



(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)



1-0053608

(51)^{2020.01} H04N 19/157; H04N 19/169; H04N 19/174; H04N 19/64; H04N 19/186; H04N 19/46; H04N 19/503; H04N 19/119; H04N 19/176 (13) B

(21) 1-2022-01470

(22) 02/09/2020

(86) PCT/CN2020/112974 02/09/2020

(87) WO2021/043157 11/03/2021

(30) PCT/CN2019/103959 02/09/2019 CN

(45) 25/11/2025 452

(43) 25/07/2022 412A

(73) 1. BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (CN)

Room B-0035, 2/F, No.3 Building No. 30, Shixing Road, Shijingshan District Beijing 100041, China

2. BYTEDANCE INC. (US)

12655 West Jefferson Boulevard Sixth Floor, Suite No. 137 Los Angeles, California 90066, United States of America

(72) XU, Jizheng (CN); DENG, Zhipin (VN); ZHANG, Li (CN); LIU, Hongbin (CN); ZHANG, Kai (CN).

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

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DỮ LIỆU VIDEO

(21) 1-2022-01470

(57) Sáng chế liên quan đến phương pháp xử lý video. Phương pháp này bao gồm các bước: xác định, để chuyển đổi giữa vùng video của video và phân biểu diễn đã được mã hóa của video này, đặc tính mã hóa trong của vùng video này dựa trên định dạng màu của video này theo quy tắc; và thực hiện việc chuyển đổi theo đặc tính mã hóa trong này.

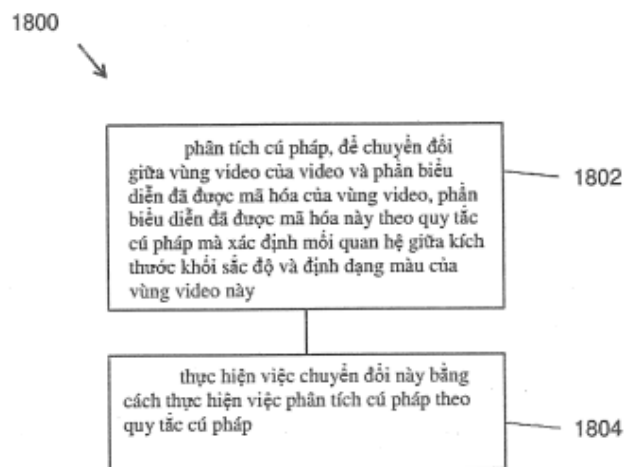


FIG. 18

Lĩnh vực kỹ thuật được đề cập

Sáng chế liên quan đến công nghệ mã hóa và giải mã video và hình ảnh.

Tình trạng kỹ thuật của sáng chế

Video kỹ thuật số chiếm băng thông lớn nhất để sử dụng trên internet và các mạng truyền thông kỹ thuật số khác. Khi số lượng thiết bị người dùng được kết nối có khả năng nhận và hiển thị video gia tăng, dự kiến nhu cầu băng thông đối với việc sử dụng video kỹ thuật số sẽ tiếp tục gia tăng.

Bản chất kỹ thuật của sáng chế

Các kỹ thuật đã biết có thể được sử dụng bởi bộ giải mã video hoặc hình ảnh hoặc bộ mã hóa theo các phương án trong đó các hình ảnh tham chiếu được sử dụng trong việc mã hóa hoặc giải mã video.

Theo một khía cạnh làm ví dụ, một phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước xác định, để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, đặc tính mã hóa trong của vùng video này dựa trên định dạng màu của video này theo quy tắc; và thực hiện việc chuyển đổi theo đặc tính mã hóa trong này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng và trong đó quy tắc định dạng này quy định phần tử cú pháp, modeType, biểu thị chế độ mã hóa của khối video này, ngang bằng với `MODE_TYPE_NO_INTER` mà hạn chế việc sử dụng chế độ mã hóa liên đối với việc chuyển đổi hoặc `MODE_TYPE_NO_INTRA` mà hạn chế việc sử dụng chế độ trong đối với việc chuyển đổi này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa video này và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng quy định rằng không có cờ chỉ ra rằng buộc chế

độ dự báo trong phần biểu diễn đã được mã hóa này trong trường hợp định dạng sắc độ của video này là 4: 2: 2, 4: 0: 0 hoặc 4: 4: 4.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định việc chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, xem liệu và/hoặc cách ràng buộc về kích thước của khối dự báo sắc độ nhỏ nhất trong vùng video này có được kích hoạt theo quy tắc hay không; và thực hiện việc chuyển đổi dựa trên việc xác định này, trong đó quy tắc này phụ thuộc vào việc định dạng màu của video này là 4: 2: 0 hay 4: 2: 2.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định, để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, xem liệu giới hạn về kích thước của khối dự báo trong sắc độ nhỏ nhất đối với vùng video này có được kích hoạt theo quy tắc hay không; và thực hiện việc chuyển đổi dựa trên việc xác định này, trong đó quy tắc này phụ thuộc vào định dạng màu của video này và/hoặc chiều rộng (M) và chiều cao (N) của vùng video này và trong đó quy tắc này quy định thêm rằng, đối với vùng video này là nút cây mã hóa với việc tách BT(cây nhị phân), khi đó, hạn chế về khối dự báo trong sắc độ nhỏ nhất sẽ bị vô hiệu hóa trong trường hợp 1) định dạng màu của video này là 4: 2: 2 và 2) tích của M và N là một trị số từ một tập hợp trị số, trong đó tập hợp trị số này bao gồm cả 64.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này theo một ràng buộc về kích thước khối dự báo trong sắc độ nhỏ nhất, trong đó phần biểu diễn đã được mã hóa tuân theo quy tắc định dạng quy định trị số của trường cú pháp trong phần biểu diễn đã được mã hóa này, do định dạng màu 4: 2: 2 của video này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, khả năng áp dụng sơ đồ phân vùng đối với khối video hiện tại theo quy tắc; và thực hiện việc chuyển đổi dựa trên việc xác định này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định việc chuyển đổi giữa khối video của video và phần biểu diễn đã được mã hóa của video này, xem liệu chế độ liên có được kích hoạt trong trường hợp việc phân vùng cây nhị phân của các mẫu độ chói có được kích hoạt đối với khối video này hay không.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định việc chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, dựa trên quy tắc, xem liệu việc sử dụng chế độ bảng màu có được phép đối với vùng video này hay không ; và thực hiện việc chuyển đổi dựa trên việc xác định này, trong đó chế độ bảng màu có việc mã hóa vùng video này bằng cách sử dụng bảng màu có các trị số mẫu đại diện.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định phần tử cú pháp, modeType, là `MODE_TYPE_IBC` mà cho phép sử dụng chế độ sao chép trong khối đối với việc chuyển đổi hoặc `MODE_TYPE_PALETTE` mà cho phép sử dụng chế độ bảng màu để chuyển đổi, trong đó chế độ sao chép trong khối bao gồm việc mã hóa khối video hiện tại bằng cách sử dụng ít nhất một vector khối trỏ đến khung video chứa khối video hiện tại này và trong đó chế độ bảng màu có việc mã hóa khối video hiện tại này bằng cách sử dụng bảng màu có các trị số mẫu đại diện.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước xác định việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, xem liệu một sơ đồ phân vùng nhất định có được phép đối với khối video hiện tại này hay không theo quy tắc phụ thuộc vào loại chế độ mã hóa dùng để biểu diễn khối video hiện tại đã được mã hóa và kích thước của khối video này ; và thực hiện việc chuyển đổi dựa trên việc xác định này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa khối video của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định rằng một đặc tính của khối video này sẽ kiểm soát xem phần tử cú pháp trong phần biểu diễn đã được mã hóa chỉ ra chế độ dự báo của khối video này.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa vùng video của thành phần thứ nhất của video này và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định xem liệu và/hoặc cách một trường cú pháp có được tạo cấu hình trong phần biểu diễn đã được mã hóa để chỉ ra một tham số lượng tử hóa vi sai đối với vùng video phụ thuộc vào sơ đồ tách dùng để tách các mẫu của thành phần thứ nhất hay không.

Theo một khía cạnh làm ví dụ khác, một phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện việc chuyển đổi giữa vùng video của thành phần thứ nhất của video này và phần biểu diễn đã được mã hóa của video này theo quy tắc, trong đó quy tắc này quy định, trong trường hợp cấu trúc mã hóa cây nhị phân và/hoặc cấu trúc mã hóa cây nhị phân cục bộ được sử dụng đối với vùng video, rằng một biến liên quan đến tham số lượng tử hóa vi sai của thành phần thứ nhất không bị sửa đổi trong khi giải mã hoặc phân tích cú pháp của thành phần thứ hai của video này.

Theo một khía cạnh làm ví dụ khác, phương pháp đã được mô tả trên đây có thể được thực hiện bởi một thiết bị mã hóa video này có bộ xử lý.

Theo một khía cạnh làm ví dụ khác, phương pháp đã được mô tả trên đây có thể được thực hiện bởi một thiết bị giải mã video này có bộ xử lý.

Theo một khía cạnh làm ví dụ khác, các phương pháp này có thể được thể hiện dưới dạng hướng dẫn thực hiện của bộ xử lý và được lưu trữ trên môi trường chương trình máy tính đọc được.

Các khía cạnh này và các khía cạnh khác được mô tả thêm trong Bản mô tả sáng chế này.

Mô tả vắn tắt các hình vẽ

FIG.1 thể hiện một ví dụ về công cụ mã hóa sao chép trong khối .

FIG.2 thể hiện một ví dụ về khối đã được mã hóa Trong chế độ bảng màu.

FIG.3 thể hiện một ví dụ về việc sử dụng công cụ dự báo bảng màu để báo hiệu các mục nhập của bảng màu.

FIG.4 thể hiện một ví dụ về các ví dụ về việc quét ngang và việc quét ngang dọc.

FIG.5 thể hiện các ví dụ về việc mã hóa các chỉ số của bảng màu.

FIG.6 thể hiện một ví dụ về 67 chế độ dự báo trong .

FIG.7 thể hiện các ví dụ về các khối lân cận bên trái và bên trên của khối hiện tại.

FIG.8 thể hiện các ví dụ về hình dạng của bộ lọc ALF (sắc độ: hình thoi 5×5 , độ chói: hình thoi 7×7).

FIG.9 thể hiện một ví dụ về phép tính Laplac đã được lấy mẫu con.

FIG.10 thể hiện một ví dụ về việc phân loại khối đã được sửa đổi trên các ranh giới ảo.

FIG.11 là một ví dụ minh họa về việc lọc ALF đã được sửa đổi đối với thành phần độ chói trên các ranh giới ảo.

FIG.12 thể hiện các ví dụ về bốn mẫu 1-D 3-pixel để phân loại pixel trong EO.

FIG.13 thể hiện bốn dải được nhóm lại với nhau và được đại diện bởi vị trí ban đầu của nó.

FIG.14 thể hiện khối lân cận trên cùng và bên trái dùng trong việc tính trọng số CIIP.

FIG.15 thể hiện việc ánh xạ độ chói với kiến trúc định tỷ lệ sắc độ.

FIG.16 thể hiện các ví dụ về SCIPU.

FIG.17 là một sơ đồ khối của một ví dụ về nền tảng phần cứng dùng để thực hiện các kỹ thuật được mô tả trong Bản mô tả sáng chế này.

FIG.18 là một sơ đồ làm ví dụ của phương pháp xử lý video.

FIG.19 thể hiện các ví dụ về vị trí của các ứng viên hợp nhất không gian.

FIG. 20 thể hiện các ví dụ về các cặp ứng viên được xem xét để kiểm tra dư thừa của các ứng viên hợp nhất không gian.

FIG. 21A đến FIG.21G là lưu đồ của các phương pháp làm ví dụ xử lý video.

Mô tả chi tiết sáng chế

Sáng chế đề xuất các kỹ thuật khác nhau mà có thể được sử dụng bởi bộ giải mã dòng bit hình ảnh hoặc video để cải thiện chất lượng của video này hoặc hình ảnh kỹ thuật số đã được giải nén hoặc đã được giải mã. Nói một cách ngắn gọn, thuật ngữ “video” được sử dụng ở đây để có cả một chuỗi các hình ảnh (theo cách gọi truyền thống là video) và các hình ảnh riêng lẻ. Hơn thế nữa, một bộ mã hóa video cũng có thể thực hiện các kỹ thuật này trong khi mã hóa để tạo lại các khung đã được giải mã dùng để mã hóa thêm.

Tiêu đề phần được sử dụng trong Bản mô tả sáng chế này để dễ hiểu và không giới hạn phạm vi của sáng chế theo các phương án và kỹ thuật nêu trong các phần tương ứng. Do đó, sáng chế theo các phương án từ một phần này có thể được kết hợp với sáng chế theo các phương án từ các phần khác.

1. Tóm tắt

Sáng chế liên quan đến công nghệ mã hóa video. Cụ thể, nó liên quan đến việc mã hóa bảng màu với việc sử dụng việc biểu diễn dựa trên màu cơ bản trong việc mã hóa video. Nó có thể được áp dụng cho tiêu chuẩn mã hóa video hiện có như HEVC hoặc tiêu chuẩn (Mã hóa video đa năng) cần được hoàn thiện. Nó cũng có thể được áp dụng đối với các tiêu chuẩn mã hóa video hoặc codec video trong tương lai.

2. Bàn luận ban đầu

Các tiêu chuẩn mã hóa video đã phát triển chủ yếu thông qua sự phát triển của các tiêu chuẩn ITU-T và ISO/IEC nổi tiếng. ITU-T sản xuất H.261 và H.263, ISO/IEC sản xuất MPEG-1 và MPEG-4 Visual và hai tổ chức cùng sản xuất H.262 / MPEG-2 Video này và H.264 /MPEG-4 Advanced Video Coding (AVC) và H.265 / HEVC. Kể từ H.262, các tiêu chuẩn mã hóa video này dựa trên cấu trúc mã hóa video hỗn hợp, trong đó dự báo thời gian cộng với việc mã hóa biến đổi được dùng. Để khám phá các công nghệ mã hóa video trong tương lai ngoài HEVC, Nhóm thăm dò video chung (Joint Video Exploration Team - JVET) được thành lập bởi VCEG

và MPEG cùng nhau vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được JVET áp dụng và đưa vào phần mềm tham chiếu có tên Mô hình thăm dò chung (Joint Exploration Model -JEM). Vào tháng 4 năm 2018, Nhóm chuyên gia video chung (JVET) giữa VCEG (Q6 /16) và ISO/IEC JTC1 SC29 / WG11 (MPEG) đã được thành lập để làm việc trên tiêu chuẩn VVC nhằm mục tiêu với tốc độ bit giảm 50% so với HEVC.

2.1 Bản sao trong khối

Bản sao trong khối (IBC), hay còn gọi là tham chiếu hình ảnh hiện tại, đã được áp dụng trong phần mở rộng Mã hóa nội dung màn hình HEVC (HEVC-SCC) và mô hình thử nghiệm VVC hiện tại (VTM-4.0). IBC mở rộng khái niệm bù chuyển động từ việc mã hóa liên khung sang việc mã hóa trong khung. Như được minh họa trên FIG.1, khối hiện tại được dự báo bởi khối tham chiếu trong cùng một hình ảnh khi IBC được áp dụng. Các mẫu trong khối tham chiếu phải được tái tạo trước khi khối hiện tại được mã hóa hoặc giải mã. Mặc dù IBC không quá hiệu quả đối với hầu hết các cảnh quay được bởi camera, nhưng nó thể hiện lợi ích mã hóa đáng kể đối với nội dung màn hình. Lý do là có rất nhiều mẫu lặp lại, như biểu tượng và ký tự văn bản trên hình ảnh nội dung màn hình IBC có thể loại bỏ phần dư thừa giữa các mẫu lặp lại này một cách có hiệu quả. Trong HEVC-SCC, một đơn vị mã hóa liên mã (CU) có thể áp dụng IBC nếu nó chọn hình ảnh hiện tại làm hình ảnh tham chiếu của nó. MV được đổi tên thành vector khối (BV) trong trường hợp này và BV luôn có độ chính xác pixel số nguyên. Để tương thích với HEVC cấu hình chính, ảnh hiện tại được đánh dấu là ảnh tham chiếu “dài hạn” trong Bộ đệm ảnh đã được giải mã (Decoded Picture Buffer-DPB). Cần lưu ý rằng theo cách tương tự, trong các tiêu chuẩn mã hóa nhiều chế độ xem/video 3D, ảnh tham chiếu giữa các chế độ xem cũng được đánh dấu là ảnh tham chiếu “dài hạn”.

Sau một BV để tìm khối tham chiếu của nó, dự báo có thể được tạo bằng cách sao chép khối tham chiếu. Phần dư có thể được lấy bằng cách lấy các tín hiệu ban đầu trừ đi các pixel tham chiếu. Sau đó, việc biến đổi và việc lượng tử hóa có thể được áp dụng như trong các chế độ mã hóa khác.

FIG.1 là hình vẽ minh họa Bản sao trong khối .

Tuy nhiên, khi khối tham chiếu nằm bên ngoài hình ảnh, hoặc chồng lên khối hiện tại, hoặc bên ngoài vùng đã được tái tạo, hoặc bên ngoài vùng hợp lệ bị hạn chế bởi một số ràng buộc, một phần hoặc tất cả các trị số pixel sẽ không được xác định. Về cơ bản, có hai giải pháp để xử lý một vấn đề như vậy. Một là không cho phép một tình huống như vậy, ví dụ, trong tuân thủ dòng bit. Cách khác là áp dụng đệm đối với các trị số pixel không xác định đó. Các phần phụ sau mô tả chi tiết các giải pháp đó.

2.2 IBC trong phần mở rộng mã hóa nội dung màn hình HEVC

Trong phần mở rộng mã hóa nội dung màn hình của HEVC, khi một khối sử dụng hình ảnh hiện tại làm khối tham chiếu, phải đảm bảo rằng toàn bộ khối tham chiếu này nằm trong vùng đã được tái tạo sẵn có, như được chỉ ra trong văn bản đặc tả sau:

Các biến offsetX và offsetY được tạo ra như sau:

$$\text{offsetX} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[0] \& 0x7 ? 2 : 0) \quad (8-104)$$

$$\text{offsetY} = (\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[1] \& 0x7 ? 2 : 0) \quad (8-105)$$

Yêu cầu về sự tuân thủ dòng bit là khi hình ảnh tham chiếu là hình ảnh hiện tại, vectơ chuyển động độ chói mvLX sẽ tuân theo các ràng buộc sau:

- Khi quy trình tạo ra tính khả dụng của khối lệnh quét z như được quy định trong điều 6.4.1 được gọi ra (xCurr, yCurr) được đặt ngang bằng với (xCb, yCb) và vị trí độ chói lân cận (xNbY, yNbY) được đặt ngang bằng với (xPb + (mvLX[0] >> 2) - offsetX, yPb + (mvLX[1] >> 2) - offsetY) làm đầu vào, đầu ra sẽ ngang bằng với TRUE.
- Khi quy trình tạo ra tính khả dụng của khối lệnh quét z như được quy định trong mục 6.4.1 được gọi ra (xCurr, yCurr) được đặt ngang bằng với (xCb, yCb) và vị trí độ chói lân cận (xNbY, yNbY) được đặt ngang bằng với (xPb + (mvLX[0] >> 2) + nPbW - 1 + offsetX, yPb + (mvLX[1] >> 2) + nPbH - 1 + offsetY) làm đầu vào, đầu ra sẽ ngang bằng với TRUE.
- Một hoặc cả hai điều kiện sau đều đúng::
 - Trị số của (mvLX[0] >> 2) + nPbW + xBI + offsetX nhỏ hơn hoặc bằng 0.

- Trị số của $(mvLX[1] \gg 2) + nPbH + Yb1 + offsetY$ nhỏ hơn hoặc bằng 0.

- Điều kiện sau phải đúng:

$$(xPb + (mvLX[0] \gg 2) + nPbSw - 1 + offsetX) / CtbSizeY - xCb / CtbSizeY$$

$\leq$

$$yCb / CtbSizeY - (yPb + (mvLX[1] \gg 2) + nPbSh - 1 + offsetY) / CtbSizeY$$

(8-106)

Như vậy, trường hợp khối tham chiếu chồng lên khối hiện tại hoặc khối tham chiếu nằm ngoài hình ảnh sẽ không xảy ra. Không cần phải đệm khối tham chiếu hoặc dự báo.

2.3 IBC ở Mô hình thử nghiệm VVC

Ở Mô hình thử nghiệm VVC hiện tại, tức là thiết kế VTM-4.0, toàn bộ khối tham chiếu phải bằng đơn vị cây mã hóa hiện tại (CTU) và không chồng lên khối hiện tại. Do đó, không cần phải đệm khối tham chiếu hoặc khối dự báo. Cờ IBC được mã hóa như một chế độ dự báo của CU hiện tại. Do đó, có tổng cộng ba chế độ dự báo, MODE_INTRA, MODE_INTER và MODE_IBC đối với mỗi CU.

2.3.1 Chế độ hợp nhất IBC

Trong chế độ hợp nhất IBC, một chỉ mục trỏ đến mục nhập trong danh mục ứng viên hợp nhất IBC được phân tích cú pháp từ dòng bit. Việc xây dựng danh mục hợp nhất IBC có thể được tóm tắt theo trình tự các bước sau:

- Bước 1: Tạo ra các ứng viên không gian
- Bước 2: Chèn các ứng viên HMVP vào
- Bước 3: Chèn từng đôi ứng viên trung bình vào

Trong việc tạo ra các ứng viên hợp nhất không gian, tối đa bốn ứng viên hợp nhất được chọn trong số các ứng viên nằm ở các vị trí được thể hiện trên Fig.19. Thứ tự của đạo hàm là A1, B1, B0, A0 và B2. Vị trí B2 chỉ được xem xét khi không có PU bất kỳ của vị trí A1, B1, B0, A0 (ví dụ, vì nó thuộc về lát cắt hoặc lớp khác) hoặc không được mã hóa là chế độ IBC. Sau khi ứng viên ở vị trí A1 được thêm vào, việc chèn các ứng viên còn lại cần được kiểm tra dư thừa để đảm bảo rằng các ứng viên có cùng thông tin chuyển động sẽ bị loại ra khỏi danh mục để hiệu quả mã hóa được cải thiện. Để giảm độ phức tạp tính toán, không phải tất cả các cặp ứng viên khả thi đều được xem xét trong kiểm tra dư thừa đã nêu. Thay vào đó, chỉ các cặp được liên

kết với một mũi tên được thể hiện trên Fig.20 ứng viên được xem xét và một ứng viên chỉ được thêm vào danh mục nếu ứng viên tương ứng được sử dụng để kiểm tra dư thừa có thông tin chuyển động không giống nhau.

Sau khi chèn các ứng viên không gian, nếu kích thước danh mục hợp nhất IBC vẫn nhỏ hơn kích thước danh mục hợp nhất IBC tối đa, thì ứng viên IBC từ bảng HMVP có thể được chèn vào. Kiểm tra dư thừa được thực hiện khi chèn các ứng viên HMVP.

Cuối cùng, từng đôi ứng viên trung bình được đưa vào danh mục hợp nhất IBC.

Khi khối tham chiếu được xác định bởi một ứng viên hợp nhất nằm bên ngoài hình ảnh hoặc chồng lên khối hiện tại, hoặc bên ngoài vùng đã được tái tạo hoặc bên ngoài vùng hợp lệ bị hạn chế bởi một số ràng buộc, ứng viên hợp nhất được gọi là ứng viên hợp nhất không hợp lệ.

Cần lưu ý rằng các ứng viên hợp nhất không hợp lệ có thể được chèn vào danh mục hợp nhất IBC.

2.3.2 Chế độ IBC AMVP

Trong chế độ IBC AMVP, chỉ mục AMVP trỏ đến mục nhập trong danh mục IBC AMVP được phân tích cú pháp từ dòng bit. Việc xây dựng danh mục IBC AMVP có thể được tóm tắt theo trình tự các bước sau:

- Bước 1: Tạo ra các ứng viên không gian
 - o Kiểm tra A0, A1 cho đến khi tìm thấy ứng viên khả dụng.
 - o Kiểm tra B0, B1, B2 cho đến khi tìm thấy ứng viên khả dụng.
- Bước 2: Chèn các ứng viên HMVP vào
- Bước 3: Chèn ứng viên zero vào

Sau khi chèn các ứng viên không gian vào, nếu kích thước danh mục IBC AMVP vẫn nhỏ hơn kích thước danh mục IBC AMVP tối đa, thì ứng viên IBC từ bảng HMVP có thể được chèn vào.

Cuối cùng, các ứng viên zero được chèn vào danh mục IBC AMVP.

2.4 Chế độ bảng màu

Ý tưởng cơ bản đằng sau chế độ bảng màu là các mẫu trong CU được biểu thị bằng một tập hợp nhỏ các trị số màu đại diện. Tập hợp này được gọi là bảng màu. Và

cũng có thể chỉ ra một mẫu nằm ngoài bảng màu này bằng cách báo hiệu một ký hiệu thoát theo sau là các trị số thành phần (có thể được lượng tử hóa). Loại mẫu này được gọi là mẫu thoát. Chế độ bảng màu được minh họa trên FIG.2.

FIG.2 thể hiện một ví dụ về khối đã được mã hóa trong chế độ bảng màu.

2.5 Chế độ bảng màu trong phần mở rộng Mã hóa nội dung màn hình HEVC (HEVC-SCC)

Trong chế độ bảng màu trong HEVC-SCC, một cách tiên đoán được sử dụng để mã hóa bảng màu và bản đồ chỉ mục.

2.5.1 Mã hóa các mục nhập trong bảng màu

Để mã hóa các mục nhập của bảng màu, một công cụ dự báo bảng màu được duy trì. Kích thước tối đa của bảng màu cũng như công cụ dự báo bảng màu được báo hiệu trong SPS. Trong HEVC-SCC, một `Palette_predictor_initializer_present_flag` (bộ khởi tạo dự báo bảng màu thể hiện cờ) được đưa vào trong PPS. Khi cờ này là 1, các mục nhập để khởi tạo bộ dự báo bảng màu được báo hiệu trong dòng bit. Bộ dự báo bảng màu này được khởi tạo ở đầu mỗi hàng CTU, mỗi lát cắt và mỗi lớp. Tùy thuộc vào trị số của `Palette_predictor_initializer_present_flag`, trình dự báo bảng màu được đặt lại bằng 0 hoặc được khởi tạo bằng cách sử dụng các mục nhập của bộ khởi tạo của trình dự báo bảng màu được báo hiệu trong PPS. Trong HEVC-SCC, bộ khởi tạo bộ dự báo bảng màu có kích thước 0 đã được kích hoạt để cho phép vô hiệu hóa rõ ràng việc khởi tạo bộ dự báo bảng màu ở cấp PPS.

Đối với mỗi mục nhập trong công cụ dự báo bảng màu, một cờ tái sử dụng được báo hiệu để chỉ ra xem liệu nó có phải là một phần của bảng màu hiện tại này hay không. Điều đó được minh họa trên FIG.3. Các cờ tái sử dụng được gửi bằng cách sử dụng mã số 0 trong thời gian chạy. Sau đó, số lượng mục nhập của bảng màu mới được báo hiệu bằng cách sử dụng mã Golomb theo hàm số mũ của bậc 0. Cuối cùng, các trị số thành phần đối với các mục nhập của bảng màu mới được báo hiệu.

FIG.3 thể hiện một ví dụ về việc sử dụng công cụ dự báo bảng màu để báo hiệu các mục nhập của bảng màu.

2.5.2 Mã hóa các chỉ số của bảng màu

Các chỉ số của bảng màu được mã hóa bằng cách sử dụng việc quét ngang và việc quét dọc ngang như được thể hiện trên FIG.4. Thứ tự quét được báo hiệu rõ ràng trong dòng bit bằng cách sử dụng `Palette_transpose_flag`. Đối với phần còn lại của tiêu mục, giả định rằng việc quét này là việc quét ngang.

FIG.4 thể hiện các ví dụ về việc quét ngang và việc quét dọc ngang.

Các chỉ số của bảng màu được mã hóa bằng cách sử dụng hai chế độ mẫu bảng màu chính: 'INDEX' và 'COPY_ABOVE'. Như đã giải thích trên đây, biểu tượng thoát cũng được báo hiệu là chế độ 'INDEX' và được gán một chỉ mục bằng kích thước bảng màu tối đa. Chế độ được báo hiệu bằng cờ ngoại trừ hàng trên cùng hoặc khi chế độ trước đó là 'COPY_ABOVE'. Trong chế độ 'COPY_ABOVE', chỉ số của bảng màu của mẫu trong hàng trên được sao chép. Trong chế độ 'INDEX', chỉ mục bảng màu được báo hiệu rõ ràng. Đối với cả chế độ 'INDEX' và 'COPY_ABOVE', một trị số chạy được báo hiệu mà quy định số lượng mẫu tiếp theo cũng được mã hóa bằng cách sử dụng cùng một chế độ. Khi biểu tượng thoát là một phần của quy trình chạy ở chế độ 'INDEX' hoặc 'COPY_ABOVE', các trị số thành phần thoát được báo hiệu đối với mỗi biểu tượng thoát. Việc mã hóa các chỉ số của bảng màu được minh họa trên FIG.5.

