



(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)



1-0036738

(51)¹⁹ H04N 19/119; H04N 19/70; H04N
19/176

(13) B

(21) 1-2019-07195

(22) 24/05/2018

(86) PCT/KR2018/005879 24/05/2018

(87) WO2018/217024 29/11/2018

(30) 10-2017-0065495 26/05/2017 KR; 10-2017-0097259 31/07/2017 KR

(45) 25/08/2023 425

(43) 25/02/2020 383A

(73) SK TELECOM CO., LTD. (KR)

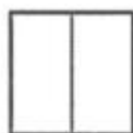
SK T-Tower, 65, Eulji-ro Jung-gu Seoul 04539, Republic of Korea

(72) LIM, Jeong-yeon (KR); LEE, Sun-young (KR); KIM, Hyeong-duck (KR); KIM, Hyo Song (KR); SON, Se-hoon (KR); SHIN, Jae-seob (KR); LEE, Gyeong-taek (KR).

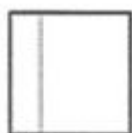
(74) Công ty Luật TNHH Phạm và Liên danh (PHAM & ASSOCIATES)

(54) PHƯƠNG PHÁP MÃ HÓA/GIẢI MÃ DỮ LIỆU VIDEO VÀ VẬT GHI MÁY TÍNH ĐỌC ĐƯỢC

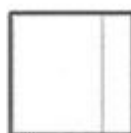
(57) Sáng chế đề xuất mã hóa hoặc giải mã video để mã hóa hiệu quả video. Các giải pháp của sáng chế đề cập đến các hình dạng được tách khác nhau của khối, các cú pháp đại diện các loại khối được tách khác nhau, và các phần tử cú pháp được biểu diễn ở mức cao cho nó.



(1) 1/2



(2) 1/4

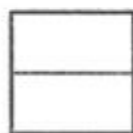


(3) 3/4

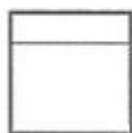


(4) tri (1/4 & 3/4)

(a) Tách khối theo phương thẳng đứng



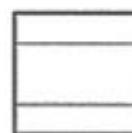
(1) 1/2



(2) 1/4



(3) 3/4



(4) tri (1/4 & 3/4)

(b) Tách khối theo phương nằm ngang

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến công nghệ mã hóa/giải mã video để mã hóa hiệu quả video. Cụ thể hơn, sáng chế đề cập đến các kỹ thuật báo hiệu thông tin phân vùng liên quan đến phân vùng khối dựa trên cây tứ phân cộng cây nhị phân (Quadtree plus Binary tree, QTBT).

Tình trạng kỹ thuật của sáng chế

Các phần mô tả này chủ yếu đề cập đến thông tin cơ bản về sáng chế và không phải là giải pháp kỹ thuật đã biết.

Trong kỹ thuật mã hóa video hiệu suất cao (High Efficiency Video Coding, HEVC), khối cây mã hóa (coding tree unit, CTU) là cây mã hóa để phản ánh các đặc tính cục bộ khác nhau trong video và được phân chia thành các khối mã hóa (coding unit, CU) bằng cách sử dụng phương tiện cây tứ phân (QuadTree, QT). Khi CU được xác định theo cách này, việc báo hiệu trong mã hóa và báo hiệu ngoài mã hóa được thực hiện dựa trên CU. Mỗi CU lại được phân chia thành các khối dự báo (prediction unit, PU). Sau khi các PU được xác định và quá trình dự báo được thực hiện, CU được phân chia thành các khối biến đổi (transformation unit, TU) cho khối còn lại.

Fig.1 là sơ đồ minh họa ví dụ trong đó CTU được tách bởi phương tiện QT và cấu trúc cây thể hiện việc tách. Một CTU khối lớn được tách thành bốn hình vuông bằng nhau, trong đó khối bên trái ở trên được phân chia và các khối bên phải ở trên không được tách thêm, trong khi

mỗi khối trong các khối bên trái và bên phải ở dưới lại được tách thành bốn hình vuông bằng nhau. Trên Fig.1(b), hình tròn, thể hiện khối, được gọi là “nút”, và nút cuối cùng, mà được gọi là “nút lá” không được tách thêm được tô màu xám. Tức là, tất cả các nút lá đề cập đến CU, vốn là khối mã cuối cùng.

Như là cú pháp mức cao để biểu diễn CTU và CU nêu trên, hai phần tử cú pháp được xác định trong tập tham số chuỗi (sequence parameter set, SPS) như được thể hiện trong Bảng 1.

BẢNG 1

<code>seq_parameter_set_rbsp() {</code>	Mô tả
...	
<code>log2_min_luma_coding_block_size_minus3</code>	ue(v)
<code>log2_diff_max_min_luma_coding_block_size</code>	ue(v)
...	

Ở đây, `log2_min_luma_coding_block_size_minus3` xác định kích thước khối CU nhỏ nhất (`min_luma_coding_block_size`) được phép dựa trên khối luma, và `log2_diff_max_min_luma_coding_block_size` xác định hiệu số giữa `min_luma_coding_block_size` và kích thước khối CU lớn nhất (`max_luma_coding_block_size`) được phép dựa trên khối luma. Bằng cách sử dụng hai phần tử cú pháp, kích thước khối luma nhỏ nhất thực sự (`MinCbSizeY`) được biểu diễn dưới dạng Phương trình 3, và kích thước CTU (`CtbSizeY`) được biểu diễn dưới dạng Phương trình 4.

$$\text{MinCbLog2SizeY} = \text{log2_min_luma_coding_block_size_minus3} + 3 \quad (1)$$

$$\text{CtbLog2SizeY} = \text{MinCbLog2SizeY} + \text{log2_diff_max_min_luma_coding_block_size} \quad (2)$$

$$\text{MinCbSizeY} = 1 \ll \text{MinCbLog2SizeY} \quad (3)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (4)$$

Thông tin tách về các khối CU trong CTU được ghi nhận trong phần

tiêu đề QT mã hóa của tiêu đề lát, như được thể hiện trong Bảng 2. Cờ 1 bit `split_cu_flag` được sử dụng để chỉ báo liệu việc tách QT đã được áp dụng cho nút cụ thể chưa. Nếu cờ bằng “0,” chỉ báo rằng việc tách không được thực hiện. Nếu cờ bằng “1,” chỉ báo rằng khối được tách thành bốn khối bằng nhau. Khi cờ được sử dụng lặp lại, thông tin vị trí và kích thước về các khối CU được tách trong CTU được trình bày.

Bảng 2

<code>coding_quadtree(x0, y0, log2CbSize, cqtDepth) {</code>	Mô tả
<code>...</code>	
<code>split_cu_flag[x0][y0]</code>	<code>ae(v)</code>
<code>...</code>	

Gần đây, cấu trúc QTBT mới được đề cập đến. Cấu trúc QTBT loại bỏ các khái niệm đã biết về CU, PU, và TU trong khi tạo tính linh hoạt trong phân vùng CU để so khớp các tính năng cục bộ khác nhau của dữ liệu video. Tức là, trong QTBT, CU có thể được định nghĩa có hình dạng chữ nhật hoặc hình vuông. Ngoài ra, trong khi kích thước CTU bằng 64 trong HEVC, sự cần thiết của các kích thước khối lớn hơn, chẳng hạn 128 và 256, đã được đề cập khi độ phân giải video tăng lên. Do vậy, khi khối được tách thành ít nhất các khối 4x4 trong CTU, thì lượng dữ liệu được yêu cầu để báo hiệu thông tin tách tăng lên.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là báo hiệu một cách hiệu quả thông tin tách liên quan đến phân vùng khối dựa trên QTBT.

Theo một khía cạnh của sáng chế, phương pháp giải mã dữ liệu video được đề xuất, bao gồm bước nhận dòng bit chứa dữ liệu được mã hóa cho khối của dữ liệu video; phân tách các phần tử cú pháp để phân vùng khối được định nghĩa ở mức cao bằng cách giải mã thông

tin tiêu đề của mức cao, khối của dữ liệu video thuộc mức cao; tính toán các tham số để phân vùng khối từ các phần tử cú pháp; dẫn xuất cấu trúc phân vùng khối QTBT cho khối của dữ liệu video từ dòng bit dựa trên các tham số được tính toán, trong đó cấu trúc phân vùng khối QTBT là cấu trúc mà BT có gốc từ nút lá của QT; và giải mã các khối phụ cuối cùng tương ứng với các nút lá của QTBT dựa trên dòng bit video.

