



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

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

(11)



1-0050411

(51)^{2020.01} H04N 19/119; H04N 19/96; H04N
19/186; H04N 19/132; H04N 19/176

(13) B

(21) 1-2022-01956

(22) 21/09/2020

(86) PCT/CN2020/116472 21/09/2020

(87) WO 2021/052494 25/03/2021

(30) PCT/CN2019/107139 21/09/2019 CN; PCT/CN2019/108760 27/09/2019 CN

(45) 25/08/2025 449

(43) 25/08/2022 413A

(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. P.R. China

2. BYTEDANCE INC. (US)

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

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

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

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

(21) 1-2022-01956

(57) Sáng chế đề xuất phương pháp xử lý video. Phương pháp bao gồm bước thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng, do khối sắc độ có kích thước $M \times N$, khối sắc độ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

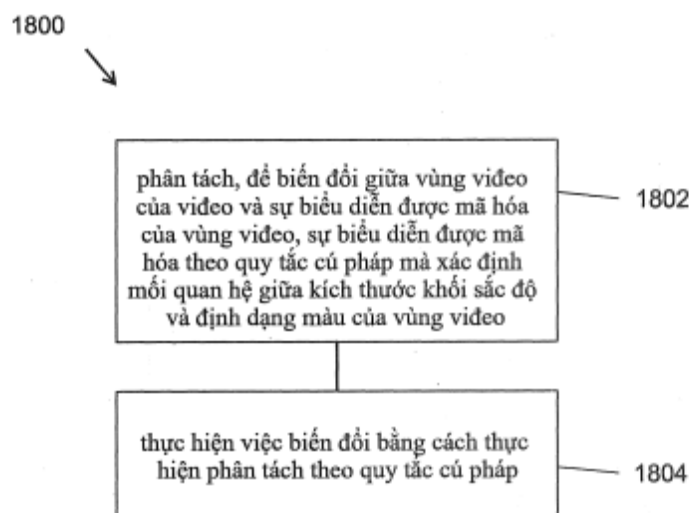


FIG. 18

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các công nghệ mã hóa và giải mã video và hình ảnh.

Tình trạng kỹ thuật của sáng chế

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

Bản chất kỹ thuật của sáng chế

Các kỹ thuật theo sáng chế có thể được sử dụng bởi các phương án của bộ giải mã hoặc bộ tạo mã video hoặc hình ảnh mà trong đó các ảnh tham chiếu được sử dụng trong việc mã hóa hoặc giải mã video.

Theo một khía cạnh ví dụ, phương pháp xử lý video được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc;

trong đó quy tắc chỉ rõ rằng, do khối sắc độ có kích thước $M \times N$, khối sắc độ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

Theo khía cạnh ví dụ khác, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được

phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh trong trường hợp sự phân chia cây kép được sử dụng, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

Theo một khía cạnh ví dụ khác nữa, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh trong trường hợp sự phân chia cây đơn được sử dụng, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

Theo một khía cạnh ví dụ khác nữa, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa khối sắc độ của vùng video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ dự báo ngoài ảnh và trong ảnh kết hợp (combined inter and intra prediction - CIIP) mà kết hợp dự báo ngoài ảnh với dự báo trong ảnh, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ.

Theo một khía cạnh ví dụ khác nữa, phương pháp xử lý video khác được bộc lộ. Phương pháp này bao gồm bước thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được mã hóa của video theo quy tắc, trong đó quy tắc chỉ rõ rằng việc liệu kích thước chuyển đổi tối đa có thể áp dụng cho vùng video có được báo tín hiệu hay không, và cách thức được báo tín hiệu như thế nào, là phụ thuộc vào kiểu chế độ mã hóa được sử dụng để biểu diễn vùng video trong sự biểu diễn được mã hóa, trong đó kiểu chế độ mã hóa là (i) chế độ mã hóa gần như không tổn thất mà trong đó thông số lượng tử hóa (quantization parameter - QP) được áp dụng cho vùng video là nhỏ hơn ngưỡng, (ii) chế độ mã hóa không

tồn thất mà trong đó việc chuyển đổi dữ liệu dư đối với vùng video được vòng tránh, hoặc (iii) chế độ mã hóa có tồn thất mà trong đó việc chuyển đổi dữ liệu dư đối với vùng video được bỏ qua.

Theo một khía cạnh ví dụ khác nữa, phương pháp được mô tả trên đây có thể được thực hiện bởi thiết bị tạo mã video mà bao gồm bộ xử lý.

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

Theo một khía cạnh ví dụ khác nữa, các phương pháp này có thể được thực hiện dưới dạng các lệnh thực thi được bằng bộ xử lý và được lưu trữ trong vật ghi chương trình máy tính đọc được.

Các khía cạnh này và khác nữa sẽ được mô tả tiếp trong bản mô tả này.

Mô tả vắn tắt các hình vẽ

FIG. 1 thể hiện ví dụ về công cụ mã hóa bản sao khối trong ảnh.

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

FIG. 3 thể hiện ví dụ về việc sử dụng bộ dự báo bảng màu để báo tín hiệu các mục nhập bảng màu.

FIG. 4 thể hiện ví dụ trong số các ví dụ về quét đường chuyển ngang và dọc.

FIG. 5 là hình vẽ thể hiện các ví dụ về mã hóa các chỉ số bảng màu.

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

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

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

FIG. 9 thể hiện ví dụ về phép tính Laplace được lấy mẫu con.

FIG. 10 thể hiện ví dụ về việc phân loại khối được cải biến ở các biên ảo.

FIG. 11 là ví dụ minh họa về việc lọc ALF cải biến đối với thành phần độ sáng ở các biên ảo.

FIG. 12 là hình vẽ thể hiện các ví dụ về bốn mẫu hình 3-điểm ảnh 1-D để phân loại điểm ảnh trong EO.

FIG. 13 thể hiện bốn băng được nhóm lại với nhau và được biểu diễn bằng vị trí bắt đầu băng của chúng.

FIG. 14 thể hiện các khối lân cận trên cùng và bên trái được sử dụng trong dẫn xuất trọng số CIIP.

FIG. 15 thể hiện cấu trúc ánh xạ độ sáng với chia tỷ lệ sắc độ.

FIG. 16 là hình vẽ thể hiện các ví dụ về SCIPU.

Các hình vẽ FIG. 17A và FIG. 17B là các sơ đồ khối của các ví dụ về nền tảng phần cứng được sử dụng để thực hiện các kỹ thuật được mô tả trong bản mô tả này.

FIG. 18 là lưu đồ của phương pháp xử lý video được nêu làm ví dụ.

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

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

FIG. 21 thể hiện lưu đồ của phương pháp xử lý video được nêu làm ví dụ.

FIG. 22 là sơ đồ khối mà thể hiện hệ thống mã hóa video được nêu làm ví dụ.

FIG. 23 là sơ đồ khối mà thể hiện bộ tạo mã theo một số phương án của sáng chế.

FIG. 24 là sơ đồ khối mà thể hiện bộ giải mã theo một số phương án của sáng chế.

Mô tả chi tiết sáng chế

Bản mô tả này đề xuất các kỹ thuật khác nhau mà có thể được sử dụng bởi bộ giải mã của các dòng bit hình ảnh hoặc video để nâng cao chất lượng của video hoặc các hình ảnh kỹ thuật số được giải nén hoặc được giải mã. Để ngắn gọn, thuật ngữ “video” được sử dụng ở đây bao gồm cả dãy của các ảnh (theo truyền thống được gọi là video) và các hình ảnh riêng lẻ. Hơn nữa, bộ tạo mã video cũng có thể thực hiện các kỹ thuật này trong suốt quá trình tạo mã nhằm tái tạo các khung được giải mã được sử dụng để tạo mã tiếp.

Các tiêu đề của các phần được sử dụng trong bản mô tả này để cho dễ hiểu và không giới hạn các phương án và các kỹ thuật ở các phần tương ứng. Như vậy, các phương án từ một phần này có thể được kết hợp với các phương án từ các phần khác.

1. Tóm tắt

Sáng chế đề cập đến các công nghệ mã hóa video. Cụ thể là, sáng chế đề xuất mã hóa bảng màu với việc sử dụng sự biểu diễn dựa trên các màu cơ bản trong quy trình mã hóa video. Sáng chế có thể được áp dụng cho chuẩn mã hóa video hiện có như HEVC, hoặc chuẩn (Versatile Video Coding) mà đang được hoàn thiện. Sáng chế cũng có thể áp dụng được cho các chuẩn mã hóa video hoặc bộ mã hóa-giải mã video tương lai.

2. Mở đầu

Các chuẩn mã hóa video đã tiến hóa chủ yếu thông qua việc phát triển các chuẩn ITU-T và ISO/IEC đã được biết rộng rãi. ITU-T đã tạo ra H.261 và H.263, ISO/IEC đã tạo ra MPEG-1 và MPEG-4 Visual, và hai tổ chức này cùng nhau tạo ra các chuẩn H.262/MPEG-2 Video và H.264/MPEG-4 Advanced Video Coding (AVC) và H.265/HEVC. Kể từ chuẩn H.262 trở đi, các chuẩn mã hóa video dựa trên cấu trúc mã hóa video lai trong đó việc dự báo theo thời gian cộng với mã hóa chuyển đổi được sử dụng. Để nghiên cứu các công nghệ mã hóa video tương lai sau HEVC, Joint Video Exploration Team (JVET) được đồng thành lập bởi VCEG và MPEG vào năm 2015. Kể từ đó, nhiều phương pháp mới đã được chấp nhận bởi JVET và được đưa vào phần mềm tham chiếu tên là Joint Exploration Model (JEM). Vào tháng tư năm 2018, Joint Video Expert Team (JVET) giữa VCEG (Q6/16) và ISO/IEC JTC1 SC29/WG11 (MPEG) đã được thành lập để làm việc với chuẩn VVC nhằm làm giảm 50% tốc độ bit so với HEVC.

2.1 Bản sao khối trong ảnh

Bản sao khối trong ảnh (Intra Block Copy - IBC), còn gọi là tham chiếu ảnh hiện tại, đã được chấp nhận trong các phần mở rộng mã hóa nội dung màn hình HEVC (HEVC-SCC Screen Content Coding) và mô hình kiểm định VVC (VTM-4.0) hiện tại. IBC mở rộng khái niệm bù chuyển động từ mã hóa khung

ngoài ảnh sang mã hóa khung trong ảnh. Như được thể hiện trên FIG. 1, khối hiện tại được dự báo bằng khối tham chiếu trong cùng ảnh khi IBC được áp dụng. Các mẫu trong khối tham chiếu cần phải được tái tạo trước khi khối hiện tại được mã hóa hoặc được giải mã. Mặc dù IBC không quá hiệu quả đối với phần lớn các dãy được chụp bằng camera, nó thể hiện các độ lợi mã hóa đáng kể đối với nội dung màn hình. Lý do là có nhiều mẫu hình lặp lại, chẳng hạn các biểu tượng và các ký tự văn bản trong ảnh nội dung màn hình. IBC có thể loại bỏ một cách hiệu quả sự dư thừa giữa các mẫu hình lặp lại này. Trong chuẩn HEVC-SCC, đơn vị mã hóa (coding unit - CU) được mã hóa ngoài ảnh có thể áp dụng IBC nếu nó chọn ảnh hiện tại làm ảnh tham chiếu của mình. Trong trường hợp này MV được đổi tên thành vectơ khối (Block Vector -BV), và BV luôn luôn có độ chính xác số nguyên-điểm ảnh. Để tương thích với tiêu sử chính HEVC, ảnh hiện tại được đánh dấu là ảnh tham chiếu “dài hạn” trong bộ đệm ảnh được giải mã (Decoded Picture Buffer - DPB). Cần lưu ý rằng tương tự như vậy, trong nhiều chuẩn mã hóa video cảnh nhìn/3D, ảnh tham chiếu cảnh nhìn ngoài ảnh cũng được đánh dấu là ảnh tham chiếu “dài hạn”.

Tiếp theo BV để tìm khối tham chiếu của nó, việc dự báo có thể được tạo ra bằng cách sao chép khối tham chiếu. Có thể thu được phần dư bằng cách trừ đi các điểm ảnh tham chiếu từ các tín hiệu gốc. Sau đó bước chuyển đổi và 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 khối trong ảnh.

Tuy nhiên, khi khối tham chiếu ở bên ngoài của ảnh, hoặc xếp chồng với khối hiện tại, hoặc bên ngoài vùng được tái tạo, hoặc bên ngoài vùng hợp lệ bị hạn chế bởi một số điều kiện ràng buộc, thì một phần hoặc tất cả các giá trị điểm ảnh là không được xác định. Về cơ bản, có hai giải pháp để giải quyết vấn đề này. Một là không cho phép tình huống này, ví dụ trong sự tương thích dòng bit. Giải pháp khác là áp dụng việc đệm đối với các giá trị điểm ảnh không được xác định đó. Các mục nhỏ sau đây mô tả chi tiết các giải pháp này.

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

Trong các phần mở rộng mã hóa nội dung màn hình của HEVC, khi khối sử dụng ảnh hiện tại làm tham chiếu, thì cần phải đảm bảo rằng toàn bộ khối

tham chiếu ở bên trong vùng được tái tạo khả dụng, như được chỉ báo trong văn bản đặc tả sau đây:

The variables `offsetX` and `offsetY` are derived as follows:

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

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

It is a requirement of bitstream conformance that when the reference picture is the current picture, the luma motion vector `mvLX` shall obey the following constraints:

- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with $(x_{\text{Curr}}, y_{\text{Curr}})$ set equal to $(x_{\text{Cb}}, y_{\text{Cb}})$ and the neighbouring luma location $(x_{\text{NbY}}, y_{\text{NbY}})$ set equal to $(x_{\text{Pb}} + (\text{mvLX}[0] \gg 2) - \text{offsetX}, y_{\text{Pb}} + (\text{mvLX}[1] \gg 2) - \text{offsetY})$ as inputs, the output shall be equal to TRUE.
- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with $(x_{\text{Curr}}, y_{\text{Curr}})$ set equal to $(x_{\text{Cb}}, y_{\text{Cb}})$ and the neighbouring luma location $(x_{\text{NbY}}, y_{\text{NbY}})$ set equal to $(x_{\text{Pb}} + (\text{mvLX}[0] \gg 2) + n_{\text{PbW}} - 1 + \text{offsetX}, y_{\text{Pb}} + (\text{mvLX}[1] \gg 2) + n_{\text{PbH}} - 1 + \text{offsetY})$ as inputs, the output shall be equal to TRUE.
- One or both of the following conditions shall be true:
 - The value of $(\text{mvLX}[0] \gg 2) + n_{\text{PbW}} + x_{\text{B1}} + \text{offsetX}$ is less than or equal to 0.
 - The value of $(\text{mvLX}[1] \gg 2) + n_{\text{PbH}} + y_{\text{B1}} + \text{offsetY}$ is less than or equal to 0.
- The following condition shall be true:

$$\begin{aligned} & (x_{\text{Pb}} + (\text{mvLX}[0] \gg 2) + n_{\text{PbSw}} - 1 + \text{offsetX}) / \text{CtbSizeY} - x_{\text{Cb}} / \text{CtbSizeY} \leq \\ & y_{\text{Cb}} / \text{CtbSizeY} - (y_{\text{Pb}} + (\text{mvLX}[1] \gg 2) + n_{\text{PbSh}} - 1 + \text{offsetY}) / \text{CtbSizeY} \end{aligned} \quad (8-106)$$

Do đó, trường hợp mà khối tham chiếu xếp chồng với khối hiện tại hoặc khối tham chiếu ở bên ngoài của ảnh sẽ không xảy ra. Không cần phải đệm khối tham chiếu hoặc khối dự báo.

2.3 IBC trong mô hình kiểm định VVC

Trong mô hình kiểm định VVC hiện tại, tức là thiết kế VTM-4.0, toàn bộ khối tham chiếu phải cùng với đơn vị cây mã hóa (coding tree unit - CTU) hiện tại và không xếp chồng với khối hiện tại. Do đó, không cần thiết phải đệm khối tham chiếu hoặc khối dự báo. Cờ IBC được mã hóa dưới dạng chế độ dự báo của CU hiện tại. Do đó, tổng cộng có 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, chỉ số chỉ vào mục nhập trong danh sách ứng viên hợp nhất IBC được phân tách từ dòng bit. Việc xây dựng danh sách hợp nhất IBC có thể được tổng kết theo chuỗi các bước sau đây:

- Bước 1: Dẫn xuất các ứng viên không gian

- Bước 2: Chèn các ứng viên HMVP
- Bước 3: Chèn các ứng viên trung bình theo cặp

Trong bước dẫn xuất các ứng viên hợp nhất không gian, tối đa bốn ứng viên hợp nhất được chọn trong số các ứng viên nằm trong các vị trí được chỉ ra trên FIG. 19. Thứ tự dẫn xuất là A1, B1, B0, A0 và B2. Vị trí B2 chỉ được xem xét khi PU bất kỳ của vị trí A1, B1, B0, A0 là không khả dụng (ví dụ vì nó thuộc về lát hoặc ô ảnh khác) hoặc không được mã hóa bằng chế độ IBC. Sau khi ứng viên ở vị trí A1 được bổ sung, việc chèn các ứng viên còn lại phải trải qua kiểm tra độ dư để đảm bảo rằng các ứng viên với cùng thông tin chuyển động được loại bỏ khỏi danh sách nhằm cải thiện hiệu quả mã hóa. Để giảm mức độ phức tạp trong tính toán, không phải tất cả các cặp ứng viên tiềm năng được xem xét trong bước kiểm tra độ dư nêu trên. Thay vào đó chỉ các cặp được liên kết với mũi tên trên FIG. 20 được xem xét và ứng viên chỉ được bổ sung vào danh sách nếu ứng viên tương ứng được sử dụng để kiểm tra độ dư không có cùng thông tin chuyển động.

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

Cuối cùng, các ứng viên trung bình theo cặp được chèn vào danh sách hợp nhất IBC.

Khi khối tham chiếu được xác định bởi ứng viên hợp nhất ở bên ngoài của ảnh, hoặc xếp chồng với khối hiện tại, hoặc bên ngoài vùng được tái tạo, hoặc bên ngoài vùng hợp lệ bị hạn chế bởi một số điều kiện ràng buộc, thì ứng viên hợp nhất được gọi là ứng viên hợp nhất không hợp lệ.

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

2.3.2 Chế độ IBC AMVP

Trong chế độ IBC AMVP, chỉ số AMVP chỉ tới mục nhập trong danh sách IBC AMVP được phân tách từ dòng bit. Việc xây dựng danh sách IBC AMVP có thể được tổng kết theo chuỗi các bước sau đây:

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

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

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

2.4 Chế độ bảng màu

Ý tưởng cơ bản của chế độ bảng màu là các mẫu trong CU được biểu diễn bởi một tập nhỏ các giá trị màu đại diện. Tập này được gọi là bảng màu. Và cũng có thể chỉ báo mẫu mà ở bên ngoài bảng màu bằng cách báo tín hiệu ký hiệu thoát được theo sau bởi (có thể được lượng tử hóa) các giá trị thành phần. Loại mẫu này được gọi là mẫu thoát. Chế độ bảng màu được minh họa trên FIG. 2.

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

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

Trong chế độ bảng màu trong HEVC-SCC, cách thức dự báo được sử dụng để mã hóa bảng màu và ánh xạ chỉ số.

2.5.1 Mã hóa các mục nhập bảng màu

Để mã hóa các mục nhập bảng màu, bộ dự báo bảng màu được duy trì. Kích thước tối đa của bảng màu cũng như bộ dự báo bảng màu được báo tín hiệu trong SPS. Trong HEVC-SCC, `palette_predictor_initializer_present_flag` được đưa vào trong PPS. Khi cờ này bằng 1, các mục nhập để khởi tạo bộ dự báo bảng màu được báo tín hiệu trong dòng bit. Bộ dự báo bảng màu được khởi tạo ở đầu của mỗi hàng CTU, mỗi lát và mỗi ô ảnh. Tùy theo giá trị của `palette_predictor_initializer_present_flag`, bộ dự báo bảng màu được thiết lập lại bằng 0 hoặc được khởi tạo bằng cách sử dụng các mục nhập bộ khởi tạo bộ dự báo bảng màu được báo tín 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 một cách rõ ràng việc khởi tạo bộ dự báo bảng màu ở bậc PPS.

Đối với mỗi mục nhập trong bộ dự báo bảng màu, cờ tái sử dụng được báo tín hiệu để chỉ báo việc liệu nó có là một phần của bảng màu hiện tại hay không. Điều này được minh họa trên FIG. 3. Các cờ tái sử dụng được gửi đi bằng cách sử dụng mã hóa chiều dài chạy của các số không. Sau đó, số lượng của các mục nhập bảng màu mới được báo tín hiệu bằng cách sử dụng mã hóa Golomb hàm mũ có bậc 0. Cuối cùng, các giá trị thành phần đối với các mục nhập bảng màu mới được báo tín hiệu.

Fig. 3 thể hiện ví dụ về việc sử dụng bộ dự báo bảng màu để báo tín hiệu các mục nhập bảng màu.

2.5.2 Mã hóa các chỉ số bảng màu

Các chỉ số bảng màu được mã hóa bằng cách sử dụng quét đường chuyên ngang và dọc như được thể hiện trên FIG. 4. Thứ tự quét được báo tín hiệu một cách rõ ràng trong dòng bit bằng cách sử dụng `palette_transpose_flag`. Đối với phần còn lại của đoạn con thì giả định rằng việc quét là ngang.

Fig. 4 là hình vẽ thể hiện các ví dụ về quét đường chuyên ngang và dọc.

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

Thứ tự cú pháp này được hoàn thành như dưới đây. Trước tiên số lượng của các giá trị chỉ số đối với CU được báo tín hiệu. Bước này được theo sau bằng việc báo tín hiệu các giá trị chỉ số thực tế đối với toàn bộ CU bằng cách sử dụng mã hóa nhị phân được cắt ngắn. Cả số lượng của các chỉ số cũng như các giá trị chỉ số đều được mã hóa trong chế độ đi vòng. Điều này nhóm các học đi vòng liên quan đến chỉ số lại với nhau. Sau đó chế độ mẫu bảng màu (nếu cần thiết) và việc chạy được báo tín hiệu theo cách thức xen kẽ. Cuối cùng, các giá trị thoát thành phần tương ứng với các mẫu thoát đối với toàn bộ CU được nhóm lại với nhau và được mã hóa trong chế độ đi vòng.

Phần tử cú pháp bổ sung, `last_run_type_flag`, được báo tín hiệu sau khi báo tín hiệu các giá trị chỉ số. Phần tử cú pháp này, cùng với số lượng của các chỉ số, loại bỏ sự cần thiết phải báo tín hiệu giá trị chạy tương ứng với lần chạy cuối cùng trong khối.

Trong HEVC-SCC, chế độ bảng màu cũng được phép đối với 4:2:2, 4:2:0, và các định dạng sắc độ đơn sắc. Việc báo tín hiệu các mục nhập bảng màu và các chỉ số bảng màu phần lớn là giống hệt đối với tất cả các định dạng sắc độ. Trong trường hợp các định dạng không đơn sắc, mỗi mục nhập bảng màu bao gồm 3 thành phần. Đối với định dạng đơn sắc, mỗi mục nhập bảng màu bao gồm một thành phần. Đối với các hướng sắc độ được lấy mẫu con, các mẫu sắc độ được liên kết với các chỉ số mẫu độ sáng mà chia hết cho 2. Sau khi tái tạo các chỉ số bảng màu đối với CU, nếu mẫu chỉ có một thành phần liên kết với nó, thì chỉ thành phần thứ nhất của mục nhập bảng màu được sử dụng. Sự khác biệt duy nhất trong việc báo tín hiệu là đối với các giá trị thành phần thoát. Đối với mỗi mẫu thoát, số lượng của các giá trị thành phần thoát được báo tín hiệu có thể là khác nhau tùy theo số lượng của các thành phần được liên kết với mẫu đó.

Trong VVC, cấu trúc mã hóa cây kép được sử dụng khi mã hóa các lát trong ảnh, cho nên thành phần độ sáng và hai thành phần sắc độ có thể có bảng màu và các chỉ số bảng màu khác nhau. Ngoài ra, hai thành phần sắc độ chia sẻ cùng bảng màu và các chỉ số bảng màu.

FIG. 5 là hình vẽ thể hiện các ví dụ về mã hóa các chỉ số bảng màu.

2.6 Mã hóa chế độ trong ảnh trong VVC

Để nắm được các hướng rìa tùy ý được trình bày trong video tự nhiên, số lượng của các chế độ trong ảnh định hướng trong VTM5 được mở rộng từ 33, như được sử dụng trong HEVC, lên tới 65. Các chế độ định hướng mới không phải trong HEVC được ký hiệu dưới dạng các mũi tên đứt nét đỏ trên FIG. 6 và các chế độ phẳng và DC vẫn giữ nguyên. Các chế độ dự báo trong ảnh định hướng đậm hơn này áp dụng cho tất cả các kích thước khối và cho cả dự báo trong ảnh độ sáng lẫn sắc độ.

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

Trong HEVC, mỗi khối được mã hóa trong ảnh có hình dạng vuông và chiều dài của mỗi cạnh của nó là lũy thừa 2. Do đó, không cần các hoạt động chia để tạo ra bộ dự báo trong ảnh bằng cách sử dụng chế độ DC. Trong VTM5, các khối có thể có hình dạng chữ nhật mà cần phải sử dụng hoạt động chia trên khối trong trường hợp tổng quát. Để tránh các hoạt động chia để dự báo DC, chỉ cạnh dài hơn được sử dụng để tính giá trị trung bình cho các khối không vuông.

Fig. 6 thể hiện ví dụ về 67 chế độ dự báo trong ảnh.

Để giảm thấp mức độ phức tạp của việc tạo danh sách chế độ xác suất lớn nhất (most probable mode - MPM), phương pháp mã hóa chế độ trong ảnh với 6 MPM được sử dụng bằng cách xem xét hai chế độ trong ảnh lân cận khả dụng. Ba khía cạnh sau đây được xem xét để xây dựng danh sách MPM:

- Các chế độ trong ảnh mặc định
- Các chế độ trong ảnh lân cận
- Các chế độ trong ảnh dẫn xuất

Danh sách 6 MPM thống nhất được sử dụng cho các khối trong ảnh không phụ thuộc vào việc liệu các công cụ mã hóa MRL và ISP có được áp dụng hay không. Danh sách MPM được xây dựng dựa trên các chế độ trong ảnh của khối lân cận bên trái và bên trên. Giả sử chế độ của khối bên trái được ký hiệu là Trái và chế độ của khối bên trên được ký hiệu là Trên, danh sách MPM thống nhất được xây dựng như dưới đây (các khối bên trái và bên trên được thể hiện trên FIG. 7):

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

- Khi khối lân cận là không khả dụng, thì chế độ trong ảnh của nó được thiết lập thành Phẳng theo mặc định.

- Nếu cả hai chế độ Bên trái và Bên trên đều là các chế độ không phải góc:

- danh sách 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, còn chế độ kia là không phải góc:

- Thiết lập chế độ Max là chế độ lớn hơn trong Bên trái và Bên trên
- danh sách MPM $\rightarrow \{\text{Planar, Max, DC, Max -1, Max +1, Max -2}\}$

- Nếu cả hai Bên trái và Bên trên đều là góc và chúng khác nhau:

- Thiết lập chế độ Max là chế độ lớn hơn trong Bên trái và Bên trên
- Nếu hiệu số của chế độ Bên trái và Bên trên nằm trong khoảng từ 2 tới 62, bao gồm cả các giá trị biên thì

- danh sách MPM $\rightarrow \{\text{Planar, Left, Above, DC, Max -1, Max +1}\}$

- Ngược lại

- danh sách MPM $\rightarrow \{\text{Planar, Left, Above, DC, Max -2, Max +2}\}$

- Nếu cả Bên trái và Bên trên đều là góc và chúng giống nhau:

- danh sách MPM $\rightarrow \{\text{Planar, Left, Left -1, Left +1, DC, Left -2}\}$

Ngoài ra, học thứ nhất của từ mã chỉ số mpm là ngữ cảnh CABAC được mã hóa. Tổng cộng ba ngữ cảnh được sử dụng, tương ứng với việc khối trong ảnh hiện tại là MRL được phép, ISP được phép, hoặc khối trong ảnh thông thường.

Trong quá trình tạo danh sách 6 MPM, việc xén bớt được sử dụng để loại bỏ các chế độ trùng lặp sao cho chỉ các chế độ duy nhất có thể được bao gồm trong danh sách MPM. Để mã hóa entropi 61 chế độ không phải MPM, phép mã hóa nhị phân được cắt ngắn (Truncated Binary Code - TBC) được sử dụng.

Để mã hóa chế độ trong ảnh sắc độ, tổng cộng 8 chế độ trong ảnh được cho phép đối với mã hóa chế độ trong ảnh sắc độ. Các chế độ này bao gồm năm chế độ trong ảnh truyền thống và ba chế độ mô hình tuyến tính giao thành phần (CCLM, LM_A, và LM_L). Việc báo tín hiệu chế độ sắc độ và quá trình dẫn xuất được thể hiện trong Bảng 2-. Việc mã hóa chế độ sắc độ phụ thuộc trực tiếp vào chế độ dự báo trong ảnh của khối độ sáng tương ứng. Do cấu trúc phân chia

khối riêng rẽ đối với các thành phần độ sáng và sắc độ được phép trong lát I, một khối sắc độ có thể tương ứng với nhiều khối độ sáng. Do đó, đối với chế độ sắc độ DM, chế độ dự báo trong ảnh của khối độ sáng tương ứng bao phủ vị trí trung tâm của khối sắc độ hiện tại được thừa hưởng trực tiếp.

