



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0025621

(51)⁷ H04N 19/102

(13) B

(21) 1-2015-01071

(22) 30/09/2013

(86) PCT/JP2013/076486 30/09/2013

(87) WO 2014/057832 A1 17/04/2014

(30) 2012-224597 09/10/2012 JP; 2012-225159 10/10/2012 JP; 2012-262803 30/11/2012 JP; 2013-004988 15/01/2013 JP; 2013-010585 23/01/2013 JP; 2013-060438 22/03/2013 JP

(45) 25/09/2020 390

(43) 27/07/2015 328A

(73) SONY CORPORATION (JP)

1-7-1 Konan, Minato-ku Tokyo, 108-0075, Japan

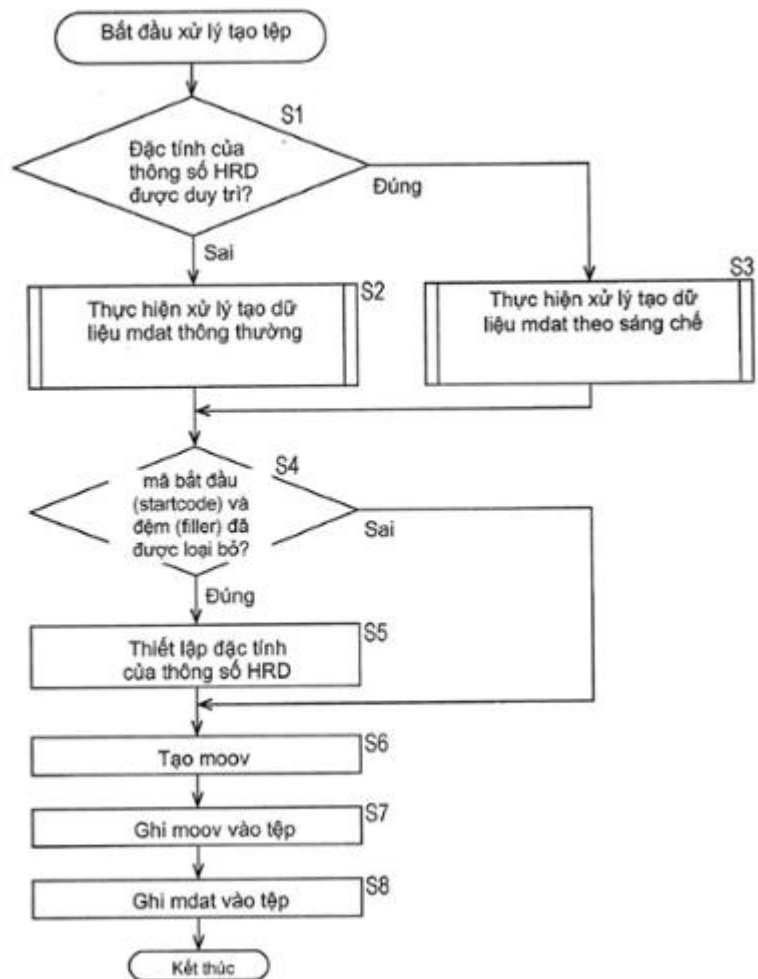
(72) SUZUKI Teruhiko (JP); HIRABAYASHI Mitsuhiro (JP); NAKANO Takehiko (JP).

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

(54) THIẾT BỊ VÀ PHƯƠNG PHÁP XỬ LÝ ẢNH

(57) Sáng chế đề cập đến thiết bị xử lý ảnh và phương pháp xử lý ảnh có thể giảm gánh nặng xử lý được yêu cầu khi tạo luồng.

Bộ tạo tệp thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và thực hiện điều khiển việc thiết lập mã bắt đầu và dữ liệu đệm để mã bắt đầu và dữ liệu đệm mà đã được thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp. Sáng chế, ví dụ, có thể được áp dụng cho thiết bị xử lý ảnh.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến thiết bị xử lý ảnh và phương pháp xử lý ảnh, và cụ thể hơn, thiết bị xử lý ảnh và phương pháp xử lý ảnh có hiệu quả giảm gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng tại thời gian lưu giữ dữ liệu trong tệp được tạo ra.

Tình trạng kỹ thuật của sáng chế

Trong H.264/AVC, đây là một trong số các đặc tả tiêu chuẩn của hệ thống mã hóa ảnh, hai loại tập thông số được gọi là tập thông số trình tự (sequence parameter set, viết tắt là SPS) và tập thông số hình ảnh (picture parameter set, viết tắt là PPS) được sử dụng để lưu giữ các thông số để mã hóa và giải mã ảnh được định nghĩa. Dữ liệu ảnh của từng lát được phân loại thành các đơn vị lớp mã hóa video (Video Coding Layer, viết tắt là VCL) lớp mạng trừu tượng (Network Abstraction Layer, viết tắt là NAL), và các tập thông số này được phân loại thành các đơn vị không phải là VCL NAL. Nói chung, trong trường hợp mà luồng đã được mã hóa bằng cách sử dụng hệ thống H.264/AVC được lưu giữ trong tệp bao gồm vùng đoạn đầu và vùng dữ liệu, SPS và PPS được chèn vào vùng đoạn đầu, và dữ liệu ảnh được chèn vào vùng dữ liệu.

Trong hoạt động chuẩn hóa mã hóa video hiệu suất cao (High Efficiency Video Coding, viết tắt là HEVC), đây là hệ thống mã hóa ảnh thế hệ tiếp theo sau H.264/AVC, việc giới thiệu tập thông số thích ứng (adaptation parameter set, viết tắt là APS), đây là tập thông số mới không phải là SPS và PPS, đã được đề xuất (xem các tài liệu phi sáng chế 1 và 2 được mô tả dưới đây).

APS cũng là một loại tập thông số. Do vậy, theo kỹ thuật hiện thời, tương tự như SPS và PPS, APS cũng được chèn vào vùng đoạn đầu của tệp. Các ví dụ về định dạng tệp bao gồm vùng đoạn đầu và vùng dữ liệu bao gồm định dạng MPEG-4 phần 14 (ISO/IEC 14496-14: 2003, dưới đây được gọi là MP4) và định dạng MPEG-4 phần 15 (ISO/IEC 14496-15: 2004, tệp AVC).

Tài liệu trích dẫn

Tài liệu phi sáng chế

Tài liệu phi sáng chế 1: Stephan Wenger, Jill Boyce, Yu-Wen Huang, Chia-Yang Tsai, Ping Wu, Ming Li, “Tập thông số thích ứng” (“Adaptation Parameter Set (APS)”), JCTVC-F747r3, tháng 7 năm 2011

Tài liệu phi sáng chế 2: Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, Thomas Wiegand, “Bản thảo đặc tả mã hóa video hiệu suất cao (HEVC) 6” (“High efficiency video coding (HEVC) text specification draft 6”), JCTVC-H1003 ver20, tháng 2 năm 2012

Bản chất kỹ thuật của sáng chế

Vấn đề kỹ thuật cần được sáng chế giải quyết

Tuy nhiên, trong trường hợp mà luồng được sử dụng để truyền hoặc luồng được sử dụng để lưu giữ dữ liệu trong tệp được tạo ra, có định dạng trong đó dữ liệu như mã bắt đầu (startcode) và dữ liệu đệm (filler data) không được phép có trong luồng. Do vậy, khi dữ liệu của luồng được đọc, nó sẽ khiến tải trọng xử lý tăng.

Sáng chế được hoàn thành trong bối cảnh như vậy và có hiệu quả giảm được gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng tại thời gian lưu giữ dữ liệu trong tệp được tạo ra.

Cách thức giải quyết vấn đề

Theo khía cạnh thứ nhất, sáng chế đề xuất thiết bị xử lý ảnh bao gồm: bộ thiết lập để thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và bộ điều khiển để thực hiện điều khiển bộ thiết lập để mã bắt đầu và dữ liệu đệm được thiết lập bởi bộ thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

Thông số quản lý bộ đệm giải mã là thông số có trong thông tin sử dụng được video (Video Usability Information, viết tắt là VUI), thông số có trong khoảng đệm thông tin cải tiến bổ sung (Supplemental Enhancement Information, viết tắt là SEI), hoặc thông số có trong SEI thời gian hình ảnh.

Bộ thiết lập có thể thiết lập dữ liệu đệm làm dữ liệu VCL.

Theo khía cạnh thứ nhất, sáng chế đề xuất phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh. Phương pháp xử lý ảnh bao gồm các bước: thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và thực hiện điều khiển việc thiết lập mã bắt đầu và dữ liệu đệm để mã bắt đầu và dữ liệu đệm mà được thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

Theo khía cạnh thứ hai, sáng chế đề xuất thiết bị xử lý ảnh bao gồm: bộ thu để thu tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và được điều khiển và thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp; và bộ giải mã để đọc mã bắt đầu và dữ liệu đệm từ tệp thu được bởi bộ thu và giải mã luồng bit bằng cách sử dụng thông số quản lý bộ đệm giải mã.

Thông số quản lý bộ đệm giải mã là thông số có trong VUI (thông tin sử dụng được video), thông số có trong khoảng đệm SEI (thông tin cải tiến bổ sung), hoặc thông số có trong SEI thời gian hình ảnh.

Dữ liệu đệm được thiết lập làm dữ liệu VCL.

Theo khía cạnh thứ hai, sáng chế đề xuất phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh. Phương pháp xử lý ảnh bao gồm các bước: thu tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và được điều khiển và thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp; và đọc mã bắt đầu và dữ liệu đệm từ tệp đã thu được và giải mã luồng bit bằng cách sử dụng thông số quản lý bộ đệm giải mã.

Theo khía cạnh thứ ba, sáng chế đề xuất thiết bị xử lý ảnh bao gồm: bộ thiết lập để thiết lập đặc tính của thông số quản lý bộ đệm giải mã trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit trong dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và bộ tạo để tạo tệp bằng cách sử dụng đặc tính được thiết lập bởi bộ thiết lập.

Bộ thiết lập có thể thiết lập thông số nhận dạng nhận dạng rằng đặc tính của thông số quản lý bộ đệm giải mã được thay đổi.

Bộ thiết lập có thể thiết lập thông số nhận dạng làm hộp tùy chọn trong mục nhập mẫu của tệp.

Theo khía cạnh thứ ba, sáng chế đề xuất phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh. Phương pháp xử lý ảnh bao gồm các bước: thiết lập đặc tính của thông số quản lý bộ đệm giải mã trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit trong dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và tạo tệp bằng cách sử dụng đặc tính đã được thiết lập.

Theo khía cạnh thứ tư, sáng chế đề xuất thiết bị xử lý ảnh bao gồm: bộ thu để thu tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và bộ giải mã để đọc thông số quản lý bộ đệm giải mã từ tệp thu được bởi bộ thu và giải mã luồng bit bằng cách sử dụng thông số đã được đọc.

Trong tệp, thông số nhận dạng nhận dạng rằng đặc tính của thông số quản lý bộ đệm giải mã được thay đổi được thiết lập.

Theo khía cạnh thứ tư, sáng chế đề xuất phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh. Phương pháp xử lý ảnh bao gồm các bước: thu tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và đọc thông số quản lý bộ đệm giải mã từ tệp đã thu được và giải mã luồng bit bằng cách sử dụng thông số đã được đọc.

Theo khía cạnh thứ nhất của sáng chế, mã bắt đầu và dữ liệu đệm được thiết lập cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh. Sau đó, việc thiết lập mã bắt đầu và dữ liệu đệm được điều khiển để mã bắt đầu và dữ liệu đệm mà được thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ

đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

Theo khía cạnh thứ hai của sáng chế, tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và được điều khiển và thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp thu được. Sau đó, mã bắt đầu và dữ liệu đệm được đọc từ tệp đã thu được, và luồng bit được giải mã bằng cách sử dụng thông số quản lý bộ đệm giải mã.

Theo khía cạnh thứ ba của sáng chế, đặc tính của thông số quản lý bộ đệm giải mã được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit trong dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh. Sau đó, tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập.

Theo khía cạnh thứ tư của sáng chế, tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh thu được. Sau đó, thông số quản lý bộ đệm giải mã được đọc từ tệp đã thu được, và luồng bit được giải mã bằng cách sử dụng thông số đã được đọc.

Ở đây, thiết bị xử lý ảnh mô tả trên đây có thể là thiết bị độc lập hoặc khối bên trong tạo thành thiết bị mã hóa ảnh hoặc thiết bị giải mã ảnh.

Hiệu quả của sáng chế

Theo các khía cạnh thứ nhất và thứ ba của sáng chế, ảnh có thể được mã hóa. Cụ thể là, gánh nặng xử lý có thể được giảm khi luồng được sử dụng tại thời gian truyền của luồng bit hoặc luồng được sử dụng tại thời gian lưu giữ dữ liệu trong tệp được tạo ra.

Ngoài ra, theo các khía cạnh thứ hai và thứ tư của sáng chế, ảnh có thể được giải mã. Cụ thể là, gánh nặng xử lý có thể được giảm khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng tại thời gian lưu giữ dữ liệu trong tệp được giải mã.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị mã hóa, được áp dụng sáng chế, theo phương án thứ nhất.

Fig.2 là sơ đồ khối minh họa ví dụ về cấu hình của bộ mã hóa.

Fig.3 là sơ đồ khối minh họa ví dụ về cấu hình của bộ tạo tệp.

Fig.4 là lưu đồ minh họa xử lý tạo tệp.

Fig.5 là lưu đồ minh họa ví dụ khác về xử lý tạo tệp.

Fig.6 là lưu đồ minh họa xử lý thông thường để tạo dữ liệu mdat.

Fig.7 là lưu đồ minh họa xử lý tạo dữ liệu mdat theo sáng chế.

Fig.8 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị giải mã, được áp dụng sáng chế, theo phương án thứ nhất.

Fig.9 là sơ đồ khối minh họa ví dụ về cấu hình của bộ đọc tệp.

Fig.10 là sơ đồ khối minh họa ví dụ về cấu hình của bộ giải mã.

Fig.11 là lưu đồ minh họa xử lý giải mã tệp.

Fig.12 là lưu đồ minh họa ví dụ khác về xử lý giải mã tệp.

Fig.13 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị mã hóa, được áp dụng sáng chế, theo phương án thứ hai.

Fig.14 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị giải mã, được áp dụng sáng chế, theo phương án thứ hai.

Fig.15 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 thông thường.

Fig.16 là sơ đồ minh họa ví dụ về cấu hình của một đơn vị truy cập của luồng AVC.

Fig.17 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 theo sáng chế.

Fig.18 là sơ đồ minh họa ví dụ khác về cấu trúc mẫu video MP4 theo sáng chế.

Fig.19 là sơ đồ minh họa ví dụ về sự mở rộng hộp tương thích HRD (HRD Conformance Box extension).

Fig.20 là sơ đồ minh họa ví dụ khác nữa về cấu trúc mẫu video MP4 theo

sáng chế.

Fig.21 là sơ đồ minh họa ví dụ về hộp được bổ sung vào bảng mẫu.

Fig.22 là sơ đồ minh họa ví dụ khác về cấu trúc mẫu video MP4 được minh họa trên Fig.20.

Fig.23 là sơ đồ minh họa ví dụ khác nữa về cấu trúc mẫu video MP4 được minh họa trên Fig.20.

Fig.24 là sơ đồ minh họa ví dụ về định dạng của hàm thông tin bổ sung mẫu.

Fig.25 là sơ đồ minh họa ví dụ khác về định dạng của hàm thông tin bổ sung mẫu.

Fig.26 là sơ đồ so sánh các mẫu của cấu trúc mẫu video MP4 theo sáng chế.

Fig.27 là sơ đồ minh họa ví dụ khác về cấu trúc mẫu video MP4 được minh họa trên Fig.23.

Fig.28 là sơ đồ minh họa ví dụ về hộp thông tin định dạng luồng byte.

Fig.29 là sơ đồ minh họa ví dụ khác về hộp thông tin định dạng luồng byte.

Fig.30 là sơ đồ minh họa ví dụ về cấu trúc dữ liệu của thông tin định dạng luồng byte được lưu giữ trong từng mục nhập mẫu.

Fig.31 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 1.

Fig.32 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 2.

Fig.33 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 3.

Fig.34 là sơ đồ minh họa ví dụ khác về cấu trúc dữ liệu của thông tin định dạng luồng byte được lưu giữ trong từng mục nhập mẫu.

Fig.35 là sơ đồ minh họa ví dụ về cấu hình của thiết bị chuyên đổi tệp

được áp dụng sáng chế.

Fig.36 là lưu đồ minh họa xử lý chuyển đổi tệp.

Fig.37 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị mã hóa, được áp dụng sáng chế, theo phương án thứ ba.

Fig.38 là sơ đồ khối minh họa ví dụ về cấu hình của bộ mã hóa.

Fig.39 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị giải mã, được áp dụng sáng chế, theo phương án thứ ba.

Fig.40 là sơ đồ khối minh họa ví dụ về cấu hình của bộ giải mã.

Fig.41 là sơ đồ khối minh họa ví dụ về cấu hình của máy tính.

Fig.42 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của thiết bị truyền hình.

Fig.43 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của điện thoại di động.

Fig.44 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của thiết bị ghi/tái tạo.

Fig.45 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của thiết bị tạo ảnh.

Fig.46 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của tập video.

Fig.47 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của bộ xử lý video.

Fig.48 là sơ đồ khối minh họa ví dụ khác về cấu hình giản lược của bộ xử lý video.

Mô tả chi tiết sáng chế

Các phương án thực hiện sáng chế (dưới đây được gọi là các phương án) sẽ được mô tả dưới đây. Phần mô tả sẽ được thực hiện theo thứ tự sau.

1. Phương án thứ nhất (thiết bị mã hóa và thiết bị giải mã của hệ thống HEVC)

2. Phương án thứ hai (thiết bị mã hóa ảnh và thiết bị giải mã ảnh)
3. Phương án thứ ba (cấu trúc mẫu video MP4)
4. Phương án thứ tư (hộp thông tin định dạng luồng byte)
5. Phương án thứ năm (thiết bị chuyển đổi tệp)
6. Phương án thứ sáu (thiết bị mã hóa và thiết bị giải mã của hệ thống AVC)
7. Ví dụ ứng dụng
8. Phương án thứ bảy (Tập hợp/Bộ/Môđun/Bộ xử lý)
1. Phương án thứ nhất

Ví dụ cấu hình của thiết bị mã hóa theo phương án thứ nhất

Fig.1 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị mã hóa theo phương án thứ nhất là thiết bị xử lý ảnh được áp dụng sáng chế.

Thiết bị mã hóa được minh họa trên Fig.1 được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 và tạo tệp trong đó luồng đã được mã hóa được lưu giữ bằng cách mã hóa ảnh như ảnh được chụp bằng cách sử dụng hệ thống HEVC.

Cụ thể hơn là, ảnh như ảnh được chụp được tạo thành theo các đơn vị khung được nhập vào bộ mã hóa 1 của thiết bị mã hóa làm tín hiệu đầu vào. Ngoài ra, tập thông số trình tự (Sequence Parameter Set, viết tắt là SPS) và tập thông số hình ảnh (Picture Parameter Set, viết tắt là PPS) được thiết lập trên tầng trước của bộ mã hóa 1 không được minh họa trên hình vẽ, thông tin sử dụng được video (Video Usability Information, viết tắt là VUI) mà thể hiện đặc trưng (khả năng sử dụng) của ảnh tương ứng với dữ liệu đã được mã hóa đối với từng trình tự, thông tin cải tiến bổ sung (Supplemental Enhancement Information, viết tắt là SEI), và tương tự được nhập vào đó.

Bộ mã hóa 1 mã hóa tín hiệu đầu vào bằng cách sử dụng các thông số có trong SPS, PPS, VUI, và SEI bằng cách sử dụng hệ thống HEVC. Sau đó, bộ mã hóa 1 tạo luồng đã được mã hóa như luồng vận chuyển MPEG-2 TS (Transport Stream, viết tắt là TS) dựa vào SPS, PPS, VUI, và SEI và dữ liệu đã được mã hóa thu nhận được nhờ kết quả của xử lý mã hóa và cấp luồng đã được mã hóa

được tạo ra đến bộ tạo tệp 2. Bộ mã hóa 1 sẽ được mô tả chi tiết sau có dựa vào Fig.2.

Bộ tạo tệp 2 tạo tệp (ví dụ, tệp của MPEG-4 phần 14 (ISO/IEC14496-14:2003, dưới đây được gọi là MP4) lưu giữ luồng đã được mã hóa (chuỗi của dữ liệu ảnh đã được mã hóa) được cấp từ bộ mã hóa 1. Bộ tạo tệp 2 sẽ được mô tả chi tiết sau có dựa vào Fig.3.

Ví dụ cấu hình của bộ mã hóa

Fig.2 là sơ đồ khối minh họa ví dụ về cấu hình của bộ mã hóa 1 được minh họa trên Fig.1. Từng bộ phận của bộ mã hóa 1 mã hóa ảnh là tín hiệu đầu vào bằng cách sử dụng các thông số có trong SPS, PPS, VUI, và SEI được cấp từ tầng trước không được minh họa trên hình vẽ, bằng cách đó tạo luồng đã được mã hóa.

Bộ mã hóa 1 được minh họa trên Fig.2 được tạo thành bao gồm: bộ chuyển đổi A/D 11; bộ đệm sắp xếp lại màn hình 12; bộ tính toán 13; bộ biến đổi trực giao 14; bộ lượng tử hóa 15; bộ mã hóa không tổn hao 16; bộ đệm tích lũy 17; bộ lượng tử hóa ngược 18; bộ biến đổi trực giao ngược 19; bộ cộng 20; bộ lọc giải khối 21; bộ nhớ khung 22; chuyển mạch 23; bộ dự báo trong 24; bộ dự báo/bù chuyển động 25; bộ lựa chọn ảnh đã được dự báo 26; và bộ điều khiển tốc độ 27.

Ngoài ra, giữa bộ lọc giải khối 21 và bộ nhớ khung 22 có bố trí bộ lọc dịch vị thích ứng 41 và bộ lọc vòng lặp thích ứng 42.

Cụ thể hơn là, bộ chuyển đổi A/D 11 của bộ mã hóa 1 thực hiện chuyển đổi A/D của ảnh, mà được tạo thành theo các đơn vị khung, được nhập như là tín hiệu đầu vào và xuất ảnh đã được chuyển đổi đến bộ đệm sắp xếp lại màn hình 12 để được lưu giữ trong đó. Bộ đệm sắp xếp lại màn hình 12 sắp xếp lại ảnh đã được lưu giữ, mà được tạo thành theo các đơn vị khung theo thứ tự hiển thị, để mã hóa theo cấu trúc nhóm hình ảnh (Group of Picture, viết tắt là GOP) và xuất ảnh đã được sắp xếp lại đến bộ tính toán 13, bộ dự báo trong 24, và bộ dự báo/bù chuyển động 25.

Bộ tính toán 13 thực hiện mã hóa bằng cách tính toán chênh lệch giữa ảnh

đã được dự báo được cấp từ bộ lựa chọn ảnh đã được dự báo 26 và ảnh mã hóa hiện thời được xuất từ bộ đệm sắp xếp lại màn hình 12. Cụ thể hơn là, bộ tính toán 13 trừ ảnh đã được dự báo được cấp từ bộ lựa chọn ảnh đã được dự báo 26 từ ảnh mã hóa hiện thời được xuất từ bộ đệm sắp xếp lại màn hình 12. Bộ tính toán 13 xuất ảnh thu nhận được như vậy đến bộ biến đổi trực giao 14 làm thông tin chênh lệch. Ngoài ra, trong trường hợp mà ảnh đã được dự báo không được cấp từ bộ lựa chọn ảnh đã được dự báo 26, bộ tính toán 13 xuất ảnh được đọc từ bộ đệm sắp xếp lại màn hình 12 đến bộ biến đổi trực giao 14 bình thường như là thông tin chênh lệch.

Bộ biến đổi trực giao 14 thực hiện biến đổi trực giao đối với thông tin chênh lệch được cấp từ bộ tính toán 13 và cấp các hệ số thu nhận được nhờ việc biến đổi trực giao đến bộ lượng tử hóa 15.

Bộ lượng tử hóa 15 lượng tử hóa các hệ số được cấp từ bộ biến đổi trực giao 14. Các hệ số đã được lượng tử hóa được nhập vào bộ mã hóa không tổn hao 16.

Bộ mã hóa không tổn hao 16 thu nhận thông tin (dưới đây được gọi là thông tin chế độ dự báo trong) thể hiện chế độ dự báo trong tối ưu từ bộ dự báo trong 24. Ngoài ra, bộ mã hóa không tổn hao 16 thu nhận thông tin thể hiện chế độ dự báo giữa tối ưu (dưới đây được gọi là thông tin chế độ dự báo giữa), vector chuyển động, thông tin được sử dụng để xác định ảnh tham chiếu, và tương tự từ bộ dự báo/bù chuyển động 25. Hơn thế nữa, bộ mã hóa không tổn hao 16 thu nhận các cờ đã được lưu giữ, chỉ số hoặc dịch vị, và thông tin loại từ bộ lọc dịch vị thích ứng 41 làm thông tin lọc dịch vị và thu nhận các hệ số lọc từ bộ lọc vòng lặp thích ứng 42.

Bộ mã hóa không tổn hao 16 thực hiện mã hóa không tổn hao như mã hóa độ dài thay đổi (ví dụ, mã hóa độ dài thay đổi thích ứng ngữ cảnh (Context-Adaptive Variable Length Coding, viết tắt là CAVLC) hoặc tương tự), mã hóa số học (ví dụ, mã hóa số học nhị phân thích ứng ngữ cảnh (Context-Adaptive Binary Arithmetic Coding, viết tắt là CABAC) hoặc tương tự) đối với các hệ số đã được lượng tử hóa được cấp từ bộ lượng tử hóa 15.

Ngoài ra, bộ mã hóa không tổn hao 16 thực hiện mã hóa không tổn hao

thông tin chế độ dự báo trong hoặc thông tin chế độ dự báo giữa, vector chuyển động, thông tin xác định ảnh tham chiếu, thông tin lọc dịch vị, các hệ số lọc, và tương tự làm thông tin mã hóa liên quan đến mã hóa. Bộ mã hóa không tốn hao 16 tạo luồng đã được mã hóa dựa vào thông tin mã hóa và các hệ số mà đã được mã hóa theo cách không tốn hao và cấp luồng đã được mã hóa được tạo ra đến bộ đệm tích lũy 17 để được tích lũy trong đó. Ở đây, thông tin mã hóa mà được mã hóa theo cách không tốn hao có thể được coi là thông tin đoạn đầu của các hệ số mà được mã hóa theo cách không tốn hao.

Bộ đệm tích lũy 17 lưu giữ tạm thời luồng đã được mã hóa được cấp từ bộ mã hóa không tốn hao 16. Ngoài ra, bộ đệm tích lũy 17 cấp luồng đã được mã hóa được lưu giữ đến bộ tạo tệp 2 được minh họa trên Fig.1.

Ngoài ra, các hệ số đã được lượng tử hóa được xuất từ bộ lượng tử hóa 15 cũng được nhập vào bộ lượng tử hóa ngược 18, được lượng tử hóa ngược, và sau đó, được cấp đến bộ biến đổi trực giao ngược 19.

Bộ biến đổi trực giao ngược 19 thực hiện biến đổi trực giao ngược đối với các hệ số được cấp từ bộ lượng tử hóa ngược 18 và cấp thông tin chênh lệch thu nhận được như vậy đến bộ cộng 20.

Bộ cộng 20 cộng thông tin chênh lệch như là thông tin giải mã hiện thời được cấp từ bộ biến đổi trực giao ngược 19 và ảnh đã được dự báo được cấp từ bộ lựa chọn ảnh đã được dự báo 26 cùng nhau, bằng cách đó thu nhận ảnh được giải mã cục bộ. Ngoài ra, trong trường hợp mà ảnh đã được dự báo không được cấp từ bộ lựa chọn ảnh đã được dự báo 26, bộ cộng 20 thiết lập thông tin chênh lệch được cấp từ bộ biến đổi trực giao ngược 19 như là ảnh được giải mã cục bộ. Bộ cộng 20 cấp ảnh được giải mã cục bộ đến bộ lọc giải khối 21 và cấp ảnh được giải mã cục bộ đến bộ nhớ khung 22 để được tích lũy trong đó.

Bộ lọc giải khối 21 lọc ảnh, mà được giải mã cục bộ, được cấp từ bộ cộng 20, bằng cách đó loại bỏ biến dạng khối. Bộ lọc giải khối 21 cấp ảnh thu nhận được như vậy đến bộ lọc dịch vị thích ứng 41.

Bộ lọc dịch vị thích ứng 41 thực hiện xử lý lọc dịch vị thích ứng (dịch vị thích ứng mẫu - Sample adaptive offset, viết tắt là SAO) chủ yếu loại bỏ dao

động đối với ảnh mà xử lý lọc giải khối thích ứng đã được thực hiện bởi bộ lọc giải khối 21.

Mô tả một cách chi tiết hơn, bộ lọc dịch vị thích ứng 41 xác định loại xử lý lọc dịch vị thích ứng đối với từng đơn vị mã hóa lớn nhất (Largest Coding Unit, viết tắt là LCU) mà là đơn vị mã hóa tối đa và thu nhận dịch vị mà được sử dụng cho xử lý lọc dịch vị thích ứng. Bộ lọc dịch vị thích ứng 41 thực hiện xử lý lọc dịch vị thích ứng của loại đã được xác định cho ảnh thu nhận được sau xử lý lọc giải khối thích ứng bằng cách sử dụng dịch vị thu nhận được. Sau đó, bộ lọc dịch vị thích ứng 41 cấp ảnh thu nhận được sau xử lý lọc dịch vị thích ứng đến bộ lọc vòng lặp thích ứng 42.

Ngoài ra, bộ lọc dịch vị thích ứng 41 bao gồm bộ đệm mà lưu giữ dịch vị. Bộ lọc dịch vị thích ứng 41 xác định liệu dịch vị được sử dụng cho xử lý lọc giải khối thích ứng đã được lưu giữ trong bộ đệm cho từng LCU hay chưa.

Trong trường hợp mà dịch vị được sử dụng cho xử lý lọc giải khối thích ứng được xác định là đã được lưu giữ trong bộ đệm, bộ lọc dịch vị thích ứng 41 thiết lập cờ lưu giữ thể hiện việc liệu dịch vị có được lưu giữ trong bộ đệm hay không ở giá trị (ở đây, 1) thể hiện là dịch vị được lưu giữ trong bộ đệm.

Sau đó, bộ lọc dịch vị thích ứng 41 cấp cờ lưu giữ được thiết lập ở "1", chỉ số thể hiện vị trí lưu giữ của dịch vị trong bộ đệm, và thông tin loại thể hiện loại của xử lý lọc dịch vị thích ứng đã được thực hiện đến bộ mã hóa không tốn hao 16 cho từng LCU.

Mặt khác, trong trường hợp mà dịch vị được sử dụng cho xử lý lọc giải khối thích ứng chưa được lưu giữ trong bộ đệm, bộ lọc dịch vị thích ứng 41 theo tuần tự lưu giữ dịch vị trong bộ đệm. Ngoài ra, bộ lọc dịch vị thích ứng 41 thiết lập cờ lưu giữ ở giá trị (ở đây, "0") thể hiện là dịch vị không được lưu giữ trong bộ đệm. Sau đó, bộ lọc dịch vị thích ứng 41 cấp cờ lưu giữ được thiết lập ở "0", dịch vị, và thông tin loại đến bộ mã hóa không tốn hao 16 cho từng LCU.

Bộ lọc vòng lặp thích ứng 42, ví dụ, thực hiện xử lý lọc vòng lặp thích ứng (ALF) đối với ảnh, mà thu nhận được sau xử lý lọc dịch vị thích ứng, được cấp từ bộ lọc dịch vị thích ứng 41 cho từng LCU. Để làm xử lý lọc vòng lặp

thích ứng, ví dụ, xử lý sử dụng bộ lọc Wiener hai chiều được sử dụng. Thấy rõ là bộ lọc không phải là bộ lọc Wiener có thể được sử dụng.

Cụ thể hơn là, bộ lọc vòng lặp thích ứng 42 tính toán các hệ số lọc được sử dụng cho xử lý lọc vòng lặp thích ứng cho từng LCU để chênh lệch giữa ảnh gốc mà là ảnh được xuất từ bộ đệm sắp xếp lại màn hình 12 và ảnh thu nhận được sau xử lý lọc vòng lặp thích ứng được giảm thiểu. Sau đó, bộ lọc vòng lặp thích ứng 42 thực hiện xử lý lọc vòng lặp thích ứng đối với ảnh thu nhận được sau xử lý lọc dịch vị thích ứng bằng cách sử dụng các hệ số lọc đã tính toán được cho từng LCU.

Bộ lọc vòng lặp thích ứng 42 cấp ảnh thu nhận được sau xử lý lọc vòng lặp thích ứng đến bộ nhớ khung 22. Ngoài ra, bộ lọc vòng lặp thích ứng 42 cấp các hệ số lọc đến bộ mã hóa không tổn hao 16.

Ở đây, mặc dù xử lý lọc vòng lặp thích ứng được thực hiện đối với từng LCU, tuy nhiên đơn vị xử lý của xử lý lọc vòng lặp thích ứng không bị giới hạn ở LCU. Tuy nhiên, bằng cách so khớp các đơn vị xử lý của bộ lọc dịch vị thích ứng 41 và bộ lọc vòng lặp thích ứng 42, xử lý có thể được thực hiện một cách hiệu quả.

Ảnh được tích lũy trong bộ nhớ khung 22 được xuất đến bộ dự báo trong 24 hoặc bộ dự báo/bù chuyển động 25 qua chuyển mạch 23 như là ảnh tham chiếu.

Bộ dự báo trong 24 thực hiện xử lý dự báo trong của từng chế độ dự báo trong của tất cả các chế độ dự báo trong mà là các tùy chọn theo các đơn vị tấp và lát bằng cách sử dụng ảnh tham chiếu, mà chưa được lọc bởi bộ lọc giải khối 21, được đọc từ bộ nhớ khung 22 qua chuyển mạch 23

Ngoài ra, bộ dự báo trong 24 tính toán các giá trị hàm chi phí (sẽ được mô tả chi tiết sau) cho tất cả các chế độ dự báo trong mà là các tùy chọn dựa vào ảnh được đọc từ bộ đệm sắp xếp lại màn hình 12 và ảnh đã được dự báo được tạo ra nhờ kết quả của xử lý dự báo trong. Sau đó, bộ dự báo trong 24 xác định chế độ dự báo trong có giá trị hàm chi phí là nhỏ nhất làm chế độ dự báo trong tối ưu.

Bộ dự báo trong 24 cấp ảnh đã được dự báo được tạo ra trong chế độ dự báo trong tối ưu và giá trị hàm chi phí tương ứng đến bộ lựa chọn ảnh đã được dự báo 26. Trong trường hợp mà bộ dự báo trong 24 được thông báo về việc lựa chọn ảnh đã được dự báo được tạo ra trong chế độ dự báo trong tối ưu từ bộ lựa chọn ảnh đã được dự báo 26, bộ dự báo trong 24 cấp thông tin chế độ dự báo trong đến bộ mã hóa không tổn hao 16.

Giá trị hàm chi phí cũng được gọi là chi phí biến dạng tỷ lệ (Rate Distortion, viết tắt là RD) và, ví dụ, được tính toán bằng cách sử dụng kỹ thuật của một trong số chế độ phức hợp cao và chế độ phức hợp thấp như được định nghĩa trong mẫu liên kết (Joint Model, viết tắt là JM) mà là phần mềm tham chiếu trong hệ thống H.264/AVC.

Cụ thể hơn là, trong trường hợp mà chế độ phức hợp cao được sử dụng làm kỹ thuật để tính toán giá trị hàm chi phí, cho tất cả các chế độ dự báo mà là các tùy chọn, xử lý cho đến mã hóa không tổn hao được thực hiện tạm thời, và giá trị hàm chi phí được thể hiện trong phương trình (1) dưới đây được tính toán cho từng chế độ dự báo.

$$\text{Chi phí(Chế độ)} (\text{Cost}(\text{Mode})) = D + \lambda \cdot R \cdots (1)$$

Ở đây, D là chênh lệch giữa ảnh gốc và ảnh đã được giải mã, R là lượng mã hóa đã được tạo ra bao gồm cho đến các hệ số của biến đổi trực giao, và λ là số nhân Lagrange mà được đưa ra làm hàm của thông số lượng tử hóa QP.

Mặt khác, trong trường hợp mà chế độ phức hợp thấp được sử dụng làm kỹ thuật để tính toán giá trị hàm chi phí, cho từng chế độ dự báo của tất cả các chế độ dự báo mà là các tùy chọn, ảnh đã được giải mã được tạo ra, và bit đầu như thông tin thể hiện chế độ dự báo được tính toán, và giá trị hàm chi phí được thể hiện trong phương trình (2) dưới đây được tính toán cho từng chế độ dự báo.

$$\text{Cost}(\text{Mode}) = D + \text{QPtoQuant}(\text{QP}) \cdot \text{Header_Bit} \cdots (2)$$

Ở đây, D là chênh lệch (biến dạng) giữa ảnh gốc và ảnh đã được giải mã, Header_Bit là bit đầu cho chế độ dự báo, và QPtoQuant là hàm được đưa ra như là hàm của thông số lượng tử hóa QP.

Trong chế độ phức hợp thấp, chỉ các ảnh đã được giải mã có thể được tạo

ra cho tất cả các chế độ dự báo, và mã hóa không tổn hao không cần được thực hiện, bằng cách đó lượng tính toán sẽ được giảm.

Bộ dự báo/bù chuyển động 25 thực hiện các xử lý dự báo/bù chuyển động trong từng chế độ dự báo giữa của tất cả các chế độ dự báo giữa mà là các tùy chọn theo các đơn vị tấm và lát. Cụ thể hơn là, bộ dự báo/bù chuyển động 25 dò vectơ chuyển động của từng chế độ dự báo giữa của tất cả các chế độ dự báo giữa mà là các tùy chọn theo các đơn vị tấm và lát dựa vào ảnh được cấp từ bộ đệm sắp xếp lại màn hình 12 và ảnh tham chiếu đã được lọc được đọc từ bộ nhớ khung 22 qua chuyển mạch 23. Sau đó, bộ dự báo/bù chuyển động 25 thực hiện xử lý bù cho ảnh tham chiếu dựa vào vectơ chuyển động theo các đơn vị tấm và lát, bằng cách đó tạo ảnh đã được dự báo.

Lúc này, bộ dự báo/bù chuyển động 25 tính toán giá trị hàm chi phí cho từng chế độ dự báo giữa của tất cả các chế độ dự báo giữa mà là các tùy chọn dựa vào ảnh được cấp từ bộ đệm sắp xếp lại màn hình 12 và ảnh đã được dự báo và xác định chế độ dự báo giữa có giá trị hàm chi phí là nhỏ nhất làm chế độ dự báo giữa tối ưu. Sau đó, bộ dự báo/bù chuyển động 25 cấp giá trị hàm chi phí của chế độ dự báo giữa tối ưu và ảnh đã được dự báo tương ứng đến bộ lựa chọn ảnh đã được dự báo 26. Ngoài ra, trong trường hợp mà bộ dự báo/bù chuyển động 25 được thông báo về việc lựa chọn của ảnh đã được dự báo được tạo ra trong chế độ dự báo giữa tối ưu từ bộ lựa chọn ảnh đã được dự báo 26, bộ dự báo/bù chuyển động 25 xuất thông tin chế độ dự báo giữa, vectơ chuyển động tương ứng, thông tin xác định ảnh tham chiếu, và tương tự đến bộ mã hóa không tổn hao 16.

Bộ lựa chọn ảnh đã được dự báo 26 xác định một trong số chế độ dự báo trong tối ưu và chế độ dự báo giữa tối ưu mà có giá trị hàm chi phí nhỏ hơn làm chế độ dự báo tối ưu dựa vào các giá trị hàm chi phí được cấp từ bộ dự báo trong 24 và bộ dự báo/bù chuyển động 25. Sau đó, bộ lựa chọn ảnh đã được dự báo 26 cấp ảnh đã được dự báo của chế độ dự báo tối ưu đến bộ tính toán 13 và bộ cộng 20. Ngoài ra, bộ lựa chọn ảnh đã được dự báo 26 thông báo cho bộ dự báo trong 24 hoặc bộ dự báo/bù chuyển động 25 về việc lựa chọn ảnh đã được dự báo của chế độ dự báo tối ưu.

Bộ điều khiển tốc độ 27 thực hiện điều khiển hoạt động lượng tử hóa được thực hiện bởi bộ lượng tử hóa 15 dựa vào dữ liệu đã được mã hóa được tích lũy trong bộ đệm tích lũy 17 để tràn trên hoặc tràn dưới không xảy ra.

Cơ sở và vấn đề

Trong định dạng tệp AVC được định nghĩa trong ISO/IEC 14496-15, không được phép đưa mã bắt đầu và dữ liệu đệm vào trong luồng bit (luồng sơ cấp, Elementary Stream). Tuy nhiên, tải trọng xử lý khi xử lý dữ liệu của định dạng tệp phương tiện cơ sở ISO, MPEG-2 TS (luồng vận chuyển, Transport Stream), và tương tự dễ bị tăng.

Ngoài ra, để tạo (biến đổi) MPEG-2 TS từ luồng bit (luồng sơ cấp), có các trường hợp mà giá trị của thông số quản lý bộ đệm giải mã trở nên khác nhau bằng cách loại bỏ mã bắt đầu và dữ liệu đệm (dữ liệu giả được sử dụng để điều chỉnh kích thước dữ liệu). Do đó, ví dụ, cần thiết lập lại khoảng đệm SEI (thông tin cải tiến bổ sung), mà dẫn đến làm tăng tải trọng xử lý khi biến đổi luồng bit.

Hoạt động của bộ tạo tệp

Do vậy, trong trường hợp mà luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng tại thời gian lưu giữ dữ liệu trong tệp được tạo ra, bộ tạo tệp 2 thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và thực hiện điều khiển việc thiết lập mã bắt đầu và dữ liệu đệm để mã bắt đầu và dữ liệu đệm, mà đã được thiết lập, tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

Ngoài ra, trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh, bộ tạo tệp 2 thiết lập đặc tính của thông số quản lý bộ đệm giải mã và tạo tệp bằng cách sử dụng đặc tính đã được thiết lập.

Ở đây, thông số quản lý bộ đệm giải mã là thông số giải mã tham chiếu giả thiết (Hypothetical Reference Decoder Parameter, viết tắt là HRD), nói cách khác, thông số mà quản lý bộ giải mã tham chiếu ảo. Thông số mà quản lý bộ đệm giải mã này là thông số có trong VUI, thông số có trong khoảng đệm SEI,

hoặc thông số có trong SEI thời gian hình ảnh.

Bằng cấu hình như trên đây, trong trường hợp mà luồng tại thời gian truyền hoặc tại thời gian lưu giữ dữ liệu trong tệp được tạo ra (các biến đổi), mã bắt đầu và dữ liệu đệm không cần được thiết lập lại, và gánh nặng xử lý có thể được giảm.

Ngoài ra, dữ liệu đệm có thể được đáp lại như dưới đây. Nói cách khác, trong hệ thống AVC hoặc HEVC, dữ liệu đệm là phi-VCL sẽ có trong bảng của các mã loại đơn vị NAL và các lớp loại đơn vị NAL. Cụ thể hơn là, khi `nal_unit_type` là 31, tên của `nal_unit_type` là `FD_NUT`, và nội dung của đơn vị NAL và cú pháp RBSP là dữ liệu đệm và `filler_data_rbsp()`, lớp loại đơn vị NAL là phi-VCL. Bằng cách thay đổi phi-VCL này thành dữ liệu VCL, dữ liệu đệm có thể sẽ có trong dữ liệu phương tiện của tệp, và do đó, nó có thể được giảm để dữ liệu đệm được loại bỏ khi biến đổi thành luồng truyền.

Cụ thể hơn là, bộ tạo tệp 2 thực hiện thiết lập mã bắt đầu và dữ liệu đệm như dưới đây.

- Mã bắt đầu và dữ liệu đệm

Do đó, sự đóng góp này đề xuất cho phép đưa mã bắt đầu và dữ liệu đệm vào trong các luồng thứ cấp chỉ cho các nhãn mới.

Trong phiên bản ban đầu của ISO/IEC 14496-15, SPS/PPS không thể có trong luồng thứ cấp. Trong ISO/IEC 14496-16 PDAM2, bây giờ được phép đưa vào SPS/PPS. Nếu được đưa mã bắt đầu và dữ liệu đệm vào trong luồng thứ cấp, thì gánh nặng cho bộ biến đổi luồng bit được giảm đáng kể. Việc ghi lại các thông số HRD, ví dụ, khoảng đệm SEI và SEI thời gian hình ảnh, là không cần thiết trong trường hợp này.

Để đảm bảo độ tương thích về sau, thay đổi này nên được áp dụng cho các nhãn mới. (nhãn AVC mới và HEVC)

- Biểu thị các thông số HRD là sai

Khi mã bắt đầu, dữ liệu đệm, v.v., được loại bỏ khỏi luồng thứ cấp để lưu giữ ISOMFF, các thông số HRD, ví dụ, khoảng đệm SEI, SEI thời gian hình ảnh, v.v., là không đúng. Tuy nhiên, bộ giải mã có thể không biết việc liệu các thông

số HRD trong luồng thứ cấp có đúng hay không. Cần có id để biểu thị các thông số HRD trong luồng thứ cấp có thể là sai.