Thứ tự cú pháp này được thực hiện như sau. Đầu tiên, số lượng trị số chỉ mục đối với CU được báo hiệu. Tiếp theo là báo hiệu các trị số chỉ mục thực tế đối với toàn bộ CU bằng cách sử dụng mã nhị phân cắt ngắn. Cả số lượng chỉ số cũng như trị số chỉ số đều được mã hóa ở chế độ bỏ qua. Điều đó nhóm các thùng bỏ qua liên quan đến chỉ mục lại với nhau. Sau đó, chế độ mẫu bảng màu (nếu cần) và chạy được báo hiệu theo cách đan xen. Cuối cùng, các trị số thoát thành phần tương ứng với các mẫu thoát đối với toàn bộ CU được nhóm lại với nhau và được mã hóa ở chế độ bỏ qua.

Một phần tử cú pháp bổ sung, `last_run_type_flag`, được báo hiệu sau khi báo hiệu các trị số chỉ mục. Phần tử cú pháp này, cùng với số lượng chỉ số, loại bỏ nhu cầu báo hiệu trị số chạy tương ứng với lần chạy cuối cùng trong khối này.

Trong HEVC-SCC, chế độ bảng màu cũng được kích hoạt đối với các định dạng 4: 2: 2, 4: 2: 0 và đơn sắc. Tín hiệu của các mục trong bảng màu và chỉ số của bảng màu gần như giống hệt nhau đối với tất cả các định dạng sắc độ. Trong trường

hợp định dạng không đơn sắc, mỗi mục nhập của bảng màu có 3 thành phần. Đối với định dạng đơn sắc, mỗi mục nhập của bảng màu có một thành phần duy nhất. Đối với các hướng sắc độ được lấy mẫu con, các mẫu sắc độ được liên kết với các chỉ số mẫu sắc độ chia hết cho 2. Sau khi tạo lại các chỉ số của bảng màu đối với CU, nếu một mẫu chỉ có một thành phần duy nhất được liên kết với nó, thì chỉ thành phần thứ nhất của mục nhập của bảng màu được sử dụng. Sự khác biệt duy nhất trong báo hiệu là đối với các trị số thành phần thoát. Đối với mỗi mẫu thoát, số lượng trị số thành phần thoát được báo hiệu có thể là khác nhau tùy thuộc vào số lượng thành phần liên kết với mẫu đó.

Trong VVC, cấu trúc mã hóa cây nhị phân được sử dụng để mã hóa trong các lát cắt, vì vậy thành phần độ chói và hai thành phần sắc độ có thể có bảng màu và chỉ số của bảng màu khác nhau. Ngoài ra, hai thành phần sắc độ dùng chung cùng một bảng màu và các chỉ số của bảng màu.

FIG.5 thể hiện các ví dụ về việc mã hóa các chỉ số của bảng màu.

2.6 Mã hóa chế độ trong VVC

Để bắt giữ các hướng cạnh tùy ý được biểu diễn trong video tự nhiên, số lượng chế độ định hướng trong VTM5 được mở rộng từ 33, như được sử dụng trong HEVC, lên 65. Các chế độ hướng mới không có trong HEVC được mô tả dưới dạng mũi tên chấm chấm màu đỏ trên FIG.6 và các chế độ phẳng và DC vẫn như cũ. Các chế độ dự báo trong theo hướng dày đặc hơn này áp dụng cho tất cả các kích thước khối và đối với cả dự báo trong độ chói và sắc độ.

Trong VTM5, một vài chế độ dự báo trong góc thông thường được thay thế thích ứng bằng các chế độ dự báo trong góc rộng đối với các khối không vuông.

Trong HEVC, mỗi khối được mã hóa trong đều có dạng hình vuông và chiều dài của mỗi cạnh của nó là lũy thừa của 2. Do đó, không cần việc tách nào để tạo ra bộ dự báo trong là chế độ DC. Trong VTM5, các khối có thể có dạng hình chữ nhật mà cần phải sử dụng phép chia đối với mỗi khối trong trường hợp chung. Để tránh các phép chia đối với việc dự báo DC, chỉ cạnh dài hơn được sử dụng để tính trị số trung bình đối với các khối không vuông.

FIG.6 thể hiện một ví dụ về 67 chế độ dự báo trong .

Để giữ đối với độ phức tạp của việc tạo danh mục chế độ có thể xảy ra nhất (MPM) ở mức thấp, phương pháp mã hóa chế độ trong với 6 MPM được sử dụng bằng cách xem xét hai chế độ trong lân cận khả dụng. Ba khía cạnh sau được coi là để xây dựng danh mục MPM:

- Chế độ trong mặc định
- Chế độ trong lân cận
- Chế độ trong đã được tạo ra

Danh mục 6 MPM thống nhất được sử dụng đối với các khối trong bất kể công cụ mã hóa MRL và ISP có được áp dụng hay không. Danh mục MPM được xây dựng dựa trên các chế độ trong của khối lân cận bên trái và bên trên. Giả sử chế độ của khối bên trái được ký hiệu là Bên trái và chế độ của khối bên trên được ký hiệu là Bên trên, danh mục MPM thống nhất được xây dựng như sau (Các khối bên trái và bên trên được thể hiện trên FIG.7):

FIG.7 là một ví dụ về các khối lân cận bên trái và bên trên của khối hiện tại.

- Khi khối lân cận không khả dụng, chế độ trong của khối đó được đặt ngang bằng với Planar(phẳng) theo mặc định.
- Nếu cả chế độ Bên trái và Bên trên là chế độ không góc:
 - o Danh mục MPM $\rightarrow \{ \text{Planar, DC, V, H, V-4, V+4} \}$
- Nếu một trong số các chế độ Bên trái và Bên trên là chế độ góc và chế độ còn lại là không góc:
 - o Đặt chế độ Tối đa làm chế độ lớn hơn ở Bên trái và Bên trên
 - o Danh mục MPM $\rightarrow \{ \text{Planar, Max, DC, Max-1, Max+1, Max-2} \}$
 - Nếu Bên trái và Bên trên đều có góc cạnh và chúng khác nhau:
 - o Đặt chế độ Max làm chế độ lớn hơn ở Bên trái và Bên trên
 - o nếu sự khác biệt của chế độ Bên trái và Bên trên nằm trong khoảng từ 2 đến 62, bao gồm cả 62
 - Danh mục MPM $\rightarrow \{ \text{Phẳng, Bên trái, Bên trên, DC, Tối đa-1, Tối đa+1} \}$
 - o Nếu không
 - Danh mục MPM $\rightarrow \{ \text{Phẳng, Bên trái, Bên trên, DC, Tối đa-2, Tối đa+2} \}$
 - Nếu Bên trái và Bên trên đều là góc và chúng giống nhau:
 - o Danh mục MPM $\rightarrow \{ \text{Phẳng, Bên trái, Bên trái-1, Bên trái+1, DC, Bên trái-2} \}$

Bên cạnh đó, thùng đầu tiên của từ mã chỉ mục mpm được mã hóa ngữ cảnh CABAC. Tổng cộng ba ngữ cảnh được sử dụng, tương ứng với việc khối trong hiện tại là MRL được kích hoạt, ISP được kích hoạt hay khối trong bình thường.

Trong khi tạo danh mục 6 MPM, việc cắt tỉa được sử dụng nhằm loại bỏ các chế độ trùng lặp để chỉ các chế độ duy nhất mới có thể được đưa vào danh mục MPM. Để mã hóa entropy của 61 chế độ không phải MPM, Mã nhị phân cắt ngắn (Truncated Binary Code -TBC) được sử dụng.

Đối với việc mã hóa Trong chế độ sắc độ, tổng cộng 8 chế độ trong được phép để mã hóa chế độ trong sắc độ. Các chế độ đó bao gồm năm chế độ trong truyền thống và ba chế độ mô hình tuyến tính chéo thành phần (CCLM, LM_A và LM_L). Quy trình báo hiệu chế độ và tạo ra sắc độ được thể hiện trong Bảng 2. Việc mã hóa chế độ sắc độ trực tiếp phụ thuộc vào chế độ dự báo trong của khối độ chói tương ứng. Vì cấu trúc phân vùng khối riêng biệt đối với các thành phần độ chói và sắc độ được kích hoạt trong các lát cắt I, khối sắc độ có thể tương ứng với nhiều khối độ chói. Do đó, đối với chế độ sắc độ DM, chế độ dự báo trong của khối độ chói tương ứng bao phủ vị trí trung tâm của khối sắc độ hiện tại được kế thừa trực tiếp.

Bảng 2- 4 - Tạo ra chế độ dự báo sắc độ từ chế độ độ chói khi kích hoạt cclm_ được kích hoạt

Chế độ dự báo sắc độ	Chế độ dự báo trong độ chói tương ứng				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	81	81	81	81	81
5	82	82	82	82	82
6	83	83	83	83	83
7	0	50	18	1	X

2.7 Điều biến mã xung vi sai khối dư đã được lượng tử hóa (QR-BDPCM)

Trong JVET-M0413, việc điều biến mã xung vi sai khối dư đã được lượng tử hóa (QR-BDPCM) được đề xuất để mã hóa nội dung màn hình một cách có hiệu quả.

Các hướng dự báo được sử dụng trong QR-BDPCM có thể là các chế độ dự báo dọc và ngang. Việc dự báo trong được thực hiện trên toàn bộ khối bằng cách sao chép mẫu theo hướng dự báo (dự báo ngang hoặc dọc) giống như việc dự báo trong. Phần dư được lượng tử hóa và delta giữa phần dư đã được lượng tử hóa và trị số lượng tử hóa dự báo (ngang hoặc dọc) của nó được mã hóa. Điều đó có thể được mô tả như sau: Đối với khối kích thước M (hàng) $\times$ N (cột), giả sử $r_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ là phần dư dự báo sau khi thực hiện việc dự báo trong theo chiều ngang (sao chép trị số pixel lân cận bên trái qua đường khối dự báo theo mỗi dòng) hoặc theo chiều dọc (sao chép dòng lân cận trên cùng sang mỗi dòng trong khối dự báo) bằng cách sử dụng các mẫu chưa được lọc từ các mẫu ranh giới khối bên trên hoặc bên trái. Giả sử $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ biểu thị phiên bản lượng tử hóa của phần dư $r_{i,j}$, trong đó phần dư là sự khác biệt giữa khối ban đầu và khối dự báo các trị số khối. Sau đó, khối DPCM được áp dụng đối với các mẫu dư đã được lượng tử hóa, mà dẫn đến mảng $M \times N$ $\tilde{r}$ được sửa đổi với các phần tử $\tilde{r}_{i,j}$. Khi BDPCM đọc được báo hiệu:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (2-7-1)$$

Đối với việc dự báo ngang, các quy tắc tương tự được áp dụng và các mẫu lượng tử còn lại thu được bằng cách

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (2-7-2)$$

Các mẫu lượng tử còn lại $\tilde{r}_{i,j}$ được gửi đến bộ giải mã.

Ở phía bộ giải mã, các tính toán trên được đảo ngược để tạo ra $Q(\tilde{r}_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$. Đối với trường hợp dự báo dọc,

$$Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \quad (2-7-3)$$

Đối với trường hợp dự báo ngang,

$$Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \quad (2-7-4)$$

Các phần dư đã được lượng tử hóa nghịch đảo, $Q^{-1}(Q(r_{i,j}))$, được thêm vào các trị số dự báo trong khối để tạo ra các trị số mẫu đã được tái tạo.

Lợi ích chính của sơ đồ này là DPCM nghịch đảo có thể được thực hiện nhanh chóng trong khi phân tích cú pháp hệ số chỉ cần thêm công cụ dự báo khi các hệ số được phân tích cú pháp hoặc nó có thể được thực hiện sau khi phân tích cú pháp.

2.8 Bộ lọc vòng lặp thích ứng

Trong VTM5, Bộ lọc vòng lặp thích ứng (Adaptive Loop Filter -ALF) với khả năng thích ứng của bộ lọc dựa trên khối được áp dụng. Đối với thành phần độ chói, một trong số 25 bộ lọc được chọn đối với mỗi khối 4×4 , dựa trên hướng và hoạt động của gradient cục bộ.

2.8.1.1 Hình dạng của bộ lọc

Trong VTM5, hai hình dạng của bộ lọc hình thoi (như được thể hiện trên FIG.8) được sử dụng. Hình dạng hình thoi 7×7 được áp dụng đối với thành phần độ chói và hình dạng hình thoi 5×5 được áp dụng đối với các thành phần sắc độ.

FIG.8 thể hiện các ví dụ về hình dạng của bộ lọc ALF (sắc độ: hình thoi 5×5 , độ chói: hình thoi 7×7)

2.8.1.2 Phân loại khối

Đối với thành phần độ chói, mỗi khối 4×4 được phân loại thành một trong số 25 lớp. Chỉ số phân loại C được tạo ra dựa trên hướng của nó D và trị số lượng tử hóa của hoạt động A, là như sau:

$$C = 5D + \hat{A} \quad (2-9-1)$$

Để tính D và $\hat{A}$, gradient của chiều ngang, dọc và hai đường chéo trước tiên được tính ngang bằng với cách sử dụng Laplacian 1-D:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \quad (2-9-2)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \quad (2-1)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (2-9-4)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$$

(2-9-5)

Trong đó các chỉ số i và j được dùng để chỉ tọa độ của mẫu phía trên bên trái trong khối 4×4 và $R(i,j)$ chỉ ra một mẫu đã được tái tạo ở tọa độ (i,j) .

Để giảm độ phức tạp của việc phân loại khối, phép tính Laplacian 1-D được lấy mẫu con được áp dụng. Như được thể hiện trên FIG.9, các vị trí được lấy mẫu con giống nhau được sử dụng để tính toán gradient của tất cả các hướng.

FIG.9 thể hiện một ví dụ về việc tính toán Laplacian được lấy mẫu con. (a) Các vị trí được lấy mẫu đối với gradient dọc (b) Các vị trí được lấy mẫu đối với gradient ngang (c) Các vị trí được lấy mẫu đối với gradient chéo (d) Các vị trí được lấy mẫu đối với gradient chéo.

Sau đó, các trị số lớn nhất và nhỏ nhất của D của các gradient ngang và dọc được đặt là:

$$g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v) \quad (2-9-6)$$

Trị số lớn nhất và nhỏ nhất của gradient theo hai hướng chéo được đặt là:

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}) \quad (2-9-7)$$

Để tạo ra trị số của hướng D , các trị số này được so sánh với nhau và với hai ngưỡng t_1 và t_2 :

Bước 1. Nếu cả $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ and $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ đều đúng, thì D được đặt ngang bằng với 0.

Bước 2. Nếu $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$, tiếp tục từ Bước 3; nếu không, hãy tiếp tục từ Bước 4.

Bước 3. Nếu $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D được đặt ngang bằng với 2; nếu không thì D được đặt ngang bằng với 1.

Bước 4. Nếu $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D được đặt ngang bằng với 4; nếu không thì D được đặt ngang bằng với 3.

Trị số hoạt động A được tính như sau:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \quad (2-9-8)$$

A tiếp tục được lượng tử hóa trong khoảng từ 0 đến 4, và trị số lượng tử hóa này được ký hiệu là $\hat{A}$.

Đối với các thành phần sắc độ trong ảnh, không có phương pháp phân loại nào được áp dụng, tức là một bộ hệ số ALF được áp dụng đối với mỗi thành phần sắc độ.

2.8.1.3 Các phép biến đổi hình học của hệ số bộ lọc và trị số cắt xén

Trước khi lọc mỗi khối sắc độ 4×4 , các phép biến đổi hình học như xoay hoặc lật theo đường chéo và dọc được áp dụng đối với các hệ số bộ lọc $f(k, l)$ và đối với các trị số cắt xén của bộ lọc tương ứng $c(k, l)$ tùy thuộc vào các trị số gradient đã được tính toán đối với khối đó. Điều đó tương đương với việc áp dụng các phép biến đổi này đối với các mẫu trong vùng đỡ bộ lọc. Ý tưởng là tạo ra các khối khác nhau mà ALF được áp dụng giống nhau hơn bằng cách căn chỉnh hướng của chúng.

Ba phép biến đổi hình học, bao gồm đường chéo, lật dọc và xoay được đưa vào:

Đường chéo: $f_D(k, l) = f(l, k)$, $c_D(k, l) = c(l, k)$,

(2-9-9)

Lật dọc: $f_V(k, l) = f(k, K - l - 1)$, $c_V(k, l) = c(k, K - l - 1)$ (2-9-10)

Xoay: $f_R(k, l) = f(K - l - 1, k)$, $c_R(k, l) = c(K - l - 1, k)$ (2-9-11)

trong đó K là kích thước của bộ lọc và $0 \leq k, l \leq K - 1$ là các hệ số tọa độ, sao cho vị trí $(0, 0)$ ở góc trên bên trái và vị trí $(K - 1, K - 1)$ ở góc dưới bên phải. Các phép biến đổi được áp dụng đối với các hệ số bộ lọc $f(k, l)$ và đối với các trị số cắt xén $c(k, l)$ tùy thuộc vào các trị số gradient đã được tính toán đối với khối đó. Mỗi quan hệ giữa phép biến đổi hình học và bốn tung độ của bốn hướng được tóm tắt trong bảng sau

Bảng 2-5 - Ánh xạ của gradient được tính toán được đối với một khối và các phép biến đổi

Trị số gradien	Biến đổi
$g_{d2} < g_{d1}$ and $g_h < g_v$	Không biến đổi
$g_{d2} < g_{d1}$ and $g_v < g_h$	Đường chéo
$g_{d1} < g_{d2}$ and $g_h < g_v$	Lật dọc
$g_{d1} < g_{d2}$ and $g_v < g_h$	Xoay

2.8.1.4 Báo hiệu thông số bộ lọc

Trong VTM5, các thông số bộ lọc ALF được báo hiệu trong Bộ thông số thích ứng (Adaptation Parameter Set -APS). Trong một APS, mà có thể báo hiệu tối đa 25 bộ hệ số bộ lọc luma và chỉ số trị số cắt xén và tối đa một bộ hệ số bộ lọc sắc độ và chỉ số trị số cắt xén. Để giảm chi phí bit, mà có thể hợp nhất các hệ số lọc của các phân loại khác nhau. Trong tiêu đề phần, các chỉ số của APS được sử dụng cho phần hiện tại được báo hiệu.

Chỉ mục trị số cắt xén, mà được giải mã từ các APS, cho phép xác định trị số cắt xén bằng cách sử dụng bảng độ chói và bảng sắc độ của các trị số cắt xén. Các trị số cắt xén này phụ thuộc vào độ sâu bit bên trong. Chính xác hơn, bảng độ chói của các trị số cắt xén và bảng sắc độ của các trị số cắt xén được thu bằng các công thức sau:

$$\text{AlfClip}_L = \left\{ \text{round} \left(2^{\frac{B-n+1}{N}} \right) \text{ đối với } n \in [1..N] \right\},$$

(2-9-12)

$$\text{AlfClip}_C = \left\{ \text{round} \left(2^{(B-8)+8\frac{(N-n)}{N-1}} \right) \text{ đối với } n \in [1..N] \right\}$$

(2-9-13)

với B bằng độ sâu bit bên trong và N bằng 4 là số lượng trị số cắt xén được phép trong VTM5.0.

Quy trình lọc có thể được kiểm soát ở mức CTB. Một cờ luôn được báo hiệu để chỉ ra ALF có được áp dụng cho CTB độ chói hay không. Một CTB độ chói có

thể chọn một bộ lọc trong số 16 bộ lọc cố định và bộ lọc từ các APS. Chỉ số bộ lọc được báo hiệu cho CTB độ chói để chỉ ra bộ lọc nào được sử dụng. 16 bộ lọc cố định được xác định trước và được mã hóa cứng trong cả bộ mã hóa và bộ giải mã.

Hệ số bộ lọc được lượng tử hóa với định mức bằng 128. Để hạn chế độ phức tạp của phép nhân, việc tuân thủ dòng bit được áp dụng sao cho trị số hệ số của vị trí không phải trung tâm sẽ nằm trong khoảng từ -2^7 đến 2^7-1 , bao gồm cả 2^7-1 . Hệ số vị trí trung tâm không được báo hiệu trong dòng bit và được coi là bằng 128.

2.8.1.5 Quy trình lọc

Ở phía bộ giải mã, khi ALF được kích hoạt đối với CTB, mỗi mẫu $R(i, j)$ trong CU cần được lọc, mà tạo ra trị số mẫu $R'(i, j)$ như được thể hiện dưới đây,

$$R'(i, j) = R(i, j) + \left(\left(\sum_{k \neq 0} \sum_{l \neq 0} f(k, l) \times K(R(i+k, j+l) - R(i, j), c(k, l)) + 64 \right) \gg 7 \right) \quad (2-9-14)$$

[001] trong đó $f(k, l)$ biểu thị các hệ số bộ lọc đã được giải mã, $K(x, y)$ là hàm cắt xén và $c(k, l)$ biểu thị các tham số cắt xén đã được giải mã. Biến k và l thay đổi giữa $-\frac{L}{2}$ và $\frac{L}{2}$ trong đó L biểu thị chiều dài bộ lọc. Hàm cắt xén $K(x, y) = \min(y, \max(-y, x))$ tương ứng với hàm $Clip3(-y, y, x)$.

2.8.1.6 Quy trình lọc ranh giới ảo để giảm bộ đệm dòng

Trong VTM5, để giảm yêu cầu về bộ đệm dòng của ALF, phân loại và lọc khối đã được sửa đổi được dùng đối với các mẫu gần ranh giới CTU ngang. Nhằm mục đích này, một ranh giới ảo được xác định là một đường bằng cách dịch chuyển ranh giới CTU theo chiều ngang với các mẫu “N” như được thể hiện trên FIG.10 với N bằng 4 đối với thành phần độ chói và 2 đối với thành phần sắc độ.

FIG.10 thể hiện một ví dụ về việc phân loại khối đã được sửa đổi trên các ranh giới ảo.

Phân loại đã được sửa đổi được áp dụng đối với thành phần độ chói như được thể hiện trên FIG.11. Hoạt động trị số A được định tỷ lệ tương ứng bằng cách tính đến số lượng mẫu đã được giảm dùng trong gradien tính toán Laplacian 1D.

Để xử việc lý lọc, thao tác đệm đối xứng trên các ranh giới ảo được sử dụng đối với cả thành phần độ chói và sắc độ. Như thể hiện trên FIG.11, khi mẫu đang được lọc nằm bên dưới ranh giới ảo, các mẫu lân cận nằm trên ranh giới ảo sẽ được

đệm. Trong khi đó, các mẫu tương ứng ở các cạnh còn lại cũng được đệm, đối xứng nhau.

FIG.11 thể hiện các ví dụ về ALF lọc đã được sửa đổi đối với các thành phần độ chói trên ranh giới ảo.

2.9 Phần bù tương thích mẫu (Sample Adaptive Offset -SAO)

Phần bù tương thích mẫu (SAO) được áp dụng cho tín hiệu đã được tái tạo sau bộ lọc tách khối bằng cách sử dụng các phần bù được bộ mã hóa quy định đối với mỗi CTB. Đầu tiên, bộ mã hóa HM đưa ra quyết định về việc có áp dụng quy trình SAO đối với lát cắt hiện tại hay không. Nếu SAO được áp dụng cho lát cắt, mỗi CTB được phân loại là một trong số năm loại SAO như được thể hiện trong Bảng 2. Khái niệm SAO là phân loại pixel thành các danh mục và giảm sự biến dạng bằng cách thêm phần bù vào các pixel của mỗi danh mục. Thao tác SAO bao gồm phần bù cạnh (EO) mà sử dụng thuộc tính cạnh để phân loại pixel trong SAO loại 1-4 và phần bù dải (BO) mà sử dụng cường độ pixel để phân loại pixel trong SAO loại 5. Mỗi CTB áp dụng có các tham số SAO bao gồm sao_merge_left_flag, sao_merge_up_flag, Loại SAO và bốn phần bù. Nếu sao_merge_left_flag ngang bằng với 1, CTB hiện tại sẽ sử dụng lại loại SAO và phần bù CTB ở bên trái. Nếu sao_merge_up_flag ngang bằng với 1, CTB hiện tại sẽ sử dụng lại loại SAO và các phần bù CTB nêu trên .

Bảng 2- 6 - Đặc điểm kỹ thuật của loại SAO

Phần bù dải SAO	Loại phần bù tương thích mẫu cần được sử dụng	Số lượng loại
0	Không	0
1	Phần bù cạnh kiểu 0-độ 1-D	4
2	Phần bù cạnh kiểu 90-độ 1-D	4
3	Phần bù cạnh kiểu 90-độ 1-D	4
4	Phần bù cạnh kiểu 90-độ 1-D	4
5	Phần bù dải	4

2.9.1 Hoạt động của mỗi loại SAO

Phần bù cạnh sử dụng bốn mẫu 3-pixel 1-D để phân loại pixel hiện tại p bằng cách xem xét thông tin hướng của cạnh, như được thể hiện trên FIG.12. Từ trái sang phải là: 0 độ, 90 độ, 135 độ và 45 độ.

FIG.12 thể hiện các ví dụ về bốn mẫu 3-pixel 1-D để phân loại pixel trong EO.

Mỗi CTB được phân loại thành một trong số năm loại theo Bảng 2-7.

Bảng 2- 7 - Quy tắc phân loại pixel đối với EO lân cận

Loại	Điều kiện	Ý nghĩa
0	Không có cái nào bên dưới	Chủ yếu là đơn điệu
1	$p < 2$ lân cận	Nhỏ nhất cục bộ
2	$p < 1$ lân cận & $p == 1$ lân cận	Cạnh
3	$p > 1$ lân cận & $p == 1$ lân cận	Cạnh
4	$p > 2$ lân cận	Lớn nhất cục bộ

Phần bù dải (BO) phân loại tất cả các pixel trong một vùng CTB thành 32 dải đồng nhất bằng cách sử dụng năm bit quan trọng nhất của trị số pixel làm chỉ số dải.

Nói cách khác, khoảng cường độ pixel được chia thành 32 đoạn bằng nhau từ 0 đến trị số cường độ lớn nhất (ví dụ, 255 đối với pixel 8 bit). Bốn băng tần liền kề được nhóm lại với nhau và mỗi nhóm được biểu thị bằng vị trí bên trái nhất của nó như được thể hiện trên FIG.13. Bộ mã hóa tìm kiếm tất cả các vị trí để có được nhóm có mức giảm méo tối đa bằng cách bù mỗi dải.

FIG.13 thể hiện một ví dụ về bốn dải được nhóm lại với nhau và được biểu thị bằng vị trí dải bắt đầu của nó.

2.10 Dự báo kết hợp liên và trong (Combined inter and intra prediction -CIIP)

Trong VTM5, khi CU được mã hóa ở chế độ hợp nhất, nếu CU chứa ít nhất 64 mẫu sắc độ (tức là, chiều rộng CU nhân với chiều cao CU bằng hoặc lớn hơn 64) và nếu cả chiều rộng CU và chiều cao CU đều nhỏ hơn 128 mẫu sắc độ, một cờ bổ sung được báo hiệu để chỉ ra chế độ dự báo liên/trong kết hợp (CIIP) có được áp dụng đối với CU hiện tại này hay không. Như tên của nó đã chỉ ra, dự báo CIIP kết hợp tín hiệu dự báo liên với tín hiệu dự báo trong. Tín hiệu dự báo liên Trong chế độ CIIP P_{inter} được suy ra ngang bằng với cách sử dụng cùng một quy trình dự báo liên áp dụng cho chế độ hợp nhất thông thường; và tín hiệu dự báo trong P_{intra} được tạo ra sau quy trình dự báo trong thông thường với chế độ phẳng. Sau đó, các tín hiệu dự báo trong và liên được kết hợp bằng cách sử dụng trung bình có trọng số, trong đó trị số của trọng số được tính tùy thuộc vào chế độ mã hóa của các khối lân cận trên cùng và bên trái (được thể hiện trên FIG.14) như sau:

- Nếu khối lân cận trên cùng có sẵn và được mã hóa trong, thì hãy đặt isIntraTop bằng 1, nếu không thì đặt isIntraTop bằng 0;

- Nếu khối lân cận bên trái có sẵn và được mã hóa trong, thì đặt isIntraLeft bằng 1, nếu không thì đặt isIntraLeft bằng 0;

- Nếu (isIntraLeft + isIntraTop) bằng 2, thì wt được đặt ngang bằng với 3;

- Nếu không, nếu (isIntraLeft + isIntraTop) bằng 1, thì wt được đặt ngang bằng với 2;

- Nếu không, hãy đặt wt bằng 1.