Bảng 2-4 – Dẫn xuất chế độ dự báo sắc độ từ chế độ độ sáng khi cclm_is được phép

Chế độ dự báo sắc độ	Chế độ dự báo trong ảnh độ sáng tương ứng				
	0	50	18	1	X ($0 \leq X \leq 66$)
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 phân khối dư lượng tử hóa (Quantized residual Block Differential Pulse-Code Modulation - QR-BDPCM)

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

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 ảnh được thực hiện trên toàn bộ khối bằng cách sao chép mẫu theo hướng dự báo (dự báo ngang hoặc dọc) tương tự như dự báo trong ảnh. Phần dư được lượng tử hóa và giá trị delta giữa phần dư được lượng tử hóa và giá trị được lượng tử hóa bộ dự báo (ngang hoặc dọc) của nó được mã hóa. Điều này có thể được mô tả như sau: Đối với khối có kích thước M (các hàng) $\times$ N (các cột), cho $r_{i,j}$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ là phần dư dự báo sau khi thực hiện dự báo trong ảnh ngang (sao chép giá trị điểm ảnh lân cận trái ngang qua khối được dự báo theo từng dòng) hoặc dọc (sao chép dòng lân cận trên cùng tới mỗi dòng trong khối được dự báo) bằng cách sử dụng các mẫu không được lọc từ các mẫu biên khối bên trên hoặc bên trái. Cho $Q(r_{i,j})$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ ký hiệu phiên bản được lượng tử hóa

của phần dư $r_{i,j}$, trong đó phần dư là hiệu số giữa khối gốc và các giá trị của khối được dự báo. Khi đó khối DPCM được áp dụng cho các mẫu phần dư được lượng tử hóa, dẫn đến mảng $\tilde{R}$ $M \times N$ được cải biến với các phần tử $\tilde{r}_{i,j}$. Khi BDPCM đọc được báo tín hiệu:

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

Đối với dự báo ngang, các quy tắc tương tự được áp dụng, và thu được các mẫu dư được lượng tử hóa bằng biểu thức

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

Các mẫu dư được lượng tử hóa $\tilde{r}_{i,j}$ được gửi tới bộ giải mã.

Về phía bộ giải mã, các tính toán trên đây được nghịch đảo để tạo ra $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$. 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 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 được lượng tử hóa nghịch đảo, $Q^{-1}(Q(r_{i,j}))$, được bổ sung vào các giá trị dự báo khối trong ảnh để tạo ra các giá trị mẫu được tái tạo.

Lợi ích chính của cách làm này là DPCM nghịch đảo có thể được thực hiện ngay trong khi đang phân tách hệ số chỉ bằng cách bổ sung bộ dự báo do các hệ số được phân tách hoặc có thể được thực hiện sau khi phân tách.

2.8 Bộ lọc vòng thích ứng

Trong VTM5, bộ lọc vòng thích ứng (Adaptive Loop Filter - ALF) với việc thích ứng bộ lọc dựa trên khối được áp dụng. Đối với thành phần độ sáng, một trong số 25 bộ lọc được chọn đối với mỗi khối 4×4 , dựa trên hướng và mức hoạt động của các gradient nội vùng.

2.8.1.1 Hình dạng bộ lọc

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

Fig. 8 là hình vẽ thể hiện các ví dụ về các hình dạng bộ lọc ALF (sắc độ: hình thoi 5×5 , độ sáng: hình thoi 7×7)

2.8.1.2 Phân loại khối

Đối với thành phần độ sáng, mỗi khối 4×4 được phân loại vào một lớp trong số 25 lớp. Chỉ số phân loại C được rút ra dựa trên tính định hướng D của nó và giá trị được lượng tử hóa của mức hoạt động $\hat{A}$, như dưới đây:

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

Để tính được D và $\hat{A}$, trước hết các gradient ngang, dọc và hai hướng đường chéo được tính toán bằng cách sử dụng hàm Laplace 1-D:

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

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

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

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (2-9-5)$$

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

Để giảm mức độ phức tạp của việc phân loại khối, phép tính Laplace 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 ví dụ về phép tính Laplace được lấy mẫu con. (a) Các vị trí được lấy mẫu con đối với gradient dọc (b) Các vị trí được lấy mẫu con đối với gradient ngang (c) Các vị trí được lấy mẫu con đối với gradient đường chéo (d) Các vị trí được lấy mẫu con đối với gradient đường chéo.

Sau đó các giá trị tối đa và tối thiểu D của các gradient theo các hướng ngang và dọc được thiết lập như:

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

Các giá trị tối đa và tối thiểu của gradien theo hai hướng đường chéo được thiết lập như:

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

Để rút ra giá trị của tính định hướng D , các giá trị này được so sánh với nhau và với hai ngưỡng t_1 và t_2 :

Bước 1. Nếu cả hai $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ và $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ đều là đúng, D được thiết lập về 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; ngược lại tiếp tục từ Bước 4.

Bước 3. Nếu $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D được thiết lập thành 2; ngược lại D được thiết lập thành 1.

Bước 4. Nếu $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D được thiết lập thành 4; ngược lại D được thiết lập thành 3.

Giá trị mức hoạt động A được tính toán như:

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

A còn được lượng tử hóa tới khoảng từ 0 tới 4, bao gồm cả các giá trị biên, và giá trị được lượng tử hóa được ký hiệu là $\hat{A}$.

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

2.8.1.3 Các chuyển đổi hình học của các hệ số bộ lọc và các giá trị xén

Trước khi lọc mỗi khối độ sáng 4×4 , các chuyển đổi hình học chẳng hạn quay hoặc lật chéo và dọc được áp dụng cho các hệ số bộ lọc $f(k, l)$ và cho các giá trị xén bộ lọc tương ứng $c(k, l)$ tùy theo các giá trị gradien được tính toán cho khối đó. Điều này tương đương với việc áp dụng các chuyển đổi này cho các mẫu trong vùng hỗ trợ bộ lọc. Ý tưởng là làm cho các khối khác nhau mà được áp dụng ALF trở nên giống nhau hơn bằng cách chỉnh định tính định hướng của chúng.

Ba chuyển đổi hình học, bao gồm lật dọc, chéo và quay được đề xuất:

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)

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

trong đó K là kích thước của bộ lọc và $0 \leq k, l \leq K - 1$ là các tọa độ hệ số, sao cho vị trí $(0,0)$ ở góc trên bên trái và vị trí $(K - 1, K - 1)$ ở góc dưới bên phải. Các chuyển đổi này được áp dụng cho các hệ số bộ lọc $f(k, l)$ và cho các giá trị xén $c(k, l)$ tùy theo các giá trị gradient được tính toán cho khối đó. Mỗi quan hệ giữa phép chuyển đổi và bốn gradient của bốn hướng được tổng kết trong bảng sau.

Bảng 2-5 - Ánh xạ gradient được tính toán cho một khối và các chuyển đổi

Các giá trị gradient	Chuyển đổi
$gd2 < gd1$ và $gh < gv$	Không có chuyển đổi
$gd2 < gd1$ và $gv < gh$	Chéo
$gd1 < gd2$ và $gh < gv$	Lật dọc
$gd1 < gd2$ và $gv < gh$	Quay

2.8.1.4 Báo tín hiệu các thông số bộ lọc

Trong VTM5, các thông số bộ lọc ALF được báo tín hiệu trong tập thông số thích ứng (Adaptation Parameter Set - APS). Trong một APS, có đến 25 tập của các hệ số bộ lọc độ sáng và các chỉ số giá trị xén, và có đến một tập của các hệ số bộ lọc sắc độ và các chỉ số giá trị xén có thể được báo tín hiệu. Để giảm các bit trên đầu, các hệ số bộ lọc có phân loại khác nhau có thể được hợp nhất. Trong tiêu đề lát, các chỉ số của các APS được sử dụng cho lát hiện tại được báo tín hiệu.

Các chỉ số giá trị xén, mà được giải mã từ APS, cho phép xác định các giá trị xén bằng cách sử dụng bảng Độ sáng của các giá trị xén và bảng Sắc độ của các giá trị xén. Các giá trị xén này phụ thuộc vào độ sâu bit nội tại. Cụ thể hơn, bảng Độ sáng của các giá trị xén và bảng Sắc độ của các giá trị xén thu được bằng các công thức sau:

$$\text{AlfClipL} = \left\{ \text{round} \left(2^{\frac{B(N-n+1)}{N}} \right) \text{ for } n \in [1..N] \right\}, \quad (2-9-12)$$

$$\text{AlfClipC} = \left\{ \text{round} \left(2^{(B-8)+8\frac{(N-n)}{N-1}} \right) \text{ for } n \in [1..N] \right\} \quad (2-9-13)$$

với B bằng độ sâu bit nội tại và N bằng 4 mà là số lượng của các giá trị xén được phép trong VTM5.0.

Quá trình lọc có thể được điều khiển ở bậc CTB. Cờ luôn luôn được báo tín hiệu để chỉ báo liệu ALF có được áp dụng cho CTB độ sáng hay không. CTB độ sáng có thể chọn tập bộ lọc trong số 16 tập bộ lọc cố định và các tập bộ lọc từ các APS. Chỉ số tập bộ lọc được báo tín hiệu đối với CTB độ sáng để chỉ báo tập bộ lọc nào được áp dụng. 16 tập bộ lọc cố định được xác định trước và được mã hóa cứng trong cả bộ tạo mã lẫn bộ giải mã.

Các hệ số bộ lọc được lượng tử hóa với chuẩn bằng 128. Nhằm hạn chế mức độ phức tạp phép nhân, sự tương thích dòng bit được áp dụng sao cho giá trị hệ số của vị trí không phải trung tâm sẽ nằm trong khoảng từ -2^7 tới $2^7 - 1$, bao gồm cả các giá trị biên. Hệ số vị trí trung tâm không được báo tín hiệu trong dòng bit và được coi là bằng 128.

2.8.1.5 Quá trình lọc

Ở phía bộ giải mã, khi ALF được phép đối với CTB, mỗi mẫu $R(i, j)$ bên trong CU được lọc, dẫn đến mẫu giá trị $R'(i, j)$ như được thể hiện dưới đây,

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

trong đó $f(k, l)$ ký hiệu các hệ số bộ lọc được giải mã, $K(x, y)$ là chức năng xén và $c(k, l)$ ký hiệu các thông số xén được giải mã. Biến số k và l biến thiên giữa $-\frac{L}{2}$ và $\frac{L}{2}$ trong đó L ký hiệu chiều dài bộ lọc. Chức năng xén $K(x, y) = \min(y, \max(-y, x))$ mà tương ứng với chức năng $\text{Clip3}(-y, y, x)$.

2.8.1.6 Quá trình lọc biên ảo để làm giảm đệm dòng

Trong VTM5, để làm giảm yêu cầu đệm dòng của ALF, phân loại khối được cải biến và việc lọc được sử dụng đối với các mẫu gần các biên CTU ngang. Nhằm mục đích này, biên ảo được xác định là dòng bằng cách dịch chuyển biên

CTU ngang với “N” mẫu như được thể hiện trên FIG. 10 với N bằng 4 đối với thành phần độ sáng và 2 đối với thành phần sắc độ.

Fig. 10 thể hiện ví dụ về phân loại khối được cải biến ở các biên ảo.

Phân loại khối được cải biến được áp dụng đối với thành phần độ sáng như được thể hiện trên FIG. 11 giá trị mức hoạt động A được định tỷ lệ một cách tương ứng bằng cách tính đến số lượng của các mẫu được giảm được sử dụng trong tính toán gradient 1D Laplace.

Để xử lý việc lọc, hoạt động đệm đối xứng ở các biên ảo được sử dụng cho cả hai thành phần Độ sáng và thành phần Sắc độ. Như được thể hiện trên FIG. 11, khi mẫu được lọc nằm bên dưới biên ảo, thì các mẫu lân cận mà nằm bên trên biên ảo được đệm. Trong khi đó, các mẫu tương ứng ở các phía khác cũng được đệm, một cách đối xứng.

Fig. 11 là hình vẽ thể hiện các ví dụ về việc lọc ALF cải biến đối với thành phần Độ sáng ở các biên ảo.

2.9 Dịch vị thích ứng mẫu (SAO)

Dịch vị thích ứng mẫu (Sample Adaptive Offset - SAO) được áp dụng cho tín hiệu được tái tạo sau khi bộ lọc giải khối bằng cách sử dụng các dịch vị được chỉ rõ đối với mỗi CTB bởi bộ tạo mã. Đầu tiên bộ tạo mã HM quyết định về việc liệu quá trình SAO có được áp dụng cho lát hiện tại hay không. Nếu SAO được áp dụng cho lát này, thì mỗi CTB được phân loại là một trong số năm kiểu SAO như được thể hiện trong Bảng 2-. Khái niệm về SAO là phân loại các điểm ảnh thành các hạng mục và làm giảm sự biến dạng bằng cách bổ sung độ dịch vào các điểm ảnh của mỗi hạng mục. Hoạt động SAO bao gồm Độ dịch rìa (Edge Offset - EO) mà sử dụng các thuộc tính rìa để phân loại điểm ảnh trong kiểu SAO 1-4 và Độ dịch băng (Band Offset - BO) mà sử dụng mật độ điểm ảnh để phân loại điểm ảnh trong kiểu SAO 5. Mỗi CTB được áp dụng có các thông số SAO bao gồm sao_merge_left_flag, sao_merge_up_flag, kiểu SAO và bốn độ dịch. Nếu sao_merge_left_flag bằng 1, CTB hiện tại sẽ sử dụng lại kiểu SAO và các dịch vị của CTB sang trái. Nếu sao_merge_up_flag bằng 1, CTB hiện tại sẽ sử dụng lại kiểu SAO và các dịch vị của CTB bên trên.

Bảng 2-6 – Đặc tả kiểu SAO

Kiểu SAO	Kiểu dịch vị thích ứng mẫu được sử dụng	Số lượng các hạng mục
0	Không	0
1	Độ dịch rìa mẫu hình 0-độ 1-D	4
2	Độ dịch rìa mẫu hình 90 độ 1-D	4
3	Độ dịch rìa mẫu hình 135 độ 1-D	4
4	Độ dịch rìa mẫu hình 45 độ 1-D	4
5	Độ dịch bằng	4

2.9.1 Hoạt động của mỗi kiểu SAO

Độ dịch rìa sử dụng bốn mẫu hình 3-điểm ảnh 1-D để phân loại điểm ảnh hiện tại p bằng cách xem xét thông tin hướng rìa, như được thể hiện trên FIG. 12. Từ trái sang phải là: 0 độ, 90 độ, 135 độ và 45 độ.

Fig. 12 là hình vẽ thể hiện các ví dụ về bốn mẫu hình 3-điểm ảnh 1-D để phân loại điểm ảnh trong EO.

Mỗi CTB được phân loại vào một trong số năm hạng mục theo Bảng 2-7.

Bảng 2-7 – Quy tắc phân loại điểm ảnh đối với EO

Phân loại	Điều kiện	Ý nghĩa
0	Không có điều nào dưới đây	Phần lớn đơn điệu
1	$p < 2$ lân cận	Tối thiểu nội tại
2	$p < 1$ lân cận & $p = 1$ lân cận	Rìa
3	$p > 1$ lân cận & $p = 1$ lân cận	Rìa
4	$p > 2$ lân cận	Tối đa nội tại

Độ dịch bằng (BO) phân loại tất cả các điểm ảnh trong một vùng CTB thành 32 băng đồng đều bằng cách sử dụng năm bit giá trị cao nhất của giá trị điểm ảnh làm chỉ số băng. Nói cách khác, khoảng mật độ điểm ảnh được chia thành 32 đoạn bằng nhau từ không tới giá trị mật độ tối đa (ví dụ 255 đối với các điểm ảnh 8 bit). Bốn băng liên kế được nhóm lại với nhau và mỗi nhóm được chỉ báo bởi vị trí ngoài cùng bên trái của nó như được thể hiện trên FIG. 13. Bộ tạo mã tra cứu tất cả vị trí lấy được nhóm với mức giảm tối đa sự biến dạng bằng cách bù độ dịch của mỗi băng.

Fig. 13 thể hiện ví dụ về bốn bảng được nhóm lại với nhau và được biểu diễn bằng vị trí bắt đầu bảng của chúng.

2.10 Dự báo ngoài ảnh và trong ảnh kết hợp (CIIP)

Trong VTM5, khi CU được mã hóa trong chế độ hợp nhất, nếu CU bao gồm ít nhất 64 mẫu độ sáng (nghĩa là, chiều rộng CU nhân với chiều cao CU bằng hoặc lớn hơn 64), và nếu cả chiều rộng CU lẫn chiều cao CU đều nhỏ hơn 128 mẫu độ sáng, cờ bổ sung được báo tín hiệu để chỉ báo rằng chế độ dự báo ngoài ảnh/trong ảnh (CIIP) kết hợp có được áp dụng cho CU hiện tại hay không. Như được tên của nó thể hiện, dự báo CIIP kết hợp tín hiệu dự báo ngoài ảnh với tín hiệu dự báo trong ảnh. Tín hiệu dự báo ngoài ảnh trong chế độ CIIP P_{inter} được rút ra bằng cách sử dụng cùng quá trình dự báo ngoài ảnh mà được áp dụng cho chế độ hợp nhất bình thường; và tín hiệu dự báo trong ảnh P_{intra} được rút ra theo sau quá trình dự báo trong ảnh bình thường với chế độ phẳng. Sau đó, các tín hiệu dự báo trong ảnh và ngoài ảnh được kết hợp bằng cách sử dụng lấu trung bình có trọng số, trong đó giá trị trọng số được tính toán tùy theo các chế độ mã hóa của các khối lân cận trên cùng và bên trái (được thể hiện trên FIG. 14) như dưới đây:

- Nếu lân cận trên cùng là khả dụng và được mã hóa trong ảnh, thì khi đó thiết lập isIntraTop bằng 1, ngược lại thiết lập isIntraTop bằng 0;
- Nếu lân cận trái là khả dụng và được mã hóa trong ảnh, thì khi đó thiết lập isIntraLeft bằng 1, ngược lại thiết lập isIntraLeft bằng 0;
- Nếu (isIntraLeft + isIntraTop) bằng 2, thì wt được thiết lập bằng 3;
- Ngược lại, nếu (isIntraLeft + isIntraTop) bằng 1, thì wt được thiết lập bằng 2;
- Ngược lại, thiết lập wt bằng 1.

Dự báo CIIP được tạo thành như dưới đây:

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

Fig. 14 là hình vẽ thể hiện các ví dụ về các khối lân cận trên cùng và bên trái được sử dụng trong việc dẫn xuất trọng số CIIP

2.11 Ánh xạ độ sáng với chia tỷ lệ sắc độ (LMCS)

Trong VTM5, công cụ mã hóa được gọi là ánh xạ độ sáng với chia tỷ lệ sắc độ (Luma mapping with chroma scaling - LMCS) được bổ sung dưới dạng khối xử lý mới trước các bộ lọc vòng. LMCS có hai thành phần chính: 1) ánh xạ trong vòng của thành phần độ sáng dựa trên các mô hình tuyến tính từng khúc thích ứng; 2) đối với các thành phần sắc độ, việc chia tỷ lệ dư sắc độ phụ thuộc độ sáng được áp dụng. FIG. 15 thể hiện cấu trúc LMCS từ viễn cảnh bộ giải mã. Các khối chấm chấm trên FIG. 15 chỉ báo nơi mà việc xử lý được áp dụng trong miền được ánh xạ; và điều này bao gồm việc lượng tử hóa nghịch đảo, chuyển đổi nghịch đảo, dự báo trong ảnh độ sáng và bổ sung dự báo độ sáng cùng với phần dư độ sáng. Các khối không được tạo mẫu hình trên FIG. 15 chỉ báo nơi mà việc xử lý được áp dụng trong miền gốc (tức là, không được ánh xạ); và điều này bao gồm các bộ lọc vòng chẳng hạn giải khối, ALF, và SAO, dự báo bù chuyển động, dự báo trong ảnh sắc độ, bổ sung dự báo sắc độ cùng với phần dư sắc độ, và lưu trữ các ảnh được giải mã làm các ảnh tham chiếu. Các khối kẻ ô vuông trên FIG. 15 là các khối chức năng LMCS mới, bao gồm ánh xạ thuận và nghịch của tín hiệu độ sáng và quá trình chia tỷ lệ sắc độ phụ thuộc độ sáng. Giống như phần lớn các công cụ khác trong VVC, LMCS có thể được phép/không được phép ở bậc dãy bằng cách sử dụng cờ SPS.

Fig. 15 là hình vẽ thể hiện các ví dụ về cấu trúc ánh xạ độ sáng với chia tỷ lệ sắc độ.

2.12 Phân chia cây kép

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

Do cây kép trong ảnh trong ảnh cho phép áp dụng cách phân chia khác nhau trong cây mã hóa sắc độ so với cây mã hóa độ sáng, cây kép đưa vào đường liên hợp mã hóa dài hơn và khoảng giá trị QTBT MinQTSIZEC và MinBtSizeY và MinTTSIZEY trong cây sắc độ cho phép các khối sắc độ nhỏ chẳng hạn 2x2, 4x2, và 2x4. Nó tạo ra khó khăn trong thiết kế bộ giải mã thực tế. Hơn nữa, một vài chế độ dự báo chẳng hạn CCLM, phẳng và chế độ góc cần phải nhân. Nhằm

tránh các vấn đề nêu trên, các kích thước khối sắc độ nhỏ ($2 \times 2/2 \times 4/4 \times 2$) bị hạn chế trong cây kép dưới dạng sự hạn chế phân chia.

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

Kích thước sắc độ nhỏ không thân thiện với phương án thực hiện phần cứng. Trong các trường hợp cây kép, các khối sắc độ với các kích thước quá nhỏ là không được phép. Tuy nhiên, trong các trường hợp cây đơn, VVC dự thảo 5 vẫn cho phép các khối sắc độ 2×2 , 2×4 , 4×2 . Để hạn chế kích thước của khối sắc độ, trong cây mã hóa đơn, SCIPU được xác định trong JVET-O0050 là nút cây mã hóa có kích thước khối sắc độ lớn hơn hoặc bằng TH mẫu sắc độ và có ít nhất một khối độ sáng con nhỏ hơn 4TH mẫu độ sáng, trong đó TH được thiết lập bằng 16 trong trường hợp này. Được yêu cầu rằng trong mỗi SCIPU, tất cả CB là ngoài ảnh, hoặc tất cả CB là không ngoài ảnh, tức là hoặc trong ảnh hoặc IBC. Trong trường hợp SCIPU không ngoài ảnh, thì còn yêu cầu là sắc độ của SCIPU không ngoài ảnh không được tách tiếp và độ sáng của SCIPU được cho phép được tách tiếp. Theo cách này, kích thước CB trong ảnh sắc độ nhỏ nhất là 16 mẫu sắc độ, và các CB sắc độ 2×2 , 2×4 , và 4×2 bị loại bỏ. Ngoài ra, việc chia tỷ lệ sắc độ không được áp dụng trong trường hợp SCIPU không ngoài ảnh.

Hai ví dụ SCIPU được thể hiện trên FIG. 16. Trên FIG. 16(a), một CB sắc độ bằng 8×4 mẫu sắc độ và ba CB độ sáng (CB độ sáng 4×8 , 8×8 , 4×8) tạo thành một SCIPU do việc tách cây tam phân (TT) từ 8×4 mẫu sắc độ có thể dẫn đến các CB sắc độ nhỏ hơn 16 mẫu sắc độ. Trên FIG. 16(b), một CB sắc độ bằng 4×4 mẫu sắc độ (phía trái của 8×4 mẫu sắc độ) và ba CB độ sáng (các CB độ sáng 8×4 , 4×4 , 4×4) tạo thành một SCIPU, và một CB sắc độ khác bằng 4×4 mẫu (phía phải của 8×4 mẫu sắc độ) và hai CB độ sáng (các CB độ sáng 8×4 , 8×4) tạo thành một SCIPU do việc tách cây nhị phân (BT) từ 4×4 mẫu sắc độ có thể dẫn đến các CB sắc độ nhỏ hơn 16 mẫu sắc độ.

Fig. 16 thể hiện các ví dụ SCIPU.

Kiểu của SCIPU được suy ra là không ngoài ảnh nếu lát hiện tại là lát I hoặc SCIPU hiện tại có phân chia độ sáng 4×4 trong đó sau khi tách tiếp một lần (do không có 4×4 ngoài ảnh được cho phép trong VVC); ngược lại, kiểu của

SCIPU (ngoài ảnh hoặc không ngoài ảnh) được chỉ báo bởi một cờ được báo tín hiệu trước khi phân tách các CU trong SCIPU.

2.14 Các điều kiện ràng buộc khối sắc độ nhỏ trong VVC dự thảo 6

Trong VVC dự thảo 6 (JVET-O2001-vE.docx), các điều kiện ràng buộc đối với các khối sắc độ nhỏ được thực hiện như dưới đây (phần liên quan được đánh dấu in nghiêng đậm nét).

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	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(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)	
cbSubdiv + 1,	coding_tree(x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC,	
	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)		
cbSubdiv + 2,	coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
cbSubdiv + 1,	coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
cbSubdiv + 2,	coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)	
	} else { /* SPLIT TT HOR */	
	y1 = y0 + (cbHeight / 4)	
	y2 = y0 + (3 * cbHeight / 4)	
	qgOnY = qgOnY && (cbSubdiv + 2 <= cu qp delta subdiv)	
	qgOnC = qgOnC && (cbSubdiv + 2 <=	
cu chroma qp offset subdiv)		
cbSubdiv + 2,	coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)	
cbSubdiv + 1,	coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
cbSubdiv + 2,	coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnY, qgOnC,	
	cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)	
	}	
	} else {	
	x1 = x0 + (cbWidth / 2)	
	y1 = y0 + (cbHeight / 2)	
cbSubdiv + 2,	coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,	
	cqtDepth + 1, 0, 0, 0, treeType, modeType)	
	if(x1 < pic width in luma samples)	
cbSubdiv + 2,	coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,	
	cqtDepth + 1, 0, 0, 1, treeType, modeType)	

if(y1 < pic_height_in_luma_samples)	
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)	
}	

2.14.1.1 Cú pháp đơn vị mã hóa

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	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)
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)	
pred_mode_plt_flag	ae(v)

}	
...	

Biến số modeTypeCondition được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 0
 - slice_type == I và qtbtt_dual_tree_intra_flag bằng 1
 - modeTypeCurr không bằng MODE_TYPE_ALL
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
 - cbWidth * cbHeight bằng 128 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Ngược lại, modeTypeCondition được thiết lập bằng 0

Quá trình tách tứ phân được cho phép

Các đầu vào của quá trình này là:

- kích thước khối mã hóa cbSize trong các mẫu độ sáng,
- độ sâu cây đa kiểu mttDepth,
- biến số treeType chỉ rõ liệu cây đơn (SINGLE_TREE) hoặc cây kép có được sử dụng để chia các CTU và, khi cây kép được sử dụng, liệu các thành phần độ sáng (DUAL_TREE_LUMA) hoặc các thành phần sắc độ (DUAL_TREE_CHROMA) có đang được xử lý hay không,
- biến số modeType chỉ rõ liệu các chế độ mã hóa trong ảnh (MODE_INTRA), IBC (MODE_IBC), bảng màu (MODE_PLT), và các chế độ mã hóa ngoài ảnh có thể được sử dụng (MODE_TYPE_ALL),

hoặc liệu chỉ các chế độ mã hóa trong ảnh, IBC, và chế độ bảng màu có thể được sử dụng (MODE_TYPE_INTRA), hoặc liệu chỉ các chế độ mã hóa ngoài ảnh có thể được sử dụng (MODE_TYPE_INTER) đối với các đơn vị mã hóa bên trong nút cây mã hóa.

Đầu ra của quá trình này là biến số allowSplitQt.

Biến số allowSplitQt được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, thì allowSplitQt được thiết lập bằng FALSE:

- treeType bằng SINGLE_TREE hoặc DUAL_TREE_LUMA và cbSize là nhỏ hơn hoặc bằng MinQtSizeY
- treeType bằng DUAL_TREE_CHROMA và cbSize / SubWidthC là nhỏ hơn hoặc bằng MinQtSizeC
- mttDepth không bằng 0
- treeType bằng DUAL_TREE_CHROMA và (cbSize / SubWidthC) là nhỏ hơn hoặc bằng 4
- treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA

- Ngược lại, allowSplitQt được thiết lập bằng TRUE.

