



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

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

(11)



1-0028841

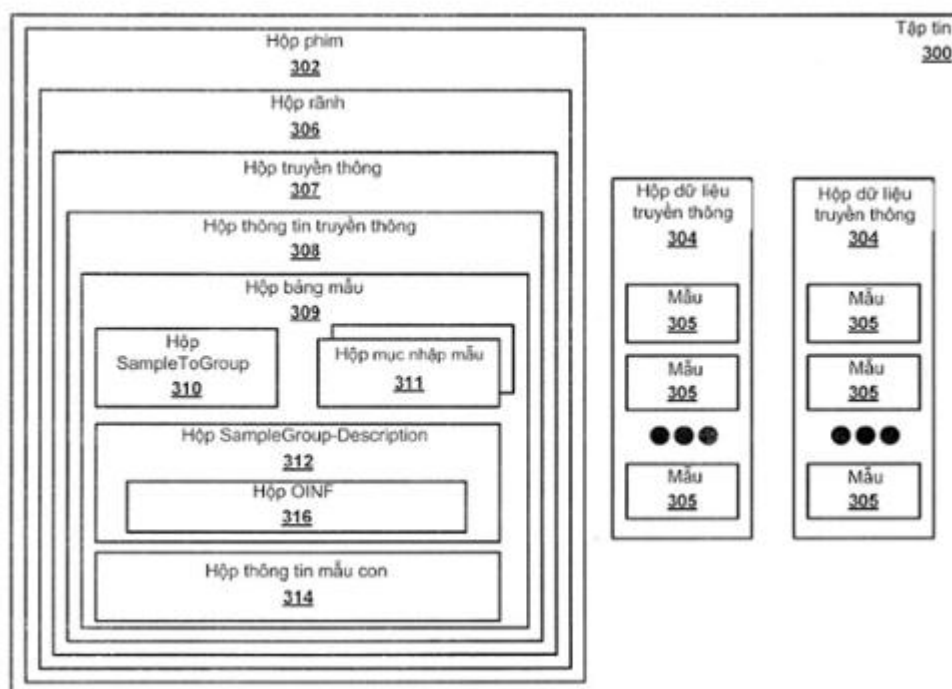
(51)⁷ H04N 19/70; H04N 21/845; H04N
21/854; H04N 21/2343

(13) B

- (21) 1-2017-02883 (22) 10/02/2016
(86) PCT/US2016/017280 10/02/2016 (87) WO 2016/130635 A1 18/08/2016
(30) 62/115,075 11/02/2015 US; 15/019,634 09/02/2016 US
(45) 25/07/2021 400 (43) 27/11/2017 356A
(73) QUALCOMM INCORPORATED (US)
ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, CA
92121-1714, United States of America
(72) HENDRY, Fnu (ID); WANG, Ye-Kui (CN).
(74) Công ty TNHH Quốc tế D&N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP, THIẾT BỊ VIDEO VÀ VẬT GHI BẤT BIẾN ĐỌC ĐƯỢC
BẰNG MÁY TÍNH ĐỂ XỬ LÝ DỮ LIỆU VIDEO NHIỀU LỚP

(57) Sáng chế đề cập đến thiết bị video để xử lý dữ liệu video nhiều lớp bao gồm phương tiện lưu trữ dữ liệu được tạo cấu hình để lưu trữ dữ liệu video nhiều lớp và một hoặc nhiều bộ xử lý được tạo cấu hình để: thu được dữ liệu video nhiều lớp; lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin; lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (operation points information-oinf) cho định dạng tập tin đó; và tạo tập tin của dữ liệu video được định dạng theo định dạng tập tin. Ngoài ra, sáng chế cũng đề cập đến phương pháp và vật ghi bất biến đọc được bằng máy tính để xử lý dữ liệu video nhiều lớp.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc mã hóa video.

Tình trạng kỹ thuật của sáng chế

Các tính năng video số có thể được tích hợp vào một loạt các thiết bị bao gồm ti-vi kỹ thuật số, các hệ thống truyền hình trực tiếp kỹ thuật số, các hệ thống truyền hình không dây, các thiết bị hỗ trợ số cá nhân (personal digital assistant - PDA), các máy tính xách tay hoặc máy tính để bàn, các máy tính bảng, các thiết bị đọc sách điện tử, các máy ảnh số, các thiết bị ghi âm số, các thiết bị phát phương tiện số, các thiết bị chơi trò chơi điện tử, bàn giao tiếp trò chơi điện tử, điện thoại di động hoặc điện thoại vô tuyến vệ tinh, được gọi là “điện thoại thông minh”, các thiết bị hội nghị truyền hình, các thiết bị truyền liên tục video, và tương tự. Các thiết bị video số thực hiện các kỹ thuật nén video, như các kỹ thuật được mô tả trong các chuẩn được xác định bởi MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Phần 10, Mã hóa video tiên tiến (Advanced Video Coding - AVC), chuẩn Mã hóa video hiệu suất cao (High Efficiency Video Coding - HEVC) hiện đang triển khai, và các dạng mở rộng của các chuẩn này. Các thiết bị video có thể truyền, thu, mã hóa, giải mã, và/hoặc lưu trữ thông tin video số một cách có hiệu quả hơn bằng cách thực hiện các kỹ thuật nén video như vậy.

Các kỹ thuật nén video thực hiện dự đoán không gian (nội hình ảnh) và/hoặc dự đoán thời gian (liên hình ảnh) để giảm bớt hoặc loại bỏ phần dư vốn có trong các chuỗi video. Đối với việc mã hóa video trên cơ sở khối, lát video (tức là, khung video hoặc một phần của khung video) có thể được phân chia thành các khối video, còn có thể được gọi là khối cây, đơn vị mã hóa (coding unit-CU) và/hoặc nút mã hóa. Các khối video ở lát được nội mã hóa (I) của hình ảnh được mã hóa nhờ sử dụng phép dự đoán theo không gian đối với các mẫu tham chiếu ở các khối lân cận trong cùng một hình ảnh. Các khối video ở lát được liên mã hóa (P hoặc B) của hình ảnh có thể sử dụng phép dự đoán theo không gian đối với các mẫu tham chiếu ở các khối lân cận trong cùng một hình ảnh hoặc dự đoán theo thời gian đối với các mẫu tham chiếu

trong hình ảnh tham chiếu khác. Hình ảnh có thể được gọi là khung, và hình ảnh tham chiếu có thể được gọi là khung tham chiếu.

Sau khi dữ liệu video được mã hóa, dữ liệu video có thể được đóng gói để truyền hoặc lưu trữ. Dữ liệu video có thể được gắn vào tập tin video theo chuẩn bất kỳ trong số nhiều chuẩn khác nhau, như định dạng tập tin phương tiện cơ sở của Tổ chức Quốc tế về Tiêu chuẩn hóa (International Organization for Standardization - ISO) và các dạng mở rộng của nó, như AVC.

Bản chất kỹ thuật của sáng chế

Nói chung, sáng chế đề cập đến việc lưu trữ nội dung video trong tập tin. Theo một số ví dụ, các kỹ thuật của sáng chế là dựa trên định dạng tập tin phương tiện cơ sở của Tổ chức Quốc tế về Tiêu chuẩn hóa (ISO base media file format - ISOBMFF). Một số ví dụ của bản mô tả này đề cập đến các phương pháp lưu trữ dòng video chứa nhiều lớp mã hóa, trong đó mỗi một lớp có thể là lớp mở rộng được, khung hình kết cấu, khung hình chiều sâu, v.v. và các phương pháp này có thể áp dụng cho lưu trữ dữ liệu mã hóa video hiệu suất cao đa khung hình (Multi-View High Efficiency Video Coding - MV-HEVC), HEVC mở rộng được (Scalable HEVC-SHVC), HEVC ba chiều (3D-HEVC), và các loại dữ liệu video khác.

Theo một ví dụ, phương pháp xử lý dữ liệu video nhiều lớp bao gồm bước thu được dữ liệu video nhiều lớp; lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin; lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin này; và tạo tập tin của dữ liệu video được định dạng theo định dạng tập tin này.

Theo ví dụ khác, phương pháp xử lý dữ liệu video nhiều lớp bao gồm việc thu được tập tin của dữ liệu video nhiều lớp được định dạng theo định dạng tập tin; đối với định dạng tập tin, xác định thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin đó; và giải mã dữ liệu video nhiều lớp dựa trên thông tin định dạng biểu diễn xác định được.

Theo ví dụ khác, thiết bị video để xử lý dữ liệu video nhiều lớp bao gồm phương tiện lưu trữ dữ liệu được tạo cấu hình để lưu trữ dữ liệu video nhiều lớp và một hoặc nhiều bộ xử lý được tạo cấu hình để: thu được dữ liệu video nhiều lớp; lưu

trữ dữ liệu video nhiều lớp dưới định dạng tập tin; lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin này; và tạo tập tin của dữ liệu video được định dạng theo định dạng tập tin này.

Theo ví dụ khác, thiết bị video để xử lý dữ liệu video nhiều lớp bao gồm phương tiện lưu trữ dữ liệu được tạo cấu hình để lưu trữ dữ liệu video nhiều lớp và một hoặc nhiều bộ xử lý được tạo cấu hình để thu được tập tin của dữ liệu video nhiều lớp được định dạng theo định dạng tập tin; đối với định dạng tập tin, xác định thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin đó; và giải mã dữ liệu video nhiều lớp dựa trên thông tin định dạng biểu diễn xác định được.

Theo ví dụ khác, thiết bị video để xử lý dữ liệu video nhiều lớp bao gồm phương tiện để thu được dữ liệu video nhiều lớp; phương tiện để lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin; phương tiện để lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin này; và phương tiện để tạo tập tin của dữ liệu video được định dạng theo định dạng tập tin này.

Theo ví dụ khác, vật ghi đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi sẽ khiến cho một hoặc nhiều bộ xử lý thu được dữ liệu video nhiều lớp; lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin; lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp thông tin điểm hoạt động (oinf) cho định dạng tập tin này; và tạo tập tin của dữ liệu video được định dạng theo định dạng tập tin này.

Chi tiết về một hoặc nhiều ví dụ theo sáng chế được trình bày trên các hình vẽ kèm theo và trong phần mô tả dưới đây. Các đặc điểm, mục đích và ưu điểm khác sẽ trở nên rõ ràng qua phần mô tả, các hình vẽ và các yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối minh họa hệ thống mã hóa và giải mã video làm ví dụ mà có thể sử dụng các kỹ thuật được mô tả trong bản mô tả này.

Fig.2 là sơ đồ khối minh họa bộ mã hóa video làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này.

Fig.3 là sơ đồ khối minh họa bộ giải mã video làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này.

Fig.4 là sơ đồ khối minh họa tập thiết bị làm ví dụ tạo thành một phần của mạng.

Fig.5A là sơ đồ khái niệm minh họa cấu trúc làm ví dụ của một tập tin theo một hoặc nhiều kỹ thuật của bản mô tả này.

Fig.5B là sơ đồ khái niệm minh họa cấu trúc làm ví dụ của một tập tin theo một hoặc nhiều kỹ thuật của bản mô tả này.

Fig.6 là sơ đồ khái niệm minh họa cấu trúc làm ví dụ của một tập tin theo một hoặc nhiều kỹ thuật của bản mô tả này.

Fig.7 là lưu đồ minh họa thao tác làm ví dụ của một thiết bị tạo tập tin theo một hoặc nhiều kỹ thuật của bản mô tả này.

Fig.8 là lưu đồ minh họa thao tác làm ví dụ của một thiết bị đọc tập tin theo một hoặc nhiều kỹ thuật của bản mô tả này.

Mô tả chi tiết sáng chế

Định dạng tập tin truyền thông cơ sở ISO (ISO base media file format - ISOBMFF) là một định dạng tập tin để lưu trữ dữ liệu truyền thông. ISOBMFF có thể mở rộng được để hỗ trợ lưu trữ dữ liệu video theo chuẩn mã hóa video cụ thể. Ví dụ, ISOBMFF trước đây đã được mở rộng để hỗ trợ lưu trữ dữ liệu video theo các chuẩn mã hóa video H.264/AVC và chuẩn mã hóa video hiệu suất cao (High Efficiency Video Coding-HEVC). Hơn nữa, ISOBMFF trước đây đã được mở rộng để hỗ trợ lưu trữ dữ liệu video theo các dạng mở rộng của chuẩn mã hóa đa khung hình (Multi-view coding-MVC) và mã hóa video mở rộng được (Scalable video coding-SVC) của H.264/AVC. MV-HEVC, 3D-HEVC, và SHVC là các dạng mở rộng của chuẩn mã hóa video HEVC hỗ trợ dữ liệu video nhiều lớp. Các đặc điểm được bổ sung vào ISOBMFF để lưu trữ dữ liệu video theo các dạng mở rộng MVC và SVC của chuẩn H.264/AVC là chưa đủ để lưu trữ hiệu quả dữ liệu video theo MV-HEVC, 3D-HEVC, và SHVC. Nói cách khác, nhiều vấn đề khác nhau có thể nảy sinh nếu sử dụng dạng mở rộng của ISOBMFF để lưu trữ dữ liệu video theo các dạng mở rộng MVC và SVC của chuẩn H.264/AVC để lưu trữ dữ liệu video theo MV-HEVC, 3D-HEVC, và SHVC.

Chẳng hạn, không giống như dòng bit phù hợp với các dạng mở rộng MVC hoặc SVC của H.264/AVC, dòng bit phù hợp với MV-HEVC, 3D-HEVC, hoặc SHVC có thể bao gồm đơn vị truy cập chứa các hình ảnh trong điểm truy cập ngẫu nhiên (Intra Random Access Point-IRAP) và các hình ảnh không IRAP. Đơn vị truy cập chứa các hình ảnh IRAP và hình ảnh không IRAP có thể được sử dụng để truy cập ngẫu nhiên theo MV-HEVC, 3D-HEVC, và SHVC. Tuy nhiên, ISO/BMFF và các dạng mở rộng hiện có của nó không đưa ra cách thức nhận dạng các đơn vị truy cập này. Điều này có thể làm cản trở khả năng của thiết bị tính toán khi thực hiện truy cập ngẫu nhiên, chuyển đổi lớp, và các chức năng tương tự khác liên quan đến dữ liệu video nhiều lớp.

Trong khi phần lớn phần mô tả về các kỹ thuật theo sáng chế mô tả MV-HEVC, 3D-HEVC, và SHVC, người đọc sẽ hiểu rằng các kỹ thuật theo sáng chế có thể được áp dụng cho các chuẩn mã hóa video khác và/hoặc dạng mở rộng của chúng.

Như sẽ được giải thích chi tiết hơn ở phần dưới đây, tập tin theo định dạng tập tin HEVC có thể bao gồm một loạt các đối tượng, được gọi là hộp. Một hộp có thể là khối kiến trúc hướng đối tượng được xác định bởi ký hiệu nhận dạng loại và độ dài duy nhất. Sáng chế mô tả kỹ thuật liên quan tới việc tạo tập tin theo định dạng tập tin, và cụ thể hơn là mô tả các kỹ thuật định vị các loại thông tin nhất định trong các hộp nhất định để nâng cao khả năng xử lý các tập tin mà gồm nhiều điểm hoạt động cho thiết bị phát lại.

Trong khi phần lớn phần mô tả về các kỹ thuật theo sáng chế mô tả MV-HEVC, 3D-HEVC, và SHVC, người đọc sẽ hiểu rằng các kỹ thuật theo sáng chế có thể được áp dụng cho các chuẩn mã hóa video khác và/hoặc dạng mở rộng của chúng.

Fig.1 là sơ đồ khối minh họa hệ thống mã hóa và giải mã video 10 làm ví dụ có thể sử dụng các kỹ thuật được mô tả trong bản mô tả này. Như được thể hiện trên Fig.1, hệ thống 10 bao gồm thiết bị nguồn 12 tạo ra dữ liệu video mã hóa sẽ được giải mã sau đó bởi thiết bị đích 14. Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong nhiều loại thiết bị, bao gồm máy tính để bàn, máy tính notebook (tức là, máy tính xách tay), máy tính bảng, hộp giải mã tín hiệu truyền hình, các máy điện thoại cầm tay như điện thoại "thông minh", bàn phím "thông minh", tivi, camera, thiết bị hiển thị, thiết bị phát phương tiện kỹ thuật số, bàn giao tiếp trò chơi video, thiết bị truyền liên tục video hoặc thiết bị tương tự. Trong một số trường hợp, thiết bị

nguồn 12 và thiết bị đích 14 có thể được trang bị để truyền thông không dây. Thiết bị nguồn 12 và thiết bị đích 14 có thể được xem là các thiết bị video.

Trong ví dụ trên Fig.1, thiết bị nguồn 12 bao gồm nguồn video 18, bộ mã hóa video 20, và giao diện đầu ra 22. Theo một số trường hợp, giao diện đầu ra 22 có thể bao gồm bộ điều biến/bộ giải điều biến (môdem) và/hoặc bộ phát. Trong thiết bị nguồn 12, nguồn video 18 có thể bao gồm nguồn như thiết bị thu video, ví dụ, máy quay video, kho lưu trữ video chứa video được thu trước đó, giao diện cấp video để nhận video từ nhà cung cấp nội dung video, và/hoặc hệ đồ họa máy tính để tạo dữ liệu đồ họa máy tính như video nguồn, hoặc kết hợp của các nguồn này. Tuy nhiên, các kỹ thuật được mô tả trong sáng chế có thể áp dụng cho việc mã hóa video nói chung, và có thể được áp dụng cho các ứng dụng không dây và/hoặc nối dây.

Bộ mã hóa video 20 có thể mã hóa video được thu, được thu trước hoặc được tạo ra bởi máy tính. Thiết bị nguồn 12 có thể truyền dữ liệu video mã hóa trực tiếp đến thiết bị đích 14 qua giao diện đầu ra 22 của thiết bị nguồn 12. Dữ liệu video được mã hóa cũng có thể (hoặc theo cách khác) được lưu trữ trên thiết bị lưu trữ 33 để truy cập sau đó bằng thiết bị đích 14 hoặc các thiết bị khác, để giải mã và/hoặc phát lại.

Thiết bị đích 14 bao gồm giao diện đầu vào 28, bộ giải mã video 30 và thiết bị hiển thị 32. Theo một số trường hợp, giao diện đầu vào 28 có thể bao gồm bộ thu và/hoặc môdem. Giao diện đầu vào 28 của thiết bị đích 14 thu dữ liệu video mã hóa qua liên kết 16. Dữ liệu video mã hóa được truyền qua liên kết 16, hoặc được lưu trữ trên thiết bị lưu trữ 33, có thể bao gồm nhiều phần tử cú pháp được tạo ra bởi thiết bị mã hóa video 20 để sử dụng bởi thiết bị giải mã video, như bộ giải mã video 30, trong việc giải mã dữ liệu video. Các phần tử cú pháp này có thể được gộp với dữ liệu video mã hóa truyền trên phương tiện truyền thông, được lưu trữ trên phương tiện lưu trữ hoặc lưu trữ trên máy chủ tập tin.

Thiết bị hiển thị 32 có thể được tích hợp vào hoặc có thể ở bên ngoài thiết bị đích 14. Trong một số ví dụ, thiết bị đích 14 có thể bao gồm thiết bị hiển thị được tích hợp và còn có thể được tạo cấu hình để tương tác với thiết bị hiển thị bên ngoài. Trong các ví dụ khác, thiết bị đích 14 có thể là thiết bị hiển thị. Nói chung, thiết bị hiển thị 32 hiển thị dữ liệu video được giải mã cho người dùng, và có thể bao gồm thiết bị hiển thị bất kỳ trong số nhiều thiết bị hiển thị như màn hình tinh thể lỏng (LCD - Liquid

Crystal Display), màn hình plasma, màn hình điốt phát sáng hữu cơ (OLED - Organic Light Emitting Diode), hoặc loại thiết bị hiển thị khác.

Mỗi bộ mã hóa video 20 và bộ giải mã video 30 có thể được thực thi dưới dạng sơ đồ bất kỳ trong nhiều sơ đồ mạch mã hóa thích hợp, như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (DSP), mạch tích hợp chuyên dụng (Application specific integrated circuits-ASIC), mảng cổng lập trình được dạng trường (Field programmable gate arrays- FPGAs), logic rời rạc, phần mềm, phần cứng, phần sụn hoặc tổ hợp bất kỳ của chúng. Khi các kỹ thuật này được thực hiện một phần trong phần mềm, thì một thiết bị có thể lưu trữ các lệnh cho phần mềm trong vật ghi bất biến đọc được bởi máy tính phù hợp và thực hiện các lệnh ở phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Mỗi bộ mã hóa video 20 và bộ giải mã video 30 có thể được bao gồm trong một hoặc nhiều bộ mã hóa hoặc giải mã, một trong hai bộ này có thể được tích hợp dưới dạng một phần của bộ mã hóa/giải mã kết hợp (CODEC) trong thiết bị tương ứng.

Thiết bị đích 14 có thể nhận dữ liệu video mã hóa để giải mã qua liên kết 16. Liên kết 16 có thể bao gồm loại phương tiện hoặc thiết bị bất kỳ có khả năng di chuyển dữ liệu video đã mã hóa từ thiết bị nguồn 12 đến thiết bị đích 14. Theo một ví dụ, liên kết 16 có thể chứa phương tiện truyền thông để cho phép thiết bị nguồn 12 truyền dữ liệu video được mã hóa trực tiếp đến thiết bị đích 14 theo thời gian thực. Dữ liệu video mã hóa có thể được điều biến theo chuẩn truyền thông, như giao thức truyền thông không dây và được truyền đến thiết bị đích 14. Phương tiện truyền thông có thể bao gồm phương tiện truyền thông không dây hoặc có dây bất kỳ, như phổ tần số vô tuyến (radio frequency - RF) hoặc một hoặc nhiều đường truyền vật lý. Phương tiện truyền thông có thể tạo thành một phần của mạng dựa trên gói, như mạng cục bộ, mạng diện rộng, hoặc mạng toàn cầu như Internet. Phương tiện truyền thông có thể bao gồm các bộ định tuyến, bộ chuyển mạch, trạm cơ sở, hoặc thiết bị khác bất kỳ mà có thể hữu dụng để tạo điều kiện thuận lợi cho việc truyền thông từ thiết bị nguồn 12 đến thiết bị đích 14.

Theo cách khác, giao diện đầu ra 22 có thể xuất dữ liệu mã hóa đến thiết bị lưu trữ 33. Tương tự, giao diện đầu vào 28 có thể truy cập vào thiết bị lưu trữ dữ liệu mã hóa 33. Thiết bị lưu trữ 33 có thể bao gồm thiết bị bất kỳ trong số các phương tiện lưu trữ dữ liệu truy cập phân tán hoặc cục bộ như đĩa cứng, đĩa Blu-ray, DVD, CD-ROM,

bộ nhớ đệm nhanh, bộ nhớ khả biến hoặc không khả biến hoặc phương tiện lưu trữ số thích hợp bất kỳ khác để lưu trữ dữ liệu video mã hóa. Theo ví dụ khác, thiết bị lưu trữ 33 có thể tương đương với máy chủ tập tin hoặc thiết bị lưu trữ trung gian khác có thể lưu trữ video mã hóa được tạo ra bởi thiết bị nguồn 12. Thiết bị đích 14 có thể truy cập dữ liệu video lưu trữ từ thiết bị lưu trữ 33 bằng cách truyền liên tục hoặc tải xuống. Máy chủ tập tin có thể là kiểu máy chủ bất kỳ có khả năng lưu trữ dữ liệu video mã hóa và truyền dữ liệu video mã hóa này đến thiết bị đích 14. Máy chủ tập tin làm ví dụ bao gồm máy chủ web (ví dụ, đối với trang web), máy chủ FTP, thiết bị lưu trữ gắn vào mạng (Network attached storage-NAS), hoặc ổ đĩa cục bộ. Thiết bị đích 14 có thể truy nhập dữ liệu video mã hóa thông qua kết nối dữ liệu tiêu chuẩn bất kỳ, bao gồm kết nối Internet. Kết nối này có thể bao gồm kênh không dây (ví dụ, kết nối Wi-Fi), kết nối có dây (ví dụ, DSL, môđem cáp, v.v.), hoặc kết hợp của cả hai kết nối trên mà thích hợp để truy cập dữ liệu video mã hóa được lưu trữ trên máy chủ tập tin. Việc truyền dữ liệu video mã hóa từ thiết bị lưu trữ 33 có thể là truyền liên tục, tải xuống hoặc kết hợp cả hai kiểu này.

Kỹ thuật theo sáng chế không nhất thiết giới hạn ở các ứng dụng hoặc lắp đặt không dây. Các kỹ thuật này có thể được áp dụng cho việc mã hóa video để hỗ trợ cho ứng dụng bất kỳ trong nhiều ứng dụng đa phương tiện, chẳng hạn như phát sóng truyền hình qua không gian, phát sóng truyền hình cáp, phát sóng truyền hình vệ tinh, truyền liên tục video, chẳng hạn qua mạng Internet, mã hóa dữ liệu video để lưu trữ trên vật ghi lưu trữ dữ liệu, giải mã video số được lưu trên vật ghi lưu trữ dữ liệu, hoặc các ứng dụng khác. Trong một số ví dụ, hệ thống 10 có thể được tạo cấu hình để hỗ trợ truyền video một chiều hoặc hai chiều để hỗ trợ các ứng dụng như truyền liên tục video, phát lại video, phát rộng video, và/hoặc điện thoại video.

Hơn nữa, theo ví dụ trên Fig.1, hệ thống mã hóa video 10 có thể bao gồm thiết bị tạo tập tin 34. Thiết bị tạo tập tin 34 có thể nhận dữ liệu video được mã hóa do thiết bị nguồn 12 tạo ra và tạo tập tin bao gồm dữ liệu video được mã hóa. Thiết bị đích 14 có thể nhận tập tin được tạo ra bởi thiết bị tạo tập tin 34 một cách trực tiếp hoặc thông qua thiết bị lưu trữ 33. Theo các ví dụ khác nhau, thiết bị tạo tập tin 34 có thể bao gồm nhiều thiết bị điện toán khác nhau. Chẳng hạn, thiết bị tạo tập tin 34 có thể bao gồm phần tử mạng nhận biết truyền thông (Media Aware Network Element-MANE), thiết bị tính toán máy chủ, thiết bị tính toán cá nhân, thiết bị tính toán chuyên dụng, thiết bị

tính toán thương mại hoặc loại thiết bị tính toán khác. Theo một số ví dụ, thiết bị tạo tập tin 34 là một phần của mạng phân phối nội dung. Thiết bị tạo tập tin 34 có thể nhận được dữ liệu video đã mã hóa từ thiết bị nguồn 12 thông qua kênh như liên kết 16. Hơn nữa, thiết bị đích 14 có thể nhận tập tin từ thiết bị tạo tập tin 34 thông qua kênh như liên kết 16.

Trong một số cấu hình, thiết bị tạo tập tin 34 có thể là thiết bị video tách biệt với thiết bị nguồn 12 và thiết bị đích 14, trong khi trong các cấu hình khác, thiết bị tạo tập tin 34 có thể được dùng như là một bộ phận của thiết bị nguồn 12 hoặc thiết bị đích 14. Trong các phương án thực hiện khi thiết bị tạo tập tin 34 là một bộ phận của thiết bị nguồn 12 hoặc thiết bị đích 14, thì khi đó thiết bị tạo tập tin 34 có thể chia sẻ một số tài nguyên giống nhau, như bộ nhớ, bộ xử lý và ổ cứng khác, được sử dụng bởi bộ mã hóa video 20 và bộ giải mã video 30. Trong các phương án thực hiện khi thiết bị tạo tập tin 34 là một thiết bị tách biệt, thì khi đó thiết bị tạo tập tin có thể bao gồm bộ nhớ, bộ xử lý, và các đơn vị ổ cứng khác của chính thiết bị đó.

