



(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)



1-0043335

(51)^{2020.01} H04N 19/124

(13) B

(21) 1-2021-01340

(22) 16/08/2019

(86) PCT/CN2019/101047 16/08/2019

(87) WO2020/035057 20/02/2020

(30) 62/764,872 16/08/2018 US; 62/727,577 06/09/2018 US; 62/786,317 28/12/2018 US

(45) 25/02/2025 443

(43) 25/05/2021 398

(73) HFI Innovation Inc. (TW)

3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan

(72) CHUANG, Tzu-Der (CN); HSU, Chih-Wei (CN); CHEN, Ching-Yeh (CN).

(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DỮ LIỆU VIDEO TRONG HỆ THỐNG
GIẢI MÃ VIDEO, VÀ PHƯƠNG TIỆN CÓ THỂ ĐỌC ĐƯỢC BẰNG MÁY TÍNH
KHÔNG TẠM THỜI ĐỂ LƯU LỆNH CHƯƠNG TRÌNH

(21) 1-2021-01340

(57) Sáng chế đề cập đến phương pháp và thiết bị xử lý dữ liệu video trong hệ thống giải mã video. Các phương pháp và các thiết bị xử lý dữ liệu video bao gồm nhận dữ liệu vào liên quan đến nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia, xác định chiều cao nút phân chia hiện thời theo dạng phân chia, và so sánh chiều cao nút phân chia hiện thời với chiều cao báo hiệu delta QP lớn nhất. Hệ thống giải mã video suy dẫn delta QP từ một hoặc nhiều thành phần cú pháp được báo hiệu trong TU liên quan đến nút phân chia hiện thời theo kết quả so sánh, khôi phục lại QP cuối cùng cho nút phân chia hiện thời dựa vào QP tham chiếu và delta QP, và giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời bằng cách sử dụng QP cuối cùng. Chiều cao được tính theo cách thức trong đó xét các định dạng phân chia và các vùng phân chia khác nhau. Ngoài ra, sáng chế còn đề cập đến phương tiện có thể đọc được bằng máy tính không tạm thời để lưu lệnh chương trình giúp cho mạch xử lý của thiết bị thực hiện phương pháp xử lý video.

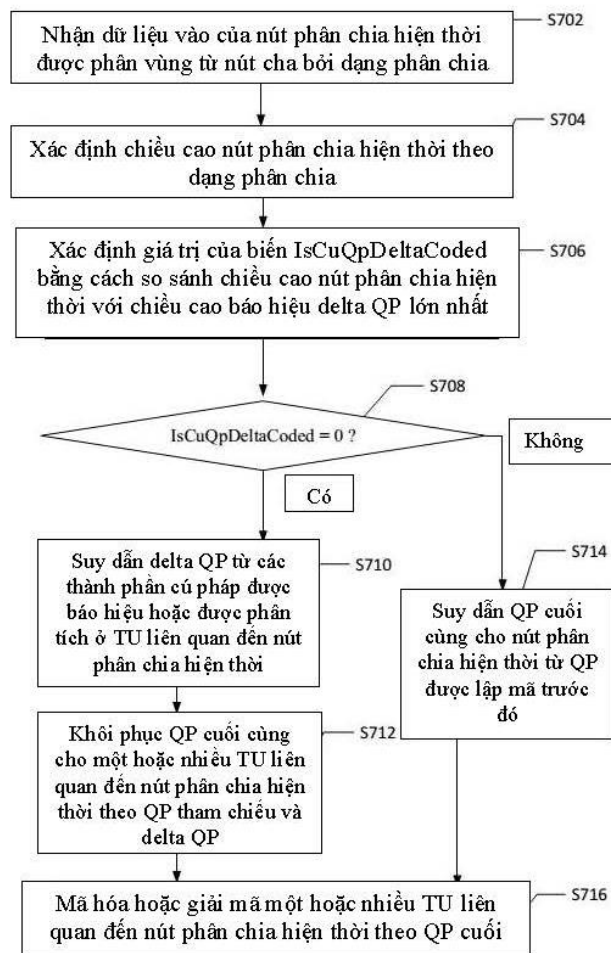


Fig. 7

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các phương pháp và các thiết bị xử lý dữ liệu video dùng để mã hóa video hoặc giải mã video. Cụ thể hơn là, sáng chế đề cập đến tham số lượng tử hóa báo hiệu cho dữ liệu video được phân vùng thành các khối vuông và không vuông.

Tình trạng kỹ thuật của sáng chế

Chuẩn mã hóa video hiệu suất cao HEVC (High-Efficiency Video Coding - HEVC) là một chuẩn mã hóa video mới nhất được phát triển bởi nhóm hợp tác chung về mã hóa video JCT-VC (Joint Collaborative Team on Video Coding - JCT-VC) gồm những chuyên gia mã hóa video đến từ Nhóm nghiên cứu thuộc ITU-T. Chuẩn HEVC dựa trên kiến trúc mã hóa trên nền khối mạch trong đó phân chia mỗi mảng (slice) thành nhiều đơn vị cây mã hóa CTU (Coding Tree Unit – CTU), về mặt thuật ngữ được gọi là đơn vị mã hóa CU (Đơn vị mã hóa - CU), là khối điểm ảnh vuông có kích thước $2N \times 2N$, và mỗi CU có thể được chia đệ quy thành bốn CU nhỏ hơn cho đến khi kích cỡ nhỏ nhất định trước đạt được. Mỗi CU bao gồm một hoặc nhiều đơn vị dự đoán PU (Đơn vị dự đoán - PU). Mỗi PU tương đương với một khối các điểm ảnh trong CU. Chuẩn HEVC. Trong kiến trúc chính HEVC, các kích cỡ lớn nhất và nhỏ nhất của CTU được định bằng các thành phần cú pháp được báo hiệu trong bộ tham số chuỗi SPS (Sequence Parameter Set - SPS) trong số các kích cỡ 8×8 , 16×16 , 32×32 , và 64×64 . Các CTU trong một mảng (slice) được xử lý theo lệnh quét mảng. Mỗi CTU còn được phân chia đệ quy thành một hoặc nhiều đơn vị mã hóa (nhiều CU) theo phương pháp phân vùng cây tứ phân QT (Quadtree – QT) để tương thích với nhiều đặc tính cục bộ khác nhau. Kích cỡ của CTU có thể là 64×64 , 32×32 , hoặc 16×16 , và mỗi CTU có thể là một CU đơn nhất hoặc được chia thành bốn đơn vị nhỏ hơn có các kích cỡ bằng nhau là $M/2 \times M/2$, là các nút chia của cây mã hóa. Nếu các đơn vị là các nút lá của cây mã hóa, thì các đơn vị đó trở thành các CU, nếu không thì, quá trình phân chia cây tứ phân sẽ phân vùng từng đơn vị một cách lặp đi lặp lại cho đến khi kích cỡ của nút đạt kích cỡ CU cho phép nhỏ nhất được định trong SPS. Ví dụ về cấu trúc phân

vùng khối cây tứ phân cho CTU được minh họa trên Fig. 1, trong đó các đường nét liền biểu thị các đường biên giới CU trong CTU 100.

Quyết định dự đoán được thực hiện tại cấp CU, trong đó mỗi CU hoặc được lập mã bằng dự đoán liên ảnh (theo thời gian) hoặc dự đoán nội ảnh (theo không gian). Vì kích cỡ CU cho phép nhỏ nhất là 8×8 , nên độ chia nhất dùng để chuyển các dạng dự đoán cơ bản khác nhau là 8×8 . Mỗi khi quá trình chia cây phân cấp CU được thực hiện xong, mỗi CU còn được chia tiếp thành một hoặc nhiều đơn vị dự đoán (nhiều PU) theo dạng phân vùng PU dùng để dự đoán trong HEVC. Fig. 2 thể hiện tám dạng phân vùng PU được định nghĩa trong chuẩn HEVC. Mỗi CU được chia thành một, hai, hoặc bốn PU theo một trong số tám dạng phân vùng PU được thể hiện trên Fig. 2. PU đóng vai trò là khối đại diện cơ bản để chia sẻ thông tin dự đoán khi quá trình dự đoán giống nhau được áp dụng cho toàn bộ các điểm ảnh trong PU và thông tin liên quan quá trình dự đoán đang truyền đến bộ giải mã trên nền tảng PU. Sau khi có được tín hiệu dư thừa được tạo ra bởi quá trình dự đoán dựa vào dạng phân chia PU, dữ liệu của tín hiệu dư thừa thuộc CU được phân chia tiếp thành một hoặc nhiều đơn vị chuyển đổi (nhiều TU) theo cấu trúc phân vùng khối QT khác để chuyển dữ liệu thừa thành các hệ số chuyển đổi để nén dữ liệu. Các đường nét đứt trên Fig. 1 biểu thị các đường biên giới TU. TU là khối đại diện cơ bản dùng để thực hiện phép biến đổi và lượng tử hóa số nguyên để thực hiện phép biến đổi số nguyên và lượng tử hóa dữ liệu thừa. Đối với mỗi TU, một ma trận biến đổi có cùng kích cỡ như TU được áp dụng cho dữ liệu thừa để tạo ra các hệ số chuyển đổi, và các hệ số chuyển đổi này được lượng tử hóa và được chuyển đến bộ giải mã trên nền tảng TU.

Các thuật ngữ như khối cây mã hóa CBT (Coding Tree Block - CTB), khối mã hóa 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 dãy mẫu hai chiều của một thành phần màu liên quan đến CTU, CU, PU, và TU một cách tương ứng. Ví dụ, CTU bao gồm một CTB luma (một luma), hai CTB chroma (hai chroma), và các thành phần cú pháp liên quan của nó. Trong hệ thống HEVC, cấu trúc phân vùng khối cây tứ phân giống nhau thường được áp dụng cho cả thành phần luma và thành phần chroma trừ khi kích cỡ nhỏ nhất đối với khối màu (khối chroma) đạt được.

Phân vùng cây nhị phân ngoài phân vùng cây tứ phân, một phương pháp phân vùng thay thế khác được gọi là phân vùng khối cây nhị phân BT (Binary Tree - BT),

trong đó khối được chia lặp đi lại thành hai khối nhỏ hơn. Phương pháp phân vùng cây nhị phân đơn giản nhất chỉ cho phép phân chia ngang đối xứng và phân chia đứng đối xứng. Đối với khối đã cho có kích cỡ $M \times N$, cờ được báo hiệu để biểu thị liệu khối đã cho được chia thành hai khối nhỏ hơn hay không, nếu cờ có biểu thị sự phân chia vừa nêu, thì thành phần cú pháp khác được báo hiệu để biểu thị dạng phân chia nào được sử dụng. Trong trường hợp của phân chia ngang, khối đã cho được chia thành hai khối có kích cỡ $M \times N/2$; và trong trường hợp của phân chia đứng, khối đã cho được chia thành hai khối có kích cỡ $M/2 \times N$. Phương pháp phân vùng BT có thể được sử dụng để phân vùng slice thành CTU, CTU thành các CU, CU thành các PU, hoặc CU thành các TU. Ví dụ về phương pháp phân vùng BT mà phân chia CTU thành các CU, và nhằm đơn giản, sẽ không có phân vùng tiếp nữa từ CU thành PU hoặc từ CU thành TU, đó là nút lá của cây nhị phân là đơn vị cơ bản để lập trình dự đoán và chuyển đổi.

Phân vùng QTBT mặc dù phương pháp phân vùng cây nhị phân hỗ trợ nhiều dạng phân vùng hơn và do đó linh hoạt hơn phương pháp phân vùng cây tứ phân, song tổng thể mức độ phức tạp và báo hiệu lập trình cao hơn để lựa chọn dạng phân vùng tốt nhất trong số tất cả các dạng phân vùng khả thi. Một phương pháp phân vùng kết hợp được gọi là cấu trúc cây nhị phân-cây tứ phân QTBT (Quad-Tree-Binary-Tree - QTBT) sẽ kết hợp phương pháp phân vùng cây tứ phân với phương pháp phân vùng cây nhị phân, điều đó giúp cân bằng hiệu quả lập trình và độ phức tạp lập trình của hai phương pháp phân vùng đã nêu. Cấu trúc QTBT được lấy làm ví dụ được thể hiện trên Fig. 3A, trong đó khối lớn chẳng hạn CTU trước tiên được phân vùng bởi phương pháp phân vùng cây tứ phân rồi đến phương pháp phân vùng cây nhị phân. Phân chia QT được thực hiện lặp đi lặp lại cho đến khi kích cỡ của khối phân chia đạt kích cỡ nút lá tứ phân cho phép nhỏ nhất, và nếu khối cây tứ phân lá không lớn hơn kích cỡ nút gốc cây nhị phân cho phép lớn nhất, thì khối cây tứ phân lá có thể được phân vùng tiếp nữa bởi phân chia BT. Phân chia BT được thực hiện lặp đi lặp lại cho đến khi kích cỡ, chiều rộng, hoặc chiều cao của khối phân chia đạt kích cỡ, chiều rộng, hoặc chiều cao nút lá cây nhị phân cho phép nhỏ nhất, hoặc chiều cao cây nhị phân đạt chiều cao cây nhị phân cho phép lớn nhất. Fig. 3A minh họa ví dụ về cấu trúc phân vùng khối theo phương pháp phân vùng QTBT và Fig. 3B minh họa sơ đồ cây mã hóa cho cấu trúc phân vùng khối QTBT được thể hiện trên Fig. 3A. Các đường nét liền trên Fig 3A và Fig 3B biểu thị quá trình phân chia cây tứ phân trong khi các đường nét đứt biểu thị

phân chia cây nhị phân. Ở mỗi nút phân chia (tức là, nút không lá) của cấu trúc cây tam phân, một cờ biểu thị dạng phân chia nào (phân chia ngang đối xứng hoặc phân chia đứng đối xứng) được sử dụng, 0 biểu thị phân chia ngang và 1 biểu thị phân chia đứng. Có thể đơn giản hóa quá trình phân vùng bằng cách bỏ phân chia từ CU thành PU và từ CU thành TU, khi các nút lá của cấu trúc phân vùng khối cây nhị phân là khối đại diện cơ bản cho cả lập trình dự đoán và chuyển đổi. Ví dụ, cấu trúc QTBT được thể hiện trên Fig. 3A phân chia khối lớn thành nhiều khối nhỏ hơn, và các khối nhỏ hơn này được xử lý bởi lập trình dự đoán và chuyển đổi mà không có thêm sự phân chia tiếp.

Một ví dụ cụ thể của cấu trúc QTBT được sử dụng để phân chia CTU có kích cỡ 128x128 thành nhiều CU, trong đó kích cỡ nút lá cây tứ phân cho phép nhỏ nhất bằng 16x16, kích cỡ nút gốc cây nhị phân cho phép lớn nhất bằng 64x64, chiều rộng và chiều cao nút lá cây nhị phân cho phép nhỏ nhất bằng 4, và chiều cao cây nhị phân cho phép lớn nhất bằng 4. CTU trước tiên được phân chia bởi cấu trúc phân vùng cây tứ phân và đơn vị cây tứ phân lá có kích cỡ nằm trong khoảng từ 16x16 đến 128x128, và nếu đơn vị cây tứ phân lá là 128x128, nó không được phân chia tiếp bởi quá trình phân chia cây nhị phân bởi vì kích cỡ đó vượt quá kích cỡ nút gốc cây nhị phân cho phép lớn nhất 64x64, nếu không thì đơn vị cây tứ phân lá có thể được phân vùng tiếp nữa bởi phân chia cây nhị phân. Đơn vị cây tứ phân lá cũng có thể là đơn vị cây nhị phân gốc, có chiều cao cây nhị phân bằng 0. Khi chiều cao cây nhị phân bằng 4, vốn là chiều cao cây nhị phân cho phép lớn nhất, không có phân chia nào được thực hiện, khi nút cây nhị phân có chiều rộng bằng 4, không có phân chia ngang nào được thực hiện, và khi nút cây nhị phân có chiều cao bằng 4, không có phân chia ngang nào được thực hiện. Các nút lá của cấu trúc QTBT được xử lý tiếp bởi lập trình dự đoán và chuyển đổi.

Cấu trúc QTBT được áp dụng riêng cho các thành phần luma và chroma ở các slice Intra (I), và được áp dụng đồng thời cho cả thành phần luma và chroma ở cả hai slice Predictive (P) và Bi-predictive (B), ngoại trừ khi các kích cỡ nhỏ nhất nhất định đạt được đối với các thành phần chroma. Ở các slice I, mỗi CTB luma có sự phân vùng khối theo cấu trúc QTBT trong khi hai CTB chroma tương ứng có phân vùng khối theo cấu trúc QTBT khác theo một phương án. Theo phương án khác, hai CTB

chroma tương ứng vừa nêu có thể cũng có phân vùng khối theo cấu trúc QTBT riêng của chúng.

Phân vùng cây tam phân: phương pháp phân vùng khác linh hoạt hơn nữa được gọi là phương pháp phân vùng cây tam phân (Ternary Tree - TT) được sử dụng để chụp những đối tượng mà nằm ở tâm khối trong khi phương pháp phân vùng cây tứ phân và phương pháp phân vùng cây nhị phân luôn luôn phân chia dọc theo tâm khối. Fig. 4 minh họa phân vùng cây tam phân đứng 42 và phân vùng cây tam phân ngang 44. Phương pháp phân vùng cây tam phân có thể giúp khả năng xác định vị trí các đối tượng nhỏ dọc theo các đường biên giới nhanh hơn, bằng cách cho phép phân vùng một phần tư theo phương thẳng đứng hoặc ngang.

Phân vùng MTT: phân vùng khối cây đa phân MTT (Multi-Type-Tree - MTT) mở rộng ý tưởng của cấu trúc cây hai cấp trong QTBT bằng cách cho phép cả hai phương pháp phân vùng cây nhị phân và cây tam phân ở cấp thứ hai của MTT. Hai cấp của các cây trong MTT được gọi là Cây vùng RT (Region Tree - RT) và Cây dự đoán PT (Prediction Tree - PT) một cách tương ứng. Ví dụ về MTT là Cây tứ phân QT cộng Cây đa phân MT (QTMT), khi cấp thứ nhất RT luôn là phân vùng QT, và cấp thứ hai PT có thể hoặc là phân vùng BT hoặc là phân vùng TT. Ví dụ, CTU trước tiên được phân vùng bởi RT, là phân vùng QT, và mỗi nút lá RT có thể được chia tiếp bởi PT, vốn hoặc là phân vùng BT hoặc là phân vùng TT. Khối được phân vùng bởi PT có thể được chia tiếp với PT cho đến khi chiều cao PT lớn nhất đạt được, ví dụ, khối có thể trước tiên được phân vùng bởi phân vùng BT đứng để tạo ra khối phụ trái và khối phụ phải, và khối phụ trái được chia tiếp bởi phân vùng TT ngang trong khi khối phụ phải được chia tiếp bởi phân vùng BT ngang. Nút lá PT là đơn vị mã hóa cơ bản (CU) cho quá trình dự đoán và biến đổi và sẽ không được chia tiếp.

Fig. 5 minh họa ví dụ về báo hiệu dạng cây dùng cho phân vùng khối theo phân vùng khối MTT. Báo hiệu RT có thể tương tự như báo hiệu cây tứ phân trong phân vùng khối QTBT. Để báo hiệu nút PT, một vùng trống phụ được báo hiệu để biểu thị liệu đó là phân vùng cây nhị phân hay phân vùng cây tam phân. Đối với phân chia khối bằng RT, vùng trống thứ nhất được báo hiệu để biểu thị liệu có phân chia RT khác hay không, nếu khối vừa nêu không được chia tiếp bởi RT (tức là vùng trống thứ nhất bằng 0), vùng trống thứ hai được báo hiệu để biểu thị liệu đó là phân chia PT hay không. Nếu khối vừa nêu cũng không được chia tiếp bởi PT (tức là vùng trống thứ hai

bằng 0), thì khối này là nút lá. Nếu khối vừa nêu sau đó được phân chia bởi PT (tức là vùng trống thứ hai bằng 1), vùng trống thứ ba được gửi để biểu thị phân vùng ngang hay đứng và sau đó là vùng trống thứ tư được gửi để phân biệt phân vùng BT hay TT.

Sau khi phân chia dữ liệu video thành những phân vùng khối MTT, các nút lá MTT là các CU được sử dụng để lập trình dự đoán và chuyển đổi. Tương tự QTBT, cấu trúc cây MTT được lập mã riêng cho các thành phần luma và chroma ở các slice I, và cấu trúc cây MTT được áp dụng đồng thời cho cả thành phần luma và chroma trong các slice P và B cho những trường hợp khi các kích cỡ nhỏ nhất nhất định đạt được đối với thành phần chroma. Nói cách khác, CTB luma có phân vùng khối theo cấu trúc MTT của nó trong khi hai CTB chroma tương ứng có phân vùng khối theo cấu trúc MTT khác. QTBT có thể được gọi là phân chia QT cộng phân chia BT hoặc phân chia QT cộng phân chia BT và phân chia TT theo những phần mô tả dưới đây.

Báo hiệu delta QP trong HEVC: trong HEVC, tham số lượng tử hóa (QP) được xác định bởi QP tham chiếu và delta QP khi tối ưu QP thích ứng hoặc tối ưu đa QP được kích hoạt. Delta QP được truyền thay vì QP để giảm lượng bit cần cho thông tin QP. Khối mã hóa CB (Coding block - CB) lớn hơn hoặc bằng kích cỡ nhỏ nhất cho báo hiệu delta QP có delta QP của riêng nó, trong khi CB nhỏ hơn kích cỡ nhỏ nhất chia sẻ delta QP với một hoặc nhiều CB khác. CB đơn nhất có delta QP của riêng nó hoặc số lượng các CB chia sẻ cùng delta QP được gọi là Nhóm lượng tử hóa QG (Quantization Group - QG) hoặc bộ lượng tử hóa.

Báo hiệu Delta QP dùng cho các thành phần luma được điều khiển bởi hai cờ, cờ kích hoạt `cu_qp_delta_enabled_flag` và cờ chiều cao `diff_cu_qp_delta_depth`. Cờ kích hoạt, `cu_qp_delta_enabled_flag`, được sử dụng để biểu thị báo hiệu delta QP được kích hoạt hoặc không được kích hoạt, và cờ chiều cao, `diff_cu_qp_delta_depth`, được sử dụng để thiết lập kích cỡ nhỏ nhất cho báo hiệu delta QP. Cờ kích hoạt có thể được báo hiệu ở cấp chuỗi ảnh, cấp khung ảnh, cấp slice, cấp tile, cấp hàng CTU. Cờ chiều cao `diff_cu_qp_delta_depth` được thể hiện trong cú pháp cấp cao, ví dụ trong PPS, và `cu_qp_delta_abs` có thể nằm trong cú pháp đơn vị chuyển đổi khi cờ kích hoạt `cu_qp_delta_enabled_flag` bằng 1. Cờ kích hoạt `cu_qp_delta_enabled_flag` bằng 0 định rõ cờ chiều cao `diff_cu_qp_delta_depth` không nằm trong PPS và `cu_qp_delta_abs` không nằm trong cú pháp đơn vị chuyển đổi. Cờ chiều cao `diff_cu_qp_delta_depth` định rõ sai biệt về chiều cao giữa kích cỡ CTB luma và kích

cỡ CB luma nhỏ nhất của các đơn vị mã hóa mà chứa `cu_qp_delta_abs` và `cu_qp_delta_sign_flag`. Giá trị của cờ chiều cao `diff_cu_qp_delta_depth` nằm trong khoảng từ 0 đến `log2_diff_max_min_luma_coding_block_size`, gồm cả giá trị của cờ chiều cao được suy bằng 0 khi cờ chiều cao này không có mặt. `log2_diff_max_min_luma_coding_block_size` định rõ sai biệt giữa kích cỡ CB luma lớn nhất và kích cỡ CB luma nhỏ nhất. Biến, `Log2MinCuQpDeltaSize`, được suy dẫn từ cờ chiều cao như sau:

$$\text{Log2MinCuQpDeltaSize} = \text{CtbLog2SizeY} - \text{diff_cu_qp_delta_depth}$$

Cú pháp của báo hiệu delta QP ở cấp TU là như sau:

```
if ( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {
    cu_qp_delta_abs
    if ( cu_qp_delta_abs )
        cu_qp_delta_sign_flag
}
```

trong đó `IsCuQpDeltaCoded` được thiết lập bằng 0 khi mã hóa hoặc giải mã CU như sau.

```
if ( cu_qp_delta_enabled_flag && log2CbSize >= Log2MinCuQpDeltaSize ) {
    IsCuQpDeltaCoded = 0
    CuQpDeltaVal = 0
}
```

trong đó `log2CbSize` là log 2 của kích cỡ nút phân chia.

