



(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)



1-0053922

(51)<sup>2020.01</sup> H04N 19/00

(13) B

---

(21) 1-2022-00958 (22) 21/08/2020  
(86) PCT/CN2020/110561 21/08/2020 (87) WO2021/036939 04/03/2021  
(30) 62/890,723 23/08/2019 US; 62/906,435 26/09/2019 US  
(45) 25/11/2025 452 (43) 25/05/2022 410A  
(73) HFI INNOVATION INC. (TW)  
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan  
(72) LIN, Zhi-Yi (TW); CHUANG, Tzu-Der (TW); CHEN, Ching-Yeh (TW).  
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

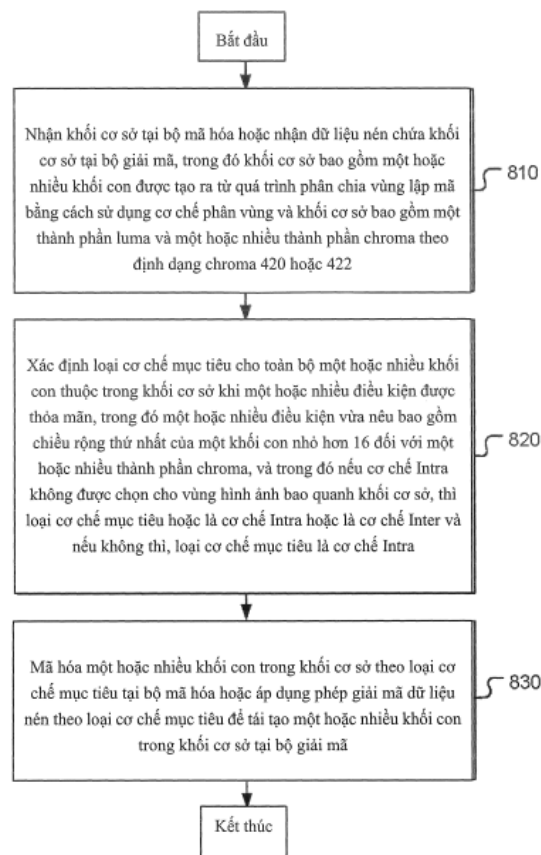
---

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ LẬP MÃ VIDEO

(21) 1-2022-00958

(57) Sáng chế đề cập đến phương pháp và thiết bị lập mã video. Theo một phương pháp của sáng chế, khối cơ sở bao gồm một thành phần luma và một hoặc nhiều thành phần chroma theo theo định dạng chroma 420 hoặc 422 được nhận tại bộ mã hóa hoặc dữ liệu nén chứa khối cơ sở được nhận tại bộ giải mã. Loại cơ chế mục tiêu được xác định cho toàn bộ các khối con thuộc trong khối cơ sở khi một hoặc nhiều điều kiện thỏa mãn. Những điều kiện vừa nêu bao gồm chiều rộng khối con bằng 2 hoặc khối con nhỏ hơn 16 đối với các thành phần màu (chroma). Nếu cơ chế nội ảnh (intra) không được chọn cho vùng hình ảnh bao quanh khối cơ sở, loại cơ chế mục tiêu là cơ chế nội ảnh hoặc liên ảnh (inter) và nếu không thì, loại cơ chế mục tiêu là cơ chế nội ảnh. Các khối con trong khối cơ sở được mã hóa hoặc được giải mã một cách phù hợp.

Fig. 8



## Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc chọn cơ chế dự đoán cho các khối nhỏ khi lập mã video. Cụ thể hơn là, sáng chế đề cập đến phương pháp và thiết bị lập mã video.

## Tình trạng kỹ thuật của sáng chế

Chuẩn lập mã video hiệu suất cao HEVC (High-Efficiency Video Coding - HEVC) được phát triển bởi dự án nghiên cứu video chung giữa các tổ chức tiêu chuẩn gồm nhóm chuyên gia lập mã video VCEG (Video coding Experts Group - VCEG) đến từ Nhóm nghiên cứu thuộc ITU-T và nhóm chuyên gia hình ảnh động MPEG ISO/IEC (Moving Picture Experts Group - MPEG), và có quan hệ với nhau được gọi là nhóm hợp tác chung về lập mã video JCT-VC (Joint Collaborative Team on Video Coding - JCT-VC). Ở HEVC, một slice được phân vùng thành nhiều đơn vị cây lập mã CTU (Coding Tree Unit - CTU). Ở kiến trúc chính, các kích cỡ lớn nhất và nhỏ nhất của CTU được xác định bằng các thành phần cú pháp theo bộ tham số chuỗi SPS (Sequence Parameter Set - SPS). Kích cỡ CTU cho phép có thể là 8x8, 16x16, 32x32, hoặc 64x64. Đối với mỗi slice, các CTU nằm trong slice được xử lý theo lệnh quét mảng.

CTU còn được phân vùng thành nhiều đơn vị lập mã CU (Coding Unit - CU) để thích ứng với nhiều đặc tính cục bộ khác nhau. Cây tứ phân, được gọi là cây lập mã, được sử dụng để phân vùng CTU thành nhiều CU. Lấy kích cỡ CTU là  $M \times M$ , trong đó  $M$  là một trong số các trị số 64, 32, hoặc 16. CTU có thể là CU đơn nhất (tức là, không có sự phân chia) hoặc có thể được phân chia thành bốn đơn vị nhỏ hơn có các kích cỡ bằng nhau (tức là, mỗi chúng có kích cỡ  $M/2 \times M/2$ ), mà tương ứng là các nút của cây lập mã. Nếu các đơn vị là các nút lá của cây lập mã, các đơn vị đó trở thành các CU. Nếu không thì, quá trình phân chia cây tứ phân có thể được lặp lại cho đến khi kích cỡ nút đạt kích cỡ CU cho phép nhỏ nhất như được định rõ trong SPS (bộ tham số chuỗi SPS). Những kết quả phân chia vừa nêu theo cấu trúc đệ quy được định rõ bởi cây lập mã (còn được gọi là cấu trúc cây phân vùng) 120 trên Fig. 1. Phân vùng CTU 110 được thể hiện trên Fig. 1, trong đó các đường nét liền biểu thị các đường biên giới CU. Quyết định liệu có lập mã vùng hình ảnh bằng cách sử dụng dự đoán liên ảnh Inter (theo thời gian) hoặc dự đoán nội ảnh Intra (theo không gian) được thực hiện tại CU. Vì kích cỡ CU nhỏ nhất có thể là 8x8, nên độ chia nhất dùng để chuyển các dạng dự đoán cơ bản khác nhau là 8x8.

Ngoài ra, theo HEVC, mỗi CU có thể được phân vùng thành một hoặc nhiều đơn vị dự đoán (PU). Được liên kết với CU, PU đóng vai trò là khối đại diện cơ bản để chia sẻ thông tin dự đoán. Bên trong mỗi PU, quá trình dự đoán giống nhau được áp dụng và thông tin liên quan được truyền đến bộ giải mã dựa trên PU. CU có thể được phân chia thành một, hai hoặc bốn PU theo dạng phân chia PU. HEVC định nghĩa tám hình dạng dùng để phân chia CU thành PU như được thể hiện trên Fig. 2, bao gồm các dạng phân vùng  $M \times M$ ,  $M \times M/2$ ,  $M/2 \times M$ ,  $M/2 \times M/2$ ,  $M \times M/4$  (U),  $M \times M/4$  (D),  $M/4 \times M$  (L) và  $M/4 \times M$  (R). Không giống CU, PU có thể chỉ được phân chia duy nhất theo HEVC. Những phân vùng được thể hiện ở hàng thứ hai là những phân vùng không đối xứng, trong đó hai phần được phân vùng có kích cỡ khác.

Sau khi thu được khối dư thừa nhờ quá trình dự đoán dựa vào dạng phân chia PU, các khối dư thừa vừa nêu của CU có thể được phân vùng thành các đơn vị biến đổi (TU) theo cấu trúc cây tứ phân khác mà tương tự như cây lập mã dùng cho CU như được thể hiện trên Fig. 1. Các đường nét liền biểu thị các đường biên giới CU và các đường chấm gạch biểu thị các đường biên giới TU. TU là khối đại diện cơ bản có các hệ số dư hoặc hệ số biến đổi để áp dụng phép biến đổi và lượng tử hóa số nguyên. Đối với mỗi TU, một biến đổi số nguyên có cùng kích cỡ với TU được áp dụng để nhận được các hệ số dư. Những hệ số vừa nêu được truyền đến bộ giải mã sau khi lượng tử hóa dựa vào TU.

Các thuật ngữ khối cây lập mã CTB (Coding Tree Block - CTB), khối lập mã CB (Coding Block - CB), khối dự đoán PB (Prediction Block - PB), và khối biến đổi TB (Transform Block - TB) được định nghĩa để định rõ mảng mẫu hai chiều có một thành phần màu liên quan đến CTU, CU, PU, và TU, một cách tương ứng. Do đó, CTU bao gồm một CTB luma, hai CTB chroma, và các thành phần cú pháp liên quan. Mỗi quan hệ tương tự cũng có đối với CU, PU, và TU. Phân vùng cây thường được áp dụng đồng thời cho cả luma và chroma, tuy nhiên những phân khác được áp dụng khi các kích cỡ nhỏ nhất nhất định đạt được đối với.

Ngoài ra, cấu trúc phân vùng cây nhị phân được đề xuất. Trong quá trình phân vùng được bộc lộ, khối có thể được phân chia đệ quy thành 2 khối nhỏ hơn. Fig. 3 thể hiện nhiều dạng phân chia khác nhau. Phân chia ngang và đứng đối xứng là những dạng phân chia đơn giản và hiệu quả nhất (tức là,  $M/2 \times M$  và  $M \times M/2$ ), được thể hiện ở hai dạng phân chia phía trên cùng trên Fig. 3. Do vậy, theo một phương án, chúng tôi chỉ sử dụng hai dạng phân chia vừa nêu.

Đối với khối cho trước có kích cỡ  $M \times N$ , cờ được báo hiệu để biểu thị liệu nó được phân chia thành hai khối nhỏ hơn. Nếu cờ vừa nêu biểu thị dạng phân chia, thì thành phần cú pháp khác được báo hiệu để cho biết dạng phân chia nào được sử dụng. Nếu phân chia đứng được sử dụng thì nó được phân chia thành hai khối có kích cỡ  $M \times N/2$ ; nếu không thì nó được phân chia thành hai khối có kích cỡ  $M/2 \times N$ . Quá trình phân chia cây nhị phân có thể được lặp lại cho đến khi kích cỡ (chiều rộng hoặc chiều cao) của khối đạt kích cỡ khối (chiều rộng hoặc chiều cao) cho phép nhỏ nhất được định theo cú pháp cấp cao chẳng hạn SPS. Vì cây nhị phân có hai dạng phân chia (tức là, ngang và đứng), nên cả chiều rộng và chiều cao khối cho phép nhỏ nhất cần được định rõ. Phân chia đứng bị cấm hoàn toàn khi các kết quả phân chia theo chiều cao khối nhỏ hơn chiều cao nhỏ nhất đã định nghĩa. Tương tự, phân chia ngang bị cấm hoàn toàn khi các kết quả phân chia theo chiều rộng khối nhỏ hơn chiều rộng nhỏ nhất đã định nghĩa.

Fig. 4 minh họa ví dụ về phân vùng khối 410 và cây nhị phân tương ứng của nó 420. Ở mỗi nút phân chia (tức là, nút không lá) của cây nhị phân, một cờ được sử dụng để cho biết dạng phân chia nào (phân chia ngang hay phân chia đứng) được sử dụng, trong đó 0 có thể biểu thị phân chia đứng và 1 có thể biểu thị phân chia ngang.

Cấu trúc cây nhị phân có thể được sử dụng để phân chia vùng hình ảnh thành nhiều khối nhỏ hơn chẳng hạn phân chia slice thành các CTU, CTU thành các CU, CU thành các PU, hoặc CU thành các TU, và .v.v. Cây nhị phân có thể được sử dụng để phân chia CTU thành các CU, trong đó nút gốc của cây nhị phân là CTU và nút lá của cây nhị phân là CU. Các nút lá có thể được xử lý tiếp bằng phép lập mã dự đoán và biến đổi. Nhằm đơn giản hóa, nên sẽ không có sự phân vùng từ CU thành PU hoặc từ CU thành TU. Điều đó có nghĩa là CU bằng PU và PU bằng TU. Vì vậy, nói cách khác, nút lá của cấu trúc QTBT là đơn vị cơ bản để dự đoán và biến đổi.