Quá trình tách nhị phân được cho phép

Các đầu vào của quá trình này là:

- chế độ tách nhị phân btSplit,
- chiều rộng khối mã hóa cbWidth trong các mẫu độ sáng,
- chiều cao khối mã hóa cbHeight trong các mẫu độ sáng,
- vị trí (x0, y0) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh,
- độ sâu cây đa kiểu mttDepth,
- độ sâu cây đa kiểu lớn nhất với độ dịch maxMttDepth,
- kích thước cây nhị phân lớn nhất maxBtSize,
- kích thước cây tứ phân nhỏ nhất minQtSize,
- chỉ số chia partIdx,

- biến số `treeType` chỉ rõ liệu cây đơn (`SINGLE_TREE`) hoặc cây kép được sử dụng để chia các CTU và, khi cây kép được sử dụng, liệu các thành phần độ sáng (`DUAL_TREE_LUMA`) hoặc các thành phần sắc độ (`DUAL_TREE_CHROMA`) có đang được xử lý,
- biến số `modeType` chỉ rõ liệu các chế độ mã hóa trong ảnh (`MODE_INTRA`), IBC (`MODE_IBC`), bảng màu (`MODE_PLT`), và các chế độ mã hóa ngoài ảnh có thể được sử dụng (`MODE_TYPE_ALL`) hay không, hoặc liệu chỉ các chế độ mã hóa trong ảnh, IBC, và bảng màu có thể được sử dụng (`MODE_TYPE_INTRA`), hoặc liệu chỉ các chế độ mã hóa ngoài ảnh có thể được sử dụng (`MODE_TYPE_INTER`) đối với các đơn vị mã hóa bên trong nút cây mã hóa.

Đầu ra của quá trình này là biến số `allowBtSplit`.

Bảng 6-2 – Đặc tả 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 số `parallelTtSplit` và `cbSize` được rút ra như được chỉ rõ trong Bảng 6-2.

Biến số `allowBtSplit` được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, `allowBtSplit` được thiết lập bằng `FALSE`:
 - `cbSize` là nhỏ hơn hoặc bằng `MinBtSizeY`
 - `cbWidth` lớn hơn `maxBtSize`
 - `cbHeight` lớn hơn `maxBtSize`
 - `mttDepth` lớn hơn hoặc bằng `maxMttDepth`
 - `treeType` bằng `DUAL_TREE_CHROMA` và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ là nhỏ hơn hoặc bằng 16
 - `treeType` bằng `DUAL_TREE_CHROMA` và `modeType` bằng `MODE_TYPE_INTRA`

- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - btSplit bằng SPLIT_BT_VER
 - $y_0 + cbHeight$ lớn hơn $pic_height_in_luma_samples$
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - btSplit bằng SPLIT_BT_VER
 - cbHeight lớn hơn MaxTbSizeY
 - $x_0 + cbWidth$ lớn hơn $pic_width_in_luma_samples$
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - btSplit bằng SPLIT_BT_HOR
 - cbWidth lớn hơn MaxTbSizeY
 - $y_0 + cbHeight$ lớn hơn $pic_height_in_luma_samples$
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - $x_0 + cbWidth$ lớn hơn $pic_width_in_luma_samples$
 - $y_0 + cbHeight$ lớn hơn $pic_height_in_luma_samples$
 - cbWidth lớn hơn minQtSize
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - btSplit bằng SPLIT_BT_HOR
 - $x_0 + cbWidth$ lớn hơn $pic_width_in_luma_samples$
 - $y_0 + cbHeight$ là nhỏ hơn hoặc bằng $pic_height_in_luma_samples$
 - Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE:
 - mttDepth lớn hơn 0
 - partIdx bằng 1
 - $MttSplitMode[x_0][y_0][mttDepth - 1]$ bằng parallelTtSplit
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE

- btSplit bằng SPLIT_BT_VER
- cbWidth là nhỏ hơn hoặc bằng MaxTbSizeY
- cbHeight lớn hơn MaxTbSizeY
- Ngược lại, nếu tất cả trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE
 - btSplit bằng SPLIT_BT_HOR
 - cbWidth lớn hơn MaxTbSizeY
 - cbHeight là nhỏ hơn hoặc bằng MaxTbSizeY
- Ngược lại, allowBtSplit được thiết lập bằng TRUE.

Quá trình tách tam phân được cho phép

Các đầu vào của quá trình này là:

- chế độ tách tam phân ttSplit,
- chiều rộng khối mã hóa cbWidth trong các mẫu độ sáng,
- chiều cao khối mã hóa cbHeight trong các mẫu độ sáng,
- vị trí (x0, y0) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh,
- độ sâu cây đa kiểu mttDepth
- độ sâu cây đa kiểu lớn nhất với độ dịch maxMttDepth,
- kích thước cây tam phân lớn nhất maxTtSize,
- biến số treeType chỉ rõ liệu cây đơn (SINGLE_TREE) hoặc cây kép được sử dụng để chia các CTU và, khi cây kép được sử dụng, liệu các thành phần độ sáng (DUAL_TREE_LUMA) hoặc các thành phần sắc độ (DUAL_TREE_CHROMA) có đang được xử lý hay không,
- biến số modeType chỉ rõ liệu các chế độ mã hóa trong ảnh (MODE_INTRA), IBC (MODE_IBC), bảng màu (MODE_PLT), và các chế độ mã hóa ngoài ảnh có thể được sử dụng (MODE_TYPE_ALL), hoặc liệu chỉ các chế độ mã hóa trong ảnh, IBC, và chế độ bảng màu có thể được sử dụng (MODE_TYPE_INTRA), hoặc liệu chỉ các chế độ mã hóa ngoài ảnh có thể được sử dụng (MODE_TYPE_INTER) đối với các đơn vị mã hóa bên trong nút cây mã hóa.

Đầu ra của quá trình này là biến số allowTtSplit.

ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize cbWidth	cbHeight

Biến số `allowTtSplit` được rút ra như dưới đây:

- cbSize là nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
- cbWidth lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- cbHeight lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- mttDepth lớn hơn hoặc bằng maxMttDepth
- $x0 + \text{cbWidth}$ lớn hơn $\text{pic_width_in_luma_samples}$
- $y0 + \text{cbHeight}$ lớn hơn $\text{pic_height_in_luma_samples}$
- treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ là nhỏ hơn hoặc bằng 32
- treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA

pred_mode_flag bằng 0 chỉ rõ rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo ngoài ảnh. pred_mode_flag bằng 1 chỉ rõ rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo trong ảnh.

- Nếu cbWidth bằng 4 và cbHeight bằng 4, pred_mode_flag được suy ra bằng 1.
- Ngược lại, nếu modeType bằng MODE_TYPE_INTRA, pred_mode_flag được suy ra bằng 1.
- Ngược lại, nếu modeType bằng MODE_TYPE_INTER, pred_mode_flag được suy ra bằng 0.
- Ngược lại, pred_mode_flag được suy ra bằng 1 khi giải mã lát I, và bằng 0 khi lần lượt giải mã lát P hoặc B.

Biến số $\text{CuPredMode}[\text{chType}][x][y]$ được rút ra như dưới đây đối với $x = x0..x0 + \text{cbWidth} - 1$ và $y = y0..y0 + \text{cbHeight} - 1$:

- Nếu pred_mode_flag bằng 0, $\text{CuPredMode}[\text{chType}][x][y]$ được thiết lập bằng MODE_INTER .
- Ngược lại (pred_mode_flag bằng 1), $\text{CuPredMode}[\text{chType}][x][y]$ được thiết lập bằng MODE_INTRA .

$\text{pred_mode_ibc_flag}$ bằng 1 chỉ rõ rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo IBC. $\text{pred_mode_ibc_flag}$ bằng 0 chỉ rõ rằng đơn vị mã hóa hiện tại không được mã hóa trong chế độ dự báo IBC.

Khi $\text{pred_mode_ibc_flag}$ không có mặt, nó được suy ra như dưới đây:

- Nếu $\text{cu_skip_flag}[x0][y0]$ bằng 1, và cbWidth bằng 4, và cbHeight bằng 4, $\text{pred_mode_ibc_flag}$ được suy ra bằng 1.
- Ngược lại, nếu cả hai cbWidth và cbHeight bằng 128, $\text{pred_mode_ibc_flag}$ được suy ra bằng 0.
- Ngược lại, nếu modeType bằng MODE_TYPE_INTER , $\text{pred_mode_ibc_flag}$ được suy ra bằng 0.
- Ngược lại, nếu treeType bằng DUAL_TREE_CHROMA , $\text{pred_mode_ibc_flag}$ được suy ra bằng 0.
- Ngược lại, $\text{pred_mode_ibc_flag}$ được suy ra bằng giá trị của $\text{sps_ibc_enabled_flag}$ khi giải mã lát I, và 0 khi giải mã lần lượt lát P hoặc B.

Khi $\text{pred_mode_ibc_flag}$ bằng 1, biến số $\text{CuPredMode}[\text{chType}][x][y]$ được thiết lập bằng MODE_IBC đối với $x = x0..x0 + \text{cbWidth} - 1$ và $y = y0..y0 + \text{cbHeight} - 1$.

3. Các vấn đề

1. Hiện tại IBC được coi là MODE_TYPE_INTRA và do đó khối sắc độ nhỏ không được cho phép, điều này dẫn đến tổn thất hiệu quả mã hóa không cần thiết.
2. Hiện tại bảng màu được coi là MODE_TYPE_INTRA và do đó khối sắc độ nhỏ không được cho phép, điều này dẫn đến tổn thất hiệu quả mã hóa không cần thiết.

3. Hiện tại các điều kiện ràng buộc khối sắc độ nhỏ không xem xét định dạng lấy mẫu con màu.
4. Hiện tại sự chia và các điều kiện ràng buộc chế độ dự báo đối với các khối nhỏ giống nhau được áp dụng cho tất cả các định dạng sắc độ. Tuy nhiên, có thể mong muốn thiết kế các cơ chế điều kiện ràng buộc khác nhau đối với các khối nhỏ trong các định dạng sắc độ 4:2:0 và 4:2:2.
5. Hiện tại việc báo tín hiệu cờ chế độ bảng màu phụ thuộc vào modeType, điều này là không mong muốn do bảng màu không thể áp dụng các điều kiện ràng buộc khối nhỏ.
6. Hiện tại cờ chế độ IBC được suy ra bằng 0 đối với lát P/B với cu_skip_flag bằng 1 nhưng MODE_TYPE bằng MODE_TYPE_INTRA, điều này không hợp lệ trong phân tách cú pháp.
7. Hiện tại, chế độ IBC độ sáng không 4x4 không được cho phép đối với các khối độ sáng SCIPU, điều này có thể là không mong muốn và có thể gây ra tổn thất hiệu quả mã hóa.
8. Khối sắc độ 2xH vẫn được cho phép, điều này không thân thiện với việc thực hiện phần cứng.
9. CIIP được coi là thuộc MODE_INTER trong khi nó sử dụng dự báo trong ảnh, điều này phá vỡ các điều kiện ràng buộc trong một số trường hợp.
10. Khi SCIPU được áp dụng, delta QP đối với sắc độ có thể được báo tín hiệu tùy theo việc tách độ sáng. Ví dụ, khi các kích thước khối hiện tại là 16x8 trong các mẫu độ sáng và được tách bằng TT dọc, cây kép nội tại có thể được áp dụng. Được chỉ rõ rằng

$$qgOnC = qgOnC \&\& (cbSubdiv + 2 \leq cu_chroma_qp_offset_subdiv)$$
 Cho nên qgOnC được thiết lập bằng 0 nếu $cbSubdiv + 2 \leq cu_chroma_qp_offset_subdiv$. Cách thiết lập có điều kiện này giả định rằng thành phần sắc độ cũng được tách bởi TT. Với cây kép nội tại, thành phần sắc độ có thể không được tách, do đó cbSubdiv có thể lớn hơn cu_chroma_qp_offset_subdiv. IsCuChromaQpOffsetCoded phải được thiết lập bằng 0 để cho phép báo tín hiệu delta QP đối với sắc độ. Tuy

nhiên, IsCuChromaQpOffsetCoded không được thiết lập bằng 0 do qgOnC được thiết lập bằng 0.

11. Kích thước chuyển đổi tối đa để mã hóa không tồn thất có thể được thiết lập khác với kích thước chuyển đổi tối đa để mã hóa tồn thất.

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

Phần liệt kê dưới đây phải được xem như các ví dụ. Các kỹ thuật này không được diễn giải theo cách hẹp. Hơn nữa, các kỹ thuật này có thể được kết hợp theo cách bất kỳ.

Trong phần mô tả này, “nút cây mã hóa $M \times N$ ” chỉ báo khối $M \times N$, với M là chiều rộng khối và N là chiều cao khối trong các mẫu độ sáng, mà có thể còn được phân chia, chẳng hạn bằng QT/BT/TT. Ví dụ, khối có thể là nút QT, hoặc nút BT, hoặc nút TT. Nút cây mã hóa có thể là đơn vị mã hóa (ví dụ, với ba thành phần màu đối với cây đơn, với hai thành phần màu sắc độ đối với mã hóa sắc độ cây kép, và chỉ thành phần màu độ sáng đối với mã hóa độ sáng cây kép), hoặc khối mã hóa độ sáng, hoặc khối mã hóa sắc độ. “đơn vị nút cây mã hóa nhỏ” có thể chỉ báo nút cây mã hóa với kích thước khối $M \times N$ bằng 32/64/128 trong các mẫu độ sáng.

Nếu không được nêu cụ thể, chiều rộng W và chiều cao H đối khối mã hóa được đo bằng các mẫu độ sáng. Ví dụ, khối mã hóa $M \times N$ nghĩa là khối độ sáng $M \times N$, và/hoặc hai khối sắc độ $(M/\text{SubWidthC}) \times (N/\text{SubHeightC})$, trong đó SubWidthC và SubHeightC được rút ra bằng định dạng sắc độ như dưới đây.

chroma_format_idc	separate_colour_plane_flag	Định dạng sắc độ	SubWidthC	SubHeightC
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 liệu có chia và/hoặc cách thức chia 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 điều kiện ràng buộc đối với các kích thước của các khối sắc độ có thể tiếp theo các điều kiện ràng buộc như vậy đối với các khối độ sáng.

- b. Trong một ví dụ, đối với định dạng màu 4:2:2, các điều kiện ràng buộc đối với các kích thước của các khối sắc độ có thể tiếp theo các điều kiện ràng buộc như vậy 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 điều kiện ràng buộc về việc chia khối nhỏ và/hoặc các chế độ dự báo không thể được áp dụng.
- d. Trong một ví dụ, các điều kiện ràng buộc về việc chia khối nhỏ và/hoặc các 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 $M \times N$ (chẳng hạn 8×8) với tách BT ngang, trong định dạng sắc độ 4:2:2, tách BT ngang có thể được cho phép đối với cả khối sắc độ lẫn khối độ sáng, trong khi trong định dạng sắc độ 4:2:0, tách BT ngang có thể được cho phép đối với khối độ sáng nhưng không được phép đối với khối sắc độ.
 - ii. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (chẳng hạn 16×4) với tách BT dọc, trong định dạng sắc độ 4:2:2, tách BT dọc có thể được cho phép đối với cả khối sắc độ lẫn khối độ sáng, trong khi trong định dạng sắc độ 4:2:0, tách BT dọc có thể được cho phép đối với khối độ sáng nhưng không được phép đối với khối sắc độ.
 - iii. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (chẳng hạn 8×16) với tách TT ngang, trong định dạng sắc độ 4:2:2, tách TT ngang có thể được cho phép đối với cả khối sắc độ lẫn khối độ sáng, trong khi trong định dạng sắc độ 4:2:0, tách TT ngang có thể được cho phép đối với khối độ sáng nhưng không được phép đối với khối sắc độ.
 - iv. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (chẳng hạn 32×4) với tách TT dọc, trong định dạng sắc độ 4:2:2, tách TT dọc có thể được cho phép đối với cả khối sắc độ lẫn khối độ sáng, trong khi trong định dạng sắc độ 4:2:0, tách TT dọc có thể được cho phép đối với khối độ sáng nhưng không được phép đối với khối sắc độ.

- v. Trong một ví dụ, đối với các định dạng màu 4:0:0, và/hoặc 4:4:4, các điều kiện ràng buộc khối nhỏ có thể không được áp dụng.
- e. Trong một ví dụ, việc liệu có cho phép SCIPU hay không là phụ thuộc vào định dạng màu.
 - i. Trong một ví dụ, SCIPU được phép đối với các định dạng màu 4:2:0 và 4:2:2.
 - ii. Trong một ví dụ, SCIPU là không được phép đối với định dạng màu 4:0:0 và/hoặc 4:4:4.
 - 1. Trong một ví dụ, modeType có thể luôn luôn bằng 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 luôn bằng 0 đối với định dạng màu 4:0:0 và/hoặc 4:4:4.
- 2. Cách thức xác định các chế độ dự báo (và/hoặc modeType) đối với các khối (con) của 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 dưới đây là đúng, modeType của các khối (con) được phân chia bởi nút cây mã hóa này có thể bằng 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 bằng hoặc MODE_TYPE_INTRA hoặc MODE_TYPE_INTER.
 - i. nút cây mã hóa MxN (chẳng hạn 8x8) với tách BT ngang
 - ii. nút cây mã hóa MxN (chẳng hạn 16x4) với tách BT dọc
 - iii. nút cây mã hóa MxN (chẳng hạn 8x16) với tách TT ngang
 - iv. nút cây mã hóa MxN (chẳng hạn 32x4) với tách TT dọc
- 3. Được 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 đơn vị mã hóa bằng MODE_TYPE_NO_INTER, MODE_INTER có thể không được phép.
- 4. Được 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 đơn vị mã hóa bằng `MODE_TYPE_NO_INTRA`, `MODE_INTRA` có thể không được phép.
- 5. Cờ điều kiện ràng buộc chế độ có thể không bao giờ được báo tín hiệu trong các định dạng sắc độ 4:2:2 và/hoặc các định dạng sắc độ 4:0:0 và/hoặc các định dạng sắc độ 4:4:4.
 - a. Trong một ví dụ, khi cờ điều kiện ràng buộc chế độ không có mặt, nó có thể được suy ra bằng 1.
 - i. Theo cách khác, khi cờ điều kiện ràng buộc chế độ không có mặt, nó có thể được suy ra bằng 0.
- 6. Việc liệu có áp dụng và/hoặc cách thức 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 liệu định dạng màu là 4:2:0 hoặc 4:2:2.
 - a. Trong một ví dụ, trong 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 có thể được phép chỉ nếu M nhân với N (được ký hiệu bằng $M \times N$) bằng 64 hoặc 32.
 - b. Trong một ví dụ, nút cây mã hóa với $M \times N = 128$ có thể không bao giờ được xử lý như khối SCIPU trong định dạng màu 4:2:2.
 - c. Trong một ví dụ, nút cây mã hóa với tách BT và $M \times N = 64$ có thể không bao giờ được xử lý như khối SCIPU trong định dạng màu 4:2:2.
 - d. Trong một ví dụ, nút cây mã hóa với `split_qt_flag` bằng 1 và $M \times N = 64$, có thể là khối SCIPU trong định dạng màu 4:2:2.
 - e. Trong một ví dụ, nút cây mã hóa với tách TT và $M \times N = 64$, có thể được xử lý như khối SCIPU trong định dạng màu 4:2:2.
 - f. Trong một ví dụ, nút cây mã hóa với tách BT và $M \times N = 32$, có thể được xử lý như khối SCIPU trong định dạng màu 4:2:2.
 - g. Trong phần mô tả trên đây, đối với khối SCIPU trong định dạng màu 4:2:2, `modeTypeCondition` có thể luôn luôn bằng 1.
 - h. Trong phần mô tả trên đây, đối với khối SCIPU trong định dạng màu 4:2:2, chỉ `MODE_TYPE_INTRA` là có thể được cho phép đối với cả

khối hiện tại trong nút mẹ lẫn tất cả các khối con bên dưới các nút lá con.

7. Trong định dạng màu 4:2:2, modeTypeCondition của khối SCIPU có thể luôn luôn bằng 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 trong định dạng màu 4:2:2, modeTypeCondition có thể không bao giờ bằng 2.
8. Trong định dạng màu 4:2:2, modeType của khối SCIPU có thể luôn luôn bằng MODE_TYPE_INTRA.
 - a. Trong một ví dụ, modeType có thể bằng MODE_TYPE_ALL hoặc MODE_TYPE_INTRA trong định dạng màu 4:2:2.
 - b. Trong một ví dụ, đối với các khối SCIPU trong định dạng màu 4:2:2, MODE_TYPE_INTER có thể không được phép.
9. Việc liệu chia khối có được cho phép hay không có thể phụ thuộc vào modeType, và/hoặc kích thước khối.
 - a. Trong một ví dụ, việc liệu tách BT và/hoặc tách TT có được cho phép đối với khối hay không có thể phụ thuộc vào modeType.
 - i. Trong một ví dụ, nếu modeType bằng MODE_TYPE_INTER, thì tách BT có thể không được phép đối với khối mã hóa hiện tại (ví dụ, allowBtSplit được thiết lập bằng sai).
 - ii. Trong một ví dụ, nếu modeType bằng MODE_TYPE_INTER, thì tách TT có thể không được phép đối với khối mã hóa hiện tại (ví dụ, allowTtSplit được thiết lập bằng sai).
 - b. Trong một ví dụ, việc liệu tách BT và/hoặc tách TT có được cho 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 $M \times N$ là nhỏ hơn hoặc bằng 32 và modeType bằng MODE_TYPE_INTER, tách BT có thể không được phép (ví dụ, allowBtSplit được thiết lập bằng sai).

- 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 $M \times N$ là nhỏ hơn hoặc bằng 64 và modeType bằng `MODE_TYPE_INTER`, tách TT có thể không được phép (ví dụ, `allowTtSplit` được thiết lập bằng sai).
10. Khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER`, việc tách cây mã hóa có thể bị hạn chế
- a. Trong một ví dụ, khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER`, tách BT có thể không được phép.
 - b. Trong một ví dụ, khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER`, tách TT có thể không được phép.
 - c. Trong một ví dụ, khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER`, tách QT có thể không được phép.
 - d. Trong một ví dụ, khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER` và kích thước khối độ sáng là nhỏ hơn hoặc bằng 32, tách BT có thể không được phép.
 - e. Trong một ví dụ, khi modeTypeCurr của cây mã hóa bằng `MODE_TYPE_INTER` và kích thước khối độ sáng là nhỏ hơn hoặc bằng 64, tách TT có thể không được phép.
11. Đơn vị mã hóa với treeType là `DUAL_TREE_LUMA` có thể được mã hóa trong chế độ ngoài ảnh.
- a. Trong một ví dụ, đơn vị mã hóa được mã hóa trong chế độ mã hóa ngoài ảnh, tức là `MODE_INTER` có thể chỉ bao gồm thành phần độ sáng thậm chí đối với các định dạng màu với nhiều thành phần màu.
 - b. Trong một ví dụ, `pred_mode_flag` có thể cần phải được phân tách đối với khối `DUAL_TREE_LUMA`.
 - c. Trong một ví dụ, đối với khối `DUAL_TREE_LUMA` được mã hóa trong chế độ ngoài ảnh, các điều kiện ràng buộc giống như của chế độ ngoài ảnh đối với `SINGLE_TREE` cũng có thể được áp dụng.
 - i. Trong một ví dụ, khối ngoài ảnh `DUAL_TREE_LUMA` 4x4 có thể không được phép.

12. Các khối trong ảnh sắc độ (và/hoặc IBC) với chiều rộng khối bằng M (chẳng hạn $M=2$) mẫu sắc độ có thể không được cho phép.
- a. Trong một ví dụ, các khối trong ảnh sắc độ $2 \times N$ (chẳng hạn $N \leq 64$) có thể không được cho phép trong cây kép.
 - i. Trong một ví dụ, khi `treeType` bằng `DUAL_TREE_CHROMA` và chiều rộng khối bằng 4 mẫu sắc độ, tách BT dọc có thể không được phép.
 - ii. Trong một ví dụ, khi `treeType` bằng `DUAL_TREE_CHROMA` và chiều rộng khối bằng 8 mẫu sắc độ, tách TT dọc có thể không được phép.
 - b. Trong một ví dụ, các khối trong ảnh sắc độ (và/hoặc IBC) $2 \times N$ (chẳng hạn $N \leq 64$) có thể không được cho phép trong cây đơn.
 - i. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (chẳng hạn $M=8$ và $N \leq 64$) với tách BT dọc, một trong số quá trình dưới đây có thể được áp dụng.
 1. Tách BT dọc có thể không được phép đối với khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$ nhưng được cho phép đối với khối độ sáng $8 \times N$.
 2. Khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$ có thể không được tách BT dọc, và có thể được mã hóa bởi `MODE_INTRA`, hoặc `MODE_IBC`.
 3. Tách BT dọc có thể được cho phép đối với cả khối độ sáng $8 \times N$ lẫn khối sắc độ $4 \times N$ hoặc $4 \times (N/2)$, nhưng cả các khối độ sáng lẫn các khối sắc độ không được mã hóa bằng `MODE_INTRA` (ví dụ, có thể được mã hóa bằng `MODE_INTER`, hoặc `MODE_IBC`).
 - ii. Trong một ví dụ, đối với nút cây mã hóa $M \times N$ (chẳng hạn $M=16$ và $N \leq 64$) với tách TT dọc, một trong số quá trình dưới đây có thể được áp dụng.
 1. Tách TT dọc có thể không được phép đối với khối sắc độ $8 \times N$ hoặc $8 \times (N/2)$ nhưng được cho phép đối với khối độ sáng $16 \times N$.
 2. Khối sắc độ $8 \times N$ hoặc $8 \times (N/2)$ có thể không được tách TT dọc và được mã hóa bởi `MODE_INTRA`, hoặc `MODE_IBC`.

3. Tách TT dọc có thể được cho phép đối với cả khối độ sáng 16xN lần khối sắc độ 8xN hoặc 8x(N/2), nhưng cả các khối độ sáng lẫn các khối sắc độ có thể không được mã hóa bởi `MODE_INTRA` (ví dụ, có thể được mã hóa bởi `MODE_INTER`, hoặc `MODE_IBC`).
13. Chế độ IBC có thể được cho phép đối với các khối độ sáng và/hoặc các khối sắc độ không phụ thuộc vào liệu có phải là kích thước khối nhỏ hay không.
 - a. Trong một ví dụ, chế độ IBC có thể được cho phép đối với các khối độ sáng bao gồm các khối độ sáng 8x4/ 8x8/16x4 và 4xN (chẳng hạn $N \leq 64$), thậm chí nếu `modeType` bằng `MODE_TYPE_INTRA`.
 - b. Trong một ví dụ, chế độ IBC có thể được cho phép đối với các khối sắc độ, thậm chí nếu `modeType` bằng `MODE_TYPE_INTRA`.
14. Việc báo tín hiệu của cờ chế độ dự báo IBC có thể phụ thuộc vào kiểu chế độ dự báo (ví dụ, `MODE_TYPE_INTRA`).
 - a. Trong một ví dụ, cờ chế độ dự báo IBC đối với khối không SKIP (ví dụ khối mã hóa mà không được mã hóa bởi chế độ bỏ qua) có thể được báo tín hiệu một cách rõ ràng trong dòng bit khi `treeType` không bằng `DUAL_TREE_CHROMA` và `modeType` bằng `MODE_TYPE_INTRA`.
15. Cờ chế độ dự báo IBC có thể được suy ra tùy theo cờ CU SKIP và kiểu chế độ (ví dụ, `modeType`).
 - a. Trong một ví dụ, nếu khối hiện tại được mã hóa với chế độ bỏ qua (chẳng hạn `cu_skip_flag` bằng 1), và `modeType` bằng `MODE_TYPE_INTRA`, cờ chế độ dự báo IBC (chẳng hạn `pred_mode_ibc_flag`) có thể được suy ra bằng 1.
16. Việc báo tín hiệu rõ ràng cờ chế độ bảng màu có thể không phụ thuộc vào `modeType`.
 - a. Trong một ví dụ, việc báo tín hiệu cờ chế độ bảng màu (chẳng hạn `pred_mode_plt_flag`) có thể phụ thuộc vào kiểu lát, kích thước khối, chế độ dự báo, v.v., nhưng `modeType` là gì không thành vấn đề.

- b. Trong một ví dụ, cờ chế độ bảng màu (chẳng hạn `pred_mode_plt_flag`) được suy ra bằng 0 khi `modeType` bằng `MODE_TYPE_INTER` hoặc `MODE_TYPE_INTRA`.
17. Chế độ IBC có thể được cho phép sử dụng khi `modeType` bằng `MODE_TYPE_INTER`
- a. Trong một ví dụ, IBC sắc độ có thể không được phép khi `modeType` bằng `MODE_TYPE_INTRA`.
 - b. Trong một ví dụ, chế độ IBC có thể được cho phép sử dụng khi `modeType` bằng `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER`.
 - c. Trong một ví dụ, chế độ IBC có thể được cho phép sử dụng không phụ thuộc vào việc `modeType` là gì.
 - d. Trong một ví dụ, bên trong một SCIPU, cả hai chế độ IBC và chế độ ngoài ảnh có thể được cho phép.
 - e. Trong một ví dụ, kích thước của khối sắc độ IBC có thể luôn luôn tương ứng với kích thước của khối độ sáng tương ứng.
 - f. Trong một ví dụ, khi `modeType` bằng `MODE_TYPE_INTER` và kích thước đơn vị mã hóa là 4x4 theo độ sáng, việc báo tín hiệu của `pred_mode_ibc_flag` có thể được bỏ qua và `pred_mode_ibc_flag` có thể được suy ra bằng 1.
18. Chế độ bảng màu có thể được cho phép sử dụng khi `modeType` là `MODE_TYPE_INTER`
- a. Trong một ví dụ, sắc độ bảng màu có thể không được phép khi `modeType` là `MODE_TYPE_INTRA`.
 - b. Trong một ví dụ, chế độ IBC có thể được cho phép sử dụng khi `modeType` bằng `MODE_TYPE_INTRA` hoặc `MODE_TYPE_INTER`.
 - c. Trong một ví dụ, chế độ IBC có thể được cho phép sử dụng không phụ thuộc vào việc `modeType` là gì.