Khi delta QP được sử dụng, QP cuối cùng cho khối mã hóa được suy dẫn dựa vào delta QP và QP tham chiếu được báo hiệu. QP tham chiếu của nhóm lượng tử hóa hiện thời được suy dẫn từ các QP gồm các nhóm lượng tử hóa được lập mã liền kề ở phía trên và bên trái (`qP_A` và `qP_L`), trong đó QP tham chiếu là trung bình của hai QP, $(qP_L + qP_A + 1) \gg 1$. Nếu bất kỳ một trong số nhóm lượng tử hóa được lập mã liền kề ở phía trên và bên trái không khả dụng, thì QP không khả dụng đó được thay thế bằng QP trước đó của nhóm lượng tử hóa được lập mã trước đó trong lệnh giải mã. QP mặc định chẳng hạn QP slice được sử dụng làm QP trước nếu nhóm lượng tử hóa

được lập mã trước không khả dụng.

Trong HEVC, báo hiệu sai lệch QP chroma được điều khiển bởi hai cờ, cờ kích hoạt chroma `cu_chroma_qp_offset_enabled_flag` và cờ kích cỡ `Log2MinCuChromaQpOffsetSize`. Cờ kích hoạt chroma biểu thị liệu báo hiệu sai lệch QP chroma được kích hoạt hay không được kích hoạt, cờ kích cỡ được sử dụng để thiết lập kích cỡ nhỏ nhất của các đơn vị trong báo hiệu sai lệch QP chroma. Nếu cờ kích hoạt chroma `cu_chroma_qp_offset_enabled_flag` bằng 1, thì cờ `cu_chroma_qp_offset_flag` xuất hiện để xác định liệu sai lệch QP chroma được sử dụng cho khối hay không. Cờ kích cỡ `Log2MinCuChromaQpOffsetSize` được tính như sau:

$$\text{Log2MinCuChromaQpOffsetSize} = \text{CtbLog2SizeY} - \text{diff_cu_chroma_qp_offset_depth}$$

trong đó `diff_cu_chroma_qp_offset_depth` định rõ sai biệt giữa kích cỡ CTB chroma và kích cỡ CB chroma nhỏ nhất của các đơn vị mã hóa mà chứa cờ `cu_chroma_qp_offset_flag`. Cú pháp của báo hiệu sai lệch QP chroma ở cấp TU là như sau:

```
if ( cu_chroma_qp_offset_enabled_flag && cbfChroma
    && !cu_transquant_bypass && !IsCuChromaQpOffsetCoded ) {
    cu_chroma_qp_offset_flag
    if (cu_chroma_qp_offset_flag )
        cu_chroma_qp_offset_idx
}
```

trong đó `IsCuChromaQpOffsetCoded` được thiết lập bằng 0 khi mã hóa hoặc giải mã CU theo công thức sau.

```
If ( cu_chroma_qp_offset_enabled_flag
    && log2CbSize >= Log2MinCuChromaQpOffsetSize ) {
    IsCuChromaQpOffsetCoded = 0
}
```

Hình dạng của các CB trong HEVC luôn là hình vuông, do đó kích cỡ khối có thể được thể hiện trực tiếp bằng một trị số một chiều, chẳng hạn một trong số chiều rộng khối và chiều cao khối. Các kích thước được suy dẫn chính xác bởi các chiều cao của quá trình phân chia cây tứ phân trong HEVC. Tuy nhiên, cũng có những CB không vuông được suy từ cấu trúc phân chia QTBT hoặc MTT, báo hiệu sai lệch delta QP và QP chroma và suy dẫn QP tham chiếu cho những CB không vuông có thể là một vấn đề. Ngoài ra, các thành phần luma và chroma có thể được lập mã riêng biệt ở các cấu trúc phân chia CU khác nhau để có hai cấu trúc phân chia CU QTBT riêng cho các thành phần luma và chroma. Báo hiệu sai lệch delta QP và QP chroma khi các cấu trúc phân chia CU riêng được sử dụng cho các thành phần luma và chroma có thể là vấn đề khác.

Bản chất kỹ thuật của sáng chế

Các phương pháp và các thiết bị xử lý dữ liệu video trong hệ thống giải mã video nhận dữ liệu vào liên quan đến nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia, xác định chiều cao nút phân chia hiện thời theo dạng phân chia, xác định giá trị của biến bằng cách so sánh chiều cao nút phân chia hiện thời với chiều cao báo hiệu delta QP lớn nhất. Các phương pháp và các thiết bị suy dẫn tiếp delta QP từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích trong TU liên quan đến nút phân chia hiện thời theo giá trị của biến, khôi phục lại QP cuối cùng cho một hoặc nhiều TU liên quan đến nút phân chia hiện thời dựa vào QP tham chiếu và delta QP nếu các thành phần cú pháp được báo hiệu cho nút phân chia hiện thời, và giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời bằng cách sử dụng QP cuối cùng. Các phương pháp và các thiết bị xử lý dữ liệu video tương ứng trong hệ thống mã hóa video nhận dữ liệu vào liên quan đến nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia, xác định chiều cao nút phân chia hiện thời theo dạng phân chia, và xác định QP cho nút phân chia hiện thời theo kết quả so sánh giữa chiều cao nút phân chia hiện thời và chiều cao báo hiệu delta QP lớn nhất. Dựa vào kết quả so sánh, phương pháp và các thiết bị còn suy dẫn delta QP từ QP cho nút phân chia hiện thời và QP tham chiếu và báo hiệu một hoặc nhiều thành phần cú pháp của delta QP trong TU liên quan đến nút phân chia hiện thời. QP được sử dụng khi mã hóa một hoặc nhiều TU trong nút phân chia hiện thời. Theo một số phương án của các phương pháp mã hóa, bước suy dẫn và báo hiệu delta QP được thực hiện nếu kết quả

so sánh biểu thị chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển trong nút phân chia hiện thời.

Dữ liệu video được phân vùng thành các CU lá vuông và không vuông theo cấu trúc phân vùng CU chẳng hạn QTBT hoặc MTT. Một số phương án báo hiệu delta QP xác định chiều cao nút phân chia hiện thời, và chiều cao báo hiệu delta QP lớn nhất được suy dẫn bởi diện tích CU báo hiệu delta QP nhỏ nhất. Chiều cao báo hiệu delta QP lớn nhất được định trước, được suy dẫn, hoặc được báo hiệu ở một trong số hoặc kết hợp ở một trong số cấp slice, cấp khung ảnh, cấp tile, cấp chuỗi ảnh. Theo một ví dụ, nút phân chia hiện thời là CU lá, và delta QP được suy dẫn từ các thành phần cú pháp tại bộ giải mã hoặc bước suy dẫn và báo hiệu delta QP được thực hiện bởi bộ mã hóa khi diện tích của nút phân chia hiện thời lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất và có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển trong nút phân chia hiện thời. Theo ví dụ khác, nút phân chia hiện thời không là CU lá và được phân vùng tiếp thành nhiều CU lá, delta QP được suy dẫn từ các thành phần cú pháp tại bộ giải mã hoặc bước suy dẫn và báo hiệu delta QP được thực hiện bởi bộ mã hóa nếu nút phân chia hiện thời có diện tích lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, diện tích của ít nhất một nút con được phân vùng từ nút phân chia hiện thời nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất, và ít nhất một trong số các CU lá trong nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển. Delta QP ở ví dụ này được sử dụng bởi một hoặc nhiều trong số các CU lá vẫn thuộc nút phân chia hiện thời.

Theo một phương án khi nút phân chia hiện thời là CU lá, delta QP được suy dẫn từ các thành phần cú pháp tại bộ giải mã hoặc bước suy dẫn và báo hiệu delta QP được thực hiện bởi bộ mã hóa nếu chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển. Theo phương án khác khi nút phân chia hiện thời không là CU lá và được phân vùng tiếp thành nhiều CU lá, delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp tại bộ giải mã hoặc bước suy dẫn và báo hiệu delta QP được thực hiện bởi bộ mã hóa nếu chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất, chiều cao của ít nhất một nút con được phân vùng từ nút phân chia hiện thời lớn hơn chiều cao báo

hiệu delta QP lớn nhất, và ít nhất một trong số các CU lá trong nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển. Delta QP được sử dụng bởi một hoặc nhiều trong số các CU lá vẫn thuộc nút phân chia hiện thời ở ví dụ này. Ví dụ về chiều cao nút phân chia hiện thời được suy dẫn từ chiều cao cây tứ phân và chiều cao cây đa phân (MTT). Ví dụ, chiều cao nút phân chia hiện thời được tính bằng cách nhân chiều cao cây tứ phân với L cộng chiều cao MTT, trong đó L là số nguyên định trước. Theo ví dụ khác, chiều cao nút phân chia hiện thời được tính bằng chiều cao cây đa phân (MTDepth), trong đó MTDepth thể hiện diện tích của nút phân chia hiện thời bằng cách xét các định dạng phân chia khác nhau và các nút chia khác nhau. Theo một phương án, chiều cao nút phân chia hiện thời là chiều cao của nút cha cộng 2 khi nút cha được phân vùng bởi phân chia QT, và chiều cao nút phân chia hiện thời là chiều cao của nút cha cộng 1 khi nút cha được phân vùng bởi phân chia BT. Nếu nút cha được phân vùng bởi phân chia TT, chiều cao của nút thứ nhất hoặc thứ ba là chiều cao của nút cha cộng 2 và chiều cao của nút thứ hai là chiều cao của nút cha cộng 1. Theo một phương án, chiều cao báo hiệu delta QP lớn nhất được báo hiệu riêng biệt cho các thành phần luma và chroma khi các cấu trúc phân vùng CU riêng được sử dụng để phân vùng dữ liệu video của các thành phần luma và chroma. Theo phương án khác trong đó nút phân chia hiện thời được phân vùng thành nhiều nút con, chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 2 nếu nút con là nút phân chia thứ nhất hoặc thứ ba của phân chia TT, chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 1 nếu nút con là nút phân chia thứ hai của phân chia TT, chiều cao của nút con là chiều cao nút phân chia hiện thời cộng 1 nếu nút con được phân vùng từ nút phân chia hiện thời bởi phân chia BT, và chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 2 nếu nút con được phân vùng từ nút phân chia hiện thời bởi phân chia cây tứ phân.

Theo một phương án của báo hiệu QP phụ thuộc vào chiều cao, khi nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba được phân vùng từ nút cha bởi phân chia TT và chiều cao nút phân chia hiện thời nhỏ hơn chiều cao báo hiệu delta QP lớn nhất, và chiều cao của nút phân chia thứ hai được phân vùng từ cùng nút cha bằng chiều cao báo hiệu delta QP lớn nhất, QP cuối cùng được suy dẫn từ delta QP được chia sẻ bởi một hoặc nhiều hơn một trong số ba nút phân chia được phân vùng từ nút cha.

Theo một số phương án, ngoài chiều cao nút phân chia hiện thời, việc có báo hiệu các thành phần cú pháp của delta QP cho nút phân chia hiện thời được xác định từ thông tin phụ, ví dụ, cờ biểu thị nếu nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba trong phân chia TT, cờ biểu thị nếu nút phân chia hiện thời là nút phân chia thứ hai trong phân chia TT, hoặc kích cỡ CU cha của nút phân chia hiện thời. Một hoặc nhiều thông tin liên quan delta QP có thể được truyền từ nút cha đến nút phân chia hiện thời để xác định liệu có báo hiệu các thành phần cú pháp của delta QP cho nút phân chia hiện thời hay không. Ví dụ, nút cha xác định cờ thiết lập lại cho nút phân chia hiện thời bằng cách xét diện tích, kích cỡ hoặc chiều cao của nút phân chia hiện thời hoặc dạng phân chia để phân vùng nút cha, và cờ thiết lập lại này được truyền đến nút phân chia hiện thời để xác định giá trị của biến cho nút phân chia hiện thời.

Theo một phương án của trường hợp đặc biệt, nhiều nhất một delta QP được suy dẫn hoặc được báo hiệu cho một hoặc nhiều nút con được phân vùng từ nút phân chia hiện thời bởi phân chia TT nếu chiều cao nút phân chia hiện thời nhỏ hơn chiều cao báo hiệu delta QP lớn nhất và chiều cao của ít nhất một nút con được phân vùng từ nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất. Delta QP được suy dẫn từ hoặc được báo hiệu bởi một hoặc nhiều thành phần cú pháp được chia sẻ bởi một hoặc nhiều hơn một trong số các nút con được phân vùng từ nút phân chia hiện thời theo phương án này. Chiều cao nút phân chia hiện thời và chiều cao nút con được phân vùng từ nút phân chia hiện thời có thể được suy dẫn từ các diện tích.

Theo một số phương án của báo hiệu QP phụ thuộc vào chiều cao, QP tham chiếu luôn là QP slice khi cấu trúc phân vùng CU là cấu trúc phân vùng QTBT hoặc MTT.

Các khía cạnh của sáng chế còn đề xuất thiết bị dùng cho hệ thống giải mã video để giải mã dữ liệu video của nút phân chia hiện thời theo QP cuối cùng được suy dẫn từ delta QP, và delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích trong TU liên quan đến nút phân chia hiện thời có chiều cao nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất. Sáng chế còn đề xuất thiết bị cho hệ thống mã hóa video mã hóa nút phân chia hiện thời bằng cách sử dụng QP được xác định theo kết quả so sánh giữa chiều cao nút phân chia hiện thời và chiều cao tín hiệu delta QP lớn nhất, và suy dẫn delta QP từ QP và QP tham chiếu sau đó báo

hiệu một hoặc nhiều thành phần cú pháp của delta QP trong TU liên quan đến nút phân chia hiện thời.

Các khía cạnh của sáng chế còn đề xuất phương tiện có thể đọc được bằng máy tính không tạm thời để lưu các lệnh chương trình giúp cho mạch xử lý của thiết bị thực hiện quá trình giải mã video để tính delta QP cho nút phân chia hiện thời từ các thành phần cú pháp được báo hiệu trong U theo chiều cao nút phân chia hiện thời. Sáng chế còn đề xuất phương tiện có thể đọc được bằng máy tính không tạm thời để lưu các lệnh chương trình giúp cho mạch xử lý của thiết bị thực hiện quá trình mã hóa video để báo hiệu delta QP cho nút phân chia hiện thời theo chiều cao nút phân chia hiện thời. Các khía cạnh và dấu hiệu khác của sáng chế sẽ trở nên rõ ràng đối với những người có kiến thức trung bình trong lĩnh vực kỹ thuật này khi xem phần mô tả các phương án chi tiết dưới đây.

Mô tả vắn tắt các hình vẽ

Nhiều phương án khác nhau của sáng chế được đề xuất dưới dạng những ví dụ sẽ được mô tả chi tiết dựa theo các hình vẽ dưới đây, trong đó các số chỉ dẫn giống nhau biểu thị các thành phần giống nhau, và cụ thể:

Fig. 1 minh họa cây mã hóa được lấy làm ví dụ để phân chia đơn vị cây mã hóa (CTU) thành các đơn vị mã hóa (các CU) và phân chia mỗi CU thành một hoặc nhiều đơn vị chuyển đổi (nhiều TU) theo chuẩn HEVC;

Fig. 2 minh họa tám dạng phân vùng đơn vị dự đoán (PU) khác nhau trong đó phân chia CU thành một hoặc nhiều PU theo chuẩn HEVC;

Fig. 3A minh họa cấu trúc phân vùng khối được lấy làm ví dụ của phương pháp phân vùng cây nhị phân-cây tứ phân QTBT (Quad-Tree-Binary-Tree - QTBT);

Fig. 3B minh họa cấu trúc cây mã hóa tương ứng với cấu trúc phân vùng khối trên Fig. 3A;

Fig. 4 minh họa phân chia cây tam phân đứng và phân chia cây tam phân ngang;

Fig. 5 minh họa ví dụ về báo hiệu dạng cây dùng cho phân vùng MTT;

Fig. 6 minh họa hai ví dụ phân vùng khối có diện tích bằng hai lần giá trị ngưỡng cho báo hiệu QP bởi phân chia TT;

Fig. 7 là lưu đồ xử lý dữ liệu video bằng cách báo hiệu delta QP cho nút chia phụ thuộc vào diện tích của nút chia theo các phương án của sáng chế;

Fig. 8 là lưu đồ xử lý dữ liệu video bằng cách tái sử dụng QP luma của khối luma đã sắp xếp cho khối chroma theo các phương án của sáng chế;

Fig. 9 minh họa sơ đồ khối hệ thống được lấy làm ví dụ dùng cho hệ thống mã hóa video tích hợp phương pháp xử lý dữ liệu theo các phương án của sáng chế; và

Fig. 10 minh họa sơ đồ khối hệ thống được lấy làm ví dụ dùng cho hệ thống giải mã video tích hợp phương pháp xử lý dữ liệu theo các phương án của sáng chế.

Mô tả chi tiết sáng chế

Sẽ dễ dàng được hiểu rằng những thành phần của sáng chế, như được mô tả và được minh họa tổng quát trên các hình vẽ ở bản mô tả này, có thể được sắp xếp và được thiết kế theo rất nhiều dạng cấu hình khác nhau. Do đó, phần mô tả chi tiết hơn về các phương án của các hệ thống và các phương pháp của sáng chế, như được thể hiện trên các hình vẽ, không nhằm giới hạn phạm vi của sáng chế, như được yêu cầu bảo hộ, mà chỉ là đại diện cho các phương án được chọn của sáng chế.

Tham chiếu toàn bộ bản mô tả này theo “một phương án”, “một số phương án”, hoặc cách diễn đạt tương tự có nghĩa là một dấu hiệu, cấu trúc, hoặc đặc tính cụ thể nào đó được mô tả theo các phương án có thể thuộc ít nhất một phương án của sáng chế. Do đó, sự xuất hiện của các cụm từ “theo một phương án” hoặc “theo một số phương án” ở những phần khác nhau trong toàn bộ bản mô tả này không có nghĩa là nhất thiết phải đề cập đến phương án giống nhau, mà các phương án này có thể được thực hiện riêng biệt hoặc kết hợp với một hoặc nhiều phương án khác. Ngoài ra, các dấu hiệu, các cấu trúc, hoặc các đặc tính đã được mô tả có thể được kết hợp theo bất kỳ cách thức phù hợp nào theo một hoặc nhiều phương án. Tuy nhiên, người có chuyên môn trong lĩnh vực kỹ thuật này sẽ biết rằng sáng chế có thể được thực hiện mà không có một hoặc nhiều hơn một trong số các phương án chi tiết cụ thể, hoặc được thực hiện với các phương pháp khác, các thành phần khác, v.v. Nói cách khác, các cấu trúc, hoặc các phần tử đã biết rõ không được thể hiện hoặc không được mô tả chi tiết nhằm tránh làm rối rắm các khía cạnh nội dung của sáng chế.

Các phương án của sáng chế minh họa các phương pháp xác định và báo hiệu QP khác nhau khi cấu trúc phân vùng QTBT hoặc MTT (ví dụ QTMT) được sử dụng cho phân vùng khối.

Báo hiệu delta QP phụ thuộc vào Diện tích phân vùng QTBT hoặc MTT: Cấu trúc phân vùng QTBT hoặc MTT tạo ra các khối mã hóa CB hình vuông (các CB) cũng như các CB hình chữ nhật, vì vậy báo hiệu delta QP không thể chỉ được đặt điều kiện bởi kích cỡ khối được đo theo một chiều như trong HEVC. Một số phương án của sáng chế theo đó xác định liệu có báo hiệu delta QP phụ thuộc vào diện tích hai chiều của nút chia hoặc phụ thuộc vào chiều cao thể hiện diện tích của nút chia. Diện tích CU báo hiệu delta QP nhỏ nhất hoặc chiều cao báo hiệu delta QP lớn nhất để điều khiển báo hiệu delta QP được định trước, được suy dẫn, hoặc được báo hiệu ở cấp slice, cấp khung ảnh, cấp tile, và/hoặc cấp chuỗi ảnh, sao cho bộ giải mã có thể xác định, suy dẫn, hoặc phân tích diện tích CU báo hiệu delta QP nhỏ nhất hoặc chiều cao báo hiệu delta QP lớn nhất vừa nêu ở cấp slice, cấp khung ảnh, cấp tile, và/hoặc cấp chuỗi ảnh. Nếu việc sử dụng các delta QP được kích hoạt, delta QP được báo hiệu cho CU lá có diện tích lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất khi có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển ở CU lá này, sao cho bộ giải mã có thể phân tích và giải mã delta QP vừa nêu. QP chỉ được tính để thực hiện lượng tử hóa ngược đối với các hệ số khác không hoặc các bảng màu dịch chuyển, do đó quá trình khôi phục lại QP không cần nếu tất cả các hệ số bằng không hoặc không có bảng màu dịch chuyển. Đối với nút phân chia hiện thời mà không phải là CU lá có kích cỡ bằng diện tích CU báo hiệu delta QP nhỏ nhất, nhiều nhất một delta QP được báo hiệu và được chia sẻ bởi toàn bộ các CU lá trong nút phân chia hiện thời, và delta QP vừa nêu được sử dụng bởi một hoặc nhiều hơn một trong số các CU lá trong nút phân chia hiện thời. Nếu một trong số các CU lá trong nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển, delta QP được báo hiệu cho toàn bộ các CU lá, nếu không thì, delta QP không được báo hiệu cho nút phân chia hiện thời. Delta QP được báo hiệu sau đó được chia sẻ bởi toàn bộ các CU lá vẫn thuộc nút phân chia hiện thời và QP cuối cùng được suy dẫn từ delta QP được sử dụng bởi một hoặc nhiều CU lá vẫn thuộc nút phân chia hiện thời. Ví dụ, nút phân chia hiện thời được phân vùng thành bốn CU lá A, B, C, và D, trong đó lệnh lập trình là CU A, CU B, CU C, và sau đó CU D, và chỉ CU C có các hệ số khác không. Tuy nhiên,

delta QP được báo hiệu và được chia sẻ cho toàn bộ các CU lá trong nút phân chia hiện thời, QP cuối cùng được suy dẫn từ delta QP và QP tham chiếu chỉ được áp dụng cho các CU C và D ở ví dụ này. QP tham chiếu được sử dụng cho các CU A và B. Delta QP có thể được báo hiệu sau khi cờ khối được lập mã thứ nhất (cbf) bằng trị số thứ nhất hoặc sau khi cờ bảng màu dịch chuyển thứ nhất bằng trị số thứ hai. Trị số thứ nhất định rõ CU lá có ít nhất một hệ số khác không, và trị số thứ hai định rõ CU lá có ít nhất một bảng màu dịch chuyển. Delta QP được báo hiệu bằng cách truyền hoặc lấy một hoặc nhiều thành phần cú pháp trong dòng bit video tại bộ mã hóa video, và delta QP được báo hiệu được suy dẫn bằng cách phân tích một hoặc nhiều thành phần cú pháp từ dòng bit video.