Theo khía cạnh khác của sáng chế, phương pháp giải mã dữ liệu video được đề xuất, bao gồm các bước nhận dòng bit chứa dữ liệu được mã hóa cho khối của dữ liệu video; phân tách các phần tử cú pháp để phân vùng khối được định nghĩa ở mức cao bằng cách giải mã thông tin tiêu đề của mức cao, khối của dữ liệu video thuộc mức cao; tính toán các tham số để phân vùng khối từ các phần tử cú pháp; dẫn xuất cấu trúc phân vùng khối QTBT cho khối của dữ liệu video từ dòng bit dựa trên các tham số được tính toán, trong đó cấu trúc phân vùng khối QTBT là cấu trúc mà BT có gốc từ nút lá của QT; và giải mã các khối phụ cuối cùng tương ứng với các nút lá của QTBT dựa trên dòng bit video.

Theo một số phương án thực hiện, các tham số để phân vùng khối bao gồm tham số thứ nhất chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong BT, tham số thứ hai chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong QT, tham số thứ ba chỉ báo kích thước khối lớn nhất của nút gốc được phép trong BT, và tham số thứ tư chỉ báo kích thước của khối của dữ liệu video.

Theo một số phương án thực hiện khác, các tham số để phân vùng khối bao gồm tham số thứ nhất chỉ báo kích thước khối nhỏ nhất của

nút lá BT được tạo qua tách đối xứng, tham số thứ hai chỉ báo kích thước khối nhỏ nhất của nút lá BT được tạo qua tách đối xứng, tham số thứ ba chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong QT, tham số thứ tư chỉ báo kích thước khối lớn nhất của nút gốc được phép trong BT, và tham số thứ năm chỉ báo kích thước của khối của dữ liệu video.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ minh họa ví dụ trong đó CTU được tách bởi phương tiện QT và cấu trúc cây đại diện việc tách;

Fig.2 là sơ đồ khối lấy làm ví dụ của thiết bị mã hóa video có thể triển khai các phương án thực hiện sáng chế;

Fig.3 là sơ đồ minh họa các loại tách BT theo sáng chế;

Fig.4 là sơ đồ khối lấy làm ví dụ của thiết bị giải mã video có thể triển khai các phương án thực hiện sáng chế;

Fig.5 là sơ đồ minh họa ví dụ tách bởi phương tiện QT và cấu trúc cây đại diện việc tách;

Fig.6 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo loại tách trong ngôn ngữ học của cú pháp tách BT theo phương án thực hiện sáng chế;

Fig.7 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo loại tách trong ngôn ngữ học của cú pháp tách BT theo phương án thực hiện khác của sáng chế;

Fig.8 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo loại tách trong ngôn ngữ học của cú pháp tách BT theo phương án thực hiện khác nữa của sáng chế;

Fig.9 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo loại tách trong ngôn ngữ học của cú pháp tách BT theo phương án thực hiện khác nữa của sáng chế;

Fig.10 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo loại tách trong ngôn ngữ học của cú pháp tách BT theo phương án thực hiện khác nữa của sáng chế;

Fig.11 là lưu đồ minh họa phương pháp xác định tách thứ hai theo ràng buộc bởi thiết bị giải mã video theo phương án thực hiện sáng chế;

Fig.12 là lưu đồ minh họa phương pháp xác định tách thứ hai theo ràng buộc bởi thiết bị giải mã video theo phương án thực hiện sáng chế;

Fig.13 là sơ đồ minh họa phân vùng BT của khối cụ thể thành ba vùng và thành hai vùng; và

Fig.14 là lưu đồ minh họa quá trình loại trừ BT trong đó thiết bị giải mã video không báo hiệu số nhị phân thứ hai của BT_split_flag.

Mô tả chi tiết sáng chế

Ở đây, một số phương án thực hiện sáng chế sẽ được mô tả chi tiết dựa vào các hình vẽ đi kèm. Nên lưu ý rằng, khi thêm các số chỉ dẫn vào các phần tử cấu thành trong các hình vẽ, các số chỉ dẫn giống nhau ký hiệu các phần tử giống nhau, mặc dù các phần tử được thể hiện trong các hình vẽ khác nhau. Ngoài ra, trong phần mô tả sau của sáng chế, mô tả chi tiết các chức năng và các cấu hình đã biết được đưa vào ở đây sẽ bị bỏ qua mặc dù nó có thể khiến đối tượng của sáng chế trở nên không rõ ràng.

Các kỹ thuật được mô tả dưới đây có thể được sử dụng trong thiết bị xử lý tín hiệu video được tạo cấu hình để mã hóa và/hoặc giải mã tín

hiệu video. Nói chung, tín hiệu video đề cập đến tín hiệu ảnh hoặc chuỗi ảnh mà mắt có thể nhận dạng. Tuy nhiên, trong bản mô tả này, cụm từ “tín hiệu video” có thể được sử dụng để cập đến chuỗi các bit đại diện ảnh được mã hóa hoặc dòng bit tương ứng với chuỗi bit. Ảnh có thể đề cập đến việc bố trí các mẫu, và có thể được gọi là khung, ảnh, hoặc tương tự. Cụ thể hơn, ảnh có thể đề cập đến mảng hai chiều của các mẫu hoặc mảng mẫu hai chiều. Mẫu có thể đề cập đến khối nhỏ nhất tạo thành ảnh, và có thể được gọi là điểm ảnh (pixel), phần tử ảnh, pel, hoặc tương tự. Mẫu có thể bao gồm thành phần độ chói (luma) và/hoặc thành phần màu (chroma). Trong bản mô tả này, mã hóa có thể được sử dụng để đề cập đến việc mã hóa, hoặc có thể đề cập chung đến mã hóa/giải mã.

Ảnh có thể bao gồm ít nhất một lát, và lát có thể bao gồm ít nhất một khối. Lát có thể được tạo cấu hình để bao gồm số lượng nguyên của các khối để xử lý song song hoặc tương tự, hoặc để đồng bộ lại việc giải mã khi dòng bit bị phá bỏ do tổn hao dữ liệu hoặc tương tự, và mỗi lát có thể được mã hóa độc lập. Khối bao gồm ít nhất một mẫu và có thể đề cập đến việc bố trí các mẫu. Khối có thể có kích thước nhỏ hơn hoặc bằng kích thước của ảnh. Khối có thể được gọi là đơn vị. Ảnh đang được mã hóa có thể được gọi là ảnh hiện tại, và khối đang được mã hóa có thể được gọi là khối hiện tại.

CTU hoặc khối cây mã hóa (coding tree block, CTB) đề cập đến đơn vị cơ bản nhất tạo thành ảnh, và có thể được tách thành các khối mã hóa ở dạng QT để tăng hiệu suất mã hóa theo kết cấu của ảnh. CB hoặc CU có thể đề cập đến đơn vị cơ bản trong đó thực hiện mã hóa, và mã hóa trong hoặc mã hóa ngoài có thể được thực hiện dựa trên CB.

Mã hóa trong có thể đề cập đến thực hiện mã hóa bằng cách sử dụng dự báo trong, và dự báo trong có thể đề cập đến thực hiện dự báo bằng cách sử dụng các mẫu được bao gồm trong cùng ảnh hoặc lát. Mã hóa ngoài có thể đề cập đến thực hiện mã hóa sử dụng dự báo ngoài, và dự báo ngoài có thể đề cập đến thực hiện dự báo sử dụng các mẫu được bao gồm trong ảnh khác ảnh hiện tại. Khối được mã hóa bằng cách sử dụng mã hóa trong hoặc khối được mã hóa trong chế độ dự báo trong có thể được gọi là khối trong, và khối được mã hóa bằng cách sử dụng mã hóa ngoài hoặc khối được mã hóa trong chế độ dự báo ngoài có thể được gọi là khối ngoài. Ngoài ra, chế độ mã hóa sử dụng dự báo trong có thể được gọi là chế độ trong, và chế độ mã hóa sử dụng dự báo ngoài có thể được gọi là chế độ ngoài.

Các giải kỹ thuật của sáng chế liên quan đến các dạng được tách khác nhau của khối, các cú pháp đại diện các loại khối được tách khác nhau, và các phần tử cú pháp được biểu diễn ở mức cao cho nhỏ.

