



(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)



1-0049905

(51)^{2021.01} H04N 19/31; H04N 19/136

(13) B

(21) 1-2022-06226

(22) 27/02/2021

(86) PCT/CN2021/078313 27/02/2021

(87) WO2021/170124 02/09/2021

(30) PCT/EP2020/055269 28/02/2020 EP; PCT/EP2020/065999 09/06/2020 EP;
PCT/EP2020/065989 09/06/2020 EP

(45) 25/08/2025 449

(43) 25/11/2022 416A

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

Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong
518129, China

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

(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP GIẢI MÃ DÒNG BIT VIDEO, PHƯƠNG PHÁP VÀ THIẾT BỊ
MÃ HÓA DÒNG BIT VIDEO, THIẾT BỊ GIẢI MÃ DÒNG BIT VIDEO ĐƯỢC
TẠO MÃ, BỘ GIẢI MÃ, BỘ MÃ HÓA, PHƯƠNG TIỆN ĐỌC ĐƯỢC BẰNG
MÁY TÍNH BẤT BIẾN, VÀ PHƯƠNG TIỆN LƯU TRỮ BẤT BIẾN

(21) 1-2022-06226

(57) Sáng chế đề cập đến ộ mã hóa, bộ giải mã, các phương pháp và các thiết bị để mã hóa và giải mã dòng bit video, và phương tiện lưu trữ bất biến.

Việc báo hiệu các phần tử cú pháp trong tập thông số chuỗi của dòng bit video được khắc phục. Cụ thể là, sáng chế đề xuất phương pháp giải mã dòng bit video trong đó tập thông số chuỗi (Sequence Parameter Set - SPS) được tạo mã mà chứa các phần tử cú pháp mà áp dụng cho chuỗi video, phương pháp này bao gồm các bước thu được giá trị của thành phần cú pháp thứ nhất từ SPS được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS hay không và thu được giá trị của thành phần cú pháp thứ hai từ SPS, ít nhất khi xác định rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

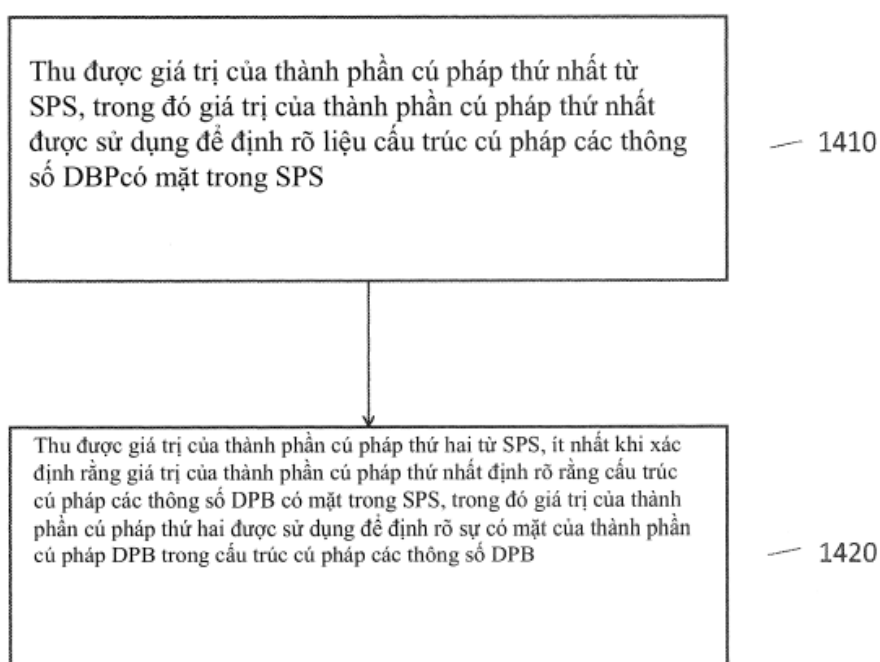


FIG.14

Lĩnh vực kỹ thuật được đề cập

Các phương án thực hiện của sáng chế nói chung liên quan đến lĩnh vực xử lý ảnh và cụ thể hơn là, đề cập đến việc báo hiệu các phần tử cú pháp trong tập thông số chuỗi.

Tình trạng kỹ thuật của sáng chế

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

Lượng dữ liệu video cần để mô tả thậm chí video tương đối ngắn có thể là quan trọng, mà có thể dẫn đến những khó khăn khi dữ liệu cần được tạo luồng hoặc nếu không thì được truyền thông qua mạng truyền thông với dung lượng dải thông hạn chế. Vì vậy, dữ liệu video thường được nén trước khi được truyền thông qua các mạng viễn thông đương thời. Kích thước của video cũng có thể là vấn đề khi video được lưu trữ trên thiết bị lưu trữ do các tài nguyên bộ nhớ có thể bị giới hạn. Các thiết bị nén video thường sử dụng phần mềm và/hoặc phần cứng ở nguồn để lập mã dữ liệu video trước khi truyền hoặc lưu trữ, nhờ đó làm giảm lượng dữ liệu cần để đại diện cho các ảnh video số. Dữ liệu đã nén sau đó được nhận tại điểm đến bởi thiết bị giải nén video để giải mã dữ liệu video. Với các tài nguyên mạng hạn chế và các nhu cầu luôn tăng lên về chất lượng video cao hơn, các kỹ thuật nén và giải nén cải tiến để nâng cao tỷ lệ

nén với việc đánh đổi một ít cho đến không phải đánh đổi về chất lượng hình ảnh luôn được mong đợi.

Cụ thể là, việc báo hiệu các phần tử cú pháp trong tập thông số chuỗi được tạo mã trong dòng bit mà được sử dụng để cung cấp thông tin bộ đệm ảnh đã được giải mã vốn không hiệu quả và thậm chí không thống nhất trong lĩnh vực kỹ thuật (xem phần mô tả chi tiết bên dưới). Vì vậy, mục đích của sáng chế là đề xuất các công nghệ về việc báo hiệu các phần tử cú pháp này với hiệu suất tạo mã được cải thiện.

Bản chất kỹ thuật của sáng chế

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

Các mục đích trên đây và các mục đích khác sẽ đạt được nhờ đối tượng của các điểm yêu cầu bảo hộ độc lập. Các hình thức thực hiện thêm nữa sẽ rõ ràng từ các điểm yêu cầu bảo hộ phụ thuộc, phần mô tả và các hình vẽ.

Theo khía cạnh thứ nhất sáng chế đề cập đến phương pháp giải mã dòng bit video được thực hiện bởi thiết bị giải mã, trong đó tập thông số chuỗi, SPS, được tạo mã trong dòng bit video và chứa các phần tử cú pháp mà áp dụng cho chuỗi video. Phương pháp bao gồm thu được giá trị của thành phần cú pháp thứ nhất (ví dụ, cờ) từ SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS hay không. Phương pháp này còn bao gồm bước thu được giá trị của thành phần cú pháp thứ hai (ví dụ, cờ) từ SPS, ít nhất khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

Ví dụ, giá trị của cấu trúc cú pháp thứ hai có thể chỉ thu được, khi xác định được rằng giá trị của thành phần cú pháp thứ nhất mà định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS. Ở đây và dưới đây, thành phần cú pháp thứ nhất có thể là `sps_ptl_dpb_hrd_params_present_flag` theo phần mô tả chi tiết bên dưới và thành phần cú pháp thứ hai có thể là `sps_sublayer_dpb_params_flag` theo phần mô tả chi tiết bên dưới. Ở đây và dưới đây, thành phần cú pháp DPB có thể là một trong số `max_dec_pic_buffering_minus1[ i ]`, `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` theo phần mô tả chi tiết bên dưới.

Vì vậy phương pháp giải mã dòng bit video được đề xuất đảm bảo hiệu quả báo hiệu của các phần tử cú pháp DPB. Cụ thể là, thành phần cú pháp thứ hai kiểm soát một cách tin cậy sự có mặt của các phần tử cú pháp DPB trong cấu trúc cú pháp các thông số DPB khi các thành phần này có mặt.

Theo một cách thực hiện, phương pháp này còn bao gồm bước thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai (ví dụ, khi thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB, cụ thể là, chỉ khi thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB) và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB. Tái tạo của chuỗi video vì vậy có thể đạt được dựa trên việc báo hiệu tin cậy và hiệu quả thông tin DPB.

Bước thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai có thể bao gồm:

thu được giá trị của thành phần cú pháp DPB từ cấu trúc cú pháp các thông số DPB, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB hoặc thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số

DPB, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB không có mặt trong cấu trúc cú pháp các thông số DPB.

Nhờ đó, có thể bảo đảm được rằng giá trị xác định của thành phần cú pháp DPB là khả dụng trong tình huống bất kỳ và có thể được sử dụng cho tái cấu trúc chuỗi video và không cần lo lắng về cách xử lý không xác định bất kỳ theo góc độ này.

Vì vậy thu được một cách tin cậy giá trị của thành phần cú pháp DPB có thể được sử dụng cho việc tạo cấu hình DPB, ví dụ, để lưu trữ các ảnh tham chiếu dùng cho xử lý dự đoán liên ảnh. Theo đó, bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tạo cấu hình DPB dựa trên giá trị của thành phần cú pháp DPB và tái cấu trúc chuỗi video nhờ sử dụng DPB.

Theo cách thay thế, bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tái cấu trúc chuỗi video dựa trên xác định rằng DPB đã được sử dụng thỏa mãn yêu cầu được định rõ bởi giá trị của thành phần cú pháp DPB. Do vậy, có thể được kiểm tra liệu DPB đã được cung cấp có thích hợp cho việc tái tạo của chuỗi video hay không.

Theo một cách thực hiện, phương pháp giải mã dòng bit video còn bao gồm thu được giá trị của thành phần cú pháp thứ ba từ SPS, trong đó giá trị của thành phần cú pháp thứ ba được sử dụng để xác định số lớn nhất của các lớp con thời gian mà có mặt trong chuỗi video. Giá trị của thành phần cú pháp thứ ba có thể bằng không, nếu chỉ có một lớp con thời gian. Việc xác định số lớn nhất của các lớp con thời gian mà có mặt trong chuỗi video được tạo điều kiện thuận lợi nhờ việc báo hiệu đơn giản thành phần cú pháp thứ ba mà có thể là ưu điểm đối với hiệu suất tạo mã.

Ở đây và dưới đây thành phần cú pháp thứ ba có thể là `sps_max_sublayers_minus1` theo phần mô tả chi tiết bên dưới.

Bước thu được giá trị của thành phần cú pháp thứ hai từ SPS có thể bao gồm

xác định xem số lớn nhất của các lớp con thời gian trong dòng bit video có lớn hơn 1 hay không dựa trên giá trị của thành phần cú pháp thứ ba, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS và thu được giá trị của thành phần cú pháp thứ hai từ SPS, khi xác định (được xác định) rằng số lớn nhất của các lớp con thời gian lớn hơn một. Ví dụ, giá trị của thành phần cú pháp thứ hai có thể thu được từ SPS, chỉ khi xác định được rằng số lớn nhất của các lớp con thời gian lớn hơn một. Nhờ đó, hiệu suất tạo mã có thể được cải thiện hơn, do trong trường hợp mà số lớn nhất của các lớp thời gian không lớn hơn một (tức là, chỉ có một lớp con thời gian), giá trị của thành phần cú pháp thứ hai có thể hoàn toàn không được đọc (ví dụ, nếu nó trở thành không còn ý nghĩa trong trường hợp này) và đối với các phần tử cú pháp DPB lớp thời gian đơn có thể luôn được báo hiệu trong SPS.

Khía cạnh thứ hai đề xuất phương pháp mã hóa dòng bit video được thực hiện bởi thiết bị mã hóa, trong đó tập thông số chuỗi, SPS, được mã hóa trong dòng bit video và chứa các phần tử cú pháp mà áp dụng cho chuỗi video, mà thể hiện các hiệu quả giống nhau như đề cập đến bên trên. Phương pháp bao gồm các bước:

xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) trong SPS;

mã hóa giá trị của thành phần cú pháp thứ nhất (ví dụ, cờ) vào trong SPS dựa trên bước xác định của sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không;

xác định sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, khi xác định (được xác định), ví dụ, chỉ khi xác định được rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó thành phần cú pháp DPB

được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video; và

mã hóa giá trị của thành phần cú pháp thứ hai (ví dụ, cờ) vào trong SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

Theo một cách thực hiện, phương pháp mã hóa còn bao gồm xác định giá trị của thành phần cú pháp DPB khi xác định (được xác định) rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB; và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

Theo một cách thực hiện, phương pháp mã hóa còn bao gồm thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số DPB và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

Bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tạo cấu hình DPB để thỏa mãn giá trị của thành phần cú pháp DPB và tái cấu trúc chuỗi video nhờ sử dụng DPB.

Theo một cách thực hiện, sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB được xác định, khi xác định được (ví dụ, chỉ khi xác định được) rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS và số lớn nhất của các lớp con thời gian trong dòng bit video lớn hơn một.

Ngoài ra, thiết bị để giải mã và thiết bị để tạo mã dòng bit video được tạo mà, một cách tương ứng, thể hiện các hiệu quả giống với các phương pháp được mô tả trên đây.

Khía cạnh thứ ba đề xuất thiết bị để giải mã dòng bit video (được tạo mã), trong đó thiết bị bao gồm:

cụm thu được được cấu hình để thu được giá trị của thành phần cú pháp thứ nhất (ví dụ, cờ) từ SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS được tạo mã trong dòng bit video hay không; và

cụm xác định được tạo cấu hình để xác định xem giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS; và trong đó

cụm thu được còn được cấu hình để thu được giá trị của thành phần cú pháp thứ hai (ví dụ, cờ) từ SPS, ít nhất khi xác định (được xác định), ví dụ, chỉ khi xác định được rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

Ví dụ, giá trị của cấu trúc cú pháp thứ hai có thể chỉ thu được bằng cụm thu được, khi xác định được rằng giá trị của thành phần cú pháp thứ nhất mà định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS.

Theo một cách thực hiện, cụm thu được còn được tạo cấu hình để thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

Thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai có thể bao gồm:

thu được giá trị của thành phần cú pháp DPB từ cấu trúc cú pháp các thông số DPB, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB hoặc

thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số DPB, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB không có mặt trong cấu trúc cú pháp các thông số DPB.

Tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tạo cấu hình DPB dựa trên giá trị của thành phần cú pháp DPB; và tái cấu trúc chuỗi video nhờ sử dụng DPB.

Trong đó DPB được sử dụng để lưu trữ các ảnh cho việc xây dựng danh sách ảnh tham chiếu.

Theo cách thay thế, bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tái cấu trúc chuỗi video dựa trên xác định rằng DPB đã được sử dụng thỏa mãn yêu cầu được định rõ bởi giá trị của thành phần cú pháp DPB.

Theo một cách thực hiện, cụm thu được còn được tạo cấu hình để thu được giá trị của thành phần cú pháp thứ ba từ SPS, trong đó giá trị của thành phần cú pháp thứ ba được sử dụng để xác định số lớn nhất của các lớp con thời gian mà có mặt trong chuỗi video.

Bước thu được giá trị của thành phần cú pháp thứ hai từ SPS có thể bao gồm xác định xem số lớn nhất của các lớp con thời gian trong dòng bit video có lớn hơn 1 hay không dựa trên giá trị của thành phần cú pháp thứ ba, khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS và thu được giá trị của thành phần cú pháp thứ

hai từ SPS, khi xác định (được xác định) rằng số lớn nhất của các lớp con thời gian lớn hơn một.

Khía cạnh thứ tư đề xuất thiết bị để mã hóa dòng bit video, trong đó thiết bị bao gồm:

cụm xác định được tạo cấu hình để xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) trong SPS;

cụm mã hóa được tạo cấu hình để mã hóa giá trị của thành phần cú pháp thứ nhất (ví dụ, cờ) vào trong SPS dựa trên bước xác định của sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không; trong đó

cụm xác định còn được cấu hình để xác định sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, khi xác định (được xác định), ví dụ, chỉ khi xác định được rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video; và trong đó

cụm mã hóa còn được cấu hình để mã hóa giá trị của thành phần cú pháp thứ hai (ví dụ, cờ) vào trong SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

Theo một cách thực hiện, cụm xác định còn được tạo cấu hình để xác định giá trị của thành phần cú pháp DPB khi xác định (được xác định) rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB; và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

Theo một cách thực hiện, cụm mã hóa còn được tạo cấu hình để thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số DPB và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

Bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB có thể bao gồm tạo cấu hình DPB để thỏa mãn giá trị của thành phần cú pháp DPB; và tái cấu trúc chuỗi video nhờ sử dụng DPB.

Theo một cách thực hiện, cụm xác định được tạo cấu hình để xác định rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB, khi cụm xác định xác định rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS và số lớn nhất của các lớp con thời gian trong dòng bit video lớn hơn một.

Các phương pháp được mô tả trên đây có thể được thực hiện trong các thiết bị giải mã hoặc mã hóa, một cách tương ứng, theo đó, bộ mã hóa được đề xuất bao gồm mạch xử lý để thực hiện phương pháp mã hóa dòng bit video theo ví dụ bất kỳ trong số các ví dụ được mô tả nêu trên. Ngoài ra, bộ mã hóa được đề xuất bao gồm một hoặc nhiều bộ xử lý và phương tiện lưu trữ đọc được bằng máy tính bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ mã hóa để thực hiện phương pháp mã hóa dòng bit video theo ví dụ bất kỳ trong số các ví dụ được mô tả nêu trên. Tương tự, bộ giải mã được đề xuất bao gồm mạch xử lý để thực hiện phương pháp giải mã dòng bit video theo ví dụ bất kỳ trong số các ví dụ được mô tả nêu trên và bộ giải mã bao gồm một hoặc nhiều bộ xử lý và phương tiện lưu trữ đọc được bằng máy tính bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình, khi được thực thi bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp giải mã dòng bit video theo ví dụ bất kỳ trong số các ví dụ được mô tả nêu trên.

Ngoài ra, sản phẩm chương trình máy tính được đề xuất bao gồm mã chương trình để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ được mô tả nêu trên khi được thực thi trên máy tính hoặc bộ xử lý. Tương tự, có đề xuất phương tiện đọc được bằng máy tính bất biến mang mã chương trình mà, khi được thực thi bởi thiết bị điện toán, khiến cho thiết bị điện toán thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ được mô tả trên đây.

Ngoài ra, phương tiện lưu trữ bất biến mà được đề xuất bao gồm dòng bit đã được mã hóa, dòng bit được tạo ra bằng cách chia ảnh hiện tại của tín hiệu video hoặc tín hiệu hình ảnh vào trong các khối, và bao gồm các phần tử cú pháp, trong đó các phần tử cú pháp bao gồm thành phần cú pháp thứ nhất trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS hay không; trong trường hợp mà giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, dòng bit còn bao gồm thành phần cú pháp thứ hai trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

Các nội dung chi tiết của một hoặc nhiều phương án được nêu trong các hình vẽ kèm theo và phần mô tả dưới đây. Các dấu hiệu, các mục đích, và các ưu điểm khác sẽ rõ ràng từ phần mô tả, hình vẽ, và yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Dưới đây, các phương án của sáng chế sẽ được mô tả chi tiết hơn dựa vào các hình và các hình vẽ kèm theo, trong đó:

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

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

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

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

FIG.4 là sơ đồ khối thể hiện một ví dụ về máy mã hóa hoặc máy giải mã;

FIG.5 là sơ đồ khối thể hiện một ví dụ khác về máy mã hóa hoặc máy giải mã;

FIG.6 là một ví dụ về thứ tự khung quét;

FIG.7 là một ví dụ về các ảnh xếp lớp, các lát cắt và các ảnh con;

FIG.8 là sơ đồ khối thể hiện một cấu trúc ví dụ của hệ thống cấp nội dung 3100 mà thực hiện dịch vụ phân phối nội dung;

FIG.9 là sơ đồ khối thể hiện cấu trúc của thiết bị đầu cuối làm ví dụ;

FIG.10 thể hiện một ví dụ về sự phân chia ảnh;

FIG.11 thể hiện ví dụ về các lớp và các lớp con của việc tạo mã video khả biến;

FIG.12 thể hiện một ví dụ khác về sự phân chia ảnh;

FIG.13 thể hiện một ví dụ khác về sự phân chia ảnh;

FIG.14 minh họa phương pháp giải mã dòng bit video theo một phương án thực hiện;

FIG.15 minh họa phương pháp mã hóa dòng bit video theo một phương án thực hiện;

FIG.16 minh họa phương pháp giải mã dòng bit video theo một phương án thực hiện;

FIG.17 minh họa phương pháp mã hóa dòng bit video theo một phương án thực hiện.

Trong phần sau đây, các ký hiệu chỉ dẫn giống hệt tham chiếu đến các dấu hiệu giống hệt hoặc ít nhất các dấu hiệu tương đương về mặt chức năng nếu không được quy định ngược lại rõ ràng.

Mô tả chi tiết sáng chế

Trong phần mô tả sau đây, việc tham khảo được thực hiện đến các hình vẽ kèm theo, mà tạo thành một phần của bản mô tả, và thể hiện, bằng cách minh họa, các khía cạnh cụ thể của các phương án của sáng chế hoặc các khía cạnh cụ thể trong đó các phương án của sáng chế có thể được sử dụng. Cần hiểu rằng các phương án của sáng chế có thể được sử dụng theo các khía cạnh khác và bao gồm các sự thay đổi về cấu trúc hoặc logic không được thể hiện trên các hình vẽ. Do đó, phần mô tả chi tiết dưới đây không được lấy làm giới hạn, và phạm vi của sáng chế được xác định bởi yêu cầu bảo hộ kèm theo.

Ví dụ, cần hiểu rằng sáng chế có liên quan đến phương pháp được mô tả cũng có thể vẫn là đúng đối với thiết bị hoặc hệ thống tương ứng được tạo cấu hình để thực hiện phương pháp này và ngược lại. Ví dụ, nếu một hoặc nhiều bước phương pháp cụ thể được mô tả, thiết bị tương ứng có thể bao gồm một hoặc nhiều bộ phận, ví dụ các bộ phận chức năng, để thực hiện một hoặc nhiều bước của phương pháp được mô tả (ví dụ một bộ phận thực hiện một hoặc nhiều bước, hoặc nhiều bộ phận, mỗi bộ phận thực hiện một hoặc nhiều hơn một bước trong số các bước đó), ngay cả khi một hoặc nhiều bộ phận này không được mô tả hoặc được minh họa rõ ràng trên các hình vẽ. Mặt khác, ví dụ, nếu thiết bị cụ thể được mô tả dựa trên một hoặc nhiều bộ phận, ví dụ các bộ phận chức năng, phương pháp tương ứng có thể bao gồm một bước để thực hiện chức năng của một hoặc nhiều bộ phận (ví dụ một bước thực hiện chức năng của một hoặc nhiều bộ phận, hoặc nhiều bước, mỗi bước thực hiện chức năng của một hoặc nhiều hơn một bộ phận trong số các bộ phận này), ngay cả khi một hoặc nhiều

bước này không được mô tả hoặc được minh họa rõ ràng trên các hình vẽ. Hơn nữa, cần hiểu rằng các dấu hiệu của các phương án và/hoặc các khía cạnh làm ví dụ khác nhau được mô tả ở đây có thể được kết hợp với nhau, trừ khi được lưu ý cụ thể là ngược lại.

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

Trong trường hợp tạo mã video không tổn hao, các hình ảnh video gốc có thể được tái tạo, tức là các hình ảnh video được tái tạo có chất lượng giống như các hình ảnh video gốc (giả sử không có tổn hao truyền dẫn hoặc tổn hao dữ liệu khác trong suốt quá trình lưu trữ hoặc truyền). Trong trường hợp tạo mã video tổn hao, việc nén thêm, ví dụ bằng cách lượng tử hóa, được thực hiện, để làm giảm lượng dữ liệu đại diện cho các hình ảnh video, mà không thể được tái tạo hoàn toàn ở bộ giải mã, tức là chất lượng của các hình ảnh video được tái tạo thấp hơn hoặc kém hơn so với chất lượng của các hình ảnh video gốc.

Một số tiêu chuẩn tạo mã video thuộc về nhóm của “các codec video lai tổn hao” (“lossy hybrid video codecs”) (tức là kết hợp dự đoán không gian và thời gian

trong miền mẫu và tạo mã biến đổi 2D để áp dụng lượng tử hóa trong miền biến đổi). Mỗi hình ảnh của chuỗi video thường được phân chia thành tập hợp của các khối không chồng chéo và việc tạo mã thường được thực hiện ở mức khối. Nói cách khác, ở bộ mã hóa, video thường được xử lý, tức là được mã hóa, ở mức khối (khối video), ví dụ bằng cách sử dụng dự đoán không gian (trong hình ảnh) và/hoặc dự đoán thời gian (liên hình ảnh) để tạo khối dự đoán, trừ khối dự đoán từ khối hiện hành (khối hiện được xử lý/cần được xử lý) để thu nhận khối dư, biến đổi khối dư và lượng tử hóa khối dư trong miền biến đổi để làm giảm lượng dữ liệu cần được truyền (nén), trong khi đó ở bộ giải mã, việc xử lý ngược so với bộ mã hóa được áp dụng vào khối được mã hóa hoặc được nén để tái tạo khối hiện hành để biểu diễn. Hơn nữa, bộ mã hóa gấp đôi vòng xử lý của bộ giải mã sao cho cả hai sẽ tạo các sự dự đoán giống hệt nhau (ví dụ các dự đoán trong ảnh và liên ảnh) và/hoặc các sự tái tạo để xử lý, tức là tạo mã, các khối tiếp theo.

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

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

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

Thiết bị nguồn 12 bao gồm bộ mã hóa 20, và ngoài ra có thể, tức là tùy chọn là, bao gồm nguồn hình ảnh 16, bộ tiền xử lý (hoặc bộ phận tiền xử lý) 18, ví dụ bộ tiền xử lý hình ảnh 18, và giao diện truyền thông hoặc bộ phận truyền thông 22.

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

Ngoài bộ tiền xử lý 18 và việc xử lý được thực hiện bởi bộ phận tiền xử lý 18, hình ảnh hoặc dữ liệu hình ảnh 17 cũng có thể được gọi là hình ảnh thô hoặc dữ liệu hình ảnh thô 17.

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

Bộ mã hóa video 20 được tạo cấu hình để nhận dữ liệu hình ảnh được tiền xử lý 19 và cung cấp dữ liệu hình ảnh được mã hóa 21 (các nội dung chi tiết thêm nữa sẽ được mô tả dưới đây, ví dụ, dựa trên Fig.2).

Giao diện truyền thông 22 của thiết bị nguồn 12 có thể được tạo cấu hình để nhận dữ liệu hình ảnh được mã hóa 21 và để truyền dữ liệu hình ảnh được mã hóa 21 (hoặc phiên bản được xử lý thêm nữa bất kỳ của nó) qua kênh truyền thông 13 đến

thiết bị khác, ví dụ thiết bị đích 14 hoặc thiết bị khác bất kỳ, để lưu trữ hoặc tái tạo trực tiếp.

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

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

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

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

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

Cả giao diện truyền thông 22 lẫn giao diện truyền thông 28 có thể được tạo cấu hình như là các giao diện truyền thông đơn hướng như được chỉ báo bởi mũi tên đối

với kênh truyền thông 13 trên Fig.1A trở từ thiết bị nguồn 12 đến thiết bị đích 14, hoặc các giao diện truyền thông hai chiều, và có thể được tạo cấu hình, ví dụ để gửi và nhận các tin nhắn, ví dụ để thiết lập kết nối, để báo nhận và trao đổi thông tin khác bất kỳ liên quan đến liên kết truyền thông và/hoặc hoạt động truyền dữ liệu, ví dụ truyền dữ liệu hình ảnh được mã hóa.

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

Bộ hậu xử lý 32 của thiết bị đích 14 được tạo cấu hình để hậu xử lý dữ liệu hình ảnh được giải mã 31 (cũng được gọi là dữ liệu hình ảnh được tái tạo), ví dụ hình ảnh được giải mã 31, để thu nhận dữ liệu hình ảnh được hậu xử lý 33, ví dụ hình ảnh được hậu xử lý 33. Việc xử lý sau được thực hiện bởi cụm xử lý sau 32 có thể bao gồm, ví dụ bước chuyển đổi định dạng màu sắc (ví dụ từ YCbCr sang RGB), hiệu chỉnh màu sắc, cắt xén, hoặc lấy mẫu lại, hoặc xử lý khác bất kỳ, ví dụ để chuẩn bị dữ liệu ảnh được giải mã 31 để hiển thị, ví dụ bởi thiết bị hiển thị 34.

Thiết bị hiển thị 34 của thiết bị đích 14 được tạo cấu hình để nhận dữ liệu hình ảnh được hậu xử lý 33 để hiển thị hình ảnh, ví dụ cho người dùng hoặc người xem. Thiết bị hiển thị 34 có thể là hoặc bao gồm loại bộ hiển thị bất kỳ để biểu diễn ảnh được tái cấu trúc, ví dụ màn hiển thị hoặc bộ hiển thị tích hợp hoặc bên ngoài. Các màn hiển thị có thể, ví dụ bao gồm các màn hiển thị tinh thể lỏng (LCD: liquid crystal display), các điốt phát quang hữu cơ (OLED: organic light emitting diode), các màn hiển thị plasma, các máy chiếu, các màn hiển thị micro LED, tinh thể lỏng trên silic (LCoS: digital light processor), bộ xử lý ánh sáng kỹ thuật số (DLP: digital light processor) hoặc loại màn hiển thị khác bất kỳ.

Mặc dù Fig.1A thể hiện thiết bị nguồn 12 và thiết bị đích 14 là các thiết bị riêng biệt, tuy nhiên các phương án của các thiết bị cũng có thể bao gồm cả hai thiết bị này

hoặc cả các chức năng, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng. Trong các phương án này, thiết bị nguồn 12 hoặc chức năng tương ứng và thiết bị đích 14 hoặc chức năng tương ứng có thể được thực thi nhờ sử dụng cùng một phần cứng và/hoặc phần mềm hoặc bởi phần cứng và/hoặc phần mềm riêng rẽ hoặc sự kết hợp bất kỳ của chúng.

Người có hiểu biết trung bình trong lĩnh vực tương ứng sẽ hiểu rõ dựa trên phần mô tả rằng sự có mặt và chia tách (chính xác) của các chức năng của các bộ phận khác nhau hoặc các chức năng trong thiết bị nguồn 12 và/hoặc thiết bị đích 14 như được thể hiện trên Fig.1A có thể thay đổi phụ thuộc vào thiết bị và ứng dụng thực tế.

Bộ mã hóa 20 (ví dụ bộ mã hóa video 20) hoặc bộ giải mã 30 (ví dụ bộ giải mã video 30) hoặc cả hai bộ mã hóa 20 và bộ giải mã 30 có thể được thực hiện thông qua mạch xử lý như được thể hiện trên FIG.1B, như một hoặc nhiều bộ vi xử lý, các bộ xử lý tín hiệu số (DSP: digital signal processor), các mạch tích hợp ứng dụng cụ thể (ASIC: application-specific integrated circuit), các mảng cổng khả lập trình dạng trường (FPGA: field-programmable gate array), logic rời rạc, phần cứng, hoạt động tạo mã video chuyên dụng hoặc các kết hợp bất kỳ của chúng. Bộ mã hóa 20 có thể được thực hiện thông qua mạch xử lý 46 để thực hiện các môđun khác nhau như được mô tả đối với bộ mã hóa 20 trên FIG.2 và/hoặc hệ thống hoặc hệ thống con bộ mã hóa bất kỳ khác đã được mô tả ở đây. Bộ giải mã 30 có thể được thực hiện thông qua mạch xử lý 46 để thực hiện các môđun khác nhau như được mô tả đối với bộ giải mã 30 trên FIG.3 và/hoặc hệ thống hoặc hệ thống con bộ giải mã bất kỳ khác đã được mô tả ở đây. Mạch xử lý có thể được tạo cấu hình để thực hiện các hoạt động khác nhau như được thảo luận sau đây. Như được thể hiện trên Fig.5, nếu các kỹ thuật được thực hiện một phần trong phần mềm, thì thiết bị có thể lưu trữ các lệnh đối với phần mềm trong phương tiện lưu trữ lâu dài đọc được bằng máy tính thích hợp và có thể thực hiện các lệnh trong phần cứng nhờ sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của

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

Thiết bị nguồn 12 và thiết bị đích 14 có thể bao gồm thiết bị bất kỳ trong phạm vi thiết bị rộng, bao gồm thiết bị bất kỳ trong số các thiết bị cầm tay hoặc cố định, ví dụ các máy tính xách tay loại notebook hoặc máy tính xách tay loại laptop, các điện thoại di động, các điện thoại thông minh, các máy tính bảng hoặc các máy tính dạng bảng, các camera, các máy tính để bàn, các thiết bị giải mã tín hiệu và truyền dữ liệu (set-top box), các máy thu hình, các thiết bị hiển thị, các máy phát phương tiện số, các bàn giao tiếp trò chơi video, các thiết bị tạo luồng video (như các máy chủ dịch vụ nội dung hoặc các máy chủ truyền nội dung), thiết bị nhận phát rộng, thiết bị truyền phát rộng, hoặc tương tự và có thể không sử dụng hệ điều hành hoặc sử dụng loại hệ điều hành bất kỳ. Trong một số trường hợp, thiết bị nguồn 12 và thiết bị đích 14 có thể được trang bị để truyền thông vô tuyến. Vì vậy, thiết bị nguồn 12 và thiết bị đích 14 có thể là các thiết bị truyền thông vô tuyến.

