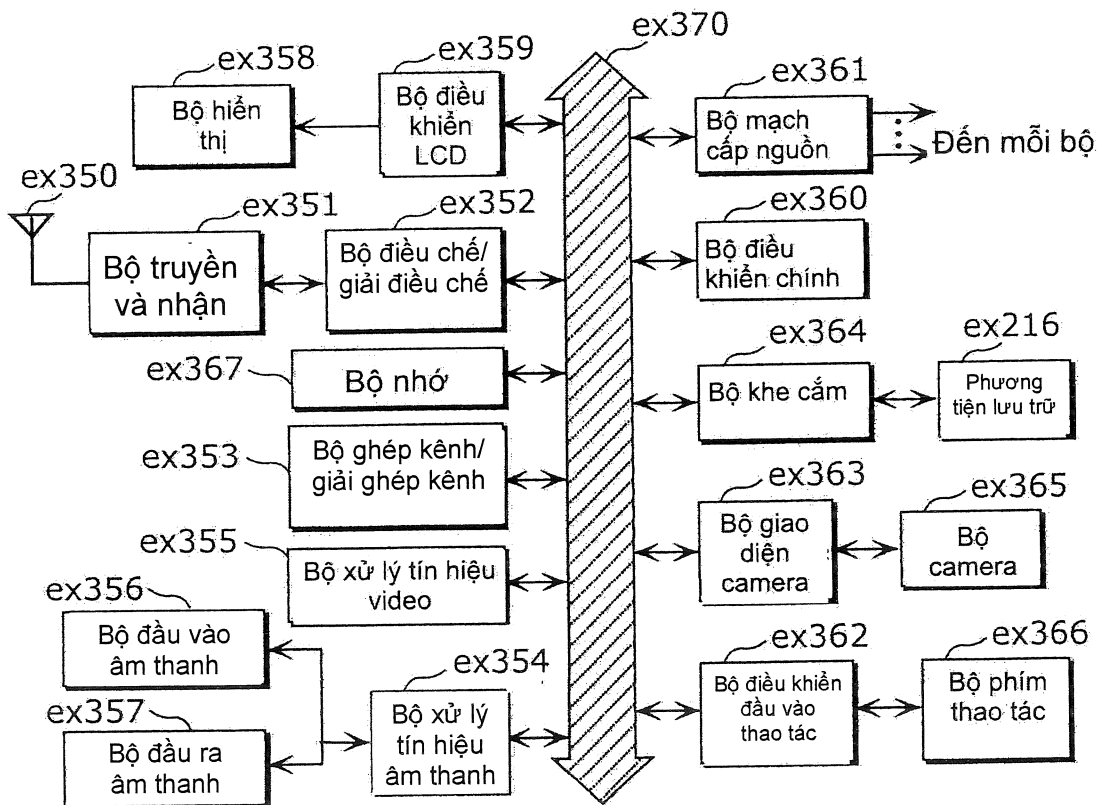


FIG. 23

Dòng video (PID=0x1011, video sơ cấp)
Dòng âm thanh (PID=0x1100)
Dòng âm thanh (PID=0x1101)
Dòng đồ họa trình diễn (PID=0x1200)
Dòng đồ họa trình diễn (PID=0x1201)
Dòng đồ họa tương tác (PID=0x1400)
Dòng video (PID=0x1B00, video thứ cấp)
Dòng video (PID=0x1B01, video thứ cấp)