Dự báo CIIP được tạo ra như sau:

$$P_{CIIP} = ((4 - wt) * P_{inter} + wt * P_{intra} + 2) \gg 2 \quad (3-2)$$

FIG.14 thể hiện các ví dụ về các khối lân cận trên cùng và bên trái dùng trong tính toán trọng số CIIP.

2.11 Ánh xạ độ chói với việc định tỷ lệ sắc độ (LMCS)

Trong VTM5, một công cụ mã hóa được gọi là ánh xạ độ chói với việc định tỷ lệ sắc độ (LMCS) được thêm vào như khối xử lý mới trước các bộ lọc vòng lặp. LMCS có hai thành phần chính: 1) ánh xạ trong vòng của thành phần độ chói dựa trên các mô hình tuyến tính mỗi mảnh thích ứng; 2) đối với các thành phần sắc độ, việc định tỷ lệ dư lượng sắc độ phụ thuộc vào độ chói được áp dụng. FIG.15 thể hiện kiến trúc LMCS từ quan điểm của bộ giải mã. Các khối chấm chấm được thể hiện trên FIG.15 chỉ ra rằng nếu việc xử lý được áp dụng trong miền được ánh xạ; và chúng bao gồm việc lượng tử hóa nghịch đảo, việc biến đổi nghịch đảo, việc dự báo trong độ chói và việc thêm dự báo độ chói cùng với phần dư độ chói. Các khối chưa được đánh dấu được thể hiện trên FIG. 15 chỉ ra rằng nếu việc xử lý được áp dụng trong miền ban đầu (tức là không được ánh xạ); và chúng bao gồm các bộ lọc vòng lặp như tách khối, ALF và SAO, dự báo bù chuyển động, dự báo trong sắc độ, thêm dự báo sắc độ cùng với dư sắc độ và lưu trữ ảnh đã được giải mã dưới dạng ảnh tham chiếu. Các khối được đánh dấu được thể hiện trên FIG. 15 là các khối chức năng mới của LMCS, bao gồm ánh xạ thuận và nghịch của tín hiệu độ chói và quy trình định tỷ lệ sắc độ phụ thuộc vào độ chói. Giống như hầu hết các công cụ khác trong VVC, LMCS có thể được kích hoạt/vô hiệu hóa ở cấp trình tự bằng cách sử dụng cờ SPS.

FIG.15 thể hiện các ví dụ về ánh xạ độ chói với kiến trúc định tỷ lệ sắc độ.

2.12 Phân vùng cây nhị phân

Trong thiết kế VVC hiện tại, đối với lát cắt I, mỗi CTU có thể được chia thành các đơn vị mã hóa với các mẫu độ chói 64x64 bằng cách sử dụng việc tách cây tứ phân ngầm định và rằng các đơn vị mã hóa này là gốc của hai cấu trúc cú pháp coding_tree riêng biệt đối với độ chói và sắc độ.

Vì cây nhị phân trong ảnh mà cho phép áp dụng việc phân vùng khác nhau trong cây mã hóa sắc độ so với cây mã hóa độ chói, nên cây nhị phân tạo ra đường ống mã hóa dài hơn và khoảng trị số QTBT MinQTSIZEC và MinBtSizeY và MinTTSIZEY trong cây sắc độ cho phép các khối sắc độ nhỏ như 2x2, 4x2 và 2x4. Nó tạo ra các khó khăn trong thiết kế bộ giải mã thực tế. Hơn nữa, một vài chế độ dự

báo như CCLM, chế độ phẳng và chế độ góc cần phép nhân. Để giảm bớt các vấn đề nêu trên, kích thước khối sắc độ nhỏ (2×2 / 2×4 / 4×2) bị hạn chế trong cây nhị phân như một hạn chế phân vùng.

2.13 Đơn vị dự báo trong sắc độ nhỏ nhất (SCIPU) trong JVET-O0050

Kích thước sắc độ nhỏ không thân thiện với việc thực hiện phân cứng. Trong các trường hợp cây nhị phân, các khối sắc độ có kích thước quá nhỏ không được phép. Tuy nhiên, trong các trường hợp cây đơn phân, phác thảo VVC 5 vẫn cho phép trong các khối sắc độ 2×2 , 2×4 , 4×2 . Để hạn chế kích thước của khối sắc độ, trong cây mã hóa đơn phân, SCIPU được định nghĩa trong JVET-O0050 là nút cây mã hóa có kích thước khối sắc độ lớn hơn hoặc bằng các mẫu sắc độ TH và có ít nhất khối độ chói con nhỏ hơn mẫu độ chói 4TH, trong đó TH được đặt ngang bằng với 16 trong đóng góp này. Yêu cầu rằng trong mỗi SCIPU, tất cả các CB là liên hoặc tất cả các CB không liên, tức là liên hoặc IBC. Trong trường hợp SCIPU không liên, cần phải tiếp tục yêu cầu rằng sắc độ của SCIPU không liên đó sẽ không được tách thêm nữa và độ chói của SCIPU được phép tách thêm. Theo cách này, kích thước trong CB sắc độ nhỏ nhất là 16 mẫu sắc độ và CB sắc độ sắc 2×2 , 2×4 và 4×2 bị loại bỏ. Ngoài ra, việc định tỷ lệ sắc độ không được áp dụng trong trường hợp SCIPU không liên.

Hai ví dụ SCIPU được thể hiện trên FIG.16. Trên FIG.16 (a), một CB sắc độ của các mẫu sắc độ 8×4 và ba CB độ chói (các CB độ chói 4×8 , 8×8 , 4×8) tạo ra một SCIPU vì cây tam phân (TT) tách ra từ 8×4 mẫu sắc độ sẽ tạo ra các CB sắc độ nhỏ hơn 16 mẫu sắc độ. Trên FIG.16(b), một CB sắc độ của các mẫu sắc độ 4×4 (phía bên trái của các mẫu sắc độ 8×4) và ba CB độ chói (các CB độ chói 8×4 , 4×4 , 4×4) tạo ra một SCIPU và CB sắc độ còn lại của 4×4 các mẫu (phía bên phải của các mẫu sắc độ 8×4) và hai CB độ chói (8×4 , 8×4 CB) tạo ra một SCIPU vì cây nhị phân (BT) tách ra từ các mẫu sắc độ 4×4 sẽ tạo ra các CB sắc độ nhỏ hơn 16 mẫu sắc độ.

FIG.16 thể hiện các ví dụ về SCIPU.

Loại SCIPU được suy ra là không liên nếu lát cắt hiện tại là lát cắt I hoặc SCIPU hiện tại có phân vùng độ chói 4×4 trong đó sau khi được tách thêm một lần (vì không có sự liên 4×4 nào được phép trong VVC); nếu không, loại SCIPU (liên

hoặc không liên) được chỉ ra bằng một cờ báo hiệu trước khi phân tích cú pháp CU trong SCIPU.

2.14 Các ràng buộc của khối sắc độ nhỏ trong bản phác thảo VVC 6

Trong VVC phác thảo 6 (JVET-O2001-vE.docx), các ràng buộc về các khối sắc độ nhỏ được thực hiện như sau (phần liên quan được đánh dấu in nghiêng đậm).

coding_tree x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, <i>treeTypeCurr</i> , <i>modeTypeCurr</i>) {	Bộ mô tả
...	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)
if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
}	
if(<i>modeTypeCondition</i> == 1)	
<i>modeType</i> = <i>MODE_TYPE_INTRA</i>	
else if(<i>modeTypeCondition</i> == 2) {	
<i>mode_constraint_flag</i>	ae(v)

<i>modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYP E_INTER</i>	
<i>} else {</i>	
<i>modeType = modeTypeCurr</i>	
<i>}</i>	
<i>treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr</i>	
<i>if(!split_qt_flag) {</i>	
<i>if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {</i>	
<i>depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0</i>	
<i>x1 = x0 + (cbWidth / 2)</i>	
<i>coding_tree(x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</i>	
<i>if(x1 < pic_width_in_luma_samples)</i>	
<i>coding_tree(x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</i>	

} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
coding_tree(x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < pic_height_in_luma_samples)	
coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
qgOnY = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)	
qgOnC = qgOnC && (cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv)	
coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)	

<code>} else { /* SPLIT_TT_HOR */</code>	
<code> y1 = y0 + (cbHeight / 4)</code>	
<code> y2 = y0 + (3 * cbHeight / 4)</code>	
<code> qgOnY = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code> qgOnC = qgOnC && (cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv)</code>	
<code> coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code> }</code>	
<code> } else {</code>	
<code> x1 = x0 + (cbWidth / 2)</code>	
<code> y1 = y0 + (cbHeight / 2)</code>	
<code> coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code> if(x1 < pic_width_in_luma_samples)</code>	
<code> coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code> if(y1 < pic_height_in_luma_samples)</code>	
<code> coding_tree(x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)</code>	

if(y1 < pic_height_in_luma_samples && x1 < pic_width_in_luma_samples)	
coding_tree(x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, 0, 0, DUAL_TREE_CHROMA, modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr, modeTypeCurr)	
}	

2.14.1.1 Cú pháp đơn vị mã hóa

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Bộ mô tả
chType = treeType == DUAL_TREE_CHROMA ? 1 : 0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	

pred_mode_flag	ae(v)
<pre> if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA) </pre>	
pred_mode_ibc_flag	ae(v)
<pre> if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER) </pre>	
pred_mode_plt_flag	ae(v)
}	
...	

Biến modeTypeCondition được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 0
- loại lát cắt == I và qtbt_dual_tree_intra_flag ngang bằng với 1
- modeTypeCurr không ngang bằng với MODE_TYPE_ALL
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và split_qt_flag ngang bằng với 1

- $cbWidth * cbHeight$ ngang bằng với 64 và $MttSplitMode [x0] [y0] [mttDepth]$ ngang bằng với $SPLIT_TT_HOR$ hoặc $SPLIT_TT_VER$
- $cbWidth * cbHeight$ ngang bằng với 32 và $MttSplitMode [x0] [y0] [mttDepth]$ ngang bằng với $SPLIT_BT_HOR$ hoặc $SPLIT_BT_VER$
- Nếu không, nếu một trong số các điều kiện sau là đúng, $modeTypeCondition$ được đặt ngang bằng với $1 + (loại\ lát\ cắt! = I? 1: 0)$
- $cbWidth * cbHeight$ ngang bằng với 64 và $MttSplitMode [x0] [y0] [mttDepth]$ ngang bằng với $SPLIT_BT_HOR$ hoặc $SPLIT_BT_VER$
- $cbWidth * cbHeight$ ngang bằng với 128 và $MttSplitMode [x0] [y0] [mttDepth]$ ngang bằng với $SPLIT_TT_HOR$ hoặc $SPLIT_TT_VER$
- Nếu không, $modeTypeCondition$ được đặt ngang bằng với 0

Quy trình tách tứ phân được phép

Đầu vào của quy trình này là:

- kích thước khối mã hóa $cbSize$ trong các mẫu độ chói,
- độ sâu cây của các loại $mttDepth$,
- biến $treeType$ quy định xem liệu cây đơn phân ($SINGLE_TREE$) hay cây nhị phân được sử dụng để phân vùng các CTU và khi cây nhị phân được sử dụng, xem liệu các thành phần độ chói ($DUAL_TREE_LUMA$) hoặc sắc độ ($DUAL_TREE_CHROMA$) hiện đang được xử lý hay không,
- biến $modeType$ quy định xem có thể sử dụng chế độ mã hóa trong ($MODE_INTRA$), IBC ($MODE_IBC$), bảng màu ($MODE_PLT$) và các chế độ mã hóa liên ($MODE_TYPE_ALL$) có thể được sử dụng hay chỉ chế độ mã hóa trong, chế độ mã hóa IBC và chế độ mã hóa bảng màu ($MODE_TYPE_INTRA$), hoặc xem liệu chỉ có thể sử dụng các chế độ mã hóa liên ($MODE_TYPE_INTER$) đối với các đơn vị mã hóa bên trong nút cây mã hóa hay không.

Đầu ra của quy trình này là biến $allowSplitQt$.

Biến $allowSplitQt$ được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, $allowSplitQt$ được đặt ngang bằng với $FALSE$:
- $treeType$ ngang bằng với $SINGLE_TREE$ hoặc $DUAL_TREE_LUMA$ và $cbSize$ nhỏ hơn hoặc ngang bằng với $MinQtSizeY$

- treeType ngang bằng với DUAL_TREE_CHROMA và cbSize / SubWidthC nhỏ hơn hoặc ngang bằng với MinQtSizeC
- mttDepth không ngang bằng với 0
- treeType ngang bằng với DUAL_TREE_CHROMA và (cbSize / SubWidthC) nhỏ hơn hoặc ngang bằng với 4
- treeType ngang bằng với DUAL_TREE_CHROMA và modeType là MODE_TYPE_INTRA
- Nếu không, allowSplitQt được đặt ngang bằng với TRUE.

Quy trình tách nhị phân được phép

Đầu vào của quy trình này là:

- một chế độ tách nhị phân btSplit,
- chiều rộng khối mã hóa cbWidth trong các mẫu độ chói,
- chiều cao khối mã hóa cbHeight trong các mẫu độ chói,
- vị trí (x0, y0) của mẫu độ chói trên cùng bên trái của khối mã hóa được coi là liên quan đến mẫu độ chói trên cùng bên trái của hình ảnh,
- độ sâu cây của các loại mttDepth,
- độ sâu cây tối đa của các loại với phần bù maxMttDepth,
- kích thước tối đa của cây nhị phân maxBtSize,
- kích thước tối thiểu của cây tứ phân minQtSize,
- một chỉ mục phân vùng partIdx,
- biến treeType quy định xem liệu cây đơn phân (SINGLE_TREE) hay cây nhị phân được sử dụng để phân vùng các CTU và khi cây nhị phân được sử dụng, xem liệu các thành phần độ chói (DUAL_TREE_LUMA) hoặc sắc độ (DUAL_TREE_CHROMA) hiện đang được xử lý hay không,
- biến modeType quy định xem có thể sử dụng chế độ mã hóa trong (MODE_INTRA), IBC (MODE_IBC), chế độ bảng màu (MODE_PLT) và các chế độ mã hóa liên (MODE_TYPE_ALL) hay chỉ có thể sử dụng chế độ mã hóa trong, chế độ IBC và chế độ bảng màu (MODE_TYPE_INTRA), hoặc xem liệu chỉ có thể sử dụng các chế độ mã hóa liên (MODE_TYPE_INTER) đối với các đơn vị mã hóa trong nút cây mã hóa hay không.

Đầu ra của quy trình này là biến allowBtSplit.

Bảng 6-2 - Đặc điểm kỹ thuật của `parallelTtSplit` và `cbSize` dựa trên `btSplit`.

	<code>btSplit == SPLIT_BT_VER</code>	<code>btSplit == SPLIT_BT_HOR</code>
<code>parallelTtSplit</code>	<code>SPLIT_TT_VER</code>	<code>SPLIT_TT_HOR</code>
<code>cbSize</code>	<code>cbWidth</code>	<code>cbHeight</code>

Các biến `llelTtSplit` và `cbSize` được tạo ra như được quy định trong Bảng 6 2.

Biến `allowBtSplit` được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, `allowBtSplit` được đặt ngang bằng với FALSE:

- `cbSize` nhỏ hơn hoặc ngang bằng với `MinBtSizeY`
- `cbWidth` lớn hơn `maxBtSize`
- `cbHeight` lớn hơn `maxBtSize`
- `mttDepth` lớn hơn hoặc ngang bằng với `maxMttDepth`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ nhỏ hơn hoặc [ngang bằng với 16
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và `modeType` là `MODE_TYPE_INTRA`
- Nếu không, nếu tất cả các điều kiện sau là đúng, `allowBtSplit` được đặt ngang bằng với FALSE
- `btSplit` ngang bằng với `SPLIT_BT_VER`
- $y0 + cbHeight$ lớn hơn `pic_height_in_luma_samples`
- Nếu không, nếu tất cả các điều kiện sau là đúng, `allowBtSplit` được đặt ngang bằng với FALSE
- `btSplit` ngang bằng với `SPLIT_BT_VER`
- `cbHeight` lớn hơn `MaxTbSizeY`
- $x0 + cbWidth$ lớn hơn `pic_width_in_luma_samples`
- Nếu không, nếu tất cả các điều kiện sau là đúng, `allowBtSplit` được đặt ngang bằng với FALSE
- `btSplit` ngang bằng với `SPLIT_BT_HOR`
- `cbWidth` lớn hơn `MaxTbSizeY`
- $y0 + cbHeight$ lớn hơn `pic_height_in_luma_samples`

- Nếu không, nếu tất cả các điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE
- $x0 + cbWidth$ lớn hơn $pic_width_in_luma_samples$
- $y0 + cbHeight$ lớn hơn $pic_height_in_luma_samples$
- $cbWidth$ lớn hơn $minQtSize$
- Nếu không, nếu tất cả các điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE
- btSplit ngang bằng với SPLIT_BT_HOR
- $x0 + cbWidth$ lớn hơn $pic_width_in_luma_samples$
- $y0 + cbHeight$ nhỏ hơn hoặc ngang bằng với $pic_height_in_luma_samples$
- Nếu không, nếu tất cả các điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE:
- $mttDepth$ lớn hơn 0
- $partIdx$ ngang bằng với 1
- $MttSplitMode[x0][y0][mttDepth - 1]$ ngang bằng với $parallelTtSplit$
- Nếu không, nếu tất cả các điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE
- btSplit ngang bằng với SPLIT_BT_VER
- $cbWidth$ nhỏ hơn hoặc ngang bằng với $MaxTbSizeY$
- $cbHeight$ lớn hơn $MaxTbSizeY$
- Nếu không, nếu tất cả các điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE
- btSplit ngang bằng với SPLIT_BT_HOR
- $cbWidth$ lớn hơn $MaxTbSizeY$
- $cbHeight$ nhỏ hơn hoặc ngang bằng với $MaxTbSizeY$
- Nếu không, allowBtSplit được đặt ngang bằng với TRUE.

Quy trình tách tam phân được phép

Đầu vào của quy trình này là:

- một chế độ tách tam phân $ttSplit$,
- chiều rộng khối mã hóa $cbWidth$ trong các mẫu độ chói,
- chiều cao khối mã hóa $cbHeight$ trong các mẫu độ chói,

- vị trí (x0, y0) của mẫu độ chói trên cùng bên trái của khối mã hóa được coi là liên quan đến mẫu độ chói trên cùng bên trái của hình ảnh,
- độ sâu cây của các loại mttDepth
- độ sâu tối đa của các cây với phân bù maxMttDepth,
- kích thước tối đa của cây tam phân maxTtSize,
- biến treeType quy định xem liệu cây đơn phân (SINGLE_TREE) hay cây nhị phân được sử dụng để phân vùng các CTU và khi cây nhị phân được sử dụng, xem liệu luma(DUAL_TREE_LUMA) hoặc các thành phần sắc độ (DUAL_TREE_CHROMA) hiện đang được xử lý hay không,
- biến modeType quy định xem có thể sử dụng chế độ mã hóa trong (MODE_INTRA), chế độ IBC (MODE_IBC), chế độ bảng màu (MODE_PLT) và các chế độ mã hóa liên (MODE_TYPE_ALL) hay chỉ có thể sử dụng chế độ mã hóa trong, chế độ IBC và chế độ bảng màu (MODE_TYPE_INTRA), hoặc xem liệu chỉ có thể sử dụng các chế độ mã hóa liên (MODE_TYPE_INTER) đối với các đơn vị mã hóa trong nút cây mã hóa hay không.

Đầu ra của quy trình này là biến allowTtSplit.

Bảng 6- 3 - Đặc điểm kỹ thuật của cbSize dựa trên ttSplit.

	ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

Biến cbSize được tạo ra như được quy định trong Bảng 6 -3.

Biến allowTtSplit được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, allowTtSplit được đặt ngang bằng với FALSE:
- cbSize nhỏ hơn hoặc ngang bằng với $2 * \text{MinTtSizeY}$
- cbWidth lớn hơn Min (MaxTbSizeY, maxTtSize)
- cbHeight lớn hơn Min (MaxTbSizeY, maxTtSize)
- mttDepth lớn hơn hoặc ngang bằng với maxMttDepth
- $x0 + \text{cbWidth}$ lớn hơn pic_width_in_luma_samples
- $y0 + \text{cbHeight}$ lớn hơn pic_height_in_luma_samples
- treeType ngang bằng với DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ nhỏ hơn hoặc bằng 32

- treeType ngang bằng với DUAL_TREE_CHROMA và modeType là MODE_TYPE_INTRA

- Nếu không, allowTtSplit được đặt ngang bằng với TRUE.

pred_mode_flag ngang bằng với 0 quy định rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo liên. pred_mode_flag ngang bằng với 1 quy định rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo trong.

Khi không có pred_mode_flag, nó được suy ra như sau:

- Nếu cbWidth ngang bằng với 4 và cbHeight ngang bằng với 4, thì suy ra pred_mode_flag ngang bằng với 1.

- Nếu không, nếu modeType là MODE_TYPE_INTRA, thì suy ra pred_mode_flag ngang bằng với 1.

- Nếu không, nếu modeType là MODE_TYPE_INTER, thì pred_mode_flag được suy ra ngang bằng với 0.

- Nếu không, pred_mode_flag được suy ra ngang bằng với 1 khi giải mã lát cắt I và ngang bằng với 0 khi lần lượt giải mã lát cắt P hoặc B.

Biến CuPredMode [chType] [x] [y] được tạo ra như sau đối với $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$:

- Nếu pred_mode_flag ngang bằng với 0, CuPredMode [chType] [x] [y] được đặt ngang bằng với MODE_INTER.

- Nếu không (pred_mode_flag ngang bằng với 1), CuPredMode [chType] [x] [y] được đặt ngang bằng với MODE_INTRA.

pred_mode_ibc_flag ngang bằng với 1 quy định rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo IBC. pred_mode_ibc_flag ngang bằng với 0 quy định rằng đơn vị mã hóa hiện tại không được mã hóa trong chế độ dự báo IBC.

Khi không có pred_mode_ibc_flag, nó được suy ra như sau:

- Nếu cu_skip_flag [x0] [y0] ngang bằng với 1, và cbWidth ngang bằng với 4 và cbHeight ngang bằng với 4, thì suy ra pred_mode_ibc_flag ngang bằng với 1.

- Nếu không, nếu cả cbWidth và cbHeight đều ngang bằng với 128, thì suy ra pred_mode_ibc_flag ngang bằng với 0.

- Nếu không, nếu modeType là MODE_TYPE_INTER, thì suy ra pred_mode_ibc_flag ngang bằng với 0.

- Nếu không, nếu `treeType` ngang bằng với `DUAL_TREE_CHROMA`, thì `pred_mode_ibc_flag` được suy ra ngang bằng với 0.

- Nếu không, `pred_mode_ibc_flag` được coi là bằng trị số của `sps_ibc_enabled_flag` khi giải mã lát cắt I và 0 khi lần lượt giải mã lát cắt P hoặc B.

Khi `pred_mode_ibc_flag` ngang bằng với 1, biến `CuPredMode[chType][x][y]` được đặt ngang bằng với `MODE_IBC` đối với $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$.

3. Ví dụ về các vấn đề kỹ thuật được giải quyết bằng các giải pháp kỹ thuật được bộc lộ

1. Hiện nay, IBC được coi là `MODE_TYPE_INTRA` và do đó khối sắc độ nhỏ không được phép, mà dẫn đến việc mất hiệu quả mã hóa không cần thiết.

2. Hiện nay, bảng màu được coi là `MODE_TYPE_INTRA` và do đó khối sắc độ nhỏ không được phép sử dụng, mà dẫn đến việc mất hiệu quả mã hóa không cần thiết.

3. Hiện nay, các ràng buộc khối sắc độ nhỏ không xem xét định dạng lấy mẫu con màu.

4. Hiện nay, các ràng buộc về việc phân vùng và chế độ dự báo giống nhau trên các khối nhỏ được áp dụng cho tất cả các định dạng sắc độ. Tuy nhiên, có thể mong muốn thiết kế các cơ chế ràng buộc khác nhau trên các khối nhỏ ở định dạng sắc độ 4: 2: 0 và 4: 2: 2.

5. Hiện nay, tín hiệu cờ chế độ bảng màu phụ thuộc vào `modeType`, đó là điều không mong muốn vì bảng màu có thể không áp dụng các ràng buộc khối nhỏ.

6. Hiện nay, cờ chế độ IBC được suy ra ngang bằng với 0 đối với lát cắt P/B với `cu_skip_flag` ngang bằng với 1 nhưng `MODE_TYPE` ngang bằng với `MODE_TYPE_INTRA`, mà là bất hợp pháp trong phân tích cú pháp.

7. Hiện nay, chế độ IBC độ chói không phải 4x4 không được phép đối với các khối độ chói SCIPU, mà có thể không được mong muốn và có thể gây mất hiệu quả mã hóa.

8. Khối sắc độ 2xH vẫn được phép, mà không thân thiện với việc thực hiện phần cứng.

9. CIIP được coi là `MODE_INTER` trong khi nó sử dụng dự báo liên, mà phá vỡ các ràng buộc trong một số trường hợp.

10. Khi SCIPU được áp dụng, QP delta đối với sắc độ có thể được báo hiệu tùy thuộc vào việc tách độ chói. Ví dụ, khi kích thước khối hiện tại là 16×8 trong các mẫu độ chói và được tách bằng TT dọc, cây nhị phân cục bộ có thể được áp dụng. Quy định rằng

$$qgOnC = qgOnC \&\& (cbSubdiv + 2 \leq cu_chroma_qp_offset_subdiv)$$

Vì vậy, qgOnC được đặt ngang bằng với 0 nếu $cbSubdiv + 2 \leq cu_chroma_qp_offset_subdiv$. Việc đặt có điều kiện này giả định rằng thành phần sắc độ cũng được tách bởi TT. Với cây nhị phân cục bộ, thành phần sắc độ có thể không được tách, do đó cbSubdiv có thể lớn hơn cu_chroma_qp_offset_subdiv. IsCuChromaQpOffsetCoded phải được đặt ngang bằng với 0 để cho phép báo hiệu QP delta đối với sắc độ. Tuy nhiên, IsCuChromaQpOffsetCoded không được đặt ngang bằng với 0 vì qgOnC được đặt ngang bằng với 0.

4. Ví dụ về các giải pháp kỹ thuật và các phương án

Danh mục dưới đây nên được coi là ví dụ. Các kỹ thuật này không nên được giải thích theo một cách hẹp. Hơn thế nữa, các kỹ thuật này có thể được kết hợp theo cách bất kỳ.

Trong Bản mô tả sáng chế này, "Nút cây mã hóa $M \times N$ " biểu thị khối $M \times N$, với M là chiều rộng khối và N là chiều cao khối trong các mẫu độ chói, mà có thể được phân vùng thêm, như bằng QT/BT/TT. Ví dụ, khối có thể là một nút QT, một nút BT hoặc một nút TT. Nút cây mã hóa có thể là một đơn vị mã hóa (ví dụ, với ba thành phần màu đối với cây đơn phân, với hai thành phần màu sắc độ đối với việc mã hóa sắc độ cây nhị phân và chỉ một thành phần màu độ chói đối với việc mã hóa độ chói cây nhị phân) hoặc khối mã hóa độ chói, hoặc khối mã hóa sắc độ. "Đơn vị nút cây mã hóa nhỏ" có thể biểu thị một nút cây mã hóa có kích thước khối $M \times N$ bằng 32/64/128 trong các mẫu độ chói.

Nếu không được đề cập một cách cụ thể, chiều rộng W và chiều cao H đối với khối mã hóa được đo trong các mẫu độ chói. Ví dụ, khối mã hóa $M \times N$ có nghĩa là khối độ chói $M \times N$ và/hoặc hai khối sắc độ $(M/SubWidthC) \times (N/SubHeightC)$, trong đó SubWidthC và SubHeightC được tạo ra bằng định dạng sắc độ như sau.