#### Cấu trúc QTBT

Cấu trúc cây nhị phân linh hoạt hơn cấu trúc cây tứ phân vì nhiều dạng phân vùng hơn có thể được hỗ trợ, và cũng là cấu trúc giúp tăng hiệu quả lập mã. Tuy nhiên, mức độ phức tạp mã hóa cũng sẽ tăng lên để lựa chọn dạng phân vùng tốt nhất. Để cân bằng giữa độ phức tạp và hiệu quả lập mã, một phương pháp kết hợp cấu trúc cây nhị phân và cây tứ phân, còn được gọi là cấu trúc cây tứ phân cộng cây nhị phân (QTBT), đã được đề xuất. Theo cấu trúc QTBT, CTU (hoặc CTB cho I slice) là nút gốc của cây tứ phân và

CTU trước tiên được phân vùng bởi cây tứ phân, trong đó sự phân chia cây tứ phân của một nút có thể được lặp lại cho đến khi nút đó đạt kích cỡ nút lá cây tứ phân cho phép nhỏ nhất (tức là, MinQTSIZE). Nếu kích cỡ nút lá cây tứ phân không lớn hơn kích cỡ nút gốc cây nhị phân cho phép lớn nhất (tức là, MaxBTSIZE), nó có thể còn được phân vùng bởi cây nhị phân. Phân chia cây nhị phân của một nút có thể được lặp lại cho đến khi nút đó đạt kích cỡ nút lá cây nhị phân cho phép nhỏ nhất (tức là, MinBTSIZE) hoặc chiều cao cây nhị phân cho phép lớn nhất (tức là, MaxBTDepth). Nút lá cây nhị phân, cụ thể là CU (hoặc CB cho I slice), sẽ được sử dụng để dự đoán (ví dụ dự đoán nội ảnh Intra hoặc dự đoán liên ảnh Inter) và biến đổi mà không có bất kỳ phân vùng nào nữa. Có hai dạng phân chia trong phân chia cây nhị phân: phân chia ngang đối xứng và phân chia đứng đối xứng. Theo cấu trúc QTBT, kích cỡ nút lá cây tứ phân cho phép nhỏ nhất, kích cỡ nút gốc cây nhị phân cho phép nhỏ nhất, chiều rộng và chiều cao nút lá cây nhị phân cho phép nhỏ nhất, và chiều cao cây nhị phân cho phép lớn nhất có thể được biểu thị theo cú pháp cấp cao chẳng hạn trong SPS. Fig. 5 minh họa ví dụ phân vùng khối 510 và cấu trúc QTBT 520 tương ứng của nó. Các đường nét liền biểu thị phân chia cây tứ phân và các đường chấm gạch biểu thị phân chia cây nhị phân. Ở mỗi nút phân chia (tức là, nút không lá) của cây nhị phân, một cờ biểu thị dạng phân chia nào (ngang hay đứng) được sử dụng, 0 có thể biểu thị phân chia ngang và 1 có thể biểu thị phân chia đứng.

Cấu trúc QTBT nêu trên có thể được sử dụng để phân chia vùng hình ảnh (ví dụ slice, CTU hoặc CU) thành nhiều khối nhỏ hơn chẳng hạn phân chia slice thành các CTU, CTU thành các CU, CU thành các PU, hoặc CU thành các TU, và .v.v. Ví dụ, QTBT có thể được sử dụng để phân vùng CTU thành các CU, trong đó nút gốc của QTBT là CTU được phân vùng thành nhiều CU bằng cấu trúc QTBT và các CU được xử lý tiếp bằng phép lập mã dự đoán và biến đổi. Nhằm đơn giản hóa, nên không có sự phân vùng từ CU thành PU hoặc từ CU thành TU. Điều đó có nghĩa là CU bằng PU và PU bằng TU. Vì vậy, nói cách khác, nút lá của cấu trúc QTBT là đơn vị cơ bản để dự đoán và biến đổi.

Ví dụ về cấu trúc QTBT được thể hiện sau đây. Đối với CTU có kích cỡ 128x128, kích cỡ nút lá cây tứ phân cho phép nhỏ nhất được thiết lập bằng 16x16, kích cỡ nút gốc cây nhị phân cho phép nhỏ nhất được thiết lập bằng 64x64, chiều rộng và chiều cao nút lá cây nhị phân cho phép nhỏ nhất đều được thiết lập bằng 4, và chiều cao cây nhị phân cho phép lớn nhất được thiết lập bằng 4. Trước tiên, CTU được phân vùng bằng cấu trúc

cây tứ phân và đơn vị cây tứ phân lá có thể có kích cỡ từ  $16 \times 16$  (tức là, kích cỡ nút lá cây tứ phân cho phép nhỏ nhất) đến  $128 \times 128$  (bằng kích cỡ CTU, tức là, không phân chia). Nếu đơn vị cây tứ phân lá là  $128 \times 128$ , nó không thể được phân chia tiếp bằng cây nhị phân vì kích cỡ đó vượt quá kích cỡ nút gốc cây nhị phân cho phép nhỏ nhất  $64 \times 64$ . Nếu không thì, đơn vị cây tứ phân lá có thể được phân chia tiếp bằng cây nhị phân. Đơn vị cây tứ phân lá, mà cũng là đơn vị cây nhị phân gốc, có chiều cao cây nhị phân là 0. Khi chiều cao cây nhị phân bằng 4 (tức là, cây nhị phân cho phép lớn nhất như được biểu thị), không có sự phân chia được thực hiện hoàn toàn. Khi khối của nút cây nhị phân tương ứng có chiều rộng bằng 4, phân chia không nằm ngang được thực hiện hoàn toàn. Khi khối của nút cây nhị phân tương ứng có chiều cao bằng 4, phân chia không thẳng đứng được thực hiện hoàn toàn. Các nút lá của QTBT được xử lý tiếp bởi dự đoán (nội ảnh Intra hoặc liên ảnh Inter) và lập mã biến đổi.

Đối với I slice, cấu trúc cây QTBT thường được áp dụng với lập mã riêng biệt luma/chroma. Ví dụ, cấu trúc cây QTBT được áp dụng riêng đối với các thành phần luma và chroma cho I slice, và được áp dụng cho cả luma và chroma (trừ khi các kích cỡ nhỏ nhất nhất định đạt được đối với chroma) đối với các P slice và B slice. Nói cách khác, ở I slice, CTB luma có phân vùng khối theo cấu trúc QTBT của nó và hai CTB chroma có phân vùng khối theo cấu trúc QTBT khác. Ở ví dụ khác, hai CTB chroma có thể còn có các phân vùng khối theo cấu trúc QTBT riêng của chúng.

### **Bản chất kỹ thuật của sáng chế**

Phương pháp và thiết bị mã hóa và giải mã video được bộc lộ. Theo một phương pháp của sáng chế, khối cơ sở bao gồm một thành phần luma và một hoặc nhiều thành phần chroma theo định dạng chroma 420 hoặc 422 được nhận tại bộ mã hóa hoặc dữ liệu nén chứa khối cơ sở tại bộ giải mã, trong đó khối cơ sở bao gồm một hoặc nhiều khối con được tạo ra từ quá trình phân chia vùng lập mã bằng cách sử dụng cây phân vùng đơn. Loại cơ chế mục tiêu được xác định cho toàn bộ một hoặc nhiều khối con thuộc trong khối cơ sở khi một hoặc nhiều điều kiện thỏa mãn, trong đó một hoặc nhiều điều kiện vừa nêu bao gồm chiều rộng thứ nhất của một khối con bằng 2 đối với một hoặc nhiều thành phần chroma, và trong đó nếu cơ chế Intra không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế Inter và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra. Một hoặc nhiều khối con vừa nêu trong khối cơ sở được mã hóa tại bộ mã hóa hoặc được giải mã tại bộ giải mã theo

loại cơ chế mục tiêu.

Theo một phương án, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng nhị phân. Theo phương án khác, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai của khối cơ sở cho một thành phần luma vừa nêu bằng 16 và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng tam phân.

Theo một phương án, khi khối cơ sở có định dạng chroma 420, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và 16 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng nhị phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 16 và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng tam phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 32 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng ngang nhị phân hoặc tam phân, hoặc chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 64 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân chia ngang tam phân.

Theo một phương án, khi khối cơ sở có định dạng chroma 422, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng nhị phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 16 và 8 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng đứng tam phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 16 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng ngang nhị phân hoặc tam phân, hoặc chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 32 một cách tương ứng và cây phân vùng đơn cho khối cơ sở bao gồm phân chia ngang tam phân.

Theo một phương án, nếu cơ chế Intra không được chọn cho vùng hình ảnh bao

quanh khối cơ sở, cú pháp thứ nhất được báo hiệu tại bộ mã hóa hoặc được phân tích tại bộ giải mã để báo hiệu hoặc cơ chế Intra hoặc cơ chế Inter được chọn cho khối cơ sở.

Vùng lập mã có thể là đơn vị cây lập mã. Vùng hình ảnh có thể là slice. Trong trường hợp này, cú pháp thứ hai ở tiêu đề slice được báo hiệu tại bộ mã hóa hoặc được phân tích tại bộ giải mã để biểu thị dạng lập mã của slice.

Phương pháp và thiết bị khác dùng để ràng buộc loại cơ chế đối với các khối nhỏ bằng cây phân vùng đơn cũng được bộc lộ. Theo phương pháp này, loại cơ chế mục tiêu cho tất cả các khối con thuộc trong khối cơ sở được xác định khi một hoặc nhiều điều kiện thỏa mãn. Những điều kiện này bao gồm kích cỡ khối thứ nhất của một khối con nhỏ hơn 16 đối với các thành phần chroma. Nếu cơ chế Intra không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế Inter và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra.

Theo một phương án, kích cỡ khối thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 4. Ví dụ, điều kiện này có thể tương ứng với khối cơ sở theo định dạng chroma 420 có kích cỡ khối bằng 64 và cây phân vùng đơn cho khối cơ sở bao gồm phân vùng cây tứ phân, phân vùng đứng tam phân, phân vùng ngang tam phân, phân vùng đứng nhị phân, hoặc phân vùng đứng nhị phân.

Theo phương án khác, kích cỡ khối thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 8.

### **Mô tả vắn tắt các hình vẽ**

Fig. 1 minh họa ví dụ phân vùng khối bằng cách sử dụng cấu trúc cây tứ phân để phân vùng đơn vị cây lập mã (CTU) thành các đơn vị lập mã (các CU);

Fig. 2 minh họa phân vùng chuyển động không đối xứng AMP (Asymmetric Motion Partition - AMP) theo chuẩn lập mã video hiệu suất cao HEVC (High Efficiency Video Coding - HEVC), trong đó AMP định nghĩa tám hình dạng dùng để phân chia CU thành PU;

Fig. 3 minh họa ví dụ về nhiều dạng phân chia nhị phân khác nhau được sử dụng bởi cấu trúc phân vùng cây nhị phân, trong đó khối có thể được phân chia đệ quy thành hai khối nhỏ hơn bằng cách sử dụng các dạng phân chia;

Fig. 4 minh họa ví dụ về phân vùng khối và cây nhị phân tương ứng của nó, trong

đó ở mỗi nút phân chia (tức là, nút không lá) của cây nhị phân, một cú pháp được sử dụng để cho biết dạng phân chia nào (phân chia ngang hay phân chia đứng) được sử dụng, trong đó 0 có thể biểu thị phân chia đứng và 1 có thể biểu thị phân chia ngang;

Fig. 5 minh họa ví dụ về phân vùng khối và QTBT tương ứng của nó, trong đó các đường nét liền biểu thị phân chia cây tứ phân và các đường chấm gạch biểu thị phân chia cây nhị phân;

Fig. 6 minh họa ví dụ về các vị trí của khối luma tham chiếu, mà có thể là khối luma thứ nhất trong khối cơ sở, khối luma cuối cùng trong khối cơ sở, khối luma bao lấy vị trí trung tâm (được biểu thị là BR-C) của khối cơ sở;

Fig. 7 minh họa lưu đồ của hệ thống lập mã được lấy làm ví dụ có phép chọn cơ chế ràng buộc cho khối nhỏ theo phương án của sáng chế; và

Fig. 8 minh họa lưu đồ khác của hệ thống lập mã được lấy làm ví dụ có phép chọn cơ chế ràng buộc cho khối nhỏ theo phương án của sáng chế.

### **Mô tả chi tiết sáng chế**

Phần mô tả dưới đây là phần mô tả cách thức tốt nhất thực hiện sáng chế. Phần mô tả này được đưa ra nhằm mục đích minh họa các nguyên lý chung của sáng chế chứ không nhằm giới hạn sáng chế. Phạm vi của sáng chế được xác định đúng nhất bằng cách tham chiếu các điểm yêu cầu bảo hộ đính kèm.

Khi dự đoán Intra, các mẫu tái tạo liền kề được sử dụng làm các mẫu tham chiếu cho khối hiện thời. Nếu kích cỡ khối quá nhỏ (ví dụ, CU chroma 2x2), thì công suất xử lý bị giới hạn vì bộ xử lý chỉ có thể dự đoán nhiều nhất tại các mẫu của khối hiện thời và dừng lại. Nếu có nhiều khối nhỏ, thì công suất xử lý không thể được cải thiện. Theo đó, theo một phương án của sáng chế, đã đề xuất việc chia sẻ chung các mẫu tham chiếu giống nhau trong một vùng để cải thiện xử lý song song.

Ở phương pháp được đề xuất dưới đây, khối cơ sở/CU cơ sở chia sẻ chung được định nghĩa. Vị trí trên cùng-bên trái của khối cơ sở là `sharedRootPos`, chiều rộng là `sharedRootWidth`, và chiều cao là `sharedRootHeight`. Phép suy diễn khối cơ sở là như sau.

Theo một phương án, trong quá trình phân vùng, nếu khối hiện thời có thể thỏa mãn một hoặc nhiều điều kiện định trước, và khối cơ sở chưa được thiết lập, thì vị trí của

khối cơ sở được thiết lập là vị trí trên cùng-bên trái của khối hiện thời, và chiều rộng và chiều cao của khối cơ sở bằng chiều rộng và chiều cao của khối hiện thời, nếu không thì, khối cơ sở không được thiết lập.

Theo một phương án, trong quá trình phân vùng, nếu một trong số các khối con có thể thỏa mãn một hoặc nhiều điều kiện định trước, và khối cơ sở chưa được thiết lập, thì vị trí trên cùng-bên trái của khối cơ sở được thiết lập là vị trí trên cùng-bên trái của khối hiện thời, và chiều rộng và chiều cao của khối cơ sở bằng chiều rộng và chiều cao của khối hiện thời; nếu không thì, khối cơ sở không được thiết lập.

Điều kiện định trước đã nêu ở trên có thể liên quan đến kích cỡ (hoặc diện tích) của khối hiện thời. Theo một phương án, điều kiện định trước là kích cỡ nhỏ hơn hoặc bằng (hoặc nhỏ hơn) giá trị ngưỡng định trước TH. Giá trị ngưỡng có thể là 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, và 4096.