- d. Trong một ví dụ, chế độ bảng màu có thể được cho phép sử dụng khi modeType bằng MODE_TYPE_INTRA hoặc MODE_TYPE_INTER.
 - e. Trong một ví dụ, chế độ bảng màu có thể được cho phép sử dụng không phụ thuộc vào việc modeType là gì.
 - f. Trong một ví dụ, bên trong một SCIPU, cả hai chế độ bảng màu và chế độ ngoài ảnh đều có thể được cho phép.
 - g. Trong một ví dụ, bên trong một SCIPU, tất cả chế độ bảng màu, chế độ IBC và chế độ ngoài ảnh có thể được cho phép.
 - h. Trong một ví dụ, kích thước của khối sắc độ bảng màu có thể luôn luôn tương ứng với kích thước của khối độ sáng tương ứng.
 - i. Trong một ví dụ, khi modeType bằng MODE_TYPE_INTER và kích thước đơn vị mã hóa là 4x4 theo độ sáng, việc báo tín hiệu của pred_mode_plt_flag có thể được bỏ qua và pred_mode_plt_flag có thể được suy ra bằng 1.
 - j. Trong một ví dụ, khi modeType bằng MODE_TYPE_INTER và kích thước đơn vị mã hóa là 4x4 theo độ sáng, một thông điệp có thể được gửi để chỉ báo việc chế độ dự báo hiện tại là chế độ IBC hoặc chế độ bảng màu.
 - k. Trong một ví dụ, việc liệu có cho phép/không cho phép chế độ bảng màu có thể phụ thuộc vào các kiểu lát và modeType.
 - i. Trong một ví dụ, đối với các lát I với MODE_TYPE_INTRA, chế độ bảng màu có thể được phép.
 - ii. Trong một ví dụ, đối với các lát P/B với MODE_TYPE_INTER, chế độ bảng màu có thể được phép.
19. Khi chế độ bảng màu được phép, cây kép nội tại có thể không được phép.
- a. Trong một ví dụ, khi chế độ bảng màu được phép, modeTypeCondition có thể luôn luôn được thiết lập bằng 0.
20. Đối với các khối sắc độ nhỏ với chiều rộng bằng M (ví dụ, M = 2) hoặc chiều cao bằng N (ví dụ, N=2), các chế độ dự báo trong ảnh được cho

phép có thể bị hạn chế là khác với các chế độ được cho phép đối với các khối sắc độ lớn.

- a. Trong một ví dụ, chỉ tập con của chế độ dự báo trong ảnh trong số các chế độ dự báo trong ảnh sắc độ khả dụng 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 với chiều rộng bằng M (ví dụ, $M = 2$) hoặc chiều cao bằng N (ví dụ, $N = 2$), các kiểu chuyển đổi có thể bị hạn chế là khác với các kiểu được cho phép đối với các khối sắc độ lớn.

- a. Trong một ví dụ, chỉ bỏ qua chuyển đổi có thể được sử dụng.
- b. Trong một ví dụ, chỉ chuyển đổi một kích thước có thể được sử dụng.
- c. Trong một ví dụ, các công cụ mã hóa mà hỗ trợ nhiều kiểu chuyển đổi là không được phép.
 - i. Theo cách khác, việc báo tín hiệu của các công cụ mã hóa mà hỗ trợ nhiều kiểu chuyển đổi được bỏ qua.

22. CIIP có thể được coi là MODE_TYPE_INTRA.

- a. Trong một ví dụ, chế độ CIIP có thể được cho phép khi phân chia cây kép được sử dụng.
 - i. Trong một ví dụ, chế độ CIIP có thể được cho phép khi kiểu 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 ảnh đối với sắc độ trong CIIP có thể bị hạn chế là chế độ dự báo trong ảnh đơn giản.

1. Trong một ví dụ, INTRA_DC có thể được sử dụng để dự báo trong ảnh 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 ảnh 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 ảnh 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 ảnh đối với sắc độ trong CIIP có thể bị hạn chế ở chế độ dự báo trong ảnh đơn giản.
1. Trong một ví dụ, INTRA_DC có thể được sử dụng để dự báo trong ảnh sắc độ.
 2. Trong một ví dụ, chế độ INTRA_ANGULAR18 có thể được sử dụng để dự báo trong ảnh sắc độ.
 3. Trong một ví dụ, chế độ INTRA_ANGULAR50 có thể được sử dụng để dự báo trong ảnh sắc độ.
23. Đối với các tiểu mục trên đây, các biến số M và/hoặc N có thể được xác định trước hoặc được báo tín hiệu.
- a. Trong một ví dụ, M và/hoặc N có thể còn phụ thuộc vào các định dạng màu (ví dụ, 4:2:0, 4:2:2, 4:4:4).
24. modeType có thể được mở rộng để bao phủ nhiều kiểu hơn.
- a. Trong một ví dụ, modeType có thể là MODE_TYPE_IBC. Khi modeType bằng MODE_TYPE_IBC, chế độ dự báo được suy ra bằng IBC.
 - i. Trong một ví dụ, pred_mode_flag không được báo tín hiệu trong trường hợp này.
 - ii. Trong một ví dụ, pred_mode_ibc_flag không được báo tín hiệu trong trường hợp này.
 - iii. Trong một ví dụ, pred_mode_plt_flag không được báo tín hiệu trong trường hợp này.

- b. Trong một ví dụ, modeType có thể là MODE_TYPE_PALETTE. Khi modeType bằng MODE_TYPE_PALETTE, chế độ dự báo được suy ra bằng Chế độ bảng màu.
 - i. Trong một ví dụ, pred_mode_flag không được báo tín hiệu trong trường hợp này.
 - ii. Trong một ví dụ, pred_mode_ibc_flag không được báo tín hiệu trong trường hợp này.
 - iii. Trong một ví dụ, pred_mode_plt_flag không được báo tín hiệu trong trường hợp này.
 - c. Trong một ví dụ, mode_constraint_flag có thể được thay thế bằng chỉ số để chỉ báo kiểu chế độ nào trong số các modeType được cho phép là được sử dụng.
25. Trong một ví dụ, việc liệu tách QT có được phép đối với khối với các kích thước $W \times H$ hay không có thể phụ thuộc vào modeType được kết hợp với các kích thước.
- a. Ví dụ, nếu modeType bằng MODE_TYPE_INTER và W bằng 8 và H bằng 8, thì tách QT không được phép.
26. Trong một ví dụ, việc liệu tách TT dọc có được phép đối với khối với các kích thước $W \times H$ hay không có thể phụ thuộc vào modeType được kết hợp với các kích thước.
- a. Ví dụ, nếu modeType bằng MODE_TYPE_INTER và W bằng 16 và H bằng 4, thì tách TT dọc không được phép.
27. Trong một ví dụ, việc liệu tách TT ngang có được phép đối với khối với các kích thước $W \times H$ hay không có thể phụ thuộc vào modeType được kết hợp với các kích thước.
- a. Ví dụ, nếu modeType bằng MODE_TYPE_INTER và W bằng 4 và H bằng 16, thì tách TT ngang không được phép.
28. Trong một ví dụ, việc liệu tách BT dọc có được phép đối với khối với các kích thước $W \times H$ hay không có thể phụ thuộc vào modeType được kết hợp với các kích thước.

- a. Ví dụ, nếu modeType bằng MODE_TYPE_INTER và W bằng 8 và H bằng 4, thì tách BT dọc không được phép.
29. Trong một ví dụ, việc liệu tách TT ngang có được phép đối với khối với các kích thước $W \times H$ hay không có thể phụ thuộc vào modeType được kết hợp với các kích thước.
- a. Ví dụ, nếu modeType bằng MODE_TYPE_INTER và W bằng 4 và H bằng 8, thì tách BT ngang không được phép.
30. Trong một ví dụ, việc 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 các kích thước khối $W \times H$.
- a. Ví dụ, chế độ dự báo của CU sắc độ được suy ra bởi modeType; nhưng chế độ dự báo của CU độ sáng được báo tín hiệu thay vì được suy ra bởi modeType.
 - i. Ví dụ, chế độ dự báo của CU độ sáng được báo tín hiệu thay vì được suy ra bởi modeType nếu $W > 4$ hoặc $H > 4$.
31. Khi SCIPU được áp dụng, việc liệu có báo tín hiệu và/hoặc cách thức báo tín hiệu thông tin liên quan đến delta QP 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 liệu có báo tín hiệu và/hoặc cách thức để báo tín hiệu thông tin liên quan đến delta QP của thành phần thứ nhất có thể phụ thuộc vào cách tách của thành phần thứ nhất và được tách rời khỏi cách tách của thành phần thứ hai.
 - b. Trong một ví dụ, thành phần thứ nhất là độ sáng và thành phần thứ hai là sắc độ.
 - c. Trong một ví dụ, thành phần thứ nhất là sắc độ và thành phần thứ hai là độ sáng.
32. Biến số bất kỳ liên quan đến delta QP của thành phần thứ nhất không thể được cải biến trong quá trình giải mã hoặc phân tách của thành phần thứ hai khi cấu trúc mã hóa cây kép và/hoặc cây kép nội tại được áp dụng.
- a. Trong một ví dụ, cấu trúc mã hóa cây kép nội tại có thể được sử dụng theo SCIPU.

- b. Trong một ví dụ, thành phần thứ nhất là độ sáng và thành phần thứ hai là sắc độ.
 - i. Biến số này có thể là `IsCuQpDeltaCoded`.
 - c. Trong một ví dụ, thành phần thứ nhất là sắc độ và thành phần thứ hai là độ sáng.
 - i. Biến số này có thể là `IsCuChromaQpOffsetCoded`.
33. Khi SCIPU được áp dụng, thông tin liên quan đến delta QP của thành phần (chẳng hạn độ sáng hoặc sắc độ) có thể được báo tín hiệu nhiều nhất là một lần trong vùng cụ thể trong đó thành phần độ sáng và thành phần sắc độ phải dùng chung cùng kiểu chế độ (chẳng hạn `MODE_TYPE_INTER` hoặc `MODE_TYPE_INTRA`).
- a. Trong một ví dụ, vùng cụ thể được coi là nhóm lượng tử hóa.
34. Các khối trong ảnh sắc độ $M \times N$ với M là chiều rộng khối và N là chiều cao khối có thể không được cho phép.
- a. Trong một ví dụ, dự báo DC có thể được sử dụng để dự báo trong ảnh sắc độ $M \times N$, ví dụ, $M = 2$ mẫu sắc độ, $N = 8$ và/hoặc 16 và/hoặc 32 mẫu sắc độ.
 - i. Trong một ví dụ, các điểm ảnh lân cận được tái tạo không thể được sử dụng để dự báo DC.
 - ii. Trong một ví dụ, các điểm ảnh tham chiếu lân cận có thể được thiết lập bằng $1 \ll (\text{độ sâu bit} - 1)$ để dự báo DC.
 - iii. Trong một ví dụ, PDPC không thể được áp dụng cho dự báo DC.
 - iv. Trong một ví dụ, thông tin chế độ DC có thể không được mã hóa đối với khối trong ảnh sắc độ $M \times N$ mà được rút ra.
 - b. Trong một ví dụ, dự báo CCLM có thể được sử dụng để dự báo trong ảnh sắc độ $M \times N$, ví dụ, $M = 2$ mẫu sắc độ, $N = 8$ và/hoặc 16 và/hoặc 32 mẫu sắc độ.
 - i. Trong một ví dụ, các điểm ảnh lân cận được tái tạo không thể được sử dụng để dự báo CCLM.
 - ii. Trong một ví dụ, đối với dự báo CCLM $\text{predC} = a \cdot \text{recL} + b$ mà giá trị dự báo sắc độ predC được rút ra từ giá trị được tái tạo độ

sáng recL, các thông số a và b có thể được khởi tạo tới các giá trị cố định mặc định (chẳng hạn $a=0$, $b=512$) ở khởi đầu của đơn vị video chẳng hạn SPS/VPS/PPS/Ảnh/ảnh con/Lát/Ô ảnh/Ô gạch/hàng CTU/CTU/VPDU/CU/ PU/TU.

1. Trong một ví dụ, bảng vào trước ra trước có thể được duy trì trong khi đang thực hiện để cập nhật các thông số CCLM a và b.
- iii. Trong một ví dụ, PDPC không thể được áp dụng cho dự báo CCLM.
- iv. Trong một ví dụ, thông tin chế độ CCLM có thể không được mã hóa đối với khối trong ảnh sắc độ $M \times N$ mà được rút ra.
- v. Theo cách khác, các phương pháp được đề xuất có thể được áp dụng cho các dạng khác của các khối sắc độ được mã hóa CCLM với chiều rộng không bằng M hoặc chiều cao không bằng N.
 1. Trong một ví dụ, các thông số CCLM của các khối được mã hóa trước đó có thể được sử dụng đối với khối hiện tại thay vì được rút ra dựa trên thông tin tái tạo của các khối lân cận của khối hiện tại.
 2. Trong một ví dụ, các thông số CCLM có thể được lưu trữ trong bảng và được cập nhật trong khi thực hiện, chẳng hạn bằng cách sử dụng FIFO.
- c. Trong một ví dụ, các khối trong ảnh sắc độ $M \times N$ có thể không được cho phép trong cây kép, ví dụ, $M = 2$ mẫu sắc độ, $N = 8$ và/hoặc 16 và/hoặc 32 mẫu sắc độ.
 - i. Trong một ví dụ, khi treeType bằng DUAL_TREE_CHROMA, thì tách X có thể không được phép đối với nút cây mã hóa với kích thước bằng $(M \times a) \times (N \times b)$, trong đó $M \times a$ là chiều rộng của nút cây mã hóa trong các mẫu sắc độ, và $N \times b$ là chiều cao của nút cây mã hóa trong các mẫu sắc độ.
 1. Trong một ví dụ, X là tách BT dọc, $a = 2$, $b = 1$
 2. Trong một ví dụ, X là tách BT ngang, $a = 1$, $b = 2$
 3. Trong một ví dụ, X là tách TT dọc, $a = 4$, $b = 1$

4. Trong một ví dụ, X là tách TT ngang, $a = 1$, $b = 4$
- d. Trong một ví dụ, các khối trong ảnh sắc độ $M \times N$ có thể không được cho phép trong cây đơn, ví dụ, $M = 2$ mẫu sắc độ, $N = 8$ và/hoặc 16 và/hoặc 32 mẫu sắc độ.
 - i. Trong một ví dụ, trong cây đơn, tách X có thể không được phép đối với các thành phần sắc độ của nút cây mã hóa với kích thước bằng $(M \cdot a_{\text{subWidthC}}) \times (N \cdot b_{\text{subHeightC}})$, trong đó $M \cdot a$ là chiều rộng của nút cây mã hóa trong các mẫu độ sáng, $N \cdot b$ là chiều cao của nút cây mã hóa trong các mẫu độ sáng, và subWidthC và subHeightC là tỷ số lấy mẫu con sắc độ theo các kích thước chiều rộng và chiều cao, ví dụ, subWidthC và subHeightC bằng 2 đối với định dạng sắc độ 4:2:0.
 1. Trong một ví dụ, X = tách BT dọc, $a = 2$, $b = 1$
 2. Trong một ví dụ, X = tách TT dọc, $a = 4$, $b = 1$
 3. Trong một ví dụ, X = tách BT ngang, $a = 1$, $b = 2$
 4. Trong một ví dụ, X = tách TT ngang, $a = 1$, $b = 2$
 5. Trong một ví dụ, X = tách TT ngang, $a = 1$, $b = 4$
 6. Trong một ví dụ, khối sắc độ không tách có thể có kích thước bằng $(M \cdot a) \times (N \cdot d)$.
 7. Trong một ví dụ, khối sắc độ không tách có thể được xác định là `MODE_TYPE_INTRA` và chỉ dự báo trong ảnh (và/hoặc các chế độ IBC và/hoặc các chế độ PLT) có thể được cho phép đối với khối sắc độ này.
 8. Trong một ví dụ, tách X có thể được cho phép đối với các thành phần độ sáng của nút cây mã hóa này, ví dụ, nút cây mã hóa độ sáng có thể được tách tiếp thành nhiều nút con độ sáng, và mỗi nút con độ sáng có thể được xác định tới `MODE_TYPE_INTER` trong khi chỉ dự báo ngoài ảnh (và/hoặc chế độ IBC) có thể được cho phép.
 9. Theo cách khác, thành phần sắc độ có thể được tách tiếp cùng với thành phần độ sáng được sắp xếp, nhưng kiểu chế độ có thể bằng

MODE_TYPE_INTER và chỉ dự báo ngoài ảnh (và/hoặc chế độ IBC) có thể được cho phép.

- ii. Trong một ví dụ, các khối trong ảnh sắc độ $M \times N$ có thể không được cho phép trong CIIP.
 1. Trong một ví dụ, CIIP có thể không được cho phép đối với khối mã hóa với kích thước bằng $(M_{\text{subWidthC}}) \times (N_{\text{subHeightC}})$ trong các mẫu độ sáng.
 - a) Khi CIIP không được cho phép, phần tử cú pháp được liên kết chẳng hạn với CIIP có thể không được mã hóa trong dòng bit.
 2. Trong một ví dụ, quá trình trộn sắc độ trong CIIP có thể chỉ được điền bằng các dự báo từ phần ngoài ảnh.
 - a) Trong một ví dụ, chỉ dự báo trong ảnh độ sáng có thể được áp dụng cho khối được mã hóa CIIP, trong khi dự báo trong ảnh sắc độ không thể được áp dụng cho khối được mã hóa CIIP với kích thước bằng $(M_{\text{subWidthC}}) \times (N_{\text{subHeightC}})$ trong các mẫu độ sáng.
 3. Trong một ví dụ, dự báo DC có thể được sử dụng để dự báo trong ảnh sắc độ $M \times N$ đối với khối CIIP, như được mô tả trong tiểu mục 34(a).
 4. Trong một ví dụ, dự báo CCLM có thể được sử dụng để dự báo trong ảnh sắc độ $M \times N$ đối với khối CIIP, như được mô tả trong tiểu mục 34(b).

Mối quan hệ giữa kích thước losslessSizeY và maxCTB/maxTbSize/maxTS

35. Kích thước chuyển đổi tối đa có thể được báo tín hiệu và/hoặc được suy ra theo các cách khác nhau đối với mã hóa không tổn thất (và/hoặc mã hóa gần như không tổn thất) và mã hóa tổn thất.

- a. Trong một ví dụ, kích thước chuyển đổi tối đa đối với mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất có thể được gọi là theo cách khác chẳng hạn “kích thước phần dư tối đa” hoặc “kích thước biểu diễn

tối đa” do không có sự chuyển đổi nào liên quan. Trong phần mô tả này, nó có thể gọi là “maxLosslessTbSizeY”.

- b. Trong một ví dụ, mã hóa không tổn thất đối với khối có thể tham chiếu tới cu_transquant_bypass_flag cho khối đó bằng 1.
- c. Trong một ví dụ, mã hóa gần như không tổn thất đối với khối có thể tham chiếu tới mã hóa với QP nhỏ hơn ngưỡng cho trước.
 - i. Trong một ví dụ, ngưỡng có thể được chỉ báo trong tiêu đề DPS/VPS/SPS/PPS/Ô ảnh/Lát/Ô gạch.
- d. Trong một ví dụ, mã hóa gần như không tổn thất đối với khối có thể tham chiếu tới mã hóa với QP nhỏ hơn ngưỡng cho trước và transform_skip_flag bằng 1 cho khối đó.
 - i. Trong một ví dụ, ngưỡng có thể được chỉ báo trong tiêu đề DPS/VPS/SPS/PPS/Ô ảnh/Lát/Ô gạch.
- e. Kích thước chuyển đổi tối đa thứ nhất đối với mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất (ví dụ, maxLosslessTbSizeY) có thể được thiết lập theo điều kiện kích thước chuyển đổi tối đa thứ hai mà có thể được sử dụng để mã hóa tổn thất, và/hoặc kích thước đơn vị cây mã hóa tối đa, và/hoặc kích thước bỏ qua chuyển đổi tối đa.
- f. Kích thước chuyển đổi tối đa đối với mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất có thể được báo tín hiệu ở bậc đơn vị video chẳng hạn phần tử cú pháp (SE) ở SPS/VPS/PPS/Ảnh con/Lát/Ô ảnh/Ô gạch/CTU/VPDU.
 - i. Trong một ví dụ, SE có thể được báo tín hiệu tùy theo kích thước CTB tối đa (ví dụ, log2_ctu_size_minus5, CtbSizeY, và v.v.)
 1. Ví dụ, cò có thể được báo tín hiệu khi log2_ctu_size_minus5 lớn hơn N (chẳng hạn N=0).
 - ii. Trong một ví dụ, SE có thể được báo tín hiệu tùy theo kích thước chuyển đổi tối đa (ví dụ, sps_max_luma_transform_size_64_flag, MaxTbSize, và v.v.)

1. Ví dụ, SE có thể được báo tín hiệu khi `sps_max_luma_transform_size_64_flag` bằng M (chẳng hạn M=1).
- iii. Trong một ví dụ, SE có thể được báo tín hiệu tùy theo kích thước bỏ qua chuyển đổi tối đa (ví dụ, `log2_transform_skip_max_size_minus2`, `MaxTsSize`, và v.v.)
 1. Ví dụ, SE có thể được báo tín hiệu khi `log2_transform_skip_max_size_minus2` lớn hơn K (chẳng hạn K=3).
- iv. Trong một ví dụ, giá trị delta giữa `log(maxLosslessTbSizeY)` và X có thể được báo tín hiệu.
 1. Ví dụ, X có thể bằng `log2_ctu_size_minus5`
 2. Ví dụ, X có thể bằng `sps_max_luma_transform_size_64_flag`
 3. Ví dụ, X có thể bằng `log2_transform_skip_max_size_minus2`
 4. Ví dụ, X có thể bằng số nguyên cố định.
 5. Trong một ví dụ, delta – K có thể được báo tín hiệu, trong đó K là số nguyên chẳng hạn 1.
- v. Trong một ví dụ, giá trị delta giữa `maxLosslessTbSizeY` và X có thể được báo tín hiệu.
 1. Ví dụ, X có thể bằng `CtbSizeY`
 2. Ví dụ, X có thể bằng `MaxTbSize`
 3. Ví dụ, X có thể bằng `MaxTsSize`
 4. Ví dụ, X có thể bằng số nguyên cố định.
 5. Trong một ví dụ, delta – K có thể được báo tín hiệu, trong đó K là số nguyên chẳng hạn 1.
- vi. Trong một ví dụ, `maxLosslessTbSizeY` của đơn vị video (chẳng hạn dãy/ảnh/ảnh con/lát/ô ảnh/CTU/VPDU/CU/PU/TU) có thể được rút ra từ SE.
- vii. SE có thể là cờ.
- viii. Khi cờ không có mặt, thì có thể được suy ra bằng số nguyên dương N.

- g. Trong một ví dụ, nếu mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất được áp dụng cho đơn vị video (chẳng hạn `cu_transquant_bypass_flag` bằng 1), kích thước chuyển đổi tối đa đối với đơn vị video có thể được thiết lập bằng `maxLosslessTbSizeY`, khác với `maxTbSizeY`.
 - h. Theo cách khác, kích thước chuyển đổi tối đa đối với mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất đối với đơn vị video (chẳng hạn dãy/ảnh/ảnh con/lát/ô ảnh/CTU/VPDU/CU/PU/TU) có thể được rút ra thay vì được báo tín hiệu.
 - i. Ví dụ, nó có thể được thiết lập bằng giá trị số nguyên cố định M (chẳng hạn $M = 32$ trong các mẫu độ sáng).
 - ii. Ví dụ, nó có thể được thiết lập bằng kích thước chuyển đổi tối đa thứ hai mà có thể được sử dụng để mã hóa tổn thất
 - iii. Ví dụ, nó có thể được thiết lập bằng kích thước đơn vị cây mã hóa tối đa,
 - iv. Ví dụ, nó có thể được thiết lập bằng kích thước VPDU,
 - v. Ví dụ, nó có thể được thiết lập bằng kích thước khối bỏ qua chuyển đổi tối đa.
 - vi. Ví dụ, nó có thể được thiết lập bằng kích thước khối mã hóa phần dư tối đa.
36. Đối với khối phần dư $M \times N$ thì bằng cách mã hóa không tổn thất (và/hoặc mã hóa gần như không tổn thất) nó có thể được chia thành nhiều hơn một các khối con phần dư nhỏ hơn tùy theo `maxLosslessTbSizeY`.
- a. Sau khi chia, chiều rộng và chiều cao của tất cả các khối con đều không lớn hơn `maxLosslessTbSizeY`.
 - b. Trong một ví dụ, nó có thể được chia thành các khối con theo cách đệ quy cho tới khi chiều rộng và chiều cao của tất cả các khối con đều không lớn hơn `maxLosslessTbSizeY`.

5. Các phương án

Các phần mới được bổ sung được làm nổi bật bằng in đậm và nghiêng, và các phần được xóa khỏi dự thảo làm việc VVC được đánh dấu bằng dấu ngoặc kép (ví dụ, `[[a]]` ký hiệu việc xóa ký tự “a”). Các cải biến dựa trên dự thảo làm việc VVC cuối cùng (JVET-O2001-v11).

5.1 Phương án ví dụ #1

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

7.4.9.4 Các ngữ nghĩa cây mã hóa

Biến số `modeTypeCondition` được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, `modeTypeCondition` được thiết lập bằng 0
 - `slice_type == I` và `qtbtt_dual_tree_intra_flag` bằng 1
 - `modeTypeCurr` không bằng `MODE_TYPE_ALL`
 - `chroma_format_idc` bằng 0
 - `chroma_format_idc` bằng 3
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, `modeTypeCondition` được thiết lập bằng 1
 - `cbWidth * cbHeight` bằng 64 và `split_qt_flag` bằng 1
 - `cbWidth * cbHeight` bằng 64 và `MttSplitMode[x0][y0][mttDepth]` bằng `SPLIT_TT_HOR` hoặc `SPLIT_TT_VER`
 - `cbWidth * cbHeight` bằng 32 và `MttSplitMode[x0][y0][mttDepth]` bằng `SPLIT_BT_HOR` hoặc `SPLIT_BT_VER`
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, `modeTypeCondition` được thiết lập bằng `1 + (slice_type != I ? 1 : 0)`
 - `cbWidth * cbHeight` bằng 64 và `MttSplitMode[x0][y0][mttDepth]` bằng `SPLIT_BT_HOR` hoặc `SPLIT_BT_VER`
 - `cbWidth * cbHeight` bằng 128 và `MttSplitMode[x0][y0][mttDepth]` bằng `SPLIT_TT_HOR` hoặc `SPLIT_TT_VER`
- Ngược lại, `modeTypeCondition` được thiết lập bằng 0

5.2 Phương án ví dụ #2

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

7.3.8.5 Cú pháp đơn vị mã hóa

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
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)
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]))	
pred_mode_plt_flag	ae(v)

5.3 Phương án ví dụ #3

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

pred_mode_ibc_flag bằng 1 chỉ rõ rằng đơn vị mã hóa hiện tại được mã hóa trong chế độ dự báo IBC. pred_mode_ibc_flag bằng 0 chỉ rõ rằng đơn vị mã hóa hiện tại không được mã hóa trong chế độ dự báo IBC.

Khi pred_mode_ibc_flag không có mặt, nó được suy ra như dưới đây:

- If `cu_skip_flag[x0][y0]` bằng 1, và `cbWidth` bằng 4, và `cbHeight` bằng 4, `pred_mode_ibc_flag` được suy ra bằng 1.
- Ngược lại, nếu cả hai `cbWidth` và `cbHeight` đều bằng 128, `pred_mode_ibc_flag` được suy ra bằng 0.
- Ngược lại, nếu `cu_skip_flag[x0][y0]` bằng 1, và `modeType` bằng `MODE_TYPE_INTRA`, `pred_mode_ibc_flag` được suy ra bằng 1.
- Ngược lại, nếu `modeType` bằng `MODE_TYPE_INTER`, `pred_mode_ibc_flag` được suy ra bằng 0.
- Ngược lại, nếu `treeType` bằng `DUAL_TREE_CHROMA`, `pred_mode_ibc_flag` được suy ra bằng 0.
- Ngược lại, `pred_mode_ibc_flag` được suy ra lần lượt bằng giá trị của `sps_ibc_enabled_flag` khi giải mã lát I, và 0 khi giải mã lát P hoặc B.

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

5.4 Phương án ví dụ #4

Phương án dưới đây về việc báo tín hiệu của cờ chế độ dự báo IBC phụ thuộc vào `MODE_TYPE_INTRA`, và/hoặc chế độ IBC được cho phép đối với các khối độ sáng không phụ thuộc vào liệu có phải là kích thước khối nhỏ hay không.