Trong một số trường hợp, hệ thống tạo mã video 10 được minh họa trên FIG.1A chỉ là một ví dụ và các công nghệ của sáng chế có thể áp dụng cho các thiết lập tạo mã video (ví dụ, mã hóa video hoặc giải mã video) mà không nhất thiết bao gồm sự truyền thông dữ liệu bất kỳ giữa các thiết bị mã hóa và giải mã. Theo các ví dụ khác, thì dữ liệu được truy hồi từ bộ nhớ cục bộ, được tạo luồng qua mạng, v.v.. Thiết bị mã hóa video có thể mã hóa và lưu trữ dữ liệu vào bộ nhớ, và/hoặc thiết bị giải mã video có thể lấy và giải mã dữ liệu từ bộ nhớ. Trong một số ví dụ, việc mã hóa và giải mã được thực hiện bởi các thiết bị mà không truyền thông với một thiết bị khác, mà chỉ đơn giản là mã hóa dữ liệu vào bộ nhớ và/hoặc lấy và giải mã dữ liệu từ bộ nhớ.

Để thuận tiện cho việc mô tả, các phương án của sáng chế được mô tả ở đây, ví dụ, bằng cách tham chiếu đến tạo mã video hiệu suất cao (High-Efficiency Video Coding, HEVC) hoặc đến phần mềm tham chiếu của tạo mã video đa năng (Versatile Video coding, VVC), tiêu chuẩn tạo mã video thế hệ tiếp theo được phát triển bởi nhóm hợp tác chung về tạo mã video (Joint Collaboration Team on Video Coding, JCT-VC) của nhóm chuyên gia tạo mã video (Video Coding Experts Group, VCEG) ITU-T và nhóm chuyên gia hình ảnh động (Motion Picture Experts Group, MPEG) ISO/IEC. Người có hiểu biết trung bình trong lĩnh vực tương ứng sẽ hiểu rằng các phương án của sáng chế không bị giới hạn ở HEVC hoặc VVC.

Bộ mã hóa và phương pháp mã hóa

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

Bộ phận tính toán dư 204, bộ phận xử lý biến đổi 206, bộ phận lượng tử hóa 208, bộ phận chọn chế độ 260 có thể được coi là tạo thành đường tín hiệu xuôi của bộ mã hóa 20, trong khi đó bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái tạo 214, bộ đệm 216, bộ lọc vòng 220, bộ đệm hình ảnh được

giải mã (decoded picture buffer, DPB) 230, bộ phận dự đoán liên ảnh 244 và bộ phận dự đoán trong ảnh 254 có thể được coi là tạo thành đường tín hiệu ngược của bộ mã hóa video 20, trong đó đường tín hiệu ngược của bộ mã hóa video 20 tương ứng với đường tín hiệu của bộ giải mã (xem bộ giải mã video 30 trên Fig.3). Bộ phận lượng tử hóa ngược 210, bộ phận xử lý biến đổi ngược 212, bộ phận tái tạo 214, bộ lọc vòng 220, bộ đệm hình ảnh được giải mã (DPB) 230, bộ phận dự đoán liên ảnh 244 và bộ phận dự đoán trong ảnh 254 cũng được coi là tạo thành “bộ giải mã lắp sẵn” của bộ mã hóa video 20.

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

Bộ mã hóa 20 có thể được tạo cấu hình để nhận, ví dụ qua đầu vào 201, hình ảnh 17 (hoặc dữ liệu hình ảnh 17), ví dụ hình ảnh của chuỗi các hình ảnh tạo thành video hoặc chuỗi video. Hình ảnh hoặc dữ liệu hình ảnh nhận được cũng có thể là hình ảnh được tiền xử lý 19 (hoặc dữ liệu hình ảnh được tiền xử lý 19). Để đơn giản, phần mô tả sau đây liên quan đến hình ảnh 17. Hình ảnh 17 cũng có thể được gọi là hình ảnh hiện hành hoặc hình ảnh cần được tạo mã (cụ thể trong tạo mã video để phân biệt hình ảnh hiện hành với các hình ảnh khác, ví dụ các hình ảnh được mã hóa và/hoặc được giải mã trước đó của cùng chuỗi video, tức là chuỗi video mà cũng bao gồm hình ảnh hiện hành).

Hình ảnh (số) được hoặc có thể được coi là mảng hoặc ma trận hai chiều của các mẫu với các giá trị cường độ. Mẫu trong mảng cũng có thể được gọi là điểm ảnh (pixel) (dạng rút gọn của phần tử hình ảnh) hoặc pel (điểm ảnh). Số lượng mẫu theo chiều (hoặc trục) ngang và thẳng đứng của mảng hoặc hình ảnh xác định kích thước và/hoặc độ phân giải của hình ảnh. Để biểu diễn màu, thông thường ba thành phần màu được sử dụng, tức là hình ảnh có thể được biểu diễn hoặc bao gồm ba mảng mẫu. Theo định dạng hoặc không gian màu RGB, hình ảnh bao gồm mảng mẫu đỏ, lục, lam tương ứng. Tuy nhiên, trong tạo mã video, mỗi điểm ảnh thường được biểu diễn theo

định dạng hoặc không gian màu độ chói và sắc độ, ví dụ YCbCr, mà bao gồm thành phần độ chói được chỉ báo bởi Y (đôi lúc L được sử dụng để thay thế) và hai thành phần sắc độ được chỉ báo bởi Cb và Cr. Thành phần độ chói (hoặc độ chói (luma) ngắn) Y đại diện cho độ sáng hoặc cường độ độ xám (ví dụ như trong hình ảnh thang màu xám), trong khi hai thành phần sắc độ (hoặc sắc độ (chroma) ngắn) Cb và Cr đại diện cho sắc độ hoặc các thành phần thông tin màu. Theo đó, hình ảnh theo định dạng YCbCr bao gồm mảng mẫu độ chói của các giá trị mẫu độ chói (Y), và hai mảng mẫu sắc độ của các giá trị sắc độ (Cb và Cr). Các hình ảnh ở định dạng RGB có thể được chuyển đổi hoặc được biến đổi thành định dạng YCbCr và ngược lại, quy trình này cũng được biết đến là biến đổi hoặc chuyển đổi màu. Nếu hình ảnh là đơn sắc, thì hình ảnh này có thể bao gồm chỉ mảng mẫu độ chói. Theo đó, hình ảnh có thể là, ví dụ, mảng của các mẫu độ chói theo định dạng đơn sắc hoặc mảng của các mẫu độ chói và hai mảng tương ứng của các mẫu sắc độ theo định dạng màu 4:2:0, 4:2:2, và 4:4:4.

Các phương án của bộ mã hóa video 20 có thể bao gồm bộ phận phân chia hình ảnh (không được thể hiện trên Fig.2) được tạo cấu hình để phân chia hình ảnh 17 thành các khối hình ảnh 203 (thường là không chồng chéo). Các khối này cũng có thể được gọi là các khối gốc, các khối macrô (H.264/AVC) hoặc các khối cây tạo mã (CTB) hoặc các đơn vị cây tạo mã (CTU) (H.265/HEVC và VVC). Bộ phận phân chia hình ảnh có thể được tạo cấu hình để sử dụng cùng kích thước khối đối với tất cả các hình ảnh của chuỗi video và lưới tương ứng xác định kích thước khối, hoặc để thay đổi kích thước khối giữa các hình ảnh hoặc các tập con hoặc các nhóm của các hình ảnh, và phân chia mỗi hình ảnh thành các khối tương ứng.

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

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

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

Theo các phương án thì bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân vùng và/hoặc mã hóa ảnh nhờ sử dụng các lát cắt (còn được gọi là các lát cắt video), trong đó ảnh có thể được phân vùng thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều lát cắt (thường là không chồng nhau), và mỗi lát cắt có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU).

Các phương án của bộ mã hóa video 20 như được thể hiện trên Fig.2 có thể còn được tạo cấu hình để phân chia và/hoặc mã hóa hình ảnh bằng cách sử dụng các nhóm ảnh xếp lớp (tile) (cũng được gọi là các nhóm ảnh xếp lớp video) và/hoặc các ảnh xếp lớp (cũng được gọi là các ảnh xếp lớp video), trong đó hình ảnh có thể được phân chia thành hoặc được mã hóa nhờ sử dụng một hoặc nhiều nhóm ảnh xếp lớp (thường là không chồng chéo), và mỗi nhóm ảnh xếp lớp có thể bao gồm, ví dụ một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều ảnh xếp lớp, trong đó mỗi ảnh xếp lớp, ví

dù có thể là hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ các khối hoàn chỉnh hoặc các khối thành phần.

Tính toán dư

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

Biến đổi

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

Bộ phận xử lý biến đổi 206 có thể được tạo cấu hình để áp dụng các phép xấp xỉ nguyên của DCT/DST, như các phép biến đổi được quy định đối với H.265/HEVC. So với biến đổi DCT trực giao, các phép xấp xỉ nguyên này thường được định tỷ lệ bởi hệ số nhất định. Để giữ chuẩn của khối dư mà được xử lý bằng các loại biến đổi thuận và ngược, các bội suất bổ sung được áp dụng như là một phần của quy trình biến đổi. Các bội suất thường được chọn dựa trên các sự ràng buộc nhất định như các bội suất là lũy thừa của hai đối với các thao tác dịch, độ sâu bit của các hệ số biến đổi, sự cân bằng giữa độ chính xác và các chi phí thực hiện, v.v.. Các bội suất cụ thể, ví dụ, được quy định đối với biến đổi ngược, ví dụ bởi bộ phận xử lý biến đổi ngược 212 (và việc biến đổi ngược tương ứng, ví dụ bởi bộ phận xử lý biến đổi ngược 312 ở bộ giải mã video 30) và các bội suất tương ứng đối với biến đổi thuận, ví dụ bởi bộ phận xử lý biến đổi 206, ở bộ mã hóa 20 có thể được quy định theo đó.

Các phương án của bộ mã hóa video 20 (tương ứng là bộ phận xử lý biến đổi 206) có thể được tạo cấu hình để kết xuất các thông số biến đổi, ví dụ loại của biến đổi hoặc các biến đổi, ví dụ trực tiếp hoặc được mã hóa hoặc được nén qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã video 30 có thể nhận và sử dụng các thông số biến đổi để giải mã.

Lượng tử hóa

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

Quy trình lượng tử hóa có thể làm giảm độ sâu bit được liên kết với một số hoặc tất cả các hệ số biến đổi 207. Ví dụ, hệ số biến đổi n bit có thể được làm tròn xuống thành hệ số biến đổi m bit trong suốt quá trình lượng tử hóa, trong đó n lớn hơn m . Mức độ lượng tử hóa có thể được thay đổi bằng cách điều chỉnh thông số lượng tử hóa (QP). Ví dụ, đối với lượng tử hóa vô hướng, việc định tỷ lệ khác nhau có thể được áp dụng để đạt được lượng tử hóa tinh hơn hoặc thô hơn. Các kích thước bước lượng tử hóa nhỏ hơn tương ứng với lượng tử hóa tinh hơn, trong khi đó các kích thước bước lượng tử hóa lớn hơn tương ứng với lượng tử hóa thô hơn. Kích thước bước lượng tử hóa có thể áp dụng được có thể được chỉ báo bởi thông số lượng tử hóa (quantization parameter, QP). Thông số lượng tử hóa có thể, ví dụ, là chỉ số đối với tập hợp được xác định trước của các kích thước bước lượng tử hóa áp dụng được. Ví dụ, các thông số lượng tử hóa nhỏ có thể tương ứng với lượng tử hóa tinh (các kích thước bước lượng tử hóa nhỏ) và các thông số lượng tử hóa lớn có thể tương ứng với lượng tử hóa thô (các kích thước bước lượng tử hóa lớn) hoặc ngược lại. Việc lượng tử hóa có thể bao gồm việc chia bởi kích thước bước lượng tử hóa và tương ứng và/hoặc khử lượng

tử hóa ngược, ví dụ bởi bộ phận lượng tử hóa ngược 210, có thể bao gồm việc nhân với kích thước bước lượng tử hóa. Các phương án theo một số tiêu chuẩn, ví dụ HEVC, có thể được tạo cấu hình để sử dụng thông số lượng tử hóa để xác định kích thước bước lượng tử hóa. Nói chung, kích thước bước lượng tử hóa có thể được tính toán dựa trên thông số lượng tử hóa nhờ sử dụng phép xấp xỉ đầu phẳng tĩnh của phương trình bao gồm phép chia. Các bội suất bổ sung có thể được đưa vào lượng tử hóa và khử lượng tử hóa để phục hồi chuẩn của khối dư, mà có thể được thay đổi do việc định tỷ lệ được sử dụng trong phép xấp xỉ đầu phẳng tĩnh của phương trình đối với kích thước bước lượng tử hóa và thông số lượng tử hóa. Theo một cách thực hiện ví dụ, việc định tỷ lệ của biến đổi ngược và khử lượng tử hóa có thể được kết hợp. Theo cách khác, các bảng lượng tử hóa được tùy biến có thể được sử dụng và được báo hiệu từ bộ mã hóa đến bộ giải mã, ví dụ trong dòng bit. Lượng tử hóa là hoạt động có tổn hao, trong đó tổn hao tăng lên cùng với việc tăng các kích thước bước lượng tử hóa.

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

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 210 được tạo cấu hình để áp dụng lượng tử hóa ngược của bộ phận lượng tử hóa 208 trên các hệ số được lượng tử hóa để thu nhận các hệ số được khử lượng tử hóa 211, ví dụ bằng cách áp dụng sự đảo ngược của sơ đồ lượng tử hóa được áp dụng bởi bộ phận lượng tử hóa 208 dựa trên hoặc sử dụng kích thước bước lượng tử hóa giống như bộ phận lượng tử hóa 208. Các hệ số được khử lượng tử hóa 211 cũng có thể được gọi là các hệ số dư được khử lượng tử hóa 211 và tương ứng - mặc dù thường không giống hệt các hệ số biến đổi do tổn hao bởi lượng tử hóa - với các hệ số biến đổi 207.

Biến đổi ngược

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

Tái tạo

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

Lọc

Bộ phận lọc vòng 220 (hoặc “bộ lọc vòng” ngấn 220), được tạo cấu hình để lọc khối được tái tạo 215 để thu nhận khối được lọc 221, hoặc nói chung, để lọc các mẫu tái tạo để thu nhận các mẫu được lọc. Bộ phận lọc vòng, ví dụ, được tạo cấu hình để làm phẳng các sự chuyển tiếp điểm ảnh, hoặc theo cách khác là nâng cao chất lượng video. Cụm bộ lọc vòng lặp 220 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp như bộ lọc giải khối, bộ lọc độ lệch phù hợp mẫu (SAO: sample-adaptive offset) hoặc một hoặc nhiều bộ lọc khác, ví dụ bộ lọc hai chiều, bộ lọc vòng lặp thích ứng (ALF), các bộ lọc làm sắc nét, làm mịn hoặc các bộ lọc cộng tác, hoặc sự kết hợp bất kỳ của chúng. Mặc dù cụm bộ lọc vòng lặp 220 được thể hiện trên FIG.2 là một bộ lọc trong vòng lặp, nhưng trong các kết cấu khác, cụm bộ lọc vòng lặp 220 có thể được triển khai dưới dạng bộ lọc sau vòng lặp. Khối được lọc 221 cũng có thể được đề cập đến khối được tái cấu trúc được lọc 221.

Các phương án của bộ mã hóa video 20 (tương ứng là bộ phận lọc vòng 220) có thể được tạo cấu hình để kết xuất các thông số lọc vòng (như thông tin độ lệch tương thích mẫu), ví dụ trực tiếp hoặc được mã hóa qua bộ phận mã hóa entropi 270, sao cho, ví dụ, bộ giải mã 30 có thể nhận và áp dụng cùng các thông số lọc vòng hoặc các bộ lọc vòng tương ứng để giải mã.

Bộ đệm hình ảnh được giải mã

Bộ đệm hình ảnh được giải mã (DPB) 230 có thể là bộ nhớ để lưu trữ các hình ảnh tham chiếu, hoặc nói chung là dữ liệu hình ảnh tham chiếu, để mã hóa dữ liệu video bằng bộ mã hóa video 20. DPB 230 có thể được tạo ra bởi loại bất kỳ trong số rất nhiều loại 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 điện trở (magnetoresistive RAM, MRAM), RAM điện trở (resistive RAM, RRAM), hoặc các loại thiết bị nhớ khác. Bộ đệm hình ảnh được giải mã (DPB) 230 có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được lọc 221. Bộ đệm hình ảnh được giải mã 230 có thể còn được tạo cấu hình để lưu trữ các khối được lọc trước đó khác, ví dụ các khối được tái tạo và được lọc trước đó 221, của cùng hình ảnh hiện hành hoặc của các hình ảnh khác, ví dụ các hình ảnh được tái tạo trước đó, và có thể cung cấp các hình ảnh được tái tạo, tức là được giải mã, trước đó hoàn chỉnh (và các khối và các mẫu tham chiếu tương ứng) và/hoặc hình ảnh hiện hành được tái tạo một phần (và các khối và các mẫu tham chiếu tương ứng), ví dụ để dự đoán liên ảnh. Bộ đệm hình ảnh được giải mã (DPB) 230 cũng có thể được tạo cấu hình để lưu trữ một hoặc nhiều khối được tái tạo không được lọc 215, hoặc nói chung là các mẫu tái tạo không được lọc, ví dụ nếu khối được tái tạo 215 không được lọc bởi bộ phận lọc vòng 220, hoặc phiên bản được xử lý thêm nữa khác bất kỳ của các khối hoặc các mẫu được tái tạo.

Chọn chế độ (phân chia & dự đoán)

Bộ phận chọn chế độ 260 bao gồm bộ phận phân chia 262, bộ phận dự đoán liên ảnh 244 và bộ phận dự đoán trong ảnh 254, và được tạo cấu hình để nhận hoặc thu nhận dữ liệu hình ảnh gốc, ví dụ khối gốc 203 (khối hiện hành 203 của hình ảnh hiện hành 17), và dữ liệu hình ảnh được tái tạo, ví dụ các mẫu hoặc các khối tái tạo được lọc và/hoặc không được lọc của cùng hình ảnh (hiện hành) và/hoặc từ một hoặc nhiều hình ảnh được giải mã trước đó, ví dụ từ bộ đệm hình ảnh được giải mã 230 hoặc các bộ đệm khác (ví dụ bộ đệm hàng, không được thể hiện trên hình vẽ). Dữ liệu hình ảnh được tái tạo được sử dụng làm dữ liệu hình ảnh tham chiếu để dự đoán, ví dụ dự đoán liên ảnh hoặc dự đoán trong ảnh, để thu nhận khối dự đoán 265 hoặc bộ dự đoán 265.

Bộ phận chọn chế độ 260 có thể được tạo cấu hình để xác định hoặc chọn sự phân chia đối với chế độ dự đoán khối hiện hành (bao gồm không phân chia) và chế độ dự đoán (ví dụ chế độ dự đoán trong ảnh hoặc liên ảnh) và tạo khối dự đoán tương ứng 265, mà được sử dụng để tính toán khối dư 205 và để tái tạo khối được tái tạo 215.

Các phương án của bộ phận chọn chế độ 260 có thể được tạo cấu hình để chọn sự phân chia và chế độ dự đoán (ví dụ từ sự phân chia và chế độ dự đoán được hỗ trợ bởi hoặc khả dụng đối với bộ phận chọn chế độ 260), mà cung cấp sự phù hợp nhất hoặc nói cách khác lượng dư tối thiểu (lượng dư tối thiểu có nghĩa là việc nén tốt hơn để truyền hoặc lưu trữ), hoặc lượng thông tin báo hiệu thừa tối thiểu (lượng thông tin báo hiệu thừa tối thiểu có nghĩa là việc nén tốt hơn để truyền hoặc lưu trữ), hoặc cân nhắc hoặc cân đối cả hai. Bộ phận chọn chế độ 260 có thể được tạo cấu hình để xác định sự phân chia và chế độ dự đoán dựa trên tối ưu hóa tỷ lệ méo (rate distortion optimization, RDO), tức là chọn chế độ dự đoán mà cung cấp tỷ lệ méo tối thiểu. Các thuật ngữ như “tốt nhất”, “tối thiểu”, “tối ưu” v.v. trong ngữ cảnh này không nhất thiết phải đề cập đến “tốt nhất”, “tối thiểu”, “tối ưu”, v.v. toàn phần mà cũng có thể đề cập đến sự hoàn thành tiêu chuẩn chấm dứt hoặc chọn giống như giá trị vượt quá hoặc nhỏ

hơn ngưỡng hoặc các ràng buộc khác có tiềm năng dẫn đến “việc chọn tối ưu phụ” nhưng làm giảm độ phức tạp và thời gian xử lý.

Nói cách khác, bộ phận phân chia 262 có thể được tạo cấu hình để phân chia khối 203 thành các phần chia khối nhỏ hơn hoặc các khối con (mà lại tạo thành các khối), ví dụ sử dụng lặp lại phân chia cây tứ phân (QT), phân chia nhị phân (BT) hoặc phân chia cây bội ba (TT) hoặc sự kết hợp bất kỳ của chúng, và để thực hiện, ví dụ, dự đoán đối với mỗi trong số các phần chia khối hoặc các khối con, trong đó việc chọn chế độ bao gồm việc chọn các cấu trúc cây của khối được phân chia 203 và các chế độ dự đoán được áp dụng vào mỗi trong số các phần chia khối hoặc các khối con.

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

Phân chia

Bộ phận phân chia 262 có thể phân chia (hoặc chia tách) khối hiện hành 203 thành các phần chia nhỏ hơn, ví dụ các khối nhỏ hơn của kích thước hình vuông hoặc hình chữ nhật. Các khối nhỏ hơn này (mà cũng có thể được gọi là các khối con) có thể còn được phân chia thành thậm chí các phần chia nhỏ hơn. Điều này cũng liên quan đến phân chia cây hoặc phân chia cây phân cấp, trong đó khối gốc, ví dụ ở mức cây gốc 0 (mức phân cấp 0, độ sâu 0), có thể được phân chia theo đệ quy, ví dụ được phân chia thành hai hoặc nhiều hơn hai khối của mức cây thấp hơn, ví dụ các nút ở mức cây 1 (mức phân cấp 1, độ sâu 1), trong đó các khối này có thể được phân chia lại thành hai hoặc nhiều hơn hai khối của mức thấp hơn tiếp theo, ví dụ mức cây 2 (mức phân cấp 2, độ sâu 2), v.v. cho đến khi việc phân chia được chấm dứt, ví dụ do tiêu chuẩn chấm dứt được đáp ứng, ví dụ độ sâu cây tối đa hoặc kích thước khối tối thiểu được đạt đến. Các khối mà không được phân chia thêm nữa cũng được gọi là các khối lá hoặc các nút lá của cây. Cây sử dụng việc phân chia thành hai phần chia được

gọi là cây nhị phân (binary-tree, BT), cây sử dụng việc phân chia thành ba phần chia được gọi là cây tam phân (TT), và cây sử dụng việc phân chia thành bốn phần chia được gọi là cây tứ phân (QT).

Như được nêu trên đây, thuật ngữ “khối” như được sử dụng ở đây có thể là một phần, cụ thể là phần hình vuông hoặc hình chữ nhật, của hình ảnh. Dựa vào, ví dụ, HEVC và VVC, khối có thể là hoặc tương ứng với đơn vị cây tạo mã (CTU), đơn vị tạo mã (CU), đơn vị dự đoán (PU), và đơn vị biến đổi (TU) và/hoặc các khối tương ứng, ví dụ khối cây tạo mã (CTB), khối tạo mã (CB), khối biến đổi (TB) hoặc khối dự đoán (PB).

Ví dụ, đơn vị cây tạo mã (CTU) có thể là hoặc bao gồm CTB của các mẫu độ chói, hai CTB tương ứng của các mẫu sắc độ của hình ảnh mà có ba mảng mẫu, hoặc CTB của các mẫu của hình ảnh đơn sắc hoặc hình ảnh mà được tạo mã nhờ sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Tương ứng, khối cây tạo mã (CTB) có thể là khối $N \times N$ của các mẫu đối với một số giá trị của N sao cho việc chia một thành phần thành các CTB là việc phân chia. Đơn vị tạo mã (CU) có thể là hoặc bao gồm khối tạo mã của các mẫu độ chói, hai khối tạo mã tương ứng của các mẫu sắc độ của hình ảnh có ba mảng mẫu, hoặc khối tạo mã của các mẫu của hình ảnh đơn sắc hoặc hình ảnh mà được tạo mã sử dụng ba mặt phẳng màu riêng rẽ và các cấu trúc cú pháp được sử dụng để tạo mã các mẫu. Tương ứng, khối tạo mã (CB) có thể là khối $M \times N$ của các mẫu đối với một số giá trị của M và N sao cho việc chia CTB thành các khối tạo mã là việc phân chia.

Trong các phương án, ví dụ, theo HEVC, đơn vị cây tạo mã (CTU) có thể được tách thành các CU bằng cách sử dụng cấu trúc cây tứ phân được biểu thị là cây mã hóa. Quyết định liệu có tạo mã khu vực ảnh sử dụng dự đoán liên hình ảnh (thời gian) hoặc trong hình ảnh (không gian) hay không được thực hiện ở mức CU. Mỗi CU có thể còn được chia thành một, hai hoặc bốn PU theo loại chia tách PU. Bên trong một

PU, cùng quy trình dự đoán được áp dụng và thông tin liên quan được truyền đến bộ giải mã trên cơ sở PU. Sau khi thu được khối dư bằng cách áp dụng quy trình dự đoán dựa vào loại phân tách PU, CU có thể được phân chia thành các đơn vị biến đổi (Transform Unit, viết tắt là TU) theo cấu trúc cây tứ phân khác tương tự như cây tạo mã cho CU.

Theo các phương án, ví dụ, theo chuẩn tạo mã video mới nhất hiện đang được phát triển, mà được đề cập đến là tạo mã video đa năng (Versatile Video Coding, viết tắt là VVC), sự phân chia cây tứ phân và cây nhị phân kết hợp (QTBT: Quad-Tree and Binary Tree), ví dụ, được sử dụng để phân chia khối tạo mã. Trong cấu trúc khối QTBT, CU có thể có hoặc là dạng hình vuông hoặc hình chữ nhật. Ví dụ, đơn vị cây tạo mã (coding tree unit, CTU) trước tiên được phân chia bởi cấu trúc cây tứ phân. Các nút lá cây tứ phân còn được phân chia bởi cấu trúc cây nhị phân hoặc cấu trúc cây tam phân (hoặc bội ba). Các nút lá cây phân chia được gọi là các đơn vị tạo mã (các CU), và phân đoạn đó được sử dụng để xử lý dự đoán và biến đổi mà không cần việc phân chia thêm nữa bất kỳ. Điều này có nghĩa rằng CU, PU và TU có cùng kích thước khối trong cấu trúc khối tạo mã QTBT. Đồng thời, nhiều phân chia, ví dụ, phân chia cây tam phân có thể được sử dụng cùng với cấu trúc khối QTBT.

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

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

Dự đoán trong ảnh

Tập hợp các chế độ dự đoán trong ảnh có thể bao gồm 35 các chế độ dự đoán trong ảnh khác nhau, ví dụ các chế độ vô hướng như chế độ DC (hoặc trung bình) và

chế độ phẳng, hoặc các chế độ có hướng, ví dụ như được định nghĩa trong HEVC, hoặc có thể bao gồm 67 chế độ dự đoán trong ảnh khác nhau, ví dụ các chế độ vô hướng như chế độ DC (hoặc trung bình) và chế độ phẳng, hoặc các chế độ có hướng, ví dụ như được định nghĩa đối với VVC.

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

Bộ phận dự đoán trong ảnh 254 (hoặc nói chung là bộ phận chọn chế độ 260) còn được tạo cấu hình để kết xuất các thông số dự đoán trong ảnh (hoặc nói chung là thông tin chỉ báo chế độ dự đoán trong ảnh được chọn đối với khối) đến bộ phận mã hóa entropi 270 ở dạng các phần tử cú pháp 266 để bao gồm vào dữ liệu hình ảnh được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể nhận và sử dụng các thông số dự đoán để giải mã.

Dự đoán liên ảnh

Tập hợp các chế độ dự đoán liên ảnh (hoặc khả thi) phụ thuộc vào các hình ảnh tham chiếu khả dụng (tức là các hình ảnh được giải mã ít nhất một phần trước đó, ví dụ được lưu trữ trong DBP 230) và các thông số dự đoán liên ảnh khác, ví dụ liệu toàn bộ hình ảnh tham chiếu hoặc chỉ một phần, ví dụ khu vực cửa sổ tìm kiếm quanh khu vực của khối hiện hành, của hình ảnh tham chiếu được sử dụng để tìm kiếm khối tham chiếu phù hợp nhất, và/hoặc ví dụ liệu việc nội suy điểm ảnh có được áp dụng, ví dụ nội suy nửa/bán điểm ảnh và/hoặc một phần tư điểm ảnh, hay không.

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

Bộ phận dự đoán liên ảnh 244 có thể bao gồm bộ phận ước lượng chuyển động (motion estimation, ME) và bộ phận bù chuyển động (motion compensation, MC) (đều không được thể hiện trên Fig.2). Bộ phận ước lượng chuyển động có thể được tạo cấu

hình để nhận hoặc thu nhận khối hình ảnh 203 (khối hình ảnh hiện hành 203 của hình ảnh hiện hành 17) và hình ảnh được giải mã 231, hoặc ít nhất một hoặc nhiều khối được tái tạo trước đó, ví dụ các khối được tái tạo của một hoặc nhiều hình ảnh được giải mã trước đó 231 khác/khác nhau, để ước lượng chuyển động. Ví dụ chuỗi video có thể bao gồm ảnh hiện thời và ảnh được giải mã trước đó 231, hoặc nói cách khác, ảnh hiện thời và ảnh được giải mã trước đó 231 có thể là một phần hoặc tạo ra chuỗi các ảnh tạo ra chuỗi video.

Bộ mã hóa 20 có thể, ví dụ, được tạo cấu hình để chọn khối tham chiếu từ các khối tham chiếu của các hình ảnh giống hoặc khác nhau của các hình ảnh khác và cung cấp hình ảnh tham chiếu (hoặc chỉ số hình ảnh tham chiếu) và/hoặc độ lệch (độ lệch không gian) giữa vị trí (các tọa độ x, y) của khối tham chiếu và vị trí của khối hiện hành như là các thông số dự đoán liên ảnh cho bộ phận ước lượng chuyển động. Độ lệch này cũng được gọi là vector chuyển động (motion vector, MV).

Bộ phận bù chuyển động được tạo cấu hình để thu nhận, ví dụ nhận, thông số dự đoán liên ảnh và để thực hiện dự đoán liên ảnh dựa trên hoặc sử dụng thông số dự đoán liên ảnh để thu nhận khối dự đoán liên ảnh 265. Bù chuyển động, được thực hiện bởi bộ phận bù chuyển động, có thể bao gồm việc tìm hoặc tạo khối dự đoán dựa trên vector chuyển động/khối được xác định bằng cách ước lượng chuyển động, có thể thực hiện các phép nội suy đến độ chính xác điểm ảnh con. Lọc nội suy có thể tạo các mẫu điểm ảnh bổ sung từ các mẫu điểm ảnh đã biết, vì vậy có khả năng tăng số lượng khối dự đoán ứng viên mà có thể được sử dụng để tạo mã khối hình ảnh. Khi nhận vector chuyển động đối với PU của khối hình ảnh hiện hành, bộ phận bù chuyển động có thể định vị khối dự đoán mà vector chuyển động trỏ đến trong một trong số các danh sách hình ảnh tham chiếu.

Bộ phận bù chuyển động cũng có thể tạo các phần tử cú pháp được liên kết với các khối và các lát video để sử dụng bởi bộ giải mã video 30 khi giải mã các khối hình

ảnh của lát video. Ngoài hoặc theo cách khác đối với các lát và các phần tử cú pháp tương ứng, các nhóm ảnh xếp lớp và/hoặc các ảnh xếp lớp và các phần tử cú pháp tương ứng có thể được tạo ra hoặc được sử dụng.