Biến `IsCuQpDeltaCoded` được sử dụng để biểu thị liệu delta QP đã được lập mã xong hoặc được cần để được báo hiệu hay không. Biến thể hiện diện tích CU báo hiệu delta QP nhỏ nhất, `MinCuQpDeltaArea`, được sử dụng để xác định trị số của biến, `IsCuQpDeltaCoded`, như sau:

```
if( cu_qp_delta_enabled_flag && CbWidth * CbHeight >= MinCuQpDeltaArea )
{
    IsCuQpDeltaCoded = 0
    CuQpDeltaVal = 0
}
```

trong đó `CbWidth` và `CbHeight` thể hiện chiều rộng khối và chiều cao khối của nút chia, một cách tương ứng. Như được mô tả trước đây, `IsCuQpDeltaCoded` biểu thị liệu delta QP đã được lập mã xong cho nút chia này hay chưa, trong đó `IsCuQpDeltaCoded = 0` xuất ra delta QP sẽ được báo hiệu cho nút chia này khi nó vẫn chưa được lập mã. Biến `IsCuQpDeltaCoded = 1` xuất ra delta QP đã được lập mã xong để delta QP mới không được báo hiệu cho nút chia này. QP cuối cùng cho một hoặc nhiều TU liên quan đến nút chia được suy dẫn bởi delta QP và QP tham chiếu nếu delta QP được báo hiệu và QP cuối cùng được suy dẫn từ QP được lập mã trước đó nếu delta QP không được báo hiệu. Theo phương án khác, biến `MinCuQpDeltaArea` ở trên được thay thế bằng trị số log 2 của diện tích CU báo hiệu delta QP nhỏ nhất, `Log2MinCuQpDeltaArea`, và biến, `IsCuQpDeltaCoded`, được thiết lập bằng 0 nếu tổng

log 2 các trị số của chiều rộng khối và chiều cao khối lớn hơn hoặc bằng Log2MinCuQpDeltaArea. Điều này có thể được mô tả bởi các mã giả dưới đây.

```

    if( cu_qp_delta_enabled_flag && (Log2CbWidth + Log2CbHeight) >=
Log2MinCuQpDeltaArea) {

        IsCuQpDeltaCoded = 0

        CuQpDeltaVal = 0

    }

```

trong đó Log2CbWidth và Log2CbHeight là log 2 các trị số của chiều rộng khối và chiều cao khối của nút chia trong QTBT, một cách tương ứng. Phương án tính biến Log2MinCuQpDeltaArea được thể hiện trong công thức sau:

$\text{Log2MinCuQpDeltaArea} = 2 * (\text{CtbLog2SizeY} - \text{diff_cu_qp_delta_depth})$. Theo một số phương án khác, biến Log2MinCuQpDeltaArea còn được sử dụng để xác định liệu biến IsCuQpDeltaCoded được thiết lập bằng 0, nhưng thay vì so sánh với giá trị tổng của log 2 các trị số của chiều rộng khối và chiều cao khối, giá trị nhỏ nhất, lớn nhất hoặc trung bình log 2 các trị số của chiều rộng khối và chiều cao khối được so sánh với biến Log2MinCuQpDeltaArea. Các phương án xác định giá trị của IsCuQpDeltaCoded có thể được miêu tả bởi các mã giả dưới đây.

```

if ( cu_qp_delta_enabled_flag
    && min( Log2CbWidth, Log2CbHeight ) >= Log2MinCuQpDeltaSize ) {

    IsCuQpDeltaCoded = 0

    CuQpDeltaVal = 0

}; hoặc

if( cu_qp_delta_enabled_flag
    && max( Log2CbWidth, Log2CbHeight) >= Log2MinCuQpDeltaSize ) {

    IsCuQpDeltaCoded = 0

    CuQpDeltaVal = 0

}; hoặc

if( cu_qp_delta_enabled_flag

```

```
&& avg( Log2CbWidth, Log2CbHeight) >= Log2MinCuQpDeltaSize ) {  
    IsCuQpDeltaCoded = 0  
    CuQpDeltaVal = 0  
}
```

Ví dụ về xác định báo hiệu delta QP dựa vào diện tích khối của nút chia được thể hiện bởi bảng cú pháp trên Bảng 1.

Coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType) {	Từ khóa
if((((x0 + (1 << log2CbSize) <= pic_width_in_luma_samples) ? 1 : 0) ((y0 + (1 << log2CbSize) <= pic_height_in_luma_samples) ? 1 : 0) (((1 << log2CbSize) <= MaxBtSizeY) ? 1 : 0)) >= 2 && log2CbSize > MinQtLog2SizeY)	
qt_split_cu_flag[x0][y0]	ae(v)
nếu (cu_qp_delta_enabled_flag && (1 << (log2CbSize*2)) > = MinCuQpDeltaArea) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType)	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType)	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, treeType)	
}	
multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, treeType) {	Từ khóa
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
mtt_split_cu_flag	ae(v)
nếu (cu_qp_delta_enabled_flag && cbWidth*cbHeight > = MinCuQpDeltaArea) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	

Bảng 1

Cú pháp của báo hiệu delta QP ở cấp TU có thể tương tự như cú pháp được sử dụng trong HEVC và được thể hiện dưới đây, trong đó biến tuyệt đối `cu_qp_delta_abs` và cờ dấu `cu_qp_delta_sign_flag` được báo hiệu khi delta QP được kích hoạt và biến `IsCuQpDeltaCoded` được thiết lập bằng 0.

```
if ( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {
    cu_qp_delta_abs
    if( cu_qp_delta_abs )
        cu_qp_delta_sign_flag
}
```

Biến `CuQpDeltaVal` thể hiện giá trị của delta QP, được suy dẫn dựa vào biến tuyệt đối `cu_qp_delta_abs` và cờ dấu `cu_qp_delta_sign_flag`. QP cuối cùng được khôi phục lại theo QP tham chiếu và delta QP `CuQpDeltaVal`. Cờ dấu `cu_qp_delta_sign_flag` được suy bằng 0 nếu nó không có mặt. Khi `cu_qp_delta_abs` có mặt, biến `IsCuQpDeltaCoded` được thiết lập bằng 1 và giá trị của delta QP `CuQpDeltaVal` được suy dẫn như sau:

$$\text{IsCuQpDeltaCoded} = 1$$

$$\text{CuQpDeltaVal} = \text{cu_qp_delta_abs} * (1 - 2 * \text{cu_qp_delta_sign_flag})$$

Các hàm báo hiệu và tính toán delta QP cho TU được gọi ra khi TU có ít nhất một hệ số khác không. Ví dụ về báo hiệu delta QP khi có ít nhất một hệ số khác không trong TB luma hoặc chroma được miêu tả bởi các bảng cú pháp được thể hiện trên Bảng 2.

<code>transform_unit(x0, y0, tbWidth, tbHeight, treeType) {</code>	Từ khóa
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA)</code>	
<code>tu_cbf_luma[x0][y0]</code>	<code>ae(v)</code>
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) {</code>	

tu_cbf_cb[x0][y0]	ae(v)
tu_cbf_cr[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0] tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0])	
delta_qp()	

delta_qp() {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	

Bảng 2

Biểu thị diện tích bằng chiều cao: Cờ chiều cao `diff_cu_qp_delta_depth` được sử dụng trong HEVC để định kích cỡ nhỏ nhất cho báo hiệu delta QP chỉ xét kích cỡ một chiều của nút chia dưới dạng chiều rộng khối và chiều cao khối của bất kỳ nút chia nào trong HEVC luôn giống nhau. Các phương án của sáng chế còn sử dụng một hoặc nhiều trị số chiều cao (depth) để định nghĩa diện tích nhỏ nhất cho báo hiệu delta QP, song trị số (các trị số) được đề xuất đó phản ánh các kích cỡ hai chiều của các nút chia bằng cách xét dạng phân chia. Một số phương án báo hiệu delta QP được đặt điều kiện bởi diện tích của nút chia được thể hiện bằng chiều cao cây tứ phân và chiều cao MTT của nút chia. Ở các phương án này, diện tích CU báo hiệu delta QP nhỏ nhất được tính bằng chiều cao báo hiệu delta QP lớn nhất, `diff_cu_qp_delta_depth`, trong đó `diff_cu_qp_delta_depth`, được định trước, được suy dẫn, hoặc được báo hiệu ở cấp slice, cấp khung ảnh, cấp tile, và/hoặc cấp chuỗi ảnh. Theo một phương án, chiều cao báo hiệu delta QP lớn nhất phải là số chẵn. Theo một ví dụ, chiều cao báo hiệu delta QP lớn nhất được tính bằng: $\text{diff_cu_qp_delta_depth} = \text{diff_cu_qp_delta_depth_div2} \ll 1$, trong đó `diff_cu_qp_delta_depth_div2` được báo hiệu ở cấp slice, cấp khung ảnh, cấp tile, cấp hàng CTU và/hoặc cấp chuỗi ảnh. Nếu delta QP được kích hoạt, cho CU lá có chiều cao nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và CU lá có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển, delta QP được báo hiệu cho CU lá này. Đối với nút chia không phải CU lá có chiều cao nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và chiều cao của ít nhất một nút con được

phân vùng từ nút chia lớn hơn chiều cao báo hiệu delta QP lớn nhất, nhiều nhất một delta QP được báo hiệu ở nút chia này. Nút chia được phân vùng tiếp thành nhiều CU lá, và delta QP được báo hiệu nếu một trong số các CU lá ở nút chia này có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển. Delta QP được báo hiệu được chia sẻ bởi toàn bộ các CU lá thuộc nút chia và QP cuối cùng được tính bởi delta QP được sử dụng bởi một hoặc nhiều trong số các CU lá ở nút chia. Delta QP có thể được báo hiệu sau khi cbf thứ nhất bằng trị số thứ nhất hoặc sau khi cờ bảng màu dịch chuyển thứ nhất bằng trị số thứ hai, trong đó trị số thứ nhất định rõ CU có ít nhất một hệ số khác không, và trị số thứ hai định rõ CU có ít nhất một bảng màu dịch chuyển.

Tương tự các phương án so sánh trước đó về sự kết hợp chiều rộng khối và chiều cao khối với diện tích CU báo hiệu delta QP nhỏ nhất, theo một số phương án, biến IsCuQpDeltaCoded được thiết lập bằng 0 bằng cách so sánh sự kết hợp của chiều cao cây tứ phân và chiều cao MTT với chiều cao báo hiệu delta QP lớn nhất. Theo một phương án, sự kết hợp của chiều cao cây tứ phân và chiều cao MTT được tính bằng cách nhân chiều cao cây tứ phân với 2 cộng chiều cao MTT, và biến IsCuQpDeltaCoded được thiết lập bằng 0 trong khi mã hóa hoặc giải mã CU theo các mã giả dưới đây:

```

if( cu_qp_delta_enabled_flag &&
  ( (2*quadTreeDepth + MTTDepth) <= diff_cu_qp_delta_depth) {

    IsCuQpDeltaCoded = 0

    CuQpDeltaVal = 0

  }

```

Sự kết hợp được lấy làm ví dụ ở trên giữa chiều cao cây tứ phân và chiều cao MTT có thể được tổng quát hóa khi nhân chiều cao cây tứ phân với L cộng chiều cao MTT, trong đó L là số nguyên định trước, ví dụ, L=2 hoặc L=1. Theo một số phương án khác, sự kết hợp của chiều cao cây tứ phân và chiều cao MTT là quadTreeDepth + (MTTDepth>>1) hoặc quadTreeDepth + (MTTDepth+K)>>N, trong đó K và N là các số nguyên định trước. Ví dụ, K=2 và N=1, K=0 và N=0, hoặc K=1 và N=1. Theo một số phương án, biến được gọi là AreaDepth được suy dẫn từ chiều cao cây tứ phân và/hoặc chiều cao MTTDepth. Biến AreaDepth được suy dẫn như sau AreaDepth = J*quadTreeDepth + ((MTTDepth+K)>>N) hoặc AreaDepth = (L*quadTreeDepth +

MTTDepth), trong đó J, K, N, và L có thể là bất kỳ số nguyên nào. Biến AreaDepth có thể thay cho tích số giữa chiều rộng khối và chiều cao khối ($cbWidth * cbHeight$) trong các phương án được mô tả trước đó, và biến này thể hiện diện tích CU báo hiệu delta QP nhỏ nhất MinCuQpDeltaArea có thể được thay thế bằng trị số chiều cao (depth) tương ứng MinCuQpDeltaAreaDepth. MinCuQpDeltaAreaDepth được báo hiệu ở cấp slice, cấp khung ảnh, cấp chuỗi ảnh, cấp hàng CTU, hoặc cấp tile, mà thể hiện diện tích, tỷ lệ diện tích, chiều cao, hoặc log 2 diện tích của MinCuQpDeltaArea hoặc trị số của diện tích, tỷ lệ diện tích, chiều cao, hoặc log 2 Diện tích của MinCuQpDeltaArea tính theo chiều rộng, chiều cao, diện tích CTU, log 2 Diện tích, hoặc MaxDepth. Để tính toán, $cbWidth * cbHeight / 4$ có thể được thay thế bằng $AreaDepth + 2$. Theo một phương án, log 2 trị số của MinCuQpDeltaArea được báo hiệu được ràng buộc là số chẵn. Theo phương án khác, trị số liên quan đến MinCuQpDeltaArea được báo hiệu, ví dụ, MinCuQpDeltaArea được tính bằng cách nhân trị số được báo hiệu (signaled_value) với 2, chẳng hạn $2 * signaled_value$ được sử dụng để suy dẫn MinCuQpDeltaArea, hoặc log 2 MinCuQpDeltaArea. Ví dụ, $MinCuQpDeltaArea = 1 << (signaled_value * 2)$ hoặc $MinCuQpDeltaArea = CTU_Diện\ tích >> (signaled_value * 2)$.

Các thành phần cú pháp liên quan đến delta QP, chẳng hạn biến tuyệt đối $cu_qp_delta_abs$ và cờ dấu $cu_qp_delta_sign_flag$, được báo hiệu ở cấp TU khi cờ đã kích hoạt biểu thị delta QP được kích hoạt và biến IsCuQpDeltaCoded biểu thị delta QP vẫn chưa được lập mã. Giá trị của delta QP CuQpDeltaVal được suy dẫn dựa vào hai thành phần cú pháp này, $cu_qp_delta_abs$ và $cu_qp_delta_sign_flag$. QP cuối cùng được khôi phục lại theo QP tham chiếu và delta QP CuQpDeltaVal.

Chiều cao báo hiệu delta QP lớn nhất $diff_cu_qp_delta_depth$ có thể được báo hiệu riêng biệt cho các thành phần luma và chroma khi các cấu trúc phân chia CU riêng được sử dụng cho các thành phần luma và chroma. Theo phương án khác, chiều cao báo hiệu delta QP lớn nhất $diff_cu_qp_delta_depth$ có thể giống nhau đối với các thành phần luma và chroma. Theo phương án khác nữa, chiều cao báo hiệu delta QP lớn nhất cho các thành phần chroma được thiết lập làm trị số phụ thuộc vào chiều cao báo hiệu delta QP lớn nhất cho các thành phần luma khi các cấu trúc phân chia CU riêng được sử dụng cho các thành phần luma và chroma. Chiều cao cây đa phân (MTDepth) được định nghĩa để biểu diễn diện tích của nút chia chính xác hơn bằng

cách sử dụng ý tưởng dùng các trị số chiều cao (depth) khác nhau cho các nút chia khác nhau mà thuộc cùng nút cha và/hoặc dùng các trị số chiều cao khác nhau cho các phương pháp phân chia khác nhau bao gồm phân chia QT, BT, và TT. Theo một phương án tính toán MTDepth, chiều cao của các nút phân chia thứ nhất và thứ ba là chiều cao của nút chia cha cộng 2 trong khi chiều cao của nút phân chia thứ hai là chiều cao của nút chia cha cộng 1 khi phân chia TT được áp dụng; và các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1 khi phân chia BT được áp dụng. Theo phương án khác, khi tính chiều cao được tạo ra bởi phân chia TT, mỗi chiều cao của ba nút phân chia là chiều cao của nút chia cha cộng 1; và khi tính các chiều cao được tạo ra bởi phân chia BT, các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1. Theo phương án khác nữa, khi tính các chiều cao được tạo ra bởi phân chia TT, các chiều cao của tất cả ba nút phân chia là chiều cao của nút chia cha cộng 2; và khi tính các chiều cao được tạo ra bởi phân chia BT, các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1. Theo một phương án, chiều cao của khối hiện thời được tăng thêm 2 khi khối hiện thời được phân vùng từ phân chia QT.

Báo hiệu delta QP phụ thuộc vào chiều cao hoặc diện tích QT: Vì cấp thứ hai của MTT được bắt đầu từ các nút lá của phân chia QT, nên phương án cần thiết lập một ràng buộc để báo hiệu các delta QP tại các nút QT. Một delta QP được lập mã và được chia sẻ cho toàn bộ các CU lá trong nút QT. Kích cỡ nhỏ nhất của các đơn vị trong báo hiệu delta QP có thể được báo hiệu giống như trong HEVC. Theo phương án khác, vì cấp thứ hai của MTT được bắt đầu từ các nút lá của phân chia QT, nên báo hiệu delta QP HEVC cho những phân vùng QT. Ví dụ, đối với nút lá QT, nếu kích cỡ, diện tích, chiều rộng, hoặc chiều cao nút lá QT lớn hơn hoặc bằng giá trị ngưỡng, hoặc nếu chiều cao QT của nút lá QT nhỏ hơn hoặc bằng giá trị ngưỡng, nhiều nhất một delta QP có thể được báo hiệu. Các CU lá thuộc nút lá QT này chia sẻ delta QP được báo hiệu. Nếu nút QT không là CU lá và kích cỡ, diện tích, chiều rộng, hoặc chiều cao bằng giá trị ngưỡng, nhiều nhất một delta QP có thể được báo hiệu. Các CU lá thuộc nút QT chia sẻ delta QP được báo hiệu. Báo hiệu Delta QP cho những phân vùng QT thì tương tự hoặc giống như báo hiệu delta QP HEVC. Các CU lá được phân vùng từ phân chia BT hoặc TT thuộc nút QT chia sẻ delta QP giống nhau. Giá trị ngưỡng có thể được định

trước, được suy dẫn, hoặc được báo hiệu ở một trong số hoặc kết hợp ở một trong số cấp slice, cấp khung ảnh, cấp tile, và cấp chuỗi ảnh.

Báo hiệu Delta QP cho trường hợp đặc biệt: Trường hợp đặc biệt xảy ra khi báo hiệu delta QP được đặt điều kiện bởi diện tích của khối và phân chia cây tam phân được sử dụng để phân chia nút cha thành ba nút phân chia, trong đó diện tích của nút chia thứ nhất hoặc thứ ba nhỏ hơn giá trị ngưỡng báo hiệu delta QP (ví dụ diện tích CU báo hiệu delta QP nhỏ nhất), và diện tích của nút chia thứ hai bằng giá trị ngưỡng báo hiệu delta QP. Ví dụ, diện tích của nút cha bằng hai lần giá trị ngưỡng báo hiệu delta QP, sau khi phân vùng nút cha bởi phân chia TT, diện tích của mỗi một trong số nút chia thứ nhất và nút chia thứ ba bằng một nửa giá trị ngưỡng trong khi diện tích của nút phân chia thứ hai bằng giá trị ngưỡng. Fig. 6 minh họa hai ví dụ của trường hợp đặc biệt mà phân vùng khối chanhờ phân chia TT. Khối 62 minh họa kích cỡ diện tích CU báo hiệu delta QP nhỏ nhất, là giá trị ngưỡng cho báo hiệu delta QP, do đó bất kỳ nút chia nào có diện tích lớn hơn hoặc bằng khối 62 có thể có delta QP của riêng nó được báo hiệu trong TU liên quan. Mỗi một trong số khối cha 64 và khối cha 66 có diện tích bằng hai lần diện tích CU báo hiệu delta QP nhỏ nhất. Khối cha 64 được phân vùng bởi phân chia TT đứng và khối cha 66 được phân vùng bởi phân chia TT ngang. Diện tích của nút chia thứ nhất 642 hoặc nút chia thứ ba 646 nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất trong khi diện tích của nút phân chia thứ hai 644 bằng diện tích CU báo hiệu delta QP nhỏ nhất. Tương tự, diện tích của nút chia thứ nhất 662 hoặc nút chia thứ ba 666 nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất trong khi diện tích của nút phân chia thứ hai 664 bằng diện tích CU báo hiệu delta QP nhỏ nhất. Ví dụ, diện tích CU báo hiệu delta QP nhỏ nhất là 32×32 , khối cha 64 có kích cỡ 64×32 , và khối cha 66 có kích cỡ 32×64 .

Một số phương án của sáng chế thiết lập quy tắc của báo hiệu delta QP cho trường hợp đặc biệt đã nêu trên. Theo một phương án, chỉ một delta QP có thể được báo hiệu cho tất cả ba nút phân chia khi diện tích của nút chia thứ nhất hoặc thứ ba nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất và diện tích của nút chia thứ hai bằng diện tích CU báo hiệu delta QP nhỏ nhất. Để tổng quát hóa quy tắc cho trường hợp đặc biệt, delta QP được báo hiệu cho tất cả các nút chia của nút cha để chia sẻ nếu ít nhất một trong số các nút chia có diện tích nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất và có ít nhất một hệ số khác không hoặc bảng màu dịch chuyển.

Một hoặc nhiều thành phần cú pháp của báo hiệu delta QP ở cấp TU có thể tương tự như trong HEVC, ví dụ, biến tuyệt đối `cu_qp_delta_abs` và cờ dấu `cu_qp_delta_sign_flag` được báo hiệu nếu cờ kích hoạt `cu_qp_delta_enabled_flag` biểu thị delta QP được kích hoạt và biến `IsCuQpDeltaCoded` được thiết lập bằng 0. Giá trị của delta QP `CuQpDeltaVal` được suy dẫn dựa vào hai thành phần cú pháp này `cu_qp_delta_abs` và `cu_qp_delta_sign_flag`, và QP cuối cùng được khôi phục lại theo QP tham chiếu và delta QP `CuQpDeltaVal`. Cờ dấu `cu_qp_delta_sign_flag` được suy bằng 0 nếu cờ dấu sign này không có mặt. Các phương pháp khác của báo hiệu delta QP ở cấp TU cũng có thể được áp dụng.

Theo phương án khác của báo hiệu delta QP cho trường hợp đặc biệt, chỉ một delta QP được báo hiệu cho nút chia thứ hai và không có delta QP mới được lập mã cho các nút phân chia thứ nhất và thứ ba. QP được sử dụng ở nút chia thứ nhất là QP cuối cùng, QP được sử dụng ở nút chia thứ hai được suy dẫn từ delta QP được báo hiệu, và QP này cũng được sử dụng trong nút chia thứ ba. Một hoặc nhiều thành phần cú pháp cho báo hiệu delta QP ở cấp TU có thể tương tự như báo hiệu delta QP trong HEVC, ví dụ, delta QP được suy dẫn dựa vào biến tuyệt đối và cờ dấu. QP cuối cùng được khôi phục lại bằng QP tham chiếu và delta QP.