7.3.8.5 Cú pháp đơn vị mã hóa

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	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>

<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 && treeType != DUAL_TREE_CHROMA) pred_mode_ibc_flag </pre>	ae(v)
---	-------

5.5 Phương án ví dụ #5

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

7.4.9.4 Các ngữ nghĩa cây mã hóa

Biến số modeTypeCondition được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 0
 - slice_type == I và qtbtt_dual_tree_intra_flag bằng 1
 - modeTypeCurr không bằng MODE_TYPE_ALL
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER và chroma_format_idc bằng 1
 - cbWidth * cbHeight bằng 128 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER và chroma_format_idc bằng 1
 - cbWidth bằng 8 và cbHeight bằng 8 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_VER và chroma_format_idc bằng 2

- cbWidth bằng 4 và cbHeight bằng 16 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR và chroma_format_idc bằng 2
- cbWidth bằng 16 và cbHeight bằng 8 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_VER và chroma_format_idc bằng 2
- cbWidth bằng 4 và cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR và chroma_format_idc bằng 2
- Ngược lại, modeTypeCondition được thiết lập bằng 0

5.6 Phương án ví dụ #6

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

7.4.9.4 Các ngữ nghĩa cây mã hóa

Biến số modeTypeCondition được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 0
 - slice_type == I và qtbt_dual_tree_intra_flag bằng 1
 - modeTypeCurr không bằng MODE_TYPE_ALL
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER

- $cbWidth * cbHeight$ bằng 128 và $MttSplitMode[x0][y0][mttDepth]$ bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
- $cbWidth$ bằng 8 và $MttSplitMode[x0][y0][mttDepth]$ bằng SPLIT_BT_VER
- $cbWidth$ bằng 16 và $MttSplitMode[x0][y0][mttDepth]$ bằng SPLIT_TT_VER
- Ngược lại, $modeTypeCondition$ được thiết lập bằng 0

5.7 Phương án ví dụ #7

Phương án dưới đây về việc không cho phép các khối trong ảnh sắc độ $2 \times N$ trong cây kép.

6.4.2 Quá trình tách nhị phân được cho phép

Biến số $allowBtSplit$ được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, $allowBtSplit$ được thiết lập bằng FALSE:
 - $cbSize$ là nhỏ hơn hoặc bằng $MinBtSizeY$
 - $cbWidth$ lớn hơn $maxBtSize$
 - $cbHeight$ lớn hơn $maxBtSize$
 - $mttDepth$ lớn hơn hoặc bằng $maxMttDepth$
 - $treeType$ bằng DUAL_TREE_CHROMA và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ là nhỏ hơn hoặc bằng 16
 - $btSplit$ bằng SPLIT_BT_VER và $treeType$ bằng DUAL_TREE_CHROMA và $(cbWidth / SubWidthC)$ là nhỏ hơn hoặc bằng 4
 - $treeType$ bằng DUAL_TREE_CHROMA và $modeType$ bằng MODE_TYPE_INTRA

...

6.4.3 Quá trình tách tam phân được cho phép

Biến số $allowTtSplit$ được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, $allowTtSplit$ được thiết lập bằng FALSE:

- cbSize là nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
 - cbWidth lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
 - cbHeight lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
 - mttDepth lớn hơn hoặc bằng maxMttDepth
 - $x0 + \text{cbWidth}$ lớn hơn $\text{pic_width_in_luma_samples}$
 - $y0 + \text{cbHeight}$ lớn hơn $\text{pic_height_in_luma_samples}$
 - treeType bằng `DUAL_TREE_CHROMA` và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ là nhỏ hơn hoặc bằng 32
 - btSplit bằng `SPLIT_TT_VER` và treeType bằng `DUAL_TREE_CHROMA` và $(\text{cbWidth} / \text{SubWidthC})$ là nhỏ hơn hoặc bằng 8
 - treeType bằng `DUAL_TREE_CHROMA` và modeType bằng `MODE_TYPE_INTRA`
- Ngược lại, allowTtSplit được thiết lập bằng TRUE.

5.8 Phương án ví dụ #8

Phương án dưới đây về việc cho phép `MODE_IBC` đối với các khối SCIPU sắc độ.

7.3.8.5 Cú pháp đơn vị mã hóa

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	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 &&</code> <code> !(((cbWidth == 4 && cbHeight == 4) modeType ==</code> <code> MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))</code>	
<code> cu_skip_flag[x0][y0]</code>	ae(v)
<code> if(cu_skip_flag[x0][y0] == 0 && slice_type != I</code> <code> && !(cbWidth == 4 && cbHeight == 4) && modeType ==</code> <code> MODE_TYPE_ALL)</code>	
<code> pred_mode_flag</code>	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]]) pred_mode_ibc_flag </pre>	ae(v)
--	-------

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

6.4.2 Quá trình tách nhị phân được cho phép

Biến số allowBtSplit được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE:
 - cbSize là nhỏ hơn hoặc bằng MinBtSizeY
 - cbWidth lớn hơn maxBtSize
 - cbHeight lớn hơn maxBtSize
 - mttDepth lớn hơn hoặc bằng maxMttDepth
 - treeType bằng DUAL_TREE_CHROMA và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ là nhỏ hơn hoặc bằng 16
 - modeType bằng MODE_TYPE_INTER và $cbWidth * cbHeight$ là nhỏ hơn hoặc bằng 32
 - treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA

...

6.4.3 Quá trình tách tam phân được cho phép

Biến số allowTtSplit được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, allowTtSplit được thiết lập bằng FALSE:
 - cbSize là nhỏ hơn hoặc bằng $2 * MinTtSizeY$
 - cbWidth lớn hơn $Min(MaxTbSizeY, maxTtSize)$

- cbHeight lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
 - mttDepth lớn hơn hoặc bằng maxMttDepth
 - $x0 + \text{cbWidth}$ lớn hơn pic_width_in_luma_samples
 - $y0 + \text{cbHeight}$ lớn hơn pic_height_in_luma_samples
 - treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ là nhỏ hơn hoặc bằng 32
 - modeType bằng MODE_TYPE_INTER và $\text{cbWidth} * \text{cbHeight}$ là nhỏ hơn hoặc bằng 64
 - treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA
- Ngược lại, allowTtSplit được thiết lập bằng TRUE.

5.10 Phương án ví dụ #10 về không cho phép chia khối khi modeType là MODE_TYPE_INTER (giải pháp 2)

6.4.2 Quá trình tách nhị phân được cho phép

Biến số allowBtSplit được rút ra như dưới đây:

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

...

6.4.3 Quá trình tách tam phân được cho phép

Biến số allowTtSplit được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, allowTtSplit được thiết lập bằng FALSE:

- cbSize là nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
- cbWidth lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- cbHeight lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
- mttDepth lớn hơn hoặc bằng maxMttDepth
- $x0 + \text{cbWidth}$ lớn hơn $\text{pic_width_in_luma_samples}$
- $y0 + \text{cbHeight}$ lớn hơn $\text{pic_height_in_luma_samples}$
- treeType bằng DUAL_TREE_CHROMA và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ là nhỏ hơn hoặc bằng 32
- modeType bằng MODE_TYPE_INTER
- treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA

– Ngược lại, allowTtSplit được thiết lập bằng TRUE.

5.11 Phương án ví dụ #11

Phương án dưới đây về việc các điều kiện ràng buộc tách tiếp của cây mã hóa khi MODE_TYPE_INTER được rút ra.

7.3.8.4 Cú pháp cây mã hóa

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Mô tả
...	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if (modeType == MODE_TYPE_INTER)	
mttDepth = max(mttDepth, maxMttDepth - 1)	
if (!split qt flag) {	
...	

5.12 Phương án ví dụ #12

Phương án dưới đây về việc các điều kiện ràng buộc đối với các phân chia khối nhỏ và các chế độ dự báo không được áp dụng khi chế độ bảng màu được phép.

7.4.9.4 Các ngữ nghĩa cây mã hóa

Biến số modeTypeCondition được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 0

- slice_type == I và qtbt_dual_tree_intra_flag bằng 1
- modeTypeCurr không bằng MODE_TYPE_ALL
- sps_palette_enabled_flag bằng 1
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
 - cbWidth * cbHeight bằng 128 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Ngược lại, modeTypeCondition được thiết lập bằng 0

5.13 Phương án ví dụ #13

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

7.4.9.4 Các ngữ nghĩa cây mã hóa

Biến số modeTypeCondition được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 0
 - slice_type == I và qtbt_dual_tree_intra_flag bằng 1
 - modeTypeCurr không bằng MODE_TYPE_ALL
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1

- cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
- cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu chroma_format_idc bằng 1 và một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
 - cbWidth * cbHeight bằng 128 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
- Ngược lại, modeTypeCondition được thiết lập bằng 0

5.14 Ví dụ #1 về bảo tín hiệu delta QP trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset,	Mô tả
partIdx, treeTypeCurr, modeTypeCurr) {	
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)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)	
{	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)	

cu chroma qp offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2,		
cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 1,		
cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2,		
cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
} else { /* SPLIT TT HOR */		
y1 = y0 + (cbHeight / 4)		
y2 = y0 + (3 * cbHeight / 4)		
cu qp delta subdiv)	qgOnYnext = qgOnY && (cbSubdiv + 2 <=	
cu chroma qp offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2,		
cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext, cbSubdiv + 1,		
cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2,		
cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
}		
} else {		
x1 = x0 + (cbWidth / 2)		
y1 = y0 + (cbHeight / 2)		
cbSubdiv + 2,	coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC,	
cqtDepth + 1, 0, 0, 0, treeType, modeType)		
if(x1 < pic_width_in_luma_samples)		
coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2,		
cqtDepth + 1, 0, 0, 1, treeType, modeType)		
if(y1 < pic_height_in_luma_samples)		

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)	
}	

5.15 Ví dụ #2 về báo tín hiệu delta QP trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, d ephOffset,	Mô tả
partIdx, treeTypeCurr, modeTypeCurr) {	
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)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)	
{	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)	

cu chroma_qp_offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
} else { /* SPLIT TT HOR */		
y1 = y0 + (cbHeight / 4)		
y2 = y0 + (3 * cbHeight / 4)		
qgOnYnext = qgOnY && (cbSubdiv + 2 <=		
cu_qp_delta subdiv)		
cu chroma_qp_offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
}		
} else {		
x1 = x0 + (cbWidth / 2)		
y1 = y0 + (cbHeight / 2)		
coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)		
if(x1 < pic_width in luma samples)		
coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)		
if(y1 < pic_height in luma samples)		

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, modeTypeCurr)	
}	

5.16 Ví dụ #3 về báo tín hiệu delta QP trong SCIPU

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, d epthOffset,	Mô tả
partIdx, treeTypeCurr, modeTypeCurr) {	
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, cbHeight, 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)	
qgOnYnext = qgOnY && (cbSubdiv + 2 <= cu_qp_delta_subdiv)	

cu chroma_qp_offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x1, y0, cbWidth / 2, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x2, y0, cbWidth / 4, cbHeight, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
} else { /* SPLIT TT HOR */		
y1 = y0 + (cbHeight / 4)		
y2 = y0 + (3 * cbHeight / 4)		
cu_qp_delta subdiv)	qgOnYnext = qgOnY && (cbSubdiv + 2 <=	
cu chroma_qp_offset subdiv)	qgOnCnext = qgOnC && (cbSubdiv + 2 <=	
coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnYnext, qgOnCnext, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnYnext, qgOnCnext, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
}		
} else {		
x1 = x0 + (cbWidth / 2)		
y1 = y0 + (cbHeight / 2)		
coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)		
if(x1 < pic_width_in_luma_samples)		
coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)		
if(y1 < pic_height_in_luma_samples)		

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, modeTypeCurr)	
}	

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

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, d epthOffset,	Mô tả
partIdx, treeTypeCurr, modeTypeCurr) {	
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	
}	
if(modeTypeCondition > 0)	
qgOnY = qgOnC = 0	
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)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)	
{	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	

cu qp delta subdiv)	qgOnY = qgOnY && (cbSubdiv + 2	<=	
cu chroma qp offset subdiv)	qgOnC = qgOnC && (cbSubdiv + 2	<=	
	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)		
	} else { /* SPLIT TT HOR */		
	y1 = y0 + (cbHeight / 4)		
	y2 = y0 + (3 * cbHeight / 4)		
cu qp delta subdiv)	qgOnY = qgOnY && (cbSubdiv + 2	<=	
cu chroma qp offset subdiv)	qgOnC = qgOnC && (cbSubdiv + 2	<=	
	coding_tree(x0, y0, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType)		
	coding_tree(x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)		
	coding_tree(x0, y2, cbWidth, cbHeight / 4, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType)		
	}		
	} else {		
	x1 = x0 + (cbWidth / 2)		
	y1 = y0 + (cbHeight / 2)		
	coding_tree(x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType)		
	if(x1 < pic width in luma samples)		
	coding_tree(x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType)		
	if(y1 < pic height in luma samples)		
	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)	
}	

5.18 Ví dụ #14 về vô hiệu hóa các khối trong ảnh sắc độ 2x8

6.4.2. Quá trình tách nhị phân được cho phép

Biến số allowBtSplit được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, allowBtSplit được thiết lập bằng FALSE:
 - cbSize là nhỏ hơn hoặc bằng MinBtSizeY
 - cbWidth lớn hơn maxBtSize
 - cbHeight lớn hơn maxBtSize
 - mttDepth lớn hơn hoặc bằng maxMttDepth
 - treeType bằng DUAL_TREE_CHROMA và $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$ là nhỏ hơn hoặc bằng 16
 - treeType bằng DUAL_TREE_CHROMA và btSplit bằng SPLIT_BT_VER và $(cbWidth / SubWidthC)$ bằng 4 và $(cbHeight / SubHeightC)$ bằng 8
 - treeType bằng DUAL_TREE_CHROMA và btSplit bằng SPLIT_BT_HOR và $(cbWidth / SubWidthC)$ bằng 2 và $(cbHeight / SubHeightC)$ bằng 16
 - treeType bằng DUAL_TREE_CHROMA và modeType bằng MODE_TYPE_INTRA

...

6.4.3. Quá trình tách tam phân được cho phép

Biến số `allowTtSplit` được rút ra như dưới đây:

- Nếu một hoặc nhiều trong số các điều kiện sau đây là đúng, `allowTtSplit` được thiết lập bằng `FALSE`:
 - `cbSize` là nhỏ hơn hoặc bằng $2 * \text{MinTtSizeY}$
 - `cbWidth` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
 - `cbHeight` lớn hơn $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$
 - `mttDepth` lớn hơn hoặc bằng `maxMttDepth`
 - $x0 + \text{cbWidth}$ lớn hơn `pic_width_in_luma_samples`
 - $y0 + \text{cbHeight}$ lớn hơn `pic_height_in_luma_samples`
 - `treeType` bằng `DUAL_TREE_CHROMA` và $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$ là nhỏ hơn hoặc bằng 32
 - `treeType` bằng `DUAL_TREE_CHROMA` và `btSplit` bằng `SPLIT_TT_VER` và $(\text{cbWidth} / \text{SubWidthC})$ is bằng 8 và $(\text{cbHeight} / \text{SubHeightC})$ bằng 8
 - `treeType` bằng `DUAL_TREE_CHROMA` và `btSplit` bằng `SPLIT_TT_HOR` và $(\text{cbWidth} / \text{SubWidthC})$ bằng 2 và $(\text{cbHeight} / \text{SubHeightC})$ bằng 32
 - `treeType` bằng `DUAL_TREE_CHROMA` và `modeType` bằng `MODE_TYPE_INTRA`
- Ngược lại, `allowTtSplit` được thiết lập bằng `TRUE`.

7.4.9.4. Các ngữ nghĩa cây mã hóa

Biến số `modeTypeCondition` được rút ra như dưới đây:

- Nếu một trong số các điều kiện sau đây là đúng, `modeTypeCondition` được thiết lập bằng 0
 - `slice_type == I` và `qtbtt_dual_tree_intra_flag` bằng 1
 - `modeTypeCurr` không bằng `MODE_TYPE_ALL`

- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1
 - cbWidth * cbHeight bằng 64 và split_qt_flag bằng 1
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth * cbHeight bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
- Ngược lại, nếu một trong số các điều kiện sau đây là đúng, modeTypeCondition được thiết lập bằng 1 + (slice_type != I ? 1 : 0)
 - cbWidth * cbHeight bằng 64 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_BT_VER
 - cbWidth * cbHeight bằng 128 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR hoặc SPLIT_TT_VER
 - cbWidth bằng 8 và (cbHeight / SubHeightC) bằng 8 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_VER
 - cbWidth bằng 16 và (cbHeight / SubHeightC) bằng 8 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_VER
 - cbWidth bằng 4 và (cbHeight / SubHeightC) bằng 16 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_BT_HOR hoặc SPLIT_TT_HOR
 - cbWidth bằng 4 và (cbHeight / SubHeightC) bằng 32 và MttSplitMode[x0][y0][mttDepth] bằng SPLIT_TT_HOR
- Ngược lại, modeTypeCondition được thiết lập bằng 0

7.4.9.7. Các ngữ nghĩa dữ liệu hợp nhất

ciip_flag[x0][y0] chỉ rõ liệu hợp nhất ảnh ngoài ảnh và dự báo ảnh trong ảnh được kết hợp có được áp dụng cho đơn vị mã hóa hiện tại hay không. Các chỉ số mảng x0, y0 chỉ rõ vị trí (x0, y0) của mẫu độ sáng trên cùng bên trái của khối mã hóa được xem xét so với mẫu độ sáng trên cùng bên trái của ảnh.

Khi ciip_flag[x0][y0] không có mặt, nó được suy ra như dưới đây:

- Nếu tất cả các điều kiện sau đây là đúng, `ciip_flag[x0][y0]` được suy ra bằng 1:

- `sps_ciip_enabled_flag` bằng 1.
 - `general_merge_flag[x0][y0]` bằng 1.
 - `merge_subblock_flag[x0][y0]` bằng 0.
 - `regular_merge_flag[x0][y0]` bằng 0.
 - `cbWidth` là nhỏ hơn 128.
 - `cbHeight` là nhỏ hơn 128.
 - `cbWidth` lớn hơn 4 và `cbHeight` không bằng 16.
 - `cbWidth * cbHeight` lớn hơn hoặc bằng 64.
- Ngược lại, `ciip_flag[x0][y0]` được suy ra bằng 0.

7.3.8.7. Cú pháp dữ liệu hợp nhất

	Descriptor
<code>merge_data(x0, y0, cbWidth, cbHeight, chType) {</code>	
<code>if (CuPredMode[chType][x0][y0] == MODE_IBC) {</code>	
<code>if (MaxNumIbcMergeCand > 1)</code>	
<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>} else {</code>	
<code>if (MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >=</code>	
<code>8)</code>	
<code>merge_subblock_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if (merge_subblock_flag[x0][y0] == 1) {</code>	
<code>if (MaxNumSubblockMergeCand > 1)</code>	
<code>merge_subblock_idx[x0][y0]</code>	<code>ae(v)</code>
<code>} else {</code>	
<code>if ((cbWidth * cbHeight) >= 64 && ((sps_ciip_enabled_flag &&</code>	
<code>cu_skip_flag[x0][y0] == 0 && cbWidth < 128 &&</code>	
<code>cbHeight < 128) </code>	
<code>(sps_triangle_enabled_flag &&</code>	
<code>MaxNumTriangleMergeCand > 1</code>	<code>&&</code>
<code>slice_type == B)))</code>	
<code>regular_merge_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if (regular_merge_flag[x0][y0] == 1) {</code>	
<code>if (sps_mmvd_enabled_flag)</code>	
<code>mmvd_merge_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if (mmvd_merge_flag[x0][y0] == 1) {</code>	
<code>if (MaxNumMergeCand > 1)</code>	
<code>mmvd_cand_flag[x0][y0]</code>	<code>ae(v)</code>
<code>mmvd_distance_idx[x0][y0]</code>	<code>ae(v)</code>
<code>mmvd_hướng_idx[x0][y0]</code>	<code>ae(v)</code>
<code>} else {</code>	
<code>if (MaxNumMergeCand > 1)</code>	
<code>merge_idx[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>} else {</code>	

if(sps_ciip_enabled_flag && sps_triangle_enabled_flag		
&&	MaxNumTriangleMergeCand > 1 && slice_type	
==	B	&&
	cu_skip_flag[x0][y0] == 0 && !(cbWidth ==	
4	&& cbWidth	= 16) &&
	(cbWidth * cbHeight) >= 64 && cbWidth < 128	
&& cbHeight < 128) {	ciip_flag[x0][y0]	ae(v)
	if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
	merge_idx[x0][y0]	ae(v)
	if(!ciip_flag[x0][y0] && MaxNumTriangleMergeCand >	
1) {	merge_triangle_split_dir[x0][y0]	ae(v)
	merge_triangle_idx0[x0][y0]	ae(v)
	merge_triangle_idx1[x0][y0]	ae(v)
	}	
	}	
	}	
	}	
	}	
	}	

FIG. 17A 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 thực hiện dưới dạng điện thoại thông minh, máy tính bảng, máy tính, bộ thu internet vạn vật (Internet of Things - IoT), và 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 các phương pháp được mô tả trong bản mô tả này. Bộ nhớ (các 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 các kỹ thuật được mô tả ở đây. Phần cứng xử lý video 1706 có thể được sử dụng để thực hiện, trong hệ mạch phần cứng, một số kỹ thuật được mô tả trong bản mô tả này.

FIG. 17B là ví dụ khác về sơ đồ khối về hệ thống xử lý video mà trong đó các kỹ thuật theo sáng chế có thể được thực hiện. FIG. 17B là sơ đồ khối thể hiện ví dụ về hệ thống xử lý video 1710 mà trong đó các kỹ thuật khác nhau được bộc lộ ở đây có thể được thực hiện. Các phương án thực hiện khác nhau có thể bao gồm một số hoặc tất cả các thành phần của hệ thống 1710. Hệ thống 1710 có thể bao gồm đầu vào 1712 để nhận nội dung video. Nội dung video có thể được nhận trong định dạng thô hoặc không nén, ví dụ, các giá trị điểm ảnh đa thành phần 8 hoặc 10 bit, hoặc có thể trong định dạng nén hoặc được tạo mã.

Đầu vào 1712 có thể biểu diễn giao diện mạng, giao diện buýt ngoại vi, hoặc giao diện lưu trữ. Các ví dụ về giao diện mạng bao gồm các giao diện dây dẫn chẳng hạn mạng Ethernet, mạng quang học thụ động (passive optical mạng - PON), v.v và các giao diện không dây chẳng hạn Wi-Fi hoặc các giao diện mạng tế bào.

Hệ thống 1710 có thể bao gồm thành phần mã hóa 1714 mà có thể thực hiện các phương pháp mã hóa hoặc tạo mã khác nhau được mô tả trong bản mô tả này. Thành phần mã hóa 1714 có thể làm giảm tốc độ bit trung bình của video từ đầu vào 1712 tới đầu ra của thành phần mã hóa 1714 để tạo ra sự biểu diễn được mã hóa của video. Do đó, các kỹ thuật mã hóa đôi khi được gọi là các kỹ thuật nén video hoặc chuyển mã video. Đầu ra của thành phần mã hóa 1714 có thể hoặc được lưu trữ, hoặc được truyền qua đường truyền thông được kết nối, như được biểu diễn bằng thành phần 1716. Sự biểu diễn dòng bit được lưu trữ hoặc được truyền thông (hoặc được mã hóa) của video được nhận ở đầu vào 1712 có thể được sử dụng bởi thành phần 1718 để tạo ra các giá trị điểm ảnh hoặc video có thể được hiển thị mà được gửi tới giao diện hiển thị 1720. Quá trình tạo ra video người dùng xem được từ sự biểu diễn dòng bit đôi khi được gọi là giải nén video. Hơn nữa, trong khi một số hoạt động xử lý video được gọi là các hoạt động hoặc các công cụ “mã hóa”, cần phải hiểu rằng các công cụ hoặc các hoạt động mã hóa được sử dụng ở bộ tạo mã và các công cụ hoặc các hoạt động giải mã tương ứng mà đảo ngược kết quả của việc mã hóa sẽ được thực hiện bởi bộ giải mã.

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

FIG. 22 là sơ đồ khối mà thể hiện ví dụ về hệ thống mã hóa video 100 mà có thể sử dụng các kỹ thuật theo sáng chế.

Như được thể hiện trên FIG. 22, hệ thống mã hóa video 100 có thể bao gồm bộ phận nguồn 110 và bộ phận đích 120. Bộ phận nguồn 110 tạo ra dữ liệu video được tạo mã mà có thể được gọi là bộ phận tạo mã video. Bộ phận đích 120 có thể giải mã dữ liệu video được tạo mã được tạo ra bởi bộ phận nguồn 110 mà có thể được gọi là bộ phận giải mã video.

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

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

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

Giao diện I/O 126 có thể bao gồm bộ thu và/hoặc môđem. Giao diện I/O 126 có thể thu nhận dữ liệu video được tạo mã từ bộ phận nguồn 110 hoặc vật ghi/máy chủ lưu trữ 130b. Bộ giải mã video 124 có thể giải mã dữ liệu video được tạo mã. Bộ phận hiển thị 122 có thể hiển thị dữ liệu video được giải mã tới người dùng. Bộ phận hiển thị 122 có thể được tích hợp với bộ phận đích 120,

hoặc có thể nằm ngoài bộ phận đích 120 mà được tạo cấu hình để giao diện với bộ phận hiển thị ngoại vi.

Bộ tạo mã video 114 và bộ giải mã video 124 có thể hoạt động theo chuẩn nén video, chẳng hạn chuẩn mã hóa video hiệu suất cao (High Efficiency Video Coding - HEVC), chuẩn mã hóa video vạn năng (Versatile Video Coding - VVC) và các chuẩn hiện tại và/hoặc tiếp theo khác.

FIG. 23 là sơ đồ khối thể hiện ví dụ về bộ tạo mã video 200, mà có thể là bộ tạo mã video 114 trong hệ thống 100 được thể hiện trên FIG. 22.

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

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

Trong các ví dụ khác, bộ tạo mã video 200 có thể bao gồm nhiều hơn, ít hơn, hoặc các thành phần chức năng khác nhau. Trong một ví dụ, đơn vị xác nhận 202 có thể bao gồm đơn vị bản sao khối trong ảnh (IBC). Đơn vị IBC có thể thực hiện sự xác nhận trong chế độ IBC mà trong đó ít nhất một ảnh tham chiếu là ảnh trong đó khối video hiện tại được bố trí.

Hơn nữa, một số thành phần, chẳng hạn đơn vị ước tính chuyển động 204 và đơn vị bù chuyển động 205 có thể được tích hợp mức cao, nhưng được trình bày một cách riêng rẽ trong ví dụ trên FIG. 5 nhằm mục đích giải thích.

Đơn vị chia 201 có thể chia ảnh thành một hoặc nhiều khối video. Bộ tạo mã video 200 và bộ giải mã video 300 có thể hỗ trợ các kích thước khối video khác nhau.

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

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

Đơn vị ước tính chuyển động 204 và đơn vị bù chuyển động 205 có thể thực hiện các hoạt động khác nhau đối với khối video hiện tại, ví dụ, tùy theo việc liệu khối video hiện tại có trong lát I, lát P, hoặc lát B.

Trong một số ví dụ, đơn vị ước tính chuyển động 204 có thể thực hiện dự báo đơn hướng đối với khối video hiện tại, và đơn vị ước tính chuyển động 204 có thể tra cứu các ảnh tham chiếu của danh sách 0 hoặc danh sách 1 đối với khối video tham chiếu cho khối video hiện tại. Sau đó đơn vị ước tính chuyển động 204 có thể tạo ra chỉ số tham chiếu mà chỉ báo ảnh tham chiếu trong danh sách 0 hoặc danh sách 1 mà bao gồm khối video tham chiếu và vectơ chuyển động mà chỉ báo sự chuyển vị không gian giữa khối video hiện tại và khối video tham chiếu. Đơn vị ước tính chuyển động 204 có thể xuất ra chỉ số tham chiếu, bộ chỉ báo hướng dự báo, và vectơ chuyển động dưới dạng thông tin chuyển động của khối video hiện tại. Đơn vị bù chuyển động 205 có thể tạo ra khối video được dự

báo của khối hiện tại dựa trên khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

Trong các ví dụ khác, đơn vị ước tính chuyển động 204 có thể thực hiện dự báo hai hướng đối với khối video hiện tại, đơn vị ước tính chuyển động 204 có thể tra cứu các ảnh tham chiếu trong danh sách 0 đối với khối video tham chiếu cho khối video hiện tại và cũng có thể tra cứu các ảnh tham chiếu trong danh sách 1 đối với khối video tham chiếu khác cho khối video hiện tại. Sau đó đơn vị ước tính chuyển động 204 có thể tạo ra các chỉ số tham chiếu mà chỉ báo các ảnh tham chiếu trong danh sách 0 và danh sách 1 bao gồm các khối video tham chiếu và các vector chuyển động mà chỉ báo các chuyển vị không gian giữa các khối video tham chiếu và khối video hiện tại. Đơn vị ước tính chuyển động 204 có thể xuất ra các chỉ số tham chiếu và các vector chuyển động của khối video hiện tại dưới dạng thông tin chuyển động của khối video hiện tại. Đơn vị bù chuyển động 205 có thể tạo ra khối video được dự báo của khối video hiện tại dựa trên các khối video tham chiếu được chỉ báo bởi thông tin chuyển động của khối video hiện tại.