Theo các ví dụ khác, thiết bị nguồn 12 hoặc thiết bị tính toán khác có thể tạo ra tập tin bao gồm dữ liệu video mã hóa. Tuy nhiên, để dễ giải thích, sáng chế mô tả thiết bị tạo tập tin 34 là thiết bị tạo tập tin. Tuy nhiên, cần hiểu rằng các mô tả này được áp dụng cho các thiết bị tính toán nói chung.

Bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo chuẩn nén video, như chuẩn mã hóa video hiệu suất cao (HEVC) hoặc dạng mở rộng của nó. Chuẩn HEVC cũng có thể được gọi là ISO/IEC 23008-2. Gần đây, thiết kế của HEVC đã được hoàn chỉnh bởi Nhóm Cộng tác Chung về Mã hóa Video (Joint Collaboration Team on Video Coding - JCT-VC) của Nhóm Chuyên gia Mã hóa Video ITU-T (Video Coding Experts Group - VCEG) và Nhóm Chuyên gia Hình Ảnh Chuyển Động ISO/IEC (Motion Picture Experts Group - MPEG). Dự thảo HEVC mới nhất và dưới đây được gọi là HEVC WD có sẵn từ http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. Dạng mở rộng đa khung hình cho HEVC, tức là MV-HEVC, cũng đang được phát triển bởi JCT-3V. Dự thảo hoạt động gần đây (Working Draft-WD) của MV-HEVC, có tựa đề “MV-HEVC Draft Text 5” và dưới đây được gọi là MV-HEVC WD5, có sẵn từ http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip. Dạng mở rộng có thể thay đổi được đối với HEVC, tức là SHVC, cũng

đang được phát triển bởi JCT-VC. Một dự thảo (Working Draft - WD) mới đây của SHVC, có tựa đề “High efficiency video coding (HEVC) scalable extension draft 3” và sau đây được gọi là tài liệu SHVC WD3, có thể xem tại http://phenix.it-sudparis.eu/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip.

Bản thảo (WD) gần đây về mở rộng dải của HEVC, có thể xem tại http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1005-v3.zip. Dự thảo làm việc gần đây (WD) của dạng mở rộng 3D của HEVC, tức là 3D-HEVC, có tựa đề “3D-HEVC Draft Text 1” có sẵn từ http://phenix.int-evry.fr/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip. Bộ hóa video 20 và bộ giải mã 30 có thể hoạt động theo một hoặc nhiều chuẩn này.

Theo cách khác, bộ mã hóa video 20 và bộ giải mã video 30 có thể hoạt động theo các tiêu chuẩn độc quyền hoặc công nghiệp khác, như tiêu chuẩn ITU-T H.264, còn được gọi là MPEG 4, Phần 10, mã hoá video tiên tiến (AVC), hoặc các dạng mở rộng của các tiêu chuẩn này. Tuy nhiên, các kỹ thuật theo sáng chế không chỉ giới hạn ở chuẩn mã hóa cụ thể bất kỳ. Các ví dụ khác về chuẩn nén video bao gồm ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual và ITU-T H.264 (còn được biết đến là ISO/IEC MPEG-4 AVC), bao gồm các dạng mở rộng mã hóa video mở rộng được (Scalable Video Coding - SVC) và mã hóa video đa khung hình (Multiview Video Coding - MVC).

Mặc dù không được thể hiện trên Fig.1, nhưng theo một số khía cạnh, mỗi bộ mã hóa video 20 và bộ giải mã video 30 có thể được tích hợp với bộ mã hóa và giải mã âm thanh và có thể bao gồm các đơn vị MUX-DEMUX thích hợp hoặc phân cứng và phần mềm khác, để xử lý mã hóa cả âm thanh và video trong một dòng dữ liệu chung hoặc trong các dòng dữ liệu riêng biệt. Nếu khả dụng, trong một số ví dụ, các đơn vị MUX-DEMUX có thể thực hiện theo giao thức bộ dồn kênh ITU H.223, hoặc các giao thức khác như giao thức gói dữ liệu người dùng (user datagram protocol – UDP).

JCT-VC đã phát triển chuẩn HEVC. Nỗ lực chuẩn hóa HEVC là dựa trên mô hình phát triển của thiết bị mã hóa video được gọi là Mô hình thử nghiệm HEVC (HM). HM giả định vài tính năng phụ trợ của các thiết bị mã hóa video liên quan đến các thiết bị sẵn có theo, ví dụ, ITU-T H.264/AVC. Ví dụ, trong khi H.264/AVC cung

cấp 9 chế độ mã hóa dự đoán nội hình ảnh, thì HEVC HM có thể cung cấp 33 chế độ mã hóa dự đoán nội hình ảnh.

Nói chung, mô hình làm việc của HM mô tả rằng khung hoặc hình ảnh video có thể được chia thành chuỗi các khối cây hoặc đơn vị mã hóa lớn nhất (Largest coding unit-LCU) bao gồm cả mẫu độ sáng lẫn mẫu độ màu. Khối cây cũng có thể được gọi là đơn vị cây mã hóa (Coding Tree Unit-CTU). Khối cây có mục đích giống như khối marco của chuẩn H.264/AVC. Lát bao gồm một số khối cây liên tiếp theo thứ tự mã hóa. Khung hoặc hình ảnh video có thể được chia thành một hoặc nhiều lát. Mỗi khối cây có thể được chia thành các đơn vị mã hóa (CU) theo cây tứ phân. Chẳng hạn, khối cây, là nút rễ của cây tứ phân, có thể được chia thành bốn nút con và mỗi nút con có thể là nút bố mẹ và có thể được chia thành bốn nút con khác. Nút con không được chia cuối cùng, là nút lá của cây tứ phân, bao gồm nút mã hóa, tức là khối video mã hóa. Dữ liệu cú pháp liên quan đến dòng bit mã hóa có thể xác định số lần tối đa một khối cây có thể phân chia và có thể xác định kích thước tối thiểu của nút mã hóa.

CU bao gồm nút mã hóa và các đơn vị dự đoán (PU) và đơn vị biến đổi (TU) gắn với nút mã hóa này. Kích cỡ của CU tương ứng với kích cỡ của nút mã hóa và phải là hình vuông. Kích thước của CU có thể nằm trong khoảng từ 8x8 điểm ảnh đến kích thước của khối cây có tối đa 64x64 điểm ảnh hoặc lớn hơn. Mỗi CU có thể bao gồm một hoặc nhiều PU và một hoặc nhiều TU. Dữ liệu cú pháp gắn với CU có thể mô tả, ví dụ, việc chia CU thành một hoặc nhiều PU. Các chế độ chia có thể khác nhau giữa việc liệu CU được mã hóa theo chế độ bỏ qua hoặc trực tiếp, mã hóa theo chế độ dự đoán nội hình ảnh, hoặc mã hóa chế độ dự đoán liên hình ảnh. PU có thể được phân chia thành hình dạng không phải hình vuông. Dữ liệu cú pháp gắn với CU cũng có thể mô tả, ví dụ, việc chia CU thành một hoặc nhiều TU theo cây tứ phân. TU có thể là hình vuông hoặc không phải hình vuông.

Chuẩn HEVC cho phép các kỹ thuật biến đổi theo các TU, có thể là khác nhau đối với các CU khác nhau. Các TU thường được định cỡ dựa trên kích thước của các PU trong CU cho trước được xác định cho LCU phân chia, mặc dù có thể không luôn đúng như vậy. Các TU thường có kích thước bằng hoặc nhỏ hơn các PU. Trong một số ví dụ, các mẫu dư tương ứng với CU có thể được chia nhỏ thành các đơn vị nhỏ hơn bằng cách sử dụng cấu trúc cây tứ phân gọi là “cây tứ phân dư” (residual quad tree - RQT). Các nút lá của RQT có thể được gọi là các TU. Các giá trị vi sai điểm ảnh gắn

với các TU có thể được biến đổi để tạo ra các hệ số biến đổi mà có thể được lượng tử hóa.

Nhìn chung, PU bao gồm dữ liệu liên quan đến quy trình dự đoán. Ví dụ, khi PU được mã hóa theo chế độ nội hình ảnh, thì PU có thể bao gồm dữ liệu mô tả chế độ dự đoán nội hình ảnh cho PU. Theo một ví dụ khác, khi PU được mã hóa theo chế độ liên hình ảnh, PU có thể bao gồm dữ liệu mà xác định vector chuyển động đối với PU. Dữ liệu xác định vector chuyển động cho PU có thể mô tả, ví dụ, thành phần ngang của vector chuyển động, thành phần dọc của vector chuyển động, độ phân giải cho vector chuyển động (ví dụ, độ chính xác một phần tư điểm ảnh hoặc độ chính xác một phần tám điểm ảnh), hình ảnh tham chiếu mà vector chuyển động hướng đến, và/hoặc danh sách hình ảnh tham chiếu (ví dụ, Danh sách 0, Danh sách 1, hoặc Danh sách C) đối với vector chuyển động.

Nhìn chung, TU được sử dụng cho quy trình biến đổi và lượng tử hóa. CU cho trước có một hoặc nhiều PU có thể còn bao gồm một hoặc nhiều đơn vị biến đổi (TU). Sau khi dự đoán, bộ mã hóa video 20 có thể tính toán giá trị dư tương ứng với PU. Các giá trị dư bao gồm các giá trị vi sai điểm ảnh mà có thể được biến đổi thành các hệ số biến đổi, được lượng tử hóa, và được quét bằng cách sử dụng các TU để tạo ra các hệ số biến đổi nối tiếp để mã hóa entropy. Bản mô tả này thường sử dụng thuật ngữ “khối video” để chỉ nút mã hóa (tức là, khối mã hóa) của CU. Trong một số trường hợp cụ thể, bản mô tả này cũng có thể sử dụng thuật ngữ “khối video” để chỉ khối cây, tức là, LCU, hoặc CU, bao gồm nút mã hóa và các PU và TU.

Chuỗi video thường bao gồm dãy các khung hoặc các hình ảnh video. Nhóm hình ảnh (group of picture - GOP) thường bao gồm một loạt một hoặc nhiều hình ảnh video. GOP có thể chứa dữ liệu cú pháp trong phần đầu của GOP, phần đầu của một hoặc nhiều hình ảnh, hoặc vị trí khác, để mô tả một số hình ảnh có trong GOP. Mỗi lát của hình có thể chứa dữ liệu cú pháp lát để mô tả chế độ mã hóa cho lát tương ứng. Bộ mã hóa video 20 thường hoạt động trên các khối video trong các lát video riêng lẻ để mã hóa dữ liệu video. Khối video có thể tương ứng với nút mã hóa trong CU. Các khối video có thể có kích cỡ cố định hoặc thay đổi, và có thể khác nhau về kích cỡ theo chuẩn mã hóa riêng.

Ví dụ, HM hỗ trợ dự đoán trong các kích cỡ PU khác nhau. Giả định rằng kích cỡ của CU cụ thể là $2N \times 2N$, HM hỗ trợ dự đoán nội hình ảnh ở các kích cỡ PU là

$2N \times 2N$ hoặc $N \times N$, và dự đoán liên hình ảnh ở các kích cỡ PU đối xứng là $2N \times 2N$, $2N \times N$, $N \times 2N$, hoặc $N \times N$. HM cũng hỗ trợ sự phân chia không đối xứng cho dự đoán liên hình ảnh ở các kích cỡ PU là $2N \times nU$, $2N \times nD$, $nL \times 2N$, và $nR \times 2N$. Trong quá trình phân chia bất đối xứng, một chiều của CU không được chia, trong khi chiều còn lại được chia thành 25% và 75%. Phần của CU tương ứng với phần chia 25% được biểu thị bằng “n” tiếp đó là chỉ báo “Lên”, “Xuống”, “Trái” hoặc “Phải”. Do vậy, ví dụ, “ $2N \times nU$ ” chỉ CU $2N \times 2N$ được chia theo chiều ngang với PU $2N \times 0,5N$ ở trên và PU $2N \times 1,5N$ ở dưới.

Theo bản mô tả này, “ $N \times N$ ” và “N nhân N” có thể được sử dụng thay thế nhau để chỉ kích thước điểm ảnh của khối video theo chiều dọc và chiều ngang, ví dụ, 16×16 điểm ảnh hoặc 16 nhân 16 điểm ảnh. Nói chung, khối 16×16 có 16 điểm ảnh theo chiều dọc ($y = 16$) và 16 điểm ảnh theo chiều ngang ($x = 16$). Tương tự, khối $N \times N$ thường có N điểm ảnh theo chiều dọc và N điểm ảnh theo chiều ngang, trong đó N là giá trị nguyên không âm. Các điểm ảnh trong một khối có thể được sắp xếp thành các hàng và các cột. Ngoài ra, các khối không nhất thiết phải có số lượng điểm theo chiều ngang bằng số lượng điểm ảnh theo chiều dọc. Ví dụ, các khối có thể có $N \times M$ điểm ảnh, trong đó M không nhất thiết bằng N.

Tiếp theo quy trình mã hóa dự đoán nội hình ảnh hoặc liên hình ảnh sử dụng các PU của CU, bộ mã hóa video 20 có thể tính toán dữ liệu dư cho các TU của CU. PU có thể bao gồm dữ liệu điểm ảnh trong miền không gian (còn được gọi là miền điểm ảnh) và TU có thể bao gồm các hệ số trong miền biến đổi sau khi áp dụng phép biến đổi, ví dụ, phép biến đổi cosin rời rạc (DCT), biến đổi số nguyên, biến đổi sóng con hoặc phép biến đổi tương tự về mặt khái niệm đối với dữ liệu video dư. Dữ liệu dư có thể tương ứng với vi sai điểm ảnh giữa các điểm ảnh của hình ảnh không mã hóa và các giá trị dự đoán tương ứng với các PU. Bộ mã hóa video 20 có thể tạo ra các TU bao gồm dữ liệu dư cho CU, và sau đó biến đổi các TU để tạo ra hệ số biến đổi cho CU.

Sau phép biến đổi bất kỳ để tạo ra các hệ số biến đổi, bộ mã hóa video 20 có thể thực hiện lượng tử hóa các hệ số biến đổi này. Lượng tử hóa thường chỉ quy trình trong đó các hệ số biến đổi được lượng tử hóa để có thể giảm khối lượng dữ liệu được sử dụng để biểu diễn các hệ số biến đổi, tạo ra sự nén thêm. Quy trình lượng tử hóa có

thể giảm độ sâu bit gắn với một số hoặc tất cả các hệ số. Ví dụ, giá trị n bit có thể được làm tròn xuống giá trị m bit trong quá trình lượng tử hóa, trong đó n lớn hơn m .

Trong một số ví dụ, bộ mã hóa video 20 có thể sử dụng thứ tự quét được định trước để quét các hệ số biến đổi lượng tử hóa nhằm tạo ra vector nối tiếp mà có thể được mã hóa entropy. Trong các ví dụ khác, bộ mã hóa video 20 có thể thực hiện quét thích nghi. Sau khi quét các hệ số biến đổi đã lượng tử hóa để tạo ra vector một chiều, bộ mã hóa video 20 có thể mã hóa entropy vector một chiều này, ví dụ, theo phương pháp mã hóa chiều dài biến đổi thích ứng với ngữ cảnh (context adaptive variable length coding-CAVLC), mã hóa thuật toán nhị phân tương thích ngữ cảnh (Context adaptive binary arithmetic coding-CABAC), mã hóa thuật toán nhị phân tương thích với ngữ cảnh dựa trên cú pháp (Syntax-based context-adaptive binary arithmetic coding-SBAC), Mã hóa entropy phân chia khoảng xác suất (Probability Interval Partitioning Entropy-PIPE) hoặc phương pháp mã hóa entropy khác. Bộ mã hóa video 20 cũng có thể mã hóa entropy các phần tử cú pháp liên quan đến dữ liệu video mã hóa để bộ giải mã video 30 sử dụng để giải mã dữ liệu video.

Để thực hiện CABAC, bộ mã hóa video 20 có thể gán ngữ cảnh trong mô hình ngữ cảnh cho ký hiệu cần được truyền. Ngữ cảnh có thể liên quan tới, ví dụ, việc các giá trị lân cận của ký hiệu có khác không hay không chẳng hạn. Để thực hiện CAVLC, bộ mã hóa video 20 có thể chọn mã độ dài biến đổi cho ký hiệu cần được truyền. Các từ mã trong mã hóa độ dài biến đổi (VLC - variable length coding) có thể được tạo ra sao cho các mã tương đối ngắn hơn tương ứng với các ký hiệu xác suất cao hơn, còn các mã dài hơn tương ứng với các ký hiệu xác suất thấp hơn. Theo cách này, việc sử dụng VLC có thể tiết kiệm bit so với, ví dụ, sử dụng các từ mã độ dài bằng nhau cho mỗi ký hiệu cần được truyền. Việc xác định xác suất có thể dựa vào ngữ cảnh được gán cho ký hiệu.

Bộ mã hóa video 20 có thể xuất dòng bit mà bao gồm chuỗi bit tạo ra dạng biểu diễn các hình ảnh mã hóa và dữ liệu liên quan. Thuật ngữ “dòng bit” có thể là thuật ngữ chung được sử dụng để chỉ dòng đơn vị lớp trừu tượng hóa mạng (Network Abstraction Layer-NAL) (ví dụ, chuỗi đơn vị NAL) hoặc dòng byte (ví dụ, đóng gói dòng đơn vị NAL chứa tiền tố mã bắt đầu và đơn vị NAL như được thể hiện bởi phụ lục B của chuẩn HEVC). Đơn vị NAL là cấu trúc cú pháp chứa chỉ báo về kiểu dữ liệu trong đơn vị NAL và byte chứa dữ liệu ở dạng tải tin chuỗi byte thô (Raw byte

sequence payload-RBSP) được đặt rải rác khi cần với các bit ngăn chặn sự mô phỏng. Mỗi đơn vị NAL có thể bao gồm tiêu đề đơn vị NAL và có thể đóng gói một RBSP. Tiêu đề đơn vị NAL có thể bao gồm phần tử cú pháp chỉ báo mã kiểu đơn vị NAL. Mã kiểu đơn vị NAL được chỉ rõ bởi tiêu đề đơn vị NAL của đơn vị NAL chỉ báo kiểu đơn vị NAL. RBSP có thể là cấu trúc cú pháp chứa số lượng byte nguyên mà được đóng gói trong đơn vị NAL. Trong một số trường hợp, RBSP bao gồm các bit 0.

Các kiểu đơn vị NAL khác nhau có thể đóng gói các loại RBSP khác nhau. Ví dụ, kiểu đơn vị NAL thứ nhất có thể đóng gói RBSP cho PPS, kiểu đơn vị NAL thứ hai có thể đóng gói RBSP cho đoạn lát, kiểu đơn vị NAL thứ ba có thể đóng gói RBSP cho SEI, v.v.. Đơn vị NAL mà đóng gói RBSP cho dữ liệu mã hóa video (như trái ngược với RBSP cho tập tham số và thông báo SEI) có thể được gọi là đơn vị NAL lớp mã hóa video (Video coding layer-VCL). Các đơn vị NAL chứa tập tham số (ví dụ, tập tham số video (video parameter set - VPS), tập tham số chuỗi (sequence parameter set - SPS, PPS, v.v.) có thể được gọi là đơn vị NAL tập tham số.

Bản mô tả này đề cập đến đơn vị NAL đóng gói RBSP cho phần đoạn dưới dạng đơn vị NAL phần mã hóa. Như được định nghĩa trong HEVC WD, đoạn lát là một số nguyên các CTU được sắp xếp liên tiếp khi quét ô và được chứa trong một đơn vị NAL. Ngược lại, theo HEVC WD, lát có thể là số nguyên các CTU có trong một đoạn lát độc lập và tất cả các đoạn lát phụ thuộc kế tiếp (nếu có) đứng trước đoạn lát độc lập tiếp theo (nếu có) trong cùng một đơn vị truy cập. Đoạn lát độc lập là đoạn lát mà các giá trị của phần tử cú pháp của tiêu đề đoạn lát cho đoạn lát đó không được suy ra từ các giá trị cho đoạn lát đứng trước. Đoạn lát phụ thuộc là đoạn lát mà các giá trị của một số phần tử cú pháp của tiêu đề đoạn lát được suy ra từ các giá trị cho đoạn lát độc lập đứng trước theo thứ tự giải mã. RBSP của đơn vị NAL đã mã hóa lát có thể bao gồm tiêu đề đoạn lát và dữ liệu lát. Tiêu đề đoạn lát là một phần của đoạn lát đã mã hóa chứa các phần tử dữ liệu liên quan đến CTU thứ nhất hoặc tất cả các CTU được biểu diễn trong đoạn lát đó. Tiêu đề lát là tiêu đề đoạn lát của đoạn lát độc lập là đoạn lát hiện thời hoặc đoạn lát độc lập gần nhất đứng trước đoạn lát hiện thời theo thứ tự giải mã.

VPS là cấu trúc cú pháp bao gồm các phần tử cú pháp áp dụng cho không hoặc nhiều chuỗi video được mã hóa (Coded video sequence-CVS) trọn vẹn. SPS là cấu trúc cú pháp bao gồm các phần tử cú pháp áp dụng cho không hoặc nhiều CVS trọn

ven. SPS có thể bao gồm phần tử cú pháp nhận dạng VPS hoạt động khi SPS hoạt động. Do vậy, các phần tử cú pháp của VPS có thể có khả năng áp dụng một cách rộng rãi hơn các phần tử cú pháp của SPS.

Tập tham số (ví dụ, VPS, SPS, PPS, v.v.) có thể chứa nhận dạng được tham chiếu, trực tiếp hoặc gián tiếp, từ tiêu đề lát của lát. Quy trình tham chiếu được biết đến là "sự kích hoạt". Do vậy, khi bộ giải mã video 30 đang giải mã một lát cụ thể, tập tham số được tham chiếu, trực tiếp hoặc gián tiếp, bằng phần tử cú pháp trong tiêu đề lát của lát cụ thể được cho là "được kích hoạt". Tùy thuộc vào kiểu tập tham số, sự kích hoạt có thể xảy ra trên cơ sở hình ảnh hoặc trên cơ sở chuỗi. Ví dụ, tiêu đề lát của lát có thể bao gồm phần tử cú pháp nhận dạng PPS. Do vậy, khi bộ mã hóa video mã hóa lát, PPS được kích hoạt. Hơn nữa, PPS có thể bao gồm phần tử cú pháp nhận dạng SPS. Do vậy, khi PPS nhận dạng SPS được kích hoạt, SPS có thể được kích hoạt. SPS có thể bao gồm một phần tử cú pháp nhận dạng VPS. Do vậy, khi SPS nhận dạng VPS được kích hoạt, VPS được kích hoạt.

Bộ giải mã video 30 có thể nhận dòng bit được tạo ra bởi bộ mã hóa video 20. Ngoài ra, bộ giải mã video 30 có thể phân tích dòng bit này để thu được các phần tử cú pháp từ dòng bit này. Bộ giải mã video 30 có thể tái tạo các hình ảnh của dữ liệu video dựa ít nhất một phần vào các phần tử cú pháp thu được từ dòng bit. Quy trình tái tạo dữ liệu video có thể thường ngược với quy trình được thực hiện bởi bộ mã hóa video 20. Ví dụ, bộ giải mã video 30 có thể sử dụng các vector chuyển động của PU để xác định các khối dự đoán cho PU của CU hiện thời. Ngoài ra, bộ giải mã video 30 có thể lượng tử hóa ngược khối hệ số của TU của CU hiện thời. Bộ giải mã video 30 có thể thực hiện các biến đổi ngược trên các khối hệ số để tái tạo khối biến đổi của TU của CU hiện thời. Bộ giải mã video 30 có thể tái tạo khối mã hóa của CU hiện thời bằng cách bổ sung các mẫu của khối dự đoán cho các PU của CU hiện thời vào các mẫu tương ứng của khối biến đổi của các TU của CU hiện thời. Bằng cách tái tạo các khối mã hóa cho mỗi CU của hình ảnh, bộ giải mã video 30 có thể tái tạo hình ảnh này.

Theo HEVC WD, CVS có thể bắt đầu từ hình ảnh làm mới giải mã tức thời (Instantaneous Decoding Refresh-IDR), hoặc hình ảnh truy cập liên kết gãy (Broken link access-BLA) hoặc hình ảnh truy cập ngẫu nhiên sạch (Clean random access - CRA) là hình ảnh đầu tiên trong dòng bit, bao gồm tất cả các hình ảnh tiếp theo không

phải hình ảnh IDR hoặc BLA. Hình ảnh IDR chỉ chứa lát I (tức là, các lát trong đó chỉ sử dụng dự đoán nội hình ảnh). Hình ảnh IDR có thể là hình ảnh đầu tiên trong dòng bit theo thứ tự mã hóa hoặc có thể xuất hiện muộn hơn trong dòng bit. Mỗi hình ảnh IDR là hình ảnh thứ nhất của CVS theo thứ tự giải mã. Theo HEVC WD, hình ảnh IDR có thể là hình ảnh trong điểm truy cập ngẫu nhiên (Intra random access point (IRAP)) mà đối với hình ảnh này, mỗi đơn vị NAL VCL có `nal_unit_type` bằng `IDR_W_RADL` hoặc `IDR_N_LP`.

Các hình ảnh IDR có thể được sử dụng để truy cập ngẫu nhiên. Tuy nhiên, các hình ảnh sau hình ảnh IDR theo thứ tự giải mã không thể sử dụng các hình ảnh được giải mã trước hình ảnh IDR làm tham chiếu. Do đó, dòng bit dựa vào các hình ảnh IDR để truy cập ngẫu nhiên có hiệu suất mã hóa thấp hơn đáng kể so với dòng bit sử dụng các loại hình ảnh truy cập ngẫu nhiên khác. Theo ít nhất một số ví dụ, đơn vị truy cập IDR là đơn vị truy cập chứa hình ảnh IDR.

Khái niệm về hình ảnh CRA được đưa vào HEVC để cho phép các hình ảnh sau hình ảnh CRA theo thứ tự giải mã, nhưng đứng trước hình ảnh CRA theo thứ tự xuất, sử dụng các hình ảnh đã được giải mã trước hình ảnh CRA để tham chiếu. Các hình ảnh đứng sau hình ảnh CRA theo thứ tự giải mã, nhưng đứng trước hình ảnh CRA theo thứ tự xuất, được gọi là các hình ảnh dẫn đầu gắn với hình ảnh CRA (hoặc các hình ảnh dẫn đầu của hình ảnh CRA). Tức là, để cải thiện hiệu quả mã hóa, khái niệm về hình ảnh CRA được đưa vào HEVC để cho phép các hình ảnh đứng sau hình ảnh CRA theo thứ tự giải mã, nhưng đứng trước hình ảnh CRA theo thứ tự xuất, sử dụng các hình ảnh đã được giải mã trước hình ảnh CRA để tham chiếu. Điểm truy cập CRA là điểm truy cập trong đó hình ảnh mã hóa là hình ảnh CRA. Theo HEVC WD, hình ảnh CRA là hình ảnh trong điểm truy cập ngẫu nhiên mà đối với hình ảnh này, mỗi đơn vị NAL VCL có `nal_unit_type` bằng `CRA_NUT`.

