



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

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

(11)



1-0051941

(51)^{2021.01} H04N 19/134; H04N 19/169; H04N
19/70; H04N 19/174; H04N 19/176;
H04N 19/136; H04N 19/172

(13) B

(21) 1-2022-05475

(22) 23/02/2021

(86) PCT/US2021/019235 23/02/2021

(87) WO 2021/173552 02/09/2021

(30) 62/980,963 24/02/2020 US

(45) 25/09/2025 450

(43) 26/12/2022 417A

(73) BYTEDANCE INC. (US)

12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California
90066, USA

(72) WANG, Ye-kui (US); ZHANG, Li (CN); ZHANG, Kai (CN).

(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP XỬ LÝ VIDEO, THIẾT BỊ XỬ LÝ DỮ LIỆU VIDEO VÀ VẬT
GHI MÁY TÍNH

(21) 1-2022-05475

(57) Sáng chế đề xuất một số kỹ thuật để mã hóa video và giải mã video. Một phương pháp ví dụ bao gồm bước tiến hành xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video có nhỏ hơn chiều cao của hàng ô ảnh của ảnh video; và thực hiện, bằng cách sử dụng sự xác định này, bước biến đổi giữa video và dòng bit của video.

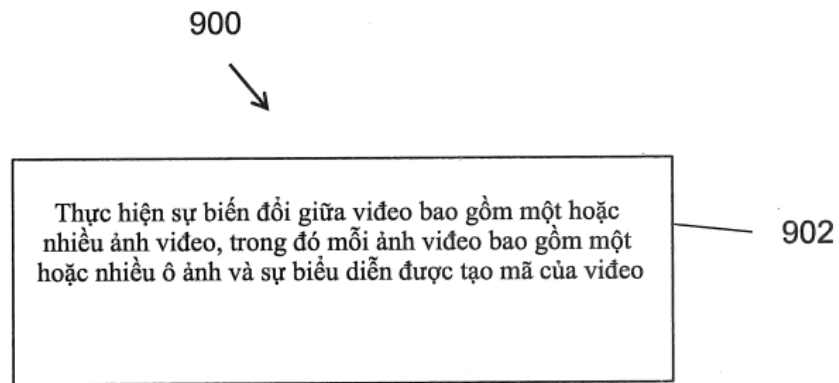


FIG. 7

Lĩnh vực kỹ thuật được đề cập

Sáng chế liên quan tới việc tạo mã và giải mã hình ảnh và video.

Tình trạng kỹ thuật của sáng chế

Video kỹ thuật số sử dụng băng thông lớn nhất trên internet và các mạng truyền thông kỹ thuật số khác. Do số lượng các thiết bị người dùng được kết nối có khả năng nhận và hiển thị video tăng lên, dự kiến rằng yêu cầu về băng thông dành cho việc sử dụng video kỹ thuật số sẽ tiếp tục tăng trưởng.

Bản chất kỹ thuật của sáng chế

Phần mô tả này bộc lộ các kỹ thuật mà có thể được sử dụng bởi các bộ mã hóa và các bộ giải mã video để xử lý sự biểu diễn được tạo mã của video bằng cách sử dụng thông tin điều khiển hữu ích để giải mã sự biểu diễn được tạo mã.

Theo một khía cạnh ví dụ, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều cột ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số cột ô ảnh được rút ra đối với mỗi cột đơn vị cây tạo mã (CTU-coding tree unit) của ô ảnh của ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều hàng ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số hàng ô ảnh được rút ra đối với mỗi hàng đơn vị cây tạo mã (CTU) của ô ảnh của ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm ít nhất một ảnh video và dòng bit của video theo quy tắc, trong đó ít nhất một ảnh video bao gồm một hoặc nhiều lát và một hoặc nhiều ảnh con, và trong đó quy tắc chỉ ra rằng thứ tự của các chỉ số lát của một hoặc nhiều lát trong ít nhất một ảnh video được chỉ báo đáp lại phần tử cú pháp liên kết với ít nhất một ảnh chỉ báo về việc liệu một lát có được bao gồm theo từng ảnh con của ít nhất một ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa đơn vị video của vùng video của video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra rằng thông tin điều khiển thứ nhất ở bậc thứ nhất vùng video trong dòng bit điều khiển việc liệu thông tin điều khiển thứ hai có được bao gồm ở bậc thứ hai của đơn vị video trong dòng bit, trong đó bậc thứ hai là nhỏ hơn bậc thứ nhất, trong đó thông tin điều khiển thứ nhất và thông tin điều khiển thứ hai bao gồm thông tin về việc liệu hoặc cách thức công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS- luma mapping and chroma scaling) có được áp dụng cho đơn vị video, và trong đó công cụ LMCS bao gồm việc sử dụng quá trình chọn thang dư sắc độ (CRS - chroma residual scaling), hoặc quá trình tái tạo dạng độ sáng (RP – luma reshaping) để biến đổi.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS) là được phép khi phần tử cú pháp thứ nhất trong tập thông số dãy được tham chiếu chỉ báo rằng công cụ LMCS là được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng khi phần tử cú pháp thứ nhất chỉ báo rằng công cụ LMCS là không được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS là được phép đối với tất cả các lát liên kết với tiêu đề ảnh của ảnh video khi phần tử cú pháp thứ hai trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề ảnh của video, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng đối với tất cả các lát liên kết với tiêu đề ảnh khi phần tử cú pháp thứ hai chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề ảnh

của video, trong đó quy tắc chỉ ra rằng công cụ LMCS được sử dụng đối với lát hiện tại liên kết với tiêu đề lát của ảnh video khi phần tử cú pháp thứ ba được bao gồm theo cách có lựa chọn trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề lát của video, và trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng đối với lát hiện tại khi phần tử cú pháp thứ ba chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề lát của video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng liệu độ phân giải hiệu vector chuyển động thích ứng (AMVR- adaptive motion vector difference resolution) có được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin dựa trên phần tử cú pháp được bao gồm theo cách có lựa chọn trong tập thông số dãy được tham chiếu (SPS) mà chỉ báo liệu AMVR có được phép hay không, trong đó quy tắc chỉ ra rằng AMVR không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin khi phần tử cú pháp chỉ báo rằng AMVR là không được phép, và trong đó quy tắc chỉ ra rằng AMVR được suy ra là không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin khi phần tử cú pháp khi phần tử cú pháp không được bao gồm trong SPS.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm ảnh video và dòng bit của video theo quy tắc, trong đó ảnh video bao gồm ảnh con, ô ảnh, và lát, và trong đó quy tắc chỉ ra rằng, do ảnh con bao gồm lát mà được phân vùng từ ô ảnh, sự biến đổi được thực hiện bằng cách không đếm chiều cao của ảnh con bằng cách sử dụng số lượng của các ô ảnh của ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm ảnh video và dòng bit của video, trong đó dòng bit chỉ báo chiều cao của ảnh con của ảnh video mà được đếm dựa trên số lượng của các đơn vị cây tạo mã (các CTU) của ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp bao gồm bước tiến hành xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video có nhỏ hơn chiều cao của hàng ô ảnh của ảnh video; và thực hiện, bằng cách sử dụng sự xác định, sự biến đổi giữa video và dòng bit của video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra thông tin thứ nhất mà được bảo tín hiệu trong sự biểu diễn được tạo mã và thông tin thứ hai mà được rút ra từ sự biểu diễn được tạo mã, trong đó ít nhất thông tin thứ nhất hoặc thông tin thứ hai liên quan tới các chỉ số hàng hoặc các chỉ số cột của một hoặc nhiều ô ảnh.

Theo khía cạnh ví dụ khác, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa đơn vị video của vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng thông tin điều khiển thứ nhất ở vùng video điều khiển việc liệu thông tin điều khiển thứ hai có được bao gồm ở bậc đơn vị video; trong đó thông tin điều khiển thứ nhất và/hoặc thông tin điều khiển thứ hai bao gồm thông tin về ánh xạ độ sáng và chọn thang sắc độ (LMCS) hoặc chọn thang dư sắc độ (CRS) hoặc quá trình tái tạo dạng (RP) được sử dụng để biến đổi.

Theo khía cạnh ví dụ khác nữa, thiết bị mã hóa video được bộc lộ. Bộ mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện các phương pháp nêu trên.

Theo khía cạnh ví dụ khác nữa, thiết bị giải mã video được bộc lộ. Bộ giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện các phương pháp nêu trên.

Theo khía cạnh ví dụ khác nữa, môi trường máy tính đọc được có mã được lưu trong đó được bộc lộ. Mã này thực hiện một trong số các phương pháp được mô tả ở đây dưới dạng mã bộ xử lý thực thi được.

Các dấu hiệu này, và khác nữa được mô tả trong phần mô tả này.

Mô tả vắn tắt các hình vẽ

FIG. 1 thể hiện ví dụ về việc phân vùng lát quét màn hình của ảnh, trong đó ảnh được chia thành 12 ô ảnh và 3 lát quét màn hình.

FIG. 2 thể hiện ví dụ về việc phân vùng lát hình chữ nhật của ảnh, trong đó ảnh được chia thành 24 ô ảnh (6 cột ô ảnh và 4 hàng ô ảnh) và 9 lát hình chữ nhật.

FIG. 3 thể hiện ví dụ về ảnh được phân vùng thành các ô ảnh và các lát hình chữ nhật, trong đó ảnh được chia thành 4 ô ảnh (2 cột ô ảnh và 2 hàng ô ảnh) và 4 lát hình chữ nhật.

FIG. 4 thể hiện ảnh mà được phân vùng thành 15 ô ảnh, 24 lát và 24 ảnh con.

FIG. 5 là sơ đồ khối của hệ thống xử lý video ví dụ.

FIG. 6 là sơ đồ khối của thiết bị xử lý video.

FIG. 7 là lưu đồ của phương pháp xử lý video ví dụ.

FIG. 8 là sơ đồ khối mà minh họa hệ thống tạo mã video theo một số phương án của sáng chế.

FIG. 9 là sơ đồ khối mà minh họa bộ mã hóa theo một số phương án của sáng chế.

FIG. 10 là sơ đồ khối mà minh họa bộ giải mã theo một số phương án của sáng chế.

Các hình vẽ từ FIG. 11 tới FIG. 19 là các lưu đồ của các phương pháp xử lý video ví dụ.

Mô tả chi tiết sáng chế

Các đầu đề của các phần được sử dụng trong bản mô tả để làm cho dễ hiểu và không giới hạn khả năng ứng dụng của các kỹ thuật và các phương án được mô tả trong mỗi phần chỉ trong phần đó. Hơn nữa, các thuật ngữ H.266 được sử

dụng trong một số phần mô tả chỉ để làm cho dễ hiểu và không nhằm giới hạn phạm vi của các kỹ thuật được mô tả. Như vậy, các kỹ thuật được mô tả ở đây cũng có thể được áp dụng cho các giao thức tạo mã-giải mã video và các thiết kế khác. Trong bản mô tả, các thay đổi biên tập được thể hiện trong văn bản bằng các ngoặc kép mở và đóng (ví dụ, [[]]) mà chỉ báo đoạn văn bản bị loại bỏ trong giữa các ngoặc kép, và đoạn văn bản in nghiêng đậm mà chỉ báo đoạn văn bản được thêm vào, so với bản soạn thảo hiện tại của đặc tả VVC.

Tổng quan

Phần mô tả này liên quan tới các công nghệ tạo mã video. Cụ thể là, về sự hỗ trợ của các ảnh con, LMCS, và AMVR. Các khía cạnh đối với các ảnh con bao gồm việc rút ra số lượng của các hàng ô ảnh và các cột ô ảnh được bao gồm trong một ảnh con cũng như việc rút ra danh sách của các địa chỉ CTU quét mảnh đối với các CTU được bao gồm trong lát khi mỗi ảnh con chỉ chứa một lát. Các khía cạnh đối với LMCS là về sự bảo tín hiệu của việc cho phép LMCS trên các bậc khác nhau. Các khía cạnh đối với AMVR là về các ngữ nghĩa của `sps_affine_amvr_enabled_flag`. Các ý tưởng này có thể được áp dụng một cách riêng rẽ hoặc theo cách kết hợp khác nhau, đối với chuẩn tạo mã video bất kỳ hoặc bộ tạo mã-giải mã video phi chuẩn mà hỗ trợ tạo mã video một lớp và/hoặc nhiều lớp, ví dụ, chuẩn tạo mã video đa năng (VVC-Versatile Video Coding) đang được phát triển.

Các chữ viết tắt

ALF	(Adaptive Loop Filter) Bộ lọc vòng thích ứng
AMVR	(Adaptive Motion Vector difference Resolution) Độ phân giải hiệu vector chuyển động thích ứng
APS	(Adaptation Parameter Set) Tập thông số thích ứng
AU	(Access Unit) Đơn vị truy nhập
AUD	(Access Unit Delimiter) Dấu cách đơn vị truy nhập
AVC	(Advanced Video Coding) Tạo mã video nâng cao
CLVS	(Coded Layer Video Sequence) Dãy video lớp được tạo mã
CPB	(Coded Picture Buffer) Bộ đệm ảnh được tạo mã
CRA	(Clean Random Access) Truy nhập ngẫu nhiên sạch

CTU	(Coding Tree Unit) Đơn vị cây tạo mã
CVS	(Coded Video Sequence) Dãy video được tạo mã
DPB	(Decoded Picture Buffer) Bộ đệm ảnh được giải mã
DPS	(Decoding Parameter Set) Tập thông số giải mã
EOB	(End Of Bitstream) Cuối dòng bit
EOS	(End Of Sequence) Cuối dãy
GDR	(Gradual Decoding Refresh) Làm mới giải mã từng bước
HEVC	(High Efficiency Video Coding) Tạo mã video hiệu quả cao
HRD	(Hypothetical Reference Decoder) Bộ giải mã tham chiếu giả
định	
IDR	(Instantaneous Decoding Refresh) Làm mới giải mã tức thời
JEM	(Joint Exploration Model) Mô hình khảo sát chung
LMCS	(Luma Mapping with Chroma Scaling) Ánh xạ độ sáng với chọn
thang sắc độ	
MCTS	(Motion-Constrained Tile Sets) Các tập ô ảnh hạn chế chuyển
động	
NAL	(Network Abstraction Layer) Lớp trừu tượng mạng
OLS	(Output Layer Set) Tập lớp đầu ra
PH(Picture Header)	Tiêu đề ảnh
PPS	(Picture Parameter Set) Tập thông số ảnh
PTL	(Profile, Tier and Level) Tiểu sử, Cấp và Bậc
PU(Picture Unit)	Đơn vị ảnh
RBSP	(Raw Byte Sequence Payload) Tải dãy byte thô
SEI	(Supplemental Enhancement Information) Thông tin tăng
cường bổ sung	
SPS	(Sequence Parameter Set) Tập thông số dãy
SVC	(Scalable Video Coding) Tạo mã video khả mở rộng
VCL	(Video Coding Layer) Lớp tạo mã video
VPS	(Video Parameter Set) Tập thông số video
VTM	(VVC Test Model) Mô hình thử VVC

VUI (Video Usability Information) Thông tin tính khả dụng của video

VVC (Versatile Video Coding) Tạo mã video đa năng

Mở đầu

Các chuẩn tạo mã video đã tiến hóa chủ yếu thông qua sự phát triển của các chuẩn ITU-T và ISO/IEC đã biết. ITU-T đã đề xuất H.261 và H.263, ISO/IEC đã đề xuất MPEG-1 và MPEG-4 Visual, và hai tổ chức này cùng đề xuất các chuẩn H.262/MPEG-2 Video và H.264/MPEG-4 Advanced Video Coding (AVC) và H.265/HEVC. Kể từ chuẩn H.262, các chuẩn tạo mã video dựa trên cấu trúc tạo mã video lai trong đó việc dự báo theo thời gian cộng tạo mã chuyển đổi được sử dụng. Để tìm kiếm các công nghệ tạo mã video tương lai sau HEVC, Joint Video Exploration Team (JVET) đã được đồng thành lập bởi VCEG và MPEG vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp nhận bởi JVET và được đưa vào phần mềm tham chiếu tên là Joint Exploration Model (JEM). Hội nghị JVET được đồng thời tổ chức mỗi lần một quý, và chuẩn tạo mã mới nhằm giảm 50% tốc độ bit so với HEVC. Chuẩn tạo mã video mới được đặt tên chính thức là chuẩn tạo mã video đa năng (VVC) trong hội nghị của JVET vào tháng tư năm 2018, và phiên bản thứ nhất của mô hình thử VVC (VTM) đã được phát hành vào lúc đó. Do có các nỗ lực liên tục đóng góp vào việc chuẩn hóa VVC, các kỹ thuật tạo mã mới được chấp nhận cho chuẩn VVC trong mỗi hội nghị JVET. Dự thảo làm việc VVC và mô hình thử VTM được cập nhật sau mỗi hội nghị. Hiện tại dự án VVC nhằm hoàn thiện kỹ thuật (FDIS) tại hội nghị tháng bảy năm 2020.

Các sơ đồ phân vùng ảnh trong HEVC

HEVC bao gồm bốn sơ đồ phân vùng ảnh khác nhau, mà chính là các lát bình thường, các lát phụ thuộc, các ô ảnh, và xử lý song song đầu sóng (WPP - Wavefront Parallel Processing), mà có thể được áp dụng đối với việc so khớp kích thước đơn vị truyền cực đại (MTU - Maximum Transfer Unit), xử lý song song, và độ trễ cuối đến cuối giảm thấp.

Các lát bình thường là tương tự như trong H.264/AVC. Mỗi lát bình thường được đóng gói trong đơn vị NAL của chính nó, và dự báo trong ảnh (dự báo mẫu

trong, dự báo thông tin chuyển động, dự báo chế độ tạo mã) và sự phụ thuộc tạo mã entropi ngang qua các biên lát là không được phép. Như vậy lát bình thường có thể được tái lập một cách độc lập từ các lát bình thường khác bên trong cùng một ảnh (mặc dù vẫn có thể có các sự phụ thuộc lẫn nhau do các hoạt động lọc vòng).

Lát bình thường là công cụ duy nhất mà có thể được sử dụng để song song hóa mà cũng khả dụng, dưới dạng đồng nhất theo cách ảo, trong H.264/AVC. Sự song song hóa dựa trên các lát bình thường không yêu cầu nhiều sự truyền thông liên bộ xử lý hoặc liên lõi (ngoại trừ chia sẻ dữ liệu liên bộ xử lý hoặc liên lõi để bù chuyển động khi giải mã ảnh được tạo mã dự báo, mà thường là nặng hơn nhiều việc chia sẻ dữ liệu liên bộ xử lý hoặc liên lõi do dự báo trong ảnh). Tuy nhiên, vì cùng lý do này, việc sử dụng các lát bình thường có thể phải chịu tổng phí tạo mã đáng kể do chi phí bit của tiêu đề lát và do thiếu dự báo ngang qua các biên lát. Hơn nữa, các lát bình thường (trái ngược với các công cụ khác được đề cập dưới đây) cũng dùng làm cơ chế chính để phân vùng dòng bit nhằm đáp ứng các yêu cầu kích thước MTU, do sự độc lập trong ảnh của các lát bình thường và rằng mỗi lát bình thường được đóng bao trong đơn vị NAL của chính nó. Trong nhiều trường hợp, mục đích của song song hóa và mục đích của so khớp kích thước MTU đặt ra các đòi hỏi trái ngược đối với sơ đồ bố trí lát trong ảnh. Việc hiện thực hóa tình huống này đã dẫn đến sự phát triển của các công cụ song song hóa được đề cập dưới đây.

Các lát phụ thuộc có các tiêu đề lát ngắn và cho phép phân vùng của dòng bit ở các biên khối cây mà không phá vỡ dự báo trong ảnh bất kỳ. Về cơ bản, các lát phụ thuộc tạo ra sự phân đoạn các lát bình thường thành nhiều đơn vị NAL, để tạo ra độ trễ cuối đến cuối giảm thấp bằng cách cho phép một phần của lát bình thường được gửi ra trước khi việc mã hóa toàn bộ lát bình thường được kết thúc.

Trong WPP, ảnh được phân vùng thành các hàng đơn của các khối cây tạo mã (CTB – coding tree block). Giải mã entropi và dự báo được cho phép để sử dụng dữ liệu từ các CTB trong các phân vùng khác. Việc xử lý song song là khả thi thông qua giải mã song song các hàng CTB, khi mà sự bắt đầu giải mã của

hàng CTB được làm trễ đi hai CTB, để đảm bảo rằng dữ liệu liên quan tới CTB bên trên và bên phải của CTB đối tượng là khả dụng trước khi CTB đối tượng được giải mã. Bằng cách sử dụng sự bắt đầu so le này (mà trông giống như đầu sóng khi được biểu diễn theo cách đồ họa), việc song song hóa là khả thi với bao nhiêu bộ xử lý/lỗi thì ảnh chứa bấy nhiêu hàng CTB. Do dự báo trong ảnh giữa các hàng khối cây lân cận bên trong ảnh được cho phép, sự truyền thông liên bộ xử lý/liên lỗi theo yêu cầu cho phép dự báo trong ảnh có thể là đáng kể. Việc phân vùng WPP không dẫn đến việc tạo ra các đơn vị NAL bổ sung so với khi không được áp dụng, như vậy WPP không phải công cụ để so khớp kích thước MTU. Tuy nhiên, nếu việc so khớp kích thước MTU là cần thiết, thì các lát bình thường có thể được sử dụng với WPP, với tổng phí tạo mã đã biết.

Các ô ảnh xác định các biên ngang và dọc mà phân vùng ảnh thành các cột ô ảnh và các hàng ô ảnh. Cột ô ảnh chạy từ đỉnh của ảnh tới đáy của ảnh. Tương tự, hàng ô ảnh chạy từ bên trái của ảnh tới bên phải của ảnh. Số lượng của các ô ảnh trong ảnh có thể được rút ra một cách đơn giản là số lượng của các cột ô ảnh nhân với số lượng của các hàng ô ảnh.

Thứ tự quét của các CTB được thay đổi để là nội vùng bên trong ô ảnh (theo thứ tự quét mảnh CTB của ô ảnh), trước khi giải mã CTB đỉnh-trái của ô ảnh tiếp theo theo thứ tự của quét mảnh ô ảnh của ảnh. Tương tự với các lát bình thường, các ô ảnh phá vỡ các phụ thuộc dự báo trong ảnh cũng như các phụ thuộc giải mã entropi. Tuy nhiên, chúng không cần phải được bao gồm vào trong các đơn vị NAL riêng lẻ (về mặt này thì giống như WPP); cho nên các ô ảnh không thể được sử dụng để so khớp kích thước MTU. Mỗi ô ảnh có thể được xử lý bởi một bộ xử lý/lỗi, và việc truyền thông liên bộ xử lý/liên lỗi mà cần thiết để dự báo trong ảnh giữa các đơn vị xử lý mà giải mã các ô ảnh lân cận được giới hạn ở việc chuyển tiêu đề lát được chia sẻ trong các trường hợp lát trải rộng hơn một ô ảnh, và việc chia sẻ liên quan đến lọc vòng của các mẫu được tái lập và siêu dữ liệu. Khi nhiều hơn một ô ảnh hoặc đoạn WPP được bao gồm trong lát, thì dịch vị byte điểm nhập đối với mỗi ô ảnh hoặc đoạn WPP mà khác với ô ảnh hoặc đoạn thứ nhất trong lát được báo tín hiệu trong tiêu đề lát.

Để đơn giản hóa, các hạn chế về việc áp dụng bốn sơ đồ phân vùng ảnh khác nhau đã được chỉ ra trong HEVC. Dãy video được tạo mã cho trước không thể bao gồm cả các ô ảnh lẫn các đầu sóng đối với phần lớn trong số các tiểu sử được chỉ ra trong HEVC. Đối với mỗi lát và ô ảnh, một trong hai hoặc cả hai điều kiện sau đây phải được thỏa mãn: 1) tất cả các khối cây được tạo mã trong lát thuộc về cùng một ô ảnh; 2) tất cả các khối cây được tạo mã trong ô ảnh thuộc về cùng một lát. Cuối cùng, đoạn đầu sóng chứa chính xác một hàng CTB, và khi WPP được sử dụng, nếu lát bắt đầu bên trong hàng CTB, nó phải kết thúc trong cùng hàng CTB đó.

Sửa đổi gần đây đối với HEVC được chỉ ra trong tài liệu phát hành của JCT-VC JCTVC-AC1005, J. Boyce, A. Ramasubramanian, R. Skupin, G. J. Sullivan, A. Tourapis, Y.-K. Wang (editors), "HEVC Additional Supplemental Enhancement Information (Draft 4)," Oct. 24, 2017, sẵn có công khai ở đây: http://phenix.int-evry.fr/jct/doc_end_user/documents/29_Macau/wg11/JCTVC-AC1005-v2.zip. Với sửa đổi này được đưa vào, HEVC chỉ ra ba thông điệp SEI liên quan với MCTS, mà chính là thông điệp SEI MCTS thời gian, thông điệp SEI tập thông tin trích MCTS, và thông điệp SEI lồng thông tin trích MCTS.

Thông điệp SEI MCTS thời gian chỉ báo sự tồn tại của các MCTS trong dòng bit và báo tín hiệu các MCTS. Đối với mỗi MCTS, các vector chuyển động được hạn chế để trở tới các vị trí mẫu toàn phần nằm trong MCTS và tới các vị trí mẫu phân số mà chỉ yêu cầu các vị trí mẫu toàn phần nằm trong MCTS để nội suy, và việc sử dụng các ứng viên vector chuyển động để dự báo vector chuyển động theo thời gian được rút ra từ các khối nằm ngoài MCTS là không được cho phép. Theo cách này, mỗi MCTS có thể được giải mã một cách độc lập mà không cần sự tồn tại của các ô ảnh không được bao gồm trong MCTS.

Thông điệp SEI các tập thông tin trích MCTS cung cấp thông tin bổ sung mà có thể được sử dụng trong việc trích dòng bit con MCTS (được chỉ ra như một phần của các ngữ nghĩa của thông điệp SEI) để tạo ra dòng bit phù hợp đối với tập MCTS. Thông tin này bao gồm số lượng các tập thông tin trích, mỗi tập xác định số lượng các tập MCTS và chứa các byte RBSP của các VPS, các SPS, và các PPS thay thế cần được sử dụng trong suốt quá trình trích dòng bit con MCTS.

Khi trích dòng bit con theo quá trình trích dòng bit con MCTS, các tập thông số (các VPS, các SPS, và các PPS) cần phải được viết lại hoặc được thay thế, các tiêu đề lát cần phải được cập nhật một chút do một hoặc tất cả các phần tử cú pháp liên quan đến địa chỉ lát (bao gồm `first_slice_segment_in_pic_flag` và `slice_segment_address`) thường phải có các giá trị khác nhau.

Phân vùng các ảnh trong VVC

Trong VVC, ảnh được chia thành một hoặc nhiều hàng ô ảnh và một hoặc nhiều cột ô ảnh. Ô ảnh là dãy của các CTU mà bao phủ vùng hình chữ nhật của ảnh. Các CTU trong ô ảnh được quét theo thứ tự quét mảnh bên trong ô ảnh đó.

Lát chứa số nguyên các ô ảnh hoàn chỉnh hoặc số nguyên các hàng CTU hoàn chỉnh liên tiếp bên trong ô ảnh của ảnh.

Hai chế độ của các lát được hỗ trợ, mà chính là chế độ lát quét mảnh và chế độ lát hình chữ nhật. Trong chế độ lát quét mảnh, lát chứa dãy của các ô ảnh hoàn chỉnh trong quét mảnh ô ảnh của ảnh. Trong chế độ lát hình chữ nhật, lát chứa hoặc số lượng các ô ảnh hoàn chỉnh mà cùng nhau tạo thành vùng hình chữ nhật của ảnh hoặc số lượng các hàng CTU hoàn chỉnh liên tiếp của một ô ảnh mà cùng nhau tạo thành vùng hình chữ nhật của ảnh. Các ô ảnh bên trong lát hình chữ nhật được quét theo thứ tự quét mảnh ô ảnh bên trong vùng hình chữ nhật tương ứng với lát đó.

Ảnh con chứa một hoặc nhiều lát mà cùng nhau bao phủ vùng hình chữ nhật của ảnh.

FIG. 1 thể hiện ví dụ về việc phân vùng lát quét mảnh của ảnh, trong đó ảnh được chia thành 12 ô ảnh và 3 lát quét mảnh.

FIG. 2 thể hiện ví dụ về việc phân vùng lát hình chữ nhật của ảnh, trong đó ảnh được chia thành 24 ô ảnh (6 cột ô ảnh và 4 hàng ô ảnh) và 9 lát hình chữ nhật.

FIG. 3 thể hiện ví dụ của ảnh được phân vùng thành các ô ảnh và các lát hình chữ nhật, trong đó ảnh được chia thành 4 ô ảnh (2 cột ô ảnh và 2 hàng ô ảnh) và 4 lát hình chữ nhật.

FIG. 4 thể hiện ví dụ về phân vùng ảnh con của ảnh, trong đó ảnh được phân vùng thành 18 ô ảnh, 12 ở bên tay trái mỗi ô bao phủ một lát gồm 4 nhân 4 CTU

và 6 ô ảnh ở bên tay phải mỗi ô bao phủ 2 lát được xếp chồng theo chiều dọc gồm 2 nhân 2 CTU, tất cả tạo ra 24 lát và 24 ảnh con có các kích thước khác nhau (mỗi lát là một ảnh con).