- Trong HEVC hoặc AVC thay đổi dữ liệu đệm làm dữ liệu VCL

Hiện tại, trong HEVC và AVC, dữ liệu đệm được phân loại là phi-VCL. Điều đó có nghĩa là bit cho dữ liệu đệm được tính là phi-VCL (thông tin đoạn đầu). Tuy nhiên, nếu xác định dữ liệu đệm là dữ liệu VCL, bit cho đệm được tính là chính luồng bit video. Trong trường hợp này, dữ liệu đệm không cần thiết được loại bỏ khỏi luồng thứ cấp và các thông số HRD không thay đổi. Bảng cho HEVC được gắn.

Hơn thế nữa, trong trường hợp mà thông số mà quản lý bộ đệm giải mã là khác nhau (không đúng), bộ giải mã không thể xác định liệu thông số quản lý bộ đệm giải mã có phải là giống nhau (đúng) hay không. Do vậy, trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit, thông số nhận dạng được sử dụng để nhận dạng việc liệu đặc tính của thông số quản lý bộ đệm giải mã có thay đổi hay không được thiết lập. Ví dụ, trường hợp sẽ được thể hiện trong đó thông số nhận dạng được lưu giữ trong hộp tùy chọn của định dạng tệp.

Ví dụ, trong trường hợp của AVC, id như vậy có thể được định nghĩa là hộp tùy chọn trong AVCSampleEntry như sau. (Cùng cách tiếp cận có thể được áp dụng cho các mục nhập mẫu AVC khác và HEVCSampleEntry.)

```
AVCSampleEntry() extends VisualSampleEntry(type') {
    //type is 'avc1' hoặc 'avc3'
    AVCCConfigurationBox config;
    MPEG4BitRateBox(); //Optional
    MPEG4ExtensionDescriptorsBox(); //Optional
    HRDConformanceBox(); //Optional
}
```

Thêm HRDConformanceBox vào AVCSampleEntry và AVC2SVVSampleEntry trong 5.4.2.1

Thêm HRDConformanceBox vào AVCSVCSampleEntry, AVC2SVCSampleEntry và SVCSampleEntry trong 6.5.3.1.

Thêm HRDConformanceBox vào AVCMVCSampleEntry, AVC2MVCSampleEntry và MVCSampleEntry trong mục 7.6.3.3.

Thêm HRDConformanceBox vào HEVC SampleEntry trong mục 8.4.1.1.

HRDConformanceBox có thể được định nghĩa như sau.

```
class HRDConformanceBox extends Box('hrdc') {
    HRDConformanceData() HRDConformance;
}

aligned(8) class HRDConformanceData {
    unsigned int(7) reserved =0;
    unsigned int(1) HRDConformanceFlag;
    unsigned int (24) reserved;
    Box[] any_box;//Optional
}
```

Thông số nhận dạng, ví dụ, được thiết lập như dưới đây.

HRDConformanceFlag là bộ nhận dạng nếu các đặc tính HRD được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., từ luồng thứ cấp. Khi HRDConformanceFlag bằng 1, các thông số HRD, ví dụ, khoảng đệm SEI, SEI thời gian hình ảnh, được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., khỏi luồng thứ cấp.

Bộ tạo tệp 2, mà được minh họa trên Fig.1, thực hiện xử lý được mô tả trên đây được tạo cấu hình như được minh họa trên Fig.3.

Ví dụ cấu hình của bộ tạo tệp

Fig.3 là sơ đồ khối minh họa ví dụ về cấu hình của bộ tạo tệp mà thực hiện xử lý được mô tả trên đây.

Trong ví dụ được minh họa trên Fig.3, bộ tạo tệp 2 được tạo ra cấu hình để

bao gồm bộ điều khiển 31, bộ thiết lập 32, và bộ ghi tệp 33.

Bộ điều khiển 31 điều khiển xử lý tạo dữ liệu mdat, mà là một phần dữ liệu VCL của tệp, mà được thực hiện bằng bộ thiết lập 32 dựa vào việc liệu đặc tính của thông số HRD mà là thông số quản lý bộ đệm giải mã sẽ được duy trì hay không. Ngoài ra, bộ điều khiển 31 điều khiển việc thiết lập đặc tính của thông số HRD hoặc việc thiết lập thông số nhận dạng của HRD trong moov mà là phần thông tin quản lý của tệp dựa vào việc liệu mã bắt đầu và dữ liệu đệm có được loại bỏ khỏi dữ liệu mdat được thiết lập bởi bộ thiết lập 32 hay không.

Bộ thiết lập 32 tạo dữ liệu mdat của tệp bằng cách sử dụng luồng đã được mã hóa được cấp từ bộ mã hóa 1 theo sự điều khiển của bộ điều khiển 31 và cấp dữ liệu mdat đã được tạo ra đến bộ ghi tệp 33. Trong khi mã bắt đầu và dữ liệu đệm được loại bỏ theo kỹ thuật thông thường, theo kỹ thuật của sáng chế, dữ liệu mdat được tạo ra với mã bắt đầu và dữ liệu đệm được thiết lập trong luồng đã được mã hóa được thiết lập.

Ngoài ra, bộ thiết lập 32 thiết lập lại đặc tính của thông số HRD hoặc thiết lập thông số nhận dạng của HRD trong moov của tệp theo sự điều khiển của bộ điều khiển 31. Sau đó, bộ thiết lập 32 tạo moov của tệp và cấp moov đã được tạo ra đến bộ ghi tệp 33.

Bộ ghi tệp 33 ghi moov được cấp từ bộ thiết lập 32 vào tệp và ghi mdat vào tệp. Tệp của MP4 được ghi bởi bộ ghi tệp 33 được truyền đến thiết bị giải mã sẽ được mô tả sau hoặc tương tự và được đọc và giải mã bởi thiết bị giải mã.

Ví dụ về xử lý tạo tệp

Tiếp theo, xử lý tạo tệp được thực hiện bởi bộ tạo tệp 2 sẽ được mô tả có dựa vào lưu đồ được thể hiện trên Fig.4.

Ở bước S1, bộ điều khiển 31 xác định liệu đặc tính của thông số HRD mà là thông số quản lý bộ đệm giải mã sẽ được duy trì hay không. Ở bước S1, trong trường hợp mà đặc tính của thông số HRD được xác định là sẽ không được duy trì, xử lý chuyển đến bước S2. Ở bước S2, bộ thiết lập 32 thực hiện xử lý tạo dữ liệu mdat thông thường theo sự điều khiển của bộ điều khiển 31. Xử lý này sẽ được mô tả sau có dựa vào Fig.6.

Trong xử lý của bước S2, mã bắt đầu được dò, dữ liệu NAL thu nhận được, và mã bắt đầu được loại bỏ. Sau đó, việc bổ sung kích thước đơn vị (Unit size), tích lũy của dữ liệu trong bộ đệm, hoặc loại bỏ dữ liệu được thực hiện dựa vào việc liệu dữ liệu thu nhận được có phải là dữ liệu VCL hay không, dữ liệu đệm, thông tin quản lý, hoặc tương tự, bằng cách đó dữ liệu mdat được tạo ra. Dữ liệu mdat đã được tạo ra được cấp đến bộ ghi tệp 33, và xử lý chuyển đến bước S4.

Mặt khác, ở bước S1, trong trường hợp mà đặc tính của thông số HRD được xác định sẽ được duy trì, xử lý chuyển đến bước S3. Ở bước S3, bộ thiết lập 32 thực hiện xử lý tạo dữ liệu mdat theo sáng chế theo sự điều khiển của bộ điều khiển 31. Xử lý này sẽ được mô tả sau có dựa vào Fig.7.

Trong xử lý của bước S3, mã bắt đầu được dò, dữ liệu NAL thu nhận được, dữ liệu thu nhận được được thiết lập làm dữ liệu VCL, kích thước đơn vị được bổ sung, và dữ liệu thu được được tích lũy trong bộ đệm, bằng cách đó dữ liệu mdat được tạo ra. Dữ liệu mdat đã được tạo ra được cấp đến bộ ghi tệp 33, và xử lý chuyển đến bước S4.

Ở bước S4, bộ điều khiển 31 xác định liệu mã bắt đầu và dữ liệu đệm đã được loại bỏ khỏi dữ liệu mdat được tạo ra ở bước S2 hoặc S3 hay chưa.

Ở bước S4, trong trường hợp mà mã bắt đầu và dữ liệu đệm được xác định đã được loại bỏ, xử lý chuyển đến bước S5. Ở bước S5, bộ thiết lập 32 thiết lập đặc tính của thông số HRD. Nói cách khác, vì đặc tính của thông số HRD là khác nhau, đặc tính của thông số HRD được tính toán lại và được thiết lập lại. Sau đó, SEI và VUI của dữ liệu mdat được ghi lại bằng thông số HRD đã được thiết lập.

Ở bước S4, trong trường hợp mà mã bắt đầu và dữ liệu đệm được xác định là chưa được loại bỏ, xử lý của bước S5 được bỏ qua, và xử lý chuyển đến bước S6.

Ở bước S6, bộ thiết lập 32 tạo moov của tệp bằng cách sử dụng thông tin kích thước của dữ liệu mà thu nhận được bởi xử lý của bước S2 hoặc S3. Ngoài ra, trong trường hợp mà xử lý của bước S3 được thực hiện, thông tin quản lý và tương tự được tích lũy trong bộ đệm cho moov cũng được sử dụng. moov đã

được tạo ra được cấp đến bộ ghi tệp 33.

Ở bước S7, bộ ghi tệp 33 ghi moov được tạo ra ở bước S6 vào tệp.

Ở bước S8, bộ ghi tệp 33 ghi mdat được tạo ra ở bước S2 hoặc S3 vào tệp.

Tệp của MP4 được ghi bởi bộ ghi tệp 33, như được mô tả trên đây, được truyền đến thiết bị giải mã sẽ được mô tả sau hoặc tương tự và được đọc và được giải mã bởi thiết bị giải mã. Lúc đó, tệp được tạo ra mà không thay đổi đặc tính của thông số HRD. Ngoài ra, ngay cả khi mã bắt đầu và dữ liệu đệm đã được loại bỏ, thông số HRD được thiết lập lại, và do đó, bằng cách tham chiếu thông số HRD, xử lý giải mã có thể được thực hiện đúng.

Trong phần mô tả của Fig.4, trong khi ví dụ đã được mô tả trong đó đặc tính của thông số HRD được thiết lập lại trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ, như sẽ được mô tả tiếp theo, thông số nhận dạng mà thể hiện việc liệu đặc tính của thông số HRD là đúng hay khác nhau có thể được thiết lập.

Ví dụ khác về xử lý tạo tệp

Tiếp theo, xử lý tạo tệp được thực hiện bằng bộ tạo tệp 2 sẽ được mô tả có dựa vào lưu đồ được thể hiện trên Fig.5.

Ở bước S11, bộ điều khiển 31 xác định liệu đặc tính của thông số HRD mà là thông số quản lý bộ đệm giải mã sẽ được duy trì hay không. Ở bước S11, trong trường hợp mà đặc tính của thông số HRD được xác định là sẽ không được duy trì, xử lý chuyển đến bước S12. Ở bước S12, bộ thiết lập 32 thực hiện xử lý tạo dữ liệu mdat thông thường theo sự điều khiển của bộ điều khiển 31. Xử lý này sẽ được mô tả sau có dựa vào Fig.5.

Trong xử lý của bước S12, mã bắt đầu được dò, dữ liệu NAL thu nhận được, và mã bắt đầu được loại bỏ. Sau đó, việc bổ sung kích thước đơn vị, tích lũy dữ liệu trong bộ đệm, hoặc loại bỏ dữ liệu được thực hiện dựa vào việc liệu dữ liệu thu nhận được có phải là dữ liệu VCL, dữ liệu đệm, thông tin quản lý, hoặc tương tự hay không, bằng cách đó dữ liệu dữ liệu mdat được tạo ra. Dữ liệu mdat đã được tạo ra được cấp đến bộ ghi tệp 33, và xử lý chuyển đến bước S14.

Mặt khác, ở bước S11, trong trường hợp mà đặc tính của thông số HRD

được xác định sẽ được duy trì, xử lý chuyển đến bước S13. Ở bước S13, bộ thiết lập 32 thực hiện xử lý tạo dữ liệu mdat theo sáng chế theo sự điều khiển của bộ điều khiển 31. Xử lý này sẽ được mô tả sau có dựa vào Fig.6.

Trong xử lý của bước S13, mã bắt đầu được dò, dữ liệu NAL thu nhận được, dữ liệu thu nhận được được thiết lập làm dữ liệu VCL, kích thước đơn vị được bổ sung, và dữ liệu thu được được tích lũy trong bộ đệm, bằng cách đó dữ liệu mdat được tạo ra. Dữ liệu mdat đã được tạo ra được cấp đến bộ ghi tệp 33, và xử lý chuyển đến bước S14.

Ở bước S14, bộ điều khiển 31 xác định liệu mã bắt đầu và dữ liệu đệm đã được loại bỏ khỏi dữ liệu mdat được tạo ra ở bước S12 hoặc S13 hay chưa.

Ở bước S14, trong trường hợp mà mã bắt đầu và dữ liệu đệm được xác định là chưa được loại bỏ, xử lý chuyển đến bước S15. Ở bước S15, bộ thiết lập 32 thiết lập thông số nhận dạng của HRD của moov là "1", và xử lý chuyển đến bước S17. Do đó, có thể xác định là thông số HRD là đúng ở phía giải mã.

Ở bước S14, trong trường hợp mà mã bắt đầu và dữ liệu đệm được xác định là đã được loại bỏ, xử lý chuyển đến bước S16. Ở bước S16, bộ thiết lập 32 thiết lập thông số nhận dạng của HRD của moov là "0", và xử lý chuyển đến bước S17. Do đó, có thể xác định là thông số HRD là lỗi (không đúng) ở phía giải mã.

Ở bước S17, bộ thiết lập 32 tạo moov của tệp bằng cách sử dụng thông tin kích thước của dữ liệu mà thu nhận được bởi xử lý của bước S12 hoặc S13. Ngoài ra, trong trường hợp mà xử lý của bước S13 được thực hiện, thông tin quản lý và tương tự được tích lũy trong bộ đệm cho moov cũng được sử dụng. moov đã được tạo ra được cấp đến bộ ghi tệp 33.

Ở bước S18, bộ ghi tệp 33 ghi moov được tạo ra ở bước S17 vào tệp.

Ở bước S19, bộ ghi tệp 33 ghi mdat được tạo ra ở bước S12 hoặc S13 vào tệp.

Tệp của MP4 được ghi bởi bộ ghi tệp 33, như được mô tả trên đây, được truyền đến thiết bị giải mã sẽ được mô tả sau hoặc tương tự và được đọc và được giải mã bởi thiết bị giải mã. Lúc đó, tệp được tạo ra mà không thay đổi đặc tính

của thông số HRD. Ngoài ra, vì thông số nhận dạng được sử dụng để nhận dạng việc liệu thông số HRD là đúng có được thiết lập hay không, bằng cách tham chiếu thông số HRD, có thể xác định rõ ràng việc liệu việc giải mã có được thực hiện hay không.

Ví dụ về xử lý tạo dữ liệu mdat thông thường

Tiếp theo, xử lý tạo dữ liệu mdat thông thường được thực hiện ở bước S2 (bước S12 được thể hiện trên Fig.5) được thể hiện trên Fig.4 sẽ được mô tả có dựa vào lưu đồ được minh họa trên Fig.6.

Bộ thiết lập 32 dò mã bắt đầu từ luồng đã được mã hóa được cấp từ bộ mã hóa 1 ở bước S21 và thu nhận dữ liệu NAL từ mã bắt đầu dò được ở bước S21 đến mã bắt đầu tiếp theo ở bước S22.

Ở bước S23, bộ thiết lập 32 loại bỏ mã bắt đầu mà đã dò được ở bước S21. Ở bước S24, xác định liệu dữ liệu thu nhận được ở bước S22 có phải là dữ liệu VCL hay không.

Ở bước S24, trong trường hợp mà dữ liệu thu nhận được ở bước S22 được xác định là dữ liệu VCL, xử lý chuyển đến bước S25. Ở bước S25, bộ thiết lập 32 bổ sung kích thước đơn vị tương ứng vào dữ liệu thu nhận được ở bước S22 và thiết lập dữ liệu thu được làm đơn vị truy cập. Sau đó, ở bước S26, bộ thiết lập 32 tích lũy đơn vị truy cập mà kích thước đơn vị đã được bổ sung vào trong bộ đệm mdat lấp sẵn bên trong, và xử lý chuyển đến bước S30.

Mặt khác, ở bước S24, trong trường hợp mà dữ liệu thu nhận được ở bước S22 được xác định không phải là dữ liệu VCL, xử lý chuyển đến bước S27. Ở bước S27, bộ thiết lập 32 xác định liệu dữ liệu thu nhận được ở bước S22 có phải là dữ liệu đệm hay không.

Ở bước S27, trong trường hợp mà dữ liệu thu nhận được ở bước S22 được xác định là dữ liệu đệm, xử lý chuyển đến bước S28. Ở bước S28, bộ thiết lập 32 loại bỏ dữ liệu đệm, và xử lý chuyển đến bước S30.

Mặt khác, ở bước S27, trong trường hợp mà dữ liệu thu nhận được ở bước S22 được xác định không phải là dữ liệu đệm, xử lý chuyển đến bước S29. Trong trường hợp như vậy, vì dữ liệu thu nhận được là thông tin quản lý như

SPS, PPS, hoặc tương tự, ở bước S29, bộ thiết lập 32 tích lũy dữ liệu thu nhận được trong bộ đệm moov lắp sẵn bên trong, và xử lý chuyển đến bước S30.

Ở bước S30, bộ thiết lập 32 dò mã bắt đầu từ luồng đã được mã hóa được cấp từ bộ mã hóa 1.

Ở bước S31, bộ thiết lập 32 xác định liệu dữ liệu hiện thời có phải là dữ liệu cuối cùng của tệp hay không. Trong trường hợp mà mã bắt đầu không dò được ở bước S30, ở bước S31, dữ liệu hiện thời được xác định là dữ liệu cuối cùng của tệp, xử lý tạo dữ liệu mdat kết thúc, và xử lý quay lại bước S2 (hoặc bước S12 được thể hiện trên Fig.5) được thể hiện trên Fig.4.

Mặt khác, trong trường hợp mà mã bắt đầu dò được ở bước S30, ở bước S31, trong trường hợp mà dữ liệu cuối cùng được xác định không phải là dữ liệu cuối cùng của tệp, xử lý chuyển đến bước S22, và các xử lý của các bước S22 và các bước sau đó được lặp lại.

Ví dụ về xử lý tạo dữ liệu mdat theo sáng chế

Ngược lại, xử lý tạo dữ liệu mdat theo sáng chế mà được thực hiện ở bước S3 (bước S13 được thể hiện trên Fig.5) được thể hiện trên Fig.4 sẽ được mô tả có dựa vào lưu đồ được minh họa trên Fig.7.

Bộ thiết lập 32 dò mã bắt đầu từ luồng đã được mã hóa được cấp từ bộ mã hóa 1 ở bước S41 và thu nhận dữ liệu NAL từ mã bắt đầu dò được ở bước S41 đến mã bắt đầu tiếp theo ở bước S42. Trong ví dụ được minh họa trên Fig.7, mã bắt đầu không được loại bỏ.

Ở bước S43, bộ thiết lập 32 xác định liệu dữ liệu thu nhận được ở bước S42 có phải là dữ liệu đệm hay không. Ở bước S43, trong trường hợp mà dữ liệu thu nhận được ở bước S42 được xác định là dữ liệu đệm, xử lý chuyển đến bước S44. Ở bước S44, bộ thiết lập 32 thiết lập dữ liệu đệm làm dữ liệu VCL.

Mặt khác, ở bước S43, trong trường hợp mà dữ liệu thu nhận được ở bước S42 được xác định không phải là dữ liệu đệm, xử lý của bước S44 được bỏ qua, và xử lý chuyển đến bước S45.

Ở bước S45, kích thước đơn vị tương ứng được bổ sung vào dữ liệu thu nhận được ở bước S42, dữ liệu thu nhận được được thiết lập làm đơn vị truy cập, và

đơn vị truy cập mà kích thước đơn vị đã được bổ sung vào ở bước S46 được tích lũy trong bộ đệm mdat lắp sẵn bên trong. Ở đây, trong trường hợp mà dữ liệu thu nhận được là thông tin quản lý như SPS hoặc PPS, dữ liệu cũng được tích lũy trong bộ đệm moov lắp sẵn bên trong và được lưu giữ trong mô tả mẫu (Sample Description) của moov làm SPS hoặc PPS mặc định ở bước S6 được thể hiện trên Fig.4 hoặc bước S17 được thể hiện trên Fig.5.

Ở đây, trong trường hợp mà dữ liệu thu nhận được là thông tin quản lý như SPS hoặc PPS, tương tự như trường hợp thông thường, dữ liệu thu nhận được có thể được lưu giữ không phải trong bộ đệm mdat mà chỉ trong bộ đệm moov lắp sẵn bên trong.

Ở bước S47, bộ thiết lập 32 dò mã bắt đầu từ luồng đã được mã hóa được cấp từ bộ mã hóa 1.

Ở bước S48, bộ thiết lập 32 xác định liệu dữ liệu hiện thời có phải là dữ liệu cuối cùng của tệp hay không. Trong trường hợp mà mã bắt đầu không dò được ở bước S47, ở bước S48, dữ liệu hiện thời được xử lý là dữ liệu cuối cùng của tệp, xử lý tạo dữ liệu mdat kết thúc, và xử lý quay lại bước S3 (bước S13 được thể hiện trên Fig.5) được thể hiện trên Fig.4.

Trong trường hợp mà mã bắt đầu dò được ở bước S47, ở bước S48, trong trường hợp mà dữ liệu hiện thời được xác định không phải là dữ liệu cuối cùng của tệp, xử lý chuyển đến bước S42, và các xử lý của các bước S42 và các bước sau đó được lặp lại.

Như trên đây, theo sáng chế, việc thiết lập mã bắt đầu và dữ liệu đệm được điều khiển để tệp được tạo cấu trúc với đặc tính của thông số quản lý bộ đệm giải mã được duy trì. Nói cách khác, mã bắt đầu và dữ liệu đệm không được loại bỏ, và tệp được tạo cấu trúc để bao gồm mã bắt đầu và dữ liệu đệm, và do đó, đặc tính của thông số quản lý bộ đệm giải mã có thể được duy trì.

Ngoài ra, vì dữ liệu đệm được thay đổi thành dữ liệu VCL, dữ liệu đệm có thể có trong dữ liệu phương tiện của tệp, và nó có thể được giảm để dữ liệu đệm được loại bỏ khi chuyển đổi thành luồng truyền.

Hơn thế nữa, trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ

khỏi dữ liệu phương tiện của tệp, đặc tính của thông số quản lý bộ đệm giải mã được thiết lập, và tệp được tạo ra bằng cách sử dụng thông số đã được thiết lập. Do đó, hoạt động giải mã có thể được thực hiện đúng.

Theo cách khác, trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp, thông số nhận dạng nhận dạng rằng đặc tính của thông số đã được thay đổi được thiết lập, và tệp được tạo ra bằng cách sử dụng thông số nhận dạng đã được thiết lập. Do đó, hoạt động giải mã có thể được thực hiện đúng.

Ví dụ cấu hình của thiết bị giải mã theo phương án thứ nhất

Fig.8 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị giải mã, mà giải mã luồng đã được mã hóa được truyền từ thiết bị mã hóa được minh họa trên Fig.1, theo phương án thứ nhất làm thiết bị xử lý ảnh được áp dụng sáng chế.

Thiết bị giải mã được minh họa trên Fig.8 được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52.

Bộ đọc tệp 51 của thiết bị giải mã thu tệp của MP4 được tạo ra bởi thiết bị mã hóa được minh họa trên Fig.1 và đọc là SPS, PPS, VUI, SEI, dữ liệu đã được mã hóa, và tương tự tạo thành luồng đã được mã hóa mà được mã hóa bằng bộ mã hóa 1 được minh họa trên Fig.1 từ tệp đã thu được. Bộ đọc tệp 51 cấp SPS, PPS, VUI, SEI, và dữ liệu đã được mã hóa đến bộ giải mã 52.

Cụ thể hơn là, bộ đọc tệp 51 thu tệp mà được điều khiển để được thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp mà bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh. Theo cách khác, trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh, bộ đọc tệp 51 thu tệp mà được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập. Sau đó, bộ đọc tệp 51 đọc tệp đã thu được và khiến bộ giải mã 52 giải mã dữ liệu đã được mã hóa của luồng bit.

Bộ giải mã 52, theo sự điều khiển của bộ đọc tệp 51, tham chiếu SPS, PPS,

VUI, SEI, và tương tự (cụ thể là, đặc tính của thông số HRD) được cấp từ bộ đọc tệp 51 và giải mã dữ liệu đã được mã hóa được cấp từ bộ đọc tệp 51 bằng cách sử dụng hệ thống HEVC. Bộ giải mã 52 cấp ảnh thu nhận được nhờ kết quả của xử lý giải mã đến tầng sau như là tín hiệu đầu ra.

Ví dụ cấu hình của bộ đọc tệp

Fig.9 là sơ đồ khối minh họa ví dụ về cấu hình của bộ đọc tệp 51.

Trong ví dụ được minh họa trên Fig.9, bộ đọc tệp 51 được tạo cấu hình để bao gồm: bộ thu tệp 71; bộ thu nhận thông số 72; bộ điều khiển giải mã 73; và bộ đọc dữ liệu 74.

Bộ thu tệp 71 thu tệp của MP4 được tạo ra bởi bộ tạo tệp 2 được minh họa trên Fig.1 và cấp tệp đã thu được đến bộ thu nhận thông số 72.

Bộ thu nhận thông số 72 thu nhận moov mà là thông tin quản lý từ tệp của MP4 và cấp moov đã thu nhận được đến bộ đọc dữ liệu 74. Ngoài ra, bộ thu nhận thông số 72 khiến bộ đọc dữ liệu 74 đọc dữ liệu từ mdat dựa vào moov đã thu nhận được.

Ngoài ra, trong trường hợp của tệp được tạo ra bằng xử lý tạo tệp được minh họa trên Fig.5, bộ thu nhận thông số 72 thu nhận thông số nhận dạng được sử dụng để xác định liệu đặc tính của thông số HRD có đúng hay không từ moov và cấp thông số nhận dạng đến bộ điều khiển giải mã 73.

Trong trường hợp mà thông số nhận dạng của HRD được cấp từ bộ thu nhận thông số 72 có mặt, bộ điều khiển giải mã 73 xác định liệu có theo thông số HRD dựa vào thông số nhận dạng của HRD hay không và điều khiển bộ giải mã 52, bằng cách đó thực hiện xử lý giải mã.

Trong khi đó, trong trường hợp của tệp được tạo ra bằng xử lý tạo tệp được minh họa trên Fig.4, trong khi thông số nhận dạng của HRD không có mặt trong moov, ngay cả khi mã bắt đầu và dữ liệu đệm của tệp của MP4 được loại bỏ, đặc tính của thông số HRD được thiết lập lại, và do đó, đặc tính là đúng. Do vậy, trong trường hợp này, bộ điều khiển giải mã 73 không thực hiện cụ thể hoạt động.

Bộ đọc dữ liệu 74, theo sự điều khiển của bộ thu nhận thông số 72, đọc

SPS, PPS, VUI, SEI, và tương tự từ moov hoặc mdat của tệp và đọc dữ liệu đã được mã hóa từ mdat của tệp. Sau đó, bộ đọc dữ liệu 74 cấp SPS, PPS, VUI, SEI, dữ liệu đã được mã hóa, và tương tự mà đã được đọc đến bộ giải mã 52.

Ví dụ cấu hình của bộ giải mã

Fig.10 là sơ đồ khối minh họa ví dụ về cấu hình của bộ giải mã 52 được minh họa trên Fig.8. Từng bộ phận của bộ giải mã 52 giải mã dữ liệu đã được mã hóa bằng cách sử dụng các thông số có trong SPS, PPS, VUI, và SEI được cấp từ bộ đọc tệp 51, bằng cách đó tạo ảnh.

Bộ giải mã 52 được minh họa trên Fig.10 được tạo thành bởi: bộ đệm tích lũy 101; bộ giải mã không tổn hao 102; bộ lượng tử hóa ngược 103; bộ biến đổi trực giao ngược 104; bộ cộng 105; bộ lọc giải khối 106; bộ đệm sắp xếp lại màn hình 107; bộ chuyển đổi D/A 108; bộ nhớ khung 109; chuyển mạch 110; bộ dự báo trong 111; bộ bù chuyển động 112; và chuyển mạch 113.

Ngoài ra, giữa bộ lọc giải khối 106 và bộ đệm sắp xếp lại màn hình 107 và bộ nhớ khung 109 có bố trí bộ lọc dịch vị thích ứng 141 và bộ lọc vòng lặp thích ứng 142.

Bộ đệm tích lũy 101 của bộ giải mã 52 thu dữ liệu đã được mã hóa từ bộ đọc tệp 51 được minh họa trên Fig.8 và tích lũy dữ liệu đã được mã hóa thu được. Bộ đệm tích lũy 101 cấp dữ liệu đã được mã hóa được tích lũy đến bộ giải mã không tổn hao 102.

Bộ giải mã không tổn hao 102 thực hiện giải mã không tổn hao như giải mã độ dài thay đổi hoặc giải mã số học đối với dữ liệu đã được mã hóa được cấp từ bộ đệm tích lũy 101, bằng cách đó thu nhận các hệ số đã được lượng tử hóa và thông tin mã hóa. Bộ giải mã không tổn hao 102 cấp các hệ số đã được lượng tử hóa đến bộ lượng tử hóa ngược 103. Ngoài ra, bộ giải mã không tổn hao 102 cấp thông tin chế độ dự báo trong và tương tự làm thông tin mã hóa đến bộ dự báo trong 111 và cấp vectơ chuyển động, thông tin được sử dụng để xác định ảnh tham chiếu, thông tin chế độ dự báo giữa, và tương tự đến bộ bù chuyển động 112. Hơn thế nữa, bộ giải mã không tổn hao 102 cấp thông tin chế độ dự báo trong hoặc thông tin chế độ dự báo giữa làm thông tin mã hóa đến chuyển

mạch 113.

Bộ giải mã không tổn hao 102 cấp thông tin lọc dịch vị làm thông tin mã hóa đến bộ lọc dịch vị thích ứng 141 và cấp các hệ số lọc đến bộ lọc vòng lặp thích ứng 142.

Bộ lượng tử hóa ngược 103, bộ biến đổi trực giao ngược 104, bộ cộng 105, bộ lọc giải khối 106, bộ nhớ khung 109, chuyển mạch 110, bộ dự báo trong 111, và bộ bù chuyển động 112 lần lượt thực hiện các xử lý tương tự như các xử lý của bộ lượng tử hóa ngược 18, bộ biến đổi trực giao ngược 19, bộ cộng 20, bộ lọc giải khối 21, bộ nhớ khung 22, chuyển mạch 23, bộ dự báo trong 24, và bộ dự báo/bù chuyển động 25 được minh họa trên Fig.2, bằng cách đó ảnh được giải mã.

Cụ thể hơn là, bộ lượng tử hóa ngược 103 thực hiện lượng tử hóa ngược đối với các hệ số đã được lượng tử hóa được cấp từ bộ giải mã không tổn hao 102 và cấp các hệ số thu nhận được như vậy đến bộ biến đổi trực giao ngược 104.

Bộ biến đổi trực giao ngược 104 thực hiện biến đổi trực giao ngược các hệ số được cấp từ bộ lượng tử hóa ngược 103 và cấp thông tin chênh lệch thu nhận được như vậy đến bộ cộng 105.

Bộ cộng 105 cộng thông tin chênh lệch làm ảnh giải mã hiện thời được cấp từ bộ biến đổi trực giao ngược 104 và ảnh đã được dự báo được cấp từ chuyển mạch 113 cùng nhau, bằng cách đó thực hiện xử lý giải mã. Bộ cộng 105 cấp ảnh thu nhận được nhờ kết quả của xử lý giải mã đến bộ lọc giải khối 106 và bộ nhớ khung 109. Ngoài ra, trong trường hợp mà ảnh đã được dự báo không được cấp từ chuyển mạch 113, bộ cộng 105 cấp ảnh mà là thông tin chênh lệch được cấp từ bộ biến đổi trực giao ngược 104 làm ảnh thu nhận được nhờ kết quả của xử lý giải mã đến bộ lọc giải khối 106 và cấp ảnh đến bộ nhớ khung 109 để được tích lũy trong đó.

Bộ lọc giải khối 106 lọc ảnh được cấp từ bộ cộng 105, bằng cách đó loại bỏ biến dạng khối. Bộ lọc giải khối 106 cấp ảnh thu nhận được như vậy đến bộ lọc dịch vị thích ứng 141.

Bộ lọc dịch vị thích ứng 141 bao gồm bộ đệm mà theo tuần tự lưu giữ các dịch vị được cấp từ bộ giải mã không tổn hao 102. Ngoài ra, bộ lọc dịch vị thích ứng 141 thực hiện xử lý lọc dịch vị thích ứng đối với ảnh thu nhận được sau khi xử lý lọc giải khối thích ứng được thực hiện bởi bộ lọc giải khối 106 đối với từng LCU dựa vào thông tin lọc dịch vị được cấp từ bộ giải mã không tổn hao 102.

Cụ thể hơn là, trong trường hợp mà cờ lưu giữ có trong thông tin lọc dịch vị là "0", bộ lọc dịch vị thích ứng 141 thực hiện xử lý lọc dịch vị thích ứng của loại mà được thể hiện bởi thông tin loại bằng cách sử dụng dịch vị có trong thông tin lọc dịch vị đối với ảnh thu nhận được sau khi xử lý lọc giải khối theo các đơn vị LCU.

Mặt khác, trong trường hợp mà cờ lưu giữ có trong thông tin lọc dịch vị là "1", bộ lọc dịch vị thích ứng 141, đối với ảnh thu nhận được sau khi xử lý lọc giải khối theo các đơn vị LCU, đọc dịch vị mà được lưu giữ ở vị trí được thể hiện bởi chỉ số có trong thông tin lọc dịch vị. Sau đó, bộ lọc dịch vị thích ứng 141 thực hiện xử lý lọc dịch vị thích ứng của loại mà được thể hiện bằng thông tin loại bằng cách sử dụng dịch vị đã được đọc. Bộ lọc dịch vị thích ứng 141 cấp ảnh thu nhận được sau khi xử lý lọc dịch vị thích ứng đến bộ lọc vòng lặp thích ứng 142.

Bộ lọc vòng lặp thích ứng 142 thực hiện xử lý lọc vòng lặp thích ứng đối với ảnh được cấp từ bộ lọc dịch vị thích ứng 141 đối với từng LCU bằng cách sử dụng hệ số lọc được cấp từ bộ giải mã không tổn hao 102. Bộ lọc vòng lặp thích ứng 142 cấp ảnh thu nhận được như vậy đến bộ nhớ khung 109 và bộ đệm sắp xếp lại màn hình 107.

Ảnh được tích lũy trong bộ nhớ khung 109 được đọc như là ảnh tham chiếu qua chuyên mạch 110 và được cấp đến bộ bù chuyển động 112 hoặc bộ dự báo trong 111.

Bộ đệm sắp xếp lại màn hình 107 lưu giữ ảnh được cấp từ bộ lọc giải khối 106 theo các đơn vị khung. Bộ đệm sắp xếp lại màn hình 107 sắp xếp lại ảnh đã được lưu giữ, mà được tạo thành theo các đơn vị khung, được sắp xếp theo thứ tự mã hóa theo thứ tự hiển thị gốc và cấp ảnh đã được sắp xếp lại đến bộ chuyển

đôi D/A 108.

Bộ chuyển đổi D/A 108 thực hiện chuyển đổi D/A đối với ảnh được tạo thành theo các đơn vị khung được cấp từ bộ đệm sắp xếp lại màn hình 107 và xuất ảnh đã được chuyển đổi đến tầng sau không được minh họa trên hình vẽ làm tín hiệu đầu ra.

Bộ dự báo trong 111 thực hiện xử lý dự báo trong của chế độ dự báo trong được thể hiện bằng thông tin chế độ dự báo trong mà được cấp từ bộ giải mã không tổn hao 102 bằng cách sử dụng ảnh tham chiếu, mà chưa được lọc bởi bộ lọc giải khối 106, được đọc từ bộ nhớ khung 109 qua chuyển mạch 110 theo các đơn vị tám và lát. Bộ dự báo trong 111 cấp ảnh đã được dự báo được tạo ra như vậy đến chuyển mạch 113.

Bộ bù chuyển động 112 đọc ảnh tham chiếu, mà đã được lọc bởi bộ lọc giải khối 106, từ bộ nhớ khung 109 qua chuyển mạch 110 dựa vào thông tin được sử dụng để xác định ảnh tham chiếu mà được cấp từ bộ giải mã không tổn hao 102 theo các đơn vị tám và lát. Bộ bù chuyển động 112 thực hiện xử lý bù chuyển động của chế độ dự báo giữa tối ưu được thể hiện bởi thông tin chế độ dự báo giữa bằng cách sử dụng vector chuyển động và ảnh tham chiếu. Bộ bù chuyển động 112 cấp ảnh đã được dự báo được tạo ra như vậy đến chuyển mạch 113.

Trong trường hợp mà thông tin chế độ dự báo trong được cấp từ bộ giải mã không tổn hao 102, chuyển mạch 113 cấp ảnh đã được dự báo được cấp từ bộ dự báo trong 111 đến bộ cộng 105. Mặt khác, trong trường hợp mà thông tin chế độ dự báo giữa được cấp từ bộ giải mã không tổn hao 102, chuyển mạch 113 cấp ảnh đã được dự báo được cấp từ bộ bù chuyển động 112 đến bộ cộng 105.

Ví dụ về xử lý giải mã tệp

Tiếp theo, xử lý giải mã tệp được thực hiện bởi thiết bị giải mã được minh họa trên Fig.8 sẽ được mô tả có dựa vào lưu đồ được minh họa trên Fig.11. Xử lý giải mã tệp này là xử lý đối với tệp mà được tạo ra bởi xử lý tạo tệp được minh họa trên Fig.4. Nói cách khác, ngay cả khi mã bắt đầu và dữ liệu đệm được loại bỏ, đặc tính của thông số HRD được thiết lập lại trong tệp này.

Ở bước S71, bộ thu tệp 71 thu tệp của MP4 và cấp tệp đã thu được đến bộ thu nhận thông số 72.

Ở bước S72, bộ thu nhận thông số 72 thu nhận moov mà là thông tin quản lý từ tệp của MP4 và cấp moov thu nhận được đến bộ đọc dữ liệu 74. Bộ đọc dữ liệu 74 thu nhận SPS và PPS từ moov ở bước S73. Ở bước S74, bộ thu nhận thông số 72 thu nhận kích thước đơn vị của mdat giải mã hiện thời dựa vào moov thu nhận được ở bước S72.

Ở bước S75, bộ thu nhận thông số 72 dò NAL dựa vào kích thước đơn vị thu nhận được ở bước S74. Lúc này, ví dụ, mã bắt đầu cũng thu nhận được và được tham chiếu theo cấu trúc tệp sẽ được mô tả sau có dựa vào Fig.17 và Fig.18.

Ở bước S76, bộ đọc dữ liệu 74 đọc dữ liệu từ NAL dò được ở bước S75 và cấp dữ liệu đã được đọc đến bộ giải mã 52. Ở đây, trong trường hợp mà NAL là SPS hoặc PPS, SPS hoặc PPS của moov mà thu nhận được ở bước S73 có giá trị mặc định, và SPS hoặc PPS của NAL được sử dụng.

Ở bước S77, bộ giải mã 52 thực hiện xử lý giải mã. Lúc đó, bộ giải mã 52 thực hiện xử lý giải mã dựa vào thông số HRD.

Ở đây, theo sáng chế, tệp được tạo cấu trúc với mã bắt đầu và dữ liệu đệm được thiết lập mà không bị loại bỏ để đặc tính của thông số HRD được duy trì, và do đó, xử lý giải mã có thể được thực hiện đúng.

Ngoài ra, ngay cả khi mã bắt đầu và dữ liệu đệm bị loại bỏ, thông số HRD được thiết lập lại trong xử lý tạo tệp được minh họa trên Fig.4, và do đó, xử lý giải mã đúng có thể được thực hiện.

Ví dụ, trong trường hợp của tệp MP4, có chỉ định tốc độ truyền của VBR hoặc CBR từ bên ngoài. Trong khi luồng được xuất đến bộ giải mã 52 từ bộ đọc dữ liệu 74 một cách bình thường trong trường hợp của VBR, trong trường hợp của CBR, dữ liệu đệm hoặc bit thêm vào được bổ sung để tạo CBR bằng bộ đọc dữ liệu 74, và thông số HRD được ghi lại. Cũng lúc đó, thông số HRD của tệp MP4 là bình thường và có thể được ghi đè để là bình thường cũng ở phía giải mã.

Ví dụ khác về xử lý giải mã tệp

Tiếp theo, xử lý giải mã tệp được thực hiện bởi thiết bị giải mã được minh họa trên Fig.8 sẽ được mô tả có dựa vào lưu đồ được minh họa trên Fig.12. Xử lý giải mã tệp này là xử lý đối với tệp được tạo ra bởi xử lý tạo tệp được minh họa trên Fig.5. Nói cách khác, trong moov của tệp này, thông số nhận dạng nhận dạng việc liệu đặc tính của thông số HRD là đúng hay khác nhau được thiết lập.

Ở bước S91, bộ thu tệp 71 thu tệp của MP4 và cấp tệp đã thu được đến bộ thu nhận thông số 72.

Ở bước S92, bộ thu nhận thông số 72 thu nhận moov mà là thông tin quản lý từ tệp của MP4 và cấp moov thu nhận được đến bộ đọc dữ liệu 74. Bộ đọc dữ liệu 74 thu nhận SPS và PPS từ moov ở bước S93. Ở bước S94, bộ thu nhận thông số 72 thu nhận thông số nhận dạng của HRD từ moov mà thu nhận được ở bước S92 và cấp thông số nhận dạng thu nhận được của HRD đến bộ điều khiển giải mã 73. Ở bước S95, bộ thu nhận thông số 72 thu nhận kích thước đơn vị của mdat giải mã hiện thời dựa vào moov thu nhận được ở bước S92.

Ở bước S96, bộ thu nhận thông số 72 dò NAL dựa vào kích thước đơn vị thu nhận được ở bước S95. Lúc này, ví dụ, mã bắt đầu cũng thu nhận được và được tham chiếu theo cấu trúc tệp sẽ được mô tả sau có dựa vào Fig.17 và Fig.18.

Ở bước S97, bộ đọc dữ liệu 74 đọc dữ liệu từ NAL dò được ở bước S96 và cấp dữ liệu đã được đọc đến bộ giải mã 52. Ở đây, trong trường hợp mà NAL là SPS hoặc PPS, SPS hoặc PPS của moov mà thu nhận được ở bước S93 có giá trị mặc định, và SPS hoặc PPS của NAL được sử dụng.

Ở bước S98, bộ điều khiển giải mã 73 xác định liệu thông số nhận dạng của HRD được cấp từ bộ thu nhận thông số 72 có phải là "1" hay không. Trong trường hợp mà thông số nhận dạng của HRD được xác định là "1" ở bước S98, xử lý chuyển đến bước S99.

Ở bước S99, bộ giải mã 52 thực hiện xử lý giải mã theo thông số HRD theo sự điều khiển của bộ điều khiển giải mã 73.

Mặt khác, trong trường hợp mà thông số nhận dạng của HRD được xác

định không phải là "1" ở bước S98, xử lý chuyển đến bước S100.

Ở bước S100, bộ giải mã 52 thực hiện xử lý giải mã với thông số HRD được bỏ qua theo sự điều khiển của bộ điều khiển giải mã 73

Ở đây, theo sáng chế, tệp được tạo cấu trúc để mã bắt đầu và dữ liệu đệm được thiết lập mà không bị loại bỏ để duy trì đặc tính của thông số HRD, và do đó, xử lý giải mã có thể được thực hiện đúng.

Ngoài ra, ngay cả khi mã bắt đầu và dữ liệu đệm bị loại bỏ, trong xử lý tạo tệp được minh họa trên Fig.5, thông số nhận dạng nhận dạng việc liệu thông số HRD này có được thiết lập đúng hay không, và do đó, xử lý giải mã có thể được thực hiện đúng.

2. Phương án thứ hai

Ví dụ cấu hình của bộ mã hóa

Fig.13 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị mã hóa ảnh 201 theo phương án để tạo tệp MP4 được mô tả trên đây. Như được minh họa trên Fig.13, thiết bị mã hóa ảnh 201 bao gồm: bộ mã hóa 211; bộ đệm VCL 212; bộ đệm phi-VCL 213; bộ tạo tệp 214; và bộ điều khiển 215.

Bộ mã hóa 211 được minh họa trên Fig.13 tương ứng với bộ mã hóa 1 được minh họa trên Fig.1. Ngoài ra, bộ đệm VCL 212, bộ đệm phi-VCL 213, bộ tạo tệp 214, và bộ điều khiển 215 được minh họa trên Fig.13 tương ứng với bộ tạo tệp 2 được minh họa trên Fig.1.