Định dạng sắc độ _i dc	Cờ mặt phẳng màu riêng biệt	Định dạng sắc độ	Chiều rộng phụ C	Chiều dài phụ C
0	0	Đơn sắc	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

1. Việc phân vùng và/hoặc cách phân vùng thành các khối nhỏ có thể phụ thuộc vào các định dạng màu.

a. Trong một ví dụ, đối với định dạng màu 4: 4: 4, các ràng buộc về kích thước của các khối sắc độ có thể tuân theo các ràng buộc đó trên các khối sắc độ.

b. Trong một ví dụ, đối với định dạng màu 4: 2: 2, các ràng buộc về kích thước của khối sắc độ có thể tuân theo các ràng buộc đó đối với định dạng màu 4: 2: 0.

c. Trong một ví dụ, đối với định dạng sắc độ 4: 0: 0 và/hoặc 4: 4: 4, các ràng buộc về việc phân vùng khối nhỏ và/hoặc chế độ dự báo có thể không được áp dụng.

d. Trong một ví dụ, các ràng buộc về việc phân vùng khối nhỏ và/hoặc chế độ dự báo có thể được áp dụng khác nhau đối với các định dạng sắc độ khác nhau.

i. Trong một ví dụ, đối với nút cây mã hóa MxN (như 8x8) với việc tách BT theo chiều ngang, ở định dạng sắc độ 4: 2: 2, việc tách BT theo chiều ngang có thể được phép đối với cả khối sắc độ và khối độ chói, trong khi ở định dạng sắc độ 4: 2: 0, việc tách BT theo chiều ngang có thể được phép đối với khối độ chói nhưng bị vô hiệu hóa đối với khối sắc độ.

ii. Trong một ví dụ, đối với nút cây mã hóa MxN (như 16x4) với việc tách BT theo chiều dọc, ở định dạng sắc độ 4: 2: 2, việc tách BT theo chiều dọc có thể được phép đối với cả khối sắc độ và khối độ chói, trong khi ở định dạng sắc độ 4: 2: 0, việc tách BT theo chiều dọc có thể được phép đối với khối độ chói nhưng bị vô hiệu hóa đối với khối sắc độ.

iii. Trong một ví dụ, đối với nút cây mã hóa MxN (như 8x16) với việc tách BT theo chiều ngang, ở định dạng sắc độ 4: 2: 2, việc tách BT theo chiều ngang có thể được

phép đối với cả khối sắc độ và khối độ chói, trong khi ở định dạng sắc độ 4: 2: 0, việc tách BT theo chiều ngang có thể được phép đối với khối độ chói nhưng bị vô hiệu hóa đối với khối sắc độ.

iv. Trong một ví dụ, đối với nút cây mã hóa MxN (như 32x4) với việc tách TT theo chiều dọc, ở định dạng sắc độ 4: 2: 2, việc tách BT theo chiều dọc có thể được phép đối với cả khối sắc độ và khối độ chói, trong khi ở định dạng sắc độ 4: 2: 0, việc tách BT theo chiều dọc có thể được phép đối với khối độ chói nhưng bị vô hiệu hóa đối với khối sắc độ.

v. Trong một ví dụ, đối với định dạng màu 4: 0: 0 và/hoặc 4: 4: 4, các ràng buộc khối nhỏ có thể không được áp dụng.

e. Trong một ví dụ, việc có kích hoạt SCIPU hay không phụ thuộc vào định dạng màu.

i. Trong một ví dụ, SCIPU được kích hoạt đối với các định dạng màu 4: 2: 0 và 4: 2: 2.

ii. Trong một ví dụ, SCIPU bị vô hiệu hóa đối với định dạng màu 4: 0: 0 và/hoặc 4: 4: 4.

1. Trong một ví dụ, modeType có thể luôn ngang bằng với MODE_TYPE_ALL đối với định dạng màu 4: 0: 0 và/hoặc 4: 4: 4.

2. Trong một ví dụ, modeTypeCondition có thể luôn ngang bằng với 0 đối với định dạng màu 4: 0: 0 và/hoặc 4: 4: 4.

2. Cách xác định các chế độ dự báo (và/hoặc modeType) đối với các khối (con) của một nút cây mã hóa có thể phụ thuộc vào các định dạng sắc độ.

a. Trong một ví dụ, nếu một trong số các điều kiện nêu dưới đây là đúng, modeType của khối(con) được phân vùng bởi nút cây mã hóa này có thể ngang bằng với MODE_TYPE_ALL đối với định dạng sắc độ 4: 2: 2, trong khi đối với định dạng sắc độ 4: 2: 0, modeType có thể ngang bằng với MODE_TYPE_INTRA hoặc MODE_TYPE_INTER.

i. Nút cây mã hóa MxN (như 8x8) với việc tách BT theo chiều ngang

ii. Nút cây mã hóa MxN (như 16x4) với việc tách BT theo chiều dọc

iii. Nút cây mã hóa MxN (như 8x16) với việc tách BT theo chiều ngang

iv. Nút cây mã hóa MxN (như 32x4) với việc tách BT theo chiều dọc

3. Đề xuất đổi tên `MODE_TYPE_INTRA` thành `MODE_TYPE_NO_INTER` và hạn chế việc sử dụng `MODE_INTER`.
 - a. Trong một ví dụ, khi `modeType` của một đơn vị mã hóa ngang bằng với `MODE_TYPE_NO_INTER`, `MODE_INTER` có thể không được phép.
4. Đề xuất đổi tên `MODE_TYPE_INTER` thành `MODE_TYPE_NO_INTRA` và hạn chế việc sử dụng `MODE_INTRA`.
 - a. Trong một ví dụ, khi `modeType` của một đơn vị mã hóa ngang bằng với `MODE_TYPE_NO_INTRA`, `MODE_INTRA` có thể không được phép.
5. Cờ ràng buộc chế độ có thể không bao giờ được báo hiệu ở các định dạng sắc độ 4: 2: 2 và/hoặc 4: 0: 0 và/hoặc 4: 4: 4.
 - a. Trong một ví dụ, khi cờ ràng buộc chế độ không xuất hiện, nó có thể được suy ra ngang bằng với 1.
 - i. Theo cách khác, khi cờ ràng buộc chế độ không xuất hiện, nó có thể được suy ra ngang bằng với 0.
6. Việc và/hoặc cách áp dụng SCIPU trên khối mã hóa $M \times N$ với M là chiều rộng khối và N là chiều cao khối có thể phụ thuộc vào việc định dạng màu 4: 2: 0 hay 4: 2: 2.
 - a. Trong một ví dụ, ở định dạng màu 4: 2: 2, đối với khối mã hóa $M \times N$ với M là chiều rộng khối và N là chiều cao khối, SCIPU chỉ có thể được kích hoạt nếu M nhân với N (ký hiệu là $M * N$) bằng 64 hoặc 32.
 - b. Trong một ví dụ, một nút cây mã hóa với $M * N = 128$ có thể không bao giờ được coi là khối SCIPU ở định dạng màu 4: 2: 2.
 - c. Trong một ví dụ, một nút cây mã hóa với việc tách BT và $M * N = 64$ có thể không bao giờ được coi là khối SCIPU ở định dạng màu 4: 2: 2.
 - d. Trong một ví dụ, một nút cây mã hóa với việc tách `_qt_flag` ngang bằng với 1 và $M * N = 64$, mà có thể là khối SCIPU ở định dạng màu 4: 2: 2.
 - e. Trong một ví dụ, một nút cây mã hóa với việc tách BT và $M * N = 64$, mà có thể được coi là khối SCIPU ở định dạng màu 4: 2: 2.
 - f. Trong một ví dụ, một nút cây mã hóa với việc tách BT và $M * N = 32$, mà có thể được coi là khối SCIPU ở định dạng màu 4: 2: 2.

g. Trong phần mô tả nêu trên, đối với khối SCIPU ở định dạng màu 4: 2: 2, modeTypeCondition có thể luôn ngang bằng với 1.

h. Trong phần mô tả nêu trên, đối với khối SCIPU ở định dạng màu 4: 2: 2, chỉ MODE_TYPE_INTRA có thể được phép đối với cả khối hiện tại trong nút cha và tất cả các khối con bên dưới các nút lá con.

7. Ở định dạng màu 4: 2: 2, modeTypeCondition của khối SCIPU có thể luôn ngang bằng với 1.

a. Trong một ví dụ, modeTypeCondition có thể bằng 0 hoặc 1 đối với định dạng màu 4: 2: 2.

b. Trong một ví dụ, đối với các khối SCIPU ở định dạng màu 4: 2: 2, modeTypeCondition có thể không bao giờ ngang bằng với 2.

8. Ở định dạng màu 4: 2: 2, modeType của khối SCIPU có thể luôn ngang bằng với MODE_TYPE_INTRA.

a. Trong một ví dụ, modeType có thể ngang bằng với MODE_TYPE_ALL hoặc MODE_TYPE_INTRA ở định dạng màu 4: 2: 2.

b. Trong một ví dụ, đối với các khối SCIPU ở định dạng màu 4: 2: 2, MODE_TYPE_INTER có thể bị vô hiệu hóa.

9. Việc phân vùng khối có được phép hay không có thể phụ thuộc vào modeType và/hoặc kích thước khối.

a. Trong một ví dụ, việc tách BT và/hoặc TT có được phép đối với khối hay không có thể phụ thuộc vào modeType.

i. Trong một ví dụ, nếu modeType là MODE_TYPE_INTER, thì việc tách BT có thể không được phép đối với khối mã hóa hiện tại (ví dụ, allowBtSplit được đặt ngang bằng với false).

ii. Trong một ví dụ, nếu modeType là MODE_TYPE_INTER, thì việc tách BT có thể không được phép đối với khối mã hóa hiện tại (ví dụ, allowTtSplit được đặt ngang bằng với false).

b. Trong một ví dụ, việc tách BT và/hoặc TT có được phép đối với khối hay không có thể phụ thuộc vào modeType và kích thước khối.

i. Trong một ví dụ, đối với khối mã hóa $M \times N$, với M là chiều rộng khối và N là chiều cao khối, khi $M * N$ nhỏ hơn hoặc bằng 32 và modeType là

MODE_TYPE_INTER, thì việc tách BT có thể không được phép (ví dụ, allowBtSplit được đặt ngang bằng với false).

ii. Trong một ví dụ, đối với khối mã hóa $M \times N$, với M là chiều rộng khối và N là chiều cao khối, khi $M * N$ nhỏ hơn hoặc bằng 64 và modeType là MODE_TYPE_INTER, thì việc tách TT có thể không được phép (ví dụ, allowTtSplit được đặt ngang bằng với false).

10. Khi modeTypeCurr của cây mã hóa ngang bằng với MODE_TYPE_INTER, việc tách cây mã hóa có thể bị hạn chế.

a. Trong một ví dụ, khi modeTypeCurr của cây mã hóa ngang bằng với MODE_TYPE_INTER, việc tách BT có thể không được phép.

b. Trong một ví dụ, khi modeTypeCurr của cây mã hóa ngang bằng với MODE_TYPE_INTER, việc tách TT có thể không được phép.

c. Trong một ví dụ, khi modeTypeCurr của cây mã hóa ngang bằng với MODE_TYPE_INTER, việc tách TT có thể không được phép.

d. Trong một ví dụ, khi modeTypeCurr của cây mã hóa ngang bằng với MODE_TYPE_INTER và kích thước khối độ chói nhỏ hơn hoặc bằng 32, việc tách BT có thể không được phép.

e. Trong một ví dụ, khi modeTypeCurr của một cây mã hóa ngang bằng với MODE_TYPE_INTER và kích thước khối độ chói nhỏ hơn hoặc bằng 64, việc tách TT có thể không được phép.

11. Một đơn vị mã hóa có treeType là DUAL_TREE_LUMA có thể được mã hóa ở chế độ liên.

a. Trong một ví dụ, đơn vị mã hóa đã được mã hóa trong chế độ mã hóa liên, tức là MODE_INTER chỉ có thể chứa thành phần độ chói ngay cả đối với các định dạng màu có nhiều thành phần màu.

b. Trong một ví dụ, mà có thể cần phân tích cú pháp pred_mode_flag đối với khối DUAL_TREE_LUMA.

c. Trong một ví dụ, đối với khối DUAL_TREE_LUMA được mã hóa ở chế độ liên, các ràng buộc tương tự của chế độ liên đối với SINGLE_TREE cũng có thể được áp dụng.

i. Trong một ví dụ, khối liên 4×4 DUAL_TREE_LUMA có thể không được phép.

12. Việc sử dụng các khối trong sắc độ (và/hoặc IBC) với chiều rộng khối bằng M (như $M = 2$) có thể không được phép.

a. Trong một ví dụ, khối trong sắc độ $2 \times N$ (như $N \leq 64$) có thể không được phép trong cây nhị phân.

i. Trong một ví dụ, khi `treeType` ngang bằng với `DUAL_TREE_CHROMA` và chiều rộng khối bằng 4 mẫu sắc độ, việc tách BT theo chiều dọc có thể bị vô hiệu hóa.

ii. Trong một ví dụ, khi `treeType` ngang bằng với `DUAL_TREE_CHROMA` và chiều rộng khối bằng 8 mẫu sắc độ, việc tách BT theo chiều dọc có thể bị vô hiệu hóa.

b. Trong một ví dụ, các khối trong sắc độ $2 \times N$ (như $N \leq 64$) (và/hoặc IBC) có thể không được phép trong cây đơn phân.

i. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (như $M = 8$ và $N \leq 64$) với việc tách BT theo chiều dọc, một trong số các quy trình nêu dưới đây có thể được áp dụng.

1. Việc tách BT theo chiều dọc có thể không được phép đối với khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$ nhưng được phép đối với khối độ chói $8 \times N$.
2. Khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$ có thể không phải là việc tách BT theo chiều dọc và nó có thể được mã hóa bởi `MODE_INTRA` hoặc `MODE_IBC`.
3. Việc tách BT theo chiều dọc có thể được phép đối với cả khối độ chói $8 \times N$ và khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$, nhưng cả khối độ chói và khối sắc độ không được mã hóa bởi `MODE_INTRA` (ví dụ, mà có thể được mã hóa bởi `MODE_INTER` hoặc `MODE_IBC`).

ii. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (như $M = 16$ và $N \leq 64$) với việc tách BT theo chiều dọc, một trong số các quy trình nêu dưới đây có thể được áp dụng.

1. Việc tách BT theo chiều dọc có thể không được phép đối với khối sắc độ $8 \times N$ hoặc $8 \times (N/2)$ nhưng được phép đối với khối độ chói $16 \times N$.
2. Khối sắc độ $8 \times N$ hoặc $8 \times (N/2)$ có thể không phải là khối TT theo chiều dọc và được mã hóa bởi `MODE_INTRA` hoặc `MODE_IBC`.
3. Việc tách BT theo chiều dọc có thể được phép đối với cả khối độ chói $16 \times N$ và khối sắc độ $8 \times N$ hoặc $8 \times (N/2)$, nhưng cả khối độ chói và khối sắc độ có thể

không được mã hóa bởi `MODE_INTRA` (ví dụ, mà có thể được mã hóa bởi `MODE_INTER` hoặc `MODE_IBC`).

13. Chế độ IBC có thể được phép đối với các khối độ chói và/hoặc khối sắc độ bất kể nó có kích thước khối nhỏ hay không.

a. Trong một ví dụ, chế độ IBC có thể được phép đối với các khối sắc độ bao gồm khối sắc độ 8×4 / 8×8 / 16×4 và $4 \times N$ (như $N \leq 64$), thậm chí nếu `modeType` là `MODE_TYPE_INTRA`.

b. Trong một ví dụ, chế độ IBC có thể được phép đối với các khối sắc độ, thậm chí nếu `modeType` là `MODE_TYPE_INTRA`.

14. Báo hiệu của cờ chế độ dự báo IBC có thể phụ thuộc vào loại chế độ dự báo (ví dụ, `MODE_TYPE_INTRA`).

a. Trong một ví dụ, cờ chế độ dự báo IBC đối với khối không phải SKIP (ví dụ, khối mã hóa không được mã hóa bởi chế độ bỏ qua) có thể được báo hiệu rõ ràng trong dòng bit khi `treeType` không ngang bằng với `DUAL_TREE_CHROMA` và `modeType` là `MODE_TYPE_INTRA`.

15. Cờ chế độ dự báo IBC có thể được suy ra tùy thuộc vào cờ CU SKIP và loại chế độ (ví dụ, `modeType`).

a. Trong một ví dụ, nếu khối hiện tại được mã hóa là chế độ SKIP (như `cu_skip_flag` ngang bằng với 1) và `modeType` là `MODE_TYPE_INTRA`, cờ chế độ dự báo IBC (như `pred_mode_ibc_flag`) có thể được suy ra ngang bằng với 1.

16. Báo hiệu rõ ràng của cờ chế độ bảng màu có thể không phụ thuộc vào `modeType`.

a. Trong một ví dụ, việc báo hiệu của cờ chế độ bảng màu (như `pred_mode_plt_flag`) có thể phụ thuộc vào loại lát cắt, kích thước khối, chế độ dự báo, v.v., bất kể `modeType` là gì.

b. Trong một ví dụ, cờ chế độ bảng màu (như `pred_mode_plt_flag`) được suy ra ngang bằng với 0 khi `modeType` là `MODE_TYPE_INTER` hoặc `MODE_TYPE_INTRA`.

17. Chế độ IBC có thể được phép sử dụng khi `modeType` là `MODE_TYPE_INTER`.

- a. Trong một ví dụ, IBC sắc độ có thể không được phép khi modeType là `MODE_TYPE_INTRA`.
- b. Trong một ví dụ, chế độ IBC có thể được phép sử dụng khi modeType là `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER`.
- c. Trong một ví dụ, chế độ IBC có thể được phép sử dụng bất kể modeType là gì.
- d. Trong một ví dụ, trong một SCIPU, cả chế độ IBC và liên đều có thể được phép.
- e. Trong một ví dụ, kích thước của khối sắc độ IBC có thể luôn tương ứng với kích thước của khối độ chói tương ứng.
- f. Trong một ví dụ, khi modeType là `MODE_TYPE_INTER` và kích thước đơn vị mã hóa là 4x4 tính theo độ chói, thì tín hiệu của `pred_mode_ibc_flag` có thể bị bỏ qua và `pred_mode_ibc_flag` có thể được suy ra ngang bằng với 1.

18. Chế độ bảng màu có thể được phép sử dụng khi modeType là `MODE_TYPE_INTER`.

- a. Trong một ví dụ, bảng màu có thể không được phép khi modeType là `MODE_TYPE_INTRA`.
- b. Trong một ví dụ, chế độ IBC có thể được phép sử dụng khi modeType là `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER`.
- c. Trong một ví dụ, chế độ IBC có thể được phép sử dụng bất kể modeType là gì.
- d. Trong một ví dụ, chế độ bảng màu có thể được phép sử dụng khi modeType là `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER`.
- e. Trong một ví dụ, chế độ bảng màu có thể được phép sử dụng bất kể modeType là gì.
- f. Trong một ví dụ, trong một SCIPU, cả chế độ bảng màu và chế độ liên đều có thể được phép.
- g. Trong một ví dụ, trong một SCIPU, tất cả các bảng màu, IBC và chế độ liên đều có thể được phép.
- h. Trong một ví dụ, kích thước của khối sắc độ bảng màu có thể luôn tương ứng với kích thước của khối độ chói tương ứng.
- i. Trong một ví dụ, khi modeType là `MODE_TYPE_INTER` và kích thước đơn vị mã hóa là 4x4 tính theo độ chói, thì tín hiệu của `pred_mode_plt_flag` có thể bị bỏ qua và `pred_mode_plt_flag` có thể được suy ra ngang bằng với 1.

j. Trong một ví dụ, khi modeType là MODE_TYPE_INTER và kích thước đơn vị mã hóa là 4x4 tính theo độ chói, một thông báo có thể được gửi để chỉ ra rằng chế độ dự báo hiện tại là chế độ IBC hoặc chế độ bảng màu.

k. Trong một ví dụ, việc kích hoạt/vô hiệu hóa chế độ bảng màu có thể phụ thuộc vào loại lát cắt và modeType.

i. Trong một ví dụ, đối với các lát cắt I ngang bằng với MODE_TYPE_INTRA, chế độ bảng màu có thể được kích hoạt.

ii. Trong một ví dụ, đối với các lát cắt P/B có MODE_TYPE_INTER, chế độ bảng màu có thể được kích hoạt.

19. Khi chế độ bảng màu được kích hoạt, cây nhị phân cục bộ có thể không được phép.

a. Trong một ví dụ, khi chế độ bảng màu được kích hoạt, modeTypeCondition có thể luôn được đặt ngang bằng với 0.

20. Đối với các khối sắc độ nhỏ có chiều rộng bằng M (ví dụ, M = 2) hoặc chiều cao bằng N (ví dụ, N = 2), các chế độ dự báo trong được phép có thể bị hạn chế khác với các chế độ cho phép đối với các khối sắc độ lớn.

a. Trong một ví dụ, chỉ một tập hợp con của chế độ dự báo trong của các chế độ dự báo trong sắc độ có sẵn có thể được sử dụng.

b. Trong một ví dụ, chỉ chế độ INTRA_DC có thể được sử dụng.

c. Trong một ví dụ, chỉ chế độ INTRA_PLANAR có thể được sử dụng.

d. Trong một ví dụ, chỉ chế độ INTRA_ANGULAR18 có thể được sử dụng.

e. Trong một ví dụ, chỉ chế độ INTRA_ANGULAR50 có thể được sử dụng.

f. Trong một ví dụ, các chế độ CCLM có thể không được phép.

21. Đối với các khối sắc độ nhỏ có chiều rộng bằng M (ví dụ, M = 2) hoặc chiều cao bằng N (ví dụ, N = 2), các kiểu biến đổi có thể bị hạn chế khác với kiểu được phép đối với các khối sắc độ lớn.

a. Trong một ví dụ, việc bỏ qua phép biến đổi có thể được sử dụng.

b. Trong một ví dụ, chỉ phép biến đổi một chiều có thể được sử dụng.

c. Trong một ví dụ, các công cụ mã hóa mà trợ giúp các loại biến đổi không được phép.

i. Theo cách khác, tín hiệu của các công cụ mã hóa mà trợ giúp các loại biến đổi bị bỏ qua.

22. CIIP có thể được coi là `MODE_TYPE_INTRA`.

a. Trong một ví dụ, chế độ CIIP có thể được phép khi phân vùng cây nhị phân được sử dụng.

i. Trong một ví dụ, chế độ CIIP có thể được phép khi loại CU là `DUAL_TREE_CHROMA`.

b. Theo cách khác, CIIP có thể được coi là `MODE_TYPE_INTER`.

i. Trong một ví dụ, khi chiều rộng khối sắc độ bằng M (ví dụ, $M = 2$), chế độ CIIP có thể không được phép.

ii. Trong một ví dụ, khi chiều rộng khối sắc độ bằng M (ví dụ, $M = 2$), các chế độ dự báo trong đối với sắc độ trong CIIP có thể bị hạn chế ở chế độ dự báo trong đơn giản.

1. Trong một ví dụ, `INTRA_DC` có thể được sử dụng để dự báo trong khối sắc độ khi chiều rộng khối sắc độ bằng M (ví dụ, $M = 2$).

2. Trong một ví dụ, `INTRA_ANGULAR18` có thể được sử dụng để dự báo trong khối sắc độ khi chiều rộng khối sắc độ bằng M (ví dụ, $M = 2$).

3. Trong một ví dụ, `INTRA_ANGULAR50` có thể được sử dụng để dự báo trong sắc độ khi chiều rộng khối sắc độ bằng M (ví dụ, $M = 2$).

iii. Trong một ví dụ, các chế độ dự báo trong đối với sắc độ trong CIIP có thể bị hạn chế ở chế độ dự báo trong đơn giản.

1. Trong một ví dụ, `INTRA_DC` có thể được sử dụng để dự báo trong sắc độ.

2. Trong một ví dụ, chế độ `INTRA_ANGULAR18` có thể được sử dụng để dự báo trong sắc độ.

3. Trong một ví dụ, chế độ `INTRA_ANGULAR50` có thể được sử dụng để dự báo trong sắc độ.

23. Đối với các gạch đầu dòng nêu trên, các biến M và/hoặc N có thể được xác định trước hoặc được báo hiệu.

a. Trong một ví dụ, M và/hoặc N có thể phụ thuộc nhiều hơn vào các định dạng màu (ví dụ, 4: 2: 0, 4: 2: 2, 4: 4: 4).

24. `modeType` có thể được mở rộng để có nhiều loại hơn.

a. Trong một ví dụ, modeType có thể là MODE_TYPE_IBC. Khi modeType là MODE_TYPE_IBC, chế độ dự báo được suy ra là chế độ IBC.

i. Trong một ví dụ, pred_mode_flag không được báo hiệu trong trường hợp này.

ii. Trong một ví dụ, pred_mode_ibc_flag không được báo hiệu trong trường hợp này.

iii. Trong một ví dụ, pred_mode_plt_flag không được báo hiệu trong trường hợp này.

b. Trong một ví dụ, modeType có thể là MODE_TYPE_PALETTE. Khi modeType là MODE_TYPE_PALETTE, chế độ dự báo được suy ra là chế độ Bảng màu.

i. Trong một ví dụ, pred_mode_flag không được báo hiệu trong trường hợp này.

ii. Trong một ví dụ, pred_mode_ibc_flag không được báo hiệu trong trường hợp này.

iii. Trong một ví dụ, pred_mode_plt_flag không được báo hiệu trong trường hợp này.

c. Trong một ví dụ, mode_constraint_flag có thể được thay thế bằng một chỉ mục để chỉ ra một trong số các modeTypes được phép được sử dụng.

25. Trong một ví dụ, việc tách QT có được phép đối với khối có kích thước $W \times H$ hay không có thể phụ thuộc vào modeType kết hợp với kích thước.

a. Ví dụ, nếu modeType là MODE_TYPE_INTER và W ngang bằng với 8 và H ngang bằng với 8, việc tách QT không được phép.

26. Trong một ví dụ, việc tách TT theo chiều dọc có được phép đối với khối có kích thước $W \times H$ hay không có thể phụ thuộc vào modeType kết hợp với kích thước.

a. Ví dụ, nếu modeType là MODE_TYPE_INTER và W ngang bằng với 16 và H ngang bằng với 4 thì không cho phép tách TT theo chiều dọc.

27. Trong một ví dụ, việc tách TT theo chiều ngang có được phép đối với khối có kích thước $W \times H$ hay không có thể phụ thuộc vào modeType kết hợp với kích thước.

a. Ví dụ, nếu modeType là MODE_TYPE_INTER và W ngang bằng với 4 và H ngang bằng với 16, việc tách TT theo chiều ngang không được phép.

28. Trong một ví dụ, việc tách BT theo chiều dọc có được phép đối với khối có kích thước $W \times H$ hay không có thể phụ thuộc vào modeType kết hợp với kích thước.

a. Ví dụ, nếu modeType là MODE_TYPE_INTER và W ngang bằng với 8 và H ngang bằng với 4 thì không cho phép tách BT theo chiều dọc .

29. Trong một ví dụ, việc tách BT theo chiều ngang có được phép đối với khối có kích thước $W \times H$ hay không có thể phụ thuộc vào modeType kết hợp với kích thước.

a. Ví dụ, nếu modeType là MODE_TYPE_INTER và W ngang bằng với 4 và H ngang bằng với 8 thì không cho phép tách BT theo chiều ngang.

30. Trong một ví dụ, việc xem liệu chế độ dự báo của CU có được suy ra bởi modeType hay không có thể phụ thuộc vào các thành phần màu và/hoặc kích thước khối $W \times H$.

a. Ví dụ, chế độ dự báo của một CU sắc độ được suy ra bởi modeType; nhưng chế độ dự báo của một CU độ chói được báo hiệu thay vì được suy ra bởi modeType.

i. Ví dụ, chế độ dự báo của một CU độ chói được báo hiệu thay vì được suy ra bởi modeType nếu $W > 4$ hoặc $H > 4$.

31. Khi SCIPU được áp dụng, việc có và/hoặc cách báo hiệu thông tin liên quan đến QP delta của thành phần thứ nhất có thể phụ thuộc vào cách tách của thành phần thứ nhất.

a. Trong một ví dụ, khi SCIPU được áp dụng, việc có và/hoặc cách báo hiệu thông tin liên quan đến QP delta của thành phần thứ nhất có thể phụ thuộc vào cách tách của thành phần thứ nhất và được tách khỏi cách tách của thành phần thứ hai.

b. Trong một ví dụ, thành phần thứ nhất là độ chói và thành phần thứ hai là sắc độ.

c. Trong một ví dụ, thành phần thứ nhất là độ chói và thành phần thứ hai là sắc độ.

32. Biến bất kỳ liên quan đến QP delta của thành phần thứ nhất không thể được sửa đổi trong khi giải mã hoặc phân tích cú pháp của thành phần thứ hai khi áp dụng cấu trúc mã hóa cây nhị phân và/hoặc cây nhị phân cục bộ.

a. Trong một ví dụ, cấu trúc mã hóa cây nhị phân cục bộ có thể được sử dụng theo SCIPU.

b. Trong một ví dụ, thành phần thứ nhất là độ chói và thành phần thứ hai là sắc độ.

i. Biến có thể là IsCuQpDeltaCoded.

c. Trong một ví dụ, thành phần thứ nhất là sắc độ và thành phần thứ hai là độ chói.

i. Biến này có thể là IsCuChromaQpOffsetCoded.