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

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

Trong một ví dụ, đơn vị ước tính chuyển động 204 có thể chỉ báo, trong kết cấu cú pháp được liên kết với khối video hiện tại, giá trị mà chỉ báo tới bộ giải mã video 300 rằng khối video hiện tại có cùng thông tin chuyển động như khối video khác.

Trong ví dụ khác, đơn vị ước tính chuyển động 204 có thể nhận dạng, trong kết cấu cú pháp được liên kết với khối video hiện tại, khối video khác và

hiệu số vector chuyển động (MVD). Hiệu số vector chuyển động chỉ báo hiệu số giữa vector chuyển động của khối video hiện tại và vector chuyển động của khối video được chỉ báo. Bộ giải mã video 300 có thể sử dụng vector chuyển động của khối video được chỉ báo và hiệu số vector chuyển động để xác định vector chuyển động của khối video hiện tại.

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

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

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

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

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

Sau khi đơn vị xử lý chuyển đổi 208 tạo ra khối video hệ số chuyển đổi được liên kết với khối video hiện tại, đơn vị lượng tử hóa 209 có thể lượng tử hóa khối video hệ số chuyển đổi được liên kết với khối video hiện tại dựa trên

một hoặc nhiều giá trị thông số lượng tử hóa (QP) được liên kết với khối video hiện tại.

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

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

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

FIG. 24 là sơ đồ khối thể hiện ví dụ về bộ giải mã video 300 mà có thể là bộ giải mã video 114 trong hệ thống 100 được thể hiện trên FIG. 22.

Bộ giải mã video 300 có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật theo sáng chế. Trong ví dụ trên FIG. 24, bộ giải mã video 300 bao gồm nhiều thành phần chức năng. Các kỹ thuật được mô tả theo sáng chế có thể được chia sẻ trong số các thành phần khác nhau của bộ giải mã video 300. Trong một số ví dụ, bộ xử lý có thể được tạo cấu hình để thực hiện kỹ thuật bất kỳ hoặc tất cả các kỹ thuật được mô tả theo sáng chế.

Trong ví dụ trên FIG. 24, bộ giải mã video 300 bao gồm đơn vị giải mã entropi 301, đơn vị bù chuyển động 302, đơn vị dự báo trong ảnh 303, đơn vị nghịch đảo lượng tử hóa 304, đơn vị nghịch đảo chuyển đổi 305, và đơn vị tái tạo 306 và bộ đệm 307. Bộ giải mã video 300 có thể, trong một số ví dụ, thực hiện chuyển tác giải mã mà nói chung là nghịch đảo với chuyển tác tạo mã được mô tả liên quan tới bộ tạo mã video 200 (ví dụ, FIG. 23).

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

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

Đơn vị bù chuyển động 302 có thể sử dụng các bộ lọc nội suy như được sử dụng bởi bộ tạo mã video 20 trong khi tạo mã khối video để tính toán các giá trị được nội suy đối với các điểm ảnh con số nguyên của khối tham chiếu. Đơn vị bù chuyển động 302 có thể xác định các bộ lọc nội suy được sử dụng bởi bộ tạo mã video 200 theo thông tin cú pháp nhận được và sử dụng các bộ lọc nội suy để tạo ra các khối dự báo.

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

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

Đơn vị tái tạo 306 có thể lấy tổng các khối dư với các khối dự báo tương ứng được tạo ra bởi đơn vị bù chuyển động 202 hoặc đơn vị dự báo trong ảnh 303 để tạo thành các khối được giải mã. Nếu mong muốn, bộ lọc giải khối cũng có thể được áp dụng cho bộ lọc các khối được giải mã nhằm loại bỏ các thành phần lạ tạo khối. Sau đó các khối video được giải mã được lưu trữ trong bộ đệm 307, mà cấp các khối tham chiếu để bù chuyển động tiếp theo.

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

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

Một số phương án của công nghệ theo sáng chế bao gồm việc ra quyết định hoặc xác định để vô hiệu hóa công cụ hoặc chế độ xử lý video. Trong một ví dụ, khi công cụ hoặc chế độ xử lý video không được phép, bộ tạo mã sẽ không sử dụng công cụ hoặc chế độ trong việc biến đổi của khối của video thành biểu diễn dòng bit của video. Trong ví dụ khác, khi công cụ hoặc chế độ xử lý video không được phép, bộ giải mã sẽ xử lý dòng bit khi biết rằng dòng bit đã không

được cải biến bằng cách sử dụng công cụ hoặc chế độ xử lý video mà đã không được phép dựa trên quyết định hoặc xác định này.

Trong bản mô tả này, thuật ngữ “xử lý video” có thể dùng để chỉ tạo mã video, giải mã video, nén video hoặc giải nén video. Ví dụ, các thuật toán nén video có thể được áp dụng trong khi biến đổi từ biểu diễn điểm ảnh của video thành biểu diễn dòng bit tương ứng hoặc ngược lại. Biểu diễn dòng bit của khối video hiện tại có thể, ví dụ, tương ứng với các bit mà hoặc có cùng vị trí hoặc trải rộng trong các vị trí khác nhau bên trong dòng bit, như được xác định bởi cú pháp. Ví dụ, khối macro có thể được tạo mã xét theo các giá trị dư lỗi được chuyển đổi và được mã hóa và cũng bằng cách sử dụng các bit trong các tiêu đề và các trường khác trong dòng bit.

Các mệnh đề sau đây có thể được thực hiện như các phương án thực hiện ví dụ của công nghệ theo sáng chế. Nhóm thứ nhất của các mệnh đề có thể bao gồm các mệnh đề sau đây về các kỹ thuật được mô tả trong mục 1 của phần trước.

1. Phương pháp xử lý video, bao gồm các bước: phân tách, để biến đổi giữa vùng video của video và sự biểu diễn được mã hóa của vùng video, sự biểu diễn được mã hóa theo quy tắc cú pháp mà xác định mối quan hệ giữa kích thước khối sắc độ và định dạng màu của vùng video; và thực hiện sự biến đổi bằng cách thực hiện phân tách theo quy tắc cú pháp.
2. Phương pháp theo mệnh đề 1, trong đó định dạng màu là 4:4:4 và trong đó quy tắc cú pháp chỉ rõ rằng khối sắc độ phải chịu điều kiện ràng buộc kích thước giống như đối với các khối độ sáng.
3. Phương pháp theo mệnh đề 1, trong đó định dạng màu là 4:2:2 và trong đó quy tắc cú pháp chỉ rõ rằng khối sắc độ phải chịu điều kiện ràng buộc kích thước giống như đối với định dạng màu 4:2:0.
4. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 đến 3, trong đó cú pháp chỉ rõ rằng các chế độ dự báo và các phân chia khối nhỏ được sử dụng theo cách phụ thuộc định dạng sắc độ.

5. Phương pháp theo mệnh đề 1, trong đó quy tắc cú pháp xác định rằng dấu hiệu kích thước được cho phép nhỏ nhất là được phép để biến đổi vùng video dựa trên định dạng màu của vùng video.

Các mệnh đề sau đây 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 đặc tính của video và định dạng sắc độ của video, chế độ mã hóa của nút cây mã hóa của video; và thực hiện biến đổi giữa sự biểu diễn được mã hóa của video và khối video của nút cây mã hóa bằng cách sử dụng chế độ mã hóa được xác định.

7. Phương pháp theo mệnh đề 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 đặc tính là:

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

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

Các mệnh đề sau đây 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, liệu kích thước đã biết của các khối sắc độ có được cho phép trong vùng video của video hay không; và thực hiện sự biến đổi giữa vùng video và sự biểu diễn được mã hóa của vùng video dựa trên xác định này.

10. Phương pháp theo mệnh đề 9, trong đó quy tắc chỉ rõ rằng các khối sắc độ $2 \times N$ là không được phép do vùng video bao gồm phân chia cây kép.

11. Phương pháp theo mệnh đề 9, trong đó quy tắc chỉ rõ rằng các khối sắc độ $2N$ là không được phép do vùng video bao gồm phân chia cây đơn.

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

Các mệnh đề sau đây 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 mà cho phép sử dụng chế độ mã hóa đối với điều kiện video, rằng chế độ mã hóa được cho phép đối với vùng video; và thực hiện sự biến đổi giữa sự biểu diễn được mã hóa của các điểm ảnh trong vùng video và các điểm ảnh của vùng video dựa trên xác định này.

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

15. Phương pháp theo mệnh đề 14, trong đó các kích thước khối nhỏ bao gồm các kích thước khối độ sáng 8x4, 8x8, 16x4 hoặc 4xN.

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

17. Phương pháp theo mệnh đề 13, trong đó quy tắc cho phép sử dụng chế độ mã hóa bảng màu để biến đổi vùng video bằng cách sử dụng chế độ mã hóa `MODE_TYPE_INTER`.

Các mệnh đề sau đây 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 sự biến đổi giữa khối video của video và sự biểu diễn được mã hóa của khối video bằng cách sử dụng chế độ mã hóa video, trong đó phần tử cú pháp báo tín hiệu chế độ mã hóa được bao gồm một cách có chọn lọc trong sự biểu diễn được mã hóa dựa trên quy tắc.

19. Phương pháp theo mệnh đề 18, trong đó chế độ mã hóa video là chế độ mã hóa khối trong ảnh và trong đó quy tắc chỉ rõ việc sử dụng kiểu của chế độ mã hóa video để điều khiển việc bao gồm phần tử cú pháp trong sự biểu diễn được mã hóa.

20. Phương pháp theo mệnh đề 19, trong đó quy tắc chỉ ra một cách rõ ràng việc báo tín hiệu khối không SKIP.

21. Phương pháp theo mệnh đề 18, trong đó quy tắc chỉ rõ báo tín hiệu theo ngầm hiểu cờ bản sao khối trong ảnh dựa trên cờ bỏ qua và kiểu chế độ của khối video.

22. Phương pháp theo mệnh đề 18, trong đó chế độ mã hóa là chế độ mã hóa bảng màu và trong đó quy tắc chỉ rõ bao gồm một cách có chọn lọc bộ chỉ báo mã hóa bảng màu dựa trên kiểu chế độ của khối video.

Các mệnh đề sau đây 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 kiểu chuyển đổi được sử dụng trong khi biến đổi giữa khối sắc độ và sự biểu diễn được mã hóa của khối sắc độ là khác với kiểu chuyển đổi được sử dụng để biến đổi khối độ sáng tương ứng; và thực hiện sự biến đổi dựa trên xác định này.

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

Các mệnh đề sau đây 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 mệnh đề bất kỳ trong số các mệnh đề từ 1 đến 24 trong đó, việc biến đổi sử dụng chế độ dự báo ngoài ảnh và trong ảnh được kết hợp làm chế độ `MODE_TYPE_INTRA`.

26. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 18 tới 22, trong đó việc biến đổi sử dụng chế độ dự báo ngoài ảnh và trong ảnh được kết hợp làm chế độ `MODE_TYPE_INTER`. Ví dụ, khi xem xét CIIP là `MODE_TYPE_INTER`, các phương pháp được mô tả trong mục từ 14 tới 17 trong phần trước có thể được áp dụng. Hoặc khi các phương pháp được mô tả trong các mục từ 14 tới 16 được áp dụng, CIIP có thể được coi là `MODE_TYPE_INTER`.

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

27. Phương pháp xử lý video, bao gồm các bước: xác định, liệu quy tắc khối sắc độ nhỏ nhất có được thực thi trong khi biến đổi giữa sự biểu diễn được mã

hóa của vùng video và các giá trị điểm ảnh của vùng video, dựa trên điều kiện mã hóa của vùng video; và thực hiện sự biến đổi dựa trên xác định này.

28. Phương pháp theo mệnh đề 17, trong đó điều kiện mã hóa bao gồm định dạng màu của vùng video.

29. Phương pháp theo mệnh đề 18, trong đó vùng video có chiều rộng bằng M điểm ảnh và chiều cao bằng N điểm ảnh, và trong đó điều kiện mã hóa còn phụ thuộc vào các giá trị của M và/hoặc N.

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

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

31. Phương pháp xử lý video, bao gồm các bước: xác định, để biến đổi giữa sự biểu diễn được mã hóa của vùng video trong định dạng 4:2:2 và các giá trị điểm ảnh của vùng video, kiểu chế độ được sử dụng để biến đổi dựa trên việc liệu quy tắc khối sắc độ nhỏ nhất có được phép đối với vùng video này hay không; và thực hiện sự biến đổi dựa trên xác định này.

32. Phương pháp theo mệnh đề 31, trong đó kiểu chế độ của vùng video được thiết lập bằng 1 do vùng video có định dạng 4:2:2 và quy tắc khối sắc độ nhỏ nhất là được phép.

33. Phương pháp theo mệnh đề 31, trong đó bước xác định kiểu chế độ bao gồm xác định kiểu chế độ là kiểu INTRA do quy tắc khối sắc độ nhỏ nhất là được phép đối với vùng video này.

34. Phương pháp theo mệnh đề 31, trong đó bước xác định kiểu chế độ bao gồm xác định rằng kiểu chế độ INTER là không được phép do quy tắc khối sắc độ nhỏ nhất là được phép đối với vùng video.

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

35. Phương pháp xử lý video, bao gồm các bước: xác định, để biến đổi giữa sự biểu diễn được mã hóa của khối video và khối video của video, liệu việc phân chia khối có được cho phép trong khi biến đổi hay không, dựa trên kiểu chế độ

được sử dụng trong khi biến đổi hoặc kích thước của khối video; và thực hiện sự biến đổi bằng cách sử dụng xác định này.

36. Phương pháp theo mệnh đề 35, trong đó việc chia khối bao gồm phân chia cây nhị phân hoặc phân chia cây tam phân.

37. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 35 tới 36 trong đó, trong trường hợp mà kiểu chế độ là chế độ INTER, việc phân chia khối dựa trên quy tắc hạn chế mà cho phép hoặc không cho phép các kiểu phân chia.

Các mệnh đề sau đây có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong mục 34 của phần trước.

38. Phương pháp xử lý video, bao gồm các bước: xác định, để biến đổi giữa sự biểu diễn được mã hóa của đoạn video của video và đoạn video, để áp dụng chế độ xử lý riêng đối với khối sắc độ có kích thước $M \times N$, trong đó M và N là các số nguyên; và thực hiện sự biến đổi dựa trên xác định này.

39. Phương pháp theo mệnh đề 38, trong đó chế độ xử lý riêng không cho phép sử dụng các khối sắc độ trong khi biến đổi.

40. Phương pháp theo mệnh đề 38, trong đó chế độ xử lý riêng sử dụng dự báo DC để dự báo trong ảnh của khối sắc độ.

41. Phương pháp theo mệnh đề 38, trong đó chế độ xử lý riêng bao gồm việc sử dụng mô hình tuyến tính giao thành phần để dự báo các hệ số trong ảnh của khối sắc độ từ các hệ số độ sáng được giảm kích thước mẫu tương ứng.

42. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 38 tới 41, trong đó chế độ xử lý riêng không cho phép sử dụng khối sắc độ do đoạn video sử dụng phân chia cây kép.

43. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 38 tới 41, trong đó chế độ xử lý riêng không cho phép sử dụng khối sắc độ do đoạn video sử dụng phân chia cây đơn.

44. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 43, trong đó việc biến đổi bao gồm tạo mã video thành sự biểu diễn được mã hóa.

45. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 43, trong đó việc biến đổi bao gồm giải mã sự biểu diễn được mã hóa để tạo ra các giá trị điểm ảnh của video.

46. Thiết bị giải mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều mệnh đề trong số các mệnh đề từ 1 tới 45.

47. Thiết bị tạo mã video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều mệnh đề trong số các mệnh đề từ 1 tới 45.

48. Sản phẩm chương trình máy tính có mã máy tính được lưu trữ trên đó, mã này, khi được thực thi bởi bộ xử lý, khiến bộ xử lý thực hiện phương pháp được nêu trong mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 45.

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

Nhóm thứ hai các mệnh đề có thể được thực hiện cùng với các kỹ thuật bổ sung được mô tả trong các mục từ 34 tới 36 của phần trước.

1. Phương pháp xử lý video (ví dụ, phương pháp 2100 được thể hiện trên FIG. 21), bao gồm các bước: thực hiện bước biến đổi 2102 giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng, do khối sắc độ có kích thước $M \times N$, khối sắc độ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

2. Phương pháp theo mệnh đề 1, trong đó quy tắc còn chỉ rõ rằng chế độ dự báo DC được sử dụng đối với khối sắc độ, và trong đó chế độ dự báo DC rút ra các giá trị dự báo của thành phần sắc độ dựa trên giá trị trung bình của các khối lân cận của khối sắc độ.

3. Phương pháp theo mệnh đề 1, trong đó quy tắc còn chỉ rõ rằng chế độ dự báo mô hình tuyến tính giao thành phần (CCLM) được sử dụng đối với khối sắc độ, trong đó chế độ dự báo CCLM sử dụng chế độ tuyến tính để rút ra các giá trị dự báo của thành phần sắc độ từ thành phần khác.

4. Phương pháp theo mệnh đề 2 hoặc 3, trong đó $M=2$ mẫu sắc độ và $N=8, 16$, hoặc 32 mẫu sắc độ.

5. Phương pháp theo mệnh đề 2 hoặc 3, trong đó quy tắc còn chỉ rõ rằng các điểm ảnh lân cận được tái tạo không được sử dụng đối với chế độ dự báo DC hoặc chế độ dự báo CCLM.
6. Phương pháp theo mệnh đề 2, trong đó quy tắc còn chỉ rõ rằng các điểm ảnh tham chiếu lân cận được thiết lập bằng $1 \ll (\text{bitDepth} - 1)$ đối với chế độ dự báo DC.
7. Phương pháp theo mệnh đề 2 hoặc 3, trong đó quy tắc còn chỉ rõ rằng phương pháp kết hợp dự báo trong ảnh phụ thuộc vị trí (position dependent intra prediction combination - PDPC) không được áp dụng cho chế độ dự báo DC hoặc chế độ dự báo CCLM, và trong đó phương pháp PDPC kết hợp các mẫu lân cận với tín hiệu dự báo của vùng video để tạo ra tín hiệu dự báo được tinh chỉnh của vùng video.
8. Phương pháp theo mệnh đề 2 hoặc 3, trong đó quy tắc còn chỉ rõ rằng thông tin về chế độ dự báo DC hoặc chế độ dự báo CCLM được rút ra mà không được mã hóa đối với khối sắc độ.
9. Phương pháp theo mệnh đề 8, trong đó quy tắc còn chỉ rõ rằng, đối với chế độ dự báo CCLM mà thỏa mãn $\text{predC} = a * \text{recL} + b$, giá trị dự báo sắc độ, predC , được rút ra từ giá trị được tái tạo độ sáng, recL , và các thông số, a và b , được khởi tạo tới các giá trị cố định mặc định ở khởi đầu của vùng video.
10. Phương pháp theo mệnh đề 9, trong đó bảng vào trước ra trước được duy trì để cập nhật các thông số, a và b .
11. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 10, trong đó quy tắc này được áp dụng cho khối sắc độ khác được mã hóa bằng cách sử dụng dự báo CCLM và có chiều rộng không bằng M hoặc chiều cao không bằng N .
12. Phương pháp theo mệnh đề 11, trong đó các thông số CCLM của các vùng video được mã hóa trước đó được sử dụng đối với vùng video này thay vì được rút ra dựa trên thông tin tái tạo của các khối lân cận của vùng video.
13. Phương pháp theo mệnh đề 11, trong đó các thông số CCLM được lưu trữ trong bảng và được cập nhật bằng cách sử dụng bảng vào trước ra trước.

14. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ mã hóa trong ảnh trong trường hợp sự phân chia cây kép được sử dụng, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.
15. Phương pháp theo mệnh đề 14, trong đó $M=2$ mẫu sắc độ và $N=8$, 16, hoặc 32 mẫu sắc độ.
16. Phương pháp theo mệnh đề 14, trong đó quy tắc còn chỉ rõ rằng, do phân chia cây kép, sơ đồ phân chia đã biết là không được phép đối với vùng video mà là nút cây mã hóa với kích thước bằng $(M^*a) \times (N^*b)$, trong đó M^*a là số lượng của các mẫu sắc độ theo chiều rộng của nút cây mã hóa và N^*b là số lượng của các mẫu sắc độ theo chiều cao của nút cây mã hóa.
17. Phương pháp theo mệnh đề 16, trong đó sơ đồ phân chia đã biết là tách cây nhị phân (BT) dọc mà trong đó vùng video được tách thành hai phần theo hướng dọc và trong đó $a=2$ và $b=1$.
18. Phương pháp theo mệnh đề 16, trong đó sơ đồ phân chia đã biết là tách cây nhị phân (BT) ngang mà trong đó vùng video được tách thành hai phần theo hướng ngang và trong đó $a=1$ và $b=2$.
19. Phương pháp theo mệnh đề 16, trong đó sơ đồ phân chia đã biết là tách cây tam phân (TT) dọc mà trong đó vùng video được tách thành ba phần theo hướng dọc và trong đó $a=4$ và $b=1$.
20. Phương pháp theo mệnh đề 16, trong đó sơ đồ phân chia đã biết là tách cây tam phân (TT) ngang mà trong đó vùng video được tách thành ba phần theo hướng ngang và trong đó $a=1$ và $b=4$.
21. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa khối sắc độ của vùng video của ảnh video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được phép được biểu diễn trong sự biểu diễn được mã hóa

bằng cách sử dụng chế độ mã hóa trong ảnh trong trường hợp sự phân chia cây đơn được sử dụng, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ, và trong đó chế độ mã hóa trong ảnh bao gồm việc mã hóa khối sắc độ dựa trên vùng video được mã hóa trước đó của ảnh video.

22. Phương pháp theo mệnh đề 21, trong đó $M=2$ mẫu sắc độ và $N=8, 16$, hoặc 32 mẫu sắc độ.

23. Phương pháp theo mệnh đề 21 hoặc 22, trong đó quy tắc còn chỉ rõ rằng, do phân chia cây đơn, sơ đồ phân chia đã biết là không được phép đối với các thành phần sắc độ của vùng video mà là nút cây mã hóa với kích thước bằng $(M \cdot a \cdot \text{subWidthC}) \times (N \cdot b \cdot \text{subHeightC})$, trong đó $M \cdot a$ là số lượng của các mẫu độ sáng theo chiều rộng của nút cây mã hóa và $N \cdot b$ là số lượng của các mẫu độ sáng theo chiều cao của nút cây mã hóa, và subWidthC và subHeightC lần lượt chỉ báo các tỷ số lấy mẫu con sắc độ theo các kích thước chiều rộng và chiều cao.

24. Phương pháp theo mệnh đề 23, trong đó sơ đồ phân chia đã biết là tách cây nhị phân (BT) dọc mà trong đó vùng video được tách thành hai phần theo hướng dọc và trong đó $a=2$ và $b=1$.

25. Phương pháp theo mệnh đề 23, trong đó sơ đồ phân chia đã biết là tách cây tam phân (TT) dọc mà trong đó vùng video được tách thành ba phần theo hướng dọc và trong đó $a=4$ và $b=1$.

26. Phương pháp theo mệnh đề 23, trong đó sơ đồ phân chia đã biết là tách cây nhị phân (BT) ngang mà trong đó vùng video được tách thành hai phần theo hướng ngang và trong đó $a=1$ và $b=2$.

27. Phương pháp theo mệnh đề 23, trong đó sơ đồ phân chia đã biết là tách cây tam phân (TT) ngang mà trong đó vùng video được tách thành ba phần theo hướng ngang và trong đó $a=1$ và $b=2$.

28. Phương pháp theo mệnh đề 23, trong đó sơ đồ phân chia đã biết là tách cây tam phân (TT) ngang mà trong đó vùng video được tách thành ba phần theo hướng ngang và trong đó $a=1$ và $b=4$.

29. Phương pháp theo mệnh đề 23, trong đó khối sắc độ có kích thước bằng $(M \times a) \times (N \times d)$.
30. Phương pháp theo mệnh đề 23, trong đó kiểu mã hóa của khối sắc độ là `MODE_TYPE_INTRA` mà cho phép sử dụng chế độ trong ảnh, chế độ bảng màu, và chế độ sao chép khối trong ảnh để biến đổi.
31. Phương pháp theo mệnh đề 23, trong đó quy tắc còn chỉ rõ rằng sơ đồ phân chia đã biết được cho phép đối với các thành phần độ sáng của nút cây mã hóa.
32. Phương pháp theo mệnh đề 31, trong đó nút cây mã hóa được tách thành các nút con và chế độ ngoài ảnh và/hoặc chế độ sao chép khối trong ảnh là được cho phép đối với nút con.
33. Phương pháp theo mệnh đề 22, trong đó quy tắc còn chỉ rõ rằng thành phần sắc độ được tách theo thành phần độ sáng được sắp xếp và chế độ ngoài ảnh và/hoặc chế độ sao chép khối trong ảnh là được cho phép đối với thành phần sắc độ này.
34. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa khối sắc độ của vùng video của video và sự biểu diễn được mã hóa của video theo quy tắc; trong đó quy tắc chỉ rõ rằng khối sắc độ có kích thước $M \times N$ không được phép được biểu diễn trong sự biểu diễn được mã hóa bằng cách sử dụng chế độ dự báo ngoài ảnh và trong ảnh kết hợp (CIIP) mà kết hợp dự báo ngoài ảnh với dự báo trong ảnh, trong đó M và N là các số nguyên mà lần lượt chỉ báo chiều rộng và chiều cao của khối sắc độ.
35. Phương pháp theo mệnh đề 34, trong đó quy tắc còn chỉ rõ rằng chế độ CIIP không được cho phép đối với các thành phần sắc độ của vùng video mà là nút cây mã hóa với kích thước bằng $(M \times \text{subWidthC}) \times (N \times \text{subHeightC})$, nhờ đó subWidthC và subHeightC lần lượt chỉ báo các tỷ số lấy mẫu con sắc độ theo các kích thước chiều rộng và chiều cao.
36. Phương pháp theo mệnh đề 34, trong đó bước thực hiện sự biến đổi bao gồm quá trình trộn sắc độ mà xác định khối dự báo sắc độ của khối sắc độ bằng cách trộn các dự báo của khối sắc độ bằng cách sử dụng dự báo ngoài ảnh trong trường hợp mà vùng video được mã hóa bằng cách sử dụng chế độ CIIP.

37. Phương pháp theo mệnh đề 36, trong đó quy tắc còn chỉ rõ rằng dự báo trong ảnh độ sáng được áp dụng cho vùng video và dự báo trong ảnh sắc độ không được áp dụng cho vùng video với kích thước bằng $(M \times \text{subWidthC}) \times (N \times \text{subHeightC})$, trong đó subWidthC và subHeightC lần lượt chỉ báo các tỷ số lấy mẫu con sắc độ theo các kích thước chiều rộng và chiều cao.
38. Phương pháp theo mệnh đề 34, trong đó quy tắc còn chỉ rõ rằng dự báo DC được sử dụng đối với khối sắc độ có kích thước $M \times N$ trong trường hợp mà vùng video được mã hóa bằng cách sử dụng chế độ CIIP.
39. Phương pháp theo mệnh đề 34, trong đó quy tắc còn chỉ rõ rằng dự báo CCLM mà sử dụng chế độ tuyến tính để rút ra các giá trị dự báo của thành phần sắc độ từ thành phần khác được sử dụng đối với khối sắc độ có kích thước $M \times N$ trong trường hợp mà vùng video được mã hóa bằng cách sử dụng chế độ CIIP.
40. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được mã hóa của video theo quy tắc, trong đó quy tắc chỉ rõ rằng việc liệu kích thước chuyển đổi tối đa có thể áp dụng cho vùng video có được báo tín hiệu hay không, và cách thức báo tín hiệu, là phụ thuộc vào kiểu chế độ mã hóa được sử dụng để biểu diễn vùng video trong sự biểu diễn được mã hóa, trong đó kiểu chế độ mã hóa là (i) chế độ mã hóa gần như không tổn thất mà trong đó thông số lượng tử hóa (QP) được áp dụng cho vùng video là nhỏ hơn ngưỡng, (ii) chế độ mã hóa không tổn thất mà trong đó việc chuyển đổi dữ liệu dư đối với vùng video được vòng tránh, hoặc (iii) chế độ mã hóa có tổn thất mà trong đó việc chuyển đổi dữ liệu dư đối với vùng video được bỏ qua.
41. Phương pháp theo mệnh đề 40, trong đó quy tắc còn chỉ rõ rằng kích thước chuyển đổi tối đa đối với chế độ mã hóa không tổn thất và/hoặc chế độ mã hóa gần như không tổn thất bằng $\text{maxLosslessTbSizeY}$.
42. Phương pháp theo mệnh đề 40, trong đó chế độ mã hóa không tổn thất được báo tín hiệu ở bậc đơn vị mã hóa bằng cách sử dụng phần tử cú pháp, `cu_transquant_bypass_flag`.