Sự báo tín hiệu của SPS/PPS/Tiêu đề ảnh/Tiêu đề lát trong VVC (như JVET-Q2001-vC)

7.3.2.3 Cú pháp RBSP tập thông số dãy

	Mô tả
seq_parameter_set_rbsp() {	
sps_seq_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)
sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <= sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	

}	
sps_subpic_id_len_minus1	ue(v)
subpic_id_mapping_explicitly_signalled_flag	u(1)
if(subpic_id_mapping_explicitly_signalled_flag) {	
subpic_id_mapping_in_sps_flag	u(1)
if(subpic_id_mapping_in_sps_flag)	
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
bit_depth_minus8	ue(v)
sps_entropy_coding_sync_enabled_flag	u(1)
if(sps_entropy_coding_sync_enabled_flag)	
sps_wpp_entry_point_offsets_present_flag	u(1)
sps_weighted_pred_flag	u(1)
sps_weighted_bipred_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	u(4)
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_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)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	

if(qtbt dual tree intra flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
if(ChromaArrayType != 0) {	
sps_joint_cbr_enabled_flag	u(1)
same_qp_table_for_chroma	u(1)
numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbr_enabled_flag ? 3 : 2)	
for(i = 0; i < numQpTables; i++) {	
qp_table_start_minus26[i]	se(v)
num_points_in_qp_table_minus1[i]	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
delta_qp_in_val_minus1[i][j]	ue(v)
delta_qp_diff_val[i][j]	ue(v)
}	
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
if(sps_alf_enabled_flag && ChromaArrayType != 0)	
sps_ccalf_enabled_flag	u(1)
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag) {	
log2_transform_skip_max_size_minus2	ue(v)
sps_bdpcm_enabled_flag	u(1)
}	
sps_ref_wraparound_enabled_flag	u(1)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
if(sps_bdof_enabled_flag)	
sps_bdof_pic_present_flag	u(1)
sps_smvd_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
if(sps_dmvr_enabled_flag)	
sps_dmvr_pic_present_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(chroma_format_idc == 1) {	
sps_chroma_horizontal_collocated_flag	u(1)
sps_chroma_vertical_collocated_flag	u(1)
}	

sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
six_minus_max_num_merge_cand	ue(v)
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
five_minus_max_num_subblock_merge_cand	ue(v)
sps_affine_type_flag	u(1)
if(sps_amvr_enabled_flag)	
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	
sps_palette_enabled_flag	u(1)
if(ChromaArrayType == 3 && !sps_max_luma_transform_size_64_flag)	
sps_act_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_minus4	ue(v)
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
if(sps_ibc_enabled_flag)	
six_minus_max_num_ibc_merge_cand	ue(v)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	
sps_lmcs_enabled_flag	u(1)
sps_lfst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
log2_parallel_merge_level_minus2	ue(v)
sps_scaling_list_enabled_flag	u(1)
sps_dep_quant_enabled_flag	u(1)
if(!sps_dep_quant_enabled_flag)	
sps_sign_data_hiding_enabled_flag	u(1)
sps_virtual_boundaries_enabled_flag	u(1)
if(sps_virtual_boundaries_enabled_flag) {	
sps_virtual_boundaries_present_flag	u(1)

if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)
if(sps_general_hrd_params_present_flag) {	
general_hrd_parameters()	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_cpb_params_present_flag	u(1)
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 : sps_max_sublayers_minus1	
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)	
}	
}	
field_seq_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters() /* Được chỉ ra trong ITU-T H.SEI ISO/IEC 23002-7 */	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

7.3.2.4 Cú pháp RBSP tập thông số ảnh

	Mô tả
pic_parameter_set_rbsp() {	
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
mixed_nalu_types_in_pic_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_conformance_window_flag	u(1)
if(pps_conformance_window_flag) {	
pps_conf_win_left_offset	ue(v)
pps_conf_win_right_offset	ue(v)
pps_conf_win_top_offset	ue(v)
pps_conf_win_bottom_offset	ue(v)
}	
scaling_window_explicit_signalling_flag	u(1)
if(scaling_window_explicit_signalling_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)

subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	
rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 && (tile_idx_delta_present_flag tileIdx % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {	
num_exp_slices_in_tile[i]	ue(v)
numExpSlicesInTiles = num_exp_slices_in_tile[i]	
for(j = 0; j < numExpSlicesInTiles; j++)	
exp_slice_height_in_ctus_minus1[j]	ue(v)
i += NumSlicesInTiles[i]	
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
cu_qp_delta_enabled_flag	u(1)
pps_chroma_tool_offsets_present_flag	u(1)

if(pps chroma tool offsets present flag) {	
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset_present_flag	u(1)
if(pps_joint_cbr_qp_offset_present_flag)	
pps_joint_cbr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i=0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbr_qp_offset_present_flag)	
joint_cbr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
rpl_info_in_ph_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
sao_info_in_ph_flag	u(1)
alf_info_in_ph_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl_present_in_ph_flag)	
wp_info_in_ph_flag	u(1)
qp_delta_info_in_ph_flag	u(1)
pps_ref_wraparound_enabled_flag	u(1)
if(pps_ref_wraparound_enabled_flag)	
pps_ref_wraparound_offset	ue(v)
picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)

rbsp_trailing_bits()	
}	

7.3.2.7 Cú pháp cấu trúc tiêu đề ảnh

picture_header_structure() {	Mô tả
gdr or irap pic flag	u(1)
if(gdr or irap pic flag)	
gdr pic flag	u(1)
ph inter slice allowed flag	u(1)
if(ph inter slice allowed flag)	
ph intra slice allowed flag	u(1)
non reference picture flag	u(1)
ph pic parameter set id	ue(v)
ph pic order cnt lsb	u(v)
if(gdr or irap pic flag)	
no output of prior pics flag	u(1)
if(gdr pic flag)	
recovery poc cnt	ue(v)
for(i = 0; i < NumExtraPhBits; i++)	
ph extra bit[i]	u(1)
if(sps poc msb flag) {	
ph poc msb present flag	u(1)
if(ph poc msb present flag)	
poc msb val	u(v)
}	
if(sps alf enabled flag && alf_info_in_ph flag) {	
ph alf enabled flag	u(1)
if(ph alf enabled flag) {	
ph num alf aps ids luma	u(3)
for(i = 0; i < ph num alf aps ids luma; i++)	
ph alf aps id luma[i]	u(3)
if(ChromaArrayType != 0)	
ph alf chroma idc	u(2)
if(ph alf chroma idc > 0)	
ph alf aps id chroma	u(3)
if(sps ccalf enabled flag) {	
ph cc alf cb enabled flag	u(1)
if(ph cc alf cb enabled flag)	
ph cc alf cb aps id	u(3)
ph cc alf cr enabled flag	u(1)
if(ph cc alf cr enabled flag)	
ph cc alf cr aps id	u(3)
}	
}	
}	
if(sps lmcs enabled flag) {	
ph lmcs enabled flag	u(1)
if(ph lmcs enabled flag) {	
ph lmcs aps id	u(2)
if(ChromaArrayType != 0)	
ph chroma residual scale flag	u(1)
}	
}	

if(sps_scaling_list_enabled_flag) {	
ph_scaling_list_present_flag	u(1)
if(ph_scaling_list_present_flag)	
ph_scaling_list_aps_id	u(3)
}	
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {	
ph_virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(rpl_info_in_ph_flag)	
ref_pic_lists()	
if(partition_constraints_override_enabled_flag)	
partition_constraints_override_flag	u(1)
if(ph_intra_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
ph_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
ph_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
ph_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(qtbt_dual_tree_intra_flag) {	
ph_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
ph_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
ph_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
ph_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(ph_inter_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph_log2_diff_min_qt_min_cb_inter_slice	ue(v)
ph_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(ph_max_mtt_hierarchy_depth_inter_slice != 0) {	
ph_log2_diff_max_bt_min_qt_inter_slice	ue(v)
ph_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	

if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag) {	
ph_temporal_mvp_enabled_flag	u(1)
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
}	
mvd_l1_zero_flag	u(1)
if(sps_fpel_mmvd_enabled_flag)	
ph_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
ph_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
ph_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
ph_disable_prof_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag)	
pred_weight_table()	
}	
if(qp_delta_info_in_ph_flag)	
ph_qp_delta	se(v)
if(sps_joint_cbr_enabled_flag)	
ph_joint_cbr_sign_flag	u(1)
if(sps_sao_enabled_flag && sao_info_in_ph_flag) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
if(sps_dep_quant_enabled_flag)	
ph_dep_quant_enabled_flag	u(1)
if(sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	

}	
if(picture header extension present flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
}	

7.3.7.1 Cú pháp tiêu đề lát chung

slice_header() {	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)
if(ph_inter_slice_allowed_flag)	
slice_type	ue(v)
if(sps_alf_enabled_flag && !alf_info_in_ph_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
slice_cc_alf_cb_enabled_flag	u(1)
if(slice_cc_alf_cb_enabled_flag)	
slice_cc_alf_cb_aps_id	u(3)
slice_cc_alf_cr_enabled_flag	u(1)
if(slice_cc_alf_cr_enabled_flag)	
slice_cc_alf_cr_aps_id	u(3)
}	
}	
}	
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(!rpl_info_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))	
ref_pic_lists()	

if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && (slice_type != I && num_ref_entries[0][RplIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplIdx[1]] > 1)) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
if(num_ref_entries[i][RplIdx[i]] > 1)	
num_ref_idx_active_minus1[i]	ue(v)
}	
if(slice_type != I) {	
if(cabac_init_present_flag)	
cabac_init_flag	u(1)
if(ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag) {	
if(slice_type == B)	
slice_collocated_from_l0_flag	u(1)
if((slice_collocated_from_l0_flag && NumRefIdxActive[0] > 1) (! slice_collocated_from_l0_flag && NumRefIdxActive[1] > 1))	
slice_collocated_ref_idx	ue(v)
}	
if(!wp_info_in_ph_flag && ((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)))	
pred_weight_table()	
}	
if(!qp_delta_info_in_ph_flag)	
slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(sps_joint_cbr_enabled_flag)	
slice_joint_cbr_qp_offset	se(v)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)
if(sps_sao_enabled_flag && !sao_info_in_ph_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)	
slice_deblocking_filter_override_flag	u(1)
if(slice_deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
slice_cb_beta_offset_div2	se(v)
slice_cb_tc_offset_div2	se(v)
slice_cr_beta_offset_div2	se(v)
slice_cr_tc_offset_div2	se(v)
}	
}	
slice_ts_residual_coding_disabled_flag	u(1)

if(ph_lmcs_enabled_flag)	
slice_lmcs_enabled_flag	u(1)
if(pic_scaling_list_enabled_flag)	
slice_scaling_list_present_flag	u(1)
if(NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Các đặc tả trong JVET-Q2001-vC đối với các ô ảnh, các lát và các ảnh con

3 Các định nghĩa

chỉ số lát bậc ảnh: chỉ số của lát tới danh sách của các lát trong ảnh theo thứ tự như chúng được báo tín hiệu trong PPS khi rect_slice_flag là bằng 1.

chỉ số lát bậc ảnh con: chỉ số của lát tới danh sách của các lát trong ảnh con theo thứ tự như chúng được báo tín hiệu trong PPS khi rect_slice_flag là bằng 1.

6.5.1 Các quá trình quét mảnh CTB, quét ô ảnh, và quét ảnh con

Biến số NumTileColumns, mà chỉ ra số lượng của các cột ô ảnh, và danh sách colWidth[i] đối với i nằm trong khoảng từ 0 tới NumTileColumn – 1, bao gồm các giá trị biên, chỉ ra chiều rộng của cột ô ảnh thứ i trong các đơn vị của các CTB, được rút ra như sau:

```

remainingWidthInCtbsY = PicWidthInCtbsY
for(i = 0; i < num_exp_tile_columns_minus1; i++) {
    colWidth[i] = tile_column_width_minus1[i] + 1
    remainingWidthInCtbsY -= colWidth[i]
}
uniformTileColWidth =
tile_column_width_minus1[num_exp_tile_columns_minus1] + 1 (23)
while(remainingWidthInCtbsY >= uniformTileColWidth) {
    colWidth[i++] = uniformTileColWidth
    remainingWidthInCtbsY -= uniformTileColWidth
}

```

```

if(remainingWidthInCtbsY > 0)
    colWidth[i++] = remainingWidthInCtbsY

```

```
NumTileColumns = i
```

Biến số NumTileRows, chỉ ra số lượng của các hàng ô ảnh, và danh sách RowHeight[j] đối với j nằm trong khoảng từ 0 tới NumTileRows – 1, bao gồm các giá trị biên, chỉ ra chiều cao của hàng ô ảnh thứ j trong các đơn vị của các CTB, được rút ra như sau:

```

    remainingHeightInCtbsY = PicHeightInCtbsY
    for(j = 0; j < num_exp_tile_rows_minus1; j++) {
        RowHeight[j] = tile_row_height_minus1[j] + 1
        remainingHeightInCtbsY -= RowHeight[j]
    }
    uniformTileRowHeight =
    tile_row_height_minus1[num_exp_tile_rows_minus1] + 1    (24)
    while(remainingHeightInCtbsY >= uniformTileRowHeight) {
        RowHeight[j++] = uniformTileRowHeight
        remainingHeightInCtbsY -= uniformTileRowHeight
    }
    if(remainingHeightInCtbsY > 0)
        RowHeight[j++] = remainingHeightInCtbsY

```

```
NumTileRows = j
```

Biến số NumTilesInPic được thiết lập bằng

```
NumTileColumns * NumTileRows.
```

Danh sách tileColBd[i] đối với i nằm trong khoảng từ 0 tới NumTileColumns, bao gồm các giá trị biên, chỉ ra vị trí của đường biên cột ô ảnh thứ i trong các đơn vị của các CTB, được rút ra như sau:

```

    for(tileColBd[0] = 0, i = 0; i < NumTileColumns; i++)
        tileColBd[i + 1] = tileColBd[i] + colWidth[i]    (25)

```

LƯU Ý 1 – kích thước của mảng tileColBd[] là lớn hơn số lượng thực tế của các cột ô ảnh trong khi suy ra CtbToTileColBd[].

Danh sách `tileRowBd[j]` đối với `j` nằm trong khoảng từ 0 tới `NumTileRows`, bao gồm các giá trị biên, chỉ ra vị trí của đường biên hàng ô ảnh thứ `j` trong các đơn vị của các CTB, được rút ra như sau:

```
for(tileRowBd[0] = 0, j = 0; j < NumTileRows; j++)
    tileRowBd[j + 1] = tileRowBd[j] + RowHeight[j]    (26)
```

LƯU Ý 2 – kích thước của mảng `tileRowBd[]` trong khi suy ra như trên là lớn hơn số lượng thực tế của các hàng ô ảnh trong khi suy ra `CtbToTileRowBd[]`.

Danh sách `CtbToTileColBd[ctbAddrX]` đối với `ctbAddrX` nằm trong khoảng từ 0 tới `PicWidthInCtbsY`, bao gồm các giá trị biên, chỉ ra sự biến đổi từ địa chỉ CTB ngang thành đường biên cột ô ảnh trái trong các đơn vị của các CTB, được rút ra như sau:

```
tileX = 0
for(ctbAddrX = 0; ctbAddrX <= PicWidthInCtbsY; ctbAddrX++) {
    if(ctbAddrX == tileColBd[tileX + 1])    (27)
        tileX++
    CtbToTileColBd[ctbAddrX] = tileColBd[tileX]
}
```

LƯU Ý 3 – kích thước của mảng `CtbToTileColBd[]` trong khi suy ra như trên là lớn hơn số lượng thực tế của chiều rộng ảnh trong các CTB trong sự rút ra báo tín hiệu `slice_data()`.

Danh sách `CtbToTileRowBd[ctbAddrY]` đối với `ctbAddrY` nằm trong khoảng từ 0 tới `PicHeightInCtbsY`, bao gồm các giá trị biên, chỉ ra sự biến đổi từ địa chỉ CTB dọc sang đường biên cột ô ảnh đỉnh trong các đơn vị của các CTB, được rút ra như sau:

```
tileY = 0
for(ctbAddrY = 0; ctbAddrY <= PicHeightInCtbsY; ctbAddrY++) {
    if(ctbAddrY == tileRowBd[tileY + 1])    (28)
        tileY++
    CtbToTileRowBd[ctbAddrY] = tileRowBd[tileY]
}
```

LƯU Ý 4 – kích thước của mảng CtbToTileRowBd[] trong khi suy ra như trên là lớn hơn số lượng thực tế của chiều cao ảnh trong các CTB trong sự báo tín hiệu slice_data().

Đối với các lát hình chữ nhật, danh sách NumCtusInSlice[i] đối với i nằm trong khoảng từ 0 tới num_slices_in_pic_minus1, bao gồm các giá trị biên, chỉ ra số lượng của CTU trong lát thứ i, danh sách SliceTopLeftTileIdx[i] đối với i nằm trong khoảng từ 0 tới num_slices_in_pic_minus1, bao gồm các giá trị biên, chỉ ra chỉ số của ô ảnh đỉnh-trái của lát, và ma trận CtbAddrInSlice[i][j] đối với i nằm trong khoảng từ 0 tới num_slices_in_pic_minus1, bao gồm các giá trị biên, và j nằm trong khoảng từ 0 tới NumCtusInSlice[i] - 1, bao gồm các giá trị biên, chỉ ra địa chỉ quét mảnh ảnh của CTB thứ j bên trong lát thứ i, được rút ra như sau:

```

if(single_slice_per_subpic_flag) {
    for(i = 0; i <= sps_num_subpics_minus1; i++)
        NumCtusInSlice[i] = 0
    for(i = 0; i < PicSizeInCtbsY; i++) {
        sliceIdx = subpic_info_present_flag ? CtbToSubpicIdx[i] : 0
        CtbAddrInSlice[sliceIdx][NumCtusInSlice[sliceIdx]] = i
        NumCtusInSlice[sliceIdx]++
    } [Ed. (YK): Eqn. 31, mà chỉ ra danh sách ctbToSubpicIdx, phải
được di chuyển trước phương trình này.]
} else {
    tileIdx = 0
    for(i = 0; i <= num_slices_in_pic_minus1; i++)
        NumCtusInSlice[i] = 0
    for(i = 0; i <= num_slices_in_pic_minus1; i++) {
        SliceTopLeftTileIdx[i] = tileIdx
        tileX = tileIdx % NumTileColumns
        tileY = tileIdx / NumTileColumns
        if(i == num_slices_in_pic_minus1) {
            slice_width_in_tiles_minus1[i] =

```

```

NumTileColumns - 1 - tileX
    slice_height_in_tiles_minus1[i] = NumTileRows - 1 - tileY
    NumSlicesInTiles[i] = 1
}
if(slice_width_in_tiles_minus1[i] == 0 &&
slice_height_in_tiles_minus1[i] == 0) {          (29)
    ctbY = tileRowBd[tileY]
    for(j = 0; j < NumSlicesInTiles[i] - 1; j++) {
        AddCtbsToSlice(i, tileColBd[tileX],
tileColBd[tileX + 1],
            ctbY, ctbY + SliceHeightInCtusMinus1[i] + 1)
            ctbY += SliceHeightInCtusMinus1[i] + 1
            i++
        }
        AddCtbsToSlice(i, tileColBd[tileX], tileColBd[tileX + 1],
ctbY, tileRowBd[tileY + 1])
    } else
        for(j = 0; j <= slice_height_in_tiles_minus1[i]; j++)
            for(k = 0; k <= slice_width_in_tiles_minus1[i]; k++)
                AddCtbsToSlice(i, tileColBd[tileX + k],
tileColBd[tileX + k + 1],
                    tileRowBd[tileY + j], tileRowBd[tileY + j + 1])
            if(tile_idx_delta_present_flag)
                tileIdx += tile_idx_delta[i]
            else {
                tileIdx += slice_width_in_tiles_minus1[i] + 1
                if(tileIdx % NumTileColumns == 0)
                    tileIdx +=
slice_height_in_tiles_minus1[i] * NumTileColumns
            }
}

```

```

    }
}

```

Trong đó hàm AddCtbsToSlice(sliceIdx, startX, stopX, startY, stopY) được chỉ ra như sau: [Ed. (YK): Cần nhắc xác định hàm này theo cách thống nhất với các hàm khác được xác định trong đặc tả.]

```

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

```

Sự tương thích dòng bit yêu cầu rằng các giá trị của NumCtusInSlice[i] đối với i nằm trong khoảng từ 0 tới num_slices_in_pic_minus1, bao gồm các giá trị biên, phải lớn hơn 0. Ngoài ra, sự tương thích dòng bit yêu cầu rằng ma trận CtbAddrInSlice[i][j] đối với i nằm trong khoảng từ 0 tới num_slices_in_pic_minus1, bao gồm các giá trị biên, và j nằm trong khoảng từ 0 tới NumCtusInSlice[i] - 1, bao gồm các giá trị biên, phải bao gồm tất cả các địa chỉ CTB trong khoảng 0 tới PicSizeInCtbsY - 1 một lần và chỉ một lần.

Danh sách CtbToSubpicIdx[ctbAddrRs] đối với ctbAddrRs nằm trong khoảng từ 0 tới PicSizeInCtbsY - 1, bao gồm các giá trị biên, chỉ ra sự biến đổi từ địa chỉ CTB trong quét màn hình sang chỉ số ảnh con, được rút ra như sau: [Ed. (YK): Cần nhắc xác định CtbToSubpicIdx[] là biến số nội vùng vì nó chỉ được sử dụng trong tiêu mục này.]

```

for(ctbAddrRs = 0; ctbAddrRs < PicSizeInCtbsY; ctbAddrRs++) {
    posX = ctbAddrRs % PicWidthInCtbsY
    posY = ctbAddrRs / PicWidthInCtbsY
    CtbToSubpicIdx[ctbAddrRs] = -1
    for(i = 0; CtbToSubpicIdx[ctbAddrRs] < 0 && i <=
sps_num_subpics_minus1; i++) {
        if((posX >= subpic_ctu_top_left_x[i]) &&
(posX <

```

```

subpic_ctu_top_left_x[i] + subpic_width_minus1[i] + 1) &&
        (posY >= subpic_ctu_top_left_y[i]) &&
        (posY <
subpic_ctu_top_left_y[i] + subpic_height_minus1[i] + 1))
        CtbToSubpicIdx[ctbAddrRs] = i
    }
}

```

Danh sách NumSlicesInSubpic[i], chỉ ra số lượng các lát hình chữ nhật trong ảnh con thứ i, được rút ra như sau:

```

for(j = 0; j <= sps_num_subpics_minus1; j++)
    NumSlicesInSubpic[j] = 0
for(i = 0; i <= num_slices_in_pic_minus1; i++) {
    posX = CtbAddrInSlice[i][0] % PicWidthInCtbsY
    posY = CtbAddrInSlice[i][0] / PicWidthInCtbsY
    for(j = 0; j <= sps_num_subpics_minus1; j++) {
        if((posX >= subpic_ctu_top_left_x[j]) &&           (32)
            (posX <
subpic_ctu_top_left_x[j] + subpic_width_minus1[j] + 1) &&
            (posY >= subpic_ctu_top_left_y[j]) &&
            (posY <
subpic_ctu_top_left_y[j] + subpic_height_minus1[j] + 1)) {
            NumSlicesInSubpic[j]++
        }
    }
}
}

```

...

7.3.4.3 Các ngữ nghĩa RBSP tập thông số ảnh

...

subpic_id_mapping_in_pps_flag bằng 1 chỉ ra rằng ánh xạ ID ảnh con được báo tín hiệu trong PPS. **subpic_id_mapping_in_pps_flag** bằng 0 chỉ ra rằng ánh xạ ID ảnh con không được báo tín hiệu trong PPS. Nếu

subpic_id_mapping_explicitly_signalled_flag bằng 0 hoặc
subpic_id_mapping_in_sps_flag là bằng 1, giá trị của
subpic_id_mapping_in_pps_flag phải bằng 0. Ngược lại
(subpic_id_mapping_explicitly_signalled_flag là bằng 1 và
subpic_id_mapping_in_sps_flag là bằng 0), giá trị của
subpic_id_mapping_in_pps_flag phải bằng 1.

pps_num_subpics_minus1 phải bằng **sps_num_subpics_minus1**.

pps_subpic_id_len_minus1 phải bằng **sps_subpic_id_len_minus1**.

pps_subpic_id[i] chỉ ra ID ảnh con của ảnh con thứ *i*. Chiều dài của phần
tử cú pháp **pps_subpic_id[i]** bằng **pps_subpic_id_len_minus1 + 1** bit.

Biến số **SubpicIdVal[i]**, đối với mỗi giá trị của *i* nằm trong khoảng từ 0 tới
sps_num_subpics_minus1, bao gồm các giá trị biên, được rút ra như sau:

```
for(i = 0; i <= sps_num_subpics_minus1; i++)
```

```
    if(subpic_id_mapping_explicitly_signalled_flag)
```

```
        SubpicIdVal[i] = subpic_id_mapping_in_pps_flag ?
```

```
pps_subpic_id[i] : sps_subpic_id[i]                (80)
```

```
    else
```

```
        SubpicIdVal[i] = i
```

Sự tương thích dòng bit yêu cầu rằng cả hai điều kiện ràng buộc sau đây
được áp dụng:

Đối với hai giá trị khác nhau bất kỳ của *i* và *j* trong khoảng từ 0 tới
sps_num_subpics_minus1, bao gồm các giá trị biên, **SubpicIdVal[i]** phải không
bằng **SubpicIdVal[j]**.

Khi ảnh hiện tại không phải là ảnh thứ nhất của CLVS, đối với mỗi giá trị
của *i* nằm trong khoảng từ 0 tới **sps_num_subpics_minus1**, bao gồm các giá trị
biên, nếu giá trị của **SubpicIdVal[i]** không bằng giá trị của **SubpicIdVal[i]** của
ảnh trước theo thứ tự giải mã trong cùng lớp, **nal_unit_type** đối với tất cả các
đơn vị NAL lát được tạo mã của ảnh con trong ảnh hiện tại với chỉ số ảnh con *i*
phải bằng giá trị cụ thể trong khoảng từ **IDR_W_RADL** tới **CRA_NUT**, bao gồm
các giá trị biên.

no_pic_partition_flag bằng 1 chỉ ra rằng không có phân vùng ảnh được áp dụng cho mỗi ảnh mà tham chiếu tới PPS. **no_pic_partition_flag** bằng 0 chỉ ra mỗi ảnh tham chiếu tới PPS có thể được phân vùng thành nhiều hơn một ô ảnh hoặc lát.

Sự tương thích dòng bit yêu cầu rằng giá trị của **no_pic_partition_flag** phải là giống nhau đối với tất cả các PPS mà được tham chiếu tới bởi các ảnh được tạo mã bên trong CLVS.

Sự tương thích dòng bit yêu cầu rằng giá trị của **no_pic_partition_flag** phải không bằng 1 khi giá trị của **sps_num_subpics_minus1 + 1** là lớn hơn 1.

pps_log2_ctu_size_minus5 plus 5 chỉ ra kích thước khối cây tạo mã độ sáng của mỗi CTU. **pps_log2_ctu_size_minus5** phải bằng **sps_log2_ctu_size_minus5**.

num_exp_tile_columns_minus1 plus 1 chỉ ra số lượng các chiều rộng cột ô ảnh được tạo ra một cách rõ ràng. Giá trị của **num_exp_tile_columns_minus1** phải nằm trong khoảng từ 0 tới $\text{PicWidthInCtbsY} - 1$, bao gồm các giá trị biên. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_exp_tile_columns_minus1** được suy ra là bằng 0.

num_exp_tile_rows_minus1 plus 1 chỉ ra số lượng các chiều cao hàng ô ảnh được tạo ra một cách rõ ràng. Giá trị của **num_exp_tile_rows_minus1** phải nằm trong khoảng từ 0 tới $\text{PicHeightInCtbsY} - 1$, bao gồm các giá trị biên. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_exp_tile_rows_minus1** được suy ra là bằng 0.

tile_column_width_minus1[i] plus 1 chỉ ra chiều rộng của cột ô ảnh thứ *i* trong các đơn vị của các CTB đối với *i* nằm trong khoảng từ 0 tới **num_exp_tile_columns_minus1 - 1**, bao gồm các giá trị biên. **tile_column_width_minus1[num_exp_tile_columns_minus1]** được sử dụng để rút ra chiều rộng của các cột ô ảnh với chỉ số lớn hơn hoặc bằng **num_exp_tile_columns_minus1** như được chỉ ra trong mục 6.5.1. Giá trị của **tile_column_width_minus1[i]** phải nằm trong khoảng từ 0 tới $\text{PicWidthInCtbsY} - 1$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_column_width_minus1[0]** được suy ra là bằng $\text{PicWidthInCtbsY} - 1$.

tile_row_height_minus1[i] plus 1 chỉ ra chiều cao của hàng ô ảnh thứ *i* trong các đơn vị của các CTB đối với *i* nằm trong khoảng từ 0 tới **num_exp_tile_rows_minus1** - 1, bao gồm các giá trị biên. **tile_row_height_minus1[num_exp_tile_rows_minus1]** được sử dụng để rút ra chiều cao của các hàng ô ảnh với chỉ số lớn hơn hoặc bằng **num_exp_tile_rows_minus1** như được chỉ ra trong mục 6.5.1. Giá trị của **tile_row_height_minus1[i]** phải nằm trong khoảng từ 0 tới **PicHeightInCtbsY** - 1, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_row_height_minus1[0]** được suy ra là bằng **PicHeightInCtbsY** - 1.

rect_slice_flag bằng 0 chỉ ra rằng các ô ảnh bên trong mỗi lát là theo thứ tự quét mảnh và thông tin lát không được báo tín hiệu trong PPS. **rect_slice_flag** bằng 1 chỉ ra rằng các ô ảnh bên trong mỗi lát bao phủ vùng hình chữ nhật của ảnh và thông tin lát được báo tín hiệu trong PPS. Khi không hiện diện, **rect_slice_flag** được suy ra là bằng 1. Khi **subpic_info_present_flag** là bằng 1, giá trị của **rect_slice_flag** phải bằng 1.

single_slice_per_subpic_flag bằng 1 chỉ ra rằng mỗi ảnh con chứa một và chỉ một lát hình chữ nhật. **single_slice_per_subpic_flag** bằng 0 chỉ ra rằng mỗi ảnh con có thể chứa một hoặc nhiều lát hình chữ nhật. Khi **single_slice_per_subpic_flag** là bằng 1, **num_slices_in_pic_minus1** được suy ra bằng **sps_num_subpics_minus1**. Khi không hiện diện, giá trị của **single_slice_per_subpic_flag** được suy ra là bằng 0. [Ed. (GJS): Cân nhắc đổi tên cờ này hoặc làm rõ theo cách khác để tránh việc diễn giải rằng cờ này chỉ thích hợp khi có nhiều hơn một ảnh con trong mỗi ảnh.]

num_slices_in_pic_minus1 plus 1 chỉ ra số lượng các lát hình chữ nhật trong mỗi ảnh tham chiếu tới PPS. Giá trị của **num_slices_in_pic_minus1** phải nằm trong khoảng từ 0 tới **MaxSlicesPerPicture** - 1, bao gồm các giá trị biên, trong đó **MaxSlicesPerPicture** được chỉ ra trong Phụ lục A. Khi **no_pic_partition_flag** là bằng 1, giá trị của **num_slices_in_pic_minus1** được suy ra là bằng 0.

tile_idx_delta_present_flag bằng 0 chỉ ra rằng các giá trị **tile_idx_delta** là không hiện diện trong PPS và tất cả các lát hình chữ nhật trong các ảnh tham

chiều tới PPS được chỉ ra theo thứ tự mảnh theo quá trình được xác định trong mục 6.5.1. [Ed. (RS): Bổ sung tham chiếu vào phương trình.] `tile_idx_delta_present_flag` bằng 1 chỉ ra rằng các giá trị `tile_idx_delta` có thể hiện diện trong PPS và tất cả các lát hình chữ nhật trong các ảnh tham chiếu tới PPS được chỉ ra theo thứ tự được chỉ báo bởi các giá trị của `tile_idx_delta`. Khi không hiện diện, giá trị của `tile_idx_delta_present_flag` được suy ra là bằng 0.