Theo phương án khác nữa của báo hiệu delta QP cho trường hợp đặc biệt, mỗi một trong số ba nút phân chia có delta QP của riêng nó, do đó nhiều nhất ba delta QP được báo hiệu cho ba nút phân chia TT. Theo phương án này, đối với nút hiện thời có diện tích bằng hai lần diện tích CU báo hiệu delta QP nhỏ nhất và được phân vùng bởi phân chia QT, mỗi nút phân chia cây tứ phân có thể có delta QP của riêng nó hoặc hai nút phân chia cây tứ phân đầu tiên được nhóm lại làm Nhóm lượng tử hóa (QG) thứ nhất trong khi hai nút phân chia cây tứ phân cuối cùng được nhóm lại làm QG thứ hai. Biến `IsCUQpDeltaCoded` được thiết lập bằng 0 cho QG thứ nhất và QG thứ hai. Đối với nút hiện thời có kích cỡ lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, hoặc đối với nút hiện thời có kích cỡ nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất nhưng kích cỡ cha của nó lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, một delta QP có thể được báo hiệu cho nút hiện thời. Biến `IsCuQpDeltaCoded` được thiết lập bằng 0 cho nút hiện thời.

Kích cỡ hoặc diện tích của nút chia hoặc nút cha có thể được suy dẫn từ chiều cao, và diện tích CU báo hiệu delta QP nhỏ nhất cũng có thể được suy dẫn từ chiều cao

báo hiệu delta QP lớn nhất. Theo một số phương án của sáng chế, chiều cao cây đa phân (MTDepth) được định nghĩa để tính chiều cao, và ý tưởng của MTDepth là sử dụng các trị số chiều cao (depth) khác nhau cho các nút chia khác nhau thuộc cùng nút cha và/hoặc sử dụng các trị số chiều cao (depth) khác nhau cho các phương pháp phân chia khác nhau bao gồm phân chia QT, BT, và TT. Ví dụ, chiều cao của nút phân chia thứ nhất hoặc thứ ba được phân vùng từ nút cha bởi phân chia TT bằng chiều cao của nút cha cộng 2, và chiều cao của nút phân chia thứ hai được phân vùng từ nút cha bằng chiều cao của nút cha cộng 1. Chiều cao của nút chia được phân vùng từ nút cha bởi phân chia BT bằng chiều cao của nút cha cộng 1. Chiều cao của nút phân chia cây tứ phân bằng MTDepth của nút cha của nó cộng 2.

Cờ biểu thị liệu nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba trong phân chia cây tam phân được sử dụng cho báo hiệu delta QP theo một số phương án. Ví dụ, trong trường hợp của ba nút phân chia cây tam phân có thể có các delta QP riêng của nó, nếu diện tích của nút phân chia hiện thời lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, hoặc nếu diện tích bằng một nửa của diện tích CU báo hiệu delta QP nhỏ nhất nhưng nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba trong phân chia cây tam phân, một delta QP có thể được báo hiệu cho nút phân chia hiện thời. Biến IsCuQpDeltaCoded được thiết lập bằng 0 cho nút hiện thời để biểu thị delta QP có thể được báo hiệu. Theo một số phương án khác, cờ biểu thị liệu nút phân chia hiện thời là nút phân chia thứ hai trong phân chia cây tam phân được sử dụng cho báo hiệu delta QP hay không. Ví dụ, trong trường hợp mà chỉ một delta QP có thể được báo hiệu cho cả ba nút phân chia cây tam phân, nếu diện tích của nút phân chia hiện thời lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, hoặc nếu diện tích bằng diện tích CU báo hiệu delta QP nhỏ nhất và nút phân chia hiện thời không là nút phân chia thứ hai trong phân chia cây tam phân, một delta QP có thể được báo hiệu cho nút phân chia hiện thời bằng cách thiết lập biến IsCuQpDeltaCoded bằng 0 cho nút phân chia hiện thời.

Theo một số phương án khác, nút phân chia hiện thời quyết định liệu delta QP có thể được báo hiệu bằng cách xét kích cỡ CU cha (parentCuArea) của nút phân chia hiện thời. Tức là, nút con quyết định liệu delta QP được báo hiệu không chỉ theo diện tích của nút con mà còn theo thông tin của khối cha của nó. Ví dụ, trong trường hợp của ba nút phân chia cây tam phân có thể có các delta QP riêng của nó, nếu diện tích

của nút phân chia hiện thời bằng hai lần diện tích CU báo hiệu delta QP nhỏ nhất, mỗi nút chia được phân vùng từ nút phân chia hiện thời có thể có delta QP của riêng nó. Nếu diện tích của nút cha của nút phân chia hiện thời lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, biến `IsCuQpDeltaCoded` được thiết lập bằng 0 cho nút phân chia hiện thời, biểu thị một delta QP có thể được báo hiệu cho nút phân chia hiện thời. Các bảng cú pháp được thể hiện trên Bảng 3 minh họa ví dụ về xác định báo hiệu QP dựa vào kích cỡ CU cha khi ba nút phân chia có các delta QP riêng của nó.

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType, parentCuArea) {	Từ khóa
? if((((x0 + (1 << log2CbSize) <= pic_width_in_luma_samples) 1 : 0) + ((y0 + (1 << log2CbSize) <= pic_height_in_luma_samples) ? 1 : 0) + (((1 << log2CbSize) <= MaxBtSizeY) ? 1 : 0)) >= 2 && log2CbSize > MinQtLog2SizeY)	
qt_split_cu_flag[x0][y0]	ae(v)
nếu (((1 << (2*log2CbSize)) >= MinCuQpDeltaArea) ((1 << (2*log2CbSize)) < MinCuQpDeltaArea && parentCuArea > MinCuQpDeltaArea)) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType, 1 << (log2CbSize*2))	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType, 1 << (log2CbSize*2))	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType, 1 << (log2CbSize*2))	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType, 1 << (log2CbSize*2))	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, treeType, parentCuArea)	
}	

multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partI dx, treeType, parentCuArea) {	Từ khóa
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
mtt_split_cu_flag	ae(v)
nếu ((cbWidth*cbHeight >= MinCuQpDeltaArea) (cbWidth*cbHeight < MinCuQpDeltaArea && parentCuArea > MinCuQpDeltaArea)) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(mtt_split_cu_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(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 0, treeType, cbWidth* cbHeight)	
if(x1 < pic_width_in_luma_samples)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, mttDepth + 1, depthOffset , 1, treeType, cbWidth* cbHeight)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 4)	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 0, treeType, cbWidth* cbHeight)	
if(y1 < pic_height_in_luma_samples)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, treeType, cbWidth* cbHeight)	

<code>} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)</code>	
<code>{</code>	
<code> x1 = x0 + (cbWidth / 4)</code>	
<code> x2 = x0 + (3 * cbWidth / 4)</code>	
<code> multi_type_tree(x0, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset,</code>	
<code> 0, treeType, cbWidth* cbHeight)</code>	
<code> multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset,</code>	
<code> 1, treeType, cbWidth* cbHeight)</code>	
<code> multi_type_tree(x2, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset,</code>	
<code> 2, treeType, cbWidth* cbHeight)</code>	
<code> } else { /* SPLIT TT HOR */</code>	
<code> y1 = y0 + (cbHeight / 4)</code>	
<code> y2 = y0 + (3 * cbHeight / 4)</code>	
<code> multi_type_tree(x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset,</code>	
<code> 0, treeType, cbWidth* cbHeight)</code>	
<code> multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset,</code>	
<code> 1, treeType, cbWidth* cbHeight)</code>	
<code> multi_type_tree(x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset,</code>	
<code> 2, treeType, cbWidth* cbHeight)</code>	
<code> }</code>	
<code> } else</code>	
<code> coding_unit(x0, y0, cbWidth, cbHeight, treeType)</code>	
<code>}</code>	

Bảng 3

Theo các phương án trước, quyết định báo hiệu delta QP được thực hiện tại nút con. Một số phương án của sáng chế xác định liệu delta QP có thể được báo hiệu cho nút chia ở nút cha của nút chia hay không. Ở mỗi nút cha, liệu delta QP có thể được báo hiệu độc lập trong mỗi nút chia được phân vùng từ nút cha được xác định. Một hoặc nhiều thông tin liên quan delta QP, chẳng hạn độ khả dụng của báo hiệu delta QP hoặc độ khả dụng của biến `IsCuQpDeltaCoded`, có thể được truyền từ nút cha đến các nút chia nếu cần. Quyết định báo hiệu delta QP sau đó được suy dẫn theo nút chia dựa vào thông tin của diện tích CU báo hiệu delta QP nhỏ nhất `MinCuQpDeltaArea`, chiều rộng, chiều cao, kích cỡ, diện tích, hoặc chiều sâu của nút hiện thời, dạng phân vùng, kích cỡ ảnh, vị trí nút hiện thời, thông tin liên quan delta QP được truyền từ nút cha, hoặc kết hợp giữa các thông tin vừa nêu. Ví dụ, biến thiết lập lại `resetQpDeltaCoded`

được truyền từ nút cha đến nút phân chia hiện thời, và nếu biến thiết lập lại là đúng và hàm báo hiệu delta QP được kích hoạt, thì biến IsCuQpDeltaCoded được thiết lập bằng 0. Nếu nút phân chia hiện thời được chia thành các nút con, biến resetQpDeltaCodedForChildren được xác định và được truyền đến các nút con. Đối với CTU, biến IsCuQpDeltaCoded có thể được thiết lập bằng 0 và biến thiết lập lại resetQpDeltaCoded for CTU được thiết lập bằng 1.

Theo một phương án, nút cha có kích cỡ lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất xác định cờ thiết lập lại cho nút chia được phân vùng từ nút cha bằng cách xét diện tích, kích cỡ, hoặc chiều cao của nút chia hoặc dạng phân chia để phân vùng nút cha. Cờ thiết lập lại được truyền đến nút chia để điều khiển liệu delta QP có thể được báo hiệu cho nút chia này hay không. Ví dụ, delta QP được báo hiệu cho tất cả các nút phân chia TT của nút cha khi nút chia thứ nhất được phân vùng từ nút cha nhỏ hơn giá trị ngưỡng cho báo hiệu delta QP, tất cả các nút phân chia TT chia sẻ delta QP này khi không có thêm delta QP được báo hiệu cho các nút chia ngay cả khi diện tích của nút chia thứ hai bằng giá trị ngưỡng cho báo hiệu delta QP.

Theo một phương án, khi diện tích của một trong số các nút chia được phân vùng từ nút cha nhỏ hơn MinCuQpDeltaArea và diện tích của nút cha lớn hơn MinCuQpDeltaArea, toàn bộ các nút chia chia sẻ nhiều nhất một delta QP. Theo một ví dụ, nếu diện tích của một nút chia được phân vùng từ nút hiện thời nhỏ hơn MinCuQpDeltaArea và diện tích của nút hiện thời lớn hơn MinCuQpDeltaArea, không một nút chia nào trong số các nút chia được phân vùng từ nút hiện thời có thể thiết lập biến IsCuQpDeltaCoded bằng 0. Ví dụ này được minh họa trên các bảng cú pháp được thể hiện trên Bảng 4.

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth, qtResetQpDeltaCoded) {	Từ khóa
if(cu_qp_delta_enabled_flag && qtResetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
qtResetQpDeltaCoded = (cbWidth*cbHeight/4 >= MinCuQpDeltaArea && qtResetQpDeltaCoded) ? 1: 0	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded)	
if(y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA, qtResetQpDeltaCoded)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA, qtResetQpDeltaCoded)	
}	
}	

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType, qtResetQpDeltaCoded) {	Từ khóa
if((((x0 + (1 << log2CbSize) <= pic_width_in_luma_samples) ? 1 : 0) + ((y0 + (1 << log2CbSize) <= pic_height_in_luma_samples) ? 1 : 0)) >= 2 && log2CbSize > MinQtLog2SizeY)	
qt_split_cu_flag[x0][y0]	ae(v)
nếu (cu_qp_delta_enabled_flag && qtResetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
qtResetQpDeltaCoded = (cbWidth*cbHeight/4 >= MinCuQpDeltaArea && qtResetQpDeltaCoded) ? 1 : 0	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
} else {	
mttResetQpDeltaCoded = ((cbWidth*cbHeight >= MinCuQpDeltaArea) && qtResetQpDeltaCoded) ? 1 : 0	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, treeType, mttResetQpDeltaCoded)	
}	
}	

multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, treeType, mttResetQpDeltaCoded) {	Từ khóa
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
mtt_split_cu_flag	ae(v)
nếu (cu_qp_delta_enabled_flag && mttResetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(mtt_split_cu_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(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
mttResetQpDeltaCoded = ((cbWidth*cbHeight/2 >= MinCuQpDeltaArea) && mttResetQpDeltaCoded) ? 1: 0	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
mttResetQpDeltaCoded = ((cbWidth*cbHeight/2 >= MinCuQpDeltaArea) && mttResetQpDeltaCoded) ? 1: 0	

multi_type_tree(x0, y0, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)	
if(y1 < pic_height_in_luma_samples)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)	
{	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
mttResetQpDeltaCoded = ((cbWidth*cbHeight/4 >= MinCuQpDeltaArea) && mttResetQpDeltaCoded) ? 1: 0	
multi_type_tree(x0, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)	
multi_type_tree(x2, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 2, treeType, mttResetQpDeltaCoded)	
} else { /* SPLIT_TT_HOR */	
y1 = y0 + (cbHeight / 4)	
y2 = y0 + (3 * cbHeight / 4)	
mttResetQpDeltaCoded = ((cbWidth*cbHeight/4 >= MinCuQpDeltaArea) && mttResetQpDeltaCoded) ? 1: 0	
multi_type_tree(x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)	
multi_type_tree(x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 2, treeType, mttResetQpDeltaCoded)	
}	
} else	

<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType)</code>	
<code>}</code>	

Bảng 4

Chiều cao cây đa phân (MTDepth) được định nghĩa để tính chiều cao của nút chia và biểu diễn diện tích của nút chia. Theo một phương án, khi tính chiều cao bằng phân chia TT, các chiều cao của các nút phân chia thứ nhất và thứ ba là chiều cao của nút chia cha cộng 2, và chiều cao của nút phân chia thứ hai là chiều cao của nút chia cha cộng 1. Khi tính chiều cao bằng phân chia BT, các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1. Ví dụ này được minh họa trên các bảng cú pháp được thể hiện trên Bảng 5. Theo phương án khác, khi tính chiều cao bằng phân chia TT, các chiều cao của tất cả ba nút phân chia là chiều cao của nút chia cha cộng 1, và khi tính chiều cao bằng phân chia BT, các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1. Theo phương án khác nữa, khi tính chiều cao bằng phân chia TT, các chiều cao của tất cả ba nút phân chia là chiều cao của nút chia cha cộng 2, và khi tính chiều cao bằng phân chia BT, các chiều cao của hai nút được phân chia là chiều cao của nút chia cha cộng 1. Theo một phương án, chiều cao được tăng thêm 2 khi nút chia được phân vùng bằng cách sử dụng quá trình phân chia cây tứ phân.

<code>coding_tree_unit() {</code>	Từ khóa
<code> xCtb = (CtbAddrInRs % PicChiều rộngInCtbsY) << CtbLog2SizeY</code>	
<code> yCtb = (CtbAddrInRs / PicChiều rộngInCtbsY) << CtbLog2SizeY</code>	
<code> if(slice sao luma flag slice sao chroma flag)</code>	
<code> sao(xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY)</code>	
<code> if(slice alf_enabled flag){</code>	
<code> alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]</code>	<code>ae(v)</code>
<code> if(alf_chroma_idc == 1 alf_chroma_idc == 3)</code>	
<code> alf_ctb_flag[1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]</code>	<code>ae(v)</code>
<code> if(alf_chroma_idc == 2 alf_chroma_idc == 3)</code>	
<code> alf_ctb_flag[2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]</code>	<code>ae(v)</code>
<code> }</code>	
<code> if(slice type == I && qtbtt_dual_tree_intra_flag)</code>	
<code> dual_tree_implicit_qt_split(xCtb, yCtb, CtbLog2SizeY, 0, 1)</code>	
<code> else</code>	
<code> coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE, 1)</code>	
<code> }</code>	

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth, resetQpDeltaCoded) {	Từ khóa
nếu (cu_qp_delta_enabled_flag && resetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
resetQpDeltaCodedForChildren = (2*(cqtDepth+1) <= diff cu_qp_delta_depth) ? 1: 0	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1, resetQpDeltaCodedForChildren)	
if(x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1, resetQpDeltaCodedForChildren)	
if(y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1, resetQpDeltaCodedForChildren)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1, resetQpDeltaCodedForChildren)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA, resetQpDeltaCoded)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA, resetQpDeltaCoded)	
}	
}	

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType, resetQpDeltaCoded) {	Từ khóa
minQtSize = (treeType == DUAL_TREE_CHROMA) ? MinQtSizeC : MinQtSizeY	
maxBtSize = (treeType == DUAL_TREE_CHROMA) ? MaxBtSizeC : MaxBtSizeY	
if((((x0 + (1 << log2CbSize) <= pic_width_in_luma_samples) ? 1 : 0) + ((y0 + (1 << log2CbSize) <= pic_height_in_luma_samples) ? 1 : 0) + (((1 << log2CbSize) <= maxBtSize) ? 1 : 0)) >= 2 && (1 << log2CbSize) > minQtSize)	
qt_split_cu_flag[x0][y0]	ae(v)
if(cu_qp_delta_enabled_flag && resetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
resetQpDeltaCodedForChildren = (2*(cqtDepth+1) <= diff_cu_qp_delta_depth) ? 1: 0	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType, resetQpDeltaCodedForChildren)	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType, resetQpDeltaCodedForChildren)	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType, resetQpDeltaCodedForChildren)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType, resetQpDeltaCodedForChildren)	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, cqtDepth, 0, 0, 0, treeType, cqtDepth<<1, resetQpDeltaCoded)	
}	

multi_type_tree(x0, y0, cbWidth, cbHeight, cqtDepth, mttDepth, depthOffset, partIdx, treeType, cqgDepth, resetQpDeltaCoded) {	Từ khóa
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
mtt_split_cu_flag	ae(v)
if(cu_qp_delta_enabled_flag && resetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
if(mtt_split_cu_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(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
resetQpDeltaCodedForChildren = ((cqgDepth + 1) <= diff_cu_qp_delta_depth) ? 1 : 0	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	
if(x1 < pic_width_in_luma_samples)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	

} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR)	
{	
depthOffset += (y0 + cbHeight > pic height in luma samples) ? 1 : 0	
y1 = y0 + (cbHeight / 2)	
resetQpDeltaCodedForChildren = ((cqgDepth + 1) <= diff cu qp delta depth) ? 1 : 0	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	
if(y1 < pic height in luma samples)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER)	
{	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
resetQpDeltaCodedForChildren = ((cqgDepth + 2) <= diff cu qp delta depth) ? 1 : 0	
multi_type_tree(x0, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, cqgDepth + 2, resetQpDeltaCodedForChildren)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	
multi_type_tree(x2, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, cqgDepth + 2, resetQpDeltaCodedForChildren)	
} else { /* SPLIT TT HOR */	
y1 = y0 + (cbHeight / 4)	
y2 = y0 + (3 * cbHeight / 4)	
resetQpDeltaCodedForChildren = ((cqgDepth + 2) <= diff cu qp delta depth) ? 1 : 0	
multi_type_tree(x0, y0, cbWidth, cbHeight / 4, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, cqgDepth + 2, resetQpDeltaCodedForChildren)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, cqgDepth + 1, resetQpDeltaCodedForChildren)	
multi_type_tree(x0, y2, cbWidth, cbHeight / 4, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, cqgDepth + 2, resetQpDeltaCodedForChildren)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, treeType)	
}	

Bảng 5

Một biến được gọi là AreaDepth có thể được suy dẫn từ chiều cao cây tứ phân và MTTDepth hoặc MTDepth. Ví dụ, AreaDepth được suy dẫn như sau: $\text{AreaDepth} = J * \text{quadTreeDepth} + ((\text{MTTDepth} + K) \gg N)$ hoặc $\text{AreaDepth} = (L * \text{quadTreeDepth} + \text{MTTDepth})$, trong đó J, K, N, và L ct là bất kỳ số nguyên nào. Ở các bảng cú pháp nêu trên, $\text{cbWidth} * \text{cbHeight}$ có thể được thay thế bằng biến AreaDepth của khối hiện thời. Diện tích CU báo hiệu delta QP nhỏ nhất MinCuQpDeltaArea có thể được thay thế bằng trị số chiều cao (depth) MinCuQpDeltaAreaDepth, trong đó MinCuQpDeltaAreaDepth được báo hiệu ở cấp slice, cấp khung ảnh, cấp chuỗi ảnh, cấp hàng CTU, hoặc cấp tile. Cú pháp báo hiệu ở cấp slice, cấp khung ảnh, cấp chuỗi ảnh, cấp hàng CTU, hoặc cấp tile có thể là diện tích, tỷ lệ diện tích, chiều cao, hoặc log2Area của MinCuQpDeltaArea hoặc trị số của diện tích, tỷ lệ diện tích, chiều cao, hoặc log2Area của MinCuQpDeltaArea tính theo chiều rộng CTU, chiều cao CTU, diện tích CTU, log2Area CTU, hoặc MaxDepth CTU. Tích số của chiều rộng khối và chiều cao khối được chia cho 4 ($\text{cbWidth} * \text{cbHeight} / 4$) có thể được thay thế bằng AreaDepth cộng 2. Theo một phương án, log2 trị số của MinCuQpDeltaArea được báo hiệu phải là số chẵn. Theo phương án khác, trị số liên quan đến MinCuQpDeltaArea được báo hiệu, và trị số được báo hiệu vừa nêu được nhân với 2 để tính ra MinCuQpDeltaArea hoặc log2 MinCuQpDeltaArea. Ví dụ, $\text{MinCuQpDeltaArea} = 1 \ll (\text{signaled_value} * 2)$ hoặc $\text{MinCuQpDeltaArea} = \text{CTU_Diện tích} \gg (\text{signaled_value} * 2)$.

Một thành phần cú pháp, `cu_qp_delta_enabled_flag` được báo hiệu hoặc được suy dẫn để xác định liệu hàm báo hiệu delta QP được kích hoạt hay không. Thành phần cú pháp này có thể được báo hiệu ở cấp chuỗi ảnh, cấp khung ảnh, cấp slice, cấp tile, hoặc cấp hàng CTU. Nếu thành phần cú pháp này bằng 0, delta QP không được báo hiệu làm delta QP không được kích hoạt. Theo một phương án, `cu_qp_delta_enabled_flag` được sử dụng khi suy dẫn biến `IsCuQpDeltaCoded`.