33. Khi SCIPU được áp dụng, thông tin liên quan đến QP delta của một thành phần (như độ chói hoặc sắc độ) có thể được báo hiệu nhiều nhất một lần trong một vùng

cụ thể trong đó thành phần độ chói và thành phần sắc độ được yêu cầu dùng chung cùng một loại chế độ (như `MODE_TYPE_INTER` hoặc `MODE_TYPE_INTRA`).

A. Trong một ví dụ, vùng cụ thể được coi là một nhóm lượng tử hóa.

5. Các phương án

Các phần mới được thêm vào được đánh dấu bằng chữ in đậm và in nghiêng, và các phần đã xóa khỏi bản phác thảo làm việc VVC được đánh dấu bằng dấu ngoặc kép (ví dụ, `[[a]]` biểu thị việc xóa ký tự “a”). Các sửa đổi dựa trên bản phác thảo làm việc VVC mới nhất (JVET-O2001-v11)

5.1 Phương án làm ví dụ số 1

Phương án nêu dưới đây là về các ràng buộc đối với việc phân vùng khối nhỏ và chế độ dự báo chỉ được áp dụng đối với các định dạng sắc độ 4: 2: 0 và 4: 4: 4 (không áp dụng đối với các định dạng sắc độ 4: 0: 0 và 4: 4: 4).

7.4.9.4 Ngữ nghĩa của cây mã hóa

Biến `modeTypeCondition` được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, `modeTypeCondition` được đặt ngang bằng với 0
- loại lát cắt == I và `qtbtt_dual_tree_intra_flag` ngang bằng với 1
- `modeTypeCurr` không ngang bằng với `MODE_TYPE_ALL`
- `chroma_format_idc` ngang bằng với 0
- `chroma_format_idc` ngang bằng với 3
- Nếu không, nếu một trong số các điều kiện sau là đúng, `modeTypeCondition` được đặt ngang bằng với 1
- `cbWidth * cbHeight` ngang bằng với 64 và `split_qt_flag` ngang bằng với 1
- `cbWidth * cbHeight` ngang bằng với 64 và `MttSplitMode [x0] [y0] [mttDepth]` ngang bằng với `SPLIT_TT_HOR` hoặc `SPLIT_TT_VER`
- `cbWidth * cbHeight` ngang bằng với 32 và `MttSplitMode [x0] [y0] [mttDepth]` ngang bằng với `SPLIT_BT_HOR` hoặc `SPLIT_BT_VER`
- Nếu không, nếu một trong số các điều kiện sau là đúng, `modeTypeCondition` được đặt ngang bằng với 1 + (loại lát cắt != I? 1: 0)
- `cbWidth * cbHeight` ngang bằng với 64 và `MttSplitMode [x0] [y0] [mttDepth]` ngang bằng với `SPLIT_BT_HOR` hoặc `SPLIT_BT_VER`

- cbWidth * cbHeight ngang bằng với 128 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Nếu không, modeTypeCondition được đặt ngang bằng với 0

5.2 Phương án làm ví dụ # 2

Phương án nêu dưới đây là về việc báo hiệu của cờ chế độ bảng màu không phụ thuộc vào modeType.

7.3.8.5 Cú pháp của đơn vị mã hóa

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Bộ mô tả
chType = treeType == DUAL_TREE_CHROMA? 1 : 0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !(((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)

<pre> if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[x0][y0] == 0 [[&& modeType != MODE_INTER]]) </pre>	
pred_mode_plt_flag	ae(v)

5.3 Phương án làm ví dụ # 3

Phương án nêu dưới đây là về cờ chế độ dự báo IBC được suy ra tùy thuộc vào cờ CU SKIP và modeType.

pred_mode_ibc_flag ngang bằng với 1 quy định rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo IBC. pred_mode_ibc_flag ngang bằng với 0 quy định rằng đơn vị mã hóa hiện tại không được mã hóa trong chế độ dự báo IBC.

Khi không có pred_mode_ibc_flag, nó được suy ra như sau:

- Nếu cu_skip_flag [x0] [y0] ngang bằng với 1, và cbWidthH ngang bằng với 4 và cbHeight ngang bằng với 4, thì suy ra pred_mode_ibc_flag ngang bằng với 1.
- Nếu không, nếu cả cbWidth và cbHeight đều ngang bằng với 128, thì suy ra pred_mode_ibc_flag ngang bằng với 0.
- Nếu không, nếu cu_skip_flag [x0] [y0] ngang bằng với 1 và modeType là MODE_TYPE_INTRA, thì suy ra pred_mode_ibc_flag ngang bằng với 1.
- Nếu không, nếu modeType là MODE_TYPE_INTER, thì suy ra pred_mode_ibc_flag ngang bằng với 0.
- Nếu không, nếu treeType ngang bằng với DUAL_TREE_CHROMA, thì pred_mode_ibc_flag được suy ra ngang bằng với 0.
- Nếu không, pred_mode_ibc_flag được coi là bằng trị số của sps_ibc_enabled_flag khi giải mã lát cắt I và 0 khi lần lượt giải mã lát cắt P hoặc B.

Khi `pred_mode_ibc_flag` ngang bằng với 1, biến `CuPredMode[chType][x][y]` được đặt ngang bằng với `MODE_IBC` đối với $x = x0..x0 + cbWidth - 1$ và $y = y0..y0 + cbHeight - 1$.

5.4 Phương án làm ví dụ # 4

Phương án nêu dưới đây là về việc báo hiệu của cờ chế độ dự báo IBC phụ thuộc vào `MODE_TYPE_INTRA` và/hoặc chế độ IBC được phép đối với các khối sắc độ bất kể nó có kích thước khối nhỏ hay không.

7.3.8.5 Cú pháp của đơn vị mã hóa

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	Bộ mô tả
<code>chType = treeType == DUAL_TREE_CHROMA ? 1 : 0</code>	
<code>if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {</code>	
<code>if(treeType != DUAL_TREE_CHROMA && !(((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA && !sps_ibc_enabled_flag))</code>	
<code>cu_skip_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)</code>	
<code>pred_mode_flag</code>	<code>ae(v)</code>
<code>if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (modeType == MODE_TYPE_INTRA && cu_skip_flag[x0][y0] == 0) (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)</code>	

pred_mode_ibc_flag	ae(v)
--------------------	-------

5.5 Phương án làm ví dụ # 5

Phương án nêu dưới đây là về việc áp dụng các ràng buộc khối bên trong khác nhau đối với các định dạng màu 4: 2: 0 và 4: 2: 2.

7.4.9.4 Ngữ nghĩa của cây mã hóa

Biến modeTypeCondition được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 0
- loại lát cắt == I và qtbtt_dual_tree_intra_flag ngang bằng với 1
- modeTypeCurr không ngang bằng với MODE_TYPE_ALL
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và split_qt_flag ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- cbWidth * cbHeight ngang bằng với 32 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1 + (loại lát cắt! = I? 1: 0)
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER và *chroma_format_idc* ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 128 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER và *chroma_format_idc* ngang bằng với 1
- *cbWidth* ngang bằng với 8 và *cbHeight* ngang bằng với 8 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_VER và *chroma_format_idc* ngang bằng với 2
- *cbWidth* ngang bằng với 4 và *cbHeight* ngang bằng với 16 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR và *chroma_format_idc* ngang bằng với 2

- *cbWidthH* ngang bằng với 16 và *cbHeight* ngang bằng với 8 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_TT_VER* và *chroma_format_idc* ngang bằng với 2
- *cbWidthH* ngang bằng với 4 và *cbHeight* ngang bằng với 32 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_TT_HOR* và *chroma_format_idc* ngang bằng với 2
- Nếu không, *modeTypeCondition* được đặt ngang bằng với 0

5.6 Phương án làm ví dụ # 6

Phương án nêu dưới đây là về việc không cho phép trong các khối sắc độ 2xN trong cây đơn phân.

7.4.9.4 Ngữ nghĩa của cây mã hóa

Biến *modeTypeCondition* được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, *modeTypeCondition* được đặt ngang bằng với 0
- loại lát cắt == I và *qtbtt_dual_tree_intra_flag* ngang bằng với 1
- *modeTypeCurr* không ngang bằng với *MODE_TYPE_ALL*
- Nếu không, nếu một trong số các điều kiện sau là đúng, *modeTypeCondition* được đặt ngang bằng với 1
- *cbWidth * cbHeight* ngang bằng với 64 và *split_qt_flag* ngang bằng với 1
- *cbWidth * cbHeight* ngang bằng với 64 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_TT_HOR* hoặc *SPLIT_TT_VER*
- *cbWidth * cbHeight* ngang bằng với 32 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_BT_HOR* hoặc *SPLIT_BT_VER*
- Nếu không, nếu một trong số các điều kiện sau là đúng, *modeTypeCondition* được đặt ngang bằng với 1 + (loại lát cắt! = I? 1: 0)
- *cbWidth * cbHeight* ngang bằng với 64 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_BT_HOR* hoặc *SPLIT_BT_VER*
- *cbWidth * cbHeight* ngang bằng với 128 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_TT_HOR* hoặc *SPLIT_TT_VER*
- *cbWidthH* ngang bằng với 8 và *MttSplitMode* [x0] [y0] [mttDepth] ngang bằng với *SPLIT_BT_VER*

- cbWidthH ngang bằng với 16 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_VER

- Nếu không, modeTypeCondition được đặt ngang bằng với 0

5.7 Phương án làm ví dụ # 7

Phương án nêu dưới đây là về việc không cho phép trong các khối sắc độ 2xN trong cây nhị phân.

6.4.2 Quy trình tách nhị phân được phép

Biến allowBtSplit được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE:

- cbSize nhỏ hơn hoặc ngang bằng với MinBtSizeY
- cbWidth lớn hơn maxBtSize
- cbHeight lớn hơn maxBtSize
- mttDepth lớn hơn hoặc ngang bằng với maxMttDepth
- treeType ngang bằng với DUAL_TREE_CHROMA và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ nhỏ hơn hoặc [ngang bằng với 16
- btSplit ngang bằng với SPLIT_BT_VER và treeType ngang bằng với DUAL_TREE_CHROMA và $(cbWidth / SubWidthC)$ nhỏ hơn hoặc ngang bằng với 4
- treeType ngang bằng với DUAL_TREE_CHROMA và modeType là MODE_TYPE_INTRA

6.4.3 Quy trình tách tam phân được phép

Biến allowTtSplit được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, allowTtSplit được đặt ngang bằng với FALSE:

- cbSize nhỏ hơn hoặc ngang bằng với $2 * MinTtSizeY$
- cbWidth lớn hơn Min (MaxTbSizeY, maxTtSize)
- cbHeight lớn hơn Min (MaxTbSizeY, maxTtSize)
- mttDepth lớn hơn hoặc ngang bằng với maxMttDepth
- $x0 + cbWidth$ lớn hơn pic_width_in_luma_samples
- $y0 + cbHeight$ lớn hơn pic_height_in_luma_samples

- treeType ngang bằng với DUAL_TREE_CHROMA và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ nhỏ hơn hoặc ngang bằng với 32
- btSplit ngang bằng với SPLIT_TT_VER và treeType ngang bằng với DUAL_TREE_CHROMA và $(cbWidth / SubWidthC)$ nhỏ hơn hoặc ngang bằng với 8
- treeType ngang bằng với DUAL_TREE_CHROMA và modeType là MODE_TYPE_INTRA
- Nếu không, allowTtSplit được đặt ngang bằng với TRUE.

5.8 Phương án làm ví dụ # 8

Phương án nêu dưới đây là về cách kích hoạt MODE_IBC đối với các khối sắc độ SCIPU.

7.3.8.5Cú pháp của đơn vị mã hóa

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Bộ mô tả
chType = treeType == DUAL_TREE_CHROMA? 1 : 0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
pred_mode_flag	ae(v)

<pre> if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (modeType == MODE_TYPE_INTRA && cu_skip_flag[x0][y0] == 0) (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0))))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && !(modeType == MODE_TYPE_ALL && treeType == DUAL_TREE_CHROMA) [[treeType != DUAL_TREE_CHROMA]]) </pre>	
pred_mode_ibc_flag	ae(v)

5.9 Phương án làm ví dụ # 9 về việc không cho phép phân vùng khối khi modeType là MODE_TYPE_INTER (giải pháp 1)

6.4.2 Quy trình tách nhị phân được phép

Biến allowBtSplit được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, allowBtSplit được đặt ngang bằng với FALSE:

- cbSize nhỏ hơn hoặc ngang bằng với MinBtSizeY
- cbWidth lớn hơn maxBtSize
- cbHeight lớn hơn maxBtSize
- mttDepth lớn hơn hoặc ngang bằng với maxMttDepth
- treeType ngang bằng với DUAL_TREE_CHROMA và (cbWidth / SubWidthC) * (cbHeight / SubHeightC) nhỏ hơn hoặc [ngang bằng với 16
- modeType là MODE_TYPE_INTER và cbWidth * cbHeight nhỏ hơn hoặc ngang bằng với 32
- treeType ngang bằng với DUAL_TREE_CHROMA và modeType là MODE_TYPE_INTRA

...

6.4.3 Quy trình tách tam phân được phép

Biến `allowTtSplit` được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, `allowTtSplit` được đặt ngang bằng với `FALSE`:
- `cbSize` nhỏ hơn hoặc ngang bằng với $2 * \text{MinTtSizeY}$
- `cbWidth` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- `cbHeight` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- `mttDepth` lớn hơn hoặc ngang bằng với `maxMttDepth`
- $x0 + \text{cbWidth}$ lớn hơn `pic_width_in_luma_samples`
- $y0 + \text{cbHeight}$ lớn hơn `pic_height_in_luma_samples`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ nhỏ hơn hoặc ngang bằng với 32
- `modeType` là `MODE_TYPE_INTER` và $\text{cbWidth} * \text{cbHeight}$ nhỏ hơn hoặc ngang bằng với 64
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và `modeType` là `MODE_TYPE_INTRA`
- Nếu không, `allowTtSplit` được đặt ngang bằng với `TRUE`.

5.10 Phương án làm ví dụ # 10 về việc không cho phép phân vùng khối khi `modeType` là `MODE_TYPE_INTER` (giải pháp 2)

6.4.2 Quy trình tách nhị phân được phép

Biến `allowBtSplit` được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, `allowBtSplit` được đặt ngang bằng với `FALSE`:
- `cbSize` nhỏ hơn hoặc ngang bằng với `MinBtSizeY`
- `cbWidth` lớn hơn `maxBtSize`
- `cbHeight` lớn hơn `maxBtSize`
- `mttDepth` lớn hơn hoặc ngang bằng với `maxMttDepth`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ nhỏ hơn hoặc [ngang bằng với 16
- `modeType` là `MODE_TYPE_INTER`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và `modeType` là `MODE_TYPE_INTRA`

...

6.4.3 Quy trình tách tam phân được phép

Biến `allowTtSplit` được tạo ra như sau:

- Nếu một hoặc nhiều điều kiện sau là đúng, `allowTtSplit` được đặt ngang bằng với `FALSE`:
- `cbSize` nhỏ hơn hoặc ngang bằng với $2 * \text{MinTtSizeY}$
- `cbWidth` lớn hơn `Min (MaxTbSizeY, maxTtSize)`
- `cbHeight` lớn hơn `Min (MaxTbSizeY, maxTtSize)`
- `mttDepth` lớn hơn hoặc ngang bằng với `maxMttDepth`
- $x0 + cbWidth$ lớn hơn `pic_width_in_luma_samples`
- $y0 + cbHeight$ lớn hơn `pic_height_in_luma_samples`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và $(cbWidth / \text{SubWidthC}) * (cbHeight / \text{SubHeightC})$ nhỏ hơn hoặc ngang bằng với 32
- `modeType` là `MODE_TYPE_INTER`
- `treeType` ngang bằng với `DUAL_TREE_CHROMA` và `modeType` là `MODE_TYPE_INTRA`
- Nếu không, `allowTtSplit` được đặt ngang bằng với `TRUE`.

5.11 Phương án làm ví dụ # 11

Phương án nêu dưới đây là về các ràng buộc trong việc tách thêm cây mã hóa khi `MODE_TYPE_INTER` được tạo ra.

7.3.8.4 Cú pháp của cây mã hóa

<code>coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mt tDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {</code>	Bộ mô tả
...	
<code>treeType = (modeType==MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : t reeTypeCurr</code>	
<code>if(modeType == MODE_TYPE_INTER)</code>	
<code>mttDepth = max(mttDepth, maxMttDepth - 1)</code>	
<code>if(!split_qt_flag) {</code>	
...	

5.12 Phương án làm ví dụ # 12

Phương án nêu dưới đây là về các ràng buộc đối với việc phân vùng khối nhỏ và chế độ dự báo không được áp dụng khi chế độ bảng màu được kích hoạt.

7.4.9.4 Ngữ nghĩa của cây mã hóa

Biến modeTypeCondition được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 0
- loại lát cắt = I và qtbtt_dual_tree_intra_flag ngang bằng với 1
- modeTypeCurr không ngang bằng với MODE_TYPE_ALL
- *sps_palette_enabled_flag* ngang bằng với 1
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và split_qt_flag ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- cbWidth * cbHeight ngang bằng với 32 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1 + (loại lát cắt! = I? 1: 0)
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER
- cbWidth * cbHeight ngang bằng với 128 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Nếu không, modeTypeCondition được đặt ngang bằng với 0

5.13 Phương án làm ví dụ # 13

Phương án nêu dưới đây là về các ràng buộc về trong khối sắc độ nhỏ đối với các định dạng màu 4: 2: 2.

7.4.9.4 Ngữ nghĩa của cây mã hóa

Biến modeTypeCondition được tạo ra như sau:

- Nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 0
- loại lát cắt = I và qtbtt_dual_tree_intra_flag ngang bằng với 1

- modeTypeCurr không ngang bằng với MODE_TYPE_ALL
- Nếu không, nếu một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và split_qt_flag ngang bằng với 1
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- cbWidth * cbHeight ngang bằng với 32 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Nếu không, nếu chroma_format_idc ngang bằng với 1 và một trong số các điều kiện sau là đúng, modeTypeCondition được đặt ngang bằng với 1 + (loại lát cắt! = I? 1: 0)
- cbWidth * cbHeight ngang bằng với 64 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_BT_HOR hoặc SPLIT_BT_VER
- cbWidth * cbHeight ngang bằng với 128 và MttSplitMode [x0] [y0] [mttDepth] ngang bằng với SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Nếu không, modeTypeCondition được đặt ngang bằng với 0

5.14 Ví dụ số 1 về tín hiệu QP delta trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mt tDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Bộ mô tả
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= cu_qp_delta_subdiv) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= cu_chroma_qp_offset_subdiv)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)

if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
}	
if(modeTypeCondition == 1)	
modeType = MODE_TYPE_INTRA	
else if(modeTypeCondition == 2) {	
mode_constraint_flag	ae(v)
modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE _INTER	
} else {	
modeType = modeTypeCurr	
}	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if(!split_qt_flag) {	

if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
coding_tree (x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(x1 < pic_width_in_luma_samples)	
coding_tree (x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
coding_tree (x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < pic_height_in_luma_samples)	
coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	

<code>} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {</code>	
<code> x1 = x0 + (cbWidth / 4)</code>	
<code> x2 = x0 + (3 * cbWidth / 4)</code>	
<code> qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code> qgOnCnext = qgOnC && (cbSubdiv + 2 <=</code> <code> cu_chroma_qp_offset_subdiv)</code>	
<code> coding_tree (x0, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree (x1, y0, cbWidth / 2, cbHeight, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 1,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree (x2, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	

<code>} else { /* SPLIT_TT_HOR */</code>	
<code> y1 = y0 + (cbHeight / 4)</code>	
<code> y2 = y0 + (3 * cbHeight / 4)</code>	
<code> qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code> qgOnCnext = qgOnC && (cbSubdiv + 2 <=</code> <code>cu_chroma_qp_offset_subdiv)</code>	
<code> coding_tree (x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 1,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree (x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code>}</code>	
<code>} else {</code>	
<code> x1 = x0 + (cbWidth / 2)</code>	
<code> y1 = y0 + (cbHeight / 2)</code>	
<code> coding_tree (x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code> if(x1 < pic_width_in_luma_samples)</code>	
<code> coding_tree (x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code> cbSubdiv + 2,</code> <code> cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code> if(y1 < pic_height_in_luma_samples)</code>	

coding_tree (x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)	
if(y1 < pic_height_in_luma_samples && x1 < pic_width_in_luma_samples)	
coding_tree (x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree (x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, 0, 0 DUAL_TREE_CHROMA , modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , modeTypeC urr)	
}	

5.15 Ví dụ số 2 về tín hiệu QP delta trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mt tDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Bộ mô tả
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= cu_qp_delta_subdiv) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= cu_chroma_qp_offset_subdiv)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)

if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
}	
if(modeTypeCondition == 1)	
modeType = MODE_TYPE_INTRA	
else if(modeTypeCondition == 2) {	
mode_constraint_flag	ae(v)
modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE _INTER	
} else {	
modeType = modeTypeCurr	
}	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if(!split_qt_flag) {	

if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
coding_tree (x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(x1 < pic_width_in_luma_samples)	
coding_tree (x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
cây mã hóa(x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < pic_height_in_luma_samples)	
coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	

<code>} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {</code>	
<code> x1 = x0 + (cbWidth / 4)</code>	
<code> x2 = x0 + (3 * cbWidth / 4)</code>	
<code> qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code> qgOnCnext = qgOnC && (cbSubdiv + 2 <=</code> <code>cu_chroma_qp_offset_subdiv)</code>	
<code> coding_tree (x0, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 2,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree (x1, y0, cbWidth / 2, cbHeight, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 1,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree (x2, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 2,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	

<code>} else { /* SPLIT_TT_HOR */</code>	
<code>y1 = y0 + (cbHeight / 4)</code>	
<code>y2 = y0 + (3 * cbHeight / 4)</code>	
<code>qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code>qgOnCnext = qgOnC && (cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv)</code>	
<code>coding_tree (x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code>coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code>coding_tree (x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code>}</code>	
<code>} else {</code>	
<code>x1 = x0 + (cbWidth / 2)</code>	
<code>y1 = y0 + (cbHeight / 2)</code>	
<code>coding_tree (x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code>if(x1 < pic_width_in_luma_samples)</code>	
<code>coding_tree (x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code>if(y1 < pic_height_in_luma_samples)</code>	

coding_tree (x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)	
if(y1 < pic_height_in_luma_samples && x1 < pic_width_in_luma_samples)	
coding_tree (x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree (x0, y0, cbWidth, cbHeight, qgOnY, 0, cbSubdiv, cqtDepth, mttDepth, 0, 0 DUAL_TREE_CHROMA , modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , modeTypeC urr)	
}	

5.16 Ví dụ số 3 về tín hiệu QP delta trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mtDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Bộ mô tả
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= cu_qp_delta_subdiv && treeTypeCurr != DUAL_TREE_CHROMA) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= cu_chroma_qp_offset_subdiv && && treeTypeCurr != DUAL_TREE_LUMA)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)

if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
mtt_split_cu_binary_flag	ae(v)
}	
if(modeTypeCondition == 1)	
modeType = MODE_TYPE_INTRA	
else if(modeTypeCondition == 2) {	
mode_constraint_flag	ae(v)
modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE _INTER	
} else {	
modeType = modeTypeCurr	
}	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if(!split_qt_flag) {	

if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
coding_tree (x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(x1 < pic_width_in_luma_samples)	
coding_tree (x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
coding_tree (x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < pic_height_in_luma_samples)	
cây mã hóa(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	

} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
<i>qgOnYnext</i> = <i>qgOnY</i> && (<i>cbSubdiv</i> + 2 <= <i>cu_qp_delta_subdiv</i>)	
<i>qgOnCnext</i> = <i>qgOnC</i> && (<i>cbSubdiv</i> + 2 <= <i>cu_chroma_qp_offset_subdiv</i>)	
coding_tree (x0, y0, cbWidth / 4, cbHeight, <i>qgOnYnext</i> , <i>qgOnCnext</i> , cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
coding_tree (x1, y0, cbWidth / 2, cbHeight, <i>qgOnYnext</i> , <i>qgOnCnext</i> , cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
coding_tree (x2, y0, cbWidth / 4, cbHeight, <i>qgOnYnext</i> , <i>qgOnCnext</i> , cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)	

<code>} else { /* SPLIT_TT_HOR */</code>	
<code> y1 = y0 + (cbHeight / 4)</code>	
<code> y2 = y0 + (3 * cbHeight / 4)</code>	
<code> qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code> qgOnCnext = qgOnC && (cbSubdiv + 2 <=</code> <code>cu_chroma_qp_offset_subdiv)</code>	
<code> coding_tree (x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code> coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 1,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code> coding_tree (x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext,</code> <code>cbSubdiv + 2,</code> <code> cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code>}</code>	
<code>} else {</code>	
<code> x1 = x0 + (cbWidth / 2)</code>	
<code> y1 = y0 + (cbHeight / 2)</code>	
<code> coding_tree (x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code>cbSubdiv + 2,</code> <code> cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code> if(x1 < pic_width_in_luma_samples)</code>	
<code> coding_tree (x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,</code> <code>cbSubdiv + 2,</code> <code> cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code> if(y1 < pic_height_in_luma_samples)</code>	

coding_tree (x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)	
if(y1 < pic_height_in_luma_samples && x1 < pic_width_in_luma_samples)	
coding_tree (x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree (x0, y0, cbWidth, cbHeight, qgOnY, 0, cbSubdiv, cqtDepth, mttDepth, 0, 0 DUAL_TREE_CHROMA , modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , modeTypeC urr)	
}	

5.17 Ví dụ # 4 về báo hiệu QP delta trong SCIPU

coding_tree (x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, dept hOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Bộ mô tả
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor allowSplitQT) &&(x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= cu_qp_delta_subdiv) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= cu_chroma_qp_offset_subdiv)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag) {	
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && allowSplitQT)	
split_qt_flag	ae(v)
if(!split_qt_flag) {	
if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer))	
mtt_split_cu_vertical_flag	ae(v)

<pre> if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag)) </pre>	
<pre> mtt_split_cu_binary_flag </pre>	ae(v)
<pre> } </pre>	
<pre> if(modeTypeCondition == 1) </pre>	
<pre> modeType = MODE_TYPE_INTRA </pre>	
<pre> else if(modeTypeCondition == 2) { </pre>	
<pre> mode_constraint_flag </pre>	ae(v)
<pre> modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE _INTER </pre>	
<pre> } else { </pre>	
<pre> modeType = modeTypeCurr </pre>	
<pre> } </pre>	
<pre> if(modeTypeCondition > 0) </pre>	
<pre> qgOnY = qgOnC = 0 </pre>	
<pre> treeType = (modeType MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr </pre>	==
<pre> if(!split_qt_flag) { </pre>	

if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
coding_tree (x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(x1 < pic_width_in_luma_samples)	
coding_tree (x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
coding_tree (x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
if(y1 < pic_height_in_luma_samples)	
coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	

} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
qgOnY = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)	
qgOnC = qgOnC && (cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv)	
coding_tree (x0, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
coding_tree (x1, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
coding_tree (x2, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)	

<code>} else { /* SPLIT_TT_HOR */</code>	
<code>y1 = y0 + (cbHeight / 4)</code>	
<code>y2 = y0 + (3 * cbHeight / 4)</code>	
<code>qgOnY = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)</code>	
<code>qgOnC = qgOnC && (cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv)</code>	
<code>coding_tree (x0, y0, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)</code>	
<code>coding_tree (x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)</code>	
<code>coding_tree (x0, y2, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)</code>	
<code>}</code>	
<code>} else {</code>	
<code>x1 = x0 + (cbWidth / 2)</code>	
<code>y1 = y0 + (cbHeight / 2)</code>	
<code>coding_tree (x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)</code>	
<code>if(x1 < pic_width_in_luma_samples)</code>	
<code>coding_tree (x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)</code>	
<code>if(y1 < pic_height_in_luma_samples)</code>	

coding_tree (x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType)	
if(y1 < pic_height_in_luma_samples && x1 < pic_width_in_luma_samples)	
coding_tree (x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType)	
}	
if(modeTypeCur == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA) {	
coding_tree (x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, 0, 0, DUAL_TREE_CHROMA , modeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , modeTypeCurr)	
}	

FIG.17 là sơ đồ khối của thiết bị xử lý video 1700. Thiết bị 1700 có thể được sử dụng để thực hiện một hoặc nhiều phương pháp được mô tả ở đây. Thiết bị 1700 có thể được tích hợp trong điện thoại thông minh, máy tính bảng, máy tính, bộ thu Internet vạn vật (IoT), v.v.. Thiết bị 1700 có thể bao gồm một hoặc nhiều bộ xử lý 1702, một hoặc nhiều bộ nhớ 1704 và phần cứng xử lý video 1706. (Các) bộ xử lý 1702 có thể được tạo cấu hình để thực hiện một hoặc nhiều phương pháp được mô tả trong Bản mô tả sáng chế này. Bộ nhớ (bộ nhớ) 1704 có thể được sử dụng để lưu trữ dữ liệu và mã được sử dụng để thực hiện các phương pháp và kỹ thuật được mô tả ở đây. Phần cứng xử lý video 1706 có thể được sử dụng để thực hiện, trong mạch phần cứng, một số kỹ thuật được mô tả trong Bản mô tả sáng chế này. Theo một số

phương án, phần cứng 1706 ít nhất có thể nằm trong bộ xử lý 1702, ví dụ, bộ đồng xử lý đồ họa.