Nếu khối cơ sở tồn tại và cơ chế Intra được sử dụng thuộc trong khối cơ sở, thì các mẫu tham chiếu được chia sẻ theo cơ chế dự đoán Intra, các mẫu tham chiếu cho cơ chế Intra, cơ chế LM, và/hoặc cơ chế CIIP thông thường được suy dẫn theo khối hiện thời và khối cơ sở; nếu không thì, các mẫu tham chiếu được suy dẫn theo cách thức nguyên thủy.

Theo một phương án, khối cơ sở không được vượt quá biên giới khung hình.

Theo một phương án, vị trí của khối cơ sở chung cũng sẽ được sử dụng để suy dẫn MPM.

Theo một phương án, khối cơ sở chung tồn tại cho cả thành phần luma và chroma.

Theo một phương án, khối cơ sở chung chỉ tồn tại cho thành phần luma.

Theo một phương án, khối cơ sở chung tồn tại cho thành phần chroma.

Theo một phương án, đối với một chuỗi tile/slice/picture/tile group/video, có một cờ để báo hiệu liệu chuỗi tile/slice/picture/tile group/video hiện thời đó có cần để xác định khối cơ sở không. Theo một phương án, ở những chuỗi có độ phân giải nhỏ hơn 1080P, cờ cấp cao này được thiết lập là false.

Khi cơ chế sao chép khối nội ảnh IBC (Intra Block Copy - IBC) được sử dụng:

- a. Theo một phương án, ở cả cây chung và cây riêng, khối tham chiếu được

suy dẫn từ vectơ khối BV (Block Vector - BV) của khối IBC không thể chồng lên khối cơ sở;

- b. Theo phương án khác, chỉ ở cây chung, khối tham chiếu được suy dẫn từ vectơ khối BV (Block Vector - BV) của khối IBC không thể chồng lên khối cơ sở; và
- c. Theo phương án khác, chỉ ở cây riêng, khối tham chiếu được suy dẫn từ vectơ khối BV (Block Vector - BV) của khối IBC không thể chồng lên khối cơ sở.

Nếu BV chroma (hoặc cả BV luma và chroma) được suy dẫn làm cho khối chroma (hoặc cả khối luma và chroma) tham chiếu chồng lên khối cơ sở chroma (hoặc cả khối cơ sở luma và chroma) tương ứng:

Theo một phương án, ở cây lưỡng, khối chroma (hoặc cả khối luma và chroma) không thể sử dụng cơ chế IBC.

Theo phương án khác, ở cây lưỡng hoặc cây chung hoặc cả cây lưỡng và cây chung, BV chroma (hoặc cả BV luma và chroma) được suy dẫn sẽ được mở rộng theo phương ngang/đứng hoặc cả phương ngang và phương đứng để làm cho khối chroma tham chiếu không chồng lên khối cơ sở chroma (hoặc cả khối cơ sở luma và chroma) tương ứng.

Theo một phương án, phương mở rộng phụ thuộc vào chiều rộng và chiều cao của vùng chồng lấn, nếu chiều rộng lớn hơn hoặc bằng chiều cao, thì BV sẽ được mở rộng theo phương đứng; nếu không thì, BV sẽ được mở rộng theo phương ngang.

Theo phương án khác, ở cây lưỡng hoặc cây chung hoặc cả cây lưỡng và cây chung, các mẫu của khối chroma (hoặc cả khối luma và chroma) tham chiếu mà chồng lên khối cơ sở chroma (hoặc cả khối cơ sở luma và chroma) tương ứng sẽ được thay thế bằng các mẫu đệm.

Theo phương án khác, ứng viên trộn IBC có thể chỉ được sử dụng nếu vị trí nằm ngoài khối cơ sở.

Theo phương án khác, phép suy dẫn ứng viên trộn IBC sẽ sử dụng biên giới của khối cơ sở khi xác định vị trí các khối liền kề.

Nếu phân vùng cây riêng (còn được gọi là cây lưỡng) được áp dụng ở I-slice, các

thành phần luma và chroma có thể sử dụng các phân vùng cây khác nhau. Đối với phân vùng cây chroma, kích cỡ CU nhỏ nhất có thể là  $2 \times 2$ , kích cỡ đó quá nhỏ và sẽ làm giảm công suất xử lý.

Theo một phương án, kích cỡ CU chroma nhỏ nhất bị ràng buộc bởi kích cỡ/diện tích, chiều rộng/chiều cao, hoặc chiều sâu. Ví dụ, kích cỡ CU chroma nhỏ nhất có thể được thiết lập bằng 16 (theo độ phân giải mẫu chroma). Theo ví dụ khác, chiều rộng hoặc chiều cao CU chroma nhỏ nhất có thể được thiết lập bằng 4.

Điều kiện ràng buộc kích cỡ có thể là phép biến đổi mã hóa hoặc phép biến đổi chuẩn.

Theo phương án khác, khái niệm CU gốc có thể được áp dụng chỉ cho cây luma, chỉ cho cây chroma, hoặc cả cây luma và cây chroma. Ví dụ, CU gốc được áp dụng chỉ cho cây chroma. Giá trị ngưỡng của kích cỡ CU chroma gốc là 64, 32, hoặc 16. Đối với CU chroma gốc này, sự phân chia CU tiếp được thực hiện. Tuy nhiên, đối với các CU con thuộc CU gốc này, các mẫu tham chiếu được chia sẻ cho dự đoán Intra và/hoặc LM.

Nếu cây phân vùng được chia sẻ chung (ví dụ, phân vùng cây chung ở I-slice hoặc I-slice có phân vùng cây chung), các phương pháp khác nhau được đề xuất nhằm tránh các khối Intra chroma nhỏ hoặc nhằm giảm sự phụ thuộc dữ liệu giữa các khối Intra chroma liên tiếp.

Phương pháp 1: Các điều kiện ràng buộc cơ chế thuộc trong khối cơ sở

Theo phương án khác, khối cơ sở được xác định, và toàn bộ các khối thuộc trong khối cơ sở này phải có cơ chế dự đoán giống nhau.

Theo một phương án, cơ chế giống nhau có nghĩa là toàn bộ các khối thuộc trong khối cơ sở phải là cơ chế dự đoán Intra/IBC, hoặc cơ chế dự đoán Inter.

Thành phần luma có thể được phân vùng tiếp thành các khối nhỏ hơn, và việc phân vùng các thành phần chroma sẽ theo sau các khối luma. Đối với các khối dự đoán Intra thuộc trong khối cơ sở, các khối chroma sẽ sử dụng các biên giới bên trái và trên cùng của khối cơ sở để tạo ra biến dự đoán Intra, điều mà gây ra sự giảm độ phụ thuộc dữ liệu giữa các khối chroma liên tiếp. Theo một phương án, đối với CU chroma này, toàn bộ các CU con có thể sử dụng một TU.

Khi cơ chế sao chép khối nội ảnh IBC (Intra Block Copy - IBC) được sử dụng:

- Nếu BV chroma được suy dẫn làm cho khối chroma tham chiếu chồng lên khối cơ sở chroma tương ứng:
  - i. Theo một phương án, ở cây lưỡng, khối chroma không thể sử dụng cơ chế IBC.
  - ii. Theo phương án khác, ở cây lưỡng, cây chung hoặc cả cây lưỡng và cây chung, BV chroma được suy dẫn sẽ được mở rộng theo phương ngang/đứng hoặc cả phương ngang và phương đứng để làm cho khối chroma tham chiếu không chồng lên khối cơ sở chroma tương ứng.
  - iii. Theo phương án khác, ở cây lưỡng, cây chung hoặc cả cây lưỡng và cây chung, các mẫu của khối chroma tham chiếu mà chồng lên khối cơ sở chroma tương ứng sẽ được thay thế bằng các mẫu đệm.

Đối với CU gốc chứa các CU con phải có cơ chế dự đoán giống nhau, các cơ chế dự đoán của mỗi CU được ràng buộc giống nhau theo một phương án. Ví dụ, các cú pháp cơ chế dự đoán của mỗi CU cũng được báo hiệu. Tuy nhiên, chúng sẽ giống nhau. Theo một phương án, sự tương hợp dòng bit mà các cơ chế dự đoán thuộc trong CU gốc sẽ giống nhau. Theo phương án khác, cơ chế dự đoán của CU thứ nhất trong khối cơ sở được báo hiệu. Đối với các CU còn lại trong khối cơ sở, cơ chế dự đoán không được báo hiệu. Theo một ví dụ, chỉ các CU cơ chế Inter cần báo hiệu cờ skip và các cú pháp liên quan. Nếu CU đã được suy diễn làm CU Intra, thì cờ skip được suy diễn là false.

#### Phương pháp 2: Hủy bỏ phân chia chroma trong khối cơ sở

Trong trường hợp mà khối cơ sở được xác định, thành phần luma của khối cơ sở này có thể được phân vùng tiếp thành các khối nhỏ hơn, nhưng các thành phần chroma của khối cơ sở này không thể được phân vùng tiếp, điều này giúp biến nhiều khối luma tương xứng với một khối chroma. Các khối thuộc trong khối cơ sở này không phải có cơ chế dự đoán giống nhau.

Theo một phương án, cơ chế dự đoán của CU chroma không phân chia có thể được báo hiệu.

Theo phương án khác, cơ chế dự đoán của CU chroma không phân chia có thể được suy diễn.

Nếu cơ chế dự đoán của các khối chroma thuộc CU cơ sở được suy diễn, thì khối luma tham chiếu được xác định. Khối luma tham chiếu có thể là khối luma thứ nhất trong khối cơ sở, khối luma cuối cùng trong khối cơ sở, khối luma bao lấy vị trí trung tâm (ví dụ dưới cùng-bên phải-chính giữa, ví dụ khối BR-C trên Fig. 6) của khối cơ sở, khối luma có diện tích lớn nhất trong khối cơ sở, khối luma bao lấy một phần tư góc của khối cơ sở, khối luma được lập mã Intra đầu tiên trong khối cơ sở, khối luma được lập mã Inter đầu tiên trong khối cơ sở. Nếu IBC được sử dụng, khối tham chiếu có thể là khối luma được lập mã IBC đầu tiên, khối luma được lập mã IBC cuối cùng, hoặc khối luma có vị trí định trước (ví dụ vị trí trung tâm) trong khối cơ sở.

Theo một phương án, nếu khối luma tham chiếu là dự đoán Intra, cơ chế dự đoán Intra của khối chroma sẽ được suy dẫn theo khối luma tham chiếu.

Theo phương án khác, nếu khối luma tham chiếu là dự đoán Intra, cơ chế dự đoán Intra của khối chroma sẽ là cơ chế định trước, ví dụ DC hoặc planar.

Theo một phương án, nếu khối luma tham chiếu là dự đoán Intra, thì khối chroma được xem là cơ chế phi dự đoán phân vùng phụ nội ảnh non-ISP (Intra Sub-partition Prediction - ISP).

Theo phương án khác, đối với khối chroma, cơ chế dự đoán của nó có thể được báo hiệu ẩn và có thể khác với khối luma.

Theo một phương án, nếu khối luma tham chiếu là dự đoán Inter, MV của khối chroma được suy dẫn từ MV của khối luma.

Theo một phương án, CU chroma không phân chia được suy diễn làm cơ chế dự đoán Inter không thể là cơ chế trộn merge.

Theo một phương án, CU chroma không phân chia được suy diễn làm cơ chế dự đoán Inter không thể là cơ chế biến đổi khối phụ SBT (Sub-Block Transform - SBT).

Theo một phương án, CU chroma không phân chia được suy diễn làm cơ chế dự đoán Inter không thể áp dụng phép lọc vectơ chuyển động tại bộ giải mã DMVR (Decoder-side Motion Vector Refinement DMVR), và MV tham chiếu là MV trước/sau phép lọc.

Theo một phương án, CU chroma không phân chia được suy diễn làm cơ chế dự đoán Inter không thể là cơ chế tam giác.

Theo ví dụ khác, MV luma được lấy trung bình trọng số của các CU luma tương ứng có thể được sử dụng cho CU chroma này. Trọng số vừa nêu có thể giống nhau cho toàn bộ các CU hoặc phụ thuộc vào diện tích CU. Theo một ví dụ, nếu CU luma tương ứng sử dụng cơ chế khối phụ (ví dụ cơ chế ATMVP hoặc cơ chế affin hoặc cơ chế MV planar), CU chroma cũng sử dụng cơ chế giống nhau nhưng với kích cỡ CU lớn hơn. Theo ví dụ khác, MV chroma được báo hiệu ẩn.

Khi các khối luma thuộc trong khối cơ sở có cơ chế dự đoán Intra và Inter, một số điều kiện ràng buộc có thể được áp dụng để suy dẫn cơ chế dự đoán Intra chroma.

Theo một phương án, thành phần chroma có thể cơ chế dự đoán Intra Inter kết hợp CIIP (Combined Intra Inter Prediction - CIIP).

Theo một phương án, khối tham chiếu có thể chỉ là cơ chế Inter. Thông tin cơ chế dự đoán Inter có thể được lấy từ khối Inter luma đầu tiên, khối luma Inter cuối cùng, hoặc khối Inter luma có diện tích lớn nhất.

Theo một phương án, nếu khối luma tham chiếu là cơ chế dự đoán Intra, cơ chế mô hình tuyến tính LM (Linear Model - LM) không được hủy bỏ.

Theo phương án khác, cơ chế dự đoán dùng cho CU chroma không phân chia có thể suy diễn một cách có điều kiện. Nếu khối luma tham chiếu là cơ chế dự đoán Inter, CU chroma được suy diễn làm cơ chế dự đoán Inter, và thông tin chuyển động được lấy từ khối luma tham chiếu. Nếu khối luma tham chiếu không là cơ chế Inter, thì cơ chế dự đoán của CU chroma không phân chia sẽ được báo hiệu.