Bộ mã hóa 211 là bộ mã hóa mà hoạt động theo hệ thống HEVC. Bộ mã hóa 211 theo tuần tự thu nhận các ảnh sẽ được mã hóa từ nguồn video như camera hoặc bộ điều hướng truyền hình được nối với thiết bị mã hóa ảnh 201. Sau đó, bộ mã hóa 211 mã hóa các ảnh thu nhận được bằng cách sử dụng các thông số có trong SPS, PPS, và APS, bằng cách đó tạo luồng bit của dữ liệu ảnh. Ngoài ra, bộ mã hóa 211 tạo SPS và PPS làm đơn vị phi-VCL NAL. Mặt khác, bộ mã hóa 211 tạo dữ liệu đệm và luồng bit của dữ liệu ảnh làm đơn vị VCL NAL. Bộ mã hóa 211 xuất dữ liệu đệm và luồng bit của dữ liệu ảnh đến bộ tạo tệp 214 qua bộ đệm VCL 212. Ngoài ra, bộ mã hóa 211 xuất SPS và PPS đến bộ tạo tệp 214 qua bộ đệm phi-VCL 213. Bộ đệm VCL 212 lưu vào bộ đệm đơn vị

VCL NAL. Bộ đệm phi-VCL 213 đơn vị phi-VCL NAL. Bộ tạo tệp 214 tạo tệp HEVC 151 mà lưu giữ chuỗi của dữ liệu ảnh đã được mã hóa. Cụ thể hơn là, bộ tạo tệp 214 chèn dữ liệu đệm và luồng bit của dữ liệu ảnh vào vùng dữ liệu (ví dụ, hộp mdat) của tệp HEVC 151 theo thứ tự giải mã làm đơn vị VCL NAL. Ngoài ra, bộ tạo tệp 214 chèn SPS và PPS vào vùng đoạn đầu (ví dụ, hộp moov) của tệp HEVC 151 làm đơn vị phi-VCL NAL. Bộ điều khiển 215 điều khiển xử lý mã hóa được thực hiện bởi thiết bị mã hóa ảnh 201.

Ngoài ra, bộ điều khiển 215 có thể điều khiển việc tạo luồng đã được mã hóa bằng cách sử dụng mô hình bộ giải mã ảo được gọi là bộ giải mã tham chiếu giả thiết (Hypothetical Reference Decoder, viết tắt là HRD) để bộ đệm của bộ giải mã không thất bại. Trong hệ thống HEVC, để làm các điểm tương thích (các điểm kiểm tra đối với tương thích chuẩn) sẽ được thỏa mãn bởi luồng đã được mã hóa, hai loại điểm tương thích bao gồm loại 1 và loại 2 được định nghĩa. Điểm tương thích của loại 1 được áp dụng cho đơn vị VCL NAL và đơn vị NAL dữ liệu lọc nhưng không được áp dụng cho đơn vị phi-VCL NAL. Điểm tương thích của loại 2 được áp dụng cho đơn vị VCL NAL, đơn vị NAL dữ liệu lọc, và đơn vị phi-VCL NAL. Theo phương án này, dữ liệu đệm được định nghĩa không phải là đơn vị phi-VCL NAL mà là đơn vị VCL NAL. Do vậy, bộ điều khiển 215 có thể thực hiện việc điều khiển tạo luồng đã được mã hóa để không chỉ luồng bit của dữ liệu ảnh mà cả dữ liệu đệm thỏa mãn điểm tương thích của loại 1.

Ví dụ cấu hình của bộ giải mã

Fig.14 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị giải mã ảnh theo phương án để giải mã ảnh từ tệp MP4 được mô tả trên đây. Như được minh họa trên Fig.14, thiết bị giải mã ảnh 251 bao gồm: bộ đệm VCL 261; bộ đệm phi-VCL 262, bộ nhớ thông số 263; bộ giải mã 264; và bộ điều khiển 265.

Ở đây, bộ đệm VCL 261, bộ đệm phi-VCL 262, bộ nhớ thông số 263, và bộ điều khiển 265 được minh họa trên Fig.14 tương ứng với bộ đọc tệp 51 được minh họa trên Fig.8. Ngoài ra, bộ giải mã 264 được minh họa trên Fig.14 tương ứng với bộ giải mã 52 được minh họa trên Fig.8.

Bộ đệm VCL 261 lưu vào bộ đệm luồng bit của dữ liệu ảnh và dữ liệu đệm được đọc từ vùng dữ liệu (ví dụ, hộp mdat) của tệp. Bộ đệm phi-VCL 262

lưu vào bộ đệm các tập thông số được đọc từ vùng đoạn đầu (ví dụ, hộp moov) của tệp. Bộ nhớ thông số 263 lưu giữ các tập thông số được bố trí trong vùng đoạn đầu của tệp thu nhận được qua bộ đệm phi-VCL 262 cùng nhau. Bộ giải mã 264 là bộ giải mã mà hoạt động theo hệ thống HEVC. Bộ giải mã 264 giải mã các ảnh từ các luồng bit mà tuần tự thu nhận được từ vùng dữ liệu của tệp qua bộ đệm VCL 261. Khi ảnh được giải mã, bộ giải mã 264 sử dụng các thông số được bố trí trong SPS và PPS được lưu giữ trong bộ nhớ thông số 263. Bộ điều khiển 265 điều khiển xử lý giải mã được thực hiện bởi thiết bị giải mã ảnh 251.

3. Phương án thứ ba

Cấu trúc mẫu video MP4 thông thường

Fig.15 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 thông thường. Đơn vị NAL mà tạo cấu trúc mẫu video được định nghĩa như được minh họa trên Fig.15.

Trong ví dụ được minh họa trên Fig.15, luồng AVC mà được tạo cấu trúc bởi các đơn vị NAL được thể hiện. Trong luồng AVC, một hình ảnh được tạo cấu trúc bởi một đơn vị truy cập. Để làm các loại đơn vị NAL tạo cấu trúc đơn vị truy cập, có các đơn vị như SPS, PPS, ký hiệu tách đơn vị truy cập, SEI, hình ảnh IDR, hình ảnh phi-IDR, đầu cuối của trình tự, và dữ liệu đệm.

Trong đặc tả MP4 thông thường, một đơn vị truy cập được lưu giữ trong một mẫu video. Nói cách khác, kích thước đơn vị được bổ sung vào từng đơn vị NAL tạo cấu trúc một đơn vị truy cập, và đơn vị NAL thu được được lưu giữ trong một mẫu video. Tuy nhiên, trong trường hợp mà đơn vị NAL của SPS, PPS, hoặc dữ liệu đệm có mặt trong đơn vị NAL tạo cấu trúc đơn vị truy cập, đơn vị NAL của SPS, PPS, hoặc dữ liệu đệm được loại trừ khỏi mẫu video.

Ví dụ, trong khi đơn vị truy cập được bố trí thứ nhất từ phía bên trái được tạo cấu trúc bởi các đơn vị NAL của ký hiệu tách đơn vị truy cập, SPS, PPS, SEI, và hình ảnh IDR, các đơn vị NAL của SPS và PPS được loại trừ khi đơn vị truy cập được lưu giữ trong mẫu.

Trong khi đơn vị truy cập được bố trí thứ hai từ phía bên trái được tạo cấu

trúc bởi các đơn vị NAL của ký hiệu tách đơn vị truy cập, SPS, SEI, hình ảnh phi-IDR, và dữ liệu đệm, các đơn vị NAL của SPS và dữ liệu đệm được loại trừ khi đơn vị truy cập được lưu giữ trong mẫu.

Trong khi đơn vị truy cập được bố trí thứ ba từ phía bên trái được tạo cấu trúc bởi các đơn vị NAL của ký hiệu tách đơn vị truy cập, SPS, SEI, và hình ảnh phi-IDR, đơn vị NAL của SPS được loại trừ khi đơn vị truy cập được lưu giữ trong mẫu.

Ví dụ cấu hình của một đơn vị truy cập của luồng AVC

Fig.16 là sơ đồ minh họa ví dụ về cấu hình của một đơn vị truy cập của luồng AVC.

Như được minh họa trên Fig.16, trên thực tế, bộ nhận dạng (mã bắt đầu) được bổ sung trước từng đơn vị NAL trong một đơn vị truy cập của luồng AVC, thông thường, các bộ nhận dạng này cũng được loại trừ khi đơn vị truy cập được lưu giữ trong mẫu.

Khi các đơn vị NAL của bộ nhận dạng (mã bắt đầu) này và dữ liệu đệm được loại trừ, như được mô tả trên đây, đặc tính của thông số HRD sẽ không được duy trì. Do vậy, theo sáng chế, đơn vị truy cập được lưu giữ trong mẫu của MP4 một cách bình thường.

Ví dụ về lưu giữ đơn vị truy cập theo sáng chế

Fig.17 và Fig.18 là các sơ đồ minh họa các ví dụ của cấu trúc mẫu video MP4 theo sáng chế.

Trong ví dụ A của Fig.17, kích thước đơn vị mô tả kích thước của đơn vị truy cập được bổ sung vào phần bắt đầu của đơn vị truy cập, và đơn vị truy cập được lưu giữ trong mẫu MP4 một cách bình thường. Ở đây, "một cách bình thường" thể hiện trạng thái trong đó mã bắt đầu được gán, và, trong trường hợp mà dữ liệu đệm có mặt, dữ liệu đệm sẽ không bị loại bỏ. Nói cách khác, "một cách bình thường" thể hiện trạng thái trong đó mã bắt đầu và dữ liệu đệm được thiết lập.

Trong trường hợp A của Fig.17, ở bước S75 được thể hiện trên Fig.11 hoặc bước S96 được thể hiện trên Fig.12 được mô tả trên đây, đơn vị truy cập

tương ứng với kích thước đơn vị dò được, và NAL dò được dựa vào mã bắt đầu có trong đơn vị truy cập. Sau đó, ở bước S76 được thể hiện trên Fig.11 hoặc bước S97 được thể hiện trên Fig.12, dữ liệu của NAL cho đến mã bắt đầu tiếp theo thu nhận được. Trường hợp A của Fig.17 thể hiện cấu trúc rất đơn giản.

Trong trường hợp B của Fig.17, kích thước đơn vị mô tả kích thước của từng đơn vị NAL bao gồm mã bắt đầu được bổ sung vào phần bắt đầu của từng đơn vị NAL tạo cấu trúc đơn vị truy cập, và đơn vị truy cập được lưu giữ trong mẫu MP4 một cách bình thường.

Trong trường hợp B của Fig.17, ở bước S75 được thể hiện trên Fig.11 hoặc bước S96 được thể hiện trên Fig.12 được mô tả trên đây, mã bắt đầu + NAL tương ứng với kích thước đơn vị dò được. Sau đó, ở bước S76 được thể hiện trên Fig.11 hoặc bước S97 được thể hiện trên Fig.12, trong số chúng, mã bắt đầu được loại bỏ, và dữ liệu của NAL thu nhận được. Trong trường hợp này, trong khi cấu trúc gần cấu trúc thông thường của "kích thước + dữ liệu", và việc nhúng là giống nhau, cần phân tích mã bắt đầu dựa vào cấu trúc dữ liệu thu nhận được.

Trong trường hợp C của Fig.18, kích thước đơn vị mô tả kích thước của từng đơn vị NAL được bổ sung vào phần bắt đầu của từng đơn vị NAL tạo cấu trúc đơn vị truy cập, và kích thước của mã bắt đầu bao gồm dữ liệu 0, nói cách khác, kích thước dịch vị cho đến NAL được bổ sung. Sau đó, đơn vị NAL mà kích thước đơn vị và kích thước dịch vị được bổ sung vào được lưu giữ một cách bình thường trong mẫu MP4.

Trong trường hợp C của Fig.18, ở bước S75 được thể hiện trên Fig.11 hoặc bước S96 được thể hiện trên Fig.12 được mô tả trên đây, kích thước dịch vị thu nhận được, và NAL dò được từ vị trí thu nhận được bằng cách thực hiện bỏ cách tương ứng với dịch vị. Sau đó, ở bước S76 được thể hiện trên Fig.11 hoặc bước S97 được thể hiện trên Fig.12, dữ liệu của NAL tương ứng với kích thước đơn vị thu nhận được.

Ngoài ra, C của Fig.18 là ví dụ về trường hợp mà FixedStartCodeSizeFlag của hộp tương thích HRD (HRD Conformance Box) được minh họa trên Fig.19 sẽ được mô tả sau là "0". Trong trường hợp mà FixedStartCodeSizeFlag là "0",

vì kích thước mã bắt đầu không được cố định, kích thước của mã bắt đầu được lưu giữ trong trường kích thước dịch vị.

Ngược lại, trong trường hợp mà FixedStartCodeSizeFlag của hộp tương thích HRD là "1", mẫu MP4 có cấu trúc được minh họa trong C' của Fig.18.

Trong ví dụ của C' được minh họa trên Fig.18, kích thước đơn vị mô tả kích thước của từng đơn vị NAL được bổ sung vào phần bắt đầu của từng đơn vị NAL mà tạo cấu trúc đơn vị truy cập, và đơn vị NAL được lưu giữ trong mẫu MP4 một cách bình thường.

Nói cách khác, trong trường hợp mà FixedStartCodeSizeFlag là "1", vì kích thước mã bắt đầu được cố định làm kích thước mã bắt đầu = 4 byte, trường kích thước dịch vị không cần được bố trí trước mã bắt đầu.

Trong trường hợp này, cấu trúc phân tách được tạo ra.

Trong trường hợp của D của Fig.18, kích thước đơn vị mô tả kích thước của từng đơn vị NAL, kích thước của mã bắt đầu, và mã bắt đầu được bổ sung vào phần bắt đầu của từng đơn vị NAL mà tạo cấu trúc đơn vị truy cập, và kích thước của mã bắt đầu bao gồm 0 dữ liệu, nói cách khác, kích thước dịch vị cho đến NAL được bổ sung. Sau đó, đơn vị NAL mà kích thước đơn vị và kích thước dịch vị được bổ sung vào được lưu giữ trong mẫu MP4 một cách bình thường.

Trong trường hợp của D của Fig.18, ở bước S75 được thể hiện trên Fig.11 hoặc bước S96 được thể hiện trên Fig.12 được mô tả trên đây, "kích thước dịch vị + mã bắt đầu + NAL" tương ứng với kích thước đơn vị dò được, kích thước dịch vị thu nhận được, và NAL dò được từ vị trí thu nhận được bằng cách thực hiện bỏ qua tương ứng với dịch vị. Sau đó, ở bước S76 được thể hiện trên Fig.11 hoặc bước S97 được thể hiện trên Fig.12, dữ liệu của NAL tương ứng với "kích thước đơn vị – (kích thước dịch vị + mã bắt đầu)" thu nhận được.

Ngoài ra, D của Fig.18 là ví dụ về trường hợp mà FixedStartCodeSizeFlag của hộp tương thích HRD là "0". Trong trường hợp mà FixedStartCodeSizeFlag là "0", vì kích thước mã bắt đầu không được cố định, kích thước của mã bắt đầu được lưu giữ trong trường kích thước dịch vị.

Ngược lại, trong trường hợp mà FixedStartCodeSizeFlag của hộp tương thích HRD là "1", mẫu MP4 có cấu trúc được minh họa trong D' của Fig.18.

Trong ví dụ của D' được minh họa trên Fig.18, kích thước đơn vị mô tả kích thước của từng đơn vị NAL bao gồm mã bắt đầu được bổ sung vào phần bắt đầu của từng đơn vị NAL mà tạo cấu trúc đơn vị truy cập, và đơn vị NAL được lưu giữ trong mẫu MP4 một cách bình thường.

Nói cách khác, trong trường hợp mà FixedStartCodeSizeFlag là "1", vì kích thước mã bắt đầu được cố định làm kích thước mã bắt đầu = 4 byte, trường kích thước dịch vị không cần được bố trí trước mã bắt đầu, và cấu trúc của D' của Fig.18 tương tự như của B của Fig.17 được mô tả trên đây.

Trong trường hợp này, trong khi việc nhúng giống như dựa vào cấu trúc thông thường của "kích thước + dữ liệu", cấu trúc phân tách được tạo ra trong đó việc bỏ qua tương ứng với kích thước dịch vị được bố trí tại phần bắt đầu của cấu trúc dữ liệu thu nhận được được thực hiện. Ngoài ra, trong trường hợp này, cần phân tích mã bắt đầu.

Ví dụ về sự mở rộng hộp tương thích HRD

Fig.19 là sơ đồ minh họa ví dụ về sự mở rộng hộp tương thích HRD. Dưới đây, ví dụ về sự mở rộng hộp tương thích HRD sẽ được minh họa một cách tương tự.

```
AVCSampleEntry() extends VisualSampleEntry('type') {
// type is 'avc3' or 'avc4', (và all hevc)
AVCConfigurationBox config;
MPEG4BitRateBox(); //Optional
MPEG4ExtensionDescriptorsBox(); //Optional
HRDConformanceBox(); //Optional
}
```

Thêm HRDConformanceBox vào AVCSampleEntry và AVC2SVVSampleEntry trong 5.4.2.1

Thêm HRDConformanceBox vào AVCSVCSampleEntry, AVC2SVCSampleEntry và SVCSampleEntry trong 6.5.3.1.

Thêm HRDConformanceBox vào AVCMVCSampleEntry, AVC2MVCSampleEntry và MVCSampleEntry trong mục 7.6.3.3.

Thêm HRDConformanceBox vào HEVC SampleEntry trong mục 8.4.1.1.

Mục nhập mẫu mà không có HRDConformanceBox chỉ được sử dụng cho các mẫu mà không có các mã bắt đầu hay đệm.

HRDConformanceBox có thể được định nghĩa như sau.

```
class HRDConformanceBox extends Box('hrdc') {
    HRDConformanceData() HRDConformance;
}

aligned(8) class HRDConformanceData {
    unsigned int(1) HRDConformanceFlag;
    unsigned int(1) FixedStartCodeSizeFlag;
    unsigned int(2) OffsetSizeMinusOne;
    unsigned int(4) reserved = 0;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}
```

HRDConformanceFlag là bộ nhận dạng nếu các đặc tính HRD được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., từ luồng thứ cấp. Khi HRDConformanceFlag bằng 1, các thông số HRD, ví dụ, khoảng đệm SEI, SEI thời gian hình ảnh, được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., từ luồng thứ cấp.

FixedStartCodeSizeFlag là bộ nhận dạng khi mã bắt đầu trong từng NAL là kích thước cố định 4 byte. Khi FixedStartCodeSizeFlag bằng 1, không có trường dịch vị cho từng NAL trong dữ liệu mẫu để biểu thị kích thước của mã

bắt đầu.

OffsetSizeMinusOne biểu thị độ dài tính theo byte của trường StartcodeLength trong mẫu video trừ một. Ví dụ, kích thước của một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

HRDConformanceBox là hữu dụng để biểu thị trạng thái của tương thích HRD ngay cả trong các tùy chọn khác.

Trên Fig.19 và ví dụ được mô tả trên đây, FixedStartCodeSizeFlag mà là cờ nhận dạng việc liệu thông tin kích thước của mã bắt đầu có mặt có được định nghĩa bổ sung trong hộp tương thích HRD hay không. Theo cách này, như được minh họa trong C' của Fig.18 và D' của Fig.18, trong trường hợp mà kích thước của mã bắt đầu được cố định, cấu trúc tệp có thể là đơn giản.

Ví dụ về lưu giữ đơn vị truy cập theo sáng chế

Fig.20 là sơ đồ minh họa ví dụ khác về cấu trúc mẫu video MP4 theo sáng chế.

Trong ví dụ được minh họa trong E của Fig.20, kích thước đơn vị được loại bỏ khỏi mdat, và luồng sơ cấp trong đó mã bắt đầu được đưa vào được lưu giữ (dự phòng) mà không thay đổi.

Trong trường hợp của E của Fig.20, hộp mới được sử dụng để duy trì dịch vị của từng đơn vị truy cập, như được minh họa trên Fig.21 và như dưới đây, được bổ sung vào bảng mẫu.

```
aligned(8) class NALunitOffsetBox
extends FullBox('nalo', version=0, 0) {
    unsigned int(32) sample_count;
    for (i=1; i <= sample_count; i++){
        unsigned int(8) offset_count;
        unsigned int(8) offset_size;
        for (j=1; j <= offset_count; j++){
```

```

unsigned int(offset_size*8) offset;

    }

}

}

```

sample_count là số nguyên mà đưa ra số lượng mục trong bảng sau.

offset_count là số nguyên mà đưa ra số lượng mục trong bảng con sau.

offset_size là số nguyên mà đưa ra kích thước byte của trường dịch vị trong bảng con sau.

offset là số nguyên mà đưa ra dịch vị của phần bắt đầu của đơn vị NAL vào mẫu chứa của nó.

Ngoài ra, để thay thế cho ví dụ được thể hiện trong E của Fig.20, như được minh họa trong ví dụ của E' của Fig.22, danh sách của các dịch vị có thể được tạo thành để được đưa vào tại phần bắt đầu của từng mẫu thay vì việc bổ sung hộp được mô tả trên đây.

Trong ví dụ được thể hiện trong E' của Fig.22, ví dụ, danh sách của năm dịch vị của năm đơn vị truy cập được đưa vào tại phần bắt đầu của mẫu.

Các tệp của các mẫu video MP4 được minh họa trên Fig.17 và Fig.18 được tạo ra bởi bộ tạo tệp 2 có cấu trúc được minh họa trên Fig.3 được mô tả trên đây qua xử lý tạo tệp được minh họa trên Fig.4 hoặc Fig.5 được mô tả trên đây. Ngoài ra, các tệp mẫu video MP4 được minh họa trên Fig.20 và Fig.22 được tạo ra bởi bộ tạo tệp 2 có cấu trúc được minh họa trên Fig.3 được mô tả trên đây qua xử lý tạo tệp được minh họa trên Fig.4 hoặc Fig.5 được mô tả trên đây.

Hơn thế nữa, các tệp của các mẫu video MP4 được minh họa trên Fig.17 và Fig.18 được giải mã bởi bộ đọc tệp 51 có cấu trúc được minh họa trên Fig.9 được mô tả trên đây qua xử lý giải mã tệp được minh họa trên Fig.11 hoặc Fig.12 được mô tả trên đây. Ngoài ra, các tệp của các tệp mẫu video MP4 được minh họa trên Fig.20 và Fig.22 được giải mã bởi bộ đọc tệp 51 có cấu trúc được minh họa trên Fig.9 được mô tả trên đây qua xử lý giải mã tệp được minh họa

trên Fig.11 hoặc Fig.12 được mô tả trên đây.

Ví dụ khác nữa về lưu giữ đơn vị truy cập theo sáng chế

Fig.23 là sơ đồ minh họa ví dụ khác nữa về cấu trúc mẫu video MP4 theo sáng chế.

Trong ví dụ được minh họa trong F của Fig.23, kích thước đơn vị được loại bỏ khỏi mdat, và luồng thứ cấp trong đó mã bắt đầu được đưa vào được lưu giữ (dự phòng) một cách bình thường mà không thay đổi bất kỳ. Ngoài ra, để lưu giữ các độ dài của từng mã bắt đầu (startcode) và đơn vị NAL, hàm thông tin bổ sung mẫu được định nghĩa trong ISO/IEC 14496-12 được áp dụng.

Đối với các độ dài của từng mã bắt đầu và đơn vị NAL, `aux_info_type = "nalz"` và `aux_info_type_parameter = 0` được đưa ra làm hàm thông tin bổ sung mẫu. Hàm thông tin bổ sung mẫu được tham chiếu bởi "saiz" mà là hộp kích thước hàm thông tin bổ sung mẫu (`SampleAuxiliaryInformationSizesBox`) và "saio" mà là hộp dịch vị hàm thông tin bổ sung mẫu.

Ví dụ về định dạng của hàm thông tin bổ sung mẫu

Fig.24 là sơ đồ minh họa ví dụ về định dạng của hàm thông tin bổ sung mẫu. Dưới đây, ví dụ về định dạng của hàm thông tin bổ sung mẫu sẽ được minh họa một cách tương tự.

```
aligned(8) class NalsizeSampleAuxiliaryDataFormat
{
    unsigned int(2) StartcodeSizeMinusOne;
    unsigned int(2) NALunitSizeMinusOne;
    unsigned int(4) reserved;
    unsigned int(32) subsample_count;
    for (i = 1; i <= subsample_count; i++){
        switch (StartcodeSizeMinusOne){
        case 0:
            unsigned int(8) Startcode Size;
```

```

break;

case 1:
    unsigned int(16) Startcode Size;
    break;

case 3:
    unsigned int(32) Startcode Size;
    break;
}

switch (NALUnitSizeMinusOne){
case 0:
    unsigned int(8) NALUnit Size;
    break;

case 1:
    unsigned int(16) NALUnit Size;
    break;

case 3:
    unsigned int(32) NALUnit Size;
    break;
}
}
}

```

StartcodeSizeMinusOne biểu thị độ dài tính theo byte của trường kích thước mã bắt đầu (Startcode Size) trừ một. Ví dụ, kích thước một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

NALUnitSizeMinusOne biểu thị độ dài tính theo byte của trường kích

thước đơn vị NAL (NALUnit Size) trừ một. Ví dụ, kích thước một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

StartcodeSize biểu thị độ dài tính theo byte của Startcode của mẫu con (subsample) trong mẫuNALUnitSize biểu thị độ dài tính theo byte của Startcode của mẫu con trong mẫu

StartcodeSizeMinusOne biểu thị giá trị thu nhận được bằng cách trừ một từ độ dài byte của trường kích thước mã bắt đầu (Startcode Size field). Ví dụ, trong trường hợp mà độ dài byte của trường kích thước mã bắt đầu là một byte, giá trị của StartcodeSizeMinusOne là "0". Giá trị của StartcodeSizeMinusOne là 0, 1, hoặc 3 tương ứng với độ dài mã của một byte, hai byte, hoặc bốn byte.

NALUnitSizeMinusOne biểu thị giá trị mà thu nhận được bằng cách trừ một từ độ dài byte của trường kích thước đơn vị NAL (NALUnit Size field). Ví dụ, trong trường hợp mà độ dài byte của trường kích thước đơn vị NAL là một byte, giá trị của NALUnitSizeMinusOne là "0". Giá trị của NALUnitSizeMinusOne là 0, 1, hoặc 3 tương ứng với độ dài mã của một byte, hai byte, hoặc bốn byte.

StartcodeSize biểu thị độ dài byte của mã bắt đầu (Startcode) của mẫu con trong mẫu.

NALUnitSize biểu thị độ dài byte của đơn vị NAL của mẫu con trong mẫu.

Định dạng có thể được tạo thành như được mô tả trên đây. Ngoài ra, định dạng của hàm thông tin bổ sung mẫu có thể được tạo thành như dưới đây.

Ví dụ khác về định dạng của hàm thông tin bổ sung mẫu

Fig.25 là sơ đồ minh họa ví dụ khác về định dạng của hàm thông tin bổ sung mẫu. Dưới đây, ví dụ khác về định dạng của hàm thông tin bổ sung mẫu sẽ được minh họa một cách tương tự.

```
aligned(8) class NalsizeSampleAuxiliaryDataFormat
{
```

```

unsigned int(1) FixedStartCodeSizeFlag;
unsigned int(2) StartcodeSizeMinusOne;
unsigned int(2) NALunitSizeMinusOne;
unsigned int(3) reserved;
unsigned int(32) subsample_count;
for (i = 1; i <= subsample_count; i++){
if (FixedStartCodeSizeFlag == 0){
switch (StartcodeSizeMinusOne){
case 0:
unsigned int(8) Startcode Size;
break;
case 1:
unsigned int(16) Startcode Size;
break;
case 3:
unsigned int(32) Startcode Size;
break;
}
}
switch (NALUnitSizeMinusOne){
case 0:
unsigned int(8) NALUnit Size;
break;
case 1:
unsigned int(16) NALUnit Size;

```

```

break;

case 3:

unsigned int(32) NALUnit Size;

break;

    }

}

}

```

FixedStartCodeSizeFlag là bộ nhận dạng khi mã bắt đầu trước từng đơn vị NAL có kích thước cố định 4 byte. Khi FixedStartCodeSizeFlag bằng 1.

StartcodeSizeMinusOne biểu thị độ dài tính theo byte của trường kích thước mã bắt đầu (Startcode Size) trừ một. Ví dụ, kích thước của một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

NALUnitSizeMinusOne biểu thị độ dài tính theo byte của trường kích thước đơn vị NAL (NALUnit Size) trừ một. Ví dụ, kích thước của một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

StartcodeSize biểu thị độ dài tính theo byte của Startcode của mẫu con trong mẫu.

NALUnitSize biểu thị độ dài tính theo byte của Startcode của mẫu con trong mẫu.

FixedStartCodeSizeFlag là thông tin cờ. FixedStartCodeSizeFlag là thông tin cờ mà biểu thị việc liệu kích thước của mã bắt đầu (mã bắt đầu) được bố trí trước từng đơn vị NAL có phải là bốn byte hay không. Giá trị của FixedStartCodeSizeFlag được thiết lập là "1" trong trường hợp mà kích thước của mã bắt đầu (mã bắt đầu) được bố trí trước từng đơn vị NAL là bốn byte và được thiết lập là "0" theo cách khác. Nói cách khác, trong trường hợp mà giá trị của FixedStartCodeSizeFlag là "1", nó biểu thị là kích thước của mã bắt đầu (mã bắt đầu) được bố trí trước từng đơn vị NAL là bốn byte. Mặt khác, trong trường

hợp mà giá trị của FixedStartCodeSizeFlag là "0", nó biểu thị là kích thước của mã bắt đầu (mã bắt đầu) được bố trí trước từng đơn vị NAL không phải là bốn byte.

Từng thông số của StartcodeSizeMinusOne, NALUnitSizeMinusOne, StartcodeSize, và NALUnitSize tương tự như của trường hợp được minh họa trên Fig.24.

Trong trường hợp mà FixedStartCodeSizeFlag của hộp HRDConformanceData là "1", khi kích thước của từng mã bắt đầu được đưa vào, có khả năng là lượng dữ liệu tăng không cần thiết. Do vậy, bằng cách tạo cấu trúc như trong ví dụ được minh họa trên Fig.25, trong trường hợp này, cấu trúc có kích thước của chỉ các đơn vị NAL có thể được tạo ra, bằng cách đó việc tăng không cần thiết lượng dữ liệu có thể được loại bỏ. Ngoài ra, trong trường hợp này, HRDConformanceBox chỉ có HRDConformanceFlag.

Ngoài ra, tệp của mẫu video MP4 được minh họa trên Fig.23 được tạo ra bởi bộ tạo tệp 2 có cấu trúc được minh họa trên Fig.3 được mô tả trên đây qua xử lý tạo tệp được minh họa trên Fig.4 hoặc Fig.5.

Hơn thế nữa, tệp của mẫu video MP4 được minh họa trên Fig.23 được giải mã bởi bộ đọc tệp 51 có cấu trúc được minh họa trên Fig.9 được mô tả trên đây qua xử lý giải mã tệp được minh họa trên Fig.11 hoặc 12 được mô tả trên đây.

Trong tùy chọn này, kích thước đơn vị được loại bỏ khỏi mdat, và luồng sơ cấp bao gồm mã bắt đầu được lưu giữ đơn thuần mà không cải biến. Và để lưu giữ độ dài của từng mã bắt đầu và đơn vị NAL, đề xuất sử dụng hàm thông tin bổ sung mẫu được định nghĩa trong ISO/IEC 14496-12. Fig.23 thể hiện ví dụ về mẫu của tùy chọn này.

Từng độ dài của mã bắt đầu và đơn vị NAL được bố trí như là thông tin bổ sung mẫu (Sample Auxiliary Information) với aux_info_type bằng 'nalz' và aux_info_type_parameter bằng 0. Thông tin bổ sung mẫu được tham chiếu bằng cách sử dụng SampleAuxiliaryInformationSizesBox ('saiz') và SampleAuxiliaryInformationOffsetsBox ('saio').

Định dạng của thông tin bổ sung mẫu cho các mẫu loại này sẽ là:

```

aligned(8) class NalsizeSampleAuxiliaryDataFormat
{
    unsigned int(1) FixedStartCodeSizeFlag;
    unsigned int(2) StartcodeSizeMinusOne;
    unsigned int(2) NALunitSizeMinusOne;
    unsigned int(3) reserved;
    unsigned int(32) subsample_count;
    for (i = 1; i <= subsample_count; i++){
        if (FixedStartCodeSizeFlag == 0){
            switch (StartcodeSizeMinusOne){
                case 0:
                    unsigned int(8) Startcode Size;
                    break;
                case 1:
                    unsigned int(16) Startcode Size;
                    break;
                case 3:
                    unsigned int(32) Startcode Size;
                    break;
            }
        }
        switch (NALUnitSizeMinusOne){
            case 0:
                unsigned int(8) NALUnit Size;
                break;

```

```

case 1:
    unsigned int(16) NALUnit Size;
    break;
case 3:
    unsigned int(32) NALUnit Size;
    break;
    }
    }
}

```

FixedStartCodeSizeFlag là bộ nhận dạng khi mã bắt đầu trước từng đơn vị NAL có kích thước cố định 4 byte. Khi FixedStartCodeSizeFlag bằng 1.

StartcodeSizeMinusOne biểu thị độ dài tính theo byte của trường kích thước mã bắt đầu (Startcode Size) trừ một. Ví dụ, kích thước một byte được biểu thị với giá trị 0. Giá trị của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

NALUnitSizeMinusOne biểu thị độ dài tính theo byte của trường kích thước đơn vị NAL (NALUnit Size) trừ một. Ví dụ, kích thước một byte được biểu thị với giá trị 0. Giá trị này của trường này sẽ là một trong số 0, 1, hoặc 3 tương ứng với độ dài được mã hóa lần lượt với 1, 2, hoặc 4 byte.

StartcodeSize biểu thị độ dài tính theo byte của Startcode của mẫu con trong mẫu

NALUnitSize biểu thị độ dài tính theo byte của đơn vị NAL của mẫu con trong mẫu

So sánh cấu trúc mẫu video MP4 theo sáng chế

Trong ví dụ được minh họa trên Fig.26, bảng được sử dụng để so sánh các cấu trúc mẫu video MP4 được mô tả trên đây được thể hiện.

Trong ví dụ về lưu giữ đơn vị truy cập được minh họa trong B của Fig.17, có ưu điểm gần cấu trúc thông thường (hiện được sử dụng) của "kích thước + dữ

liệu".

Trong ví dụ lưu giữ đơn vị truy cập được minh họa trong C của Fig.18 hoặc C' của Fig.18, có các ưu điểm gần cấu trúc thông thường (hiện được sử dụng) của "kích thước + dữ liệu" và không thay đổi ngữ nghĩa của kích thước đơn vị. Ngoài ra, trong trường hợp mà kích thước của mã bắt đầu được cố định, có ưu điểm là cấu trúc dữ liệu tương tự như thiết kế định dạng tệp thông thường (hiện được sử dụng).

Trong ví dụ về lưu giữ đơn vị truy cập được minh họa trong D của Fig.18 hoặc D' của Fig.18, có ưu điểm gần cấu trúc thông thường (hiện được sử dụng) của "kích thước + dữ liệu".

Trong ví dụ về lưu giữ đơn vị truy cập được minh họa trong E của Fig.20 hoặc E' của Fig.22, có ưu điểm là việc lưu giữ đơn vị truy cập trong luồng thứ cấp là đơn giản (nói cách khác, mã bắt đầu không cần được loại bỏ, và kích thước đơn vị và kích thước mã bắt đầu không cần được bổ sung).

Trong trường hợp mà độ nhất quán thiết kế của mẫu là yếu tố quan trọng, C của Fig.18 hoặc C' của Fig.18 là tùy chọn tốt nhất. Tuy nhiên, E của Fig.20 hoặc E' của Fig.22 là phương pháp tốt nhất để lưu giữ luồng thứ cấp mà không thay đổi bất kỳ.

Tất cả các ví dụ mô tả trên đây về lưu giữ đơn vị truy cập đã được đề xuất để lưu giữ mã bắt đầu trong mdat (trong đóng góp này, các tùy chọn dưới đây được đề xuất để lưu giữ mã bắt đầu trong mdat (cho cả định dạng tệp AVC và HEVC)).

Ngoài ra, cấu trúc mẫu video MP4 được minh họa trên Fig.23 được mô tả trên đây có thể được tạo thành như được minh họa trong Fig.27 dưới đây.

Ví dụ khác về lưu giữ đơn vị truy cập theo sáng chế

Fig.27 là sơ đồ minh họa ví dụ khác về cấu trúc mẫu video MP4 được minh họa trên Fig.23.

Trong ví dụ của G của Fig.27, tương tự như ví dụ được minh họa trong F của Fig.23, kích thước đơn vị được loại bỏ khỏi mdat, và luồng thứ cấp trong đó mã bắt đầu được đưa vào được lưu giữ (dự phòng) một cách bình thường mà

không thay đổi bất kỳ. Tuy nhiên, khác với ví dụ được minh họa trong F của Fig.23, trong luồng thứ cấp, từng mã bắt đầu (startcode) được đưa vào đơn vị NAL, và để lưu giữ độ dài của đơn vị NAL bao gồm mã bắt đầu, hàm thông tin bổ sung mẫu được định nghĩa trong ISO/IEC 14496-12 được áp dụng.

Đối với độ dài của từng đơn vị NAL, `aux_info_type = "nalz"` và `aux_info_type_parameter = 0` được đưa ra làm hàm thông tin bổ sung mẫu. Hàm thông tin bổ sung mẫu được tham chiếu bởi "saiz" mà là hộp kích thước hàm thông tin bổ sung mẫu (SampleAuxiliaryInformationSizesBox) được sử dụng và "saio" mà là hộp dịch vị hàm thông tin bổ sung mẫu.

Nói cách khác, trong ví dụ được minh họa trong G của Fig.27, dữ liệu hàm thông tin bổ sung mẫu, độ dài của mã bắt đầu không được lưu giữ, và chỉ độ dài của đơn vị NAL bao gồm mã bắt đầu được lưu giữ. Do đó, ví dụ của G của Fig.27 giống như ví dụ được minh họa trong F của Fig.23 ngoại trừ độ dài của mã bắt đầu không được lưu giữ.

Tập của mẫu video MP4 được minh họa trên Fig.27 được tạo ra bởi bộ tạo tập 2 có cấu trúc được minh họa trên Fig.3 được mô tả trên đây qua xử lý tạo tập được minh họa trên Fig.4 hoặc Fig.5 được mô tả trên đây.

Ngoài ra, tập của mẫu video MP4 được minh họa trên Fig.27 được giải mã bởi bộ đọc tập 51 có cấu trúc được minh họa trên Fig.9 được mô tả trên đây qua xử lý giải mã tập được minh họa trên Fig.11 hoặc 12 được mô tả trên đây.

4. Phương án thứ tư

Ở đây, thông thường, khi luồng sơ cấp của MP4 cần được lưu giữ trong MPEG2-PS, cần trích kích thước đơn vị và chèn mã bắt đầu. Ngược lại, khi luồng sơ cấp của MPEG2-PS cần được lưu giữ trong MP4, cần trích mã bắt đầu và chèn kích thước đơn vị. Nói cách khác, ít nhất, hai loại luồng có mặt, và khi luồng được tạo ra như là tập, luồng cần được thay đổi, và cờ thể hiện tương thích/không tương thích với tương thích HRD là cần thiết.

Ngoài ra, trong trường hợp mà có các thay đổi khác nhau trong luồng theo việc có mặt/không có mặt của mã bắt đầu, có quan ngại là độ tương thích có thể bị mất.

Ngược lại, định dạng luồng byte, là luồng bao gồm mã bắt đầu, sẽ được lưu giữ trong tệp của MPEG2-PS được định nghĩa. Bằng cách cho phép luồng được lưu giữ trong tệp MP4, độ tương thích giữa hai tệp sẽ được tăng.

Theo sáng chế, trong mẫu, để nhận dạng việc liệu luồng byte có mặt hay không, hộp thông tin định dạng luồng byte được sử dụng để lưu giữ thông tin của định dạng luồng byte được định nghĩa.

Hộp thông tin định dạng luồng byte

Tiếp theo, hộp thông tin định dạng luồng byte sẽ được mô tả có dựa vào Fig.28.

Hộp thông tin định dạng luồng byte được định nghĩa là tùy chọn cho mục nhập mẫu mà tập thông số hoặc tương tự có thể được chèn vào. Mục nhập mẫu mà tập thông số hoặc tương tự có thể được chèn vào có thể được lưu giữ cũng như là luồng byte. Ngoài ra, lúc đó, để đưa ra tín hiệu là mục nhập mẫu là luồng byte, hộp này sẽ được lưu giữ.

Trong ví dụ được minh họa trên Fig.28, đặc tả của hộp thông tin định dạng luồng byte được minh họa.

Hộp thông tin định dạng luồng bit (Byte Stream Format Information Box) có thể được lưu giữ trong mục nhập mẫu 'avc3', 'avc4', 'mvc3', 'mvc4' hoặc 'hev1' (trong hộp mục nhập mẫu của rãnh phương tiện).

Hộp này sẽ được lưu giữ khi dữ liệu mẫu được tạo cấu trúc bởi định dạng luồng bit (Byte Stream Format) như được định nghĩa trong ISO/IEC 14496-10.

Hộp thông tin mẫu con (Sub-Sample Information Box) có thể được sử dụng để lưu giữ một hoặc nhiều kích thước đơn vị NAL luồng byte (Byte stream NAL unit size) liên kế.

Mẫu con (sub-sample) được định nghĩa là một hoặc nhiều kích thước đơn vị NAL luồng byte liên kế như được định nghĩa trong ISO/IEC 14496-10.

Loại hộp (Box Type): 'bsfi'

Mục (Container): Mục nhập mẫu (Sample Entry) ('avc3', 'avc4', 'mvc3', 'mvc4' hoặc 'hev1')

Mục thiết yếu (mandatory): Không

Số lượng (quantity): Không hoặc một

```
Class ByteStreamFormatInformationBox extends Box('bsfi') {
    ByteStreamFormatInformation() ByteStreamInformation;
}
```

Hộp thông tin định dạng luồng byte được lưu giữ (chứa) trong mục nhập mẫu của "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1" trong hộp mục nhập mẫu của rãnh phương tiện. Mục nhập mẫu của "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1" là mục nhập mẫu mà tập thông số hoặc tương tự có thể được chèn.

Hộp này được lưu giữ (chứa) khi dữ liệu mẫu được tạo cấu trúc trong định dạng luồng byte được định nghĩa trong ISO/IEC 14496-10.

Hộp thông tin mẫu con có thể được sử dụng để lưu giữ kích thước đơn vị NAL bao gồm một hoặc nhiều mã bắt đầu liên tiếp.

Mẫu con được định nghĩa là một hoặc nhiều đơn vị NAL luồng byte liên tiếp được định nghĩa trong ISO/IEC 14496-10.

Loại hộp (Box Type) là "bsfi", và mục (Container) có trong đó là mục nhập mẫu của "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1". Không có mục thiết yếu (mandatory), và khối lượng (quantity) là "0" hoặc "1".

Hộp thông tin định dạng luồng byte mở rộng hộp của "bsfi" trong đó thông tin định dạng luồng byte được mô tả.

Trong hộp thông tin định dạng luồng byte có cấu trúc như được mô tả trên đây, như được minh họa trên Fig.29, bằng cách chỉ chèn mục nhập mẫu của "avc5", "avc6", "mvc5", "mvc6" hoặc "hevc2", mà là các định dạng mới, mục nhập mẫu có thể được mở rộng.

Fig.29 là sơ đồ minh họa ví dụ khác về hộp thông tin định dạng luồng byte.

Hộp thông tin định dạng luồng bit (Byte Stream Format Information Box) sẽ được lưu giữ trong mục nhập mẫu 'avc5', 'avc6', 'mvc5', 'mvc6' hoặc 'hev2'

(trong hộp mục nhập mẫu của rãnh phương tiện).

Hộp này sẽ được lưu giữ khi dữ liệu mẫu được tạo cấu trúc bởi định dạng luồng byte (Byte Stream Format) như được định nghĩa trong ISO/IEC 14496-10.

Hộp thông tin mẫu con (Sub-Sample Information Box) có thể được sử dụng để lưu giữ một hoặc nhiều kích thước đơn vị NAL luồng byte (Byte stream NAL unit size) liên kề.

Mẫu con (sub-sample) được định nghĩa là một hoặc nhiều kích thước đơn vị NAL luồng byte liên kề như được định nghĩa trong ISO/IEC 14496-10.

Loại hộp (Box Type): 'bsfi'

Mục (Container): Mục nhập mẫu (Sample Entry) ('avc5', 'avc6', 'mvc5', 'mvc6' hoặc 'hev2')

Mục thiết yếu (mandatory): Không

Số lượng (quantity): Không hoặc một

```
Class ByteStreamFormatInformationBox extends Box('bsfi') {
    ByteStreamFormatInformation() ByteStreamInformation;
}
```

Trong khi "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1" là loại của bộ mã hóa giải mã trong đó tập thông số có thể được chèn, trong trường hợp của "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1", có thời gian khi mã bắt đầu được chèn vào trong và thời gian khi mã bắt đầu không được chèn.