FIG.18 là sơ đồ của phương pháp 1800 xử lý video. Phương pháp 1800 bao gồm các bước :phân tích cú pháp (1802), để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của vùng video,phần biểu diễn đã được mã hóa này theo quy tắc cú pháp xác định mối quan hệ giữa kích thước khối sắc độ và định dạng màu của vùng video; và thực hiện (1804) việc chuyển đổi bằng cách thực hiện việc phân tích cú pháp theo quy tắc cú pháp.

Công nghệ đã được bộc lộ theo một số phương án bao gồm việc đưa ra phán quyết hoặc quyết định kích hoạt chế độ hoặc công cụ xử lý video. Trong một ví dụ, khi công cụ hoặc chế độ xử lý video được kích hoạt, bộ mã hóa sẽ sử dụng hoặc thực hiện công cụ hoặc chế độ xử lý khối video, nhưng không nhất thiết phải sửa đổi dòng bit thu được dựa trên việc sử dụng công cụ hoặc chế độ đó. Tức là, một chuyển đổi từ khối video sang biểu diễn dòng bit của video này sẽ sử dụng công cụ hoặc chế độ xử lý video khi nó được kích hoạt dựa trên phán quyết hoặc quyết định. Trong một ví dụ khác, khi công cụ hoặc chế độ xử lý video được kích hoạt, bộ giải mã sẽ xử lý dòng bit với sự hiểu biết rằng dòng bit đã được sửa đổi dựa trên công cụ hoặc chế độ xử lý video. Tức là, việc chuyển đổi từ phần biểu diễn dòng bit của video này thành khối video cần được thực hiện bằng cách sử dụng công cụ hoặc chế độ xử lý video đã được kích hoạt dựa trên phán quyết hoặc quyết định.

Công nghệ đã được bộc lộ theo một số phương án bao gồm việc đưa ra phán quyết hoặc quyết định tắt chế độ hoặc công cụ xử lý video. Trong một ví dụ, khi công cụ hoặc chế độ xử lý video bị vô hiệu hóa, bộ mã hóa sẽ không sử dụng công cụ hoặc chế độ trong khi chuyển đổi khối video thành phần biểu diễn dòng bit của video này. Trong một ví dụ khác, khi công cụ hoặc chế độ xử lý video bị vô hiệu hóa, bộ giải mã sẽ xử lý dòng bit với thông tin rằng dòng bit chưa được sửa đổi bằng cách sử dụng công cụ hoặc chế độ xử lý video đã bị vô hiệu hóa dựa trên phán quyết hoặc quyết định.

Trong Bản mô tả sáng chế này, cụm từ “xử lý video” có thể được dùng để chỉ việc mã hóa video, giải mã video, nén video hoặc giải nén video. Ví dụ, các thuật toán nén video này có thể được áp dụng trong khi chuyển đổi từ phần biểu diễn

pixel của video này sang biểu diễn dòng bit tương ứng hoặc ngược lại. Ví dụ, việc biểu diễn dòng bit của khối video hiện tại có thể tương ứng với các bit nằm cùng vị trí hoặc trải rộng ở các vị trí khác nhau trong dòng bit, như được định nghĩa bởi cú pháp này. Ví dụ, một macroblock có thể được mã hóa theo các trị số còn lại của lỗi đã được chuyển đổi và mã hóa và cũng sử dụng các bit trong tiêu đề và các trường khác trong dòng bit này.

Tập hợp các mục thứ nhất sau có thể được thực hiện theo một số phương án.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 1 của phần trước.

1. Phương pháp xử lý video bao gồm các bước: phân tích cú pháp, để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của vùng video, phần biểu diễn đã được mã hóa này theo quy tắc cú pháp xác định mối quan hệ giữa kích thước của khối sắc độ và định dạng màu của vùng video; và thực hiện việc chuyển đổi bằng cách thực hiện việc phân tích cú pháp theo quy tắc cú pháp.

2. Phương pháp theo mục 1, trong đó định dạng màu 4: 4: 4 và trong đó quy tắc cú pháp quy định rằng khối sắc độ phải chịu cùng một ràng buộc về kích thước như đối với khối độ chói.

3. Phương pháp theo mục 1, trong đó định dạng màu 4: 2: 2 và trong đó quy tắc cú pháp quy định rằng khối sắc độ phải chịu cùng một ràng buộc về kích thước như đối với định dạng màu 4: 2: 0.

4. Phương pháp theo mục bất kỳ trong số các mục 1-3, trong đó cú pháp quy định rằng chế độ dự báo và phân vùng khối nhỏ được sử dụng theo cách thức phụ thuộc vào định dạng sắc độ.

5. Phương pháp theo mục 1, trong đó quy tắc cú pháp quy định rằng tính năng kích thước được phép nhỏ nhất được kích hoạt để chuyển đổi vùng video này dựa trên định dạng màu của vùng video này.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 2 của phần trước.

6. Phương pháp xử lý video bao gồm các bước: xác định, dựa trên thuộc tính của video này và định dạng sắc độ của video này, chế độ mã hóa của nút cây mã hóa của video này; và thực hiện việc chuyển đổi giữa phần biểu diễn đã được mã

hóa của video này và khối video của nút cây mã hóa bằng cách sử dụng chế độ mã hóa đã được xác định.

7. Phương pháp theo mục 6, trong đó chế độ mã hóa được xác định là `MODE_TYPE_ALL` đối với định dạng sắc độ là 4: 2: 2, `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER` đối với định dạng sắc độ là 4: 2: 0 trong trường hợp thuộc tính là:

- i. nút mã hóa là một nút cây mã hóa $M \times N$ với việc tách cây nhị phân theo chiều ngang;
- ii. nút mã hóa là một nút cây mã hóa $M \times N$ với việc tách cây nhị phân theo chiều dọc;
- iii. nút mã hóa là một nút cây mã hóa $M \times N$ với việc tách cây tam phân theo chiều ngang; hoặc là
- iv. nút mã hóa là một nút cây mã hóa $M \times N$ với việc tách cây tam phân theo chiều dọc.

8. Phương pháp theo mục 7, trong đó $M = 8$, hoặc 16 hoặc 32 và $N = 4$ hoặc 8 hoặc 16.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 12 của phần trước.

9. Phương pháp xử lý video bao gồm các bước: xác định, dựa trên quy tắc, xem liệu một kích thước nhất định của khối sắc độ có được phép trong vùng video của video hay không; và thực hiện việc chuyển đổi giữa vùng video này và phần biểu diễn đã được mã hóa của vùng video này dựa trên việc xác định này.

10. Phương pháp theo mục 9, trong đó quy tắc này quy định rằng khối sắc độ $2 \times N$ không được phép do vùng video này có phân vùng cây nhị phân.

11. Phương pháp theo mục 9, trong đó quy tắc này quy định rằng khối sắc độ $2 \times N$ không được phép do vùng video này có một phân vùng cây đơn phân.

12. Phương pháp theo mục 10 hoặc 11, trong đó $N \leq 64$.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục 13, 14 và 15 của phần trước

13. Phương pháp xử lý video bao gồm các bước: xác định, dựa trên quy tắc cho phép sử dụng chế độ mã hóa đối với điều kiện video, rằng chế độ mã hóa được

phép đối với một vùng video; và thực hiện việc chuyển đổi giữa phần biểu diễn đã được mã hóa của pixel trong vùng video này và pixel của vùng video này dựa trên việc xác định này.

14. Phương pháp theo mục 13, trong đó điều kiện video này là kích thước khối và trong đó quy tắc cho phép sử dụng chế độ sao chép trong khối đối với các kích thước khối nhỏ của khối độ chói.

15. Phương pháp theo mục 14, trong đó kích thước khối nhỏ bao gồm kích thước khối sắc độ 8x4, 8x8, 16x4 hoặc 4xN.

16. Phương pháp theo mục 13, trong đó quy tắc cho phép sử dụng chế độ sao chép trong khối để chuyển đổi vùng video bằng cách sử dụng chế độ mã hóa `MODE_TYPE_INTER`.

Phương pháp theo mục 13, trong đó quy tắc cho phép sử dụng chế độ mã hóa bảng màu để chuyển đổi vùng video là chế độ mã hóa `MODE_TYPE_INTER`.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục 16, 17, 18 của phần trước.

18. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa khối video của video và phần biểu diễn đã được mã hóa của khối video này bằng cách sử dụng chế độ mã hóa video, trong đó phần tử cú pháp báo hiệu chế độ mã hóa được đưa vào phần biểu diễn đã được mã hóa dựa trên quy tắc.

19. Phương pháp theo mục 18, trong đó chế độ mã hóa video này là chế độ mã hóa trong khối và trong đó quy tắc này quy định việc sử dụng một loại chế độ mã hóa video để kiểm soát việc đưa phần tử cú pháp vào phần biểu diễn đã được mã hóa.

20. Phương pháp theo mục 19, trong đó quy tắc này quy định việc báo hiệu rõ ràng khối không phải SKIP.

21. Phương pháp theo mục 18, trong đó quy tắc này quy định việc báo hiệu ngầm cho cờ sao chép trong khối dựa trên cờ bỏ qua và kiểu chế độ của khối video này.

22. Phương pháp theo mục 18, trong đó chế độ mã hóa là chế độ mã hóa bảng màu và trong đó quy tắc này quy định việc đưa vào theo cách chọn lọc chỉ báo mã hóa bảng màu dựa trên loại chế độ của khối video này.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 21 của phần trước.

23. Phương pháp xử lý video bao gồm các bước: xác định, do khối sắc độ có kích thước nhỏ hơn kích thước ngưỡng, rằng loại biến đổi dùng trong khi chuyển đổi giữa khối sắc độ và phân biểu diễn đã được mã hóa của khối sắc độ là khác với kiểu chuyển đổi dùng để chuyển đổi khối độ chói tương ứng; và thực hiện việc chuyển đổi dựa trên việc xác định này.

24. Phương pháp theo mục 23, trong đó kích thước ngưỡng là $M \times N$, trong đó M là 2.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 22 của phần trước.

25. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 24, trong đó việc chuyển đổi sử dụng chế độ dự báo trong và kết hợp dưới dạng chế độ `MODE_TYPE_INTRA`.

26. Phương pháp theo mục bất kỳ trong số các mục từ 18 đến 22, trong đó việc chuyển đổi sử dụng chế độ dự báo trong và kết hợp dưới dạng chế độ `MODE_TYPE_INTER`. Ví dụ, khi coi CIIP là `MODE_TYPE_INTER`, các phương pháp được mô tả trong mục 14-17 trong phần trước có thể được áp dụng. Hoặc khi các phương pháp được mô tả trong mục 14-16 được áp dụng, CIIP có thể được coi là `MODE_TYPE_INTER`.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục 3-6 của phần trước.

27. Phương pháp xử lý video bao gồm các bước: quy định xem quy tắc khối sắc độ nhỏ nhất có được thực hiện trong khi chuyển đổi giữa phân biểu diễn đã được mã hóa của vùng video này và các trị số pixel của vùng video hay không, dựa trên điều kiện mã hóa của vùng video này; và thực hiện việc chuyển đổi dựa trên việc xác định này.

28. Phương pháp theo mục 27, trong đó điều kiện mã hóa bao gồm định dạng màu của vùng video.

29. Phương pháp theo mục 28, trong đó vùng video này có chiều rộng là M pixel và chiều cao là N pixel, và trong đó điều kiện mã hóa còn phụ thuộc vào các trị số của M và/hoặc N.

30. Phương pháp theo mục 29, trong đó quy tắc khối sắc độ nhỏ nhất được kích hoạt do vùng video này có định dạng màu 4: 2: 2 và $M * N = 32$ hoặc $M * N = 64$.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục 7-11 của phần trước.

31. Phương pháp xử lý video bao gồm các bước: xác định, để chuyển đổi giữa phần biểu diễn đã được mã hóa của vùng video ở định dạng 4: 2: 2 và các trị số pixel của vùng video, loại chế độ được sử dụng đối với việc chuyển đổi dựa trên quy tắc khối sắc độ nhỏ nhất có được kích hoạt đối với vùng video này hay không; và thực hiện việc chuyển đổi dựa trên việc xác định này.

32. Phương pháp theo mục 31, trong đó loại chế độ của vùng video được đặt ngang bằng với 1 do vùng video này có định dạng 4: 2: 2 và quy tắc khối sắc độ nhỏ nhất đang được kích hoạt.

33. Phương pháp theo mục 31, trong đó việc xác định kiểu chế độ là việc xác định kiểu chế độ là kiểu INTRA do quy tắc khối sắc độ nhỏ nhất được kích hoạt đối với vùng video.

34. Phương pháp theo mục 31, trong đó việc xác định loại chế độ là việc quy định rằng loại chế độ INTER bị vô hiệu hóa do quy tắc khối sắc độ nhỏ nhất được kích hoạt đối với vùng video.

Các mục sau có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục 7-11 của phần trước.

35. Phương pháp xử lý video bao gồm các bước: xác định việc chuyển đổi giữa phần biểu diễn đã được mã hóa của khối video này và khối video của video, xem liệu phân vùng khối có được phép trong khi chuyển đổi hay không, dựa trên loại chế độ dùng trong khi chuyển đổi hoặc cỡ của khối video này ; và thực hiện việc chuyển đổi bằng cách sử dụng việc xác định này.

36. Phương pháp theo mục 35, trong đó việc phân vùng khối là việc phân vùng cây nhị phân hoặc việc phân vùng cây tam phân

37. Phương pháp theo mục bất kỳ trong số các mục 35-36, trong đó trong trường hợp loại chế độ là chế độ INTER, việc phân vùng khối dựa trên quy tắc hạn chế mà cho phép hoặc không cho phép các loại phân vùng.

38. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 37, trong đó việc chuyển đổi là việc mã hóa video thành phân biểu diễn đã được mã hóa .

39. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 37, trong đó việc chuyển đổi là việc giải mã phân biểu diễn đã được mã hóa để tạo ra các trị số pixel của video này.

40. Thiết bị giải mã video này có bộ xử lý được tạo cấu hình để thực hiện phương pháp nêu trong một hoặc nhiều mục 1 đến 39.

41. Thiết bị mã hóa video này có bộ xử lý được tạo cấu hình để thực hiện phương pháp nêu trong một hoặc nhiều mục 1 đến 39.

42. Sản phẩm chương trình máy tính có mã máy tính được lưu trên đó, mã này, khi được thực hiện bởi bộ xử lý, sẽ khiến cho bộ xử lý này thực hiện phương pháp theo mục bất kỳ trong số các mục từ 1 đến 39.

43. Phương pháp, thiết bị hoặc hệ thống được mô tả trong Bản mô tả sáng chế này.

Tập hợp các mục thứ hai mô tả các dấu hiệu và khía cạnh nhất định của các kỹ thuật đã được bộc lộ trong phần trước (ví dụ, mục 1, 3-11, 18, 19 và 24).

1. Phương pháp xử lý video (ví dụ, phương pháp 2110 được thể hiện trên FIG.21A), bao gồm các bước: xác định (2112), để chuyển đổi giữa vùng video của video và phân biểu diễn đã được mã hóa của video này, mã hóa trong phần đặc tính của vùng video này dựa trên định dạng màu của video này theo quy tắc; và thực hiện (2114) việc chuyển đổi theo đặc tính mã hóa trong .

2. Phương pháp theo mục 1, trong đó quy tắc quy định rằng trong trường hợp định dạng màu của vùng video này là 4: 0: 0 hoặc 4: 4: 4, thì đặc tính mã hóa trong là tất cả các chế độ mã hóa đều được kích hoạt đối với vùng video này và trong đó phân biểu diễn đã được mã hóa có trị số `MODE_TYPE_ALL` đối với phần tử cú pháp chỉ ra loại chế độ dùng cho vùng video.

3. Phương pháp theo các mục 1-2, trong đó quy tắc này quy định rằng trong trường hợp định dạng màu 4: 0: 0 hoặc 4: 4: 4, phân biểu diễn đã được mã hóa có

phần tử cú pháp biểu thị điều kiện xác định một loại chế độ được đặt ngang bằng với 0.

4. Phương pháp theo mục bất kỳ trong số các mục 1-3, trong đó quy tắc này quy định rằng việc hạn chế kích thước được phép nhỏ nhất đối với khối sắc độ được mã hóa trong vùng video này có được kích hoạt hay không tùy thuộc vào định dạng màu.

5. Phương pháp theo mục 4, trong đó quy tắc này quy định rằng việc hạn chế được kích hoạt đối với các định dạng 4: 2: 0 và 4: 2: 2.

6. Phương pháp theo mục 4, trong đó quy tắc này quy định rằng việc hạn chế bị vô hiệu hóa đối với các định dạng 4: 0: 0 và 4: 4: 4.

7. Phương pháp xử lý video (ví dụ, phương pháp 2120 được thể hiện trên FIG.21B), bao gồm các bước: thực hiện (2122) việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa tuân theo quy tắc định dạng và trong đó quy tắc định dạng này quy định phần tử cú pháp, modeType, biểu thị chế độ mã hóa của khối video này, tương đương với MODE_TYPE_NO_INTER mà hạn chế việc sử dụng chế độ mã hóa liên để chuyển đổi, hoặc MODE_TYPE_NO_INTRA mà hạn chế việc sử dụng chế độ trong để chuyển đổi.

8. Phương pháp xử lý video (ví dụ, phương pháp 2120 được thể hiện trên FIG.21B), bao gồm các bước: thực hiện (2122) việc chuyển đổi giữa video này và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng quy định rằng cờ chỉ ra ràng buộc chế độ dự báo không được đưa vào phần biểu diễn đã được mã hóa trong trường hợp định dạng sắc độ của video này là 4: 2: 2, 4: 0: 0 hoặc 4: 4: 4.

9. Phương pháp theo mục 8, trong đó trong trường hợp không có cờ, trị số tương ứng được suy ra bằng 0 hoặc 1.

10. Phương pháp xử lý video (ví dụ, phương pháp 2130 được thể hiện trên FIG.21C), bao gồm các bước: xác định (2132), để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, xem liệu việc và/hoặc cách hạn chế kích thước của khối dự báo trong sắc độ nhỏ nhất trong vùng video được kích hoạt theo quy tắc; và thực hiện (2134) việc chuyển đổi dựa trên việc xác định

này, trong đó quy tắc này phụ thuộc vào việc định dạng màu của video này là 4: 2: 0 hay 4: 2: 2.

11. Phương pháp xử lý video bao gồm các bước: xác định, để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, xem liệu việc hạn chế kích thước của khối dự báo trong sắc độ nhỏ nhất trong vùng video này có được kích hoạt hay không theo qui định; và thực hiện việc chuyển đổi dựa trên việc xác định này, trong đó quy tắc này phụ thuộc vào định dạng màu của video này và/hoặc chiều rộng (M) và chiều cao (N) của vùng video này và trong đó quy tắc này quy định thêm rằng, đối với vùng video này là nút cây mã hóa với việc tách BT(cây nhị phân), khi đó hạn chế về khối dự báo trong sắc độ nhỏ nhất sẽ bị vô hiệu hóa trong trường hợp 1) định dạng màu của video này là 4: 2: 2 và 2) tích của M và N là một trị số từ một tập hợp trị số, trong đó tập hợp trị số này bao gồm cả 64.

12. Phương pháp theo mục 11, trong đó quy tắc này quy định thêm rằng việc hạn chế về khối dự báo trong sắc độ nhỏ nhất được kích hoạt trong trường hợp 1) định dạng màu của video này là 4: 2: 2 và 2) mà bộ trị số này bao gồm cả 32.

13. Phương pháp theo mục 11, trong đó quy tắc này quy định thêm rằng việc hạn chế về khối dự báo trong sắc độ nhỏ nhất bị vô hiệu hóa trong trường hợp 1) định dạng màu của video này là 4: 2: 2 và 2) mà bộ trị số này bao gồm cả 128.

14. Phương pháp theo mục 11, trong đó quy tắc này quy định thêm, đối với vùng video này là nút cây mã hóa với việc tách qt_flag ngang bằng với 1, hạn chế về khối dự báo trong sắc độ nhỏ nhất được kích hoạt trong trường hợp định dạng màu của video này là 4: 2: 2.

15. Phương pháp theo mục 11, trong đó quy tắc này quy định thêm, đối với vùng video này là nút cây mã hóa với việc tách BT(cây tam phân), hạn chế đối với khối dự báo trong sắc độ nhỏ nhất được kích hoạt trong trường hợp định dạng màu của video này là 4: 2: 2.

16. Phương pháp theo mục 11, trong đó quy tắc này quy định thêm, đối với vùng video này là nút cây mã hóa với việc tách BT(cây nhị phân), hạn chế đối với khối dự báo trong sắc độ nhỏ nhất được kích hoạt trong trường hợp 1) định dạng màu của video này là 4: 2: 2 và 2) mà bộ trị số này bao gồm cả 32.

17. Phương pháp theo mục bất kỳ trong số các mục từ 11 đến 16, trong đó quy tắc này quy định thêm, đối với khối dự báo trong sắc độ nhỏ nhất trong video này có định dạng màu 4: 2: 2, modeTypeCondition luôn ngang bằng với 1.

18. Phương pháp theo mục bất kỳ trong số các mục từ 11 đến 17, trong đó quy tắc này quy định thêm, đối với khối dự báo trong sắc độ nhỏ nhất trong video này có định dạng màu 4: 2: 2, chỉ MODE_TYPE_INTRA mới cho phép sử dụng chế độ trong, một chế độ bảng màu và một chế độ sao chép trong khối để chuyển đổi được phép .

19. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này theo việc hạn chế về kích thước của khối dự báo trong sắc độ nhỏ nhất, trong đó phần biểu diễn đã được mã hóa tuân theo quy tắc định dạng quy định trị số của trường cú pháp trong phần biểu diễn đã được mã hóa, do định dạng màu 4: 2: 2 của video này.

20. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với modeTypeCondition của khối SCIPU và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, modeTypeCondition luôn là 1.

21. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với modeTypeCondition của khối SCIPU và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, modeTypeCondition là 0 hoặc 1.

22. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với modeTypeCondition của khối SCIPU và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, modeTypeCondition không là .

23. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với modeType của khối SCIPU và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, modeType luôn ngang bằng với MODE_TYPE_INTRA mà cho phép sử dụng chế độ trong, chế độ bảng màu và chế độ sao chép trong khối .

24. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với modeType của khối SCIPU và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, modeType là 1) MODE_TYPE_ALL mà cho phép sử dụng chế độ mã hóa liên, chế độ trong, chế độ bảng màu và chế độ sao chép trong khối

để chuyển đổi hoặc 2) `MODE_TYPE_INTRA` mà cho phép sử dụng chế độ trong, chế độ bảng màu và chế độ sao chép trong khối .

25. Phương pháp theo mục 19, trong đó trường cú pháp tương ứng với `modeType` của khối `SCIPU` và trong đó quy tắc định dạng này quy định thêm, do định dạng màu 4: 2: 2, `modeType` không tương ứng với `MODE_TYPE_INTER` mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi.

26. Phương pháp xử lý video (ví dụ, phương pháp 2140 được thể hiện trên FIG.21D), bao gồm các bước: xác định (2142), để chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, một khả năng áp dụng của sơ đồ phân vùng đối với khối video hiện tại theo quy tắc ; và thực hiện (2144) việc chuyển đổi dựa trên việc xác định này.

27. Phương pháp theo mục 26, trong đó quy tắc này quy định rằng bước xác định xác định khả năng áp dụng dựa trên ít nhất một trong số các loại chế độ dùng trong khi chuyển đổi hoặc cỡ của khối video này và trong đó sơ đồ phân vùng có việc tách BT (cây nhị phân) và/hoặc việc tách TT (cây tam phân).

28. Phương pháp theo mục 27, trong đó trong trường hợp loại chế độ ngang bằng với `MODE_TYPE_INTER` mà cho phép sử dụng chỉ chế độ liên để chuyển đổi, việc tách BT không được phép đối với khối video hiện tại.

29. Phương pháp theo mục 27, trong đó trong trường hợp loại chế độ ngang bằng với `MODE_TYPE_INTER` mà cho phép sử dụng chỉ chế độ liên để chuyển đổi, việc tách TT không được phép đối với khối video hiện tại.

30. Phương pháp theo mục 27, trong đó trong trường hợp $M * N$ nhỏ hơn hoặc bằng 32 và kiểu chế độ ngang bằng với `MODE_TYPE_INTER` mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi, việc tách BT không được phép, nhờ đó M và N tương ứng với chiều cao và chiều rộng của khối video hiện tại này .

31. Phương pháp theo mục 27, trong đó trong trường hợp $M * N$ nhỏ hơn hoặc bằng 64 và loại chế độ ngang bằng với `MODE_TYPE_INTER` mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi, thì việc tách TT không được phép, nhờ đó M và N tương ứng với chiều cao và chiều rộng của khối video hiện tại này .

32. Phương pháp theo mục 26, trong đó quy tắc này quy định hạn chế một sơ đồ phân vùng nhất định dựa trên một phần tử cú pháp, `modeTypeCurr`, mà có trong

phần biểu diễn đã được mã hóa và mô tả một loại chế độ dùng trong khi chuyển đổi, và trong đó sơ đồ phân vùng nhất định có việc tách BT (cây nhị phân), việc tách TT (cây tam phân) và/hoặc việc tách QT (cây tứ phân).

33. Phương pháp theo mục 32, trong đó, do modeTypeCurr ngang bằng với MODE_TYPE_INTER mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi, việc tách BT không được phép.

34. Phương pháp theo mục 32, trong đó, do modeTypeCurr ngang bằng với MODE_TYPE_INTER mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi, việc tách TT không được phép.

35. Phương pháp theo mục 32, trong đó, do modeTypeCurr ngang bằng với MODE_TYPE_INTER mà cho phép chỉ sử dụng một chế độ liên để chuyển đổi, việc tách QT không được phép.

36. Phương pháp theo mục 32, trong đó việc tách BT không được phép trong trường hợp modeTypeCurr ngang bằng với MODE_TYPE_INTER mà cho phép sử dụng chỉ chế độ liên để chuyển đổi và kích thước khối sắc độ nhỏ hơn hoặc bằng 32.

37. Phương pháp theo mục 32, trong đó việc tách TT không được phép trong trường hợp modeTypeCurr ngang bằng với MODE_TYPE_INTER mà cho phép sử dụng chỉ chế độ liên để chuyển đổi và kích thước khối sắc độ nhỏ hơn hoặc bằng 64.

38. Phương pháp xử lý video (ví dụ, phương pháp 2150 được thể hiện trên FIG.21E), bao gồm các bước: xác định (2152), để chuyển đổi giữa khối video của video và phần biểu diễn đã được mã hóa của video này, cho dù chế độ được kích hoạt theo quy tắc và thực hiện (2154) chuyển đổi dựa trên việc xác định này, trong đó quy tắc này quy định rằng chế độ liên được kích hoạt trong trường hợp việc phân vùng cây nhị phân của các mẫu sắc độ được kích hoạt đối với khối video.

39. Phương pháp theo mục 38, trong đó phần biểu diễn đã được mã hóa có trường cú pháp mà ngang bằng với DUAL_TREE_LUMA.

40. Phương pháp theo mục 38, trong đó đơn vị mã hóa đã được mã hóa ở chế độ liên chỉ chứa các mẫu sắc độ đối với các định dạng màu có nhiều thành phần màu.

41. Phương pháp theo mục 38, trong đó phần biểu diễn đã được mã hóa có cờ chỉ báo về chế độ dự báo được áp dụng đối với khối video này và cờ này được phân tích cú pháp đối với khối video tương ứng với khối sắc độ có kiểu cây nhị phân.