Khi cơ chế sao chép khối nội ảnh IBC (Intra Block Copy - IBC) được sử dụng, chỉ khối IBC có BV mà không làm cho khối chroma tham chiếu chồng lên khối cơ sở chroma tương ứng có thể được chọn. Theo phương án khác, chỉ khối IBC có BV mà không làm cho khối luma và chroma tham chiếu chồng lên khối cơ sở luma và chroma tương ứng có thể được chọn.

Theo phương án khác, nếu khối luma tham chiếu được chọn là cơ chế IBC, thì khối tham chiếu luma được suy dẫn không thể chồng lên khối cơ sở.

Theo phương án khác, nếu khối luma tham chiếu được chọn là cơ chế IBC, và nếu khối tham chiếu luma được suy dẫn chồng lên khối cơ sở, BV luma sẽ được mở rộng.

Theo phương án khác, nếu khối luma tham chiếu được chọn là cơ chế IBC, và nếu

khối tham chiếu luma được suy dẫn chồng lên khối cơ sở, chỉ BV chroma được suy dẫn sẽ được mở rộng.

Phương pháp 3: Các điều kiện ràng buộc cơ chế và hủy bỏ một cách có điều kiện phép phân chia chroma thuộc trong khối cơ sở

Theo phương án khác, khối cơ sở được xác định; thành phần luma của khối cơ sở này có thể được phân vùng tiếp thành các khối nhỏ hơn; và liệu các thành phần chroma của khối cơ sở có thể được phân chia tiếp hay không được quyết định bởi cơ chế dự đoán của các khối luma thuộc cùng khối cơ sở và các kích cỡ khối chroma tương ứng.

Theo một phương án, nếu toàn bộ các khối thuộc khối cơ sở hiện thời là cơ chế dự đoán Inter, thì việc phân vùng các thành phần chroma theo sau các khối luma. Nếu toàn bộ các khối thuộc khối cơ sở hiện thời là cơ chế Intra/IBC, thì các thành phần chroma của khối cơ sở này không thể được phân chia tiếp, điều này giúp biến nhiều khối luma tương xứng với một khối chroma. Tương tự, vì cơ chế Inter không được hỗ trợ theo phân vùng luma 4x4, khi nút phân vùng có một hoặc nhiều nút con trong đó kích cỡ khối luma bằng 4x4, thì nút vừa nêu được suy diễn làm cơ chế non-Inter (cơ chế Intra hoặc cơ chế IBC).

Theo một phương án, cờ mode\_constraint\_flag được báo hiệu để biểu thị cơ chế dự đoán của khối cơ sở. Toàn bộ các khối thuộc trong khối cơ sở sẽ hoặc là cơ chế dự đoán Inter hoặc cơ chế dự đoán Intra/IBC. Ví dụ, khi mode\_constraint\_flag bằng 1, toàn bộ các khối thuộc trong khối cơ sở là các khối được lập mã Inter, nếu không thì, toàn bộ các khối thuộc trong khối cơ sở là các khối được lập mã Intra/IBC.

Liệu có giải mã mode\_constraint\_flag hay không phụ thuộc vào biến modeTypeCondition. Biến modeTypeCondition được suy dẫn như sau:

- Nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 0
  - slice\_type == I và qtbt\_dual\_tree\_intra\_flag bằng 1
  - modeTypeCurr không bằng MODE\_TYPE\_ALL
- Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 1
  - cbWidth \* cbHeight bằng THRES và split\_qt\_flag bằng 1

- $cbWidth * cbHeight$  bằng THRES và  $MttSplitMode[x0][y0][mttDepth]$  bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER
- $cbWidth * cbHeight$  bằng  $(THRES \gg 1)$  và  $MttSplitMode[x0][y0][mttDepth]$  bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER
- Nếu không thì, nếu một trong số những điều kiện sau đúng, `modeTypeCondition` được thiết lập bằng  $1 + (slice\_type \neq I ? 1 : 0)$ 
  - $cbWidth * cbHeight$  bằng THRES và  $MttSplitMode[x0][y0][mttDepth]$  bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER
  - $cbWidth * cbHeight$  bằng  $(THRES \ll 1)$  và  $MttSplitMode[x0][y0][mttDepth]$  bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER
- Nếu không thì, `modeTypeCondition` được thiết lập bằng 0.

Ở các phép suy dẫn nêu trên,  $MttSplitMode[x0][y0][mttDepth]$  biểu diễn các phép phân chia nhị phân và tam phân ngang và đứng đơn vị lập mã thuộc cây đa phân MTT (Multi-Type Tree - MTT). Biến THRES được suy dẫn như sau:

Nếu định dạng màu là 4:4:4, THRES được thiết lập bằng 16,

Nếu không thì, nếu định dạng màu là 4:2:2, THRES được thiết lập bằng 32,

Nếu không thì, nếu định dạng màu là 4:2:0, THRES được thiết lập bằng 64.

Theo phương án khác, biến THRES được suy dẫn như sau:

Nếu định dạng màu là 4:4:4, THRES được thiết lập bằng 8/16/32/64,

Nếu không thì, nếu định dạng màu là 4:2:2, THRES được thiết lập bằng 16/32/64/128,

Nếu không thì, nếu định dạng màu là 4:2:0, THRES được thiết lập bằng 32/64/128/256.

Theo phương án khác, định dạng chroma được sử dụng làm một trong số những điều kiện dùng để xác định `modeTypeCondition`. Phép suy dẫn `modeTypeCondition` cho định dạng chroma 422 và định dạng chroma 420 là khác nhau. `ModeTypeCondition` luôn được thiết lập là 0 khi định dạng chroma bằng định dạng chroma 444.

ModeTypeCondition được suy dẫn như sau:

- Nếu `chroma_format_idc` bằng 1 (khi định dạng chroma 4:2:0 được sử dụng), biến `modeTypeCondition` được suy dẫn như sau:
  - Nếu một trong số những điều kiện sau đúng, `modeTypeCondition` được thiết lập bằng 0
    - `slice_type == I` và `qtbtt_dual_tree_intra_flag` bằng 1
    - `modeTypeCurr` không bằng `MODE_TYPE_ALL`
- Nếu không thì, nếu một trong số những điều kiện sau 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` (theo phương án khác, điều kiện này có thể được lược bỏ)
- Nếu không thì, nếu một trong số những điều kiện sau đúng, `modeTypeCondition` được thiết lập bằng  $1 + (\text{slice\_type} \neq I ? 1 : 0)$ 
  - `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`.

Theo phương án khác, 2 điều kiện dưới đây được thêm vào:

- `cbWidth` bằng 8 và `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_BT_VER`
- `cbWidth` bằng 16 và `split_qt_flag` bằng 0 và `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_TT_VER`.

Theo phương án khác, 4 điều kiện dưới đây được thêm vào:

- `cbWidth` bằng 8 và `cbHeight` bằng 16 và

- MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- cbWidth bằng 16 và cbHeight bằng 16 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
  - cbWidth bằng 4 và cbHeight bằng 32 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc  
SPLIT\_TT\_HOR
  - cbWidth bằng 4 và cbHeight bằng 64 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
  - Nếu không thì, modeTypeCondition được thiết lập bằng 0
  - Nếu không thì, nếu chroma\_format\_idc bằng 2 (khi định dạng chroma 4:2:2 được sử dụng), biến modeTypeCondition được suy dẫn như sau:
    - Nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 0
      - slice\_type == I và qtbtt\_dual\_tree\_intra\_flag bằng 1
      - modeTypeCurr không bằng MODE\_TYPE\_ALL
  - Nếu không thì, nếu một trong số những điều kiện sau 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.

Theo một phương án, thiết lập modeTypeCondition được thể hiện như sau:

- nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng  $1 + (\text{slice\_type} \neq I ? 1 : 0)$ 
  - cbWidth bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- cbWidth bằng 16 và split\_qt\_flag bằng 0 và

MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER.

Theo phương án khác, thiết lập modeTypeCondition được thể hiện như sau:

- nếu không thì, nếu một trong số những điều kiện sau đúng,  
modeTypeCondition được thiết lập bằng  $1 + (\text{slice\_type} \neq I ? 1 : 0)$
- cbWidth bằng 8 và cbHeight bằng 8 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- cbWidth bằng 16 và cbHeight bằng 8 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- cbWidth bằng 4 và cbHeight bằng 16 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc  
SPLIT\_TT\_HOR
- cbWidth bằng 4 và cbHeight bằng 32 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- Nếu không thì, modeTypeCondition được thiết lập bằng 0
- Nếu không thì, nếu chroma\_format\_idc bằng 3 (khi định dạng chroma 4:4:4 được sử dụng), biến modeTypeCondition được thiết lập bằng 0.
- Nếu không thì, (chroma\_format\_idc bằng 0, khi monochrome được sử dụng),  
biến modeTypeCondition được thiết lập bằng 0.

Theo phương án khác, định dạng chroma được sử dụng làm một trong số những điều kiện dùng để xác định modeTypeCondition. Phép suy dẫn modeTypeCondition cho định dạng chroma 422 và định dạng chroma 420 là khác nhau. ModeTypeCondition luôn được thiết lập là 0 khi định dạng chroma bằng định dạng chroma 444. ModeTypeCondition được suy dẫn như sau:

- Nếu một trong số những điều kiện sau 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 hoặc 3

- Nếu không thì, nếu một trong số những điều kiện sau 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
- Nếu không thì, một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 1 + (slice\_type != I ? 1 : 0)
  - nếu chroma\_format\_idc bằng 1 và cbWidth \* cbHeight bằng 64 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER
  - nếu chroma\_format\_idc bằng 1 và cbWidth \* cbHeight bằng 128 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER.

Theo một phương án, 2 điều kiện dưới đây được thêm vào:

- cbWidth bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- cbWidth bằng 16 và split\_qt\_flag bằng 0 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER.

Theo phương án khác, 8 điều kiện dưới đây được thêm vào:

- nếu chroma\_format\_idc bằng 1 và cbWidth bằng 8 và cbHeight bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- nếu chroma\_format\_idc bằng 1 và cbWidth bằng 16 và cbHeight bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- nếu chroma\_format\_idc bằng 1 và cbWidth bằng 4 và cbHeight bằng 32 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_TT\_HOR
- nếu chroma\_format\_idc bằng 1 và cbWidth bằng 4 và cbHeight bằng 64 và

MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR

- nếu chroma\_format\_idc bằng 2 và cbWidth bằng 8 và cbHeight bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- nếu chroma\_format\_idc bằng 2 và cbWidth bằng 16 và cbHeight bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- nếu chroma\_format\_idc bằng 2 và cbWidth bằng 4 và cbHeight bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_TT\_HOR
- nếu chroma\_format\_idc bằng 2 và cbWidth bằng 4 và cbHeight bằng 32 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- Nếu không thì, modeTypeCondition được thiết lập bằng 0.

Ở phương án này, modeTypeCondition có thể được thiết lập bằng 2 chỉ khi định dạng chroma 4:2:0 được sử dụng. Do vậy, mode\_constraint\_flag chỉ được báo hiệu/được phân tích khi định dạng chroma 4:2:0 được sử dụng. Khi định dạng chroma không là 4:2:0, mode\_constraint\_flag không được báo hiệu. Tương tự, modeTypeCondition có thể được thiết lập bằng 1 chỉ khi định dạng chroma 4:2:0 hoặc 4:2:2 được sử dụng. Do vậy, dạng khối cơ sở được suy diễn làm MODE\_TYPE\_INTRA khi định dạng chroma 4:2:0 hoặc 4:2:2 được sử dụng. Đối với định dạng chroma 4:2:2, modeTypeCondition có thể chỉ bằng 0 hoặc 1. Đối với định dạng chroma 4:4:4, modeTypeCondition có thể chỉ bằng 0. Đối với định dạng chroma 4:2:0, modeTypeCondition có thể chỉ bằng 0 hoặc 1 hoặc 2.

Theo phương án khác, điều kiện ràng buộc không được xác định bằng kích cỡ CB (ví dụ cbWidth \* cbHeight). Thay vào đó, nó được xác định bằng cách tính cbWidth, cbHeight, SubWidthC, SubHeightC, trong đó SubWidthC, SubHeightC được xác định bằng định dạng chroma. ModeTypeCondition có thể được suy diễn như sau. Liệu có giải mã mode\_constraint\_flag hay không phụ thuộc vào biến modeTypeCondition. Biến modeTypeCondition được suy diễn như sau:

- Nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 0
- slice\_type == I và qtbtt\_dual\_tree\_intra\_flag bằng 1

- modeTypeCurr không bằng MODE\_TYPE\_ALL
- Nếu không thì, nếu một trong số những điều kiện sau 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
- Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 1 + (slice\_type != I ? 1 : 0)
  - (cbWidth/SubWidthC) \* (cbHeight/SubHeightC) bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER
  - (cbWidth/SubWidthC) \* (cbHeight/SubHeightC) bằng 32 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER.

Theo trên, modeTypeCurr biểu diễn modeType của khối hiện thời. Biến modeType xác định liệu cơ chế lập mã Intra (tức là, MODE\_INTRA), IBC (tức là, MODE\_IBC), palette (tức là, MODE\_PLT), và Inter có thể được sử dụng (tức là, MODE\_TYPE\_ALL), hay liệu chỉ cơ chế lập mã Intra, IBC, và palette có thể được sử dụng (tức là, MODE\_TYPE\_INTRA), hay liệu chỉ cơ chế Inter có thể được sử dụng (tức là, MODE\_TYPE\_INTER) cho các đơn vị lập mã nằm trong nút cây lập mã.