Ngược lại, như được minh họa trên Fig.29, bằng cách mở rộng mục nhập mẫu đến "avc5", "avc6", "mvc5", "mvc6", "hev2", hoặc tương tự, ví dụ, để làm loại của bộ mã hóa giải mã trong đó mã bắt đầu được chèn và tham chiếu mục nhập mẫu này, nó có thể được nhận dạng việc liệu luồng byte có mặt trong mẫu hay không.

Ngoài ra, trong trường hợp mà mục nhập mẫu là "avc3", "avc4", "mvc3", "mvc4", hoặc "hevc1", bằng cách tạo cấu trúc thông tin định dạng luồng byte như dưới đây, nó có thể được nhận dạng việc liệu luồng byte có mặt trong mẫu

hay không (nói cách khác, việc liệu mã bắt đầu có được chèn vào trong hay không).

Hơn thế nữa, cấu trúc luồng byte của mẫu con có thể được thể hiện.

Ngoài ra, trong trường hợp của "avc5", "avc6", "mvc5", "mvc6", hoặc "hev2", bằng cách tạo cấu trúc thông tin định dạng luồng byte như dưới đây, cấu trúc của luồng byte của mẫu con có thể được thể hiện.

Ví dụ về cấu trúc dữ liệu của thông tin định dạng luồng byte

Fig.30 là sơ đồ minh họa ví dụ về cấu trúc dữ liệu của thông tin định dạng luồng byte được lưu giữ trong từng mục nhập mẫu.

```
aligned(8) class ByteStreamFormatInformation {
    unsigned int(1) StartcodePresentFlag;
    unsigned int(1) ZeroBytePresentFlag;
    unsigned int(1) LeadingZeroBytesPresentFlag;
    unsigned int(1) TrailingZeroBytesPresentFlag;
    unsigned int(4) reserved = 0;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}
```

StartcodePresentFlag là cờ thể hiện việc có mặt/không có mặt của Startcode trong mẫu. Trong trường hợp mà từng mẫu con trong số tất cả các mẫu con, mà có trong mẫu, bao gồm Startcode ba byte trước đơn vị NAL, StartcodePresentFlag là "1". ZeroBytePresentFlag là cờ mà biểu thị việc có mặt/không có mặt của dữ liệu không của một byte trước Startcode, và trong trường hợp mà cờ này là "1", nó biểu thị là Startcode là bốn byte.

Ngoài ra, thay vì StartcodePresentFlag và ZeroBytePresentFlag, Startcode có thể được biểu thị là ba byte hoặc bốn byte bằng cách sử dụng cờ hai bit.

LeadingZeroBytePresentFlag là cờ mà biểu thị việc có mặt/không có mặt của dữ liệu không trong mẫu. Trong số các mẫu con có trong mẫu, trong trường

hợp mà các 0x00 của một byte được đưa vào trước ba byte hoặc bốn byte của ít nhất một mẫu con, LeadingZeroBytePresentFlag là "1". Trong trường hợp mà LeadingZeroBytePresentFlag là "0", nó có thể thu nhận được phần bắt đầu của mẫu con là mã bắt đầu độ dài cố định.

TrailingZeroBytePresentFlag là cờ mà biểu thị việc có mặt/không có mặt của dữ liệu không trong mẫu. Trong số các mẫu con có trong mẫu, trong trường hợp mà các 0x00 của một byte được đưa vào sau đơn vị NAL của ba byte của ít nhất một mẫu con, TrailingZeroBytePresentFlag là "1".

Ngoài ra, trong số các cờ như vậy, trong khi tất cả các cờ có thể được bố trí, tất cả các cờ là không thiết yếu, và cờ khác bất kỳ có thể được bổ sung.

Ví dụ về lưu giữ đơn vị NAL theo sáng chế

Tiếp theo, ví dụ về lưu giữ đơn vị NAL của trường hợp mà thông tin định dạng luồng byte mô tả trên đây được sử dụng sẽ được mô tả. Fig.31 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 1 trong đó 0x00 (dữ liệu không) có mặt trước và sau đơn vị NAL. Nói cách khác, ví dụ được minh họa trên Fig.31 là ví dụ về cấu trúc mẫu video MP4 của trường hợp mà tất cả StartcodePresentFlag, ZeroBytePresentFlag, LeadingZeroBytePresentFlag, và TrailingZeroBytePresentFlag, mà là các cờ của thông tin định dạng luồng byte được mô tả trên đây, là "1's".

Trong trường hợp của ví dụ được minh họa trên Fig.31, ví dụ được minh họa trong đó mẫu được tạo cấu trúc bởi năm mẫu con, và kích thước của năm mẫu con được lưu giữ trong kích thước mẫu làm kích thước mẫu.

Mẫu con được tạo cấu trúc bởi dữ liệu không đầu (LeadingZeroBytes), Startcode (4 byte), đơn vị NAL, và dữ liệu không cuối (TrailingZeroBytes), và kích thước từ dữ liệu không đầu đến dữ liệu không cuối được lưu giữ trong hộp thông tin mẫu con làm kích thước của mẫu con. Nói cách khác, vì kích thước của mẫu con có mặt trong hộp thông tin mẫu con, ranh giới của mẫu con có thể thu nhận được.

Thay vì hộp thông tin mẫu con, cấu trúc như được minh họa trong ví dụ của G của Fig.27 cũng có thể được sử dụng. Để làm hàm thông tin bổ sung mẫu,

`aux_info_type = "bsfi"` và `aux_info_type_parameter = 0` được đưa ra. Hàm thông tin bổ sung mẫu được tham chiếu bởi "saiz" mà là hộp kích thước hàm thông tin bổ sung mẫu được sử dụng (`SampleAuxiliaryInformationSizesBox`) và "saio" mà là hộp dịch vị hàm thông tin bổ sung mẫu.

Ngoài ra, có các trường hợp mà hoạt động như vậy được thực hiện tại tốc độ bit không đổi (constant bitrate, viết tắt là CBR) dùng cho kinh doanh. Do đó, mẫu 1 này, ví dụ, là thích hợp cho trường hợp mà tất cả các mẫu dùng cho kinh doanh được sử dụng cho các luồng được tạo thành bởi các hình ảnh trong.

Hơn thế nữa, ranh giới giữa `LeadingZeroBytes` và `TrailingZeroBytes`, ví dụ, có thể được thiết lập bởi hoạt động của ứng dụng.

Fig.32 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 2 trong đó mẫu con bắt đầu từ `Startcode`, và `0x00` (dữ liệu không) không có mặt. Nói cách khác, ví dụ được minh họa trên Fig.32 minh họa ví dụ về cấu trúc mẫu video MP4 của trường hợp mà `StartcodePresentFlag` và `ZeroBytePresentFlag` là 1's, và `LeadingZeroBytePresentFlag` và `TrailingZeroBytePresentFlag` là 0's.

Trong trường hợp của ví dụ được minh họa trên Fig.32, ví dụ được minh họa trong đó mẫu được tạo cấu trúc bởi năm mẫu con, và kích thước của năm mẫu con được lưu giữ trong hộp kích thước mẫu làm kích thước mẫu.

Mẫu con được tạo cấu trúc bởi `Startcode` (bốn byte) và đơn vị NAL, và kích thước từ `Startcode` đến đơn vị NAL được lưu giữ trong hộp thông tin mẫu con làm kích thước của mẫu con. Nói cách khác, vì kích thước của mẫu con có mặt trong hộp thông tin mẫu con, ranh giới của mẫu con có thể thu nhận được.

Mẫu 2 này là ví dụ đơn giản nhất.

Fig.33 là sơ đồ minh họa ví dụ về cấu trúc mẫu video MP4 trong trường hợp của mẫu 3 trong đó mẫu con được bắt đầu từ `Startcode`, và `0x00` (dữ liệu không) có mặt sau NAL. Nói cách khác, ví dụ được minh họa trên Fig.33 minh họa ví dụ về cấu trúc mẫu video MP4 của trường hợp mà `StartcodePresentFlag`, `ZeroBytePresentFlag`, và `TrailingZeroBytePresentFlag` là 1's, và `LeadingZeroBytePresentFlag` là 0.

Trong trường hợp của ví dụ được minh họa trên Fig.33, ví dụ được minh họa trong đó mẫu được tạo cấu trúc bởi năm mẫu con, và kích thước của năm mẫu con được lưu giữ trong hộp kích thước mẫu làm kích thước mẫu.

Mẫu con được tạo cấu trúc bởi Startcode (bốn byte), đơn vị NAL, và dữ liệu không cuối (TrailingZeroBytes), và kích thước từ Startcode đến dữ liệu không cuối được lưu giữ trong hộp thông tin mẫu con làm kích thước của mẫu con. Nói cách khác, vì kích thước của mẫu con có mặt trong hộp thông tin mẫu con, ranh giới của mẫu con có thể thu nhận được.

Ngoài ra, trong trường hợp của mẫu 3 này, ví dụ, trong bộ soạn thảo của các luồng trong đó tất cả các mẫu dùng cho kinh doanh được tạo thành bởi các hình ảnh trong, nó có thể được biết là độ dài cố định được thiết lập bởi TrailingZero, và do đó, có ưu điểm là luồng có thể được soạn thảo dễ dàng. Ngoài ra, có các trường hợp mà mã bắt đầu ba byte cũng được sử dụng cho hệ thống hội nghị truyền hình hoặc tương tự.

Theo cách này, thông tin định dạng luồng byte được định nghĩa bằng cách tạo thông tin định dạng luồng byte như là hộp. Ví dụ, vì mẫu con được tạo thành như là một trong số mẫu 1 đến 3 được mô tả trên đây có dựa vào Fig.31 đến Fig.33, nó có thể được nhận dạng việc liệu luồng byte có nằm trong mẫu hay không.

Các tệp của các mẫu video MP4 được minh họa trên Fig.31 đến Fig.33 được tạo ra bởi bộ tạo tệp 2 có cấu trúc được minh họa trên Fig.3 được mô tả trên đây qua xử lý tạo tệp được minh họa trên Fig.4 hoặc Fig.5 được mô tả trên đây.

Ngoài ra, các tệp của các mẫu video MP4 được minh họa trên Fig.31 đến Fig.33 được giải mã bởi bộ đọc tệp 51 có cấu trúc được minh họa trên Fig.9 được mô tả trên đây qua xử lý giải mã tệp được minh họa trên Fig.11 hoặc Fig.12 được mô tả trên đây.

Ví dụ khác về cấu trúc dữ liệu thông tin định dạng luồng byte

Fig.34 là sơ đồ minh họa ví dụ về cấu trúc dữ liệu của thông tin định dạng luồng byte được lưu giữ trong từng mục nhập mẫu. Nói cách khác, trong ví dụ

được minh họa trên Fig.34, ví dụ khác về cấu trúc dữ liệu được mô tả trên đây có dựa vào Fig.30 được minh họa.

```
aligned(8) class ByteStreamFormatInformation {
    unsigned int(2) StartcodePresentFlag;
    unsigned int(1) LeadingZeroBytesPresentFlag;
    unsigned int(1) TrailingZeroBytesPresentFlag;
    unsigned int(4) reserved = 0;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}
```

StartcodePresentFlag là cờ hai bit mà minh họa cấu trúc của Startcode của từng mẫu con trong số tất cả các mẫu con.

"StartcodePresentFlag = 00b" thể hiện "dự phòng".

"StartcodePresentFlag = 01b" thể hiện là Startcode ba byte được đưa vào trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu.

"StartcodePresentFlag = 10b" thể hiện là Startcode ba byte và dữ liệu ZeroByte một byte (nói cách khác, bốn byte tạo thành) được đưa vào trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu.

"StartcodePresentFlag = 11b" thể hiện là, trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu, Startcode ba byte được đưa vào trong, và dữ liệu ZeroByte một byte có thể được đưa vào trong. Nói cách khác, trong trường hợp này, vì có khả năng là Startcode ba byte và Startcode bốn byte được trộn, không thể đảm bảo việc liệu StartcodePresentFlag là 01b hoặc 10b.

Ví dụ này là ví dụ khác về cấu trúc được mô tả có dựa vào Fig.30 và là ví dụ trong đó kích thước của Startcode có thể thu nhận được. Ngoài ra, do cờ hai bit, ba byte, bốn byte, hoặc hỗn hợp của chúng xuất hiện.

LeadingZeroBytePresentFlag thể hiện là, trong số các mẫu con có trong

mẫu, trước Startcode của ba byte hoặc bốn byte của ít nhất một mẫu con, từng 0x00 được tạo thành bởi một byte được đưa vào.

TrailingZeroBytePresentFlag thể hiện là, trong số các mẫu con có trong mẫu, các 0x00 của một byte có thể được đưa vào trong sau đơn vị NAL của ít nhất một mẫu con.

Ngoài ra, cũng trong ví dụ được minh họa trên Fig.34, trong khi tất cả các cờ như vậy có thể có mặt, tuy nhiên, các cờ là không cần thiết, và cờ khác có thể được bổ sung.

5. Phương án thứ năm

Ví dụ cấu hình của thiết bị chuyển đổi tệp

Fig.35 là sơ đồ khối minh họa ví dụ về cấu hình của thiết bị chuyển đổi tệp như là thiết bị xử lý ảnh được áp dụng sáng chế. Thiết bị chuyển đổi tệp 300 được minh họa trên Fig.35, ví dụ, chuyển đổi MPEG-2 TS thành tệp MP4.

Trong ví dụ được minh họa trên Fig.35, thiết bị chuyển đổi tệp 300 được tạo cấu hình bao gồm bộ phân tích luồng video 311 và bộ tạo tệp 2 được minh họa trên Fig.1.

Bộ phân tích luồng video 311 phân tích định dạng của luồng video đầu vào. Trong trường hợp mà định dạng được phân tích là MPEG-2 TS, bộ phân tích luồng video 311 cấp luồng video đến bộ tạo tệp 2.

Bộ tạo tệp 2 thực hiện xử lý chuyển đổi MPEG-2 TS thành tệp MP4. Nói cách khác, bộ tạo tệp 2, tương tự như ví dụ được minh họa trên Fig.1, tạo tệp MP4 trong đó MPEG-2 TS được lưu giữ.

Ví dụ về xử lý chuyển đổi tệp

Tiếp theo, xử lý chuyển đổi tệp được thực hiện bởi thiết bị chuyển đổi tệp 300 được minh họa trên Fig.35 sẽ được mô tả có dựa vào lưu đồ được minh họa trên Fig.36.

Bộ phân tích luồng video 311, ở bước S311, phân tích định dạng của luồng video đầu vào, và trong trường hợp mà định dạng được phân tích là MPEG-2 TS, cấp luồng video đến bộ tạo tệp 2.

Ở bước S312, bộ tạo tệp 2 tạo tệp MP4 trong đó MPEG-2 TS được lưu giữ. Xử lý tạo tệp này giống như xử lý tạo tệp được mô tả trên đây có dựa vào Fig.5, và do vậy phần mô tả lặp lại sẽ không được thể hiện.

Thiết bị chuyển đổi tệp 300 có thể được tạo cấu hình như được mô tả trên đây.

Mặc dù thiết bị mã hóa và thiết bị giải mã sử dụng hệ thống HEVC đã được mô tả trên đây, tuy nhiên, sáng chế, như được mô tả dưới đây, cũng có thể áp dụng cho thiết bị mã hóa và thiết bị giải mã sử dụng hệ thống AVC.

6. Phương án thứ sáu

Ví dụ cấu hình của thiết bị mã hóa

Fig.37 là sơ đồ khối minh họa ví dụ khác về cấu hình của thiết bị mã hóa như là thiết bị xử lý ảnh được áp dụng sáng chế.

Trong cấu hình được minh họa trên Fig.37, cùng số chỉ dẫn được gán cho cùng bộ phận cấu hình giống như được minh họa trên Fig.1. Ngoài ra, phần mô tả lặp lại sẽ không được thể hiện một cách thích hợp.

Cấu hình của thiết bị mã hóa được minh họa trên Fig.37 khác với cấu hình được minh họa trên Fig.1 ở chỗ bộ mã hóa 401 được bố trí thay vì bộ mã hóa 1. Cấu hình của thiết bị mã hóa được minh họa trên Fig.37 là chung với cấu hình được minh họa trên Fig.1 ở chỗ bộ tạo tệp 2 được bố trí.

Nói cách khác, ảnh như ảnh được chụp được tạo thành theo các đơn vị khung được nhập vào bộ mã hóa 401 làm tín hiệu đầu vào. Ngoài ra, SPS được thiết lập trên tầng trước của bộ mã hóa 401 không được minh họa trên hình vẽ, VUI mà thể hiện đặc trưng của ảnh tương ứng với dữ liệu đã được mã hóa đối với từng trình tự, SEI, và tương tự được nhập vào đó.

Bộ mã hóa 401 mã hóa tín hiệu đầu vào bằng cách sử dụng các thông số có trong SPS, PPS, VUI, và SEI bằng cách sử dụng hệ thống AVC. Sau đó, bộ mã hóa 401 tạo luồng đã được mã hóa như MPEG-2 TS dựa vào SPS, PPS, VUI, và SEI và dữ liệu đã được mã hóa thu nhận được nhờ kết quả của xử lý mã hóa và cấp luồng đã được mã hóa được tạo ra đến bộ tạo tệp 2. Bộ mã hóa 401 sẽ được mô tả chi tiết sau có dựa vào Fig.38.

Bộ tạo tệp 2 tạo tệp lưu giữ luồng đã được mã hóa (chuỗi của dữ liệu ảnh đã được mã hóa) được cấp từ bộ mã hóa 401.

Nói cách khác, thiết bị mã hóa được minh họa trên Fig.36 khác với thiết bị mã hóa được minh họa trên Fig.1 chỉ ở chỗ xử lý mã hóa sử dụng hệ thống AVC được thực hiện.

Ví dụ cấu hình của bộ mã hóa

Fig.38 là sơ đồ khối minh họa ví dụ về cấu hình của bộ mã hóa 401 được minh họa trên Fig.37.

Trong cấu hình được minh họa trên Fig.38, cùng số chỉ dẫn được gán cho cùng bộ phận cấu hình giống như được minh họa trên Fig.2. Ngoài ra, phần mô tả lặp lại sẽ không được thể hiện một cách thích hợp.

Bộ mã hóa 401 được minh họa trên Fig.38 được tạo cấu hình bao gồm: bộ chuyển đổi A/D 11; bộ đệm sắp xếp lại màn hình 12; bộ tính toán 13; bộ biến đổi trực giao 14; bộ lượng tử hóa 15; bộ mã hóa không tổn hao 16; bộ đệm tích lũy 17; bộ lượng tử hóa ngược 18; bộ biến đổi trực giao ngược 19; bộ cộng 20; bộ lọc giải khối 21; bộ nhớ khung 22; chuyển mạch 23; bộ dự báo trong 24; bộ dự báo/bù chuyển động 25; bộ lựa chọn ảnh đã được dự báo 26; và bộ điều khiển tốc độ 27.

Nói cách khác, cấu hình của bộ mã hóa 401 được minh họa trên Fig.38 khác với cấu hình được minh họa trên Fig.2 chỉ ở chỗ bộ lọc dịch vị thích ứng 141 và bộ lọc vòng lặp thích ứng 142 được loại trừ, và bộ mã hóa không tổn hao 16 thực hiện mã hóa không sử dụng hệ thống HEVC mà hệ thống AVC. Do đó, bộ mã hóa 401 thực hiện xử lý mã hóa không phải theo các đơn vị CU mà theo các đơn vị khối.

Đích cho xử lý mã hóa được thực hiện bởi bộ mã hóa không tổn hao 16 về cơ bản giống như của bộ mã hóa không tổn hao 16 được minh họa trên Fig.2 ngoại trừ các thông số của bộ lọc dịch vị thích ứng và bộ lọc vòng lặp thích ứng. Nói cách khác, bộ mã hóa không tổn hao 16, tương tự như bộ mã hóa không tổn hao 16 được minh họa trên Fig.2, thu nhận thông tin chế độ dự báo trong từ bộ dự báo trong 24. Ngoài ra, bộ mã hóa không tổn hao 16 thu nhận thông tin chế

độ dự báo giữa, vectơ chuyển động, thông tin được sử dụng để xác định ảnh tham chiếu, và tương tự từ bộ dự báo/bù chuyển động 25.

Bộ mã hóa không tổn hao 16, tương tự như bộ mã hóa không tổn hao 16 được minh họa trên Fig.2, thực hiện mã hóa không tổn hao như mã hóa độ dài thay đổi (ví dụ, CAVLC hoặc tương tự), mã hóa số học (ví dụ, CABAC hoặc tương tự) đối với các hệ số đã được lượng tử hóa được cấp từ bộ lượng tử hóa 15.

Ngoài ra, bộ mã hóa không tổn hao 16, tương tự như bộ mã hóa không tổn hao 16 được minh họa trên Fig.2, thực hiện mã hóa không tổn hao thông tin chế độ dự báo trong hoặc thông tin chế độ dự báo giữa, vectơ chuyển động, thông tin xác định ảnh tham chiếu, thông tin lọc dịch vị, các hệ số lọc, và tương tự như là thông tin liên quan đến mã hóa. Bộ mã hóa không tổn hao 16 cấp thông tin mã hóa và các hệ số mà đã được mã hóa theo cách không tổn hao đến bộ đệm tích lũy 17 như là dữ liệu đã được mã hóa để được tích lũy trong đó. Ở đây, thông tin mã hóa mà được mã hóa theo cách không tổn hao có thể được coi là thông tin đoạn đầu của các hệ số mà đã được mã hóa theo cách không tổn hao.

Bộ lọc giải khối 21 lọc ảnh, mà được giải mã cục bộ, được cấp từ bộ cộng 20, bằng cách đó loại bỏ biến dạng khối. Bộ lọc giải khối 21 cấp ảnh thu nhận được như vậy đến bộ nhớ khung 22 để được tích lũy trong đó.

Ảnh được tích lũy trong bộ nhớ khung 22 được xuất đến bộ dự báo trong 24 hoặc bộ dự báo/bù chuyển động 25 qua chuyển mạch 23 như là ảnh tham chiếu.

Sáng chế có thể áp dụng cho thiết bị mã hóa của hệ thống AVC.

Ví dụ cấu hình của thiết bị giải mã

Fig.39 là sơ đồ khối minh họa ví dụ khác về cấu hình của thiết bị giải mã, được áp dụng sáng chế, mà đọc và giải mã tệp được tạo ra bằng thiết bị mã hóa được minh họa trên Fig.37.

Trong cấu hình được minh họa trên Fig.39, cùng số chỉ dẫn được gán cho cùng bộ phận cấu hình giống như được minh họa trên Fig.8. Ngoài ra, phần mô tả lặp lại sẽ không được thể hiện một cách thích hợp.

Cấu hình của thiết bị giải mã ảnh 251 được minh họa trên Fig.39 khác với cấu hình được minh họa trên Fig.8 ở chỗ bộ giải mã 451 được bố trí thay vì bộ giải mã 52. Cấu hình của thiết bị giải mã ảnh 251 là chung với cấu hình được minh họa trên Fig.8 ở chỗ bộ đọc tệp 51 được bố trí.

Bộ đọc tệp 51 của thiết bị giải mã thu tệp của MP4 được tạo ra bởi thiết bị mã hóa được minh họa trên Fig.37 và đọc SPS, PPS, VUI, SEI, dữ liệu đã được mã hóa, và tương tự tạo thành luồng đã được mã hóa mà được mã hóa bởi bộ mã hóa 401 được minh họa trên Fig.37 từ tệp đã thu được. Bộ đọc tệp 51 cấp SPS, PPS, VUI, SEI, và dữ liệu đã được mã hóa đến bộ giải mã 451.

Bộ giải mã 451, theo sự điều khiển của bộ đọc tệp 51, tham chiếu SPS, PPS, VUI, SEI, và tương tự được cấp từ bộ đọc tệp 51 và giải mã dữ liệu đã được mã hóa được cấp từ bộ đọc tệp 51 bằng cách sử dụng hệ thống AVC. Bộ giải mã 451 cấp ảnh thu nhận được nhờ kết quả của xử lý giải mã đến tầng sau như là tín hiệu đầu ra.

Nói cách khác, thiết bị giải mã được minh họa trên Fig.39 khác với thiết bị giải mã được minh họa trên Fig.8 chỉ ở chỗ xử lý giải mã sử dụng hệ thống AVC được thực hiện.

Ví dụ cấu hình của bộ giải mã

Fig.40 là sơ đồ khối minh họa ví dụ về cấu hình của bộ giải mã 451 được minh họa trên Fig.39.

Trong cấu hình được minh họa trên Fig.40, cùng số chỉ dẫn được gán cho cùng bộ phận cấu hình giống như được minh họa trên Fig.10. Ngoài ra, phần mô tả lặp lại sẽ không được thể hiện một cách thích hợp.

Bộ giải mã 451 được minh họa trên Fig.40 được tạo cấu hình bởi: bộ đệm tích lũy 101; bộ giải mã không tổn hao 102; bộ lượng tử hóa ngược 103; bộ biến đổi trực giao ngược 104; bộ cộng 105; bộ lọc giải khối 106; bộ đệm sắp xếp lại màn hình 107; bộ chuyển đổi D/A 108; bộ nhớ khung 109; chuyển mạch 110; bộ dự báo trong 111; bộ bù chuyển động 112; và chuyển mạch 113.

Cấu hình của bộ giải mã 451 được minh họa trên Fig.40 khác với cấu hình được minh họa trên Fig.10 chỉ ở chỗ bộ lọc dịch vị thích ứng 141 và bộ lọc vòng

lắp thích ứng 142 được loại trừ, và bộ giải mã không tốn hao 102 thực hiện giải mã bằng cách không sử dụng hệ thống HEVC mà hệ thống AVC. Do đó, bộ giải mã 451 thực hiện xử lý giải mã không theo các đơn vị CU mà theo các đơn vị khối.

Đích đối với xử lý giải mã được thực hiện bởi bộ giải mã không tốn hao 102 về cơ bản giống như của trường hợp của bộ giải mã không tốn hao 102 được minh họa trên Fig.10 ngoại trừ các thông số của bộ lọc dịch vị thích ứng và bộ lọc vòng lắp thích ứng. Nói cách khác, bộ giải mã không tốn hao 102, tương tự như bộ giải mã không tốn hao 102 được minh họa trên Fig.10, thực hiện giải mã không tốn hao như giải mã độ dài thay đổi hoặc giải mã số học đối với dữ liệu đã được mã hóa được cấp từ bộ đệm tích lũy 101, bằng cách đó thu nhận các hệ số đã được lượng tử hóa và thông tin mã hóa. Bộ giải mã không tốn hao 102 cấp các hệ số đã được lượng tử hóa đến bộ lượng tử hóa ngược 103.

Ngoài ra, bộ giải mã không tốn hao 102, tương tự như bộ giải mã không tốn hao 102 được minh họa trên Fig.10, cấp thông tin chế độ dự báo trong và tương tự như là thông tin mã hóa đến bộ dự báo trong 111 và cấp vector chuyển động, thông tin được sử dụng để xác định ảnh tham chiếu, thông tin chế độ dự báo giữa, và tương tự đến bộ bù chuyển động 112. Hơn thế nữa, bộ giải mã không tốn hao 102 cấp thông tin chế độ dự báo trong hoặc thông tin chế độ dự báo giữa như là thông tin mã hóa đến chuyển mạch 113.

Bộ lọc giải khối 106 lọc ảnh được cấp từ bộ cộng 105, bằng cách đó loại bỏ biến dạng khối. Bộ lọc giải khối 106 cấp ảnh thu nhận được như vậy đến bộ nhớ khung 109 và bộ đệm sắp xếp lại màn hình 107.

Sáng chế cũng có thể áp dụng cho thiết bị giải mã của hệ thống AVC.

Trong phần mô tả được thể hiện trên đây, mặc dù ví dụ của định dạng tệp MP4 đã được mô tả, tuy nhiên, định dạng tệp không bị giới hạn ở định dạng tệp MP4 hoặc định dạng tệp AVC. Trong trường hợp mà mục đích và ưu điểm theo sáng chế là giống nhau, sáng chế có thể được áp dụng một cách tương tự cho định dạng tệp khác, luồng được sử dụng tại thời gian truyền, hoặc luồng được sử dụng khi được lưu giữ trong tệp.

Ngoài ra, sáng chế, ví dụ, có thể áp dụng cho thiết bị mã hóa ảnh và thiết bị giải mã ảnh mà được sử dụng khi thông tin ảnh (luồng bit) được nén bằng cách sử dụng biến đổi trực giao như biến đổi cosin rời rạc và bù chuyển động, giống như hệ thống HEVC hoặc tương tự, thu được qua môi trường mạng như phát rộng vệ tinh, truyền hình cáp, Internet, hoặc điện thoại di động. Hơn thế nữa, sáng chế có thể được áp dụng cho thiết bị mã hóa ảnh và thiết bị giải mã ảnh mà được sử dụng khi thông tin được xử lý trên phương tiện lưu giữ như đĩa quang, đĩa từ, hoặc bộ nhớ tia chớp.

Chuỗi các xử lý được mô tả trên đây có thể được thực hiện bằng phần cứng hoặc bằng phần mềm. Trong trường hợp mà chuỗi các xử lý được thực hiện bằng phần mềm, chương trình tạo thành phần mềm được cài vào máy tính. Ở đây, máy tính bao gồm máy tính mà được lắp trong phần cứng chuyên dụng, máy tính như máy tính cá nhân đa năng mà có thể thực hiện các chức năng khác nhau bằng cách cài vào các chương trình, và tương tự.

Fig.41 là sơ đồ khối minh họa ví dụ về cấu hình phần cứng của máy tính mà thực hiện chuỗi các xử lý được mô tả trên đây bằng cách sử dụng chương trình.

Trong máy tính 800, bộ xử lý trung tâm (Central Processing Unit, viết tắt là CPU) 801, bộ nhớ chỉ đọc (Read Only Memory, viết tắt là ROM) 802, và bộ nhớ truy cập ngẫu nhiên (Random Access Memory, viết tắt là RAM) 803 được liên kết qua bus 804.

Ngoài ra, giao diện nhập/xuất 805 được nối với bus 804. Bộ nhập 806, bộ xuất 807, bộ lưu giữ 808, bộ truyền thông 809, và ổ đĩa 810 được nối với giao diện nhập/xuất 805.

Bộ nhập 806 được tạo thành bởi bàn phím, chuột, micro, và tương tự. Bộ xuất 807 được tạo cấu trúc bởi màn hiển thị, loa, và tương tự. Bộ lưu giữ 808 được tạo thành bởi đĩa cứng, bộ nhớ bất khả biến, và tương tự. Bộ truyền thông 809 được tạo thành bởi giao diện mạng và tương tự. Ổ đĩa 810 dẫn động đĩa từ, đĩa quang, đĩa từ quang, hoặc phương tiện tháo lắp được 811 như bộ nhớ bán dẫn.

Trong máy tính có cấu hình như trên đây, CPU 801, ví dụ, tải chương trình được lưu giữ trong bộ lưu giữ 808 vào RAM 803 qua giao diện nhập/xuất 805 và buýt 804 và thực hiện chương trình đã được tải, bằng cách đó thực hiện chuỗi các xử lý được mô tả trên đây.

Chương trình được thực hiện bởi máy tính 800 (CPU 801), ví dụ, có thể được cung cấp bằng cách được ghi trên phương tiện tháo lắp được 811 như là phương tiện đóng gói hoặc tương tự. Ngoài ra, chương trình có thể được cung cấp qua môi trường hữu tuyến hoặc vô tuyến như mạng cục bộ, Internet, hoặc phát rộng vệ tinh số.

Bằng cách lắp phương tiện tháo lắp được 811 vào ổ đĩa 810, chương trình có thể được cài vào bộ lưu giữ 808 qua giao diện nhập/xuất 805. Ngoài ra, chương trình có thể thu được bởi bộ truyền thông 809 qua môi trường hữu tuyến hoặc vô tuyến và được cài vào bộ lưu giữ 808. Hơn thế nữa, chương trình có thể được cài vào ROM 802 hoặc bộ lưu giữ 808 từ trước.

Ngoài ra, chương trình được thực hiện bởi máy tính có thể là chương trình mà thực hiện các xử lý theo chuỗi thời gian dọc theo trình tự được mô tả trong bản mô tả sáng chế này hoặc chương trình mà thực hiện các xử lý theo cách song song hoặc tại thời gian cần thiết như tại thời gian được gọi.

Hơn thế nữa, trong bản mô tả sáng chế này, bước mô tả chương trình được ghi trên phương tiện ghi bao gồm không chỉ xử lý được thực hiện theo chuỗi thời gian trình tự được mô tả mà cả xử lý mà được thực hiện theo cách song song hoặc riêng rẽ mà không nhất thiết được xác định theo chuỗi thời gian.

Ngoài ra, trong bản mô tả sáng chế này, hệ thống thể hiện toàn bộ thiết bị mà được tạo thành bởi các thiết bị (các máy).

Hơn thế nữa, cấu hình được mô tả trên đây như là một thiết bị (hoặc bộ xử lý) có thể được chia để được tạo thành bởi các thiết bị (hoặc các bộ xử lý). Ngược lại, cấu hình được mô tả trên đây như là các thiết bị (hoặc các bộ xử lý) có thể được bố trí để được tạo cấu hình như là một thiết bị (hoặc bộ xử lý). Ngoài ra, cấu hình mà chưa được mô tả trên đây có thể được bổ sung vào cấu hình của từng thiết bị (hoặc từng bộ xử lý). Chỉ cần cấu hình tổng thể và hoạt

động tổng thể của hệ thống gần như giống nhau, một phần của cấu hình của thiết bị cụ thể (hoặc bộ xử lý cụ thể) có thể được tạo cấu hình để có trong cấu hình của thiết bị khác (hoặc bộ xử lý khác). Nói cách khác, sáng chế không bị giới hạn ở các phương án được mô tả trên đây, và các thay đổi khác nhau có thể được thực hiện trong khoảng không nằm ngoài khái niệm của sáng chế.

Nói cách khác, sáng chế không bị giới hạn ở các phương án được mô tả trên đây, và các thay đổi khác nhau có thể được thực hiện trong khoảng không nằm ngoài khái niệm của sáng chế.

Ví dụ, sáng chế có thể áp dụng điện toán đám mây trong đó một chức năng được chia và xử lý một cách phối hợp bằng các thiết bị qua mạng.

Ngoài ra, từng bước được mô tả trong từng lưu đồ được mô tả trên đây có thể được thực hiện bằng một thiết bị hoặc được thực hiện bằng các thiết bị theo cách chia sẻ.

Hơn thế nữa, trong trường hợp mà các xử lý có trong một bước, các xử lý có trong một bước có thể được thực hiện bằng một thiết bị hoặc được thực hiện bằng các thiết bị theo cách chia sẻ.

Thiết bị mã hóa ảnh và thiết bị giải mã ảnh theo các phương án được mô tả trên đây có thể áp dụng cho các thiết bị điện tử khác nhau như bộ truyền hoặc bộ thu dùng cho phát rộng hữu tuyến như phát rộng vệ tinh hoặc TV cáp, cáp trên Internet, cáp đến thiết bị đầu cuối qua truyền thông tế bào, hoặc tương tự, thiết bị ghi mà ghi ảnh trên phương tiện như đĩa quang, đĩa từ, hoặc bộ nhớ tia chớp, hoặc thiết bị tái tạo mà tái tạo ảnh từ phương tiện lưu giữ. Dưới đây, bốn ví dụ ứng dụng sẽ được mô tả.

7. Ví dụ ứng dụng

Ví dụ ứng dụng thứ nhất: Bộ thu truyền hình

Fig.42 minh họa ví dụ về cấu hình giản lược của thiết bị truyền hình mà được áp dụng phương án mô tả trên đây. Thiết bị truyền hình 900 bao gồm anten 901, bộ điều hướng 902, bộ phân kênh 903, bộ giải mã 904, bộ xử lý tín hiệu video 905, bộ hiển thị 906, bộ xử lý tín hiệu audio 907, loa 908, giao diện ngoài 909, bộ điều khiển 910, giao diện người dùng 911, và buýt 912.

Bộ điều hướng 902 trích tín hiệu của kênh mong muốn từ tín hiệu phát rộng thu được qua anten 901 và giải điều biến tín hiệu đã được trích. Sau đó, bộ điều hướng 902 xuất luồng bit đã được mã hóa thu nhận được qua điều biến đến bộ phân kênh 903. Nói cách khác, bộ điều hướng 902 có chức năng làm phương tiện truyền của thiết bị truyền hình 900 mà thu luồng đã được mã hóa trong đó ảnh được mã hóa.

Bộ phân kênh 903 tách luồng video và luồng audio của chương trình sẽ được xem từ luồng bit đã được mã hóa và xuất từng luồng đã được tách đến bộ giải mã 904. Ngoài ra, bộ phân kênh 903 trích dữ liệu bổ sung như hướng dẫn chương trình điện tử (Electronic Program Guide, viết tắt là EPG) từ luồng bit đã được mã hóa và cấp dữ liệu đã được trích đến bộ điều khiển 910. Hơn thế nữa, bộ phân kênh 903 có thể thực hiện khử xáo trộn trong trường hợp mà luồng bit đã được mã hóa được xáo trộn.

Bộ giải mã 904 giải mã luồng video và luồng audio được nhập từ bộ phân kênh 903. Sau đó, bộ giải mã 904 xuất dữ liệu video được tạo ra bằng xử lý giải mã đến bộ xử lý tín hiệu video 905. Ngoài ra, bộ giải mã 904 xuất dữ liệu audio được tạo ra bằng xử lý giải mã đến bộ xử lý tín hiệu audio 907.

Bộ xử lý tín hiệu video 905 tái tạo dữ liệu video được nhập từ bộ giải mã 904 và khiến bộ hiển thị 906 hiển thị video. Bộ xử lý tín hiệu video 905 cũng có thể khiến bộ hiển thị 906 hiển thị màn hình ứng dụng được cấp qua mạng. Ngoài ra, bộ xử lý tín hiệu video 905 có thể thực hiện xử lý bổ sung như loại bỏ nhiễu cho dữ liệu video theo thiết lập. Hơn thế nữa, bộ xử lý tín hiệu video 905 có thể tạo ảnh giao diện người dùng đồ họa (Graphical User Interface, viết tắt là GUI) như thực đơn, nút, và con trỏ và chồng ảnh đã được tạo ra trên ảnh đầu ra.

Bộ hiển thị 906 được dẫn động theo tín hiệu dẫn động được cấp từ bộ xử lý tín hiệu video 905 để hiển thị video hoặc ảnh trên màn hình video của thiết bị hiển thị (ví dụ, màn hiển thị tinh thể lỏng, màn hiển thị plasma, màn hiển thị phát sáng điện tử hữu cơ (Organic ElectroLuminescence Display, viết tắt là OLED (màn hiển thị EL hữu cơ), hoặc tương tự).

Bộ xử lý tín hiệu audio 907 thực hiện xử lý tái tạo như chuyển đổi D/A và khuếch đại đối với dữ liệu audio được nhập từ bộ giải mã 904 và khiến loa 908

xuất audio. Ngoài ra, bộ xử lý tín hiệu audio 907 có thể thực hiện xử lý bổ sung như loại bỏ nhiễu cho dữ liệu audio.

Giao diện ngoài 909 là giao diện để nối thiết bị truyền hình 900 với thiết bị bên ngoài hoặc mạng. Ví dụ, luồng video hoặc luồng audio thu được qua giao diện ngoài 909 có thể được giải mã bởi bộ giải mã 904. Nói cách khác, giao diện ngoài 909 cũng có chức năng làm phương tiện truyền của thiết bị truyền hình 900 mà thu luồng đã được mã hóa trong đó ảnh được mã hóa.

Bộ điều khiển 910 bao gồm bộ xử lý như CPU và bộ nhớ như RAM hoặc ROM. Bộ nhớ lưu giữ chương trình được thực hiện bởi CPU, dữ liệu chương trình, dữ liệu EPG, dữ liệu thu nhận được qua mạng, và tương tự. Chương trình được lưu giữ trong bộ nhớ, ví dụ, được đọc bởi CPU khi kích hoạt thiết bị truyền hình 900 và được thực hiện. CPU điều khiển hoạt động của thiết bị truyền hình 900, ví dụ, theo tín hiệu hoạt động được nhập từ giao diện người dùng 911 bằng cách thực hiện chương trình.

Giao diện người dùng 911 được nối với bộ điều khiển 910. Giao diện người dùng 911, ví dụ, bao gồm nút và chuyển mạch để người dùng vận hành thiết bị truyền hình 900, bộ thu dùng cho tín hiệu điều khiển từ xa, và tương tự. Giao diện người dùng 911 dò thao tác của người dùng qua các bộ phận như vậy, tạo tín hiệu hoạt động, và xuất tín hiệu hoạt động đã được tạo ra đến bộ điều khiển 910.

Buýt 912 nối bộ điều hướng 902, bộ phân kênh 903, bộ giải mã 904, bộ xử lý tín hiệu video 905, bộ xử lý tín hiệu audio 907, giao diện ngoài 909, và bộ điều khiển 910 với nhau.

Trong thiết bị truyền hình 900 có cấu hình theo cách này, bộ giải mã 904 có chức năng của thiết bị giải mã ảnh (ví dụ, thiết bị giải mã được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52 được minh họa trên Fig.8) theo phương án mô tả trên đây. Do đó, khi giải mã ảnh mà được thực hiện bằng thiết bị truyền hình 900, gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng khi được lưu giữ trong tệp được giải mã có thể được giảm.

Ví dụ ứng dụng thứ hai: Điện thoại di động

Fig.43 minh họa ví dụ về cấu hình giản lược của điện thoại di động mà được áp dụng phương án mô tả trên đây. Điện thoại di động 920 bao gồm anten 921, bộ truyền thông 922, bộ mã hóa giải mã audio 923, loa 924, micro 925, bộ camera 926, bộ xử lý ảnh 927, bộ dồn kênh/tách 928, bộ ghi/tái tạo 929, bộ hiển thị 930, bộ điều khiển 931, bộ thao tác 932, và buýt 933.

Anten 921 được nối với bộ truyền thông 922. Loa 924 và micro 925 được nối với bộ mã hóa giải mã audio 923. Bộ thao tác 932 được nối với bộ điều khiển 931. Buýt 933 nối bộ truyền thông 922, bộ mã hóa giải mã audio 923, bộ camera 926, bộ xử lý ảnh 927, bộ dồn kênh/tách 928, bộ ghi/tái tạo 929, bộ hiển thị 930, và bộ điều khiển 931 với nhau.

Điện thoại di động 920 thực hiện hoạt động như truyền/thu tín hiệu audio, truyền/thu thư điện tử hoặc dữ liệu ảnh, chụp ảnh, và ghi dữ liệu trong các chế độ khác nhau bao gồm chế độ cuộc gọi thoại, chế độ truyền thông dữ liệu, chế độ tạo ảnh, và chế độ điện thoại truyền hình.

Trong chế độ cuộc gọi thoại, tín hiệu audio tương tự được tạo ra bởi micro 925 được cấp đến bộ mã hóa giải mã audio 923. Bộ mã hóa giải mã audio 923 chuyển đổi tín hiệu audio tương tự thành dữ liệu audio, thực hiện chuyển đổi A/D đối với dữ liệu audio đã được chuyển đổi, và nén dữ liệu audio. Sau đó, bộ mã hóa giải mã audio 923 xuất dữ liệu audio đã được nén đến bộ truyền thông 922. Bộ truyền thông 922 mã hóa và điều biến dữ liệu audio để tạo tín hiệu truyền. Sau đó, bộ truyền thông 922 truyền tín hiệu truyền đã được tạo ra đến trạm cơ sở (không được minh họa trên hình vẽ) qua anten 921. Ngoài ra, bộ truyền thông 922 khuếch đại tín hiệu vô tuyến thu được qua anten 921 và thực hiện chuyển đổi tần số đối với tín hiệu vô tuyến, bằng cách đó thu nhận tín hiệu thu. Sau đó, bộ truyền thông 922 tạo dữ liệu audio bằng cách giải điều biến và giải mã tín hiệu thu và xuất dữ liệu audio đã được tạo ra đến bộ mã hóa giải mã audio 923. Bộ mã hóa giải mã audio 923 thực hiện giải nén và chuyển đổi D/A đối với dữ liệu audio, bằng cách đó tạo tín hiệu audio tương tự. Sau đó, bộ mã hóa giải mã audio 923 cấp tín hiệu audio đã được tạo ra đến loa 924 để khiến audio được xuất.

Trong chế độ truyền thông dữ liệu, ví dụ, bộ điều khiển 931 tạo dữ liệu ký tự tạo thành thư điện tử theo thao tác của người dùng được thực hiện qua bộ thao tác 932. Ngoài ra, bộ điều khiển 931 khiến bộ hiển thị 930 hiển thị các ký tự. Bộ điều khiển 931 tạo dữ liệu thư điện tử theo chỉ lệnh truyền từ người dùng qua bộ thao tác 932 và xuất dữ liệu thư điện tử đã được tạo ra đến bộ truyền thông 922. Bộ truyền thông 922 mã hóa và giải điều biến dữ liệu thư điện tử, bằng cách đó tạo tín hiệu truyền. Sau đó, bộ truyền thông 922 truyền tín hiệu truyền đã được tạo ra đến trạm cơ sở (không được minh họa trên hình vẽ) qua anten 921. Ngoài ra, bộ truyền thông 922 thực hiện khuếch đại và chuyển đổi tần số đối với tín hiệu vô tuyến thu được qua anten 921, bằng cách đó thu nhận tín hiệu thu. Sau đó, bộ truyền thông 922 giải điều biến và giải mã tín hiệu thu để khôi phục dữ liệu thư điện tử và xuất dữ liệu thư điện tử đã được phục hồi thu được đến bộ điều khiển 931. Bộ điều khiển 931 khiến bộ hiển thị 930 hiển thị nội dung của thư điện tử và lưu giữ dữ liệu thư điện tử trong phương tiện lưu giữ của bộ ghi/tái tạo 929.