42. Phương pháp theo mục 38, trong đó quy tắc này quy định thêm để áp dụng các ràng buộc tương tự về chế độ liên bất kể phân vùng cây nhị phân hay phân vùng cây đơn phân của các mẫu sắc độ được kích hoạt đối với khối video.

43. Phương pháp xử lý video (ví dụ, phương pháp 2160 được thể hiện trên FIG.21F), bao gồm các bước: xác định (2162), để chuyển đổi giữa vùng video của video và phần biểu diễn đã được mã hóa của video này, dựa trên quy tắc, xem liệu việc sử dụng chế độ bảng màu có được phép đối với vùng video này hay không; và thực hiện (2164) chuyển đổi dựa trên việc xác định này, trong đó chế độ bảng màu có việc mã hóa vùng video này bằng cách sử dụng bảng màu có các trị số mẫu đại diện.

44. Phương pháp theo mục 43, quy tắc quy định rằng chế độ bảng màu được phép trong trường hợp loại chế độ của vùng video ngang bằng với `MODE_TYPE_INTRA` mà cho phép sử dụng chế độ trong, chế độ bảng màu và chế độ sao chép trong khối để chuyển đổi hoặc `MODE_TYPE_INTER` chỉ cho phép sử dụng chế độ liên để chuyển đổi.

45. Phương pháp theo mục 43, trong đó quy tắc này quy định rằng chế độ bảng màu được phép độc lập với loại chế độ của vùng video.

46. Phương pháp theo mục 43, trong đó quy tắc này dựa trên kiểu lát cắt và kiểu chế độ của vùng video.

47. Phương pháp theo mục 46, trong đó quy tắc này quy định rằng chế độ bảng màu được phép đối với lát cắt I có loại chế độ ngang bằng với `MODE_TYPE_INTRA` mà cho phép sử dụng chế độ trong, chế độ bảng màu và chế độ sao chép trong khối để chuyển đổi.

48. Phương pháp theo mục 46, trong đó quy tắc này quy định rằng chế độ bảng màu được phép đối với một lát cắt P/B có loại chế độ ngang bằng với `MODE_TYPE_INTRA` mà cho phép sử dụng chế độ trong, chế độ bảng màu và chế độ sao chép trong khối để chuyển đổi.

49. Phương pháp theo mục 43, trong đó quy tắc này quy định thêm rằng cây nhị phân cục bộ không được phép trong trường hợp chế độ bảng màu được phép.

50. Phương pháp theo mục 43, trong đó điều kiện `modeTypeCondition` luôn được đặt ngang bằng với 0 trong trường hợp chế độ bảng màu được kích hoạt.

51. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định phần tử cú pháp, modeType, là MODE_TYPE_IBC mà cho phép sử dụng chế độ sao chép trong khối để chuyển đổi hoặc MODE_TYPE_PALETTE mà cho phép sử dụng chế độ bảng màu để chuyển đổi, trong đó chế độ sao chép trong khối bao gồm việc mã hóa khối video hiện tại bằng cách sử dụng ít nhất một vector khối trỏ đến khung video chứa khối video hiện tại và trong đó chế độ bảng màu có việc mã hóa khối video hiện tại bằng cách sử dụng bảng các trị số mẫu đại diện.

52. Phương pháp theo mục 51, trong đó quy tắc định dạng này quy định thêm rằng phần biểu diễn đã được mã hóa không có pred_mode_flag, pred_mode_ibc_flag và/hoặc pre_mode_plt_flag, trong trường hợp modeType là MODE_TYPE_IBC hoặc MODE_TYPE_PALETTE.

53. Phương pháp theo mục 51, trong đó phần biểu diễn đã được mã hóa có một chỉ mục chỉ ra loại chế độ dùng để chuyển đổi thay vì mode_constraint_flag.

54. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 53, trong đó việc chuyển đổi là việc mã hóa video thành phần biểu diễn đã được mã hóa.

55. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 53, trong đó việc chuyển đổi là việc giải mã phần biểu diễn đã được mã hóa để tạo ra video.

56. Thiết bị xử lý video này có bộ xử lý được tạo cấu hình để thực hiện phương pháp nêu trong một hoặc nhiều mục từ 1 đến 55.

57. Phương tiện đọc được bởi máy tính lưu trữ mã chương trình mà khi được thực hiện, bộ xử lý thực hiện phương pháp nêu trong một mục bất kỳ hoặc nhiều mục từ 1 đến 55.

58. Phương tiện máy tính đọc được mà lưu trữ phần biểu diễn đã được mã hóa hoặc phần biểu diễn dòng bit đã được tạo ra theo phương pháp bất kỳ đã được mô tả trên đây.

Tập hợp các mục thứ ba mô tả các dấu hiệu và khía cạnh nhất định của các kỹ thuật được bộc lộ trong phần trước (ví dụ, mục 25-33).

1. Phương pháp xử lý video bao gồm các bước: xác định việc chuyển đổi giữa khối video hiện tại của video và phần biểu diễn đã được mã hóa của video này, xem liệu một sơ đồ phân vùng nhất định có được phép đối với khối video hiện tại theo quy tắc hay không mà phụ thuộc vào loại chế độ mã hóa dùng để biểu diễn khối video hiện tại trong phần biểu diễn đã được mã hóa và kích thước của khối video này ; và thực hiện việc chuyển đổi dựa trên việc xác định này.

2. Phương pháp theo mục 1, trong đó sơ đồ phân vùng nhất định có một QT (cây tứ phân) được tách trong đó khối video hiện tại được chia thành bốn phần theo cả chiều ngang và chiều dọc, chia TT(cây tam phân) theo chiều dọc trong đó khối video hiện tại được chia thành ba phần theo chiều dọc, chia TT theo chiều ngang trong đó khối video hiện tại được chia thành ba phần theo chiều ngang, chia BT (cây nhị phân) theo chiều dọc trong đó khối video hiện tại được chia thành hai phần theo chiều dọc và/hoặc tách BT theo chiều ngang, trong đó khối video hiện tại được chia thành hai phần theo chiều ngang.

3. Phương pháp theo mục 1, trong đó quy tắc quy định rằng việc tách QT (cây tứ phân) không được phép đối với khối video hiện tại trong trường hợp 1) mà phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` tương ứng với loại chế độ mã hóa trong đó chỉ chế độ trong được phép đối với khối video hiện tại và 2) mà cả chiều rộng và chiều cao của khối video này là 8.

4. Phương pháp theo mục 1, trong đó quy tắc quy định rằng việc tách TT (cây tam phân) không được phép đối với khối video hiện tại trong trường hợp 1) mà phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` tương ứng với loại chế độ mã hóa trong đó chế độ liên chỉ được phép đối với khối video hiện tại và 2) tích chiều rộng và chiều cao của khối video này là 64.

5. Phương pháp theo mục 4, trong đó quy tắc này quy định thêm rằng việc tách TT (cây tam phân) theo chiều dọc không được phép trong trường hợp phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` và chiều rộng và chiều cao của khối video này lần lượt là 16 và 4.

6. Phương pháp theo mục 4, trong đó quy tắc này quy định thêm rằng việc tách TT (cây tam phân) theo chiều ngang không được phép trong trường hợp phần

biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` và chiều rộng và chiều cao của khối video này lần lượt là 4 và 16.

7. Phương pháp theo mục 1, trong đó quy tắc quy định rằng việc tách BT (cây nhị phân) không được phép đối với khối video hiện tại trong trường hợp 1) mà phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` tương ứng với loại chế độ mã hóa trong đó chỉ chế độ trong được phép đối với khối video hiện tại và 2) tích chiều rộng và chiều cao của khối video này là 32.

8. Phương pháp theo mục 7, trong đó quy tắc này quy định thêm rằng việc tách BT (cây nhị phân) theo chiều dọc không được phép trong trường hợp phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` và chiều rộng và chiều cao của khối video này lần lượt là 8 và 4.

9. Phương pháp theo mục 7, trong đó quy tắc này quy định rằng việc tách BT (cây nhị phân) theo chiều ngang không được phép trong trường hợp phần biểu diễn đã được mã hóa có trị số `MODE_TYPE_INTER` và chiều rộng và chiều cao của khối video này lần lượt là 4 và 8.

10. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa khối video của video và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định rằng một đặc điểm của khối video này kiểm soát xem một phần tử cú pháp trong phần biểu diễn đã được mã hóa có chỉ ra chế độ dự báo của khối video này hay không.

11. Phương pháp theo mục 10, trong đó đặc tính của khối video này có ít nhất một trong số các thành phần màu hoặc kích thước của khối video này.

12. Phương pháp theo mục 10 hoặc 11, trong đó quy tắc định dạng này quy định thêm rằng phần tử cú pháp chỉ ra chế độ dự báo của khối video này tương ứng với khối sắc độ.

13. Phương pháp theo mục bất kỳ trong số các mục từ 10 đến 12, trong đó quy tắc định dạng này quy định thêm rằng phần tử cú pháp không chỉ ra chế độ dự báo của khối video này tương ứng với khối sắc độ và chế độ dự báo của khối video này tương ứng với khối sắc độ có trong phần biểu diễn đã được mã hóa.

14. Phương pháp theo mục 13, trong đó khối video này có chiều rộng và chiều cao lớn hơn 4.

15. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa vùng video của thành phần thứ nhất của video này và phần biểu diễn đã được mã hóa của video này, trong đó phần biểu diễn đã được mã hóa này tuân theo quy tắc định dạng, trong đó quy tắc định dạng này quy định xem liệu và/hoặc cách một trường cú pháp được tạo cấu hình trong phần biểu diễn đã được mã hóa để chỉ ra một tham số lượng tử hóa khác biệt đối với vùng video phụ thuộc vào sơ đồ tách dùng để tách các mẫu của thành phần thứ nhất.

16. Phương pháp theo mục 15, trong đó quy tắc định dạng này quy định thêm xem liệu và/hoặc cách trường cú pháp được tạo cấu hình trong phần biểu diễn đã được mã hóa có độc lập với sơ đồ tách dùng để tách các mẫu của thành phần thứ hai của video này hay không.

17. Phương pháp theo mục 15 hoặc 16, trong đó thành phần thứ nhất là thành phần độ chói và thành phần thứ hai là thành phần sắc độ.

18. Phương pháp theo mục 15 hoặc 16, trong đó thành phần thứ nhất là thành phần sắc độ và thành phần thứ hai là thành phần độ chói.

19. Phương pháp theo mục 15, trong đó quy tắc định dạng này quy định thêm để có thông tin liên quan đến tham số lượng tử hóa vi sai nhiều nhất một lần trong một vùng cụ thể trong đó thành phần độ chói và thành phần sắc độ dùng chung cùng một kiểu chế độ.

20. Phương pháp theo mục 19, trong đó vùng cụ thể tương ứng với một nhóm lượng tử hóa.

21. Phương pháp xử lý video bao gồm các bước: thực hiện việc chuyển đổi giữa vùng video của thành phần thứ nhất của video này và phần biểu diễn đã được mã hóa của video này theo quy tắc, trong đó quy tắc này quy định, trong trường hợp cây nhị phân và/hoặc cấu trúc mã hóa cây nhị phân cục bộ được áp dụng đối với vùng video, biến liên quan đến tham số lượng tử hóa vi sai của thành phần thứ nhất không bị sửa đổi trong quy trình giải mã hoặc phân tích cú pháp của thành phần thứ hai của video này.

22. Phương pháp theo mục 21, trong đó cấu trúc cây nhị phân cục bộ được áp dụng đối với vùng video trong trường hợp ràng buộc về kích thước được phép nhỏ nhất đối với khối sắc độ được áp dụng đối với vùng video này.

23. Phương pháp theo mục 21 hoặc 22, trong đó thành phần thứ nhất là thành phần độ chói và thành phần thứ hai là thành phần sắc độ.

24. Phương pháp theo mục 21 hoặc 22, trong đó thành phần thứ nhất là thành phần sắc độ và thành phần thứ hai là thành phần độ chói.

25. Phương pháp theo mục bất kỳ trong số các mục từ 21 đến 24, trong đó tham số lượng tử hóa vi sai chỉ ra sự khác biệt về trị số lượng tử hóa được áp dụng đối với khối video này và trị số lượng tử hóa trước đó được áp dụng đối với khối video lân cận.

26. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 25, trong đó bước thực hiện việc chuyển đổi bao gồm việc tạo ra phần biểu diễn đã được mã hóa từ video này.

27. Phương pháp theo mục bất kỳ trong số các mục từ 1 đến 25, trong đó bước thực hiện việc chuyển đổi là bước tạo ra video từ phần biểu diễn đã được mã hóa.

28. Thiết bị xử lý video này có bộ xử lý được tạo cấu hình để thực hiện phương pháp nêu trong một hoặc nhiều mục từ 1 đến 27.

29. Phương tiện đọc được bởi máy tính lưu trữ mã chương trình mà khi được thực hiện, khiến cho bộ xử lý thực hiện phương pháp nêu trong mục bất kỳ hoặc nhiều mục 1 đến 27.

30. Phương tiện máy tính đọc được lưu trữ phần biểu diễn đã được mã hóa hoặc phần biểu diễn dòng bit đã được tạo ra theo phương pháp bất kỳ đã được mô tả trên đây.

Các giải pháp đã được bộc lộ và các giải pháp khác, ví dụ, phương án, mô-đun và các hoạt động chức năng được mô tả trong Bản mô tả sáng chế này có thể được thực hiện trong mạch điện tử kỹ thuật số hoặc trong phần mềm máy tính, phần lõi(sụn) hoặc phần cứng, bao gồm các cấu trúc đã biết trong Bản mô tả sáng chế này và cấu trúc tương đương của chúng, hoặc tổ hợp của một hoặc nhiều trong số chúng. Sáng chế theo các phương án đã được bộc lộ và theo các phương án khác có thể

được thực hiện dưới dạng một hoặc nhiều sản phẩm chương trình máy tính, tức là một hoặc nhiều mô-đun hướng dẫn chương trình máy tính được mã hóa trên phương tiện máy tính đọc được để thực hiện hoặc để điều khiển hoạt động của thiết bị xử lý dữ liệu. Phương tiện máy tính đọc được có thể là thiết bị lưu trữ có thể đọc được bằng máy, nền lưu trữ có thể đọc được bằng máy, thiết bị nhớ, môi trường tạo ra tín hiệu lan truyền mà máy có thể đọc được hoặc kết hợp của một hoặc nhiều trong số chúng. Cụm từ “thiết bị xử lý dữ liệu” bao gồm tất cả các thiết bị, thiết bị và máy móc để xử lý dữ liệu, bao gồm, ví dụ như một bộ xử lý lập trình được, một máy tính hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị này có thể có, ngoài phần cứng, mã tạo ra môi trường thực hiện cho chương trình máy tính được đề cập, ví dụ, mã cấu thành phần sụn của bộ xử lý, ngăn xếp giao thức, hệ thống quản lý cơ sở dữ liệu, hệ điều hành hoặc tổ hợp của một hoặc nhiều trong số chúng. Tín hiệu lan truyền là tín hiệu được tạo ra nhân tạo, ví dụ, tín hiệu điện, quang hoặc điện từ do máy tạo ra, được tạo ra nhằm mã hóa thông tin để truyền đến thiết bị thu thích hợp.

Chương trình máy tính (còn được gọi là chương trình, phần mềm, ứng dụng phần mềm, tập lệnh hoặc mã) có thể được viết bằng dạng ngôn ngữ lập trình bất kỳ, kể cả ngôn ngữ biên dịch hoặc thông dịch và nó có thể được thực hiện dưới mọi dạng, kể cả dưới dạng một chương trình độc lập hoặc dưới dạng mô-đun, thành phần, chương trình 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 một tệp trong hệ thống tệp. Một chương trình có thể được lưu trữ trong một phần của tệp chứa các chương trình hoặc dữ liệu khác (ví dụ, một hoặc nhiều tập lệnh được lưu trữ trong tài liệu ngôn ngữ đánh dấu), trong một tệp duy nhất dành riêng cho chương trình được đề cập hoặc trong nhiều tệp phối hợp (ví dụ, các tệp lưu trữ một hoặc nhiều mô-đun, chương trình con hoặc các phần mã). Một chương trình máy tính có thể được triển khai để thực hiện trên một máy tính hoặc trên nhiều máy tính được đặt tại một nơi hoặc được phân phối ở nhiều nơi và được kết nối với nhau bằng mạng truyền thông.

Các quy trình và luồng logic được mô tả trong Bản mô tả sáng chế này có thể được thực hiện bởi một hoặc nhiều bộ xử lý lập trình đang thực hiện một hoặc nhiều chương trình máy tính để thực hiện các chức năng bằng cách hoạt động trên dữ liệu đầu vào và tạo đầu ra. Các quy trình và luồng logic cũng có thể được thực hiện bởi

và thiết bị cũng có thể được thực hiện như, mạch logic mục đích đặc biệt, ví dụ, FPGA (mảng cổng có thể lập trình trường) hoặc ASIC (mạch tích hợp dành riêng cho ứng dụng)

Các bộ xử lý thích hợp đối với việc thực hiện một chương trình máy tính bao gồm, ví dụ, cả bộ vi xử lý mục đích chung và mục đích đặc biệt, và một bộ bất kỳ hoặc nhiều bộ xử lý của loại máy tính kỹ thuật số bất kỳ. Nói chung, một bộ xử lý sẽ nhận các lệnh và dữ liệu từ bộ nhớ chỉ đọc hoặc bộ nhớ truy cập ngẫu nhiên hoặc cả hai. Các thành phần thiết yếu của máy tính là một bộ xử lý để thực hiện các lệnh và một hoặc nhiều thiết bị nhớ để lưu trữ các lệnh và dữ liệu. Nói chung, một máy tính cũng sẽ bao gồm, hoặc được kết hợp hoạt động để nhận dữ liệu từ hoặc truyền dữ liệu đến, hoặc cả hai, một hoặc nhiều thiết bị lưu trữ chung để lưu trữ dữ liệu, ví dụ, đĩa từ, đĩa quang từ, hoặc đĩa quang. Tuy nhiên, một máy tính không nhất thiết phải có các thiết bị như vậy. Phương tiện có thể đọc được của máy tính thích hợp để lưu trữ dữ liệu và hướng dẫn chương trình máy tính có tất cả các dạng thiết bị nhớ, phương tiện và bộ nhớ bất khả biến, kể cả thiết bị nhớ bán dẫn chẳng hạn, ví dụ, EPROM, EEPROM và thiết bị nhớ flash; đĩa từ, ví dụ, đĩa cứng bên trong hoặc đĩa tháo lắp được; đĩa quang từ; và đĩa CD ROM và DVD-ROM. Bộ xử lý và bộ nhớ có thể được bổ sung hoặc kết hợp trong mạch logic mục đích đặc biệt.

Trong khi Bản mô tả sáng chế này chứa nhiều thông tin chi tiết cụ thể, nhưng chúng không nên được hiểu là giới hạn phạm vi của đối tượng bất kỳ hoặc của những gì có thể được yêu cầu bảo hộ, nhưng đúng hơn là mô tả các dấu hiệu mà có thể là cụ thể đối với sáng chế theo các phương án cụ thể của các kỹ thuật cụ thể. Một số tính năng nhất định mà được mô tả trong Bản mô tả sáng chế này theo các phương án riêng biệt cũng có thể được thực hiện kết hợp trong sáng chế theo một phương án duy nhất. Ngược lại, các dấu hiệu khác nhau được mô tả trong phạm vi của sáng chế theo một phương án cũng có thể được thực hiện theo nhiều phương án riêng biệt hoặc trong tổ hợp con thích hợp bất kỳ. Hơn nữa, mặc dù các dấu hiệu có thể đã được mô tả trên đây là hoạt động trong các tổ hợp nhất định và thậm chí được yêu cầu bảo hộ ban đầu như vậy, một hoặc nhiều dấu hiệu từ tổ hợp được yêu cầu bảo hộ trong một số trường hợp có thể bị loại bỏ ra khỏi tổ hợp này và tổ hợp được yêu cầu bảo hộ có thể hướng đến một tổ hợp con hoặc biến thể của một tổ hợp con.

Tương tự, trong khi các hoạt động được thể hiện trên các hình vẽ theo một thứ tự cụ thể, thì không nên được hiểu là yêu cầu rằng các hoạt động đó phải được thực hiện theo thứ tự cụ thể được biểu diễn hoặc theo thứ tự tuần tự, hoặc tất cả các hoạt động được minh họa phải được thực hiện, để đạt được các kết quả mong muốn. Hơn nữa, việc tách các thành phần hệ thống khác nhau theo các phương án được mô tả trong Bản mô tả sáng chế này không nên được hiểu là yêu cầu việc tách như vậy trong sáng chế theo tất cả các phương án.

Chỉ một vài cách thực hiện và ví dụ được mô tả và các cách thực hiện, cải tiến và biến thể khác có thể được tạo ra dựa trên những gì đã được mô tả và minh họa trong Bản mô tả sáng chế này.

YÊU CẦU BẢO HỘ

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

xác định, để chuyển đổi giữa lát cắt video bao gồm khối cha luma và khối cha chroma và dòng bit của video, liệu có chia khối cha luma và khối cha chroma thành một hoặc nhiều khối luma và một hoặc nhiều khối chroma một cách tương ứng hay không, trong đó kiểu chế độ của khối luma hoặc kiểu chế độ của khối chroma được xác định dựa vào trị số của biến, và trong đó trị số của biến được suy ra dựa vào định dạng màu của lát cắt video; và

thực hiện việc chuyển đổi dựa vào việc xác định này,

trong đó khối cha luma được tạo ra từ khối cây mã hóa (CTB) dựa vào sơ đồ phân vùng luma bao gồm việc phân vùng đệ quy, và

trong đó khối cha chroma được tạo ra từ khối cây mã hóa chroma (CTB) dựa vào sơ đồ phân vùng chroma có cùng việc phân vùng đệ quy với sơ đồ luma.

2. Phương pháp theo điểm 1, trong đó khi định dạng màu của lát cắt video là 4:0:0 hoặc 4:4:4, trị số của biến được thiết lập là 0.

3. Phương pháp theo điểm 2, trong đó khi trị số của biến là 0, kiểu chế độ của khối luma và kiểu chế độ của khối chroma là `MODE_TYPE_ALL`, và trong đó kiểu chế độ có trị số `MODE_TYPE_ALL` quy định rằng tất cả các chế độ mã hóa đều khả thi đối với khối luma và khối chroma.

4. Phương pháp theo điểm 3, trong đó khi trị số của biến là 0, cờ ràng buộc chế độ không có trong dòng bit, và trong đó trị số của cờ ràng buộc chế độ quy định liệu các chế độ mã hóa chỉ báo liên có thể được sử dụng hay không.

5. Phương pháp theo điểm 1, trong đó trị số của biến còn được suy ra dựa vào kích thước của khối cha luma và chế độ phân vùng được xác định trước thứ nhất của khối cha luma.

6. Phương pháp theo điểm 5, trong đó đáp ứng với khối cha luma có kích thước là 64, định dạng màu của lát cắt video không phải là 4:0:0 và 4:4:4, và chế độ phân vùng được xác định trước thứ nhất có việc tách QT (cây tứ phân), biến bằng 1.

7. Phương pháp theo điểm 5, trong đó đáp ứng với khối cha luma có kích thước là 32, định dạng màu của lát cắt video không phải là 4:0:0 và 4:4:4, và chế độ phân vùng được xác định trước thứ nhất có việc tách BT (cây nhị phân), biến bằng 1.

8. Phương pháp theo điểm 5, trong đó đáp ứng với loại lát cắt của lát cắt video ngang bằng với lát cắt I, định dạng màu của lát cắt video là 4:2:0, khối cha luma có kích thước là 64, và chế độ phân vùng được xác định trước thứ nhất có việc tách BT (cây nhị phân), biến được thiết lập là 1, và trong đó đáp ứng với định dạng màu của lát cắt video là 4:2:2, khối cha luma có kích thước là 64, và chế độ phân vùng được xác định trước thứ nhất có việc tách BT (cây nhị phân), biến được thiết lập là 0.

9. Phương pháp theo điểm 5, trong đó đáp ứng với loại lát cắt của lát cắt video ngang bằng với lát cắt I, định dạng màu của lát cắt video là 4:2:0, khối cha luma có kích thước là 128, và chế độ phân vùng được xác định trước thứ nhất có việc tách TT (cây tam phân), biến được thiết lập là 1, và trong đó đáp ứng với định dạng màu của lát cắt video là 4:2:2, khối cha luma có kích thước là 128, và chế độ phân vùng được xác định trước thứ nhất có việc tách TT (cây tam phân), biến được thiết lập là 0.

10. Phương pháp theo điểm 5, trong đó đáp ứng với trị số của biến là 1, khối cha chroma không được để tách và chế độ phân vùng được xác định trước thứ nhất được phép đối với khối cha luma.

11. Phương pháp theo điểm 5, trong đó đáp ứng với loại lát cắt không phải là lát cắt I, định dạng màu của lát cắt video là 4:2:0, khối cha luma có kích thước là 64, và chế

độ phân vùng được xác định trước thứ nhất có việc tách BT (cây nhị phân), biến được thiết lập là 2, và trong đó đáp ứng với định dạng màu của lát cắt video là 4:2:2, khối cha luma có kích thước là 64, và chế độ phân vùng được xác định trước thứ nhất có việc tách BT (cây nhị phân), biến được thiết lập là 0.

12. Phương pháp theo điểm 5, trong đó đáp ứng với loại lát cắt không phải là lát cắt I, định dạng màu của lát cắt video là 4:2:0, khối cha luma có kích thước là 128, và chế độ phân vùng được xác định trước thứ nhất có việc tách TT (cây tam phân), biến được thiết lập là 2, và trong đó đáp ứng với định dạng màu của lát cắt video là 4:2:2, khối cha luma có kích thước là 128, và chế độ phân vùng được xác định trước thứ nhất có việc tách TT (cây tam phân), biến được thiết lập là 0.

13. Phương pháp theo điểm 5, trong đó đáp ứng với trị số của biến là 2, cờ ràng buộc chế độ có trong dòng bit, và trong đó cờ ràng buộc chế độ quy định liệu các chế độ mã hóa chỉ báo liên có thể được sử dụng hay không.

14. Phương pháp theo điểm 13, trong đó đáp ứng với trị số của biến là 2 và cờ ràng buộc chế độ quy định các chế độ mã hóa chỉ báo liên không thể được sử dụng, khối cha chroma không được để tách và việc phân vùng xác định trước thứ nhất được phép đối với khối cha luma.

15. Phương pháp theo điểm 1, trong đó việc chuyển đổi bao gồm việc giải mã video thành dòng bit.

16. Phương pháp theo điểm 1, trong đó việc chuyển đổi bao gồm việc giải mã video từ dòng bit.

17. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ không tạm thời có các lệnh trong đó, trong đó các lệnh khi được thực thi bởi bộ xử lý, khiến bộ xử lý: xác định, để chuyển đổi giữa lát cắt video bao gồm khối cha luma và khối cha chroma và dòng bit của video, liệu có chia khối cha luma và khối cha chroma thành một hoặc nhiều khối luma và một hoặc nhiều khối chroma một cách tương ứng hay không, trong đó kiểu chế độ của khối luma hoặc kiểu chế độ của khối chroma được xác định dựa vào trị số của biến, và trong đó trị số của biến được suy ra dựa vào định dạng màu của lát cắt video; và thực hiện việc chuyển đổi dựa vào việc xác định này, trong đó khối cha luma được tạo ra từ khối cây mã hóa (CTB) dựa vào sơ đồ phân vùng luma bao gồm việc phân vùng đệ quy, và trong đó khối cha chroma được tạo ra từ khối cây mã hóa chroma (CTB) dựa vào sơ đồ phân vùng chroma có cùng việc phân vùng đệ quy với sơ đồ luma.

18. Thiết bị theo điểm 17, trong đó khi định dạng màu của lát cắt video là 4:0:0 hoặc 4:4:4, trị số của biến được thiết lập là 0, trong đó khi trị số của biến là 0, kiểu chế độ của khối luma và kiểu chế độ của khối chroma là `MODE_TYPE_ALL`, và trong đó kiểu chế độ có trị số `MODE_TYPE_ALL` quy định rằng tất cả các chế độ mã hóa đều khả thi đối với khối luma và khối chroma.