FIG. 24

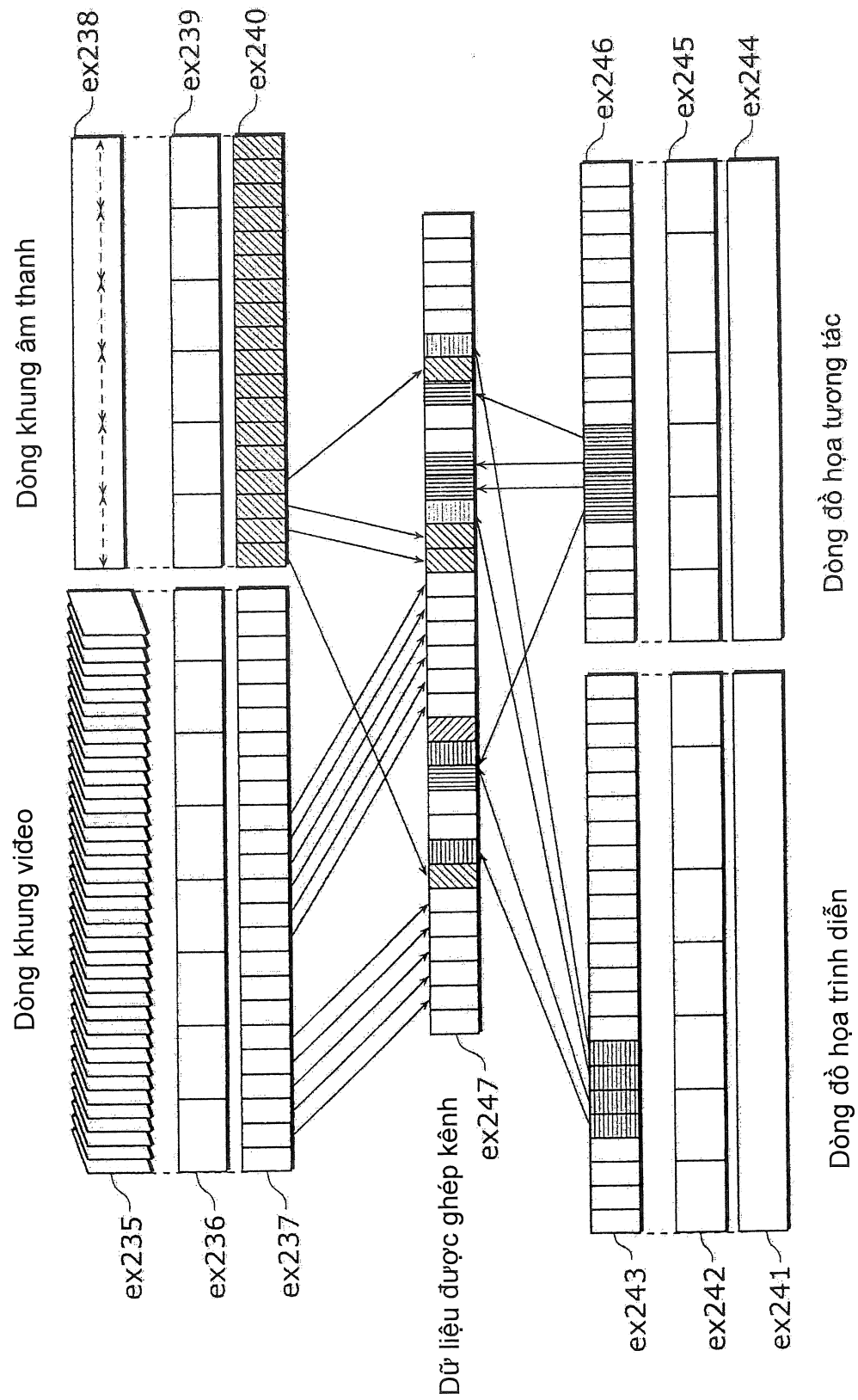


FIG. 25

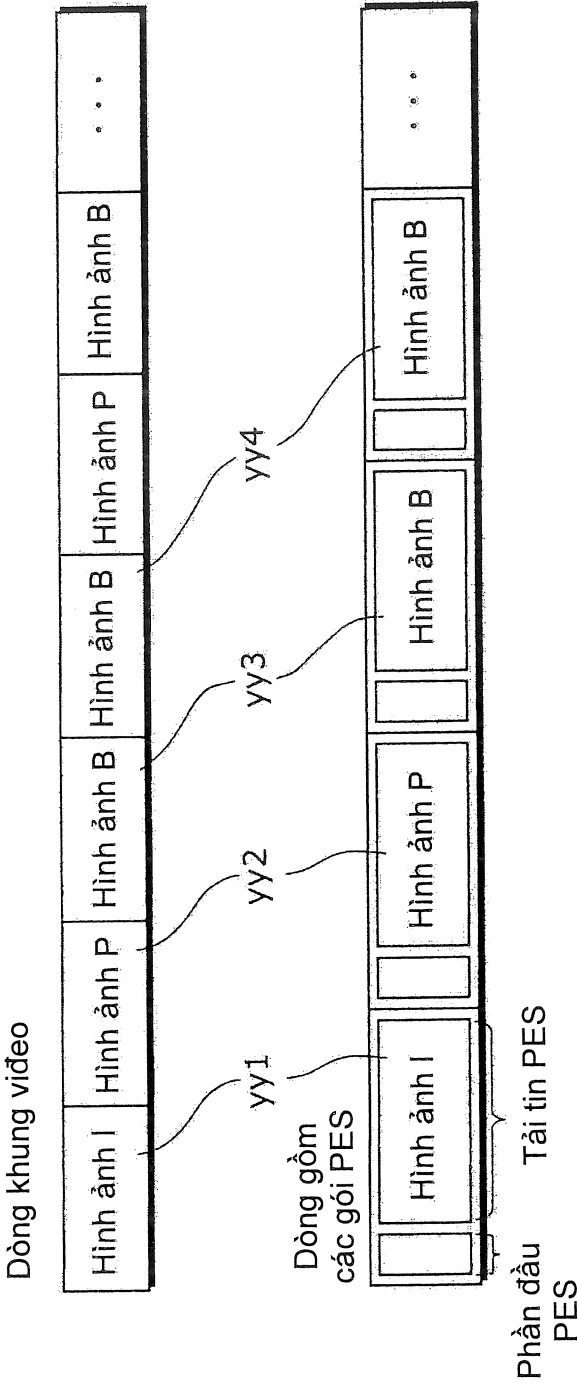