Các hình ảnh dẫn đầu của hình ảnh CRA có thể được giải mã chính xác nếu việc giải mã bắt đầu từ hình ảnh IDR hoặc hình ảnh CRA đứng trước hình ảnh CRA theo thứ tự giải mã. Tuy nhiên, hình ảnh dẫn đầu của hình ảnh CRA có thể không được giải mã khi việc truy cập ngẫu nhiên từ hình ảnh CRA xảy ra. Do vậy, bộ giải mã video thường giải mã các hình ảnh dẫn đầu của hình ảnh CRA trong quá trình giải mã truy cập ngẫu nhiên. Để ngăn chặn sự lan truyền lỗi từ các hình ảnh tham chiếu mà có thể là không khả dụng tùy thuộc vào quá trình giải mã bắt đầu ở đâu, không có

hình ảnh nào đứng sau hình ảnh CRA theo cả thứ tự giải mã lẫn thứ tự xuất có thể sử dụng hình ảnh bất kỳ đứng trước hình ảnh CRA theo thứ tự giải mã hoặc thứ tự xuất (bao gồm các hình ảnh dẫn đầu) để tham chiếu.

Khái niệm về hình ảnh BLA được đưa vào HEVC sau khi đưa các hình ảnh CRA và dựa trên khái niệm về hình ảnh CRA. Hình ảnh BLA thường bắt đầu từ dòng bit ghép nối ở vị trí hình ảnh CRA và trong dòng bit được ghép nối, hình ảnh CRA tại điểm ghép nối được biến đổi thành hình ảnh BLA. Do vậy, hình ảnh BLA có thể là hình ảnh CRA trong dòng bit ban đầu và hình ảnh CRA được biến đổi thành hình ảnh BLA bởi bộ ghép nối dòng bit sau khi ghép nối dòng bit ở vị trí hình ảnh CRA. Theo một số trường hợp, đơn vị truy cập chứa hình ảnh RAP có thể được gọi là đơn vị truy cập RAP. Đơn vị truy cập BLA là đơn vị truy cập chứa hình ảnh BLA. Theo HEVC WD, hình ảnh BLA có thể là hình ảnh trong điểm truy cập ngẫu nhiên mà đối với hình ảnh này, mỗi đơn vị NAL VCL có `nal_unit_type` bằng `BLA_W_LP`, `BLA_W_RADL`, hoặc `BLA_N_LP`.

Nói chung, hình ảnh IRAP chỉ chứa các lát I và có thể là hình ảnh BLA, hình ảnh CRA hoặc hình ảnh IDR. Chẳng hạn, HEVC WD chỉ ra rằng hình ảnh IRAP có thể là hình ảnh mã hóa mà đối với hình ảnh này mỗi đơn vị NAL VCL có `nal_unit_type` nằm trong khoảng từ `BLA_W_LP` đến `RSV_IRAP_VCL23`, bao gồm cả các giá trị đầu mút. Hơn nữa, HEVC WD chỉ ra rằng hình ảnh đầu tiên trong dòng bit theo thứ tự giải mã phải là hình ảnh IRAP. Bảng 7-1 của HEVC WD thể hiện các mã kiểu đơn vị NAL và nhóm kiểu đơn vị NAL. Bảng 7-1 của HEVC WD được đưa ra dưới đây.

Bảng 7-1 – Các mã kiểu đơn vị NAL và nhóm kiểu đơn vị NAL

nal_unit_type	Tên của nal_unit_type	Nội dung của đơn vị NAL và cấu trúc cú pháp RBSP	Nhóm kiểu đơn vị NAL
0 1	TRAIL_N TRAIL_R	Đoạn lát mã hóa của hình ảnh kế tiếp không TSA, không STSA slice_segment_layer_rbsp()	VCL
2 3	TSA_N TSA_R	Đoạn lát mã hóa của hình ảnh TSA slice_segment_layer_rbsp()	VCL
4 5	STSA_N STSA_R	Đoạn lát mã hóa của hình ảnh STSA slice_segment_layer_rbsp()	VCL
6 7	RADL_N RADL_R	Đoạn lát mã hóa của hình ảnh RADL slice_segment_layer_rbsp()	VCL
8 9	RASL_N RASL_R	Đoạn lát mã hóa của hình ảnh RASL slice_segment_layer_rbsp()	VCL
10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Kiểu đơn vị NAL VCL không tham chiếu của lớp con không IRAP dự trữ	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Kiểu đơn vị NAL VCL tham chiếu của lớp con không IRAP dự trữ	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Đoạn lát mã hóa của hình ảnh BLA slice_segment_layer_rbsp()	VCL
19 20	IDR_W_RADL IDR_N_LP	Đoạn lát mã hóa của hình ảnh IDR slice_segment_layer_rbsp()	VCL
21	CRA_NUT	Đoạn lát mã hóa của hình ảnh CRA slice_segment_layer_rbsp()	VCL

22	RSV_IRAP_VCL22	Kiểu đơn vị NAL VCL IRAP dự trữ	VCL
23	RSV_IRAP_VCL23		
24..31	RSV_VCL24.. RSV_VCL31	Kiểu đơn vị NAL VCL không IRAP dự trữ	VCL
32	VPS_NUT	Tập tham số video video_parameter_set_rbsp()	không VCL
33	SPS_NUT	Tập tham số chuỗi seq_parameter_set_rbsp()	không VCL
34	PPS_NUT	Tập tham số hình ảnh pic_parameter_set_rbsp()	không VCL
35	AUD_NUT	Ký hiệu tách đơn vị truy nhập access_unit_delimiter_rbsp()	không VCL
36	EOS_NUT	Cuối của chuỗi end_of_seq_rbsp()	không VCL
37	EOB_NUT	Cuối của dòng bit end_of_bitstream_rbsp()	không VCL
38	FD_NUT	Dữ liệu đệm filler_data_rbsp()	không VCL
39	PREFIX_SEI_NUT	Thông tin nâng cao bổ sung	không
40	SUFFIX_SEI_NUT	sei_rbsp()	VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Dự trữ	không VCL
48..63	UNSPEC48.. UNSPEC63	Không xác định	không VCL

Một sự khác nhau giữa hình ảnh BLA và hình ảnh CRA là như sau. Đối với hình ảnh CRA, hình ảnh dẫn đầu liên quan có thể được giải mã chính xác nếu việc giải mã bắt đầu từ hình ảnh RAP đứng trước hình ảnh CRA theo thứ tự giải mã. Tuy

nhiên, hình ảnh dẫn đầu liên quan đến hình ảnh CRA không thể được giải mã chính xác khi xảy ra sự truy cập ngẫu nhiên từ hình ảnh CRA này (tức là, khi việc giải mã bắt đầu từ hình ảnh CRA hoặc nói cách khác, khi hình ảnh CRA là hình ảnh đầu tiên trong dòng bit). Ngược lại, không có trường hợp nào trong đó hình ảnh dẫn đầu liên quan đến hình ảnh BLA có thể giải mã được, ngay cả khi việc giải mã bắt đầu từ hình ảnh RAP đứng trước hình ảnh BLA theo thứ tự giải mã.

Một số hình ảnh dẫn đầu liên quan đến hình ảnh CRA cụ thể hoặc hình ảnh BLA cụ thể có thể được giải mã chính xác ngay cả khi hình ảnh CRA hoặc hình ảnh BLA cụ thể là hình ảnh thứ nhất trong dòng bit. Các hình ảnh dẫn đầu này có thể được gọi là hình ảnh dẫn đầu giải mã được (decodable leading picture-DLP) hoặc hình ảnh dẫn đầu giải mã truy cập ngẫu nhiên (Random Access Decodable Leading-RADL). Theo HEVC WD, hình ảnh RADL có thể là hình ảnh mã hóa mà đối với hình ảnh này, mỗi đơn vị NAL VCL có `nal_unit_type` bằng `RADL_R` hoặc `RADL_N`. Hơn nữa, HEVC WD chỉ ra rằng tất cả các hình ảnh RADL là hình ảnh dẫn đầu và rằng các hình ảnh RADL không được sử dụng làm hình ảnh tham chiếu cho quá trình giải mã các hình ảnh kế tiếp của cùng một hình ảnh IRAP liên quan. Khi có mặt, tất cả các hình ảnh RADL đứng trước, theo thứ tự giải mã, tất cả các hình ảnh kế tiếp của cùng một hình ảnh IRAP liên quan. HEVC WD chỉ ra rằng đơn vị truy cập RADL có thể là đơn vị truy cập trong đó hình ảnh mã hóa là hình ảnh RADL. Hình ảnh kế tiếp có thể là hình ảnh đứng sau hình ảnh IRAP liên quan (tức là, hình ảnh IRAP trước đó theo thứ tự giải mã) theo thứ tự xuất.

Các hình ảnh dẫn đầu khác có thể được gọi là hình ảnh dẫn đầu không giải mã được (non-decodable leading picture-NLP) hoặc hình ảnh dẫn đầu bỏ qua truy cập ngẫu nhiên (Random Access Skipped Leading -RSDL). Theo HEVC WD, hình ảnh RSDL có thể là hình ảnh mã hóa mà đối với hình ảnh này, mỗi đơn vị NAL VCL có `nal_unit_type` bằng `RASL_R` hoặc `RASL_N`. Tất cả các hình ảnh RASL là hình ảnh dẫn đầu của hình ảnh BLA hoặc CRA liên quan.

Nếu các tập tham số cần thiết là khả dụng khi chúng cần được kích hoạt, hình ảnh IRAP và tất cả các hình ảnh không RASL tiếp theo theo thứ tự giải mã có thể được giải mã chính xác mà không cần thực hiện quá trình giải mã hình ảnh bất kỳ đứng trước hình ảnh IRAP theo thứ tự giải mã. Có thể có các hình ảnh trong dòng bit chỉ chứa lát I không phải là hình ảnh IRAP.

Theo kỹ thuật mã hóa đa khung hình, có thể có nhiều khung hình của cùng một cảnh từ các điểm nhìn khác nhau. Thuật ngữ “điểm truy cập” có thể được sử dụng để chỉ tập hợp các hình ảnh tương ứng với cùng một thời điểm thời gian. Do vậy, dữ liệu video có thể được khái niệm hóa là một loạt đơn vị truy cập xảy ra theo thời gian. "Thành phần khung hình" có thể là dạng biểu diễn mã hóa của khung hình trong một đơn vị truy cập. Trong bản mô tả này, “khung hình” có thể chỉ chuỗi hoặc tập hợp các thành phần khung hình liên quan tới cùng một ký hiệu nhận dạng khung hình. Thành phần khung hình có thể chứa thành phần khung hình kết cấu và thành phần khung hình chiều sâu. Trong bản mô tả này, “khung hình” có thể là chuỗi hoặc tập gồm một hoặc nhiều thành phần khung hình có cùng một ký hiệu nhận dạng khung hình.

Thành phần khung hình kết cấu (tức là, hình ảnh kết cấu) có thể là dạng biểu diễn mã hóa của kết cấu của khung hình trong một đơn vị truy cập. Khung hình kết cấu có thể là chuỗi các thành phần khung hình kết cấu có liên quan tới giá trị giống nhau của chỉ số thứ tự khung hình. Chỉ số thứ tự khung hình của một khung hình có thể chỉ báo vị trí máy quay của khung hình này so với vị trí của các khung hình khác. Thành phần khung hình chiều sâu (tức là, hình ảnh chiều sâu) có thể là dạng biểu diễn mã hóa về chiều sâu của khung hình trong một đơn vị truy cập. Khung hình chiều sâu có thể là một tập hoặc chuỗi gồm một hoặc nhiều thành phần khung hình chiều sâu có liên quan tới giá trị giống nhau của chỉ số thứ tự khung hình.

Theo MV-HEVC, 3D-HEVC và SHVC, bộ mã hóa video có thể tạo ra dòng bit bao gồm một dãy đơn vị NAL. Các đơn vị NAL khác nhau của dòng bit có thể được kết hợp với các lớp khác nhau của dòng bit. Một lớp có thể được định nghĩa là tập đơn vị NAL VCL và đơn vị NAL không VCL liên quan có cùng ký hiệu nhận dạng lớp. Một lớp có thể tương đương với một khung hình trong mã hóa video đa khung hình. Theo mã hóa video đa khung hình, một lớp có thể chứa tất cả các thành phần khung hình của cùng một lớp với các thời điểm khác nhau. Mỗi thành phần khung hình có thể là hình ảnh mã hóa của cảnh video thuộc một khung hình cụ thể ở một thời điểm cụ thể. Trong một số ví dụ về mã hóa video 3D, một lớp có thể chứa tất cả các hình ảnh chiều sâu mã hóa của một khung hình cụ thể hoặc các hình ảnh kết cấu mã hóa của một khung hình cụ thể. Trong các ví dụ khác về mã hóa video 3D, một lớp có thể chứa cả thành phần khung hình kết cấu và thành phần khung hình chiều sâu của một khung hình cụ thể. Tương tự, trong trường hợp mã hóa video có thể mở rộng được,

một lớp thường tương ứng với các hình ảnh mã hóa có các đặc điểm video khác với hình ảnh mã hóa trong các lớp khác. Các đặc điểm video này thường bao gồm độ phân giải không gian và mức chất lượng (ví dụ, tỷ lệ tín hiệu so với tiếng ồn). Theo HEVC và các dạng mở rộng của nó, khả năng mở rộng về thời gian có thể đạt được trong một lớp bằng cách xác định nhóm hình ảnh có mức thời gian cụ thể làm lớp con.

Đối với mỗi lớp tương ứng của dòng bit, dữ liệu ở lớp thấp hơn có thể được giải mã mà không cần tham chiếu đến dữ liệu ở lớp cao hơn bất kỳ. Theo phương pháp mã hóa video có thể mở rộng được, ví dụ dữ liệu ở lớp cơ sở có thể được giải mã mà không cần tham chiếu đến dữ liệu ở lớp nâng cao. Nói chung, đơn vị NAL chỉ có thể đóng gói dữ liệu của một lớp đơn. Do vậy, đơn vị NAL đóng gói dữ liệu của lớp còn lại cao nhất của dòng bit có thể được loại bỏ ra khỏi dòng bit mà không làm ảnh hưởng đến khả năng giải mã dữ liệu ở các lớp còn lại của dòng bit. Theo kỹ thuật mã hóa đa khung hình và 3D-HEVC, các lớp cao hơn có thể bao gồm các thành phần khung hình bổ sung. Theo SHVC, các lớp cao hơn có thể bao gồm dữ liệu nâng cao tỷ lệ tín hiệu so với tiếng ồn (signal to noise ratio-SNR), dữ liệu nâng cao về không gian và/hoặc dữ liệu nâng cao về thời gian. Theo MV-HEVC, 3D-HEVC và SHVC, một lớp có thể được gọi là "lớp cơ sở" nếu bộ giải mã video có thể giải mã các hình ảnh trong lớp này mà không cần tham chiếu đến dữ liệu của lớp khác bất kỳ. Lớp cơ sở có thể phù hợp với đặc điểm kỹ thuật cơ bản của HEVC (ví dụ, HEVC WD).

Theo SVC, các lớp không phải là lớp cơ sở có thể được gọi là "lớp nâng cao" và có thể cung cấp thông tin làm tăng chất lượng nhìn thấy của dữ liệu video được giải mã từ dòng bit. SVC có thể làm tăng độ phân giải không gian, tỷ lệ tín hiệu so với tiếng ồn (tức là, chất lượng) hoặc tốc độ thời gian. Theo kỹ thuật mã hóa video có thể mở rộng (ví dụ, SHVC), "dạng biểu diễn lớp" có thể là dạng biểu diễn mã hóa của lớp không gian trong một đơn vị truy cập đơn lẻ. Để thuận tiện cho việc giải thích, bản mô tả này có thể đề cập đến các thành phần khung hình và/hoặc dạng biểu diễn lớp dưới dạng "thành phần khung hình/ dạng biểu diễn lớp" hoặc đơn giản là "các hình ảnh".

Để sử dụng các lớp theo chuẩn HEVC, các tiêu đề của các đơn vị NAL bao gồm phần tử cú pháp `nuh_layer_id`, trước đó được gọi là phần tử cú pháp `nuh_reserved_zero_6bits` trong nhiều bản dự thảo ra đời trước bản chuẩn HEVC cuối cùng. Theo chuẩn HEVC cơ sở, phần tử cú pháp `nuh_layer_id` bị giới hạn ở giá trị là 0. Tuy nhiên, theo MV-HEVC, 3D-HEVC, và SVC, phần tử cú pháp `nuh_layer_id` có

thể lớn hơn 0 để chỉ rõ ký hiệu nhận dạng lớp. Các đơn vị NAL của một dòng bit có phần tử cú pháp `nuh_layer_id` mô tả các giá trị khác nhau thuộc các lớp khác nhau của dòng bit.

Trong một số ví dụ, phần tử cú pháp `nuh_layer_id` của đơn vị NAL bằng 0 nếu đơn vị NAL liên quan đến lớp cơ sở trong mã hóa đa khung hình (ví dụ, MV-HEVC), mã hóa 3DV (ví dụ, 3D-HEVC), hoặc mã hóa video mở rộng được (ví dụ, SHVC). Dữ liệu ở lớp cơ sở của dòng bit có thể được giải mã mà không cần tham chiếu đến dữ liệu ở lớp bất kỳ khác của dòng bit. Nếu đơn vị NAL không liên quan đến lớp cơ sở trong mã hóa đa khung hình, 3DV, hoặc mã hóa video mở rộng được, phần tử cú pháp `nuh_layer_id` của đơn vị NAL có thể có giá trị khác không.

Hơn thế nữa, một số thành phần khung hình/ dạng biểu diễn lớp trong một lớp có thể được giải mã mà không cần tham chiếu đến thành phần khung hình/ dạng biểu diễn lớp khác trong cùng một lớp. Do vậy, đơn vị NAL đóng gói dữ liệu của một số thành phần khung hình/ dạng biểu diễn lớp của một lớp có thể được loại bỏ ra khỏi dòng bit mà không làm ảnh hưởng đến khả năng giải mã các thành phần khung hình/ dạng biểu diễn lớp khác trong lớp này. Việc loại bỏ các đơn vị NAL đóng gói dữ liệu của các thành phần khung hình/ dạng biểu diễn lớp này có thể làm giảm tốc độ khung hình của dòng bit. Tập con của các thành phần khung hình/ dạng biểu diễn lớp trong một lớp có thể được giải mã mà không cần tham chiếu đến các thành phần khung hình/ dạng biểu diễn lớp khác trong lớp có thể được gọi là "lớp con" hoặc "lớp con thời gian".

Các đơn vị NAL có thể bao gồm các phần tử cú pháp `temporal_id` mô tả các ký hiệu nhận dạng thời gian (ví dụ, TemporalID) của các đơn vị NAL. Ký hiệu nhận dạng thời gian của đơn vị NAL nhận dạng lớp con mà đơn vị NAL thuộc vào. Do vậy, mỗi lớp con của dòng bit có thể có ký hiệu nhận dạng thời gian khác nhau. Nói chung, nếu ký hiệu nhận dạng của đơn vị NAL thứ nhất của một lớp nhỏ hơn ký hiệu nhận dạng của đơn vị NAL thứ hai của cùng một lớp, dữ liệu được đóng gói bởi đơn vị NAL thứ nhất có thể được giải mã mà không cần tham chiếu đến dữ liệu được đóng gói bởi đơn vị NAL thứ hai.

Một dòng bit có thể kết hợp với nhiều điểm hoạt động. Mỗi điểm hoạt động của dòng bit được kết hợp với tập ký hiệu nhận dạng lớp (ví dụ, tập các giá trị `nuh_layer_id`) và ký hiệu nhận dạng thời gian. Tập các ký hiệu nhận dạng lớp có thể

được biểu thị là `OpLayerIdSet` và ký hiệu nhận dạng thời gian có thể được biểu thị là `TemporalID`. Nếu ký hiệu nhận dạng lớp của một đơn vị NAL nằm trong tập các ký hiệu nhận dạng lớp của điểm hoạt động và ký hiệu nhận dạng thời gian của đơn vị NAL nhỏ hơn hoặc bằng ký hiệu nhận dạng thời gian của điểm hoạt động, đơn vị NAL được kết hợp với điểm hoạt động. Do vậy, điểm hoạt động có thể tương đương với tập con các đơn vị NAL trong dòng bit. HEVC định nghĩa điểm hoạt động là dòng bit được tạo ra từ dòng bit khác bằng hoạt động của quá trình tách dòng bit con với dòng bit khác, `TemporalId` đích cao nhất, và danh sách ký hiệu nhận dạng lớp đích làm đầu vào.

Như được giới thiệu ở trên, bản mô tả này đề cập đến việc lưu trữ nội dung video trong một tập tin dựa trên định dạng tập tin truyền thông cơ sở ISO (ISOBMFF). Cụ thể, bản mô tả này mô tả các kỹ thuật để lưu trữ các dòng video chứa nhiều lớp mã hóa, trong đó mỗi lớp có thể là lớp mở rộng được, khung hình kết cấu, khung hình chiều sâu hoặc các loại lớp hoặc khung hình khác. Các kỹ thuật của bản mô tả này có thể được áp dụng để, ví dụ, lưu trữ dữ liệu video MV-HEVC, dữ liệu video SHVC, dữ liệu video 3D-HEVC, và/hoặc các loại dữ liệu video khác.

Các định dạng tập tin và các chuẩn định dạng tập tin sẽ được mô tả khái quát. Các chuẩn định dạng tập tin bao gồm định dạng tập tin truyền thông cơ sở ISO (ISOBMFF, ISO/IEC 14496-12, sau đây được gọi là “ISO/IEC 14996-12”) và các chuẩn định dạng tập tin khác bắt nguồn từ ISOBMFF, bao gồm định dạng tập tin MPEG-4 (ISO/IEC 14496-14), định dạng tập tin 3GPP (3GPP TS 26.244) và định dạng tập tin AVC (ISO/IEC 14496-15, sau đây được gọi là “ISO/IEC 14996-15”). Do đó, ISO/IEC 14496-12 chỉ rõ định dạng tập tin phương tiện cơ sở ISO. Các tài liệu khác mở rộng định dạng tập tin truyền thông cơ sở ISO cho các ứng dụng cụ thể. Ví dụ, ISO/IEC 14496-15 mô tả việc lưu trữ video được tạo cấu trúc bởi đơn vị NAL theo định dạng tập tin truyền thông cơ sở ISO. H.264/AVC và HEVC, cũng như các dạng mở rộng của chúng, là các ví dụ về video được tạo cấu trúc bởi đơn vị NAL. ISO/IEC 14496-15 bao gồm các phần mô tả việc mang các đơn vị NAL H.264/AVC. Ngoài ra, phần 8 của ISO/IEC 14496-15 mô tả việc mang các đơn vị NAL HEVC.

ISOBMFF được sử dụng làm cơ sở cho nhiều định dạng đóng gói mã hóa, như định dạng tập tin AVC, cũng như cho nhiều định dạng bộ chứa đa phương tiện như định dạng tập tin MPEG-4, định dạng tập tin 3GPP (3GP) và định dạng tập tin DVB.

Bên cạnh các phương tiện truyền liên tục như âm thanh và video thì các phương tiện tĩnh như ảnh cũng như siêu dữ liệu có thể được lưu trữ trong tập tin theo chuẩn ISOBMFF. Các tập tin được tạo cấu trúc theo chuẩn ISOBMFF có thể được sử dụng cho nhiều mục đích, bao gồm phát lại tập tin truyền thông cục bộ, tải liên tục tập tin từ xa, các đoạn để tạo dòng tương thích động theo HTTP (DASH), bộ chứa cho nội dung được tạo dòng và các lệnh đóng gói của nó và ghi lại các dòng truyền thông thời gian thực nhận được. Do vậy, mặc dù ban đầu được thiết kế để lưu trữ, ISOBMFF đã tỏ ra có giá trị đối với việc tạo dòng, ví dụ, để tải liên tục hoặc DASH. Đối với mục đích tạo dòng, các đoạn phim được xác định theo ISOBMFF có thể được sử dụng.

Tập tin theo định dạng tập tin HEVC có thể bao gồm một loạt các đối tượng, được gọi là hộp. Một hộp có thể là khối kiến trúc hướng đến đối tượng được xác định bởi ký hiệu nhận dạng loại duy nhất và chiều dài. Chẳng hạn, một hộp có thể là cấu trúc phân tử cú pháp theo ISOBMFF, bao gồm loại hộp được mã hóa bởi bốn ký tự, tổng số byte của hộp này và tải tin. Nói cách khác, một hộp có thể là cấu trúc cú pháp bao gồm loại hộp được mã hóa, tổng số byte của hộp này và tải tin. Theo một số trường hợp, tất cả các dữ liệu trong một tập tin theo định dạng tập tin HEVC có thể được chứa trong các hộp và có thể không có dữ liệu nào trong tập tin không ở trong hộp. Do vậy, tập tin ISOBMFF có thể bao gồm một chuỗi các hộp và các hộp có thể chứa các hộp khác. Chẳng hạn, tải tin của hộp có thể bao gồm một hoặc nhiều hộp khác. Fig.5A, Fig.5B và Fig.6, như được mô tả chi tiết trong phần khác của bản mô tả này, thể hiện các hộp làm ví dụ trong tập tin, theo một hoặc nhiều kỹ thuật của bản mô tả này.

Tập tin theo ISOBMFF có thể bao gồm các loại hộp khác nhau. Ví dụ, tập tin theo ISOBMFF có thể bao gồm hộp kiểu tập tin, hộp dữ liệu truyền thông, hộp phim ảnh, hộp đoạn phim, v.v... Theo ví dụ này, hộp loại tập tin bao gồm thông tin về kiểu tập tin và thông tin về tính tương thích. Hộp dữ liệu truyền thông có thể chứa các mẫu (ví dụ, các hình ảnh mã hóa). Hộp phim (“moov”) chứa siêu dữ liệu cho các dòng truyền thông liên tục có trong tập tin. Mỗi dòng truyền thông liên tục này có thể được biểu diễn trong tập tin dưới dạng rãnh. Chẳng hạn, hộp phim có thể chứa siêu dữ liệu về phim (ví dụ, mối liên hệ logic và định thời giữa các mẫu và cả các con trỏ đến vị trí của các mẫu). Hộp phim có thể bao gồm một số kiểu hộp con. Các hộp con trong một hộp phim có thể bao gồm một hoặc nhiều hộp rãnh. Hộp rãnh có thể bao gồm thông

tin về một rãnh riêng lẻ của phim. Hộp rãnh có thể bao gồm tiêu đề của rãnh mô tả thông tin chung của một rãnh đơn lẻ. Ngoài ra, hộp rãnh có thể bao gồm hộp truyền thông chứa hộp thông tin truyền thông. Hộp thông tin truyền thông có thể bao gồm hộp bảng mẫu chứa dữ liệu chỉ ra các mẫu truyền thông trong rãnh. Thông tin trong hộp bảng mẫu có thể được sử dụng để định vị mẫu theo thời gian và cho mỗi mẫu của rãnh, kiểu, kích thước và bộ chứa và độ dịch vào bộ chứa đó của mẫu. Do vậy, siêu dữ liệu cho rãnh có thể được chứa trong hộp rãnh (“trak”), còn nội dung truyền thông của rãnh này được chứa trong hộp dữ liệu truyền thông (Media Data box-mdat) hoặc trực tiếp trong một tập tin riêng. Nội dung truyền thông đối với các rãnh bao gồm (ví dụ, bao gồm) chuỗi các mẫu, như đơn vị truy cập âm thanh hoặc video.