Theo một phương án, 2 điều kiện dưới đây được thêm vào:

- (cbWidth/SubWidthC) bằng 4 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- (cbWidth/SubWidthC) bằng 8 và split\_qt\_flag bằng 0 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER.

Theo phương án khác, 4 điều kiện dưới đây được thêm vào:

- (cbWidth/SubWidthC) bằng 4 và (cbHeight/SubHeightC) bằng 8 và

MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER

- (cbWidth/SubWidthC) bằng 8 và (cbHeight/SubHeightC) bằng 8 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- (cbWidth/SubWidthC) bằng 2 và (cbHeight/SubHeightC) bằng 16 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc  
SPLIT\_TT\_HOR
- (cbWidth/SubWidthC) bằng 2 và (cbHeight/SubHeightC) bằng 32 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- Nếu không thì, modeTypeCondition được thiết lập bằng 0.

Theo phương án khác, điều kiện ràng buộc không được xác định bằng kích cỡ cb (ví dụ cbWidth \* cbHeight). Nó được xác định bằng cách tính định dạng chroma, cbWidth, cbHeight, SubWidthC, SubHeightC, trong đó SubWidthC, SubHeightC được xác định bằng định dạng chroma. ModeTypeCondition có thể được suy dẫn như sau. Biến modeTypeCondition được suy dẫn như sau:

- Nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 0
  - slice\_type == I và qtbt\_dual\_tree\_intra\_flag bằng 1
  - modeTypeCurr không bằng MODE\_TYPE\_ALL
  - chroma\_format\_idc == 0
  - chroma\_format\_idc == 3
- Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 1
  - (cbWidth / SubWidthC) \* (cbHeight / SubHeightC) bằng 16 và  
split\_qt\_flag bằng 1
  - (cbWidth / SubWidthC) \* (cbHeight / SubHeightC) bằng 16 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc  
SPLIT\_TT\_VER
  - (cbWidth / SubWidthC) \* (cbHeight / SubHeightC) bằng 8 và  
MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc

## SPLIT\_BT\_VER

- Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng  $1 + (\text{slice\_type} \neq I ? 1 : 0)$
- $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$  bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER
- $(\text{cbWidth} / \text{SubWidthC}) * (\text{cbHeight} / \text{SubHeightC})$  bằng 32 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER.

Theo một phương án, 2 điều kiện dưới đây được thêm vào:

- $(\text{cbWidth}/\text{SubWidthC})$  bằng 4 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- $(\text{cbWidth}/\text{SubWidthC})$  bằng 8 và split\_qt\_flag bằng 0 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER.

Theo phương án khác, 4 điều kiện dưới đây được thêm vào:

- $(\text{cbWidth}/\text{SubWidthC})$  bằng 4 và  $(\text{cbHeight}/\text{SubHeightC})$  bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER
- $(\text{cbWidth}/\text{SubWidthC})$  bằng 8 và  $(\text{cbHeight}/\text{SubHeightC})$  bằng 8 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- $(\text{cbWidth}/\text{SubWidthC})$  bằng 2 và  $(\text{cbHeight}/\text{SubHeightC})$  bằng 16 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_TT\_HOR
- $(\text{cbWidth}/\text{SubWidthC})$  bằng 2 và  $(\text{cbHeight}/\text{SubHeightC})$  bằng 32 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- Nếu không thì, modeTypeCondition được thiết lập bằng 0.

Theo phương án khác, điều kiện ràng buộc không được xác định bằng kích cỡ cb (ví dụ  $\text{cbWidth} * \text{cbHeight}$ ). Thay vào đó, nó được xác định bằng cách tính chroma\_format\_idc, cbWidth, cbHeight, SubWidthC, SubHeightC, trong đó SubWidthC, SubHeightC được xác định bằng định dạng chroma. ModeTypeCondition

có thể được suy dẫn như sau. Biến `modeTypeCondition` được suy dẫn như sau:

- Nếu một trong số những điều kiện sau 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 == 0`
  - `chroma_format_idc == 3`
- Nếu không thì, nếu một trong số những điều kiện sau đúng, `modeTypeCondition` được thiết lập bằng 1 được suy dẫn như sau:

Biến `minLuma` được thiết lập bằng:

- $(cbWidth * cbHeight) >> 2$  nếu `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_TT_HOR` hoặc `SPLIT_TT_VER` hoặc `split_qt_flag` bằng 1
- $(cbWidth * cbHeight) >> 1$  nếu `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_BT_HOR` hoặc `SPLIT_BT_VER`
- `cbWidth * cbHeight`, nếu không thì.

Biến `minChroma` được thiết lập bằng `minLuma / (SubWidthC * SubHeightC)`.

Theo một phương án,

- `modeTypeCondition` được thiết lập bằng 0 nếu `minChroma` lớn hơn 8.
- Nếu không thì, `modeTypeCondition` được thiết lập bằng `1 + (slice_type == I || minLuma == 16?) 0 : 1`

Theo phương án khác, biến `minLumaWidth` được thiết lập bằng:

- $cbWidth >> 2$  nếu `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_TT_VER`
- $cbWidth >> 1$  nếu `MttSplitMode[ x0 ][ y0 ][ mttDepth ]` bằng `SPLIT_BT_VER` hoặc `split_qt_flag` bằng 1
- `cbWidth`, nếu không thì.

Biến `minChromaWidth` được thiết lập bằng `minLumaWidth / SubWidthC`.

Biến minLumaHeight được thiết lập bằng:

- cbHeight >> 2 nếu MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- cbHeight >> 1 nếu MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc split\_qt\_flag bằng 1
- cbHeight, nếu không thì.

Biến minChromaHeight được thiết lập bằng minLumaHeight / SubHeightC.

- modeTypeCondition được thiết lập bằng 0 nếu minChroma lớn hơn 8 và minChromaWidth lớn hơn 2 và minChromaHeight lớn hơn 8.
- Nếu không thì, modeTypeCondition được thiết lập bằng 1 + (slice\_type == I || minLuma == 16?) 0 : 1.

Theo phương án khác, biến minLumaWidth được thiết lập bằng:

- cbWidth >> 2 nếu MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER và split\_qt\_flag bằng 0
- cbWidth >> 1 nếu MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER hoặc split\_qt\_flag bằng 1
- cbWidth, nếu không thì.

Biến minChromaWidth được thiết lập bằng minLumaWidth / SubWidthC.

- modeTypeCondition được thiết lập bằng 0 nếu minChroma lớn hơn 8 và minChromaWidth lớn hơn 2.
- Nếu không thì, modeTypeCondition được thiết lập bằng 1 + (slice\_type == I || minLuma == 16?) 0 : 1.

Theo phương án khác, điều kiện ràng buộc không được xác định bằng kích cỡ cb (ví dụ cbWidth \* cbHeight). Nó được xác định bằng cách tính cbWidth, cbHeight, SubWidthC, SubHeightC, trong đó SubWidthC, SubHeightC được xác định bằng định dạng chroma. ModeTypeCondition có thể được suy dẫn như sau. Biến modeTypeCondition được suy dẫn như sau:

- Nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập

bằng 0

–slice\_type == I và qtbtt\_dual\_tree\_intra\_flag bằng 1

–modeTypeCurr không bằng MODE\_TYPE\_ALL

– Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng 1

– cbWidth \* cbHeight bằng 64, cbWidth \* cbHeight nhỏ hơn  $64 * \text{SubWidthC} * \text{SubHeightC}$ , và split\_qt\_flag bằng 1

– cbWidth \* cbHeight bằng 64, cbWidth \* cbHeight nhỏ hơn  $64 * \text{SubWidthC} * \text{SubHeightC}$ , và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER

– cbWidth \* cbHeight bằng 32, cbWidth \* cbHeight nhỏ hơn  $32 * \text{SubWidthC} * \text{SubHeightC}$ , và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER

– Nếu không thì, nếu một trong số những điều kiện sau đúng, modeTypeCondition được thiết lập bằng  $1 + (\text{slice\_type} \neq I ? 1 : 0)$

– cbWidth \* cbHeight nhỏ hơn  $32 * \text{SubWidthC} * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_BT\_VER

– cbWidth \* cbHeight nhỏ hơn  $64 * \text{SubWidthC} * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR hoặc SPLIT\_TT\_VER.

Theo một phương án, 2 điều kiện dưới đây được thêm vào:

– cbWidth bằng  $4 * \text{SubWidthC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER

– cbWidth bằng  $8 * \text{SubWidthC}$  và split\_qt\_flag bằng 0 và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER.

Theo phương án khác, 4 điều kiện dưới đây được thêm vào:

– cbWidth bằng  $4 * \text{SubWidthC}$  và cbHeight bằng  $8 * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_VER

- cbWidth bằng  $8 * \text{SubWidthC}$  và cbHeight bằng  $8 * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_VER
- cbWidth bằng  $2 * \text{SubWidthC}$  và cbHeight bằng  $16 * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_BT\_HOR hoặc SPLIT\_TT\_HOR
- cbWidth bằng  $2 * \text{SubWidthC}$  và cbHeight bằng  $32 * \text{SubHeightC}$  và MttSplitMode[ x0 ][ y0 ][ mttDepth ] bằng SPLIT\_TT\_HOR
- Nếu không thì, modeTypeCondition được thiết lập bằng 0.

Theo phương án khác, khi monochrome được sử dụng, modeTypeCondition luôn được thiết lập bằng 0.

Khi modeTypeCondition bằng 2, mode\_constraint\_flag sẽ được giải mã, và modeType được thiết lập theo mode\_constraint\_flag. Khi modeTypeCondition bằng 1, modeType được suy diễn làm MODE\_TYPE\_INTRA, toàn bộ các khối thuộc trong các khối cơ sở được dự đoán intra/IBC. Khi modeTypeCondition bằng 0, modeType không được thay đổi. Bảng cú pháp là như sau:

Bảng 1. Bảng cú pháp ví dụ

|                                                                                                                                                                                                               |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| coding_tree( x0, y0, cbWidth, cbHeight, qgOn, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr , modeTypeCurr ) {                                                                             | Biên mô tả |
| if( ( allowSplitBtVer    allowSplitBtHor    allowSplitTtVer    allowSplitTtHor    allowSplitQT )<br>&&( x0 + cbWidth <= pic_width_in_luma_samples )<br>&&( y0 + cbHeight <= pic_chiều cao_in_luma_samples ) ) |            |
| split_cu_flag                                                                                                                                                                                                 | ae(v)      |
| if( cu_qp_delta_enabled_flag && qgOn && cbSubdiv <= cu_qp_delta_subdiv ) {                                                                                                                                    |            |
| IsCuQpDeltaCoded = 0                                                                                                                                                                                          |            |

|                                                                                                                                                              |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| CuQpDeltaVal = 0                                                                                                                                             |       |
| CuQgTopLeftX = x0                                                                                                                                            |       |
| CuQgTopLeftY = y0                                                                                                                                            |       |
| }                                                                                                                                                            |       |
| if( split_cu_flag ) {                                                                                                                                        |       |
| if( ( allowSplitBtVer    allowSplitBtHor    allowSplitTtVer   <br>allowSplitTtHor ) &&<br>allowSplitQT )                                                     |       |
| split_qt_flag                                                                                                                                                | ae(v) |
| if( !split_qt_flag ) {                                                                                                                                       |       |
| if( ( allowSplitBtHor    allowSplitTtHor ) &&<br>( allowSplitBtVer    allowSplitTtVer ) )                                                                    |       |
| mtt_split_cu_vertical_flag                                                                                                                                   | ae(v) |
| if( ( allowSplitBtVer && allowSplitTtVer &&<br>mtt_split_cu_vertical_flag )   <br>( allowSplitBtHor && allowSplitTtHor<br>&& !mtt_split_cu_vertical_flag ) ) |       |
| mtt_split_cu_binary_flag                                                                                                                                     | ae(v) |
| }                                                                                                                                                            |       |
| if( modeTypeCondition == 2 ) {                                                                                                                               |       |
| mode_constraint_flag                                                                                                                                         | ae(v) |
| modeType =<br>mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE_INTER                                                                                       |       |

|                                                                                                                                      |  |
|--------------------------------------------------------------------------------------------------------------------------------------|--|
| }                                                                                                                                    |  |
| else                                                                                                                                 |  |
| modeType = ( modeTypeCondition ==<br>1 ) ? MODE_TYPE_INTRA : modeTypeCurr                                                            |  |
| treeType = ( modeType ==<br>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, qgOn, cbSubdiv + 1,<br>cqtDepth, mttDepth + 1, depthOffset, 0, treeType ,<br>modeType )  |  |
| if( x1 < pic_width_in_luma_samples )                                                                                                 |  |
| coding_tree( x1, y0, cbWidth / 2, cbHeightY, qgOn, cbSubdiv + 1,<br>cqtDepth, mttDepth + 1, depthOffset, 1, treeType ,<br>modeType ) |  |
| } else if( MttSplitMode[ x0 ][ y0 ][ mttDepth ] == SPLIT_BT_HOR ) {                                                                  |  |
| .....                                                                                                                                |  |

Theo một phương án, không phụ thuộc định dạng màu, nếu kích cỡ CB chroma của một trong số các khối con nhỏ hơn 16 mẫu chroma, và khối cơ sở chưa được thiết lập, thì vị trí trên cùng-bên trái của khối cơ sở được thiết lập là vị trí trên cùng-bên trái của khối hiện thời, và chiều rộng và chiều cao của khối cơ sở bằng chiều rộng và chiều cao của khối hiện thời, nếu không thì, khối cơ sở không được thiết lập. Giá trị ngưỡng có

thể là 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, và 4096.

Khi khối cơ sở được xác định là cơ chế Inter, phân vùng khối luma 4x4 bị hủy bỏ. Do vậy, phân vùng khối mà làm cho một hoặc nhiều khối phụ con cơ kích cỡ khối luma bằng 4x4 bị hủy bỏ. Cú pháp của cờ phân vùng khối đó được bỏ qua và được coi là không bị phân chia bởi các dạng phân vùng khối.