FIG. 26

Dòng gồm các gói TS

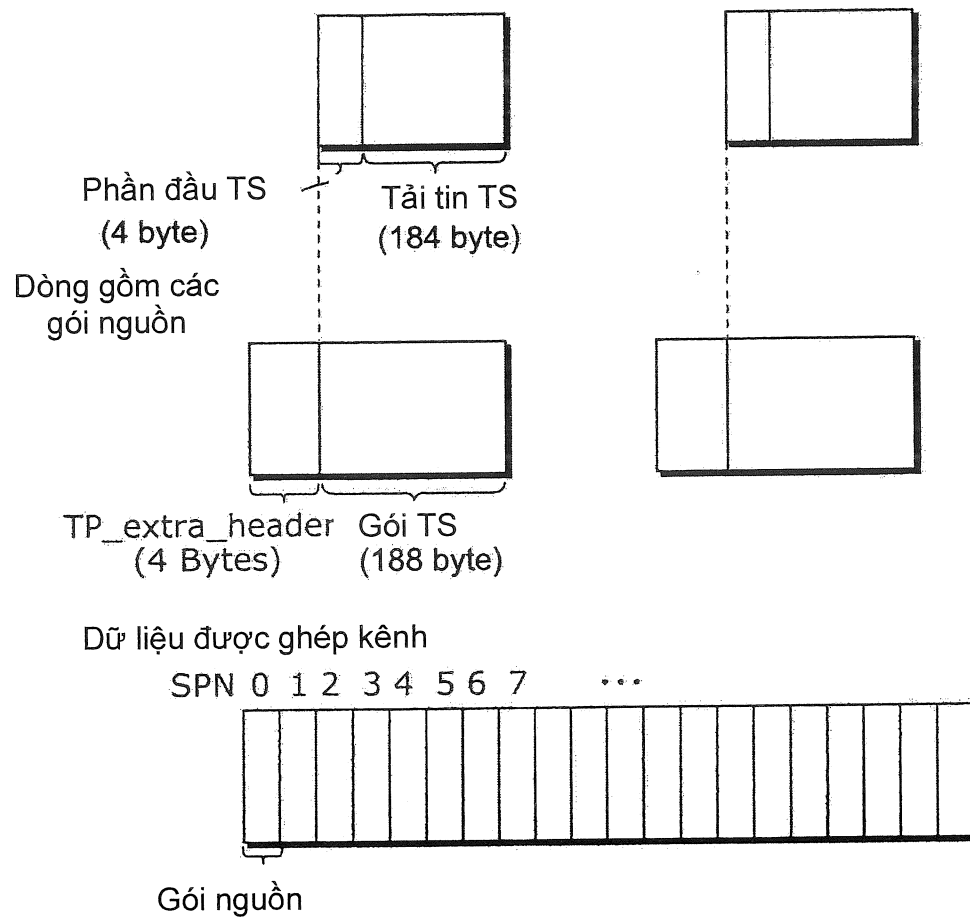


FIG. 27

Cấu trúc dữ liệu của PMT