Mã hóa entropi

Bộ phận mã hóa entropi 270 được tạo cấu hình để áp dụng, ví dụ, thuật toán hoặc sơ đồ mã hóa entropi (ví dụ, sơ đồ tạo mã độ dài biến số (variable length coding, VLC), sơ đồ VLC thích ứng ngữ cảnh (context adaptive VLC, CAVLC), sơ đồ tạo mã số học, nhị phân hóa, tạo mã số học nhị phân thích ứng ngữ cảnh (context adaptive binary arithmetic coding, CABAC), tạo mã số học nhị phân thích ứng ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding, SBAC), tạo mã entropi phân chia khoảng xác suất (probability interval partitioning entropy, PIPE) hoặc hệ phương pháp hoặc kỹ thuật mã hóa entropi khác) hoặc rẽ nhánh (không nén) trên các hệ số được lượng tử hóa 209, các thông số dự đoán liên ảnh, các thông số dự đoán trong ảnh, các thông số lọc vòng và/hoặc các phần tử cú pháp khác để thu nhận dữ liệu hình ảnh được mã hóa 21 mà có thể được kết xuất qua đầu ra 272, ví dụ ở dạng dòng bit được mã hóa 21, sao cho, ví dụ, bộ giải mã video 30 có thể nhận và sử dụng các thông số để giải mã. Dòng bit được mã hóa 21 có thể được truyền đến bộ giải mã video 30, hoặc được lưu trữ trong bộ nhớ để truyền hoặc truy hồi sau đó bởi bộ giải mã video 30.

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

Bộ giải mã và phương pháp giải mã

Fig.3 thể hiện một ví dụ về bộ giải mã video 30 mà được tạo cấu hình để thực hiện các kỹ thuật của sáng chế. Bộ giải mã video 30 được tạo cấu hình để nhận dữ liệu hình ảnh được mã hóa 21 (ví dụ, dòng bit được mã hóa 21), ví dụ được mã hóa bởi bộ mã hóa 20, để thu nhận hình ảnh được giải mã 331. Dữ liệu hình ảnh hoặc dòng bit được mã hóa bao gồm thông tin để giải mã dữ liệu hình ảnh được mã hóa, ví dụ dữ liệu mà đại diện cho các khối hình ảnh của lát video được mã hóa (và/hoặc các nhóm ảnh xếp lớp hoặc các ảnh xếp lớp) và các phần tử cú pháp được liên kết.

Trong ví dụ trên Fig.3, bộ giải mã 30 bao gồm cụm giải mã entropy 304, cụm lượng tử hóa ngược 310, cụm xử lý biến đổi ngược 312, cụm tái cấu trúc 314 (ví dụ, bộ cộng 314), bộ lọc vòng 320, bộ đệm ảnh được giải mã (DBP) 330, cụm ứng dụng chế độ 360, cụm dự đoán liên ảnh 344 và cụm dự đoán trong ảnh 354. Bộ phận dự đoán liên ảnh 344 có thể là hoặc bao gồm bộ phận bù chuyển động. Theo một số ví dụ, bộ giải mã video 30 có thể thực hiện đường giải mã nói chung nghịch đảo với đường tạo mã mà đã được mô tả đối với bộ mã hóa video 100 từ FIG.2.

Như được giải thích đối với bộ mã hóa 20, cụm lượng tử hóa ngược 210, cụm xử lý chuyển đổi ngược 212, cụm tái cấu trúc 214, bộ lọc vòng lặp 220, bộ đệm ảnh đã được giải mã (DPB) 230, cụm dự đoán liên ảnh 344 và cụm dự đoán trong ảnh 354 còn được gọi là các thành phần tạo ra “bộ giải mã tích hợp” của bộ mã hóa video 20. Theo đó, bộ phận lượng tử hóa ngược 310 có thể giống hệt về chức năng với bộ phận lượng tử hóa ngược 110, bộ phận xử lý biến đổi ngược 312 có thể giống hệt về chức năng với bộ phận xử lý biến đổi ngược 212, bộ phận tái tạo 314 có thể giống hệt về chức năng với bộ phận tái tạo 214, bộ lọc vòng 320 có thể giống hệt về chức năng với bộ lọc vòng 220, và bộ đệm hình ảnh được giải mã 330 có thể giống hệt về chức năng với bộ đệm hình ảnh được giải mã 230. Do đó, các phần giải thích được cung cấp đối với các bộ phận và các chức năng tương ứng của bộ mã hóa video 20 áp dụng tương ứng vào các bộ phận và các chức năng tương ứng của bộ giải mã video 30.

Giải mã entropi

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

Lượng tử hóa ngược

Bộ phận lượng tử hóa ngược 310 có thể được tạo cấu hình để nhận các thông số lượng tử hóa (QP) (hoặc nói chung là thông tin liên quan đến lượng tử hóa ngược) và các hệ số được lượng tử hóa từ dữ liệu hình ảnh được mã hóa 21 (ví dụ bằng cách phân tách và/hoặc giải mã, ví dụ bằng bộ phận giải mã entropi 304) và để áp dụng, dựa trên các thông số lượng tử hóa, lượng tử hóa ngược trên các hệ số được lượng tử hóa được giải mã 309 để thu nhận các hệ số được khử lượng tử hóa 311, mà cũng có thể được gọi là các hệ số biến đổi 311. Quy trình lượng tử hóa ngược có thể bao gồm

việc sử dụng thông số lượng tử hóa được xác định bởi bộ mã hóa video 20 đối với mỗi khối video trong lát video (hoặc ảnh xếp lớp hoặc nhóm ảnh xếp lớp) để xác định mức độ lượng tử hóa và, tương tự, mức độ lượng tử hóa ngược mà sẽ được áp dụng.

Biến đổi ngược

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

Tái tạo

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

Lọc

Bộ phận lọc vòng 320 (trong vòng mã hóa hoặc sau vòng mã hóa) được tạo cấu hình để lọc khối được tái tạo 315 để thu nhận khối được lọc 321, ví dụ để làm trơn nhẵn các sự chuyển tiếp điểm ảnh, hoặc nếu không thì nâng cao chất lượng video. Cụm bộ lọc vòng lặp 320 có thể bao gồm một hoặc nhiều bộ lọc vòng lặp như bộ lọc giải khối, bộ lọc độ lệch phù hợp mẫu (SAO: sample-adaptive offset) hoặc một hoặc nhiều bộ lọc khác, ví dụ bộ lọc hai chiều, bộ lọc vòng lặp thích ứng (ALF), các bộ lọc

làm sắc nét, làm mịn hoặc các bộ lọc cộng tác, hoặc sự kết hợp bất kỳ của chúng. Mặc dù cụm bộ lọc vòng lặp 320 được thể hiện trên FIG.3 là bộ lọc trong vòng lặp, trong các kết cấu khác, cụm bộ lọc vòng lặp 320 có thể được triển khai dưới dạng bộ lọc sau vòng lặp.

Bộ đệm ảnh giải mã

Các khối video được giải mã 321 của hình ảnh sau đó được lưu trữ trong bộ đệm hình ảnh được giải mã 330, mà lưu trữ các hình ảnh được giải mã 331 như là các hình ảnh tham chiếu để bù chuyển động sau đó đối với các hình ảnh khác và/hoặc để kết xuất việc hiển thị tương ứng.

Bộ giải mã 30 được tạo cấu hình để kết xuất hình ảnh được giải mã 311, ví dụ qua đầu ra 312, để trình diễn cho người dùng hoặc cho người dùng xem.

Dự đoán

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

Khi lát video được mã hóa như là lát được mã hóa trong ảnh (I), bộ phận dự đoán trong ảnh 354 của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo khối dự đoán 365 đối với khối hình ảnh của lát video hiện hành dựa trên chế độ dự đoán trong ảnh được báo hiệu và dữ liệu từ các khối được giải mã trước đó của hình ảnh

hiện hành. Khi hình ảnh video được tạo mã như là lát được tạo mã liên ảnh (tức là, B hoặc P), bộ phận dự đoán liên ảnh 344 (ví dụ, bộ phận bù chuyển động) của bộ phận ứng dụng chế độ 360 được tạo cấu hình để tạo ra các khối dự đoán 365 đối với khối video của lát video hiện hành dựa trên các vectơ chuyển động và các phần tử cú pháp khác được nhận từ bộ phận giải mã entropi 304. Đối với dự đoán liên ảnh, các khối dự đoán có thể được tạo ra từ một trong số các hình ảnh tham chiếu trong một trong số các danh sách hình ảnh tham chiếu. Bộ giải mã video 30 có thể xây dựng các danh sách khung tham chiếu, List (danh sách) 0 và List (danh sách) 1, sử dụng các kỹ thuật xây dựng mặc định dựa trên các hình ảnh tham chiếu được lưu trữ trong DPB 330. Điều giống hoặc tương tự có thể được áp dụng cho hoặc bởi các phương án sử dụng các nhóm ảnh xếp lớp (ví dụ, các nhóm ảnh xếp lớp video) và/hoặc các ảnh xếp lớp (ví dụ, các ảnh xếp lớp video) ngoài hoặc theo cách khác đối với các lát (ví dụ, các lát video), ví dụ video có thể được tạo mã nhờ sử dụng các nhóm ảnh xếp lớp và/hoặc các ảnh xếp lớp I, P hoặc B.

Bộ phận ứng dụng chế độ 360 được tạo cấu hình để xác định thông tin dự đoán đối với khối video của lát video hiện hành bằng cách phân tách các vectơ chuyển động hoặc thông tin liên quan và các phần tử cú pháp khác, và sử dụng thông tin dự đoán để tạo ra các khối dự đoán đối với khối video hiện hành đang được giải mã. Ví dụ, bộ phận ứng dụng chế độ 360 sử dụng một số trong số các phần tử cú pháp được nhận để xác định chế độ dự đoán (ví dụ, dự đoán trong ảnh hoặc liên ảnh) được sử dụng để tạo mã các khối video của lát video, loại lát dự đoán liên ảnh (ví dụ, lát B, lát P, hoặc lát GPB), thông tin xây dựng đối với một hoặc nhiều danh sách trong số các danh sách hình ảnh tham chiếu đối với lát, các vectơ chuyển động đối với mỗi khối video được mã hóa liên ảnh của lát, trạng thái dự đoán liên ảnh đối với mỗi khối video được mã hóa liên ảnh của lát, và thông tin khác để giải mã các khối video trong lát video hiện hành. Điều giống hoặc tương tự có thể được áp dụng cho hoặc bởi các phương án sử

dụng các nhóm ảnh xếp lớp (ví dụ, các nhóm ảnh xếp lớp video) và/hoặc các ảnh xếp lớp (ví dụ, các ảnh xếp lớp video) ngoài hoặc theo cách khác đối với các lát (ví dụ, các lát video), ví dụ video có thể được tạo mã nhờ sử dụng các nhóm ảnh xếp lớp và/hoặc các ảnh xếp lớp I, P hoặc B.

Theo các phương án thì bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân vùng và/hoặc giải mã ảnh nhờ sử dụng các lát cắt (còn được gọi là các lát cắt video), trong đó ảnh có thể được phân vùng thành hoặc được giải mã nhờ sử dụng một hoặc nhiều lát cắt (thường là không chồng nhau), và mỗi lát cắt có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU).

Các phương án của bộ giải mã video 30 như được thể hiện trên Fig.3 có thể được tạo cấu hình để phân chia và/hoặc giải mã hình ảnh bằng cách sử dụng các nhóm ảnh xếp lớp (cũng được gọi là các nhóm ảnh xếp lớp video) và/hoặc các ảnh xếp lớp (cũng được gọi là các ảnh xếp lớp video), trong đó hình ảnh có thể được phân chia thành hoặc được giải mã nhờ sử dụng một hoặc nhiều nhóm ảnh xếp lớp (thường là không chồng chéo), và mỗi nhóm ảnh xếp lớp có thể bao gồm, ví dụ một hoặc nhiều khối (ví dụ, các CTU) hoặc một hoặc nhiều ảnh xếp lớp, trong đó mỗi ảnh xếp lớp, ví dụ có thể là hình chữ nhật và có thể bao gồm một hoặc nhiều khối (ví dụ, các CTU), ví dụ các khối hoàn chỉnh hoặc các khối thành phần.

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

Cần hiểu rằng, trong bộ mã hóa 20 và bộ giải mã 30, kết quả xử lý của bước hiện hành có thể được xử lý thêm và sau đó được kết xuất sang tiếp theo bước. Ví dụ, sau khi lọc nội suy, dẫn xuất vector chuyển động hoặc lọc vòng, thao tác thêm nữa, như cắt hoặc dịch chuyển, có thể được thực hiện trên kết quả xử lý của lọc nội suy, dẫn xuất vector chuyển động hoặc lọc vòng.

Cần lưu ý rằng các thao tác thêm nữa có thể được áp dụng vào các vector chuyển động được dẫn xuất của khối hiện hành (bao gồm nhưng không bị giới hạn ở việc điều khiển các vector chuyển động điểm của chế độ afin, các vector chuyển động khối con trong afin, các chế độ phẳng, ATMVP, các vector chuyển động thời gian, v.v.). Ví dụ, giá trị của vector chuyển động bị ràng buộc ở khoảng được xác định trước theo bit đại diện của nó. Nếu bit đại diện của vector chuyển động là bitDepth, thì khoảng này là $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)-1}$, trong đó “^” có nghĩa là mũ. Ví dụ, nếu bitDepth được thiết lập bằng 16, thì khoảng này là $-32768 \sim 32767$; nếu bitDepth được thiết lập bằng 18, thì khoảng này là $-131072 \sim 131071$. Ví dụ, giá trị của vector chuyển động được dẫn xuất (ví dụ các MV của bốn khối con 4x4 trong một khối 8x8) bị ràng buộc sao cho chênh lệch lớn nhất giữa các phần nguyên của bốn MV khối con 4x4 không lớn hơn N điểm ảnh, như không lớn hơn 1 điểm ảnh. Ở đây đề xuất hai phương pháp để ràng buộc vector chuyển động theo bitDepth.

Phương pháp 1: loại bỏ MSB tràn (bit có nghĩa nhiều nhất) bởi các phép toán sau đây:

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

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

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

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

(4)

trong đó mvx là thành phần nằm ngang của vector chuyển động của khối hình ảnh hoặc khối con, mvy là thành phần thẳng đứng của vector chuyển động của khối hình ảnh hoặc khối con, và ux và uy biểu thị giá trị trung gian;

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

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

(5)

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

(6)

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

(7)

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

(8)

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

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

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

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

trong đó v_x là thành phần nằm ngang của vector chuyển động của khối hình ảnh hoặc khối con, v_y là thành phần thẳng đứng của vector chuyển động của khối hình ảnh hoặc khối con; x , y và z lần lượt tương ứng với ba đầu vào giá trị của xử lý xén MV, và định nghĩa của hàm Clip3 là như sau:

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

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

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

Bộ xử lý 430 được thực thi bởi phần cứng và phần mềm. Bộ xử lý 430 có thể được thực thi như là một hoặc nhiều chip CPU, nhân (ví dụ, bộ xử lý nhiều nhân), các FPGA, các ASIC, và các DSP. Bộ xử lý 430 truyền thông với các cổng vào 410, các

bộ phận nhận 420, các bộ phận truyền 440, các cổng ra 450, và bộ nhớ 460. Bộ xử lý 430 bao gồm môđun tạo mã 470. Môđun tạo mã 470 thực thi các phương án được bộc lộ được mô tả trên đây. Ví dụ, môđun tạo mã 470 thực hiện, xử lý, chuẩn bị, hoặc cung cấp các thao tác tạo mã khác nhau. Việc bao gồm môđun tạo mã 470 do đó sẽ mang lại sự cải tiến quan trọng cho chức năng của thiết bị tạo mã video 400 và đem lại sự biến đổi thiết bị tạo mã video 400 sang trạng thái khác. Theo cách khác, môđun tạo mã 470 được thực thi dưới dạng các lệnh được lưu trữ trong bộ nhớ 460 và được thực hiện bởi bộ xử lý 430.

Bộ nhớ 460 có thể bao gồm một hoặc nhiều đĩa, ổ băng, và ổ trạng thái rắn và có thể được sử dụng làm thiết bị lưu trữ dữ liệu tràn, để lưu trữ các chương trình khi các chương trình này được chọn để thực hiện, và để lưu trữ các lệnh và dữ liệu được đọc trong suốt quá trình thực hiện chương trình. Bộ nhớ 460 có thể là, ví dụ, khả biến và/hoặc bất khả biến và có thể là bộ nhớ chỉ đọc (ROM), bộ nhớ truy cập ngẫu nhiên (RAM), bộ nhớ khả lập địa chỉ nội dung bậc ba (ternary content-addressable memory, TCAM), và/hoặc bộ nhớ truy cập ngẫu nhiên tĩnh (SRAM).

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

Bộ xử lý 502 trong máy 500 có thể là bộ xử lý trung tâm. Theo cách khác, bộ xử lý 502 có thể là loại thiết bị khác bất kỳ, hoặc nhiều thiết bị, có khả năng thao tác hoặc xử lý thông tin hiện có hoặc được phát triển trong tương lai. Mặc dù các cách thực hiện được bộc lộ có thể được thực hành với một bộ xử lý đơn lẻ như được thể hiện, ví dụ, bộ xử lý 502, các ưu điểm về tốc độ và hiệu năng có thể đạt được nhờ sử dụng nhiều hơn một bộ xử lý.

Bộ nhớ 504 trong máy 500 có thể là thiết bị nhớ chỉ đọc (ROM) hoặc thiết bị nhớ truy cập ngẫu nhiên (RAM) theo một cách thực hiện. Loại thiết bị lưu trữ thích hợp khác

bất kỳ có thể được sử dụng làm bộ nhớ 504. Bộ nhớ 504 có thể bao gồm mã và dữ liệu 506 mà được truy cập bởi bộ xử lý 502 nhờ sử dụng bus 512. Bộ nhớ 504 có thể còn bao gồm hệ điều hành 508 và các chương trình ứng dụng 510, các chương trình ứng dụng 510 bao gồm ít nhất một chương trình mà cho phép bộ xử lý 502 thực hiện các phương pháp được mô tả ở đây. Ví dụ, các chương trình ứng dụng 510 có thể bao gồm các ứng dụng từ 1 đến N, mà còn bao gồm ứng dụng tạo mã video mà thực hiện các phương pháp này được mô tả ở đây.

Máy 500 cũng có thể bao gồm một hoặc nhiều thiết bị đầu ra, như bộ hiển thị 518. Bộ hiển thị 518 có thể là, theo một ví dụ, bộ hiển thị nhạy chạm mà kết hợp bộ hiển thị với phần tử nhạy chạm mà có thể hoạt động để nhận biết các đầu vào chạm. Bộ hiển thị 518 có thể được ghép nối với bộ xử lý 502 qua bus 512.

Mặc dù được thể hiện ở đây dưới dạng một bus đơn lẻ, nhưng bus 512 của máy 500 có thể bao gồm nhiều bus. Hơn nữa, bộ phận lưu trữ thứ cấp 514 có thể được ghép nối trực tiếp với các thành phần khác của máy 500 hoặc có thể được truy cập qua mạng và có thể bao gồm một bộ phận tích hợp đơn lẻ như thẻ nhớ hoặc nhiều bộ phận như nhiều thẻ nhớ. Do đó, thiết bị 500 có thể được thực hiện theo nhiều cấu hình khác nhau.

Các tập thông số

Các tập thông số về cơ bản tương tự và chia sẻ các mục đích thiết kế cơ bản giống nhau —cụ thể là hiệu suất tốc độ bit, tính co giãn lỗi, và cung cấp các giao diện lớp hệ thống. Có thứ bậc của các tập thông số trong HEVC (H.265), bao gồm tập thông số video (VPS: Video Parameter Set), tập thông số chuỗi (SPS: Sequence Parameter Set) và tập thông số ảnh (PPS: Picture Parameter Set) vốn tương tự với các bản sao của chúng trong AVC và VVC. Mỗi lát cắt tham chiếu đến một PPS, SPS và VPS chủ động để truy nhập thông tin dùng cho việc giải mã lát cắt. PPS chứa thông tin mà áp dụng cho tất cả các lát cắt trong ảnh, và do đó tất cả các lát cắt trong ảnh phải tham chiếu đến PPS giống nhau. Các lát cắt trong khác nhau các ảnh cũng được phép tham

chiều đến PPS giống nhau. Tương tự, SPS chứa thông tin mà áp dụng cho tất cả các ảnh trong chuỗi video giống nhau đã được tạo mã.

Mặc dù PPS có thể khác nhau đối với các ảnh riêng biệt, nhưng nhiều hoặc tất cả các ảnh trong chuỗi video đã được tạo mã thường tham chiếu đến PPS giống nhau. Việc sử dụng lại các tập thông số là có hiệu quả tốc độ bit do tránh được nhu cầu gửi thông tin chia sẻ nhiều lần. Cũng có thất thoát lớn do nó cho phép tập thông số nội dung được mang bởi một số lớn liên kết truyền thông bên ngoài có thể tin cậy hoặc được lặp lại thường xuyên bên trong dòng bit để đảm bảo rằng sẽ không bị thất thoát.

Việc tạo mã video khả biến, lớp và tập thông số video (VPS)

Việc tạo mã video khả biến cung cấp cơ chế để tạo mã video trong nhiều lớp, mỗi lớp thể hiện sự trình diễn chất lượng của cảnh video giống nhau. Lớp đề (BL) có sự trình diễn với chất lượng thấp nhất. Một hoặc nhiều lớp tăng cường (EL: enhancement layer) có thể được tạo mã bước tham chiếu các lớp dưới và cung cấp chất lượng video được cải thiện. Việc giải mã tập hợp con của các lớp của dòng bit video được tạo mã khả biến dẫn tới video với chất lượng thấp hơn nhưng vẫn chấp nhận được. Điều này cho phép giảm chất lượng tốt hơn so với các dòng bit video bất khả biến, trong đó việc giảm tốc độ bit thường khiến cho giảm nhiều hơn chất lượng video.

Có nhiều loại khả năng thay đổi tồn tại trong chuỗi video khả biến, bao gồm khả năng thay đổi theo thời gian, khả năng thay đổi không gian và khả năng thay đổi chất lượng. FIG.11 cung cấp ví dụ thể hiện cả khả năng thay đổi không gian và thời gian. Trên FIG.11, hai lớp được tạo mã trong các độ phân giải khác nhau. BL có độ phân giải thấp trong khi EL có độ phân giải cao hơn, và khả năng thay đổi theo không gian đạt được nhờ tạo ra bộ giải mã để giải mã hoặc BL, EL, hoặc cả hai.

Ngoài khả năng thay đổi theo không gian, khả năng thay đổi theo thời gian đạt được bên trong lớp tạo mã. Trong ví dụ này, mỗi tạo mã lớp được chia thành hai lớp

con thời gian, được gắn nhãn bởi ID thời gian 0 và IID 1, một cách tương ứng. Khả năng thay đổi theo thời gian đạt được nhờ cung cấp sự giải mã hoặc lớp con thời gian 0 (với ID thời gian bằng 0) hoặc cả hai lớp con 0 và 1.

Các ảnh trong mỗi lớp được ấn định với id lớp, ví dụ với thành phần cú pháp `nuh_layer_id`. Chuỗi video lớp được tạo mã (CLVS) là chuỗi của các ảnh với giá trị giống nhau của `nuh_layer_id` mà bao gồm, theo thứ tự giải mã, chuỗi video lớp tạo mã đặc biệt bắt đầu tạo mã ảnh (CLVSS, ví dụ trong ảnh), theo sau bởi không hoặc nhiều ảnh mà không là các ảnh CLVSS, bao gồm tất cả các ảnh phía sau lên tới nhưng không bao gồm bất kỳ tiếp theo các ảnh mà là ảnh CLVSS.

Chuỗi video đã được tạo mã (CVS: coded video sequence) bao gồm một trong số các nhiều chuỗi video lớp được tạo mã (CLVS). Theo ví dụ trên FIG.11, giả sử các ảnh thứ nhất của BL và EL là ảnh CLVSS và tất cả khác các ảnh không là các ảnh CLVSS, CVS này bao gồm hai CLVS.

Tập thông số chuỗi (SPS)

SPS chứa các thông số mà áp dụng cho một các lớp của chuỗi video đã được tạo mã và không thay đổi từ ảnh đến ảnh bên trong chuỗi video đã được tạo mã.

Trong một số trường hợp đặc biệt, SPS có thể không được sử dụng bởi các ảnh bất kỳ trong CLVS.

Cũng có thể có trường hợp mà SPS được chia sẻ bởi các CLVS khác nhau.

Trong bản thảo đặc tả kỹ thuật VVC cuối cùng (tức là http://phenix.it-sudparis.eu/jvet/doc_end_user/cac_tai_lieu/19_Teleconference/wg11/JVET-S0152-v5.zip, được gọi là bản thảo VVC trong phần sau cho đơn giản), định nghĩa của SPS là như sau:

tập thông số chuỗi (SPS): Cấu trúc cú pháp chứa các phần tử cú pháp mà áp dụng cho không hoặc nhiều CLVS toàn cục như được xác định bởi nội dung của thành phần cú pháp được tìm thấy trong PPS được tham chiếu đến bởi thành phần cú pháp được tìm thấy trong mỗi tiêu đề ảnh.

Đối với định nghĩa của PPS và tiêu đề ảnh, xem bản thảo VVC.

Cụ thể là, SPS bao gồm thông tin về việc báo hiệu bộ đệm ảnh đã được giải mã (tức là dpb).

Một số phần của bảng sau đây thể hiện ảnh chụp của một phần của báo hiệu dpb trong SPS trong VVC:

Báo hiệu thông tin bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer).

Bộ đệm ảnh giải mã

Bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) là bộ đệm mà được sử dụng để lưu trữ ảnh được giải mã để tham khảo, ví dụ, làm ảnh tham chiếu cho sự dự đoán liên ảnh. Theo một ví dụ được bộc lộ trong bản thảo VVC, các phần tử cú pháp liên quan của các thông số cho DBP trong tập thông số chuỗi (SPS) được làm nổi bật.

Bảng 1: Các phần tử cú pháp liên quan tới ảnh được giải mã trong SPS

...	
if(sps_ptl_dpb_hrd_params_present_flag) {	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
}	
...	

`sps_ptl_dpb_hrd_params_present_flag` bằng 1 chỉ định rằng cấu trúc cú pháp `profile_tier_level()` và cấu trúc cú pháp `dpb_parameters()` có mặt trong SPS, và cấu trúc cú pháp `general_hrd_parameters()` và cấu trúc cú pháp `ols_hrd_parameters()` cũng có thể có mặt trong SPS. `sps_ptl_dpb_hrd_params_present_flag` bằng 0 chỉ định rằng không có cấu trúc nào trong số bốn cấu trúc cú pháp này có mặt trong SPS.

Khi `sps_video_parameter_set_id` lớn hơn 0 và có OLS mà chứa chỉ một lớp với `nuh_layer_id` bằng `nuh_layer_id` của SPS, hoặc khi `sps_video_parameter_set_id` bằng 0, giá trị của `sps_ptl_dpb_hrd_params_present_flag` sẽ bằng 1.

OLS (tập hợp lớp đầu ra) là tập hợp của các lớp mà một hoặc nhiều lớp được định rõ cho chúng là các lớp đầu ra. Lớp đầu ra là lớp của tập hợp lớp đầu ra mà được xuất ra.

Các bảng cú pháp cho các cấu trúc cú pháp `profile_tier_level()`, `dpb_parameters()`, `general_hrd_parameters()`, `ols_hrd_parameters()` có thể được tìm thấy trong bản thảo VVC.

Thành phần cú pháp `sps_max_sublayers_minus1` chỉ báo số lượng các lớp con thời gian khả dụng. Khi số lượng các lớp con thời gian khả dụng lớn hơn 1 (ví dụ giá trị của thành phần cú pháp `sps_max_sublayers_minus1` lớn hơn 0), giá trị của thành phần cú pháp `sps_sublayer_dpb_params_flag` được báo hiệu trong dòng bit. Thành phần cú pháp `sps_sublayer_dpb_params_flag` bắt buộc báo hiệu thông tin giải mã ảnh cho mỗi khả dụng lớp con (khi của nó giá trị bằng 1) hoặc chỉ báo hiệu cho lớp con thời gian cao nhất (khi của nó giá trị bằng 0). Khi giá trị của `sps_sublayer_dpb_params_flag` không được thể hiện, ví dụ chỉ có một lớp con thời gian (giá trị của `sps_max_sublayers_minus1` == 0), giá trị của `sps_max_sublayers_minus1` được suy ra là 0.

Theo một số ví dụ, giá trị của thành phần cú pháp `sps_ptl_dpb_hrd_params_present_flag` chỉ báo liệu cấu trúc báo hiệu `dpb_parameter()`

có được báo hiệu trong SPS hay không. Khi giá trị của thành phần cú pháp `sps_ptl_dpb_hrd_params_present_flag` bằng 1, cấu trúc ngay được báo hiệu `dpb_parameter()` được gọi, với số lượng các lớp con thời gian khả dụng trừ đi 1 (`sps_max_sublayers_minus1`) và cờ `sps_sublayer_dpb_params_flag` làm thông số thứ nhất và thứ hai, một cách tương ứng.

Theo một ví dụ, cấu trúc báo hiệu của `dpb_parameter()` trong bản thảo VVC được xác định như sau:

Bảng 2: định nghĩa của cấu trúc cú pháp `dpb_parameters`

<code>dpb_parameters(maxSubLayersMinus1, subLayerInfoFlag) {</code>	Bộ mô tả
<code>for(i = (subLayerInfoFlag ? 0 : maxSubLayersMinus1);</code>	
<code> i <= maxSubLayersMinus1; i++) {</code>	
<code> max_dec_pic_buffering_minus1[i]</code>	ue(v)
<code> max_num_reorder_pics[i]</code>	ue(v)
<code> max_latency_increase_plus1[i]</code>	ue(v)
<code>}</code>	
<code>}</code>	

Cấu trúc cú pháp `dpb_parameters()` cung cấp thông tin của kích cỡ DPB, số lượng sắp xếp lại ảnh tối đa, và độ trễ tối đa cho một hoặc nhiều OLS.

Khi cấu trúc cú pháp `dpb_parameters()` được bao hàm trong VPS, các OLS mà cấu trúc cú pháp `dpb_parameters()` áp dụng vào đó được định rõ bởi VPS. Khi cấu trúc cú pháp `dpb_parameters()` được bao hàm trong SPS, nó áp dụng cho OLS mà chỉ bao gồm lớp là lớp thấp nhất trong số các lớp mà viện dẫn đến SPS, và lớp thấp nhất này là lớp độc lập.

$\text{max_dec_pic_buffering_minus1}[i]$ cộng 1 chỉ định kích cỡ được yêu cầu tối đa của DPB trong các cụm của các bộ đệm lưu trữ ảnh khi Htid bằng i . Giá trị của $\text{max_dec_pic_buffering_minus1}[i]$ sẽ nằm trong khoảng 0 đến $\text{MaxDpbSize} - 1$, tuyến không loại, trong đó MaxDpbSize như được mô tả trong mệnh đề A.4.2. Khi i lớn hơn 0, $\text{max_dec_pic_buffering_minus1}[i]$ sẽ lớn hơn hoặc bằng $\text{max_dec_pic_buffering_minus1}[i - 1]$. Khi $\text{max_dec_pic_buffering_minus1}[i]$ không có mặt đối với i nằm trong khoảng từ 0 đến $\text{maxSubLayersMinus1} - 1$, tuyến không loại, do subLayerInfoFlag bằng 0, bằng $\text{max_dec_pic_buffering_minus1}[\text{maxSubLayersMinus1}]$.

$\text{max_num_reorder_pics}[i]$ chỉ định số lượng các ảnh được phép tối đa của OLS mà có thể ở trước bất kỳ ảnh trong OLS theo thứ tự giải mã và theo sau ảnh đó theo thứ tự xuất khi Htid bằng i . Giá trị của $\text{max_num_reorder_pics}[i]$ sẽ nằm trong khoảng 0 đến $\text{max_dec_pic_buffering_minus1}[i]$, tuyến không loại. Khi i lớn hơn 0, $\text{max_num_reorder_pics}[i]$ sẽ lớn hơn hoặc bằng $\text{max_num_reorder_pics}[i - 1]$. Khi $\text{max_num_reorder_pics}[i]$ không có mặt đối với i nằm trong khoảng từ 0 đến $\text{maxSubLayersMinus1} - 1$, tuyến không loại, do subLayerInfoFlag bằng 0, nó được suy ra là bằng $\text{max_num_reorder_pics}[\text{maxSubLayersMinus1}]$.