Fig.2 là sơ đồ khối lấy làm ví dụ của thiết bị mã hóa video có thể triển khai các phương án thực hiện sáng chế.

Thiết bị mã hóa video bao gồm bộ tách khối 210, bộ dự báo 220, bộ trừ 230, bộ biến đổi 240, bộ lượng tử hóa 245, bộ mã hóa 250, bộ lượng tử hóa đảo 260, bộ biến đổi ngược 265, bộ cộng 270, khối bộ lọc 280, và bộ nhớ 290. Mỗi phần tử của thiết bị mã hóa video có thể được triển khai dưới dạng vi mạch phân cứng, hoặc có thể được triển khai dưới dạng phần mềm, và bộ vi xử lý có thể được triển khai để thực thi các chức năng của phần mềm tương ứng với các phần tử riêng rẽ.

Bộ tách khối 210 tách mỗi ảnh tạo thành video thành các CTU, và sau đó tách đệ quy các CTU bằng cách sử dụng cấu trúc cây. Nút lá

trong cấu trúc cây là CU, vốn là đơn vị mã hóa cơ bản. Cấu trúc QT, trong đó nút (hoặc nút cha mẹ) được tách thành các nút phụ (hoặc các nút con) có cùng kích thước, hoặc cấu trúc QTBT kết hợp cấu trúc QT và cấu trúc BT, trong đó nút được tách thành hai nút phụ, có thể được sử dụng làm cấu trúc cây.

Trong cấu trúc QTBT, CTU trước hết có thể được tách theo cấu trúc QT. Việc tách QT có thể được lặp lại cho đến khi kích thước của khối tách đạt kích thước khối nhỏ nhất MinQTSIZE của nút lá được phép trong QT. Nếu nút lá của QT không lớn hơn kích thước khối lớn nhất MaxBTSIZE của nút gốc được phép trong BT, có thể còn được phân vùng thành cấu trúc BT. BT có thể có các loại tách. Fig.3 là sơ đồ minh họa các loại tách BT theo sáng chế. Chẳng hạn, trong một số ví dụ, có thể có hai loại tách, vốn là loại tách ngang khối cụ thể thành hai khối có cùng kích thước (tức là, tách đối xứng ngang) và loại và loại tách dọc khối cụ thể thành hai khối có cùng kích thước (tức là, tách dọc đối xứng) (được chỉ báo bởi 1/2 trên Fig.3). Ngoài ra, có thể có các loại tách thực hiện tách khối cụ thể thành hai khối bất đối xứng. Tách bất đối xứng có thể bao gồm tách khối cụ thể thành hai khối chữ nhật ở tỷ lệ kích thước 1:3 (được chỉ báo bởi 1/4 trên Fig.3), tách khối cụ thể thành hai khối chữ nhật ở tỷ lệ kích thước 3:1 (được chỉ báo bởi 3/4 trên Fig.3), hoặc tách khối cụ thể thành ba khối chữ nhật ở tỷ lệ kích thước 1:2:1 (được chỉ báo bởi tri(1/4 & 3/4) trên Fig.3). Có thể còn bao gồm tách bất đối xứng nhỏ hơn ở tỷ lệ (chẳng hạn, 1:7, 7:1, v.v.) khác ngoài 1:3 và 3:1.

Thông tin tách được tạo bởi bộ tách khối 210 bằng cách tách CTU bởi cấu trúc QTBT được mã hóa bởi bộ mã hóa 250 và được truyền đến thiết bị giải mã video.

Sau đây, khối tương ứng với CU (tức là, nút lá của QTBT) được mã hóa hoặc giải mã được gọi là “khối hiện tại”.

Bộ dự báo 220 tạo khối dự báo bằng cách dự báo khối hiện tại. Bộ dự báo 220 bao gồm bộ dự báo trong 222 và bộ dự báo ngoài 224.

Bộ dự báo trong 222 dự báo các điểm ảnh trong khối hiện tại bằng cách sử dụng các điểm ảnh (các điểm ảnh tham chiếu) được đặt quanh khối hiện tại trong ảnh hiện tại trong đó khối hiện tại được bao gồm. Có các chế độ dự báo trong theo các hướng dự báo, và các điểm ảnh ngoại vi và phương trình được sử dụng được định nghĩa khác nhau theo mỗi chế độ dự báo. Cụ thể là, bộ dự báo trong 222 có thể xác định chế độ dự báo trong được sử dụng trong mã hóa khối hiện tại. Trong một số ví dụ, bộ dự báo trong 222 có thể mã hóa khối hiện tại bằng cách sử dụng vài chế độ dự báo trong và lựa chọn chế độ dự báo trong thích hợp để sử dụng từ các chế độ được kiểm thử. Chẳng hạn, bộ dự báo trong 222 có thể tính toán các giá trị biến dạng tốc độ bằng cách sử dụng phân tích biến dạng tốc độ của vài chế độ dự báo trong được kiểm thử, và có thể lựa chọn chế độ dự báo trong có các đặc tính biến dạng tốc độ trong số các chế độ kiểm thử.

Bộ dự báo ngoài 224 tìm kiếm khối gần tương tự với khối hiện tại trong ảnh tham chiếu được mã hóa và giải mã sớm hơn ảnh hiện tại, và tạo khối dự báo cho khối hiện tại bằng cách sử dụng khối được tìm kiếm. Sau đó, bộ dự báo ngoài tạo vector chuyển động tương ứng với dịch chuyển giữa khối hiện tại trong ảnh hiện tại và khối dự báo trong ảnh tham chiếu. Thông tin chuyển động bao gồm thông tin về ảnh tham chiếu và vector chuyển động được sử dụng để dự báo khối hiện tại được mã hóa bởi bộ mã hóa 250 và được truyền đến thiết bị giải mã video.

Bộ trừ 230 lấy khối hiện tại trừ đi khối dự báo được tạo bởi bộ dự báo trong 222 hoặc bộ dự báo ngoài 224 để tạo khối dư.

Bộ biến đổi 240 biến đổi các tín hiệu dư trong khối dư có các giá trị điểm ảnh trong miền không gian thành các hệ số biến đổi trong miền tần số. Bộ biến đổi 240 có thể biến đổi các tín hiệu dư trong khối dư bằng cách sử dụng kích thước của khối hiện tại làm khối biến đổi, hoặc có thể tách khối dư thành các khối phụ nhỏ hơn và biến đổi các tín hiệu dư trong các khối biến đổi tương ứng với các kích thước của các khối phụ. Có thể có các phương pháp khác nhau tách khối dư thành các khối phụ nhỏ hơn. Chẳng hạn, khối dư có thể được tách thành các khối phụ có cùng kích thước định trước, hoặc có thể được tách theo cách thức của QT lấy khối dư làm nút gốc.

Bộ lượng tử hóa 245 lượng tử hóa các hệ số biến đổi được xuất ra từ bộ biến đổi 240 và xuất ra các hệ số biến đổi được lượng tử hóa đến bộ mã hóa 250.

Bộ mã hóa 250 mã hóa các hệ số biến đổi được lượng tử hóa bằng cách sử dụng phương tiện mã hóa chẳng hạn CABAC để tạo dòng bit. Bộ mã hóa 250 mã hóa thông tin chẳng hạn các tham số liên quan đến tách khối (MinBtSizeY, SymMinBtSizeY, AsymMinBtSizeY, MinQtSizeY, MaxBtSizeY, CtbSizeY), cờ tách QT, và cờ tách BT, được liên kết với tách khối, sao cho thiết bị giải mã video tách khối theo cách thức tương tự như trong thiết bị mã hóa video.

Bộ mã hóa 250 mã hóa thông tin về loại dự báo chỉ báo liệu khối hiện tại được mã hóa bởi dự báo trong hoặc dự báo ngoài, và mã hóa thông tin dự báo trong hoặc thông tin dự báo ngoài theo loại dự báo.

Bộ lượng tử hóa đảo 260 lượng tử hóa đảo các hệ số biến đổi được

lượng tử hóa được xuất ra từ bộ lượng tử hóa 245 để tạo các hệ số biến đổi. Bộ biến đổi ngược 265 biến đổi các hệ số biến đổi được xuất ra từ bộ lượng tử hóa đảo 260 từ miền tần số sang miền không gian và tái tạo khối dư.