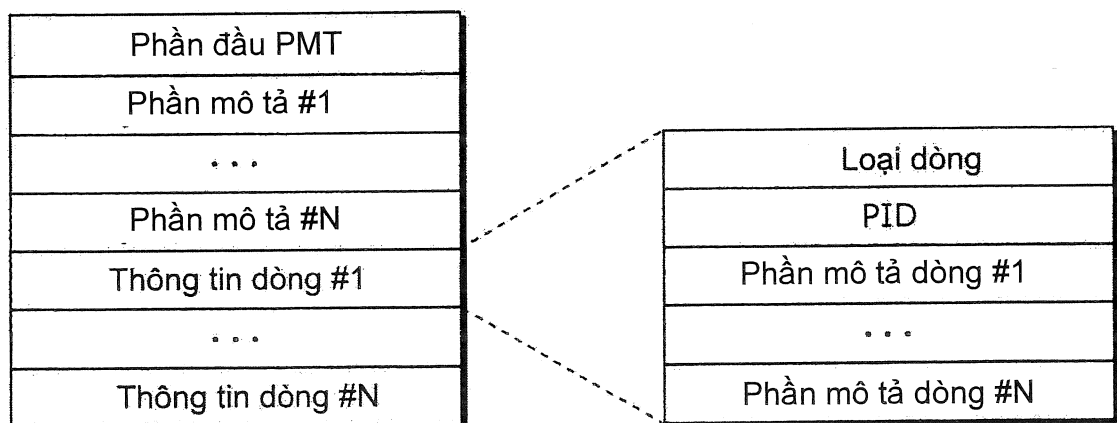


FIG. 28

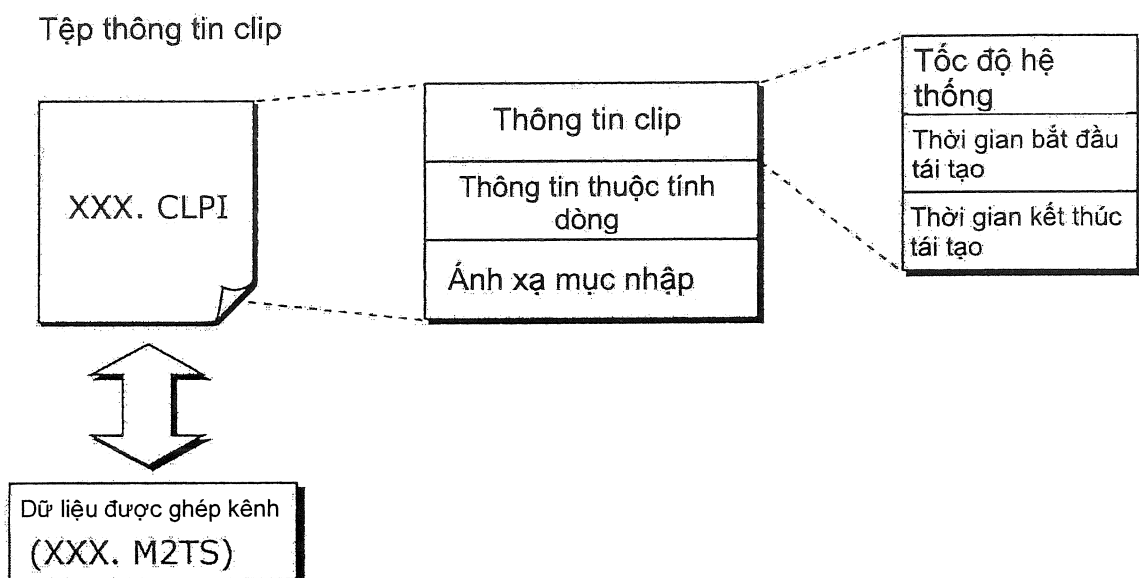