$\text{max_latency_increase_plus1}[i]$ không bằng 0 được sử dụng để tính toán giá trị của $\text{MaxLatencyPictures}[i]$, mà chỉ định số lớn nhất của các ảnh trong OLS mà có thể ở trước bất kỳ ảnh trong OLS theo thứ tự xuất và theo sau ảnh đó theo thứ tự giải mã khi Htid bằng i .

Khi $\text{max_latency_increase_plus1}[i]$ không bằng 0, giá trị của $\text{MaxLatencyPictures}[i]$ được chỉ định rõ như sau:

$$\text{MaxLatencyPictures}[i] = \text{max_num_reorder_pics}[i] + \text{max_latency_increase_plus1}[i] - 1 \quad (112)$$

Khi `max_latency_increase_plus1[ i ]` bằng 0, không có giới hạn tương ứng được trình bày.

Giá trị của `max_latency_increase_plus1[ i ]` sẽ nằm trong khoảng 0 đến $232 - 2$, tuyến không loại. Khi `max_latency_increase_plus1[ i ]` không có mặt đối với i nằm trong khoảng từ 0 đến `maxSubLayersMinus1 - 1`, tuyến không loại, do `subLayerInfoFlag` bằng 0, nó được suy ra là bằng `max_latency_increase_plus1[ maxSubLayersMinus1 ]`.

Thông tin chi tiết hơn đối với các phần tử cú pháp và các biến số được sử dụng để giải thích các thông số dpb (`max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và `max_latency_increase_plus1[i]`) xem bản thảo VVC.

Cấu trúc báo hiệu `dpb_parameters` hoặc báo hiệu một thông tin bộ đệm ảnh đã được giải mã của lớp con hoặc mỗi thông tin ảnh được giải mã của lớp con, được điều khiển bởi giá trị của `subLayerInfoFlag`. Khi giá trị của `subLayerInfoFlag` bằng 0, bộ đệm ảnh đã được giải mã thông tin của lớp con cao nhất trong số các lớp con thời gian khả dụng được báo hiệu ($i = \text{maxSubLayersMinus1}$). Khi giá trị của `subLayerInfoFlag` bằng 1, bộ đệm ảnh đã được giải mã thông tin của mỗi lớp con trong số các lớp con thời gian khả dụng được báo hiệu (giá trị của i nằm trong khoảng từ 0 đến `maxSubLayersMinus1`, tuyến không loại).

Vấn đề về các ngữ nghĩa của thành phần cú pháp `sps_sublayer_dpb_params_flag`

Trong bản thảo VVC, các ngữ nghĩa của `sps_sublayer_dpb_params_flag` được xác định như sau:

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong

SPS. Khi không có mặt, giá trị của `sps_sub_dpb_params_info_present_flag` được suy là bằng 0.

Các ngữ nghĩa nêu trên có hai vấn đề. Thứ nhất, có lỗi chính tả trong câu cuối cùng

Khi không có mặt, giá trị của `sps_sub_dpb_params_info_present_flag` được suy là bằng 0

`sps_sub_dpb_params_info_present_flag` không được xác định ở bất kỳ nơi nào khác và câu cuối cùng sẽ được sửa đổi như sau:

Khi không có mặt, giá trị của `sps_sublayer_dpb_params_flag` được suy là bằng 0

Thứ hai, các ngữ nghĩa của `sps_sublayer_dpb_params_flag` không đủ chính xác. Do phạm vi của `i` trong `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và `max_latency_increase_plus1[i]` không được xác định. Có thể quan sát được từ bảng 2 rằng khi `i` bằng `maxSubLayersMinus1` (tương ứng với `sps_max_sublayers_minus1` trong bảng 1), cho dù `sps_sublayer_dpb_params_flag` bằng 0 hoặc 1, các phần tử cú pháp `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và `max_latency_increase_plus1[i]` luôn được báo hiệu. `sps_sublayer_dpb_params_flag` không kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và `max_latency_increase_plus1[i]` trong trường hợp này, vốn có mâu thuẫn với định nghĩa hiện tại.

Theo một số ví dụ, thành phần cú pháp `sps_sublayer_dpb_params_flag` chỉ được sử dụng như thông số thứ hai của `dpb_parameters`.

Đối với các phương án sau đây, giả sử rằng lỗi chính tả (vấn đề thứ nhất) đã được sửa chữa, và các phương án thực hiện được đề xuất tập trung vào khắc phục vấn đề thứ hai, ngữ nghĩa không chính xác của `sps_sublayer_dpb_params_flag`

Phương án 1

Theo phương án thứ nhất, các ngữ nghĩa của `sps_sublayer_dpb_params_flag` được thay đổi như sau:

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong SPS đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không loại, khi `sps_max_sublayers_minus1` lớn hơn 0. Khi `sps_max_sublayers_minus1` bằng 0, giá trị của `sps_sublayer_dpb_params_flag` được suy là bằng 0, và `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và `max_latency_increase_plus1[ 0 ]` luôn chỉ được báo hiệu cho lớp con viện dẫn đến SPS.

Theo cách này, các ngữ nghĩa của `sps_sublayer_dpb_params_flag` được làm sáng tỏ. Chỉ khi có nhiều hơn một lớp con (`sps_max_sublayers_minus1` lớn hơn 0), thì kiểm soát sự có mặt của `minus1[ i ]`, `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters()` đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`. Ngược lại (`sps_max_sublayers_minus1` bằng 0), việc báo hiệu `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và `max_latency_increase_plus1[ 0 ]` luôn được báo hiệu, cho dù `sps_sublayer_dpb_params_flag` có giá trị nào. Do đó, các ngữ nghĩa có thể cũng được thay đổi như sau theo phương án 2

Phương án 2

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong SPS đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không

loại, khi `sps_max_sublayers_minus1` lớn hơn 0. Khi `sps_max_sublayers_minus1` bằng 0, giá trị của `sps_sublayer_dpb_params_flag` được suy là bằng 1, và `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và `max_latency_increase_plus1[ 0 ]` luôn chỉ được báo hiệu cho lớp con viện dẫn đến SPS.

Hoặc thậm chí loại bỏ quy tắc suy ra đối với giá trị của `sps_sublayer_dpb_params_flag`, như được thể hiện ở phương án thực hiện 3

Phương án 3

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong SPS đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không loại, khi `sps_max_sublayers_minus1` lớn hơn 0. Khi `sps_max_sublayers_minus1` bằng 0, `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và `max_latency_increase_plus1[ 0 ]` luôn chỉ được báo hiệu cho lớp con viện dẫn đến SPS.

Hoặc với một số giải thích khi `sps_max_sublayers_minus1` bằng 0 như được thể hiện ở phương án thực hiện 4

Phương án 4

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong SPS đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không loại, khi `sps_max_sublayers_minus1` lớn hơn 0. Khi `sps_max_sublayers_minus1` bằng 0, giá trị của `sps_sublayer_dpb_params_flag` không có ý nghĩa, và `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và

`max_latency_increase_plus1[ 0 ]` luôn chỉ được báo hiệu cho lớp con viện dẫn đến SPS.

Sự sửa đổi cho ý tưởng 1 với các phương án thực hiện 1 đến 4

Cấu trúc dữ liệu `dpb_parameter()` không chỉ được gọi trong SPS, mà còn có thể được gọi trong VPS. Phần dưới đây là ảnh chụp của VPS nơi mà `dpb_parameter()` được gọi.

Bảng 3: cấu trúc cú pháp `dpb_parameters` trong VPS

<code>for(i = 0; i < VpsNumDpbParams; i++) {</code>	
<code>if(!vps_default_ptl_dpb_hrd_max_tid_flag)</code>	
<code>vps_dpb_max_tid[i]</code>	u(3)
<code>dpb_parameters(vps_dpb_max_tid[i], vps_sublayer_dpb_params_present_flag)</code>	
<code>}</code>	

Trong đó việc định rõ số lượng các cấu trúc cú pháp `dpb_parameters()` trong VPS.

`vps_default_ptl_dpb_hrd_max_tid_flag` bằng 1 định rõ rằng các phần tử cú pháp `vps_ptl_max_tid[ i ]`, `vps_dpb_max_tid[ i ]`, và `vps_hrd_max_tid[ i ]` không có mặt và được suy là bằng giá trị mặc định `vps_max_sublayers_minus1`. `vps_default_ptl_dpb_hrd_max_tid_flag` bằng 0 định rõ rằng các phần tử cú pháp `vps_ptl_max_tid[ i ]`, `vps_dpb_max_tid[ i ]`, và `vps_ptl_max_tid[ i ]` có mặt. Khi không có mặt, giá trị của `vps_default_ptl_dpb_hrd_max_tid_flag` được suy là bằng 1.

Đối với sự giải thích chi tiết của `vps_ptl_max_tid[ i ]` và `vps_hrd_max_tid[ i ]` xem bản thảo VVC.

`vps_dpb_max_tid[ i ]` chỉ định TemporalId của lớp con cao nhất sự thể hiện mà các thông số dpb có thể có mặt cho nó trong cấu trúc cú pháp `dpb_parameters( )` thứ `i` trong VPS. Giá trị của `vps_dpb_max_tid[ i ]` sẽ nằm trong khoảng 0 đến `vps_max_sublayers_minus1`, tuyến không loại. Khi không có mặt, giá trị của `vps_dpb_max_tid[ i ]` được suy là bằng `vps_max_sublayers_minus1`

và

`vps_sublayer_dpb_params_present_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ ]`, `max_num_reorder_pics[ ]`, và `max_latency_increase_plus1[ ]` các phần tử cú pháp trong các cấu trúc cú pháp `dpb_parameters( )` trong VPS. Khi không có mặt, `vps_sub_dpb_params_info_present_flag` được suy là bằng 0.

Các ngữ nghĩa của `vps_sublayer_dpb_params_present_flag` có vấn đề tương tự với các ngữ nghĩa của `sps_sublayer_dpb_params_flag`, như được mô tả trước đó.

Kích thước của mảng thành phần cú pháp `max_dec_pic_buffering_minus1[ ]`, `max_num_reorder_pics[ ]`, và `max_latency_increase_plus1[ ]` không được xác định trong các ngữ nghĩa. Có thể quan sát được từ bảng 2 rằng khi chỉ có một lớp con (tương ứng với `vps_dpb_max_tid[ i ]` trong bảng 4 bằng 0), cho dù `vps_sublayer_dpb_params_present_flag` bằng 0 hoặc 1, các phần tử cú pháp `max_dec_pic_buffering_minus1[ ]`, `max_num_reorder_pics[ ]`, và `max_latency_increase_plus1[ ]` luôn được báo hiệu. `vps_sublayer_dpb_params_present_flag` không kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ ]`, `max_num_reorder_pics[ ]`, và `max_latency_increase_plus1[ ]` trong trường hợp này, vốn có mâu thuẫn với định nghĩa hiện tại.

Phương án thực hiện 9

Theo một ví dụ, các ngữ nghĩa của `vps_sublayer_dpb_params_present_flag` được thay đổi như sau:

`vps_sublayer_dpb_params_present_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ j ]`, `max_num_reorder_pics[ j ]`, và các phần tử cú pháp `max_latency_increase_plus1[ j ]` trong cấu trúc cú pháp `dpb_parameters( )` trong VPS đối với `j` nằm trong khoảng từ 0 đến `vps_dpb_max_tid[ i ] - 1`, tuyến không loại, khi `vps_dpb_max_tid[ i ]` trong VPS lớn hơn 0. Khi không có mặt, `vps_sub_dpb_params_info_present_flag` được suy là bằng 0.

Lưu ý rằng `j` thay cho `i` được sử dụng để định rõ phạm vi của mảng thành phần cú pháp `max_dec_pic_buffering_minus1[ ]`, `max_num_reorder_pics[ ]`, và `max_latency_increase_plus1[ ]` để tránh rối loạn của `i` trong `vps_dpb_max_tid[ i ]`.

Lưu ý, vấn đề 2 và các giải pháp tương ứng, `dependentFlag` v.v.

Khi dự đoán liên ảnh được thực hiện, ảnh tham chiếu có thể là các ảnh được giải mã từ cùng một lớp, hoặc ảnh tham chiếu từ lớp dưới. Trong trường hợp sau, ảnh tham chiếu trong lớp dưới được gọi là ảnh tham chiếu liên lớp (ILRP), và lớp dưới mà chứa một hoặc nhiều ILRP được gọi là lớp tham chiếu (hoặc phụ thuộc).

Để hỗ trợ dự đoán liên lớp, một số biến số trung gian được sử dụng theo các phần tử cú pháp được báo hiệu trong VPS. Ví dụ, trong phương trình 37 của bản thảo VVC, quy trình suy ra đối với các biến số `dependencyFlag[i][j]`, `NumDirectRefLayers[i]`, `DirectRefLayerIdx[i][ d ]`, `NumRefLayers[ i ]`, `RefLayerIdx[ i ][ r ]`, và `LayerUsedAsRefLayerFlag[ j ]` được xác định như sau:

Các biến số `dependencyFlag[ i ][ j ]`, `NumDirectRefLayers[ i ]`, `DirectRefLayerIdx[ i ][ d ]`, `NumRefLayers[ i ]`, `RefLayerIdx[ i ][ r ]`, và `LayerUsedAsRefLayerFlag[ j ]` được suy ra như sau:

```

    for( i = 0; i  <=  vps_max_layers_minus1; i++ ) {
    for( j = 0; j  <=  vps_max_layers_minus1; j++ ) {
        dependencyFlag[ i ][ j ] = vps_direct_ref_layer_flag[ i ][ j ]
        for( k = 0; k < i; k++ )
            if( vps_direct_ref_layer_flag[ i ][ k ]  &&  dependencyFlag[ k ][ j ] )
                dependencyFlag[ i ][ j ] = 1
    }
    LayerUsedAsRefLayerFlag[ i ] = 0
}
for( i = 0; i  <=  vps_max_layers_minus1; i++ ) {
    for( j = 0, d = 0, r = 0; j  <=  vps_max_layers_minus1; j++ ) { (37)
        if( vps_direct_ref_layer_flag[ i ][ j ] ) {
            DirectRefLayerIdx[ i ][ d++ ] = j
            LayerUsedAsRefLayerFlag[ j ] = 1
        }
        if( dependencyFlag[ i ][ j ] )
            RefLayerIdx[ i ][ r++ ] = j
    }
    NumDirectRefLayers[ i ] = d
    NumRefLayers[ i ] = r
}

```

dependencyFlag[i][j] bằng 1 định rõ rằng lớp với chỉ số j là lớp tham chiếu đối với lớp với chỉ số i. dependencyFlag[i][j] bằng 0 định rõ rằng lớp với chỉ số j là lớp không tham chiếu đối với lớp với chỉ số i.

Tính phụ thuộc giữa lớp với chỉ số i và lớp tham chiếu j của nó có thể là trực tiếp hoặc gián tiếp. Nếu lớp thứ j là lớp phụ thuộc trực tiếp của lớp thứ i, các phụ thuộc

được báo hiệu với thành phần cú pháp `vps_direct_ref_layer_flag[ i ][ j ]` đối với phạm vi `i` từ 0 đến `vps_max_layers_minus1`, và đối với `j` từ 0 đến `i-1` trong VPS. Như được thể hiện trong bảng 4, với các phần tử cú pháp không liên quan đến sáng chế được loại bỏ:

<code>for(i = 0; i <= vps_max_layers_minus1; i++) {</code>	
<code>...</code>	u(6)
<code>if(i > 0 && !vps_all_independent_layers_flag) {</code>	
<code> vps_independent_layer_flag[i]</code>	u(1)
<code> if(!vps_independent_layer_flag[i]) {</code>	
<code> ...</code>	u(1)
<code> for(j = 0; j < i; j++) {</code>	
<code> vps_direct_ref_layer_flag[i][j]</code>	u(1)
<code> ...</code>	
<code> }</code>	
<code>}</code>	
<code>}</code>	

Bảng cú pháp nêu trên thể hiện khi không phải tất cả các lớp là các lớp độc lập, cách mà VPS được báo hiệu sự phụ thuộc giữa các lớp.

`vps_max_layers_minus1` đại diện cho số lượng lớp được phép tối đa trong mỗi CVS tham chiếu đến VPS. Thông tin phụ thuộc lớp được báo hiệu chỉ từ lớp thấp nhất thứ hai (tức là `i=1`) và khi không phải tất cả lớp là lớp độc lập, được chỉ báo bởi thành phần cú pháp `vps_all_independent_layers_flag`.

`vps_all_independent_layers_flag` bằng 1 định rõ rằng tất cả các lớp được định rõ bởi VPS được tạo mã một cách độc lập mà không cần sử dụng dự đoán liên lớp. `vps_all_independent_layers_flag` bằng 0 định rõ rằng một hoặc nhiều trong số các lớp được định rõ bởi VPS có thể sử dụng dự đoán liên lớp. Khi không có mặt, giá trị của `vps_all_independent_layers_flag` được suy là bằng 1.

Trong VVC, lớp thấp nhất (tức là tương ứng với `vps_independent_layer_flag[0]`) luôn là lớp độc lập. Do vậy thành phần cú pháp `vps_independent_layer_flag[i]` được báo hiệu từ lớp thấp nhất thứ hai (tức là `vps_independent_layer_flag[1]`), và các ngữ nghĩa của nó được loại bỏ thẳng về phía trước.

`vps_independent_layer_flag[i]` bằng 1 định rõ rằng lớp với chỉ số i không sử dụng dự đoán liên lớp. `vps_independent_layer_flag[i]` bằng 0 định rõ rằng lớp với chỉ số i có thể sử dụng dự đoán liên lớp và các phần tử cú pháp `vps_direct_ref_layer_flag[i][j]` đối với j nằm trong khoảng từ 0 đến $i-1$, tuyến không loại, được thể hiện trong VPS. Khi không có mặt, giá trị của `vps_independent_layer_flag[i]` được suy là bằng 1.

Chỉ khi lớp không là lớp độc lập (tức là `vps_independent_layer_flag[i]` bằng 0), thì sau đó lớp phụ thuộc trực tiếp của nó được báo hiệu, được chỉ báo bởi `vps_direct_ref_layer_flag[i][j]`, đối với j nằm trong khoảng từ 0 đến $i-1$.

Các ngữ nghĩa của `vps_direct_ref_layer_flag[i][j]` được xác định như sau:

`vps_direct_ref_layer_flag[i][j]` bằng 0 định rõ rằng lớp với chỉ số j không là lớp tham chiếu trực tiếp đối với lớp với chỉ số i . `vps_direct_ref_layer_flag[i][j]` bằng 1 định rõ rằng lớp với chỉ số j là lớp tham chiếu trực tiếp đối với lớp với chỉ số i . Khi `vps_direct_ref_layer_flag[i][j]` không có mặt đối với i và j nằm trong khoảng từ 0 đến `vps_max_layers_minus1`, tuyến không loại, nó được suy ra là bằng 0. Khi `vps_independent_layer_flag[i]` bằng 0, sẽ có ít nhất một giá trị của j nằm trong

khoảng từ 0 đến $i-1$, tuyến không loại, sao cho giá trị của $vps_direct_ref_layer_flag[i][j]$ bằng 1.

Khi lớp là lớp phụ thuộc (tức là $vps_independent_layer_flag[i]$ của nó bằng 0), sự ràng buộc trong các ngữ nghĩa của $vps_direct_ref_layer_flag[i][j]$ định rõ rằng ít nhất một trong số các lớp dưới của nó j nằm trong khoảng từ 0 đến $i-1$, tuyến không loại, sẽ là lớp tham chiếu đối với lớp i .

Trong trường hợp sự phụ thuộc gián tiếp, nó có thể xuất hiện khi lớp i phụ thuộc trực tiếp vào lớp k ($k < i$) và lớp k cũng phụ thuộc vào lớp j ($j < k$). Trong trường hợp này, lớp với chỉ số i phụ thuộc vào lớp j , và $dependencyFlag[i][j]$ tương ứng bằng 1.

Vấn đề của quy trình thu được. Do duy nhất lớp có thể chỉ phụ thuộc vào lớp dưới, vòng lặp tìm kiếm lớp dưới phụ thuộc j trong quy trình suy ra có thể được đơn giản hóa. Không cần vòng lặp trên tất cả lớp trong VPS, nhưng mà chỉ cần vòng lặp cho các lớp mà thấp hơn lớp hiện tại i .

Phương án thực hiện 5

Theo một ví dụ, quy trình suy ra của các biến số $dependencyFlag[i][j]$, $NumDirectRefLayers[i]$, $DirectRefLayerIdx[i][d]$, $NumRefLayers[i]$, $RefLayerIdx[i][r]$, và $LayerUsedAsRefLayerFlag[j]$ được thay đổi như sau:

Các biến số $dependencyFlag[i][j]$, $NumDirectRefLayers[i]$, $DirectRefLayerIdx[i][d]$, $NumRefLayers[i]$, $RefLayerIdx[i][r]$, và $LayerUsedAsRefLayerFlag[j]$ được suy ra như sau:

```
for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0; j < i; j++ ) {
        dependencyFlag[i][j] = vps_direct_ref_layer_flag[i][j]
        for( k = 0; k < i; k++ )
```

```

        if(          vps_direct_ref_layer_flag[ i ][ k ]          &&
dependencyFlag[ k ][ j ] )
        dependencyFlag[ i ][ j ] = 1
    }
    LayerUsedAsRefLayerFlag[ i ] = 0
}
for( i = 0; i  <=  vps_max_layers_minus1; i++ ) {
    for( j = 0, d = 0, r = 0; j  < i; j++ ) {
(37)
        if( vps_direct_ref_layer_flag[ i ][ j ] ) {
            DirectRefLayerIdx[ i ][ d++ ] = j
            LayerUsedAsRefLayerFlag[ j ] = 1
        }
        if( dependencyFlag[ i ][ j ] )
            RefLayerIdx[ i ][ r++ ] = j
    }
    NumDirectRefLayers[ i ] = d
    NumRefLayers[ i ] = r
}

```

Phương án thực hiện 6

Theo ví dụ khác, quy trình suy ra của các biến số `dependencyFlag[i][j]`, `NumDirectRefLayers[i]`, `DirectRefLayerIdx[i][ d ]`, `NumRefLayers[ i ]`, `RefLayerIdx[ i ][ r ]`, và `LayerUsedAsRefLayerFlag[ j ]` được thay đổi như sau:

Các biến số `dependencyFlag[ i ][ j ]`, `NumDirectRefLayers[ i ]`, `DirectRefLayerIdx[ i ][ d ]`, `NumRefLayers[ i ]`, `RefLayerIdx[ i ][ r ]`, và `LayerUsedAsRefLayerFlag[ j ]` được suy ra như sau:

```

for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0; j < i; j++ ) {
        dependencyFlag[ i ][ j ] = vps_direct_ref_layer_flag[ i ][ j ]
        for( k = j+1; k < i; k++ )
            if( vps_direct_ref_layer_flag[ i ][ k ] &&
                dependencyFlag[ k ][ j ] )
                dependencyFlag[ i ][ j ] = 1
    }
    LayerUsedAsRefLayerFlag[ i ] = 0
}

for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0, d = 0, r = 0; j < i; j++ ) {
(37)
        if( vps_direct_ref_layer_flag[ i ][ j ] ) {
            DirectRefLayerIdx[ i ][ d++ ] = j
            LayerUsedAsRefLayerFlag[ j ] = 1
        }
        if( dependencyFlag[ i ][ j ] )
            RefLayerIdx[ i ][ r++ ] = j
    }
    NumDirectRefLayers[ i ] = d

```

```

        NumRefLayers[ i ] = r
    }

```

Phương án thực hiện 7

Theo ví dụ khác, quy trình suy ra của các biến số $\text{dependencyFlag}[i][j]$, $\text{NumDirectRefLayers}[i]$, $\text{DirectRefLayerIdx}[i][d]$, $\text{NumRefLayers}[i]$, $\text{RefLayerIdx}[i][r]$, và $\text{LayerUsedAsRefLayerFlag}[j]$ được thay đổi như sau:

Các biến số $\text{dependencyFlag}[i][j]$, $\text{NumDirectRefLayers}[i]$, $\text{DirectRefLayerIdx}[i][d]$, $\text{NumRefLayers}[i]$, $\text{RefLayerIdx}[i][r]$, và $\text{LayerUsedAsRefLayerFlag}[j]$ được suy ra như sau:

```

for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0; j <= vps_max_layers_minus1; j++ ) {
        dependencyFlag[ i ][ j ] = vps_direct_ref_layer_flag[ i ][ j ]
        for( k = j+1; k < i; k++ )
            if( vps_direct_ref_layer_flag[ i ][ k ] &&
                dependencyFlag[ k ][ j ] )
                dependencyFlag[ i ][ j ] = 1
    }
    LayerUsedAsRefLayerFlag[ i ] = 0
}
for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0, d = 0, r = 0; j <= vps_max_layers_minus1; j++ ) {
(37)
        if( vps_direct_ref_layer_flag[ i ][ j ] ) {
            DirectRefLayerIdx[ i ][ d++ ] = j
            LayerUsedAsRefLayerFlag[ j ] = 1

```

```

    }

    if( dependencyFlag[ i ][ j ] )

        RefLayerIdx[ i ][ r++ ] = j

    }

    NumDirectRefLayers[ i ] = d

    NumRefLayers[ i ] = r

    }

```

Chỉ vòng lặp `for( k = 0; k < i; k++ )` được thay đổi thành `for( k = j+1; k < i; k++ )`.

Phương án thực hiện 8

Theo cách tương ứng, các ngữ nghĩa của `vps_direct_ref_layer_flag[i][j]` được thay đổi như sau:

`vps_direct_ref_layer_flag` bằng 0 định rõ rằng lớp với chỉ số `j` không là lớp tham chiếu trực tiếp đối với lớp với chỉ số `i`. `vps_direct_ref_layer_flag[ i ][ j ]` bằng 1 định rõ rằng lớp với chỉ số `j` là lớp tham chiếu trực tiếp đối với lớp với chỉ số `i`. Khi `vps_direct_ref_layer_flag[ i ][ j ]` không có mặt đối với `i` nằm trong khoảng từ 1 đến `vps_max_layers_minus1` và đối với `j` nằm trong khoảng từ 0 đến `i-1`, đều tuyến không loại, nó được suy ra là bằng 0. Khi `vps_independent_layer_flag[ i ]` bằng 0, sẽ có ít nhất một giá trị của `j` nằm trong khoảng từ 0 đến `i - 1`, tuyến không loại, sao cho giá trị của `vps_direct_ref_layer_flag[ i ][ j ]` bằng 1.

Có một số phần tử cú pháp biến số khác biểu thị sự phụ thuộc tương tự với `vps_direct_ref_layer_flag[i][j]`, phạm vi của chúng sẽ được thay đổi theo cách tương tự.

Theo phần mô tả nêu trên phương pháp giải mã dòng bit video và phương pháp mã hóa dòng bit video được đưa ra trong bản mô tả này. Theo cách tương ứng, thiết bị để giải mã dòng bit video (được tạo mã) và thiết bị để giải mã dòng bit video được đưa ra trong bản mô tả này.

FIG.14 minh họa phương pháp giải mã dòng bit video được thực hiện bởi thiết bị giải mã, trong đó tập thông số chuỗi, SPS, được tạo mã trong dòng bit video và chứa các phần tử cú pháp mà áp dụng cho chuỗi video. Phương pháp bao gồm bước thu được 1410 (ví dụ, bằng cách phân tích dòng bit) giá trị của thành phần cú pháp thứ nhất (ví dụ, `sps_ptl_dpb_hrd_params_present_flag` theo phần mô tả nêu trên) từ SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS hay không. Phương pháp này còn bao gồm bước thu được 1420 (ví dụ, bằng cách phân tích dòng bit) giá trị của thành phần cú pháp thứ hai (ví dụ, `sps_sublayer_dpb_params_flag` theo phần mô tả nêu trên) từ SPS, ít nhất khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB (ví dụ, một trong số `max_dec_pic_buffering_minus1[ i ]`, `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` theo phần mô tả chi tiết nêu trên) trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

Lưu ý rằng dòng bit có thể thu được nhờ mạng không dây hoặc mạng có dây. Dòng bit có thể được truyền từ vị trí web, máy chủ, hoặc nguồn ở xa khác nhờ sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường thuê bao số (DSL), hoặc các kỹ thuật vô tuyến như hồng ngoại, radiô, vi sóng, WIFI, Bluetooth, LTE hoặc 5G.

Dòng bit có thể là một chuỗi bit, ở dạng dòng đơn vị lớp trừu tượng mạng (network abstraction layer, NAL) hoặc dòng bai (byte), mà tạo thành sự biểu diễn một chuỗi các đơn vị truy cập (access unit, AU) tạo thành một hoặc nhiều các chuỗi video được tạo mã (coded video sequence, CVS).

Theo một ví dụ cụ thể, các định dạng dòng bit quy định mối quan hệ giữa dòng đơn vị lớp trừu tượng mạng (NAL) và dòng bai, một trong số đó được gọi là dòng bit.

Dòng bit có thể có một trong số hai định dạng: định dạng dòng đơn vị NAL hoặc định dạng dòng bai. Định dạng dòng đơn vị NAL về mặt khái niệm là loại "cơ bản" hơn. Định dạng dòng đơn vị NAL bao gồm chuỗi của các cấu trúc cú pháp được gọi là các đơn vị NAL. Chuỗi này được sắp xếp theo thứ tự giải mã. Có các ràng buộc gập phải về thứ tự giải mã (và các nội dung) của các đơn vị NAL trong dòng đơn vị NAL.

Định dạng dòng bai có thể được xây dựng từ định dạng dòng đơn vị NAL bằng cách sắp xếp các đơn vị NAL theo thứ tự giải mã và đặt ở trước mỗi đơn vị NAL một tiền tố mã bắt đầu và các bai có giá trị không (zero) hoặc lớn hơn không (zero) để tạo thành dòng của các bai. Định dạng dòng đơn vị NAL có thể được trích xuất từ định dạng dòng bai bằng cách tìm kiếm vị trí của mẫu hình tiền tố mã bắt đầu duy nhất trong dòng của các bai này.

FIG.15 minh họa phương pháp mã hóa dòng bit video được thực hiện bởi thiết bị mã hóa, trong đó tập thông số chuỗi, SPS, được mã hóa trong dòng bit video và chứa các phần tử cú pháp mà áp dụng cho chuỗi video. Phương pháp bao gồm bước xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) trong SPS 1510. Giá trị của thành phần cú pháp thứ nhất (ví dụ, `sps_ptl_dpb_hrd_params_present_flag` theo phần mô tả nêu trên) được mã hóa 1520 vào trong SPS dựa trên bước xác định sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không. Ngoài ra, phương pháp được minh họa trên FIG.15 bao gồm xác định 1530 sự có mặt của thành phần cú pháp DPB (ví dụ, một của `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và `max_latency_increase_plus1[i]` theo phần mô tả chi

tiết nêu trên) trong cấu trúc cú pháp các thông số DPB, khi xác định (được xác định), ví dụ, chỉ khi xác định được rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS ở bước 1510, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video. Giá trị của thành phần cú pháp thứ hai (ví dụ, `sps_sublayer_dpb_params_flag` theo phần mô tả nêu trên) được mã hóa 1540 vào trong SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

Các phương pháp được mô tả trên đây có thể là được cấy trong thiết bị giải mã video hoặc thiết bị mã hóa video (tạo ra dòng bit), một cách tương ứng, như đã được mô tả dưới đây.

Như được thể hiện trên FIG.16 thiết bị giải mã video 1600 được đề xuất ở đây theo một phương án thực hiện bao gồm cụm thu được 1610 (ví dụ, bao gồm bộ phân tách) và cụm xác định 1620. Như được thể hiện trên FIG.17 thiết bị mã hóa video 1700 được đề xuất ở đây theo một phương án thực hiện bao gồm cụm xác định 1710 và cụm mã hóa 1720.