Bộ cộng 270 thêm khối dư được tái tạo vào khối dự báo được tạo bởi bộ dự báo 220 để tái tạo khối hiện tại. Các điểm ảnh trong khối hiện tại được tái tạo được sử dụng làm các mẫu tham chiếu khi thực hiện dự báo trong của khối tiếp theo theo thứ tự.

Khối bộ lọc 280 bỏ lọc khối các đường biên giữa các khối được tái tạo để loại bỏ các đối tượng chặn gây ra bằng cách mã hóa/giải mã theo từng khối và lưu trữ các khối trong bộ nhớ 290. Khi tất cả các khối trong một ảnh được tái tạo, ảnh được tái tạo được sử dụng làm ảnh tham chiếu để dự báo ngoài của khối trong ảnh tiếp theo được mã hóa.

Sau đây, thiết bị giải mã video sẽ được mô tả.

Fig.4 là sơ đồ khối lấy làm ví dụ của thiết bị giải mã video có thể triển khai các phương án thực hiện sáng chế.

Thiết bị giải mã video bao gồm bộ giải mã 410, bộ lượng tử hóa đảo 420, bộ biến đổi ngược 430, bộ dự báo 440, bộ cộng 450, khối bộ lọc 460, và bộ nhớ 470. Như trong trường hợp của thiết bị mã hóa video trên Fig.2, mỗi phần tử của thiết bị mã hóa video có thể được triển khai như là vi mạch phần cứng, hoặc có thể được triển khai dưới dạng phần mềm, và bộ vi xử lý có thể được triển khai để thực thi các chức năng của phần mềm tương ứng với các phần tử tương ứng.

Bộ giải mã 410 giải mã dòng bit được nhận từ thiết bị mã hóa video, trích rút thông tin liên quan đến tách khối để xác định khối hiện tại để được giải mã, và trích rút thông tin dự báo cần để tái tạo khối hiện tại

và thông tin về tín hiệu dư.

Bộ giải mã 410 trích rút thông tin về kích thước CTU từ cú pháp mức cao chẳng hạn tập tham số chuỗi (SPS) hoặc tập tham số ảnh (PPS), xác định kích thước của CTU, và tách ảnh thành các CTU có kích thước định trước. Sau đó, bộ giải mã xác định CTU làm lớp cao nhất, tức là, nút gốc, của cấu trúc cây, và trích rút thông tin tách về CTU để tách CTU bằng cách sử dụng cấu trúc cây (chẳng hạn, cấu trúc QTBT).

Khi xác định khối hiện tại sẽ được giải mã qua việc tách cấu trúc cây, bộ giải mã 410 trích rút thông tin về loại dự báo chỉ báo liệu khối hiện tại được dự báo trong hoặc dự báo ngoài.

Khi thông tin loại dự báo chỉ báo dự báo trong, bộ giải mã 410 trích rút phần tử cú pháp cho thông tin dự báo trong (chế độ dự báo trong) về khối hiện tại.

Bộ giải mã 410 trích rút thông tin về các hệ số biến đổi được lượng tử hóa của khối hiện tại làm thông tin về tín hiệu dư.

Bộ lượng tử hóa đảo 420 lượng tử hóa đảo các hệ số biến đổi được lượng tử hóa. Bộ biến đổi ngược 430 biến đổi ngược các hệ số biến đổi được lượng tử hóa đảo từ miền tần số sang miền không gian để tái tạo các tín hiệu dư, và nhờ đó tạo khối dư cho khối hiện tại.

Bộ dự báo 440 bao gồm bộ dự báo trong 442 và bộ dự báo ngoài 444. Bộ dự báo trong 442 được kích hoạt khi loại dự báo của khối hiện tại là dự báo trong, và bộ dự báo ngoài 444 được kích hoạt khi loại dự báo của khối hiện tại là dự báo ngoài.

Bộ dự báo trong 442 xác định chế độ dự báo trong của khối hiện tại từ các chế độ dự báo trong từ phần tử cú pháp cho chế độ dự báo trong

trích rút từ bộ giải mã 410, và dự báo khối hiện tại bằng cách sử dụng các điểm ảnh tham chiếu quanh khối hiện tại theo chế độ dự báo trong.

Bộ dự báo ngoài 444 xác định thông tin chuyển động về khối hiện tại bằng cách sử dụng phần tử cú pháp của chế độ dự báo trong được trích rút từ bộ giải mã 410, và dự báo khối hiện tại bằng cách sử dụng thông tin chuyển động được xác định.

Bộ cộng 450 thêm khối dư được xuất ra từ bộ biến đổi ngược và khối dự báo được xuất ra từ bộ dự báo ngoài hoặc bộ dự báo trong để tái tạo khối hiện tại. Các điểm ảnh trong khối hiện tại được tái tạo được tận dụng làm các mẫu tham chiếu để dự báo trong của khối được giải mã sau.

Khối bộ lọc 460 bỏ lọc khối các đường biên giữa các khối được tái tạo để loại bỏ các đối tượng chặn gây ra bởi giải mã theo từng khối và lưu trữ các khối bị bỏ lọc khối trong bộ nhớ 470. Khi tất cả các khối trong một ảnh được tái tạo, ảnh được tái tạo được sử dụng làm ảnh tham chiếu cho dự báo ngoài của các khối trong ảnh tiếp theo sẽ được giải mã.

Các ví dụ của sáng chế thường liên quan đến kỹ thuật báo hiệu thông tin phân vùng khối QTBT và kỹ thuật xác định phân vùng khối từ thông tin phân vùng khối QTBT. Các kỹ thuật cụ thể của sáng chế có thể được thực hiện bởi bộ mã hóa 250 của thiết bị mã hóa video và bộ giải mã 410 của thiết bị giải mã video. Tức là, chẳng hạn, bộ mã hóa 250 và bộ giải mã 410 có thể thực hiện các phương án thực hiện sáng chế được mô tả dựa vào Fig.5 đến Fig.14 dưới đây. Trong các ví dụ khác, một hoặc nhiều khối khác của thiết bị mã hóa video và thiết bị giải mã video có thể dùng để thực hiện các phương án thực hiện sáng chế bên cạnh đó.

Trước hết, phương pháp biểu diễn thông tin phân vùng QT trong cấu trúc phân vùng khối QTBT sẽ được mô tả.

Trong cấu trúc QTBT, CTU, vốn là nút gốc, trước hết có thể được tách theo cách thức QT. Việc tách QT có thể được lặp lại đệ quy. Cờ thứ nhất (QT_split_flag) được sử dụng để chỉ báo liệu khối cụ thể có được tách QT hay không. Chẳng hạn, như được thể hiện trong Bảng 3, cờ thứ nhất = 1 chỉ báo rằng khối cụ thể được tách thành bốn khối có cùng kích thước, và cờ thứ nhất = 0 chỉ báo rằng khối cụ thể không được tách QT.

Bảng 3

Cờ tách QT	Giá trị
Không tách	0
Tách	1

Fig.5 là sơ đồ minh họa ví dụ tách QT và cấu trúc cây đại diện tách QT. “0” hoặc “1” được đánh dấu trên mỗi nút là giá trị của QT_split_flag chỉ báo liệu nút tương ứng được tách QT. Trên Fig.5(b), các nút được đánh dấu xám là các nút lá của QT, và không còn được tách QT. Ở đây, QT_split_flag cho các khối được báo hiệu theo thứ tự z-scan. Nói theo cách khác, các nút trong cấu trúc cây được minh họa trên Fig.5(b) được báo hiệu theo thứ tự “tìm kiếm chiều sâu trước”. Như được mô tả trên đây, các nút lá của QT trong cấu trúc phân vùng QTBT có thể còn được phân vùng trong cấu trúc BT. Tức là, việc tách BT có thể được thực hiện đệ quy trên các khối hoàn thành phân vùng QT. Có thể có các loại tách trong BT. Chẳng hạn, trong một số ví dụ, có thể có các loại tách khối cụ thể theo phương ngang hoặc phương dọc thành hai khối có cùng kích thước (được chỉ báo bởi 1/2 trên Fig.3). Ngoài ra, có thể có các loại tách khối cụ thể thành hai khối bất đối xứng. Việc tách bất đối xứng có thể bao gồm tách khối cụ thể thành hai khối chữ nhật ở

tỷ lệ kích thước 1:3 (được chỉ báo bởi 1/4 trên Fig.3) và tách khối cụ thể thành hai khối chữ nhật ở tỷ lệ kích thước 3:1 (được chỉ báo bởi 3/4 trên Fig.3). Việc tách bất đối xứng cũng có thể bao gồm tách khối cụ thể thành ba (bộ ba) khối chữ nhật ở tỷ lệ kích thước 1:2:1 (được chỉ báo bởi '1/4 & 3/4' trên Fig.3).