`slice_width_in_tiles_minus1[i]` plus 1 chỉ ra chiều rộng của lát hình chữ nhật thứ *i* trong các đơn vị của các cột ô ảnh. Giá trị của `slice_width_in_tiles_minus1[i]` phải nằm trong khoảng từ 0 tới `NumTileColumns - 1`, bao gồm các giá trị biên.

Khi `slice_width_in_tiles_minus1[i]` không hiện diện, điều sau đây được áp dụng:

Nếu `NumTileColumns` là bằng 1, giá trị của `slice_width_in_tiles_minus1[i]` được suy ra là bằng 0.

Ngược lại, giá trị của `slice_width_in_tiles_minus1[i]` được suy ra như được chỉ ra trong mục 6.5.1. [Ed. (YK): Bổ sung tham chiếu vào phương trình.]

`slice_height_in_tiles_minus1[i]` plus 1 chỉ ra chiều cao của lát hình chữ nhật thứ *i* trong các đơn vị của các hàng ô ảnh. Giá trị của `slice_height_in_tiles_minus1[i]` phải nằm trong khoảng từ 0 tới `NumTileRows - 1`, bao gồm các giá trị biên.

Khi `slice_height_in_tiles_minus1[i]` không hiện diện, điều sau đây được áp dụng:

Nếu `NumTileRows` là bằng 1, hoặc `tile_idx_delta_present_flag` là bằng 0 và `tileIdx % NumTileColumns` là lớn hơn 0), giá trị của `slice_height_in_tiles_minus1[i]` được suy ra là bằng 0.

Ngược lại (`NumTileRows` không bằng 1, và `tile_idx_delta_present_flag` là bằng 1 hoặc `tileIdx % NumTileColumns` là bằng 0), khi `tile_idx_delta_present_flag` là bằng 1 hoặc `tileIdx % NumTileColumns` là bằng 0, giá trị của `slice_height_in_tiles_minus1[i]` được suy ra là bằng `slice_height_in_tiles_minus1[i - 1]`.

num_exp_slices_in_tile[i] chỉ ra số lượng các chiều cao lát được tạo ra một cách rõ ràng trong ô ảnh hiện tại mà chứa nhiều hơn một lát hình chữ nhật. Giá trị của **num_exp_slices_in_tile[i]** phải nằm trong khoảng từ 0 tới **RowHeight[tileY] - 1**, bao gồm các giá trị biên, trong đó **tileY** là chỉ số hàng ô ảnh chứa lát thứ **i**. Khi không hiện diện, giá trị của **num_exp_slices_in_tile[i]** được suy ra là bằng 0. Khi **num_exp_slices_in_tile[i]** là bằng 0, giá trị của biến số **NumSlicesInTiles[i]** được rút ra là bằng 1.

exp_slice_height_in_ctus_minus1[j] plus 1 chỉ ra chiều cao của lát hình chữ nhật thứ **j** trong ô ảnh hiện tại trong các đơn vị của các hàng CTU. Giá trị của **exp_slice_height_in_ctus_minus1[j]** phải nằm trong khoảng từ 0 tới **RowHeight[tileY] - 1**, bao gồm các giá trị biên, trong đó **tileY** là chỉ số hàng ô ảnh của ô ảnh hiện tại.

Khi **num_exp_slices_in_tile[i]** là lớn hơn 0, biến số **NumSlicesInTiles[i]** và **SliceHeightInCtusMinus1[i + k]** đối với **k** nằm trong khoảng từ 0 tới **NumSlicesInTiles[i] - 1** được rút ra như sau:

```

remainingHeightInCtbsY =
RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns]
numExpSliceInTile = num_exp_slices_in_tile[i]
for(j = 0; j < numExpSliceInTile - 1; j++) {
    SliceHeightInCtusMinus1[i++] = exp_slice_height_in_ctu_minus1[j]
    remainingHeightInCtbsY -= SliceHeightInCtusMinus1[j]
}
uniformSliceHeightMinus1 = SliceHeightInCtusMinus1[i - 1] (81)
while(remainingHeightInCtbsY >= (uniformSliceHeightMinus1 + 1)) {
    SliceHeightInCtusMinus1[i++] = uniformSliceHeightMinus1
    remainingHeightInCtbsY -= (uniformSliceHeightMinus1 + 1)
    j++
}
if(remainingHeightInCtbsY > 0) {
    SliceHeightInCtusMinus1[i++] = remainingHeightInCtbsY
    j++
}

```

}

NumSlicesInTiles[i] = j

tile_idx_delta[i] chỉ ra hiệu số giữa chỉ số ô ảnh của ô ảnh thứ nhất trong lát hình chữ nhật thứ i và chỉ số ô ảnh của ô ảnh thứ nhất trong lát hình chữ nhật thứ (i + 1). Giá trị của **tile_idx_delta[i]** phải nằm trong khoảng từ $-\text{NumTilesInPic} + 1$ tới $\text{NumTilesInPic} - 1$, bao gồm các giá trị biên. Khi không hiện diện, giá trị của **tile_idx_delta[i]** được suy ra là bằng 0. Khi hiện diện, giá trị của **tile_idx_delta[i]** phải không bằng 0.

...

7.4.2.4.5 Thứ tự của các đơn vị NAL VCL và sự liên kết của chúng với các ảnh được tạo mã

Thứ tự của các đơn vị NAL VCL bên trong ảnh được tạo mã được ràng buộc như sau:

Đối với hai đơn vị NAL lát được tạo mã bất kỳ A và B của ảnh được tạo mã, giả sử **subpicIdxA** và **subpicIdxB** là các giá trị chỉ số bậc ảnh con của chúng, và **sliceAddrA** và **sliceddrB** là các giá trị **slice_address** của chúng.

Khi một trong hai điều kiện sau đây là đúng, đơn vị NAL lát được tạo mã A phải đứng trước đơn vị NAL lát được tạo mã B:

subpicIdxA là nhỏ hơn **subpicIdxB**.

subpicIdxA là bằng **subpicIdxB** và **sliceAddrA** là nhỏ hơn **sliceAddrB**.

7.4.8.1 Các ngữ nghĩa tiêu đề lát chung

Biến số **CuQpDeltaVal**, chỉ ra hiệu số giữa thông số lượng tử hóa độ sáng đối với đơn vị tạo mã chứa **cu_qp_delta_abs** và dự báo của nó, được thiết lập bằng 0. Tất cả các biến số **CuQpOffsetCb**, **CuQpOffsetCr**, và **CuQpOffsetCbCr**, mà chỉ ra các giá trị cần được sử dụng khi xác định các giá trị tương ứng của các thông số lượng tử hóa **Qp'Cb**, **Qp'Cr**, và **Qp'CbCr** đối với đơn vị tạo mã chứa **cu_chroma_qp_offset_flag**, đều được thiết lập bằng 0.

picture_header_in_slice_header_flag bằng 1 chỉ ra rằng cấu trúc cú pháp PH hiện diện trong tiêu đề lát. **picture_header_in_slice_header_flag** bằng 0 chỉ ra rằng cấu trúc cú pháp PH không hiện diện trong tiêu đề lát.

Sự tương thích dòng bit yêu cầu rằng giá trị của `picture_header_in_slice_header_flag` phải là giống nhau trong tất cả các lát được tạo mã trong CLVS.

Khi `picture_header_in_slice_header_flag` là bằng 1 đối với lát được tạo mã, sự tương thích dòng bit yêu cầu rằng không đơn vị NAL VCL nào có `nal_unit_type` bằng PH_NUT là hiện diện trong CLVS.

Khi `picture_header_in_slice_header_flag` là bằng 0, tất cả các lát được tạo mã trong ảnh hiện tại phải có `picture_header_in_slice_header_flag` là bằng 0, và PU hiện tại phải có đơn vị NAL PH.

`slice_subpic_id` chỉ ra ID ảnh con của ảnh con mà chứa lát. Nếu `slice_subpic_id` hiện diện, giá trị của biến số `CurrSubpicIdx` được rút ra phải sao cho `SubpicIdVal[CurrSubpicIdx]` là bằng `slice_subpic_id`. Ngược lại (`slice_subpic_id` không hiện diện), `CurrSubpicIdx` được rút ra là bằng 0. Chiều dài của `slice_subpic_id` bằng `sps_subpic_id_len_minus1 + 1` bit.

`slice_address` chỉ ra địa chỉ lát của lát. Khi không hiện diện, giá trị của `slice_address` được suy ra là bằng 0. Khi `rect_slice_flag` là bằng 1 và `NumSlicesInSubpic[CurrSubpicIdx]` là bằng 1, giá trị của `slice_address` được suy ra là bằng 0.

Nếu `rect_slice_flag` là bằng 0, điều sau đây được áp dụng:

Địa chỉ lát là chỉ số ô ảnh quét màn hình.

Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log}_2(\text{NumTilesInPic}))$ bit.

Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới $\text{NumTilesInPic} - 1$, bao gồm các giá trị biên.

Ngược lại (`rect_slice_flag` là bằng 1), điều sau đây được áp dụng:

Địa chỉ lát là chỉ số lát bậc ảnh con của lát.

Chiều dài của `slice_address` bằng $\text{Ceil}(\text{Log}_2(\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}]))$ bit.

Giá trị của `slice_address` phải nằm trong khoảng từ 0 tới $\text{NumSlicesInSubpic}[\text{CurrSubpicIdx}] - 1$, bao gồm các giá trị biên.

Sự tương thích dòng bit yêu cầu rằng các điều kiện ràng buộc sau đây được áp dụng:

Nếu `rect_slice_flag` là bằng 0 hoặc `subpic_info_present_flag` là bằng 0, giá trị của `slice_address` phải không bằng giá trị của `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.

Ngược lại, cặp giá trị `slice_subpic_id` và `slice_address` phải không bằng cặp giá trị `slice_subpic_id` và `slice_address` của đơn vị NAL lát được tạo mã bất kỳ khác của cùng ảnh được tạo mã.

Các hình dạng của các lát của ảnh phải sao cho mỗi CTU, khi được giải mã, phải có toàn bộ đường biên trái và toàn bộ đường biên đỉnh của nó chứa đường biên ảnh hoặc chứa các biên của (các) CTU được giải mã trước đó.

`sh_extra_bit[i]` có thể bằng 1 hoặc 0. Các bộ giải mã tương thích với phiên bản này của Đặc tả này phải bỏ qua giá trị của `sh_extra_bit[i]`. Giá trị của nó không ảnh hưởng tới sự tương thích của bộ giải mã với các tiêu sử được chỉ ra trong phiên bản này của đặc tả.

`num_tiles_in_slice_minus1` plus 1, khi hiện diện, chỉ ra số lượng của các ô ảnh trong lát. Giá trị của `num_tiles_in_slice_minus1` phải nằm trong khoảng từ 0 tới `NumTilesInPic - 1`, bao gồm các giá trị biên.

Biến số `NumCtusInCurrSlice`, mà chỉ ra số lượng của các CTU trong lát hiện tại, và danh sách `CtbAddrInCurrSlice[i]`, đối với `i` nằm trong khoảng từ 0 tới `NumCtusInCurrSlice - 1`, bao gồm các giá trị biên, chỉ ra địa chỉ quét mảnh ảnh của CTB thứ `i` bên trong lát, được rút ra như sau:

```

if(rect_slice_flag) {
    picLevelSliceIdx = slice_address
    for(j = 0; j < CurrSubpicIdx; j++)
        picLevelSliceIdx += NumSlicesInSubpic[j]
    NumCtusInCurrSlice = NumCtusInSlice[picLevelSliceIdx]
    for(i = 0; i < NumCtusInCurrSlice; i++)
        CtbAddrInCurrSlice[i] = CtbAddrInSlice[picLevelSliceIdx][i]
(117)
} else {
    NumCtusInCurrSlice = 0
    for(tileIdx = slice_address; tileIdx <=

```

```

slice_address + num_tiles_in_slice_minus1; tileIdx++) {
    tileX = tileIdx % NumTileColumns
    tileY = tileIdx / NumTileColumns
    for(ctbY = tileRowBd[tileY]; ctbY < tileRowBd[tileY + 1];
ctbY++) {
        for(ctbX = tileColBd[tileX]; ctbX < tileColBd[tileX + 1];
ctbX++) {
            CtbAddrInCurrSlice[NumCtusInCurrSlice] =
ctbY * PicWidthInCtb + ctbX
            NumCtusInCurrSlice++
        }
    }
}

```

Các biến số SubpicLeftBoundaryPos, SubpicTopBoundaryPos, SubpicRightBoundaryPos, và SubpicBotBoundaryPos được rút ra như sau:

```

if(subpic_treated_as_pic_flag[CurrSubpicIdx]) {
    SubpicLeftBoundaryPos =
subpic_ctu_top_left_x[CurrSubpicIdx] * CtbSizeY
    SubpicRightBoundaryPos =
Min(pic_width_max_in_luma_samples - 1,
    (subpic_ctu_top_left_x[CurrSubpicIdx] +
    subpic_width_minus1[CurrSubpicIdx] + 1) * CtbSizeY - 1)
    SubpicTopBoundaryPos =
subpic_ctu_top_left_y[CurrSubpicIdx] * CtbSizeY
    SubpicBotBoundaryPos =
Min(pic_height_max_in_luma_samples - 1,
    (subpic_ctu_top_left_y[CurrSubpicIdx] +
    subpic_height_minus1[CurrSubpicIdx] + 1) * CtbSizeY - 1)
}

```

...

Ánh xạ độ sáng với chọn thang sắc độ (LMCS)

LMCS bao gồm hai khía cạnh: ánh xạ độ sáng (quá trình tái tạo dạng, được ký hiệu là RP) và chọn thang dư sắc độ phụ thuộc độ sáng (CRS). Đối với tín hiệu độ sáng, chế độ LMCS hoạt động dựa trên hai miền có liên quan trong đó bao gồm miền thứ nhất là miền gốc và miền thứ hai là miền được tái tạo dạng mà ánh xạ các mẫu độ sáng tới các giá trị cụ thể theo các mô hình tái tạo dạng. Ngoài ra, đối với tín hiệu sắc độ, việc chọn thang dư có thể được áp dụng trong đó các hệ số gộp được rút ra từ các mẫu độ sáng.

Các phần tử cú pháp và các ngữ nghĩa có liên quan trong SPS, tiêu đề ảnh (PH) và tiêu đề lát (SH) được mô tả như sau:

Các bảng cú pháp

7.3.2.3 Cú pháp RBSP tập thông số dãy

	Mô tả
<code>seq_parameter_set_rbsp() {</code>	
<code>sps_seq_parameter_set_id</code>	u(4)
<code>sps_video_parameter_set_id</code>	u(4)
<code>sps_max_sublayers_minus1</code>	u(3)
<code>sps_reserved_zero_4bits</code>	u(4)
<code>...</code>	
<code>sps_affine_enabled_flag</code>	u(1)
<code>if(sps_affine_enabled_flag) {</code>	
<code>five_minus_max_num_subblock_merge_cand</code>	ue(v)
<code>sps_affine_type_flag</code>	u(1)
<code>if(sps_amvr_enabled_flag)</code>	
<code>sps_affine_amvr_enabled_flag</code>	u(1)
<code>sps_affine_prof_enabled_flag</code>	u(1)
<code>if(sps_affine_prof_enabled_flag)</code>	
<code>sps_prof_pic_present_flag</code>	u(1)
<code>}</code>	
<code>...</code>	
<code>sps_lmcs_enabled_flag</code>	u(1)
<code>sps_lfst_enabled_flag</code>	u(1)
<code>...</code>	
<code>}</code>	

7.3.2.7 Cú pháp cấu trúc tiêu đề ảnh

	Mô tả
<code>picture_header_structure() {</code>	
<code>gdr_or_irap_pic_flag</code>	u(1)
<code>if(gdr_or_irap_pic_flag)</code>	
<code>gdr_pic_flag</code>	u(1)
<code>...</code>	
<code>if(sps_lmcs_enabled_flag) {</code>	
<code>ph_lmcs_enabled_flag</code>	u(1)
<code>if(ph_lmcs_enabled_flag) {</code>	

ph_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
ph_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_scaling_list_enabled_flag) {	
ph_scaling_list_present_flag	u(1)
if(ph_scaling_list_present_flag)	
ph_scaling_list_aps_id	u(3)
}	
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {	
ph_virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
}	

7.3.7 Cú pháp tiêu đề lát

7.3.7.1 Cú pháp tiêu đề lát chung

slice_header() {	Mô tả
picture_header_in_slice_header_flag	u(1)
slice_ts_residual_coding_disabled_flag	u(1)
if(ph_lmcs_enabled_flag)	
slice_lmcs_enabled_flag	u(1)
if(pic_scaling_list_enabled_flag)	
slice_scaling_list_present_flag	u(1)
if(NumEntryPoints > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < NumEntryPoints; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

Các ngữ nghĩa

sps_lmcs_enabled_flag bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ được sử dụng trong CLVS. **sps_lmcs_enabled_flag** bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ không được sử dụng trong CLVS.

ph_lmcs_enabled_flag bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ được phép đối với tất cả các lát liên kết với PH. **ph_lmcs_enabled_flag** bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ có thể không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi không hiện diện, giá trị của **ph_lmcs_enabled_flag** được suy ra là bằng 0.

ph_lmcs_aps_id chỉ ra **adaptation_parameter_set_id** của LMCS APS mà các lát liên kết với PH tham chiếu tới. **TemporalId** của đơn vị NAL APS có **aps_params_type** bằng LMCS_APS và **adaptation_parameter_set_id** bằng **ph_lmcs_aps_id** phải nhỏ hơn hoặc bằng **TemporalId** của ảnh liên kết với PH.

ph_chroma_residual_scale_flag bằng 1 chỉ ra rằng việc chọn thang dư sắc độ là được phép đối với tất cả các lát liên kết với PH. **ph_chroma_residual_scale_flag** bằng 0 chỉ ra rằng việc chọn thang dư sắc độ có thể không được phép đối với một, hoặc nhiều, hoặc tất cả các lát liên kết với PH. Khi **ph_chroma_residual_scale_flag** không hiện diện, nó được suy ra là bằng 0.

slice_lmcs_enabled_flag bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là được phép đối với lát hiện tại. **slice_lmcs_enabled_flag** bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ không được phép đối với lát hiện tại. Khi **slice_lmcs_enabled_flag** không hiện diện, nó được suy ra là bằng 0.

Độ phân giải hiệu vector chuyển động thích ứng (AMVR) đối với các khối được tạo mã afin

AMVR afin là công cụ tạo mã mà cho phép khối được tạo mã ngoài afin truyền các hiệu số MV trong các độ phân giải khác nhau, như với độ chính xác tới 1/4 mẫu độ sáng (mặc định, với **amvr_flag** được thiết lập bằng 0), 1/16 mẫu độ sáng, 1 mẫu độ sáng.

Các phần tử cú pháp và các ngữ nghĩa có liên quan trong SPS được mô tả như sau:

Các bảng cú pháp

7.3.2.3 Cú pháp RBSP tập thông số dãy

	Mô tả
<code>seq_parameter_set_rbsp() {</code>	
<code>sps_seq_parameter_set_id</code>	u(4)
<code>sps_video_parameter_set_id</code>	u(4)
<code>sps_max_sublayers_minus1</code>	u(3)
<code>sps_reserved_zero_4bits</code>	u(4)
<code>...</code>	
<code>sps_affine_enabled_flag</code>	u(1)
<code>if(sps_affine_enabled_flag) {</code>	
<code>five_minus_max_num_subblock_merge_cand</code>	ue(v)
<code>sps_affine_type_flag</code>	u(1)
<code>if(sps_amvr_enabled_flag)</code>	
<code>sps_affine_amvr_enabled_flag</code>	u(1)
<code>sps_affine_prof_enabled_flag</code>	u(1)
<code>if(sps_affine_prof_enabled_flag)</code>	
<code>sps_prof_pic_present_flag</code>	u(1)
<code>}</code>	
<code>...</code>	
<code>sps_lmcs_enabled_flag</code>	u(1)
<code>sps_lfnst_enabled_flag</code>	u(1)
<code>...</code>	
<code>}</code>	

Các ngữ nghĩa

sps_affine_amvr_enabled_flag bằng 1 chỉ ra rằng độ phân giải hiệu vector chuyển động thích ứng được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin. **sps_affine_amvr_enabled_flag** bằng 0 chỉ ra rằng độ phân giải hiệu vector chuyển động thích ứng không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin. Khi không hiện diện, giá trị của **sps_affine_amvr_enabled_flag** được suy ra là bằng 0.

D.7 Thông điệp SEI thông tin bậc ảnh con

D.7.1 Cú pháp thông điệp SEI thông tin bậc ảnh con

subpic_level_info(payloadSize) {	Mô tả
num_ref_levels_minus1	u(3)
sli_cbr_constraint_flag	u(1)
explicit_fraction_present_flag	u(1)
if(explicit_fraction_present_flag)	
sli_num_subpics_minus1	ue(v)
while(!byte_aligned())	
sli_alignment_zero_bit	f(1)
for(i = 0; i <= num_ref_levels_minus1; i++) {	
ref_level_idc[i]	u(8)
if(explicit_fraction_present_flag)	
for(j = 0; j <= sli_num_subpics_minus1; j++)	
ref_level_fraction_minus1[i][j]	u(8)
}	
}	

D.7.2 Các ngữ nghĩa thông điệp SEI thông tin bậc ảnh con

Thông điệp SEI thông tin bậc ảnh con chứa thông tin về bậc mà các dãy ảnh con trong dòng bit tương thích với khi kiểm tra sự tương thích của các dòng bit được trích ra mà chứa các dãy ảnh con theo Phụ lục A.

Khi thông điệp SEI thông tin bậc ảnh con hiện diện đối với ảnh bất kỳ của CLVS, thông điệp SEI thông tin bậc ảnh con phải hiện diện đối với ảnh thứ nhất của CLVS. Thông điệp SEI thông tin bậc ảnh con tiếp tục đối với lớp hiện tại theo thứ tự giải mã từ ảnh hiện tại cho đến cuối của CLVS. Tất cả các thông điệp SEI thông tin bậc ảnh con mà áp dụng cho cùng CLVS phải có cùng nội dung. Dãy ảnh con chứa tất cả các ảnh con bên trong CLVS mà có cùng giá trị của chỉ số ảnh con.

Sự tương thích dòng bit yêu cầu rằng, khi thông điệp SEI thông tin bậc ảnh con hiện diện đối với CLVS, giá trị của `subpic_treated_as_pic_flag[i]` phải bằng 1 đối với mỗi giá trị của `i` nằm trong khoảng từ 0 tới `sps_num_subpics_minus1`, bao gồm các giá trị biên.

`num_ref_levels_minus1` plus 1 chỉ ra số lượng các bậc tham chiếu được báo tín hiệu đối với mỗi ảnh con trong số các ảnh con `sps_num_subpics_minus1 + 1`.

`sli_cbr_constraint_flag` bằng 0 chỉ ra rằng để giải mã các dòng bit con thu được từ sự trích ra ảnh con bất kỳ của dòng bit theo mục C.7 bằng cách sử dụng

HRD mà sử dụng đặc tả CPB bất kỳ trong dòng bit con được trích ra, thì bộ lập lịch biểu dòng giả định (HSS- hypothetical stream scheduler) hoạt động trong chế độ tốc độ bit gián đoạn. `sli_cbr_constraint_flag` bằng 1 chỉ ra rằng HSS hoạt động trong chế độ tốc độ bit cố định (CBR – constant bit rate).

`explicit_fraction_present_flag` bằng 1 chỉ ra rằng các phần tử cú pháp `ref_level_fraction_minus1[i]` là hiện diện. `explicit_fraction_present_flag` bằng 0 chỉ ra rằng các phần tử cú pháp `ref_level_fraction_minus1[i]` là không hiện diện.

`sli_num_subpics_minus1` plus 1 chỉ ra số lượng của các ảnh con trong các ảnh của CLVS. Khi hiện diện, giá trị của `sli_num_subpics_minus1` phải bằng giá trị của `sps_num_subpics_minus1` trong SPS mà được tham chiếu tới bởi các ảnh trong CLVS.

`sli_alignment_zero_bit` phải bằng 0.

`ref_level_idc[i]` chỉ báo bậc mà mỗi ảnh con tương thích với như được chỉ ra trong Phụ lục A. Các dòng bit phải không chứa các giá trị của `ref_level_idc` khác với các giá trị được chỉ ra trong Phụ lục A. Các giá trị khác của `ref_level_idc[i]` được bảo lưu để dùng trong tương lai bởi ITU-T | ISO/IEC. Sự tương thích dòng bit yêu cầu rằng giá trị của `ref_level_idc[i]` phải nhỏ hơn hoặc bằng `ref_level_idc[k]` đối với giá trị bất kỳ của `k` lớn hơn `i`.

`ref_level_fraction_minus1[i][j]` plus 1 chỉ ra phần của các giới hạn bậc liên kết với `ref_level_idc[i]` mà ảnh con thứ `j` tương thích với như được chỉ ra trong mục A.4.1.

Biến số `SubpicSizeY[j]` được thiết lập bằng $(\text{subpic_width_minus1}[j] + 1) * \text{CtbSizeY} * (\text{subpic_height_minus1}[j] + 1) * \text{CtbSizeY}$.

Khi không hiện diện, giá trị của `ref_level_fraction_minus1[i][j]` được suy ra bằng

$\text{Ceil}(256 * \text{SubpicSizeY}[j] \div \text{PicSizeInSamplesY} * \text{MaxLumaPs}(\text{general_level_idc}) \div \text{MaxLumaPs}(\text{ref_level_idc}[i]) - 1)$. [Ed. (HD): ở đây `PicSizeInSamplesY` không là `PicSizeMaxInSamplesY` như được chỉ ra trong mục A.4.2?]

Biến số `RefLevelFraction[i][j]` được thiết lập bằng $\text{ref_level_fraction_minus1}[i][j] + 1$.

Các biến số SubpicNumTileCols[j] và SubpicNumTileRows[j] được rút ra như sau:

```

for(i = 0; i <= sps_num_subpics_minus1; i++) {
    SubpicNumTileCols[i] = 1
    SubpicNumTileRows[i] = 1
    for(ctbAddrRs = subpic_ctu_top_left_x[i] + 1; ctbAddrRs <=
        subpic_ctu_top_left_x[i] + subpic_width_minus1[i];
        ctbAddrRs++)
        if(CtbToTileColBd[ctbAddrRs] !=
            CtbToTileColBd[ctbAddrRs - 1])
            SubpicNumTileCols[i]++
            (D.5)
    for(ctbAddrRs = (subpic_ctu_top_left_y[i] + 1) * PicWidthInCtbsY;
        ctbAddrRs <= (subpic_ctu_top_left_y[i] +
            subpic_height_minus1[i]) * PicWidthInCtbsY;
        ctbAddrRs += PicWidthInCtbsY)
        if(CtbToTileRowBd[ctbAddrRs] !=
            CtbToTileRowBd[ctbAddrRs - PicWidthInCtbsY])
            SubpicNumTileRows[i]++
}

```

Các biến số SubpicCpbSizeVcl[i][j] và SubpicCpbSizeNal[i][j] được rút ra như sau:

$$\text{SubpicCpbSizeVcl}[i][j] = \text{Floor}(\text{CpbVclFactor} * \text{MaxCPB} * \text{RefLevelFraction}[i][j] \div 256) \quad (\text{D.6})$$

$$\text{SubpicCpbSizeNal}[i][j] = \text{Floor}(\text{CpbNalFactor} * \text{MaxCPB} * \text{RefLevelFraction}[i][j] \div 256) \quad (\text{D.7})$$

Với MaxCPB được rút ra từ ref_level_idc[i] như được chỉ ra trong mục A.4.2.