FIG. 29

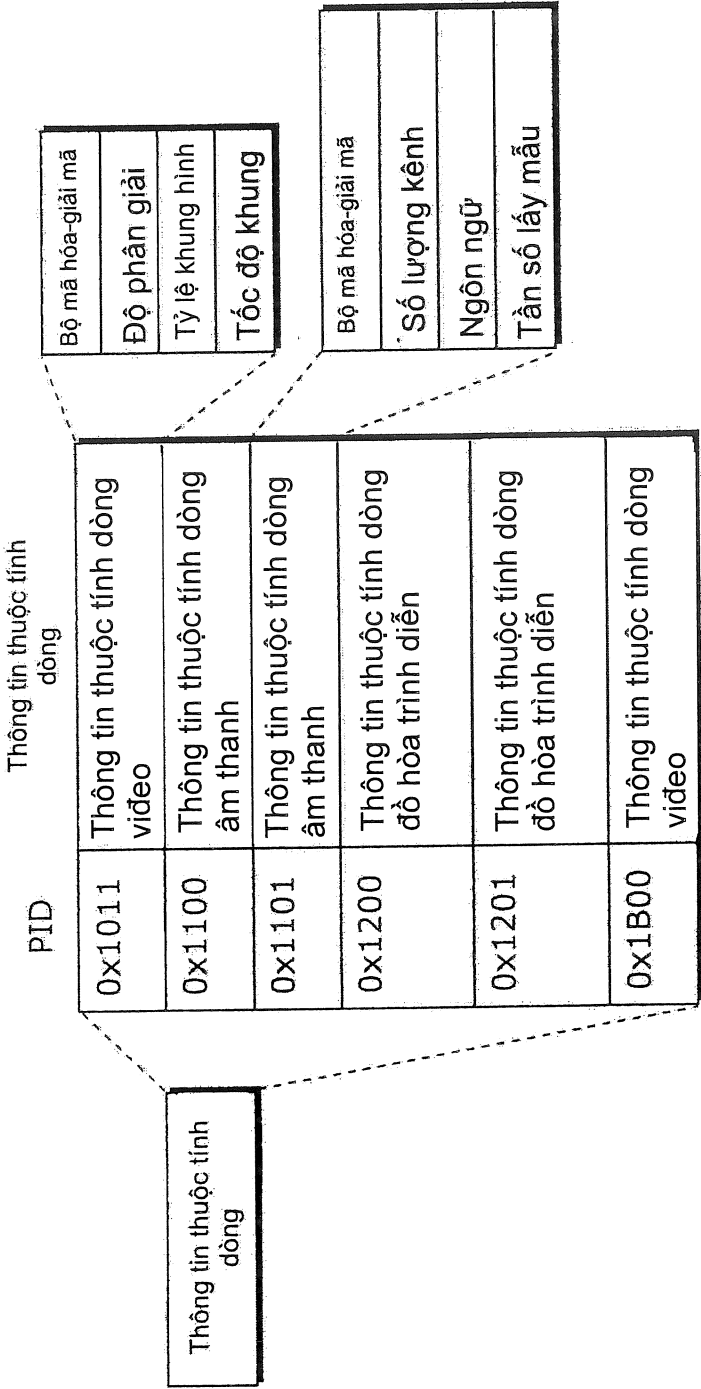


FIG. 30

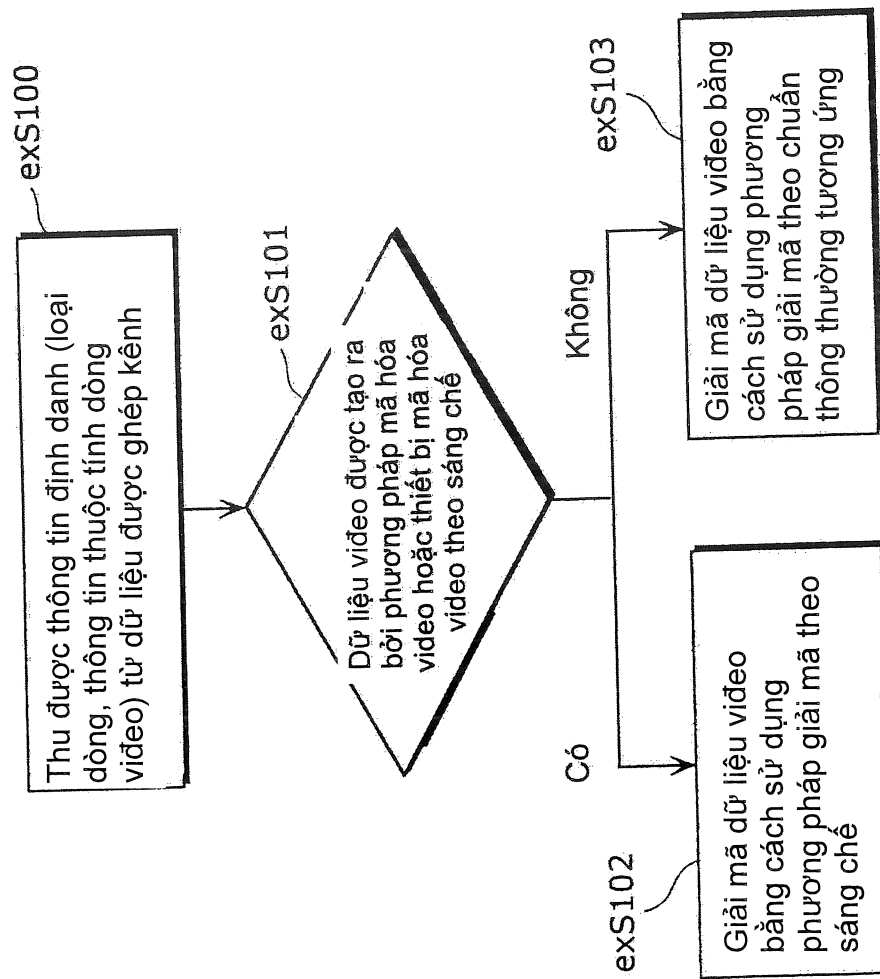