Cụm thu được 1610 có trong thiết bị giải mã video 1600 được thể hiện trên FIG.16 được cấu hình để thu được giá trị của thành phần cú pháp thứ nhất (ví dụ, `sps_ptl_dpb_hrd_params_present_flag` theo phần mô tả nêu trên) từ SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) có mặt trong SPS được tạo mã trong dòng bit video hay không. Cụm xác định 1620 có trong thiết bị giải mã video 1600 được thể hiện trên FIG.16 được tạo cấu hình để xác định xem giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không. Ngoài ra, cụm thu được 1610 được cấu hình để thu

được giá trị của thành phần cú pháp thứ hai (ví dụ, `sps_sublayer_dpb_params_flag` theo phần mô tả nêu trên) từ SPS, ít nhất khi xác định (được xác định) rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB (ví dụ, một trong số `max_dec_pic_buffering_minus1[ i ]`, `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` theo phần mô tả chi tiết nêu trên) trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

Cụm xác định 1710 có trong thiết bị mã hóa video 1700 được thể hiện trên FIG.17 được tạo cấu hình để xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) trong SPS; Cụm mã hóa 1720 có trong thiết bị mã hóa video 1700 được thể hiện trên FIG.17 được tạo cấu hình để mã hóa giá trị của thành phần cú pháp thứ nhất (ví dụ, `sps_ptl_dpb_hrd_params_present_flag` theo phần mô tả nêu trên) vào trong SPS dựa trên bước xác định của sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không. Ngoài ra, cụm xác định 1710 được tạo cấu hình để xác định sự có mặt của thành phần cú pháp DPB (ví dụ, một của `max_dec_pic_buffering_minus1[ i ]`, `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` theo phần mô tả chi tiết nêu trên) trong cấu trúc cú pháp các thông số DPB, khi xác định (được xác định), ví dụ, chỉ khi xác định được rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video. Ngoài ra, cụm mã hóa 1720 được tạo cấu hình để mã hóa giá trị của phần tử cú pháp thứ hai (ví dụ, `sps_sublayer_dpb_params_flag` theo phần mô tả nêu

trên) vào trong SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

Thiết bị giải mã video 1600 được thể hiện trên FIG.16 có thể là hoặc có thể được chứa bởi bộ giải mã 30 được thể hiện trên FIG.1A , 1B, và 3 và bộ giải mã video 3206 được thể hiện trên FIG.9. Ngoài ra, thiết bị giải mã 1700 có thể được chứa bởi thiết bị mã hóa video 400 được thể hiện trên FIG.4, thiết bị 500 được thể hiện trên FIG.5 và thiết bị đầu cuối 3106 được thể hiện trên FIG.8. Thiết bị mã hóa 1700 được thể hiện trên FIG.17 có thể là hoặc có thể được chứa bởi bộ mã hóa 20 được thể hiện trên FIG.1A, Fig.1B và Fig.3). Ngoài ra, thiết bị mã hóa 1700 có thể được chứa bởi thiết bị mã hóa video 400 được thể hiện trên FIG.4, thiết bị 500 được thể hiện trên FIG.5 và thiết bị chụp 3102 được thể hiện trên FIG.8.

Cụ thể là, SPS bao gồm thông tin về việc báo hiệu ảnh con.

Một số phần của bảng sau đây thể hiện ảnh chụp của một phần của báo hiệu ảnh con trong SPS trong ITU JVET-Q2001-v13, với liên kết tải xuống như sau:

<http://phenix.it->

sudparis.eu/jvet/doc_end_user/documents/17_Brussels/wg11/JVET-Q2001-v13.zip.

Trong phần còn lại của ứng dụng tài liệu này sẽ được gọi là bản thảo VVC 8 cho đơn giản.

<code>seq_parameter_set_rbsp() {</code>	Bộ mô tả
<code>sps_seq_parameter_set_id</code>	u(4)
<code>sps_video_parameter_set_id</code>	u(4)

sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)
sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	

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

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

- `subpic_ctu_top_left_x[ i ]`, biểu thị thành phần nằm ngang của tọa độ phía trên bên trái của ảnh con i trong ảnh; hoặc
- `subpic_ctu_top_left_y[ i ]` biểu thị thành phần thẳng đứng của tọa độ phía trên bên trái của ảnh con i trong ảnh; hoặc
- `subpic_width_minus1[ i ]` biểu thị chiều rộng của ảnh con i trong ảnh; hoặc
- `subpic_height_minus1[ i ]` biểu thị chiều cao của ảnh con i trong ảnh.

Một số phần tử cú pháp chỉ báo số lượng các ảnh con bên trong ảnh, ví dụ `sps_num_subpics_minus1`.

Phân chia của ảnh thành các CTU, các lát cắt, các ảnh xếp lớp và các ảnh con

Phân chia ảnh thành các CTU

Ảnh được chia thành chuỗi các đơn vị cây tạo mã (coding tree unit, CTU). Thuật ngữ CTU đôi khi được sử dụng hoán đổi cho nhau như CTB (coding tree block: khối cây tạo mã). Theo một ví dụ, thuật ngữ CTU là giống với định nghĩa CTU trong ITU-T H.265. Đối với ảnh mà có ba mảng mẫu, CTU bao gồm $N \times N$ khối của các mẫu luma, và hai khối tương ứng của các mẫu chroma. FIG.6 thể hiện một ví dụ về ảnh được chia thành các CTU. Kích thước của các CTU phải giống nhau ngoại trừ các CTU được bố trí ở các đường bao ảnh (trong đó các CTU chưa hoàn thành có thể có mặt).

Phân chia ảnh thành các ảnh xếp lớp

Theo một số ví dụ, khi các ảnh xếp lớp được phép, ảnh được chia thành các nhóm được tạo hình chữ nhật của các CTU được tách riêng bởi các đường bao thẳng đứng và/hoặc nằm ngang. Các đường bao ảnh xếp lớp thẳng đứng và nằm ngang giao nhau với ảnh từ đường bao ảnh phía dưới và bên trái đến đường bao ảnh bên phải một

cách tương ứng. Các chỉ báo thông tin liên quan đến vị trí của các đường bao ảnh xếp lớp nằm ngang và các đường bao ảnh xếp lớp thẳng đứng được tạo mã trong dòng bit.

FIG.7 minh họa việc phân chia ảnh thành 9 các ảnh xếp lớp. Trong ví dụ này, các đường bao ảnh xếp lớp được đánh dấu với các đường nét đứt đậm.

Khi có nhiều hơn 1 sự chia ảnh xếp lớp theo phương thẳng đứng, thứ tự quét của các CTU được thay đổi đối với thứ tự khung quét của các CTU trong ảnh. Các CTU được quét theo quy tắc sau đây:

1. Các ảnh xếp lớp được quét từ trái sang phải và từ phía trên xuống phía dưới theo thứ tự khung quét, mà được gọi là thứ tự quét ảnh xếp lớp. Điều này có nghĩa là bắt đầu từ phía trên bên trái ảnh xếp lớp, trước tiên tất cả các ảnh xếp lớp trong hàng ảnh xếp lớp giống nhau được quét từ trái sang phải. Sau đó bắt đầu với ảnh xếp lớp thứ nhất trong hàng ảnh xếp lớp thứ hai (hàng ảnh xếp lớp mà ở ngay bên dưới), tất cả các ảnh xếp lớp trong hàng ảnh xếp lớp thứ hai được quét từ trái sang phải. Quy trình được lặp lại cho tới khi tất cả các ảnh xếp lớp được quét.

2. Đối với ảnh xếp lớp, các CTU trong ảnh xếp lớp này được quét theo thứ tự khung quét. Đối với mỗi hàng CTU, các CTU được quét từ trái sang phải và các hàng CTU được quét từ phía trên xuống phía dưới. FIG.7 minh họa thứ tự quét của các CTU trong các ảnh xếp lớp, số lượng tương ứng với các CTU chỉ báo thứ tự quét.

Ảnh xếp lớp cung cấp sự phân chia ảnh theo cách sao cho mỗi ảnh xếp lớp có thể giải mã một cách độc lập từ các ảnh xếp lớp khác của ảnh giống nhau, trong đó bước giải mã đề cập đến giải mã entropy, giải mã dư, và giải mã dự đoán. Ngoài ra với các ảnh xếp lớp có thể phân chia ảnh thành các vùng có kích thước tương tự. Do đó có thể xử lý các ảnh xếp lớp của ảnh song song với nhau, vốn là ưu tiên cho các môi trường xử lý đa lõi trong đó mỗi lõi xử lý là giống nhau.

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

Việc xử lý đề cập đến mã hóa hoặc giải mã các CTU trong bộ mã hóa hoặc bộ giải mã. Thứ tự quét đề cập đến chỉ mục của sự phân chia cụ thể trong ảnh. Các CTU được tạo chỉ mục khi tăng thứ tự trong ảnh theo thứ tự quét được định rõ. Thứ tự quét CTU trong ảnh xếp lớp có nghĩa là cách các CTU bên trong ảnh xếp lớp được tạo chỉ mục, vốn có thể không là thứ tự giống nhau mà theo đó chúng được xử lý (tức là thứ tự xử lý).

Phân chia ảnh thành các lát cắt

Khái niệm lát cắt cung cấp sự phân chia ảnh theo cách sao cho mỗi lát cắt có thể giải mã một cách độc lập từ các lát cắt khác của ảnh giống nhau, trong đó việc giải mã đề cập đến giải mã entropy, dư và dự đoán. Sự khác biệt đối với các ảnh xếp lớp là các lát cắt có thể có các hình dạng tùy ý mà không nhất định là chữ nhật (linh hoạt hơn trong các khả năng phân chia), và mục đích của việc phân chia lát cắt là xử lý không song song nhưng kích thước gói đối chiếu trong các môi trường truyền và sự phục hồi lỗi.

Lát cắt có thể bao gồm ảnh hoàn thành hoặc bao gồm một phần của ảnh. Trong ITU-T H.265, lát cắt bao gồm các CTU liên tiếp của ảnh theo thứ tự xử lý. Lát cắt được xác định theo địa chỉ CTU bắt đầu, địa chỉ CTU bắt đầu được báo hiệu trong tiêu đề lát cắt hoặc trong tập thông số ảnh hoặc trong cụm khác. Theo một ví dụ lát cắt có thể được xác định theo địa chỉ ảnh xếp lớp bắt đầu khi các lát cắt cần bao gồm số nguyên các ảnh xếp lớp.

Trong bản thảo 8 của VVC, lát cắt bao gồm số nguyên các ảnh xếp lớp hoặc số nguyên liên tiếp các hàng CTU bên trong ảnh xếp lớp của ảnh. Do đó, đường bao lát cắt thẳng đứng cũng là đường bao ảnh xếp lớp thẳng đứng. Có thể có trường hợp trong đó đường bao nằm ngang của lát cắt không là đường bao ảnh xếp lớp, các đường bao CTU nằm ngang có thể được chứa bên trong ảnh xếp lớp; theo một ví dụ, khi ảnh

xếp lớp được chia thành nhiều lát cắt hình chữ nhật, mỗi lát cắt bao gồm số nguyên các hàng CTU hoàn thành liên tiếp bên trong ảnh xếp lớp.

Theo một số ví dụ, có hai chế độ lát cắt, chế độ lát cắt khung quét và chế độ lát cắt hình chữ nhật. Trong chế độ lát quét kiểu màn hình, lát bao gồm một chuỗi các ảnh xếp lớp trong việc quét kiểu màn hình ảnh xếp lớp của hình ảnh. Trong chế độ lát cắt hình chữ nhật, lát cắt bao gồm một số lượng ảnh xếp lớp mà tạo ra theo dạng chung vùng hình chữ nhật của ảnh, hoặc lát cắt bao gồm số lượng các hàng CTU liên tiếp của một ảnh xếp lớp mà tạo ra theo dạng chung vùng hình chữ nhật của ảnh. Các ảnh xếp lớp trong lát hình chữ nhật được quét theo thứ tự quét kiểu màn hình ảnh xếp lớp trong vùng hình chữ nhật tương ứng với lát đó.

Tất cả các lát cắt của ảnh tạo ra theo dạng chung toàn bộ ảnh, tức là tất cả các CTU của ảnh có trong một trong số các lát cắt của ảnh. Tương tự các quy tắc áp dụng cho các ảnh xếp lớp và các ảnh con.

Phân chia của ảnh thành các ảnh con

Ảnh con có thể là sự phân chia hình chữ nhật của ảnh. Ảnh con có thể toàn bộ ảnh hoặc một phần của ảnh. Ảnh con là sự phân chia của ảnh theo cách sao cho mỗi ảnh con có thể giải mã một cách độc lập từ các ảnh con khác của toàn bộ chuỗi video. Trong bản thảo VVC 8, khi `subpic_treated_as_pic_flag[i]` chỉ báo là đúng (true) (ví dụ giá trị của `subpic_treated_as_pic_flag[i]` bằng 1) đối với ảnh con *i*, mà ảnh con *i* có thể giải mã một cách độc lập từ các ảnh con khác của toàn bộ chuỗi video.

Sự khác biệt giữa ảnh con và các ảnh xếp lớp hoặc các lát cắt là, các ảnh con tạo ra chuỗi video có thể giải mã một cách độc lập. Đối với các ảnh xếp lớp và các lát cắt, việc giải mã độc lập được thực hiện trong một ảnh của chuỗi video.

Trong bản thảo VVC 8, ảnh con bao gồm một hoặc nhiều lát cắt mà che phủ chung vùng hình chữ nhật của ảnh. Do đó, mỗi đường bao ảnh con luôn là đường bao

lát cắt, và mỗi đường bao ảnh con thẳng đứng luôn là đường bao ảnh xếp lớp thẳng đứng.

FIG.10 cung cấp một ví dụ về các ảnh xếp lớp, các lát cắt và các ảnh con.

Theo một ví dụ như được thể hiện trên FIG.8, ảnh được phân chia thành 216 CTU, 4 ảnh xếp lớp, 4 lát cắt và 3 ảnh con. Giá trị của `sps_num_subpics_minus1` bằng 2, và các phần tử cú pháp liên quan đến vị trí có các giá trị sau đây:

Đối với ảnh con 0

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

Đối với ảnh con 1

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

Đối với ảnh con 2

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

Báo hiệu thông tin bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer).

Bộ đệm ảnh giải mã

Bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) là bộ đệm mà được sử dụng để lưu trữ ảnh được giải mã để tham khảo, ví dụ, làm ảnh tham chiếu cho sự dự đoán liên ảnh. Theo một ví dụ được bộc lộ trong bản thảo VVC (ví dụ ITU JVET-Q2001-v13), các phần tử cú pháp liên quan của các thông số cho DBP trong tập thông số chuỗi (SPS) được làm nổi bật.

Bảng 6: Các phần tử cú pháp liên quan tới ảnh được giải mã trong SPS

sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_max_sublayers_minus1 > 0)	
<u>sps_sublayer_dpb_params_flag</u>	<u>u(1)</u>
<u>if(sps_ptl_dpb_hrd_params_present_flag)</u>	
<u>dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)</u>	
long_term_ref_pics_flag	u(1)

sps_max_sublayers_minus1 plus 1 chỉ định số lớn nhất của các lớp con thời gian mà có thể có mặt trong mỗi CLVS tham chiếu đến SPS. Giá trị của

`sps_max_sublayers_minus1` sẽ nằm trong khoảng 0 đến `vps_max_sublayers_minus1`, tuyến không loại.

`sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của `max_dec_pic_buffering_minus1[ i ]`, các phần tử cú pháp `max_num_reorder_pics[ i ]`, và `max_latency_increase_plus1[ i ]` trong cấu trúc cú pháp `dpb_parameters( )` trong SPS. Khi không có mặt, giá trị của `sps_sub_dpb_params_info_present_flag` được suy ra là bằng 0.

`sps_ptl_dpb_hrd_params_present_flag` bằng 1 chỉ định rằng cấu trúc cú pháp `profile_tier_level( )` và cấu trúc cú pháp `dpb_parameters( )` có mặt trong SPS, và cấu trúc cú pháp `general_hrd_parameters( )` và cấu trúc cú pháp `ols_hrd_parameters( )` cũng có thể có mặt trong SPS. `sps_ptl_dpb_hrd_params_present_flag` bằng 0 định rõ rằng không có cấu trúc nào trong số bốn cấu trúc cú pháp này có mặt trong SPS. Giá trị của `sps_ptl_dpb_hrd_params_present_flag` sẽ bằng `vps_independent_layer_flag[ GeneralLayerIdx[ nuh_layer_id ] ]`.

Thành phần cú pháp `sps_max_sublayers_minus1` chỉ báo số lượng các lớp con thời gian khả dụng. Khi số lượng các lớp con thời gian khả dụng lớn hơn 1 (ví dụ giá trị của thành phần cú pháp `sps_max_sublayers_minus1` lớn hơn 0), giá trị của thành phần cú pháp `sps_sublayer_dpb_params_flag` được báo hiệu trong dòng bit. Do Thành phần cú pháp `sps_sublayer_dpb_params_flag` bắt buộc báo hiệu thông tin giải mã ảnh cho mỗi khả dụng lớp con (khi của nó giá trị bằng 1) hoặc chỉ báo hiệu cho lớp con thời gian cao nhất (khi của nó giá trị bằng 0). Khi giá trị của `sps_sublayer_dpb_params_flag` không được thể hiện, ví dụ chỉ có một lớp con thời gian (giá trị của `sps_max_sublayers_minus1` == 0), giá trị của `sps_max_sublayers_minus1` được suy ra là 0.

Theo một số ví dụ, giá trị của thành phần cú pháp `sps_ptl_dpb_hrd_params_present_flag` chỉ báo liệu cấu trúc báo hiệu `dpb_parameter()`

có được báo hiệu trong SPS hay không. Khi giá trị của thành phần cú pháp `sps_ptl_dpb_hrd_params_present_flag` bằng 1, cấu trúc ngay được báo hiệu `dpb_parameter( )` được gọi, với số lượng các lớp con thời gian khả dụng trừ đi 1 (`sps_max_sublayers_minus1`) và cờ `sps_sublayer_dpb_params_flag` làm thông số thứ nhất và thứ hai, một cách tương ứng.

Theo một ví dụ, cấu trúc báo hiệu của `dpb_parameter()` trong bản thảo VVC (ví dụ ITU JVET-Q2001-v13) được xác định như sau:

Bảng 7: định nghĩa của `dpb_parameters`

<code>dpb_parameters(maxSubLayersMinus1, subLayerInfoFlag) {</code>	Bộ mô tả
<code>for(i = (subLayerInfoFlag ? 0 : maxSubLayersMinus1);</code> <code>i <= maxSubLayersMinus1; i++) {</code>	
<code>max_dec_pic_buffering_minus1[i]</code>	ue(v)
<code>max_num_reorder_pics[i]</code>	ue(v)
<code>max_latency_increase_plus1[i]</code>	ue(v)
<code>}</code>	
<code>}</code>	

Cấu trúc cú pháp `dpb_parameters( )` cung cấp thông tin của kích cỡ DPB, số lượng sắp xếp lại ảnh tối đa, và độ trễ tối đa cho một hoặc nhiều OLS.

(tập hợp lớp đầu ra (OLS): Tập hợp của các lớp bao gồm tập hợp được định rõ của các lớp, trong đó một hoặc nhiều lớp trong tập hợp của các lớp được định rõ là các lớp đầu ra)

Khi cấu trúc cú pháp `dpb_parameters( )` được bao hàm trong VPS, các OLS mà cấu trúc cú pháp `dpb_parameters( )` áp dụng vào đó được định rõ bởi VPS. Khi cấu

trúc cú pháp `dpb_parameters()` được bao hàm trong SPS, nó áp dụng cho OLS mà chỉ bao gồm lớp là lớp thấp nhất trong số các lớp mà viện dẫn đến SPS, và lớp thấp nhất này là lớp độc lập.

`max_dec_pic_buffering_minus1[i]` cộng 1 chỉ định kích cỡ được yêu cầu tối đa của DPB trong các cụm của các bộ đệm lưu trữ ảnh khi `Htid` bằng `i`. Giá trị của `max_dec_pic_buffering_minus1[i]` sẽ nằm trong khoảng 0 đến `MaxDpbSize - 1`, tuyến không loại, trong đó `MaxDpbSize` như được mô tả trong mệnh đề A.4.2. Khi `i` lớn hơn 0, `max_dec_pic_buffering_minus1[i]` sẽ lớn hơn hoặc bằng `max_dec_pic_buffering_minus1[i - 1]`. Khi `max_dec_pic_buffering_minus1[i]` không có mặt đối với `i` nằm trong khoảng từ 0 đến `maxSubLayersMinus1 - 1`, tuyến không loại, do `subLayerInfoFlag` bằng 0, bằng `max_dec_pic_buffering_minus1[maxSubLayersMinus1]`.

`max_num_reorder_pics[i]` chỉ định số lượng các ảnh được phép tối đa của OLS mà có thể ở trước bất kỳ ảnh trong OLS theo thứ tự giải mã và theo sau ảnh đó theo thứ tự xuất khi `Htid` bằng `i`. Giá trị của `max_num_reorder_pics[i]` sẽ nằm trong khoảng 0 đến `max_dec_pic_buffering_minus1[i]`, tuyến không loại. Khi `i` lớn hơn 0, `max_num_reorder_pics[i]` sẽ lớn hơn hoặc bằng `max_num_reorder_pics[i - 1]`. Khi `max_num_reorder_pics[i]` không có mặt đối với `i` nằm trong khoảng từ 0 đến `maxSubLayersMinus1 - 1`, tuyến không loại, do `subLayerInfoFlag` bằng 0, nó được suy ra là bằng `max_num_reorder_pics[maxSubLayersMinus1]`.

`max_latency_increase_plus1[i]` không bằng 0 được sử dụng để tính toán giá trị của `MaxLatencyPictures[i]`, mà chỉ định số lớn nhất của các ảnh trong OLS mà có thể ở trước bất kỳ ảnh trong OLS theo thứ tự xuất và theo sau ảnh đó theo thứ tự giải mã khi `Htid` bằng `i`.

Khi `max_latency_increase_plus1[i]` không bằng 0, giá trị của `MaxLatencyPictures[i]` được chỉ định rõ như sau:

$$\text{MaxLatencyPictures}[i] = \text{max_num_reorder_pics}[i] + \text{max_latency_increase_plus1}[i] - 1 \quad (7-111)$$

Khi $\text{max_latency_increase_plus1}[i]$ bằng 0, không có giới hạn tương ứng được trình bày.

Giá trị của $\text{max_latency_increase_plus1}[i]$ sẽ nằm trong khoảng 0 đến $232 - 2$, tuyến không loại. Khi $\text{max_latency_increase_plus1}[i]$ không có mặt đối với i nằm trong khoảng từ 0 đến $\text{maxSubLayersMinus1} - 1$, tuyến không loại, do subLayerInfoFlag bằng 0, nó được suy ra là bằng $\text{max_latency_increase_plus1}[\text{maxSubLayersMinus1}]$.

Cấu trúc báo hiệu dpb_parameters hoặc báo hiệu một thông tin bộ đệm ảnh đã được giải mã của lớp con hoặc mỗi thông tin ảnh được giải mã của lớp con, được điều khiển bởi giá trị của subLayerInfoFlag . Khi giá trị của subLayerInfoFlag bằng 0, bộ đệm ảnh đã được giải mã thông tin của lớp con cao nhất trong số các lớp con thời gian khả dụng được báo hiệu ($i = \text{maxSubLayersMinus1}$). Khi giá trị của subLayerInfoFlag bằng 1, bộ đệm ảnh đã được giải mã thông tin của mỗi lớp con trong số các lớp con thời gian khả dụng được báo hiệu (giá trị của i nằm trong khoảng từ 0 đến $\text{maxSubLayersMinus1}$, tuyến không loại).

Theo một số ví dụ, thành phần cú pháp $\text{sps_sublayer_dpb_params_flag}$ chỉ được sử dụng như thông số thứ hai của dpb_parameters .

Phương án 10

Theo phương án thực hiện thứ mười, giá trị của thành phần cú pháp $\text{sps_sublayer_dpb_params_flag}$ được tạo mã trong dòng bit, dựa trên giá trị của thành phần cú pháp $\text{sps_ptl_dpb_hrd_params_present_flag}$ như sau:

Bảng 8: Các phần tử cú pháp trong SPS

sps_poc_msb_flag	$u(1)$
--	--------------------------

if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_max_sublayers_minus1 > 0 && sps_ptl_dpb_hrd_params_present_flag)	
sps_sublayer_dpb_params_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
long_term_ref_pics_flag	u(1)

Theo phương án này, khi giá trị của sps_max_sublayers_minus1 lớn hơn 0 và giá trị của sps_ptl_dpb_hrd_params_present_flag bằng 0, giá trị của thành phần cú pháp sps_sublayer_dpb_params_flag không cần được tạo mã trong dòng bit.

Phương án 11

Theo phương án thực hiện thứ mười một, giá trị của thành phần cú pháp sps_sublayer_dpb_params_flag được tạo mã trong dòng bit, dựa trên giá trị của thành phần cú pháp sps_ptl_dpb_hrd_params_present_flag như sau:

Bảng 9: Các phần tử cú pháp trong PPS

sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_ptl_dpb_hrd_params_present_flag){	
if (sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
}	
long_term_ref_pics_flag	u(1)

Theo phương án này, khi giá trị của sps_max_sublayers_minus1 lớn hơn 0 và giá trị của sps_ptl_dpb_hrd_params_present_flag bằng 0, giá trị của thành phần cú pháp sps_sublayer_dpb_params_flag không cần được tạo mã trong dòng bit. Ngoài ra, các phần tử cú pháp được tái cấu trúc với thiết kế rõ ràng hơn. Thành phần cú pháp sps_ptl_dpb_hrd_params_present_flag được nâng lên để bắt buộc báo hiệu của cả thành phần cú pháp sps_sublayer_dpb_params_flag và cấu trúc báo hiệu dpb_parameters().

Phương án 12

Theo phương án thực hiện thứ mười hai, giá trị của thành phần cú pháp `sps_sublayer_dpb_params_flag` được tạo mã trong dòng bit, dựa trên giá trị của thành phần cú pháp `sps_ptl_dpb_hrd_params_present_flag` như sau:

Bảng 10: Các phần tử cú pháp trong PPS

sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_ptl_dpb_hrd_params_present_flag){	
if (sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
firstSubLayer = sps_sublayer_dpb_params_flag ? 0 : sps_max_sublayers_minus1	
dpb_parameters(firstSubLayer , sps_max_sublayers_minus1)	
}	
long_term_ref_pics_flag	u(1)

Theo cách tương ứng, cấu trúc báo hiệu của `dpb_parameters` được hiệu chỉnh như sau:

<code>dpb_parameters(firstSubLayer, maxSubLayersMinus1) {</code>	Bộ mô tả
<code>for(i = firstSubLayer; i <= maxSubLayersMinus1; i++) {</code>	
<code> max_dec_pic_buffering_minus1[i]</code>	ue(v)
<code> max_num_reorder_pics[i]</code>	ue(v)
<code> max_latency_increase_plus1[i]</code>	ue(v)
<code>}</code>	
<code>}</code>	

Theo phương án này, khi giá trị của `sps_max_sublayers_minus1` lớn hơn 0 và giá trị của `sps_ptl_dpb_hrd_params_present_flag` bằng 0, giá trị của thành phần cú pháp `sps_sublayer_dpb_params_flag` không cần được tạo mã trong dòng bit. So với phương án thực hiện 11, cấu trúc báo hiệu của `dpb_parameter()` được báo hiệu từ lớp con thứ nhất đến lớp con cao nhất, chỉ báo bởi thông số `firstSubLayer` và `maxSubLayersMinus1`, một cách tương ứng. Việc chọn lớp con thứ nhất là đầu ra của `dpb_parameter()`. Các ngữ nghĩa của `sps_sublayer_dpb_params_flag` là rõ ràng hơn theo cách này.

Theo một ví dụ, như được bộc lộ trong bản thảo VVC (ví dụ ITU JVET-Q2001-v13), có hai chế độ cho lát cắt, chế độ lát cắt khung quét và chế độ lát cắt hình chữ nhật. Tuy nhiên, theo định nghĩa hiện tại, lát cắt có thể là hoặc trong chế độ khung quét hoặc trong chế độ hình chữ nhật.

Định nghĩa của các lát cắt được mô tả như sau:

Lát bao gồm số lượng nguyên của các ảnh xếp lớp hoàn chỉnh hoặc số lượng nguyên của các hàng CTU hoàn chỉnh liên tiếp trong ảnh xếp lớp của hình ảnh. Do đó,

mỗi đường bao lát cắt thẳng đứng cũng luôn là đường bao ảnh xếp lớp thẳng đứng. Có thể có trường hợp là đường bao nằm ngang của lát cắt không là đường bao ảnh xếp lớp nhưng bao gồm các đường bao CTU nằm ngang bên trong ảnh xếp lớp; điều này xảy ra khi ảnh xếp lớp chia thành nhiều lát cắt hình chữ nhật, mỗi trong số mà bao gồm số nguyên các hàng CTU hoàn thành liên tiếp bên trong ảnh xếp lớp.

Hai chế độ của các lát được hỗ trợ, cụ thể là chế độ lát quét kiểu mảnh và chế độ lát hình chữ nhật. Trong chế độ lát quét kiểu mảnh, lát chứa một chuỗi các ảnh xếp lớp hoàn chỉnh trong việc quét kiểu mảnh ảnh xếp lớp của hình ảnh. Trong chế độ lát hình chữ nhật, lát chứa số lượng ảnh xếp lớp hoàn chỉnh tạo chung thành vùng hình chữ nhật của hình ảnh hoặc số lượng hàng CTU hoàn chỉnh liên tiếp của một ảnh xếp lớp mà tạo chung thành vùng hình chữ nhật của hình ảnh. Các ảnh xếp lớp trong lát hình chữ nhật được quét theo thứ tự quét kiểu mảnh ảnh xếp lớp trong vùng hình chữ nhật tương ứng với lát đó.

Ràng buộc duy nhất đối với lát cắt khung quét là nó “chứa chuỗi của các ảnh xếp lớp hoàn thành trong ảnh xếp lớp khung quét của ảnh”. Như được thể hiện trên FIG.12, ví dụ, chỉ có hai ảnh xếp lớp và một lát cắt trong ảnh. Lát cắt này có thể là hoặc trong chế độ lát cắt khung quét hoặc chế độ lát cắt hình chữ nhật. Điều này có thể gây ra rối loạn cho hai chế độ của lát cắt.

Phương án thực hiện 13

Có đề xuất tiếp tục giới hạn định nghĩa của lát cắt khung quét như sau:

Trong chế độ lát cắt khung quét, lát cắt chứa chuỗi của các ảnh xếp lớp hoàn thành trong ảnh xếp lớp khung quét của ảnh và ảnh này chứa ít nhất một lát cắt tạo ra hình dạng không là hình chữ nhật.

Ví dụ, với định nghĩa này của lát cắt khung quét, ảnh như được thể hiện trên FIG.13 là lát cắt khung quét, nhưng ảnh như được thể hiện trên FIG.12 có thể chỉ sử dụng chế độ lát cắt hình chữ nhật.

Theo cách tương ứng, việc báo hiệu cho ảnh xếp lớp cấu trúc được thay đổi.

Theo một ví dụ, báo hiệu của ảnh xếp lớp thông tin là như sau trong PPS:

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

no_pic_partition_flag bằng 1 chỉ định rằng không có sự phân chia ảnh được áp dụng cho mỗi ảnh tham chiếu đến PPS. no_pic_partition_flag bằng 0 chỉ định mỗi ảnh tham chiếu đến PPS có thể là được phân chia thành nhiều hơn một ảnh xếp lớp hoặc lát cắt.

Có yêu cầu về sự hợp chuẩn dòng bit mà giá trị của no_pic_partition_flag sẽ giống nhau cho tất cả các PPS mà được tham chiếu đến bởi các ảnh được tạo mã trong CLVS.

Có yêu cầu về sự hợp chuẩn dòng bit mà giá trị của no_pic_partition_flag sẽ không bằng 1 khi giá trị của sps_num_subpics_minus1 + 1 lớn hơn 1.

$\text{pps_log2_ctu_size_minus5}$ cộng 5 chỉ định kích cỡ khối cây tạo mã luma của mỗi CTU. $\text{pps_log2_ctu_size_minus5}$ bằng $\text{sps_log2_ctu_size_minus5}$.

$\text{num_exp_tile_columns_minus1}$ cộng 1 chỉ định số lượng của các chiều rộng cột ảnh xếp lớp được cung cấp tường minh. Giá trị của $\text{num_exp_tile_columns_minus1}$ sẽ nằm trong khoảng từ 0 đến $\text{PicWidthInCtbsY} - 1$, tuyến không loại. Khi $\text{no_pic_partition_flag}$ bằng 1, giá trị của $\text{num_exp_tile_columns_minus1}$ được suy ra là bằng 0.

$\text{num_exp_tile_rows_minus1}$ cộng 1 chỉ định số lượng các chiều cao hàng ảnh xếp lớp được cung cấp tường minh. Giá trị của $\text{num_exp_tile_rows_minus1}$ sẽ nằm trong khoảng từ 0 đến $\text{PicHeightInCtbsY} - 1$, tuyến không loại. Khi $\text{no_pic_partition_flag}$ bằng 1, giá trị của $\text{num_tile_rows_minus1}$ được suy ra là bằng 0.