19. Vật lưu trữ đọc được bằng máy tính không tạm thời lưu trữ các lệnh khiến bộ xử lý: xác định, để chuyển đổi giữa lát cắt video bao gồm khối cha luma và khối cha chroma và dòng bit của video, liệu có chia khối cha luma và khối cha chroma thành một hoặc nhiều khối luma và một hoặc nhiều khối chroma một cách tương ứng hay không, trong đó kiểu chế độ của khối luma hoặc kiểu chế độ của khối chroma được xác định dựa vào trị số của biến, và trong đó trị số của biến được suy ra dựa vào định dạng màu của lát cắt video; và thực hiện việc chuyển đổi dựa vào việc xác định này,

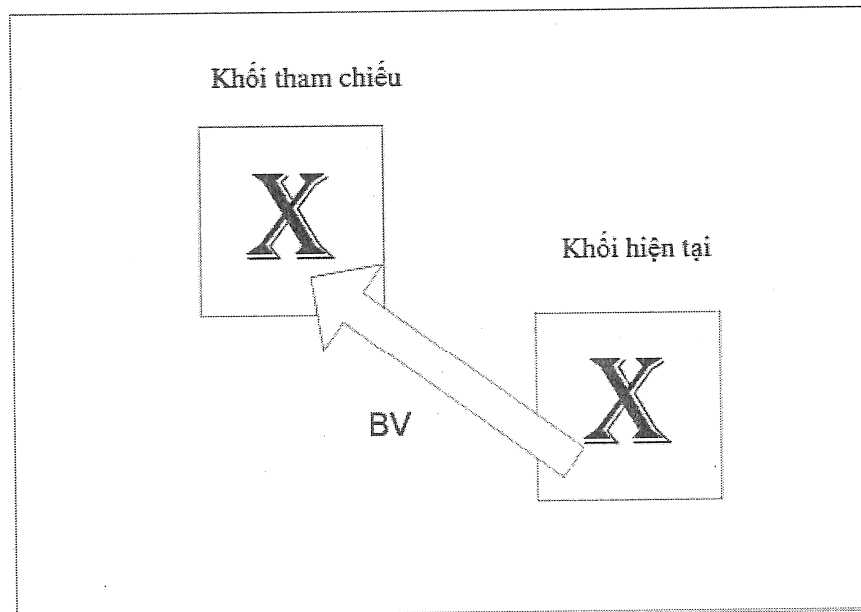
trong đó khối cha luma được tạo ra từ khối cây mã hóa (CTB) dựa vào sơ đồ phân vùng luma bao gồm việc phân vùng đệ quy, và trong đó khối cha chroma được tạo ra từ khối cây mã hóa chroma (CTB) dựa vào sơ đồ phân vùng chroma có cùng việc phân vùng đệ quy với sơ đồ luma.

20. Vật lưu trữ đọc được bằng máy tính không tạm thời lưu trữ dòng bit của video được tạo ra bởi phương pháp được thực hiện bởi thiết bị xử lý video, trong đó phương pháp này bao gồm các bước:

xác định, đối với lát cắt video bao gồm khối cha luma và khối cha chroma, liệu có chia khối cha luma và khối cha chroma thành một hoặc nhiều khối luma và một hoặc nhiều khối chroma một cách tương ứng hay không, trong đó kiểu chế độ của khối luma hoặc kiểu chế độ của khối chroma được xác định dựa vào trị số của biến, và trong đó trị số của biến được suy ra dựa vào định dạng màu của lát cắt video; và tạo ra dòng bit dựa vào việc xác định này,

trong đó khối cha luma được tạo ra từ khối cây mã hóa (CTB) dựa vào sơ đồ phân vùng luma bao gồm việc phân vùng đệ quy, và trong đó khối cha chroma được tạo ra từ khối cây mã hóa chroma (CTB) dựa vào sơ đồ phân vùng chroma có cùng việc phân vùng đệ quy với sơ đồ luma.

1/26



Ảnh hiện tại

FIG. 1

2/26

Khối đã được mã hóa
trong chế độ bảng màu

Bảng màu

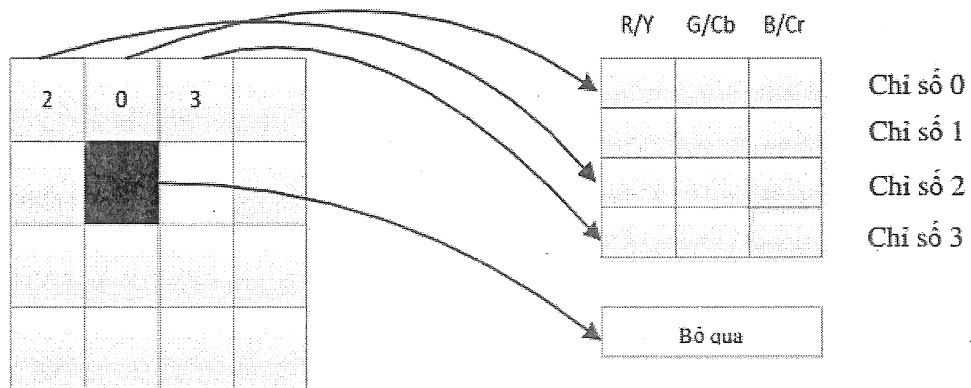


FIG. 2

3/26

Bảng màu trước				Bảng màu hiện tại				
Chỉ số	G/Y	B/Cb	R/Cr	Pred flag	Chỉ số	G/Y	B/Cb	R/Cr
0	G0	B0	R0	1	0	G0	B0	R0
1	G1	B1	R1	0				
2	G2	B2	R2	1	1	G2	B2	R2
3	G3	B3	R3	1	2	G3	B3	R3
4	G4	B4	R4	0				
5	G5	B5	R5	0				
					3	G3N	B3N	R3N
					4	G4N	B4N	R4N

 Chỉ mục của bảng màu sử dụng lại (3)
 Chỉ mục của bảng màu mới (2), được báo hiệu

FIG. 3

4/26

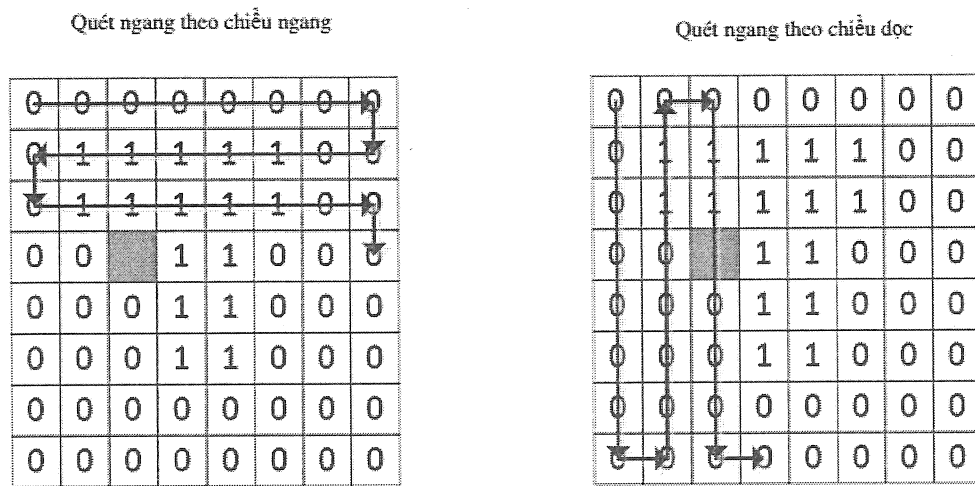


FIG. 4

5/26

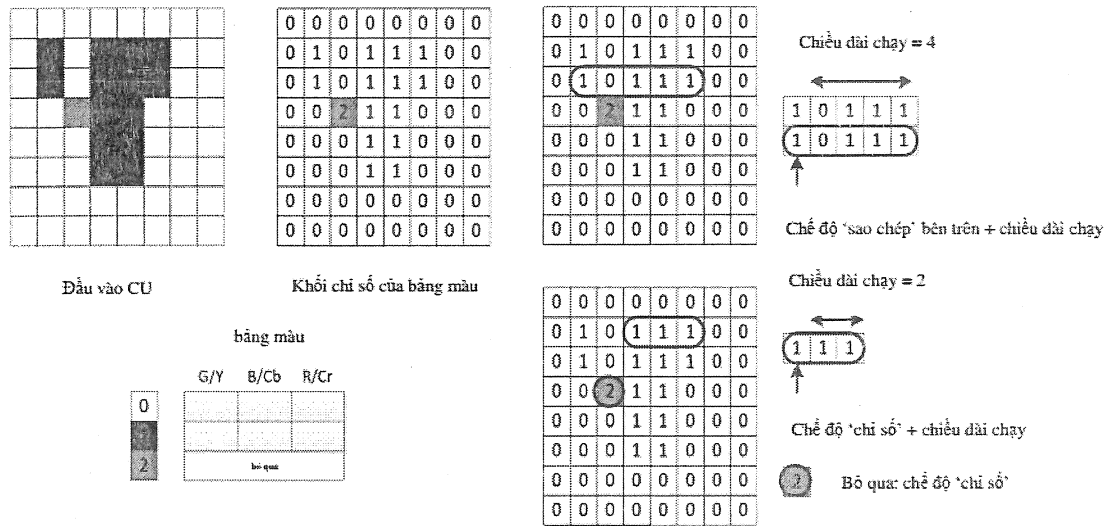


FIG. 5

6/26

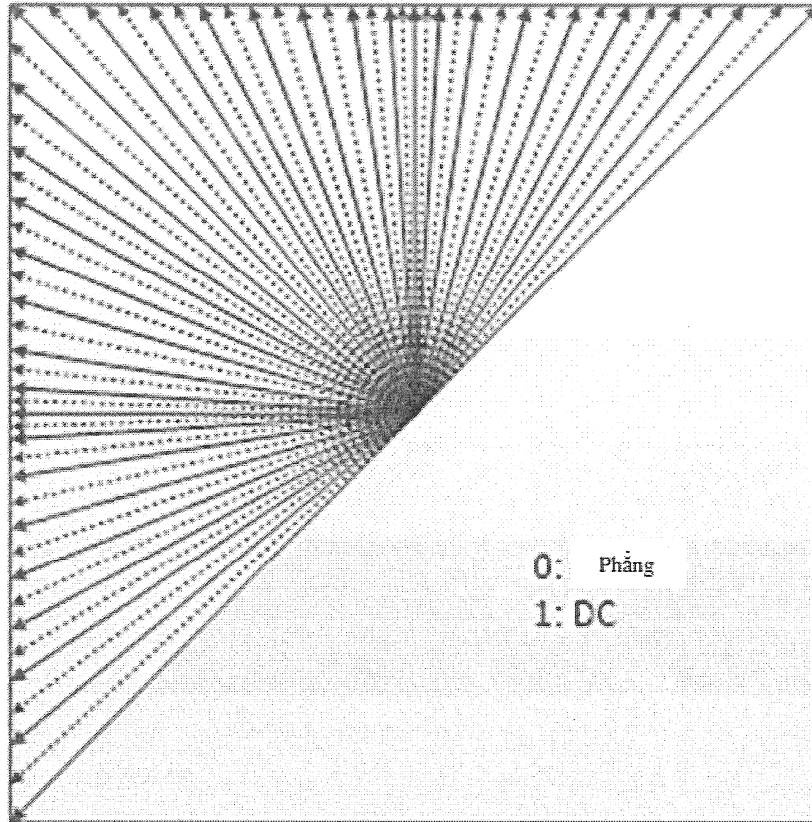


FIG. 6

7/26

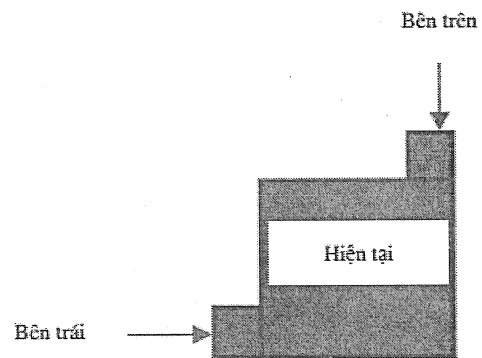


FIG. 7

8/26

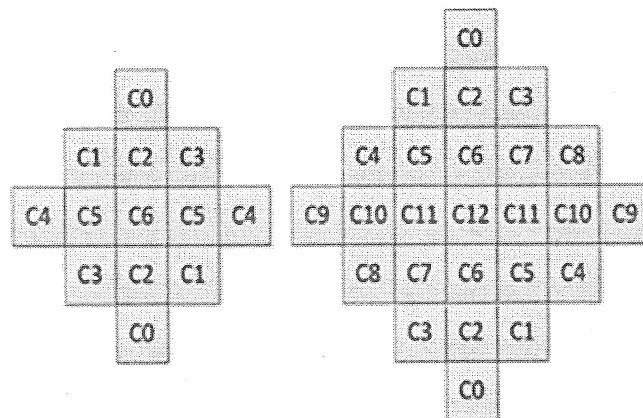


FIG. 8

9/26

V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V

H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H

D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1

D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2

FIG. 9

10/26

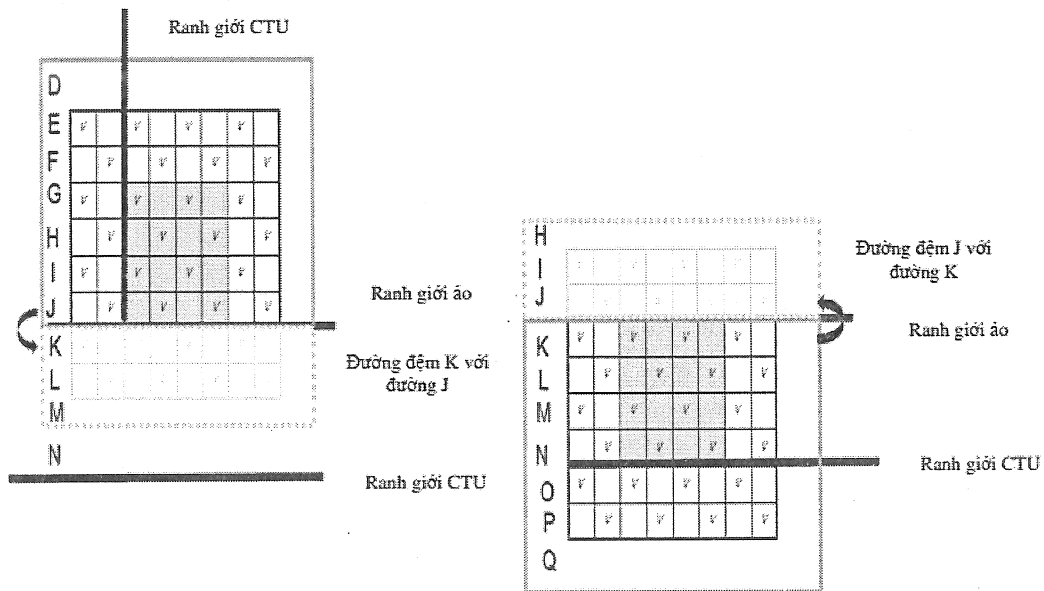


FIG. 10

11/26

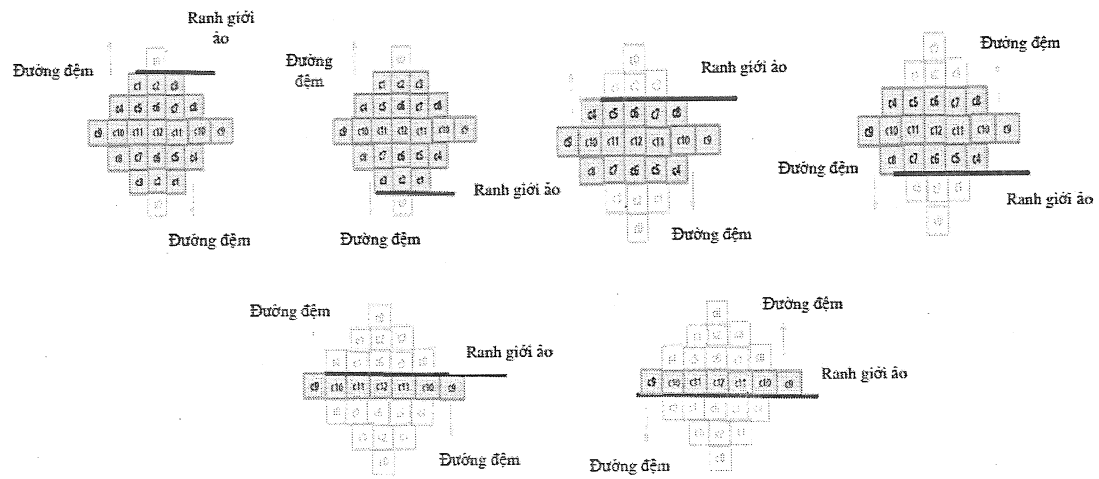


FIG. 11

12/26

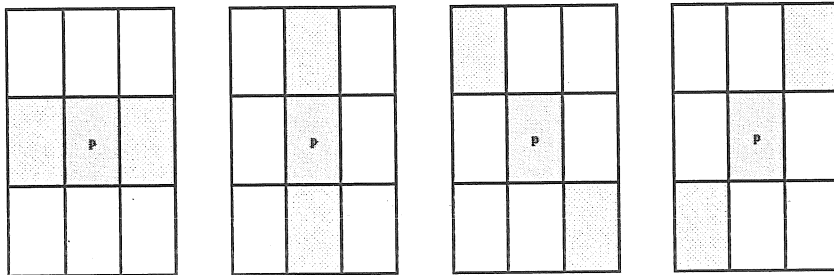


FIG. 12

13/26

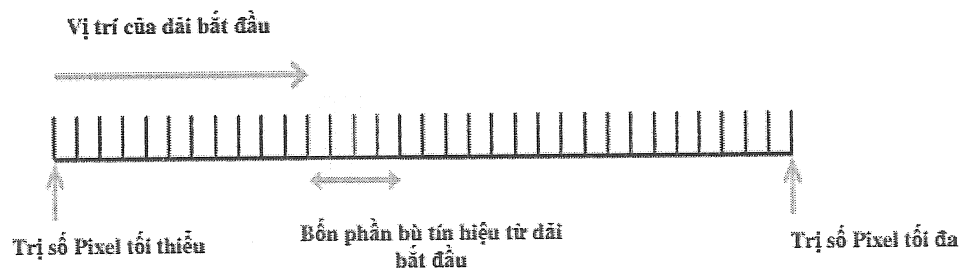


FIG. 13

14/26

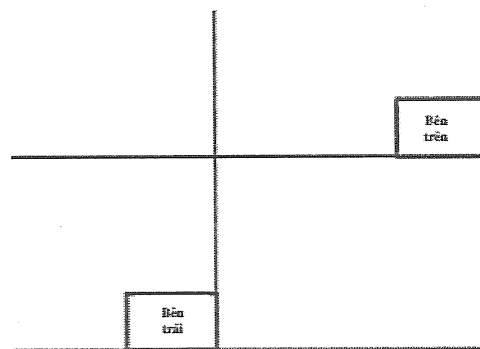


FIG. 14

15/26

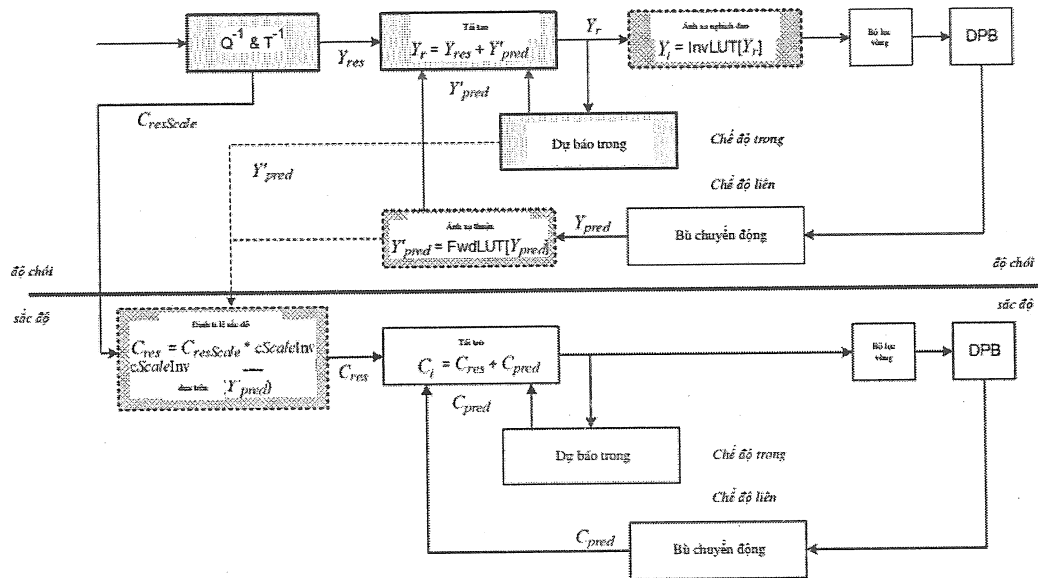


FIG. 15

16/26

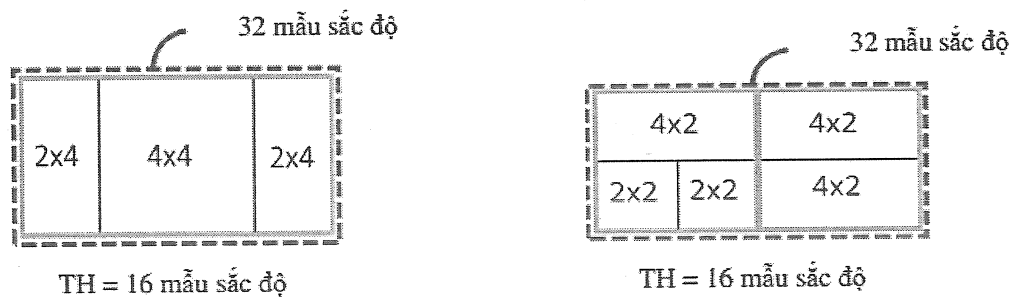


FIG. 16

17/26

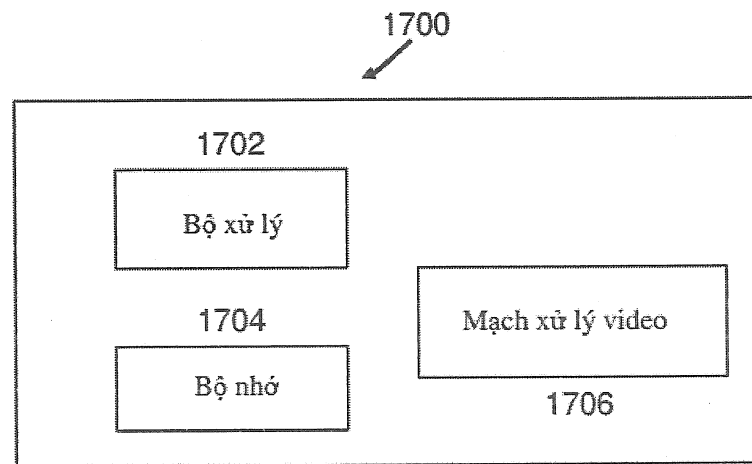


FIG. 17

18/26

1800

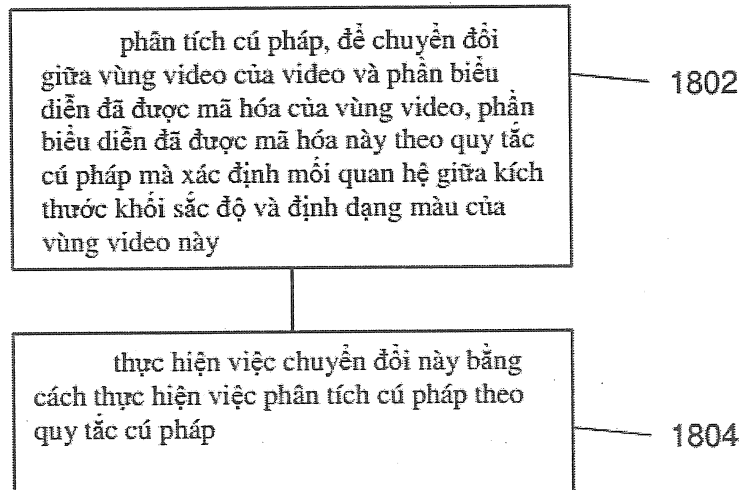


FIG. 18

19/26

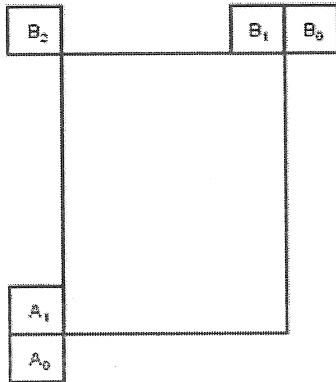


FIG. 19

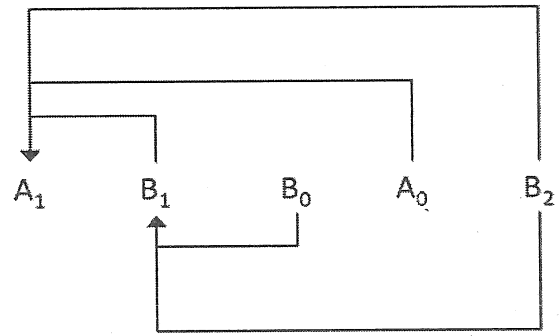


FIG. 20

20/26

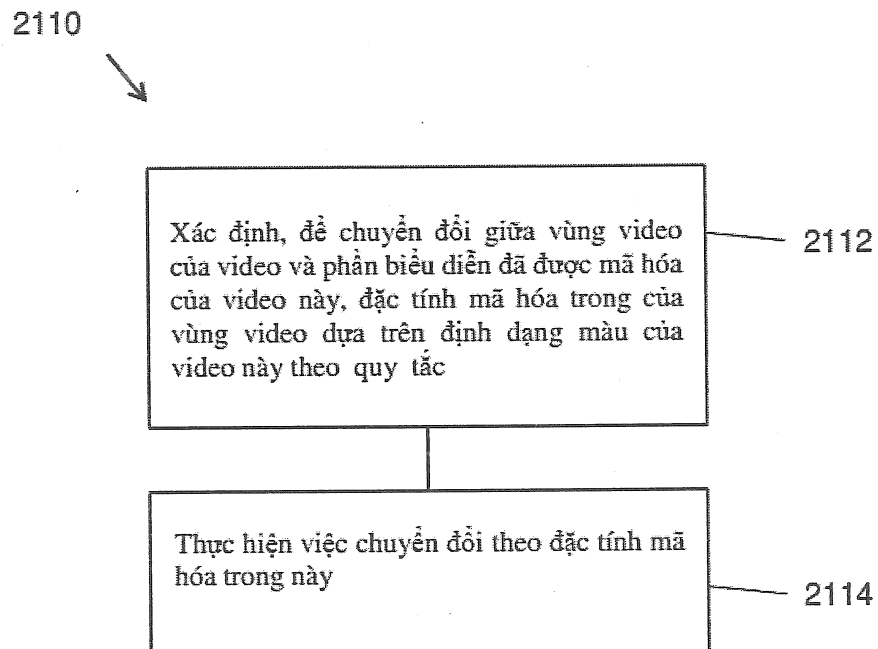


FIG. 21A

21/26

2120



Thực hiện việc chuyển đổi giữa video và
phần biểu diễn đã được mã hóa của video
này

2122

FIG. 21B

22/26

2130



Xác định, để chuyển đổi giữa vùng video của video và phân biểu diễn đã được mã hóa của video này, xem liệu việc và/hoặc cách giới hạn kích thước của khối dự đoán màu sắc nhỏ nhất trong khối dự đoán đối với vùng video này có được kích hoạt theo quy tắc hay không

2132

Thực hiện việc chuyển đổi dựa trên việc xác định này

2134

FIG. 21C

23/26

2140



Xác định, để chuyển đổi giữa khối video hiện tại của video và phân biểu diễn đã được mã hóa của video này, khả năng áp dụng sơ đồ phân vùng đối với khối video hiện tại theo quy tắc

2142

Thực hiện việc chuyển đổi dựa trên việc xác định này

2144

FIG. 21D

24/26

2150



Xác định, để chuyển đổi giữa khối video của video và phân biểu diễn đã được mã hóa của video này, xem liệu một chế độ liên có được kích hoạt theo quy tắc hay không

2152

Thực hiện việc chuyển đổi dựa trên việc xác định này

2154

FIG. 21E

25/26

2160



Xác định, để chuyển đổi giữa vùng video của video và phân biểu diễn đã được mã hóa của video này, dựa trên quy tắc, xem liệu việc sử dụng chế độ bảng màu có được phép đối với vùng video này hay không

2162

Thực hiện việc chuyển đổi dựa trên việc xác định này

2164

FIG. 21F

26/26

2170

Xác định, để chuyển đổi giữa khối video hiện tại của video và phân biểu diễn đã được mã hóa của video này, xem liệu một sơ đồ phân vùng nhất định có được phép đổi với khối video hiện tại này hay không theo quy tắc mà phụ thuộc vào loại chế độ mã hóa dùng để biểu diễn khối video hiện tại này trong phân biểu diễn đã được mã hóa và cờ của khối video hiện tại này

2172

Thực hiện việc chuyển đổi dựa trên việc xác định này

2174

FIG. 21G