Bộ ghi/tái tạo 929 bao gồm phương tiện lưu giữ đọc và ghi được tùy ý. Ví dụ, phương tiện lưu giữ có thể là phương tiện lưu giữ lắp sẵn bên trong như RAM và bộ nhớ tia chớp hoặc có thể là phương tiện lưu giữ loại lắp bên ngoài như đĩa cứng, đĩa từ, đĩa từ quang, đĩa quang, bộ nhớ buýt nối tiếp vạn năng (Universal Serial Bus, viết tắt là USB), hoặc thẻ nhớ.

Trong chế độ tạo ảnh, ví dụ, bộ camera 926 tạo ảnh đối tượng để tạo dữ liệu ảnh và xuất dữ liệu ảnh đã được tạo ra đến bộ xử lý ảnh 927. Bộ xử lý ảnh 927 mã hóa dữ liệu ảnh được nhập từ bộ camera 926 và lưu giữ luồng đã được mã hóa trong phương tiện lưu giữ của bộ ghi/tái tạo 929.

Ngoài ra, trong chế độ điện thoại truyền hình, ví dụ, bộ dồn kênh/tách 928 dồn kênh luồng video được mã hóa bởi bộ xử lý ảnh 927 và luồng audio được nhập từ bộ mã hóa giải mã audio 923 và xuất luồng đã được dồn kênh thu được đến bộ truyền thông 922. Bộ truyền thông 922 mã hóa và điều biến luồng, bằng cách đó tạo tín hiệu truyền. Sau đó, bộ truyền thông 922 truyền tín hiệu truyền đã được tạo ra đến trạm cơ sở (không được minh họa trên hình vẽ) qua anten 921. Ngoài ra, bộ truyền thông 922 thực hiện khuếch đại và chuyển đổi tần số

đối với tín hiệu vô tuyến thu được qua anten 921, bằng cách đó thu nhận tín hiệu thu. Tín hiệu truyền và tín hiệu thu thu nhận được với luồng bit đã được mã hóa nằm trong. Sau đó, bộ truyền thông 922 khôi phục luồng bằng cách giải điều biến và giải mã tín hiệu thu và xuất luồng đã được khôi phục đến bộ dồn kênh/tách 928. Bộ dồn kênh/tách 928 tách luồng video và luồng audio từ luồng đầu vào và xuất luồng video và luồng audio lần lượt đến bộ xử lý ảnh 927 và bộ mã hóa giải mã audio 923. Bộ xử lý ảnh 927 giải mã luồng video để tạo dữ liệu video. Dữ liệu video được cấp đến bộ hiển thị 930, và chuỗi các ảnh được hiển thị bởi bộ hiển thị 930. Bộ mã hóa giải mã audio 923 thực hiện giải nén và chuyển đổi D/A đối với luồng audio, bằng cách đó tạo tín hiệu audio tương tự. Sau đó, bộ mã hóa giải mã audio 923 cấp tín hiệu audio đã được tạo ra đến loa 924 để khiến audio được xuất.

Trong điện thoại di động 920 được tạo cấu hình theo cách này, bộ xử lý ảnh 927 có các chức năng của thiết bị mã hóa ảnh (ví dụ, thiết bị mã hóa được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 được minh họa trên Fig.1) và thiết bị giải mã ảnh (ví dụ, thiết bị giải mã được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52 được minh họa trên Fig.8) theo các phương án được mô tả trên đây. Do đó, khi mã hóa/giải mã ảnh mà được thực hiện bằng điện thoại di động 920, gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng khi được lưu giữ trong tệp được giải mã có thể được giảm.

Ví dụ ứng dụng thứ ba: Thiết bị ghi/tái tạo

Fig.44 là sơ đồ khối minh họa ví dụ về cấu hình giản lược của thiết bị ghi/tái tạo mà được áp dụng phương án được mô tả trên đây. Thiết bị ghi/tái tạo 940, ví dụ, mã hóa dữ liệu audio và dữ liệu video của chương trình phát rộng đã thu được và ghi dữ liệu đã được mã hóa trên phương tiện ghi. Ngoài ra, thiết bị ghi/tái tạo 940, ví dụ, có thể mã hóa dữ liệu audio và dữ liệu video được thu nhận từ thiết bị khác và ghi dữ liệu đã được mã hóa trên phương tiện ghi. Hơn thế nữa, thiết bị ghi/tái tạo 940, ví dụ, tái tạo dữ liệu được ghi trên phương tiện ghi bằng cách sử dụng màn hình và loa theo chỉ lệnh của người dùng. Lúc đó, thiết bị ghi/tái tạo 940 giải mã dữ liệu audio và dữ liệu video.

Thiết bị ghi/tái tạo 940 bao gồm bộ điều hướng 941, giao diện ngoài 942,

bộ mã hóa 943, ổ đĩa cứng (Hard Disk Drive, viết tắt là HDD) 944, ổ đĩa 945, bộ lựa chọn 946, bộ giải mã 947, hiển thị trên màn hình (On-Screen Display, viết tắt là OSD) 948, bộ điều khiển 949, và giao diện người dùng 950.

Bộ điều hướng 941 trích tín hiệu của kênh mong muốn từ tín hiệu phát rộng được thu qua anten (không được minh họa trên hình vẽ) và giải điều biến tín hiệu đã được trích. Sau đó, bộ điều hướng 941 xuất luồng bit đã được mã hóa được thu nhận bởi xử lý giải điều biến đến bộ lựa chọn 946. Nói cách khác, bộ điều hướng 941 có chức năng làm phương tiện truyền của thiết bị ghi/tái tạo 940.

Giao diện ngoài 942 là giao diện được sử dụng để nối thiết bị ghi/tái tạo 940 và thiết bị bên ngoài hoặc mạng. Giao diện ngoài 942, ví dụ, có thể là giao diện IEEE 1394, giao diện mạng, giao diện USB, giao diện bộ nhớ tia chớp hoặc tương tự. Ví dụ, dữ liệu video và dữ liệu audio được thu qua giao diện ngoài 942 được nhập vào bộ mã hóa 943. Nói cách khác, giao diện ngoài 942 có chức năng làm phương tiện truyền của thiết bị ghi/tái tạo 940.

Trong trường hợp mà dữ liệu video và dữ liệu audio được nhập từ giao diện ngoài 942 không được mã hóa, bộ mã hóa 943 mã hóa dữ liệu video và dữ liệu audio. Sau đó, bộ mã hóa 943 xuất luồng bit đã được mã hóa đến bộ lựa chọn 946.

HDD 944 ghi luồng bit đã được mã hóa trong đó dữ liệu nội dung như video và audio được nén, các chương trình khác nhau, và dữ liệu khác trong đĩa cứng bên trong. Khi video và audio được tái tạo, HDD 944 đọc dữ liệu của chúng từ đĩa cứng.

Ổ đĩa 945 ghi dữ liệu trên phương tiện ghi đã được lắp và đọc dữ liệu phương tiện ghi đã được lắp. Phương tiện ghi được lắp vào ổ đĩa 945, ví dụ, có thể là đĩa DVD (DVD-Video, DVD-RAM, DVD-R, DVD-RW, DVD+R, DVD+RW, hoặc tương tự), đĩa Blu-ray (nhãn hiệu đã được đăng ký), hoặc tương tự.

Khi video và audio được ghi, bộ lựa chọn 946 lựa chọn luồng bit đã được mã hóa được nhập từ bộ điều hướng 941 hoặc bộ mã hóa 943 và xuất luồng bit

đã được mã hóa được lựa chọn đến HDD 944 hoặc ổ đĩa 945. Ngoài ra, khi video và audio được tái tạo, bộ lựa chọn 946 xuất luồng bit đã được mã hóa được nhập từ HDD 944 hoặc ổ đĩa 945 đến bộ giải mã 947.

Bộ giải mã 947 giải mã luồng bit đã được mã hóa để tạo dữ liệu video và dữ liệu audio. Sau đó, bộ giải mã 947 xuất dữ liệu video đã được tạo ra đến OSD 948. Ngoài ra, bộ giải mã 947 xuất dữ liệu audio đã được tạo ra đến loa ngoài.

OSD 948 tái tạo dữ liệu video được nhập từ bộ giải mã 947, bằng cách đó hiển thị video. OSD 948 có thể chồng ảnh của GUI như thực đơn, nút, con trỏ, hoặc tương tự trên video được hiển thị.

Bộ điều khiển 949 bao gồm bộ xử lý như CPU và bộ nhớ như RAM hoặc ROM. Bộ nhớ lưu giữ chương trình được thực hiện bởi CPU, dữ liệu chương trình, và tương tự. Chương trình được lưu giữ trong bộ nhớ, ví dụ, được đọc và thực hiện bởi CPU khi kích hoạt thiết bị ghi/tái tạo 940. CPU điều khiển hoạt động của thiết bị ghi/tái tạo 940, ví dụ, theo tín hiệu hoạt động được nhập từ giao diện người dùng 950 bằng cách thực hiện chương trình.

Giao diện người dùng 950 được nối với bộ điều khiển 949. Giao diện người dùng 950, ví dụ, bao gồm nút và chuyển mạch để người dùng vận hành thiết bị ghi/tái tạo 940 và bộ thu dùng cho tín hiệu điều khiển từ xa. Giao diện người dùng 950 dò thao tác của người dùng qua các bộ phận cấu thành để tạo tín hiệu hoạt động và xuất hoạt động đã được tạo ra đến bộ điều khiển 949.

Trong thiết bị ghi/tái tạo 940 được tạo cấu hình theo cách này, bộ mã hóa 943 có chức năng của thiết bị mã hóa ảnh (ví dụ, thiết bị mã hóa được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 được minh họa trên Fig.1) theo phương án được mô tả trên đây. Ngoài ra, bộ giải mã 947 có chức năng của thiết bị giải mã ảnh (ví dụ, thiết bị giải mã được tạo cấu hình bởi bộ đọc tệp 51 và bộ giải mã 52 được minh họa trên Fig.8) theo phương án được mô tả trên đây. Do đó, khi mã hóa/giải mã ảnh mà được thực hiện bởi thiết bị ghi/tái tạo 940, gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng khi được lưu giữ trong tệp được giải mã có thể được giảm.

Ví dụ ứng dụng thứ tư: Thiết bị tạo ảnh

Fig.45 minh họa ví dụ về cấu hình giản lược của thiết bị tạo ảnh mà được áp dụng phương án được mô tả trên đây. Thiết bị tạo ảnh 960 tạo ảnh đối tượng để tạo ảnh, mã hóa dữ liệu ảnh, và ghi dữ liệu ảnh đã được mã hóa trên phương tiện ghi.

Thiết bị tạo ảnh 960 bao gồm khối quang 961, bộ tạo ảnh 962, bộ xử lý tín hiệu 963, bộ xử lý ảnh 964, bộ hiển thị 965, giao diện ngoài 966, bộ nhớ 967, ổ chứa phương tiện 968, OSD 969, bộ điều khiển 970, giao diện người dùng 971, và buýt 972.

Khối quang 961 được nối với bộ tạo ảnh 962. Bộ tạo ảnh 962 được nối với bộ xử lý tín hiệu 963. Bộ hiển thị 965 được nối với bộ xử lý ảnh 964. Giao diện người dùng 971 được nối với bộ điều khiển 970. Buýt 972 nối bộ xử lý ảnh 964, giao diện ngoài 966, bộ nhớ 967, ổ chứa phương tiện 968, OSD 969, và bộ điều khiển 970 với nhau.

Khối quang 961 bao gồm thấu kính điều tiêu, cơ cấu màn chắn, và tương tự. Khối quang 961 tạo thành ảnh quang của đối tượng trên bề mặt tạo ảnh của bộ tạo ảnh 962. Bộ tạo ảnh 962 bao gồm cảm biến ảnh chẳng hạn như thiết bị ghép nối điện tích (Charge Coupled Device, viết tắt là CCD) và bán dẫn oxit kim loại bù (Complementary Metal Oxide Semiconductor, viết tắt là CMOS) và chuyển đổi ảnh quang được tạo ra trên bề mặt tạo ảnh thành tín hiệu ảnh như là tín hiệu điện qua chuyển đổi quang điện. Sau đó, bộ tạo ảnh 962 xuất tín hiệu ảnh đến bộ xử lý tín hiệu 963.

Bộ xử lý tín hiệu 963 thực hiện các xử lý tín hiệu camera khác nhau như hiệu chỉnh khớp xoay, hiệu chỉnh gama, hiệu chỉnh màu, và tương tự đối với tín hiệu ảnh được nhập từ bộ tạo ảnh 962. Bộ xử lý tín hiệu 963 xuất dữ liệu ảnh sau các xử lý tín hiệu camera đến bộ xử lý ảnh 964.

Bộ xử lý ảnh 964 mã hóa dữ liệu ảnh được nhập từ bộ xử lý tín hiệu 963 để tạo dữ liệu đã được mã hóa. Sau đó, bộ xử lý ảnh 964 xuất dữ liệu đã được mã hóa đã được tạo ra đến giao diện ngoài 966 hoặc ổ chứa phương tiện 968. Ngoài ra, bộ xử lý ảnh 964 giải mã dữ liệu đã được mã hóa được nhập từ giao diện ngoài 966 hoặc ổ chứa phương tiện 968 để tạo dữ liệu ảnh. Sau đó, bộ xử lý ảnh 964 xuất dữ liệu ảnh đã được tạo ra đến bộ hiển thị 965. Ngoài ra, bộ xử lý

ảnh 964 có thể xuất dữ liệu ảnh được nhập từ bộ xử lý tín hiệu 963 đến bộ hiển thị 965 để hiển thị ảnh. Hơn thế nữa, bộ xử lý ảnh 964 có thể chồng dữ liệu để hiển thị mà thu nhận được từ OSD 969 trên ảnh được xuất đến bộ hiển thị 965.

OSD 969, ví dụ, tạo ảnh của GUI như thực đơn, nút, con trỏ, hoặc tương tự và xuất ảnh đã được tạo ra đến bộ xử lý ảnh 964.

Giao diện ngoài 966, ví dụ, được tạo cấu hình như là thiết bị đầu cuối nhập/xuất USB. Giao diện ngoài 966, ví dụ, nối thiết bị tạo ảnh 960 và máy in khi ảnh được in. Ngoài ra, ổ đĩa được nối với giao diện ngoài 966 nếu cần thiết. Phương tiện tháo lắp được như đĩa từ hoặc đĩa quang được lắp vào ổ đĩa, và chương trình được đọc từ phương tiện tháo lắp được có thể được cài vào thiết bị tạo ảnh 960. Hơn thế nữa, giao diện ngoài 966 có thể được tạo cấu hình như là giao diện mạng mà được nối với mạng như LAN, Internet, hoặc tương tự. Nói cách khác, giao diện ngoài 966 có chức năng làm phương tiện truyền của thiết bị tạo ảnh 960.

Phương tiện ghi được lắp vào ổ chứa phương tiện 968, ví dụ, có thể là phương tiện tháo lắp được ghi và đọc được tùy ý như đĩa từ, đĩa từ quang, đĩa quang, hoặc bộ nhớ bán dẫn. Ngoài ra, nó có thể được tạo cấu hình để phương tiện ghi được lắp cố định vào ổ chứa phương tiện 968 để tạo cấu hình bộ lưu giữ không di động như ổ đĩa cứng lắp sẵn bên trong hoặc ổ đĩa trạng thái rắn (Solid State Drive, viết tắt là SSD).

Bộ điều khiển 970 bao gồm bộ xử lý như CPU và bộ nhớ như RAM hoặc ROM. Bộ nhớ lưu giữ chương trình được thực hiện bởi CPU, dữ liệu chương trình, và tương tự. Chương trình được lưu giữ trong bộ nhớ được đọc bởi CPU, ví dụ, khi kích hoạt thiết bị tạo ảnh 960 và được thực hiện. CPU điều khiển hoạt động của thiết bị tạo ảnh 960, ví dụ, theo tín hiệu hoạt động được nhập từ giao diện người dùng 971 bằng cách thực hiện chương trình.

Giao diện người dùng 971 được nối với bộ điều khiển 970. Giao diện người dùng 971, ví dụ, bao gồm các nút, chuyển mạch, và tương tự để người dùng vận hành thiết bị tạo ảnh 960. Giao diện người dùng 971 dò thao tác của người dùng qua các bộ phận cấu thành như vậy, tạo tín hiệu hoạt động, và xuất tín hiệu hoạt động đã được tạo ra đến bộ điều khiển 970.

Trong thiết bị tạo ảnh 960 được tạo cấu hình như vậy, bộ xử lý ảnh 964 có các chức năng của thiết bị mã hóa ảnh (ví dụ, thiết bị mã hóa được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 được minh họa trên Fig.1) và thiết bị giải mã ảnh (ví dụ, thiết bị giải mã được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52 được minh họa trên Fig.8) theo các phương án được mô tả trên đây. Do đó, khi mã hóa/giải mã ảnh mà được thực hiện bởi thiết bị tạo ảnh 960, gánh nặng xử lý được yêu cầu khi luồng được sử dụng tại thời gian truyền hoặc luồng được sử dụng khi được lưu giữ trong tệp được giải mã có thể được giảm.

8. Phương án thứ bảy

Các ví dụ khác

Mặc dù các ví dụ về thiết bị, hệ thống, và tương tự được áp dụng sáng chế đã được mô tả trên đây, tuy nhiên, sáng chế không bị giới hạn như vậy. Do vậy, sáng chế cũng có thể được thực hiện như là tất cả các bộ phận được lắp trong thiết bị hoặc thiết bị tạo thành hệ thống như bộ xử lý như tích hợp cỡ lớn (Large Scale Integration, viết tắt là LSI) hệ thống hoặc tương tự, môđun sử dụng các bộ xử lý và tương tự, bộ phận sử dụng các môđun và tương tự, và tập hợp thu nhận được bằng cách bổ sung chức năng khác vào bộ phận (nói cách khác, một phần của cấu hình của thiết bị).

Tập video

Ví dụ về trường hợp mà sáng chế được thực hiện như là một tập hợp sẽ được mô tả có dựa vào Fig.46. Fig.46 minh họa ví dụ về cấu hình giản lược của tập video được áp dụng sáng chế.

Gần đây, việc thực hiện đa chức năng trong các thiết bị điện tử đang tiến triển, và trong quá trình phát triển hoặc sản xuất từng thiết bị điện tử, trong trường hợp mà một phần của cấu hình được thực hiện khi bán, cung cấp, hoặc tương tự, một cách thường xuyên, không chỉ có trường hợp mà một phần được thực hiện như là cấu hình có một chức năng mà cả trường hợp mà một phần được thực hiện như là một tập hợp có đa chức năng bằng cách kết hợp các cấu hình có các chức năng liên quan.

Tập video 1300 được minh họa trên Fig.46 có cấu hình mà có đa chức

năng và thu nhận được bằng cách kết hợp thiết bị có chức năng liên quan đến mã hóa/giải mã (một trong số mã hóa và giải mã hoặc cả hai) đối với ảnh với các thiết bị có các chức năng khác liên quan đến chức năng.

Như được minh họa trên Fig.46, tập video 1300 bao gồm: nhóm môđun như môđun video 1311, bộ nhớ ngoài 1312, môđun quản lý công suất 1313, và môđun đầu trước 1314 và các thiết bị có các chức năng liên quan như đầu nối 1321, camera 1322, và cảm biến 1323.

Môđun là bộ phận mà có các chức năng có sự gắn kết thu nhận được bằng cách tập hợp một vài chức năng thành phần liên quan với nhau. Cấu hình vật lý cụ thể là tùy ý, và ví dụ, cấu hình có thể được cân nhắc trong đó các bộ xử lý có các chức năng tương ứng, các thành phần mạch điện tử như điện trở và tụ điện, và các thiết bị khác được bố trí để được tích hợp trong bản mạch dây nối hoặc tương tự. Ngoài ra, có thể cân nhắc việc tạo môđun mới bằng cách kết hợp môđun với các môđun khác, bộ xử lý, hoặc tương tự.

Trong trường hợp của ví dụ được minh họa trên Fig.46, môđun video 1311 được tạo thành bằng cách kết hợp các bộ phận cấu hình có các chức năng liên quan đến xử lý ảnh và bao gồm: bộ xử lý ứng dụng, bộ xử lý video, môđem dải rộng 1333, và môđun RF 1334.

Bộ xử lý được tạo thành bằng cách tích hợp bộ phận cấu hình có chức năng định trước trên chip bán dẫn qua hệ thống trên chip (System On a Chip, viết tắt là SoC), và ví dụ, có bộ xử lý gọi là LSI (tích hợp cỡ lớn) hoặc tương tự. Cấu hình có chức năng định trước có thể là mạch logic (cấu hình phần cứng), cấu hình bao gồm CPU, ROM, RAM, và tương tự và chương trình (cấu hình phần mềm) được thực hiện bằng cách sử dụng các thành phần, hoặc cấu hình thu nhận được bằng cách kết hợp cả hai. Ví dụ, có thể tạo cấu hình để bộ xử lý bao gồm các mạch logic, CPU, ROM, RAM, và tương tự, một số chức năng của chúng được thực hiện bằng các mạch logic (cấu hình phần cứng), và các chức năng khác được thực hiện bằng chương trình (cấu hình phần mềm) được thực hiện bởi CPU.

Bộ xử lý ứng dụng 1331 được minh họa trên Fig.46 là bộ xử lý mà thực hiện ứng dụng liên quan đến xử lý ảnh. Để thực hiện chức năng định trước, ứng

dụng được thực hiện bởi bộ xử lý ứng dụng 1331 này không chỉ thực hiện xử lý tính toán mà còn có thể điều khiển các bộ phận cấu hình ở bên trong/bên ngoài môđun video 1311 như bộ xử lý video 1332 và tương tự nếu cần thiết.

Bộ xử lý video 1332 là bộ xử lý mà có chức năng liên quan đến mã hóa/giải mã (một trong số mã hóa và giải mã hoặc cả mã hóa và giải mã) đối với ảnh.

Môđem dải rộng 1333 là bộ xử lý (hoặc môđun) mà xử lý liên quan đến truyền thông dải rộng hữu tuyến hoặc vô tuyến (hoặc hữu tuyến hoặc vô tuyến) mà được thực hiện qua các đường truyền thông dải rộng như Internet hoặc mạng chuyển mạch điện thoại công cộng. Ví dụ, môđem dải rộng 1333 chuyển đổi dữ liệu (tín hiệu số) sẽ được truyền thành tín hiệu tương tự qua xử lý điều biến số hoặc tương tự hoặc chuyển đổi tín hiệu tương tự thu được thành dữ liệu (tín hiệu số) qua xử lý giải điều biến. Ví dụ, môđem dải rộng 1333 có thể thực hiện điều biến/giải điều biến số đối với thông tin tùy ý như dữ liệu ảnh được xử lý bởi bộ xử lý video 1332, luồng trong đó dữ liệu ảnh được mã hóa, chương trình ứng dụng, và dữ liệu thiết lập.

Môđun RF 1334 là môđun mà thực hiện chuyển đổi tần số, điều biến/giải điều biến, khuếch đại, xử lý lọc, và tương tự đối với tín hiệu tần số radio (Radio Frequency, viết tắt là RF) mà được truyền hoặc thu qua anten. Ví dụ, môđun RF 1334 thực hiện chuyển đổi tần số và tương tự đối với tín hiệu dải gốc được tạo ra bằng môđem dải rộng 1333, bằng cách đó tạo tín hiệu RF. Ngoài ra, ví dụ, môđun RF 1334 thực hiện chuyển đổi tần số và tương tự đối với tín hiệu RF thu được qua môđun đầu trước 1314, bằng cách đó tạo tín hiệu dải gốc.

Như được biểu thị bằng đường chấm 1341 trên Fig.46, bộ xử lý ứng dụng 1331 và bộ xử lý video 1332 có thể được tích hợp để được tạo thành như là một bộ xử lý.

Bộ nhớ ngoài 1312 là môđun mà được bố trí bên ngoài môđun video 1311 và có thiết bị nhớ được sử dụng bởi môđun video 1311. Trong khi thiết bị nhớ của bộ nhớ ngoài 1312 có thể được thực hiện bởi cấu hình vật lý nhất định, nói chung, thiết bị nhớ thường được sử dụng để lưu giữ dữ liệu có khối lượng lớn như dữ liệu ảnh được tạo thành theo các đơn vị khung. Do đó, tốt hơn là thiết bị

nhớ được thực hiện bởi bộ nhớ bán dẫn có điện dung lớn như bộ nhớ truy cập ngẫu nhiên động (Dynamic Random Access Memory, viết tắt là DRAM) với giá thành tương đối thấp.

Môđun quản lý công suất 1313 quản lý và điều khiển việc cấp điện đến môđun video 1311 (từng bộ phận cấu hình nằm bên trong môđun video 1311).

Môđun đầu trước 1314 là môđun mà cung cấp chức năng đầu trước (mạch tại đầu truyền/đầu thu ở phía anten) cho môđun RF 1334. Như được minh họa trên Fig.46, môđun đầu trước 1314, ví dụ, bao gồm bộ anten 1351, bộ lọc 1352, và bộ khuếch đại 1353.

Bộ anten 1351 bao gồm anten mà truyền và thu các tín hiệu vô tuyến và các bộ phận cấu hình bên ngoài. Bộ anten 1351 truyền tín hiệu được cấp từ bộ khuếch đại 1353 như là tín hiệu vô tuyến và cấp tín hiệu vô tuyến thu được đến bộ lọc 1352 như là tín hiệu điện (tín hiệu RF). Bộ lọc 1352 thực hiện xử lý lọc và tương tự đối với tín hiệu RF thu được qua bộ anten 1351 và cấp tín hiệu RF sau xử lý đến môđun RF 1334. Bộ khuếch đại 1353 khuếch đại tín hiệu RF được cấp từ môđun RF 1334 và cấp tín hiệu RF đã được khuếch đại đến bộ anten 1351.

Đầu nối 1321 là môđun có chức năng liên quan đến việc kết nối với bên ngoài. Cấu hình vật lý của đầu nối 1321 là tùy ý. Ví dụ, đầu nối 1321 bao gồm bộ phận cấu hình có chức năng truyền thông theo chuẩn truyền thông không phải là chuẩn truyền thông mà môđem dải rộng 1333 tương thích, thiết bị đầu cuối nhập/xuất bên ngoài, và tương tự.

Ví dụ, đầu nối 1321 có thể được tạo cấu hình bao gồm môđun có chức năng truyền thông tương thích với chuẩn truyền thông radio như Bluetooth (nhãn hiệu đã được đăng ký), IEEE 802.11 (ví dụ, Wi-Fi (Wireless Fidelity; nhãn hiệu đã được đăng ký)), truyền thông trường gần (Near Field Communication, viết tắt là NFC), hoặc kết hợp dữ liệu hồng ngoại (Infrared Data Association, viết tắt là IrDA), anten truyền và thu các tín hiệu tương thích với chuẩn, và tương tự. Ngoài ra, ví dụ, đầu nối 1321 có thể được tạo cấu hình bao gồm môđun có chức năng truyền thông tương thích với chuẩn truyền thông vô tuyến như buýt nối tiếp vạn năng (Universal Serial Bus, USB) hoặc giao diện

đa phương tiện độ nét cao (High-Definition Multimedia Interface, viết tắt là HDMI) (nhãn hiệu đã được đăng ký) và các thiết bị đầu cuối tương thích với chuẩn. Hơn thế nữa, ví dụ, đầu nối 1321 có thể được tạo cấu hình để có chức năng truyền dữ liệu (tín hiệu) khác của thiết bị đầu cuối nhập/xuất tương tự hoặc tương tự.

Ngoài ra, đầu nối 1321 có thể được tạo cấu hình bao gồm thiết bị của đích truyền dữ liệu (tín hiệu). Ví dụ, đầu nối 1321 có thể được tạo cấu hình bao gồm ổ đĩa (bao gồm không chỉ ổ đĩa của phương tiện tháo lắp được mà cả đĩa cứng, ổ đĩa trạng thái rắn (Solid State Drive, viết tắt là SSD), lưu trữ gắn mạng (Network Attached Storage, viết tắt là NAS), và tương tự) mà đọc/ghi dữ liệu từ/vào phương tiện ghi như đĩa từ, đĩa quang, đĩa từ quang, hoặc bộ nhớ bán dẫn. Ngoài ra, đầu nối 1321 có thể được tạo cấu hình bao gồm thiết bị đầu ra (màn hình, loa, hoặc tương tự) của ảnh hoặc tiếng nói.

Camera 1322 là môđun có chức năng thu nhận dữ liệu ảnh của đối tượng bằng cách tạo ảnh đối tượng. Dữ liệu ảnh thu nhận được bằng xử lý tạo ảnh được thực hiện bằng camera 1322, ví dụ, được cấp đến bộ xử lý video 1332 và được mã hóa.

Cảm biến 1323 là môđun có chức năng của cảm biến tùy ý như cảm biến âm thanh, cảm biến siêu âm, cảm biến quang, cảm biến độ chói, cảm biến hồng ngoại, cảm biến ảnh, cảm biến quay, cảm biến góc, cảm biến vận tốc góc, cảm biến vận tốc, cảm biến gia tốc, cảm biến độ nghiêng, cảm biến nhận dạng từ, cảm biến va đập, hoặc cảm biến nhiệt độ. Dữ liệu dò được bởi cảm biến 1323, ví dụ, được cấp đến bộ xử lý ứng dụng 1331 và được sử dụng bởi cảm biến ứng dụng và tương tự.

Cấu hình được mô tả trên đây như là môđun có thể được thực hiện như là bộ xử lý. Ngược lại, cấu hình được mô tả trên đây như là bộ xử lý có thể được thực hiện như là môđun.

Trong tập video 1300 có cấu hình được mô tả trên đây, như sẽ được mô tả sau, sáng chế có thể áp dụng cho bộ xử lý video 1332. Do đó, tập video 1300 có thể được thực hiện như là tập hợp được áp dụng sáng chế.

Ví dụ về cấu hình của bộ xử lý video

Fig.47 minh họa ví dụ về cấu hình giản lược của bộ xử lý video 1332 (Fig.46) được áp dụng sáng chế.

Trong trường hợp của ví dụ được minh họa trên Fig.47, bộ xử lý video 1332 có chức năng thu các đầu vào của tín hiệu video và tín hiệu audio và mã hóa tín hiệu video và tín hiệu audio theo hệ thống định trước và chức năng giải mã dữ liệu video và dữ liệu audio đã được mã hóa và tái tạo và xuất tín hiệu video và tín hiệu audio.

Như được minh họa trên Fig.47, bộ xử lý video 1332 bao gồm: bộ xử lý đầu vào video 1401; bộ phóng to/thu nhỏ ảnh thứ nhất 1402; bộ phóng to/thu nhỏ ảnh thứ hai 1403; bộ xử lý đầu ra video 1404; bộ nhớ khung 1405; và bộ điều khiển bộ nhớ 1406. Ngoài ra, bộ xử lý video 1332 bao gồm: cỗ máy mã hóa/giải mã 1407; bộ đệm luồng cơ bản (Elementary Stream, viết tắt là ES) video 1408A và 1408B; và các bộ đệm ES audio 1409A và 1409B. Hơn thế nữa, bộ xử lý video 1332 bao gồm: bộ mã hóa audio 1410; bộ giải mã audio 1411; bộ dồn kênh (Multiplexer, viết tắt là MUX) 1412; bộ phân kênh (Demultiplexer, viết tắt là DMUX) 1413; và bộ đệm luồng 1414.

Bộ xử lý đầu vào video 1401, ví dụ, thu nhận tín hiệu video được nhập từ đầu nối 1321 (Fig.46) hoặc tương tự và chuyển đổi tín hiệu video thành dữ liệu ảnh số. Bộ phóng to/thu nhỏ ảnh thứ nhất 1402 thực hiện chuyển đổi định dạng, xử lý phóng to/thu nhỏ ảnh, và tương tự đối với dữ liệu ảnh. Bộ phóng to/thu nhỏ ảnh thứ hai 1403 thực hiện xử lý phóng to/thu nhỏ ảnh theo định dạng của đích đầu ra qua bộ xử lý đầu ra video 1404, cùng chuyển đổi định dạng như của bộ phóng to/thu nhỏ ảnh thứ nhất 1402, xử lý phóng to/thu nhỏ ảnh, và tương tự đối với dữ liệu ảnh. Bộ xử lý đầu ra video 1404 thực hiện chuyển đổi định dạng, chuyển đổi thành tín hiệu tương tự, và tương tự đối với dữ liệu ảnh và xuất tín hiệu thu được, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự như là tín hiệu video được tái tạo.

Bộ nhớ khung 1405 là bộ nhớ dùng cho dữ liệu ảnh mà được chia sẻ bởi bộ xử lý đầu vào video 1401, bộ phóng to/thu nhỏ ảnh thứ nhất 1402, bộ phóng to/thu nhỏ ảnh thứ hai 1403, bộ xử lý đầu ra video 1404, và cỗ máy mã hóa/giải

mã 1407. Bộ nhớ khung 1405, ví dụ, được thực hiện bởi bộ nhớ bán dẫn như DRAM.

Bộ điều khiển bộ nhớ 1406 thu tín hiệu đồng bộ từ cỗ máy mã hóa/giải mã 1407 và điều khiển việc truy cập vào bộ nhớ khung 1405 để ghi/đọc theo lịch truy cập để truy cập bộ nhớ khung 1405 mà được ghi trong bảng quản lý truy cập 1406A. Bảng quản lý truy cập 1406A được cập nhật bởi bộ điều khiển bộ nhớ 1406 theo các xử lý được thực hiện bởi cỗ máy mã hóa/giải mã 1407, bộ phóng to/thu nhỏ ảnh thứ nhất 1402, bộ phóng to/thu nhỏ ảnh thứ hai 1403, và tương tự.

Cỗ máy mã hóa/giải mã 1407 thực hiện xử lý mã hóa đối với dữ liệu ảnh và xử lý giải mã đối với luồng video mà là dữ liệu thu nhận được bằng cách mã hóa dữ liệu ảnh. Ví dụ, cỗ máy mã hóa/giải mã 1407 mã hóa dữ liệu ảnh được đọc từ bộ nhớ khung 1405 và theo tuần tự ghi dữ liệu ảnh vào bộ đệm ES video 1408A như là luồng video. Ngoài ra, ví dụ, cỗ máy mã hóa/giải mã 1407 theo tuần tự đọc và giải mã các luồng video được cấp từ bộ đệm ES video 1408B và theo tuần tự ghi các luồng video đã được giải mã vào bộ nhớ khung 1405 như là dữ liệu ảnh. Cỗ máy mã hóa/giải mã 1407 sử dụng bộ nhớ khung 1405 như là vùng làm việc trong các xử lý mã hóa và giải mã này. Ngoài ra, cỗ máy mã hóa/giải mã 1407 xuất tín hiệu đồng bộ đến bộ điều khiển bộ nhớ 1406, ví dụ, tại thời gian khi xử lý đối với từng khối macro được bắt đầu.

Bộ đệm ES video 1408A lưu vào bộ đệm luồng video được tạo ra bằng cỗ máy mã hóa/giải mã 1407 và cấp luồng video đã được lưu vào bộ đệm đến bộ dồn kênh (MUX) 1412. Bộ đệm ES video 1408B lưu vào bộ đệm luồng video được cấp từ bộ phân kênh (DMUX) 1413 và cấp luồng video đã được lưu vào bộ đệm đến cỗ máy mã hóa/giải mã 1407.

Bộ đệm ES audio 1409A lưu vào bộ đệm luồng audio được tạo bởi bộ mã hóa audio 1410 và cấp luồng audio đã được lưu vào bộ đệm đến bộ dồn kênh (MUX) 1412. Bộ đệm ES audio 1409B lưu vào bộ đệm luồng audio được cấp từ bộ phân kênh (DMUX) 1413 và cấp luồng audio đã được lưu vào bộ đệm đến bộ giải mã audio 1411.

Bộ mã hóa audio 1410, ví dụ, chuyển đổi tín hiệu audio, ví dụ, được nhập

từ đầu nối 1321 (Fig.46) hoặc tương tự thành tín hiệu số và mã hóa tín hiệu số đã được chuyển đổi theo hệ thống định trước như hệ thống audio MPEG hoặc hệ thống mã audio số 3 (AudioCode number 3, viết tắt là AC3). Bộ mã hóa audio 1410 theo tuần tự ghi các luồng audio từng luồng này là dữ liệu thu nhận được bằng cách mã hóa tín hiệu audio vào bộ đệm ES audio 1409A. Bộ giải mã audio 1411 giải mã luồng audio được cấp từ bộ đệm ES audio 1409B, và ví dụ, thực hiện chuyển đổi thành tín hiệu tương tự, và tương tự đối với luồng audio đã được giải mã và cấp tín hiệu thu được, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự như là tín hiệu audio đã được tái tạo.

Bộ dồn kênh (MUX) 1412 dồn kênh luồng video và luồng audio. Phương pháp của xử lý dồn kênh này (nói cách khác, định dạng của luồng bit được tạo ra bởi xử lý dồn kênh) là tùy ý. Ngoài ra, trong xử lý dồn kênh, bộ dồn kênh (MUX) 1412 có thể bổ sung thông tin đoạn đầu định trước và tương tự vào luồng bit. Nói cách khác, bộ dồn kênh (MUX) 1412 có thể chuyển đổi định dạng của luồng qua xử lý dồn kênh. Ví dụ, bằng cách dồn kênh luồng video và luồng audio, bộ dồn kênh (MUX) 1412 chuyển đổi các luồng thành luồng vận chuyển mà là luồng bit của định dạng truyền. Ngoài ra, ví dụ, bằng cách dồn kênh luồng video và luồng audio, bộ dồn kênh (MUX) 1412 chuyển đổi các luồng thành dữ liệu (dữ liệu tệp) của định dạng tệp ghi.

Bộ phân kênh (DMUX) 1413 phân kênh luồng bit trong đó luồng video và luồng audio được dồn kênh bằng cách sử dụng phương pháp tương ứng với xử lý dồn kênh được thực hiện bởi bộ dồn kênh (MUX) 1412. Nói cách khác, bộ phân kênh (DMUX) 1413 trích luồng video và luồng audio từ luồng bit được đọc từ bộ đệm luồng 1414 (tách luồng video và luồng audio khỏi nhau). Nói cách khác, bộ phân kênh (DMUX) 1413 có thể chuyển đổi định dạng của luồng qua xử lý phân kênh (chuyển đổi ngược của chuyển đổi được thực hiện bởi bộ dồn kênh (MUX) 1412). Ví dụ, bộ phân kênh (DMUX) 1413 thu nhận luồng vận chuyển được cấp, ví dụ, từ đầu nối 1321 (Fig.46), môdem dải rộng 1333 (Fig.46), hoặc tương tự qua bộ đệm luồng 1414 và phân kênh luồng vận chuyển được cấp, bằng cách đó chuyển đổi luồng vận chuyển thành luồng video và luồng audio. Ngoài ra, ví dụ, bộ phân kênh (DMUX) 1413 thu nhận dữ liệu tệp,

ví dụ, được đọc từ các phương tiện ghi khác nhau bằng đầu nối 1321 (Fig.46) qua bộ đệm luồng 1414 và phân kênh dữ liệu tệp thu nhận được, bằng cách đó chuyển đổi dữ liệu tệp thành luồng video và luồng audio.

Bộ đệm luồng 1414 lưu vào bộ đệm luồng bit. Ví dụ, bộ đệm luồng 1414 lưu vào bộ đệm luồng vận chuyển được cấp từ bộ dồn kênh (MUX) 1412 và cấp luồng vận chuyển đã được lưu vào bộ đệm, ví dụ, đến đầu nối 1321 (Fig.46), môdem dải rộng 1333 (Fig.46), hoặc tương tự tại thời gian định trước hoặc dựa vào yêu cầu từ bên ngoài, hoặc tương tự.

Ngoài ra, ví dụ, bộ đệm luồng 1414 lưu vào bộ đệm dữ liệu tệp được cấp từ bộ dồn kênh (MUX) 1412 và cấp dữ liệu tệp đã được lưu vào bộ đệm, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự tại thời gian định trước, yêu cầu từ bên ngoài, hoặc tương tự để được ghi trên các phương tiện ghi khác nhau.

Hơn thế nữa, bộ đệm luồng 1414 lưu vào bộ đệm luồng vận chuyển, ví dụ, thu nhận được qua đầu nối 1321 (Fig.46), môdem dải rộng 1333 (Fig.46), hoặc tương tự và cấp luồng vận chuyển đã được lưu vào bộ đệm đến bộ phân kênh (DMUX) 1413 tại thời gian định trước hoặc dựa vào yêu cầu từ bên ngoài hoặc tương tự.

Ngoài ra, bộ đệm luồng 1414 lưu vào bộ đệm dữ liệu tệp được đọc từ các phương tiện ghi khác nhau bằng đầu nối 1321 (Fig.46) hoặc tương tự và cấp dữ liệu tệp đã được lưu vào bộ đệm đến bộ phân kênh (DMUX) 1413 tại thời gian định trước hoặc yêu cầu từ bên ngoài hoặc tương tự.

Tiếp theo, ví dụ về hoạt động của bộ xử lý video 1332 có cấu hình như vậy sẽ được mô tả. Ví dụ, tín hiệu video được nhập từ đầu nối 1321 (Fig.46) hoặc tương tự vào bộ xử lý video 1332 được chuyển đổi thành dữ liệu ảnh số của hệ thống định trước như hệ thống 4:2:2 Y/Cb/Cr trong bộ xử lý đầu vào video 1401 và theo tuần tự được ghi vào bộ nhớ khung 1405. Dữ liệu ảnh số này được đọc bởi bộ phóng to/thu nhỏ ảnh thứ nhất 1402 hoặc bộ phóng to/thu nhỏ ảnh thứ hai 1403, chuyển đổi định dạng thành hệ thống định trước như hệ thống 4:2:0 Y/Cb/Cr và xử lý phóng to/thu nhỏ được thực hiện đối với dữ liệu ảnh số đã được đọc, và dữ liệu ảnh số thu được được ghi lại vào bộ nhớ khung 1405. Dữ liệu ảnh này được mã hóa bằng cỗ máy mã hóa/giải mã 1407 và được ghi

vào bộ đệm ES video 1408A như là luồng video.

Ngoài ra, tín hiệu audio được nhập từ đầu nối 1321 (Fig.46) hoặc tương tự vào bộ xử lý video 1332 được mã hóa bằng bộ mã hóa audio 1410 và được ghi vào bộ đệm ES audio 1409A như là luồng audio.

Luồng video được lưu vào bộ đệm ES video 1408A và luồng audio được lưu vào bộ đệm ES audio 1409A được đọc và dồn kênh bằng bộ dồn kênh (MUX) 1412 và được chuyển đổi thành luồng vận chuyển, dữ liệu tệp, hoặc tương tự. Luồng vận chuyển được tạo ra bởi bộ dồn kênh (MUX) 1412 được lưu vào bộ đệm luồng 1414 và sau đó, được xuất đến mạng ngoài, ví dụ, qua đầu nối 1321 (Fig.46), môđem dải rộng 1333 (Fig.46), hoặc tương tự. Ngoài ra, dữ liệu tệp được tạo ra bởi bộ dồn kênh (MUX) 1412 được lưu vào bộ đệm luồng 1414 và sau đó, được xuất, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự và được ghi trên các phương tiện ghi khác nhau.

Ngoài ra, luồng vận chuyển được nhập vào bộ xử lý video 1332 từ mạng ngoài, ví dụ, qua đầu nối 1321 (Fig.46), môđem dải rộng 1333 (Fig.46), hoặc tương tự được lưu trong bộ đệm luồng 1414 và sau đó, được phân kênh bằng bộ phân kênh (DMUX) 1413. Ngoài ra, dữ liệu tệp được đọc từ các phương tiện ghi khác nhau và được nhập vào bộ xử lý video 1332, ví dụ, bằng đầu nối 1321 (Fig.46) hoặc tương tự được lưu trong bộ đệm luồng 1414 và sau đó, được phân kênh bằng bộ phân kênh (DMUX) 1413. Nói cách khác, luồng vận chuyển hoặc dữ liệu tệp được nhập vào bộ xử lý video 1332 được tách thành luồng video và luồng audio bằng bộ phân kênh (DMUX) 1413.

Luồng audio được cấp đến bộ giải mã audio 1411 qua bộ đệm ES audio 1409B và được giải mã, và tín hiệu audio được tái tạo. Ngoài ra, luồng video được ghi vào bộ đệm ES video 1408B và sau đó theo tuần tự được đọc và được giải mã bởi cỗ máy mã hóa/giải mã 1407 và được ghi vào bộ nhớ khung 1405. Dữ liệu ảnh đã được giải mã được xử lý để được phóng to hoặc thu nhỏ bởi bộ phóng to/thu nhỏ ảnh thứ hai 1403 và được ghi vào bộ nhớ khung 1405. Sau đó, dữ liệu ảnh đã được giải mã được đọc bởi bộ xử lý đầu ra video 1404, được chuyển đổi thành định dạng khác theo hệ thống định trước như hệ thống 4:2:2 Y/Cb/Cr hoặc tương tự, và được chuyển đổi tiếp thành tín hiệu tương tự, và tín

hiệu video được tái tạo và xuất.