Theo một phương án phân vùng nút phân chia hiện thời thành nhiều nút con, khi diện tích của một nút con nhỏ hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất và diện tích của nút phân chia hiện thời lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, hoặc khi toàn bộ diện tích của các nút con lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, mỗi nút con có thể báo hiệu delta QP của riêng nó. Biến

resetQpDeltaCodedForChildren được thiết lập bằng 1 cho toàn bộ các nút con, và biến IsCuQpDeltaCoded có thể được thiết lập bằng 0 cho toàn bộ các nút con trong trường hợp này. Khi diện tích của nút chia hiện thời bằng hoặc nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất, nhiều nhất một delta QP có thể được báo hiệu cho toàn bộ các nút con vẫn thuộc nút phân chia hiện thời. Biến resetQpDeltaCodedForChildren được thiết lập bằng 0 cho toàn bộ các nút con trong trường hợp này. Theo ví dụ tương tự khác, khi diện tích nút con lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, mỗi một trong số các nút con có thể báo hiệu delta QP của riêng nó. Biến resetQpDeltaCodedForChildren được thiết lập bằng 1 cho các nút con. Theo phương án này, đối với CU lá có kích cỡ bằng hoặc lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, nhiều nhất một delta QP có thể được báo hiệu, và đối với CU lá có kích cỡ nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất nhưng nút cha của nó có kích cỡ lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, nhiều nhất một delta QP có thể được báo hiệu. Đối với nút hiện thời CU không lá, nếu diện tích của một nút con nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất và diện tích của nút hiện thời bằng diện tích CU báo hiệu delta QP nhỏ nhất, nút hiện thời này chia sẻ giá trị delta QP giống nhau cho tất cả các nút con của nó. Nhiều nhất một delta QP có thể được báo hiệu hoặc được phân tích cho nút hiện thời này và QP cuối cùng được suy dẫn từ delta QP được sử dụng bởi một hoặc nhiều nút con. Đối với nút hiện thời CU không lá có kích cỡ nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất nhưng nút cha của nó có kích cỡ lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất, nút hiện thời này chia sẻ giá trị delta QP giống nhau cho toàn bộ các nút con được phân vùng từ nút cha. Nhiều nhất một delta QP có thể được báo hiệu cho nút hiện thời này và QP cuối cùng được suy dẫn từ delta QP được sử dụng bởi một hoặc nhiều nút được phân vùng từ nút cha. Đối với CU lá, nếu biến thiết lập lại resetQpDeltaCoded bằng 1 hoặc nếu biến IsCuQpDeltaCoded bằng 0, nhiều nhất một delta QP có thể được báo hiệu. Ví dụ, đối với nút phân chia hiện thời chẳng hạn CU hoặc TU, nếu biến IsCuQpDeltaCoded bằng 0 và cbf luma hoặc cbf chroma bằng 1, và dạng cây không phải là DUAL_TREE_CHROMA khi lập trình cây song song được kích hoạt, một delta QP được báo hiệu. Vị trí bên trái trên cùng của nút phân chia hiện thời được lưu khi thiết lập lại biến IsCuQpDeltaCoded, khi vị trí được lưu có thể được sử dụng để suy dẫn QP tham chiếu. Delta QP và QP tham chiếu được sử dụng để khôi phục lại QP cuối cùng cho nút phân chia hiện thời.

Theo phương án được lấy làm ví dụ, khi diện tích của một trong số các nút con nhỏ hơn MinCuQpDeltaArea, không có nút con nào trong số các nút con có thể thiết lập biến IsCuQpDeltaCoded bằng 0, và khi các diện tích của toàn bộ các nút con lớn hơn hoặc bằng MinCuQpDeltaArea, mỗi nút con có thể có delta QP của riêng nó. Theo một ví dụ, biến resetQpDeltaCodedForChildren được thiết lập bằng 1 khi các diện tích của toàn bộ các nút con lớn hơn hoặc bằng MinCuQpDeltaArea, nếu không thì, khi diện tích của một nút con nhỏ hơn MinCuQpDeltaArea, không có nút con nào trong số các nút con có thể thiết lập biến IsCuQpDeltaCoded bằng 0, và biến resetQpDeltaCodedForChildren được thiết lập bằng 0. Đối với nút phân chia hiện thời, để xác định giá trị của biến resetQpDeltaCodedForChildren, diện tích nút hiện thời được chia cho 4 được sử dụng khi nút phân chia hiện thời được phân vùng bởi phân chia QT hoặc phân chia TT, hoặc diện tích nút hiện thời được chia cho 2 được sử dụng khi nút phân chia hiện thời được phân vùng bởi phân chia BT.

Nếu kích cỡ CU lớn hơn diện tích CU báo hiệu delta QP nhỏ nhất MinCuQpDeltaArea và CU được phân chia thành một vài TU, và kích cỡ TU bằng hoặc lớn hơn MinCuQpDeltaArea, một delta QP có thể được báo hiệu cho CU có kích cỡ bằng hoặc lớn hơn MinCuQpDeltaArea. Các kích cỡ của CU và các TU, cũng như diện tích CU báo hiệu delta QP nhỏ nhất có thể được suy dẫn hoặc được thể hiện bằng các chiều cao chẳng hạn các MTDepth và chiều cao báo hiệu delta QP lớn nhất.

Lưu đồ đại diện cho báo hiệu delta QP phụ thuộc vào diện tích Fig. 7 là lưu đồ minh họa phương án của hệ thống giải mã video xử lý dữ liệu video phân vùng thành các CU lá theo cấu trúc phân vùng CU, trong đó delta QP được báo hiệu phụ thuộc vào chiều cao thể hiện thể hiện diện tích khối hai chiều. Hệ thống giải mã video nhận dữ liệu vào liên quan đến nút phân chia hiện thời ở bước S702. Nút phân chia hiện thời hoặc là CU lá hoặc là CU không lá được phân vùng từ nút cha bởi dạng phân chia, và nó là khối vuông hoặc khối hình chữ nhật. Chiều cao nút phân chia hiện thời được xác định theo dạng phân chia ở bước S704, ví dụ, chiều cao được xác định từ diện tích của nút phân chia hiện thời. Giá trị của biến IsCuQpDeltaCoded được xác định bằng cách so sánh chiều cao nút phân chia hiện thời với chiều cao tín hiệu delta QP lớn nhất ở bước S706. Chiều cao tín hiệu delta QP lớn nhất có thể được định trước, được suy dẫn, hoặc được báo hiệu ở cấp slice, cấp khung ảnh, cấp tile, hoặc cấp chuỗi ảnh. Giá trị của biến IsCuQpDeltaCoded được kiểm tra ở bước S708, và nếu nó bằng 1,

QP cuối cùng cho nút phân chia hiện thời được suy dẫn từ QP được lập mã trước đó ở bước S714. Nếu $IsCuQpDeltaCoded = 0$, delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích trong TU liên quan đến nút phân chia hiện thời ở bước S710, và QP cuối cùng được khôi phục lại cho một hoặc nhiều TU liên quan đến nút phân chia hiện thời theo QP tham chiếu và delta QP ở bước S712. Hệ thống giải mã video giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời theo QP cuối cùng ở bước S716. Theo một số phương án, vì QP chỉ được cần khi có hệ số khác không hoặc bảng màu dịch chuyển, bước S714 hoặc bước S710 và bước S712 chỉ được tiến hành cho nút phân chia hiện thời khi có ít nhất một hệ số khác không hoặc bảng màu dịch chuyển trong nút phân chia hiện thời.

Theo ví dụ về trường hợp đặc biệt, nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba được phân vùng từ nút cha bởi phân chia TT, và chiều cao nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất nhưng chiều cao của nút phân chia thứ hai được phân vùng từ cùng nút cha bằng chiều cao báo hiệu delta QP lớn nhất, delta QP được suy dẫn từ các thành phần cú pháp được chia sẻ bởi tất cả ba nút phân chia được phân vùng từ nút cha. Theo ví dụ khác của trường hợp đặc biệt, nhiều nhất một delta QP được suy dẫn cho toàn bộ các nút con được phân vùng từ nút phân chia hiện thời bởi phân chia TT nếu chiều cao nút phân chia hiện thời nhỏ hơn chiều cao báo hiệu delta QP lớn nhất và chiều cao của ít nhất một nút con được phân vùng từ nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất. Delta QP được chia sẻ bởi tất cả các nút con được phân vùng cho nút phân chia hiện thời ở ví dụ này.

Mặc dù các phương án và các ví dụ đã nêu là các phương pháp giải mã được thực hiện trong các hệ thống giải mã video, nhưng các phương án của sáng chế cũng có thể được áp dụng cho các hệ thống mã hóa video. Phương pháp mã hóa bao gồm nhận dữ liệu video đầu vào và phân vùng dữ liệu video thành CU lá theo cấu trúc phân vùng CU chẳng hạn QTBT hoặc MTT. Một số bước trong phương pháp mã hóa tương đương với một số bước trong phương pháp giải mã, ví dụ, phương pháp mã hóa còn bao gồm xác định chiều cao của nút phân chia hiện thời theo dạng phân chia và so sánh chiều cao vừa nêu với chiều cao báo hiệu delta QP lớn nhất. Theo kết quả so sánh, QP được xác định cho nút phân chia hiện thời. Ví dụ, nếu nút phân chia hiện thời là CU lá, QP được lập mã trước đó được sử dụng nếu chiều cao nút phân chia hiện thời

lớn hơn chiều cao báo hiệu delta QP lớn nhất; QP mới khác được xác định làm QP cho nút phân chia hiện thời. Delta QP được suy dẫn từ QP và QP tham chiếu nếu QP mới được xác định cho nút phân chia hiện thời, và sau đó một hoặc nhiều thành phần cú pháp liên quan đến delta QP này được báo hiệu ở TU liên quan đến nút phân chia hiện thời, sao cho bộ giải mã có thể phân tích một hoặc nhiều thành phần cú pháp để suy dẫn delta QP. Phương pháp mã hóa này bao gồm mã hóa nút phân chia hiện thời bằng cách sử dụng QP trong quá trình lượng tử hóa.

Báo hiệu QP cho các cấu trúc phân vùng CU luma và chroma riêng Lập trình cây song song được thực hiện để cho phép các cấu trúc phân vùng CU riêng phân chia dữ liệu video của các thành phần luma và chroma. Các phương pháp báo hiệu hoặc suy dẫn QP nêu trên cho các thành phần luma có thể được mở rộng cho báo hiệu hoặc suy dẫn QP cho các thành phần chroma. Theo một phương án, một phần trong số toàn bộ các thông số liên quan đến báo hiệu delta QP bao gồm một hoặc nhiều giá trị ngưỡng, kết quả quyết định, cờ điều khiển từ các nút cha, các điều kiện, và diện tích CU báo hiệu delta QP nhỏ nhất hoặc chiều cao chẳng hạn MinCuQpDeltaArea được nhân đôi cho các thành phần luma và chroma. Một bộ được sử dụng cho các thành phần luma trong khi bộ khác được sử dụng cho các thành phần chroma. Những giá trị ngưỡng và những điều kiện có thể giống nhau trong hai bộ vừa nêu hoặc có thể khác nhau theo các thành phần màu khác nhau.

Theo một số phương án sử dụng lập trình cây song song cho các thành phần luma và chroma, báo hiệu delta QP được bỏ qua cho các thành phần chroma. Ví dụ, báo hiệu delta QP chroma được thay thế bằng báo hiệu sai lệch QP chroma. Tương tự phương pháp báo hiệu delta QP được mô tả trước đó cho các thành phần luma, các sai lệch QP chroma có thể được báo hiệu khi chiều cao của khối chroma nhỏ hơn hoặc bằng giá trị ngưỡng cho báo hiệu QP. Theo một phương án, chiều cao được tính bằng diện tích, và giá trị ngưỡng cho báo hiệu QP cũng được tính bằng chiều cao, ví dụ chiều cao báo hiệu QP lớn nhất cho các thành phần chroma. Một số phương án của sáng chế tái sử dụng QP luma tương ứng cho các thành phần chroma khi báo hiệu delta QP được bỏ qua cho các thành phần chroma. Tức là, chỉ một delta QP được báo hiệu hoặc được phân tích cho các thành phần luma và QP cho các thành phần luma được xác định theo delta QP. Delta QP chroma được suy dẫn từ delta QP của khối luma đã sắp xếp theo một phương án, và QP chroma được suy dẫn từ QP của khối

luma đã sắp xếp theo phương án khác. Phương án tái sử dụng delta QP luma hoặc các QP luma cho các thành phần chroma cho phép hoặc không cho phép quá trình tái sử dụng ở cấp chuỗi ảnh, cấp khung ảnh, cấp tile, hoặc cấp slice và các delta QP riêng được báo hiệu hoặc được phân tích cho các thành phần luma và chroma nếu việc tái sử dụng không được kích hoạt.

Theo một số phương án tái sử dụng QP luma tương ứng cho các thành phần chroma, các sai lệch QP chroma được báo hiệu thay cho các delta QP chroma, trong đó báo hiệu sai lệch QP chroma cho các thành phần chroma có thể tương tự như báo hiệu Delta QP cho các thành phần luma. Sai lệch QP chroma là sai biệt giữa QP chroma của khối chroma hiện thời và QP luma tham chiếu. Vì hai cấu trúc phân vùng CU riêng được sử dụng để phân chia dữ liệu video của các thành phần luma và chroma khi lập trình cây song song được kích hoạt, các phương pháp phân chia khác có thể được sử dụng để phân chia các CTB luma và chroma thuộc cùng CTU. Ví dụ, CTB luma trước tiên được phân vùng bởi phân chia QT, và sau đó hai phân vùng QT trên cùng được chia tiếp bởi phân chia BT đứng; tuy nhiên, các CTB chroma tương ứng được phân vùng chỉ bởi phân chia BT đứng. Ở ví dụ về lập trình cây song song, mỗi CB lá chroma tương ứng với ba CB lá luma thay vì một CB lá luma, và vì mỗi một trong số ba CB lá luma có thể có QP riêng của nó, nên một trong số các QP được chọn làm QP luma tham chiếu để suy dẫn QP chroma. Các phương án của sáng chế xác định QP luma tham chiếu cho khối chroma hiện thời bằng cách sắp xếp khối luma đã sắp xếp khi lập trình cây song song được sử dụng để phân chia các thành phần luma và chroma. QP luma tham chiếu là QP luma của khối luma đã sắp xếp, chẳng hạn CU, PU, hoặc TU đã sắp xếp, hoặc QP luma của bộ delta QP đã sắp xếp. Một số ví dụ của khối luma đã sắp xếp bao gồm khối, CU, PU, hoặc TU mà bao lấy mẫu đã sắp xếp của một mẫu định trước của khối chroma hiện thời. Các ví dụ về mẫu định trước có thể là mẫu bên trái trên cùng, mẫu bên phải trên cùng, mẫu bên trái dưới cùng, mẫu bên phải dưới cùng, hoặc mẫu ở chính giữa của khối chroma hiện thời. QP chroma cuối cùng được khôi phục lại bằng sai lệch QP chroma được báo hiệu và QP luma tham chiếu thu được từ khối luma đã sắp xếp.

Theo một số phương án khác, hai delta QP riêng được báo hiệu cho các thành phần luma và chroma một cách tương ứng. Thành phần cú pháp để định giá trị ngưỡng cho báo hiệu delta QP, chẳng hạn diện tích CU báo hiệu delta QP nhỏ nhất

MinCuQpDeltaArea, hoặc Log 2 trị số của diện tích nhỏ nhất hoặc kích cỡ nhỏ nhất, Log2MinCuQpDeltaArea hoặc Log2MinCuQpDeltaSize, có thể cũng được lập mã riêng biệt ở cấp chuỗi ảnh, cấp khung ảnh, cấp tile, hoặc cấp slice cho các thành phần luma và chroma. Theo một phương án, delta QP chroma hoặc QP chroma của khối chroma được dự đoán từ delta QP luma hoặc QP luma của khối luma đã sắp xếp. Các bảng cú pháp được thể hiện trên Bảng 6 minh họa ví dụ về báo hiệu delta QP riêng cho các thành phần luma và chroma khi lập trình cây song song được sử dụng.

<code>coding_tree_unit() {</code>	Từ khóa
<code>xCtb = (CtbAddrInRs % PicChiều rộngInCtbsY) << CtbLog2SizeY</code>	
<code>yCtb = (CtbAddrInRs / PicChiều rộngInCtbsY) << CtbLog2SizeY</code>	
<code>if(slice_type == I && qtbtt_dual_tree_intra_flag) {</code>	
<code> dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0, 1, 1)</code>	
<code>else</code>	
<code> coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE, 1)</code>	
<code>}</code>	

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth, qtResetQpDeltaCoded, qtResetQpDeltaCodedChroma) {	Từ khóa
nếu (cu_qp_delta_enabled_flag && qtResetQpDeltaCoded) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
nếu (cu_qp_delta_enabled_flag && qtResetQpDeltaCodedChroma) {	
IsCuQpDeltaCodedChroma = 0	
CuQpDeltaValChroma = 0	
}	
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
qtResetQpDeltaCoded = (cbWidth*cbHeight/4 >= MinCuQpDeltaArea && qtResetQpDeltaCoded) ? 1: 0	
qtResetQpDeltaCodedChroma = (cbWidth*cbHeight/4 >= MinCuQpDeltaAreaChroma && qtResetQpDeltaCodedChroma) ? 1: 0	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded, qtResetQpDeltaCodedChroma)	
if(x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded, qtResetQpDeltaCodedChroma)	
if(y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded, qtResetQpDeltaCodedChroma)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1, qtResetQpDeltaCoded, qtResetQpDeltaCodedChroma)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA, qtResetQpDeltaCoded)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA, qtResetQpDeltaCodedChroma)	
}	
}	

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType, qtResetQpDeltaCoded) {	Từ khóa
if((((x0 + (1 << log2CbSize) <= pic_width_in_luma_samples) ? 1 : 0) ((y0 + (1 << log2CbSize) <= pic_height_in_luma_samples) ? 1 : 0) + (((1 << log2CbSize) <= MaxBtSizeY) ? 1 : 0)) >= 2 && log2CbSize > MinQtLog2SizeY)	
qt_split_cu_flag[x0][y0]	ae(v)
nếu (cu_qp_delta_enabled_flag && qtResetQpDeltaCoded) {	
IsCuQpDeltaCoded = (treeType != DUAL_TREE_CHROMA)? 0 : IsCuQpDeltaCoded	
CuQpDeltaVal = (treeType != DUAL_TREE_CHROMA)? 0 : CuQpDeltaVal	
IsCuQpDeltaCodedChroma = (treeType == DUAL_TREE_CHROMA)? 0 : IsCuQpDeltaCodedChroma	
CuQpDeltaValChroma = (treeType == DUAL_TREE_CHROMA)? 0 : CuQpDeltaValChroma	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
qtResetQpDeltaCoded = (cbWidth*cbHeight/4 >= ((treeType != DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) ? qtResetQpDeltaCoded : 1: 0	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType, qtResetQpDeltaCoded)	
} else {	

mttResetQpDeltaCoded = ((cbWidth*cbHeight >= ((treeType != DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) && qtResetQpDeltaCoded) ? 1 : 0	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, treeType, mttResetQpDeltaCoded)	
}	
}	

multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, treeType, mttResetQpDeltaCoded) {	Từ khóa
if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
mtt_split_cu_flag	ae(v)
nếu (cu_qp_delta_enabled_flag && mttResetQpDeltaCoded) {	
IsCuQpDeltaCoded = (treeType != DUAL_TREE_CHROMA)? 0 : IsCuQpDeltaCoded	
CuQpDeltaVal = (treeType != DUAL_TREE_CHROMA)? 0 : CuQpDeltaVal	
IsCuQpDeltaCodedChroma = (treeType == DUAL_TREE_CHROMA)? 0 : IsCuQpDeltaCodedChroma	
CuQpDeltaValChroma = (treeType == DUAL_TREE_CHROMA)? 0 : CuQpDeltaValChroma	
}	
if(mtt_split_cu_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(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > pic_width_in_luma_samples) ? 1 : 0	
x1 = x0 + (cbWidth / 2)	
mttResetQpDeltaCoded = ((cbWidth*cbHeight/2 >= ((treeType != DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) && mttResetQpDeltaCoded) ? 1 : 0	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)	
if(x1 < pic_width_in_luma_samples)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > pic_height_in_luma_samples) ? 1 : 0	

$y1 = y0 + (cbHeight / 2)$	
$mttResetQpDeltaCoded = ((cbWidth*cbHeight/2 \geq ((treeType \neq DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) \&\& mttResetQpDeltaCoded) ? 1 : 0$	
$multi_type_tree(x0, y0, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)$	
$if(y1 < pic_height_in_luma_samples)$	
$multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)$	
$\} \text{ else if(MttSplitMode[} x0 \text{][} y0 \text{][} mttDepth \text{]} = \text{ SPLIT_TT_VER })$ $\{$	
$x1 = x0 + (cbWidth / 4)$	
$x2 = x0 + (3 * cbWidth / 4)$	
$mttResetQpDeltaCoded = ((cbWidth*cbHeight/4 \geq ((treeType \neq DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) \&\& mttResetQpDeltaCoded) ? 1 : 0$	
$multi_type_tree(x0, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)$	
$multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)$	
$multi_type_tree(x2, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 2, treeType, mttResetQpDeltaCoded)$	
$\} \text{ else } \{ /* \text{ SPLIT_TT_HOR } */$	
$y1 = y0 + (cbHeight / 4)$	
$y2 = y0 + (3 * cbHeight / 4)$	
$mttResetQpDeltaCoded = ((cbWidth*cbHeight/4 \geq ((treeType \neq DUAL_TREE_CHROMA)? MinCuQpDeltaArea : MinCuQpDeltaAreaChroma)) \&\& mttResetQpDeltaCoded) ? 1 : 0$	
$multi_type_tree(x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 0, treeType, mttResetQpDeltaCoded)$	
$multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, treeType, mttResetQpDeltaCoded)$	

multi_type_tree(x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 2 , treeType, mttResetQpDeltaCoded)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, treeType)	
}	

Bảng 6

Các thành phần cú pháp của báo hiệu delta QP chroma ở cấp TU có thể tương tự như các thành phần cú pháp của HEVC được sử dụng cho báo hiệu delta QP luma, và được thể hiện theo công thức sau:

```

if ( cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {

    cu_qp_delta_abs

    if( cu_qp_delta_abs )

        cu_qp_delta_sign_flag

}

```

Trị số delta QP chroma CuQpDeltaVal được suy dẫn dựa vào biến tuyệt đối cu_qp_delta_abs và cờ dấu cu_qp_delta_sign_flag. QP chroma cuối cùng được khôi phục lại theo QP chroma tham chiếu và CuQpDeltaVal. Cờ dấu cu_qp_delta_sign_flag được suy bằng 0 nếu nó không có mặt. Khi biến tuyệt đối cu_qp_delta_abs có mặt, các biến IsCuQpDeltaCoded được thiết lập bằng 1 và trị số delta QP chroma CuQpDeltaVal được suy dẫn như sau :

IsCuQpDeltaCoded = 1

CuQpDeltaVal = cu_qp_delta_abs * (1 - 2 * cu_qp_delta_sign_flag)

Hàm tính delta QP chroma được gọi ra khi dạng cây là cây song song và có ít nhất một hệ số khác không trong TU chroma. Các bảng cú pháp được lấy làm ví dụ được thể hiện trên Bảng 7.

transform_unit(x0, y0, tbWidth, tbHeight, treeType) {	Từ khóa
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA)	
tu_cbf_luma[x0][y0]	ae(v)

if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) {	
tu_cbf_cb[x0][y0]	ae(v)
tu_cbf_cr[x0][y0]	ae(v)
}	
if((treeType == SINGLE_TREE && (tu_cbf_luma[x0][y0] tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0])) (treeType == DUAL_TREE_LUMA && tu_cbf_luma[x0][y0])))	
delta_qp()	
if((treeType == DUAL_TREE_CHROMA && (tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0])))	
chroma_delta_qp()	

delta_qp() {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	

chroma_delta_qp() {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCodedChroma) {	
cu_chroma_qp_delta_abs	ae(v)
if(cu_chroma_qp_delta_abs)	
cu_chroma_qp_delta_sign_flag	ae(v)
}	
}	

Bảng 7

Các phương pháp báo hiệu delta QP chroma khác ở cấp TU có thể cũng được sử dụng thay cho phương pháp báo hiệu HEVC hiện có. Báo hiệu delta QP chroma được mô tả ở trên có thể được thay thế bằng báo hiệu sai lệch QP chroma và cờ, IsCuQpDeltaCodedChroma được thay thế bằng IsCuChromaQpOffsetCoded, và ví dụ về báo hiệu sai lệch QP chroma ở cấp TU được thể hiện trên các bảng cú pháp của Bảng 8.

transform_unit(x0, y0, tbWidth, tbHeight, treeType) {	Từ khóa
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA)	
tu_cbf_luma[x0][y0]	ae(v)
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) {	
tu_cbf_cb[x0][y0]	ae(v)
tu_cbf_cr[x0][y0]	ae(v)
}	
if((treeType == SINGLE_TREE && (tu_cbf_luma[x0][y0] tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0])) (treeType == DUAL_TREE_LUMA && tu_cbf_luma[x0][y0]))	
delta_qp()	
if((treeType == DUAL_TREE_CHROMA && (tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0]))	
chroma_qp_offset ()	

chroma_qp_offset() {	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_offset_idx	ae(v)
}	
}	

Bảng 8

Theo phương án này, sai lệch QP chroma của khối chroma hiện thời được suy dẫn từ một hoặc nhiều thành phần cú pháp được báo hiệu ở TU liên quan đến khối chroma hiện thời khi TU có ít nhất một hệ số khác không cho các thành phần chroma. Ví dụ được thể hiện trên Bảng 7 kiểm tra nếu bất kỳ một trong số các thành phần Cb chroma và Cr chroma có ít nhất một hệ số khác không, và sai lệch QP chroma được báo hiệu khi có ít nhất một hệ số khác không cho các thành phần chroma.