Theo một phương án, khi modeTypeCurr bằng MODE\_TYPE\_INTER, phân chia BT bị hủy bỏ khi diện tích CU/CB bằng 32, và phân chia TT bị hủy bỏ khi diện tích CU/CB bằng 64. Trong đó, modeTypeCurr được quyết định bởi nút phân vùng cha và được chuyển thành nút hiện thời. Do vậy, phụ thuộc vào biến (modeTypeCurr) được chuyển từ nút cha, phân chia BT hoặc TT bị hủy bỏ khi CU/CB bằng 32 hoặc 64, một cách tương ứng. Việc hủy bỏ phân vùng nghĩa là cú pháp phân chia không được báo hiệu và được suy diễn bằng 0. Ví dụ, allowBtSplit hoặc allowTtSplit được thiết lập bằng 0/False khi modeTypeCurr bằng MODE\_TYPE\_INTER và cbWidth \* cbHeight bằng 32 hoặc 64, một cách tương ứng.

Phương pháp 4: Các phép biến đổi giải khối khi hủy bỏ phân chia phân vùng chroma

Ở Inter slice (hoặc cây chung), nếu khối cơ sở chứa thành phần luma và thành phần chroma có phân vùng khác nhau, quá trình giải khối cho luma và chroma sẽ được tách riêng. Tức là, các thiết lập độ lớn biên giới BS (Boundary Strength - BS) của các khối/các CU luma không thể được tái sử dụng bởi các khối /các CU chroma. Thiết lập độ lớn biên giới (BS) của các khối/các CU chroma sẽ được thiết lập một cách riêng rẽ, và sau đó áp dụng phép lọc cho các thành phần chroma.

Theo một phương án, nếu CU hiện thời là CU chroma, và CU liền kề là CU luma, thì trị số độ lớn biên giới (BS) cho thành phần chroma sẽ được thiết lập bằng 2.

Theo một phương án, nếu CU hiện thời là CU chroma, và CU liền kề là CU luma, thì trị số độ lớn biên giới (BS) cho thành phần chroma sẽ được thiết lập bằng 1.

Phương pháp 5: Các phép biến đổi thứ cấp suy giảm RST (Reduced Secondary Transform - RST) khi hủy bỏ phân chia phân vùng chroma

Ở Inter slice (hoặc cây chung), nếu khối cơ sở chứa thành phần luma và thành phần chroma có phân vùng khác nhau, thì quyết định của RST đối với các khối luma và khối

chroma ở miền cơ sở là khác nhau.

Theo một phương án, RST không được áp dụng cho và được báo hiệu ở khối chroma trong khối cơ sở.

Theo phương án khác, RST chỉ được áp dụng và được báo hiệu cho khối chroma trong khối cơ sở nếu chiều rộng khối và chiều cao khối đều lớn hơn 2 mẫu chroma.

Thiết lập mức hệ số khác không trong khối cơ sở là giống như các thiết lập ở cây lưỡng.

Ví dụ, đối với các khối luma trong khối cơ sở, giá trị ngưỡng được chuyển thành LFNST\_SIG\_LUMA, và đối với khối chroma trong khối cơ sở, giá trị ngưỡng được chuyển thành LFNST\_SIG\_CHROMA, trong đó LFNST\_SIG\_LUMA và LFNST\_SIG\_CHROMA là giá trị ngưỡng cho các thành phần luma và chroma, một cách tương ứng.

Phép suy dẫn các biến thuộc tính trong khối cơ sở cũng giống như các thiết lập trong cây lưỡng.

Ví dụ, nếu ở cây đơn, chỉ số mts nhỏ hơn 2, và khối hiện thời là trong khối cơ sở, thì biến thuộc tính sẽ được tăng thêm 1; nếu không thì (ở cây đơn, chỉ số mts nhỏ hơn 2, nhưng không thuộc trong khối cơ sở), các biến thuộc tính sẽ không được tăng thêm 1.

Phương pháp 6: Các phép biến đổi dự đoán khối phụ Intra ISP (Intra Subpartition Prediction - ISP) khi hủy bỏ phân chia phân vùng chroma

Đối với các khối luma trong khối cơ sở, sẽ không có phân vùng phụ chroma.

Phương pháp 7: Các phép biến đổi Delta QP (DQP) khi hủy bỏ phân chia phân vùng chroma

Ở cây đơn, cu level delta QP sẽ không được báo hiệu ở khối chroma trong khối cơ sở.

Phương pháp 8: Báo hiệu cơ chế dự đoán khi hủy bỏ phân chia phân vùng chroma

Báo hiệu cơ chế dự đoán trong khối cơ sở có thể được suy diễn.

- Khi IBC không được sử dụng:

Đối với các khối luma trong khối cơ sở:

- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, và khối cơ sở là cơ chế dự đoán Intra/IBC, cơ chế dự đoán được suy diễn làm cơ chế dự đoán Intra, không có cú pháp cần được báo hiệu để biểu thị cơ chế dự đoán.
- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, và khối cơ sở là cơ chế dự đoán Inter, cơ chế dự đoán được suy diễn làm cơ chế dự đoán Inter, không có cú pháp cần được báo hiệu.

Đối với khối chroma:

- Theo một phương án, cơ chế dự đoán được suy diễn làm cơ chế dự đoán Intra.
- Khi IBC được sử dụng:

Đối với các khối luma trong khối cơ sở:

- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, và khối cơ sở là cơ chế dự đoán Inter, thì khối luma được suy diễn làm cơ chế dự đoán Inter, không có cú pháp cần được báo hiệu.
- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, và khối cơ sở là cơ chế dự đoán Intra/IBC, và IBC không được hủy bỏ ở khối hiện thời, cơ chế dự đoán được suy diễn làm cơ chế dự đoán Intra, không có cú pháp cần được báo hiệu.

Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, nếu khối cơ sở là cơ chế dự đoán Intra/IBC, và nếu IBC được sử dụng ở khối hiện thời, thì chỉ cờ dự đoán IBC cần được báo hiệu. Nếu cờ dự đoán IBC là true, thì cơ chế dự đoán là cơ chế IBC.

Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, nếu khối cơ sở là cơ chế dự đoán Intra/IBC, và nếu IBC được sử dụng ở khối hiện thời, thì chỉ cờ dự đoán IBC cần được báo hiệu. Nếu cờ dự đoán IBC là true, thì cơ chế dự đoán là cơ chế Intra.

Đối với khối chroma, theo một phương án, nếu IBC không được hủy bỏ ở khối hiện thời và cơ chế dự đoán được suy diễn làm cơ chế dự đoán Intra, thì không có cú

pháp cần được báo hiệu. Theo một phương án, nếu IBC được sử dụng ở khối hiện thời, thì chỉ cờ dự đoán IBC cần được báo hiệu. Nếu cờ dự đoán IBC là true, thì cơ chế dự đoán là cơ chế IBC. Theo một phương án, nếu IBC bị hủy bỏ ở khối hiện thời, thì chỉ cờ dự đoán IBC cần được báo hiệu. Nếu cờ dự đoán IBC là true, thì cơ chế dự đoán là cơ chế Intra.

Phương pháp 9: Báo hiệu cờ Skip khi hủy bỏ phân chia phân vùng chroma

Báo hiệu cơ chế dự đoán trong khối cơ sở có thể được suy diễn như sau.

- Khi IBC không được sử dụng:

Đối với các khối luma trong khối cơ sở:

- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, và khối cơ sở là cơ chế là cơ chế Intra/IBC, cờ skip không cần được báo hiệu, và được suy diễn là false.

Đối với khối chroma, cơ chế skip không thể được sử dụng, do đó không yêu cầu cú pháp cho cờ skip.

- Khi IBC được sử dụng:

Đối với các khối luma trong khối cơ sở:

- Theo một phương án, nếu vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, nếu khối cơ sở là cơ chế Intra/IBC và IBC không được hủy bỏ ở khối hiện thời, cờ skip không cần được báo hiệu, và được suy diễn là false.

Theo phương án khác, nếu cờ skip là true, và vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, khối cơ sở là cơ chế Intra/IBC, thì cơ chế dự đoán của khối hiện thời được suy diễn làm cơ chế IBC, không cần báo hiệu cờ IBC. Nếu không thì (nếu cờ skip là true, và vị trí trên cùng-bên trái của khối luma không giống như vị trí trên cùng-bên trái của khối cơ sở, khối cơ sở là cơ chế Inter), cơ chế dự đoán của khối hiện thời được suy diễn làm cơ chế Inter.

Đối với khối chroma, cơ chế skip không thể được sử dụng, do đó không yêu cầu cú pháp cho cờ skip.

Slice đã nêu ở trên có thể là một slice, một tile, một tile group, hoặc một picture.

Theo VTM-6, cây lưỡng cực bộ được sử dụng để ngăn CU Intra chroma nhỏ để nâng cao công suất xử lý. Ví dụ, nếu CU 128 mẫu luma được phân vùng bằng phân chia cây tam phân TT (Ternary-Tree - TT), và SCIPU được lập mã theo cơ chế Intra, thì SCIPU là cây lưỡng cực bộ. Ở SCIPU này, khối chroma không thể được phân chia và tạo thành CU 8x4 (tức là, 32 mẫu chroma).

Theo khái niệm của cây lưỡng cực bộ hoặc SCIPU, nó cố gắng không làm cho CU chroma nhỏ hơn 16 mẫu chroma. Tuy nhiên, CU 32 mẫu chroma có thể được phân chia tiếp thành hai CU mẫu 16 chroma mà không vi phạm điều kiện ràng buộc (tức là, không có CU chroma có mẫu chroma nhỏ hơn 16 mẫu). Nhằm nâng cao hiệu suất lập mã, dạng CU chroma vừa nêu có thể được phân chia tiếp thành hai CU chroma bằng phân chia cây tam phân BT (Binary-Tree - BT). Ví dụ, có thể được phân chia thành hai CU chroma 4x4 hoặc hai CU chroma 8x2. Tương tự, nếu kích cỡ SCIPU/cây lưỡng cực bộ là 8x16 theo kích cỡ mẫu luma, tương ứng là 4x8 theo kích cỡ mẫu chroma, nó có thể được phân chia tiếp thành hai CU chroma 4x4 hoặc hai CU chroma. Để đạt được điều đó, theo một phương án, điều kiện ràng buộc phân chia BT được sửa đổi. Khi dạng cây của nút/CU hiện thời bằng DUAL\_TREE\_CHROMA và SCIPU modeType bằng MODE\_TYPE\_INTRA, và  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$  bằng 32, hoặc lớn hơn hoặc bằng 32, BT sẽ không phải bị hủy bỏ (tức là, BT có thể được sử dụng). Tuy nhiên, khi dạng cây của nút/CU hiện thời bằng DUAL\_TREE\_CHROMA và SCIPU modeType bằng MODE\_TYPE\_INTRA, và  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC) / 2$  nhỏ hơn 16 mẫu, BT bị hủy bỏ. Theo ví dụ khác, khi dạng cây của nút/CU hiện thời bằng DUAL\_TREE\_CHROMA và SCIPU modeType bằng MODE\_TYPE\_INTRA, và  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$  nhỏ hơn hoặc bằng 16, BT bị hủy bỏ.

Theo một phương án, một phần của cú pháp nguyên thủy được thể hiện dưới đây. Biến allowBtSplit được suy dẫn như sau:

Nếu một hoặc nhiều trong số những điều kiện sau là true, allowBtSplit được thiết lập là FALSE:

- cbSize 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  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$  nhỏ hơn hoặc bằng 16
- treeType bằng DUAL\_TREE\_CHROMA và modeType bằng MODE\_TYPE\_INTRA.

Cú pháp có thể được sửa đổi như dưới đây. Biến allowBtSplit được suy dẫn như sau:

Nếu một hoặc nhiều trong số những điều kiện sau là true, allowBtSplit được thiết lập là FALSE:

- cbSize 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  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$  nhỏ hơn hoặc bằng 16
- treeType bằng DUAL\_TREE\_CHROMA và modeType bằng MODE\_TYPE\_INTRA và the  $(cbWidth / SubWidthC) * (cbHeight / SubHeightC)$  nhỏ hơn hoặc bằng 16.

Theo phương án khác, hai điều kiện sau được thêm vào:

- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và  $cbWidth / SubWidthC$  nhỏ hơn hoặc bằng 4
- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và modeType bằng MODE\_TYPE\_INTRA và  $cbWidth / SubWidthC$  nhỏ hơn hoặc bằng 4.

Theo phương án khác, hai điều kiện sau được thêm vào:

- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và  $cbWidth / SubWidthC$  nhỏ hơn hoặc bằng 4 và  $cbHeight / SubHeightC$  nhỏ hơn hoặc bằng 8

- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và modeType bằng MODE\_TYPE\_INTRA và cbWidth / SubWidthC nhỏ hơn hoặc bằng 4 và cbHeight / SubHeightC nhỏ hơn hoặc bằng 8.

Ngoài ra, cú pháp có thể được sửa đổi như dưới đây. Biến allowBtSplit được suy dẫn như sau:

Nếu một hoặc nhiều trong số những điều kiện sau là true, allowBtSplit được thiết lập là FALSE:

- cbSize 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 ) nhỏ hơn hoặc bằng 16
- // treeType bằng DUAL\_TREE\_CHROMA và modeType bằng MODE\_TYPE\_INTRA //, trong đó ký tự chữ giữa cặp gạch chéo ngược “//” biểu thị phép xóa.

Theo phương án khác, một điều kiện sau được thêm vào:

- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và cbWidth / SubWidthC nhỏ hơn hoặc bằng 4.