Trong trường hợp mà sáng chế áp dụng cho bộ xử lý video 1332 được tạo cấu hình như vậy, sáng chế liên quan đến từng phương án được mô tả trên đây có thể áp dụng cho cỗ máy mã hóa/giải mã 1407. Nói cách khác, ví dụ, cỗ máy mã hóa/giải mã 1407 có thể được tạo cấu hình để có các chức năng của thiết bị mã hóa ảnh (Fig.1) được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 theo phương án thứ nhất và thiết bị giải mã ảnh (Fig.8) được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52. Bằng cách tạo cấu hình như vậy, bộ xử lý video 1332 có thể có các ưu điểm giống như được mô tả trên đây có dựa vào từ Fig.1 đến Fig.36.

Ngoài ra, trong cỗ máy mã hóa/giải mã 1407, sáng chế (nói cách khác, các chức năng của thiết bị mã hóa ảnh và thiết bị giải mã ảnh theo từng phương án được mô tả trên đây) có thể được thực hiện bởi phần cứng như các mạch logic hoặc phần mềm như chương trình được nhúng hoặc có thể được thực hiện bởi cả phần cứng và phần mềm.

Ví dụ khác về cấu hình của bộ xử lý video

Fig.48 minh họa ví dụ khác về cấu hình giản lược của bộ xử lý video 1332 (Fig.46) được áp dụng sáng chế. Trong trường hợp của ví dụ được minh họa trên Fig.48, bộ xử lý video 1332 có chức năng mã hóa và giải mã dữ liệu video theo hệ thống định trước.

Cụ thể hơn là, như được minh họa trên Fig.48, bộ xử lý video 1332 bao gồm: bộ điều khiển 1511; giao diện hiển thị 1512; cỗ máy hiển thị 1513; cỗ máy xử lý ảnh 1514; và bộ nhớ trong 1515. Ngoài ra, bộ xử lý video 1332 bao gồm: cỗ máy mã hóa/giải mã 1516; giao diện bộ nhớ 1517; bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518; giao diện mạng 1519; và giao diện video 1520.

Bộ điều khiển 1511 điều khiển các hoạt động của các bộ xử lý bên trong bộ xử lý video 1332 như giao diện hiển thị 1512, cỗ máy hiển thị 1513, cỗ máy xử lý ảnh 1514, và cỗ máy mã hóa/giải mã 1516.

Như được minh họa trên Fig.48, bộ điều khiển 1511, ví dụ, bao gồm CPU chính 1531, CPU phụ 1532, và bộ điều khiển hệ thống 1533. CPU chính 1531 thực hiện chương trình được sử dụng để điều khiển hoạt động của từng bộ xử lý

bên trong bộ xử lý video 1332 và tương tự. CPU 1531 tạo tín hiệu điều khiển theo chương trình hoặc tương tự và cấp tín hiệu điều khiển đến từng bộ xử lý (nói cách khác, điều khiển hoạt động của từng bộ xử lý). CPU 1532 đóng vai trò bổ sung của CPU chính 1531. Ví dụ, CPU phụ 1532 thiết lập xử lý con, thường trình con, hoặc tương tự của chương trình được thực hiện bởi CPU chính 1531. Bộ điều khiển hệ thống 1533 điều khiển các hoạt động của CPU chính 1531 và CPU phụ 1532 bằng cách thực hiện chỉ định các chương trình sẽ được thực hiện bởi CPU chính 1531 và CPU phụ 1532 và tương tự.

Giao diện hiển thị 1512 xuất dữ liệu ảnh, ví dụ, đến đầu nối 1321 (Fig.46) và tương tự theo sự điều khiển của bộ điều khiển 1511. Ví dụ, giao diện hiển thị 1512 chuyển đổi dữ liệu ảnh mà là dữ liệu số thành tín hiệu tương tự và xuất tín hiệu tương tự như là tín hiệu video đã được tái tạo hoặc dữ liệu ảnh mà là dữ liệu số đến thiết bị màn hình của đầu nối 1321 (Fig.46) hoặc tương tự.

Cỗ máy hiển thị 1513, theo sự điều khiển của bộ điều khiển 1511, thực hiện các xử lý chuyển đổi khác nhau như chuyển đổi định dạng, chuyển đổi kích thước, và chuyển đổi gam màu cho dữ liệu ảnh để so khớp đặc tả phần cứng của thiết bị màn hình hiển thị ảnh và tương tự.

Cỗ máy xử lý ảnh 1514 thực hiện xử lý ảnh định trước như xử lý lọc được sử dụng để tăng chất lượng ảnh và tương tự cho dữ liệu ảnh theo sự điều khiển của bộ điều khiển 1511.

Bộ nhớ trong 1515 là bộ nhớ được chia sẻ bởi cỗ máy hiển thị 1513, cỗ máy xử lý ảnh 1514, và cỗ máy mã hóa/giải mã 1516 và nằm bên trong bộ xử lý video 1332. Bộ nhớ trong 1515, ví dụ, được sử dụng để chuyển dữ liệu giữa cỗ máy hiển thị 1513, cỗ máy xử lý ảnh 1514, và cỗ máy mã hóa/giải mã 1516. Ví dụ, bộ nhớ trong 1515 lưu giữ dữ liệu được cấp từ cỗ máy hiển thị 1513, cỗ máy xử lý ảnh 1514, hoặc cỗ máy mã hóa/giải mã 1516 và cấp dữ liệu đến cỗ máy hiển thị 1513, cỗ máy xử lý ảnh 1514, hoặc cỗ máy mã hóa/giải mã 1516 nếu cần thiết (ví dụ, theo yêu cầu). Bộ nhớ trong 1515 này có thể được thực hiện bằng cách sử dụng thiết bị nhớ bất kỳ. Tuy nhiên, nói chung, bộ nhớ trong thường được sử dụng để lưu giữ dữ liệu có khối lượng nhỏ như dữ liệu ảnh theo các đơn vị khối và các thông số, và do đó, tốt hơn là thực hiện bộ nhớ trong

bằng cách sử dụng bộ nhớ bán dẫn có điện dung tương đối nhỏ (so với bộ nhớ ngoài 1312) và có đáp ứng cao như bộ nhớ truy cập ngẫu nhiên tĩnh (Static Random Access Memory, viết tắt là SRAM).

Cỗ máy mã hóa/giải mã 1516 thực hiện các xử lý liên quan đến mã hóa và giải mã dữ liệu ảnh. Hệ thống mã hóa/giải mã mà cỗ máy mã hóa/giải mã 1516 tương thích là tùy ý, và số lượng hệ thống mã hóa/giải mã có thể là một hoặc nhiều hơn. Ví dụ, có thể được tạo cấu hình để cỗ máy mã hóa/giải mã 1516 có thể có chức năng mã hóa giải mã đối với các hệ thống mã hóa/giải mã và được tạo cấu hình để thực hiện mã hóa dữ liệu ảnh hoặc giải mã dữ liệu đã được mã hóa bằng cách sử dụng một trong số các hệ thống mã hóa/giải mã đã được lựa chọn.

Trong ví dụ được minh họa trên Fig.48, cỗ máy mã hóa/giải mã 1516, ví dụ, bao gồm MPEG-2 Video 1541, AVC/H.264 1542, HEVC/H.265 1543, HEVC/H.265 (có thể mở rộng) 1544, HEVC/H.265 (đa cảnh) 1545, và MPEG-DASH 1551 như là các khối chức năng cho xử lý liên quan đến mã hóa giải mã.

MPEG-2 Video 1541 là khối chức năng mà mã hóa hoặc giải mã dữ liệu ảnh theo hệ thống MPEG-2. AVC/H.264 1542 là khối chức năng mà mã hóa hoặc giải mã dữ liệu ảnh theo hệ thống AVC. HEVC/H.265 1543 là khối chức năng mà mã hóa hoặc giải mã dữ liệu ảnh theo hệ thống HEVC. HEVC/H.265 (có thể mở rộng) 1544 là khối chức năng mà thực hiện mã hóa hoặc giải mã có thể thay đổi đối với dữ liệu ảnh theo hệ thống HEVC. HEVC/H. 265 (đa cảnh) 1545 là khối chức năng mà thực hiện mã hóa hoặc giải mã đa cảnh đối với dữ liệu ảnh theo hệ thống HEVC.

MPEG-DASH 1551 là khối chức năng mà truyền và thu dữ liệu ảnh theo hệ thống luồng thích ứng động MPEG trên HTTP (MPEG-Dynamic Adaptive Streaming (MPEG-DASH) over HTTP). MPEG-DASH là kỹ thuật thực hiện tạo luồng video bằng cách sử dụng giao thức truyền tải siêu văn bản (HyperText Transfer Protocol, viết tắt là HTTP), và một trong số các đặc trưng là dữ liệu đã được mã hóa thích hợp trong số các mẫu dữ liệu đã được mã hóa có độ phân giải khác nhau và tương tự, mà được chuẩn bị từ trước, được lựa chọn và truyền theo

các đơn vị đoạn. MPEG-DASH 1551 thực hiện tạo luồng mà tương thích với chuẩn, điều khiển truyền luồng, và tương tự và sử dụng MPEG-2 video 1541 đến HEVC/H.265 (đa cảnh) 1545 được mô tả trên đây để mã hóa và giải mã dữ liệu ảnh.

Giao diện bộ nhớ 1517 là giao diện được sử dụng cho bộ nhớ ngoài 1312. Dữ liệu được cấp từ cỗ máy xử lý ảnh 1514 hoặc cỗ máy mã hóa/giải mã 1516 được cấp đến bộ nhớ ngoài 1312 qua giao diện bộ nhớ 1517. Ngoài ra, dữ liệu được đọc từ bộ nhớ ngoài 1312 được cấp đến bộ xử lý video 1332 (cỗ máy xử lý ảnh 1514 hoặc cỗ máy mã hóa/giải mã 1516) qua giao diện bộ nhớ 1517.

Bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 thực hiện dồn kênh và phân kênh các loại dữ liệu khác nhau liên quan đến ảnh như luồng bit của dữ liệu đã được mã hóa, dữ liệu ảnh, và tín hiệu video. Phương pháp dồn kênh/phân kênh là tùy ý. Ví dụ, khi thực hiện dồn kênh, bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 có thể không bố trí các mẫu dữ liệu vào một mà cũng bổ sung thông tin đoạn đầu định trước hoặc tương tự vào dữ liệu. Ngoài ra, khi thực hiện phân kênh, bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 có thể không chỉ chia một mẫu dữ liệu thành các bộ phận mà cũng bổ sung thông tin đoạn đầu định trước hoặc tương tự vào từng bộ phận dữ liệu đã được chia. Nói cách khác, bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 có thể chuyển đổi định dạng của dữ liệu qua xử lý dồn kênh/phân kênh. Ví dụ, bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 có thể chuyển đổi luồng bit thành luồng vận chuyển mà là luồng bit của định dạng truyền hoặc dữ liệu (dữ liệu tệp) của định dạng tệp ghi bằng cách dồn kênh luồng bit. Thấy rõ là chuyển đổi ngược có thể được thực hiện bằng xử lý phân kênh.

Giao diện mạng 1519 là giao diện, ví dụ, chuyên dụng cho môđem dải rộng 1333 (Fig.46), đầu nối 1321 (Fig.46), hoặc tương tự. Giao diện video 1520 là giao diện, ví dụ, chuyên dụng cho đầu nối 1321 (Fig.46), camera 1322 (Fig.46), hoặc tương tự.

Tiếp theo, ví dụ về hoạt động của bộ xử lý video 1332 như vậy sẽ được mô tả. Ví dụ, khi luồng vận chuyển thu được từ mạng ngoài, ví dụ, qua đầu nối 1321 (Fig.46), môđem dải rộng 1333 (Fig.46), hoặc tương tự, luồng vận chuyển

được cấp đến bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 qua giao diện mạng 1519, được phân kênh, và được giải mã bởi cỗ máy mã hóa/giải mã 1516. Đối với dữ liệu ảnh thu nhận được bằng xử lý giải mã được thực hiện bằng cỗ máy mã hóa/giải mã 1516, xử lý ảnh định trước được thực hiện, ví dụ, bằng cỗ máy xử lý ảnh 1514, và chuyển đổi định trước được thực hiện bằng cỗ máy hiển thị 1513. Sau đó, dữ liệu ảnh thu được được cấp, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự qua giao diện hiển thị 1512, và ảnh được hiển thị trên màn hình. Ngoài ra, dữ liệu ảnh, ví dụ, thu nhận được bằng xử lý giải mã được thực hiện bằng cỗ máy mã hóa/giải mã 1516 được ghi bằng cỗ máy mã hóa/giải mã 1516, được dồn kênh bằng bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518, được chuyển đổi thành dữ liệu tệp, được xuất, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự qua giao diện video 1520, và được ghi trên các phương tiện ghi khác nhau.

Ngoài ra, dữ liệu tệp của dữ liệu đã được mã hóa thu nhận được bằng cách mã hóa dữ liệu ảnh, được đọc từ phương tiện ghi không được minh họa trên hình vẽ, ví dụ, bằng đầu nối 1321 (Fig.46) hoặc tương tự được cấp đến bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518 qua giao diện video 1520, được phân kênh, và được giải mã bởi cỗ máy mã hóa/giải mã 1516. Đối với dữ liệu ảnh thu nhận được bằng xử lý giải mã được thực hiện bằng cỗ máy mã hóa/giải mã 1516, xử lý ảnh định trước được thực hiện bằng cỗ máy xử lý ảnh 1514 và chuyển đổi định trước được thực hiện bằng cỗ máy hiển thị 1513. Sau đó, dữ liệu ảnh thu được được cấp, ví dụ, đến đầu nối 1321 (Fig.46) hoặc tương tự qua giao diện hiển thị 1512, và ảnh được hiển thị trên màn hình. Ngoài ra, dữ liệu ảnh, ví dụ, thu nhận được bằng xử lý giải mã được thực hiện bằng cỗ máy mã hóa/giải mã 1516 được ghi bằng cỗ máy mã hóa/giải mã 1516, được dồn kênh bằng bộ dồn kênh/bộ phân kênh (MUX DMUX) 1518, được chuyển đổi thành luồng vận chuyển, được cấp, ví dụ, đến đầu nối 1321 (Fig.46), môđem dải rộng 1333 (Fig.46), hoặc tương tự qua giao diện mạng 1519, và được truyền đến thiết bị khác không được minh họa trên hình vẽ.

Ở đây, việc truyền/thu dữ liệu ảnh và dữ liệu khác giữa các bộ phận xử lý bên trong bộ xử lý video 1332, ví dụ, được thực hiện bằng cách sử dụng bộ nhớ

trong 1515 hoặc bộ nhớ ngoài 1312. Ngoài ra, mô đun quản lý công suất 1313, ví dụ, điều khiển việc cấp điện đến bộ điều khiển 1511.

Trong trường hợp mà sáng chế áp dụng cho bộ xử lý video 1332 được tạo cấu hình như vậy, sáng chế theo từng phương án được mô tả trên đây có thể áp dụng cho cỗ máy mã hóa/giải mã 1516. Nói cách khác, ví dụ, cỗ máy mã hóa/giải mã 1516 có thể được tạo cấu hình để bao gồm các khối chức năng thực hiện thiết bị mã hóa ảnh (Fig.1) được tạo thành bởi bộ mã hóa 1 và bộ tạo tệp 2 theo phương án thứ nhất và thiết bị giải mã ảnh (Fig.8) được tạo thành bởi bộ đọc tệp 51 và bộ giải mã 52. Bằng cách tạo cấu hình như vậy, bộ xử lý video 1332 có thể có các ưu điểm giống như được mô tả trên đây có dựa vào Fig.1 đến Fig.36.

Ngoài ra, trong cỗ máy mã hóa/giải mã 1516, sáng chế (nói cách khác, các chức năng của thiết bị mã hóa ảnh và thiết bị giải mã ảnh theo từng phương án được mô tả trên đây) có thể được thực hiện bằng phần cứng như các mạch logic hoặc phần mềm như chương trình được nhúng hoặc có thể được thực hiện bằng cả phần cứng và phần mềm.

Mặc dù hai ví dụ của cấu hình của bộ xử lý video 1332 đã được mô tả trên đây, tuy nhiên, cấu hình của bộ xử lý video 1332 là tùy ý và có thể là cấu hình không phải là hai ví dụ được mô tả trên đây. Ở đây, bộ xử lý video 1332 có thể được tạo cấu hình như là một chip bán dẫn hoặc các chip bán dẫn. Ví dụ, bộ xử lý video có thể được tạo cấu hình như là LSI chồng ba chiều. Ngoài ra, bộ xử lý video có thể được thực hiện bởi các LSI.

Ví dụ áp dụng cho thiết bị

Tập video 1300 có thể được tạo trong các thiết bị khác nhau mà xử lý dữ liệu ảnh. Ví dụ, tập video 1300 có thể được tạo trong thiết bị truyền hình 900 (Fig.42), điện thoại di động 920 (Fig.43), thiết bị ghi/tái tạo 940 (Fig.44), thiết bị tạo ảnh 960 (Fig.45), và tương tự. Bằng cách tạo tập video 1300 trong đó, thiết bị có thể thu nhận các ưu điểm tương tự như được mô tả trên đây có dựa vào Fig.1 đến Fig.36.

Ngoài ra, mặc dù cấu hình là một phần của các cấu hình của tập video

1300 được mô tả trên đây, trong trường hợp mà cấu hình bao gồm bộ xử lý video 1332, cấu hình có thể được thực hiện như là cấu hình được áp dụng sáng chế. Ví dụ, chỉ bộ xử lý video 1332 có thể được thực hiện như là bộ xử lý video được áp dụng sáng chế. Ngoài ra, ví dụ, như được mô tả trên đây, bộ xử lý được biểu thị bằng đường chấm 1341, môđun video 1311, và tương tự có thể được thực hiện như là bộ xử lý, môđun, và tương tự được áp dụng sáng chế. Hơn thế nữa, ví dụ, môđun video 1311, bộ nhớ ngoài 1312, môđun quản lý công suất 1313, và môđun đầu trước 1314 có thể được kết hợp để được thực hiện như là bộ video 1361 được áp dụng sáng chế. Theo cấu hình bất kỳ, các ưu điểm tương tự như được mô tả trên đây có dựa vào Fig.1 đến Fig.36 có thể thu nhận được.

Nói cách khác, cấu hình bất kỳ có thể được tạo trong các thiết bị mà xử lý dữ liệu ảnh, tương tự như trường hợp của tập video 1300, chỉ cần cấu hình bao gồm bộ xử lý video 1332. Ví dụ, bộ xử lý video 1332, bộ xử lý được biểu thị bằng đường chấm 1341, môđun video 1311, hoặc bộ video 1361 có thể được tạo trong thiết bị truyền hình 900 (Fig.42), điện thoại di động 920 (Fig.43), thiết bị ghi/tái tạo 940 (Fig.44), thiết bị tạo ảnh 960 (Fig.45), và tương tự. Bằng cách tạo cấu hình bất kỳ mà cần được áp dụng sáng chế trong thiết bị, tương tự như trường hợp của tập video 1300, thiết bị có thể thu nhận các ưu điểm tương tự như được mô tả trên đây có dựa vào từ Fig.1 đến Fig.36.

Trong bản mô tả này, ví dụ đã được mô tả trong đó các loại thông tin như mã bắt đầu, dữ liệu đệm, thông số quản lý bộ đệm giải mã, thông số nhận dạng, và FixedStartCodeSizeFlag được dồn kênh trong luồng đã được mã hóa và được truyền từ phía mã hóa đến phía giải mã. Tuy nhiên, kỹ thuật truyền thông tin như vậy không bị giới hạn ở kỹ thuật như vậy. Ví dụ, thông tin như vậy có thể được truyền hoặc ghi như là dữ liệu riêng rẽ liên kết với luồng bit đã được mã hóa mà không được dồn kênh trong luồng đã được mã hóa. Ở đây, thuật ngữ "được liên kết" thể hiện là ảnh (nó có thể là một phần của ảnh như lát, khối, hoặc tương tự) có trong luồng bit và thông tin tương ứng với ảnh thu nhận được với việc được liên kết với nhau khi giải mã ảnh và thông tin. Nói cách khác, thông tin có thể được truyền trong đường truyền không phải là cho ảnh (hoặc luồng bit). Ngoài ra, thông tin có thể được ghi trên phương tiện ghi không phải là cho ảnh (hoặc

luồng bit) (hoặc vùng ghi khác của cùng phương tiện ghi). Hơn thế nữa, thông tin và ảnh (hoặc luồng bit), ví dụ, có thể được liên kết với nhau theo các đơn vị của các bộ phận tùy ý như đa khung, một khung, hoặc một phần của khung.

Mặc dù các phương án của sáng chế đã được mô tả một cách chi tiết có dựa vào các hình vẽ kèm theo, tuy nhiên, sáng chế không bị giới hạn ở các ví dụ như vậy. Thấy rõ là người có hiểu biết trung bình trong kỹ thuật của sáng chế có thể thực hiện các thay đổi hoặc cải biến khác nhau trong phạm vi kỹ thuật được mô tả trong yêu cầu bảo hộ, và hiển nhiên, hiểu được rằng các thay đổi hoặc cải như vậy thuộc phạm vi kỹ thuật của sáng chế.

Danh sách số chỉ dẫn

- 1 Bộ mã hóa
- 2 Bộ tạo tệp
- 31 Bộ điều khiển
- 32 Bộ thiết lập
- 33 Bộ ghi tệp
- 51 Bộ đọc tệp
- 52 Bộ giải mã
- 71 Bộ thu tệp
- 72 Bộ thu nhận thông số
- 73 Bộ điều khiển giải mã
- 74 Bộ đọc dữ liệu
- 201 Thiết bị mã hóa ảnh
- 251 Thiết bị giải mã ảnh
- 300 Thiết bị chuyển đổi tệp
- 311 Bộ phân tích luồng video
- 401 Bộ mã hóa
- 451 Bộ giải mã

YÊU CẦU BẢO HỘ

1. Thiết bị xử lý ảnh bao gồm:

bộ thiết lập để thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

bộ điều khiển để thực hiện điều khiển bộ thiết lập để mã bắt đầu và dữ liệu đệm được thiết lập bởi bộ thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

2. Thiết bị theo điểm 1, trong đó thông số quản lý bộ đệm giải mã là thông số có trong thông tin sử dụng được video (Video Usability Information, viết tắt là VUI), thông số có trong khoảng đệm thông tin cải tiến bổ sung (Supplemental Enhancement Information, viết tắt là SEI), hoặc thông số có trong SEI thời gian hình ảnh.

3. Thiết bị theo điểm 1, trong đó bộ thiết lập thực hiện thiết lập dữ liệu đệm làm dữ liệu VCL (VCL: lớp mã hóa video).

4. Phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh, phương pháp xử lý ảnh này bao gồm các bước:

thiết lập mã bắt đầu và dữ liệu đệm cho tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

thực hiện điều khiển việc thiết lập mã bắt đầu và dữ liệu đệm để mã bắt đầu và dữ liệu đệm mã được thiết lập tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp.

5. Thiết bị xử lý ảnh bao gồm:

bộ thu để thu tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và được điều khiển và thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tệp với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tệp; và

bộ giải mã để đọc mã bắt đầu và dữ liệu đệm từ tệp thu được bởi bộ thu và

giải mã luồng bit bằng cách sử dụng thông số quản lý bộ đệm giải mã.

6. Thiết bị theo điểm 5, trong đó thông số quản lý bộ đệm giải mã là thông số có trong VUI (thông tin sử dụng được video), thông số có trong khoảng đệm SEI (thông tin cải tiến bổ sung), hoặc thông số có trong SEI thời gian hình ảnh.

7. Thiết bị theo điểm 5, trong đó dữ liệu đệm được thiết lập làm dữ liệu VCL.

8. Phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh, phương pháp xử lý ảnh bao gồm các bước:

thu tập bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh và được điều khiển và thiết lập để mã bắt đầu và dữ liệu đệm tạo cấu trúc tập với đặc tính của thông số quản lý bộ đệm giải mã đang được duy trì trong dữ liệu phương tiện của tập; và

đọc mã bắt đầu và dữ liệu đệm từ tập đã thu được và giải mã luồng bit bằng cách sử dụng thông số quản lý bộ đệm giải mã.

9. Thiết bị xử lý ảnh bao gồm:

bộ thiết lập để thiết lập đặc tính của thông số quản lý bộ đệm giải mã trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit trong dữ liệu phương tiện của tập bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

bộ tạo để tạo tập bằng cách sử dụng đặc tính được thiết lập bởi bộ thiết lập.

10. Thiết bị theo điểm 9, trong đó bộ thiết lập thực hiện thiết lập thông số nhận dạng nhận dạng rằng đặc tính của thông số quản lý bộ đệm giải mã được thay đổi.

11. Thiết bị theo điểm 10, trong đó bộ thiết lập thực hiện thiết lập thông số nhận dạng làm hộp tùy chọn trong mục nhập mẫu của tập.

12. Phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh, phương pháp xử lý ảnh này bao gồm các bước:

thiết lập đặc tính của thông số quản lý bộ đệm giải mã trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi luồng bit trong dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

tạo tệp bằng cách sử dụng đặc tính đã được thiết lập.

13. Thiết bị xử lý ảnh bao gồm:

bộ thu để thu tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

bộ giải mã để đọc thông số quản lý bộ đệm giải mã từ tệp thu được bởi bộ thu và giải mã luồng bit bằng cách sử dụng thông số đã được đọc.

14. Thiết bị theo điểm 13, trong đó thông số nhận dạng nhận dạng rằng đặc tính của thông số quản lý bộ đệm giải mã được thay đổi được thiết lập trong tệp.

15. Phương pháp xử lý ảnh bằng cách sử dụng thiết bị xử lý ảnh, phương pháp xử lý ảnh này bao gồm các bước:

thu tệp được tạo ra bằng cách sử dụng đặc tính đã được thiết lập của thông số quản lý bộ đệm giải mã mà được thiết lập trong trường hợp mà mã bắt đầu và dữ liệu đệm được loại bỏ khỏi dữ liệu phương tiện của tệp bao gồm luồng bit thu nhận được bằng cách mã hóa ảnh; và

đọc thông số quản lý bộ đệm giải mã từ tệp đã thu được và giải mã luồng bit bằng cách sử dụng thông số đã được đọc.