Các biến số SubpicBitRateVcl[i][j] và SubpicBitRateNal[i][j] được rút ra như sau:

$$\text{SubpicBitRateVcl}[i][j] = \text{Floor}(\text{CpbVclFactor} * \text{MaxBR} * \text{RefLevelFraction}[i][j] \div 256) \quad (\text{D.8})$$

$$\text{SubpicBitRateNal}[i][j] = \text{Floor}(\text{CpbNalFactor} * \text{MaxBR} * \text{RefLevelFraction}[i][j] \div 256) \text{ (D.9)}$$

Với MaxBR được rút ra từ ref_level_idc[i] như được chỉ ra trong mục A.4.2.

LƯU Ý 1 – Khi ảnh con được trích ra, dòng bit thu được có CpbSize (hoặc được chỉ báo trong SPS hoặc được suy ra) lớn hơn hoặc bằng SubpicCpbSizeVcl[i][j] và SubpicCpbSizeNal[i][j] và BitRate (hoặc được chỉ báo trong SPS hoặc được suy ra) lớn hơn hoặc bằng SubpicBitRateVcl[i][j] và SubpicBitRateNal[i][j].

Sự tương thích dòng bit yêu cầu rằng các dòng bit thu được từ việc trích ra ảnh con thứ j đối với j nằm trong khoảng từ 0 tới sps_num_subpics_minus1, bao gồm các giá trị biên, và tương thích với tiêu sử với general_tier_flag bằng 0 và bậc bằng ref_level_idc[i] đối với i nằm trong khoảng từ 0 tới num_ref_level_minus1, bao gồm các giá trị biên, phải tuân theo các điều kiện ràng buộc sau đây đối với mỗi phép thử sự tương thích dòng bit như được chỉ ra trong Phụ lục C:

$\text{Ceil}(256 * \text{SubpicSizeY}[j] \div \text{RefLevelFraction}[i][j])$ phải nhỏ hơn hoặc bằng MaxLumaPs, trong đó MaxLumaPs được chỉ ra trong Bảng A.1 đối với bậc ref_level_idc[i].

Giá trị của $\text{Ceil}(256 * (\text{subpic_width_minus1}[j] + 1) * \text{CtbSizeY} \div \text{RefLevelFraction}[i][j])$ phải nhỏ hơn hoặc bằng $\text{Sqrt}(\text{MaxLumaPs} * 8)$.

Giá trị của $\text{Ceil}(256 * (\text{subpic_height_minus1}[j] + 1) * \text{CtbSizeY} \div \text{RefLevelFraction}[i][j])$ phải nhỏ hơn hoặc bằng $\text{Sqrt}(\text{MaxLumaPs} * 8)$.

Giá trị của SubpicNumTileCols[j] phải nhỏ hơn hoặc bằng MaxTileCols và của SubpicNumTileRows[j] phải nhỏ hơn hoặc bằng MaxTileRows, trong đó MaxTileCols và MaxTileRows được chỉ ra trong Bảng A.1 đối với bậc ref_level_idc[i].

Giá trị của SubpicNumTileCols[j] * SubpicNumTileRows[j] phải nhỏ hơn hoặc bằng MaxTileCols * MaxTileRows * RefLevelFraction[i][j], trong đó

MaxTileCols và MaxTileRows được chỉ ra trong Bảng A.1 đối với bậc $\text{ref_level_idc}[i]$.

Tổng của các biến số NumBytesInNalUnit đối với AU 0 tương ứng với ảnh con thứ j phải nhỏ hơn hoặc bằng $\text{FormatCapabilityFactor} * (\text{Max}(\text{SubpicSizeY}[j], fR * \text{MaxLumaSr} * \text{RefLevelFraction}[i][j] \div 256) + \text{MaxLumaSr} * (\text{AuCpbRemovalTime}[0] - \text{AuNominalRemovalTime}[0]) * \text{RefLevelFraction}[i][j]) \div (256 * \text{MinCr})$ đối với giá trị của SubpicSizeInSamplesY bằng AU 0, trong đó MaxLumaSr và FormatCapabilityFactor là các giá trị lần lượt được chỉ ra trong Bảng A.2 và Bảng A.3, mà áp dụng cho AU 0, ở bậc $\text{ref_level_idc}[i]$, và MinCr được rút ra như được chỉ báo trong A.4.2. [Ed. (RS): fR là biến số tiêu sử cụ thể được xác định trong A.4.2.]

Tổng của các biến số NumBytesInNalUnit đối với AU n (với n lớn hơn 0) tương ứng với ảnh con thứ j phải nhỏ hơn hoặc bằng $\text{FormatCapabilityFactor} * \text{MaxLumaSr} * (\text{AuCpbRemovalTime}[n] - \text{AuCpbRemovalTime}[n - 1]) * \text{RefLevelFraction}[i][j] \div (256 * \text{MinCr})$, trong đó MaxLumaSr và FormatCapabilityFactor là các giá trị lần lượt được chỉ ra trong Bảng A.2 và Bảng A.3, mà áp dụng cho AU n , ở bậc $\text{ref_level_idc}[i]$, và MinCr được rút ra như được chỉ báo trong A.4.2.

Đối với tập ảnh con bất kỳ mà chứa một hoặc nhiều ảnh con và bao gồm danh sách của các chỉ số ảnh con SubpicSetIndices và số lượng của các ảnh con trong tập ảnh con NumSubpicsInSet, thông tin bậc của tập ảnh con được rút ra.

Biến số SubpicSetAccLevelFraction[i] đối với phân bậc tổng so với bậc tham chiếu $\text{ref_level_idc}[i]$, và các biến số SubpicSetCpbSizeVcl[i], SubpicSetCpbSizeNal[i], SubpicSetBitRateVcl[i], và SubpicSetBitRateNal[i] của tập ảnh con, được rút ra như sau:

```
for (i = 0; i <= num_ref_level_minus1; i++) {
    SubpicSetAccLevelFraction[i] = 0
    SubpicSetCpbSizeVcl[i] = 0
    SubpicSetCpbSizeNal[i] = 0
```

```

SubpicSetNumTiles[i] = 0
for (j = 0; j < NumSubpicsInSet; j++) {
    CurrSubpicIdx = SubpicSetIndices[j]
    SubpicSetAccLevelFraction[i] +=
RefLevelFraction[i][CurrSubpicIdx] (D.10)
    SubpicSetCpbSizeVcl[i] +=
SubpicSetCpbSizeVcl[i][CurrSubpicIdx]
    SubpicSetCpbSizeNal[i] +=
SubpicSetCpbSizeNal[i][CurrSubpicIdx]
    SubpicSetBitRateVcl[i] +=
SubpicSetBitRateVcl[i][CurrSubpicIdx]
    SubpicSetBitRateNal[i] +=
SubpicSetBitRateNal[i][CurrSubpicIdx]
    SubpicSetNumTiles[i] +=
SubpicNumTileCols[CurrSubpicIdx] *
    SubpicNumTileRow[CurrSubpicIdx]
}
}

```

Giá trị của bộ chỉ báo bậc dãy tập ảnh con, SubpicSetLevelIdc, được rút ra như sau:

```

SubpicSetLevelIdc = general_level_idc
for (i = num_ref_level_minus1; i >= 0; i--)
    if(SubpicSetNumTiles[i] <= (MaxTileCols * MaxTileRows) &&
(D.11)

```

```

SubpicSetAccLevelFraction[i] <= 256)

```

```

SubpicSetLevelIdc = ref_level_idc[i]

```

trong đó MaxTileCols và MaxTileRows được chỉ ra trong Bảng A.1 đối với ref_level_idc[i].

Dòng bit tập ảnh con tương thích với tiêu sử với general_tier_flag bằng 0 và bậc bằng SubpicSetLevelIdc phải tuân theo các điều kiện ràng buộc sau đây đối với mỗi phép thử sự tương thích dòng bit như được chỉ ra trong Phụ lục C:

Đối với các thông số HRD VCL, $\text{SubpicSetCpbSizeVcl}[i]$ phải nhỏ hơn hoặc bằng $\text{CpbVclFactor} * \text{MaxCPB}$, trong đó CpbVclFactor được chỉ ra trong Bảng A.3 và MaxCPB được chỉ ra trong Bảng A.1 trong các đơn vị bằng CpbVclFactor bit.

Đối với các thông số HRD NAL, $\text{SubpicSetCpbSizeNal}[i]$ phải nhỏ hơn hoặc bằng $\text{CpbNalFactor} * \text{MaxCPB}$, trong đó CpbNalFactor được chỉ ra trong Bảng A.3, và MaxCPB được chỉ ra trong Bảng A.1 trong các đơn vị bằng CpbNalFactor bit.

Đối với các thông số HRD VCL, $\text{SubpicSetBitRateVcl}[i]$ phải nhỏ hơn hoặc bằng $\text{CpbVclFactor} * \text{MaxBR}$, trong đó CpbVclFactor được chỉ ra trong Bảng A.3 và MaxBR được chỉ ra trong Bảng A.1 trong các đơn vị bằng CpbVclFactor bit.

Đối với các thông số HRD NAL, $\text{SubpicSetBitRateNal}[i]$ phải nhỏ hơn hoặc bằng $\text{CpbNalFactor} * \text{MaxCR}$, trong đó CpbNalFactor được chỉ ra trong Bảng A.3, và MaxBR được chỉ ra trong Bảng A.1 trong các đơn vị bằng CpbNalFactor bit.

LƯU Ý 2 – Khi tập ảnh con được trích ra, dòng bit thu được có CpbSize (hoặc được chỉ báo trong SPS hoặc được suy ra) là lớn hơn hoặc bằng $\text{SubpicSetCpbSizeVcl}[i][j]$ và $\text{SubpicSetCpbSizeNal}[i][j]$ và BitRate (hoặc được chỉ báo trong SPS hoặc được suy ra) là lớn hơn hoặc bằng $\text{SubpicSetBitRateVcl}[i][j]$ và $\text{SubpicSetBitRateNal}[i][j]$.

Các vấn đề kỹ thuật được giải quyết bởi các giải pháp kỹ thuật được mô tả

Các thiết kế hiện có đối với các ảnh con và LMCS trong VVC có các vấn đề sau đây:

Việc rút ra danh sách $\text{SubpicNumTileRows}[]$ (chỉ ra số lượng của các hàng ô ảnh được bao gồm trong ảnh con) là Phương trình D.5 là không đúng, do giá trị chỉ số idx trong $\text{CtbToTileRowBd}[\text{idx}]$ trong phương trình có thể lớn hơn giá trị cho phép lớn nhất. Hơn nữa, độ sai lệch của cả $\text{SubpicNumTileRows}[]$ lẫn $\text{SubpicNumTileCols}[]$ (chỉ ra số lượng của các cột ô ảnh được bao gồm trong

ảnh con) sử dụng hoạt động dựa trên CTU, mà phức tạp một cách không cần thiết.

Việc rút ra của mảng CtbAddrInSlice trong Phương trình 29 khi `single_slice_per_subpic_flag` là bằng 1 là không đúng, do các giá trị của các địa chỉ CTB quét mảnh trong mảng đối với mỗi lát cần phải theo giải mã thứ tự của các CTU thay vì theo thứ tự quét mảnh của các CTU.

Sự báo tín hiệu LMCS là không hiệu quả. Khi `ph_lmcs_enabled_flag` bằng 1, trong phần lớn các trường hợp LMCS sẽ là được phép đối với tất cả các lát của ảnh. Tuy nhiên, trong thiết kế VVC hiện tại, đối với trường hợp khi LMCS là được phép đối với tất cả các lát của ảnh, thì không chỉ `ph_lmcs_enabled_flag` là bằng 1, `slice_lmcs_enabled_flag` với giá trị 1 cần phải được báo tín hiệu đối với mỗi lát.

Các ngữ nghĩa của `ph_lmcs_enabled_flag` là xung đột với việc thúc đẩy báo tín hiệu cờ LMCS bậc lát khi `ph_lmcs_enabled_flag` là đúng. Trong VVC hiện tại, khi `ph_lmcs_enabled_flag` là đúng, thì tất cả các lát phải cho phép LMCS. Do đó, không cần phải báo tín hiệu thêm các cờ cho phép LMCS trong tiêu đề lát.

Ngoài ra, khi tiêu đề ảnh báo rằng LMCS là được phép, thường là, đối với tất cả các lát, thì tất cả LMCS là được phép. Việc điều khiển LCMS trong tiêu đề lát chủ yếu để xử lý các trường hợp góc. Do đó, nếu cờ LMCS PH là đúng và cờ LMCS SH luôn luôn được báo tín hiệu, điều này có thể gây ra các bit không cần thiết được báo tín hiệu cho các trường hợp người dùng phổ thông.

Các ngữ nghĩa của cờ AMVR afin SPS là không đúng do đối với mỗi CU được tạo mã ngoài afin, AMVR afin có thể là được phép hoặc không được phép.

Các ví dụ về các kỹ thuật và các phương án

Để giải quyết các vấn đề nêu trên, và một số vấn đề khác mà không được nhắc đến, các phương pháp như được tổng kết dưới đây được đề xuất. Các mục này cần được xem như các ví dụ để giải thích các ý tưởng chung và không được diễn giải theo cách hẹp. Hơn nữa, các mục này có thể được áp dụng một cách riêng rẽ hoặc được kết hợp theo cách bất kỳ.

Liên quan tới các ảnh con để giải quyết các vấn đề thứ nhất và thứ hai

Một hoặc nhiều trong số cách tiếp cận sau đây được mô tả:

Chỉ số cột ô ảnh của mỗi cột CTU của ảnh được rút ra.

Việc rút ra số lượng của các cột ô ảnh được bao gồm trong ảnh con là dựa trên các chỉ số cột ô ảnh của các CTU xa nhất bên trái và/hoặc xa nhất bên phải được bao gồm trong ảnh con.

Chỉ số hàng ô ảnh của mỗi hàng CTU của ảnh được rút ra.

Việc rút ra số lượng của các hàng ô ảnh được bao gồm trong ảnh con là dựa trên các chỉ số hàng ô ảnh của các CTU đỉnh và/hoặc đáy được bao gồm trong ảnh con.

Thuật ngữ chỉ số lát bậc ảnh được xác định như sau: chỉ số, được xác định khi `rect_slice_flag` là bằng 1, của lát tới danh sách của các lát trong ảnh theo thứ tự như các lát được báo tín hiệu trong PPS khi `single_slice_per_subpic_flag` là bằng 0, hoặc theo thứ tự tăng các chỉ số ảnh con của ảnh con tương ứng với các lát khi `single_slice_per_subpic_flag` là bằng 1.

Trong một ví dụ, chiều cao của ảnh con không thể được tính theo các ô ảnh khi ảnh con chứa lát mà được phân vùng từ ô ảnh.

Trong một ví dụ, chiều cao của ảnh con có thể được tính theo các CTU thay vì các ô ảnh.

Việc liệu chiều cao của ảnh con có nhỏ hơn một hàng ô ảnh hay không được rút ra.

Trong một ví dụ, việc liệu chiều cao của ảnh con có nhỏ hơn một hàng ô ảnh hay không được rút ra là đúng khi ảnh con chỉ bao gồm các CTU từ một hàng ô ảnh và khi hoặc các CTU đỉnh trong ảnh con không là các CTU đỉnh của hàng ô ảnh hoặc các CTU đáy trong ảnh con không là các CTU đáy của hàng ô ảnh.

Khi được chỉ báo rằng mỗi ảnh con chỉ chứa một lát và chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh, đối với mỗi lát với chỉ số lát bậc ảnh i của ảnh, giá trị của `CtbAddrInSlice[i][j]` đối với j nằm trong khoảng từ 0 tới số lượng của các CTU trong lát trừ 1, bao gồm các giá trị biên, được rút ra bằng địa chỉ CTU quét mảnh ảnh của CTU thứ j trong quét mảnh CTU của ảnh con.

Trong một ví dụ, việc liệu chiều cao của ảnh con có nhỏ hơn một hàng ô ảnh hay không được rút ra là đúng khi khoảng cách giữa các CTU đỉnh trong ảnh con và các CTU đáy trong ảnh con là nhỏ hơn chiều cao của ô ảnh tính theo các CTU.

Khi được chỉ báo rằng mỗi ảnh con chỉ chứa một lát và chiều cao của ảnh con là lớn hơn hoặc bằng một hàng ô ảnh, đối với mỗi lát với chỉ số lát bậc ảnh i của ảnh, giá trị của $CtbAddrInSlice[i][j]$ đối với j nằm trong khoảng từ 0 tới số lượng của các CTU trong lát trừ 1, bao gồm các giá trị biên, được rút ra bằng địa chỉ CTU quét mảnh ảnh của CTU thứ j theo thứ tự sau đây của các CTU:

Các CTU trong các ô ảnh khác nhau trong ảnh con được xếp thứ tự sao cho CTU thứ nhất trong ô ảnh thứ nhất với giá trị của chỉ số ô ảnh nhỏ hơn đi trước CTU thứ hai trong ô ảnh thứ hai với giá trị của chỉ số ô ảnh lớn hơn.

Các CTU bên trong một ô ảnh trong ảnh con được xếp thứ tự trong quét mảnh CTU của ô ảnh.

Liên quan tới LMCS để giải quyết vấn đề thứ ba (bao gồm các vấn đề con)

Sự điều khiển hai bậc của LMCS (mà bao gồm hai khía cạnh: ánh xạ độ sáng (quá trình tái tạo dạng, được ký hiệu là RP) và chọn thang dư sắc độ phụ thuộc độ sáng (CRS)) được đề xuất, trong đó sự điều khiển bậc cao hơn (ví dụ, bậc ảnh) và bậc thấp hơn (ví dụ, bậc lát) được sử dụng và việc liệu thông tin điều khiển bậc thấp hơn có hiện diện hay không phụ thuộc vào thông tin điều khiển bậc cao. Ngoài ra, điều sau đây được áp dụng:

Trong ví dụ thứ nhất, một hoặc nhiều trong số các gạch đầu dòng con dưới đây được áp dụng:

Bộ chỉ báo thứ nhất (ví dụ, **ph_lmcs_enabled_type**) có thể được báo tín hiệu ở bậc cao hơn (ví dụ, trong tiêu đề ảnh (PH)) để chỉ ra cách thức LMCS được phép ở bậc thấp hơn mà là giá trị không nhị phân.

Trong một ví dụ, khi bộ chỉ báo thứ nhất là bằng X (ví dụ, $X = 2$), nó chỉ ra rằng LMCS được phép đối với tất cả các lát liên kết với PH; khi bộ chỉ báo thứ nhất là bằng Y ($Y \neq X$) (ví dụ, $Y = 1$), nó chỉ ra rằng LMCS được phép đối với một, hoặc nhiều, nhưng không phải tất cả các lát liên kết với PH; khi bộ chỉ báo thứ nhất là bằng Z ($Z \neq X$ và $Z \neq Y$) (ví dụ, $Z = 0$), nó chỉ ra rằng LMCS là không được phép đối với tất cả các lát liên kết với PH.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ nhất không hiện diện, giá trị của bộ chỉ báo được suy ra là bằng giá trị mặc định, chẳng hạn Z.

Trong một ví dụ, khi bộ chỉ báo thứ nhất là bằng X (ví dụ, $X = 2$), nó chỉ ra rằng LMCS là không được phép đối với tất cả các lát liên kết với PH; khi bộ chỉ báo thứ nhất là bằng Y ($Y \neq X$) (ví dụ, $Y = 1$), nó chỉ ra rằng LMCS là không được phép đối với một, hoặc nhiều, nhưng không phải tất cả các lát liên kết với PH; khi bộ chỉ báo thứ nhất là bằng Z ($Z \neq X$ và $Z \neq Y$) (ví dụ, $Z = 0$), nó chỉ ra rằng LMCS được phép đối với tất cả các lát liên kết với PH.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ nhất không hiện diện, giá trị của bộ chỉ báo được suy ra là bằng giá trị mặc định, chẳng hạn X.

Theo cách khác, hơn nữa, bộ chỉ báo thứ nhất có thể được báo tín hiệu một cách có điều kiện theo giá trị của cờ cho phép LMCS trong bậc dãy (ví dụ, **sps_lmcs_enabled_flag**).

Theo cách khác, hơn nữa, bộ chỉ báo thứ nhất có thể được tạo mã với $u(v)$, hoặc $u(2)$ hoặc $ue(v)$.

Theo cách khác, hơn nữa, bộ chỉ báo thứ nhất có thể được tạo mã với mã đơn phân bị chặt.

Theo cách khác, hơn nữa, thông tin APS LMCS (ví dụ, **ph_lmcs_aps_id**) được sử dụng bởi các lát và/hoặc cờ cho phép CS (ví dụ, **ph_chroma_residual_scale_flag**) có thể được báo tín hiệu theo kiểm tra điều kiện của các giá trị của bộ chỉ báo thứ nhất.

Bộ chỉ báo thứ hai của việc cho phép/không cho phép LMCS đối với bậc thấp hơn (ví dụ, **slice_lmcs_enabled_flag**) có thể được báo tín hiệu ở bậc thấp hơn (ví dụ, trong tiêu đề lát) và nó có thể được báo tín hiệu một cách có điều kiện bằng cách kiểm tra giá trị của bộ chỉ báo thứ nhất.

Trong một ví dụ, bộ chỉ báo thứ hai có thể được báo tín hiệu theo kiểm tra điều kiện là ‘bộ chỉ báo thứ nhất là bằng Y’.

Theo cách khác, bộ chỉ báo thứ hai có thể được báo tín hiệu theo kiểm tra điều kiện là ‘giá trị của bộ chỉ báo thứ nhất $\gg 1$ ’ hoặc ‘giá trị của bộ chỉ báo thứ nhất/2’ hoặc ‘giá trị của bộ chỉ báo thứ nhất $\& 0x01$ ’.

Theo cách khác, hơn nữa, nó có thể được suy ra là được phép khi bộ chỉ báo thứ nhất là bằng X; hoặc được suy ra là không được phép khi bộ chỉ báo thứ nhất là bằng Z.

Trong ví dụ thứ hai, một hoặc nhiều trong số các gạch đầu dòng con dưới đây được áp dụng:

Nhiều hơn một bộ chỉ báo có thể được báo tín hiệu có thể được báo tín hiệu ở bậc cao hơn (ví dụ, trong tiêu đề ảnh (PH)) để chỉ ra cách thức LMCS được phép ở bậc thấp hơn mà là giá trị không nhị phân.

Trong một ví dụ, hai bộ chỉ báo có thể được báo tín hiệu trong PH.

Trong một ví dụ, bộ chỉ báo thứ nhất chỉ ra liệu có ít nhất một lát liên kết với PH mà cho phép LMCS hay không. Và bộ chỉ báo thứ hai chỉ ra liệu tất cả các lát liên kết với PH có cho phép LMCS hay không.

Theo cách khác, hơn nữa, bộ chỉ báo thứ hai có thể được báo tín hiệu một cách có điều kiện theo giá trị của bộ chỉ báo thứ nhất, ví dụ, khi bộ chỉ báo thứ nhất chỉ ra rằng có ít nhất một lát mà cho phép LMCS.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ hai không hiện diện, thì được suy ra rằng tất cả các lát cho phép LMCS.

Theo cách khác, hơn nữa, bộ chỉ báo thứ ba có thể được báo tín hiệu một cách có điều kiện trong SH theo giá trị của bộ chỉ báo thứ hai, ví dụ, khi bộ chỉ báo thứ hai chỉ ra rằng không phải tất cả các lát cho phép LMCS.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ ba không hiện diện, nó có thể được suy ra theo giá trị của bộ chỉ báo thứ nhất và/hoặc thứ hai (ví dụ, được suy ra là bằng giá trị của bộ chỉ báo thứ nhất).

Theo cách khác, bộ chỉ báo thứ nhất chỉ ra liệu có ít nhất một lát liên kết với PH mà không cho phép LMCS hay không. Và bộ chỉ báo thứ hai chỉ ra liệu tất cả các lát liên kết với PH có không cho phép LMCS hay không.

Theo cách khác, hơn nữa, bộ chỉ báo thứ hai có thể được báo tín hiệu một cách có điều kiện theo giá trị của bộ chỉ báo thứ nhất, ví dụ, khi bộ chỉ báo thứ nhất chỉ ra rằng có ít nhất một lát mà không cho phép LMCS.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ hai không hiện diện, thì được suy ra rằng tất cả các lát liên kết với PH không cho phép LMCS.

Theo cách khác, hơn nữa, bộ chỉ báo thứ ba có thể được báo tín hiệu một cách có điều kiện trong SH theo giá trị của bộ chỉ báo thứ hai, ví dụ, khi bộ chỉ báo thứ hai chỉ ra rằng không phải tất cả các lát không cho phép LMCS.

Theo cách khác, hơn nữa, khi bộ chỉ báo thứ ba không hiện diện, nó có thể được suy ra theo giá trị của bộ chỉ báo thứ nhất và/hoặc thứ hai (ví dụ, được suy ra là bằng giá trị của bộ chỉ báo thứ nhất).

Theo cách khác, hơn nữa, bộ chỉ báo thứ nhất có thể được báo tín hiệu một cách có điều kiện theo giá trị của cờ cho phép LMCS trong bậc dãy (ví dụ, `sps_lmcs_enabled_flag`).

Bộ chỉ báo thứ ba của việc cho phép/không cho phép LMCS đối với bậc thấp hơn (ví dụ, `slice_lmcs_enabled_flag`) có thể được báo tín hiệu ở bậc thấp hơn (ví dụ, trong tiêu đề lát) và nó có thể được báo tín hiệu một cách có điều kiện bằng cách kiểm tra giá trị của bộ chỉ báo thứ nhất và/hoặc bộ chỉ báo thứ hai.

Trong một ví dụ, bộ chỉ báo thứ ba có thể được báo tín hiệu theo kiểm tra điều kiện là ‘không phải tất cả các lát cho phép LMCS’ hoặc ‘không phải tất cả các lát không cho phép LMCS’.

Trong ví dụ khác nữa, bộ chỉ báo thứ nhất và/hoặc bộ chỉ báo thứ hai và/hoặc bộ chỉ báo thứ ba được nêu trong ví dụ thứ nhất/thứ hai có thể được sử dụng để điều khiển việc sử dụng RP hoặc CRS thay vì LMCS.

Các ngữ nghĩa của ba cờ LMCS trong SPS/PH/SH được cập nhật như sau:

`sps_lmcs_enabled_flag` bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ *[[được sử dụng]] có thể được sử dụng* trong CLVS. `sps_lmcs_enabled_flag` bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ không được sử dụng trong CLVS.

`ph_lmcs_enabled_flag` bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ *[[được phép]] có thể được sử dụng* đối với tất cả các lát liên kết với PH. `ph_lmcs_enabled_flag` bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ *[[có thể không được phép đối với một, hoặc nhiều, hoặc]] là không được sử dụng đối với* tất cả các lát liên kết với PH. Khi không hiện diện, giá trị của `ph_lmcs_enabled_flag` được suy ra là bằng 0.

`slice_lmcs_enabled_flag` bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là *được phép* **được sử dụng** đối với lát hiện tại. `slice_lmcs_enabled_flag` bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là không *được phép* **được sử dụng** đối với lát hiện tại. Khi `slice_lmcs_enabled_flag` không hiện diện, nó được suy ra là bằng 0.

Sự báo tín hiệu LMCS PH và/hoặc SH được thay đổi sao cho khi LMCS được sử dụng đối với tất cả các lát của ảnh, thì sự báo tín hiệu LMCS không hiện diện trong SH.

Theo cách khác, hơn nữa, cách thức LMCS được suy ra phụ thuộc vào sự báo tín hiệu LMCS PH.

Trong một ví dụ, nó được suy ra là được phép khi LMCS được sử dụng đối với tất cả các lát của ảnh; và nó được suy ra không được phép khi LMCS không được sử dụng đối với tất cả các lát của ảnh.

Liên quan tới AMVR afin

Các ngữ nghĩa của cờ AMVR afin trong SPS được cập nhật như sau:

`sps_affine_amvr_enabled_flag` bằng 1 chỉ ra rằng độ phân giải hiệu vector chuyển động thích ứng *là* **có thể** được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin. `sps_affine_amvr_enabled_flag` bằng 0 chỉ ra rằng độ phân giải hiệu vector chuyển động thích ứng không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin. Khi không hiện diện, giá trị của `sps_affine_amvr_enabled_flag` được suy ra là bằng 0.

Các phương án

Phương án 1: hỗ trợ các ảnh con

Phương án là dành cho mục 1 và các tiểu mục của nó.

3 Các định nghĩa

chỉ số lát bậc ảnh: chỉ số, được xác định khi `rect_slice_flag` là bằng 1, của lát tới danh sách của các lát trong ảnh theo thứ tự như các lát được báo tín hiệu trong PPS khi `single_slice_per_subpic_flag` là bằng 0, hoặc theo thứ tự

tăng các chỉ số ảnh con của ảnh con tương ứng với các lát khi $single_slice_per_subpic_flag$ là bằng 1.

[[**chỉ số lát bậc ảnh**: chỉ số của lát tới danh sách của các lát trong ảnh theo thứ tự như chúng được báo tín hiệu trong PPS khi $rect_slice_flag$ là bằng 1.]]

6.5.1 Các quá trình quét mảnh CTB, quét ô ảnh, và quét ảnh con

...

Các danh sách $CtbToTileColBd[ctbAddrX]$ và $ctbToTileColIdx[ctbAddrX]$ đối với $ctbAddrX$ nằm trong khoảng từ 0 tới $PicWidthInCtbsY$, bao gồm các giá trị biên, chỉ ra sự biến đổi từ địa chỉ CTB ngang lần lượt sang đường biên cột ô ảnh trái trong các đơn vị của các CTB và **tới chỉ số cột ô ảnh**, được rút ra như sau:

```
tileX = 0
for(ctbAddrX = 0; ctbAddrX <= PicWidthInCtbsY; ctbAddrX++) {
    if(ctbAddrX == tileColBd[tileX + 1])
        (27)
        tileX++
    CtbToTileColBd[ctbAddrX] = tileColBd[tileX]
     $ctbToTileColIdx[ctbAddrX] = tileX$ 
}
```

LƯU Ý 3 – kích thước của **các mảng** $CtbToTileColBd[]$ và $ctbToTileColIdx[]$ trong khi suy ra như trên **là** lớn hơn chiều rộng ảnh thực tế trong các CTB.