FIG. 31

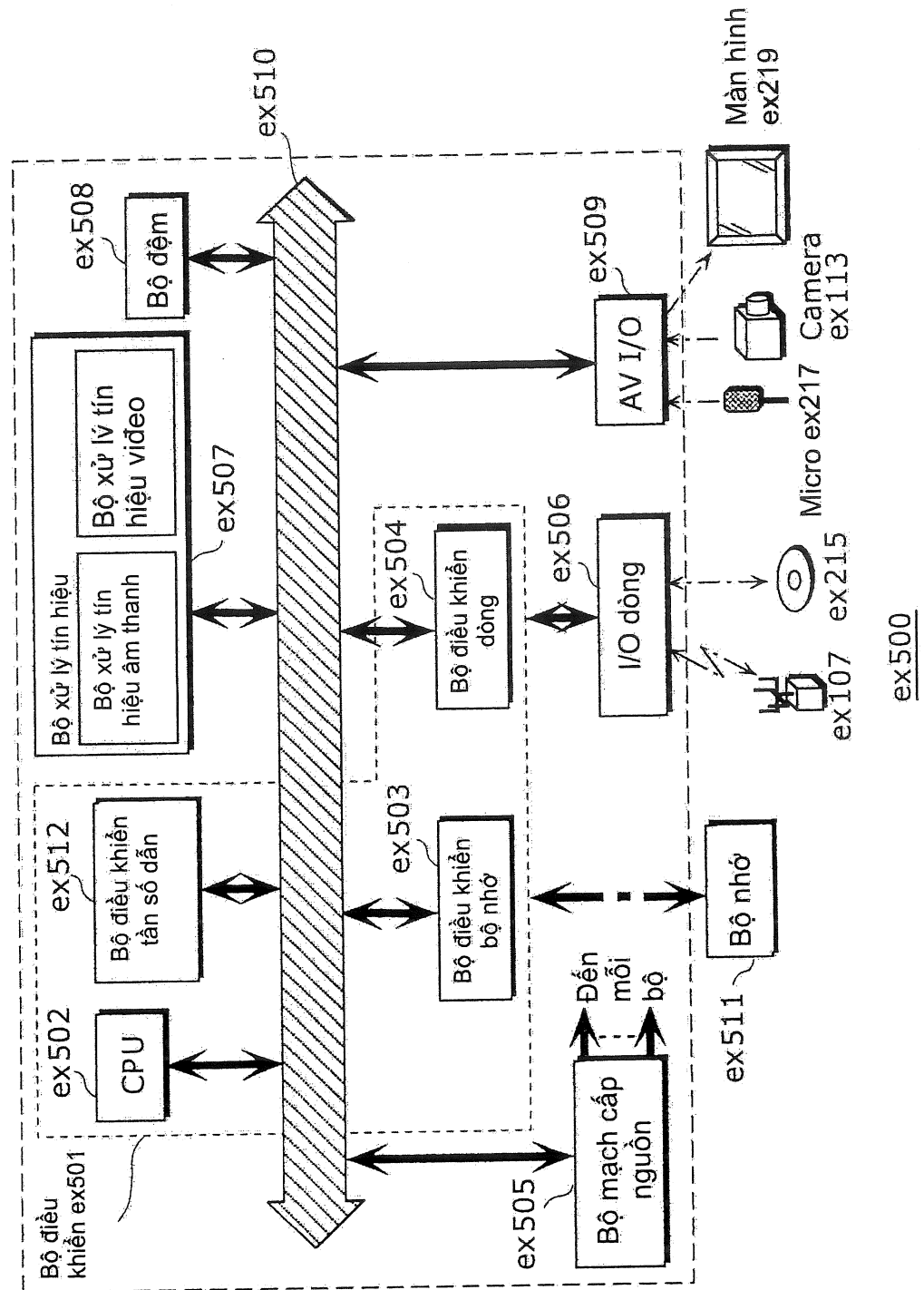


FIG. 32

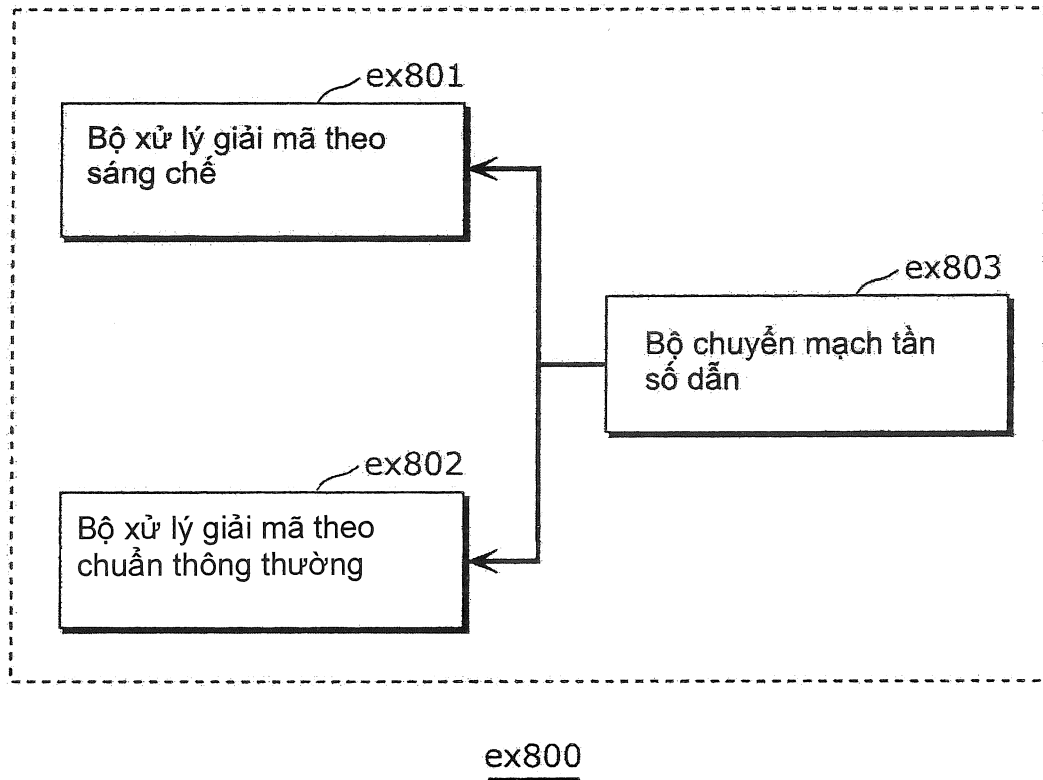