$\text{tile_column_width_minus1}[i]$ cộng 1 chỉ định chiều rộng của thứ i cột ảnh xếp lớp trong các cụm của các CTB đối với i nằm trong khoảng từ 0 đến $\text{num_exp_tile_columns_minus1} - 1$, tuyến không loại. $\text{tile_column_width_minus1}[\text{num_exp_tile_columns_minus1}]$ được sử dụng để suy ra chiều rộng của các cột ảnh xếp lớp với chỉ số lớn hơn hoặc bằng $\text{num_exp_tile_columns_minus1}$ như được mô tả trong mệnh đề 6.5.1. Giá trị của $\text{tile_column_width_minus1}[i]$ sẽ nằm trong khoảng 0 đến $\text{PicWidthInCtbsY} - 1$, tuyến không loại. Khi không có mặt, giá trị của $\text{tile_column_width_minus1}[0]$ được suy ra là bằng $\text{PicWidthInCtbsY} - 1$.

$\text{tile_row_height_minus1}[i]$ cộng 1 chỉ định chiều cao của hàng ảnh xếp lớp thứ i trong các cụm của các CTB đối với i nằm trong khoảng từ 0 đến $\text{num_exp_tile_rows_minus1} - 1$, tuyến không loại. $\text{tile_row_height_minus1}[\text{num_exp_tile_rows_minus1}]$ được sử dụng để suy ra chiều cao của các hàng ảnh xếp lớp với chỉ số lớn hơn hoặc bằng

`num_exp_tile_rows_minus1` như được mô tả trong mệnh đề 6.5.1. Giá trị của `tile_row_height_minus1[ i ]` sẽ nằm trong khoảng 0 đến `PicHeightInCtbsY - 1`, tuyến không loại. Khi không có mặt, giá trị của `tile_row_height_minus1[ 0 ]` được suy ra là bằng `PicHeightInCtbsY - 1`.

`rect_slice_flag` bằng 0 chỉ định rằng các ảnh xếp lớp trong mỗi lát cắt có thứ tự khung quét và thông tin lát cắt không được báo hiệu trong PPS. `rect_slice_flag` bằng 1 chỉ định rằng các ảnh xếp lớp trong mỗi lát cắt bao phủ vùng hình chữ nhật của ảnh và thông tin lát cắt được báo hiệu trong PPS. Khi không có mặt, `rect_slice_flag` được suy ra là bằng 1. Khi `subpic_info_present_flag` bằng 1, giá trị của `rect_slice_flag` sẽ bằng 1.

`single_slice_per_subpic_flag` bằng 1 chỉ định rằng mỗi ảnh con bao gồm một và chỉ một lát cắt hình chữ nhật. `single_slice_per_subpic_flag` bằng 0 chỉ định rằng mỗi ảnh con có thể bao gồm một hoặc nhiều lát cắt hình chữ nhật. Khi `single_slice_per_subpic_flag` bằng 1, `num_slices_in_pic_minus1` được suy ra là bằng `sps_num_subpics_minus1`. Khi không có mặt, giá trị của `single_slice_per_subpic_flag` được suy ra là bằng 0. [Ed. (GJS): Xem xét việc đổi tên cờ này hoặc làm sáng tỏ theo cách khác để tránh sự dự đoán rằng cờ này chỉ liên quan khi có nhiều hơn một ảnh con trong mỗi ảnh.]

Khi số lượng ảnh xếp lớp trong ảnh lớn hơn 1, cờ `rect_slice_flag` được phân tích cú pháp từ dòng bit, giá trị của `rect_slice_flag` biểu thị liệu chế độ lát cắt hình chữ nhật (khi giá trị bằng 1) được sử dụng cho ảnh hoặc chế độ lát cắt khung quét (khi giá trị bằng 0) được sử dụng cho ảnh.

Theo cách tương ứng, trong tiêu đề lát

<code>slice_header() {</code>	Bộ mô tả
--------------------------------	-----------------

picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
if(subpic_info_present_flag)	
slice_subpic_id	u(v)
if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1))	
slice_address	u(v)
for(i = 0; i < NumExtraPhBits; i++)	
sh_extra_bit[i]	u(1)
if(!rect_slice_flag && NumTilesInPic > 1)	
num_tiles_in_slice_minus1	ue(v)

picture_header_in_slice_header_flag bằng 1 định rõ rằng cấu trúc cú pháp PH có mặt trong tiêu đề lát. picture_header_in_slice_header_flag bằng 0 định rõ rằng cấu trúc cú pháp PH không có mặt trong tiêu đề lát.

Có yêu cầu về sự hợp chuẩn dòng bit mà giá trị của picture_header_in_slice_header_flag sẽ giống nhau cho tất cả các lát cắt được tạo mã trong CLVS.

Khi picture_header_in_slice_header_flag bằng 1 đối với lát cắt được tạo mã, nó là yêu cầu của sự hợp chuẩn dòng bit mà không có cụm VCL NAL với nal_unit_type bằng PH_NUT sẽ có mặt trong CLVS.

Khi `picture_header_in_slice_header_flag` bằng 0, tất cả các lát cắt được tạo mã trong ảnh hiện tại sẽ có `picture_header_in_slice_header_flag` bằng 0, và PU hiện tại sẽ có cụm PH NAL.

`slice_subpic_id` chỉ định ID ảnh con của ảnh con mà chứa lát cắt. Nếu `slice_subpic_id` có mặt, giá trị của biến `CurrSubpicIdx` được suy ra sao cho `SubpicIdVal[ CurrSubpicIdx ]` bằng `slice_subpic_id`. Ngược lại (`slice_subpic_id` không thể hiện), `CurrSubpicIdx` được suy là bằng 0. Chiều dài của `slice_subpic_id` là `sps_subpic_id_len_minus1 + 1` bit.

`slice_address` chỉ định địa chỉ lát cắt của lát cắt. Khi không có mặt, giá trị của `slice_address` được suy là bằng 0. Khi `rect_slice_flag` bằng 1 và `NumSlicesInSubpic[ CurrSubpicIdx ]` bằng 1, giá trị của `slice_address` được suy là bằng 0.

Nếu `rect_slice_flag` bằng 0, thì áp dụng nội dung sau đây:

- Địa chỉ lát cắt là chỉ số ảnh xếp lớp khung quét.
- Độ dài của `slice_address` là $\text{Ceil}(\text{Log}_2(\text{NumTilesInPic}))$ bit.
- Giá trị của `slice_address` sẽ nằm trong khoảng 0 đến $\text{NumTilesInPic} - 1$, tuyến không loại.

Ngược lại (`rect_slice_flag` bằng 1), thì áp dụng nội dung sau đây:

- Địa chỉ lát cắt là chỉ số lát cắt mức ảnh con của lát cắt.
- Độ dài của `slice_address` là $\text{Ceil}(\text{Log}_2(\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}]))$ bit.
- Giá trị của `slice_address` sẽ nằm trong khoảng 0 đến $\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}] - 1$, tuyến không loại.

Đây là yêu cầu của sự phù hợp dòng bit mà các ràng buộc sau đây áp dụng:

- Nếu `rect_slice_flag` bằng 0 hoặc `subpic_info_present_flag` bằng 0, giá trị của `slice_address` sẽ không bằng giá trị của `slice_address` của cụm NAL lát cắt được tạo mã khác bất kỳ của ảnh đã được tạo mã giống nhau.
- Theo cách khác/ngược lại, cặp giá trị `slice_subpic_id` và `slice_address` sẽ không bằng cặp giá trị `slice_subpic_id` và `slice_address` của cụm NAL lát cắt được tạo mã bất kỳ khác của cùng ảnh đã được tạo mã.
- Các hình dạng của các lát cắt của ảnh sẽ là các hình dạng sao cho mỗi CTU, khi được giải mã, sẽ có đường bao toàn bộ bên trái và đường bao phía trên toàn bộ của nó bao gồm đường bao ảnh hoặc bao gồm các đường bao của (các) CTU được giải mã trước đó.

`sh_extra_bit[ i ]` có thể là bằng 1 hoặc 0. Các bộ giải mã phù hợp với này phiên bản của bản mô tả này sẽ bỏ qua giá trị của `sh_extra_bit[ i ]`. Giá trị của nó không ảnh hưởng đến sự hợp chuẩn bộ giải mã với các biên dạng được định rõ trong phiên bản này của đặc tả kỹ thuật.

`num_tiles_in_slice_minus1 + 1`, khi có mặt, chỉ định số lượng các ảnh xếp lớp trong lát cắt. Giá trị của `num_tiles_in_slice_minus1` sẽ nằm trong khoảng 0 đến `NumTilesInPic - 1`, tuyến không loại.

Có đề xuất điều chỉnh cơ chế báo hiệu nêu trên như sau trong PPS và tiêu đề lát:

<code>if(!no_pic_partition_flag) {</code>	
<code>pps_log2_ctu_size_minus5</code>	<code>u(2)</code>
<code>num_exp_tile_columns_minus1</code>	<code>ue(v)</code>
<code>num_exp_tile_rows_minus1</code>	<code>ue(v)</code>
<code>for(i = 0; i <= num_exp_tile_columns_minus1; i++)</code>	

tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 3)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)

slice_header() {	Bộ mô tả
picture_header_in_slice_header_flag	u(1)
if(picture_header_in_slice_header_flag)	
picture_header_structure()	
if(subpic_info_present_flag)	
slice_subpic_id	u(v)
if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 3))	
slice_address	u(v)
for(i = 0; i < NumExtraPhBits; i++)	
sh_extra_bit[i]	u(1)
if(!rect_slice_flag && NumTilesInPic > 3)	
num_tiles_in_slice_minus1	ue(v)

Cụ thể là, các phương án sau đây cũng được đề xuất ở đây (trong sự liệt kê mới).

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

thu được dòng bit, tập thông số chuỗi, SPS, được tạo mã trong dòng bit;

thu được giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` theo dòng bit, trong đó giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` được sử dụng để chỉ báo số lớn nhất của các lớp con thời gian mà có mặt trong lớp video sequence được tạo mã, CLVS, tham chiếu đến SPS;

thu được giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` theo dòng bit, trong đó giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` được sử dụng để chỉ báo liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) (ví dụ `dpb_parameters`) có mặt trong SPS hay không;

phân tích giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` từ dòng bit, khi giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` lớn hơn giá trị mặc định thứ nhất (ví dụ giá trị mặc định thứ nhất bằng 0) và khi giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` bằng giá trị mặc định thứ hai (ví dụ giá trị mặc định thứ hai bằng 1), trong đó giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của thành phần cú pháp (ví dụ `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và/hoặc `max_latency_increase_plus1[i]` đối với `i` nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không loại, khi `sps_max_sublayers_minus1` lớn hơn 0) trong cấu trúc cú pháp các thông số DPB (ví dụ `dpb_parameters`) trong SPS.

2. Phương pháp theo phương án thực hiện thứ nhất, trong đó phương pháp này còn bao gồm bước:

thiết lập giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` thành giá trị mặc định thứ ba (ví dụ giá trị mặc định thứ ba bằng 0 hoặc 1), khi giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` nhỏ hơn hoặc bằng giá trị mặc định thứ nhất hoặc khi giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` không bằng giá trị mặc định thứ hai.

3. Phương pháp theo phương án 1 hoặc 2, trong đó giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được tạo mã trong SPS.

4. Phương pháp tạo mã được thực hiện bởi thiết bị giải mã hoặc thiết bị mã hóa, phương pháp này bao gồm các bước:

thu được dòng bit, tập thông số chuỗi, SPS, được tạo mã trong dòng bit;

thu được giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` theo dòng bit, trong đó giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` được sử dụng để chỉ định số lớn nhất của các lớp con thời gian mà có mặt trong chuỗi video lớp được tạo mã, CLVS, tham chiếu đến SPS;

thu được giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` theo dòng bit, trong đó giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` được sử dụng để định rõ liệu cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (DPB: decoded picture buffer) (ví dụ `dpb_parameters`) có mặt trong SPS hay không;

xác định xem giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` có bằng giá trị mặc định thứ hai (ví dụ giá trị mặc định thứ hai bằng 1) hay không;

xác định xem giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` có lớn hơn giá trị mặc định thứ nhất (ví dụ giá trị mặc

định thứ nhất bằng 0) hay không, khi xác định được rằng giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` bằng giá trị mặc định thứ hai;

phân tích giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` từ dòng bit, khi xác định được rằng giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` lớn hơn giá trị mặc định thứ nhất (ví dụ giá trị mặc định thứ nhất bằng 0), trong đó giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được sử dụng để kiểm soát sự có mặt của thành phần cú pháp (ví dụ `max_dec_pic_buffering_minus1[i]`, `max_num_reorder_pics[i]`, và/hoặc `max_latency_increase_plus1[i]` đối với i nằm trong khoảng từ 0 đến `sps_max_sublayers_minus1 - 1`, tuyến không loại, khi `sps_max_sublayers_minus1` lớn hơn 0) trong cấu trúc cú pháp các thông số DPB (ví dụ `dpb_parameters`) trong SPS.

5. Phương pháp theo phương án thực hiện thứ tư, trong đó phương pháp này còn bao gồm bước:

thiết lập giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` thành giá trị mặc định thứ ba (ví dụ giá trị mặc định thứ ba bằng 0 hoặc 1), khi xác định được rằng giá trị của thành phần cú pháp thứ nhất `sps_max_sublayers_minus1` nhỏ hơn hoặc bằng giá trị mặc định thứ nhất hoặc khi xác định được rằng giá trị của thành phần cú pháp thứ hai `sps_ptl_dpb_hrd_params_present_flag` không bằng giá trị mặc định thứ hai.

6. Phương pháp theo phương án 4 hoặc 5, trong đó giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được tạo mã trong SPS (theo các ví dụ khác, giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được tạo mã trong tập thông số ảnh, PPS hoặc giá trị của thành phần cú pháp thứ ba `sps_sublayer_dpb_params_flag` được tạo mã trong tập thông số video, VPS).

7. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 6, trong đó giá trị của thành phần cú pháp thứ nhất được tạo mã trong tập thông số ảnh, PPS

hoặc giá trị của thành phần cú pháp thứ nhất được tạo mã trong tập thông số video, VPS (theo một ví dụ khác, giá trị của thành phần cú pháp thứ hai được tạo mã trong tập thông số ảnh, PPS hoặc giá trị của thành phần cú pháp thứ hai được tạo mã trong tập thông số video, VPS).

8. Phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 7, trong đó khi `sps_max_sublayers_minus1` bằng 0, `max_dec_pic_buffering_minus1[ 0 ]`, `max_num_reorder_pics[ 0 ]`, và `max_latency_increase_plus1[ 0 ]` luôn chỉ được báo hiệu cho lớp con viện dẫn đến SPS.

9. Phương pháp tạo mã được thực hiện bởi thiết bị giải mã hoặc thiết bị mã hóa, phương pháp này bao gồm các bước:

xác định lớp tham chiếu gián tiếp với chỉ số j của lớp với chỉ số i , chỉ từ ít nhất một lớp với chỉ số thấp hơn i ;

khi dự đoán liên lớp được phép đối với ảnh hiện tại, thu được ảnh tham chiếu của nó trong lớp tham chiếu gián tiếp với chỉ số j ,

dự đoán ảnh hiện tại nhờ sử dụng ảnh tham chiếu từ lớp tham chiếu gián tiếp với chỉ số j .

10. Phương pháp theo phương án 9, trong đó bước xác định lớp tham chiếu gián tiếp của lớp với chỉ số i bao gồm: xác định rằng có lớp với chỉ số k , chỉ trong số ít nhất một lớp với chỉ số thấp hơn i và cao hơn j , là lớp tham chiếu trực tiếp với chỉ số i , trong đó lớp tham chiếu gián tiếp với chỉ số j là lớp tham chiếu của lớp tham chiếu trực tiếp với chỉ số k .

11. Phương pháp theo phương án 9, trong đó bước xác định lớp tham chiếu trực tiếp gián tiếp của lớp với chỉ số i bao gồm: xác định rằng có lớp với chỉ số k , chỉ trong số ít nhất một lớp với chỉ số thấp hơn i , là lớp tham chiếu trực tiếp với chỉ số i , trong đó lớp tham chiếu gián tiếp với chỉ số j là lớp tham chiếu của lớp tham chiếu trực tiếp với chỉ số k .

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

xác định lớp tham chiếu gián tiếp với chỉ số j của lớp với chỉ số i , khi xác định được rằng có lớp với chỉ số k , chỉ trong số ít nhất một lớp với chỉ số thấp hơn i và cao hơn j , là lớp tham chiếu trực tiếp của lớp với chỉ số i , và trong đó lớp với chỉ số j là lớp tham chiếu của lớp tham chiếu trực tiếp với chỉ số k .

khi dự đoán liên lớp được phép đối với ảnh hiện tại, thu được ảnh tham chiếu của nó trong lớp tham chiếu gián tiếp với chỉ số j ,

dự đoán ảnh hiện tại nhờ sử dụng ảnh tham chiếu từ lớp tham chiếu gián tiếp với chỉ số j .

13. Phương pháp tạo mã được thực hiện bởi thiết bị giải mã hoặc thiết bị mã hóa, phương pháp này bao gồm các bước:

thu được dòng bit, tập thông số ảnh, PPS, được tạo mã trong dòng bit;

thu được số lượng các cột ảnh xếp lớp NumTileColumns cho ảnh hiện tại (ví dụ ảnh hiện tại chứa ít nhất một lát cắt tạo ra hình dạng không là hình chữ nhật) theo dòng bit;

thu được số lượng các hàng ảnh xếp lớp NumTileRows cho ảnh hiện tại theo dòng bit;

thu được giá trị của biến NumTilesInPic theo số lượng các cột ảnh xếp lớp NumTileColumns và số lượng các hàng ảnh xếp lớp NumTileRows;

phân tích giá trị của thành phần cú pháp rect_slice_flag từ dòng bit, khi giá trị của biến NumTilesInPic lớn hơn giá trị định trước (ví dụ giá trị định trước bằng 3), trong đó giá trị của thành phần cú pháp rect_slice_flag được sử dụng để định rõ xem lát cắt thông tin có được báo hiệu trong PPS hay không.

14. Phương pháp theo phương án 13, trong đó phương pháp còn bao gồm bước:

thiết lập giá trị của thành phần cú pháp `rect_slice_flag` thành giá trị mặc định thứ tư (ví dụ giá trị mặc định thứ tư bằng 1), khi giá trị của biến `NumTilesInPic` nhỏ hơn hoặc bằng giá trị định trước.

15. Phương pháp theo phương án 13 hoặc 14, trong đó giá trị của biến thiên `NumTilesInPic` bằng `NumTileColumns * NumTileRows`.

16. Phương pháp theo phương án bất kỳ trong số các phương án 13 đến 15, trong đó phương pháp này còn bao gồm bước:

phân tích giá trị của địa chỉ lát cắt `slice_address` của lát cắt hiện tại từ dòng bit, khi giá trị của thành phần cú pháp `rect_slice_flag` bằng giá trị mặc định thứ năm (ví dụ giá trị mặc định thứ năm bằng 0) và khi giá trị của biến `NumTilesInPic` lớn hơn giá trị định trước, trong đó lát cắt hiện tại có trong ảnh hiện tại.

17. Phương pháp theo phương án 16, trong đó giá trị của địa chỉ lát cắt được tạo mã trong tiêu đề lát cắt cho lát cắt hiện tại.

18. Phương pháp theo phương án bất kỳ trong số các phương án 13 đến 17, trong đó phương pháp này còn bao gồm bước:

thu được số lượng các ảnh xếp lớp `num_tiles_in_slice_minus1` có trong lát cắt hiện tại, khi giá trị của thành phần cú pháp `rect_slice_flag` bằng giá trị mặc định thứ năm và khi giá trị của biến `NumTilesInPic` lớn hơn giá trị định trước.

19. Phương pháp theo phương án 18, trong đó số lượng ảnh xếp lớp `num_tiles_in_slice_minus1` được tạo mã trong tiêu đề lát cắt cho lát cắt hiện tại.

20. Bộ giải mã (30) bao gồm mạch xử lý để thực hiện phương pháp theo phương án bất kỳ trong số các phương án 1 đến 19.

21. Sản phẩm chương trình máy tính chứa mã chương trình để thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 19 khi được thực thi trên máy tính hoặc bộ xử lý.

22. Bộ giải mã, bộ giải mã này bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bằng máy tính bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực hiện bởi các bộ xử lý, trong đó chương trình này, khi được thực hiện bởi các bộ xử lý, tạo cấu hình bộ giải mã để thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 19.

23. Phương tiện phi nhất thời đọc được bằng máy tính có mang mã chương trình mà khi được thực thi bởi thiết bị máy tính thì khiến thiết bị máy tính đó thực hiện phương pháp theo phương án bất kỳ trong số các phương án từ 1 đến 19.

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

FIG.14 là sơ đồ khối thể hiện hệ thống cấp nội dung 3100 để thực hiện dịch vụ phân bố nội dung. Hệ thống cung cấp nội dung 3100 này bao gồm thiết bị chụp 3102, thiết bị đầu cuối 3106, và tùy chọn là bao gồm bộ hiển thị 3126. Thiết bị chụp 3102 truyền thông với thiết bị đầu cuối 3106 qua liên kết truyền thông 3104. Liên kết truyền thông có thể bao gồm kênh truyền thông 13 được mô tả trên đây. Liên kết truyền thông 3104 bao gồm nhưng không giới hạn ở WIFI, cáp Ethernet, vô tuyến (3G/4G/5G), USB, hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Thiết bị chụp 3102 tạo ra dữ liệu, và có thể mã hóa dữ liệu bằng phương pháp mã hóa như được thể hiện trong các phương án trên đây. Theo cách khác, thiết bị chụp 3102 có thể phân phối dữ liệu đến máy chủ tạo luồng (không được thể hiện trên hình vẽ trên các hình vẽ), và máy chủ mã hóa dữ liệu và truyền dữ liệu được mã hóa đến thiết bị đầu cuối 3106. Thiết bị chụp 3102 bao gồm nhưng không giới hạn ở camera, điện thoại thông minh hoặc thiết bị dạng khối (Pad) thông minh, máy tính hoặc máy tính xách tay, hệ thống hội nghị video, PDA, thiết bị được lắp trên xe, hoặc sự kết hợp

của thiết bị bất kỳ trong số đó, hoặc tương tự. Ví dụ, thiết bị chụp 3102 có thể bao gồm thiết bị nguồn 12 như được mô tả trên đây. Khi dữ liệu bao gồm video, bộ mã hóa video 20 có trong thiết bị chụp 3102 có thể thực hiện thực tế việc xử lý mã hóa video. Khi dữ liệu bao gồm audio (tức là, tiếng nói), bộ mã hóa audio có trong thiết bị chụp 3102 có thể thực hiện thực tế việc xử lý mã hóa audio. Đối với một số tình huống thực tế, thiết bị chụp 3102 phân phối dữ liệu video và audio được mã hóa bằng cách dồn kênh chúng với nhau. Đối với các tình huống thực tế khác, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được dồn kênh. Thiết bị chụp 3102 phân phối dữ liệu audio được mã hóa và dữ liệu video được mã hóa đến thiết bị đầu cuối 3106 một cách riêng rẽ.

Trong hệ thống cung cấp nội dung 3100, thiết bị đầu cuối 310 nhận và tái tạo dữ liệu được mã hóa. Thiết bị đầu cuối 3106 có thể là thiết bị có khả năng nhận và phục hồi dữ liệu, như điện thoại hoặc thiết bị dạng khối thông minh 3108, máy tính hoặc máy tính xách tay 3110, máy ghi video mạng (network video recorder, NVR)/ máy ghi video số (digital video recorder, DVR) 3112, TV 3114, thiết bị giải mã tín hiệu và truyền dữ liệu (set top box, STB) 3116, hệ thống hội nghị video 3118, hệ thống giám sát video 3120, thiết bị hỗ trợ số cá nhân (PDA) 3122, thiết bị được lắp trên xe 3124, hoặc sự kết hợp của bất kỳ thiết bị nào trong số đó, hoặc tương tự có khả năng giải mã dữ liệu được mã hóa nêu trên. Ví dụ, thiết bị đầu cuối 3106 có thể bao gồm thiết bị đích 14 như được mô tả trên đây. Khi dữ liệu được mã hóa bao gồm video, bộ giải mã video 30 có trong thiết bị đầu cuối được dành ưu tiên để thực hiện việc giải mã video. Khi dữ liệu được mã hóa bao gồm audio, bộ giải mã audio có trong thiết bị đầu cuối được dành ưu tiên để thực hiện việc xử lý giải mã audio.

Đối với thiết bị đầu cuối có bộ hiển thị của nó, ví dụ, điện thoại hoặc thiết bị dạng khối thông minh 3108, máy tính hoặc máy tính xách tay 3110, máy ghi video mạng (NVR)/ máy ghi video số (DVR) 3112, TV 3114, thiết bị hỗ trợ số cá nhân (PDA)

3122, hoặc thiết bị được lắp trên xe 3124, thiết bị đầu cuối có thể cấp dữ liệu được giải mã cho bộ hiển thị của nó. Đối với thiết bị đầu cuối không có bộ hiển thị, ví dụ, STB 3116, hệ thống hội nghị video 3118, hoặc hệ thống giám sát video 3120, bộ hiển thị ngoài 3126 được liên kết trong đó để nhận và thể hiện dữ liệu được giải mã.

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

FIG.9 là hình vẽ thể hiện cấu trúc của ví dụ về thiết bị đầu cuối 3106. Sau khi thiết bị đầu cuối 3106 nhận luồng từ thiết bị chụp 3102, bộ phận tiến hành giao thức 3202 phân tích giao thức truyền của luồng này. Giao thức bao gồm nhưng không giới hạn ở giao thức tạo luồng thời gian thực (Real Time Streaming Protocol, RTSP), giao thức truyền siêu văn bản (Hyper Text Transfer Protocol, HTTP), giao thức tạo luồng trực tiếp HTTP (HTTP Live streaming protocol, HLS), MPEG-DASH, giao thức truyền thời gian thực (Real-time Transport protocol, RTP), giao thức nhắn tin thời gian thực (Real Time Messaging Protocol, RTMP), hoặc loại kết hợp bất kỳ của chúng, hoặc tương tự.

Sau khi bộ phận tiến hành giao thức 3202 xử lý luồng, tệp luồng được tạo ra. Tệp này được kết xuất đến bộ phận phân kênh 3204. Bộ phận phân kênh 3204 có thể tách dữ liệu được dồn kênh thành dữ liệu audio được mã hóa và dữ liệu video được mã hóa. Như được mô tả trên đây, đối với một số tình huống thực tế, ví dụ trong hệ thống hội nghị video, dữ liệu audio được mã hóa và dữ liệu video được mã hóa không được dồn kênh. Trong tình huống này, dữ liệu được mã hóa được truyền đến bộ giải mã video 3206 và bộ giải mã audio 3208 mà không qua bộ phận phân kênh 3204.

Qua việc xử lý phân kênh, luồng cơ bản (elementary stream, ES) video, ES audio, và tùy chọn là phụ đề được tạo ra. Bộ giải mã video 3206, mà bao gồm bộ giải mã video 30 như được giải thích trong các phương án nêu trên, giải mã ES video bằng

phương pháp giải mã như được thể hiện trong các phương án nêu trên để tạo khung video, và cung cấp dữ liệu này cho bộ phận đồng bộ 3212. Bộ giải mã audio 3208, giải mã ES audio để tạo ra khung audio, và cấp dữ liệu này đến cụm đồng bộ 3212. Theo cách thay thế, khung video có thể chứa trong bộ đệm (không được thể hiện trên FIG.9) trước khi cấp nó tới cụm đồng bộ 3212. Tương tự, khung âm thanh có thể chứa trong bộ đệm (không được thể hiện trên FIG.9) trước khi cấp nó tới cụm đồng bộ 3212.

Bộ phận đồng bộ 3212 đồng bộ hóa khung video và khung audio, và cung cấp video/audio cho bộ hiển thị video/audio 3214. Ví dụ, bộ phận đồng bộ 3212 đồng bộ hóa việc trình diễn thông tin video và audio. Thông tin có thể tạo mã trong cú pháp nhờ sử dụng các nhãn thời gian liên quan đến việc trình diễn audio được tạo mã và dữ liệu trực quan và các nhãn thời gian liên quan đến việc truyền chính luồng dữ liệu.

Nếu phụ đề có trong luồng, thì bộ giải mã phụ đề 3210 giải mã phụ đề, và đồng bộ hóa nó với khung video và khung audio, và cung cấp video/audio/phụ đề cho bộ hiển thị video/audio/phụ đề 3216.

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

Các toán tử toán học

Các toán tử toán học được sử dụng trong sáng chế tương tự như các toán tử toán học được sử dụng trong ngôn ngữ lập trình C. Tuy nhiên, các kết quả của phép chia số nguyên và các phép toán dịch số học được xác định chính xác hơn, và các phép toán bổ sung được xác định, như mũ hóa và chia giá trị thực. Các quy ước đánh số và đếm thường bắt đầu từ 0, ví dụ, "thứ nhất" tương đương với thứ 0, "thứ hai" tương đương với thứ 1, v.v..

Các toán tử số học

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

$+$ Cộng

$-$ Trừ (dưới dạng toán tử hai đối số) hoặc phủ định (dưới dạng toán tử tiếp đầu ngữ một ngôi)

$*$ Phép nhân, bao gồm phép nhân ma trận

x^y Mũ hóa. Quy định x lũy thừa y . Trong các ngữ cảnh khác, cách ký hiệu này được sử dụng cho chỉ số trên mà không được dự định để hiểu là mũ hóa.

$/$ Phép chia nguyên với việc cắt bỏ phần kết quả về phía không (zero). Ví dụ, $7 / 4$ và $-7 / -4$ được cắt bỏ thành 1 và $-7 / 4$ và $7 / -4$ được cắt bỏ thành -1 .

$\div$ Được sử dụng để thể hiện phép chia trong các phương trình toán học trong đó không có việc cắt bỏ hoặc làm tròn được dự định.

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

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

$x \% y$ Phép chia lấy dư. Phần dư của x được chia bởi y , được xác định chỉ cho các số nguyên x và y với $x \geq 0$ và $y > 0$.

Các toán tử logic

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

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

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

$!$ logic Boolean "không"

$x ? y : z$ nếu x là TRUE (đúng) hoặc không bằng 0, thì đánh giá giá trị của y ; ngược lại, thì đánh giá giá trị của z .

Các toán tử quan hệ

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

>	Lớn hơn
>=	Lớn hơn hoặc bằng
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
==	Bằng
!=	Không bằng

Khi toán tử quan hệ được áp dụng vào phần tử cú pháp hoặc biến số mà đã được ấn định giá trị "na" (not applicable - không áp dụng được), giá trị "na" được xử lý như là giá trị phân biệt đối với phần tử cú pháp hoặc biến số. Giá trị "na" được coi là không bằng giá trị khác bất kỳ.

Các toán tử theo từng bit

Các toán tử theo từng bit sau đây được định nghĩa như sau:

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

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

$\wedge$ "exclusive or" ("hoặc tuyệt đối") theo từng bit. Khi thao tác trên các đối số nguyên, thao tác trên biểu diễn bù hai của giá trị nguyên. Khi thao tác trên đối số nhị phân mà chứa ít bit hơn so với đối số khác, đối số ngắn hơn được mở rộng bằng cách bổ sung các bit có trọng số cao hơn bằng 0.

$x \gg y$ Dịch chuyển phải số học của biểu diễn nguyên bù hai của x theo y chữ số nhị phân. Hàm này được định nghĩa chỉ đối với các giá trị nguyên không âm của y . Các bit được dịch chuyển vào các bit có trọng số lớn nhất (các MSB) do phép dịch chuyển phải có giá trị bằng MSB của x trước phép toán dịch chuyển.

$x \ll y$ Dịch chuyển trái số học của biểu diễn nguyên bù hai của x theo y chữ số nhị phân. Hàm này được định nghĩa chỉ đối với các giá trị nguyên không âm của y . Các bit được dịch chuyển vào các bit có trọng số nhỏ nhất (các LSB) do phép dịch chuyển trái có giá trị bằng 0.

Các toán tử gán

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

- $=$ Toán tử gán
- $++$ Tăng, tức là, $x++$ tương đương với $x = x + 1$; khi được sử dụng trong chỉ số mảng, đánh giá đến giá trị của biến số trước phép toán tăng.
- $--$ Giảm, tức là, $x--$ tương đương với $x = x - 1$; khi được sử dụng trong chỉ số mảng, đánh giá đến giá trị của biến số trước phép toán giảm.
- $+=$ Tăng theo lượng được quy định, tức là, $x += 3$ tương đương với $x = x + 3$, và $x += (-3)$ tương đương với $x = x + (-3)$.

$\text{--}=\text{}$ Giảm theo lượng được quy định, tức là, $x \text{ --} 3$ tương đương với $x = x - 3$, và
 $x \text{ --} (-3)$ tương đương với $x = x - (-3)$.