Theo phương án khác, một điều kiện sau được thêm vào:

- btSplit bằng SPLIT\_BT\_VER và treeType bằng DUAL\_TREE\_CHROMA và cbWidth / SubWidthC nhỏ hơn hoặc bằng 4 và cbHeight / SubHeightC nhỏ hơn hoặc bằng 8.

Theo đó, đã đề xuất cho phép CU chroma được phân chia tiếp (ví dụ phân chia BT) theo SCIPU dạng Intra hoặc theo cây lưỡng cực bộ khi kích cỡ CU chroma là 32 mẫu chroma như được mô tả ở trên.

Bất kỳ phương pháp nào trong số các phương pháp đã đề xuất ở trên có thể được thực hiện trong các bộ mã hóa và/hoặc các bộ giải mã. Ví dụ, bất kỳ phương pháp nào trong số các phương pháp đã đề xuất có thể được thực hiện theo dự đoán hoặc phân vùng

khối Intra của bộ mã hóa và/hoặc bộ giải mã. Ngoài ra, bất kỳ phương pháp nào trong số các phương pháp đã đề xuất có thể được thực hiện dưới dạng vi mạch được tích hợp vào phép dự đoán hoặc phân vùng khối Intra của bộ mã hóa và/hoặc bộ giải mã, để cấp thông tin cần cho dự đoán hoặc phân vùng khối Intra.

Fig. 7 minh họa lưu đồ của hệ thống mã hóa và giải mã được lấy làm ví dụ phép chọn cơ chế ràng buộc cho các khối nhỏ theo phương án của sáng chế. Các bước được thể hiện trên lưu đồ, cũng như các lưu đồ khác trong bản mô tả này, có thể được thực hiện dưới dạng các mã chương trình có thể xử lý bằng một hoặc nhiều bộ xử lý (ví dụ, một hoặc nhiều CPU) tại bộ mã hóa và/hoặc bộ giải mã. Các bước được thể hiện trên lưu đồ có thể còn được thực hiện dựa vào phần cứng chẳng hạn một hoặc nhiều thiết bị điện tử hoặc bộ xử lý được bố trí để thực hiện các bước trên lưu đồ. Theo phương pháp này, khối cơ sở được nhận tại bộ mã hóa hoặc dữ liệu nén chứa khối cơ sở được nhận tại bộ giải mã ở bước 710, trong đó khối cơ sở bao gồm một hoặc nhiều khối con được tạo ra từ quá trình phân chia vùng lập mã bằng cách sử dụng cây phân vùng đơn và khối cơ sở bao gồm một thành phần luma và một hoặc nhiều thành phần chroma theo định dạng chroma 420 hoặc 422. Loại cơ chế mục tiêu cho toàn bộ một hoặc nhiều khối con thuộc trong khối cơ sở ở bước 720 khi một hoặc nhiều điều kiện thỏa mãn, trong đó một hoặc nhiều điều kiện vừa nêu bao gồm chiều rộng thứ nhất của một khối con bằng 2 đối với một hoặc nhiều thành phần chroma, và trong đó nếu cơ chế Intra không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế Inter và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra. Một hoặc nhiều khối con vừa nêu trong khối cơ sở được mã hóa theo loại cơ chế mục tiêu tại bộ mã hóa hoặc một hoặc nhiều khối con vừa nêu trong khối cơ sở được giải mã theo loại cơ chế mục tiêu tại bộ giải mã ở bước 730.

Fig. 8 minh họa lưu đồ khác của hệ thống lập mã được lấy làm ví dụ có phép chọn cơ chế ràng buộc cho các khối nhỏ theo phương án của sáng chế. Theo phương pháp này, khối cơ sở được nhận tại bộ mã hóa hoặc dữ liệu nén chứa khối cơ sở được nhận tại bộ giải mã ở bước 810, trong đó khối cơ sở bao gồm một hoặc nhiều khối con được tạo ra từ quá trình phân chia vùng lập mã bằng cách sử dụng cây phân vùng đơn và khối cơ sở bao gồm một thành phần luma và một hoặc nhiều thành phần chroma theo định dạng chroma 420 hoặc 422. Loại cơ chế mục tiêu cho toàn bộ một hoặc nhiều khối con thuộc trong khối cơ sở ở bước 820 khi một hoặc nhiều điều kiện thỏa mãn, trong đó một hoặc

nhiều điều kiện vừa nêu bao gồm kích cỡ khối con thứ nhất nhỏ hơn 16 đối với một hoặc nhiều thành phần chroma, và trong đó nếu cơ chế Intra không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế Inter và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra. Một hoặc nhiều khối con vừa nêu trong khối cơ sở được mã hóa theo loại cơ chế mục tiêu tại bộ mã hóa hoặc một hoặc nhiều khối con vừa nêu trong khối cơ sở được giải mã theo loại cơ chế mục tiêu tại bộ giải mã ở bước 830.

Các lưu đồ được thể hiện là nhằm minh họa ví dụ về lập mã video theo sáng chế. Người có chuyên môn trong lĩnh vực kỹ thuật này có thể sửa đổi từng bước, sắp xếp lại các bước, phân chia bước, hoặc kết hợp các bước để thực hiện sáng chế mà không trệtch khỏi nội dung kỹ thuật của sáng chế. Ở bản mô tả này, cú pháp và các ngữ nghĩa cụ thể được sử dụng để minh họa các ví dụ thực hiện các phương án của sáng chế. Người có chuyên môn có thể thực hiện sáng chế bằng cách thay thế cú pháp và các ngữ nghĩa vừa nêu bằng cú pháp và các ngữ nghĩa tương đương mà không trệtch khỏi nội dung kỹ thuật của sáng chế.

Phần mô tả vừa nêu được đưa ra nhằm cho phép người có kiến thức trung bình trong cùng lĩnh vực kỹ thuật này thực hiện sáng chế như được trình bày trong phần diễn tả ứng dụng cụ thể và yêu cầu của nó. Nhiều sửa đổi khác nhau cho các phương án đã được mô tả sẽ là điều hiển nhiên đối với những chuyên gia trong cùng lĩnh vực kỹ thuật này, và các nguyên lý chung được định nghĩa ở bản mô tả này có thể được áp dụng cho các phương án khác. Vì vậy, sáng chế không có ý định bị giới hạn ở các phương án cụ thể đã được thể hiện và được mô tả, mà được ý định là tuân theo phạm vi rộng nhất phù hợp với những nguyên lý và dấu hiệu mới được bộc lộ ở bản mô tả này. Ở phần mô tả chi tiết sáng chế ở trên, nhiều phần chi tiết cụ thể khác nhau được minh họa nhằm giúp hiểu toàn bộ sáng chế. Tuy nhiên, sẽ được hiểu bởi những chuyên gia trong cùng lĩnh vực kỹ thuật này rằng sáng chế có thể được thực hiện.

Phương án của sáng chế như được mô tả ở trên có thể được thực hiện theo nhiều lập trình phần cứng, phần mềm khác nhau hoặc kết hợp cả phần cứng và phần mềm. Ví dụ, phương án của sáng chế có thể là một hoặc nhiều vi mạch được tích hợp vào chip nén video hoặc mã chương trình được tích hợp vào phần mềm nén video để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Phương án của sáng chế có thể còn là mã chương trình được xử lý trên bộ xử lý tín hiệu số DSP (Digital Signal Processor - DSP)

để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Sáng chế có thể còn liên quan đến số lượng chức năng được thực hiện bởi bộ xử lý máy tính, bộ xử lý tín hiệu số, bộ vi xử lý, hoặc vi mạch tích hợp FPGA (Field Programmable Gate Array-FPGA). Những bộ xử lý này có thể được cấu hình để thực hiện những nhiệm vụ cụ thể theo sáng chế, bằng cách xử lý mã phần mềm hoặc mã vi chương trình có thể đọc được bằng máy mà định ra các phương pháp thực hiện của sáng chế. Mã phần mềm hoặc mã vi chương trình có thể được phát triển theo các ngôn ngữ lập trình khác và các định dạng hoặc cách thức khác. Mã phần mềm có thể còn được tương thích với nhiều nền tảng khác nhau. Tuy nhiên, các định dạng, cách thức và ngôn ngữ mã khác nhau của các mã chương trình và các phương thức cấu hình mã khác để thực hiện những nhiệm vụ theo sáng chế sẽ không trệch khỏi nội dung và phạm vi của sáng chế.

Sáng chế có thể được thực hiện theo các dạng cụ thể khác mà không trệch khỏi nội dung và các đặc tính cơ bản. Các ví dụ đã mô tả được xem xét theo toàn bộ các khía cạnh chỉ là nhằm minh họa chứ không nhằm giới hạn. Vì vậy, phạm vi của sáng chế được thể hiện theo các điểm yêu cầu bảo hộ chứ không phải phần mô tả đã nêu ở trên. Mọi thay đổi mà nằm trong nội dung ý nghĩa và phạm vi nghĩa tương đương của các điểm yêu cầu bảo hộ đều được xem là thuộc phạm vi của sáng chế.

Sáng chế này hưởng quyền ưu tiên của đơn xin cấp bằng sáng chế Hoa Kỳ số 62/890,723 nộp ngày 23/08/2019 và đơn xin cấp bằng sáng chế Hoa Kỳ số 62/906,435 nộp ngày 26/09/2019. Nội dung của chúng được tham khảo kết hợp ở đây.

## YÊU CẦU BẢO HỘ

### 1. Phương pháp lập mã video, phương pháp này bao gồm:

nhận khối cơ sở tại bộ mã hóa hoặc nhận dữ liệu nén chứa khối cơ sở tại bộ giải mã, trong đó khối cơ sở bao gồm một hoặc nhiều khối con được tạo ra từ quá trình phân chia vùng lập mã bằng cách sử dụng cơ chế phân vùng và khối cơ sở bao gồm một thành phần chói (luma) và một hoặc nhiều thành phần màu (chroma) theo định dạng chroma 420 hoặc 422;

xác định loại cơ chế mục tiêu cho toàn bộ một hoặc nhiều khối con thuộc trong khối cơ sở khi một hoặc nhiều điều kiện được thỏa mãn, trong đó một hoặc nhiều điều kiện vừa nêu bao gồm chiều rộng thứ nhất của một khối con bằng 2 đối với một hoặc nhiều thành phần chroma, và trong đó nếu cơ chế nội ảnh (Intra) được suy luận không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế liên ảnh (Inter) và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra được suy luận; và

mã hóa một hoặc nhiều khối con vừa nêu trong khối cơ sở theo loại cơ chế mục tiêu tại bộ mã hóa hoặc áp dụng phép giải mã đối với dữ liệu nén theo loại cơ chế mục tiêu để tái tạo một hoặc nhiều khối con vừa nêu trong khối cơ sở tại bộ giải mã.

2. Phương pháp theo điểm 1, trong đó chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng nhị phân.

3. Phương pháp theo điểm 1, trong đó chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai của khối cơ sở cho một thành phần luma vừa nêu bằng 16 và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng tam phân.

4. Phương pháp theo điểm 1, trong đó khi khối cơ sở có định dạng chroma 420, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và 16 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng nhị phân, chiều rộng thứ hai và chiều cao của khối cơ sở

cho một thành phần luma vừa nêu bằng 16 và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng tam phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 32 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng ngang nhị phân hoặc tam phân, hoặc chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 64 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân chia ngang tam phân.

5. Phương pháp theo điểm 1, trong đó khi khối cơ sở có định dạng chroma 422, chiều rộng thứ nhất của một khối con đối với một hoặc nhiều thành phần chroma bằng 2 tương ứng với một điều kiện mà chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 8 và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng nhị phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 16 và 8 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng đứng tam phân, chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 16 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân vùng ngang nhị phân hoặc tam phân, hoặc chiều rộng thứ hai và chiều cao của khối cơ sở cho một thành phần luma vừa nêu bằng 4 và 32 một cách tương ứng và cơ chế phân vùng cho khối cơ sở bao gồm phân chia ngang tam phân.

6. Phương pháp theo điểm 1, trong đó nếu cơ chế Intra được suy luận không được chọn cho vùng hình ảnh bao quanh khối cơ sở, cú pháp thứ nhất được báo hiệu tại bộ mã hóa hoặc được phân tích tại bộ giải mã để báo hiệu hoặc cơ chế Intra hoặc cơ chế Inter được chọn cho khối cơ sở.

7. Phương pháp theo điểm 1, trong đó vùng lập mã tương ứng là đơn vị cây lập mã.

8. Phương pháp theo điểm 1 trong đó vùng hình ảnh tương ứng là slice.

9. Phương pháp theo điểm 8, trong đó cú pháp thứ hai được báo hiệu tại bộ mã hóa hoặc được phân tích tại bộ giải mã ở tiêu đề slice để biểu thị dạng lập mã của slice.

10. Thiết bị lập mã video, thiết bị này bao gồm một hoặc nhiều mạch điện tử hoặc bộ xử lý được bố trí để:

nhận khối cơ sở tại bộ mã hóa hoặc nhận dữ liệu nén chứa khối cơ sở tại bộ giải mã, trong đó khối cơ sở bao gồm một hoặc nhiều khối con được tạo ra từ quá trình phân chia vùng lập mã bằng cách sử dụng cơ chế phân vùng và khối cơ sở bao gồm một thành phần

luma và một hoặc nhiều thành phần chroma theo định dạng chroma 420 hoặc 422;

xác định loại cơ chế mục tiêu cho toàn bộ một hoặc nhiều khối con thuộc trong khối cơ sở khi một hoặc nhiều điều kiện thỏa mãn, trong đó một hoặc nhiều điều kiện vừa nêu bao gồm chiều rộng thứ nhất của một khối con bằng 2 đối với một hoặc nhiều thành phần chroma, và trong đó nếu cơ chế Intra được suy luận không được chọn cho vùng hình ảnh bao quanh khối cơ sở, thì loại cơ chế mục tiêu hoặc là cơ chế Intra hoặc là cơ chế Inter và nếu không thì, loại cơ chế mục tiêu là cơ chế Intra được suy luận; và

mã hóa một hoặc nhiều khối con vừa nêu trong khối cơ sở theo loại cơ chế mục tiêu tại bộ mã hóa hoặc áp dụng phép giải mã đối với dữ liệu nén theo loại cơ chế mục tiêu để tái tạo một hoặc nhiều khối con vừa nêu trong khối cơ sở tại bộ giải mã.