Các danh sách $CtbToTileRowBd[ctbAddrY]$ và $ctbToTileRowIdx[ctbAddrY]$ đối với $ctbAddrY$ nằm trong khoảng từ 0 tới $PicHeightInCtbsY$, bao gồm các giá trị biên, chỉ ra sự biến đổi từ địa chỉ CTB dọc lần lượt sang đường biên cột ô ảnh đỉnh trong các đơn vị của các CTB và **tới chỉ số hàng ô ảnh**, được rút ra như sau:

```
tileY = 0
for(ctbAddrY = 0; ctbAddrY <= PicHeightInCtbsY; ctbAddrY++) {
    if(ctbAddrY == tileRowBd[tileY + 1])
        (28)
```

```

tileY++
CtbToTileRowBd[ctbAddrY] = tileRowBd[tileY]
ctbToTileRowIdx[ctbAddrY] = tileY
}

```

LƯU Ý 4 – các kích thước của các mảng CtbToTileRowBd[] và ctbToTileRowIdx[] trong khi suy ra như trên là lớn hơn chiều cao ảnh thực tế trong các CTB.

Các danh sách SubpicWidthInTiles[i] và SubPicHeightInTiles[i], đối với i nằm trong khoảng từ 0 tới sps_num_subpics_minus1, bao gồm các giá trị biên, lần lượt chỉ ra chiều rộng và chiều cao của ảnh con thứ i trong các cột ô ảnh và các hàng ô ảnh, và danh sách subPicHeightLessThanOneTileFlag[i], đối với i nằm trong khoảng từ 0 tới sps_num_subpics_minus1, bao gồm các giá trị biên, chỉ ra việc liệu chiều cao của ảnh con thứ i có nhỏ hơn một hàng ô ảnh hay không, được rút ra như sau:

```

for(i = 0; i <= sps_num_subpics_minus1; i++) {
    leftX = subpic_ctu_top_left_x[i]
    rightX = leftX + subpic_width_minus1[i]
    SubpicWidthInTiles[i] =
ctbToTileColIdx[rightX] + 1 - ctbToTileColIdx[leftX]    (29)
    topY = subpic_ctu_top_left_y[i]
    bottomY = topY + subpic_height_minus1[i]
    SubPicHeightInTiles[i] =
ctbToTileRowIdx[bottomY] + 1 - ctbToTileRowIdx[topY]
    if(SubPicHeightInTiles[i] == 1 && bottomY + 1 - topY <
RowHeight[ctbToTileRowIdx[topY]])
        subPicHeightLessThanOneTileFlag[i] = 1
}

```

Theo cách khác, trong phương trình nêu trên, điều kiện sau đây:

```

if(SubPicHeightInTiles[i] == 1 && (bottomY + 1 - topY <
RowHeight[ctbToTileRowIdx[topY]]))

```

Có thể được thay đổi như sau:

```

        if(SubPicHeightInTiles[i] == 1 && (topY != CtbToTileRowBd[topY])
||

```

```

        bottomY + 1 != CtbToTileRowBd[bottomY + 1])

```

LƯU Ý 5 – Khi ô ảnh được phân vùng thành nhiều lát hình chữ nhật và chỉ tập con của các lát hình chữ nhật của ô ảnh là được bao gồm trong ảnh con thứ *i*, thì ô ảnh được tính là một ô ảnh trong giá trị của *SubPicHeightInTiles[i]*.

Khi *rect_slice_flag* là bằng 1, danh sách *NumCtusInSlice[i]* đối với *i* nằm trong khoảng từ 0 tới *num_slices_in_pic_minus1*, bao gồm các giá trị biên, chỉ ra số lượng của các CTU trong lát thứ *i*, danh sách *SliceTopLeftTileIdx[i]* đối với *i* nằm trong khoảng từ 0 tới *num_slices_in_pic_minus1*, bao gồm các giá trị biên, chỉ ra chỉ số ô ảnh của ô ảnh chứa CTU thứ nhất trong lát, và ma trận *CtbAddrInSlice[i][j]* đối với *i* nằm trong khoảng từ 0 tới *num_slices_in_pic_minus1*, bao gồm các giá trị biên, và *j* nằm trong khoảng từ 0 tới *NumCtusInSlice[i] - 1*, bao gồm các giá trị biên, chỉ ra địa chỉ quét mảnh ảnh của CTB thứ *j* bên trong lát thứ *i*, và biến số *NumSlicesInTiles[i]*, chỉ ra số lượng của các lát trong ô ảnh chứa lát thứ *i*, được rút ra như sau:

```

        if(single_slice_per_subpic_flag) {
            for(i = 0; i <= sps_num_subpics_minus1; i++) {
                NumCtusInSlice[i] = 0
                if(subPicHeightLessThanOneTileFlag[i]) /* Lát chứa số lượng các
hàng CTU trong ô ảnh. */
                    AddCtbsToSlice(i, subpic_ctu_top_left_x[i],
                        subpic_ctu_top_left_x[i] + subpic_width_minus1[i] + 1,
subpic_ctu_top_left_y[i],
                        subpic_ctu_top_left_y[i] + subpic_height_minus1[i] + 1)
                else { /* Lát chứa số lượng các ô ảnh hoàn chỉnh bao phủ vùng
hình chữ nhật. */
                    tileX = CtbToTileColBd[subpic_ctu_top_left_x[i]]
                    tileY = CtbToTileRowBd[subpic_ctu_top_left_y[i]]
                    for(j = 0; j < SubPicHeightInTiles[i]; j++)

```

```

        for(k = 0; k < SubpicWidthInTiles[i]; k++)
            AddCtbsToSlice(i, tileColBd[tileX + k],
tileColBd[tileX + k + 1],
tileRowBd[tileY + j], tileRowBd[tileY + j + 1])
    }
}
[[
    for(i = 0; i <= sps_num_subpics_minus1; i++)
        NumCtusInSlice[i] = 0
    for(i = 0; i < PicSizeInCtbsY; i++) {
        sliceIdx = subpic_info_present_flag ? CtbToSubpicIdx[i] : 0
        CtbAddrInSlice[sliceIdx][NumCtusInSlice[sliceIdx]] = i
        NumCtusInSlice[sliceIdx]++
    }
} else {
    tileIdx = 0
    for(i = 0; i <= num_slices_in_pic_minus1; i++)
        NumCtusInSlice[i] = 0
    for(i = 0; i <= num_slices_in_pic_minus1; i++) {
        SliceTopLeftTileIdx[i] = tileIdx
        tileX = tileIdx % NumTileColumns
        tileY = tileIdx / NumTileColumns
        if(i < num_slices_in_pic_minus1) {
            sliceWidthInTiles[i] = slice_width_in_tiles_minus1[i] + 1
            SliceHeightInTiles[i] = slice_height_in_tiles_minus1[i] + 1
        } else {
            sliceWidthInTiles[i] = NumTileColumns - tileX
            SliceHeightInTiles[i] = NumTileRows - tileY
            NumSlicesInTiles[i] = 1
        }
        if(sliceWidthInTiles[i] == 1 && SliceHeightInTiles[i] == 1) {
(30)

```

```

        if(num_exp_slices_in_tile[i] == 0) {
            NumSlicesInTiles[i] = 1
            SliceHeightInCtus[i] =
RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns]
        } else {
            remainingHeightInCtbsY =
RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns]
            for(j = 0; j < num_exp_slices_in_tile[i] - 1; j++) {
                SliceHeightInCtus[i + j] =
exp_slice_height_in_ctus_minus1[i][j] + 1
                remainingHeightInCtbsY -= SliceHeightInCtus[i + j]
            }
            uniformSliceHeight = exp_slice_height_in_ctus_minus1[i][j] +
1
            while(remainingHeightInCtbsY >= uniformSliceHeight) {
                SliceHeightInCtus[i + j] = uniformSliceHeight
                remainingHeightInCtbsY -= uniformSliceHeight
                j++
            }
            if(remainingHeightInCtbsY > 0) {
                SliceHeightInCtus[i + j] = remainingHeightInCtbsY
                j++
            }
            NumSlicesInTiles[i] = j
        }
        ctbY = tileRowBd[tileY]
        for(j = 0; j < NumSlicesInTiles[i]; j++) {
            AddCtbsToSlice(i, tileColBd[tileX], tileColBd[tileX + 1],
                ctbY, ctbY + SliceHeightInCtus[i])
            ctbY += SliceHeightInCtus[i]
            if(j < NumSlicesInTiles[i] - 1)

```

```

        i++
    }
} else
    for(j = 0; j < SliceHeightInTiles[i]; j++)
        for(k = 0; k < sliceWidthInTiles[i]; k++)
            AddCtbsToSlice(i, tileColBd[tileX + k],
tileColBd[tileX + k + 1],
tileRowBd[tileY + j], tileRowBd[tileY + j + 1])
if(i < num_slices_in_pic_minus1) {
    if(tile_idx_delta_present_flag)
        tileIdx += tile_idx_delta[i]
    else {
        tileIdx += sliceWidthInTiles[i]
        if(tileIdx % NumTileColumns == 0)
            tileIdx += (SliceHeightInTiles[i] - 1) * NumTileColumns
    }
}
}
}
}

```

....

D.7.2 Các ngữ nghĩa thông điệp SEI thông tin bậc ảnh con

...

ref_level_fraction_minus1[i][j] plus 1 chỉ ra phần của các giới hạn bậc liên kết với **ref_level_idc[i]** mà ảnh con thứ j tương thích với như được chỉ ra trong mục A.4.1.

Biến số **SubpicSizeY[j]** được thiết lập bằng $(\text{subpic_width_minus1}[j] + 1) * \text{CtbSizeY} * (\text{subpic_height_minus1}[j] + 1) * \text{CtbSizeY}$.

Khi không hiện diện, giá trị của **ref_level_fraction_minus1[i][j]** được suy ra bằng

$\text{Ceil}(256 * \text{SubpicSizeY}[j] \div \text{PicSizeInSamplesY} * \text{MaxLumaPs}(\text{general_level_idc}) \div \text{MaxLumaPs}(\text{ref_level_idc}[i]) - 1.$

Biến số $\text{RefLevelFraction}[i][j]$ được thiết lập bằng $\text{ref_level_fraction_minus1}[i][j] + 1.$

[[Các biến số $\text{SubpicNumTileCols}[j]$ và $\text{SubpicNumTileRows}[j]$ được rút ra như sau:

```

for(i = 0; i <= sps_num_subpics_minus1; i++) {
    SubpicNumTileCols[i] = 1
    SubpicNumTileRows[i] = 1
    for(ctbAddrRs = subpic_ctu_top_left_x[i] + 1; ctbAddrRs <=
        subpic_ctu_top_left_x[i] + subpic_width_minus1[i];
        ctbAddrRs++)
        if(CtbToTileColBd[ctbAddrRs] !=
            CtbToTileColBd[ctbAddrRs - 1])
            SubpicNumTileCols[i]++
            (D.5)
    for(ctbAddrRs = (subpic_ctu_top_left_y[i] + 1) * PicWidthInCtbsY;
        ctbAddrRs <= (subpic_ctu_top_left_y[i] +
            subpic_height_minus1[i]) * PicWidthInCtbsY;
            ctbAddrRs += PicWidthInCtbsY)
        if(CtbToTileRowBd[ctbAddrRs] !=
            CtbToTileRowBd[ctbAddrRs - PicWidthInCtbsY])
            SubpicNumTileRows[i]++
    }
}

```

...

Giá trị của ***SubpicWidthInTiles[j]*** phải nhỏ hơn hoặc bằng MaxTileCols và của ***SubPicHeightInTiles[j]*** phải nhỏ hơn hoặc bằng MaxTileRows , trong đó MaxTileCols và MaxTileRows được chỉ ra trong Bảng A.1 đối với bậc $\text{ref_level_idc}[i]$.

Giá trị của ***SubpicWidthInTiles[j]*** * ***SubPicHeightInTiles[j]*** phải nhỏ hơn hoặc bằng $\text{MaxTileCols} * \text{MaxTileRows} * \text{RefLevelFraction}[i][j]$, trong đó

MaxTileCols và MaxTileRows được chỉ ra trong Bảng A.1 đối với bậc $\text{ref_level_idc}[i]$.

...

Biến số $\text{SubpicSetAccLevelFraction}[i]$ đối với phần bậc tổng so với bậc tham chiếu $\text{ref_level_idc}[i]$, và các biến số $\text{SubpicSetCpbSizeVcl}[i]$, $\text{SubpicSetCpbSizeNal}[i]$, $\text{SubpicSetBitRateVcl}[i]$, và $\text{SubpicSetBitRateNal}[i]$ của tập ảnh con, được rút ra như sau:

```

for (i = 0; i <= num_ref_level_minus1; i++) {
    SubpicSetAccLevelFraction[i] = 0
    SubpicSetCpbSizeVcl[i] = 0
    SubpicSetCpbSizeNal[i] = 0
    SubpicSetNumTiles[i] = 0
    for (j = 0; j < NumSubpicsInSet; j++) {
        CurrSubpicIdx = SubpicSetIndices[j]
        SubpicSetAccLevelFraction[i] +=
RefLevelFraction[i][CurrSubpicIdx]                (D.10)
        SubpicSetCpbSizeVcl[i] +=
SubpicSetCpbSizeVcl[i][CurrSubpicIdx]
        SubpicSetCpbSizeNal[i] +=
SubpicSetCpbSizeNal[i][CurrSubpicIdx]
        SubpicSetBitRateVcl[i] +=
SubpicSetBitRateVcl[i][CurrSubpicIdx]
        SubpicSetBitRateNal[i] +=
SubpicSetBitRateNal[i][CurrSubpicIdx]
        SubpicSetNumTiles[i] +=
SubpicWidthInTiles[CurrSubpicIdx] *
SubPicHeightInTiles[CurrSubpicIdx]
    }
}

```

...

Phương án 2: hỗ trợ LMCS

Theo phương án này, cú pháp và các ngữ nghĩa của các phần tử cú pháp liên quan LMCS trong tiêu đề ảnh được cải biến, sao cho khi LMCS được sử dụng đối với tất cả các lát của ảnh, sự báo tín hiệu LMCS không hiện diện trong SH.

7.3.2.7 Cú pháp cấu trúc tiêu đề ảnh

picture_header_structure() {	Mô tả
gdr or irap pic flag	u(1)
if(gdr or irap pic flag)	
gdr pic flag	u(1)
ph inter slice allowed flag	u(1)
if(ph inter slice allowed flag)	
ph intra slice allowed flag	u(1)
non reference picture flag	u(1)
ph pic parameter set id	ue(v)
ph pic order cnt lsb	u(v)
if(gdr or irap pic flag)	
no output of prior pics flag	u(1)
if(gdr pic flag)	
recovery poc cnt	ue(v)
for(i = 0; i < NumExtraPhBits; i++)	
ph extra bit[i]	u(1)
if(sps poc msb flag) {	
ph poc msb present flag	u(1)
if(ph poc msb present flag)	
poc msb val	u(v)
}	
if(sps alf_enabled_flag && alf_info_in_ph_flag) {	
...	
}	
if(sps lmcs_enabled_flag) {	
[[ph_lmcs_enabled_flag]] <i>ph_lmcs_enabled_type</i>	[[u(1)]] <i>u(2)</i>
if([[ph_lmcs_enabled_flag]] <i>ph_lmcs_enabled_type</i>) {	
ph_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
ph_chroma_residual_scale_flag	u(1)
}	
}	
if(picture_header_extension_present_flag) {	
...	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
}	

7.3.7.1 Cú pháp tiêu đề lát chung

slice_header() {	Mô tả
picture_header_in_slice_header_flag	u(1)
slice_ts_residual_coding_disabled_flag	u(1)
if([[ph_lmcs_enabled_flag]] <i>ph_lmcs_enabled_type</i> == <i>N</i>)	
slice_lmcs_enabled_flag	u(1)
if(pic_scaling_list_enabled_flag)	
slice_scaling_list_present_flag	u(1)
...	
}	

ph_lmcs_enabled_type[[cờ]] bằng *M* (ví dụ, *M* = 1)[[1]] chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ được phép đối với tất cả các lát liên kết với PH. *ph_lmcs_enabled_flag* bằng *N* (ví dụ, *N* = 2) chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ là được phép đối với ít nhất một lát và không được phép đối với ít nhất một lát liên kết với PH. *ph_lmcs_enabled_flag* bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ [[có thể không được phép đối với một, hoặc nhiều, hoặc]] là không được phép đối với tất cả các lát liên kết với PH. Khi không hiện diện, giá trị của *ph_lmcs_enabled_flag* được suy ra là bằng 0.

slice_lmcs_enabled_flag bằng 1 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ được phép đối với lát hiện tại. *slice_lmcs_enabled_flag* bằng 0 chỉ ra rằng ánh xạ độ sáng với chọn thang sắc độ không được phép đối với lát hiện tại. Khi *slice_lmcs_enabled_flag* không hiện diện, nó được suy ra là bằng [[0]] (*ph_lmcs_enabled_type* ? 1 : 0).

Trong các ví dụ nêu trên, các giá trị của *M* và *N* có thể lần lượt được thiết lập bằng 1 và 2. Theo cách khác, các giá trị của *M* và *N* có thể lần lượt được thiết lập bằng 2 và 1.

FIG. 5 là sơ đồ khối thể hiện hệ thống xử lý video ví dụ 1900 mà trong đó các kỹ thuật khác nhau được mô tả ở đây có thể được thực hiện. Các phương án thực hiện khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 1900. Hệ thống 1900 có thể bao gồm đầu vào 1902 để nhận nội dung video. Nội dung video có thể được nhận trong định dạng thô hoặc không nén, ví dụ, các giá trị điểm ảnh đa thành phần 8 hoặc 10 bit, hoặc có thể trong định dạng được nén hoặc được mã hóa. Đầu vào 1902 có thể là giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện bộ lưu trữ. Các ví dụ về giao diện mạng bao gồm các

giao diện dây dẫn như mạng Ethernet, mạng cáp quang thụ động (PON- passive optical network), v.v. và các giao diện không dây như Wi-Fi hoặc các giao diện dạng ô.

Hệ thống 1900 có thể bao gồm thành phần tạo mã 1904 mà có thể thực hiện các phương pháp tạo mã hoặc mã hóa khác nhau mà được mô tả trong bản mô tả này. Thành phần tạo mã 1904 có thể làm giảm tốc độ bit trung bình của video từ đầu vào 1902 tới đầu ra của thành phần tạo mã 1904 để tạo ra sự biểu diễn được tạo mã của video. Do đó, các kỹ thuật tạo mã này đôi khi được gọi là các kỹ thuật nén video hoặc chuyển mã video. Đầu ra của thành phần tạo mã 1904 có thể hoặc được lưu, hoặc được truyền qua đường truyền thông được kết nối, như được biểu diễn bởi thành phần 1906. Sự biểu diễn dòng bit được lưu hoặc được truyền thông (hoặc được tạo mã) của video được nhận ở đầu vào 1902 có thể được sử dụng bởi thành phần 1908 để tạo ra các giá trị điểm ảnh hoặc video có thể hiển thị mà được gửi tới giao diện hiển thị 1910. Quá trình tạo ra video người dùng xem được từ sự biểu diễn dòng bit đôi khi được gọi là giải nén video. Hơn nữa, trong khi những hoạt động xử lý video đã biết được gọi là các hoạt động hoặc các công cụ “tạo mã”, thì cần hiểu rằng các công cụ hoặc các hoạt động tạo mã được sử dụng ở bộ mã hóa và các công cụ hoặc các hoạt động giải mã tương ứng mà đảo ngược các kết quả của sự tạo mã sẽ được thực hiện bởi bộ giải mã.

Các ví dụ về giao diện buýt ngoại vi hoặc giao diện hiển thị có thể bao gồm buýt nối tiếp vạn năng (USB - universal serial bus) hoặc giao diện đa phương tiện độ phân giải cao (HDMI - high definition multimedia interface) hoặc Displayport, và v.v.. Các ví dụ về các giao diện bộ lưu trữ bao gồm giao diện SATA (serial advanced technology attachment), PCI, IDE, và tương tự. Các kỹ thuật được mô tả trong bản mô tả có thể được thực hiện trong các thiết bị điện tử khác nhau như các điện thoại di động, các máy tính xách tay, các điện thoại thông minh hoặc các thiết bị khác mà có khả năng thực hiện xử lý dữ liệu kỹ thuật số và/hoặc hiển thị video.

FIG. 6 là sơ đồ khối của thiết bị xử lý video 3600. Thiết bị 3600 có thể được sử dụng để thực hiện một hoặc nhiều trong số các phương pháp được mô tả ở đây. Thiết bị 3600 có thể được thực hiện trong điện thoại thông minh, máy tính

bảng, máy tính, bộ thu internet vạn vật (IoT - Internet of Things), và v.v.. Thiết bị 3600 có thể bao gồm một hoặc nhiều bộ xử lý 3602, một hoặc nhiều bộ nhớ 3604 và phần cứng xử lý video 3606. (Các) bộ xử lý 3602 có thể được tạo cấu hình để thực hiện một hoặc nhiều phương pháp được mô tả trong bản mô tả này. Bộ nhớ (các bộ nhớ) 3604 có thể được sử dụng để lưu trữ dữ liệu và mã được sử dụng để thực hiện các phương pháp và các kỹ thuật được mô tả ở đây. Phần cứng xử lý video 3606 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được mô tả trong bản mô tả.

FIG. 8 là sơ đồ khối mà minh họa hệ thống tạo mã video 100 ví dụ mà có thể sử dụng các kỹ thuật theo sáng chế.

Như được thể hiện trên FIG. 8, hệ thống tạo mã video 100 có thể bao gồm thiết bị nguồn 110 và thiết bị đích 120. Thiết bị nguồn 110 tạo ra dữ liệu video được mã hóa mà có thể được gọi là thiết bị mã hóa video. Thiết bị đích 120 có thể giải mã dữ liệu video được mã hóa được tạo ra bởi thiết bị nguồn 110 mà có thể được gọi là thiết bị giải mã video.

Thiết bị nguồn 110 có thể bao gồm nguồn video 112, bộ mã hóa video 114, và giao diện đầu vào/đầu ra (I/O) 116.

Nguồn video 112 có thể bao gồm nguồn như thiết bị quay video, giao diện để nhận dữ liệu video từ nhà cung cấp nội dung video, và/hoặc hệ thống đồ họa máy tính để tạo ra dữ liệu video, hoặc sự kết hợp của các nguồn này. Dữ liệu video có thể bao gồm một hoặc nhiều ảnh. Bộ mã hóa video 114 mã hóa dữ liệu video từ nguồn video 112 để tạo ra dòng bit. Dòng bit có thể bao gồm dãy của các bit mà tạo thành sự biểu diễn được tạo mã của dữ liệu video. Dòng bit có thể bao gồm các ảnh được tạo mã và dữ liệu liên kết. Ảnh được tạo mã là sự biểu diễn được tạo mã của ảnh. Dữ liệu liên kết có thể bao gồm các tập thông số dãy, các tập thông số ảnh, và các cấu trúc cú pháp khác. Giao diện I/O 116 có thể bao gồm bộ điều biến/giải điều biến (môđem) và/hoặc bộ truyền. Dữ liệu video được mã hóa có thể được truyền trực tiếp tới thiết bị đích 120 qua giao diện I/O 116 thông qua mạng 130a. Dữ liệu video được mã hóa cũng có thể được lưu trong vật lưu trữ/máy chủ 130b để truy nhập bởi thiết bị đích 120.

Thiết bị đích 120 có thể bao gồm giao diện I/O 126, bộ giải mã video 124, và thiết bị hiển thị 122.

Giao diện I/O 126 có thể bao gồm bộ thu và/hoặc môđem. Giao diện I/O 126 có thể thu thập dữ liệu video được mã hóa từ thiết bị nguồn 110 hoặc vật lưu trữ/máy chủ 130b. Bộ giải mã video 124 có thể giải mã dữ liệu video được mã hóa. Thiết bị hiển thị 122 có thể hiển thị dữ liệu video được giải mã cho người dùng. Thiết bị hiển thị 122 có thể được tích hợp với thiết bị đích 120, hoặc có thể là ngoại vi đối với thiết bị đích 120 mà được tạo cấu hình để giao diện với thiết bị hiển thị ngoại vi.

Bộ mã hóa video 114 và bộ giải mã video 124 có thể hoạt động theo chuẩn nén video, như chuẩn tạo mã video hiệu quả cao (HEVC), chuẩn tạo mã video đa năng (VVM) và các chuẩn hiện tại và/hoặc tiếp theo khác.

FIG. 9 là sơ đồ khối minh họa ví dụ của bộ mã hóa video 200, mà có thể là bộ mã hóa video 114 trong hệ thống 100 được minh họa trên FIG. 8.

Bộ mã hóa video 200 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật theo sáng chế. Trong ví dụ trên FIG. 9, bộ mã hóa video 200 bao gồm nhiều thành phần chức năng. Các kỹ thuật được mô tả trong bản mô tả này có thể được chia sẻ giữa các thành phần khác nhau của bộ mã hóa video 200. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả trong bản mô tả này.

Các thành phần chức năng của bộ mã hóa video 200 có thể bao gồm khối phân vùng 201, khối xác nhận 202 mà có thể bao gồm khối chọn chế độ 203, khối ước tính chuyển động 204, khối bù chuyển động 205 và khối dự báo trong 206, khối tạo dư 207, khối chuyển đổi 208, khối lượng tử hóa 209, khối lượng tử hóa ngược 210, khối chuyển đổi ngược 211, khối tái tạo 212, bộ đệm 213, và khối mã hóa entropi 214.

Trong các ví dụ khác, bộ mã hóa video 200 có thể bao gồm nhiều hơn, ít hơn, hoặc các thành phần chức năng khác nhau. Trong một ví dụ, khối xác nhận 202 có thể bao gồm khối sao chép khối trong (IBC). Khối IBC có thể thực hiện việc xác nhận trong chế độ IBC mà trong đó ít nhất một ảnh tham chiếu là ảnh mà khối video hiện tại nằm trong đó.

Hơn nữa, một số thành phần, như khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể được tích hợp ở mức độ cao, nhưng được biểu diễn trong ví dụ trên FIG. 9 một cách riêng biệt cho mục đích giải thích.

Khối phân vùng 201 có thể phân vùng ảnh thành một hoặc nhiều khối video. Bộ mã hóa video 200 và bộ giải mã video 300 có thể hỗ trợ các kích thước khối video khác nhau.

Khối chọn chế độ 203 có thể chọn một chế độ trong số các chế độ tạo mã, trong hoặc ngoài, ví dụ, dựa trên các kết quả lỗi, và cấp khối được tạo mã trong hoặc ngoài thu được tới khối tạo dư 207 để tạo ra khối dữ liệu dư và tới khối tái tạo 212 để tái tạo khối được mã hóa để sử dụng dưới dạng ảnh tham chiếu. Trong một số ví dụ, khối chọn chế độ 203 có thể chọn sự kết hợp của chế độ xác nhận trong và ngoài (CIIP – combination of intra and inter predication) mà trong đó việc xác nhận là dựa trên tín hiệu xác nhận ngoài và tín hiệu xác nhận trong. Khối chọn chế độ 203 cũng có thể chọn độ phân giải đối với vector chuyển động (ví dụ, độ chính xác điểm ảnh con hoặc điểm ảnh nguyên) đối với khối trong trường hợp xác nhận ngoài.

Để thực hiện dự báo ngoài trên khối video hiện tại, khối ước tính chuyển động 204 có thể tạo ra thông tin chuyển động đối với khối video hiện tại bằng cách so sánh một hoặc nhiều khung tham chiếu từ bộ đệm 213 tới khối video hiện tại. Khối bù chuyển động 205 có thể xác định khối video được dự báo đối với khối video hiện tại dựa trên thông tin chuyển động và các mẫu được giải mã của các ảnh từ bộ đệm 213 mà khác với ảnh liên kết với khối video hiện tại.

Khối ước tính chuyển động 204 và khối bù chuyển động 205 có thể thực hiện các hoạt động khác nhau đối với khối video hiện tại, ví dụ, phụ thuộc vào việc liệu khối video hiện tại đang ở trong lát I, lát P, hoặc lát B.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể thực hiện dự báo đơn hướng đối với khối video hiện tại, và khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu của danh sách 0 hoặc danh sách 1 đối với khối video tham chiếu đối với khối video hiện tại. Khi đó khối ước tính chuyển động 204 có thể tạo ra chỉ số tham chiếu mà chỉ báo ảnh tham chiếu trong danh sách 0 hoặc danh sách 1 mà chứa khối video tham chiếu và vector chuyển động mà chỉ

bảo sự dịch chuyển không gian giữa khối video hiện tại và khối video tham chiếu. Khối ước tính chuyển động 204 có thể xuất ra chỉ số tham chiếu, bộ chỉ báo hướng dự báo, và vector chuyển động làm thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo ra khối video được dự báo của khối hiện tại dựa trên khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong các ví dụ khác, khối ước tính chuyển động 204 có thể thực hiện dự báo hai hướng đối với khối video hiện tại, khối ước tính chuyển động 204 có thể tìm kiếm các ảnh tham chiếu trong danh sách 0 đối với khối video tham chiếu đối với khối video hiện tại và cũng có thể tìm kiếm các ảnh tham chiếu trong danh sách 1 đối với khối video tham chiếu khác đối với khối video hiện tại. Khi đó khối ước tính chuyển động 204 có thể tạo ra các chỉ số tham chiếu mà chỉ báo các ảnh tham chiếu trong danh sách 0 và danh sách 1 chứa các khối video tham chiếu và các vector chuyển động mà chỉ báo sự dịch chuyển không gian giữa các khối video tham chiếu và khối video hiện tại. Khối ước tính chuyển động 204 có thể xuất ra các chỉ số tham chiếu và các vector chuyển động của khối video hiện tại làm thông tin chuyển động của khối video hiện tại. Khối bù chuyển động 205 có thể tạo ra khối video được dự báo của khối video hiện tại dựa trên các khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể xuất ra tập đầy đủ của thông tin chuyển động để xử lý giải mã của bộ giải mã.