Như là thông tin phân vùng BT về khối cụ thể, cờ thứ hai (BT_split_flag) có thể được sử dụng. Do hình dạng của khối tách BT thay đổi tùy thuộc vào hướng tách và loại tách, cờ thứ hai (BT_split_flag) có thể xác định liệu khối cụ thể được tách BT, và cũng xác định hướng tách và loại tách trong trường hợp tách BT.

1. Ngôn ngữ học cho cú pháp tách BT

Phương án thực hiện thứ nhất

Theo phương án thực hiện, giả sử rằng loại 1/2 xuất hiện thường xuyên nhất trong số bốn loại tách, các bit nhỏ nhất có thể được phân phối đến loại 1/2, và các bit lớn hơn có thể được phân phối đến các loại 3/4 và loại bộ ba (tri). Fig.6 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo các loại tách. Bảng 4 thể hiện ví dụ ngôn ngữ học của cú pháp tách BT, tức là, từ mã cho mỗi loại tách của cờ tách BT theo phương án thực hiện.

Bảng 4

Cờ tách BT	Từ mã
No_split	0
Split_Hor_1/2	100
Split_Hor_1/4	1010
Split_Hor_3/4	10110
Split_Hor_Tri	10111
Split_Ver_1/2	110
Split_Ver_1/4	1110

Split_Ver_3/4	11110
Split_Ver_Tri	11111

Phương án thực hiện thứ hai

Theo phương án thực hiện, giả sử rằng loại 1/2 xuất hiện thường xuyên nhất trong số bốn loại tách được minh họa trên Fig.3, các bit nhỏ nhất có thể được phân phối đến loại 1/2, và các bit lớn hơn có thể được phân phối đến các loại 1/4 và 3/4. Fig.7 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo các loại tách. Bảng 5 thể hiện ví dụ ngôn ngữ học của cú pháp tách BT, tức là, từ mã cho mỗi loại tách của cờ tách BT theo phương án thực hiện. Bit thứ nhất của các từ mã trong bảng 5 có thể chỉ báo liệu khối cụ thể được tách BT. Khi thực hiện tách BT, bit thứ nhất có thể xác định hướng tách, và các bit còn lại xác định loại tách.

Bảng 5

Cờ tách BT	Từ mã
No_split	0
Split_Hor_1/2	100
Split_Hor_1/4	10110
Split_Hor_3/4	10111
Split_Hor_Tri	1010
Split_Ver_1/2	110
Split_Ver_1/4	11110
Split_Ver_3/4	11111
Split_Ver_Tri	1110

Phương án thực hiện thứ ba

Theo phương án thực hiện, phương pháp phân phối số bit tương tự cho mỗi loại tách được đề xuất giả sử rằng bốn loại tách được phân phối đều (tức là, tần số xuất hiện là giống nhau). Fig.8 minh họa biểu diễn dạng cây của việc phân phối bit theo các loại tách. Ở đây, các vị trí của bốn loại tách có thể được chuyển đổi. Bảng 6 thể hiện ví dụ của ngôn ngữ học của cú pháp tách BT, tức là, từ mã cho mỗi loại tách của cờ tách

BT theo phương án thực hiện. Ở đây, khi các vị trí của bốn loại tách được chuyển đổi, cấu trúc bit của từ mã cho mỗi loại tách được thay đổi tương ứng.

Bảng 6

Cờ tách BT	Từ mã
No_split	0
Split_Hor_1/2	1000
Split_Hor_1/4	1001
Split_Hor_3/4	1010
Split_Hor_Tri	1011
Split_Ver_1/2	1100
Split_Ver_1/4	1101
Split_Ver_3/4	1110
Split_Ver_Tri	1111

Phương án thực hiện thứ tư

Phương án thực hiện đề cập đến phương pháp báo hiệu cho các loại tách BT khi chỉ ba loại tách (1/2, 1/4, 3/4) được sử dụng ngoại trừ loại Tri trong số bốn loại tách được minh họa trên Fig.3. Theo phương án thực hiện, giả sử rằng loại 1/2 xuất hiện thường xuyên nhất trong số ba loại tách, các bit nhỏ nhất có thể được phân phối đến loại 1/2 và các bit lớn hơn có thể được phân phối đến các loại 1/4 và 3/4. Fig.9 minh họa biểu diễn dạng cây của việc phân phối bit theo các loại tách. Bảng 7 thể hiện ví dụ ngôn ngữ học của cú pháp tách BT, tức là, từ mã theo cờ tách BT theo phương án thực hiện.

Bảng 7

Cờ tách BT	Từ mã
No_split	0
Split_Hor_1/2	100
Split_Hor_1/4	1010
Split_Hor_3/4	1011
Split_Ver_1/2	110
Split_Ver_1/4	1110
Split_Ver_3/4	1111

Phương án thực hiện thứ năm

Phương án thực hiện đề cập đến phương pháp báo hiệu cho các loại tách BT khi chỉ hai loại tách (Binary, Triple) được sử dụng ngoại trừ các loại 1/4 và 3/4 trong số bốn loại tách được minh họa trên Fig.3. Theo phương án thực hiện, phương pháp phân phối số lượng bit tương tự đến hai loại được đề xuất khi giả thiết rằng hai loại tách xuất hiện ở các tần số giống nhau. Fig.10 là sơ đồ minh họa biểu diễn dạng cây của việc phân phối bit theo các loại tách. Bảng 8 thể hiện ví dụ ngôn ngữ học của cú pháp tách BT, tức là, từ mã theo cờ tách BT theo phương án thực hiện.

Bảng 8

Cờ tách BT	Từ mã
No_split	0
Split_Hor_1/2	100
Split_Hor_Tri	101
Split_Ver_1/2	110
Split_Ver_Tri	111

2. Các giới hạn tách BT đệ quy

Nói chung, việc tách BT có cấu trúc phân cấp và được tách đệ quy. Theo khía cạnh của sáng chế, đề xuất rằng các giới hạn đặt lên cấu trúc BT được tách đệ quy để giảm các bit của cú pháp đại diện các loại tách. Sau đây, phương pháp tách BT có điều kiện có các giới hạn khác nhau sẽ được mô tả. Trong các ví dụ được mô tả dưới đây, giả sử rằng các loại tách thực hiện tách BT bao gồm bốn loại, bao gồm các loại 1/2, 1/4, 3/4, và Tri được minh họa trên Fig.3.

Giới hạn tách #1

Một phương pháp tách BT là để chỉ định một hoặc nhiều loại 1/4, 3/4, và Tri (ngoại trừ 1/2) trên Fig.3 như là các nút lá. Theo cách khác, khi

khối cụ thể được tách BT thành loại tách được chỉ định như là các nút lá, các khối phụ của khối cụ thể không còn được tách nữa. Do vậy, ngoại trừ thông tin phân vùng về khối cụ thể, thông tin phân vùng (chỉ báo không tách) về các khối phụ không cần được báo hiệu.

Chẳng hạn, trong trường hợp tổng quát trong đó giới hạn này không được thêm, khi khối cụ thể được tách BT theo phương ngang ở dạng loại $1/4$, và một trong các khối phụ là ‘không tách’, các bit đại diện các phần tách này là “1010” (split_hor_1/4) và “0” (không tách) theo, chẳng hạn, các từ mã được liệt kê trong Bảng 2. Tuy nhiên, dưới sự giới hạn của việc giới hạn loại $1/4$ ở các nút lá, các bit cho các phần tách là “1010” (split_hor_1/4), và bit cho “0” (không tách) không còn cần nữa. Do vậy, sáng chế có thể giảm số bit bởi giới hạn tách.

Giới hạn tách #2

Như là phương pháp khác tách BT, khi thực hiện tách khối loại $1/2$, phương pháp tách BT tổng quát có thể được áp dụng lặp lại sau khi khối được tách. Khi thực hiện tách khối một trong các loại $1/4$, $3/4$, và Tri, phương tiện tách giới hạn được áp dụng cho khối tách khi tách tiếp theo. Ở đây, phương tiện tách BT giới hạn nghĩa là, khi loại tách BT hiện tại là tách thẳng đứng (một trong các loại $1/4$, $3/4$, và Tri), loại tách BT tiếp theo bị giới hạn ở tách theo phương ngang, và rằng, khi loại tách BT hiện tại là tách theo phương ngang (một trong các loại $1/4$, $3/4$, và Tri), loại tách BT tiếp theo bị giới hạn ở tách theo phương thẳng đứng.