Theo một phương án, báo hiệu QP chroma và các toán tử và hàm IsCuChromaQpOffsetCoded và MinCuQpDeltaAreaChroma, được sử dụng để báo hiệu các sai lệch QP chroma nếu lập trình cây chia sẻ được sử dụng. Báo hiệu QP chroma được sử dụng cho báo hiệu delta QP chroma nếu lập trình cây riêng được sử

dụng. Theo phương án khác, cho dù các cấu trúc phân vùng CU cho các thành phần luma và chroma là lập trình cây song song hay cây chia sẻ, báo hiệu QP chroma nêu trên được sử dụng để báo hiệu các delta QP chroma. Tức là, các delta QP riêng cho các thành phần luma và chroma không chỉ được áp dụng cho lập trình cây song song mà còn được áp dụng cho lập trình cây chia sẻ trong đó cấu trúc phân vùng CU được chia sẻ giữa các thành phần luma và chroma.

Theo phương án khác nữa, cho dù các cấu trúc phân vùng CU cho các thành phần luma và chroma là lập trình cây song song hay lập trình cây chia sẻ, báo hiệu QP chroma nêu trên được sử dụng để báo hiệu các sai lệch QP chroma. Trong những trường hợp khi phép chuyển đổi và lượng tử hóa được bỏ qua, tức là khi cờ `cu_transquant_bypass_flag` là đúng, delta QP và/hoặc báo hiệu sai lệch QP chroma có thể được bỏ qua.

Theo một số phương án của báo hiệu delta QP chroma, cờ, `diff_cu_qp_delta_depth_c`, được báo hiệu để xác định sai biệt giữa kích cỡ khối cây mã hóa chroma và kích cỡ khối mã hóa chroma nhỏ nhất của các đơn vị mã hóa cho báo hiệu delta QP chroma. Tương tự báo hiệu delta QP luma, delta QP chroma có thể được suy dẫn từ các thành phần cú pháp `cu_qp_delta_abs` và `cu_qp_delta_sign_flag` cho các thành phần chroma. Giá trị của cờ `diff_cu_qp_delta_depth_c` bị giới hạn trong phạm vi từ 0 đến `log2_diff_max_min_luma_coding_block_size`. Khi cờ này không có mặt, giá trị được suy bằng 0, điều đó có nghĩa là kích cỡ nhỏ nhất cho báo hiệu delta QP chroma bằng kích cỡ khối cây mã hóa chroma.

Theo một phương án, cờ được báo hiệu để biểu thị liệu kích cỡ nhỏ nhất cho báo hiệu delta QP chroma, được thể hiện bằng `MinCuQpDeltaAreaC` hoặc `Log2MinCuQpDeltaAreaC`, có cùng tỷ lệ với tỷ lệ mẫu phụ chroma. Ví dụ, theo định dạng màu 4:2:0, cờ này biểu thị liệu `MinCuQpDeltaAreaC` cho các thành phần chroma bằng một phần tư `MinCuQpDeltaArea` cho các thành phần luma, hay liệu `Log2MinCuQpDeltaAreaC` bằng `Log2MinCuQpDeltaArea-2`.

Theo phương án khác, ba delta QP riêng được báo hiệu cho mỗi một trong số ba thành phần màu, luma Y, chroma Cb, và chroma Cr nếu khả thi. Ví dụ, ba delta QP được báo hiệu cho ba thành phần màu khi `ChromaArrayType` lớn hơn 0.

Lưu đồ tái sử dụng QP Luma cho Chroma Fig. 8 là lưu đồ minh họa phương án của xử lý dữ liệu video trong hệ thống mã hóa hoặc giải mã video tái sử dụng QP luma cho khối chroma hiện thời. Khối chroma hiện thời có thể là CB chroma trong CU lá hoặc CU không lá. Hệ thống mã hóa hoặc giải mã video nhận dữ liệu video vào của khối chroma hiện thời ở bước S802. Dữ liệu video đầu vào được phân vùng theo hai cấu trúc phân vùng CU riêng cho các thành phần luma và chroma khi lập trình cây song song được kích hoạt. QP luma của khối luma đã sắp xếp được xác định ở bước S804 bằng cách định vị khối đã sắp xếp mà bao lấy mẫu đã sắp xếp của một mẫu định trước của khối chroma hiện thời. Một số ví dụ của khối luma đã sắp xếp bao gồm CU, PU, TU đã sắp xếp, hoặc bộ delta QP của khối chroma hiện thời. Một số ví dụ về mẫu định trước của khối chroma hiện thời bao gồm mẫu bên trái trên cùng, mẫu bên phải trên cùng, mẫu bên trái dưới cùng, mẫu bên phải dưới cùng, hoặc mẫu ở chính giữa của khối chroma hiện thời. Ở bước S806, QP luma của khối luma đã sắp xếp được tái sử dụng để suy dẫn QP chroma cho khối chroma hiện thời. Ví dụ, QP chroma được suy dẫn bởi QP luma và sai lệch QP chroma được xác định từ một hoặc nhiều thành phần cú pháp được báo hiệu trong TU liên quan đến khối chroma hiện thời. Sai lệch QP chroma có thể được báo hiệu bằng cách sử dụng phương pháp tương tự như phương pháp báo hiệu delta QP được mô tả trước đó cho các thành phần luma, ví dụ, sai lệch QP chroma được suy dẫn cho khối chroma hiện thời khi chiều cao của khối chroma hiện thời nhỏ hơn hoặc bằng giá trị ngưỡng cho báo hiệu QP. Các sai lệch QP chroma và các delta QP luma được báo hiệu riêng khi lập trình cây song song được kích hoạt làm các QG chroma để chia sẻ các sai lệch QP chroma có thể khác với QG luma để chia sẻ delta QP. Diện tích của khối chroma hiện thời và giá trị ngưỡng cho báo hiệu QP có thể được tính bằng các chiều cao. Một hoặc nhiều TU liên quan đến khối chroma hiện thời được mã hóa hoặc được giải mã bằng cách sử dụng QP chroma ở bước S808.

Suy dẫn QP tham chiếu Khó xác định Nhóm lượng tử hóa (QG) của khối mã hóa hiện thời và nhóm lượng tử hóa trước trong lệnh giải mã khi suy dẫn QP tham chiếu cho khối mã hóa theo phân vùng cây đa phân. Theo một phương án, QP slice luôn được sử dụng làm QP tham chiếu cho các khối được phân vùng bởi cấu trúc phân vùng QTBT hoặc MTT. Theo phương án khác, QP slice được sử dụng làm QP tham chiếu cho sự phân vùng hoặc QT, nhưng trong trường hợp của phân vùng QT, phương

pháp trong HEVC được sử dụng để suy dẫn QP tham chiếu. Theo phương án khác nữa, QP của nhóm lượng tử hóa trước trong lệnh giải mã không được sử dụng làm tham chiếu, QP của nhóm lượng tử hóa lập trình liền kề được sử dụng hoặc QP slice được sử dụng nếu nhóm lượng tử hóa lập trình liền kề không khả dụng. Theo phương án khác nữa, QP của nhóm lượng tử hóa trước trong lệnh giải mã không được sử dụng làm tham chiếu, nhưng QP được lập mã cuối cùng trong CTU cuối cùng được sử dụng nếu nhóm lượng tử hóa lập trình liền kề không khả dụng. Trong trường hợp khi QP được lập mã cuối cùng trong CTU cuối cùng cũng không khả dụng, QP slice được sử dụng.

Theo một số phương án khác, QP tham chiếu được suy dẫn theo độ khả dụng của QP bên trái và QP phía trên. Ví dụ, giá trị trung bình của QP bên trái và QP phía trên được sử dụng làm QP tham chiếu nếu các QP bên trái và phía trên cùng khả dụng, nếu không thì QP bên trái được sử dụng làm QP tham chiếu nếu QP bên trái khả dụng hoặc QP phía trên được sử dụng làm QP tham chiếu nếu QP phía trên khả dụng. Nếu không một QP nào trong số các QP bên trái và phía trên khả dụng, QP slice được sử dụng làm QP tham chiếu. Theo ví dụ khác, QP bên trái được kiểm tra trước tiên và được sử dụng làm QP tham chiếu nếu nó khả dụng, nếu không thì QP đã nêu được sử dụng làm QP tham chiếu nếu nó khả dụng, và QP slice được sử dụng làm QP tham chiếu nếu cả QP bên trái và phía trên không khả dụng. Theo ví dụ khác nữa, QP đã nêu được kiểm tra trước tiên và được sử dụng làm QP tham chiếu nếu nó khả dụng, nếu không thì QP bên trái được sử dụng làm QP tham chiếu nếu nó khả dụng, và QP slice được sử dụng làm QP tham chiếu nếu cả QP bên trái và phía trên không khả dụng.

Theo một phương án, độ khả dụng của QP lập trình liền kề phụ thuộc vào địa chỉ CTU. QP lập trình liền kề khả dụng nếu QP lập trình liền kề và QP lập trình hiện thời ở cùng CTU; nếu không thì QP lập trình liền kề không khả dụng. Theo phương án khác, độ khả dụng của QP lập trình liền kề phụ thuộc vào địa chỉ hàng CTU. Ví dụ, QP lập trình liền kề khả dụng nếu QP lập trình liền kề và QP lập trình hiện thời ở cùng hàng CTU; nếu không thì, QP lập trình liền kề không khả dụng. Theo một phương án, QP tham chiếu được suy dẫn cho QG. Nhiều nhất một delta QP có thể được báo hiệu trong QG. Kích cỡ hoặc hình dạng của QG có thể là không vuông. QG có thể chứa một hoặc nhiều CU. Thứ tự mã hóa hoặc giải mã các CU trong QG là liên tục. Khi hệ thống giải mã video đang giải mã các CU trong QG, nếu không có số dư khác không

cũng như không có bảng màu dịch chuyển ở CU và delta QP không được báo hiệu trong QG này, thì QP của CU đó được thiết lập làm QP tham chiếu. Nếu CU có ít nhất một số dư khác không hoặc ít nhất một bảng màu dịch chuyển, và delta QP vẫn chưa được báo hiệu trong QG vừa nêu, thì delta QP được báo hiệu. QP được suy dẫn từ QP tham chiếu và delta QP. QP được khôi phục lại được sử dụng cho CU vừa nêu và tất cả các CU còn lại trong QG vừa nêu. Nếu CU có ít nhất một số dư khác không hoặc ít nhất một bảng màu dịch chuyển, và delta QP đã được báo hiệu trong QG vừa nêu, thì delta QP không được báo hiệu.

Phân vùng h

oặc xác định Nhóm lượng tử hóa Có ba sơ đồ, sơ đồ A, sơ đồ B, và sơ đồ C, được giải thích dưới đây cho phân vùng hoặc xác định Nhóm lượng tử hóa (QG). Giá trị ngưỡng cho báo hiệu QP MinCuQpDeltaArea được định rõ để xác định QG. QG thể hiện CU vùng, nút, hoặc lá có biến IsCuQpDeltaCoded bằng 0.

Sơ đồ A Theo ví dụ về sơ đồ A, đối với nút phân chia hiện thời có kích cỡ lớn hơn hoặc bằng giá trị ngưỡng MinCuQpDeltaArea, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho nút phân chia hiện thời, và đối với nút phân chia hiện thời có kích cỡ nhỏ hơn MinCuQpDeltaArea và kích cỡ của nút cha lớn hơn MinCuQpDeltaArea, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho nút phân chia hiện thời. Nếu kích cỡ của nút phân chia hiện thời nhỏ hơn MinCuQpDeltaArea và kích cỡ của nút cha bằng hoặc nhỏ hơn MinCuQpDeltaArea, IsCuQpDeltaCoded không được thiết lập bằng 0. Theo ví dụ khác, nếu kích cỡ của nút phân chia hiện thời bằng hoặc nhỏ hơn MinCuQpDeltaArea và kích cỡ của nút cha của nó lớn hơn MinCuQpDeltaArea, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho nút phân chia hiện thời. Theo ví dụ khác nữa, đối với nút phân chia hiện thời được phân vùng từ nút cha có kích cỡ lớn hơn MinCuQpDeltaArea, biến CuQpDeltaCoded được thiết lập bằng 0 cho nút hiện thời.

Theo ví dụ khác của sơ đồ A, nếu kích cỡ của nút phân chia hiện thời lớn hơn MinCuQpDeltaArea và kích cỡ của một trong số các nút con lớn hơn hoặc bằng MinCuQpDeltaArea, biến IsCuDeltaCoded được thiết lập bằng 0 cho toàn bộ các nút con; nếu không thì, biến IsCuDeltaCoded không được thiết lập bằng 0 cho toàn bộ các nút con nếu nút phân chia hiện thời nhỏ hơn hoặc bằng MinCuQpDeltaArea hoặc các kích cỡ của tất cả các nút con nhỏ hơn MinCuQpDeltaArea. Theo ví dụ khác, nếu kích

cỡ của nút phân chia hiện thời lớn hơn MinCuQpDeltaArea, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho toàn bộ các nút con; nếu không thì, biến IsCuQpDeltaCoded không được thiết lập bằng 0 cho tất cả các nút con nếu nút phân chia hiện thời nhỏ hơn hoặc bằng MinCuQpDeltaArea.

Các ví dụ về sơ đồ A nêu trên có thể được mô tả bằng ý tưởng xác định QG. Theo một ví dụ, đối với CU lá có kích cỡ lớn hơn hoặc bằng MinCuQpDeltaArea, CU lá là QG. Đối với CU lá có kích cỡ nhỏ hơn MinCuQpDeltaArea và kích cỡ của CU cha của nó lớn hơn MinCuQpDeltaArea, CU lá là QG. Đối với CU không lá hoặc nút hiện thời, nếu kích cỡ của CU không lá hoặc nút hiện thời bằng MinCuQpDeltaArea, CU không lá hoặc nút hiện thời vừa nêu là QG. Đối với CU không lá hoặc nút hiện thời, nếu kích cỡ của nó nhỏ hơn MinCuQpDeltaArea và kích cỡ của CU cha của nó hoặc nút cha lớn hơn MinCUQpDeltaArea, CU không lá hoặc nút hiện thời vừa nêu là QG. Nếu không thì, CU lá hoặc CU không lá vừa nêu hoặc nút hiện thời không là QG.

Theo ví dụ khác, đối với CU lá có kích cỡ lớn hơn hoặc bằng MinCuQpDeltaArea, CU lá là QG, hoặc nếu kích cỡ nhỏ hơn MinCuQpDeltaArea và kích cỡ của CU cha của nó lớn hơn MinCuQpDeltaArea, CU lá là QG; nếu không thì, CU lá vừa nêu không là QG. Đối với CU không lá, nếu kích cỡ của nó bằng MinCuQpDeltaArea, hoặc nếu kích cỡ của nó nhỏ hơn MinCuQpDeltaArea và kích cỡ của CU cha của nó lớn hơn MinCuQpDeltaArea, CU không lá vừa nêu là QG; nếu không thì, CU không lá vừa nêu không là QG.

Theo ví dụ khác của sơ đồ A, đối với nút hiện thời có kích cỡ bằng hoặc nhỏ hơn MinCuQpDeltaArea, và nút cha của nó không là QG, nút hiện thời là QG. Đối với CU lá, nếu kích cỡ của nó lớn hơn hoặc bằng MinCuQpDeltaArea, CU lá là QG. Theo ví dụ khác, đối với nút hiện thời có kích cỡ bằng hoặc nhỏ hơn MinCuQpDeltaArea, và nút cha của nó không là QG, nút hiện thời là QG. Đối với CU lá, nếu nút cha của nó không là QG, CU lá là QG.

Đối với QG, biến IsCuQpDeltaCoded được thiết lập bằng 0. QP tham chiếu hoặc còn được gọi là QP dự đoán được suy dẫn cho QG. Ví dụ, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho QG và vị trí của QG này được lưu, ví dụ về một vị trí là vị trí bên trái trên cùng của QG. Vị trí được lưu có thể được sử dụng để suy dẫn QP tham chiếu. Theo ví dụ khác, đối với QG, biến IsCuQpDeltaCoded được thiết lập bằng 0 và

QP tham chiếu được suy dẫn. Theo ví dụ khác, biến IsCUQpDeltaCoded được thiết lập bằng 0 tại CU lá đầu tiên trong QG, và QP tham chiếu được suy dẫn tại CU lá đầu tiên trong QG.

Ở sơ đồ A, nếu kích cỡ của CU hoặc nút hiện thời bằng $2 * \text{MinCuQpDeltaArea}$ và phân chia QT hoặc phân chia TT được áp dụng để phân chia CU hoặc nút hiện thời, mỗi nút con là QG. Biến IsCuQpDeltaCoded được thiết lập bằng 0 cho toàn bộ các nút con. Nếu kích cỡ của CU hoặc nút hiện thời bằng MinCuQpDeltaArea , CU hoặc nút hiện thời là QG. Biến IsCuQpDeltaCoded được thiết lập bằng 0 cho CU hoặc nút hiện thời. Nếu kích cỡ của CU lá hiện thời lớn hơn MinCuQpDeltaArea , CU lá hiện thời là QG. Biến IsCuQpDeltaCoded được thiết lập bằng 0 cho CU lá hiện thời. Nếu kích cỡ của CU hoặc nút hiện thời lớn hơn hoặc bằng MinCuQpDeltaArea , biến IsCuQpDeltaCoded được thiết lập bằng 0 cho CU hoặc nút hiện thời.

Sơ đồ B Ở sơ đồ B xác định QG, một số phương án xác định liệu các nút chia của nút cha là các QG ở nút cha. Ví dụ về sơ đồ B thiết lập biến IsCuQpDeltaCoded bằng 0 cho toàn bộ các nút con của nút hiện thời nếu kích cỡ của nút con nhỏ nhất lớn hơn hoặc bằng giá trị ngưỡng cho báo hiệu QP MinCuQpDeltaArea . Theo ví dụ khác, biến IsCuQpDeltaCoded được thiết lập bằng 0 cho toàn bộ các nút con của nút hiện thời nếu các kích cỡ của tất cả các nút con lớn hơn hoặc bằng MinCuQpDeltaArea . Theo ví dụ khác, biến được thiết lập bằng 0 cho toàn bộ các nút con của nút hiện thời nếu các kích cỡ của tất cả các nút con lớn hơn hoặc bằng MinCuQpDeltaArea ; nếu không thì nếu kích cỡ của một nút con nhỏ hơn MinCuQpDeltaArea , biến IsCuQpDeltaCoded không được thiết lập bằng 0 cho toàn bộ các nút con. Không một nút chia nào trong số các nút chia có thể thiết lập biến IsCuQpDeltaCoded to 0.

QG thể hiện CU vùng, nút, hoặc lá có biến IsCuQpDeltaCoded được thiết lập bằng 0. Theo một ví dụ, CU lá là QG nếu kích cỡ của nó lớn hơn giá trị ngưỡng MinCuQpDeltaArea . Theo ví dụ khác, nút phân chia hiện thời là QG nếu kích cỡ của nó lớn hơn MinCuQpDeltaArea và Kích cỡ của một nút con của nút phân chia hiện thời nhỏ hơn MinCuQpDeltaArea . Theo ví dụ khác, nút phân chia hiện thời là QG nếu nút phân chia hiện thời có kích cỡ lớn hơn MinCuQpDeltaArea và kích cỡ của một nút con của nút phân chia hiện thời nhỏ hơn MinCuQpDeltaArea . Đối với nút hiện thời, nếu các kích cỡ của tất cả các nút con lớn hơn hoặc bằng MinCuQpDeltaArea , nút phân chia hiện thời không là QG. Theo ví dụ khác, CU không lá là QG nếu nút cha của

nó không là QG và Kích cỡ của một nút con nhỏ hơn MinCuQpDeltaArea. CU lá là QG nếu kích cỡ của nó lớn hơn hoặc bằng MinCuQpDeltaArea và nút cha của nó không là QG.

Theo ví dụ khác của Sơ đồ B, nút phân chia hiện thời là QG nếu nút cha của nó không là QG và kích cỡ của một nút con của nút phân chia hiện thời nhỏ hơn giá trị ngưỡng cho báo hiệu QP MinCuQpDeltaArea. Đối với CU lá, nếu kích cỡ của nó lớn hơn MinCuQpDeltaArea, CU lá là QG. Theo ví dụ khác, nút phân chia hiện thời là QG nếu nút cha của nó không là QG và kích cỡ của một nút con nhỏ hơn MinCuQpDeltaArea. Đối với CU lá, nếu kích cỡ của nó lớn hơn hoặc bằng MinCuQpDeltaArea, và nút cha của nó không là QG, CU lá là QG. Theo ví dụ khác nữa, nút phân chia hiện thời là QG nếu nút cha của nó không là QG và Kích cỡ của một nút con nhỏ hơn MinCuQpDeltaArea. Đối với CU lá, nếu nút cha của nó không là QG, CU lá là QG.

Đối với QG, biến IsCuQpDeltaCoded được thiết lập bằng 0. QP tham chiếu hoặc QP dự đoán được suy dẫn cho QG. Ví dụ, biến IsCuQpDeltaCoded được thiết lập bằng 0 và vị trí chẳng hạn vị trí bên trái trên cùng của QG được lưu. Vị trí được lưu có thể được sử dụng để suy dẫn QP tham chiếu. Theo ví dụ khác, biến IsCuQpDeltaCoded được thiết lập bằng 0 và QP tham chiếu được suy dẫn đối với QG. Theo ví dụ khác, biến IsCuQpDeltaCoded được thiết lập bằng 0 tại CU lá thứ nhất trong QG, và QP tham chiếu được suy dẫn tại CU lá thứ nhất trong QG.

Ở sơ đồ B, nút phân chia hiện thời là QG nếu kích cỡ nút phân chia hiện thời bằng hai lần giá trị ngưỡng MinCuQpDeltaArea và phân chia QT hoặc TT được áp dụng để phân vùng nút phân chia hiện thời. Biến IsCuQpDeltaCoded được thiết lập bằng 0 cho nút phân chia hiện thời. Nếu kích cỡ CU lá hiện thời lớn hơn MinCuQpDeltaArea, CU lá hiện thời là QG, và biến IsCuQpDeltaCoded được thiết lập bằng 0 cho CU lá hiện thời.