43. Phương pháp theo mệnh đề 40, trong đó ngưỡng được báo tín hiệu trong tập thông số bộ giải mã (DPS), tập thông số video (VPS), tập thông số dãy (SPS), tập thông số ảnh (PPS), ô ảnh, lát, hoặc tiêu đề ô gạch.
44. Phương pháp theo mệnh đề 40, trong đó `transform_skip_flag` bằng 1 đối với chế độ mã hóa gần như không tổn thất của vùng video.
45. Phương pháp theo mệnh đề 40, trong đó quy tắc còn chỉ rõ rằng kích thước chuyển đổi tối đa thứ nhất đối với chế độ mã hóa không tổn thất và/hoặc chế độ mã hóa gần như không tổn thất của vùng video được thiết lập dựa trên i) kích thước chuyển đổi tối đa khác được sử dụng đối với chế độ mã hóa có tổn thất, ii) kích thước đơn vị cây mã hóa tối đa, và/hoặc iii) kích thước bỏ qua chuyển đổi tối đa.
46. Phương pháp theo mệnh đề 40, trong đó kích thước chuyển đổi tối đa đối với chế độ mã hóa không tổn thất hoặc chế độ mã hóa gần như không tổn thất của vùng video được báo tín hiệu ở bậc đơn vị video tương ứng với phân tử cú pháp (SE) ở tập thông số dãy (SPS), tập thông số video (VPS), tập thông số ảnh (PPS), ảnh con, lát, ô ảnh, ô gạch, đơn vị cây mã hóa (CTU), hoặc đơn vị dữ liệu đường liên hợp ảo (VPDU).
47. Phương pháp theo mệnh đề 46, trong đó SE được báo tín hiệu tùy theo kích thước khối cây mã hóa (CTB) tối đa, kích thước chuyển đổi tối đa, và/hoặc kích thước bỏ qua chuyển đổi tối đa.
48. Phương pháp theo mệnh đề 47, trong đó SE được báo tín hiệu trong trường hợp mà `log2_ctu_size_minus5` là lớn hơn N, trong đó N là số nguyên.
49. Phương pháp theo mệnh đề 47, trong đó SE được báo tín hiệu trong trường hợp mà `sps_max_luma_transform_size_64_flag` bằng M, trong đó M là số nguyên.
50. Phương pháp theo mệnh đề 47, trong đó SE được báo tín hiệu trong trường hợp mà `log2_transform_skip_max_size_minus2` là lớn hơn K.
51. Phương pháp theo mệnh đề 47, trong đó hiệu số giữa `log(maxLosslessTbSizeY)` và X được báo tín hiệu, và trong đó `maxLosslessTbSizeY` chỉ báo kích thước chuyển đổi tối đa đối với chế độ mã hóa gần như không tổn thất hoặc chế độ mã hóa không tổn thất của vùng video.

52. Phương pháp theo mệnh đề 51, trong đó X bằng $\log_2_ctu_size_minus5$, $sps_max_luma_transform_size_64_flag$, $\log_2_transform_skip_max_size_minus2$, hoặc số nguyên cố định.
53. Phương pháp theo mệnh đề 47, trong đó hiệu số giữa $maxLosslessTbSizeY$ và X được bảo tín hiệu, và trong đó $maxLosslessTbSizeY$ chỉ báo kích thước chuyển đổi tối đa đối với chế độ mã hóa gần như không tổn thất hoặc chế độ mã hóa không tổn thất của vùng video.
54. Phương pháp theo mệnh đề 47, trong đó giá trị tương ứng với hiệu số giữa $maxLosslessTbSizeY$ và X minus K được bảo tín hiệu, trong đó $maxLosslessTbSizeY$ chỉ báo kích thước chuyển đổi tối đa đối với chế độ mã hóa gần như không tổn thất hoặc chế độ mã hóa không tổn thất của vùng video và K là số nguyên.
55. Phương pháp theo mệnh đề 53 hoặc 54, trong đó X bằng $CtbSizeY$, $MaxTbSize$, $MaxTsSize$, hoặc số nguyên cố định.
56. Phương pháp theo mệnh đề 46, trong đó $maxLosslessTbSizeY$ của vùng video được rút ra từ SE, và trong đó $maxLosslessTbSizeY$ chỉ báo kích thước chuyển đổi tối đa đối với chế độ mã hóa gần như không tổn thất hoặc chế độ mã hóa không tổn thất của vùng video.
57. Phương pháp theo mệnh đề 46, trong đó phần tử cú pháp tương ứng với cờ.
58. Phương pháp theo mệnh đề 57, trong đó trong trường hợp mà cờ không có mặt, thì giá trị mà là số nguyên dương được suy ra.
59. Phương pháp theo mệnh đề 40, trong đó quy tắc còn chỉ rõ rằng kích thước chuyển đổi tối đa được rút ra đối với chế độ mã hóa không tổn thất và/hoặc chế độ mã hóa gần như không tổn thất của vùng video.
60. Phương pháp theo mệnh đề 59, trong đó kích thước chuyển đổi tối đa được thiết lập bằng i) giá trị số nguyên cố định, ii) kích thước chuyển đổi tối đa khác được sử dụng đối với chế độ mã hóa có tổn thất, iii) kích thước đơn vị cây mã hóa tối đa, iv) kích thước VPDU, v) kích thước khối bỏ qua chuyển đổi tối đa, hoặc vi) kích thước khối mã hóa phần dư tối đa.

61. Phương pháp xử lý video, bao gồm các bước: thực hiện sự biến đổi giữa vùng video của video và sự biểu diễn được mã hóa của video theo quy tắc, trong đó quy tắc chỉ rõ rằng khối dư của vùng video được chia thành một hoặc nhiều khối dư con để mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất dựa trên kích thước chuyển đổi tối đa của vùng video mà mã hóa không tổn thất và/hoặc mã hóa gần như không tổn thất được áp dụng tới đó, và trong đó mã hóa không tổn thất bao gồm vòng tránh chuyển đổi khối dư đối với vùng video và mã hóa gần như không tổn thất sử dụng thông số lượng tử hóa (QP) mà nhỏ hơn ngưỡng.

62. Phương pháp theo mệnh đề 61, trong đó quy tắc còn chỉ rõ rằng chiều rộng và chiều cao của một hoặc nhiều khối dư con thu được sau khi chia không lớn hơn kích thước chuyển đổi tối đa.

63. Phương pháp theo mệnh đề 62, trong đó quy tắc còn chỉ rõ lặp lại việc chia khối dư cho tới khi chiều rộng và chiều cao của một hoặc nhiều khối dư con không lớn hơn kích thước chuyển đổi tối đa.

64. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 63, trong đó bước biến đổi bao gồm tạo mã video thành sự biểu diễn được mã hóa.

65. Phương pháp theo mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 63, trong đó bước biến đổi bao gồm giải mã sự biểu diễn được mã hóa để tạo ra video.

66. Thiết bị xử lý video bao gồm bộ xử lý được tạo cấu hình để thực hiện phương pháp được nêu trong một hoặc nhiều mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 65.

67. Vật ghi máy tính đọc được lưu trữ mã chương trình mà, khi được thực thi, khiến bộ xử lý thực hiện phương pháp được nêu trong một hoặc nhiều mệnh đề bất kỳ trong số các mệnh đề từ 1 tới 65.

68. Vật ghi máy tính đọc được mà lưu trữ sự biểu diễn được mã hóa hoặc biểu diễn dòng bit được tạo ra theo phương pháp bất kỳ trong số các phương pháp được mô tả trên đây.

Các giải pháp, các ví dụ, các phương án, các môđun và các hoạt động chức năng được bộc lộ và khác nữa mà được mô tả trong phần mô tả này có thể được thực hiện trong hệ mạch điện tử kỹ thuật số, hoặc trong phần mềm, phần

sụn, hoặc phần cứng máy tính, bao gồm các kết cấu được bộc lộ trong phần mô tả này và các kết cấu tương đương, hoặc trong sự kết hợp một hoặc nhiều trong số chúng. Các phương án được bộc lộ và 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 môđun của các lệnh chương trình máy tính được tạo mã trong vật ghi máy tính đọc được để được thực thi bởi, hoặc để điều khiển hoạt động của, thiết bị xử lý dữ liệu. Vật ghi máy tính đọc được có thể là bộ phận lưu trữ máy tính đọc được, nền lưu trữ máy tính đọc được, bộ phận nhớ, sản phẩm thực hiện tín hiệu lan truyền máy đọc được, hoặc sự kết hợp của một hoặc nhiều trong số chúng. Thuật ngữ “thiết bị xử lý dữ liệu” bao hàm tất cả các thiết bị, các bộ phận, và các máy để xử lý dữ liệu, bao gồm ví dụ bộ xử lý lập trình được, máy tính, hoặc nhiều bộ xử lý hoặc máy tính. Thiết bị có thể bao gồm, ngoài phần cứng ra, mã mà tạo ra môi trường thực thi đối với chương trình máy tính đang được đề cập, ví dụ, mã mà tạo thành phần sụn bộ xử lý, ngăn xếp giao thức, hệ thống quản lý cơ sở dữ liệu, hệ điều hành, hoặc sự kết hợp của một hoặc nhiều trong số chúng. Tín hiệu được lan truyền là tín hiệu được tạo ra một cách nhân tạo, ví dụ, tín hiệu điện, quang hoặc điện từ do máy tạo ra, mà được tạo ra để tạo mã thông tin để truyền tới thiết bị thu phù hợp.

Chương trình máy tính (còn được biết như chương trình, phần mềm, ứng dụng phần mềm, tập lệnh, hoặc mã) có thể được viết dưới dạng bất kỳ của ngôn ngữ lập trình, bao gồm các ngôn ngữ biên dịch hoặc diễn giải, và có thể được khai triển dưới dạng bất kỳ, bao gồm dưới dạng chương trình độc lập hoặc dưới dạng môđun, thành phần, thủ tục con, hoặc đơn vị khác phù hợp để sử dụng trong môi trường máy tính. Chương trình máy tính không nhất thiết tương ứng với tập tin trong hệ thống tập tin. Chương trình có thể được lưu trữ trong phần của tập tin mà 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 tập tin dành riêng cho chương trình đang được đề cập, hoặc trong nhiều tập tin phối hợp (ví dụ, các tập tin mà lưu trữ một hoặc nhiều môđun, chương trình con, hoặc các đoạn mã). Chương trình máy tính có thể được triển khai để được thực thi trên một máy tính

hoặc trên nhiều máy tính mà được bố trí tại một địa điểm hoặc được phân tán trên nhiều địa điểm và được kết nối với nhau bởi mạng truyền thông.

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

Các bộ xử lý phù hợp để thực thi chương trình máy tính bao gồm, ví dụ, các bộ vi xử lý cả đa năng lẫn chuyên dụng, và một hoặc nhiều bộ xử lý bất kỳ thuộc kiểu máy tính kỹ thuật số bất kỳ. Một cách tổng quát, bộ xử lý sẽ nhận các lệnh và dữ liệu từ bộ nhớ chỉ đọc hoặc bộ nhớ truy cập ngẫu nhiên hoặc từ cả hai. Các phần tử thiết yếu của máy tính là bộ xử lý để thực hiện các lệnh và một hoặc nhiều bộ phận nhớ để lưu trữ các lệnh và dữ liệu. Một cách tổng quát, máy tính cũng sẽ bao gồm, hoặc được gắn kết theo cách vận hành được để nhận dữ liệu từ hoặc truyền dữ liệu tới, hoặc cả hai, một hoặc nhiều bộ phận lưu trữ dung lượng lớn để lưu trữ dữ liệu, ví dụ, các đĩa từ tính, quang từ, hoặc các đĩa quang. Tuy nhiên, máy tính không cần có các bộ phận này. Vật ghi máy tính đọc được phù hợp để lưu trữ các lệnh chương trình máy tính và dữ liệu bao gồm tất cả các dạng của bộ nhớ bất biến, môi trường và các thiết bị nhớ, bao gồm ví dụ các bộ phận nhớ bán dẫn, ví dụ, EPROM, EEPROM, và các bộ phận nhớ nhanh; các đĩa từ tính, ví dụ, các ổ cứng trong hoặc các ổ cứng di động; các đĩa quang từ; và các đĩa CD ROM và DVD-ROM. Bộ xử lý và bộ nhớ có thể được bổ sung bởi, hoặc được kết hợp trong, hệ mạch logic chuyên dụng.

Trong khi phần mô tả này bao gồm nhiều đặc điểm cụ thể, các đặc điểm này không được diễn giải như là các hạn chế đối với phạm vi của đối tượng bất kỳ hoặc của những gì có thể được yêu cầu bảo hộ, mà như là các phần mô tả của các dấu hiệu mà có thể cụ thể với các phương án cụ thể của các kỹ thuật cụ thể. Các dấu hiệu đã biết mà được mô tả trong phần mô tả này trong ngữ cảnh của các phương án riêng rẽ cũng có thể được thực hiện một cách kết hợp trong một

phương án. Ngược lại, các dấu hiệu khác nhau mà được mô tả trong ngữ cảnh của một phương án cũng có thể được thực hiện trong nhiều phương án một cách riêng rẽ hoặc trong tập hợp con phù hợp bất kỳ. Hơn nữa, mặc dù các dấu hiệu có thể được mô tả trên đây như là hoạt động trong các sự kết hợp đã biết và thậm chí ngay từ đầu được nêu như vậy, một hoặc nhiều dấu hiệu từ tập hợp được yêu cầu bảo hộ có thể trong một số trường hợp bị cắt bỏ từ sự kết hợp này, và tập hợp được yêu cầu bảo hộ có thể hướng đến tập hợp con hoặc biến thể của tập hợp con.

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

Chỉ một vài phương án thực hiện và các ví dụ được mô tả và các phương án thực hiện, các cải tiến và các biến thể khác có thể được thực hiện dựa trên những gì được mô tả và được thể hiện trong phần mô tả này.

YÊU CẦU BẢO HỘ

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

xác định, để biến đổi giữa khối sắc độ của video và dòng bit của video, xem liệu có tách khối sắc độ theo quy tắc liên quan tới kiểu cây của khối sắc độ và kích thước của khối sắc độ hay không; và

thực hiện biến đổi dựa trên bước xác định,

trong đó quy tắc chỉ rõ rằng việc tách khối sắc độ có kiểu cây là DUAL_TREE_CHROMA để tạo ra khối con sắc độ có chiều rộng bằng 2 là không được cho phép;

trong đó quy tắc còn xác định rằng khi khối độ sáng tương ứng với khối sắc độ có 64 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây DUAL_TREE_CHROMA sử dụng tách nhị phân không được phép, và khối độ sáng tương ứng với khối sắc độ được phép cho tách nhị phân.

2. Phương pháp theo điểm 1, trong đó quy tắc chỉ rõ rằng khi kiểu cây của khối sắc độ là bằng DUAL_TREE_CHROMA và chiều rộng của khối sắc độ là bằng 4, khối sắc độ là không được cho phép đối với tách nhị phân dọc.

3. Phương pháp theo điểm 1, trong đó quy tắc chỉ rõ rằng khi kiểu cây của khối sắc độ là bằng DUAL_TREE_CHROMA và chiều rộng của khối sắc độ là bằng 8, khối sắc độ là không được cho phép đối với tách tam phân dọc.

4. Phương pháp theo điểm 1, trong đó khi kiểu chế độ của khối sắc độ là MODE_TYPE_INTRA, quy tắc còn chỉ rõ rằng việc tách khối sắc độ để tạo ra khối con sắc độ có chiều rộng bằng 4 và chiều cao bằng 2 cũng không được cho phép.

5. Phương pháp theo điểm 4, trong đó quy tắc còn chỉ rõ rằng việc tách khối độ sáng tương ứng với khối sắc độ để tạo ra khối con độ sáng là được cho phép.

6. Phương pháp theo điểm 5, trong đó quy tắc còn chỉ rõ rằng khối độ sáng tương ứng với khối sắc độ có 128 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây DUAL_TREE_CHROMA sử dụng tách tam phân không được phép và khối độ sáng được phép cho tách tam phân.

7. Phương pháp theo điểm 1, trong đó chế độ dự báo ngoài ảnh-trong ảnh kết hợp là không được cho phép để được áp dụng cho khối sắc độ với chiều rộng nhỏ hơn 4.

8. Phương pháp theo điểm 7, trong đó chế độ dự báo ngoài ảnh-trong ảnh kết hợp là không được cho phép để được áp dụng cho khối độ sáng với chiều rộng bằng 4 và chiều cao bằng 16.

9. Phương pháp theo điểm 1, trong đó bước biến đổi bao gồm tạo mã khối sắc độ thành dòng بیت.

10. Phương pháp theo điểm 1, trong đó bước biến đổi bao gồm giải mã khối sắc độ từ dòng بیت.

11. Thiết bị xử lý dữ liệu video bao gồm bộ xử lý và bộ nhớ bất biến với các lệnh trong đó, trong đó các lệnh này khi được xử lý bởi bộ xử lý, khiến bộ xử lý:

xác định, để biến đổi giữa khối sắc độ của video và dòng بیت của video, xem liệu có tách khối sắc độ theo quy tắc liên quan tới kiểu cây của khối sắc độ và kích thước của khối sắc độ hay không; và

thực hiện biến đổi dựa trên bước xác định,

trong đó quy tắc chỉ rõ rằng việc tách khối sắc độ có kiểu cây là DUAL_TREE_CHROMA để tạo ra khối con sắc độ có chiều rộng bằng 2 là không được cho phép;

trong đó quy tắc còn xác định rằng khi khối độ sáng tương ứng với khối sắc độ có 64 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây DUAL_TREE_CHROMA sử dụng tách nhị phân không được phép, và khối độ sáng tương ứng với khối sắc độ được phép cho tách nhị phân.

12. Thiết bị theo điểm 11, trong đó quy tắc chỉ rõ rằng khi kiểu cây của khối sắc độ là bằng DUAL_TREE_CHROMA và chiều rộng của khối sắc độ là bằng 4, khối sắc độ là không được cho phép đối với tách nhị phân dọc.

13. Thiết bị theo điểm 11, trong đó quy tắc chỉ rõ rằng khi kiểu cây của khối sắc độ là bằng DUAL_TREE_CHROMA và chiều rộng của khối sắc độ là bằng 8, khối sắc độ là không được cho phép đối với tách tam phân dọc.

14. Thiết bị theo điểm 11, trong đó khi kiểu chế độ của khối sắc độ là `MODE_TYPE_INTRA`, quy tắc còn chỉ rõ rằng việc tách khối sắc độ để tạo ra khối con sắc độ có chiều rộng bằng 4 và chiều cao bằng 2 cũng không được cho phép.

15. Thiết bị theo điểm 14, trong đó quy tắc còn chỉ rõ rằng việc tách khối độ sáng tương ứng với khối sắc độ để tạo ra khối con độ sáng là được cho phép.

16. Thiết bị theo điểm 15, trong đó quy tắc còn chỉ rõ rằng khối độ sáng tương ứng với khối sắc độ có 128 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây `DUAL_TREE_CHROMA` sử dụng tách tam phân không được phép và khối độ sáng được phép cho tách tam phân.

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

xác định, để biến đổi giữa khối sắc độ của video và dòng bit của video, xem liệu có tách khối sắc độ theo quy tắc liên quan tới kiểu cây của khối sắc độ và kích thước của khối sắc độ hay không; và

thực hiện biến đổi dựa trên bước xác định,

trong đó quy tắc chỉ rõ rằng việc tách khối sắc độ có kiểu cây là `DUAL_TREE_CHROMA` để tạo ra khối con sắc độ có chiều rộng bằng 2 là không được cho phép.

trong đó quy tắc còn xác định rằng khi khối độ sáng tương ứng với khối sắc độ has 64 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây `DUAL_TREE_CHROMA` sử dụng tách nhị phân không được phép, và khối độ sáng tương ứng với khối sắc độ được phép cho tách nhị phân.

18. Vật ghi máy tính đọc được bất biến theo điểm 17, trong đó quy tắc xác định rằng khi loại cây của khối sắc độ bằng `DUAL_TREE_CHROMA` và chiều rộng của khối sắc độ bằng 4, khối sắc độ không được phép với phép tách nhị phân dọc.

19. Vật ghi máy tính đọc được bất biến theo điểm 17, trong đó quy tắc xác định rằng khi loại cây của khối sắc độ bằng `DUAL_TREE_CHROMA` và chiều rộng

của khối sắc độ bằng 8, khối sắc độ không được phép với phép tách tam phân dọc.

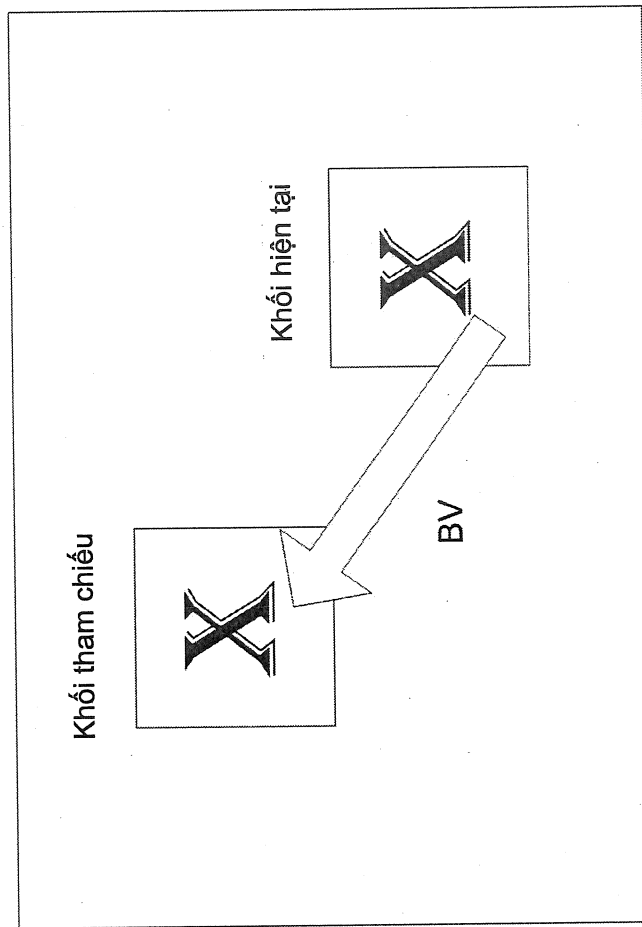
20. Phương pháp lưu trữ dòng bit của video, trong đó phương pháp này bao gồm các bước:

xác định, đối với khối sắc độ của video, xem liệu có tách khối sắc độ theo quy tắc liên quan tới kiểu cây của khối sắc độ và kích thước của khối sắc độ hay không; và

tạo ra dòng bit dựa trên bước xác định, trong đó quy tắc chỉ rõ rằng việc tách khối sắc độ có kiểu cây là DUAL_TREE_CHROMA để tạo ra khối con sắc độ có chiều rộng bằng 2 là không được cho phép;

trong đó quy tắc còn xác định rằng khi khối độ sáng tương ứng với khối sắc độ có 64 mẫu độ sáng, loại lát của khối sắc độ là lát I, và định dạng sắc độ của khối sắc độ bằng 4:2:0, bước tách khối sắc độ có loại cây DUAL_TREE_CHROMA sử dụng tách nhị phân không được phép, và khối độ sáng tương ứng với khối sắc độ được phép cho tách nhị phân.

1/24



Ảnh hiện tại

FIG. 1

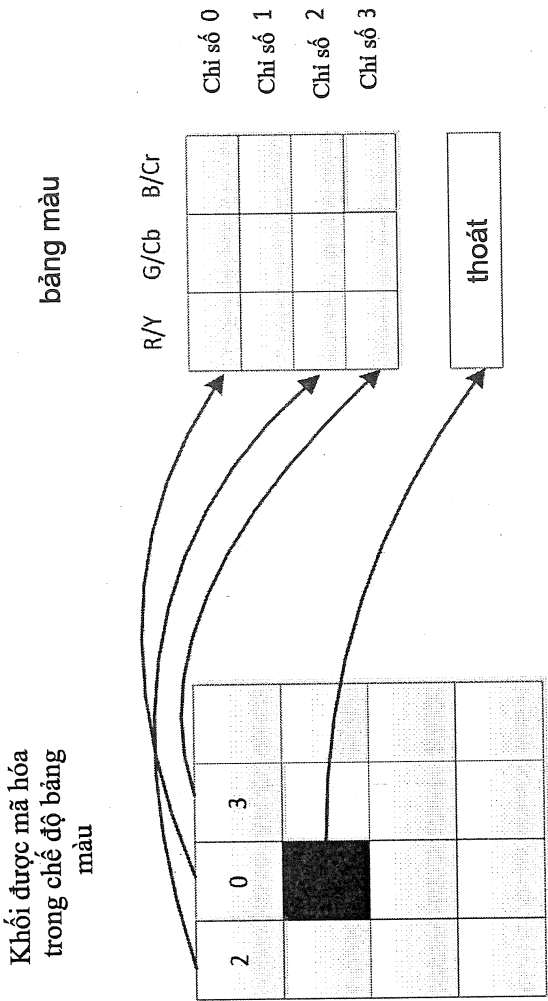


FIG. 2

FIG. 3

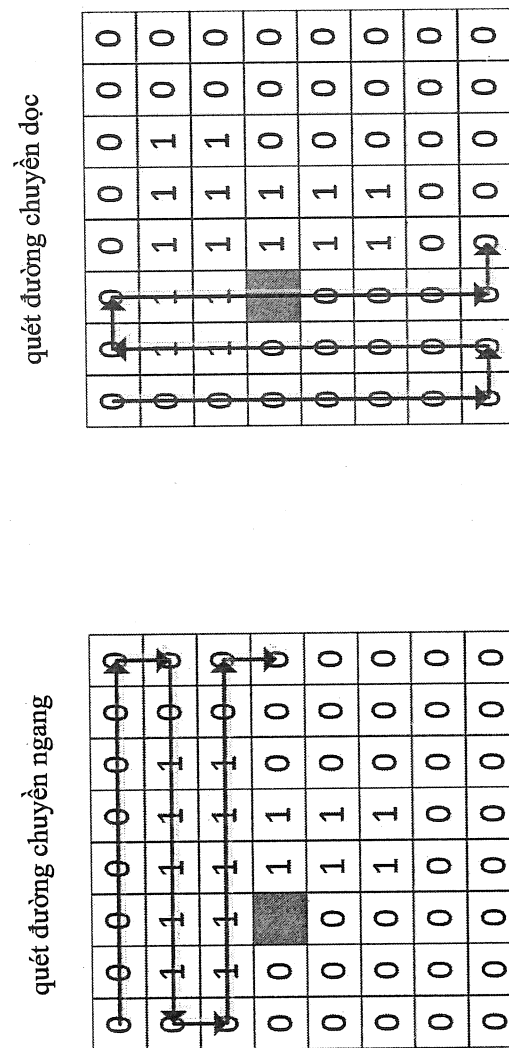
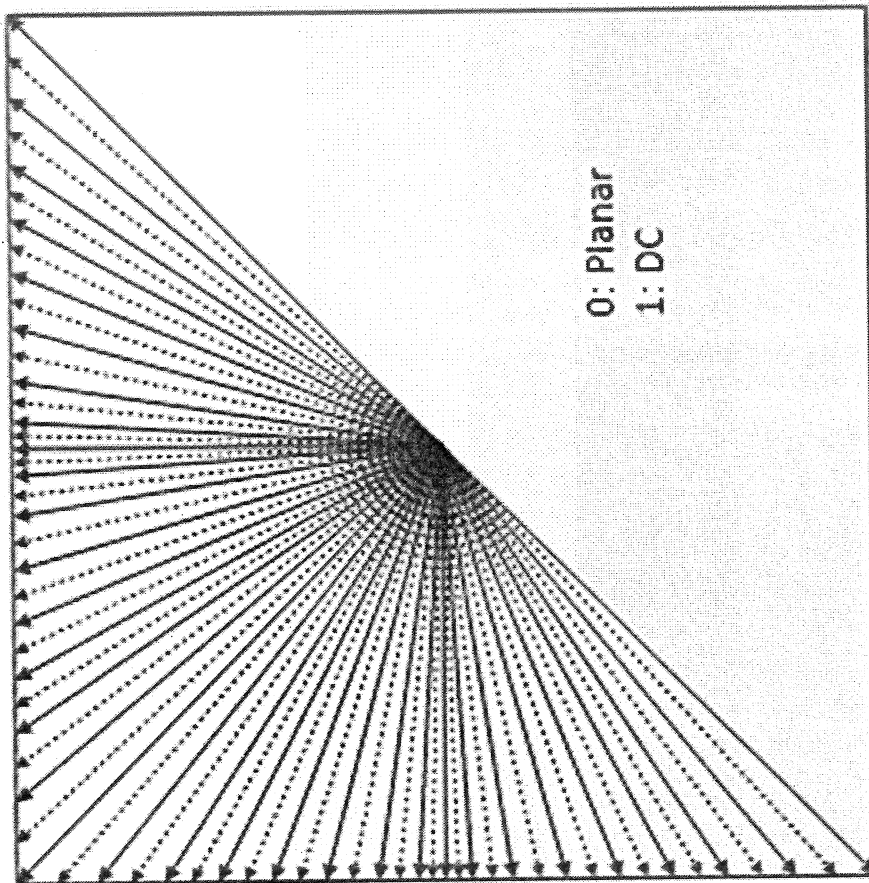