Chẳng hạn, trong trường hợp tổng quát trong đó giới hạn này không được thêm vào, khi xuất ra tách loại $1/4$ theo phương nằm ngang, và sau đó một trong các khối phụ được tách theo phương thẳng đứng ở dạng của loại $1/2$, các bit đại diện các phần tách này là “1010” (split_hor_1/4)

và “110” (split_ver_1/2) theo, chẳng hạn, các từ mã được liệt kê trong bảng 2. Trong bảng 2, từ mã (“110”) chỉ báo phân tách split_ver_1/2 bao gồm “1” chỉ báo tách, “1” chỉ báo “phương thẳng đứng”, và “0” chỉ báo loại 1/2. Tuy nhiên, dưới điều kiện mà tách tiếp theo sau khi tách $\frac{1}{4}$ theo phương ngang bị giới hạn ở tách theo phương thẳng đứng, các bit đại diện các phân tách này là “1010” (split_hor_1/4) và “10” (split_ver_1/2). Tức là, bit chỉ báo “phương thẳng đứng” không cần thiết. Kết quả là, loại tách split_ver_1/2 được biểu diễn dưới dạng “10”. Nhờ đó, một bit có thể được dành riêng.

Các từ mã mới cho các loại tách theo các giới hạn hiện tại các giới hạn hiện tại theo các phương án thực hiện từ thứ nhất đến thứ năm liên quan đến các bảng từ 2 đến 6 được tóm tắt trong bảng 9, và phương pháp xác định, bởi thiết bị giải mã video, việc tách thứ hai theo các giới hạn hiện tại được minh họa trên lưu đồ trên Fig.11.

Bảng 9

Cờ tách BT	Phương án thực hiện thứ nhất	Phương án thực hiện thứ hai	Phương án thực hiện thứ ba	Phương án thực hiện thứ tư	Phương án thực hiện thứ năm
No_split	0	0	0	0	0
Split_1/2	10	10	100	10	10
Split_1/4	110	1110	101	110	-
Split_3/4	1110	1111	110	111	-
Split_Tri	1111	110	111	-	11

Fig.11 minh họa phương pháp xác định phân vùng BT của khối phụ sau khi khối cụ thể được tách BT, trong đó khối thứ nhất tách chỉ báo tách BT khối cụ thể, và khối thứ hai tách chỉ báo tách BT khối phụ. Như được minh họa trên Fig.11, thiết bị giải mã video trước hết phân tách nhị phân thứ nhất (tức là, bit thứ nhất) của BT_split_flag (S1110). Nhị phân thứ nhất của BT_split_flag chỉ báo liệu khối phụ lại được tách BT. Khi

nhị phân thứ nhất bằng 0 (KHÔNG ở bước S1120), khối phụ không được tách BT, và việc tách khối phụ kết thúc. Khi nhị phân thứ nhất bằng 1 (CÓ ở bước S1120), khối phụ được tách BT, và thiết bị giải mã video xác định liệu khối thứ nhất tách đã xuất hiện theo phương thẳng đứng (bước S1130). Khi khối thứ nhất tách đã xuất hiện theo phương thẳng đứng (CÓ ở bước S1130), có thể suy ra rằng khối thứ hai tách (tức là, tách BT khối phụ) tách theo phương nằm ngang theo giới hạn tách #2. Do vậy, thiết bị giải mã video phân tách các số nhị phân còn lại của BT_split_flag để xác định một trong bốn loại tách theo phương nằm ngang (bước S1260). Khi khối thứ nhất tách đã xuất hiện theo phương nằm ngang (Không ở bước S1130), có thể suy ra rằng khối thứ hai tách (tức là, tách BT khối phụ) tách theo phương thẳng đứng theo Giới hạn tách #2. Do vậy, thiết bị giải mã video phân tích cú pháp các số nhị phân còn lại của BT_split_flag để xác định một trong bốn loại tách theo phương thẳng đứng (bước S1260).

Giới hạn tách #3

Phương tiện tách BT giới hạn khác là để giới hạn các loại tách sao cho, khi thực hiện tách khối theo một trong các loại 1/4, 3/4, và Tri, chỉ loại tách 1/2 được phép để tách tiếp theo khối tách. Theo cách khác, khi tách BT hiện tại là một trong các loại 1/4, 3/4, và Tri, chỉ loại tách 1/2 được phép là loại tách BT tiếp theo.

Chẳng hạn, trong trường hợp tổng quát trong đó giới hạn này không được thêm vào, khi tách loại 1/4 xuất hiện theo phương nằm ngang, và sau đó một trong các khối phụ được tách theo phương thẳng đứng ở dạng của loại 1/2, các bit đại diện các phần tách này là “1010” (split_hor_1/4) và “110” (split_ver_1/2) theo, chẳng hạn, các từ mã

được liệt kê trong bảng 2. Trong bảng 2, từ mã (“110”) chỉ báo phân tách split_ver_1/2 bao gồm “1” chỉ báo tách, “1” chỉ báo “phương thẳng đứng”, và “0” chỉ báo loại 1/2. Tuy nhiên, dưới điều kiện mà tách tiếp theo sau khi loại tách 1/4 tách theo phương ngang bị giới hạn ở loại 1/2, các bit đại diện các phân tách này là “1010” (split_hor_1/4) và “11” (split_ver_1/2). Tức là, bit chỉ báo “1/2” không cần thiết. Kết quả là, loại tách split_ver_1/2 được biểu diễn dưới dạng “11.” Nhờ đó, một bit có thể được để dành.

Các từ mã mới cho các loại tách theo giới hạn hiện tại theo các phương án thực hiện từ thứ nhất đến thứ năm được tóm tắt trong bảng 10, và phương pháp xác định, bởi thiết bị giải mã video, tách theo giới hạn hiện tại thứ hai được minh họa trên lưu đồ trên Fig.12.

Bảng 10

BT_split_flag	Phương án thực hiện thứ nhất	Phương án thực hiện thứ hai	Phương án thực hiện thứ ba	Phương án thực hiện thứ tư	Phương án thực hiện thứ năm
No split	0	0	0	0	0
Split_Hor	10	10	10	10	10
Split_Ver	11	1	11	11	11

Như được minh họa trên Fig.12, thiết bị giải mã video trước hết phân tích cú pháp nhị phân thứ nhất (tức là, bit thứ nhất) của BT_split_flag (bước S1210). Nhị phân thứ nhất của BT_split_flag chỉ báo liệu khối phụ lại được tách BT. Khi nhị phân thứ nhất bằng 0 (Không ở bước S1220), khối phụ không được tách BT, và kết thúc tách khối phụ. Khi nhị phân thứ nhất bằng 1 (Có ở bước S1220), khối phụ được tách BT, và thiết bị giải mã video phân tích cú pháp nhị phân thứ hai (tức là, bit thứ hai) của BT_split_flag (bước S1230). Nhị phân thứ hai của BT_split_flag chỉ báo liệu hướng tách của khối thứ hai tách (tức là, tách

BT khối phụ) là phương nằm ngang phương thẳng đứng. Theo giới hạn tách #2, loại 1/2 tách được phép dưới dạng loại khối thứ hai tách. Do đó, khi nhị phân thứ hai của BT_split_flag chỉ báo tách theo phương nằm ngang, có thể suy ra rằng khối thứ hai tách được tách_hor_1/2. Một cách tương tự, khi nhị phân thứ hai của BT_split_flag chỉ báo tách theo phương thẳng đứng, có thể suy ra rằng khối thứ hai tách được tách_ver_1/2 (S1240).

Giới hạn tách #4

Phương pháp khác tách BT là áp dụng cả giới hạn theo phương và giới hạn cho phép chỉ loại 1/2. Chẳng hạn, khi loại tách BT hiện tại thực hiện tách (của một trong các loại 1/4, 3/4, và Tri) theo phương thẳng đứng, loại tách BT tiếp theo bị giới hạn ở loại 1/2 tách theo phương nằm ngang. Tuy nhiên, khi loại tách BT hiện tại thực hiện tách (của một trong các loại 1/4, 3/4, và Tri) theo phương nằm ngang, loại tách BT tiếp theo bị giới hạn ở loại 1/2 tách theo phương thẳng đứng. Cuối cùng, ngôn ngữ học của cú pháp BT_split_flag cho loại tách giới hạn này là không tách (“0”) và tách (“1”), trong đó “tách” có thể đề cập đến loại 1/2 tách theo phương nằm ngang hoặc phương thẳng đứng tùy thuộc loại tách BT trước đó.