Sơ đồ C Sơ đồ C tương tự như sơ đồ B, ngoại trừ khi kích cỡ nút phân chia hiện thời bằng hai lần giá trị ngưỡng MinCuQpDeltaArea và phân chia QT được áp dụng, hai phân vùng QT đầu tiên sẽ được nhóm làm QG trong khi hai phân vùng QT cuối cùng sẽ được nhóm làm QG khác. Đối với QG, biến IsCuQpDeltaCoded được thiết lập bằng 0 và QP tham chiếu hoặc QP dự đoán được suy dẫn cho QG. Ví dụ, đối với

QG, biến IsCuQpDeltaCoded được thiết lập bằng 0 và vị trí, chẳng hạn vị trí bên trái trên cùng, của QG vừa nêu được lưu. Vị trí được lưu có thể được sử dụng để suy dẫn QP tham chiếu. Theo ví dụ khác, đối với QG, biến IsCuQpDeltaCoded được thiết lập bằng 0 và QP tham chiếu được suy dẫn. Theo ví dụ khác, biến IsCuQpDeltaCode được thiết lập bằng 0 tại CU lá thứ nhất trong QG, và QP tham chiếu được suy dẫn tại CU lá thứ nhất trong QG.

Ý tưởng của các phương pháp phân vùng QG cũng có thể được áp dụng để chia sẻ danh mục trộn, vùng ước lượng trộn MER (Merge Estimation Region - MER), hoặc dự đoán biên giới chung nội ảnh.

Các phương án của sáng chế có thể được thực hiện trong các bộ mã hóa video và/hoặc các bộ giải mã video. Ví dụ, các phương pháp đã nêu có thể được thực hiện trong môđun giải mã entropy, môđun lập trình hệ số, môđun lượng tử hóa, hoặc môđun điều khiển phân vùng của bộ mã hóa video, và/hoặc môđun phân tích entropy, môđun giải mã hệ số, môđun phi lượng tử hóa, hoặc môđun điều khiển phân vùng của bộ giải mã video. Ngoài ra, bất kỳ một trong số các phương pháp đã nêu có thể được thực hiện dưới dạng mạch tích hợp với môđun mã hóa entropy, môđun lập trình hệ số, môđun lượng tử hóa, hoặc môđun điều khiển phân vùng của bộ mã hóa video và/hoặc môđun phân tích entropy, môđun giải mã hệ số, môđun phi lượng tử hóa, hoặc môđun điều khiển phân vùng của bộ giải mã video, để cấp thông tin cần bởi môđun phân tích entropy hoặc môđun điều khiển phân vùng. Các bộ mã hóa video phải có cấu trúc cú pháp đã nêu để tạo ra dòng bit đúng, và các bộ giải mã video có thể giải mã dòng bit một cách chính xác chỉ khi nếu quá trình phân tích được tuân theo cấu trúc cú pháp đã nêu. Khi bất kỳ thành phần cú pháp nào được bỏ qua trong dòng bit, các bộ mã hóa video và các bộ giải mã thiết lập giá trị của thành phần cú pháp làm giá trị suy luận để đảm bảo các kết quả mã hóa và giải mã được khớp nhau.

Fig. 9 minh họa sơ đồ khối hệ thống được lấy làm ví dụ cho bộ mã hóa video 900 thực hiện nhiều phương án khác nhau của sáng chế. Môđun dự đoán nội ảnh 910 để xuất các biến dự báo nội ảnh dựa vào dữ liệu video được khôi phục lại của khung hình hiện thời. Môđun dự đoán liên ảnh 912 thực hiện ước lượng chuyển động ME (Motion Estimation - ME) và bù chuyển động MC (Motion Compensation - MC) để tạo ra các biến dự báo liên ảnh dựa vào dữ liệu video tham chiếu từ hình hoặc các khung hình khác. Hoặc môđun dự đoán nội ảnh 910 hoặc môđun dự đoán liên ảnh 912 cấp biến dự

báo được chọn của khối hiện thời đến bộ cộng 916 để tạo ra các số dư bằng cách trừ biến dự báo đã chọn từ dữ liệu video gốc của khối hiện thời. Các số dư của khối hiện thời được xử lý tiếp bởi Môđun chuyển đổi (T) 918 và sau đó bởi Môđun lượng tử hóa (Q) 920. Tín hiệu dư thừa được biến đổi và được lượng tử hóa sau đó được mã hóa bởi Bộ mã hóa Entropy 934 để tạo thành dòng bit video. Ở môđun lượng tử hóa 920, các số dư luma được chuyển đổi của mỗi nhóm lượng tử hóa được lượng tử hóa bởi QP luma, và QP luma được xác định từ delta QP. Delta QP được báo hiệu dựa vào chiều cao của nút chia theo một số phương án của sáng chế. Chiều cao của nút chia phụ thuộc vào dạng phân chia. Theo một số phương án khác, QP luma của khối luma đã sắp xếp được tái sử dụng để suy dẫn QP chroma cho khối màu, trong đó khối luma đã sắp xếp là khối bao lấy mẫu đã sắp xếp của một mẫu định trước của khối chroma. Dòng bit video sau đó được chèn thông tin phụ bao gồm delta QP cho mỗi nhóm lượng tử hóa. Tín hiệu dư thừa được biến đổi và được lượng tử hóa của khối hiện thời được xử lý bởi môđun lượng tử hóa ngược (IQ) 922 và môđun chuyển đổi ngược (IT) 924 để khôi phục các số dư dự đoán. Như được thể hiện trên Fig. 9, các số dư được khôi phục bằng cách thêm hậu tố vào biến dự báo đã chọn tại môđun khôi phục lại (REC) 926 để tạo ra dữ liệu video được khôi phục lại. Dữ liệu video được khôi phục lại có thể được lưu trong Bộ đệm khung hình tham chiếu (Ref. Pict. Buffer) 932 và được sử dụng để dự đoán các khung hình khác. Kết quả là, dữ liệu video được khôi phục lại từ REC 926 có thể dùng cho quá trình mã hóa, Bộ lọc xử lý vòng (ILPF) 928 được áp dụng cho dữ liệu video được khôi phục lại trước khi lưu trong Bộ đệm khung hình tham chiếu 932 để nâng cao hơn nữa chất lượng hình ảnh. Các thành phần cú pháp được cấp đến Bộ mã hóa Entropy 934 để tích hợp vào dòng bit video.

Bộ giải mã video tương ứng 1000 cho Bộ mã hóa video 900 trên Fig. 9 được thể hiện trên Fig. 10. Dòng bit video được mã hóa bởi bộ mã hóa video là đầu vào cho bộ giải mã video 1000 và được giải mã bởi bộ giải mã Entropy 1010 để phân tích và khôi phục tín hiệu dư thừa được biến đổi và được lượng tử hóa và thông tin hệ thống khác. Bộ giải mã Entropy 1010 phân tích một hoặc nhiều thành phần cú pháp của delta QP cho mỗi nhóm lượng tử hóa, trong đó báo hiệu delta QP phụ thuộc vào các chiều cao của các nút chia. Chiều cao của nút chia được phân vùng bởi dạng phân chia được suy dẫn theo dạng phân chia. Quá trình giải mã của bộ giải mã 1000 tương tự như vòng lặp khôi phục lại tại bộ mã hóa 900, ngoại trừ bộ giải mã 1000 chỉ yêu cầu dự đoán bù

chuyển động trong môđun dự đoán liên ảnh 1014. Mỗi khối được giải mã hoặc bởi môđun dự đoán nội ảnh 1012 hoặc Môđun dự đoán liên ảnh 1014. Bộ chuyển đổi 1016 chọn biến dự báo nội ảnh từ môđun dự đoán nội ảnh 1012 hoặc biến dự báo liên ảnh từ môđun dự đoán liên ảnh 1014 theo thông tin cơ chế giải mã. Tín hiệu dư thừa được biến đổi và được lượng tử hóa được khôi phục bởi môđun lượng tử hóa ngược (IQ) 1020 và môđun chuyển đổi ngược (IT) 1022. Môđun IQ 1020 còn được gọi là môđun phi lượng tử hóa. Bằng cách so sánh chiều cao của nút chia và giá trị ngưỡng cho báo hiệu QP, môđun lượng tử hóa ngược 1020 xác định QP luma cho nút chia từ delta QP được phân tích bởi bộ giải mã Entropy 1010 và QP tham chiếu hoặc xác định QP luma từ QP được lập mã trước đó. Theo một số phương án khác, QP chroma cho khối chroma được xác định từ QP luma của khối luma đã sắp xếp, và khối luma đã sắp xếp là khối mà bao lấy mẫu đã sắp xếp của một mẫu định trước của khối chroma hiện thời. Tín hiệu dư thừa tái sinh được khôi phục lại bằng cách thêm hậu tố cho biến dự báo ở môđun khôi phục lại 1018 để tạo ra video được khôi phục lại. Video được khôi phục lại được xử lý tiếp bởi Bộ lọc xử lý vòng (ILPF) 1024 để tạo ra video được giải mã cuối cùng. Nếu khung hình đang được giải mã là khung hình tham chiếu, video được khôi phục lại của khung hình đang được giải mã cũng được lưu trong bộ đệm khung hình tham chiếu (Ref. Pict. Buffer) 1028 cho các khung hình sau trong lệnh giải mã.

Nhiều bộ phận khác nhau của bộ mã hóa video 900 và bộ giải mã video 1000 trên Fig. 9 và Fig. 10 có thể được thực thi bằng phần cứng, một hoặc nhiều bộ xử lý được cấu hình để xử lý các lệnh chương trình được lưu trong bộ nhớ, hoặc kết hợp giữa phần cứng và bộ xử lý. Ví dụ, bộ xử lý xử lý các lệnh chương trình để điều khiển việc nhận dữ liệu video đầu vào. Bộ xử lý được trang bị có một hoặc nhiều lõi xử lý. Theo một số ví dụ, bộ xử lý xử lý các lệnh chương trình để thực hiện các phép toán trong một số bộ phận trên bộ mã hóa 900 và bộ giải mã 1000, và bộ nhớ được kết nối điện với bộ xử lý được sử dụng để lưu các lệnh chương trình, thông tin tương ứng với các hình ảnh được khôi phục lại của các khối, và/hoặc dữ liệu trung gian trong quá trình mã hóa hoặc giải mã. Theo một số ví dụ, bộ mã hóa video 900 có thể báo hiệu thông tin chứa một hoặc nhiều thành phần cú pháp trong dòng bit video, và bộ giải mã video tương ứng 1000 suy dẫn thông tin vừa nêu bằng cách phân tích và giải mã một hoặc nhiều thành phần cú pháp. Bộ nhớ theo một số phương án bao gồm phương tiện có thể đọc được bằng máy tính không tạm thời, chẳng hạn bộ nhớ bán dẫn hoặc bộ nhớ trạng

thái rắn, bộ nhớ truy cập ngẫu nhiên (RAM), bộ nhớ chỉ đọc (ROM), ổ đĩa cứng, ổ đĩa quang, hoặc phương tiện lưu trữ phù hợp khác. Bộ nhớ có thể còn kết hợp hai hoặc nhiều hơn hai phương tiện có thể đọc được bằng máy tính không tạm thời được liệt kê ở trên. Như được thể hiện trên Fig. 9 và Fig. 10, bộ mã hóa 900 và bộ giải mã 1000 có thể được thực hiện trên cùng thiết bị điện tử, do đó các thành phần chức năng khác nhau của bộ mã hóa 900 và bộ giải mã 1000 có thể được chia sẻ hoặc được tái sử dụng nếu được thực hiện trên cùng thiết bị điện tử. Ví dụ, một hoặc nhiều hơn một trong số môđun khôi phục lại 926, Môđun chuyển đổi ngược 924, môđun lượng tử hóa ngược 922, bộ lọc xử lý vòng 928, và bộ đệm khung hình tham chiếu 932 trên Fig. 9 có thể cũng được sử dụng để làm chức năng như môđun khôi phục lại 1018, môđun chuyển đổi ngược 1022, môđun lượng tử hóa ngược 1020, bộ lọc xử lý vòng 1024, và bộ đệm khung hình tham chiếu 1028 trên Fig. 10, một cách tương ứng. Các phương án của phương pháp xử lý dữ liệu có báo hiệu QP phụ thuộc vào các chiều cao liên quan đến dạng phân chia cho hệ thống lập trình video hoặc tái sử dụng các QP luma cho các thành phần chroma có thể được thực hiện trên 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 được mô tả ở trên. Đối với các ví dụ, việc xác định diện tích của nút phân chia hiện thời hoặc việc xác định QP luma của khối luma đã sắp xếp có thể được thực hiện trong mã chương trình sẽ được xử lý ở bộ xử lý máy tính, bộ xử lý tín hiệu số (DSP), 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 đoạn mã phần mềm hoặc đoạn mã vi chương trình có thể đọc được bằng máy mà định ra các phương pháp cụ thể được thực hiện bởi sáng chế. Sáng chế có thể được thực hiện theo các dạng cụ thể khác mà không khác với các nội dung hoặc các đặc tính cơ bản của sáng chế. Các ví dụ đã nêu được chỉ nhằm mục đích minh họa chứ không nhằm mục đích giới hạn sáng chế. Vì vậy, phạm vi của sáng chế được miêu tả theo các điểm yêu cầu bảo hộ đính kèm chứ không phải phần mô tả đã nêu ở trên. Mọi sửa đổi mà vẫn thuộc nội dung ý nghĩa và phạm vi tương đương của các điểm yêu cầu bảo hộ đều thuộc phạm vi của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp xử lý dữ liệu video trong hệ thống giải mã video, trong đó dữ liệu video được phân vùng thành các đơn vị mã hóa lá (các CU) theo cấu trúc phân vùng CU, bao gồm:

nhận dữ liệu vào liên quan đến nút phân chia hiện thời, trong đó nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia;

xác định chiều cao nút phân chia hiện thời theo dạng phân chia;

xác định giá trị của biến bằng cách so sánh chiều cao nút phân chia hiện thời với chiều cao báo hiệu delta tham tham số lượng tử hóa (delta QP) lớn nhất;

suy dẫn delta QP từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích ở đơn vị chuyển đổi (TU) liên quan đến nút phân chia hiện thời theo giá trị của biến;

khôi phục lại QP cuối cùng cho một hoặc nhiều TU liên quan đến nút phân chia hiện thời dựa vào QP tham chiếu và delta QP nếu một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích cho nút phân chia hiện thời; và

giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời bằng cách sử dụng QP cuối cùng.

2. Phương pháp theo điểm 1, trong đó chiều cao báo hiệu delta QP lớn nhất được định trước, được suy dẫn, được phân tích hoặc được báo hiệu ở một trong số hoặc kết hợp ở một trong số cấp slice, cấp khung ảnh, cấp tile, và cấp chuỗi ảnh.

3. Phương pháp theo điểm 1, trong đó chiều cao nút phân chia hiện thời được xác định bởi diện tích của nút phân chia hiện thời và chiều cao báo hiệu delta QP lớn nhất được suy dẫn bởi diện tích CU báo hiệu delta QP nhỏ nhất.

4. Phương pháp theo điểm 3, trong đó diện tích của nút phân chia hiện thời được xác định từ chiều rộng khối và chiều cao khối của nút phân chia hiện thời, và diện tích CU báo hiệu delta QP nhỏ nhất được định trước, được suy dẫn, được phân tích hoặc được báo hiệu ở một trong số hoặc kết hợp ở một trong số cấp slice, cấp khung ảnh, cấp tile, và cấp chuỗi ảnh.

5. Phương pháp theo điểm 3, trong đó nút phân chia hiện thời là CU lá, và delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp khi diện tích của nút phân chia hiện thời lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất và có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển trong nút phân chia hiện thời.
6. Phương pháp theo điểm 3, trong đó nút phân chia hiện thời không là CU lá và được phân vùng tiếp thành nhiều CU lá, delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp nếu nút phân chia hiện thời có kích cỡ lớn hơn hoặc bằng diện tích CU báo hiệu delta QP nhỏ nhất, diện tích của ít nhất một nút con được phân vùng từ nút phân chia hiện thời nhỏ hơn diện tích CU báo hiệu delta QP nhỏ nhất, và ít nhất một trong số các CU lá trong nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển, trong đó delta QP được sử dụng bởi một hoặc nhiều trong số các CU lá vẫn thuộc nút phân chia hiện thời.
7. Phương pháp theo điểm 1, trong đó nút phân chia hiện thời là CU lá, delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp nếu chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển.
8. Phương pháp theo điểm 1, trong đó khi nút phân chia hiện thời không là CU lá và được phân vùng tiếp thành nhiều CU lá, delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp nếu chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất, chiều cao của ít nhất một nút con được phân vùng từ nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất, và ít nhất một trong số các CU lá trong nút phân chia hiện thời có ít nhất một hệ số khác không hoặc ít nhất một bảng màu dịch chuyển, trong đó delta QP được sử dụng bởi một hoặc nhiều trong số các CU lá vẫn thuộc nút phân chia hiện thời.
9. Phương pháp theo điểm 1, trong đó chiều cao báo hiệu delta QP lớn nhất được báo hiệu hoặc được phân tích riêng cho các thành phần luma và chroma khi các cấu trúc phân vùng CU riêng được sử dụng để phân vùng dữ liệu video của các thành phần luma và chroma.
10. Phương pháp theo điểm 1, trong đó chiều cao nút phân chia hiện thời là chiều cao của nút cha cộng 2 nếu nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba của phân chia cây tam phân (TT), chiều cao nút phân chia hiện thời là chiều cao của

nút cha cộng 1 nếu nút phân chia hiện thời là nút phân chia thứ hai của phân chia TT, chiều cao nút phân chia hiện thời là chiều cao của nút cha cộng 1 nếu nút phân chia hiện thời được phân vùng từ nút cha bởi phân chia cây nhị phân (BT), và chiều cao nút phân chia hiện thời là chiều cao của nút cha cộng 2 nếu nút phân chia hiện thời được phân vùng từ nút cha bởi phân chia cây tứ phân.

11. Phương pháp theo điểm 1, trong đó nút phân chia hiện thời được phân vùng tiếp nữa thành nhiều nút con, trong đó chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 2 nếu nút con là nút phân chia thứ nhất hoặc thứ ba của phân chia cây tam phân (TT), chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 1 nếu nút con là nút phân chia thứ hai của phân chia TT, chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 1 nếu nút con được phân vùng từ nút phân chia hiện thời bởi phân chia cây nhị phân (BT), và chiều cao của nút con là chiều cao của nút phân chia hiện thời cộng 2 nếu nút con được phân vùng từ nút phân chia hiện thời bởi phân chia cây tứ phân.

12. Phương pháp theo điểm 1, trong đó khi nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba được phân vùng từ nút cha bởi phân chia cây tam phân TT (Ternary Tree - TT), chiều cao nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất, chiều cao của nút phân chia thứ hai được phân vùng từ nút cha bằng chiều cao báo hiệu delta QP lớn nhất, và delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp được sử dụng bởi một hoặc nhiều hơn một trong số ba nút phân chia được phân vùng từ nút cha.

13. Phương pháp theo điểm 1, trong đó cờ biểu thị liệu nút phân chia hiện thời là nút phân chia thứ nhất hoặc thứ ba trong phân chia cây tam phân (TT), cờ biểu thị liệu nút phân chia hiện thời là nút phân chia thứ hai trong phân chia TT, kích cỡ CU cha, hoặc kết hợp giữa chúng cũng được sử dụng để xác định giá trị của biến cho nút phân chia hiện thời.

14. Phương pháp theo điểm 1, trong đó một hoặc nhiều thông tin liên quan delta QP được truyền từ nút cha đến nút phân chia hiện thời để xác định giá trị của biến cho nút phân chia hiện thời.

15. Phương pháp theo điểm 14, trong đó một hoặc nhiều thông tin liên quan delta QP bao gồm cờ thiết lập lại cho nút phân chia hiện thời, và cờ thiết lập lại này được xác

định bằng cách xét diện tích, kích cỡ, hoặc chiều cao của nút phân chia hiện thời hoặc dạng phân chia để phân vùng nút cha.

16. Phương pháp theo điểm 1, trong đó một hoặc nhiều thông tin liên quan delta QP được truyền từ nút phân chia hiện thời đến một hoặc nhiều nút con được phân vùng từ nút phân chia hiện thời để xác định giá trị của biến cho một hoặc nhiều nút con.

17. Phương pháp theo điểm 16, trong đó một hoặc nhiều thông tin liên quan delta QP bao gồm cờ thiết lập lại cho một hoặc nhiều nút con, và cờ thiết lập lại này được xác định bằng cách xét diện tích, kích cỡ, hoặc chiều cao của một hoặc nhiều các nút con và dạng phân chia để phân vùng nút phân chia hiện thời.

18. Phương pháp theo điểm 17, trong đó cờ thiết lập lại được xác định bằng cách so sánh chiều cao của nút hiện thời cộng biến thứ hai với chiều cao báo hiệu delta QP lớn nhất, trong đó biến thứ hai bằng 2 nếu nút phân chia hiện thời được phân vùng bởi phân chia cây tam phân (TT) hoặc bởi phân chia cây tứ phân, biến thứ hai bằng 1 nếu nút phân chia hiện thời được phân vùng bởi cây nhị phân (BT).

19. Phương pháp theo điểm 1, trong đó nhiều nhất một delta QP được suy dẫn cho toàn bộ các nút con được phân vùng từ nút phân chia hiện thời bởi phân chia cây tam phân TT (Ternary Tree - TT) nếu chiều cao nút phân chia hiện thời nhỏ hơn hoặc bằng chiều cao báo hiệu delta QP lớn nhất và chiều cao của ít nhất một nút con được phân vùng từ nút phân chia hiện thời lớn hơn chiều cao báo hiệu delta QP lớn nhất, và delta QP được suy dẫn từ một hoặc nhiều thành phần cú pháp được sử dụng bởi một hoặc nhiều hơn một trong số các nút con được phân vùng từ nút phân chia hiện thời.

20. Phương pháp theo điểm 1, trong đó QP tham chiếu là QP slice khi cấu trúc phân vùng CU là cấu trúc phân vùng cây nhị phân-cây tứ phân QTBT (Quad-Tree-Binary-Tree - QTBT) hoặc cấu trúc phân vùng cây đa phân (MTT).

21. Thiết bị xử lý dữ liệu video trong hệ thống giải mã video, trong đó dữ liệu video được phân vùng thành các đơn vị mã hóa lá (các CU) theo một hoặc nhiều cấu trúc phân vùng CU, thiết bị này bao gồm một hoặc nhiều mạch điện tử được cấu hình để:

nhận dữ liệu vào liên quan đến nút phân chia hiện thời, trong đó nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia;

xác định chiều cao nút phân chia hiện thời theo dạng phân chia;

xác định giá trị của biến bằng cách so sánh chiều cao nút phân chia hiện thời với chiều cao báo hiệu delta tham tham số lượng tử hóa (delta QP) lớn nhất;

suy dẫn delta QP từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích ở đơn vị chuyển đổi (TU) liên quan đến nút phân chia hiện thời theo giá trị của biến;

khôi phục lại QP cuối cùng cho một hoặc nhiều TU liên quan đến nút phân chia hiện thời dựa vào QP tham chiếu và delta QP nếu một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích cho nút phân chia hiện thời; và

giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời bằng cách sử dụng QP cuối cùng.

22. Phương tiện có thể đọc được bằng máy tính không tạm thời để lưu lệnh chương trình giúp cho mạch xử lý của thiết bị thực hiện phương pháp xử lý video, và phương pháp này bao gồm:

nhận dữ liệu vào liên quan đến nút phân chia hiện thời, trong đó nút phân chia hiện thời được phân vùng từ nút cha bởi dạng phân chia;

xác định chiều cao nút phân chia hiện thời theo dạng phân chia;

xác định giá trị của biến bằng cách so sánh chiều cao nút phân chia hiện thời với chiều cao báo hiệu delta tham tham số lượng tử hóa (delta QP) lớn nhất;

suy dẫn delta QP từ một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích ở đơn vị chuyển đổi (TU) liên quan đến nút phân chia hiện thời theo giá trị của biến;

khôi phục lại QP cuối cùng cho một hoặc nhiều TU liên quan đến nút phân chia hiện thời dựa vào QP tham chiếu và delta QP nếu một hoặc nhiều thành phần cú pháp được báo hiệu hoặc được phân tích cho nút phân chia hiện thời; và

giải mã một hoặc nhiều TU liên quan đến nút phân chia hiện thời bằng cách sử dụng QP cuối cùng.