Ký hiệu khoảng

Ký hiệu sau đây được sử dụng để quy định khoảng của các giá trị:

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

Các hàm toán học

Các hàm toán học sau đây được định nghĩa:

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

$\text{Asin}(x)$ hàm sin lượng giác ngược, thao tác trên đối số x mà
 nằm trong khoảng từ $-1,0$ đến $1,0$, bao gồm cả hai giá trị này, với giá trị
 đầu ra nằm trong khoảng từ
 $-\pi/2$ đến $\pi/2$, bao gồm cả hai giá trị này, theo đơn vị
 radian

$\text{Atan}(x)$ hàm tang lượng giác ngược, thao tác trên đối số x , với
 giá trị đầu ra nằm trong khoảng từ $-\pi/2$ đến $\pi/2$, bao gồm cả hai giá trị
 này, theo đơn vị radian

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0,5)$$

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

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

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

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

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

Thứ tự ưu tiên phép toán

Khi thứ tự ưu tiên trong biểu thức không được chỉ ra rõ ràng bằng cách sử dụng ngoặc, các quy tắc sau đây áp dụng:

- Các phép toán có mức ưu tiên cao hơn được đánh giá là trước phép toán bất kỳ có mức ưu tiên thấp hơn.
- Các phép toán có cùng mức ưu tiên được đánh giá là tuần tự từ trái sang phải.

Bảng dưới đây quy định mức ưu tiên của các phép toán từ cao nhất đến thấp nhất; vị trí cao hơn trong bảng chỉ ra mức ưu tiên cao hơn.

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

Bảng: Mức ưu tiên phép toán từ cao nhất (ở đỉnh bảng) đến thấp nhất (ở đáy bảng)

các phép toán (với các toán hạng x, y, và z)
"x++", "x--"
"!x", "-x" (như là toán tử tiếp đầu ngữ một ngôi)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (như là toán tử hai đối số), " $\sum_{i=x}^y f(i)$ "
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x.y"
"x = y", "x += y", "x -= y"

Phần mô tả bằng văn bản của các phép toán logic

Trong phần văn bản này, câu lệnh của các phép toán logic như sẽ được mô tả toán học theo hình thức sau đây:

nếu (điều kiện 0)

câu lệnh 0

còn nếu (điều kiện 1)

câu lệnh 1

...

còn không /* chú thích mang tính thông tin về điều kiện còn lại */

câu lệnh n

có thể được mô tả theo cách sau đây:

... như sau / ... phần sau đây áp dụng:

- Nếu điều kiện 0, câu lệnh 0
- Nếu không, nếu điều kiện 1, câu lệnh 1
- ...
- Nếu không (chú thích mang tính thông tin về điều kiện còn lại), câu lệnh n

Mỗi "Nếu ... Nếu không, nếu ... Nếu không, ..." câu lệnh trong văn bản này được giới thiệu với "... như sau" hoặc "... phần sau đây áp dụng" được theo ngay sau bởi "Nếu ...". Điều kiện cuối cùng của "Nếu ... Nếu không, nếu ... Nếu không, ..." luôn là "Nếu không, ...". Được đan xen "Nếu ... Nếu không, nếu ... Nếu không, ..." các câu lệnh có thể được nhận dạng bằng cách so khớp "... như sau" hoặc "... phần sau đây áp dụng" với kết thúc "Nếu không, ...".

Trong phần văn bản này, câu lệnh của các phép toán logic như sẽ được mô tả toán học theo hình thức sau đây:

nếu (điều kiện 0a && điều kiện 0b)

câu lệnh 0

còn không nếu (điều kiện 1a || điều kiện 1b)

câu lệnh 1

...

còn không

câu lệnh n

có thể được mô tả theo cách sau đây:

... như sau / ... phần sau đây áp dụng:

- Nếu tất cả các điều kiện sau đây là đúng, câu lệnh 0:
 - điều kiện 0a
 - điều kiện 0b
- Nếu không, nếu một hoặc nhiều điều kiện sau đây là đúng, câu lệnh 1:
 - điều kiện 1a
 - điều kiện 1b
- ...
- Nếu không, câu lệnh n

Trong phần văn bản này, câu lệnh của các phép toán logic như sẽ được mô tả toán học theo hình thức sau đây:

nếu (điều kiện 0)

câu lệnh 0

nếu (điều kiện 1)

câu lệnh 1

có thể được mô tả theo cách sau đây:

Khi điều kiện 0, câu lệnh 0

Khi điều kiện 1, câu lệnh 1.

Mặc dù các phương án của sáng chế đã được mô tả sơ bộ dựa trên việc tạo mã video, nhưng cần lưu ý rằng các phương án của hệ thống tạo mã 10, bộ mã hóa 20 và bộ giải mã 30 (và tương ứng là hệ thống 10) và các phương án khác được mô tả ở đây cũng có thể được tạo cấu hình để xử lý hoặc tạo mã hình ảnh tĩnh, tức là xử lý hoặc tạo mã hình ảnh riêng độc lập với hình ảnh trước hoặc liên tiếp bất kỳ như trong việc tạo mã video. Nói chung, chỉ các bộ phận dự đoán liên ảnh 244 (bộ mã hóa) và 344 (bộ giải mã) có thể không khả dụng trong trường hợp việc tạo mã xử lý hình ảnh được giới hạn ở một hình ảnh 17. Tất cả các chức năng khác (cũng được gọi là các công cụ hoặc các kỹ thuật) của bộ mã hóa video 20 và bộ giải mã video 30 có thể được sử dụng tương đương để xử lý hình ảnh tĩnh, ví dụ tính toán dư 204/304, biến đổi 206, lượng tử hóa 208, lượng tử hóa ngược 210/310, (biến đổi ngược 212/312, phân chia 262/362, dự đoán trong ảnh 254/354, và/hoặc lọc vòng 220, 320, và mã hóa entropi 270 và giải mã entropi 304.

Các phương án, ví dụ của bộ mã hóa 20 và bộ giải mã 30, và các chức năng được mô tả ở đây, ví dụ dựa vào bộ mã hóa 20 và bộ giải mã 30, có thể được thực hiện

trong phần cứng, phần mềm, phần sụn, hoặc sự kết hợp bất kỳ của chúng. Nếu được thực hiện trong phần mềm, các chức năng này có thể được lưu trữ trên phương tiện đọc được bằng máy tính hoặc được truyền qua phương tiện truyền thông dưới dạng một hoặc nhiều lệnh hoặc mã và được thực hiện bởi bộ phận xử lý dựa trên phần cứng. Phương tiện 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 như phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông bao gồm phương tiện bất kỳ hỗ trợ truyền chương trình máy tính từ nơi này đến nơi khác, ví dụ, theo giao thức truyền thông. Theo cách này, phương tiện đọc được bằng máy tính nói chung có thể tương ứng với (1) phương tiện lưu trữ đọc được bằng máy tính hữu hình mà là lâu dài hoặc (2) phương tiện truyền thông như sóng 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 lấy các lệnh, mã và/hoặc các cấu trúc dữ liệu để thực thi các kỹ thuật được mô tả trong sáng chế. Sản phẩm chương trình máy tính có thể bao gồm phương tiện đọc được bằng máy tính.

Ví dụ, và không phải giới hạn, phương tiện lưu trữ đọc được bằng máy tính như vậy có thể bao gồm RAM, ROM, EEPROM, CD-ROM hoặc bộ lưu trữ đĩa quang khác, bộ lưu trữ đĩa từ, hoặc các thiết bị lưu trữ từ tính khác, bộ nhớ tác động nhanh (flash memory), 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 được gọi là phương tiện đọc được bằng máy tính. Ví dụ, nếu các lệnh được truyền từ vị trí web, máy chủ, hoặc nguồn ở xa khác sử dụng cáp đồng trục, cáp sợi quang, cáp xoắn đôi, đường thuê bao số (DSL), hoặc các kỹ thuật vô tuyến như hồng ngoại, radiô, và vi sóng, thì cáp đồng trục, cáp sợi quang, cáp xoắn đôi, DSL, hoặc các kỹ thuật vô tuyến như hồng ngoại, radiô, và vi sóng có trong định nghĩa về phương tiện. Tuy nhiên, nên hiểu rằng phương tiện lưu trữ đọc

được bằng 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 tạm thời khác, mà thay vào đó được hướng đến phương tiện lưu trữ lâu dài, hữu hình. Đĩa (disk) và đĩa (disc), như được sử dụng ở đây, bao gồm đĩa (disc) compact (CD), đĩa laser, đĩa quang, đĩa đa năng số (DVD), đĩa mềm và đĩa Blu-ray, trong đó các đĩa (disk) thường tái tạo dữ liệu về mặt từ tính, trong khi các đĩa (disc) tái tạo dữ liệu về mặt quang học bằng laser. Các sự kết hợp của các nội dung trên đây cũng sẽ có trong phạm vi của phương tiện đọc được bằng máy tính.

Các lệnh có thể được thực hiện bởi một hoặc nhiều bộ xử lý, như một hoặc nhiều bộ xử lý tín hiệu số (digital signal processor, DSP), các bộ vi xử lý đa năng, các mạch tích hợp chuyên dụng (application specific integrated circuit, ASIC), các mảng logic lập trình được dạng trường (FPGA), hoặc các loại tương đương được tích hợp hoặc mạch logic rời rạc. Theo đó, thuật ngữ “bộ xử lý”, như được sử dụng ở đây có thể đề cập đến cấu trúc bất kỳ trong số cấu trúc trên đây hoặc cấu trúc khác bất kỳ thích hợp để 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 cung cấp trong phần cứng chuyên dụng và/hoặc các mô đun phần mềm được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp trong codec kết hợp. Ngoài ra, các kỹ thuật 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 rất nhiều loại thiết bị hoặc máy, bao gồm máy cầm tay vô tuyến, mạch tích hợp (IC) hoặc tập hợp các IC (ví dụ, bộ chip). Các thành phần, các mô đun, hoặc các bộ phận khác nhau được mô tả trong sáng chế để nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật được bộc lộ, mà không nhất thiết phải đòi hỏi thực hiện bởi các bộ phận phần cứng khác nhau. Ngược lại, như được mô tả trên đây, các cụm khác nhau có thể được kết hợp trong cụm phần cứng bộ mã hóa/giải mã hoặc được tạo

ra bởi một tập hợp của các cụm phần cứng phối hợp, bao gồm một hoặc nhiều bộ xử lý như được mô tả trên đây, có dựa vào phần mềm và/hoặc phần sụn thích hợp.

YÊU CẦU BẢO HỘ

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

thu được giá trị của thành phần cú pháp thứ nhất từ tập thông số chuỗi (Sequence Parameter Set - SPS), trong đó giá trị của thành phần cú pháp thứ nhất này được sử dụng để định rõ xem cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (Decoded Picture Buffer - DPB) có mặt trong SPS hay không, trong đó SPS này được tạo mã trong dòng bit video và chứa các thành phần cú pháp mà áp dụng cho chuỗi video; và

thu được giá trị của thành phần cú pháp thứ hai từ SPS, ít nhất khi xác định rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB là có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai này được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB này được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video đó.

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

thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai; và

tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB này.

3. Phương pháp theo điểm 2, trong đó bước thu được giá trị của thành phần cú pháp DPB dựa trên giá trị của thành phần cú pháp thứ hai là bao gồm bước:

thu được giá trị của thành phần cú pháp DPB từ cấu trúc cú pháp các thông số DPB, khi xác định rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB là có mặt trong cấu trúc cú pháp các thông số DPB; hoặc

thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác mà được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số DPB, khi xác định rằng giá trị của thành phần cú pháp thứ hai định rõ rằng thành phần cú pháp DPB không có mặt trong cấu trúc cú pháp các thông số DPB.

4. Phương pháp theo điểm 2, trong đó bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB là bao gồm các bước: tạo cấu hình DPB dựa trên giá trị của thành phần cú pháp DPB; và tái cấu trúc chuỗi video nhờ sử dụng DPB này.

5. Phương pháp theo điểm 2, trong đó bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB là bao gồm bước: tái cấu trúc chuỗi video dựa trên việc xác định rằng DPB được sử dụng có thỏa mãn yêu cầu được định rõ bởi giá trị của thành phần cú pháp DPB.

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

thu được giá trị của thành phần cú pháp thứ ba từ SPS, trong đó giá trị của thành phần cú pháp thứ ba này được sử dụng để xác định số lượng lớn nhất của các lớp con thời gian mà có mặt trong chuỗi video.

7. Phương pháp theo điểm 6, trong đó bước thu được giá trị của thành phần cú pháp thứ hai từ SPS là bao gồm các bước:

xác định xem số lượng lớn nhất của các lớp con thời gian trong dòng bit video có lớn hơn 1 hay không, dựa trên giá trị của thành phần cú pháp thứ ba, khi xác định rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS; và

thu được giá trị của thành phần cú pháp thứ hai từ SPS, khi xác định rằng số lượng lớn nhất của các lớp con thời gian là lớn hơn một.

8. Phương pháp mã hóa dòng bit video được thực hiện bởi thiết bị mã hóa, trong đó phương pháp này bao gồm các bước:

xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (Decoded Picture Buffer - DPB) trong tập thông số chuỗi (Sequence Parameter Set - SPS), trong đó SPS này được mã hóa trong dòng bit video và chứa các thành phần cú pháp mà áp dụng cho chuỗi video;

mã hóa giá trị của thành phần cú pháp thứ nhất vào SPS dựa trên bước xác định của sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không;

xác định sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, khi xác định rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video đó; và

mã hóa giá trị của thành phần cú pháp thứ hai vào SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

9. Phương pháp theo điểm 8, trong đó phương pháp này còn bao gồm các bước: xác định giá trị của thành phần cú pháp DPB khi xác định rằng thành phần cú pháp DPB có mặt trong cấu trúc cú pháp các thông số DPB; và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

10. Phương pháp theo điểm 8, trong đó phương pháp này còn bao gồm các bước: thiết lập giá trị của thành phần cú pháp DPB bằng giá trị của thành phần cú pháp DPB khác mà được áp dụng cho lớp con thời gian cao nhất trong cấu trúc cú pháp các thông số DPB; và tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB.

11. Phương pháp theo điểm 9, trong đó bước tái cấu trúc chuỗi video dựa trên giá trị của thành phần cú pháp DPB là bao gồm các bước: tạo cấu hình DPB để thỏa mãn giá trị của thành phần cú pháp DPB; và tái cấu trúc chuỗi video nhờ sử dụng DPB này.

12. Phương pháp theo điểm 8, trong đó sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB được xác định, khi xác định được rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS và số lượng lớn nhất của các lớp con thời gian trong dòng bit video là lớn hơn một.

13. Thiết bị để giải mã dòng bit video được tạo mã, trong đó thiết bị này bao gồm:

cụm thu được được tạo cấu hình để thu được giá trị của thành phần cú pháp thứ nhất từ tập thông số chuỗi (Sequence Parameter Set - SPS) mà được tạo mã trong dòng bit video, trong đó giá trị của thành phần cú pháp thứ nhất này được sử dụng để định rõ xem cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (Decoded Picture Buffer - DPB) có mặt trong SPS hay không; và

cụm xác định được tạo cấu hình để xác định xem giá trị của thành phần cú pháp thứ nhất có định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không; và trong đó

cụm thu được còn được tạo cấu hình để thu được giá trị của thành phần cú pháp thứ hai từ SPS, ít nhất khi xác định rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB là có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai này được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB này được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

14. Thiết bị để mã hóa dòng bit video, trong đó thiết bị này bao gồm:

cụm xác định được tạo cấu hình để xác định sự có mặt của cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (Decoded Picture Buffer - DPB) trong tập thông số chuỗi (Sequence Parameter Set - SPS) được mã hóa trong dòng bit video đó; và

cụm mã hóa được tạo cấu hình để mã hóa giá trị của thành phần cú pháp thứ nhất vào SPS dựa trên bước xác định của sự có mặt của cấu trúc cú pháp các thông số DPB trong SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ xem cấu trúc cú pháp các thông số DPB có mặt trong SPS hay không; trong đó

cụm xác định còn được tạo cấu hình để xác định sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, khi xác định rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó thành phần cú pháp DPB được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video; và trong đó

cụm mã hóa còn được tạo cấu hình để mã hóa giá trị của thành phần cú pháp thứ hai vào SPS dựa trên bước xác định của sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB.

15. Bộ giải mã, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bằng máy tính bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình này, khi được thực thi bởi các bộ xử lý, thì tạo cấu hình cho bộ giải mã này để thực hiện phương pháp theo điểm 1.

16. Bộ mã hóa, bao gồm:

một hoặc nhiều bộ xử lý; và

phương tiện lưu trữ đọc được bằng máy tính bất biến được ghép nối với các bộ xử lý và lưu trữ chương trình để thực thi bởi các bộ xử lý, trong đó chương trình này, khi được thực thi bởi các bộ xử lý, thì tạo cấu hình cho bộ mã hóa này để thực hiện phương pháp theo điểm 8.

17. Phương tiện đọc được bằng máy tính bất biến có mang mã chương trình mà khi được thực thi bởi thiết bị máy tính thì khiến thiết bị máy tính đó thực hiện phương pháp theo điểm 1.

18. Phương tiện lưu trữ bất biến bao gồm dòng bit đã được mã hóa, dòng bit này được tạo ra bằng cách chia ảnh hiện tại của tín hiệu video hoặc tín hiệu hình ảnh thành các

khởi, và bao gồm các thành phần cú pháp, trong đó các thành phần cú pháp này bao gồm thành phần cú pháp thứ nhất trong tập thông số chuỗi (Sequence Parameter Set - SPS), trong đó giá trị của thành phần cú pháp thứ nhất này được sử dụng để định rõ xem cấu trúc cú pháp các thông số bộ đệm hình ảnh đã được giải mã (Decoded Picture Buffer - DPB) có mặt trong SPS hay không; trong trường hợp mà giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB là có mặt trong SPS, thì dòng bit này còn bao gồm thành phần cú pháp thứ hai trong SPS, trong đó giá trị của thành phần cú pháp thứ hai này được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB, trong đó thành phần cú pháp DPB này được áp dụng cho lớp con thời gian ngoại trừ lớp con thời gian cao nhất trong chuỗi video.

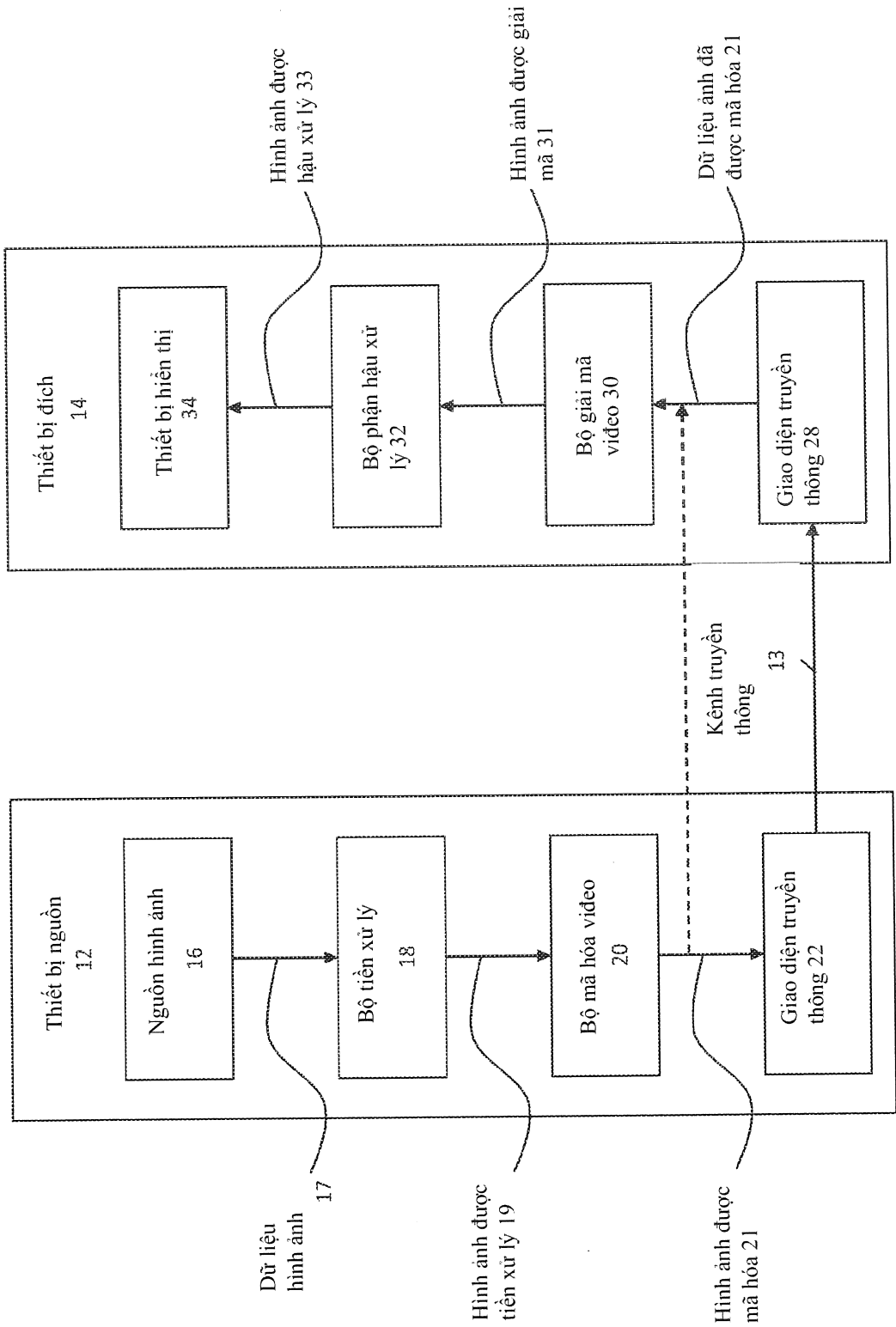


Fig. 1A

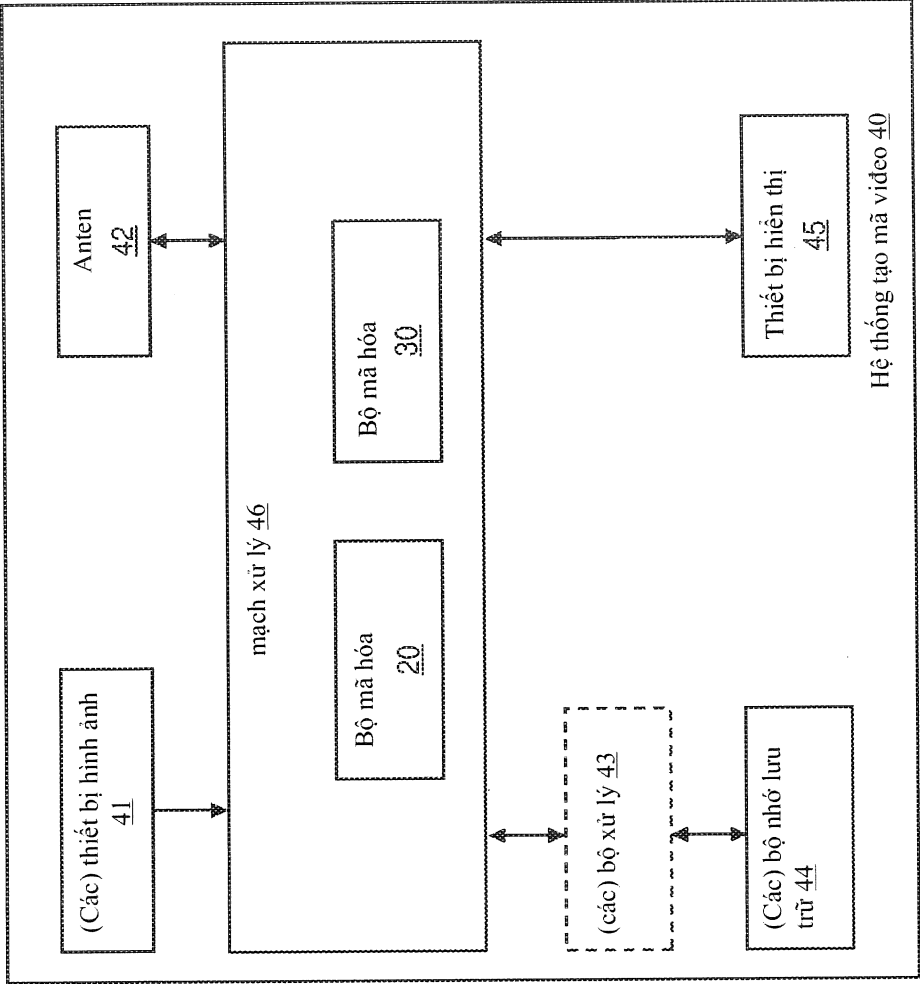
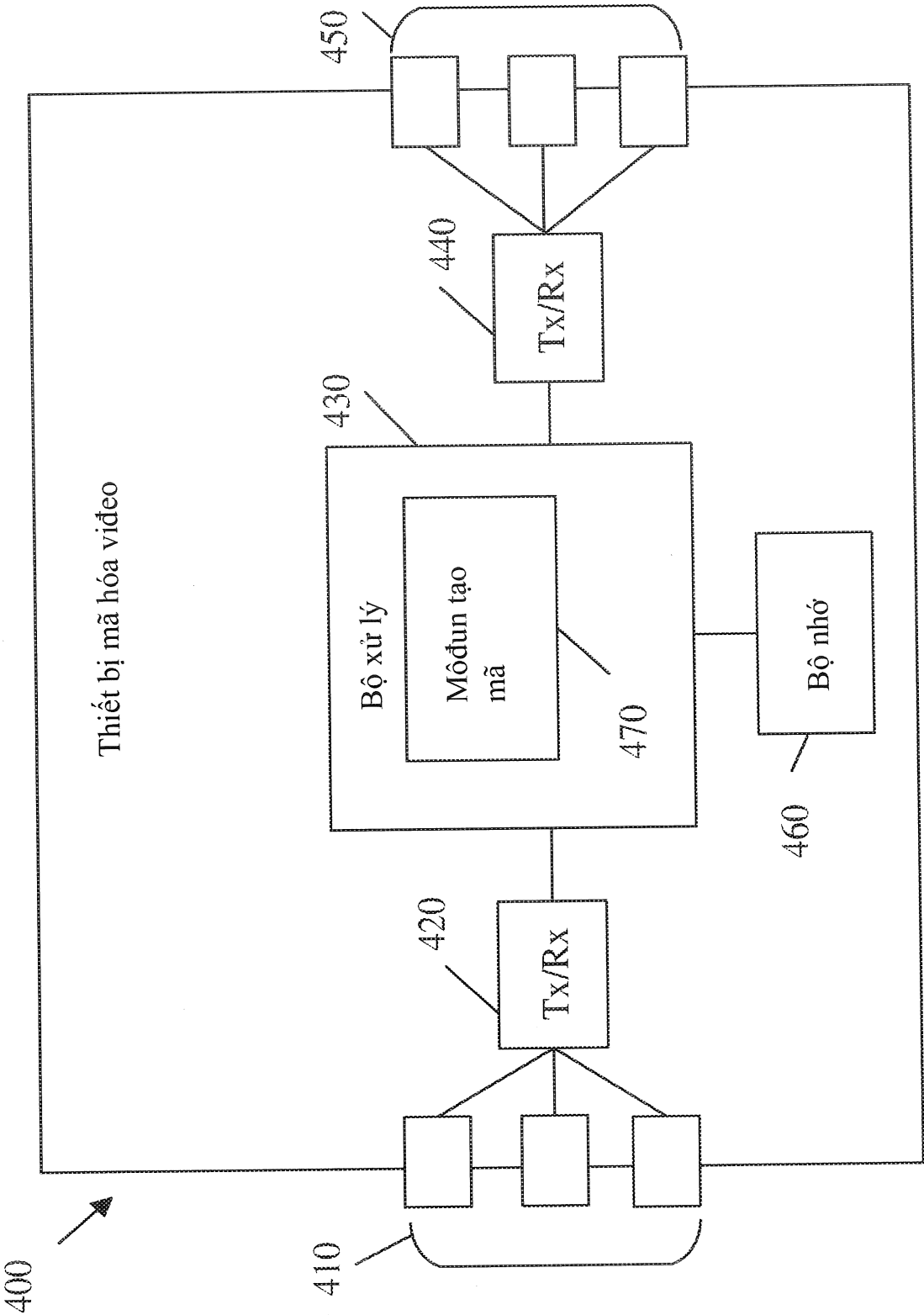