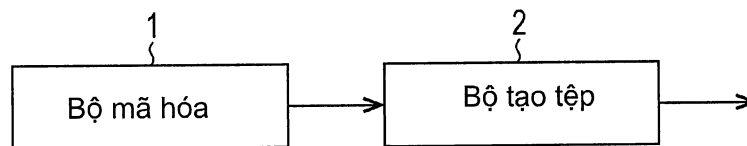
FIG. 1

FIG. 2

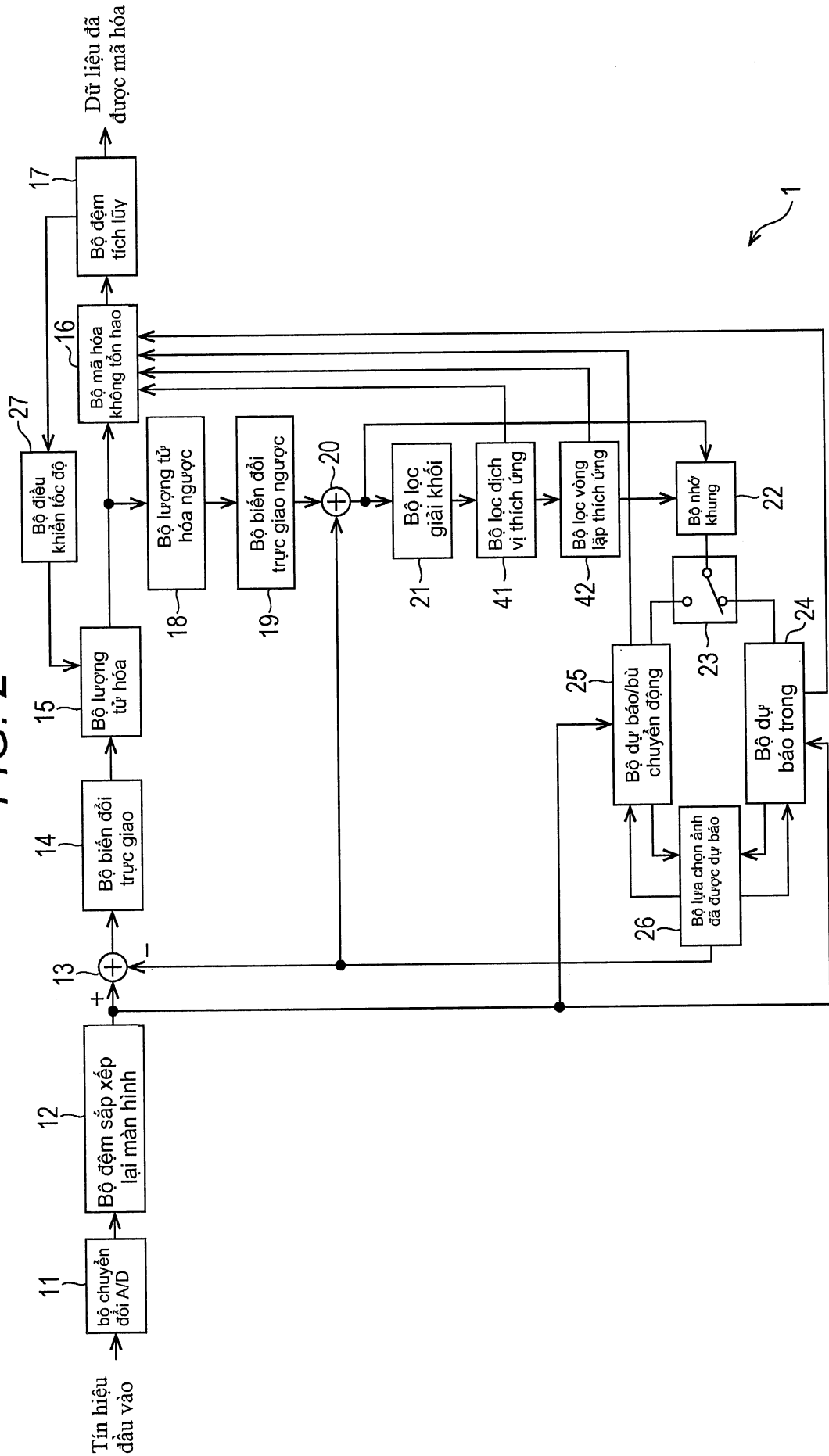


FIG. 3

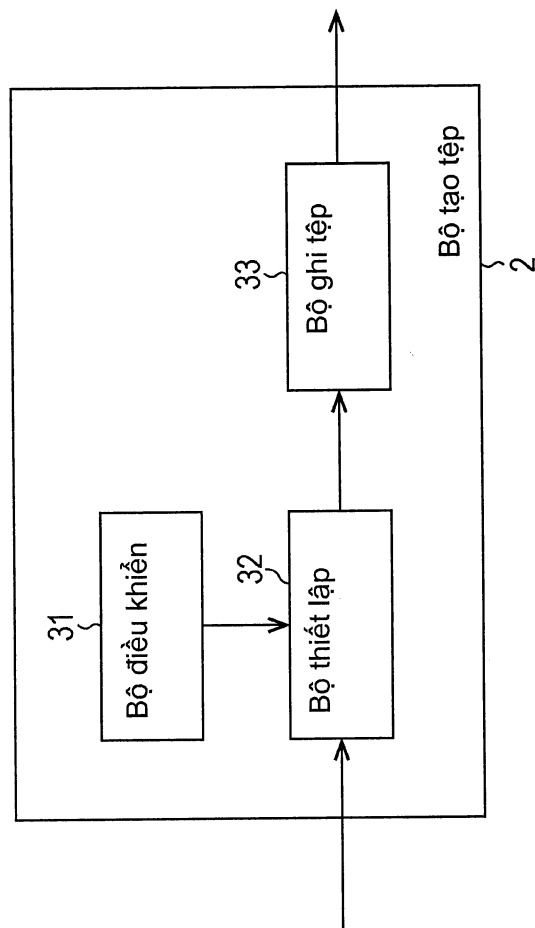


FIG. 4

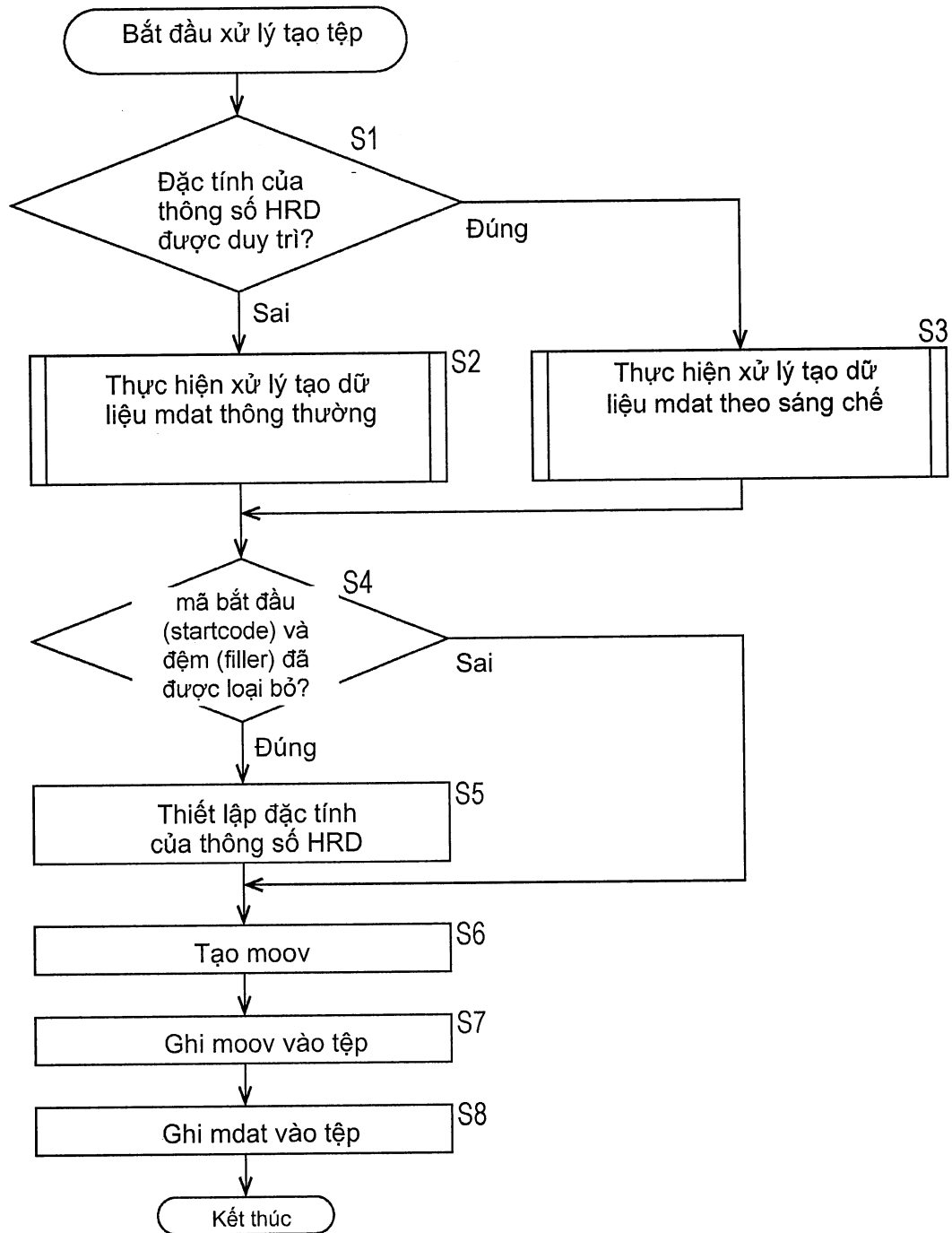


FIG. 5

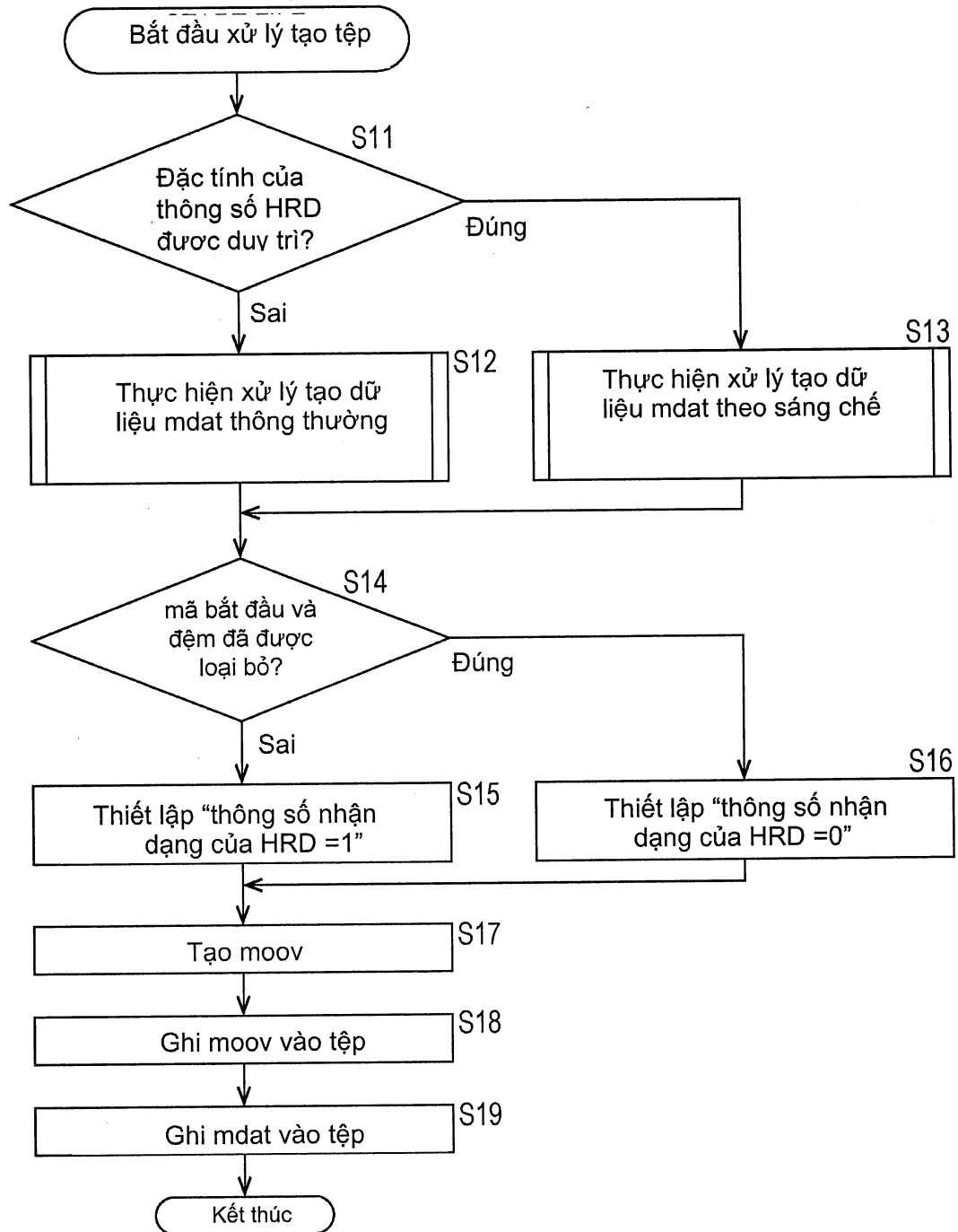


FIG. 6

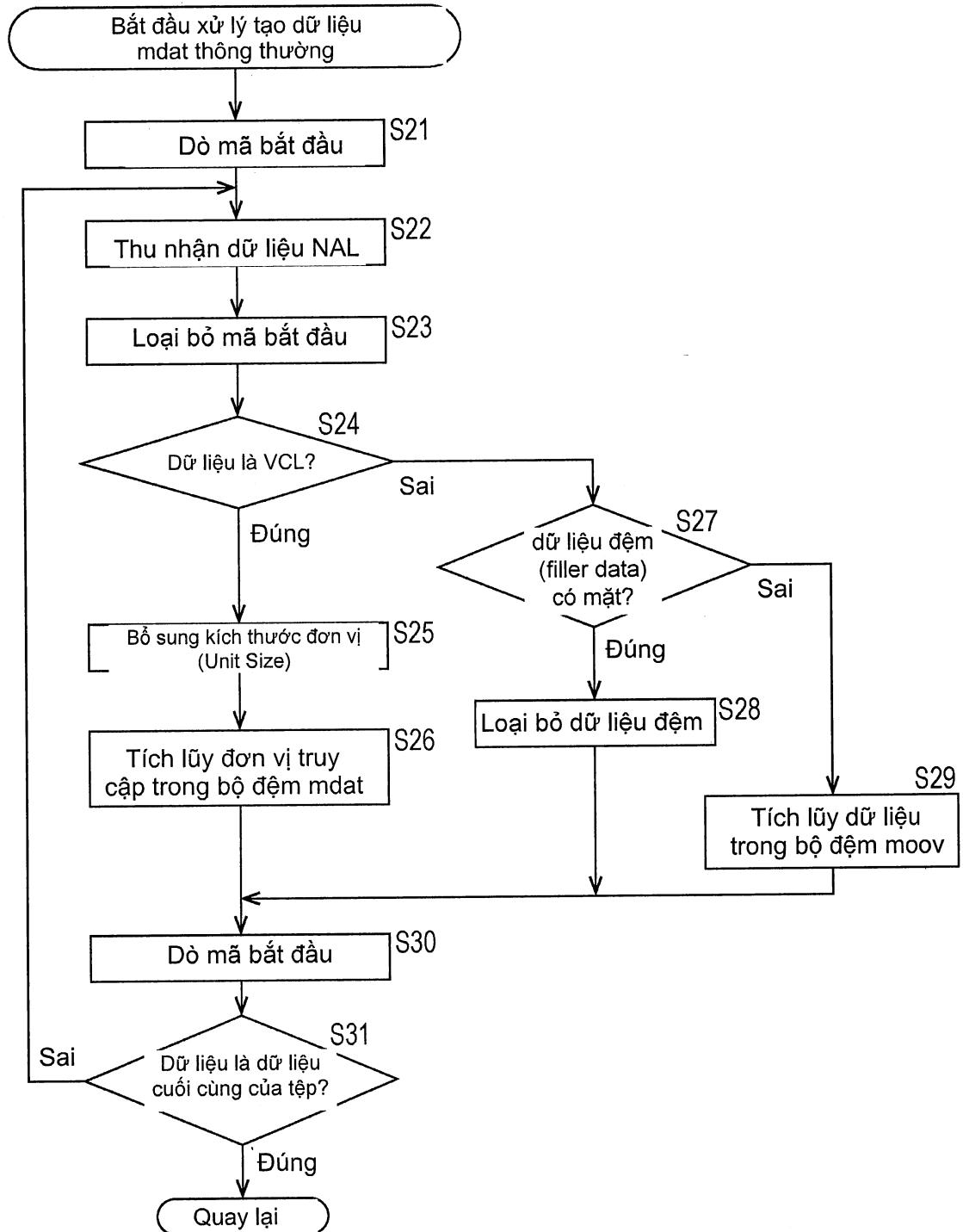


FIG. 7

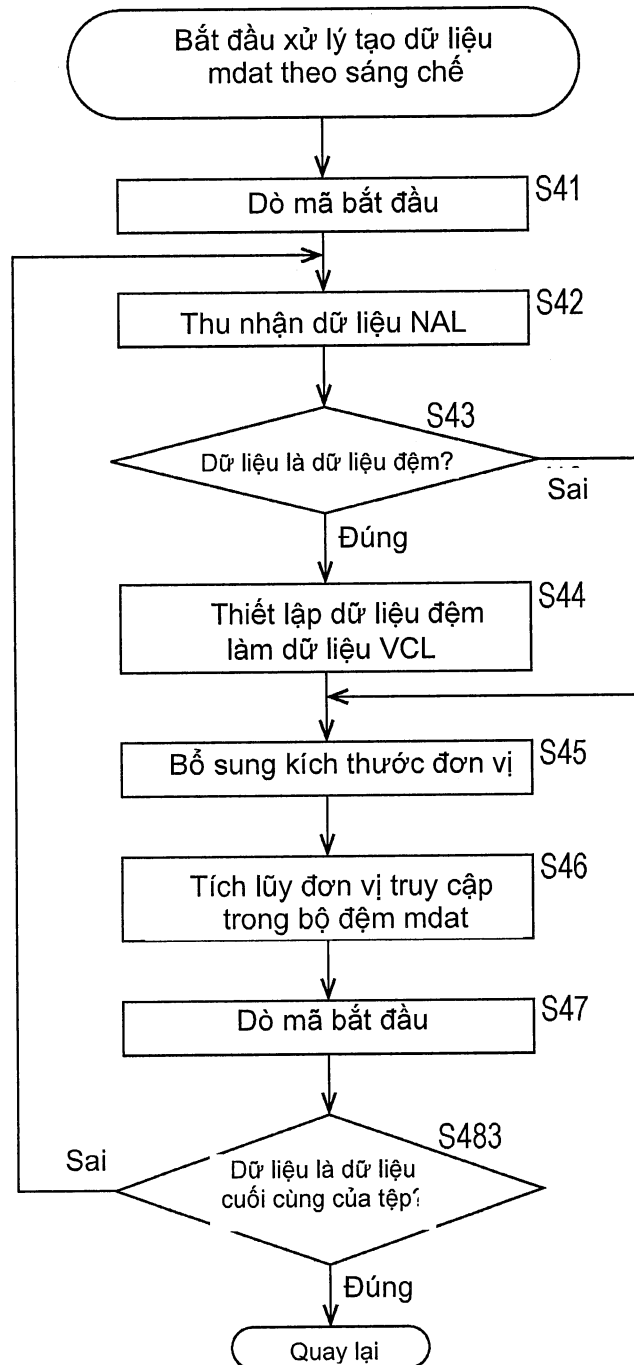


FIG. 8

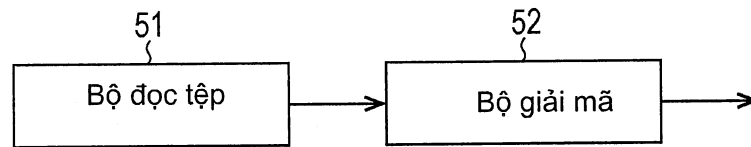


FIG. 9

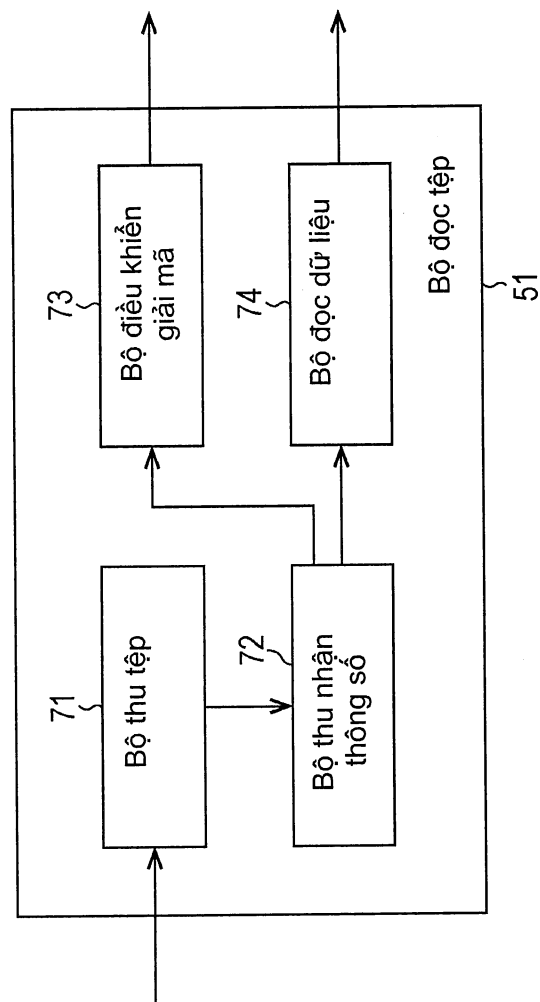


FIG. 10

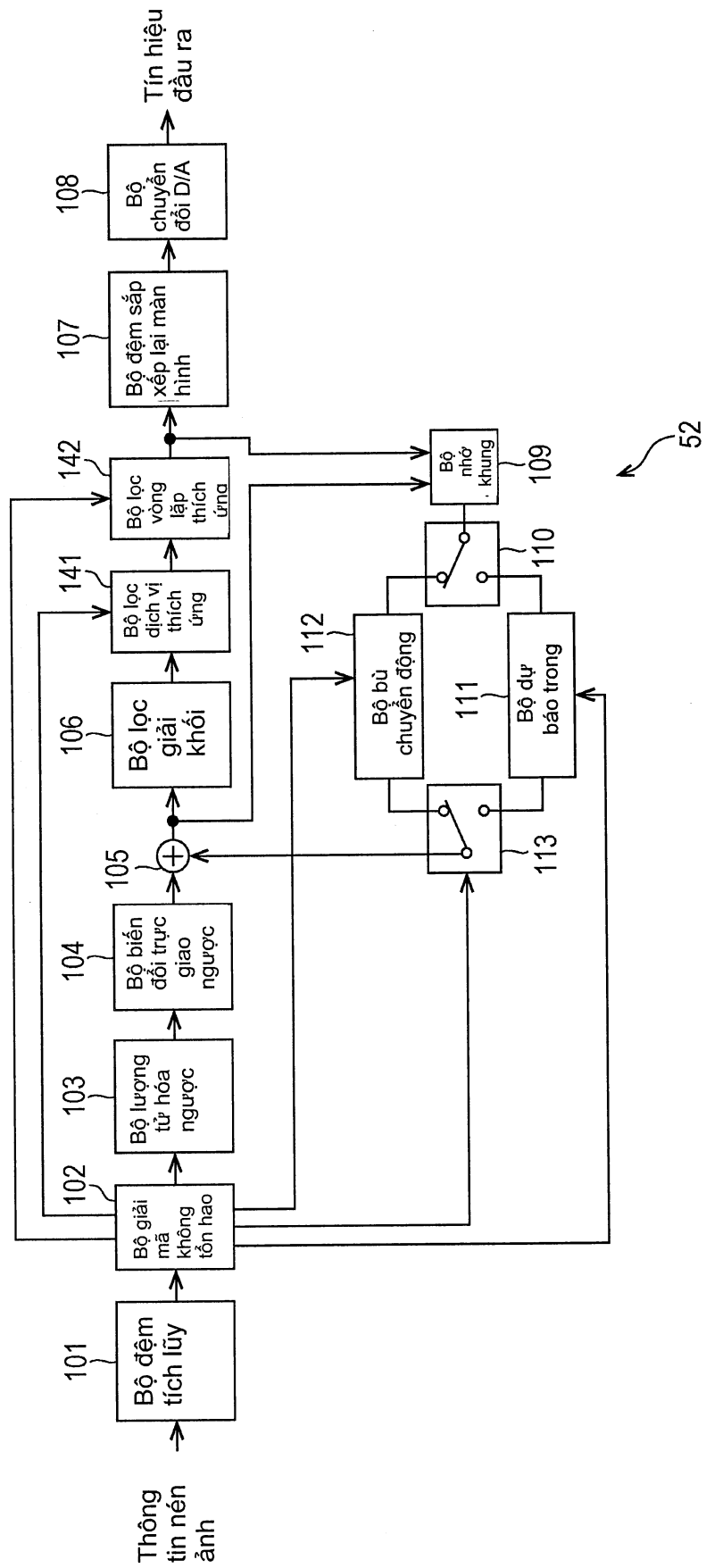


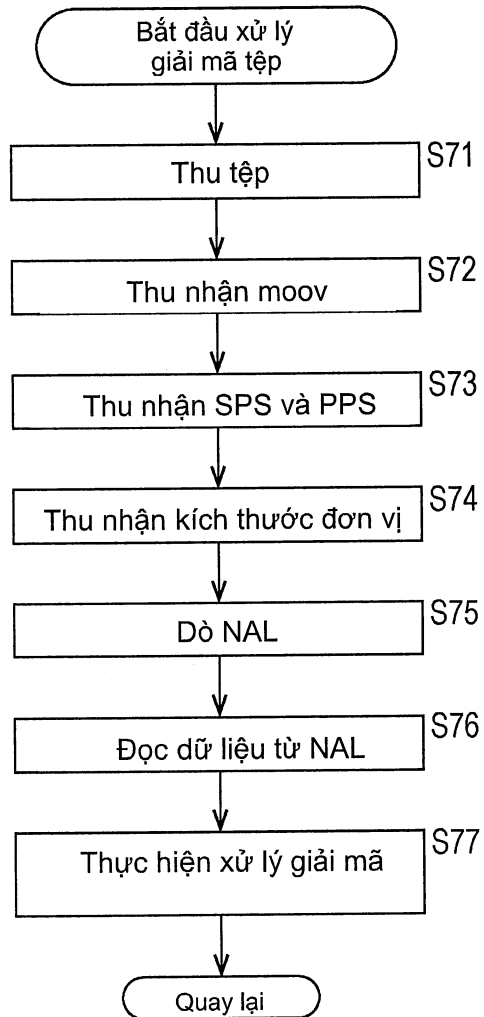
FIG. 11

FIG. 12

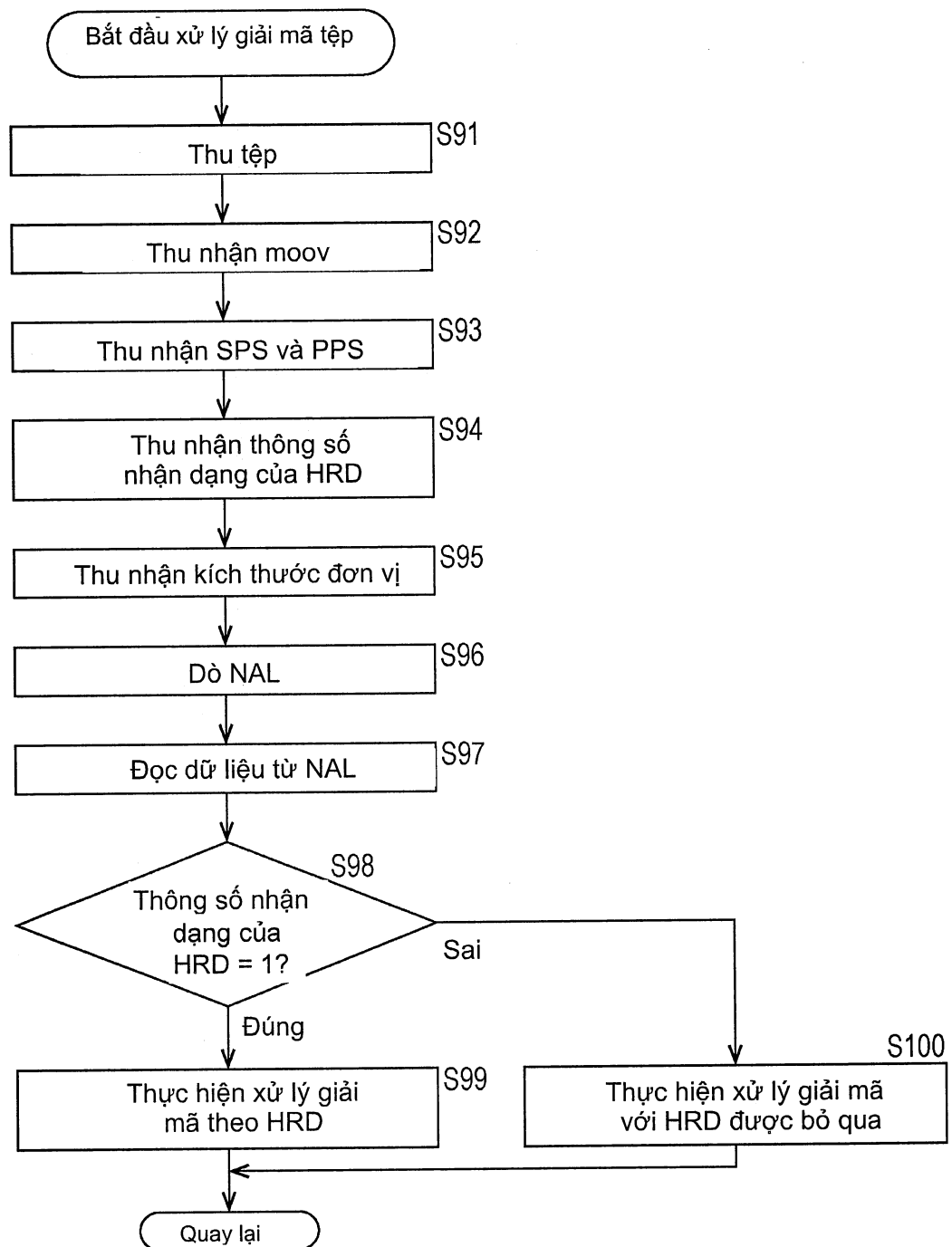


FIG. 13

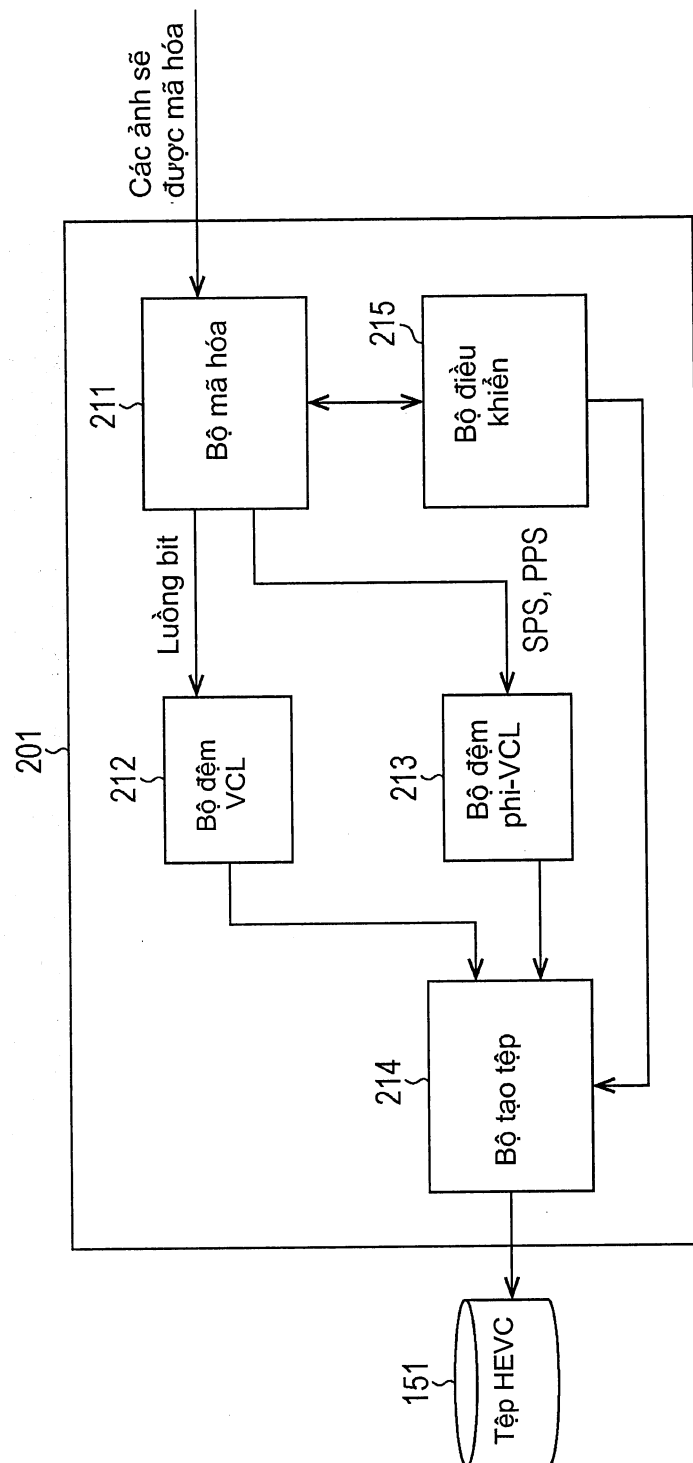


FIG. 14

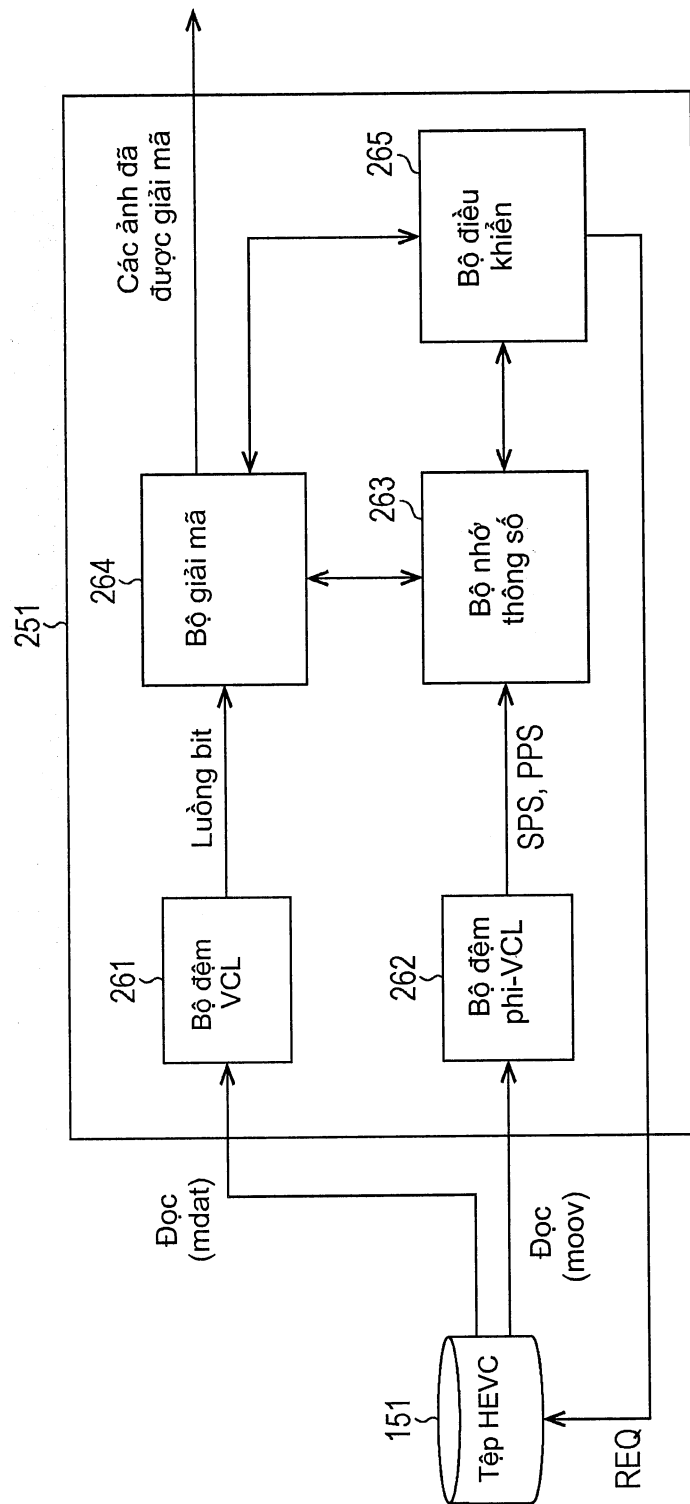


FIG. 15

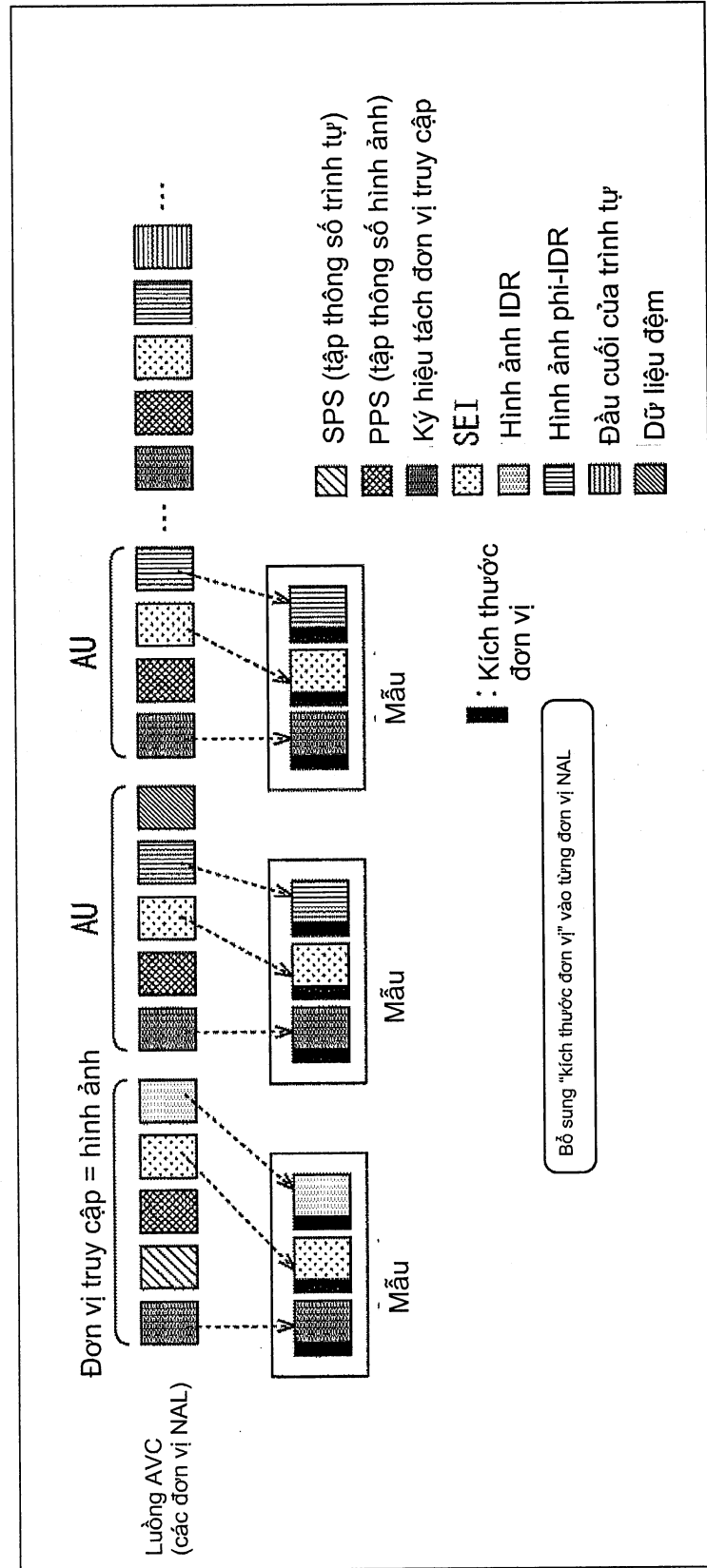


FIG. 16

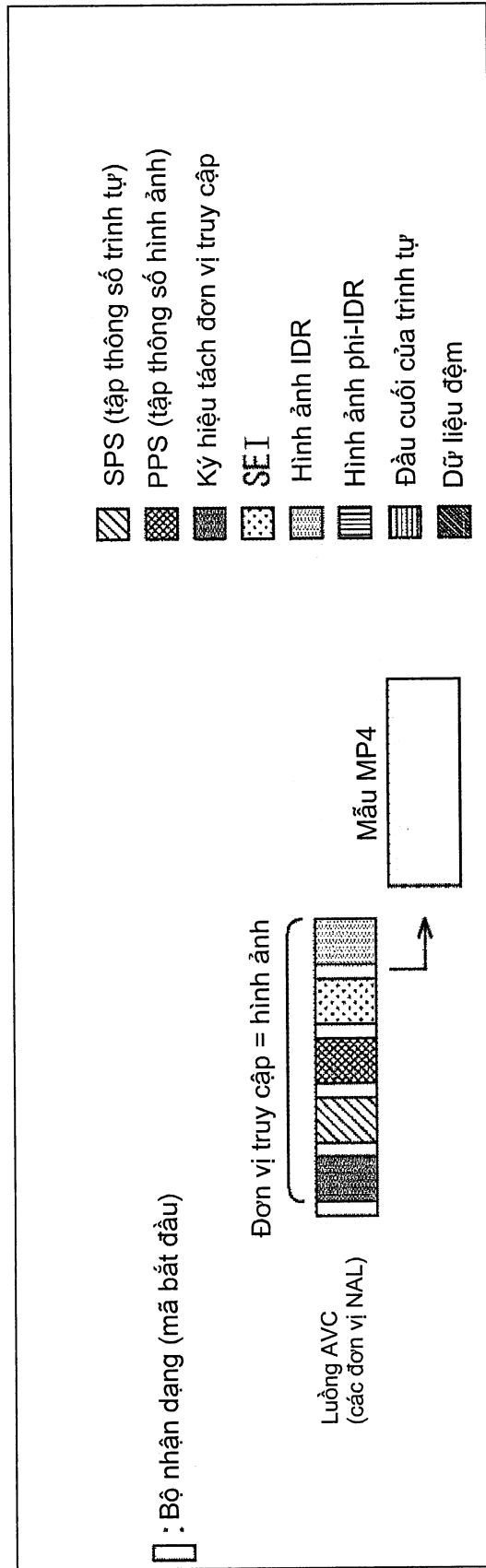


FIG. 17

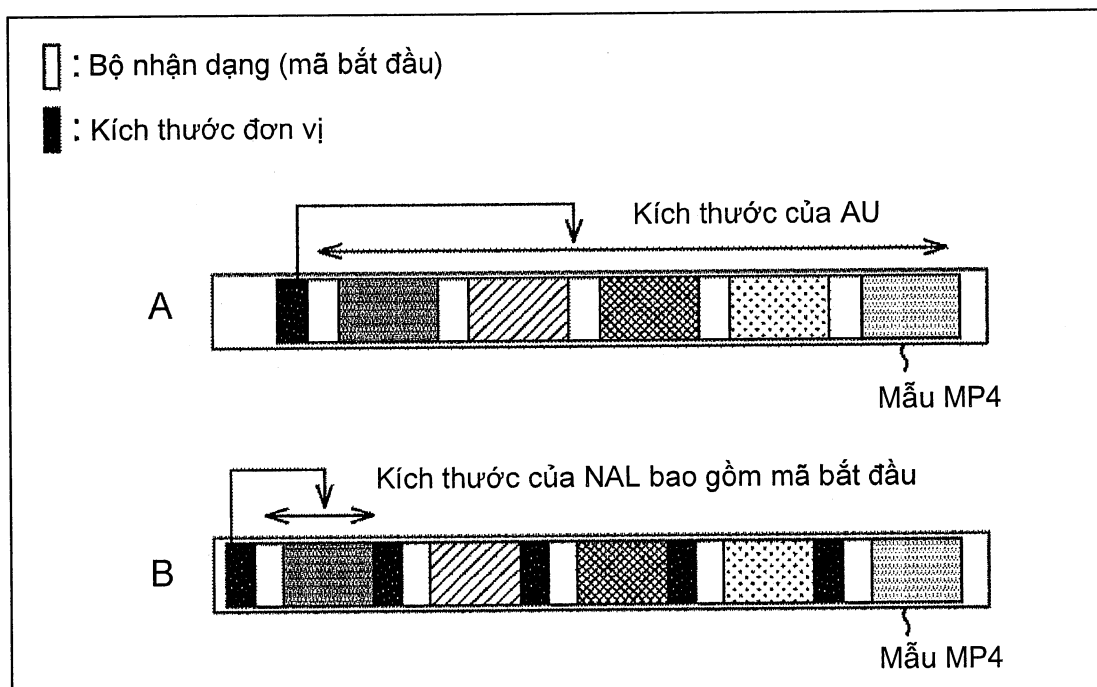


FIG. 18

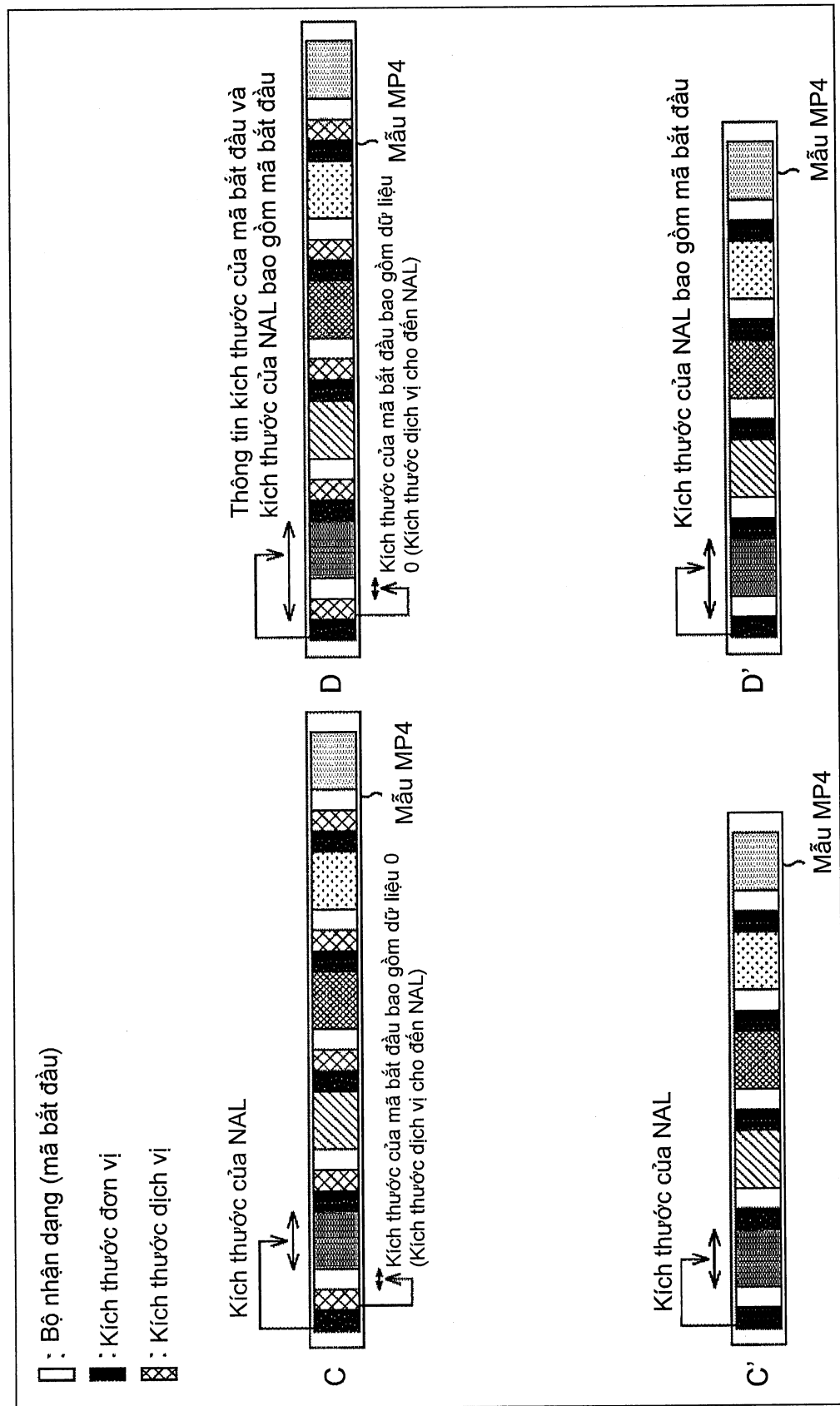


FIG. 19

```

AVCSampleEntry() extends VisualSampleEntry(type') {
    //type is 'avc1' or 'avc3'

    AVConfigurationBox config;
    MPEG4BitRateBox(); //Optional
    MPEG4ExtensionDescriptorsBox(); //Optional
    HRDConformanceBox(); //Optional
}

HRDConformanceBox có thể được định nghĩa như sau

class HRDConformanceBox extends Box('hrdc') {
    HRDConformanceData() HRDConformance;
}

aligned(8) class HRDConformanceData {
    unsigned int(6) reserved = 0;
    unsigned int(1) FixedStartCodeSizeFlag;
    unsigned int(1) HRDConformanceFlag;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}

```

FixedStartCodeSizeFlag là bộ nhận dạng khi mã bắt đầu trong từng NAL là kích thước cố định 4 byte. Khi FixedStartCodeSizeFlag bằng 1, không có trường dịch vị cho từng NAL trong dữ liệu mẫu để biểu thị kích thước của mã bắt đầu

HRDConformanceFlag là bộ nhận dạng nếu các đặc trưng HRD được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., từ luồng thứ cấp. Khi HRDConformanceFlag bằng 1, các thông số HRD, ví dụ, khoảng đệm SEI, SEI thời gian ảnh, được thay đổi bằng cách loại bỏ mã bắt đầu, dữ liệu đệm, v.v., từ luồng thứ cấp

FIG. 20

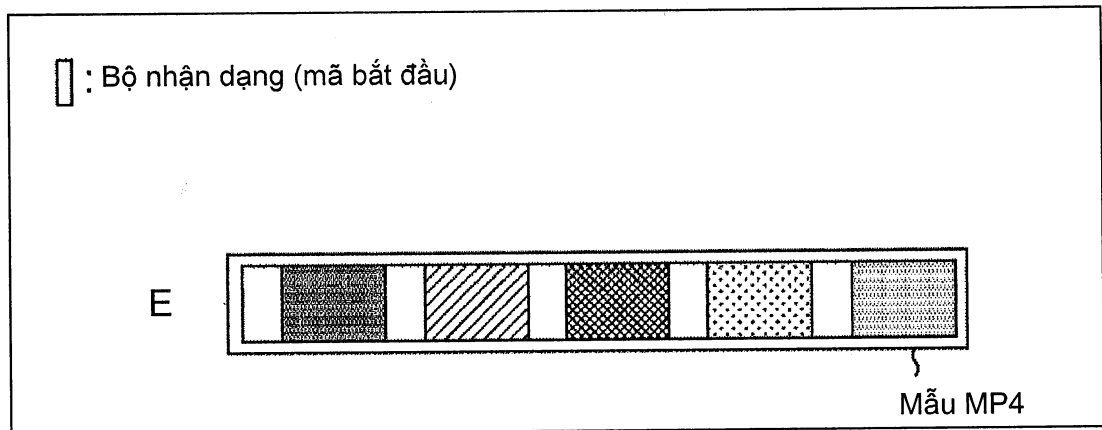


FIG. 21

```

aligned(8) class NALunitOffsetBox
extends FullBox( 'nalo' , version=0, 0) {
    unsigned int(32) sample_count;
    for (i=1; i <= sample_count; i++) {
        unsigned int(32) offset_count;
        for (j=1; j <= offset_count; j++) {
            unsigned int(32) offset;
        }
    }
}

```

sample_count là số nguyên mà đưa ra số lượng mục nhập trong bảng sau
offset_count là số nguyên mà đưa ra số lượng mục nhập trong bảng con sau
offset là số nguyên mà đưa ra dịch vị của phần bắt đầu của đơn vị NAL vào mẫu chứa của nó.

FIG. 22

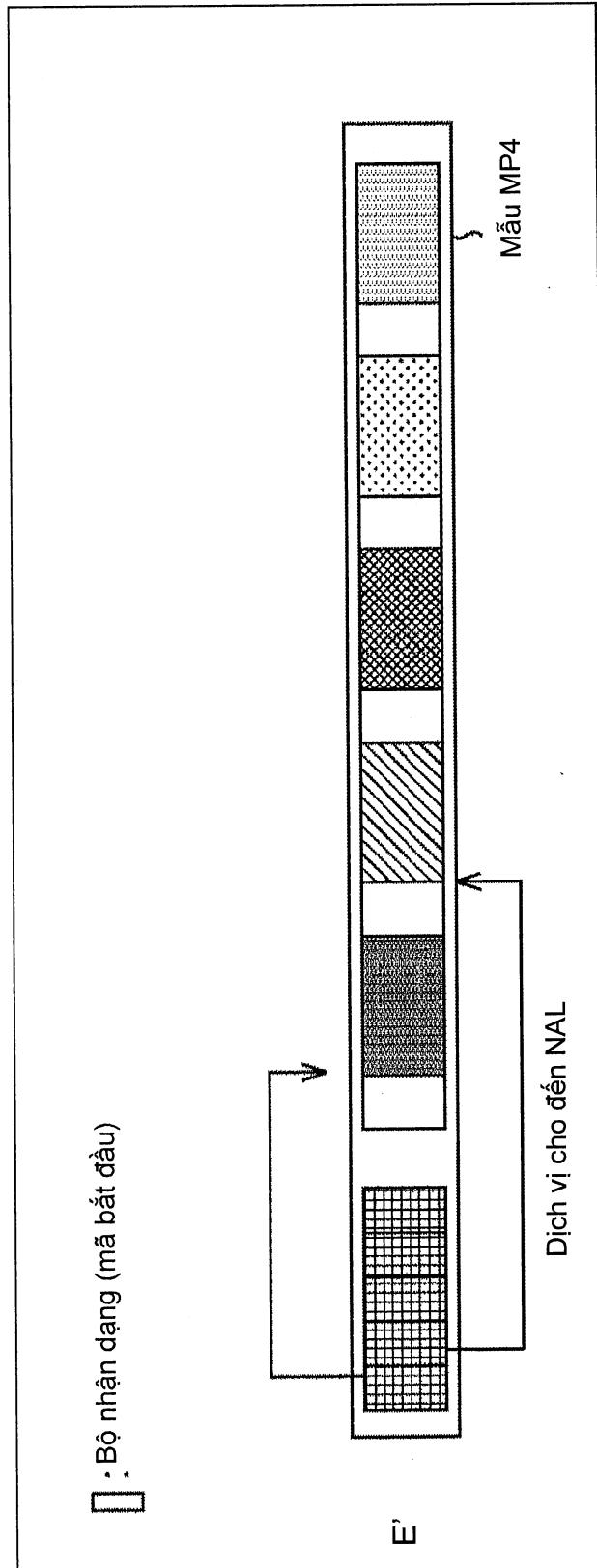


FIG. 23

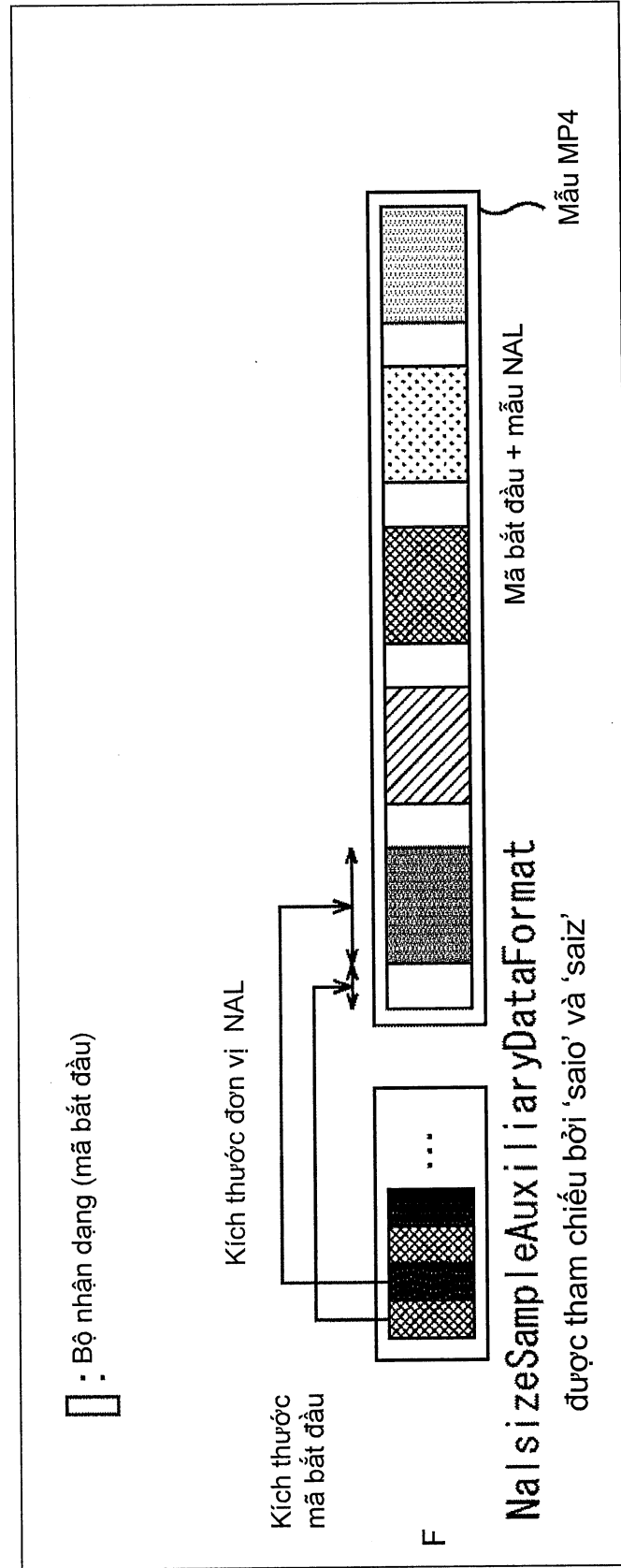


FIG. 24

```

aligned(8) class NalsizeSampleAuxiliaryDataFormat
{
    unsigned int(2) StartcodeSizeMinusOne;
    unsigned int(2) NALunitSizeMinusOne;
    unsigned int(4) reserved;
    unsigned int(32) subsample_count;
    for (i=1; i <= subsample_count; i++) {
        switch (StartcodeSizeMinusOne) {
            case 0:
                unsigned int(8) Startcode Size;
                break;
            case 1:
                unsigned int(16) Startcode Size;
                break;
            case 3:
                unsigned int(32) Startcode Size;
                break;

        }

        switch (NALUnitSizeMinusOne) {
            case 0:
                unsigned int(8) NALUnit Size;
                break;
            case 1:
                unsigned int(16) NALUnit Size;
                break;
            case 3:
                unsigned int(32) NALUnit Size;
                break;

        }
    }
}

```

FIG. 25

```

aligned(8) class NalSizeSampleAuxiliaryDataFormat
{
    unsigned int(1) FixedStartCodeSizeFlag;
    unsigned int(2) StartcodeSizeMinusOne;
    unsigned int(2) NALUnitSizeMinusOne;
    unsigned int(3) reserved;
    unsigned int(32) subsample_count;
    for (i=1; i <= subsample_count; i++){
        if (FixedStartCodeSizeFlag == 0){
            switch (StartcodeSizeMinusOne){
                case 0:
                    unsigned int(8) Startcode Size;
                    break;
                case 1:
                    unsigned int(16) Startcode Size;
                    break;
                case 3:
                    unsigned int(32) Startcode Size;
                    break;
            }
        }
        switch (NALUnitSizeMinusOne){
            case 0:
                unsigned int(8) NALUnit Size;
                break;
            case 1:
                unsigned int(16) NALUnit Size;
                break;
            case 3:
                unsigned int(32) NALUnit Size;
                break;
        }
    }
}

```

FIG. 26

	Ưu điểm
B của Fig.17	Tương tự cấu trúc hiện thời của kích thước và dữ liệu
C của Fig.18 hoặc C' của Fig.18	Tương tự cấu trúc hiện thời của kích thước và dữ liệu. Các ngữ nghĩa của kích thước đơn vị không thay đổi, nếu kích thước của mã bắt đầu được cố định, cấu trúc dữ liệu là giống như thiết kế định dạng tệp hiện thời
D của Fig.18 hoặc D' của Fig.18	Tương tự cấu trúc hiện thời của kích thước và dữ liệu
E của Fig.20 hoặc E' của Fig.22	Cách đơn giản để lưu giữ luồng sơ cấp. (Không cần loại bỏ mã bắt đầu, hoặc bổ sung kích thước đơn vị hay kích thước mã bắt đầu.)

FIG. 27

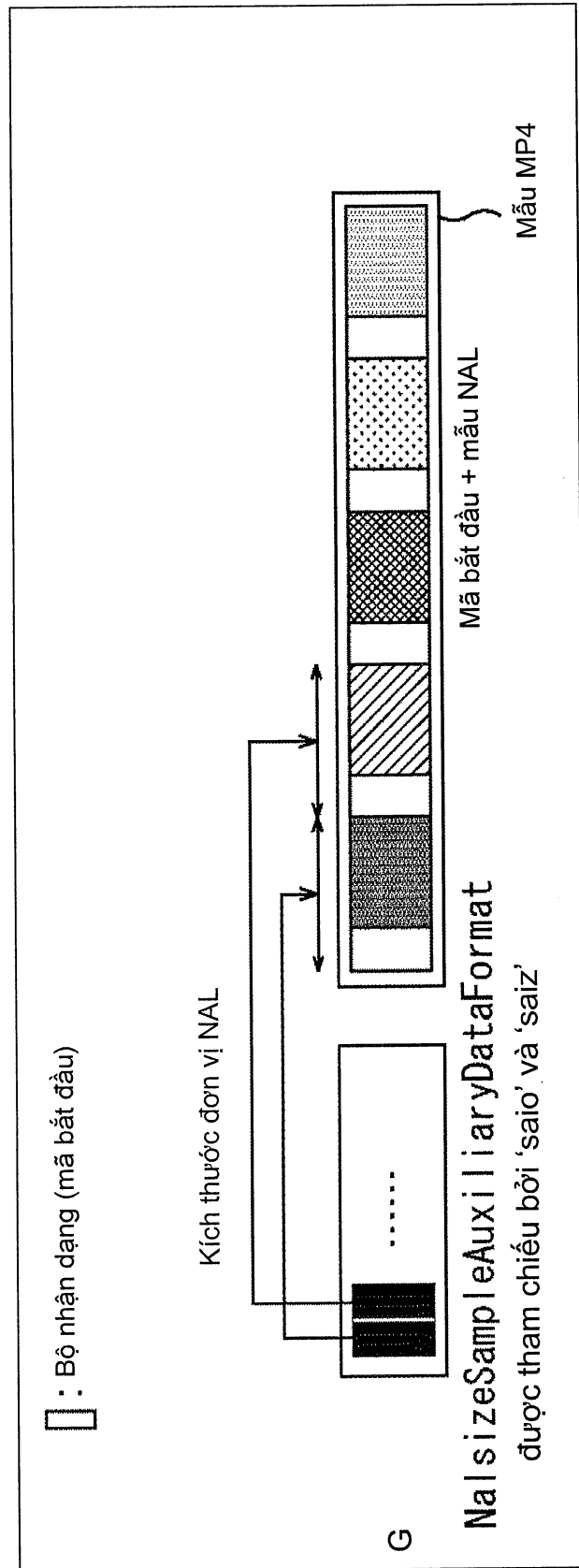


FIG. 28

- Hộp thông tin định dạng luồng byte (Byte Stream Format Information Box) có thể được lưu giữ trong mục nhập mẫu 'avc3', 'avc4', 'mvc3', 'mvc4' hoặc 'hev1' (trong hộp mục nhập mẫu (Sample entry) của rãnh phương tiện)
- Hộp này sẽ được lưu giữ khi dữ liệu mẫu được tạo cấu trúc bởi định dạng luồng byte (Byte Stream Format) như được định nghĩa trong ISO/IEC 14496-10.
- Hộp thông tin mẫu con (Sub-Sample Information Box) có thể được sử dụng để lưu giữ một hoặc nhiều kích thước đơn vị NAL luồng byte (Byte stream NAL unit size) liên kế
- Mẫu con (sub-sample) được định nghĩa là một hoặc nhiều kích thước đơn vị NAL luồng byte liên kế như được định nghĩa trong ISO/IEC 14496-10.