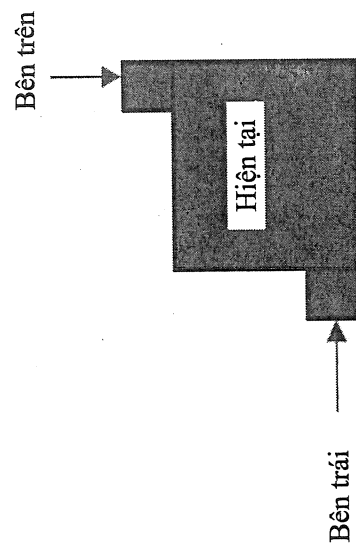


Fig. 5

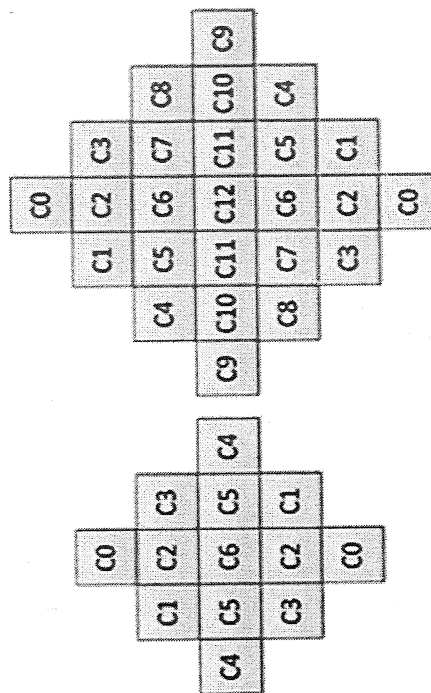
6/24

**FIG. 6**

7/24

**FIG. 7**

8/24

**FIG. 8**

	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	

	D ²		D ²		D ²		D ²		D ²
		D ²		D ²		D ²		D ²	
	D ²		D ²		D ²		D ²		D ²
		D ²		D ²		D ²		D ²	
	D ²		D ²		D ²		D ²		D ²
		D ²		D ²		D ²		D ²	
	D ²		D ²		D ²		D ²		D ²
		D ²		D ²		D ²		D ²	
	D ²		D ²		D ²		D ²		D ²
		D ²		D ²		D ²		D ²	

	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	

$D1$		$D1$		$D1$		$D1$		$D1$
	$D1$		$D1$		$D1$		$D1$	
$D1$		$D1$		$D1$		$D1$		$D1$
	$D1$		$D1$		$D1$		$D1$	
$D1$		$D1$		$D1$		$D1$		$D1$
	$D1$		$D1$		$D1$		$D1$	
$D1$		$D1$		$D1$		$D1$		$D1$
	$D1$		$D1$		$D1$		$D1$	
$D1$		$D1$		$D1$		$D1$		$D1$

9. G. F.

10/24

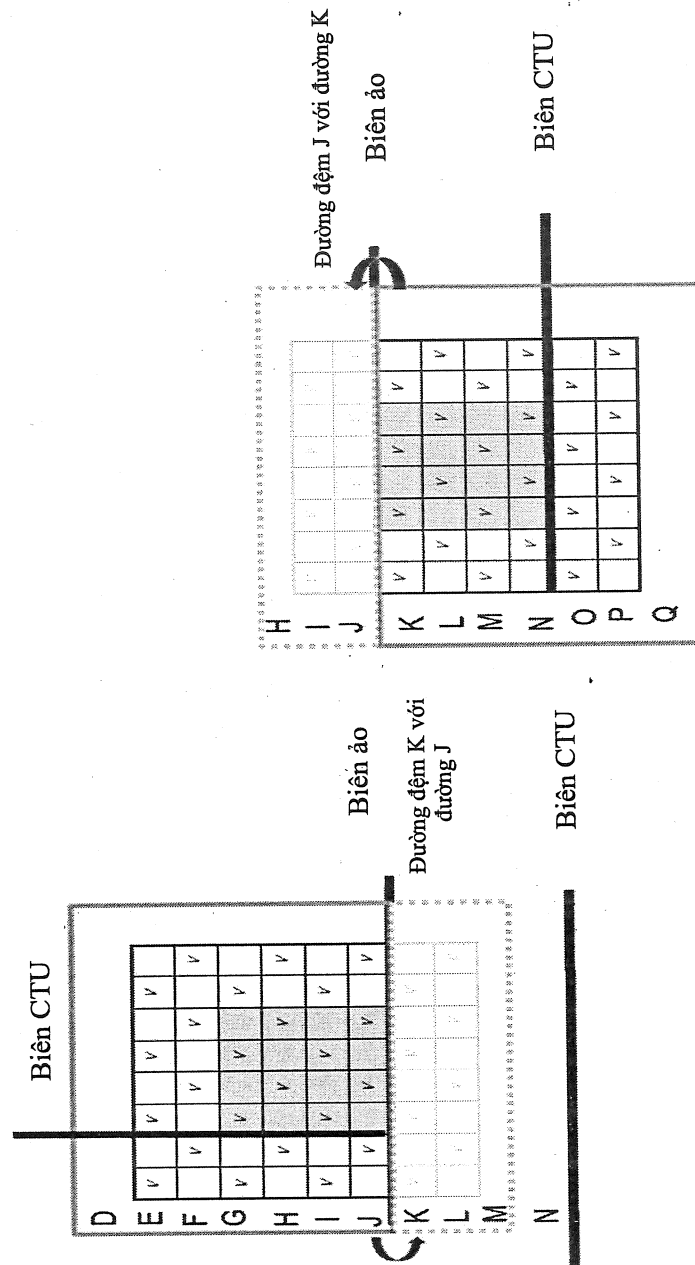


FIG. 10

11/24

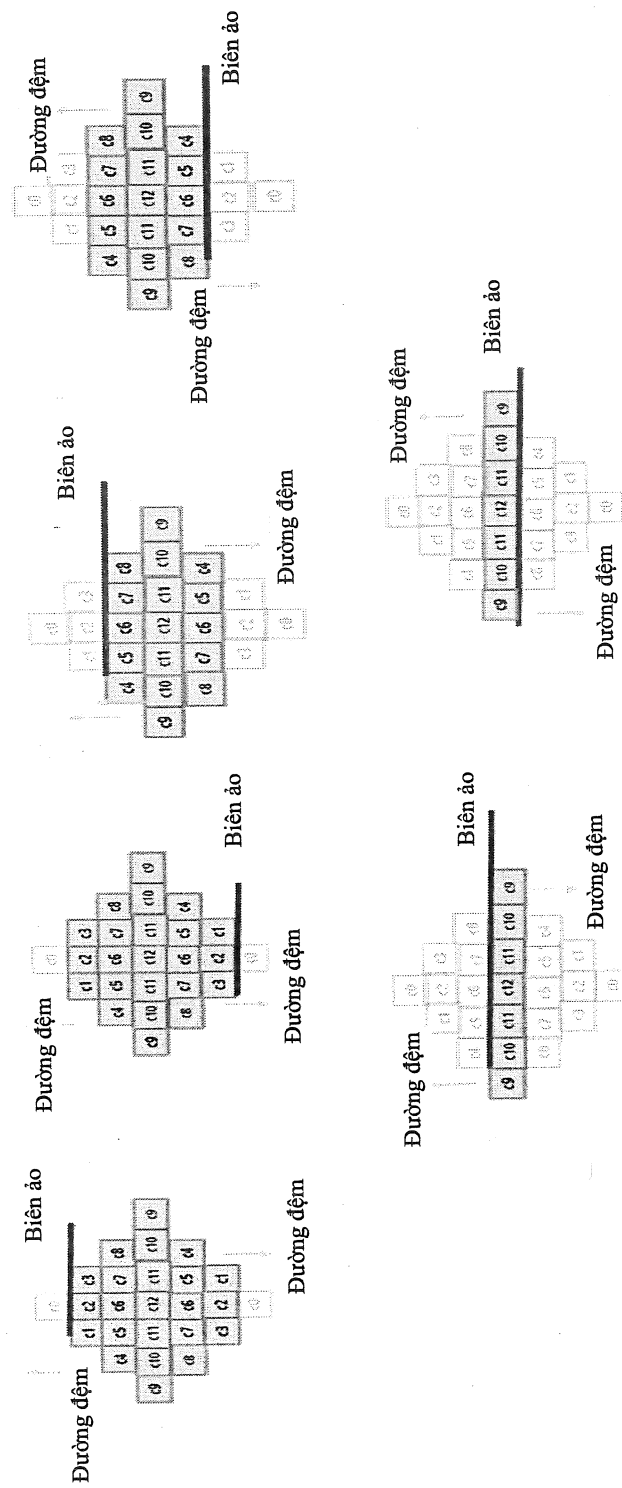


FIG. 11

12/24

	P	

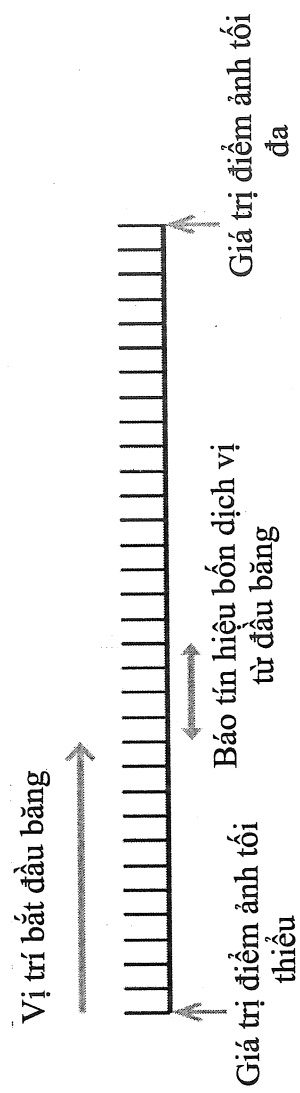
	P	

	P	

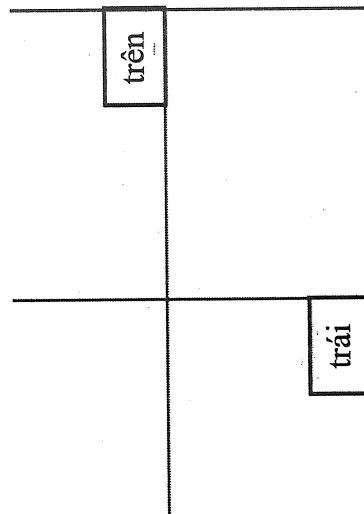
	P	

FIG. 12

13/24

**FIG. 13**

14/24

**FIG. 14**

15/24

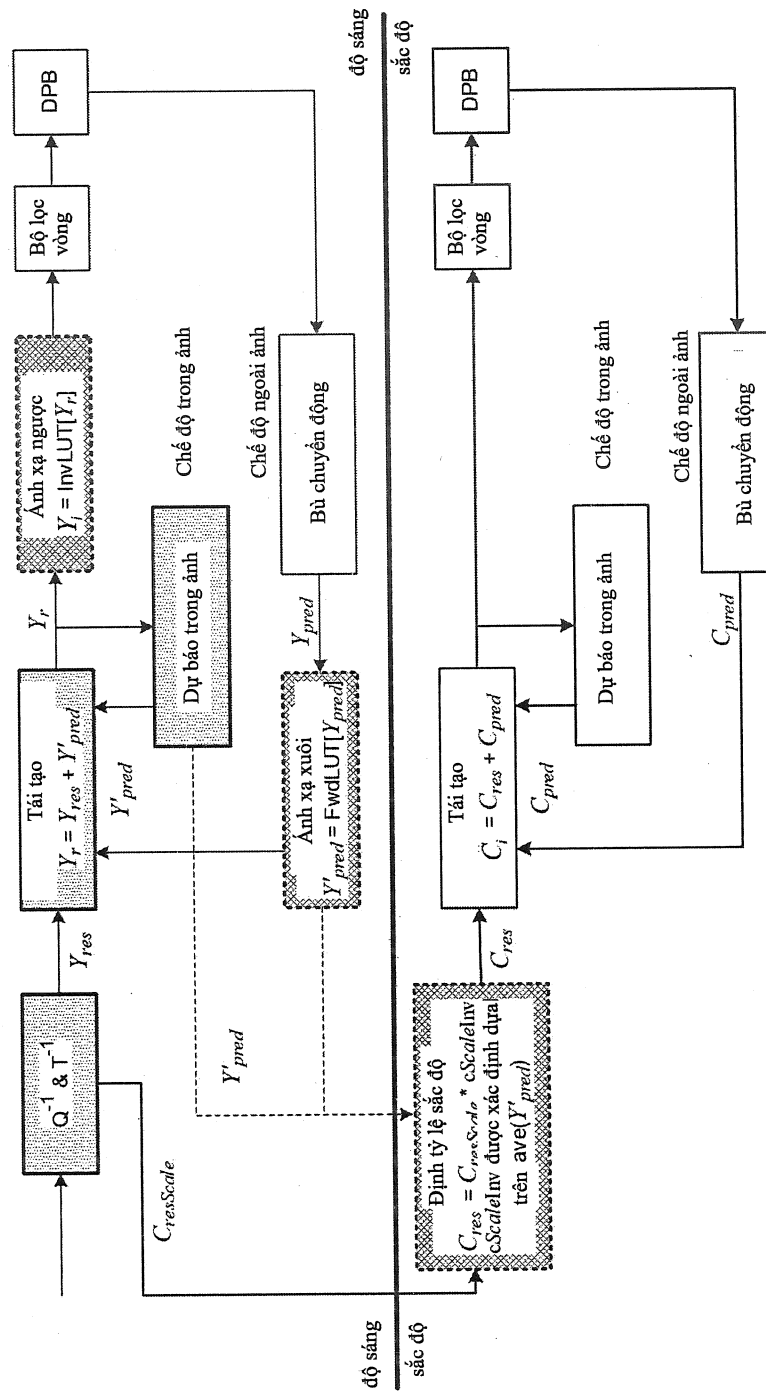


FIG. 15

16/24

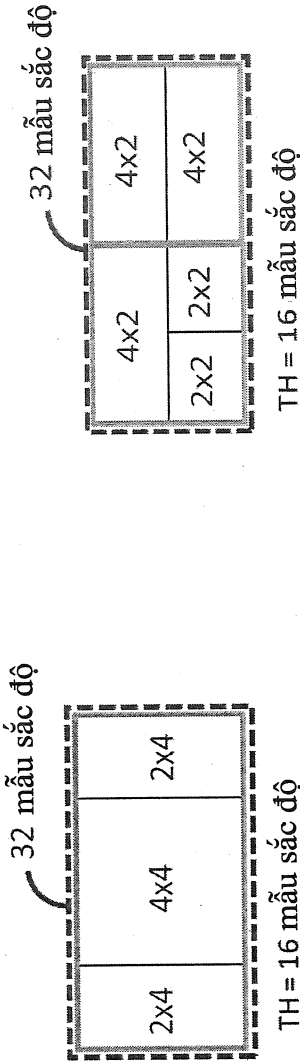
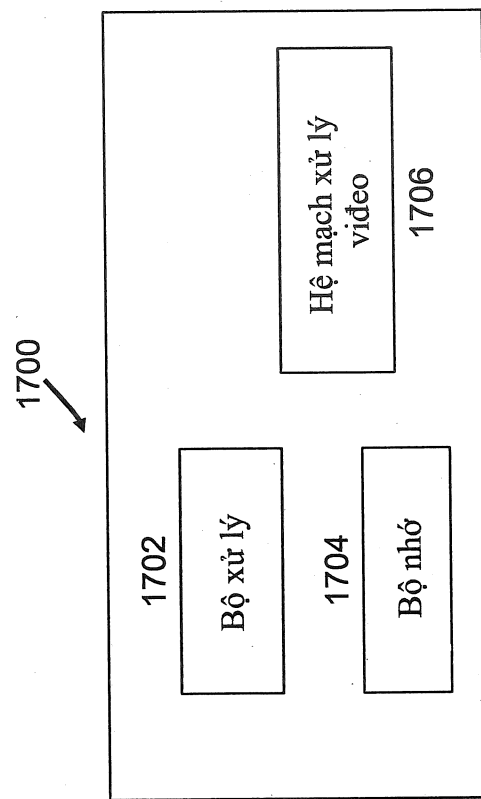
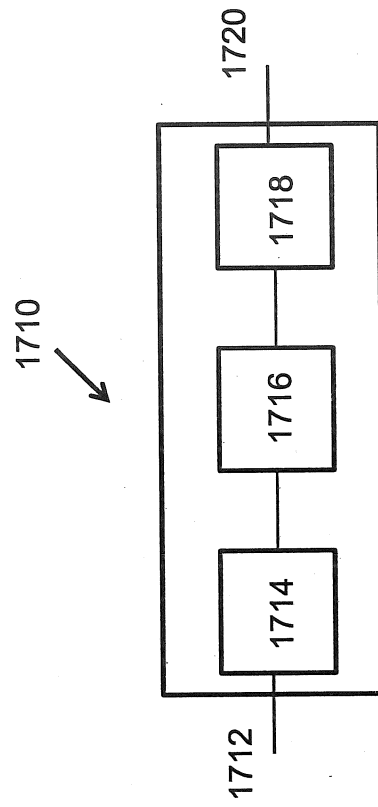


FIG. 16

17/24

**FIG. 17A**

18/24

**FIG. 17B**

19/24

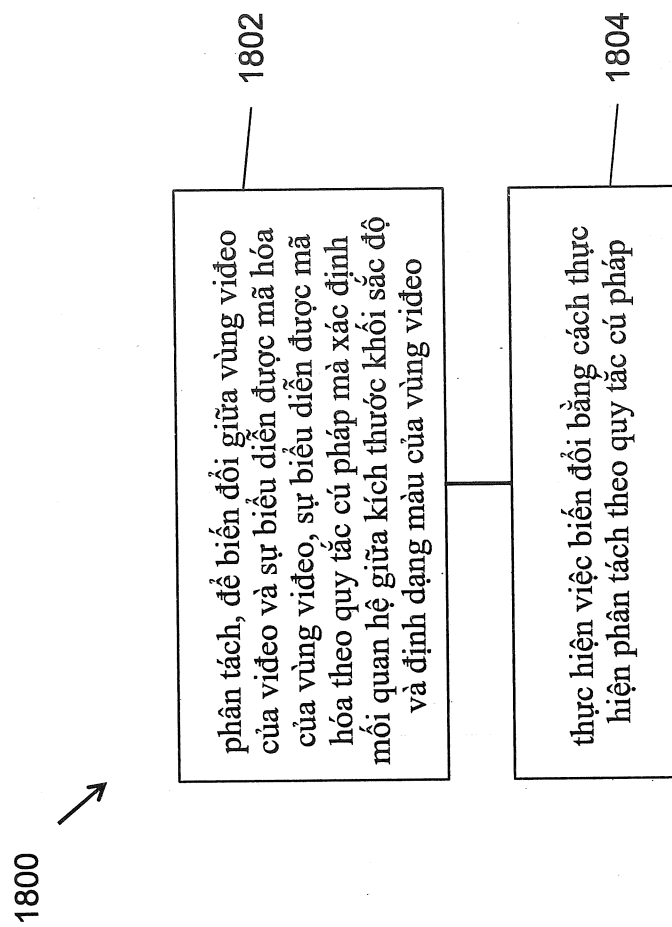


FIG. 18

20/24

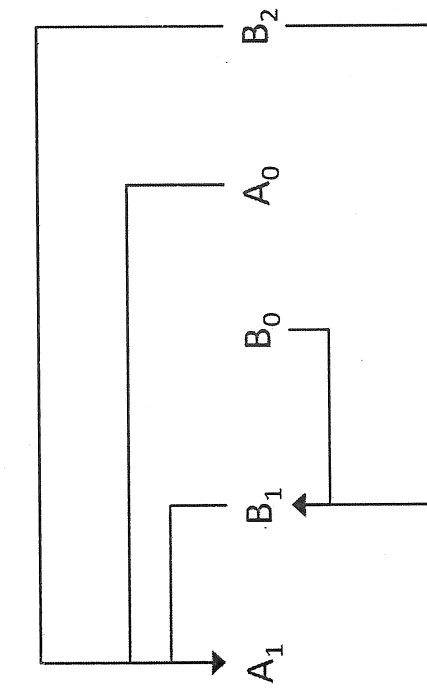


FIG. 20

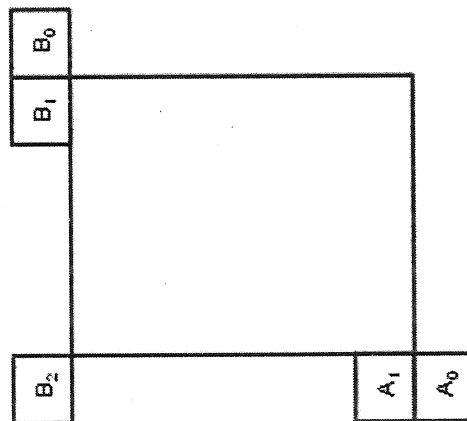
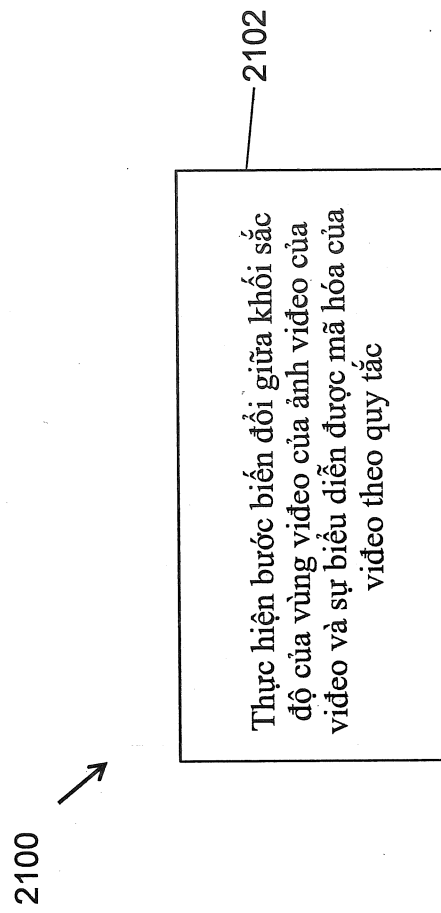


FIG. 19

21/24

**FIG. 21**

22/24

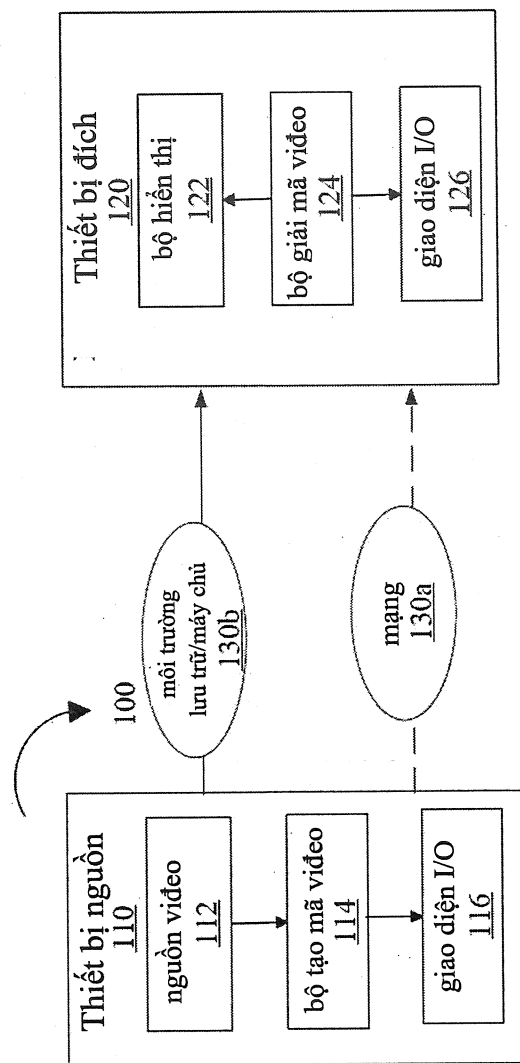


FIG. 22

23/24

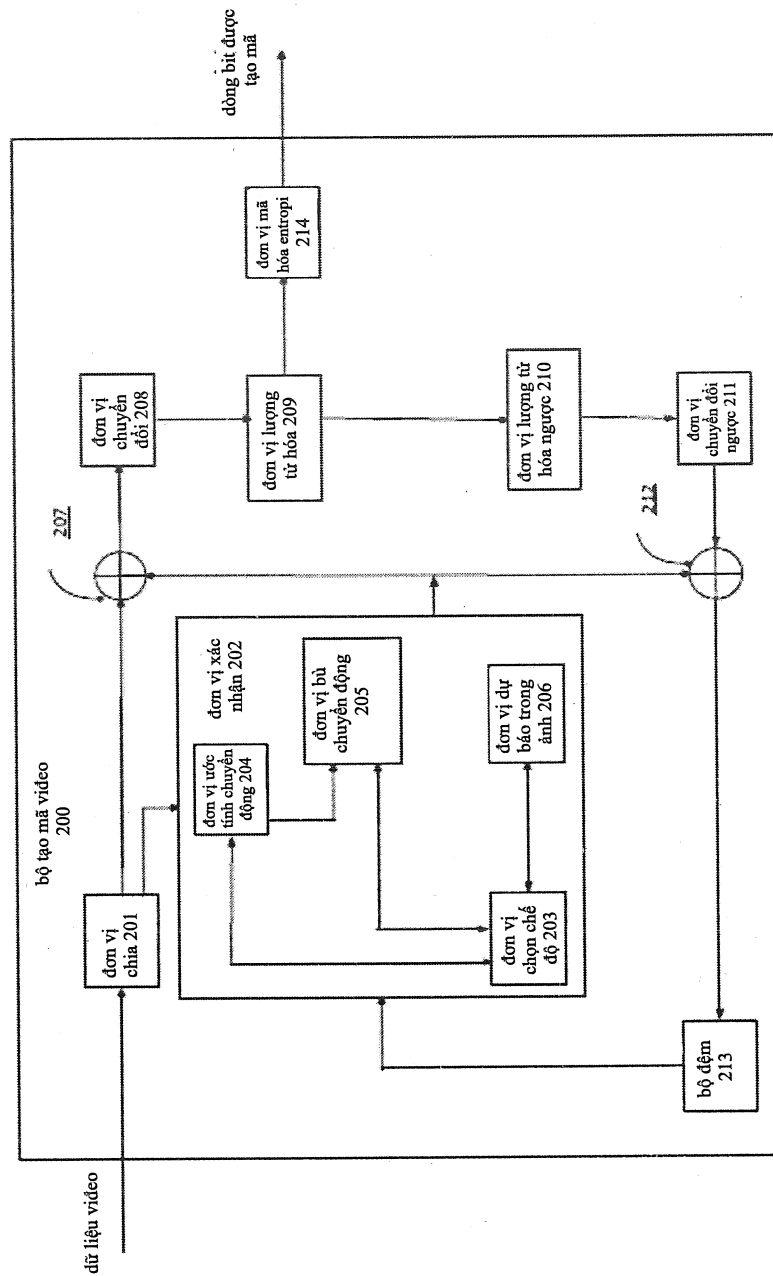


FIG. 23

24/24

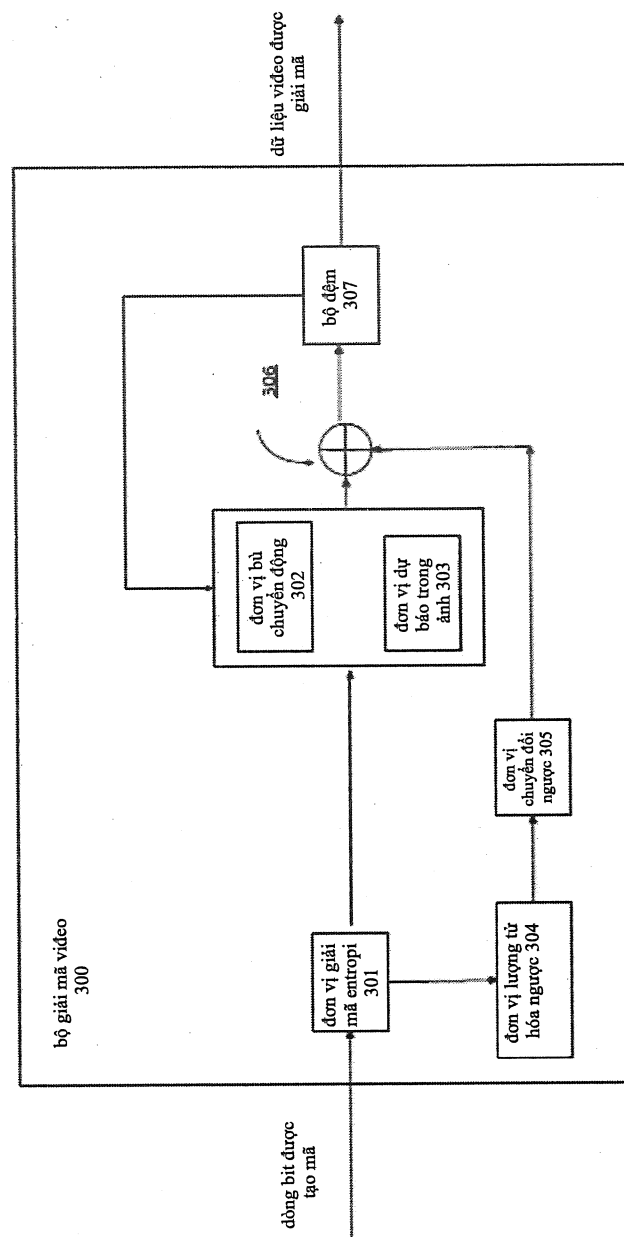


FIG. 24