Fig. 1B



Cổng đường lên

Cổng đường
xuống

Fig. 4

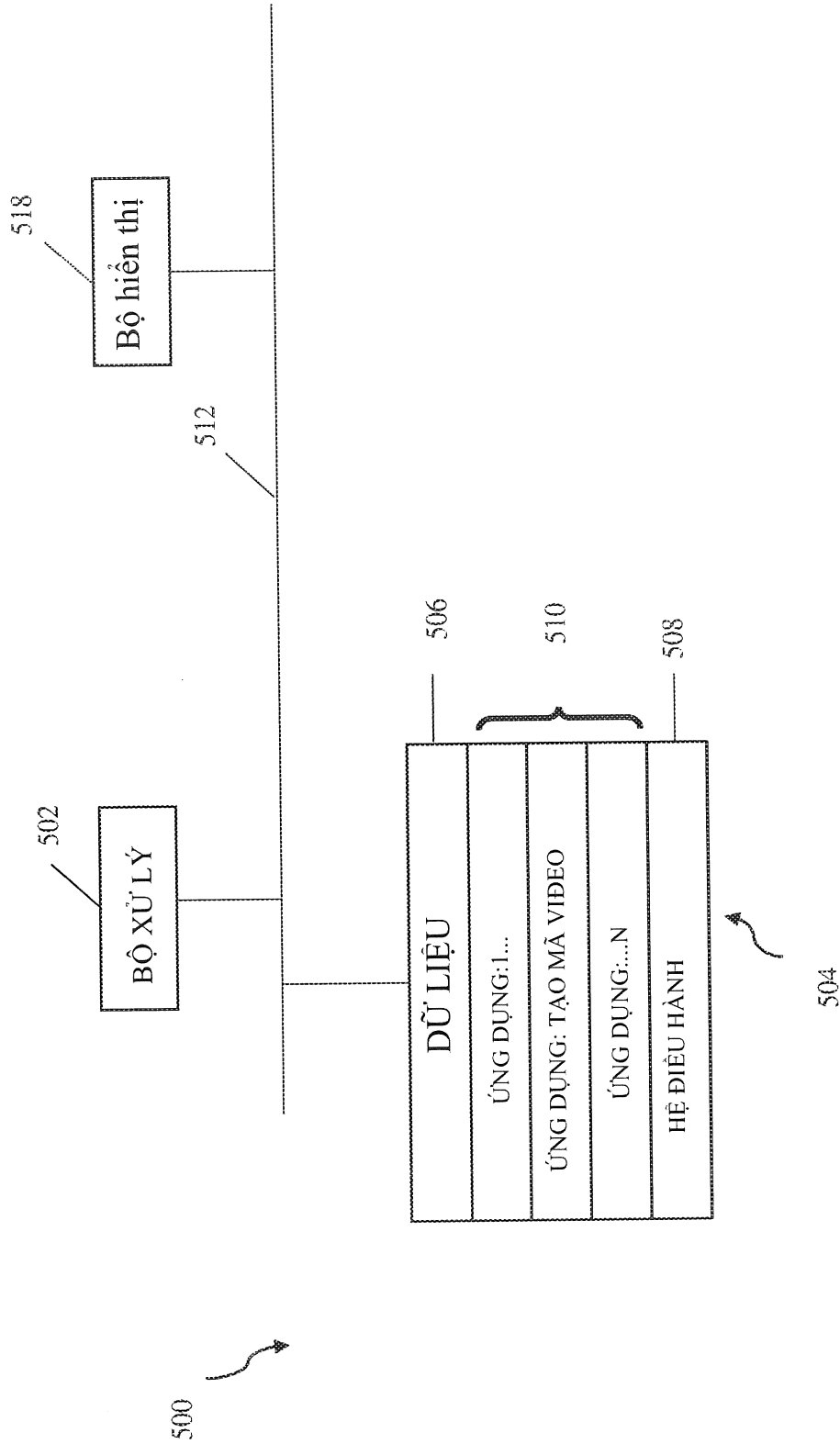


Fig. 5

7/18

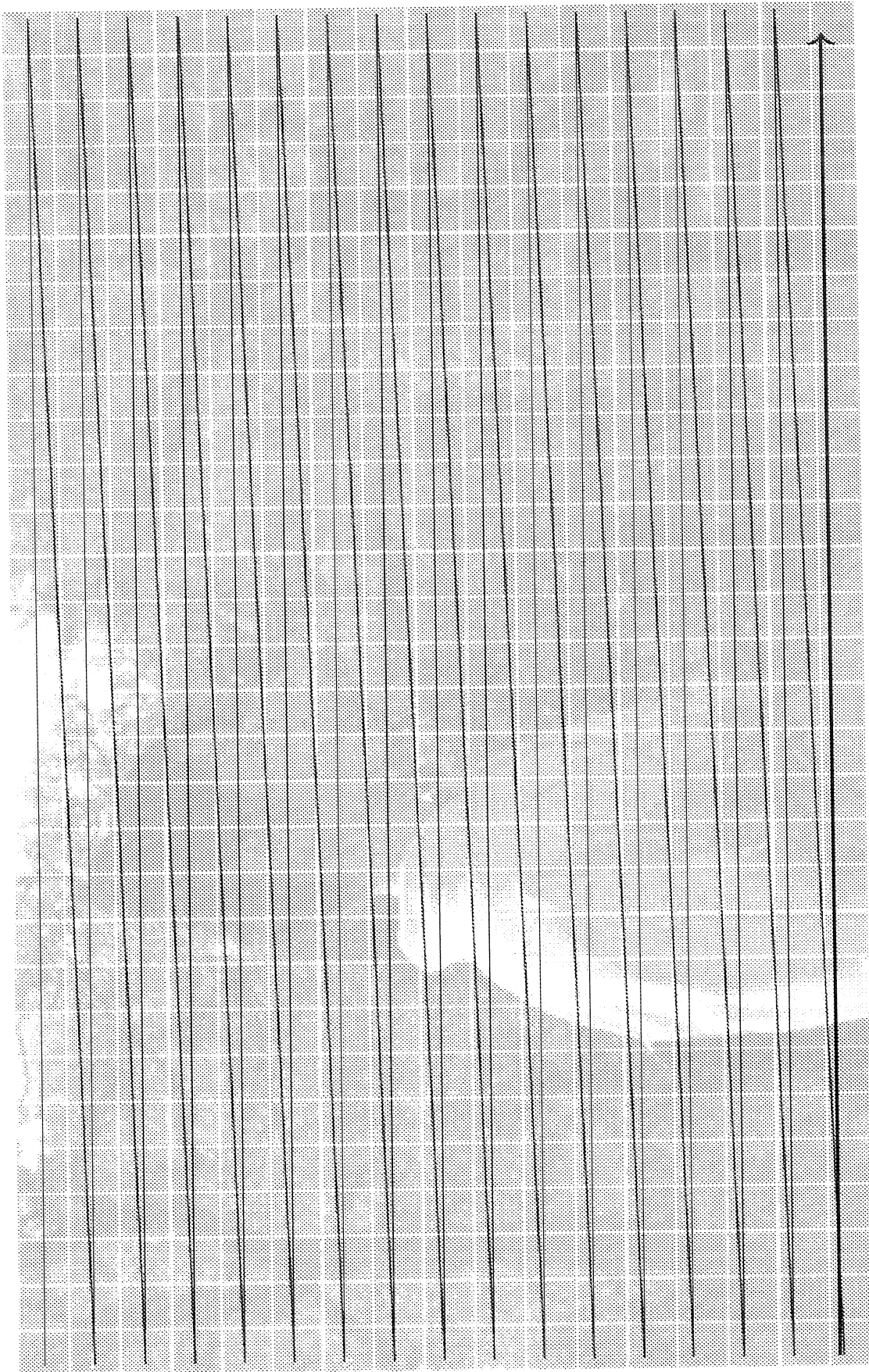


FIG. 6

9/18

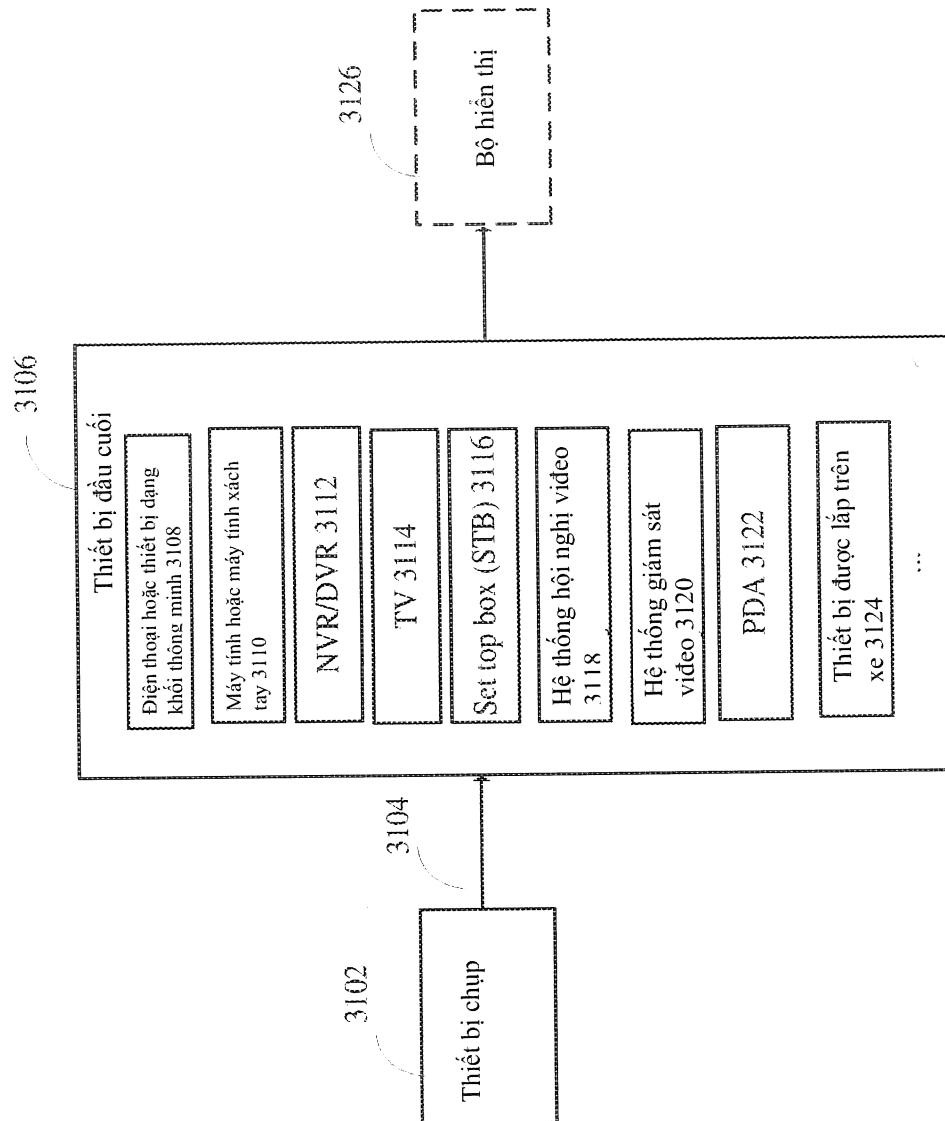


Fig. 8

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

FIG. 7

10/18

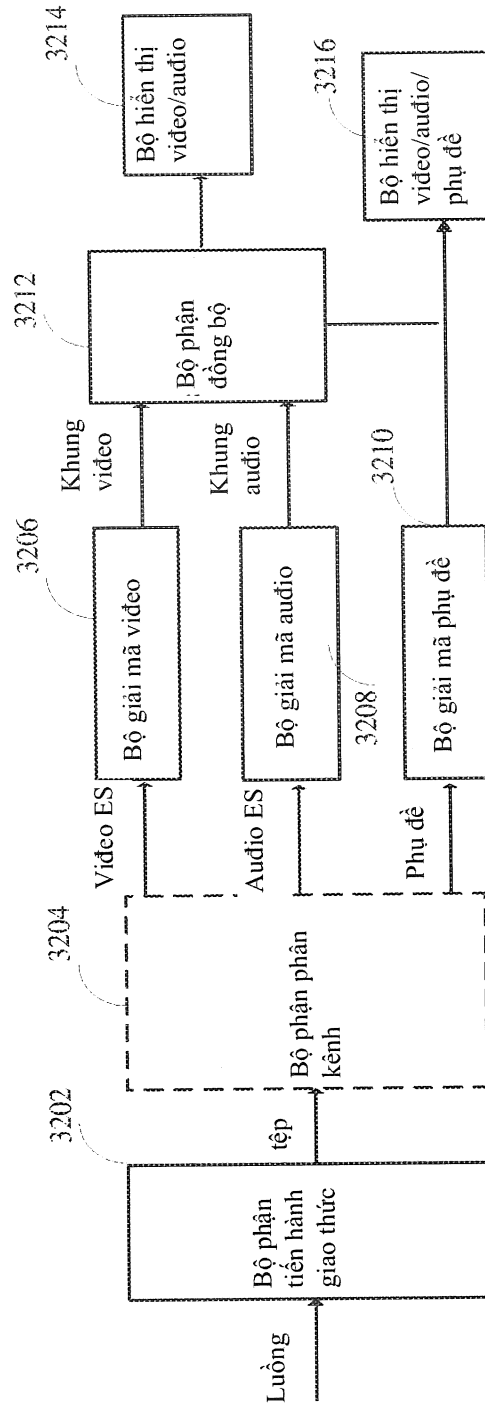


Fig. 9

11/18

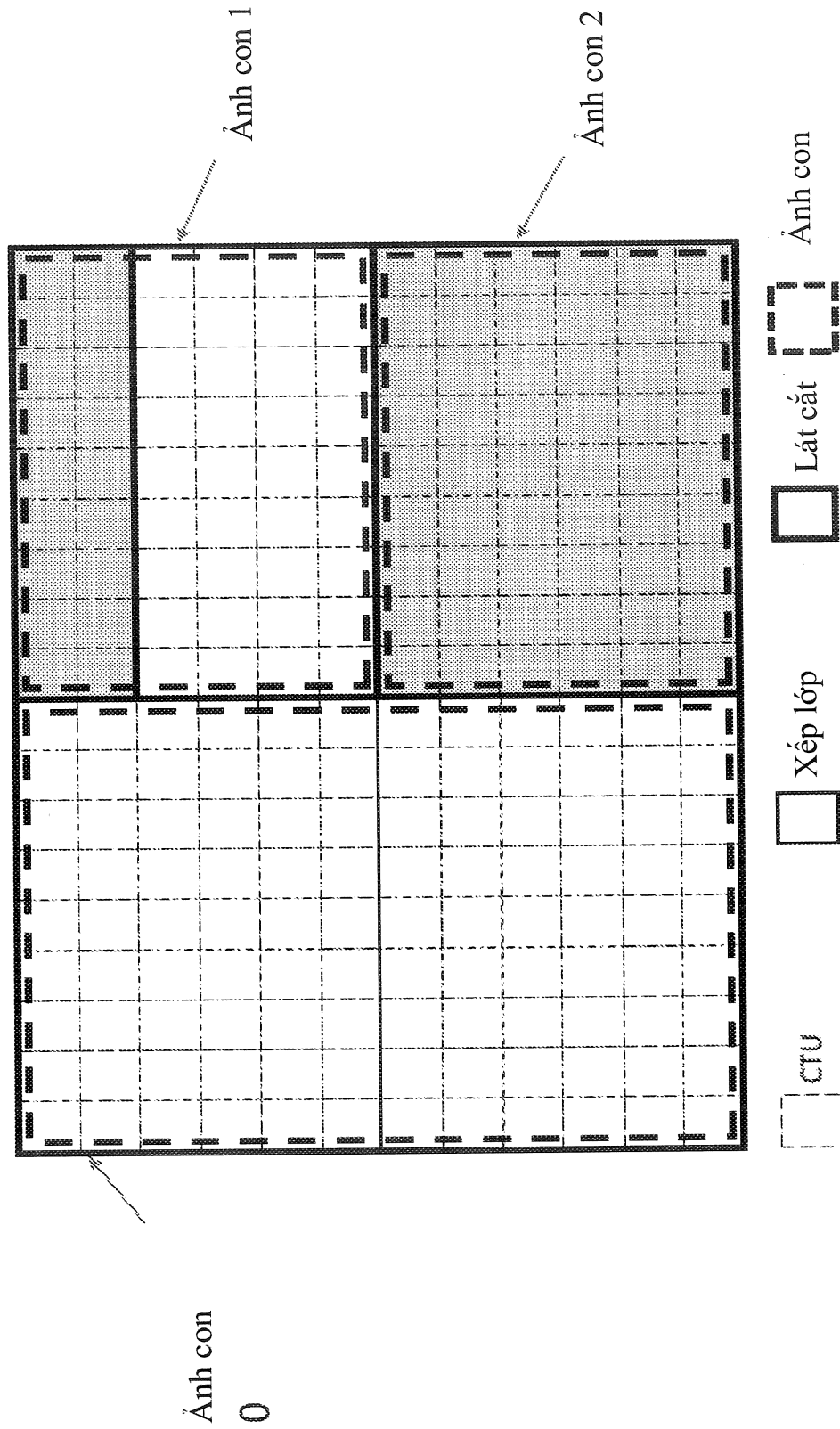


FIG. 10

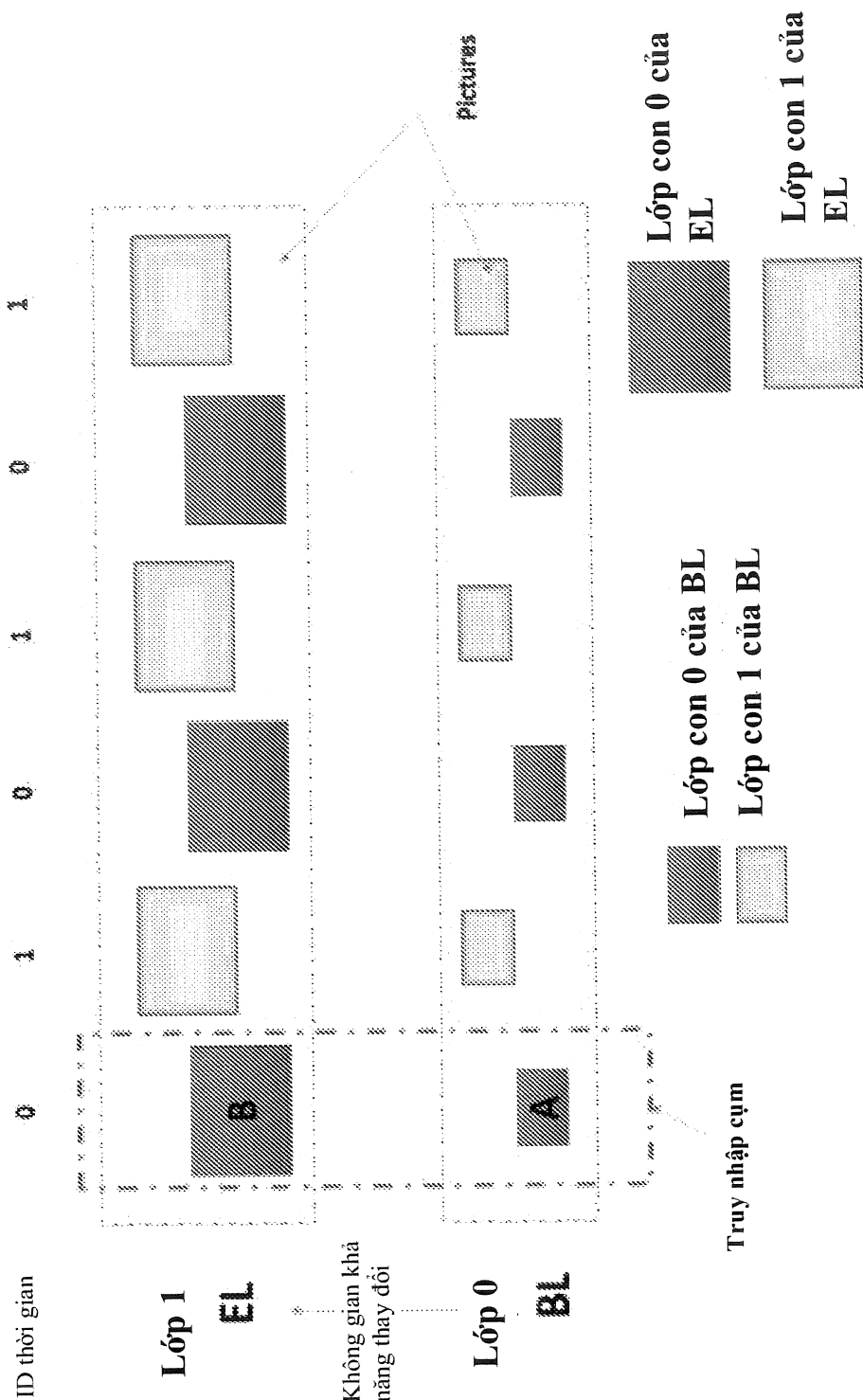


FIG. 11

13/18

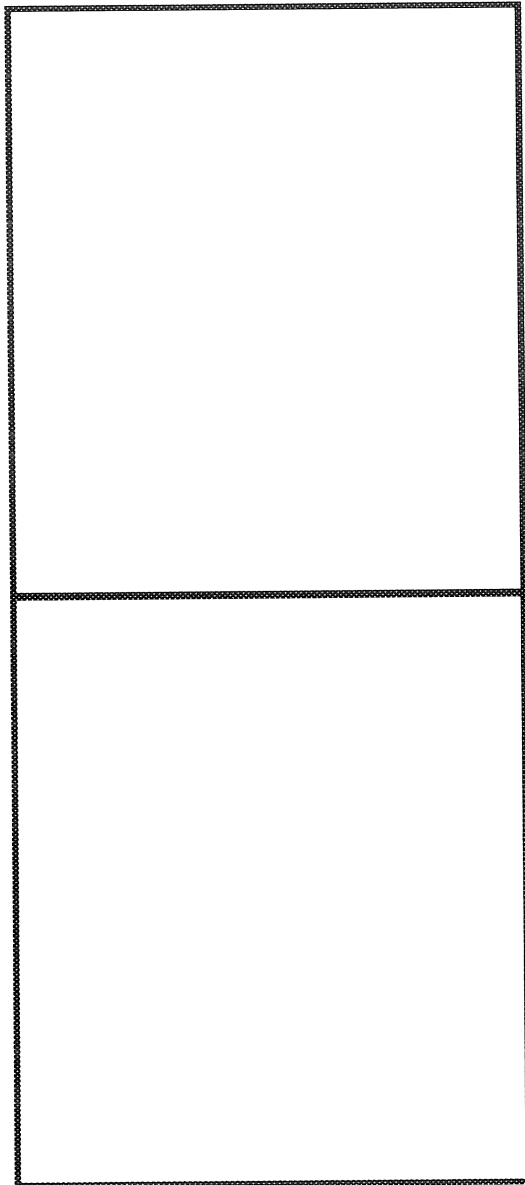


FIG. 12

14/18

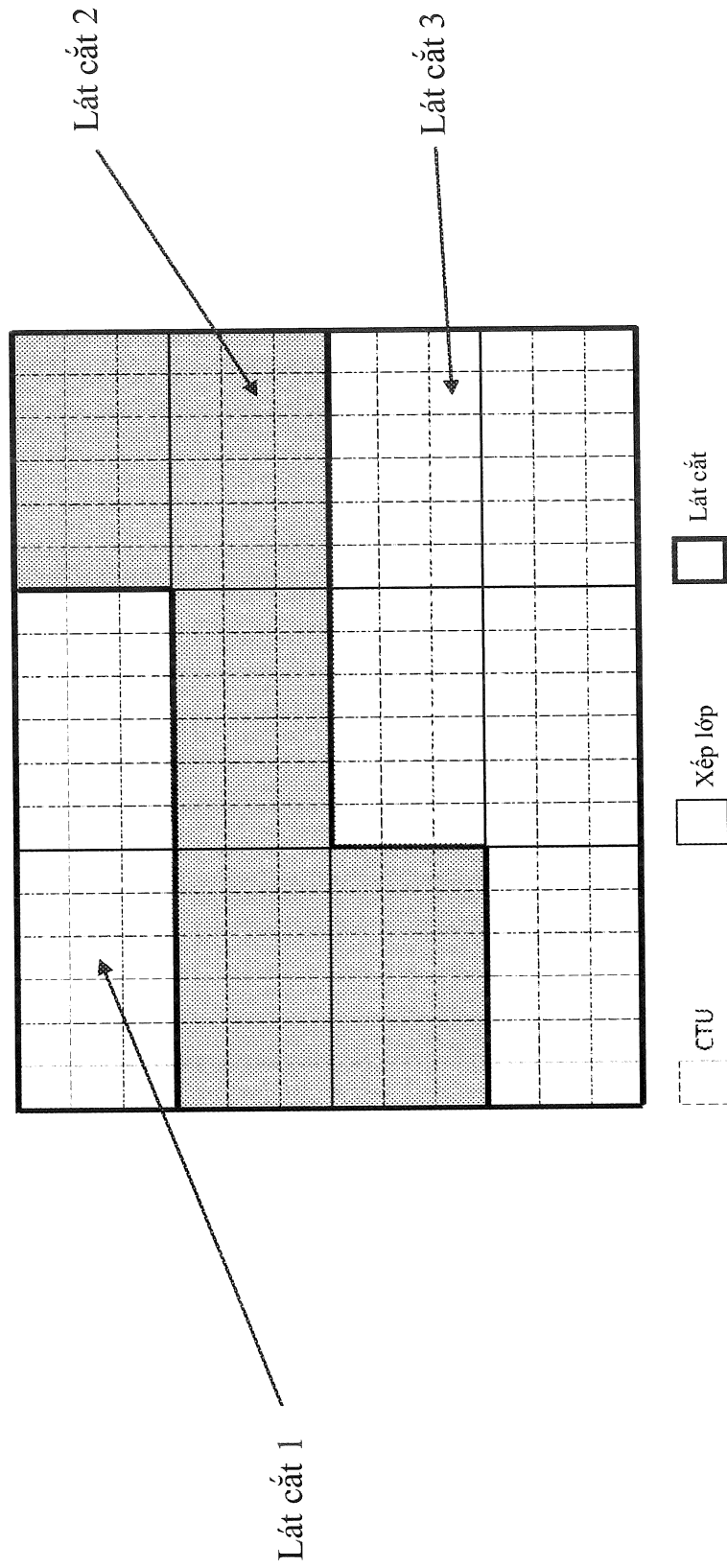


FIG. 13

Thu được giá trị của thành phần cú pháp thứ nhất từ SPS, trong đó giá trị của thành phần cú pháp thứ nhất được sử dụng để định rõ liệu cấu trúc cú pháp các thông số DBP có mặt trong SPS

— 1410

Thu được giá trị của thành phần cú pháp thứ hai từ SPS, ít nhất khi xác định rằng giá trị của thành phần cú pháp thứ nhất định rõ rằng cấu trúc cú pháp các thông số DPB có mặt trong SPS, trong đó giá trị của thành phần cú pháp thứ hai được sử dụng để định rõ sự có mặt của thành phần cú pháp DPB trong cấu trúc cú pháp các thông số DPB

— 1420

Fig. 14

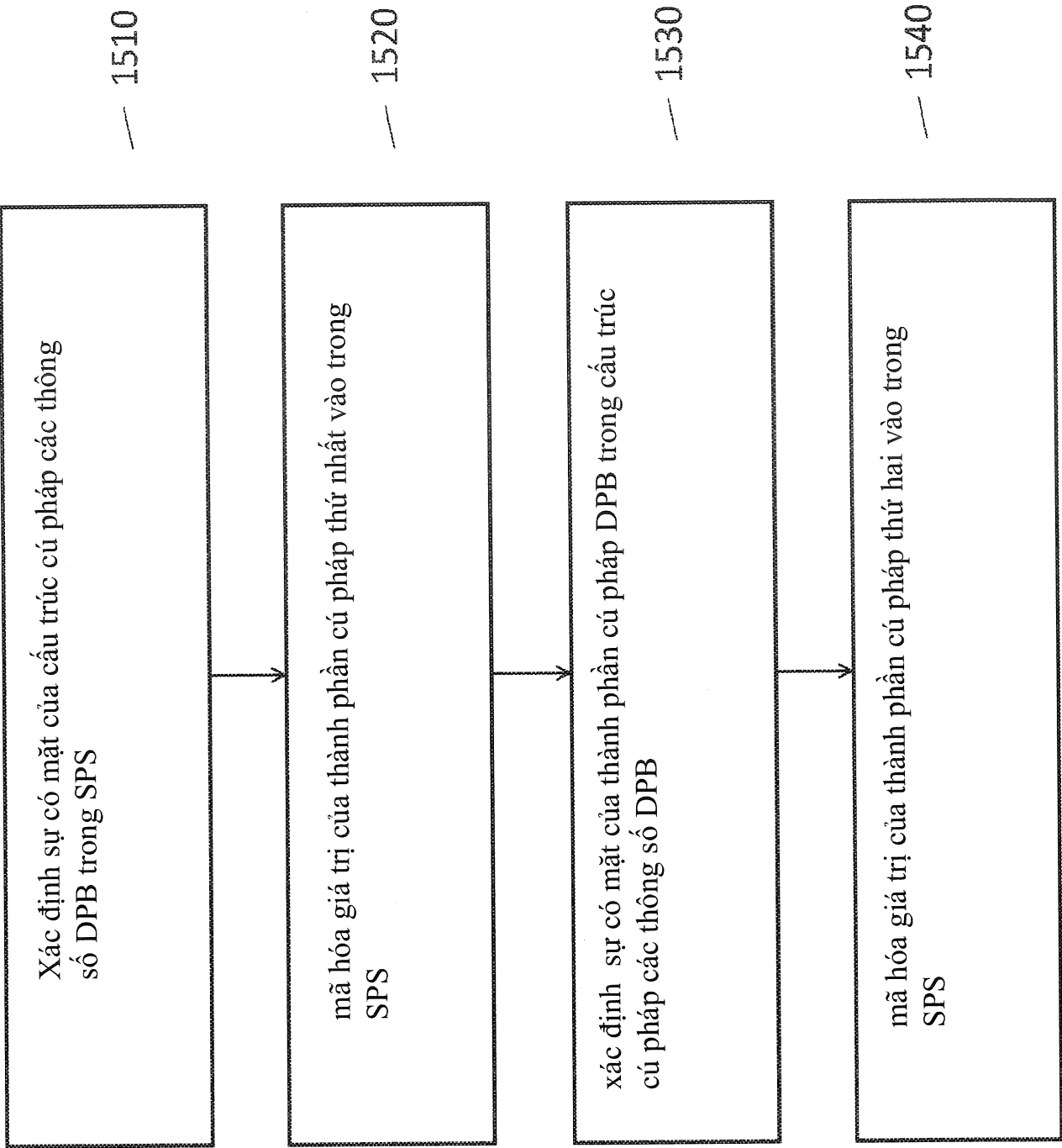


Fig. 15

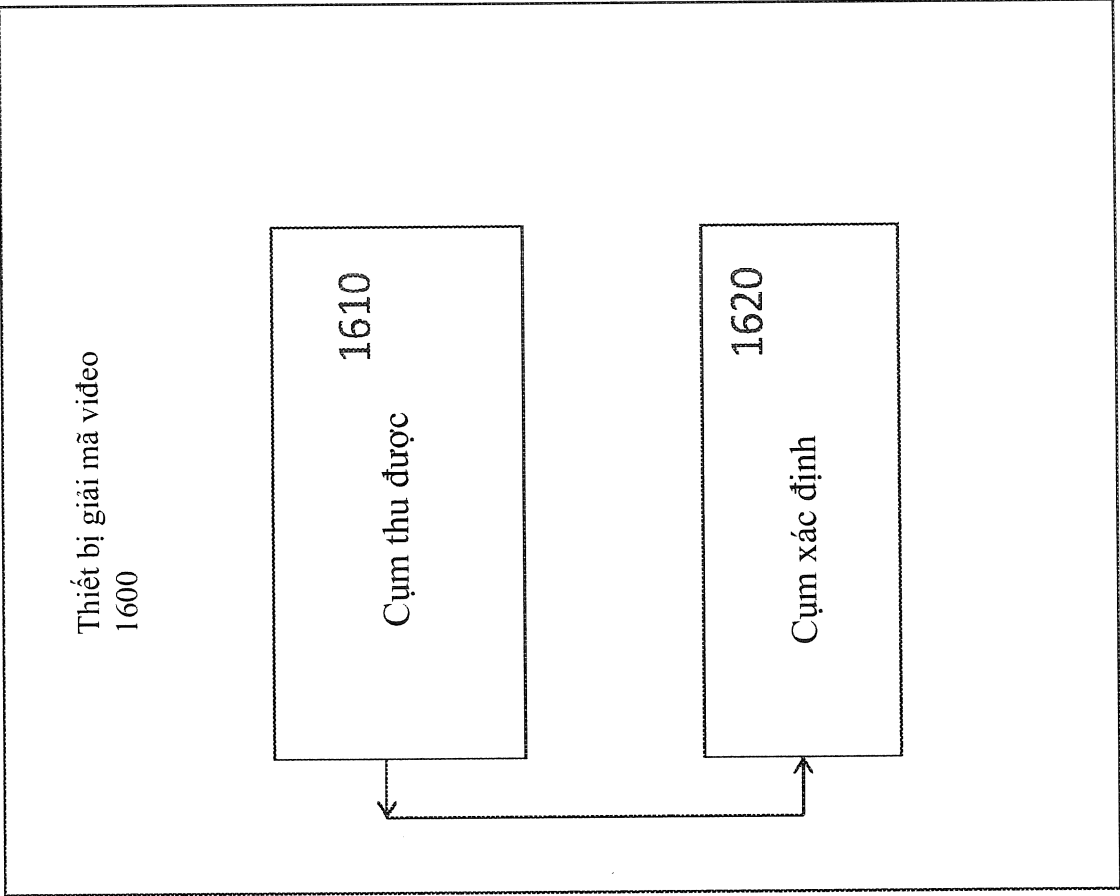


Fig. 16

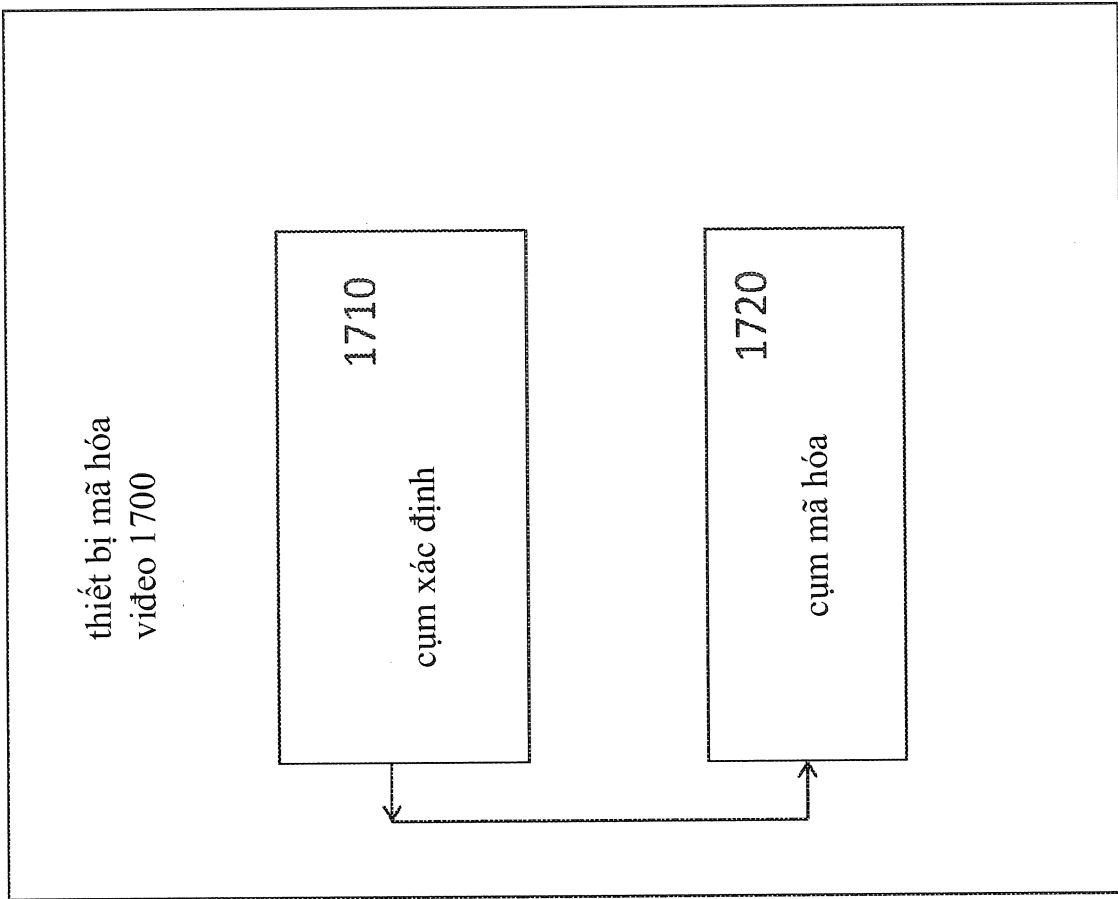


Fig. 17