ISOBMFF chỉ rõ các loại rãnh sau đây: rãnh phương tiện chứa dòng phương tiện sơ cấp, rãnh gợi ý bao gồm các lệnh truyền phương tiện hoặc biểu diễn dòng gói thu được, và rãnh siêu dữ liệu được định thời bao gồm siêu dữ liệu được đồng bộ theo thời gian. Siêu dữ liệu cho mỗi rãnh bao gồm danh sách các mục nhập mô tả mẫu, mỗi mục nhập cung cấp định dạng mã hóa hoặc đóng gói dùng trong rãnh và dữ liệu khởi tạo cần thiết để xử lý định dạng đó. Mỗi mẫu được kết hợp với một trong số các mục nhập mô tả mẫu của rãnh.

ISOBMFF có thể mô tả siêu dữ liệu đặc trưng cho mẫu bằng các cơ chế khác nhau. Các hộp đặc trưng trong hộp bảng mẫu (“stbl”) đã được chuẩn hóa để đáp ứng với các nhu cầu chung. Ví dụ, hộp mẫu đồng bộ hóa (“Sync Sample box-stss”) là một hộp nằm trong hộp bảng mẫu. Hộp mẫu đồng bộ hóa được sử dụng để lập danh sách các mẫu truy cập ngẫu nhiên của rãnh. Bản mô tả này có thể đề cập đến mẫu được lập danh sách bởi hộp mẫu đồng bộ hóa làm mẫu đồng bộ hóa. Theo ví dụ khác, cơ chế tạo nhóm mẫu có thể lập sơ đồ mẫu theo kiểu tạo nhóm 4 đặc điểm thành các nhóm mẫu có chung đặc tính được mô tả là mục nhập mô tả nhóm mẫu trong tập tin. Một số kiểu tạo nhóm đã được mô tả trong ISOBMFF.

Hộp bảng mẫu bao gồm một hoặc nhiều hộp SampleToGroup và một hoặc nhiều hộp mô tả nhóm mẫu (tức là, hộp SampleGroupDescription). Hộp SampleToGroup có thể được sử dụng để xác định nhóm mẫu mà mẫu này thuộc về, cùng với mô tả liên quan về nhóm mẫu này. Nói cách khác, hộp SampleToGroup có thể chỉ ra nhóm mà mẫu này thuộc về. Hộp SampleToGroup có thể là kiểu hộp “sbgp”. Hộp SampleToGroup có thể bao gồm phần tử kiểu tạo nhóm (ví dụ,

grouping_type). Phần tử kiểu tạo nhóm có thể là một số nguyên nhận dạng kiểu (tức là, tiêu chí được sử dụng để tạo nhóm mẫu) của việc tạo nhóm mẫu. Hơn nữa, hộp SampleToGroup có thể bao gồm một hoặc nhiều mục nhập. Mỗi mục nhập trong hộp SampleToGroup có thể liên quan đến một loạt các mẫu liên tiếp khác nhau, không chồng lấp trong rãnh. Mỗi mục nhập có thể chỉ ra phần tử đếm mẫu (ví dụ, sample_count) và phần tử chỉ số mô tả nhóm (ví dụ, group_description_index). Phần tử đếm mẫu của một mục nhập có thể chỉ ra số lượng mẫu kết hợp với mục nhập này. Nói cách khác, phần tử đếm mẫu của mục nhập có thể là số nguyên xác định số mẫu liên tiếp có cùng bộ mô tả nhóm mẫu. Phần tử chỉ số mô tả nhóm có thể nhận dạng hộp SampleGroupDescription chứa phần mô tả về các mẫu kết hợp với mục nhập. Phần tử chỉ số mô tả nhóm của nhiều mục nhập có thể nhận dạng cùng một hộp SampleGroupDescription.

Thiết kế định dạng tập tin hiện nay có thể có một hoặc nhiều vấn đề. Để lưu trữ nội dung video của một bộ mã hóa-giải mã video cụ thể dựa trên ISO/BMFF, việc mô tả về định dạng của tập tin cho bộ mã hóa-giải mã video đó có thể là cần thiết. Để lưu trữ dòng video chứa nhiều lớp như MV-HEVC và SHVC, có thể sử dụng lại một số khái niệm của định dạng tập tin SVC và MVC. Tuy nhiên, nhiều phần không thể sử dụng trực tiếp cho các dòng video SHVC và MV-HEVC. Việc áp dụng trực tiếp định dạng tập tin HEVC có ít nhất là các nhược điểm sau: Dòng bit SHVC và MV-HEVC có thể bắt đầu bằng đơn vị truy cập chứa hình ảnh IRAP ở lớp cơ sở, nhưng cũng có thể chứa các hình ảnh không IRAP khác ở các lớp khác hoặc ngược lại. Mẫu đồng bộ hóa hiện nay không cho phép chỉ ra điểm như vậy để truy cập ngẫu nhiên.

Bản mô tả này mô tả các giải pháp có thể thực hiện đối với các vấn đề trên, cũng như đưa ra các cải thiện tiềm năng, có thể lưu trữ hiệu quả và linh hoạt dòng video chứa nhiều lớp. Các kỹ thuật được mô tả trong bản mô tả này có thể áp dụng cho định dạng tập tin bất kỳ để lưu trữ nội dung video được mã hóa bởi bộ mã hóa video bất kỳ, mặc dù phần mô tả là dành riêng cho việc lưu trữ các dòng video SHVC và MV-HEVC dựa trên định dạng tập tin HEVC, được mô tả trong mệnh đề 8 của ISO/IEC 14496-15.

Phương án thực hiện làm ví dụ về một số kỹ thuật của bản mô tả được mô tả dưới đây. Phương án thực hiện làm ví dụ được mô tả dưới đây là dựa trên phần mô tả tích hợp mới nhất 14496-15 trong tài liệu xuất MPEG W13478. Các thay đổi của Phụ

lục A (được thể hiện ở phần văn bản gạch chân) và các phần bổ sung (Phần 9 cho SHVC và Phần 10 cho MV-HEVC) được bao gồm dưới đây. Nói cách khác, các ví dụ cụ thể của sáng chế có thể thay đổi Phụ lục A của ISO/IEC 14496-15 và có thể bổ sung các phần 9 và/hoặc 10 vào ISO/IEC 14496-15. Phần chữ được gạch chân và gạch chân hai dòng có thể đặc biệt thích hợp cho các ví dụ của sáng chế. Mặc dù thuật ngữ SHVC được sử dụng ở các phần khác nhau trong các ví dụ được mô tả trong bản mô tả này, thiết kế của bản mô tả này thực sự không chỉ để hỗ trợ bộ mã hóa-giải mã SHVC, mà thay vào đó là tất cả các bộ mã hóa-giải mã nhiều lớp bao gồm MV-HEVC, 3D-HEVC đều có thể được hỗ trợ, trừ khi có chỉ dẫn rõ ràng khác.

Thông số kỹ thuật ISO/BMFF xác định sáu loại điểm truy cập dòng (Stream Access Point - SAP) để sử dụng với DASH. Hai loại SAP đầu tiên (loại 1 và loại 2), tương ứng với các hình ảnh IDR theo chuẩn H.264/AVC và HEVC. Loại SAP thứ ba (loại 3) tương ứng với các điểm truy cập ngẫu nhiên GOP mở tức là các hình ảnh BLA hoặc CRA trong HEVC. Loại SAP thứ tư (loại 4) tương ứng với các điểm truy cập ngẫu nhiên GDR.

Trong định dạng tập tin L-HEVC hiện thời, một số thông tin bậc cao (ví dụ, thông tin về các lớp trong dòng bit, tốc độ bit, tốc độ khung, các lớp thời gian con, tính song song, các điểm hoạt động, v.v.) được báo hiệu trong LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDecoderConfigurationRecord, thông tin nội dung rãnh ('tcon') và OperationPointsInformationBox ('oinf'). Theo một ví dụ, thiết kế cú pháp của các hộp nêu trên là như sau:

Dựa trên cấu trúc hiện tại của các hộp này và thông tin được chứa trong đó, nhằm phát lại nội dung trong tập tin, máy phát có thể được cấu tạo để trước tiên là tìm hộp 'oinf' (chỉ có một hộp này trong tập tin) để biết có các điểm hoạt động nào, và sau đó chọn một trong số các điểm hoạt động cần phát. Bộ phát video sau đó có thể kiểm tra các hộp 'tcon' (một ô trong mỗi rãnh chứa video L-HEVC) để biết các rãnh nào chứa các lớp của các điểm hoạt động được chọn.

//Mục nhập mẫu LHVC và HEVCLHVC

```
class LHEVCCConfigurationBox extends Box('lhvc') {
    LHVCDecoderConfigurationRecord() LHEVCCConfig;
}
```

```

class HEVCLHVCSampleEntry() extends HEVCSampleEntry() {
    LHEVCConfigurationBox    lhvcconfig;
    MPEG4BitRateBox ();          // tùy chọn
    MPEG4ExtensionDescriptorsBox (); // tùy chọn
    extra_boxes                boxes;    // tùy chọn
}

// Sử dụng phần này nếu rãnh không tương thích với chuẩn HEVC
class LHEVCSampleEntry() extends VisualSampleEntry ('lhv1', or 'lhe1') {
    LHVCConfigurationBox    lhvcconfig;

    MPEG4BitRateBox ();          // tùy chọn
    MPEG4ExtensionDescriptorsBox (); // tùy chọn
    Box extra_boxes[];
}

aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;
    bit(3) reserved = '111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    bit(6) reserved = '111111'b;
    unsigned int(2) chromaFormat;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthLumaMinus8;

```

```

bit(5) reserved = '11111'b;
unsigned int(3) bitDepthChromaMinus8;
bit(16) avgFrameRate;
bit(2) constantFrameRate;
bit(3) numTemporalLayers;
bit(1) temporalIdNested;
unsigned int(2) lengthSizeMinusOne;
unsigned int(8) numOfArrays;
for (j=0; j < numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) reserved = 0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i< numNalus; i++) {
        unsigned int(16) nalUnitLength;
        bit(8*nalUnitLength) nalUnit;
    }
}
unsigned int(16) operationPointIdx;
}

class TrackContentsInfoBox extends FullBox('tcon', version = 0, 0)){
    unsigned int (2) reserved
    unsigned int (6) num_layers_in_track
    for (i=0; i<num_layers_in_track; i++){
        unsigned int (4) reserved
        unsigned int (6) layer_id
        unsigned int (3) min_sub_layer_id
        unsigned int (3) max_sub_layer_id
    }
}

```

```

class OperationPointsInformation extends FullBox('oinf', version = 0, 0){
    unsigned int(16) scalability_mask
    unsigned int(2) reserved
    unsigned int(6) num_profile_tier_level
    for (i=1; i<=num_profile_tier_level; i++) {
        unsigned int(2) general_profile_space;
        unsigned int(1) general_tier_flag;
        unsigned int(5) general_profile_idc;
        unsigned int(32) general_profile_compatibility_flags;
        unsigned int(48) general_constraint_indicator_flags;
        unsigned int(8) general_level_idc;
    }
    unsigned int(16) num_operation_points
    for (i=0; i<num_operation_points) {
        unsigned int(16) operation_point_id
        unsigned int(8) max_temporal_id;
        unsigned int(8) layer_count
        for (i=0; i<layer_count; i++) {
            unsigned int(8) ptl_idx
                unsigned int(6) layer_id;
                unsigned int(1) is_outputlayer;
                unsigned int(1) is_alterate_outputlayer;
            }
        }
    unsigned int(8) max_layer_count
    for (i=0; i<max_layer_count; i++) {
        unsigned int(8) dependent_layerID
        unsigned int(8) num_layers_dependent_on
        for (j=0; j<num_layers_dependent_on; j++) {
            unsigned int(8) dependent_on_layerID
        }
        for (j = 0; j < 16; j++) {

```

```

        if (scalability mask & (1 << j))
            unsigned int(8) dimension_identifier
    }
}
}

```

Dựa trên cấu trúc hiện tại của các hộp này và thông tin được chứa trong đó, nhằm phát lại nội dung trong tập tin, máy phát có thể được cấu tạo để trước tiên là tìm hộp 'oinf' (chỉ có một hộp này trong tập tin) để biết có các điểm hoạt động nào, và sau đó chọn một trong số các điểm hoạt động cần phát. Bộ phát video sau đó có thể kiểm tra các hộp 'tcon' (một ô trong mỗi rãnh chứa video L-HEVC) để biết các rãnh nào chứa các lớp của các điểm hoạt động được chọn.

Xét đến công dụng cơ bản của thiết kế hiện tại, sáng chế đề xuất nhiều thông tin hơn, như định dạng biểu diễn (bao gồm độ phân giải không gian, độ sâu bit, và định dạng màu sắc), tốc độ bit, và tốc độ khung, được bao gồm trong hộp 'oinf' để cho phép lựa chọn các điểm hoạt động. Mục nhập mẫu trong mỗi rãnh không có một tập hợp gồm các thông tin như thế, mà chỉ dành cho một điểm hoạt động cụ thể. Khi nhiều điểm hoạt động được chứa trong một rãnh, thông tin cho các điểm hoạt động khác bị mất.

Một vấn đề khác liên quan đến thực tế là ngữ nghĩa của nhiều trường trong LHEVCDecoderConfigurationRecord không rõ ràng, và một số trong đó rất lộn xộn. Ví dụ, biên dạng, tầng và độ (profile, tier and level - PTL), chromaFormat, bitDepthLumaMinus8, và bitDepthChromaMinus8 là các đặc điểm riêng cho từng lớp, nhưng hiện nay đang được cho là áp dụng vào điểm hoạt động được chỉ báo bởi operationPointIdx. Khi điểm hoạt động chứa nhiều hơn một lớp, thì ngữ nghĩa thường là không rõ ràng.

Thực tế, dựa trên các bước sử dụng cơ bản thông thường của thiết kế này, một số thông tin trong mục nhập mẫu không thực sự hữu dụng, đặc biệt là khi có đủ thông tin trong hộp 'oinf' để chọn điểm hoạt động.

Một vấn đề khác nữa là, trong SHVC và MV-HEVC, PTL chỉ được báo hiệu cho mỗi lớp thiết yếu (tức là, lớp mà là lớp xuất hoặc lớp được tham chiếu đến trực tiếp hoặc gián tiếp bởi lớp xuất trong điểm hoạt động hoặc cả hai), và không dành cho

lớp không thiết yếu bất kỳ (lớp mà không phải là lớp thiết yếu). Do đó trong thiết kế định dạng tập tin, có thể không cần báo hiệu PTL cho các lớp không thiết yếu.

Tóm tắt các phương pháp và kỹ thuật được mô tả theo sáng chế được liệt kê dưới đây. Các phương án thực hiện cụ thể làm ví dụ được trình bày ở các phần sau đó. Các phương pháp và các kỹ thuật của sáng chế có thể được áp dụng độc lập hoặc kết hợp.

Kỹ thuật thứ nhất theo sáng chế bao gồm việc loại bỏ báo hiệu MPEG4BitRateBox() sau LHEVCConfigurationBox trong mục nhập mẫu LHEVC và mục nhập mẫu HEVCLHVC. Thay vào đó, kích hoạt báo hiệu thông tin tốc độ bit cho mỗi điểm hoạt động trong hộp 'oinf'.

Kỹ thuật thứ hai của sáng chế bao gồm việc báo hiệu thông tin trên định dạng biểu diễn (bao gồm độ phân giải không gian, độ sâu bit, và định dạng màu sắc) cho mỗi điểm hoạt động trong hộp 'oinf'.

Kỹ thuật thứ ba theo sáng chế bao gồm việc loại bỏ thông tin PTL, thông tin định dạng biểu diễn, và thông tin về tốc độ khung ra khỏi LHEVCDecoderConfigurationRecord, chúng có sẵn trong hộp 'oinf' hoặc được đề xuất thêm vào hộp 'oinf'. Thông tin còn lại trong LHEVCDecoderConfigurationRecord áp dụng cho tất cả các lớp có trong rãnh. Theo ví dụ khác về kỹ thuật thứ ba này, thiết kế của LHEVCDecoderConfigurationRecord được làm lại sao cho thông tin định dạng biểu diễn và thông tin về tốc độ khung, và có thể là các tham số/ thông tin bổ sung (ví dụ, thông tin về tính song song), được báo hiệu cho mỗi lớp. Phần tử cú pháp không được đánh dấu int(2) parallelismType trong LHEVCDecoderConfigurationRecord có thể chỉ báo loại nào của (các) đặc điểm giải mã song song có thể được dùng để giải mã hình ảnh trong lớp. Ô, mặt đầu sóng, và lát là các ví dụ về cơ chế phân đoạn hình ảnh có thể được dùng để hỗ trợ cho quá trình xử lý song song.

Kỹ thuật thứ tư theo sáng chế bao gồm việc loại bỏ operationPointIdx ra khỏi LHEVCDecoderConfigurationRecord. Theo ví dụ khác về kỹ thuật thứ tư này, việc báo hiệu danh sách các chỉ số điểm hoạt động mà liên quan đến rãnh trong LHEVCDecoderConfigurationRecord được cho phép.

Kỹ thuật thứ năm theo sáng chế bao gồm việc thay đổi ngữ nghĩa của trường layer_count trong hộp 'oinf' để chỉ đếm các lớp cần thiết của điểm hoạt động.

Các phương án thực hiện làm ví dụ của phương pháp và kỹ thuật theo sáng chế được mô tả dưới đây. Các ví dụ dưới đây thể hiện các thay đổi về ký tự so với định dạng tập tin HEVC và LHEVC. Ký tự được thêm vào được thể hiện ở giữa các ký hiệu nhận dạng [START INSERTION] và [END INSERTION]. Ký tự bị xóa đi vào được thể hiện ở giữa các ký hiệu nhận dạng [START DELETION] và [END DELETION].

Phương án thực hiện thứ nhất được mô tả dưới đây.

Phần này mô tả các cải biến cụ thể đối với việc báo hiệu LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDecoderConfigurationRecord và OperationPointsInformationBox ('oinf') cho các kỹ thuật theo sáng chế 1, 2, 3 (không bao gồm ví dụ a.), 4 (không bao gồm ví dụ a.) và 5.

```
class LHEVCConfigurationBox extends Box('lhvC') {
    LHVCDecoderConfigurationRecord() LHEVCConfig;
}
class HEVCLHVCSampleEntry() extends HEVCSampleEntry() {
    LHEVCConfigurationBox    lhvconfig;
    [START DELETION] MPEG4BitRateBox ();
// [END DELETION] tùy chọn
    MPEG4ExtensionDescriptorsBox (); // tùy chọn
    extra_boxes                boxes;    // tùy chọn
}
// Sử dụng phần này nếu rãnh không tương thích với chuẩn HEVC
class LHEVCSampleEntry() extends VisualSampleEntry ('lhv1', or 'lhe1') {
    LHVConfigurationBox    lhvconfig;
    [START DELETION]MPEG4BitRateBox ();
// [END DELETION] tùy chọn
    MPEG4ExtensionDescriptorsBox (); // tùy chọn
    Box extra_boxes[];
}
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    [START DELETION] unsigned int(2) general_profile_space;
```

```

unsigned int(1) general_tier_flag;
unsigned int(5) general_profile_idc;
unsigned int(32) general_profile_compatibility_flags;
unsigned int(48) general_constraint_indicator_flags;
unsigned int(8) general_level_idc; [END DELETION]
bit(1) complete_representation;
bit(3) reserved = '111'b;
unsigned int(12) min_spatial_segmentation_idc;
bit(6) reserved = '111111'b;
unsigned int(2) parallelismType;
[START DELETION] bit(6) reserved = '111111'b;
unsigned int(2) chromaFormat;
bit(5) reserved = '11111'b;
unsigned int(3) bitDepthLumaMinus8;
bit(5) reserved = '11111'b;
unsigned int(3) bitDepthChromaMinus8;
bit(16) avgFrameRate;
bit(2) constantFrameRate; [END DELETION]
[START INSERTION] bit(2) reserved = '11'b; [END INSERTION]
bit(3) numTemporalLayers;
bit(1) temporalIdNested;
unsigned int(2) lengthSizeMinusOne;
unsigned int(8) numOfArrays;
for (j=0; j < numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) reserved = 0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i< numNalus; i++) {
        unsigned int(16) nalUnitLength;
        bit(8*nalUnitLength) nalUnit;
    }
}

```

```

    }
    [START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
}

class OperationPointsInformation extends FullBox('oinf', version = 0, 0){
    unsigned int(16) scalability_mask
    unsigned int(2) reserved
    unsigned int(6) num_profile_tier_level
    for (i=1; i<=num_profile_tier_level; i++) {
        unsigned int(2) general_profile_space;
        unsigned int(1) general_tier_flag;
        unsigned int(5) general_profile_idc;
        unsigned int(32) general_profile_compatibility_flags;
        unsigned int(48) general_constraint_indicator_flags;
        unsigned int(8) general_level_idc;
    }
    unsigned int(16) num_operation_points
    for (i=0; i<num_operation_points) {
        unsigned int(16) operation_point_id
        unsigned int(8) max_temporal_id;
        unsigned int(8) layer_count;
        for (i=0; i<layer_count; i++) {
            unsigned int(8) ptl_idx
            unsigned int(6) layer_id;
            unsigned int(1) is_outputlayer;
            unsigned int(1) is_alternate_outputlayer;
        }

        [START INSERTION]
        unsigned int(16) minPicWidth;
        unsigned int(16) minPicHeight;
        unsigned int(16) maxPicWidth;
        unsigned int(16) maxPicHeight;
        unsigned int(2) maxChromaFormat;
    }
}

```

```

    unsigned int(3) maxBitDepthMinus8;
    unsigned int(1) reserved
    unsigned int(1) frame_rate_info_flag
    unsigned int(1) bit_rate_info_flag
    if (frame_rate_info_flag) {
        bit(16) avgFrameRate;
        unsigned int(6) reserved
        bit(2) constantFrameRate;
    }
    if (bit_rate_info_flag) {
        unsigned int(32) maxBitRate;
        unsigned int(32) avgBitRate;
    }[END INSERTION]
}

unsigned int(8) max_layer_count
for (i=0; i<max_layer_count; i++) {
    unsigned int(8) dependent_layerID
    unsigned int(8) num_layers_dependent_on
    for (j=0; j< num_layers_dependent_on; j++) {
        unsigned int(8) dependent_on_layerID
    }
    for (j = 0; j < 16; j++) {
        if (scalability_mask & (1 << j))
            unsigned int(8) dimension_identifier
    }
}
}

```

...

layer_count: Trường này chỉ báo số lượng [START INSERTION] lớp [END INSERTION] cần thiết mà là một phần của [START INSERTION] điểm hoạt động [END INSERTION] [START DELETION] [END DELETION].

...

[START INSERTION]

minPicWidth chỉ rõ giá trị nhỏ nhất của chỉ số độ rộng khối sáng như được xác định bởi tham số pic_width_in_luma_samples theo ISO/IEC 23008-2 dành cho luồng của điểm hoạt động.

minPicHeight chỉ rõ giá trị nhỏ nhất của chỉ số độ cao khối sáng như được xác định bởi tham số pic_height_in_luma_samples theo ISO/IEC 23008-2 cho luồng của điểm hoạt động.

maxPicWidth chỉ rõ giá trị lớn nhất của chỉ số độ rộng khối sáng như được xác định bởi tham số pic_width_in_luma_samples theo ISO/IEC 23008-2 dành cho luồng của điểm hoạt động.

maxPicHeight chỉ rõ giá trị lớn nhất của chỉ số độ cao khối sáng như được xác định bởi tham số pic_height_in_luma_samples theo ISO/IEC 23008-2 cho luồng của điểm hoạt động.

maxChromaFormat chỉ rõ giá trị lớn nhất của chỉ số chroma_format như được xác định bởi tham số chroma_format_idc theo ISO/IEC 23008-2 cho luồng của điểm hoạt động.

maxBitDepthMinus8 chỉ rõ giá trị lớn nhất của các chỉ số độ sâu bit độ sáng và độ màu như được xác định lần lượt bởi các tham số bit_depth_luma_minus8 và bit_depth_chroma_minus8 theo ISO/IEC 23008-2 cho luồng của điểm hoạt động.

frame_rate_info_flagequal bằng 0 chỉ báo rằng không có thông tin về tốc độ khung dành cho điểm hoạt động. Giá trị 1 chỉ báo rằng có thông tin về tốc độ khung dành cho điểm hoạt động.

bit_rate_info_flagequal bằng 0 chỉ báo rằng không có thông tin tốc độ bit dành cho điểm hoạt động. Giá trị 1 chỉ báo rằng có thông tin về tốc độ bit dành cho điểm hoạt động.

avgFrameRate cho biết tốc độ khung trung bình trong các đơn vị khung /(256 giây) dành cho điểm hoạt động. Giá trị 0 chỉ báo tốc độ khung trung bình không xác định.

constantFrameRate bằng 1 chỉ báo rằng luồng của điểm hoạt động có tốc độ khung không đổi. Giá trị 2 chỉ báo rằng dạng biểu diễn của mỗi lớp thời gian trong luồng của điểm hoạt động có tốc độ khung không đổi. Giá trị 0 chỉ báo rằng luồng của điểm hoạt động có thể có hoặc không có tốc độ khung không đổi.

maxBitRate cho biết tốc độ bit lớn nhất tính theo đơn vị bit/giây của luồng của điểm hoạt động, qua cửa sổ bất kỳ của một giây.

avgBitRate cho biết tốc độ bit trung bình tính theo đơn vị bit/giây của luồng của điểm hoạt động.

...

[END INSERTION]

Phương án thực hiện thứ hai được mô tả dưới đây.

Phần này mô tả các cải biến cụ thể đối với việc báo hiệu LHVCDecoderConfigurationRecord cho ví dụ 3(a).

```
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    [START DELETION] unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;
    bit(3) reserved = '111'b; [END DELETION]
    [START INSERTION] bit(2) reserved = '11'b; [END INSERTION]
    [START INSERTION] unsigned int(6) num_layers; [END INSERTION]
    for (j=0; j < num_layers; j++) {
        [START INSERTION] unsigned int(8) layer_id; [END INSERTION]
        unsigned int(12) min_spatial_segmentation_idc;
        bit(6) reserved = '111111'b;
        unsigned int(2) parallelismType;
        bit(6) reserved = '111111'b;
        unsigned int(2) chromaFormat;
        [START INSERTION] bit(6) reserved = '111111'b; [END INSERTION]
        [START DELETION] bit(5) reserved = '11111'b; [END DELETION]
        unsigned int(3) bitDepthLumaMinus8;
        bit(5) reserved = '11111'b;
```

```

    unsigned int(3) bitDepthChromaMinus8;
    [START INSERTION] bit(5) reserved = '11111'b; [END INSERTION]
[START DELETION] bit(16) avgFrameRate;
bit(2) constantFrameRate; [END DELETION]
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    [START INSERTION] bit(4) reserved = '1111'b; [END INSERTION]
}
[START INSERTION] bit(1) complete_representation; [END INSERTION]
unsigned int(2) lengthSizeMinusOne;
[START INSERTION] bit(5) reserved = '11111'b; [END INSERTION]
unsigned int(8) numOfArrays;
for (j=0; j < numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) reserved = 0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i < numNalus; i++) {
        unsigned int(16) nalUnitLength;
        bit(8*nalUnitLength) nalUnit;
    }
}
[START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
}
[START INSERTION]
num_layers chỉ rõ số lượng lớp trong rãnh
layer_id chỉ rõ giá trị ID lớp mà thông tin trong vòng lặp được cung cấp đến đó.
[END INSERTION]

```

Phương án thực hiện thứ ba được mô tả dưới đây.