Chẳng hạn, trong trường hợp tổng quát trong đó giới hạn này không được thêm vào, khi tách loại 1/4 xuất hiện theo phương thẳng đứng, và sau đó một trong các khối phụ được tách theo phương thẳng đứng của loại 1/2, các bit đại diện các phần tách này là “1010” (split_hor_1/4) và “110” (split_ver_1/2) theo, chẳng hạn, các từ mã được liệt kê trong bảng 2. Trong bảng 2, từ mã (“110”) chỉ báo phần tách split_ver_1/2 bao gồm “1” chỉ báo tách, “1” chỉ báo “phương thẳng đứng”, và “0” chỉ báo loại

1/2. Tuy nhiên, khi tách sau khi loại tách 1/4 tách theo phương ngang bị giới hạn ở loại 1/4 tách theo phương thẳng đứng, các bit đại diện các phần tách này là “1010” (split_hor_1/4) và “1” (split_ver_1/2). Tức là, các bit chỉ báo phương thẳng đứng và loại 1/2 không cần thiết. Kết quả là, loại tách split_ver_1/2 được biểu diễn dưới dạng “1”. Nhờ đó, 2 bit có thể được để dành.

Các từ mã mới cho các loại tách theo giới hạn tách \$4 cho tất cả năm phương án thực hiện nêu trên được tóm tắt trong bảng 11.

Bảng 11

BT_split_flag	Phương án thực hiện thứ nhất	Phương án thực hiện thứ hai	Phương án thực hiện thứ ba	Phương án thực hiện thứ tư	Phương án thực hiện thứ năm
No_split	0	0	0	0	0
Split	1	1	1	1	1

Bằng cách giới hạn việc tách ở loại tách đệ quy của BT như nêu trên, bit chỉ báo phương và/hoặc bit chỉ báo loại có thể được để dành. Có thể thấy từ các bảng 9 đến 11 rằng các bit nhỏ hơn có thể được phân phối bằng cách áp các giới hạn lên các loại tách thay vì lên từ mã cho loại tách khối tổng quát. Trong các giới hạn tách #2 đến #4 nêu trên, giới hạn được áp lên tách thứ hai sau khi tách thứ nhất (của một trong các loại 1/4, 3/4, Tri). Khối sau khi tách thứ nhất có hình dạng như được thể hiện trong Fig.13, và khối được tách thành ba phần hoặc hai phần theo loại tách thứ nhất. Chẳng hạn, khi tách thứ nhất có loại Tri, khối được phân chia thành ba khối phụ A, B, và C như được thể hiện trong Fig.13(a). Khi tách thứ nhất có loại 1/4 hoặc 3/4, khối được phân chia thành hai khối phụ A và B như được thể hiện trong Fig.13(b).

Tách thứ hai sau khi tách thứ nhất có thể được áp dụng riêng rẽ cho mỗi trong (hai hoặc ba) khối phụ được phân chia như được thể hiện

trong Fig.13, hoặc việc tách tương tự có thể được áp dụng cho tất cả các khối phụ. Chẳng hạn, trong trường hợp trong đó khối được phân chia thành ba khối phụ như được thể hiện trong Fig.13(a), khi giả sử rằng tất cả các khối phụ được tách giống nhau, thì loại tách của ba khối phụ có thể được biểu diễn bằng cách hiển thị thông tin tách chỉ một lần. Ngược lại, trong trường hợp trong đó khối được phân chia thành ba khối phụ như được thể hiện trong Fig.13(a), khi giả sử rằng mỗi khối phụ trong các khối phụ phân chia được tách riêng rẽ, thông tin riêng rẽ về tách thứ hai cần cho mỗi khối trong khối phụ A, khối phụ B, và khối phụ C.

Theo sáng chế, giới hạn tách #2 đến giới hạn tách #4 áp dụng cho hai trường hợp nêu trên (tức là, trường hợp trong đó việc tách tương tự được áp dụng cho tất cả các khối phụ và trường hợp trong đó tách được áp dụng cho mỗi khối trong các khối phụ một cách riêng rẽ). Chẳng hạn, trong trường hợp giới hạn tách #4, khi tách thứ nhất thuộc loại Tri và thông tin tách được tạo cho mỗi khối trong các khối phụ (khối phụ A, khối phụ B, và khối phụ C) được phân chia trong tách thứ hai, tổng cộng 3 bit bao gồm 1 bit (không tách hoặc tách) cho thông tin tách về khối phụ A, 1 bit cho thông tin tách về khối phụ B, và 1 bit cho thông tin tách về khối phụ C được yêu cầu. Ngược lại, khi tách thứ nhất có loại Tri và thông tin tách tương tự được tạo cho tất cả các khối phụ được phân chia trong tách thứ hai, 1 bit cần để biểu diễn thông tin tách về khối phụ A, khối phụ B, và khối phụ C.

Cấu trúc phân vùng CTU-đến-CU, cú pháp (QT_split_flag và BT_split_flag) để đại diện thông tin tách, và ngôn ngữ học (tức là, nhị phân hóa) của cú pháp đã được mô tả trên đây.

3. Các tham số cho phân vùng khối QTBT

Để báo hiệu thông tin về việc tách khối bởi cấu trúc QTBT hiệu quả hơn, các tham số sau có thể được sử dụng thêm vào. Các tham số này có thể được xác định như là mức video, mức chuỗi, hoặc mức ảnh.

Theo một số phương án thực hiện việc tận dụng các loại tách BT như được thể hiện trong Fig.3 trong cấu trúc phân vùng khối QTBT, các tham số sau có thể được sử dụng.

- *MinBtSizeY*: Kích thước khối nhỏ nhất của các nút lá được phép trong BT

- *MinQtSizeY*: Kích thước khối nhỏ nhất của các nút lá được phép trong QT

- *MaxBtSizeY*: Kích thước khối lớn nhất của nút gốc được phép bởi BT

- *CtbSizeY*: Kích thước CTU

Theo cách khác, theo một số phương án thực hiện tận dụng các loại tách BT như được thể hiện trong Fig.3 trong cấu trúc phân vùng khối QTBT, các tham số được liệt kê dưới đây có thể được sử dụng. Ở đây, giả sử rằng loại 1/2 trong số các loại tách trên Fig.3 được phân loại là loại đối xứng, các loại 1/4 và 3/4 được phân loại là các loại bất đối xứng, và loại Tri phân chia khối trung gian đối xứng và cả hai khối bên bất đối xứng.

- *SymMinBtSizeY*: Kích thước khối nhỏ nhất của các nút lá BT được tách bằng cách tách đối xứng

- *AsymMinBtSizeY*: Kích thước khối nhỏ nhất của các nút lá BT được tách bằng cách tách bất đối xứng

- *MinQtSizeY*: Kích thước khối nhỏ nhất của các nút lá được phép

trong QT

- *MaxBtSizeY*: Kích thước khối lớn nhất của nút gốc được phép trong BT

- *CtbSizeY*: kích thước CTU

Các tham số cho các kích thước của các khối này cho phép biểu diễn hiệu quả hơn thông tin về phân vùng khối bằng cấu trúc QTBT như sau. Ở đây, *MinBtSizeY* chỉ báo kích thước nhỏ nhất của CU (hoặc CB). Do đó, khi CU được phân chia đối xứng và bất đối xứng, *AsymMinBtSizeY* có thể đại diện *MinBtSizeY*. Theo cách khác, trong tình huống cụ thể, *SymMinBtSizeY* có thể đại diện *MinBtSizeY*.

(1) Khi *CtbSizeY* lớn hơn kích thước biến đổi lớn nhất định trước, thiết bị mã hóa video không mã hóa thông tin tách (QT_split_flag) về QT cho CTB tương ứng, và thiết bị giải mã video tự động thiết lập thông tin về 1. Tức là, CTB cụ thể được tách QT vô điều kiện lên đến kích thước biến đổi lớn nhất. Kích thước biến đổi lớn nhất có thể được định trước mặc định hoặc có thể được báo hiệu như là một phần tử ngôn ngữ học trong tiêu đề.