1/5

Fig. 1

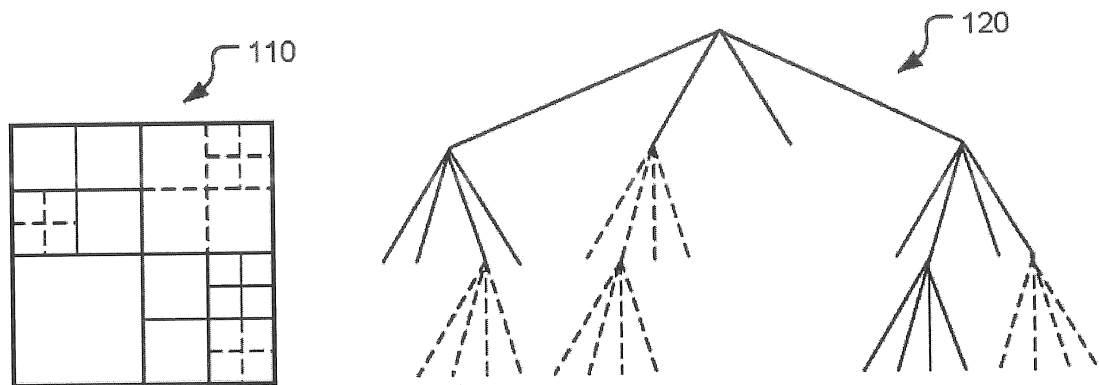
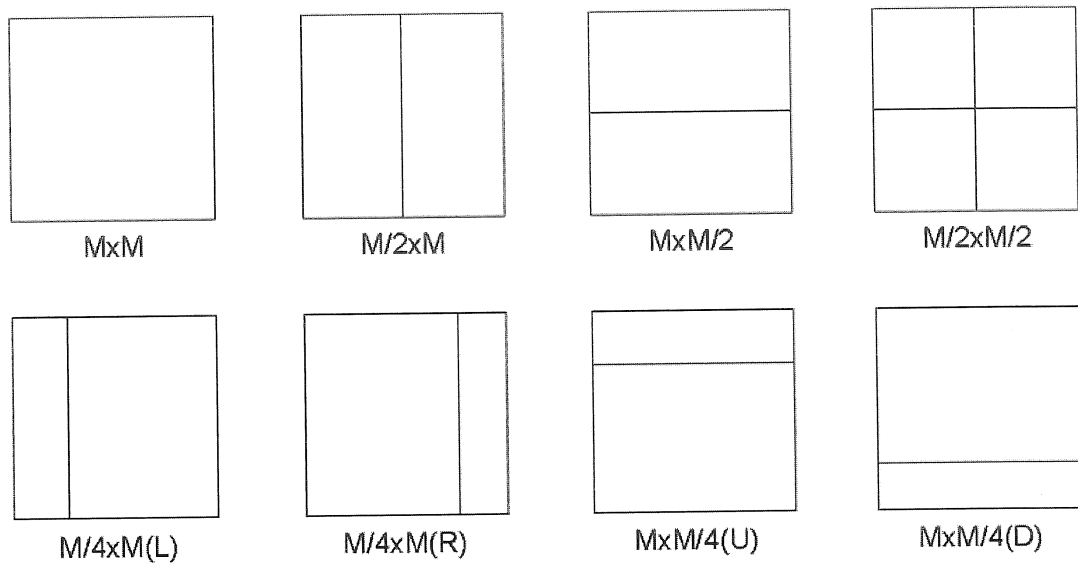


Fig. 2



2/5

Fig. 3

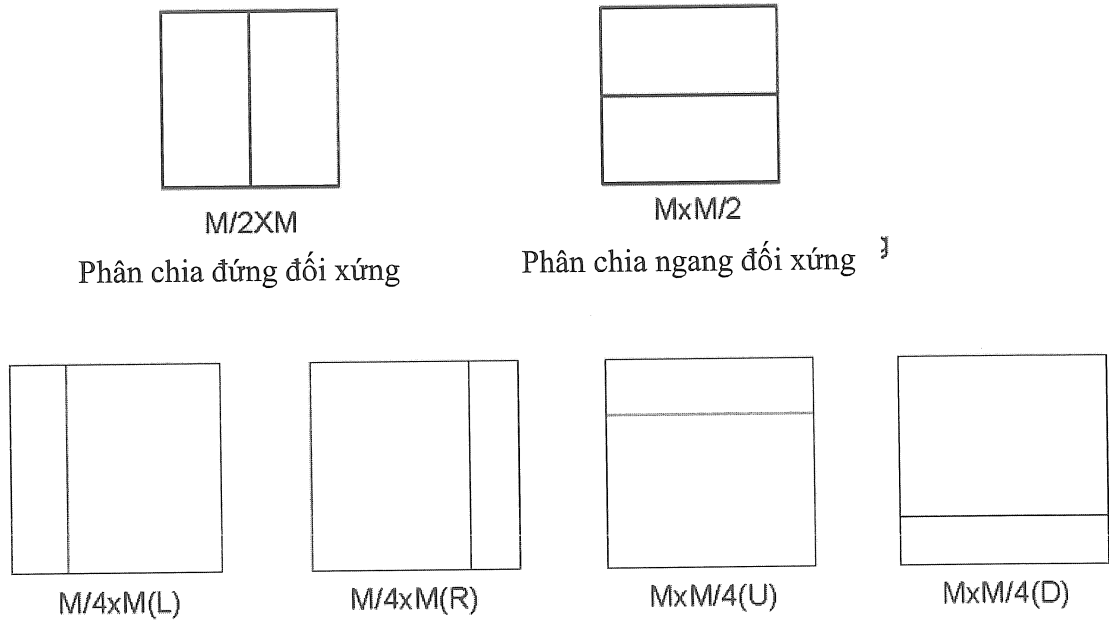
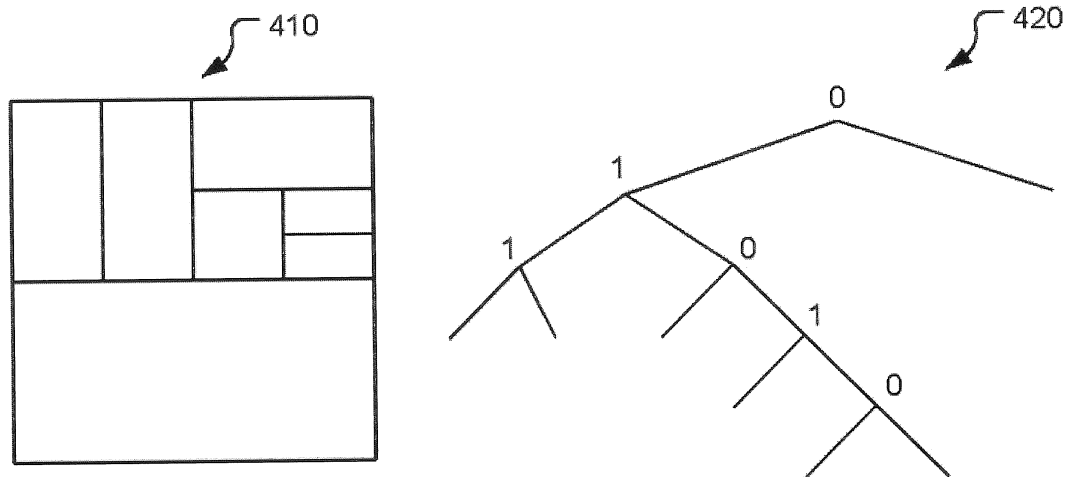


Fig. 4



3/5

Fig. 5

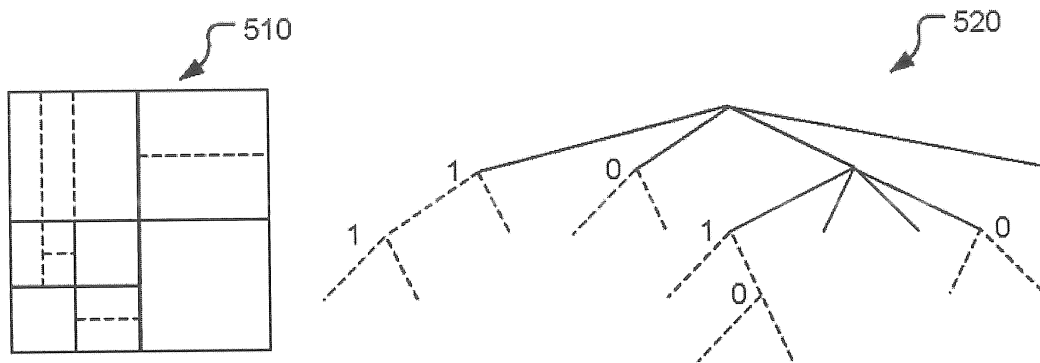
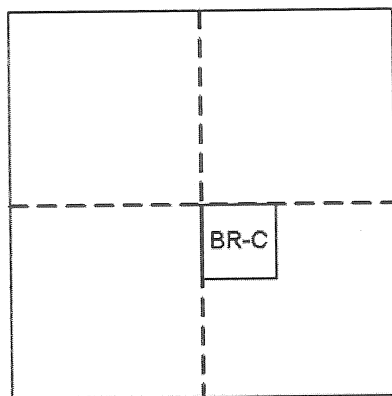


Fig. 6



4/5

Fig. 7

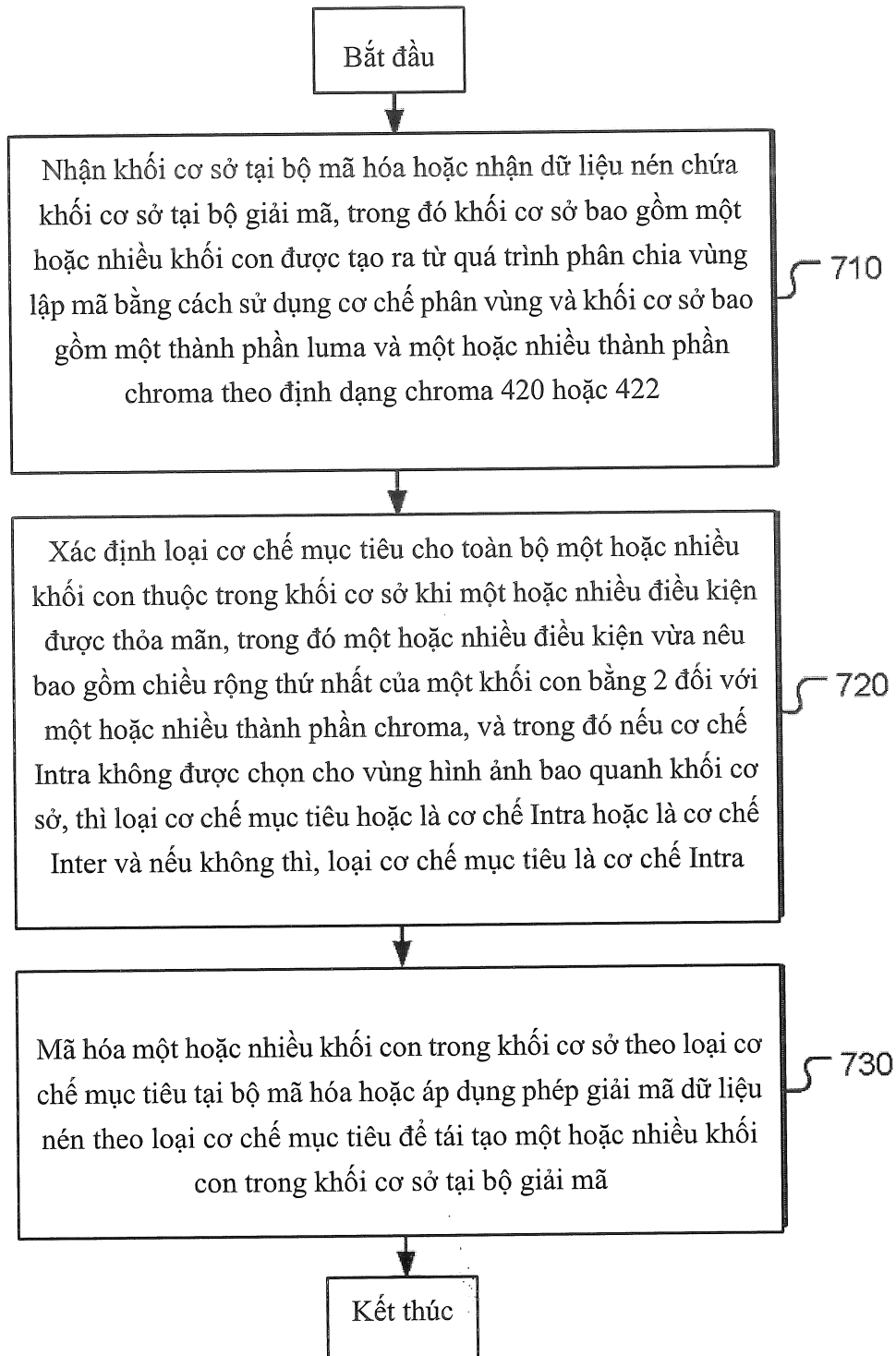


Fig. 8