Phần này mô tả các cải biến cụ thể đối với việc báo hiệu
LHVCDecoderConfigurationRecord cho ví dụ 4(a).

```

aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;    bit(3) reserved = '111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    bit(6) reserved = '111111'b;
    unsigned int(2) chromaFormat;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthLumaMinus8;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthChromaMinus8;
    bit(16) avgFrameRate;
    bit(2) constantFrameRate;
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j < numOfArrays; j++) {
        bit(1) array_completeness;
        unsigned int(1) reserved = 0;
        unsigned int(6) NAL_unit_type;
        unsigned int(16) numNalus;
        for (i=0; i < numNalus; i++) {
            unsigned int(16) nalUnitLength;
            bit(8*nalUnitLength) nalUnit;
        }
    }
}

```

```

    }
}
[START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
[START INSERTION]
    unsigned int(16) numOfOperationPoints;
    for (j=0; j < numOfOperationPoints; j++) {
        unsigned int(16) operationPointIdx;
    } [END INSERTION]
}

```

[START INSERTION] numOperationPoints: Trường này báo hiệu số lượng của các điểm hoạt động có sẵn dành cho rãnh. [END INSERTION]

operationPointIdx: Trường này báo hiệu chỉ số của điểm hoạt động được ghi trong hộp thông tin về điểm hoạt động. [START DELETION] Các giá trị general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flags, general_constraint_indicator_flag và general_level_idc trong LHEVCDecoderConfigurationRecord sẽ giống với các giá trị tương ứng của điểm hoạt động thứ operationPointIdx trong hộp thông tin về điểm hoạt động. [END DELETION] [START INSERTION] Giá trị max_temporal_id trong điểm hoạt động thứ operationPointIdx trong hộp thông tin về điểm hoạt động sẽ nhỏ hơn hoặc bằng giá trị numTemporalLayers. [END INSERTION]

Lưu ý: Rãnh có thể liên quan đến một hoặc [START DELETION] biểu diễn [END DELETION] nhiều hơn một bộ lớp xuất [START DELETION] và do đó là nhiều hơn một biên dạng [START DELETION]. Bộ phát có thể tìm ra các lớp nào cần giải mã và các lớp nào cần xuất tương ứng với thông tin biên dạng trong LHEVCDecoderConfigurationRecord[START INSERTION] dành cho điểm hoạt động được chọn với chỉ số operationPointIdx [END INSERTION] bằng cách nghiên cứu thông tin được cung cấp cho điểm hoạt động thứ operationPointIdx trong hộp thông tin về điểm hoạt động.

Lưu ý: Đối với mỗi lớp hình ảnh phụ có trong rãnh, tốt hơn là nó bao gồm, trong nalUnit, đơn vị NAL SEI chứa thông báo thông báo SEI khai báo, như thông

báo SEI về thông tin biểu diễn độ sâu cho các lớp hình ảnh phụ có độ sâu, chỉ rõ các đặc tính của lớp hình ảnh phụ.

Fig.2 là sơ đồ khối minh họa bộ mã hóa video 20 làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này. Bộ mã hóa video 20 có thể được tạo cấu hình để xuất một khung hình, đa khung hình, dữ liệu video mở rộng được, dữ liệu video 3D và các loại dữ liệu video khác. Bộ mã hóa video 20 có thể được tạo cấu hình để xuất video cho thực thể sau xử lý 27. Thực thể sau xử lý 27 được dự định là một ví dụ về thực thể video, như MANE hoặc thiết bị ghép nối/soạn thảo, có thể xử lý dữ liệu video mã hóa từ bộ mã hóa video 20. Trong một số trường hợp, thực thể sau xử lý có thể là một ví dụ về thực thể mạng. Trong một số hệ thống mã hóa video, thực thể sau xử lý 27 và bộ mã hóa video 20 có thể là các bộ phận của các thiết bị riêng biệt, trong khi ở các trường hợp khác, chức năng được mô tả liên quan đến thực thể sau xử lý 27 có thể được thực hiện bởi cùng một thiết bị bao gồm bộ mã hóa 20. Thực thể sau xử lý 27 có thể là thiết bị video. Theo một số ví dụ, thực thể sau xử lý 27 có thể giống như thiết bị tạo tập tin 34 trên Fig.1.

Bộ mã hóa video 20 có thể thực hiện mã hóa nội hình ảnh và liên hình ảnh các khối video trong các lát video. Việc mã hóa nội hình ảnh dựa trên dự đoán không gian để giảm hoặc loại bỏ phần dư thừa không gian trong video nằm trong khung hoặc hình ảnh video xác định. Mã hóa liên hình ảnh dựa vào sự dự đoán theo thời gian để làm giảm hoặc loại bỏ sự dư thừa thời gian trong video trong các khung hoặc hình ảnh liên kế của chuỗi video. Chế độ nội hình ảnh (chế độ I) có thể chỉ chế độ nén dựa trên không gian bất kỳ. Chế độ liên hình ảnh, như chế độ dự đoán một chiều (chế độ P) hoặc dự đoán hai chiều (chế độ B), có thể chỉ chế độ nén dựa vào thời gian bất kỳ.

Trong ví dụ trên Fig.2, bộ mã hóa video 20 bao gồm đơn vị phân chia 37, đơn vị xử lý dự đoán 41, đơn vị lọc 63, bộ nhớ hình ảnh tham chiếu 64, bộ cộng 50, đơn vị xử lý biến đổi 52, đơn vị lượng tử hóa 54 và đơn vị mã hóa entropy 56. Đơn vị xử lý dự đoán 41 bao gồm đơn vị ước lượng chuyển động 42, đơn vị bù chuyển động 44 và đơn vị xử lý dự đoán nội hình ảnh 46. Đối với việc tái tạo khối video, bộ mã hóa video 20 còn bao gồm đơn vị lượng tử hóa ngược 58, đơn vị xử lý biến đổi ngược 60, và bộ cộng 62. Đơn vị lọc 63 được dự định là một hoặc nhiều bộ lọc vòng lặp như bộ lọc tách khối, bộ lọc vòng lặp tương thích (adaptive loop filter-ALF), và bộ lọc dịch chuyển tương thích mẫu (sample adaptive offset-SAO). Mặc dù bộ lọc 63 được thể

hiện trên Fig.2 là bộ lọc vòng lặp, theo cấu hình khác, đơn vị lọc 63 có thể được thực hiện như bộ lọc sau vòng lặp.

Bộ nhớ dữ liệu video 35 của bộ mã hóa video 20 có thể lưu trữ dữ liệu video cần mã hóa bởi các thành phần của bộ mã hóa video 20. Dữ liệu video được lưu trữ trong bộ nhớ dữ liệu video 35 có thể thu được, ví dụ, từ nguồn video 18. Bộ nhớ hình ảnh tham chiếu 64 có thể là bộ nhớ hình ảnh tham chiếu lưu trữ dữ liệu video tham chiếu để sử dụng trong việc mã hóa dữ liệu video của bộ mã hóa video 20, ví dụ, các chế độ mã hóa nội hình ảnh hoặc liên hình ảnh. Bộ nhớ dữ liệu video 35 và bộ nhớ hình ảnh tham chiếu 64 có thể được tạo ra bởi thiết bị bất kỳ trong số các thiết bị nhớ, như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM), bao gồm DRAM đồng bộ (synchronous DRAM - SDRAM), RAM từ trở (magnetoresistive RAM - MRAM), RAM điện trở (resistive RAM - RRAM), hoặc các loại thiết bị nhớ khác. Bộ nhớ dữ liệu video 35 và bộ nhớ hình ảnh tham chiếu 64 có thể được cung cấp bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Trong các ví dụ khác nhau, bộ nhớ dữ liệu video 35 có thể nằm trên chip cùng các bộ phận khác của bộ mã hóa video 20, hoặc ngoài chip so với các bộ phận này.

Như được thể hiện trên Fig.2, bộ mã hóa video 20 nhận dữ liệu video và đơn vị phân chia 37 phân chia dữ liệu thành các khối video. Bước phân chia này cũng có thể bao gồm bước phân chia thành các lát, ô, hoặc các đơn vị lớn hơn khác, cũng như phân chia khối video, ví dụ, theo cấu trúc cây tứ phân gồm các LCU và CU. Nói chung bộ mã hóa video 20 minh họa các thành phần mà mã hóa các khối video trong lát video cần được mã hóa. Lát có thể được chia thành nhiều khối video (và có thể được chia thành tập hợp các khối video được gọi là các ô). Đơn vị xử lý dự đoán 41 có thể chọn một trong số các chế độ mã hóa có thể có, như một trong số các chế độ mã hóa nội hình ảnh hoặc một trong số các chế độ mã hóa liên hình ảnh, cho khối video hiện thời dựa trên các kết quả lỗi (ví dụ, tốc độ mã hóa và độ méo). Đơn vị xử lý dự đoán 41 có thể cung cấp khối mã hóa nội hình ảnh hoặc liên hình ảnh thu được cho bộ cộng 50 để tạo ra khối dữ liệu dư và cho bộ cộng 62 để tái tạo khối hiện thời để dùng làm hình ảnh tham chiếu.

Đơn vị xử lý dự đoán nội hình ảnh 46 nằm trong đơn vị xử lý dự đoán 41 có thể thực hiện mã hóa dự đoán nội hình ảnh khối video hiện thời so với một hoặc nhiều khối lân cận trong cùng một khung hoặc lát dưới dạng khối hiện thời được mã hóa để

tạo ra sự nén không gian. Đơn vị ước lượng chuyển động 42 và đơn vị bù chuyển động 44 nằm trong đơn vị xử lý dự đoán 41 thực hiện mã hóa dự đoán liên hình ảnh khối video hiện thời so với một hoặc nhiều khối dự đoán trong một hoặc nhiều hình ảnh tham chiếu để tạo ra sự nén về thời gian.

Đơn vị ước lượng chuyển động 42 có thể được tạo cấu hình để xác định chế độ dự đoán liên hình ảnh cho lát video theo kiểu định trước cho chuỗi video. Kiểu định trước có thể biểu thị các lát video theo trình tự là lát P, lát B hoặc lát GPB. Đơn vị ước lượng chuyển động 42 và đơn vị bù chuyển động 44 có thể được tích hợp cao, nhưng được minh họa riêng cho các mục đích khái niệm. Việc ước lượng chuyển động, được thực hiện bởi đơn vị ước lượng chuyển động 42, là quá trình tạo ra các vector chuyển động, các vector này ước lượng chuyển động cho các khối video. Vector chuyển động, ví dụ, có thể biểu thị sự dịch chuyển của PU của khối video trong khung hoặc hình video hiện thời so với khối dự đoán trong hình ảnh tham chiếu.

Khối dự đoán là khối được thấy là khớp gần nhất với PU của khối video cần được mã hóa xét về mặt vi sai điểm ảnh, giá trị này có thể được xác định bằng tổng vi sai tuyệt đối (SAD - Sum of Absolute Difference), tổng vi sai bình phương (SSD - Sum of Square Difference), hoặc các số đo vi sai khác. Theo một số ví dụ, bộ mã hóa video 20 có thể tính các giá trị cho các vị trí điểm ảnh dưới số nguyên của các hình ảnh tham chiếu được lưu trữ trong bộ nhớ hình ảnh tham chiếu 64. Ví dụ, bộ mã hóa video 20 có thể nội suy các giá trị của các vị trí điểm ảnh một phần tư, các vị trí điểm ảnh một phần tám, hoặc các vị trí điểm ảnh phân số khác của hình ảnh tham chiếu. Do vậy, đơn vị ước lượng chuyển động 42 có thể thực hiện tìm kiếm chuyển động so với các vị trí điểm ảnh toàn phần và các vị trí điểm ảnh phân số và xuất ra vector chuyển động với độ chính xác điểm ảnh phân số.

Đơn vị ước lượng chuyển động 42 tính toán vector chuyển động cho PU của khối video trong lát được mã hóa liên hình ảnh bằng cách so sánh vị trí của PU với vị trí của khối dự đoán hình ảnh tham chiếu. Hình ảnh tham chiếu có thể được chọn từ danh sách hình ảnh tham chiếu thứ nhất (Danh sách 0) hoặc danh sách hình ảnh tham chiếu thứ hai (Danh sách 1), mỗi danh sách này nhận dạng một hoặc nhiều hình ảnh tham chiếu được lưu trữ trong bộ nhớ hình ảnh tham chiếu 64. Đơn vị ước lượng chuyển động 42 gửi vector chuyển động đã được tính toán đến đơn vị mã hóa entropy 56 và đơn vị bù chuyển động 44.

Quy trình bù chuyển động, được thực hiện bởi đơn vị bù chuyển động 44, có thể bao gồm tìm nạp hoặc tạo ra khối dự đoán dựa vào vector chuyển động được xác định bởi đơn vị ước lượng chuyển động 42, có thể thực hiện các kỹ thuật nội suy theo độ chính xác dưới điểm ảnh. Khi nhận vector chuyển động cho PU của khối video hiện thời, đơn vị bù chuyển động 44 có thể định vị khối dự đoán mà vector chuyển động trỏ vào trong một trong số các danh sách hình ảnh tham chiếu. Bộ mã hóa video 20 có thể tạo ra khối video dư bằng cách lấy các giá trị điểm ảnh của khối video hiện thời đang được mã hóa trừ đi các giá trị điểm ảnh của khối dự đoán, tạo ra các giá trị vi sai điểm ảnh. Các giá trị vi sai điểm ảnh tạo ra dữ liệu dư cho khối, và có thể bao gồm cả thành phần vi sai độ sáng và độ màu. Bộ cộng 50 là thành phần hoặc các thành phần thực hiện phép trừ này. Đơn vị bù chuyển động 44 cũng có thể tạo ra các phần tử cú pháp gắn với các khối video và lát video để bộ giải mã video 30 dùng trong việc giải mã các khối video của lát video.

Đơn vị xử lý dự đoán nội hình ảnh 46 có thể dự đoán liên hình ảnh khối hiện thời, thay cho dự đoán liên hình ảnh được thực hiện bởi đơn vị ước lượng chuyển động 42 và đơn vị bù chuyển động 44, như nêu trên. Cụ thể là, đơn vị xử lý dự đoán nội hình ảnh 46 có thể xác định chế độ dự đoán nội hình ảnh nhằm sử dụng để mã hóa khối hiện thời. Theo một số ví dụ, đơn vị xử lý dự đoán nội hình ảnh 46 có thể mã hóa khối hiện thời bằng cách sử dụng các chế độ dự đoán nội hình ảnh khác nhau, ví dụ, trong các quy trình mã hóa riêng biệt và đơn vị xử lý dự đoán nội hình ảnh 46 (hoặc đơn vị chọn chế độ 40, theo một số ví dụ) có thể chọn chế độ dự đoán nội hình ảnh thích hợp để sử dụng từ các chế độ đã thử nghiệm. Ví dụ, đơn vị xử lý dự đoán nội hình ảnh 46 có thể tính toán giá trị tốc độ-độ méo bằng cách sử dụng phân tích tốc độ-độ méo đối với các chế độ dự đoán nội hình ảnh khác nhau và chọn chế độ dự đoán nội hình ảnh có đặc điểm tốc độ-độ méo tốt nhất trong số các chế độ đã thử nghiệm. Phân tích tốc độ-độ méo thường xác định mức độ méo (hoặc lỗi) giữa khối mã hóa và khối gốc chưa mã hóa mà khối này được mã hóa để tạo ra khối mã hóa, cũng như tốc độ bit (tức là, số lượng bit) được sử dụng để tạo ra khối mã hóa. Đơn vị xử lý dự đoán nội hình ảnh 46 có thể tính toán tỷ lệ giữa mức độ méo và tốc độ cho các khối mã hóa khác nhau để xác định chế độ dự đoán nội hình ảnh nào có giá trị tốc độ-độ méo tốt nhất cho khối này.

Trong trường hợp bất kỳ, sau khi chọn chế độ dự đoán nội hình ảnh cho một khối, đơn vị xử lý dự đoán nội hình ảnh 46 có thể cung cấp thông tin chỉ báo chế độ dự đoán nội hình ảnh được chọn cho khối này đến đơn vị mã hóa entropy 56. Đơn vị mã hóa entropy 56 có thể mã hóa thông tin chỉ báo chế độ dự đoán nội hình ảnh được chọn theo kỹ thuật theo sáng chế. Bộ mã hóa video 20 có thể bao gồm trên dữ liệu cấu hình dòng bit được truyền, bao gồm nhiều bảng chỉ số chế độ dự đoán nội hình ảnh và nhiều bảng chỉ số chế độ dự đoán nội hình ảnh được hiệu chỉnh (còn được gọi là bảng ánh xạ từ mã), các định nghĩa về ngữ cảnh mã hóa cho các khối khác nhau và các chỉ báo về chế độ dự đoán nội hình ảnh có xác suất cao nhất, bảng chỉ số chế độ dự đoán nội hình ảnh và bảng chỉ số chế độ dự đoán nội hình ảnh đã hiệu chỉnh để sử dụng cho từng ngữ cảnh.

Sau khi đơn vị xử lý dự đoán 41 tạo ra khối dự đoán cho khối video hiện thời bằng cách dự đoán liên hình ảnh hoặc dự đoán nội hình ảnh, bộ mã hóa video 20 có thể tạo ra khối video dư bằng cách lấy khối video hiện thời trừ đi khối dự đoán. Dữ liệu video dư trong khối dư có thể được bao gồm trong một hoặc nhiều TU và được áp dụng cho đơn vị xử lý biến đổi 52. Đơn vị xử lý biến đổi 52 này biến đổi dữ liệu video dư thành các hệ số biến đổi dư bằng cách sử dụng kỹ thuật biến đổi như kỹ thuật biến đổi cosin rời rạc (DCT - discrete cosine transform) hoặc kỹ thuật biến đổi tương tự về mặt khái niệm. Đơn vị xử lý biến đổi 52 có thể biến đổi dữ liệu video dư từ miền điểm ảnh thành miền biến đổi, như miền tần số.

Đơn vị xử lý biến đổi 52 có thể gửi các hệ số biến đổi thu được đến đơn vị lượng tử hóa 54. Đơn vị lượng tử hóa 54 lượng tử hóa các hệ số biến đổi để làm giảm tiếp tốc độ bit. Quy trình lượng tử hóa có thể giảm chiều sâu bit gắn với một số hoặc tất cả các hệ số. Mức độ lượng tử hóa có thể được thay đổi bằng cách điều chỉnh tham số lượng tử hóa. Theo một số ví dụ, đơn vị lượng tử hóa 54 sau đó có thể thực hiện quét ma trận bao gồm các hệ số biến đổi được lượng tử hóa. Theo cách khác, đơn vị mã hóa entropy 56 có thể thực hiện bước quét này.

Sau khi lượng tử hóa, đơn vị mã hóa entropy 56 có thể mã hóa entropy các phần tử cú pháp biểu diễn các hệ số biến đổi được lượng tử hóa. Ví dụ, đơn vị mã hóa entropy 56 có thể thực hiện mã hóa CAVLC, CABAC, SBAC, PIPE hoặc phương pháp hoặc kỹ thuật mã hóa entropy khác. Sau khi mã hóa entropy bằng đơn vị mã hóa entropy 56, dòng bit được mã hóa có thể được truyền đến bộ giải mã video 30, hoặc

được lưu trữ để truyền sau hoặc khôi phục bởi bộ giải mã video 30. Đơn vị mã hóa entropy 56 cũng có thể mã hóa entropy các vector chuyển động và các phần tử cú pháp khác cho lát video hiện thời được mã hóa.

Đơn vị lượng tử hóa ngược 58 và đơn vị xử lý biến đổi ngược 60 lần lượt áp dụng kỹ thuật lượng tử hóa ngược và biến đổi ngược để tái tạo khối dư trong miền điểm ảnh để sử dụng sau đó làm khối tham chiếu của hình ảnh tham chiếu. Đơn vị bù chuyển động 44 có thể tính toán khối tham chiếu bằng cách cộng khối dư vào khối dự đoán của một trong số các hình ảnh tham chiếu trong một trong số các danh sách hình ảnh tham chiếu. Đơn vị bù chuyển động 44 cũng có thể áp dụng một hoặc nhiều bộ lọc nội suy vào khối dư được tái tạo để tính toán các giá trị điểm ảnh dưới số nguyên để sử dụng trong việc ước lượng chuyển động. Bộ cộng 62 cộng khối dư được tái tạo vào khối dự đoán được bù chuyển động được tạo ra bởi đơn vị bù chuyển động 44 để tạo ra khối tham chiếu để lưu trữ trong bộ nhớ hình ảnh tham chiếu 64. Khối tham chiếu có thể được sử dụng bởi đơn vị ước lượng chuyển động 42 và đơn vị bù chuyển động 44 làm khối tham chiếu để dự đoán nội hình ảnh khối trong khung hoặc hình ảnh video tiếp theo.

Bộ mã hóa video 20 là một ví dụ về bộ mã hóa video được tạo cấu hình để tạo ra dữ liệu video có thể được lưu trữ bằng cách sử dụng kỹ thuật định dạng tập tin được mô tả trong bản mô tả này.

Fig.3 là sơ đồ khối minh họa bộ giải mã video 30 làm ví dụ mà có thể thực hiện các kỹ thuật được mô tả trong bản mô tả này. Bộ giải mã video 30 có thể được tạo cấu hình để giải mã một khung hình, đa khung hình, dữ liệu video mở rộng được, dữ liệu video 3D, và các loại dữ liệu video khác. Trong ví dụ trên Fig.3, bộ giải mã video 30 bao gồm đơn vị giải mã entropy 80, đơn vị xử lý dự đoán 81, đơn vị lượng tử hóa ngược 86, đơn vị xử lý biến đổi ngược 88, bộ cộng 90, đơn vị lọc 91 và bộ nhớ hình ảnh tham chiếu 92. Đơn vị xử lý dự đoán 81 bao gồm đơn vị bù chuyển động 82 và đơn vị xử lý dự đoán nội hình ảnh 84. Bộ giải mã video 30 có thể, trong một số ví dụ, thực hiện quy trình giải mã thường nghịch đảo với quy trình mã hóa đã được mô tả đối với bộ mã hóa video 20 từ Fig 2.

Bộ nhớ đệm hình ảnh mã hóa (coded picture buffer-CPB) có thể nhận và lưu trữ dữ liệu video đã mã hóa (ví dụ, đơn vị NAL) của dòng bit. Dữ liệu video được lưu trữ trong bộ CPB 79 có thể được thu, ví dụ, từ liên kết 16, ví dụ, từ nguồn video cục

bộ, như máy quay phim, thông qua truyền thông mạng có dây hoặc không dây của dữ liệu video, hoặc bằng cách truy cập vật ghi dữ liệu vật lý. CPB 79 có thể tạo bộ nhớ dữ liệu video mà lưu trữ dữ liệu video được mã hóa từ dòng bit video được mã hóa. CPB 79 có thể là bộ nhớ hình ảnh tham chiếu mà lưu trữ dữ liệu video tham chiếu để dùng để giải mã dữ liệu video bởi bộ giải mã video 30, ví dụ, các chế độ mã hóa nội hình ảnh hoặc liên hình ảnh. CPB 79 và bộ nhớ hình ảnh tham chiếu 92 có thể được tạo ra bởi thiết bị nhớ bất kỳ, như bộ nhớ truy cập ngẫu nhiên động (DRAM), bao gồm DRAM đồng bộ (SDRAM), RAM từ trở (MRAM), RAM điện trở (RRAM), hoặc các loại thiết bị nhớ khác. CPB 79 và bộ nhớ hình ảnh tham chiếu 92 có thể được cung cấp bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Trong các ví dụ khác nhau, CPB 79 có thể nằm trên chip cùng các bộ phận khác của bộ giải mã video 30, hoặc không nằm trên chip so với các bộ phận đó.

Trong quá trình giải mã, bộ giải mã video 30 nhận dòng bit video mã hóa mà biểu diễn các khối video của lát video được mã hóa và các phần tử cú pháp gắn kèm từ bộ mã hóa video 20. Bộ giải mã video 30 có thể nhận dòng bit video mã hóa từ mạng 29. Thực thể mạng 29 có thể là, ví dụ, máy chủ, MANE, bộ soạn thảo/bộ nối video, hoặc thiết bị như vậy khác được tạo cấu hình để thực thi một hoặc nhiều kỹ thuật được mô tả trên đây. Thực thể mạng 29 có thể bao gồm hoặc không bao gồm bộ mã hóa video, như bộ mã hóa video 20. Một số kỹ thuật được mô tả trong bản mô tả này có thể được thực thi bởi thực thể mạng 29 trước khi thực thể mạng 29 truyền dòng bit video mã hóa đến bộ giải mã video 30. Trong một số hệ thống giải mã video, thực thể mạng 29 và bộ giải mã video 30 có thể là các bộ phận của các thiết bị riêng biệt, còn theo trường hợp khác, chức năng được mô tả liên quan đến thực thể mạng 29 có thể được thực hiện bởi cùng một thiết bị bao gồm bộ mã hóa 30. Thực thể mạng 29 có thể là được xem là thiết bị video. Hơn nữa, theo một số ví dụ, thực thể mạng 29 là thiết bị tạo tập tin 34 trên Fig.1.

Đơn vị giải mã entropy 80 của bộ giải mã 30 giải mã entropy các phần tử cú pháp cụ thể của dòng bit để tạo ra các hệ số lượng tử hóa, vector chuyển động và các phần tử cú pháp khác. Đơn vị giải mã entropy 80 chuyển tiếp các vector chuyển động và các phần tử cú pháp khác đến đơn vị xử lý dự đoán 81. Bộ giải mã video 30 có thể nhận các phần tử cú pháp ở mức lát video và/hoặc mức khối video.

Khi lát video được mã hóa dưới dạng lát được mã hóa nội hình ảnh (I), đơn vị xử lý dự đoán nội hình ảnh 84 của đơn vị xử lý dự đoán 81 có thể tạo ra dữ liệu dự đoán cho khối video của lát video hiện thời dựa trên chế độ dự đoán nội hình ảnh được báo hiệu và dữ liệu từ các khối được giải mã trước đó của khung hoặc hình ảnh hiện thời. Khi khung video được mã hóa dưới dạng lát được mã hóa liên hình ảnh (tức là, B, P hoặc GPB), đơn vị bù chuyển động 82 của đơn vị xử lý dự đoán 81 tạo ra các khối dự đoán cho khối video của lát video hiện thời dựa vào các vector chuyển động và các phần tử cú pháp khác nhận được từ đơn vị giải mã entropy 80. Các khối dự đoán có thể được tạo ra từ một trong các hình ảnh tham chiếu trong một trong các danh sách hình ảnh tham chiếu. Bộ giải mã video 30 có thể tạo ra danh sách khung tham chiếu, Danh sách 0 và Danh sách 1, bằng cách sử dụng kỹ thuật tạo mặc định dựa trên các hình ảnh tham chiếu được lưu trữ trong bộ nhớ hình ảnh tham chiếu 92.