Loại hộp (Box Type): 'bsfi'

Mục (Container):

Mục nhập mẫu

('avc3', 'avc4', 'mvc3', 'mvc4' hoặc 'hev1')

Mục thiết yếu

(Mandatory):

Không

Số lượng (Quantity): Không hoặc một

```

• Class ByteStreamFormatInformationBox extends Box('bsfi') {
    ByteStreamFormatInformation() ByteStreamInformation;
}

```

FIG. 29

- Hộp thông tin định dạng luồng byte sẽ được lưu giữ trong mục nhập mẫu 'avc5', 'avc6', 'mvc5', 'mvc6' hoặc 'hev2' (trong hộp mục nhập mẫu của rãnh phương tiện)
- Hộp này sẽ được lưu giữ khi dữ liệu mẫu được tạo cấu trúc bởi định dạng luồng bit như được định nghĩa trong ISO/IEC 14496-10.
- Hộp thông tin mẫu con có thể được sử dụng để lưu giữ một hoặc nhiều kích thước đơn vị NAL luồng bit liên kế ;
- Mẫu con được định nghĩa là một hoặc nhiều kích thước đơn vị NAL luồng bit liên kế như được định nghĩa trong ISO/IEC 14496-10.

Loại hộp (Box Type): 'bsf i'

Mục (Container):

Mục nhập mẫu

- Mục thiết yếu
- Không

- (Mandatory):
- Số lượng (Quantity): Không hoặc một

- Class ByteStreamFormatInformationBox extends Box('bsfi') {
 ByteStreamFormatInformation;
}

-131-

FIG. 30

```

• aligned(8) class ByteStreamFormatInformation{
    unsigned int(1) StartcodePresentFlag;
    unsigned int(1) ZeroBytePresentFlag;
    unsigned int(1) LeadingZeroBytesPresentFlag;
    unsigned int(1) TrailingZeroBytesPresentFlag;
    unsigned int(4) reserved =0;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}

```

Mã bắt đầu ba byte được đưa vào trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con được đưa vào trong mẫu StartcodePresentFlag. Dữ liệu không 1 byte được đưa vào trước mã bắt đầu ZeroBytePresentFlag.

Các 0x00 của một byte có thể được đưa vào trước mã bắt đầu ba byte hoặc bốn byte của ít nhất một mẫu con trong số các mẫu con được đưa vào trong mẫu LeadingZeroBytePresentFlag.

Các 0x00 của một byte có thể được đưa vào sau đơn vị NAL của ít nhất một mẫu con trong số các mẫu con được đưa vào trong mẫu TrailingZeroBytePresentFlag.

FIG. 31

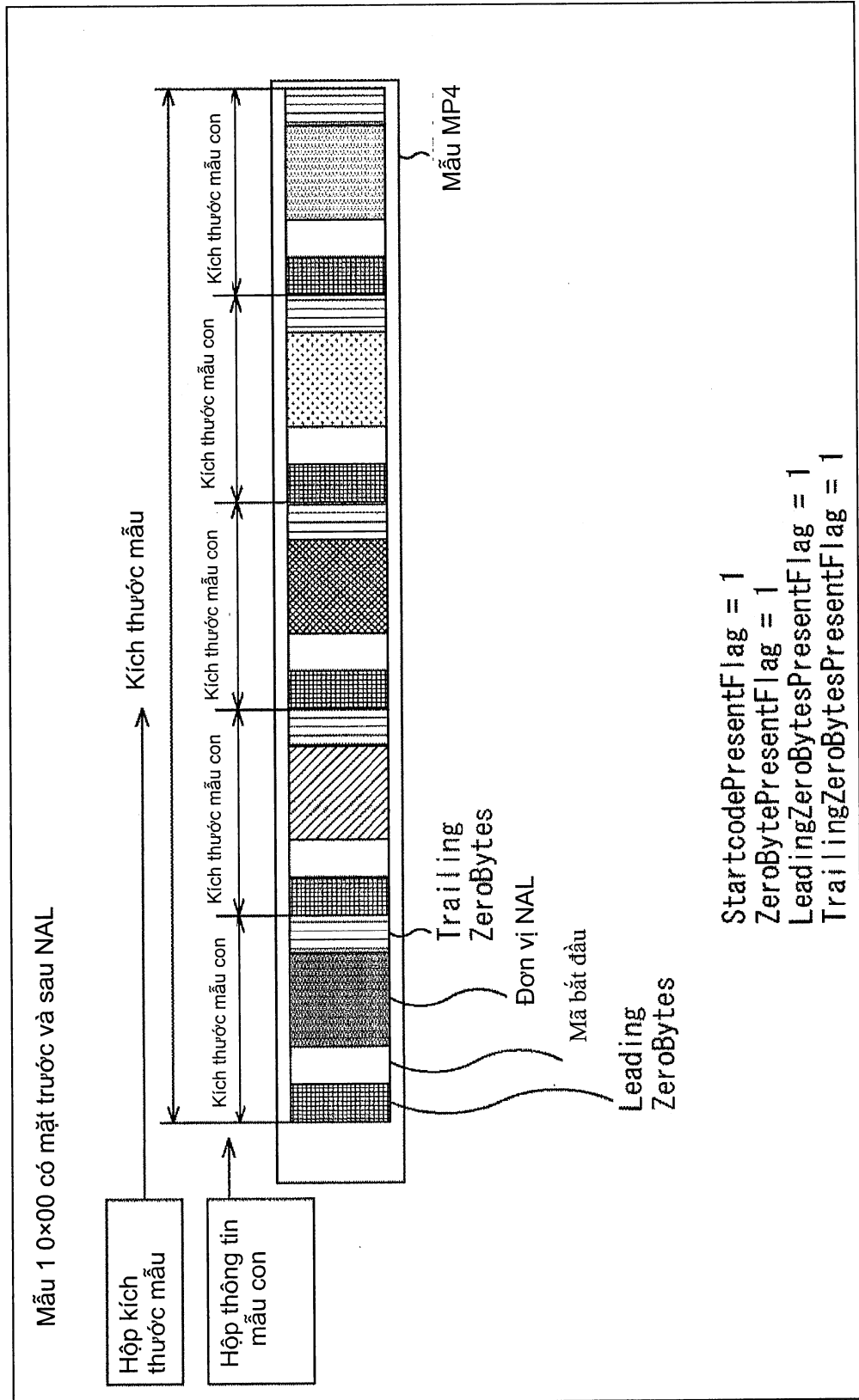


FIG. 32

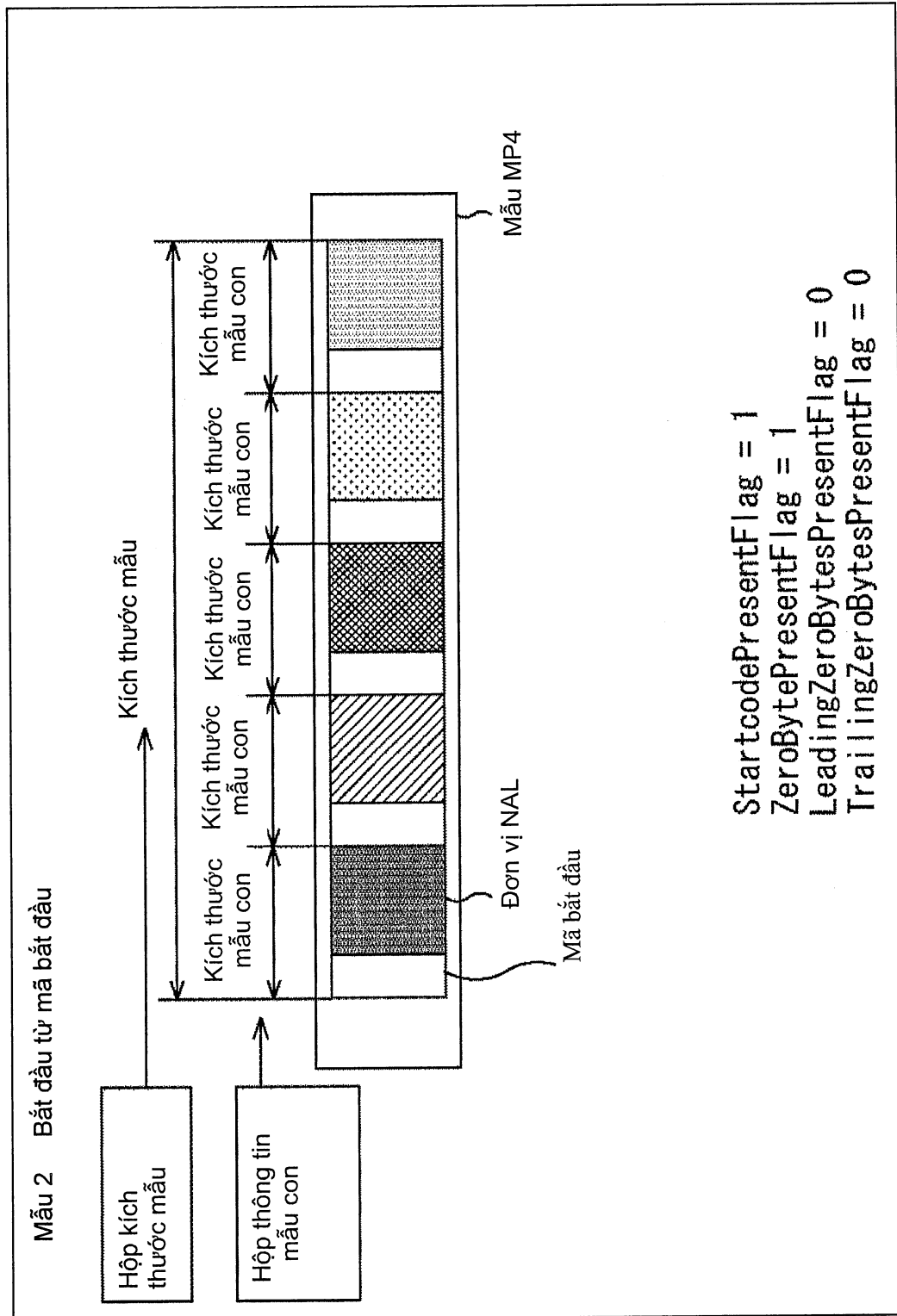


FIG. 34

```

• aligned(8) class ByteStreamFormatInformation {
    unsigned int(2) StartcodePresentFlag;
    unsigned int(1) LeadingZeroBytesPresentFlag;
    unsigned int(1) TrailingZeroBytesPresentFlag;
    unsigned int(4) reserved = 0;
    unsigned int(24) reserved;
    Box[] any_box; //Optional
}

```

Cờ 2 bit thể hiện cấu trúc của mã bắt đầu của từng mẫu con trong số tất cả các mẫu con của StartcodePresentFlag

StartcodePresentFlag = 00b dự phòng

StartcodePresentFlag = 01b Mã bắt đầu 3 byte được đưa vào trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu

StartcodePresentFlag = 10b Mã bắt đầu 3 byte và dữ liệu ZeroByte 1 byte được đưa vào trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu (tạo thành bởi 4 byte)

StartcodePresentFlag = 11b trước đơn vị NAL trong từng mẫu con trong số tất cả các mẫu con có trong mẫu, mã bắt đầu 3 byte được đưa vào, và dữ liệu ZeroByte 1 byte có thể được đưa vào (có khả năng là các mã bắt đầu 3 byte/4 byte được trộn => 01b/10b không được đảm bảo)

LeadingZeroBytesPresentFlag Các 0x00 1 byte có thể được đưa vào trước mã bắt đầu 3 byte hoặc

4 byte trong ít nhất một mẫu con trong số các mẫu con có trong mẫu.

TrailingZeroBytesPresentFlag Các 0x00 1 byte có thể được đưa vào sau đơn vị NAL trong ít nhất

một mẫu con trong số các mẫu con có trong mẫu.

FIG. 35

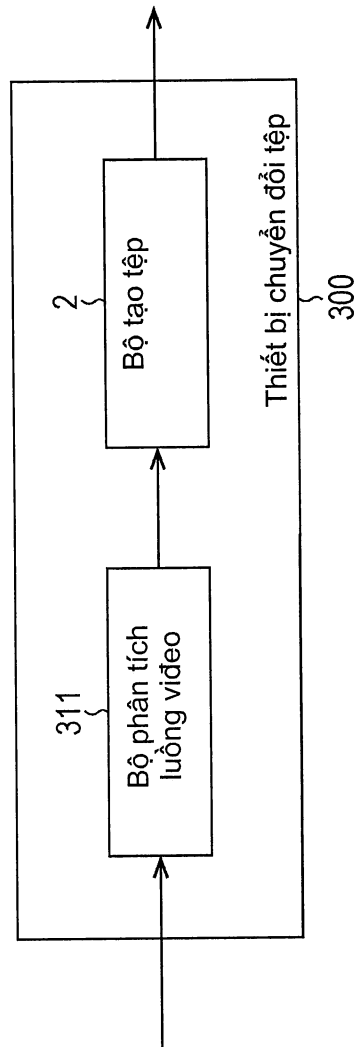


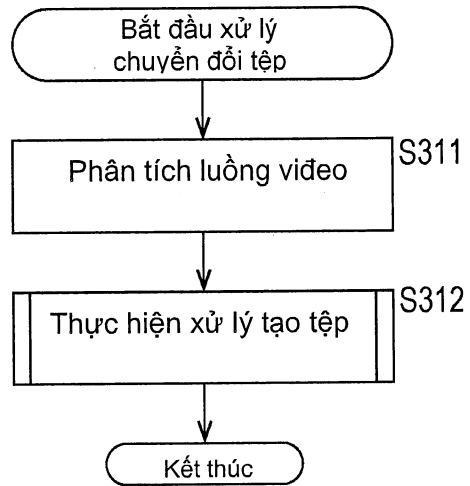
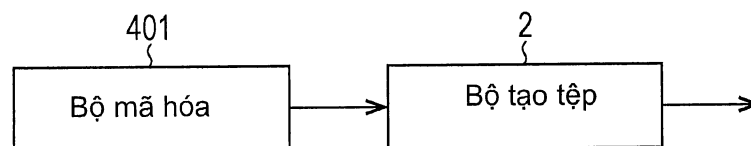
FIG. 36*FIG. 37*

FIG. 38

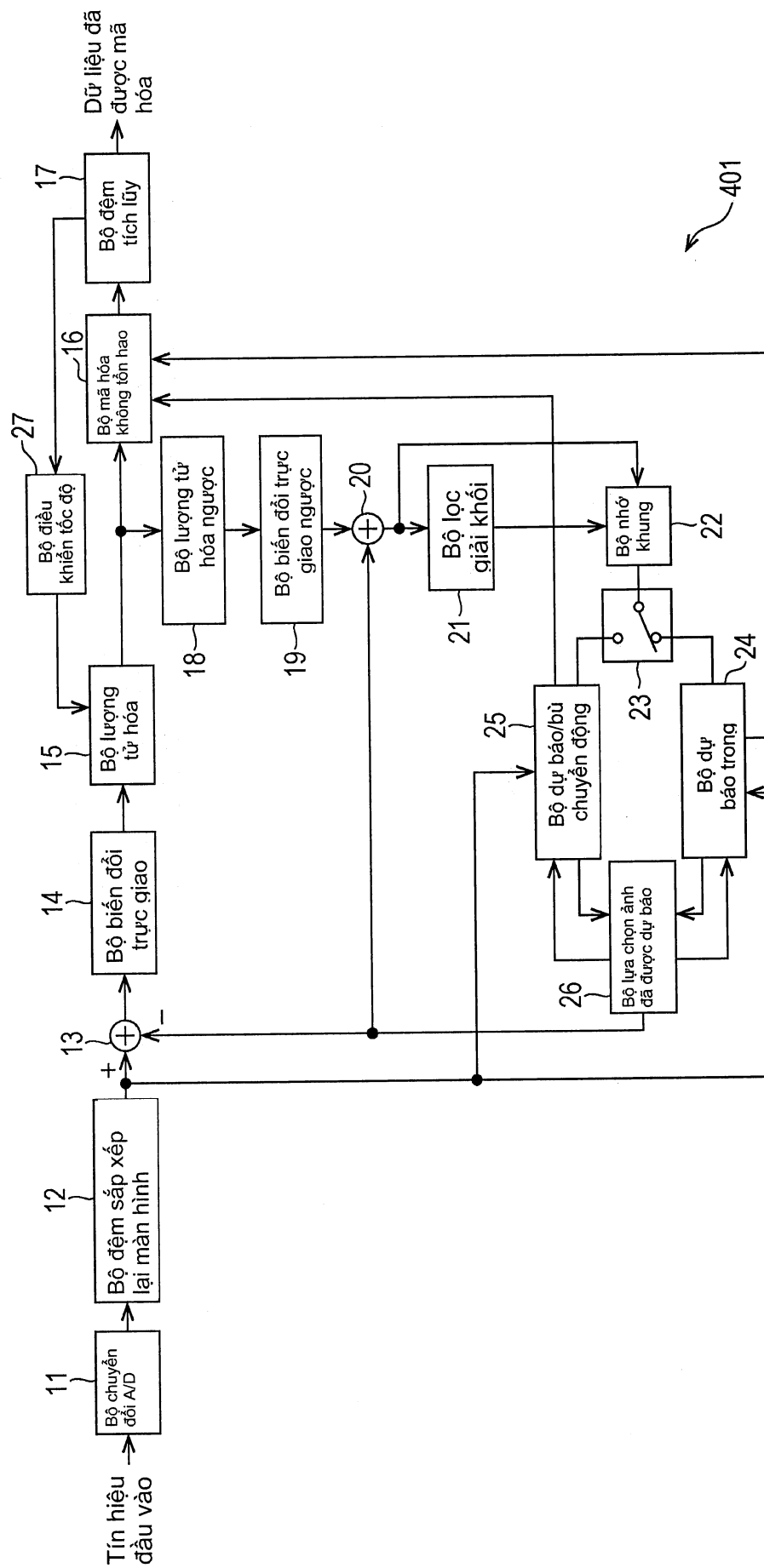


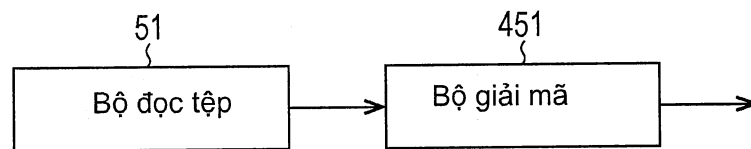
FIG. 39

FIG. 41

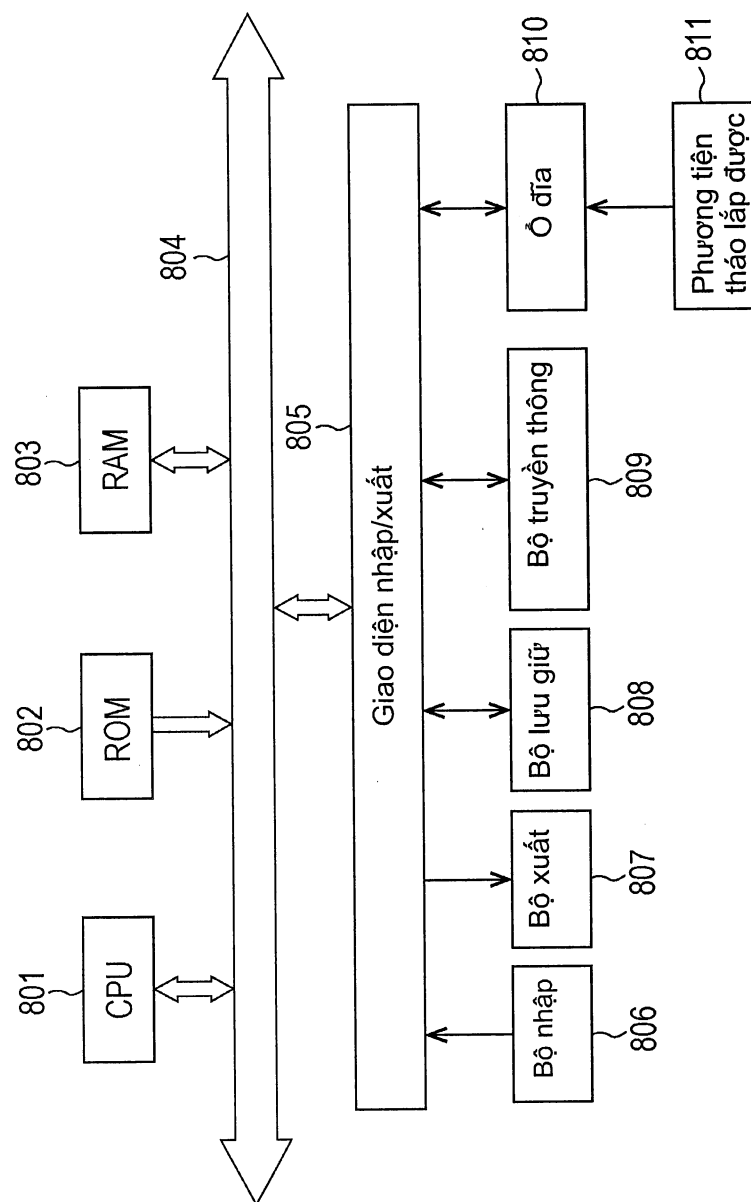


FIG. 42

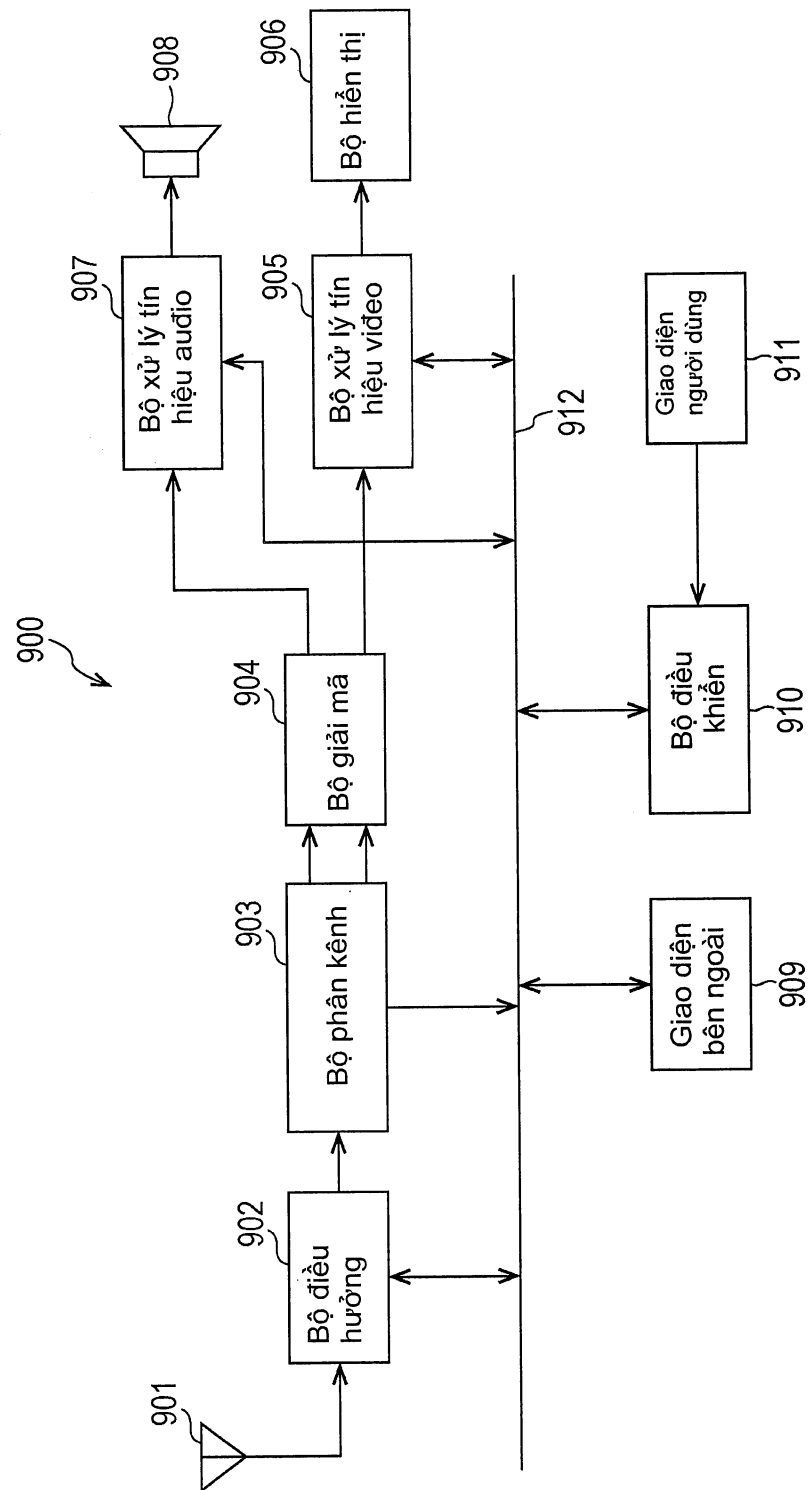


FIG. 43

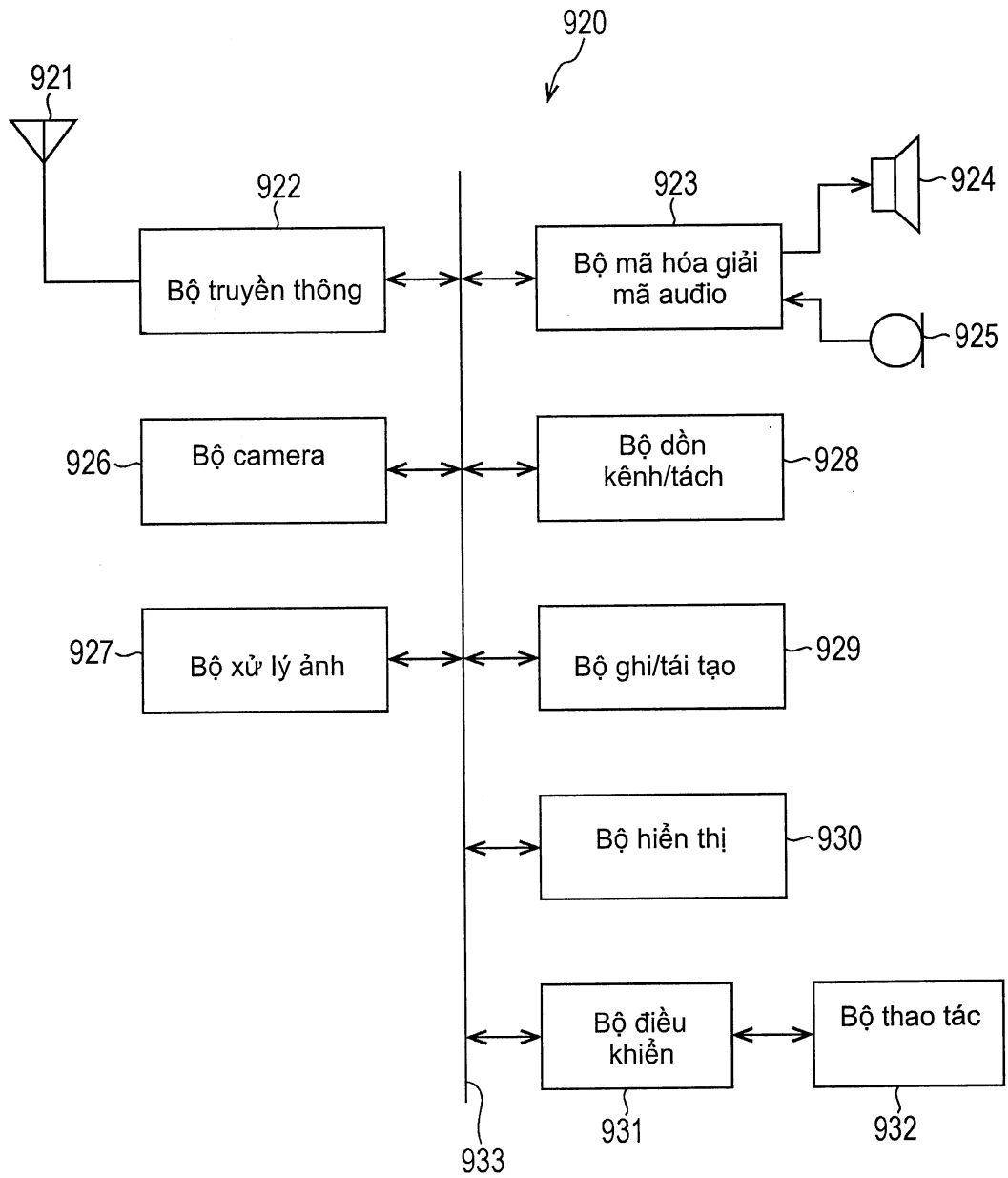


FIG. 44

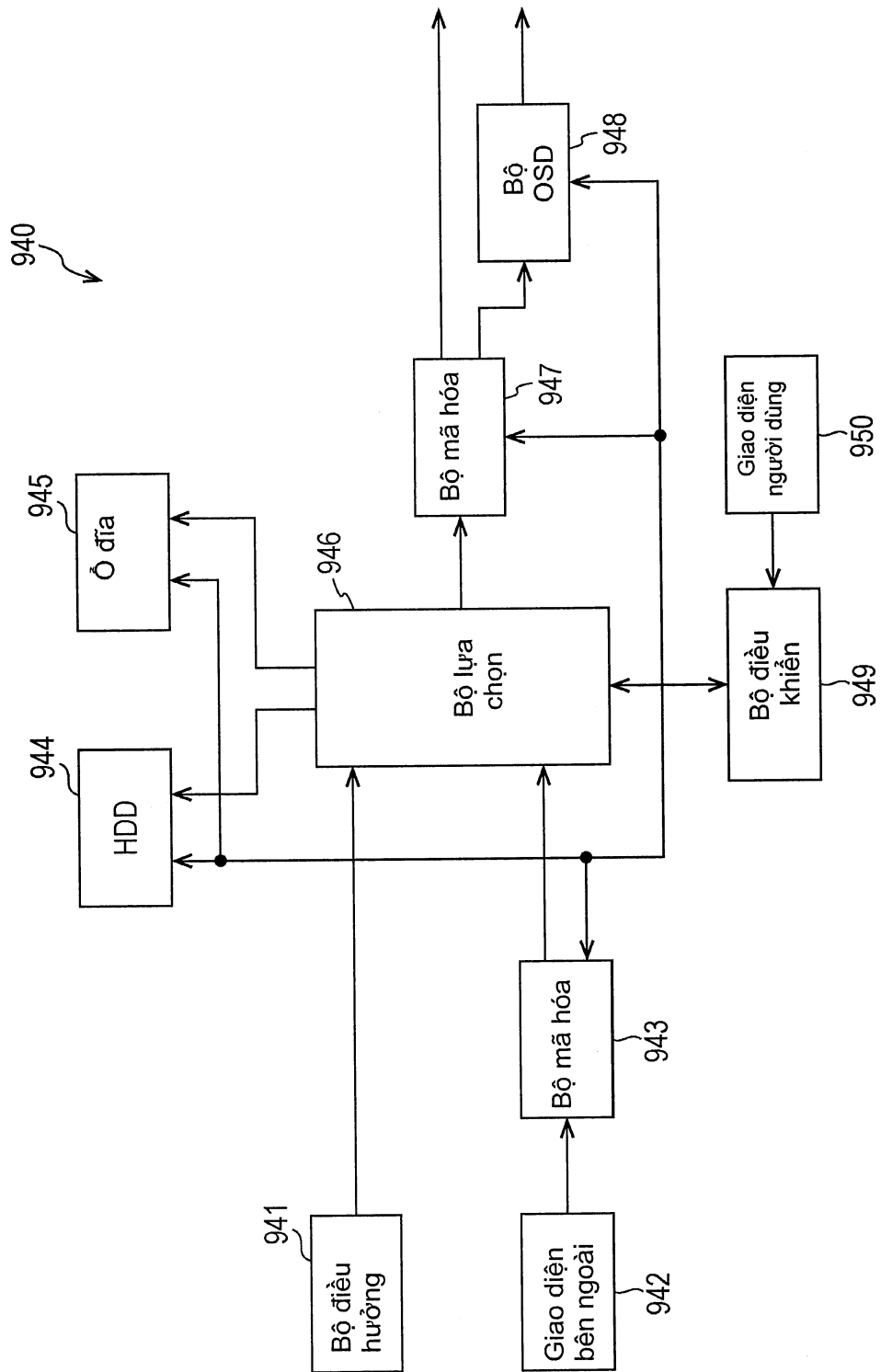


FIG. 45

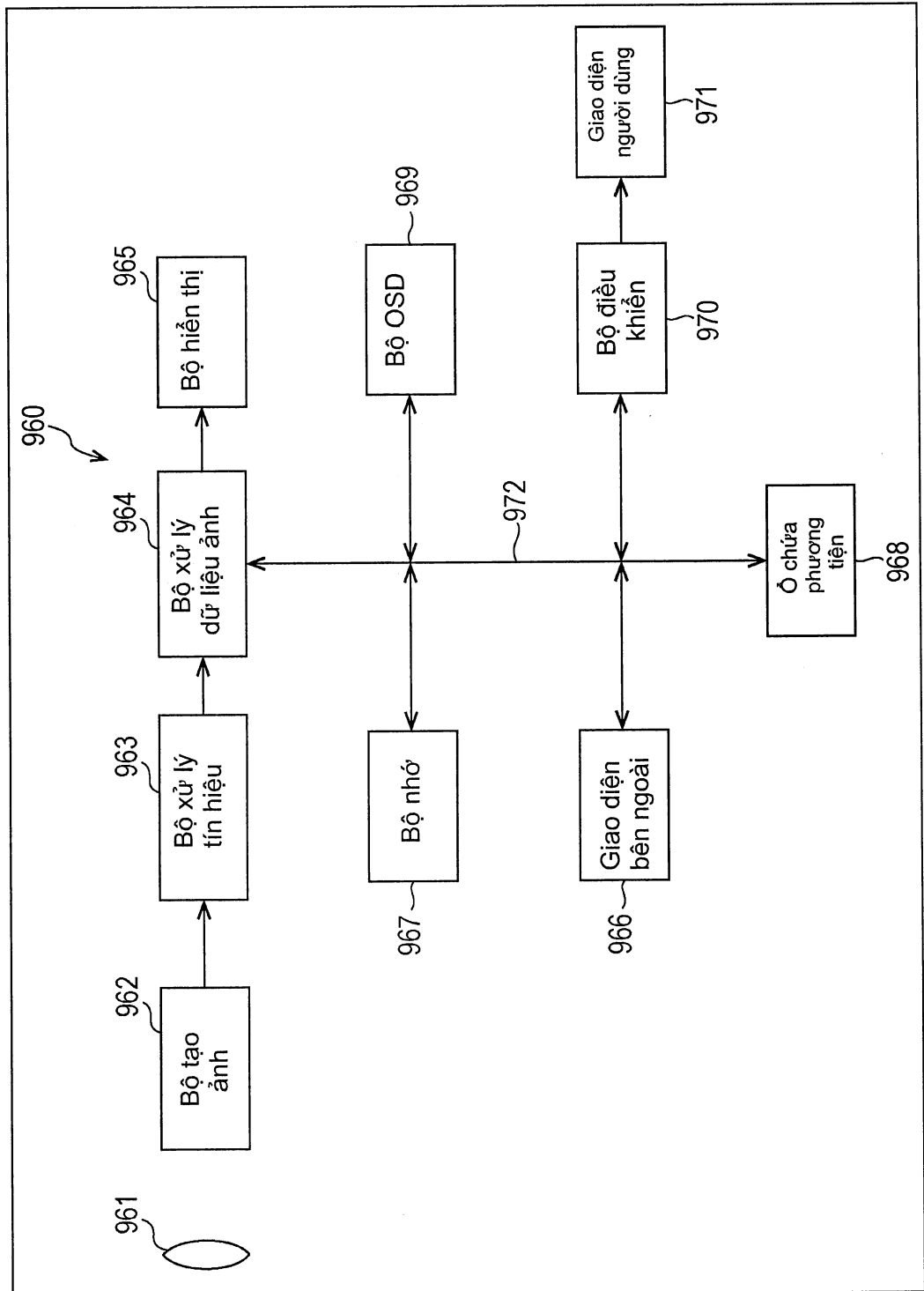


FIG. 46

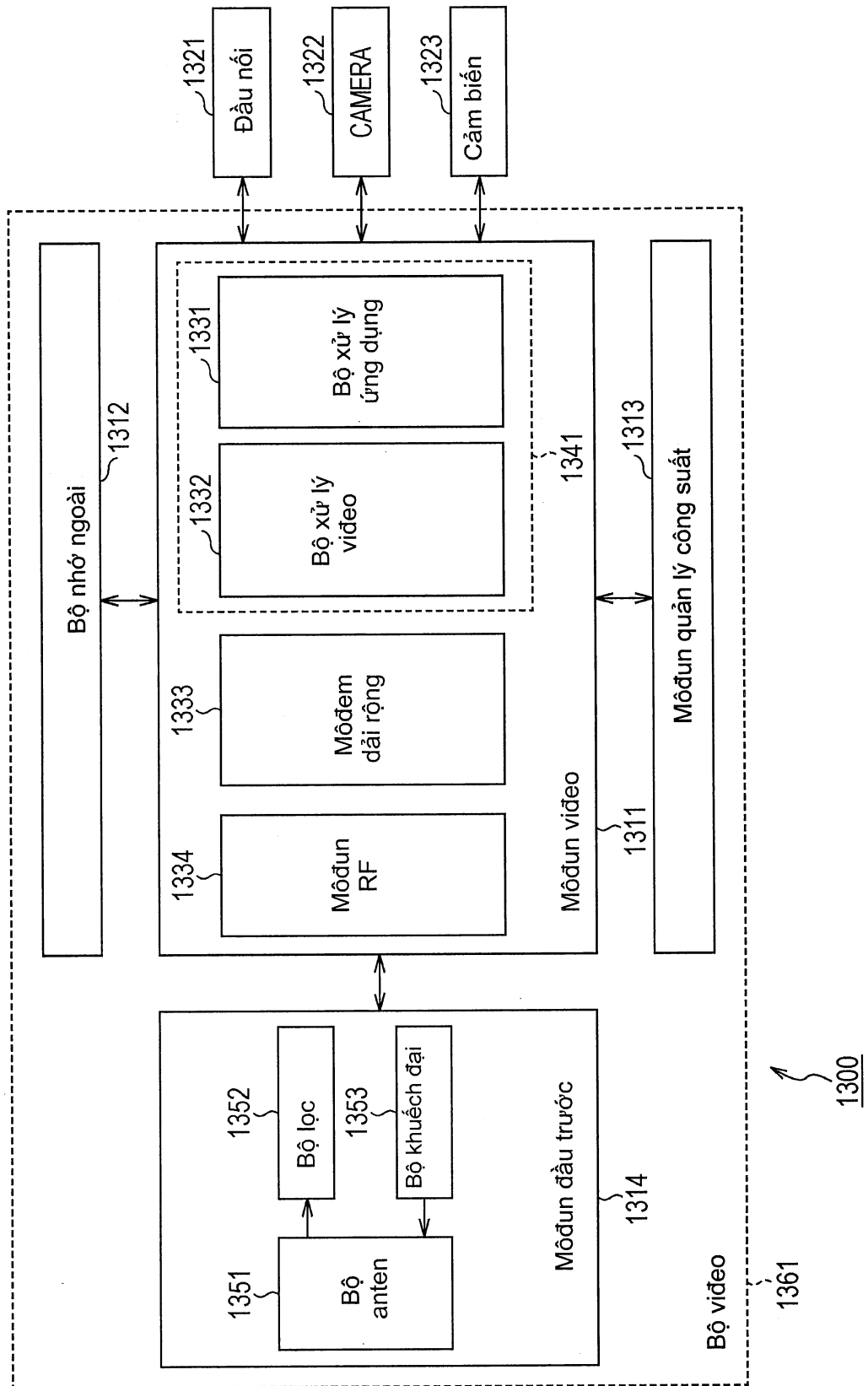


FIG. 47

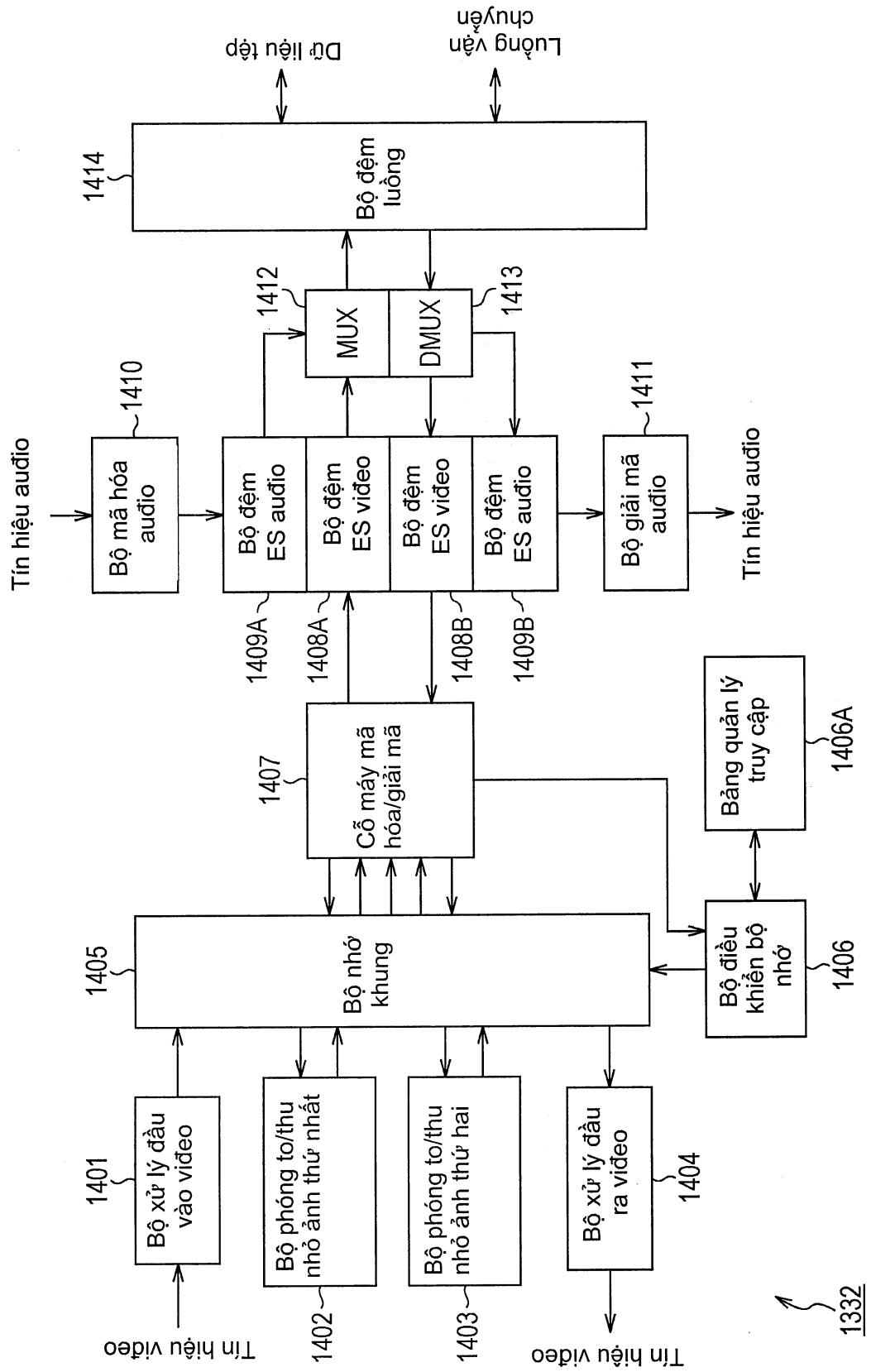


FIG. 48