Trong một số ví dụ, khối ước tính chuyển động 204 có thể không xuất ra tập đầy đủ của thông tin chuyển động đối với video hiện tại. Thay vào đó, khối ước tính chuyển động 204 có thể báo tín hiệu thông tin chuyển động của khối video hiện tại nhờ tham chiếu tới thông tin chuyển động của khối video khác. Ví dụ, khối ước tính chuyển động 204 có thể xác định rằng thông tin chuyển động của khối video hiện tại là đủ giống với thông tin chuyển động của khối video lân cận.

Trong một ví dụ, khối ước tính chuyển động 204 có thể chỉ báo, trong cấu trúc cú pháp liên kết với khối video hiện tại, giá trị mà chỉ báo tới bộ giải mã

video 300 rằng khối video hiện tại có thông tin chuyển động giống như khối video khác.

Trong một ví dụ khác, khối ước tính chuyển động 204 có thể nhận dạng, trong cấu trúc cú pháp liên kết với khối video hiện tại, khối video khác và hiệu số vector chuyển động (MVD – motion vector difference). Hiệu số vector chuyển động chỉ báo hiệu số giữa vector chuyển động của khối video hiện tại và vector chuyển động của khối video được chỉ báo. Bộ giải mã video 300 có thể sử dụng vector chuyển động của khối video được chỉ báo và hiệu số vector chuyển động để xác định vector chuyển động của khối video hiện tại.

Như đã đề cập trên đây, bộ mã hóa video 200 có thể báo tín hiệu theo cách dự báo vector chuyển động. Hai ví dụ về các kỹ thuật báo tín hiệu theo cách dự báo mà có thể được thực hiện bởi bộ mã hóa video 200 bao gồm xác nhận vector chuyển động cao cấp (AMVP – advanced motion vector predication) và báo tín hiệu chế độ kết hợp.

Khối dự báo trong 206 có thể thực hiện dự báo trong trên khối video hiện tại. Khi khối dự báo trong 206 thực hiện dự báo trong trên khối video hiện tại, khối dự báo trong 206 có thể tạo ra dữ liệu dự báo đối với khối video hiện tại dựa trên các mẫu được giải mã của các khối video khác trong cùng một ảnh. Dữ liệu dự báo đối với khối video hiện tại có thể bao gồm khối video được dự báo và các phần tử cú pháp khác nhau.

Khối tạo dư 207 có thể tạo ra dữ liệu dư đối với khối video hiện tại bằng cách trừ đi (ví dụ, được chỉ báo bởi dấu trừ) (các) khối video được dự báo của khối video hiện tại từ khối video hiện tại. Dữ liệu dư của khối video hiện tại có thể bao gồm các khối video dư mà tương ứng với các thành phần mẫu khác nhau của các mẫu trong khối video hiện tại.

Trong các ví dụ khác, có thể không có dữ liệu dư đối với khối video hiện tại đối với khối video hiện tại, ví dụ trong chế độ bỏ qua, và khối tạo dư 207 có thể không thực hiện hoạt động trừ.

Khối xử lý chuyển đổi 208 có thể tạo ra một hoặc nhiều khối video hệ số chuyển đổi đối với khối video hiện tại bằng cách áp dụng một hoặc nhiều phép chuyển đổi vào khối video dư liên kết với khối video hiện tại.

Sau khi khối xử lý chuyển đổi 208 tạo ra khối video hệ số chuyển đổi liên kết với khối video hiện tại, khối lượng tử hóa 209 có thể lượng tử hóa khối video hệ số chuyển đổi liên kết với khối video hiện tại dựa trên một hoặc nhiều giá trị thông số lượng tử hóa (QP) liên kết với khối video hiện tại.

Khối lượng tử hóa ngược 210 và khối chuyển đổi ngược 211 có thể lần lượt áp dụng lượng tử hóa ngược và các chuyển đổi ngược vào khối video hệ số chuyển đổi, để tái tạo khối video dư từ khối video hệ số chuyển đổi. Khối tái tạo 212 có thể bổ sung khối video dư được tái lập vào các mẫu tương ứng từ một hoặc nhiều khối video được dự báo được tạo ra bởi khối xác nhận 202 để tạo ra khối video được tái lập liên kết với khối hiện tại để lưu trữ trong bộ đệm 213.

Sau khi khối tái tạo 212 tái tạo khối video, hoạt động lọc vòng có thể được thực hiện để giảm các thành phần lạ tạo khối video trong khối video.

Khối mã hóa entropi 214 có thể nhận dữ liệu từ các thành phần chức năng khác của bộ mã hóa video 200. Khi khối mã hóa entropi 214 nhận dữ liệu, khối mã hóa entropi 214 có thể thực hiện một hoặc nhiều hoạt động mã hóa entropi để tạo ra dữ liệu được mã hóa entropi và xuất ra dòng bit mà bao gồm dữ liệu được mã hóa entropi.

Một số phương án của công nghệ được mô tả bao gồm việc ra quyết định hoặc xác định để cho phép công cụ hoặc chế độ xử lý video. Trong một ví dụ, khi công cụ hoặc chế độ xử lý video được phép, bộ mã hóa sẽ sử dụng hoặc thực hiện công cụ hoặc chế độ trong việc xử lý của khối của video, nhưng có thể không nhất thiết phải cải biến dòng bit thu được dựa trên việc sử dụng công cụ hoặc chế độ này. Nghĩa là, sự biến đổi từ khối của video thành dòng bit (hoặc sự biểu diễn dòng bit) của video sẽ sử dụng công cụ hoặc chế độ xử lý video khi nó được phép dựa trên quyết định hoặc sự xác định này. Trong một ví dụ khác, khi công cụ hoặc chế độ xử lý video được phép, bộ giải mã sẽ xử lý dòng bit khi biết rằng dòng bit đã được cải biến dựa trên công cụ hoặc chế độ xử lý video. Nghĩa là, sự biến đổi từ dòng bit của video thành khối của video sẽ được thực hiện bằng cách sử dụng công cụ hoặc chế độ xử lý video mà đã được phép dựa trên quyết định hoặc sự xác định này.

FIG. 10 là sơ đồ khối minh họa ví dụ của bộ giải mã video 300 mà có thể là bộ giải mã video 114 trong hệ thống 100 được minh họa trên FIG. 8.

Bộ giải mã video 300 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật theo sáng chế. Trong ví dụ trên FIG. 10, bộ giải mã video 300 bao gồm nhiều thành phần chức năng. Các kỹ thuật được mô tả trong bản mô tả này có thể được chia sẻ giữa các thành phần khác nhau của bộ giải mã video 300. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả trong bản mô tả này.

Trong ví dụ trên FIG. 10, bộ giải mã video 300 bao gồm khối giải mã entropi 301, khối bù chuyển động 302, khối dự báo trong 303, khối lượng tử hóa ngược 304, khối chuyển đổi ngược 305, và khối tái tạo 306 và bộ đệm 307. Bộ giải mã video 300 có thể, trong một số ví dụ, thực hiện chuyển tác giải mã thường là nghịch đảo với chuyển tác mã hóa được mô tả đối với bộ mã hóa video 200 (FIG. 9).

Khối giải mã entropi 301 có thể truy xuất dòng bit được mã hóa. Dòng bit được mã hóa có thể bao gồm dữ liệu video được tạo mã entropi (ví dụ, các khối được mã hóa của dữ liệu video). Khối giải mã entropi 301 có thể giải mã dữ liệu video được tạo mã entropi, và từ dữ liệu video được giải mã entropi, khối bù chuyển động 302 có thể xác định thông tin chuyển động bao gồm các vector chuyển động, độ chính xác vector chuyển động, các chỉ số danh sách ảnh tham chiếu, và thông tin chuyển động khác. Khối bù chuyển động 302 có thể, ví dụ, xác định thông tin này bằng cách thực hiện AMVP và chế độ kết hợp.

Khối bù chuyển động 302 có thể tạo ra các khối bù chuyển động, có thể bằng cách thực hiện nội suy dựa trên các bộ lọc nội suy. Các bộ nhận dạng đối với các bộ lọc nội suy mà được sử dụng với độ chính xác điểm ảnh con có thể được bao gồm trong các phần tử cú pháp.

Khối bù chuyển động 302 có thể sử dụng các bộ lọc nội suy như được sử dụng bởi bộ mã hóa video 200 trong khi mã hóa khối video để tính toán các giá trị nội suy đối với các điểm ảnh con số nguyên của khối tham chiếu. Khối bù chuyển động 302 có thể xác định các bộ lọc nội suy được sử dụng bởi bộ mã hóa

video 200 theo thông tin cú pháp nhận được và sử dụng các bộ lọc nội suy để tạo ra các khối dự báo.

Khối bù chuyển động 302 có thể sử dụng một số thông tin cú pháp để xác định các kích thước của các khối được sử dụng để mã hóa (các) khung và/hoặc (các) lát của dãy video được mã hóa, phân vùng thông tin mà mô tả cách thức mỗi khối macro của ảnh của dãy video được mã hóa được phân vùng, các chế độ mà chỉ báo cách thức mỗi phân vùng được mã hóa, một hoặc nhiều khung tham chiếu (và các danh sách khung tham chiếu) đối với mỗi khối được mã hóa ngoài, và thông tin khác để giải mã dãy video được mã hóa.

Khối dự báo trong 303 có thể sử dụng các chế độ dự báo trong ví dụ được nhận trong dòng bit để tạo thành khối dự báo từ các khối liền kề trong không gian. Khối lượng tử hóa ngược 303 lượng tử hóa ngược, tức là, giải lượng tử hóa, các hệ số khối video được lượng tử hóa được tạo ra trong dòng bit và được giải mã bởi khối giải mã entropi 301. Khối chuyển đổi ngược 303 áp dụng chuyển đổi ngược.

Khối tái tạo 306 có thể cộng các khối dự với các khối dự báo tương ứng được tạo ra bởi khối bù chuyển động 202 hoặc khối dự báo trong 303 để tạo thành các khối được giải mã. Nếu mong muốn, bộ lọc giải khối cũng có thể được áp dụng để lọc các khối được giải mã nhằm loại bỏ các thành phần lạ tạo khối. Sau đó các khối video được giải mã được lưu trong bộ đệm 307, mà cấp các khối tham chiếu để bù chuyển động/xác nhận trong tiếp sau và còn tạo ra video được giải mã để trình diễn trên thiết bị hiển thị.

Việc liệt kê các giải pháp được ưu tiên bởi một số phương án được đề cập sau đây.

Các giải pháp sau đây thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 1).

1. Phương pháp xử lý video (ví dụ, phương pháp 900 được mô tả trên FIG. 7), bao gồm các bước: thực hiện (902) sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra thông tin thứ nhất mà

được báo tín hiệu trong sự biểu diễn được tạo mã và thông tin thứ hai mà được rút ra từ sự biểu diễn được tạo mã, trong đó ít nhất thông tin thứ nhất hoặc thông tin thứ hai liên quan tới các chỉ số hàng hoặc các chỉ số cột của một hoặc nhiều ô ảnh.

2. Phương pháp theo giải pháp 1, trong đó quy tắc định dạng chỉ ra rằng chỉ số cột ô ảnh của mỗi cột đơn vị cây tạo mã của mỗi ảnh video được rút ra.

3. Phương pháp theo giải pháp 1, trong đó quy tắc định dạng chỉ ra rằng chỉ số hàng ô ảnh của mỗi hàng đơn vị cây tạo mã của mỗi ảnh video được rút ra.

Các giải pháp sau đây thể hiện các phương án ví dụ của các kỹ thuật được đề cập trong phần trước (ví dụ, mục 2). Trong các giải pháp này, vùng video có thể là ảnh video và đơn vị video có thể là khối video hoặc đơn vị cây tạo mã hoặc lát video.

4. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa đơn vị video của vùng video của video và sự biểu diễn được tạo mã của video, trong đó sự biểu diễn được tạo mã tương thích với quy tắc định dạng; trong đó quy tắc định dạng chỉ ra rằng thông tin điều khiển thứ nhất ở vùng video điều khiển việc liệu thông tin điều khiển thứ hai có được bao gồm ở bậc đơn vị video; trong đó thông tin điều khiển thứ nhất và/hoặc thông tin điều khiển thứ hai bao gồm thông tin về ánh xạ độ sáng và chọn thang sắc độ (LMCS) hoặc chọn thang dư sắc độ (CRS) hoặc quá trình tái tạo dạng (RP) được sử dụng để biến đổi.

5. Phương pháp theo giải pháp 4, trong đó thông tin điều khiển thứ nhất bao gồm bộ chỉ báo mà chỉ báo việc liệu thông tin điều khiển thứ hai có được bao gồm trong sự biểu diễn được tạo mã.

6. Phương pháp theo các giải pháp 4 và 5, trong đó giá trị cụ thể của thông tin điều khiển thứ nhất chỉ báo rằng LMCS là không được phép đối với tất cả các đơn vị video trong vùng video.

7. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 4 tới 6, trong đó thông tin điều khiển thứ hai điều khiển việc cho phép LMCS ở đơn vị video.

8. Phương pháp theo giải pháp 4, trong đó thông tin điều khiển thứ nhất bao gồm nhiều bộ chỉ báo.

9. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 tới 8, trong đó bước biến đổi bao gồm mã hóa video thành sự biểu diễn được tạo mã.

10. Phương pháp theo giải pháp bất kỳ trong số các giải pháp từ 1 tới 8, trong đó bước biến đổi bao gồm giải mã sự biểu diễn được tạo mã để tạo ra các giá trị điểm ảnh của video.

11. Thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều giải pháp trong số các giải pháp từ 1 tới 10.

12. Thiết bị mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều giải pháp trong số các giải pháp từ 1 tới 10.

13. Sản phẩm chương trình máy tính có mã máy tính được lưu trong đó, mã này, khi được thực thi bởi bộ xử lý, khiến bộ xử lý thực hiện phương pháp được nêu trong giải pháp bất kỳ trong số các giải pháp từ 1 tới 10.

14. Phương pháp, thiết bị hoặc hệ thống được mô tả trong bản mô tả.

Trong các giải pháp được mô tả ở đây, bộ mã hóa có thể tương thích với quy tắc định dạng bằng cách tạo ra sự biểu diễn được tạo mã theo quy tắc định dạng. Trong các giải pháp được mô tả ở đây, bộ giải mã có thể sử dụng quy tắc định dạng để phân tích các phần tử cú pháp trong sự biểu diễn được tạo mã khi biết về sự có mặt và vắng mặt của các phần tử cú pháp theo quy tắc định dạng để tạo ra video được giải mã.

FIG. 11 là lưu đồ của phương pháp ví dụ 1100 của quá trình xử lý video. Hoạt động 1102 việc bao gồm thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều cột ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số cột ô ảnh được rút ra đối với mỗi cột đơn vị cây tạo mã (CTU) của ô ảnh của ảnh video.

Theo một số phương án của phương pháp 1100, chỉ số cột ô ảnh đối với cột ô ảnh thứ `ctbAddrX`, được ký hiệu là `ctbToTileColIdx[ctbAddrX]`, được rút ra như sau:

```

tileX = 0
for(ctbAddrX = 0; ctbAddrX <= PicWidthInCtbsY; ctbAddrX++) {
    if(ctbAddrX == tileColBd[tileX + 1])
        tileX++
    ctbToTileColIdx[ctbAddrX] = tileX
},

```

trong đó PicWidthInCtbsY biểu diễn chiều rộng của ảnh video trong các đơn vị của các khối cây được tạo mã (các CTB), và trong đó tileColBd[i] biểu diễn vị trí của đường biên cột ô ảnh thứ i trong các đơn vị của các CTB.

Theo một số phương án của phương pháp 1100, mỗi ảnh video còn bao gồm một hoặc nhiều ảnh con, mỗi ảnh con bao gồm một hoặc nhiều lát mà cùng nhau tạo thành tập con hình chữ nhật của ảnh video, và quy tắc định dạng còn chỉ ra rằng chiều rộng của ảnh con trong các đơn vị của các cột ô ảnh được bao gồm được rút ra dựa trên các chỉ số cột ô ảnh của CTU xa nhất bên trái và/hoặc CTU xa nhất bên phải được bao gồm trong ảnh con.

Theo một số phương án của phương pháp 1100, chiều rộng của ảnh con thứ i trong các đơn vị của các cột ô ảnh, được ký hiệu là SubpicWidthInTiles[i], được rút ra như sau:

```

for(i = 0; i <= sps_num_subpics_minus1; i++) {
    leftX = sps_subpic_ctu_top_left_x[i]
    rightX = leftX + sps_subpic_width_minus1[i]
    SubpicWidthInTiles[i] =
        ctbToTileColIdx[rightX] + 1 - ctbToTileColIdx[leftX]
},

```

trong đó sps_num_subpics_minus1 biểu diễn số lượng của các ảnh con trong ảnh video, trong đó sps_subpic_ctu_top_left_x[i] biểu diễn vị trí ngang của CTU đỉnh-trái của ảnh con thứ i, trong đó sps_subpic_width_minus1[i] plus 1 chỉ ra chiều rộng của ảnh con thứ i, và trong đó ctbToTileColIdx[rightX] và ctbToTileColIdx[leftX] lần lượt biểu diễn các chỉ số cột ô ảnh của CTU xa nhất bên trái và CTU xa nhất bên phải, được bao gồm trong ảnh con.

Theo một số phương án của phương pháp 1100, đáp lại việc ô ảnh được phân vùng thành nhiều lát hình chữ nhật và chỉ tập con của các lát hình chữ nhật của ô ảnh được bao gồm trong ảnh con, ô ảnh này được đếm như một ô ảnh trong giá trị của chiều rộng của ảnh con.

FIG. 12 là lưu đồ của phương pháp ví dụ 1200 của quá trình xử lý video. Hoạt động 1202 bao gồm việc thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều hàng ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số hàng ô ảnh được rút ra đối với mỗi hàng đơn vị cây tạo mã (CTU) của ô ảnh của ảnh video.

Theo một số phương án của phương pháp 1200, chỉ số hàng ô ảnh đối với hàng ô ảnh thứ `ctbAddrY`, được ký hiệu là `ctbToTileRowIdx[ctbAddrY]`, được rút ra như sau:

```
tileY = 0
for(ctbAddrY = 0; ctbAddrY <= PicHeightInCtbsY; ctbAddrY++) {
    if(ctbAddrY == tileRowBd[tileY + 1])
        tileY++
    ctbToTileRowIdx[ctbAddrY] = tileY
},
```

trong đó `PicHeightInCtbsY` biểu diễn chiều cao của ảnh video trong các đơn vị của các khối cây được tạo mã (các CTB), và trong đó `tileRowBd[i]` biểu diễn vị trí của đường biên hàng ô ảnh thứ *i* trong các đơn vị của các CTB.

Theo một số phương án của phương pháp 1200, mỗi ảnh video còn bao gồm một hoặc nhiều ảnh con, mỗi ảnh con bao gồm một hoặc nhiều lát mà cùng nhau tạo thành tập con hình chữ nhật của ảnh video, và quy tắc định dạng còn chỉ ra rằng chiều cao của ảnh con trong các đơn vị của các hàng ô ảnh được rút ra dựa trên các chỉ số hàng ô ảnh của CTU đỉnh và/hoặc CTU đáy được bao gồm trong ảnh con.

Theo một số phương án của phương pháp 1200, chiều cao của ảnh con thứ i trong các đơn vị của các hàng ô ảnh, được ký hiệu là $\text{SubPicHeightInTiles}[i]$, được rút ra như sau:

```
for(i = 0; i <= sps_num_subpics_minus1; i++) {
    topY = sps_subpic_ctu_top_left_y[i]
    bottomY = topY + sps_subpic_height_minus1[i]
    SubPicHeightInTiles[i] =
        ctbToTileRowIdx[botY] + 1 - ctbToTileRowIdx[topY]
},
```

trong đó $\text{sps_num_subpics_minus1}$ biểu diễn số lượng của các ảnh con trong ảnh video, trong đó $\text{sps_subpic_ctu_top_left_y}[i]$ biểu diễn vị trí dọc của các CTU đỉnh-trái của ảnh con thứ i , trong đó $\text{sps_subpic_height_minus1}[i]$ plus 1 chỉ ra chiều cao của ảnh con thứ i , và trong đó $\text{ctbToTileRowIdx}[\text{botY}]$ và $\text{ctbToTileRowIdx}[\text{topY}]$ lần lượt biểu diễn các chỉ số hàng ô ảnh của CTU đáy và CTU đỉnh, được bao gồm trong ảnh con.

Theo một số phương án của phương pháp 1200, đáp lại ô ảnh được phân vùng thành nhiều lát hình chữ nhật và chỉ tập con của các lát hình chữ nhật của ô ảnh được bao gồm trong ảnh con, ô ảnh này được đếm như một ô ảnh trong giá trị của chiều cao của ảnh con.

FIG. 13 là lưu đồ của phương pháp ví dụ 1300 của quá trình xử lý video. Hoạt động 1302 bao gồm thực hiện sự biến đổi giữa video bao gồm ít nhất một ảnh video và dòng bit của video theo quy tắc, trong đó ít nhất một ảnh video bao gồm một hoặc nhiều lát và một hoặc nhiều ảnh con, và trong đó quy tắc chỉ ra rằng thứ tự của các chỉ số lát của một hoặc nhiều lát trong ít nhất một ảnh video được chỉ báo đáp lại phần tử cú pháp liên kết với ít nhất một ảnh chỉ báo về việc liệu một lát có được bao gồm trên ảnh con của ít nhất một ảnh video.

Theo một số phương án của phương pháp 1300, quy tắc còn chỉ ra rằng các chỉ số lát được chỉ báo đáp lại mỗi lát trong ít nhất một ảnh video là lát hình chữ nhật. Theo một số phương án của phương pháp 1300, quy tắc chỉ ra rằng trong trường hợp mà phần tử cú pháp chỉ báo rằng mỗi ảnh con trong số một hoặc nhiều ảnh con bao gồm một lát hình chữ nhật, thì thứ tự tương ứng với các giá

trị tăng lên của các chỉ số ảnh con của một hoặc nhiều ảnh con trong ảnh video, và các chỉ số ảnh con của một hoặc nhiều ảnh con được chỉ báo trong tập thông số dây (SPS) được tham chiếu tới bởi ít nhất một ảnh video. Theo một số phương án của phương pháp 1300, quy tắc chỉ ra rằng trong trường hợp mà phân tử cú pháp chỉ báo rằng mỗi ảnh con bao gồm một hoặc nhiều lát hình chữ nhật, thì thứ tự này tương ứng với thứ tự mà trong đó một hoặc nhiều lát được bao gồm trong tập thông số ảnh (PPS) được tham chiếu tới bởi ít nhất một ảnh video. Theo một số phương án của phương pháp 1300, phân tử cú pháp được bao gồm trong tập thông số ảnh (PPS) được tham chiếu tới bởi ít nhất một ảnh video.

FIG. 14 là lưu đồ của phương pháp ví dụ 1400 của quá trình xử lý video. Hoạt động 1402 bao gồm thực hiện sự biến đổi giữa đơn vị video của vùng video của video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra rằng thông tin điều khiển thứ nhất ở bậc thứ nhất vùng video trong dòng bit điều khiển việc liệu thông tin điều khiển thứ hai có được bao gồm ở bậc thứ hai của đơn vị video trong dòng bit, trong đó bậc thứ hai là nhỏ hơn bậc thứ nhất, trong đó thông tin điều khiển thứ nhất và thông tin điều khiển thứ hai bao gồm thông tin về việc liệu hoặc cách thức công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS) có được áp dụng cho đơn vị video, và trong đó công cụ LMCS bao gồm việc sử dụng chọn thang dư sắc độ (CRS), hoặc quá trình tái tạo dạng độ sáng (RP) để biến đổi.

Theo một số phương án của phương pháp 1400, thông tin điều khiển thứ nhất bao gồm theo cách có lựa chọn bộ chỉ báo thứ nhất mà chỉ báo việc liệu công cụ LMCS có là được phép đối với một hoặc nhiều lát ở bậc thứ nhất của vùng video để chỉ ra liệu công cụ LMCS có là được phép ở bậc thứ hai của đơn vị video, và bộ chỉ báo thứ nhất là giá trị không nhị phân. Theo một số phương án của phương pháp 1400, bậc thứ nhất của vùng video bao gồm tiêu đề ảnh. Theo một số phương án của phương pháp 1400, bậc thứ nhất của vùng video bao gồm tiêu đề ảnh, thông tin điều khiển thứ nhất bao gồm bộ chỉ báo thứ nhất, công cụ LMCS là được phép đối với tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ nhất, công cụ LMCS là được phép đối với ít hơn tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ hai, công cụ LMCS là không

được phép đối với tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ ba, và giá trị thứ nhất, giá trị thứ hai, và giá trị thứ ba là khác nhau. Theo một số phương án của phương pháp 1400, giá trị của bộ chỉ báo thứ nhất được suy ra là bằng giá trị mặc định khi thông tin điều khiển thứ nhất loại trừ bộ chỉ báo thứ nhất.

Theo một số phương án của phương pháp 1400, bậc thứ nhất của vùng video bao gồm tiêu đề ảnh, thông tin điều khiển thứ nhất bao gồm bộ chỉ báo thứ nhất, công cụ LMCS là không được phép đối với tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ nhất, công cụ LMCS là không được phép đối với ít hơn tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ hai, công cụ LMCS là được phép đối với tất cả các lát của tiêu đề ảnh khi bộ chỉ báo thứ nhất là bằng giá trị thứ ba, và giá trị thứ nhất, giá trị thứ hai, và giá trị thứ ba là khác nhau. Theo một số phương án của phương pháp 1400, việc liệu bộ chỉ báo thứ nhất có được bao gồm theo cách có lựa chọn trong thông tin điều khiển thứ nhất dựa trên giá trị của phần tử cú pháp trong dòng bit mà chỉ báo việc liệu công cụ LMCS có là được phép ở bậc đấy. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ nhất được tạo mã với $u(v)$ hoặc $u(2)$ hoặc $ue(v)$. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ nhất được tạo mã với mã đơn phân bị chặt.

Theo một số phương án của phương pháp 1400, thông tin tập thông số thích ứng (APS) của công cụ LMCS được sử dụng bởi một hoặc nhiều lát và/hoặc phần tử cú pháp chọn thang sắc độ được bao gồm trong dòng bit dựa trên giá trị của bộ chỉ báo thứ nhất mà chỉ báo việc liệu công cụ LMCS có là được phép đối với một hoặc nhiều lát ở bậc thứ nhất của vùng video. Theo một số phương án của phương pháp 1400, thông tin điều khiển thứ hai bao gồm theo cách có lựa chọn bộ chỉ báo thứ hai mà chỉ báo việc liệu công cụ LMCS là được phép hoặc không được phép đối với một hoặc nhiều lát ở bậc thứ hai của đơn vị video, và bộ chỉ báo thứ hai được bao gồm trong dòng bit dựa trên giá trị của bộ chỉ báo thứ nhất được bao gồm trong thông tin điều khiển thứ nhất, và bộ chỉ báo thứ nhất chỉ báo việc liệu công cụ LMCS là được phép hoặc không được phép đối với một hoặc nhiều lát ở bậc thứ hai của đơn vị video. Theo một số phương án

của phương pháp 1400, thông tin điều khiển thứ hai bao gồm tiêu đề lát. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ hai được bao gồm trong thông tin điều khiển thứ hai đáp lại bộ chỉ báo thứ nhất bằng giá trị thứ nhất. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ hai được bao gồm trong thông tin điều khiển thứ hai đáp lại việc thực hiện kiểm tra điều kiện là: bộ chỉ báo thứ nhất $\gg 1$, hoặc bộ chỉ báo thứ nhất/2, hoặc bộ chỉ báo thứ nhất & 0x01, trong đó $\gg$ mô tả hoạt động dịch phải, và trong đó & mô tả logic và hoạt động thao tác bit.

Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ hai được suy ra để chỉ báo rằng công cụ LMCS là được phép đối với một hoặc nhiều lát ở bậc thứ hai của đơn vị video đáp lại việc bộ chỉ báo thứ nhất bằng giá trị thứ nhất, hoặc bộ chỉ báo thứ hai được suy ra để chỉ báo rằng công cụ LMCS là không được phép đối với một hoặc nhiều lát ở bậc thứ hai của đơn vị video đáp lại việc bộ chỉ báo thứ nhất bằng giá trị thứ ba, và giá trị thứ nhất, giá trị thứ hai của bộ chỉ báo thứ nhất, và giá trị thứ ba là khác nhau. Theo một số phương án của phương pháp 1400, thông tin điều khiển thứ nhất bao gồm nhiều bộ chỉ báo mà chỉ báo việc liệu công cụ LMCS là được phép đối với một hoặc nhiều lát ở bậc thứ nhất của vùng video để chỉ ra việc liệu công cụ LMCS có là được phép ở bậc thứ hai của đơn vị video, và nhiều bộ chỉ báo có các giá trị không phải nhị phân. Theo một số phương án của phương pháp 1400, nhiều bộ chỉ báo bao gồm ít nhất hai bộ chỉ báo được bao gồm trong tiêu đề ảnh. Theo một số phương án của phương pháp 1400, ít nhất hai bộ chỉ báo này bao gồm bộ chỉ báo thứ nhất mà chỉ ra liệu công cụ LMCS có là được phép đối với ít nhất một lát liên kết với tiêu đề ảnh, và ít nhất hai bộ chỉ báo bao gồm theo cách có lựa chọn bộ chỉ báo thứ hai mà chỉ ra liệu công cụ LMCS có là được phép đối với tất cả các lát liên kết với tiêu đề ảnh. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ hai là hiện diện theo cách có lựa chọn trong nhiều bộ chỉ báo dựa trên giá trị của bộ chỉ báo thứ nhất.