Đơn vị bù chuyển động 82 xác định thông tin dự đoán cho khối video của lát video hiện thời bằng cách phân tích các vector chuyển động và phần tử cú pháp và sử dụng thông tin dự đoán để tạo ra khối dự đoán cho khối video hiện thời được giải mã. Ví dụ, đơn vị bù chuyển động 82 sử dụng một số phần tử cú pháp trong số các phần tử cú pháp nhận được để xác định chế độ dự đoán (ví dụ, dự đoán nội hình ảnh hoặc liên hình ảnh) được sử dụng để mã hóa các khối video, loại lát dự đoán liên hình ảnh (ví dụ, lát B, lát P hoặc lát GPB), thông tin cấu trúc cho một hoặc nhiều danh sách hình ảnh tham chiếu cho lát này, các vector chuyển động cho mỗi khối video được mã hóa liên hình ảnh của lát này, trạng thái dự đoán liên hình ảnh đối với mỗi khối video được mã hóa liên hình ảnh của lát này và thông tin khác để giải mã các khối video trong lát video hiện thời.

Đơn vị bù chuyển động 82 cũng có thể thực hiện kỹ thuật phép nội suy dựa trên các bộ lọc nội suy. Đơn vị bù chuyển động 82 có thể sử dụng các bộ lọc nội suy như được dùng bởi bộ mã hóa video 20 trong khi mã hóa các khối video để tính toán các giá trị đã nội suy cho các điểm ảnh dưới số nguyên của các khối tham chiếu. Trong trường hợp này, đơn vị bù chuyển động 82 có thể xác định các bộ lọc nội suy được sử dụng bởi bộ mã hóa video 20 từ các phần tử cú pháp nhận được và có thể sử dụng các bộ lọc nội suy này để tạo ra các khối dự đoán.

Đơn vị lượng tử hóa ngược 86 lượng tử hóa ngược, tức là, khử lượng tử hóa, các hệ số biến đổi đã lượng tử hóa được cung cấp trong dòng bit và được giải mã bởi

đơn vị giải mã entropy 80. Quy trình lượng tử hóa ngược có thể bao gồm sử dụng tham số lượng tử hóa được tính bởi bộ mã hóa 20 cho mỗi khối video trong lát video để xác định mức độ lượng tử hóa và, tương tự, mức độ lượng tử hóa ngược cần được áp dụng. Đơn vị xử lý biến đổi ngược 88 áp dụng quy trình biến đổi ngược, ví dụ, DCT ngược, biến đổi số nguyên chiều ngược, hoặc quy trình biến đổi ngược tương tự về mặt khái niệm, vào các hệ số biến đổi để tạo ra các khối dư trong miền điểm ảnh.

Sau khi đơn vị bù chuyển động 82 tạo ra khối dự đoán cho khối video hiện thời dựa vào các vectơ chuyển động và các phần tử cú pháp khác, bộ giải mã video 30 tạo khối video được giải mã bằng cách cộng các khối dư từ đơn vị xử lý biến đổi ngược 88 với các khối dự đoán tương ứng được tạo bởi đơn vị bù chuyển động 82. Bộ cộng 90 là thành phần hoặc các thành phần mà thực hiện phép tính cộng này. Nếu muốn, bộ lọc vòng lặp (hoặc trong vòng lặp mã hóa hoặc sau vòng lặp mã hóa) cũng có thể được sử dụng để làm trơn sự chuyển tiếp điểm ảnh hoặc theo cách khác cải thiện chất lượng video. Đơn vị lọc 91 được dự định là một hoặc nhiều bộ lọc vòng lặp như bộ lọc tách khối, bộ lọc vòng lặp tương thích (adaptive loop filter-ALF), và bộ lọc dịch chuyển tương thích mẫu (sample adaptive offset-SAO). Mặc dù đơn vị lọc 91 được thể hiện trên Fig.3 là đang ở trong bộ lọc vòng lặp, nhưng trong các cấu hình khác, đơn vị lọc 91 có thể được dùng làm bộ lọc sau vòng lặp. Các khối video được giải mã trong khung hoặc hình ảnh cho trước khi đó được lưu trữ trong bộ nhớ hình ảnh tham chiếu 92, lưu trữ các hình ảnh tham chiếu dùng cho việc bù chuyển động sau này. Bộ nhớ hình ảnh tham chiếu 92 cũng lưu trữ video đã giải mã để sau đó biểu diễn trên thiết bị hiển thị, như thiết bị hiển thị 32 trên Fig.1.

Bộ giải mã video 30 trên Fig.3 là một ví dụ về bộ giải mã video được tạo cấu hình để giải mã dữ liệu video có thể được lưu trữ bằng cách sử dụng kỹ thuật định dạng tập tin được mô tả trong bản mô tả này.

Fig.4 là sơ đồ khối minh họa tập thiết bị ví dụ tạo thành một phần của mạng 100. Theo ví dụ này, mạng 100 bao gồm các thiết bị định tuyến 104A, 104B (thiết bị định tuyến 104) và thiết bị chuyển mã 106. Thiết bị định tuyến 104 và thiết bị chuyển mã 106 được dự định là một số lượng nhỏ các thiết bị có thể là một phần của mạng 100. Các thiết bị mạng khác như bộ ngắt mạch, ống nối ngoài, cổng nối, tường lửa, cầu nối và thiết bị tương tự khác cũng có thể được bao gồm trong mạng 100. Ngoài ra, các thiết bị mạng khác có thể được cung cấp dọc theo đường mạng giữa thiết bị máy

chủ 102 và thiết bị khách 108. Thiết bị máy chủ 102 có thể tương ứng với thiết bị nguồn 12 (Fig.1), còn thiết bị khách 108 có thể tương ứng với thiết bị đích 14 (Fig.1), theo một số ví dụ.

Nói chung, thiết bị định tuyến 104 thực hiện một hoặc nhiều giao thức định tuyến để trao đổi dữ liệu mạng qua mạng 100. Theo một số ví dụ, thiết bị định tuyến 104 có thể được tạo cấu hình để thực hiện các thao tác ủy nhiệm hoặc lưu trữ. Do vậy, theo một số ví dụ, thiết bị định tuyến 104 có thể được gọi là thiết bị ủy nhiệm. Nói chung, thiết bị định tuyến 104 chạy các giao thức định tuyến để tìm ra các tuyến qua mạng 100. Bằng cách chạy các giao thức định tuyến này, thiết bị định tuyến 104B có thể phát hiện ra đường mạng từ chính thiết bị này đến thiết bị máy chủ 102 qua thiết bị định tuyến 104A.

Kỹ thuật của bản mô tả này có thể được thực hiện bởi thiết bị mạng như thiết bị định tuyến 104 và thiết bị chuyển mã 106, nhưng cũng có thể được thực hiện bởi thiết bị khách 108. Theo cách này, thiết bị định tuyến 104, thiết bị chuyển mã 106, và thiết bị khách 108 là các ví dụ về thiết bị được tạo cấu hình để thực hiện các kỹ thuật theo sáng chế. Hơn nữa, thiết bị trên Fig.1, và bộ mã hóa 20 được minh họa trên Fig.2, và bộ giải mã 30 được minh họa trên Fig.3 là các ví dụ về thiết bị có thể được tạo cấu hình để thực hiện một hoặc nhiều kỹ thuật theo sáng chế.

Fig.5A là sơ đồ khái niệm minh họa cấu trúc làm ví dụ của tập tin 300, theo một hoặc nhiều kỹ thuật của sáng chế. Trong ví dụ trên Fig.5A, tập tin 300 bao gồm hộp phim 302 và nhiều hộp dữ liệu truyền thông 304. Mặc dù được minh họa trong ví dụ trên Fig.5A là ở trong cùng một tập tin, theo các ví dụ khác hộp phim 302 và hộp dữ liệu truyền thông 304 có thể ở trong các tập tin riêng biệt. Như nêu trên, một hộp có thể là khối kiến trúc hướng đến đối tượng được xác định bởi ký hiệu nhận dạng loại duy nhất và chiều dài. Chẳng hạn, một hộp có thể là cấu trúc phần tử cú pháp theo ISOBMFF, bao gồm loại hộp được mã hóa bởi bốn ký tự, tổng số byte của hộp này và tải tin.

Hộp phim 302 có thể chứa siêu dữ liệu cho các rãnh của tập tin 300. Mỗi rãnh của tập tin 300 có thể bao gồm dòng dữ liệu truyền thông liên tục. Mỗi hộp dữ liệu truyền thông 304 có thể bao gồm một hoặc nhiều mẫu 305. Mỗi mẫu 305 có thể bao gồm một đơn vị truy cập âm thanh hoặc video. Như được mô tả ở phần khác trong bản mô tả này, mỗi đơn vị truy cập có thể bao gồm nhiều hình ảnh mã hóa trong mã hóa

đa khung hình (ví dụ, MV-HEVC và 3D-HEVC) và mã hóa video mở rộng được (ví dụ, SHVC). Chẳng hạn, đơn vị truy cập có thể bao gồm một hoặc nhiều hình ảnh mã hóa cho mỗi lớp.

Hơn thế nữa, theo ví dụ trên Fig.5A, hộp phim 302 bao gồm hộp rãnh 306. Hộp rãnh 306 có thể chứa siêu dữ liệu cho một rãnh của tập tin 300. Theo các ví dụ khác, hộp phim 302 có thể bao gồm nhiều hộp rãnh cho các rãnh khác nhau của tập tin 300. Hộp rãnh 306 bao gồm hộp truyền thông 307. Hộp truyền thông 307 có thể chứa tất cả các đối tượng thể hiện thông tin về dữ liệu truyền thông trong rãnh. Hộp truyền thông 307 bao gồm hộp thông tin truyền thông 308. Hộp thông tin truyền thông 308 có thể chứa tất cả các đối tượng thể hiện thông tin đặc trưng về truyền thông trong rãnh. Hộp thông tin truyền thông 308 bao gồm hộp bảng mẫu 309. Hộp bảng mẫu 309 có thể mô tả siêu dữ liệu dành riêng cho mẫu.

Trong ví dụ trên Fig.5A, hộp bảng mẫu 309 bao gồm hộp SampleToGroup 310 và hộp SampleGroupDescription (mô tả nhóm mẫu) 312, và hộp SampleGroupDescription 312 bao gồm hộp oinf 316. Theo cách khác, hộp bảng mẫu 309 có thể bao gồm các hộp khác ngoài hộp SampleToGroup 310 và hộp SampleGroupDescription 312, và/hoặc có thể bao gồm nhiều hộp SampleToGroup và hộp SampleGroupDescription. Hộp SampleToGroup 310 có thể sắp xếp các mẫu (ví dụ, mẫu cụ thể trong số các mẫu 305) vào nhóm mẫu. Hộp SampleGroupDescription 312 có thể mô tả một đặc tính chung của các mẫu trong nhóm gồm các mẫu (tức là, nhóm mẫu). Hơn nữa, hộp bảng mẫu 309 có thể bao gồm các hộp mục nhập mẫu 311. Mỗi hộp mục nhập mẫu 311 có thể tương ứng với một mẫu trong nhóm mẫu. Theo một số ví dụ, hộp mục nhập mẫu 311 là các trường hợp của loại mục nhập mẫu truy cập được ngẫu nhiên mà mở rộng loại mô tả nhóm mẫu cơ sở.

Theo một hoặc nhiều kỹ thuật của bản mô tả này, hộp SampleGroupDescription 312 có thể mô tả mỗi mẫu của nhóm mẫu chứa ít nhất một hình ảnh IRAP. Theo cách này, thiết bị tạo tập tin 34 có thể tạo ra tập tin bao gồm hộp rãnh 306 chứa siêu dữ liệu cho một rãnh trong tập tin 300. Dữ liệu truyền thông cho rãnh này bao gồm chuỗi các mẫu 305. Mỗi mẫu này có thể là đơn vị truy cập video của dữ liệu video nhiều lớp (ví dụ, dữ liệu video SHVC, MV-HEVC hoặc 3D-HEVC). Hơn nữa, là một phần của quá trình tạo tập tin 300, thiết bị tạo tập tin 34 có thể tạo ra, trong tập tin 300, một hộp phụ (tức là, hộp bảng mẫu 309) ghi lại tất cả các mẫu 305

chứa ít nhất một hình ảnh IRAP. Nói cách khác, hộp phụ này nhận dạng tất cả các mẫu 305 chứa ít nhất một hình ảnh IRAP. Trong ví dụ trên Fig.5A, hộp phụ này xác định một nhóm mẫu mà ghi (ví dụ, nhận dạng) tất cả các mẫu 305 chứa ít nhất một hình ảnh IRAP. Nói cách khác, hộp phụ này mô tả các mẫu 305 chứa ít nhất một hình ảnh IRAP thuộc nhóm mẫu.

Theo các kỹ thuật của sáng chế, hộp SampleGroupDescription 312 có thể bao gồm hộp oinf 316. Hộp oinf có thể lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video. Thông tin định dạng biểu diễn có thể bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng màu sắc. Ngoài ra, hộp oinf có thể lưu trữ số đếm lớp mà chỉ báo số lượng lớp điểm hoạt động cần thiết của dữ liệu video. Hộp oinf còn có thể lưu trữ thêm thông tin về tốc độ bit cho mỗi điểm hoạt động của dữ liệu video. Do đó, có thể không cần báo hiệu hộp tốc độ bit sau hộp cấu hình vì thông tin về tốc độ bit đang được báo hiệu trong hộp oinf.

Ngoài ra, có thể không cần lưu trữ thông tin PTL biên dạng, tầng và độ, thông tin định dạng biểu diễn, và thông tin về tốc độ khung trong bản ghi cấu hình bộ giải mã của định dạng tập tin. Toàn bộ các thông tin khác trong bản ghi cấu hình bộ giải mã có thể liên quan đến tất cả các lớp của dữ liệu video trong rãnh. Bản ghi cấu hình bộ giải mã cho mỗi lớp dữ liệu video có thể lưu trữ thông tin định dạng biểu diễn và thông tin về tốc độ khung. Bản ghi cấu hình bộ giải mã có thể lưu trữ thông tin về tính song song cho mỗi lớp dữ liệu video. Các tập tin thường chỉ bao gồm một bản ghi cấu hình bộ giải mã cho một rãnh, nhưng một rãnh có thể chứa một hoặc nhiều lớp và một hoặc nhiều điểm hoạt động. Thông tin PTL, thông tin định dạng biểu diễn, và thông tin về tốc độ khung có thể liên quan đến mỗi lớp hoặc mỗi OP. Do đó, không giống với định dạng tập tin HEVC mà chỉ hỗ trợ một lớp, bản ghi cấu hình bộ giải mã có thể không thể hỗ trợ sự liên kết đối với định dạng tập tin LHEVC hỗ trợ nhiều lớp một cách thích hợp.

Bản ghi cấu hình bộ giải mã có thể không lưu trữ chỉ số điểm hoạt động trong bản ghi cấu hình bộ giải mã, trong đó chỉ số điểm hoạt động nghĩa là chỉ số của điểm hoạt động được ghi trong hộp thông tin về điểm hoạt động. Việc lưu trữ chỉ số điểm hoạt động trong bản ghi cấu hình bộ giải mã có thể khiến thiết bị phát rãnh (tức là liên kết với bản ghi cấu hình bộ giải mã đó) để phát điểm hoạt động được chỉ đến bởi chỉ số điểm hoạt động đó. Tuy nhiên, có thể có nhiều điểm hoạt động có sẵn. Việc loại bỏ

chỉ số điểm hoạt động có thể cho phép thiết bị phát lại nhận dạng tất cả các điểm hoạt động được hỗ trợ bởi tập tin. Bản ghi cấu hình bộ giải mã có thể lưu trữ danh sách các chỉ số điểm hoạt động liên quan đến rãnh của dữ liệu video. Bản ghi cấu hình bộ giải mã có thể, ví dụ, thu được từ thông tin trong hộp mục nhập mẫu 311 trên Fig.5A.

Bản ghi cấu hình bộ giải mã lưu trữ thông tin như kích thước của trường độ dài dùng trong mỗi mẫu để chỉ báo độ dài của các đơn vị NAL của nó cũng như các tập tham số, nếu được lưu trữ trong mục nhập mẫu. Bản ghi cấu hình bộ giải mã có thể, ví dụ, được tạo khung ngoài (ví dụ, kích thước của nó phải được hỗ trợ bởi cấu trúc chứa nó). Bản ghi cấu hình bộ giải mã còn có thể chứa trường phiên bản để nhận dạng phiên bản kỹ thuật được tuân theo, có các thay đổi thích hợp cho bản ghi đang được chỉ báo bằng sự thay đổi số lượng phiên bản. Ngược lại, các dạng mở rộng thích hợp với bản ghi này có thể không đòi hỏi thay đổi mã phiên bản cấu hình. Bản ghi cấu hình bộ giải mã còn có thể bao gồm các giá trị cho nhiều phần tử cú pháp HEVC như `general_profile_space`, `general_tier_flag`, `general_profile_idc`, `general_profile_compatibility_flags`, `general_constraint_indicator_flags`, `general_level_idc`, `min_spatial_segmentation_idc`, `chroma_format_idc`, `bit_depth_luma_minus8` và `bit_depth_chroma_minus8`, mà được quy định trong HEVC. Bản ghi cấu hình bộ giải mã có thể chứa thông tin khái quát mà liên kết, với rãnh mà chứa bản ghi cấu hình, số lượng của các lớp con theo thời gian, thông tin phân đoạn, loại song song được hỗ trợ, và các tập tham số các đơn vị NAL (ví dụ VPS, SPS, PPS, SEI, v.v.)

Hơn nữa, theo một hoặc nhiều kỹ thuật của sáng chế, mỗi hộp mục nhập mẫu 311 có thể bao gồm giá trị (ví dụ, `all_pics_are_IRAP`) chỉ ra liệu tất cả các hình ảnh mã hóa trong mẫu tương ứng có phải là hình ảnh IRAP hay không. Theo một số ví dụ, giá trị này bằng 1 chỉ rõ rằng không phải tất cả các hình ảnh mã hóa trong mẫu là hình ảnh IRAP. Giá trị này bằng 0 chỉ rõ rằng không đòi hỏi mỗi hình ảnh mã hóa trong mỗi mẫu của nhóm mẫu là hình ảnh IRAP.

Theo một số ví dụ, khi không phải tất cả các hình ảnh mã hóa của mẫu là hình ảnh IRAP, thiết bị tạo tập tin 34 có thể bao gồm, trong một trong số các hộp mục nhập mẫu 311 cho một mẫu cụ thể, một giá trị (ví dụ, `num_IRAP_pics`) chỉ ra số lượng hình ảnh IRAP trong mẫu cụ thể. Ngoài ra, thiết bị tạo tập tin 34 có thể bao gồm, trong mục nhập mẫu cho mẫu cụ thể, các giá trị chỉ ra ký hiệu nhận dạng lớp của các hình

ảnh IRAP trong mẫu cụ thể. Thiết bị tạo tập tin 34 cũng có thể bao gồm, trong mục nhập mẫu cho mẫu cụ thể, giá trị chỉ ra kiểu đơn vị NAL của các đơn vị NAL VCL trong các hình ảnh IRAP của mẫu cụ thể.

Hơn thế nữa, theo ví dụ trên Fig.5A, hộp bảng mẫu 309 bao gồm hộp thông tin mẫu con 314. Mặc dù ví dụ trên Fig.5A chỉ thể hiện một hộp thông tin mẫu con, nhưng hộp bảng mẫu 309 có thể bao gồm nhiều hộp thông tin mẫu con. Nói chung, một hộp thông tin mẫu con được thiết kế để chứa thông tin mẫu con. Mẫu con là một khoảng byte liền kề của một mẫu. ISO/IEC 14496-12 chỉ ra rằng định nghĩa cụ thể về mẫu con được đưa ra đối với một hệ thống mã hóa xác định, như H.264/AVC hoặc HEVC.

Phần 8.4.8 của ISO/IEC 14496-15 chỉ rõ định nghĩa về mẫu con cho HEVC. Cụ thể là, phần 8.4.8 của ISO/IEC 14496-15 chỉ rõ rằng đối với việc sử dụng hộp thông tin mẫu con (8.7.7 trong ISO/IEC 14496-12) trong dòng HEVC, mẫu con được định nghĩa trên cơ sở giá trị của trường cờ của hộp thông tin mẫu con. Theo một hoặc nhiều kỹ thuật của bản mô tả này, nếu trường cờ trong hộp thông tin mẫu con 314 bằng 5, mẫu con tương ứng với hộp thông tin mẫu con 314 chứa một hình ảnh mã hóa và các đơn vị NAL không VCL liên quan. Các đơn vị NAL không VCL liên quan có thể bao gồm các đơn vị NAL chứa các thông báo SEI có thể áp dụng cho hình ảnh mã hóa và các đơn vị NAL chứa các tập tham số (ví dụ, VPS, SPS, PPS, v.v.) có thể áp dụng cho hình ảnh mã hóa.

Do vậy, theo một ví dụ, thiết bị tạo tập tin 34 có thể tạo ra tập tin (ví dụ, tập tin 300) bao gồm hộp rãnh (ví dụ, hộp rãnh 306) chứa siêu dữ liệu cho một rãnh trong tập tin. Theo một ví dụ, dữ liệu truyền thông cho một rãnh có thể bao gồm chuỗi các mẫu, mỗi mẫu là một đơn vị truy cập video của dữ liệu video nhiều lớp (ví dụ, dữ liệu video SHVC, MV-HEVC, hoặc 3D-HEVC). Hơn nữa, theo ví dụ này, là một phần của thiết bị tạo tập tin 34 tạo tập tin, thiết bị tạo tập tin 34 có thể tạo ra, trong tập tin này, hộp thông tin mẫu con (ví dụ, hộp thông tin mẫu con 314) chứa các cờ mô tả kiểu thông tin mẫu con có trong hộp thông tin mẫu con. Khi cờ có giá trị cụ thể, mẫu con tương ứng với hộp thông tin mẫu con chứa chính xác một hình ảnh mã hóa và không hoặc nhiều đơn vị NAL không VCL liên quan đến hình ảnh mã hóa này.

Hơn nữa theo một hoặc nhiều kỹ thuật của bản mô tả này, nếu trường cờ của hộp thông tin mẫu con 314 bằng 0, hộp thông tin mẫu con 314 còn bao gồm giá trị

DiscardableFlag, giá trị NoInterLayerPredFlag, giá trị LayerId và giá trị TempID. Nếu trường cờ của hộp thông tin mẫu con 314 bằng 5, hộp thông tin mẫu con 314 có thể bao gồm giá trị DiscardableFlag, giá trị VclNalUnitType, giá trị LayerId, giá trị TempID, giá trị NoInterLayerPredFlag, giá trị SubLayerRefNalUnitFlag và giá trị dự trữ.

SubLayerRefNalUnitFlag bằng 0 chỉ ra rằng tất cả các đơn vị NAL trong mẫu con là đơn vị NAL VCL của hình ảnh không tham chiếu của lớp con như được mô tả trong ISO/IEC 23008-2 (tức là HEVC). Giá trị SubLayerRefNalUnitFlag bằng 1 chỉ ra rằng tất cả các đơn vị NAL trong mẫu con là đơn vị NAL VCL của một hình ảnh tham chiếu của lớp con như được mô tả trong ISO/IEC 23008-2 (tức là, HEVC). Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, cờ bổ sung chỉ ra liệu tất cả các đơn vị NAL trong mẫu con có phải là đơn vị NAL VCL của hình ảnh không tham chiếu của lớp con hay không.

Giá trị DiscardableFlag (cờ có thể loại bỏ) chỉ ra giá trị discardable_flag của các đơn vị NAL VCL trong mẫu con. Như được mô tả trong phần A.4 của ISO/IEC 14496-15, discardable_flag được thiết lập là 1 nếu và chỉ nếu tất cả các đơn vị NAL được tách hoặc gộp có discardable_flag được thiết lập là 1 và nếu không thì thiết lập là 0. Đơn vị NAL có thể có discardable_flag được thiết lập là 1 nếu dòng bit chứa đơn vị NAL có thể được giải mã chính xác mà không có đơn vị NAL. Do vậy, đơn vị NAL là "có thể loại bỏ" nếu dòng bit chứa đơn vị NAL có thể được giải mã chính xác mà không cần đơn vị NAL. Tất cả các đơn vị NAL VCL trong mẫu con sẽ có cùng một giá trị discardable_flag. Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thì thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, cờ bổ sung (ví dụ, discardable_flag) chỉ ra liệu tất cả các đơn vị NAL VCL của mẫu con có thể loại bỏ hay không.

Giá trị NoInterLayerPredFlag chỉ ra giá trị của inter_layer_pred_enabled_flag của các đơn vị NAL VCL trong mẫu con. inter_layer_pred_enabled_flag được thiết lập là 1 nếu và chỉ nếu tất cả các đơn vị NAL VCL được tách hoặc gộp có inter_layer_pred_enabled_flag được thiết lập là 1 và nếu không thì thiết lập là 0. Tất cả các đơn vị NAL VCL trong mẫu con sẽ có cùng một giá trị inter_layer_pred_enabled_flag. Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin

mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, cờ bổ sung (ví dụ, `inter_layer_pred_enabled_flag`) chỉ ra liệu dự đoán liên lớp có thể được thực hiện cho tất cả các đơn vị NAL VCL của mẫu con hay không.

`LayerId` chỉ ra giá trị `nuh_layer_id` của các đơn vị NAL trong mẫu con. Tất cả các đơn vị NAL trong mẫu con sẽ có cùng một giá trị `nuh_layer_id`. Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thì thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, giá trị bổ sung (ví dụ, `discardable_flag`) chỉ báo ký hiệu nhận dạng lớp của mỗi đơn vị NAL của mẫu con.

`TempId` chỉ ra giá trị `TemporalId` của các đơn vị NAL trong mẫu con. Tất cả các đơn vị NAL trong mẫu con sẽ có cùng một giá trị `TemporalId`. Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thì thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, giá trị bổ sung (ví dụ, `TempId`) chỉ báo ký hiệu nhận dạng thời gian của mỗi đơn vị NAL của mẫu con.

`VclNalUnitType` (kiểu đơn vị NAL VCL) chỉ ra phần tử cú pháp `nal_unit_type` của các đơn vị NAL VCL trong mẫu con. Phần tử cú pháp `nal_unit_type` là phần tử cú pháp trong tiêu đề đơn vị NAL của đơn vị NAL. Phần tử cú pháp `nal_unit_type` mô tả kiểu của RBSP có trong đơn vị NAL. Tất cả các đơn vị NAL VCL `nal_unit_type` trong mẫu con có cùng giá trị `nal_unit_type`. Do vậy, khi thiết bị tạo tập tin 34 tạo ra hộp thông tin mẫu con 314 và các cờ có giá trị cụ thể (ví dụ, 5), thiết bị tạo tập tin 34 bao gồm, trong hộp thông tin mẫu con 314, cờ bổ sung (ví dụ, `VclNalUnitType`) chỉ ra kiểu đơn vị NAL của các đơn vị NAL VCL của mẫu con. Tất cả các đơn vị NAL VCL trong mẫu con có cùng kiểu đơn vị NAL.

Fig.5B là sơ đồ khái niệm minh họa cấu trúc làm ví dụ thay thế của tập tin 300, theo một hoặc nhiều kỹ thuật của sáng chế. Trong ví dụ trên Fig.5B, thay vì hộp `oinf 316` có trong hộp mô tả nhóm mẫu 312, như được thể hiện trên Fig.5A, hộp `oinf 316` được bao gồm trong hộp thông tin truyền thông 308 dưới dạng một hộp riêng so với hộp bảng mẫu 309. Nội dung và chức năng của các hộp khác nhau trên Fig.3B có thể giống với những gì được mô tả theo Fig.5A.