FIG. 33

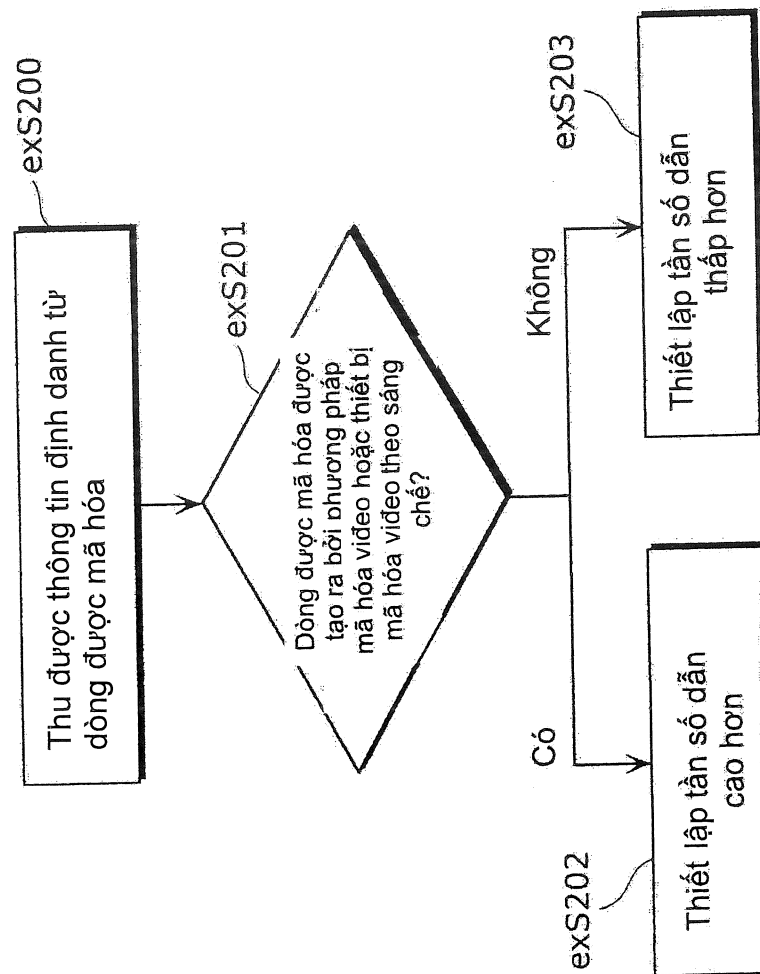


FIG. 34

Chuẩn tương ứng	Tần số dẫn