Theo một số phương án của phương pháp 1400, giá trị của bộ chỉ báo thứ nhất chỉ báo rằng công cụ LMCS là được phép đối với ít nhất một lát. Theo một số phương án của phương pháp 1400, công cụ LMCS được suy ra là được phép

đối với tất cả các lát liên kết với tiêu đề ảnh đáp lại việc bộ chỉ báo thứ hai là không có mặt từ dòng bit. Theo một số phương án của phương pháp 1400, ít nhất hai bộ chỉ báo bao gồm bộ chỉ báo thứ ba mà được bao gồm theo cách có lựa chọn trong tiêu đề lát dựa trên giá trị thứ hai của bộ chỉ báo thứ hai. Theo một số phương án của phương pháp 1400, giá trị thứ hai của bộ chỉ báo thứ hai chỉ báo rằng công cụ LMCS là không được phép đối với tất cả các lát. Theo một số phương án của phương pháp 1400, giá trị đối với bộ chỉ báo thứ ba được suy ra dựa trên giá trị thứ nhất của bộ chỉ báo thứ nhất và/hoặc giá trị thứ hai của bộ chỉ báo thứ hai đáp lại việc bộ chỉ báo thứ ba là không có mặt từ dòng bit. Theo một số phương án của phương pháp 1400, ít nhất hai bộ chỉ báo bao gồm bộ chỉ báo thứ nhất mà chỉ ra việc liệu công cụ LMCS có là không được phép đối với ít nhất một lát liên kết với tiêu đề ảnh, và ít nhất hai bộ chỉ báo bao gồm theo cách có lựa chọn bộ chỉ báo thứ hai mà chỉ ra việc liệu công cụ LMCS có là không được phép đối với tất cả các lát liên kết với tiêu đề ảnh. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ hai là hiện diện trong nhiều bộ chỉ báo dựa trên giá trị của bộ chỉ báo thứ nhất. Theo một số phương án của phương pháp 1400, giá trị của bộ chỉ báo thứ nhất chỉ ra rằng công cụ LMCS là không được phép đối với ít nhất một lát. Theo một số phương án của phương pháp 1400, công cụ LMCS được suy ra là không được phép đối với tất cả các lát liên kết với tiêu đề ảnh đáp lại việc bộ chỉ báo thứ hai là không có mặt từ dòng bit. Theo một số phương án của phương pháp 1400, ít nhất hai bộ chỉ báo bao gồm theo cách có lựa chọn bộ chỉ báo thứ ba trong tiêu đề lát dựa trên giá trị thứ hai của bộ chỉ báo thứ hai.

Theo một số phương án của phương pháp 1400, giá trị thứ hai của bộ chỉ báo thứ hai chỉ ra rằng công cụ LMCS là được phép đối với tất cả các lát. Theo một số phương án của phương pháp 1400, giá trị đối với bộ chỉ báo thứ ba được suy ra dựa trên giá trị thứ nhất của bộ chỉ báo thứ nhất và/hoặc giá trị thứ hai của bộ chỉ báo thứ hai đáp lại việc bộ chỉ báo thứ ba là không có mặt từ dòng bit. Theo một số phương án của phương pháp 1400, nhiều bộ chỉ báo bao gồm theo cách có lựa chọn bộ chỉ báo thứ nhất dựa trên giá trị của phần tử cú pháp mà chỉ báo việc liệu công cụ LMCS có là được phép ở bậc đây. Theo một số phương án của

phương pháp 1400, nhiều bộ chỉ báo bao gồm theo cách có lựa chọn bộ chỉ báo thứ ba mà chỉ báo việc liệu công cụ LMCS có là được phép hoặc không được phép ở bậc thứ hai của đơn vị video, và bộ chỉ báo thứ ba là hiện diện theo cách có lựa chọn dựa trên giá trị thứ nhất của bộ chỉ báo thứ nhất và/hoặc giá trị thứ hai của bộ chỉ báo thứ hai. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ ba là hiện diện theo cách có lựa chọn dựa trên bộ chỉ báo thứ hai mà chỉ báo rằng công cụ LMCS là không được phép đối với tất cả các lát hoặc rằng công cụ LMCS không phải là không được phép đối với tất cả các lát. Theo một số phương án của phương pháp 1400, bộ chỉ báo thứ nhất, bộ chỉ báo thứ hai, và/hoặc bộ chỉ báo thứ ba điều khiển việc sử dụng CRS hoặc độ sáng RP.

FIG. 15 là lưu đồ của phương pháp ví dụ 1500 của quá trình xử lý video. Hoạt động 1502 bao gồm thực hiện sự biến đổi giữa video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS) là được phép khi phần tử cú pháp thứ nhất trong tập thông số dãy được tham chiếu chỉ báo rằng công cụ LMCS là được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng khi phần tử cú pháp thứ nhất chỉ báo rằng công cụ LMCS là không được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS là được phép đối với tất cả các lát liên kết với tiêu đề ảnh của ảnh video khi phần tử cú pháp thứ hai trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề ảnh của video, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng đối với tất cả các lát liên kết với tiêu đề ảnh khi phần tử cú pháp thứ hai chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề ảnh của video, trong đó quy tắc chỉ ra rằng công cụ LMCS được sử dụng đối với lát hiện tại liên kết với tiêu đề lát của ảnh video khi phần tử cú pháp thứ ba được bao gồm theo cách có lựa chọn trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề lát của video, và trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng đối với lát hiện tại khi phần tử cú pháp thứ ba chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề lát của video.

Theo một số phương án của phương pháp 1500, quy tắc chỉ ra rằng phần tử cú pháp thứ ba không được bao gồm tiêu đề lát trong dòng bit khi công cụ LMCS được sử dụng đối với tất cả các lát của ảnh video. Theo một số phương án của

phương pháp 1500, việc liệu công cụ LMCS là được phép hoặc không được phép là dựa trên phần tử cú pháp thứ hai. Theo một số phương án của phương pháp 1500, công cụ LMCS là được phép khi công cụ LMCS được sử dụng đối với tất cả các lát của ảnh video, và công cụ LMCS là không được phép khi công cụ LMCS không được sử dụng đối với tất cả các lát của ảnh video.

FIG. 16 là lưu đồ của phương pháp ví dụ 1600 của quá trình xử lý video. Hoạt động 1602 bao gồm thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng việc liệu độ phân giải hiệu vector chuyển động thích ứng (AMVR) có được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin dựa trên phần tử cú pháp được bao gồm theo cách có lựa chọn trong tập thông số dãy (SPS) được tham chiếu mà chỉ báo việc liệu AMVR có được phép, trong đó quy tắc chỉ ra rằng AMVR không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin khi phần tử cú pháp chỉ báo rằng AMVR là không được phép, và trong đó quy tắc chỉ ra rằng AMVR được suy ra là không được sử dụng trong tạo mã vector chuyển động của chế độ ngoài ảnh afin khi phần tử cú pháp khi phần tử cú pháp không được bao gồm trong SPS.

FIG. 17 là lưu đồ của phương pháp ví dụ 1700 của quá trình xử lý video. Hoạt động 1702 bao gồm thực hiện sự biến đổi giữa video bao gồm ảnh video và dòng bit của video theo quy tắc, trong đó ảnh video bao gồm ảnh con, ô ảnh, và lát, và trong đó quy tắc chỉ ra rằng, do ảnh con bao gồm lát mà được phân vùng từ ô ảnh, sự biến đổi được thực hiện bằng cách không đếm chiều cao của ảnh con bằng cách sử dụng số lượng các ô ảnh của ảnh video.

Theo một số phương án của phương pháp 1700, chiều cao của ảnh con được đếm dựa trên số lượng của các đơn vị cây tạo mã (các CTU). Theo một số phương án của phương pháp 1700, chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh.

FIG. 18 là lưu đồ của phương pháp ví dụ 1800 của quá trình xử lý video. Hoạt động 1802 bao gồm thực hiện sự biến đổi giữa video bao gồm ảnh video và dòng bit của video, trong đó dòng bit chỉ báo chiều cao của ảnh con của ảnh video mà được đếm dựa trên số lượng của các đơn vị cây tạo mã (các CTU) của ảnh video.

Theo một số phương án của phương pháp 1800, chiều cao của ảnh con không được dựa trên số lượng của các ô ảnh của ảnh video. Theo một số phương án của phương pháp 1800, chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh.

FIG. 19 là lưu đồ của phương pháp ví dụ 1900 của quá trình xử lý video. Hoạt động 1902 bao gồm tiến hành xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video có là nhỏ hơn chiều cao của hàng ô ảnh của ảnh video. Hoạt động 1904 bao gồm thực hiện, bằng cách sử dụng sự xác định, biến đổi giữa video và dòng bit của video.

Theo một số phương án của phương pháp 1900, quy tắc chỉ ra rằng chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh khi: ảnh con chỉ bao gồm các đơn vị cây tạo mã (các CTU) từ một hàng ô ảnh, và hoặc nhóm thứ nhất của các CTU nằm trên đỉnh của ảnh con là không giống như nhóm thứ hai của các CTU nằm trên đỉnh của một hàng ô ảnh, hoặc nhóm thứ ba của các CTU nằm trên đáy của ảnh con là không giống như nhóm thứ tư của các CTU nằm trên đáy của một hàng ô ảnh. Theo một số phương án của phương pháp 1900, khi mỗi ảnh con của ảnh video chỉ bao gồm một lát và chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh, đối với mỗi lát với chỉ số lát bậc ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ được rút ra từ địa chỉ CTU quét màn hình ảnh của CTU thứ j trong quét màn hình CTU của ảnh con, và j nằm trong khoảng từ 0 tới số lượng của các CTU trong lát trừ 1, bao gồm các giá trị biên.

Theo một số phương án của phương pháp 1900, quy tắc chỉ ra rằng chiều cao của ảnh con là nhỏ hơn một hàng ô ảnh khi: khoảng cách giữa nhóm thứ nhất của các CTU nằm trên đỉnh của ảnh con và nhóm thứ hai của các CTU nằm trên đáy của ảnh con là nhỏ hơn chiều cao thứ hai của ô ảnh của ảnh video, trong đó chiều cao thứ hai của ô ảnh là dựa trên số lượng của các CTU của ảnh con. Theo một số phương án của phương pháp 1900, khi mỗi ảnh con của ảnh video chỉ bao gồm một lát và chiều cao của ảnh con là lớn hơn hoặc bằng một hàng ô ảnh, đối với mỗi lát với chỉ số lát bậc ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ được rút ra từ địa chỉ CTU quét màn hình ảnh của CTU thứ j theo thứ tự của các CTU trong ảnh con, và j nằm trong khoảng từ 0 tới số lượng của các CTU trong lát trừ 1, bao gồm các giá trị biên.

Theo một số phương án của phương pháp 1900, thứ tự của các CTU trong ảnh con là sao cho CTU thứ nhất trong ô ảnh thứ nhất với chỉ số ô ảnh thứ nhất được đặt trước CTU thứ hai trong ô ảnh thứ hai với chỉ số ô ảnh thứ hai, và giá trị của chỉ số ô ảnh thứ nhất là nhỏ hơn giá trị của chỉ số ô ảnh thứ hai. Theo một số phương án của phương pháp 1900, thứ tự của các CTU trong ảnh con là sao cho các CTU bên trong một ô ảnh trong ảnh con được xếp thứ tự trong quét mảnh của các CTU trong một ô ảnh.

Theo một số phương án của (các) phương pháp từ 1100 tới 1900, bước thực hiện sự biến đổi bao gồm mã hóa video thành dòng bit. Theo một số phương án của (các) phương pháp từ 1100 tới 1900, bước thực hiện sự biến đổi bao gồm mã hóa video thành dòng bit, và phương pháp còn bao gồm bước lưu trữ dòng bit trong vật ghi máy tính đọc được bất biến. Theo một số phương án của (các) phương pháp từ 1100 tới 1900, bước thực hiện sự biến đổi bao gồm giải mã video từ dòng bit.

Theo một số phương án, thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới 1900. Theo một số phương án, thiết bị mã hóa video bao gồm bộ xử lý được tạo cấu hình để thực hiện các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới 1900. Theo một số phương án, sản phẩm chương trình máy tính có các lệnh máy tính được lưu trong đó, các lệnh này, khi được thực thi bởi bộ xử lý, khiến bộ xử lý thực hiện các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới 1900. Theo một số phương án, vật lưu trữ máy tính đọc được bất biến mà lưu trữ dòng bit được tạo ra theo các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới 1900. Theo một số phương án, vật lưu trữ máy tính đọc được bất biến lưu trữ các lệnh mà khiến bộ xử lý thực hiện các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới 1900. Theo một số phương án, phương pháp tạo ra dòng bit, bao gồm các bước: tạo ra dòng bit của video theo các hoạt động được mô tả đối với một hoặc nhiều phương pháp bất kỳ trong số các phương pháp từ 1100 tới

1900, và lưu trữ dòng bit trong môi trường chương trình máy tính đọc được. Theo một số phương án, phương pháp, thiết bị, dòng bit được tạo ra theo phương pháp được mô tả hoặc hệ thống được mô tả trong bản mô tả này.

Trong bản mô tả, thuật ngữ “xử lý video” có thể dùng để chỉ việc mã hóa video, giải mã video, nén video hoặc giải nén video. Ví dụ, các thuật toán nén video có thể được áp dụng trong khi biến đổi từ sự biểu diễn điểm ảnh của video thành sự biểu diễn dòng bit tương ứng hoặc ngược lại. Sự biểu diễn dòng bit của khối video hiện tại có thể, ví dụ, tương ứng với các bit mà hoặc là cùng nằm ở hoặc trải dài trong các vị trí khác nhau bên trong dòng bit, như được xác định bởi cú pháp. Ví dụ, khối macro có thể được mã hóa theo các giá trị dư lỗi được chuyển đổi và được tạo mã và cũng bằng cách sử dụng các bit trong các tiêu đề và các trường khác trong dòng bit. Hơn nữa, trong khi biến đổi, bộ giải mã có thể phân tích dòng bit khi biết rằng một số trường có thể hiện diện, hoặc không hiện diện, dựa trên xác định, như được mô tả trong các giải pháp trên đây. Tương tự, bộ mã hóa có thể xác định rằng các trường cú pháp đã biết là được hoặc không được bao gồm và tạo ra sự biểu diễn được tạo mã một cách tương ứng bằng cách bao gồm hoặc loại trừ các trường cú pháp này khỏi sự biểu diễn được tạo mã.

Các giải pháp, các ví dụ, các phương án, các môđun được mô tả và khác nữa và các hoạt động chức năng được mô tả trong bản mô tả này có thể được thực hiện trong hệ mạch điện tử kỹ thuật số, hoặc trong phần mềm, phần sụn, hoặc phần cứng máy tính, bao gồm các cấu trúc được mô tả trong bản mô tả này và các phương án tương đương về mặt cấu trúc của chúng, hoặc trong các sự kết hợp của một hoặc nhiều trong số chúng. Các phương án được mô tả và khác nữa có thể được thực hiện dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, tức là, một hoặc nhiều môđun của các lệnh chương trình máy tính được mã hóa trong môi trường máy tính đọc được để thực thi bởi, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Môi trường máy tính đọc được có thể là thiết bị lưu trữ máy đọc được, nền lưu trữ máy đọc được, thiết bị nhớ, sản phẩm thực hiện tín hiệu được lan truyền máy đọc được, hoặc sự kết hợp của một hoặc nhiều đối tượng trong số chúng. Thuật ngữ “thiết bị xử lý dữ liệu” bao hàm tất cả các thiết bị, các bộ phận, và các máy để xử lý dữ liệu, bao gồm ví dụ bộ xử lý lập

trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, ngoài phần cứng, mã mà tạo nên môi trường thực thi đối với chương trình máy tính đang được xem xét, ví dụ, mã mà cấu thành phần sụn bộ xử lý, ngăn xếp giao thức, hệ thống quản lý cơ sở dữ liệu, hệ điều hành, hoặc sự kết hợp của một hoặc nhiều đối tượng trong số chúng. Tín hiệu được lan truyền là tín hiệu được tạo ra một cách nhân tạo, ví dụ, tín hiệu điện, quang học, hoặc điện từ được tạo ra bằng máy, mà được tạo ra để mã hóa thông tin để truyền tới thiết bị thu thích hợp.

Chương trình máy tính (còn được biết như chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết dưới dạng bất kỳ của ngôn ngữ lập trình, bao gồm các ngôn ngữ biên dịch hoặc diễn dịch, và có thể được triển khai dưới dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, thủ tục con, hoặc đơn vị khác thích hợp để sử dụng trong môi trường máy tính. Chương trình máy tính không nhất thiết phải tương ứng với tập tin trong hệ thống tập tin. Chương trình có thể được lưu trong phần của tập tin mà giữ chương trình khác hoặc dữ liệu (ví dụ, một hoặc nhiều tập lệnh được lưu trong tài liệu ngôn ngữ đánh dấu), trong một tập tin dành riêng cho chương trình đang được xem xét, hoặc trong nhiều tập tin được kết hợp (ví dụ, các tập tin mà lưu trữ một hoặc nhiều môđun, các chương trình con, hoặc các phần của mã). Chương trình máy tính có thể được triển khai để được thực thi trên một máy tính hoặc trên nhiều máy tính mà nằm ở một địa điểm hoặc được phân bố trên nhiều địa điểm và được liên kết bởi mạng truyền thông.

Các quá trình và các luồng logic được mô tả trong bản mô tả này có thể được thực hiện bởi một hoặc nhiều bộ xử lý lập trình được mà thực thi một hoặc nhiều chương trình máy tính để thực hiện các chức năng bằng cách điều hành dữ liệu đầu vào và tạo ra đầu ra. Các quá trình và các luồng logic cũng có thể được thực hiện bởi, và thiết bị cũng có thể được thực hiện dưới dạng, hệ mạch logic chuyên dụng, ví dụ, mảng công trường lập trình được FPGA (field programmable gate array) hoặc mạch tích hợp chuyên dụng ASIC (application specific integrated circuit).

Các bộ xử lý thích hợp để thực thi chương trình máy tính bao gồm, ví dụ, cả bộ vi xử lý đa năng lẫn chuyên dụng, và một hoặc nhiều bộ xử lý bất kỳ của loại máy tính kỹ thuật số bất kỳ. Thông thường, bộ xử lý sẽ nhận các lệnh và dữ liệu từ chỉ bộ nhớ chỉ đọc hoặc bộ nhớ truy cập ngẫu nhiên hoặc cả hai. Các phần tử cơ bản của máy tính là bộ xử lý để thực hiện các lệnh và một hoặc nhiều thiết bị nhớ để lưu trữ các lệnh và dữ liệu. Thông thường, máy tính sẽ còn bao gồm, hoặc được kết nối theo cách hoạt động được để nhận dữ liệu từ hoặc chuyển dữ liệu tới, hoặc cả hai, một hoặc nhiều thiết bị lưu trữ dung lượng lớn để lưu trữ dữ liệu, ví dụ, các đĩa từ, quang từ, hoặc các đĩa quang học. Tuy nhiên, máy tính không cần phải có các thiết bị này. Môi trường máy tính đọc được thích hợp để lưu trữ các lệnh chương trình máy tính và dữ liệu bao gồm tất cả các dạng của bộ nhớ bất biến, môi trường và các thiết bị nhớ, bao gồm ví dụ các thiết bị nhớ bán dẫn, ví dụ, EPROM, EEPROM, và các thiết bị nhớ nhanh; các đĩa từ, ví dụ, các ổ cứng trong hoặc ổ cứng lưu động; các đĩa quang từ; và các đĩa CD ROM và DVD-ROM. Bộ xử lý và bộ nhớ có thể được bổ sung bằng, hoặc được tích hợp trong, hệ mạch logic chuyên dụng.

Trong khi bản mô tả này chứa nhiều dấu hiệu cụ thể, chúng không được coi là mang tính giới hạn đối với phạm vi của đối tượng bất kỳ hoặc của những gì mà có thể được yêu cầu bảo hộ, mà chỉ là sự mô tả các dấu hiệu mà có thể là cụ thể đối với các phương án cụ thể của các kỹ thuật cụ thể. Các dấu hiệu nhất định mà được mô tả trong bản mô tả này trong ngữ cảnh của các phương án riêng biệt cũng có thể được thực hiện trong sự kết hợp trong một phương án. Ngược lại, các dấu hiệu khác nhau mà được mô tả trong ngữ cảnh của một phương án cũng có thể được thực hiện trong nhiều phương án một cách riêng biệt hoặc trong sự kết hợp nhỏ thích hợp bất kỳ. Hơn nữa, mặc dù các dấu hiệu có thể được mô tả trên đây là hoạt động trong các sự kết hợp đã biết và thậm chí ban đầu được yêu cầu bảo hộ như vậy, một hoặc nhiều dấu hiệu từ sự kết hợp được yêu cầu bảo hộ có thể trong một số trường hợp được loại bỏ khỏi sự kết hợp này, và sự kết hợp được yêu cầu bảo hộ có thể hướng tới sự kết hợp nhỏ hoặc biến thể của sự kết hợp nhỏ.

Tương tự, trong khi các hoạt động được mô tả trên các hình vẽ theo thứ tự cụ thể, điều này không được hiểu là yêu cầu rằng các hoạt động này được thực hiện theo thứ tự cụ thể được thể hiện hoặc theo thứ tự liên tiếp, hoặc rằng tất cả các hoạt động được minh họa được thực hiện, để đạt được các kết quả mong muốn. Hơn nữa, sự tách biệt các thành phần hệ thống khác nhau trong các phương án được mô tả trong bản mô tả này không được hiểu là yêu cầu rằng sự tách biệt này có trong tất cả các phương án.

Chỉ một vài phương án thực hiện và các ví dụ là được mô tả và các phương án thực hiện khác nữa, các nâng cấp và các biến thể có thể được thực hiện dựa trên những gì được mô tả và được minh họa trong bản mô tả này.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý video bao gồm các bước:

xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video nhỏ hơn chiều cao của hàng ô của ảnh video; và

thực hiện, sử dụng bước xác định này, biến đổi giữa video và dòng bit của video,

trong đó quy tắc xác định rằng chiều cao của ảnh phụ nhỏ hơn một hàng ô khi khoảng cách giữa nhóm thứ nhất của các đơn vị cây tạo mã (coding tree unit, CTU) được đặt trên cùng của ảnh phụ và nhóm thứ hai của các CTU được đặt ở dưới cùng của ảnh phụ nhỏ hơn chiều cao thứ hai của ô của ảnh video, và

trong đó chiều cao thứ hai của ô dựa trên số lượng của CTU của ô.

2. Phương pháp theo điểm 1, trong đó quy tắc xác định rằng chiều cao của ảnh phụ nhỏ hơn một hàng ô khi:

ảnh phụ chỉ bao gồm các CTU từ một hàng ô, và

nhóm thứ nhất của các CTU được đặt trên cùng của ảnh phụ không giống như nhóm thứ hai của các CTU được đặt trên cùng của một hàng ô, hoặc nhóm thứ ba của các CTU được đặt ở dưới cùng của ảnh phụ không giống như nhóm thứ tư của các CTU được đặt ở dưới cùng của một hàng ô.

3. Phương pháp theo điểm 1, trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ nhỏ hơn một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ khối cây tạo mã (coding tree block, CTB) quét mảnh ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j khi quét mảnh CTB của ảnh phụ, và trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút.

4. Phương pháp theo điểm 1, trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ lớn hơn hoặc bằng một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét mảnh ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j theo thứ tự của các CTB trong ảnh phụ,

và trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút.

5. Phương pháp theo điểm 4, trong đó thứ tự của các CTB trong ảnh phụ là sao cho CTB thứ nhất trong ô thứ nhất với chỉ số ô thứ nhất được đặt trước CTB thứ hai trong ô thứ hai có chỉ số ô thứ hai, và trong đó giá trị của chỉ số ô thứ nhất nhỏ hơn giá trị của chỉ số ô thứ hai.

6. Phương pháp theo điểm 4, trong đó thứ tự của các CTB trong một ô trong ảnh phụ có thứ tự khi quét mảnh các CTB trong một ô.

7. Phương pháp theo điểm 1, trong đó thực hiện biến đổi bao gồm bước mã hóa video thành dòng bit.

8. Phương pháp theo điểm 1, trong đó thực hiện biến đổi bao gồm bước giải mã video từ dòng bit.

9. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ bất biến có các lệnh trên đó, trong đó các lệnh khi thực thi bằng bộ xử lý, khiến bộ xử lý:

xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video nhỏ hơn chiều cao của hàng ô của ảnh video; và

thực hiện, sử dụng bước xác định này, biến đổi giữa video và dòng bit của video,

trong đó quy tắc xác định rằng chiều cao của ảnh phụ nhỏ hơn một hàng ô khi khoảng cách giữa nhóm thứ nhất của các CTU được đặt trên cùng của ảnh phụ và nhóm thứ hai của các CTU được đặt ở dưới cùng của ảnh phụ nhỏ hơn chiều cao thứ hai của ô của ảnh video, và

trong đó chiều cao thứ hai của ô dựa trên số lượng của CTU của ô.

10. Thiết bị theo điểm 9, trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ nhỏ hơn một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của `CtbAddrInSlice[i][j]` xác định địa chỉ CTB quét mảnh ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j khi quét mảnh CTB của ảnh phụ, và trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị

đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút.

11. Thiết bị theo điểm 9,

trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ lớn hơn hoặc bằng một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét mảnh ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j theo thứ tự của các CTB trong ảnh phụ,

trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút,

trong đó thứ tự của các CTB trong ảnh phụ sao cho CTB thứ nhất trong ô thứ nhất với chỉ số ô thứ nhất được đặt trước CTB thứ hai trong ô thứ hai có chỉ số ô thứ hai,

trong đó giá trị của chỉ số ô thứ nhất nhỏ hơn giá trị của chỉ số ô thứ hai, và

trong đó thứ tự của các CTB trong một ô trong ảnh phụ có thứ tự khi quét mảnh của các CTB trong một ô.

12. Vật ghi máy tính đọc được bất biến lưu trữ các lệnh mà khiến bộ xử lý:

xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video nhỏ hơn chiều cao của hàng ô của ảnh video; và

thực hiện, sử dụng bước xác định này, biến đổi giữa video và dòng bit của video,

trong đó quy tắc xác định rằng chiều cao của ảnh phụ nhỏ hơn một hàng ô khi khoảng cách giữa nhóm thứ nhất của các CTU được đặt trên cùng của ảnh phụ và nhóm thứ hai của các CTU được đặt ở dưới cùng của ảnh phụ nhỏ hơn chiều cao thứ hai của ô của ảnh video, và

trong đó chiều cao thứ hai của ô dựa trên số lượng của CTU của ô.

13. Vật ghi máy tính đọc được bất biến theo điểm 12,

trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ nhỏ hơn một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét mảnh ảnh của

CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j khi quét mảnh CTB của ảnh phụ, và

trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút.

14. Vật ghi máy tính đọc được bất biến theo điểm 12,

trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ lớn hơn hoặc bằng một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét mảnh ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét mảnh ảnh của CTB thứ j theo thứ tự của các CTB trong ảnh phụ,

trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút,

trong đó thứ tự của các CTB trong ảnh phụ sao cho CTB thứ nhất trong ô thứ nhất với chỉ số ô thứ nhất được đặt trước CTB thứ hai trong ô thứ hai có chỉ số ô thứ hai,

trong đó giá trị của chỉ số ô thứ nhất nhỏ hơn giá trị của chỉ số ô thứ hai, và

trong đó thứ tự của các CTB trong một ô trong ảnh phụ có thứ tự khi quét mảnh của các CTB trong một ô.

15. Vật ghi máy tính đọc được lưu trữ dòng bit của video được tạo bởi phương pháp được thực hiện bởi thiết bị xử lý video, trong đó phương pháp bao gồm các bước:

xác định, theo quy tắc, về việc liệu chiều cao của ảnh con của ảnh video của video nhỏ hơn chiều cao của hàng ô của ảnh video; và

tạo, sử dụng bước xác định này, dòng bit của video,

trong đó quy tắc xác định rằng chiều cao của ảnh phụ nhỏ hơn một hàng ô khi khoảng cách giữa nhóm thứ nhất của các CTU được đặt trên cùng của ảnh phụ và nhóm thứ hai của các CTU được đặt ở dưới cùng của ảnh phụ nhỏ hơn chiều cao thứ hai của ô của ảnh video, và

trong đó chiều cao thứ hai của ô dựa trên số lượng của CTU của ô.

16. Vật ghi máy tính đọc được bất biến theo điểm 15,

trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ nhỏ hơn một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét màn hình ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét màn hình ảnh của CTB thứ j khi quét màn hình CTB của ảnh phụ, và

trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút.

17. Vật ghi máy tính đọc được bất biến theo điểm 15,

trong đó khi mỗi ảnh phụ của ảnh video bao gồm chỉ một lát và chiều cao của ảnh phụ lớn hơn hoặc bằng một hàng ô, đối với mỗi lát có chỉ số lát mức ảnh i của ảnh video, giá trị của $CtbAddrInSlice[i][j]$ xác định địa chỉ CTB quét màn hình ảnh của CTB thứ j trong lát được dẫn xuất từ địa chỉ CTB quét màn hình ảnh của CTB thứ j theo thứ tự của các CTB trong ảnh phụ,

trong đó i nằm trong khoảng từ 0 đến số lượng lát trong ảnh video trừ đi 1, bao gồm hai giá trị đầu mút và j nằm trong khoảng từ 0 đến số lượng của các CTU trong lát trừ đi 1, bao gồm hai giá trị đầu mút,

trong đó thứ tự của các CTB trong ảnh phụ sao cho CTB thứ nhất trong ô thứ nhất với chỉ số ô thứ nhất được đặt trước CTB thứ hai trong ô thứ hai có chỉ số ô thứ hai,

trong đó giá trị của chỉ số ô thứ nhất nhỏ hơn giá trị của chỉ số ô thứ hai, và

trong đó thứ tự của các CTB trong một ô trong ảnh phụ có thứ tự khi quét màn hình của các CTB trong một ô.