Fig.6 là sơ đồ khái niệm minh họa cấu trúc làm ví dụ của tập tin 300, theo một hoặc nhiều kỹ thuật của bản mô tả này. Như được mô tả trong phần 8.4.9 của ISO/IEC

14496-15, HEVC cho phép các mẫu định dạng tập tin chỉ được sử dụng để tham chiếu và không được xuất. Ví dụ, HEVC cho phép hình ảnh tham chiếu không được hiển thị trong video.

Hơn nữa, phần 8.4.9 của ISO/IEC 14496-15 mô tả rằng khi mẫu bất kỳ không xuất ra có mặt trong một rãnh, tập tin sẽ bị ràng buộc như sau.

Mẫu không xuất được xác định thời gian hợp thành nằm ngoài khoảng thời gian của mẫu được xuất.

Danh sách soạn thảo được sử dụng để loại bỏ thời gian hợp thành của các mẫu không được xuất.

Khi rãnh bao gồm hộp CompositionOffsetBox ('ctts'), phiên bản 1 của hộp CompositionOffsetBox sẽ được sử dụng, giá trị của sample_offset được thiết lập bằng -2^{31} cho mỗi mẫu không xuất, CompositionToDecodeBox ('cslg') sẽ được chứa trong SampleTableBox ('stbl') của rãnh, và

khi CompositionToDecodeBox có mặt cho rãnh này, giá trị của trường leastDecodeToDisplayDelta trong hộp sẽ bằng với độ lệch hợp thành nhỏ nhất trong CompositionOffsetBox ngoại trừ các giá trị sample_offset cho các mẫu không xuất. LƯU Ý: Do vậy, giá trị leastDecodeToDisplayDelta lớn hơn -2^{31} .

Như được mô tả trong ISO/IEC 14496-12, CompositionOffsetBox cung cấp độ lệch giữa thời gian giải mã và thời gian hợp thành. CompositionOffsetBox bao gồm tập các giá trị sample_offset. Mỗi giá trị sample_offset là một số nguyên không âm đưa ra độ lệch giữa thời gian hợp thành và thời gian giải mã. Thời gian hợp thành chỉ thời gian mà mẫu được xuất. Thời gian giải mã chỉ thời gian mà mẫu được giải mã.

Như nêu trên, đơn vị NAL lát mã hóa có thể bao gồm tiêu đề đoạn lát. Tiêu đề đoạn lát là một phần của đoạn lát đã mã hóa và có thể chứa các phần tử dữ liệu liên quan đến CTU thứ nhất hoặc tất cả các CTU trong đoạn lát. Theo HEVC, tiêu đề đoạn lát bao gồm phần tử cú pháp pic_output_flag. Nói chung phần tử cú pháp pic_output_flag có trong tiêu đề đoạn lát thứ nhất của một lát của một hình ảnh. Do vậy, bản mô tả này có thể đề cập đến pic_output_flag của tiêu đề đoạn lát thứ nhất của lát của hình ảnh là pic_output_flag của hình ảnh.

Như được mô tả trong phần 7.4.7.1 của HEVC WD, phần tử cú pháp pic_output_flag tác động đến quá trình xuất và loại bỏ hình ảnh giải mã như được mô

tả trong Phụ lục C của HEVC WD. Nói chung, nếu phần tử cú pháp `pic_output_flag` của tiêu đề đoạn lát đối với một đoạn lát là 1, hình ảnh bao gồm một lát tương ứng với tiêu đề đoạn lát này được xuất ra. Theo cách khác, nếu phần tử cú pháp `pic_output_flag` của tiêu đề đoạn lát đối với một đoạn lát là 0, hình ảnh bao gồm lát tương ứng với tiêu đề đoạn lát này có thể được giải mã để sử dụng làm hình ảnh tham chiếu, nhưng không được xuất ra.

Theo một hoặc nhiều kỹ thuật của sáng chế, các tham chiếu trong phần 8.4.9 theo ISO/IEC 14496-15 đến chuẩn HEVC có thể được thay thế bằng các tham chiếu tương ứng đến chuẩn SHVC, MV-HEVC, hoặc 3D-HEVC. Ngoài ra, theo một hoặc nhiều kỹ thuật của sáng chế, khi đơn vị truy nhập chứa một số hình ảnh được mã hóa mà có `pic_output_flag` bằng 1 và một số hình ảnh được mã hóa mà có `pic_output_flag` bằng 0, ít nhất hai rãnh cần phải được dùng để lưu trữ luồng. Đối với mỗi rãnh tương ứng trong số các rãnh này, tất cả các hình ảnh được mã hóa trong mỗi mẫu của rãnh tương ứng có cùng giá trị `pic_output_flag`. Do vậy, tất cả các hình ảnh mã hóa trong rãnh thứ nhất trong số các rãnh có `pic_output_flag` bằng 0 và tất cả các hình ảnh mã hóa trong rãnh thứ hai trong số các rãnh có `pic_output_flag` bằng 1.

Do vậy, theo ví dụ trên Fig.6, thiết bị tạo tập tin 34 có thể tạo ra tập tin 400. Tương tự với tập tin 300, theo ví dụ trên Fig.5A, tập tin 400 bao gồm hộp phim 402 và một hoặc nhiều hộp dữ liệu truyền thông 404. Mỗi hộp dữ liệu truyền thông 404 có thể tương ứng với rãnh khác của tập tin 400. Hộp phim 402 có thể chứa siêu dữ liệu cho các rãnh của tập tin 400. Mỗi rãnh của tập tin 400 có thể bao gồm dòng dữ liệu truyền thông liên tục. Mỗi hộp dữ liệu truyền thông 404 có thể bao gồm một hoặc nhiều mẫu 405. Mỗi mẫu 405 có thể bao gồm một đơn vị truy cập âm thanh hoặc video.

Như nêu trên, theo một số ví dụ, khi đơn vị truy cập chứa một số hình ảnh mã hóa có `pic_output_flag` bằng 1 và một số hình ảnh mã hóa khác có `pic_output_flag` bằng 0, ít nhất hai rãnh phải được sử dụng để lưu trữ dòng này. Do vậy, theo ví dụ trên Fig.6, hộp phim 402 bao gồm hộp rãnh 406 và hộp rãnh 408. Mỗi hộp rãnh 406 và 408 chứa siêu dữ liệu cho một rãnh khác của tập tin 400. Chẳng hạn, hộp rãnh 406 có thể chứa siêu dữ liệu cho một rãnh có các hình ảnh mã hóa với `pic_output_flag` bằng 0, và không có hình ảnh có `pic_output_flag` bằng 1. Hộp rãnh 408 có thể chứa

siêu dữ liệu cho một rãnh có các hình ảnh mã hóa với `pic_output_flag` bằng 1, và không có hình ảnh có `pic_output_flag` bằng 0.

Do vậy, theo một ví dụ, thiết bị tạo tập tin 34 có thể tạo ra tập tin (ví dụ, tập tin 400) bao gồm hộp dữ liệu truyền thông (ví dụ, hộp dữ liệu truyền thông 404) chứa (ví dụ, bao gồm) nội dung truyền thông. Nội dung truyền thông bao gồm chuỗi các mẫu (ví dụ, mẫu 405). Mỗi mẫu có thể là đơn vị truy cập dữ liệu video nhiều lớp. Theo ví dụ này, khi thiết bị tạo tập tin 34 tạo tập tin, đáp lại việc xác định rằng ít nhất một đơn vị truy cập của dòng bit bao gồm hình ảnh mã hóa có cờ xuất ảnh bằng 1 và hình ảnh mã hóa có cờ xuất ảnh bằng 0, thiết bị tạo tập tin 34 có thể sử dụng ít nhất hai rãnh để lưu trữ dòng bit trong tập tin đó. Đối với mỗi rãnh tương ứng từ ít nhất hai rãnh này, tất cả các hình ảnh được mã hóa trong mỗi mẫu của rãnh tương ứng có cùng giá trị cờ xuất ảnh. Các hình ảnh có cờ xuất ảnh bằng 1 được phép xuất và các hình ảnh có cờ xuất ảnh bằng 0 được phép được sử dụng làm hình ảnh tham chiếu nhưng không được phép xuất ra.

Fig.7 là lưu đồ minh họa thao tác làm ví dụ của một thiết bị tạo tập tin 34, theo một hoặc nhiều kỹ thuật của bản mô tả này. Thao tác trên Fig.7, cùng với các thao tác được minh họa trong các lưu đồ khác của bản mô tả này, là các ví dụ. Các thao tác làm ví dụ khác cùng với kỹ thuật của bản mô tả này có thể bao gồm nhiều hơn, ít hơn hoặc các thao tác khác.

Trong ví dụ trên Fig.7, thiết bị tạo tập tin 34 tạo ra tập tin. Là một phần của việc tạo ra tập tin, thiết bị tạo tập tin 34 thu được dữ liệu video nhiều lớp (170) và lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin (172). Thiết bị tạo tập tin 34 lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp oinf của định dạng tập tin (174). Thiết bị tạo tập tin 34 tạo tập tin dữ liệu video được định dạng theo định dạng tập tin (176). Thông tin định dạng biểu diễn có thể bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng màu sắc. Ngoài ra hoặc theo cách khác, thiết bị tạo tập tin 34 có thể lưu trữ thông tin tốc độ bit cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp oinf của định dạng tập tin và/hoặc có thể không báo hiệu hộp tốc độ bit sau hộp cấu hình của định dạng tập tin. Ngoài ra hoặc theo cách khác, thiết bị tạo tập tin 34 có thể không lưu trữ thông tin về biên dạng, tầng và độ (PTL), thông tin định dạng biểu diễn, và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã của định dạng tập tin và

liên quan đến tất cả thông tin trong bản ghi cấu hình bộ giải mã với tất cả các lớp của dữ liệu video nhiều lớp trong rãnh. Ngoài ra hoặc theo cách khác, thiết bị tạo tập tin 34 có thể lưu trữ số đếm lớp trong hộp oinf của định dạng tập tin, trong đó số đếm lớp chỉ báo số lượng lớp cần thiết của một điểm hoạt động của dữ liệu video nhiều lớp.

Hộp oinf có thể được bao gồm trong hộp thông tin phương tiện, và hộp oinf có thể được bao gồm trong hộp mô tả nhóm mẫu. Hộp mô tả nhóm mẫu có thể được bao gồm trong hộp bảng mẫu, và hộp bảng mẫu có thể được bao gồm trong hộp thông tin phương tiện.

Thiết bị tạo tập tin 34 có thể lưu trữ thông tin định dạng biểu diễn và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp. Ngoài ra hoặc theo cách khác, thiết bị tạo tập tin 34 có thể lưu trữ thông tin về tính song song trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp. Thiết bị tạo tập tin 34 có thể không lưu trữ chỉ số điểm hoạt động trong bản ghi cấu hình bộ giải mã của định dạng tập tin. Ngoài ra hoặc theo cách khác, thiết bị tạo tập tin 34 có thể lưu trữ danh sách các chỉ số điểm hoạt động liên quan đến rãnh của dữ liệu video nhiều lớp trong bản ghi cấu hình bộ giải mã theo định dạng tập tin.

Fig.8 là lưu đồ minh họa thao tác làm ví dụ của thiết bị đọc tập tin, như thiết bị đích 14, thực thể sau xử lý 27, hoặc thực thể mạng 29. Thao tác trên Fig.8, cùng với các thao tác được minh họa trong các lưu đồ khác của bản mô tả này, là các ví dụ. Các thao tác làm ví dụ khác cùng với kỹ thuật của bản mô tả này có thể bao gồm nhiều hơn, ít hơn hoặc các thao tác khác.

Trong ví dụ trên Fig.8, thiết bị đọc tập tin thu được tập tin dữ liệu video nhiều lớp được định dạng theo định dạng tập tin (180). Thiết bị đọc tập tin, đối với định dạng tập tin, xác định thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong hộp oinf của định dạng tập tin (182). Thiết bị đọc tập tin, có thể kết hợp với bộ giải mã video như bộ giải mã video 30, giải mã dữ liệu video nhiều lớp dựa trên thông tin định dạng biểu diễn xác định được (184).

Theo một hoặc nhiều ví dụ, các chức năng đã được mô tả có thể được thực hiện trong phần cứng, phần mềm, phần sụn, hoặc tổ hợp bất kỳ của chúng. Nếu được thực hiện trong phần mềm, thì các chức năng này có thể được lưu trữ hoặc truyền qua, dưới dạng một hoặc nhiều lệnh hoặc mã, vật ghi đọc được bởi máy tính và được thực thi bởi bộ xử lý dựa trên phần cứng. Vật ghi đọc được bằng máy tính có thể bao gồm

phương tiện lưu trữ đọc được bằng máy tính, mà tương ứng với phương tiện hữu hình chẳng hạn như phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm cả phương tiện bất kỳ mà tạo điều kiện thuận lợi cho việc truyền chương trình máy tính từ nơi này sang nơi khác, ví dụ theo giao thức truyền thông. Theo cách này, vật ghi đọc được bằng máy tính thông thường có thể tương ứng với (1) phương tiện lưu trữ hữu hình đọc được bằng máy tính mà không chuyển tiếp hoặc (2) phương tiện truyền thông như tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để tìm kiếm các lệnh, mã và/hoặc các cấu trúc dữ liệu để thực hiện các kỹ thuật được mô tả trong sáng chế này. Sản phẩm chương trình máy tính có thể bao gồm vật ghi đọc được bằng máy tính.

Ví dụ, và không giới hạn, các vật ghi đọc được bởi máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc bộ nhớ đĩa quang khác, bộ nhớ đĩa từ, hoặc các thiết bị lưu trữ từ tính khác, bộ nhớ nhanh, hoặc phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng các lệnh hoặc các cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Ngoài ra, kết nối bất kỳ cũng được gọi đúng là vật ghi đọc được bởi máy tính. Ví dụ, nếu các lệnh được truyền từ một trang mạng, máy chủ, hoặc nguồn từ xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường dây thuê bao số (digital subscriber line - DSL), hoặc các công nghệ không dây chẳng hạn như tia hồng ngoại, sóng vô tuyến và vi sóng, thì cáp đồng trục, cáp sợi quang, cáp xoắn đôi, DSL, hoặc các công nghệ không dây chẳng hạn như tia hồng ngoại, sóng vô tuyến, và vi sóng được bao gồm trong định nghĩa về vật ghi. Tuy nhiên, cần hiểu rằng vật ghi đọc được bởi máy tính và phương tiện lưu trữ dữ liệu không bao gồm các kết nối, các sóng mang, các tín hiệu, hoặc các phương tiện chuyển tiếp khác, mà thay vào đó là hướng đến vật ghi lưu trữ hữu hình, không chuyển tiếp. Đĩa từ và đĩa quang, như được sử dụng ở đây, bao gồm đĩa nén (Compact Disc - CD), đĩa laze, đĩa quang, đĩa kỹ thuật số đa năng (Digital Versatile Disc - DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa từ thường sao chép dữ liệu theo cách từ tính, trong khi đó đĩa quang sao chép dữ liệu theo cách quang học bằng các tia laze. Kết hợp của cả hai loại này cũng được bao gồm trong phạm vi của vật ghi đọc được bằng máy tính.

Các lệnh có thể được thực hiện bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu kỹ thuật số (Digital Signal Processors - DSP), các bộ vi xử lý đa năng, các mạch tích hợp chuyên dụng (Application Specific Integrated Circuits - ASIC), các mảng logic lập trình được dạng trường (field programmable logic arrays - FPGAs), hoặc mạch logic rời rạc hoặc tích hợp tương đương khác. Do vậy, thuật ngữ “bộ xử lý” được sử dụng ở đây có thể chỉ cấu trúc bất kỳ trong cấu trúc nêu trên hoặc cấu trúc khác bất kỳ thích hợp cho việc thực hiện các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, chức năng được mô tả ở đây có thể được tạo ra trong các môđun phần cứng và/hoặc phần mềm chuyên dụng được tạo cấu hình để mã hóa và giải mã, hoặc được đưa vào trong bộ mã hóa-giải mã kết hợp. Ngoài ra, các kỹ thuật này có thể được thực hiện đầy đủ trong một hoặc nhiều mạch hoặc phần tử logic.

Các kỹ thuật của sáng chế có thể được thực hiện trong nhiều loại cơ cấu hoặc thiết bị khác nhau, bao gồm máy cầm tay không dây, mạch tích hợp (integrated circuit - IC) hoặc bộ IC (ví dụ, bộ chip). Các thành phần, môđun hoặc đơn vị khác nhau được mô tả trong bản mô tả này để nhấn mạnh các khía cạnh chức năng của thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, nhưng không nhất thiết đòi hỏi việc thực hiện bởi các đơn vị phần cứng khác nhau. Thay vào đó, như được mô tả trên đây, các đơn vị khác nhau có thể được kết hợp trong đơn vị phần cứng codec hoặc được tạo ra bởi tập hợp các đơn vị phần cứng liên kết hoạt động, bao gồm một hoặc nhiều bộ xử lý như được mô tả trên đây, kết hợp với phần mềm và/hoặc phần sụn thích hợp.

Các ví dụ khác nhau đã được mô tả. Các ví dụ này và các ví dụ khác đều nằm trong phạm vi của yêu cầu bảo hộ dưới đây.

Yêu cầu bảo hộ

1. Phương pháp xử lý dữ liệu video nhiều lớp, phương pháp này bao gồm các bước:

thu được dữ liệu video nhiều lớp có nhiều hơn một điểm hoạt động;

lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin, trong đó định dạng tập tin bao gồm nhóm mẫu thông tin điểm hoạt động (operating points information-oinf) mà nhận dạng các điểm hoạt động có trong dữ liệu video nhiều lớp;

lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong nhóm mẫu oinf, trong đó thông tin định dạng biểu diễn bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng màu sắc; và tạo tập tin dữ liệu video được định dạng theo định dạng tập tin.

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

lưu trữ thông tin tốc độ bit cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong nhóm mẫu oinf của định dạng tập tin; và

không báo hiệu nhóm mẫu tốc độ bit sau nhóm mẫu cấu hình của định dạng tập tin.

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

không lưu trữ thông tin về biên dạng, tầng và độ (profile, tier, and level-PTL), thông tin định dạng biểu diễn và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã của định dạng tập tin; và

liên kết tất cả thông tin trong bản ghi cấu hình bộ giải mã với tất cả các lớp của dữ liệu video nhiều lớp trong rãnh.

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

lưu trữ thông tin định dạng biểu diễn và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp.

5. Phương pháp theo điểm 4, trong đó phương pháp này còn bao gồm bước:

lưu trữ thông tin về tính song song trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp.

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

không lưu trữ chỉ số điểm hoạt động trong bản ghi cấu hình bộ giải mã của định dạng tập tin.

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

lưu trữ danh sách gồm các chỉ số điểm hoạt động liên quan đến rãnh của dữ liệu video nhiều lớp trong bản ghi cấu hình bộ giải mã của định dạng tập tin.

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

lưu trữ số đếm lớp trong nhóm mẫu oinf của định dạng tập tin, trong đó số đếm lớp chỉ báo số lượng lớp cần thiết của một điểm hoạt động của dữ liệu video nhiều lớp.

9. Phương pháp theo điểm 1, trong đó nhóm mẫu oinf được bao gồm trong nhóm mẫu thông tin phương tiện.

10. Phương pháp theo điểm 9, trong đó nhóm mẫu oinf còn được bao gồm trong nhóm mẫu mô tả nhóm mẫu, trong đó nhóm mẫu mô tả nhóm mẫu được bao gồm trong nhóm mẫu bảng mẫu, và trong đó nhóm mẫu bảng mẫu được bao gồm trong nhóm mẫu thông tin phương tiện.

11. Phương pháp theo điểm 1, trong đó mỗi điểm hoạt động của dữ liệu video nhiều lớp lần lượt bao gồm dòng bit được tạo ra từ một dòng bit khác bằng hoạt động của quá trình tách dòng bit con với dòng bit khác này.

12. Thiết bị video để xử lý dữ liệu video nhiều lớp, thiết bị này bao gồm:

phương tiện lưu trữ dữ liệu được tạo cấu hình để lưu trữ dữ liệu video nhiều lớp; và

một hoặc nhiều bộ xử lý được tạo cấu hình để:

thu được dữ liệu video nhiều lớp có nhiều hơn một điểm hoạt động;

lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin, trong đó định dạng tập tin bao gồm nhóm mẫu thông tin điểm hoạt động (oinf) mà nhận dạng các điểm hoạt động có trong dữ liệu video nhiều lớp;

lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong nhóm mẫu oinf, trong đó thông tin định dạng biểu diễn bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng màu sắc; và tạo tập tin dữ liệu video được định dạng theo định dạng tập tin.

13. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

lưu trữ thông tin tốc độ bit cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong nhóm mẫu oinf của định dạng tập tin; và

không báo hiệu nhóm mẫu tốc độ bit sau nhóm mẫu cấu hình của định dạng tập tin.

14. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

không lưu trữ thông tin về biên dạng, tầng và độ (profile, tier, and level-PTL), thông tin định dạng biểu diễn và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã của định dạng tập tin; và

liên kết tất cả thông tin trong bản ghi cấu hình bộ giải mã với tất cả các lớp của dữ liệu video nhiều lớp trong rãnh.

15. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

lưu trữ thông tin định dạng biểu diễn và thông tin tốc độ khung trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp.

16. Thiết bị theo điểm 15, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

lưu trữ thông tin về tính song song trong bản ghi cấu hình bộ giải mã cho mỗi lớp của dữ liệu video nhiều lớp.

17. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:

không lưu trữ chỉ số điểm hoạt động trong bản ghi cấu hình bộ giải mã của định dạng tập tin.

18. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:
 lưu trữ danh sách chỉ số điểm hoạt động liên quan đến rãnh của dữ liệu video
 nhiều lớp trong bản ghi cấu hình bộ giải mã của định dạng tập tin.
19. Thiết bị theo điểm 12, trong đó một hoặc nhiều bộ xử lý còn được tạo cấu hình để:
 lưu trữ số đếm lớp trong nhóm mẫu oinf của định dạng tập tin, trong đó số đếm
 lớp chỉ báo số lượng lớp cần thiết của một điểm hoạt động của dữ liệu video nhiều
 lớp.
20. Thiết bị theo điểm 12, trong đó nhóm mẫu oinf được bao gồm trong nhóm mẫu
 thông tin phương tiện.
21. Thiết bị theo điểm 20, trong đó nhóm mẫu oinf còn được bao gồm trong nhóm
 mẫu mô tả nhóm mẫu, trong đó nhóm mẫu mô tả nhóm mẫu được bao gồm trong
 nhóm mẫu bảng mẫu, và trong đó nhóm mẫu bảng mẫu được bao gồm trong nhóm
 mẫu thông tin phương tiện.
22. Thiết bị theo điểm 12, trong đó mỗi điểm hoạt động của dữ liệu video nhiều lớp
 lần lượt bao gồm dòng bit được tạo ra từ dòng bit khác bằng hoạt động của quá trình
 tách dòng bit con với dòng bit khác này.
23. Thiết bị video để xử lý dữ liệu video nhiều lớp, thiết bị này bao gồm:
 phương tiện thu được dữ liệu video nhiều lớp có nhiều hơn một điểm hoạt
 động;
 phương tiện lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin, trong đó
 định dạng tập tin bao gồm nhóm mẫu thông tin điểm hoạt động (oinf) mà nhận dạng
 các điểm hoạt động có trong dữ liệu video nhiều lớp;
 phương tiện lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của
 dữ liệu video nhiều lớp trong nhóm mẫu oinf, trong đó thông tin định dạng biểu diễn
 bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng
 màu sắc; và

phương tiện tạo tập tin dữ liệu video được định dạng theo định dạng tập tin.

24. Thiết bị theo điểm 23, trong đó nhóm mẫu oinf được bao gồm trong nhóm mẫu thông tin phương tiện.

25. Thiết bị theo điểm 24, trong đó nhóm mẫu oinf còn được bao gồm trong nhóm mẫu mô tả nhóm mẫu, trong đó nhóm mẫu mô tả nhóm mẫu được bao gồm trong nhóm mẫu bảng mẫu, và trong đó nhóm mẫu bảng mẫu được bao gồm trong nhóm mẫu thông tin phương tiện.

26. Vật ghi bất biến đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi khiến cho một hoặc nhiều bộ xử lý thực hiện các bước:

thu được dữ liệu video nhiều lớp có nhiều hơn một điểm hoạt động;

lưu trữ dữ liệu video nhiều lớp dưới định dạng tập tin, trong đó định dạng tập tin bao gồm nhóm mẫu thông tin điểm hoạt động (oinf) mà nhận dạng các điểm hoạt động có trong dữ liệu video nhiều lớp;

lưu trữ thông tin định dạng biểu diễn cho mỗi điểm hoạt động của dữ liệu video nhiều lớp trong nhóm mẫu oinf, trong đó thông tin định dạng biểu diễn bao gồm một hoặc nhiều trong số độ phân giải không gian, độ sâu bit, hoặc định dạng màu sắc; và

tạo tập tin dữ liệu video được định dạng theo định dạng tập tin.

27. Vật ghi bất biến đọc được bằng máy tính theo điểm 26, trong đó nhóm mẫu oinf được bao gồm trong nhóm mẫu thông tin phương tiện.

28. Vật ghi bất biến đọc được bằng máy tính theo điểm 27, trong đó nhóm mẫu oinf còn được bao gồm trong nhóm mẫu mô tả nhóm mẫu, trong đó nhóm mẫu mô tả nhóm mẫu được bao gồm trong nhóm mẫu bảng mẫu, và trong đó nhóm mẫu bảng mẫu được bao gồm trong nhóm mẫu thông tin phương tiện.