MPEG-4 AVC	500 MHz
MPEG-2	350 MHz
⋮	⋮

FIG. 35A

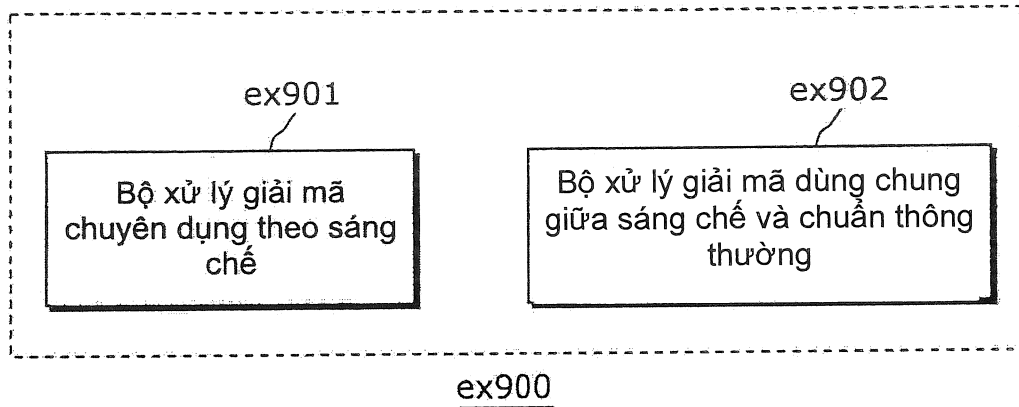


FIG. 35B