1/8

Fig. 1.

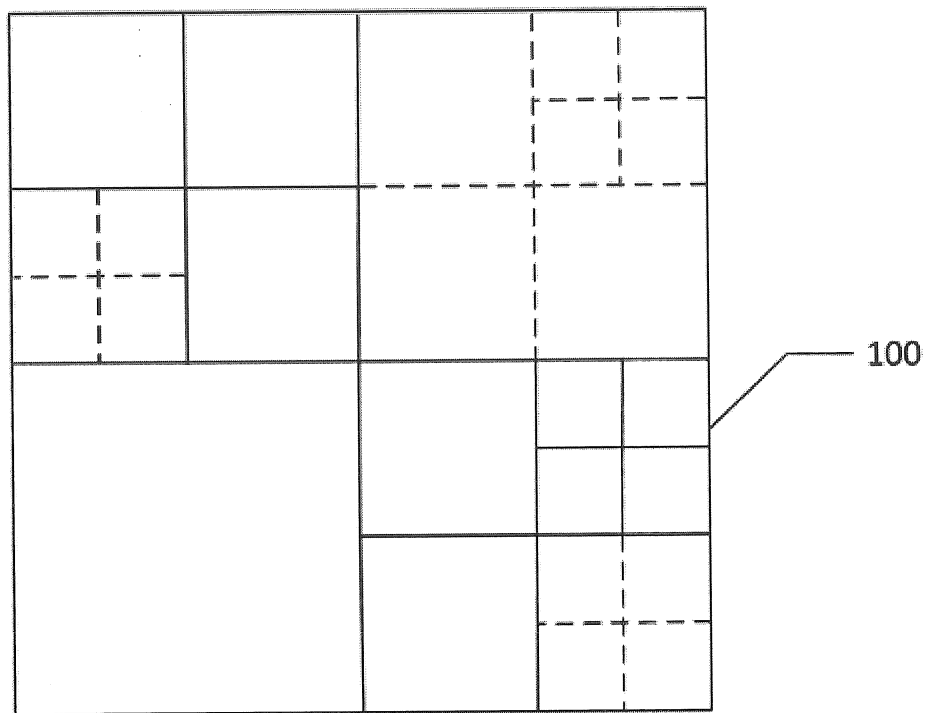
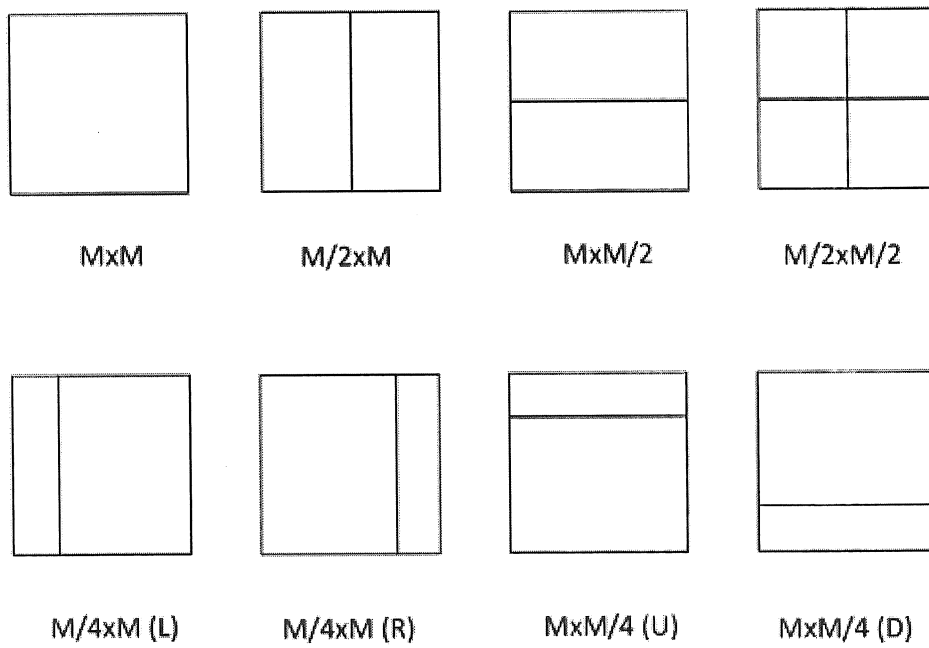
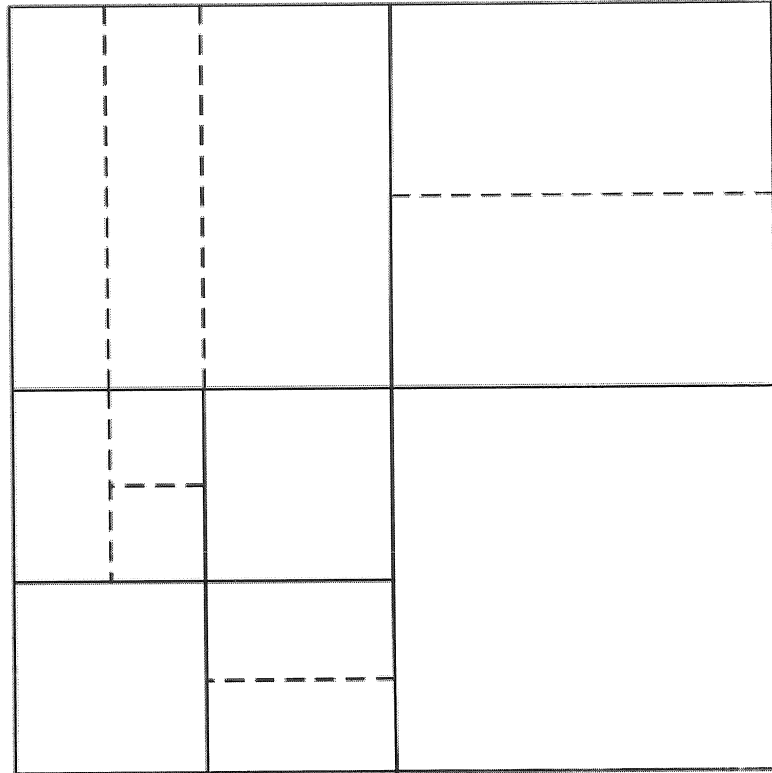
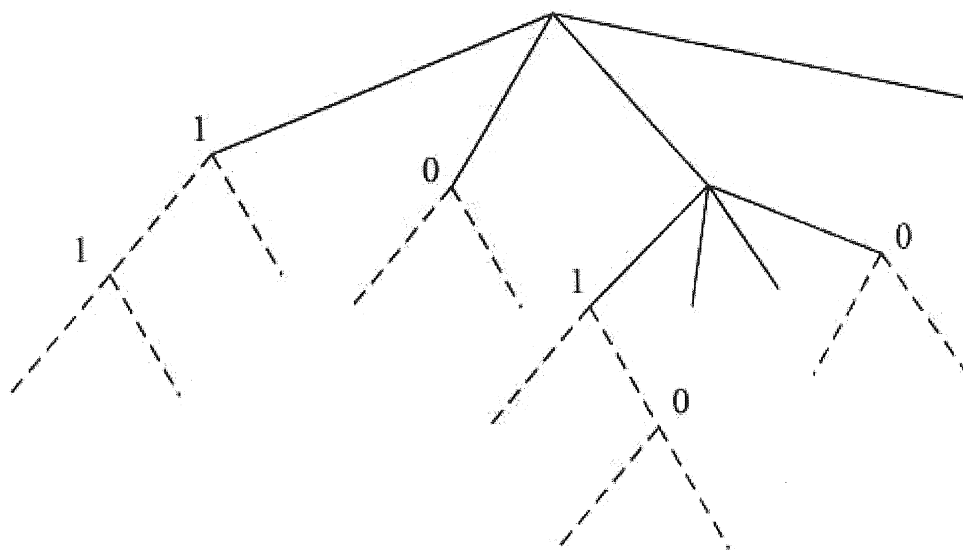


Fig. 2



2/8

Fig. 3A**Fig. 3B**

3/8

Fig. 4

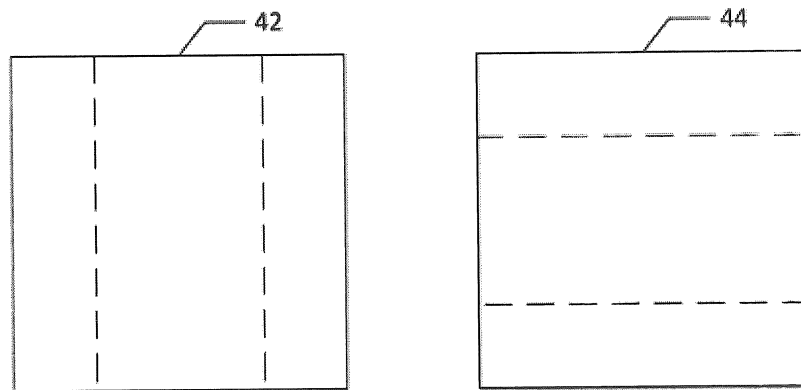
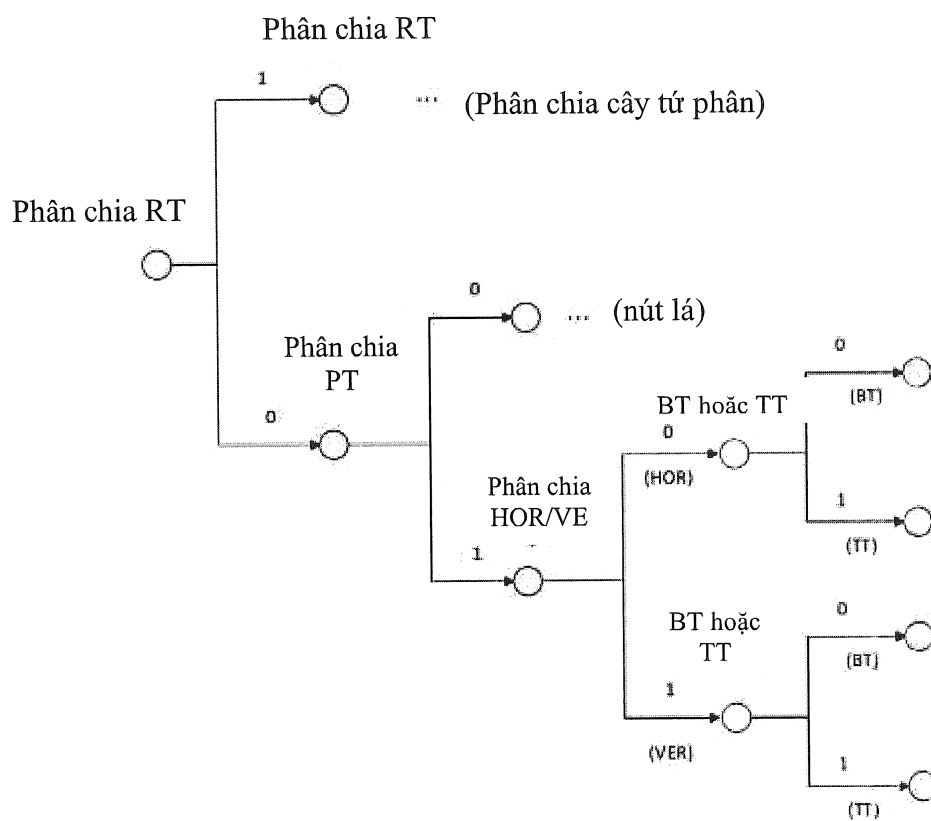
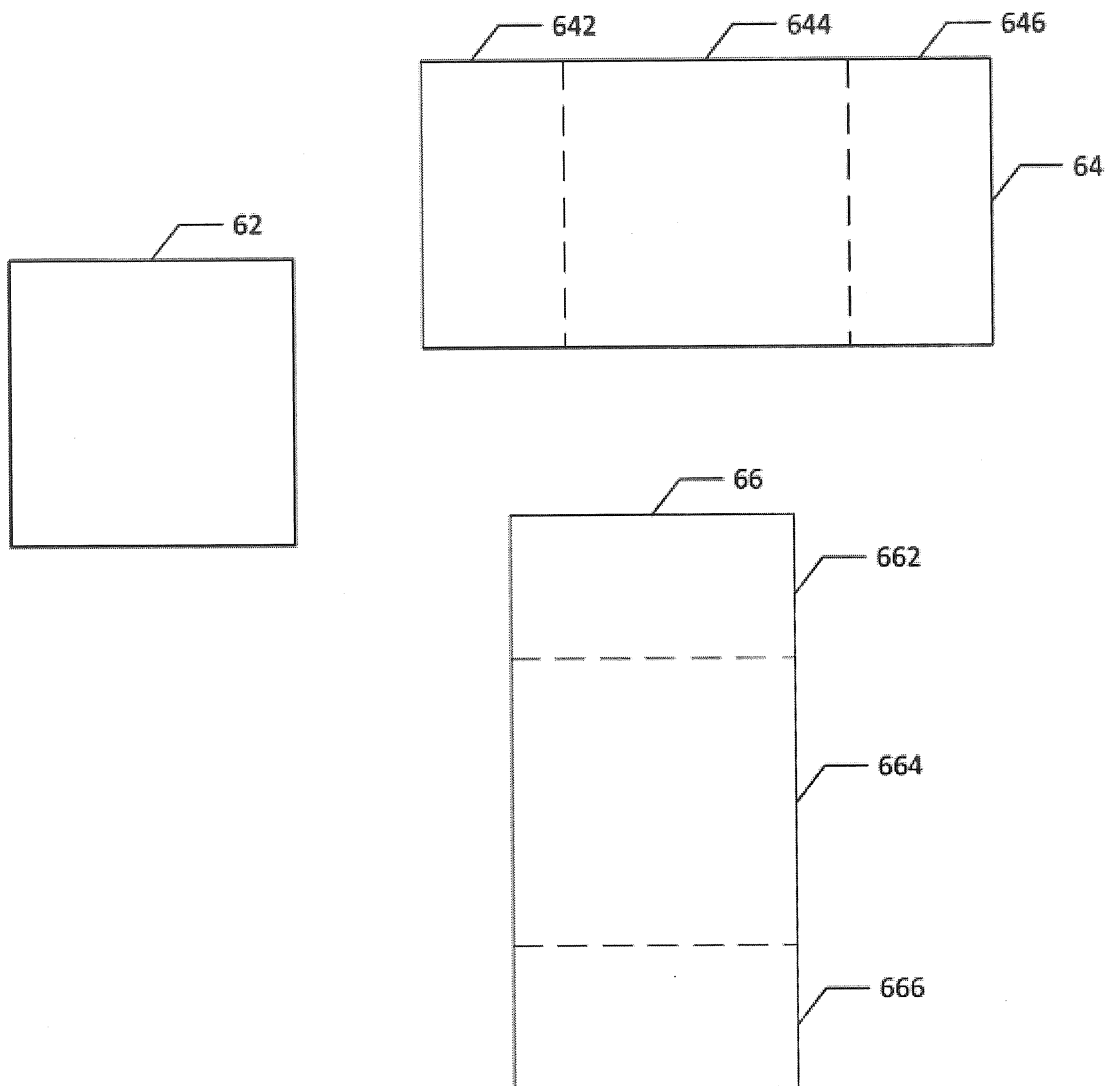


Fig. 5



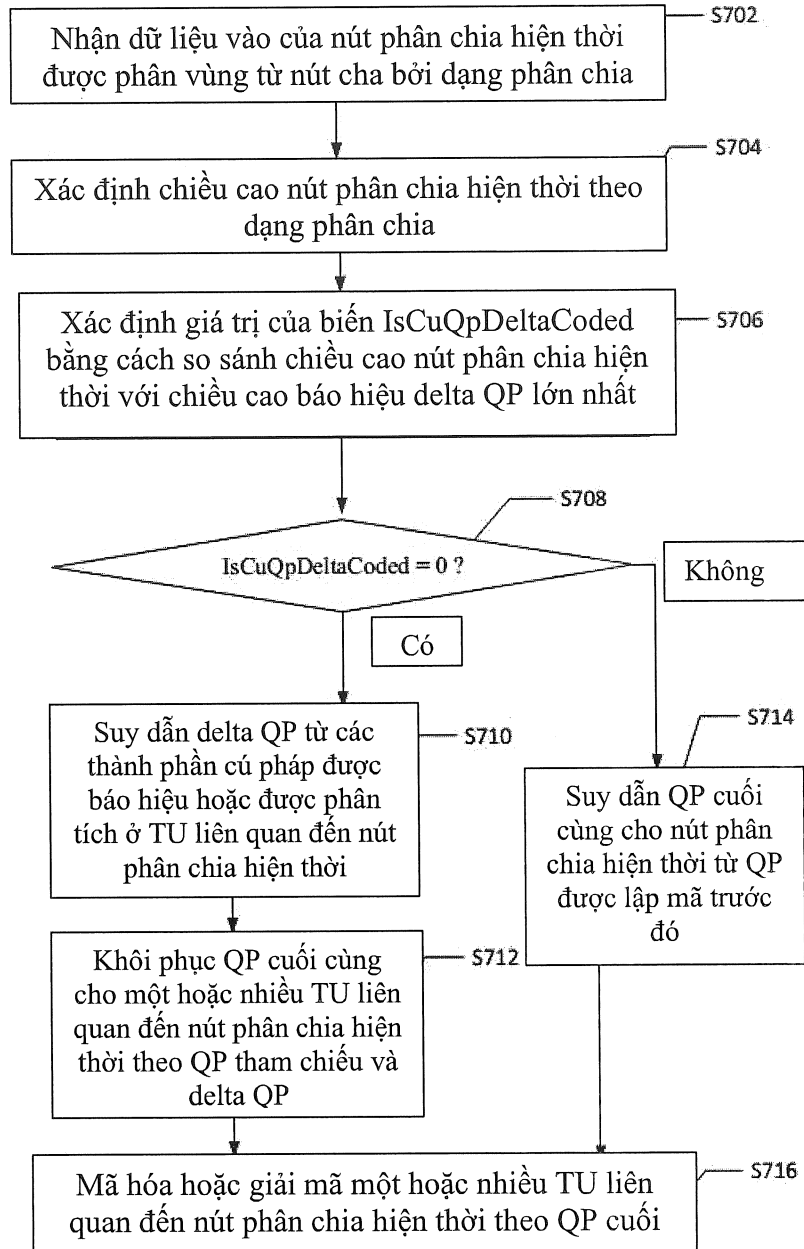
4/8

Fig. 6



5/8

Fig. 7



6/8

Fig. 8

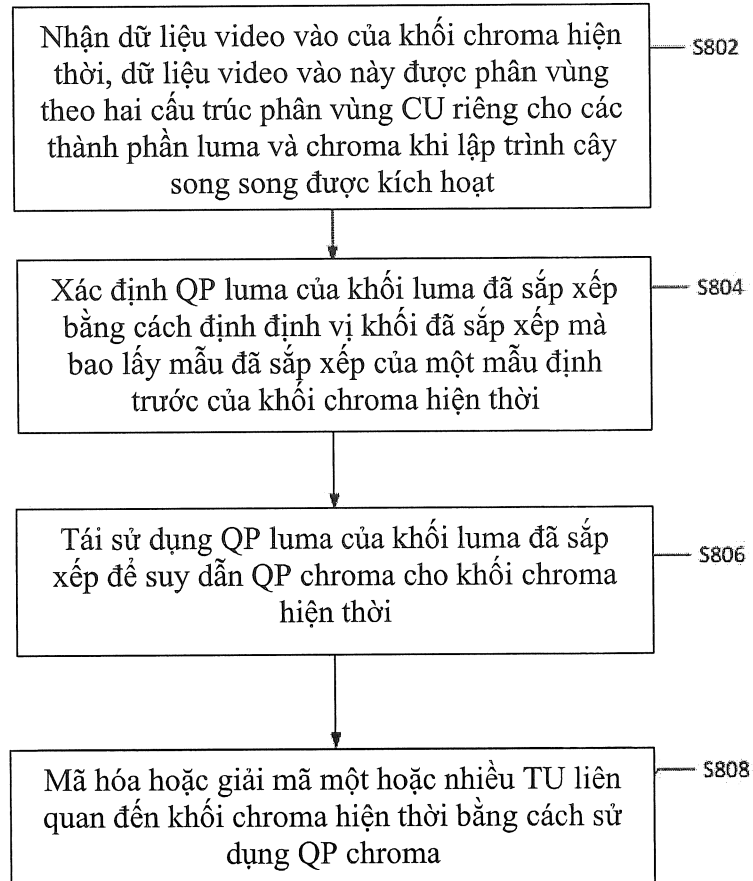
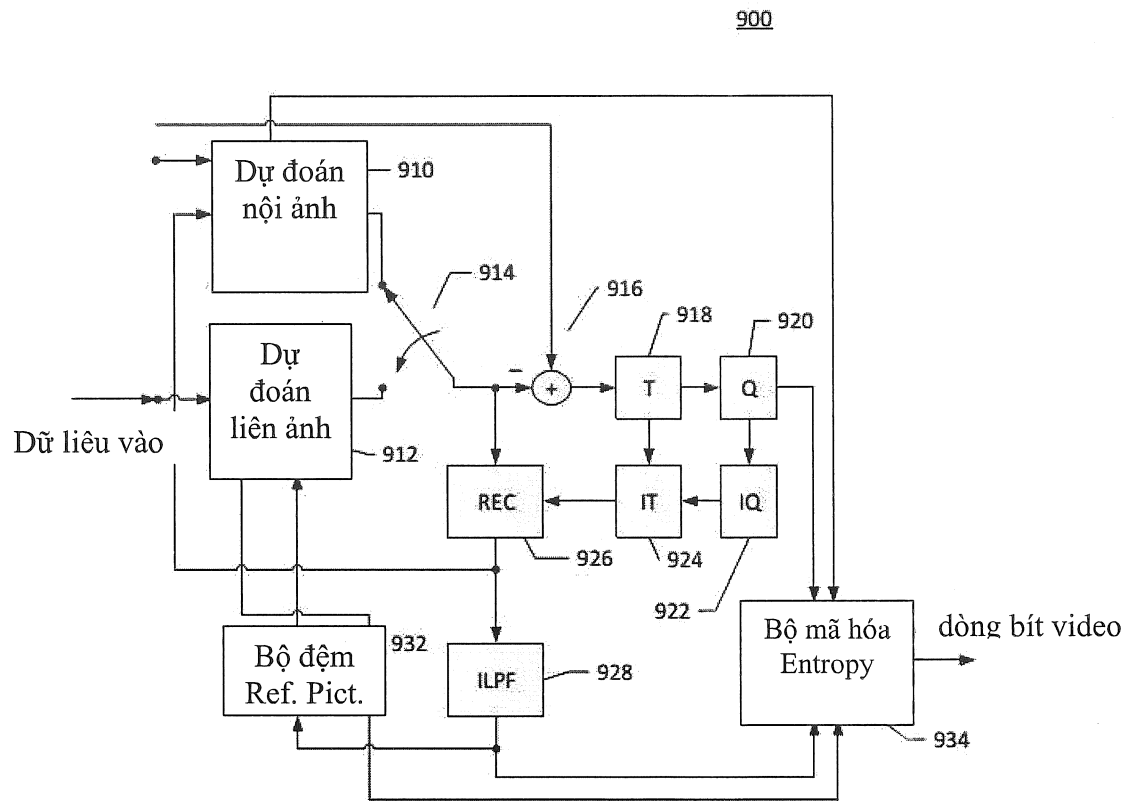


Fig. 9



8/8

Fig. 10