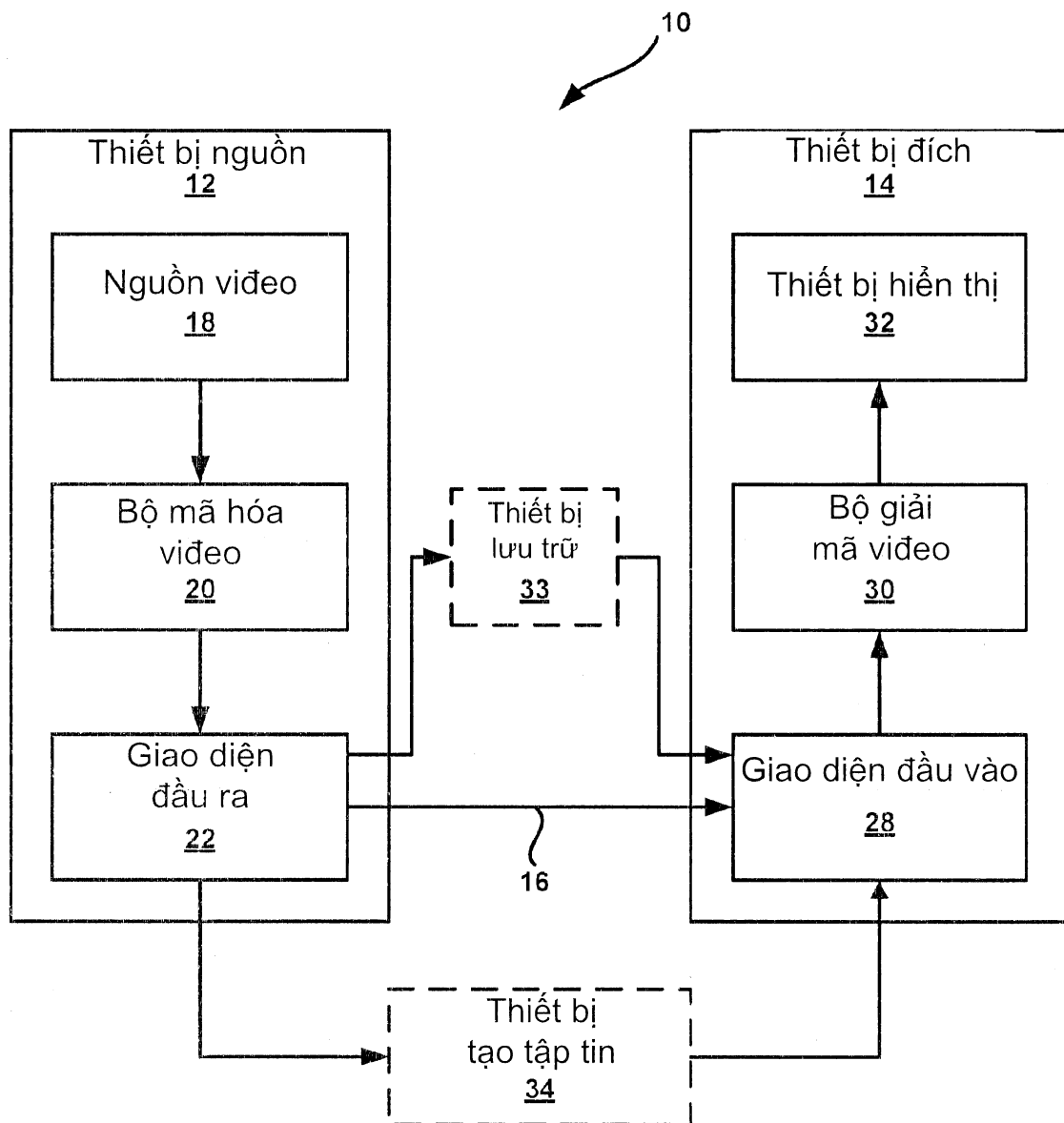


FIG. 1

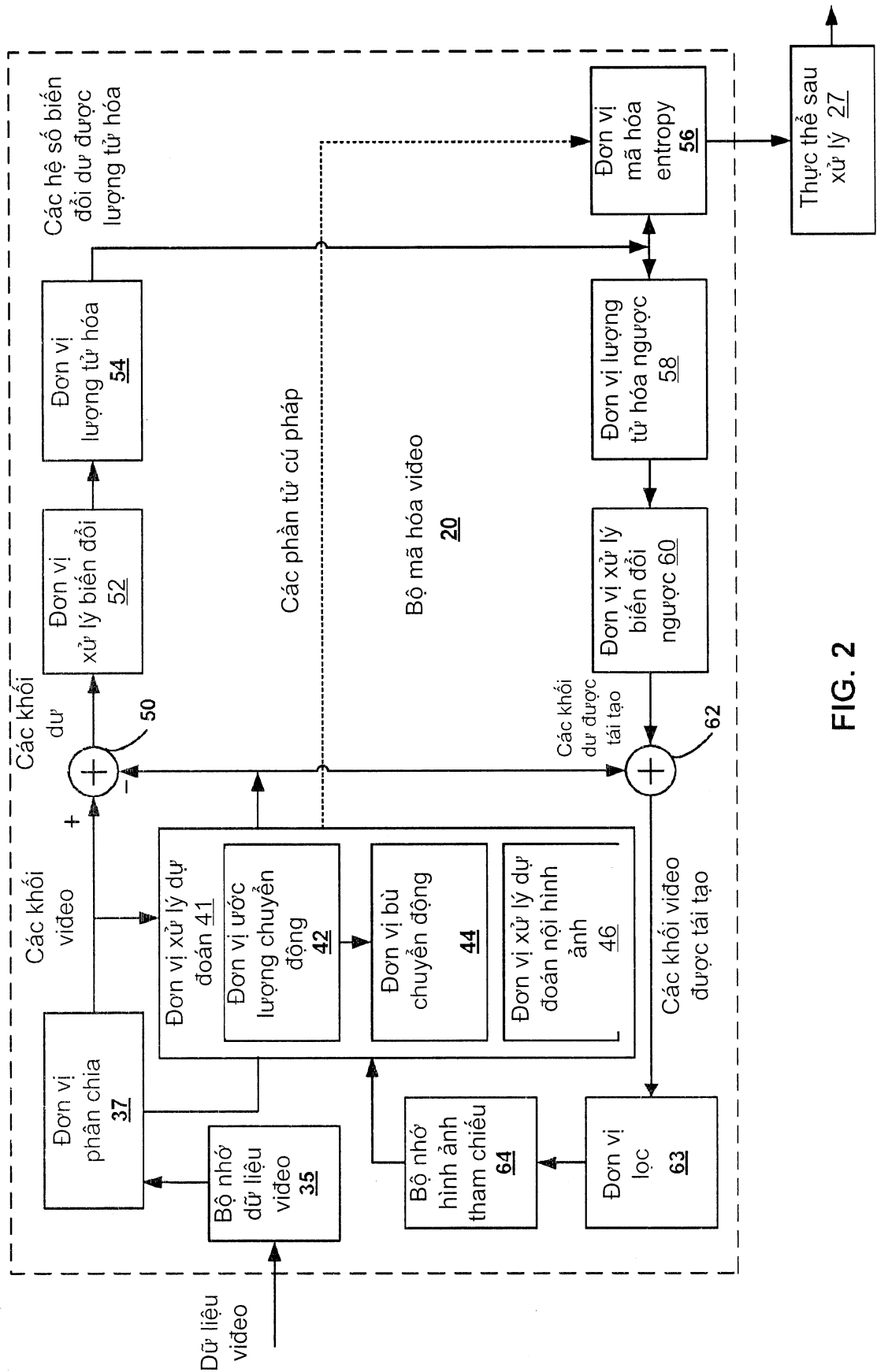


FIG. 2

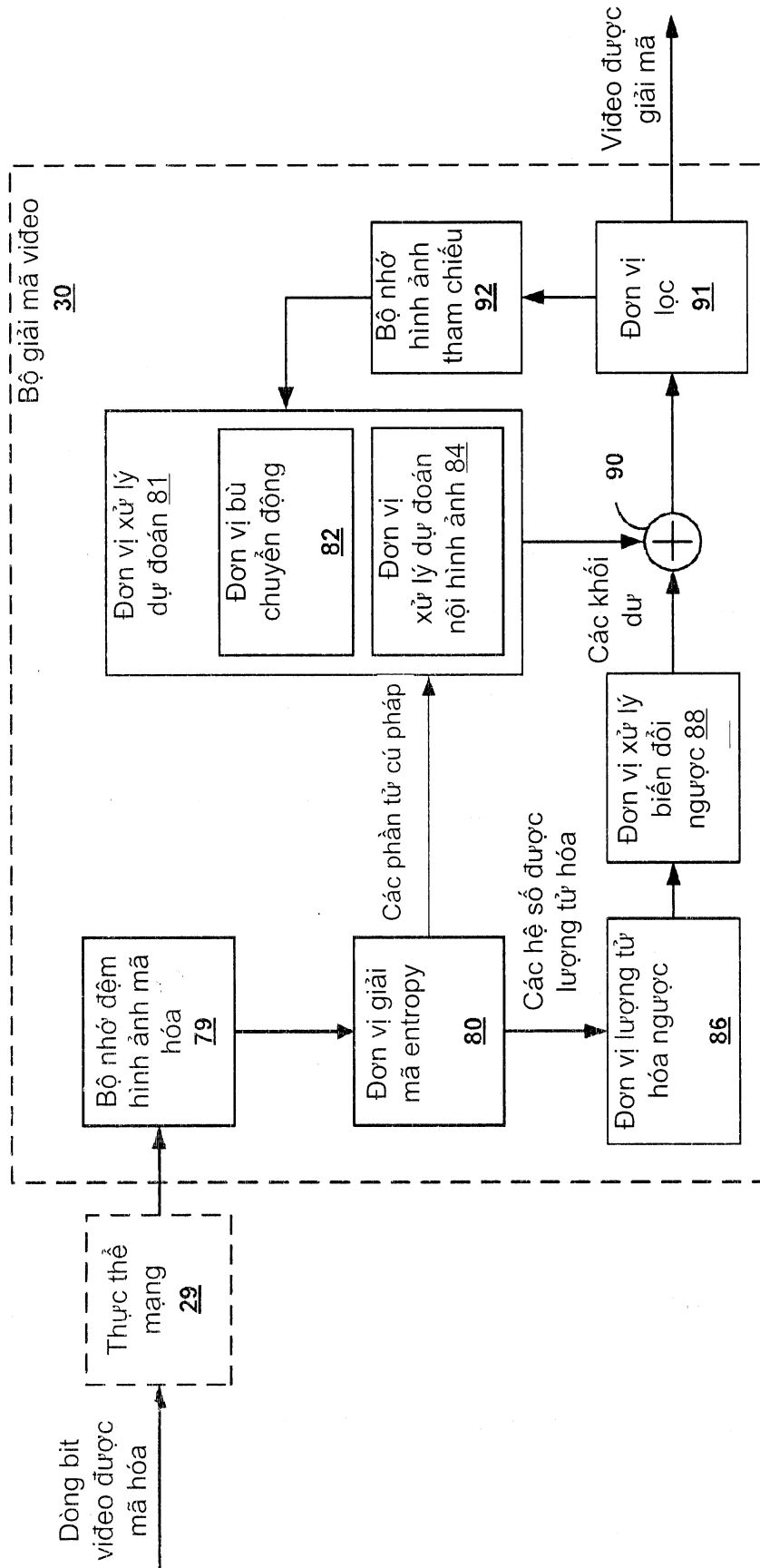


FIG. 3

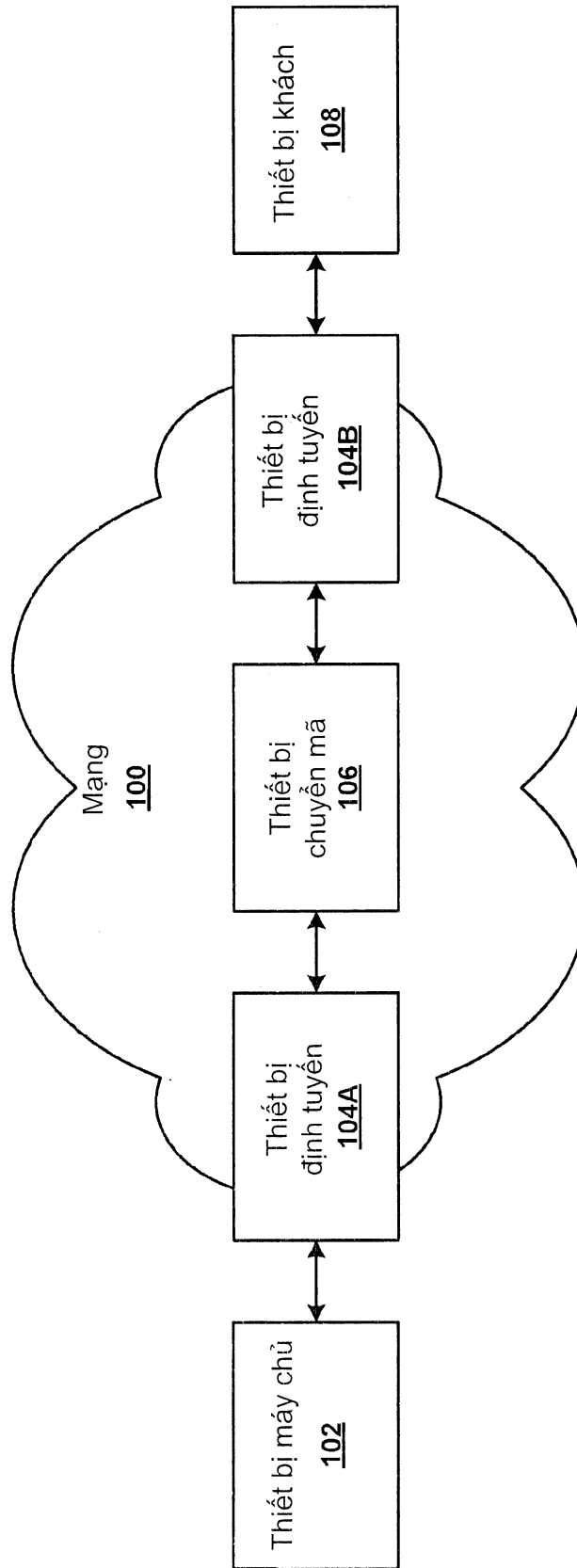


FIG. 4

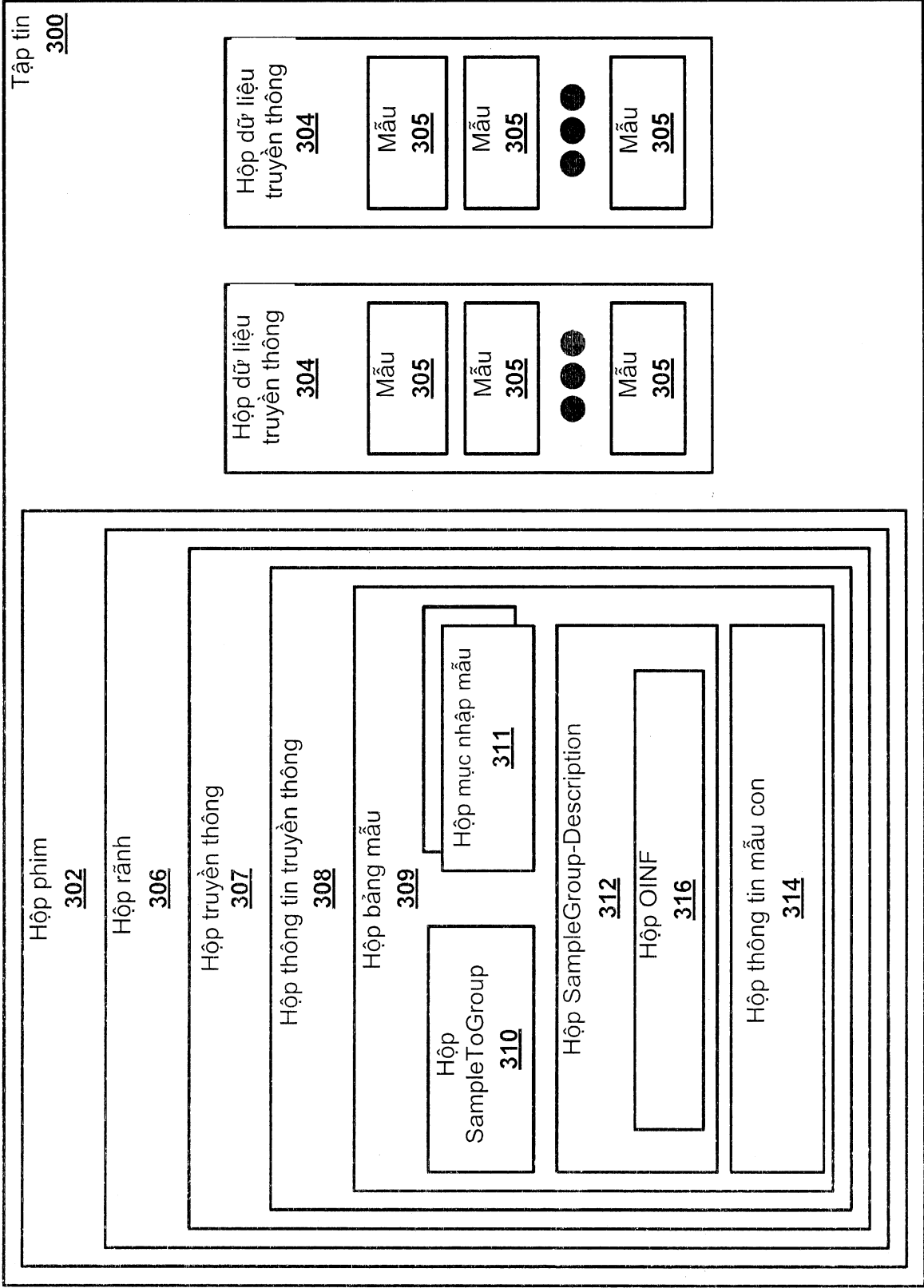


FIG. 5A

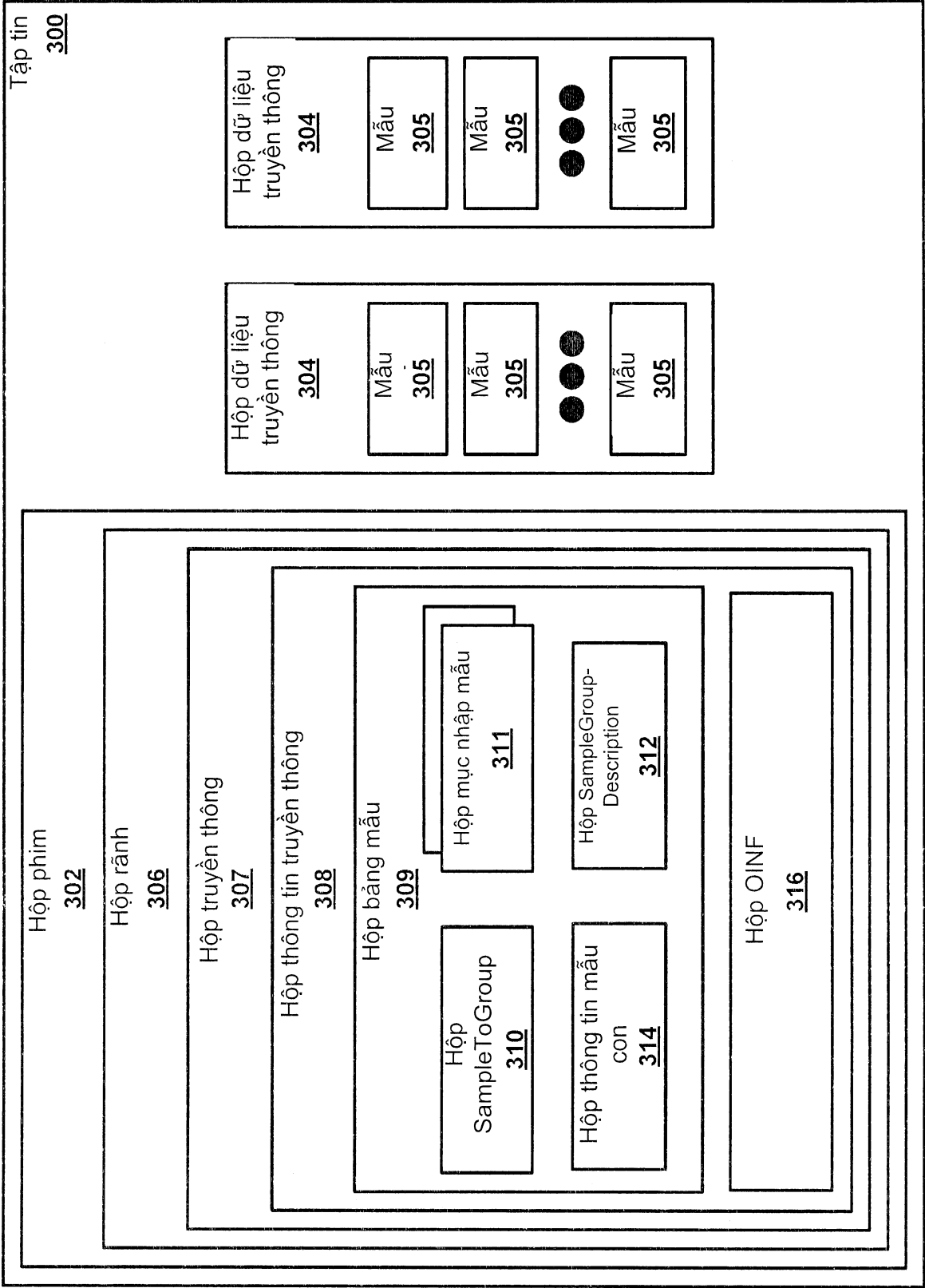
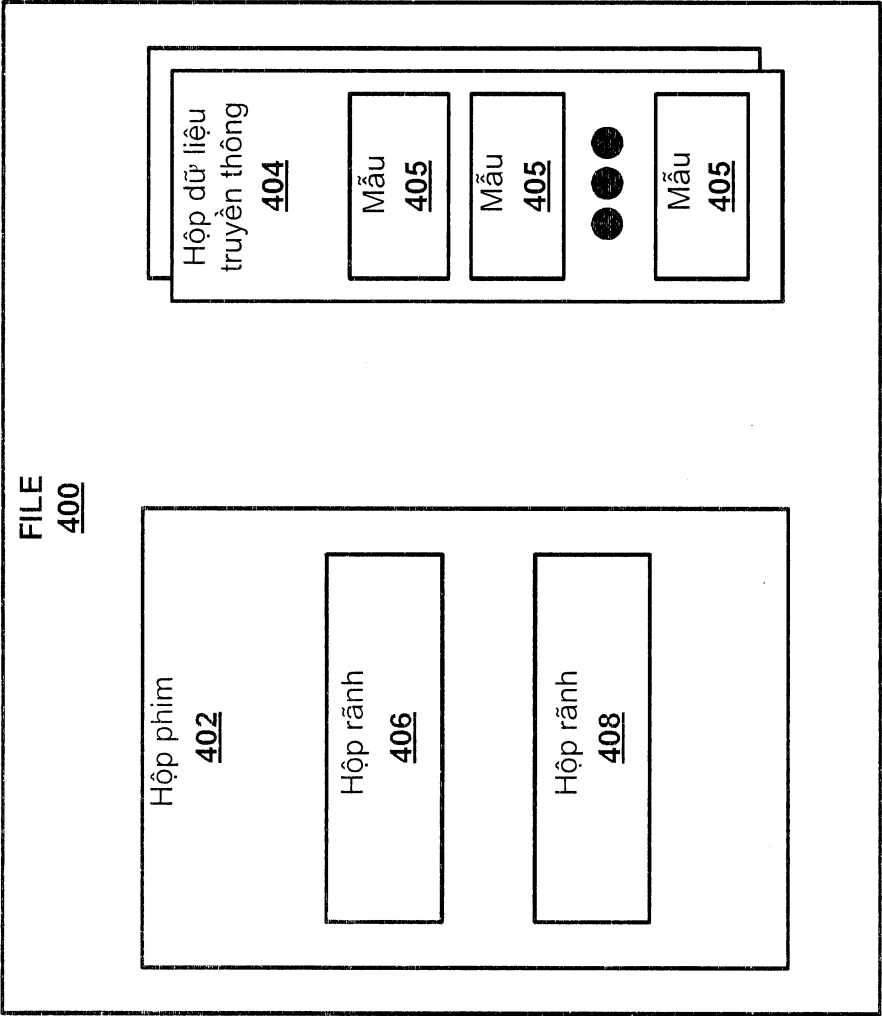


FIG. 5B



Mẫu

FIG. 6

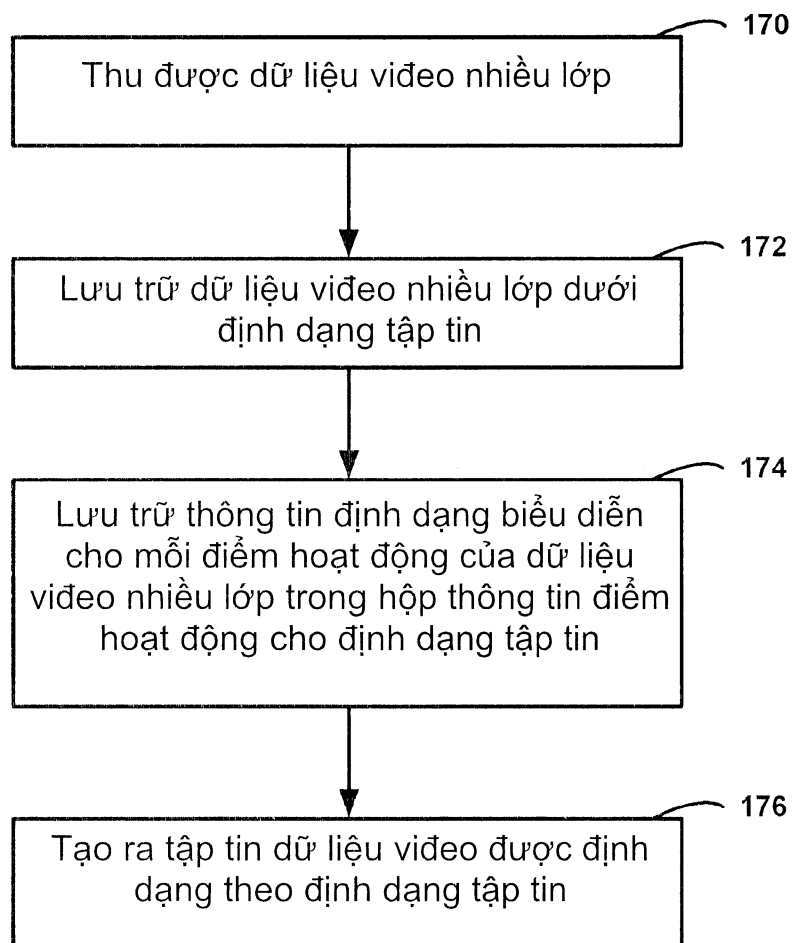


FIG. 7

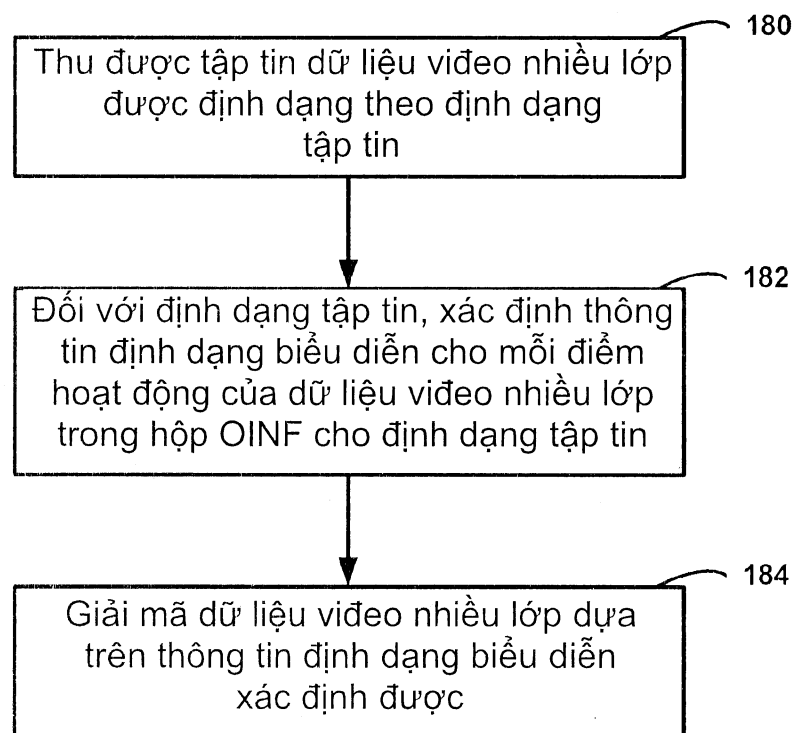


FIG. 8