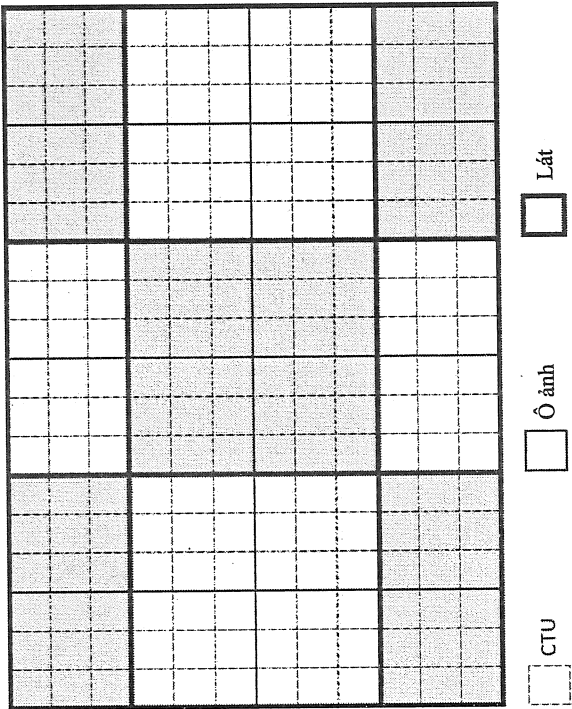
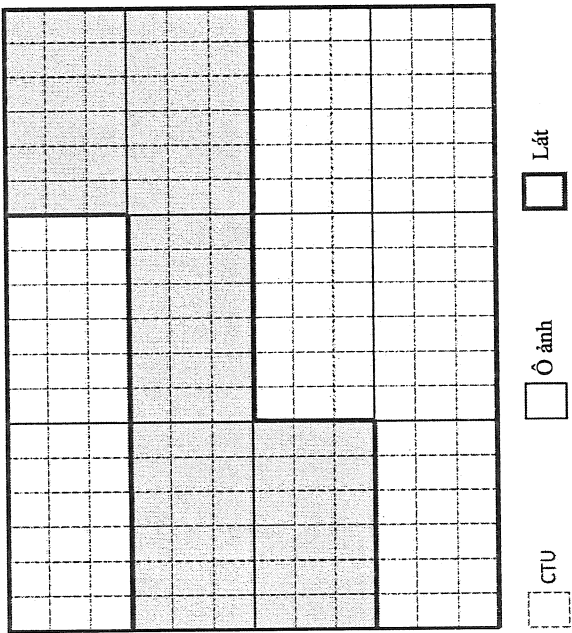


FIG. 1



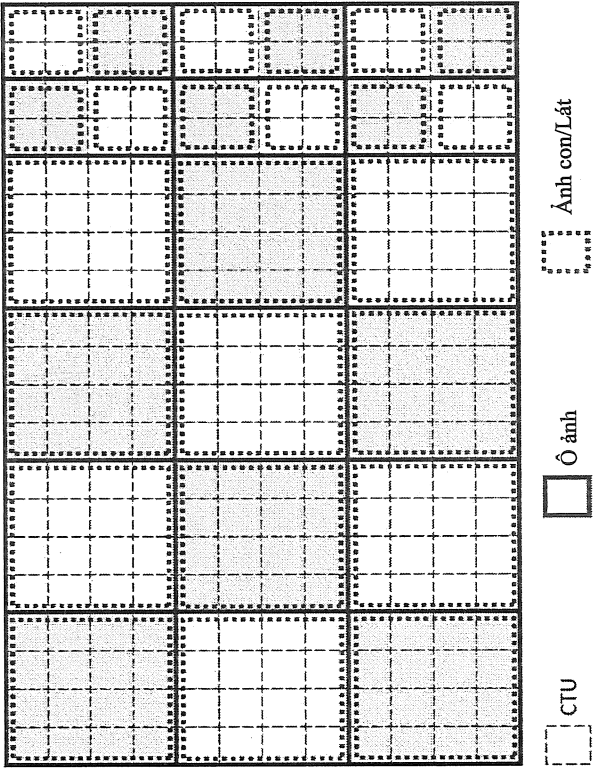


FIG. 3

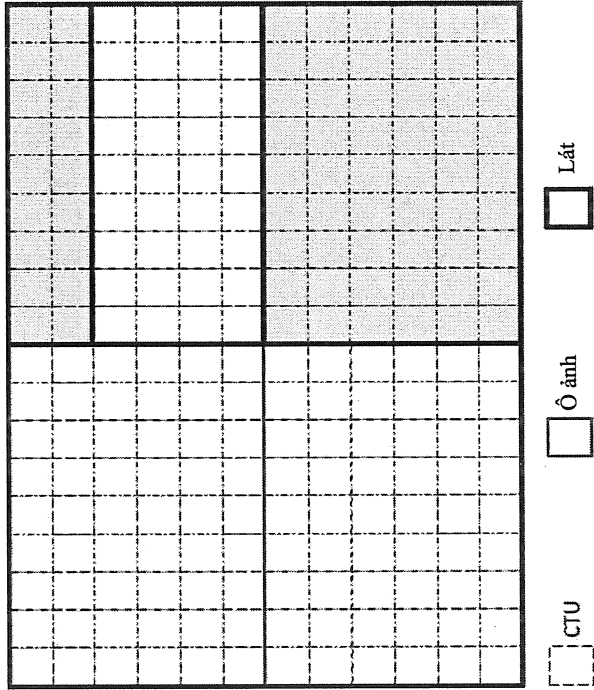
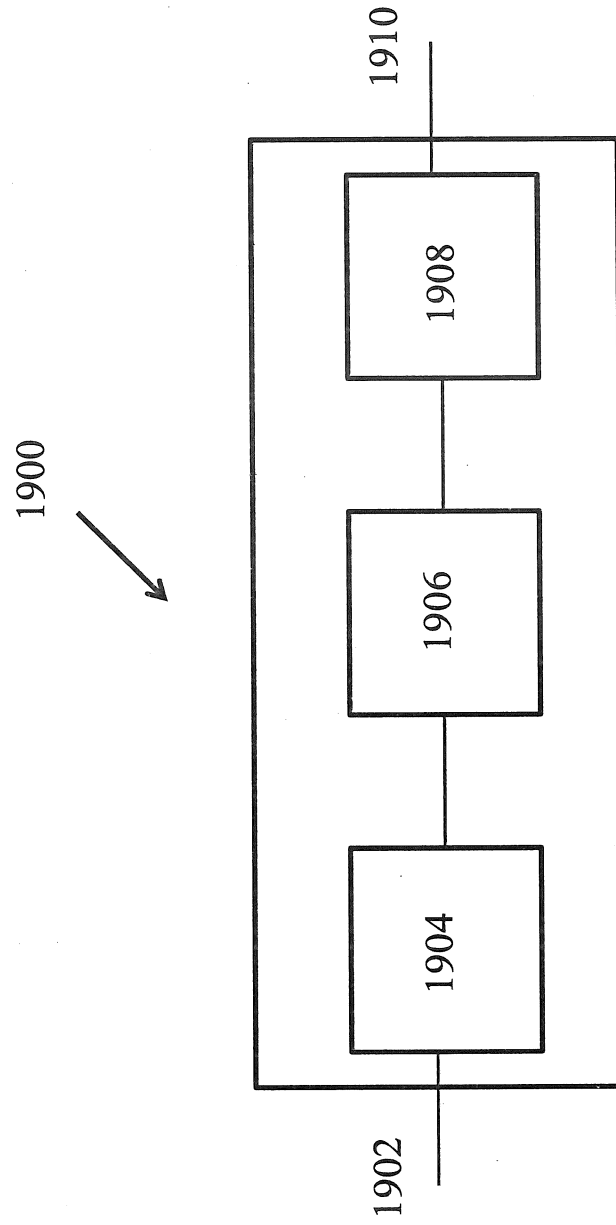


FIG. 4

**FIG. 5**

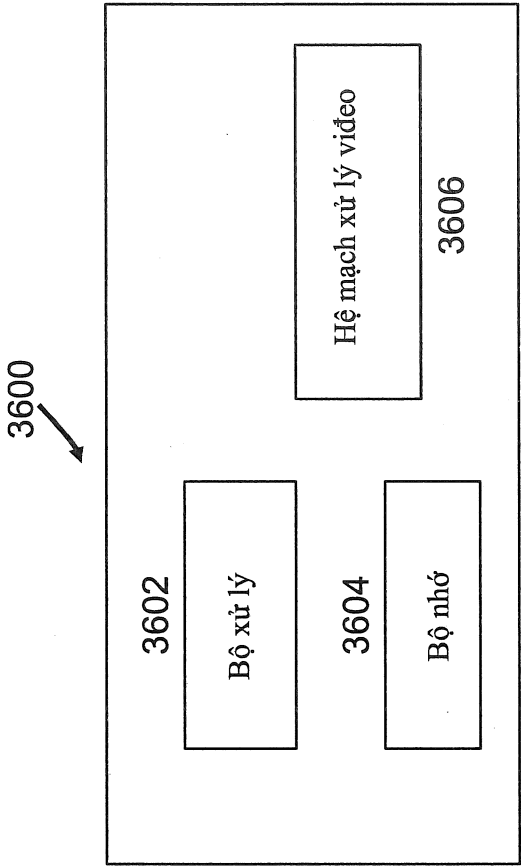
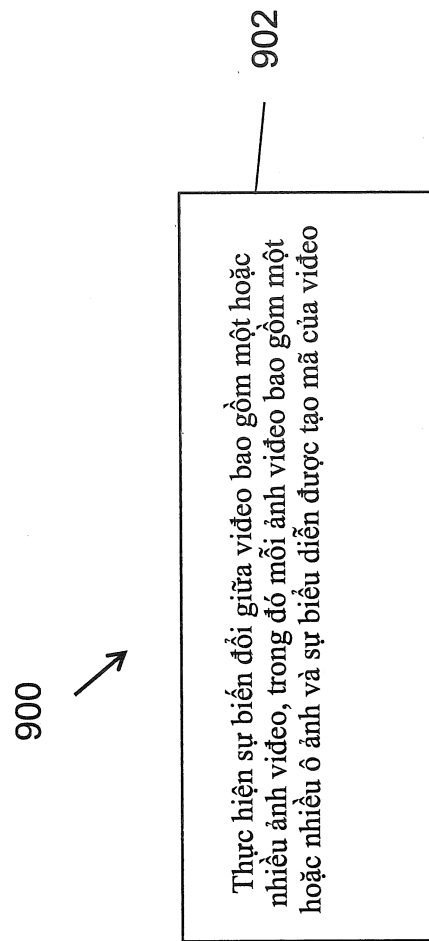


FIG. 6

**FIG. 7**

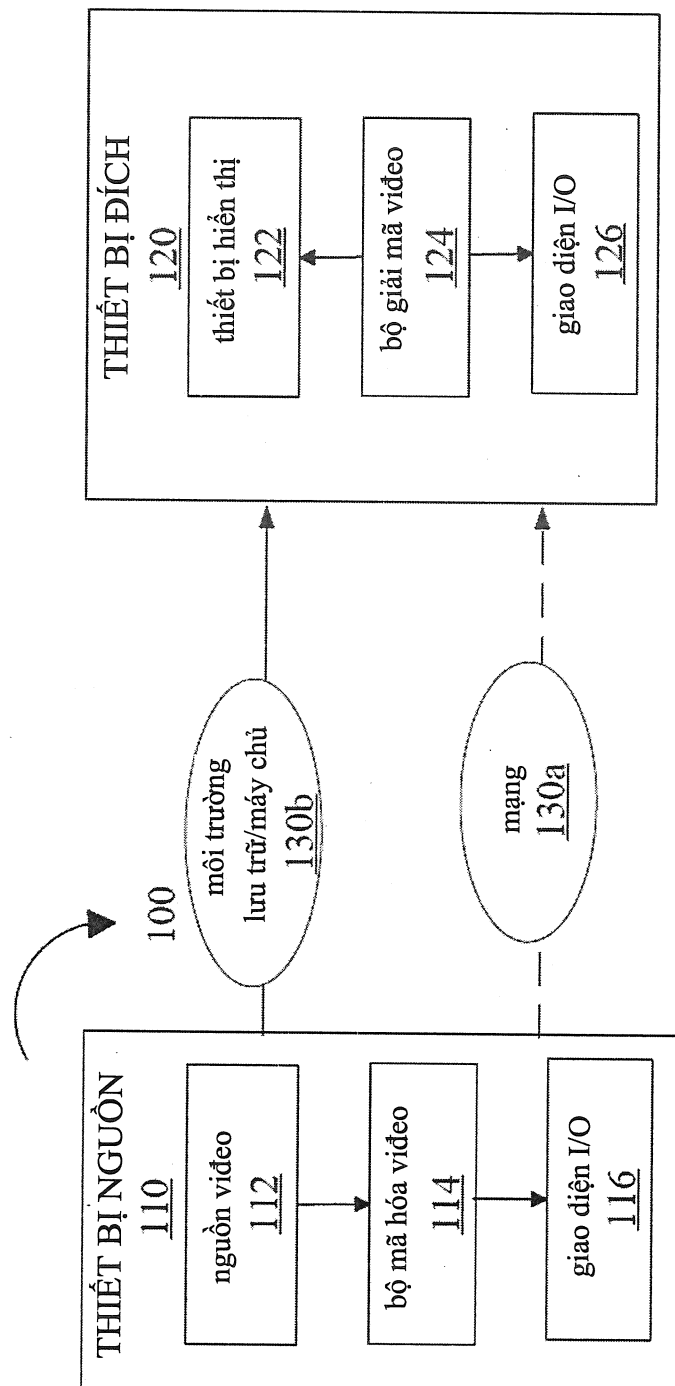


FIG. 8

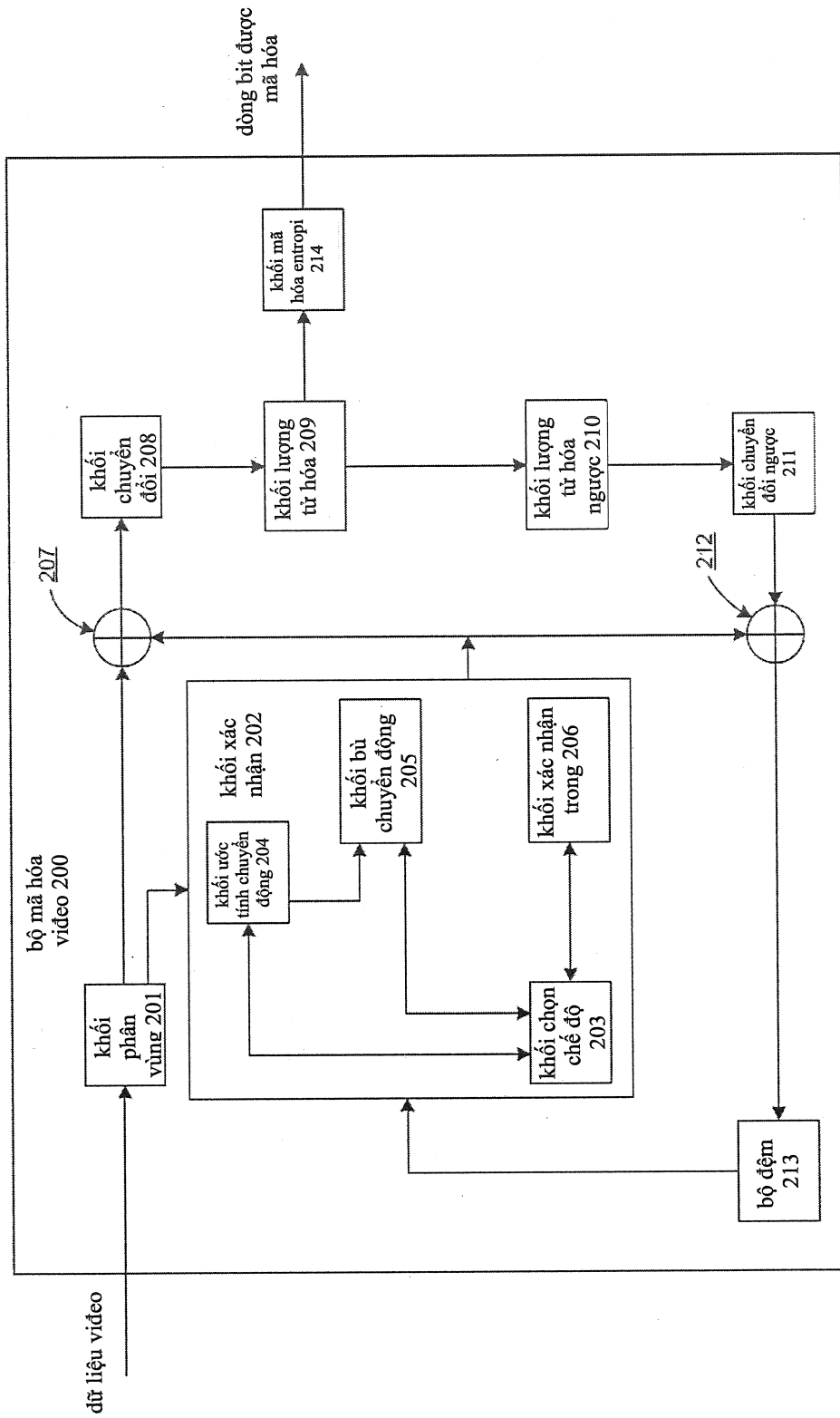


FIG. 9

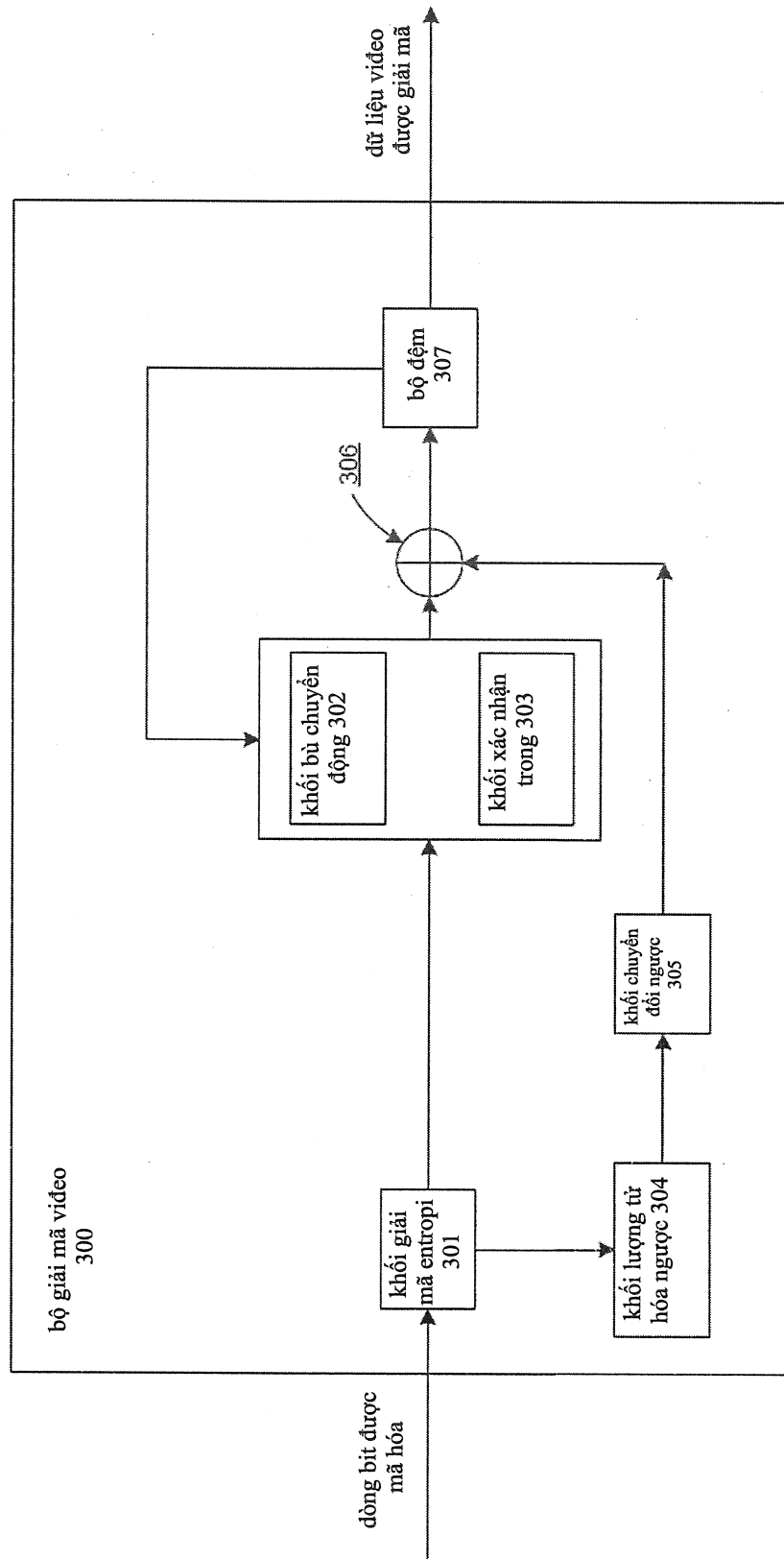


FIG. 10

1100



Thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều cột ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số cột ô ảnh được rút ra đối với mỗi cột đơn vị cây tạo mã (CTU) của ô ảnh của ảnh video

1102

FIG. 11

1200



1202

Thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video, trong đó mỗi ảnh video bao gồm một hoặc nhiều ô ảnh mà bao gồm một hoặc nhiều hàng ô ảnh, trong đó dòng bit tương thích với quy tắc định dạng, và trong đó quy tắc định dạng chỉ ra rằng chỉ số hàng ô ảnh được rút ra đối với mỗi hàng đơn vị cây tạo mã (CTU) của ô ảnh của ảnh video

FIG. 12

1300



1302

Thực hiện sự biến đổi giữa video bao gồm ít nhất một ảnh video và dòng bit của video theo quy tắc, trong đó ít nhất một ảnh video bao gồm một hoặc nhiều lát và một hoặc nhiều ảnh con, và trong đó quy tắc chỉ ra rằng thứ tự của các chỉ số lát của một hoặc nhiều lát trong ít nhất một ảnh video được chỉ báo lặp lại phân tử cú pháp liên kết với ít nhất một ảnh chỉ báo việc liệu một lát có được bao gồm trên mỗi ảnh con của ít nhất một ảnh video hay không

FIG. 13

1400



1402

Thực hiện sự biến đổi giữa đơn vị video của vùng video của video và dòng bit của video, trong đó dòng bit tương thích với quy tắc định dạng, trong đó quy tắc định dạng chỉ ra rằng thông tin điều khiển thứ nhất ở bậc thứ nhất của vùng video trong dòng bit điều khiển việc liệu thông tin điều khiển thứ hai có được bao gồm ở bậc thứ hai của đơn vị video trong dòng bit, trong đó bậc thứ hai nhỏ hơn bậc thứ nhất, trong đó thông tin điều khiển thứ nhất và thông tin điều khiển thứ hai bao gồm thông tin về việc liệu hoặc cách thức mà công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS) có được áp dụng cho đơn vị video, và trong đó công cụ LMCS bao gồm sử dụng việc chọn thang dư sắc độ (CRS), hoặc quá trình tái tạo dạng độ sáng (RP) để biến đổi

FIG. 14

1500



1502

Thực hiện sự biến đổi giữa video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng công cụ ánh xạ độ sáng và chọn thang sắc độ (LMCS) là được phép khi phân tử cú pháp thứ nhất trong tập thông số đây được tham chiếu chỉ báo rằng công cụ LMCS là được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng khi phân tử cú pháp thứ nhất chỉ báo rằng công cụ LMCS là không được phép, trong đó quy tắc chỉ ra rằng công cụ LMCS là được phép đối với tất cả các lát liên kết với tiêu đề ảnh của ảnh video khi phân tử cú pháp thứ hai trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề ảnh của video, trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng từ cú pháp thứ hai chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề ảnh của video, trong đó quy tắc chỉ ra rằng công cụ LMCS được sử dụng đối với lát liên kết với tiêu đề lát của ảnh video khi phân tử cú pháp thứ ba được bao gồm theo cách có lựa chọn trong dòng bit chỉ báo rằng công cụ LMCS là được phép ở bậc tiêu đề lát của video, và trong đó quy tắc chỉ ra rằng công cụ LMCS không được sử dụng đối với lát hiện tại khi phân tử cú pháp thứ ba chỉ báo rằng công cụ LMCS là không được phép ở bậc tiêu đề lát của video

FIG. 15

1600



Thực hiện sự biến đổi giữa video bao gồm một hoặc nhiều ảnh video và dòng bit của video theo quy tắc, trong đó quy tắc chỉ ra rằng việc liệu độ phân giải hiệu vector chuyển động thích ứng (AMVR) có được sử dụng trong tạo mã vector chuyển động của chế độ ngoại ảnh afin dựa trên phần tử cú pháp được bao gồm theo cách có lựa chọn trong tập thông số dây (SPS) được tham chiếu mà chỉ báo việc liệu AMVR có được phép, trong đó quy tắc chỉ ra rằng AMVR không được sử dụng trong tạo mã vector chuyển động của chế độ ngoại ảnh afin khi phần tử cú pháp chỉ báo rằng AMVR là không được phép, và trong đó quy tắc chỉ ra rằng AMVR được suy ra là không được sử dụng trong tạo mã vector chuyển động của chế độ ngoại ảnh afin khi phần tử cú pháp khi phần tử cú pháp không được bao gồm trong SPS

1602

FIG. 16

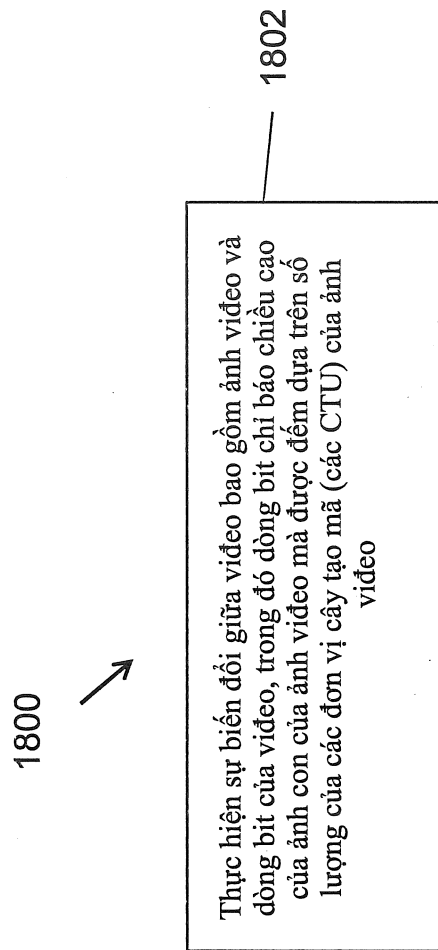
1700



1702

Thực hiện sự biến đổi giữa video bao gồm ảnh video và dòng bit của video theo quy tắc, trong đó ảnh video bao gồm ảnh con, ô ảnh, và lát, và trong đó quy tắc chỉ ra rằng, do ảnh con bao gồm lát mà được phân vùng từ ô ảnh, sự biến đổi được thực hiện bằng cách không đếm chiều cao của ảnh con bằng cách sử dụng số lượng các ô ảnh của ảnh video

FIG. 17

**FIG. 18**

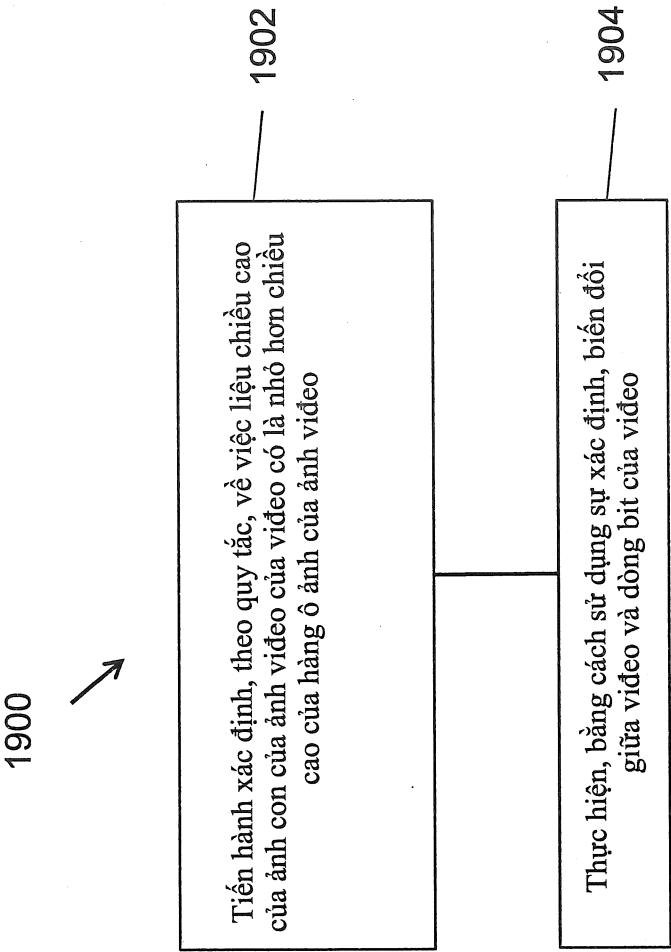


FIG. 19