(2) Việc tách BT không được phép cho khối có kích thước lớn hơn *MaxBtSizeY* trong QT. Do vậy, khi kích thước của nút lá của QT lớn hơn *MaxBtSizeY*, có thể suy ra rằng khối không được phép tách BT, và do đó thiết bị mã hóa video không báo hiệu thông tin tách (BT_split_flag) về BT cho khối. Khi khối cụ thể là nút lá của QT và kích thước của nó lớn hơn *MaxBtSizeY*, thiết bị giải mã video không giải mã BT_split_flag, hoặc không thực hiện tách thêm khối tương ứng nữa. Tức là, thiết bị giải mã video xác định khối như là nút lá (tức là, CU) của QTBT.

(3) Trong QT, khối có cùng kích thước với *MinQtSizeY* không được

tách nữa. Do vậy, thiết bị mã hóa video không mã hóa thông tin tách (QT_split_flag) về QT của khối. Tức là, khi khối được tách qua tách QT lặp lại để có kích thước của *MinQtSizeY*, khối không thực hiện tách QT thêm, hoặc không mã hóa thông tin tách QT cho khối nữa. Ở đây, thông tin tách QT chỉ báo không tách (“0”) và không được mã hóa. Ngoài ra, khi khối được tách để có kích thước của *MinQtSizeY* qua việc tách QT lặp lại, thiết bị giải mã video không giải mã thông tin tách (QT_split_flag) về QT cho khối (tức là, thiết bị tự động thiết lập bit bằng 0), hoặc không thực hiện tách QT nữa trên khối.

(4) Khi cả chiều rộng và chiều cao của khối cụ thể bằng *MinBtSizeY* (bao gồm *SymMinBtSizeY* hoặc *AsymMinBtSizeY*) trong khi tách BT, không thực hiện tách BT thêm. Do vậy, thiết bị mã hóa video không mã hóa thông tin tách (BT_split_flag) về BT cho khối cụ thể, mà tự động thiết lập thông tin bằng 0. Tức là, khi chiều rộng và chiều cao của khối được tách cho kích thước của *MinBtSizeY* trong khi tách BT lặp lại, thiết bị giải mã video không giải mã thông tin tách (BT_split_flag) về BT cho khối (tức là, thiết bị tự động thiết lập thông tin bằng 0), nhưng xác định khối như là nút lá (tức là, CU) của QTBT.

(5) Khi chiều rộng hoặc chiều cao của khối cụ thể bằng *MinBtSizeY* và nhị phân thứ nhất của BT_split_flag bằng “1”, có thể suy ra liệu hướng tách BT của khối cụ thể là nằm ngang hoặc phương thẳng đứng. Do vậy, nhị phân thứ hai của BT_split_flag, chỉ báo hướng tách BT, là không cần thiết. Tức là, khi chiều rộng của khối cụ thể bằng *MinBtSizeY* và nhị phân thứ nhất của BT_split_flag bằng “1”, có thể suy ra rằng việc tách BT khối cụ thể là tách theo phương ngang (“0”), và do vậy thiết bị mã hóa/giải mã không mã hóa/giải mã nhị phân thứ hai của

BT_split_flag cho khối cụ thể. Ngược lại, khi chiều cao của khối cụ thể bằng *MinBtSizeY* và nhị phân thứ nhất của BT_split_flag bằng “1”, có thể suy ra rằng tách BT khối cụ thể là tách theo phương thẳng đứng (“1”), và do vậy thiết bị mã hóa/giải mã không mã hóa/giải mã nhị phân thứ hai của BT_split_flag cho khối cụ thể. Fig.14 là lưu đồ minh họa quá trình cho việc loại trừ BT trong đó thiết bị giải mã video không báo hiệu nhị phân thứ hai của BT_split_flag.

Như được minh họa trên Fig.14, thiết bị giải mã video trước khi phân tích cú pháp nhị phân thứ nhất (tức là, bit thứ nhất) của BT_split_flag (bước S1410). Nhị phân thứ nhất của BT_split_flag chỉ báo liệu khối phụ lại được tách BT. Khi nhị phân thứ nhất bằng 0 (Không ở bước S1420), khối cụ thể không được tách BT, và kết thúc tách khối cụ thể. Khi nhị phân thứ nhất bằng 1 (Có ở bước S1420), khối phụ được tách BT, và thiết bị giải mã video xác định liệu độ rộng của khối bằng *MinBtSizeY* (S1430). Khi chiều rộng của khối cụ thể bằng *MinBtSizeY* (có ở bước S1430), thiết bị suy ra hướng tách BT của khối cụ thể là hướng nằm ngang, và phân tích cú pháp các nhị phân còn lại để xác định loại tách (tức là, một trong các loại trên Fig.3(b)) (bước S1480). Khi chiều rộng của khối cụ thể không bằng *MinBtSizeY* (Không ở bước S1430), thiết bị còn xác định liệu độ cao của khối cụ thể bằng *MinBtSizeY*. Khi độ cao của khối cụ thể bằng *MinBtSizeY* (Có ở bước S1440), thiết bị suy ra rằng hướng tách BT của khối cụ thể là phương thẳng đứng, và phân tích cú pháp các nhị phân còn lại để xác định loại tách (tức là, một trong các loại trên Fig.3(a)) (bước S1480). Khi độ cao của khối cụ thể không bằng *MinBtSizeY* (Không ở bước S1440), nhị phân thứ hai của BT_split_flag được phân tách. Trong trường hợp này,

nhị phân thứ hai chỉ báo hướng tách BT của khối cụ thể. Các nhị phân còn lại được phân tách để xác định loại tách cụ thể (bước S1480).

4. Cú pháp mức cao (High-Level Syntax, HLS)

Như nêu trên, các tham số này có thể được xác định như là mức video, mức chuỗi, hoặc mức ảnh. Do đó, thông tin đại diện các tham số này có thể được bao gồm trong tiêu đề của mức cao bao gồm SPS, tập tham số ảnh (picture parameter set, PPS), và tập tham số video (video parameter set, VPS). Trong phần mô tả dưới đây, các HLS để trình bày hiệu quả các tham số này được đề xuất.

Phương án thực hiện thứ nhất (HLS)

Các phần tử cú pháp được đề xuất theo phương án thực hiện được nêu dưới đây. Xem xét rằng kích thước khối có thể được biểu diễn dưới dạng lũy thừa 2, các phần tử cú pháp dưới đây được biểu diễn dưới dạng hàm logarit của giá trị thực được định nghĩa với cơ số 2.

1) **log2_luma_min_BT_size_minus2**: Định nghĩa kích thước khối BT nhỏ nhất được phép trong BT (kích thước khối BT nhỏ nhất) dựa trên khối luma.

2) **log2_diff_luma_min_QT_min_BT_size**: Định nghĩa hiệu số (kích thước khối QT nhỏ nhất - kích thước khối BT nhỏ nhất) giữa kích thước khối nhỏ nhất (kích thước khối QT nhỏ nhất) của các nút lá được phép trong QT và kích thước khối nhỏ nhất (kích thước khối BT nhỏ nhất) được phép trong BT, dựa trên khối luma.

3) **log2_diff_luma_max_BT_min_QT_size**: Định nghĩa hiệu số (kích thước khối BT lớn nhất - kích thước khối QT nhỏ nhất) giữa kích thước khối lớn nhất (kích thước khối BT lớn nhất) của nút gốc được

phép trong BT và kích thước khối nhỏ nhất (kích thước khối QT nhỏ nhất) của các nút lá được phép trong QT, dựa trên khối luma.

4) **log2_diff_luma_max_QT_max_BT_size**: Định nghĩa hiệu số (kích thước khối QT lớn nhất - kích thước khối BT lớn nhất: tức là, kích thước của khối CTU) giữa kích thước khối lớn nhất (kích thước khối QT lớn nhất) được phép trong QT và kích thước khối lớn nhất (kích thước khối BT lớn nhất) của nút gốc được phép trong BT, dựa trên khối luma.

Bảng 12 thể hiện dưới dạng ví dụ các phần tử cú pháp được định nghĩa trong tiêu đề SPS. Như nêu trên, các phần tử cú pháp có thể được định nghĩa trong tiêu đề PPS hoặc tương tự.

Bảng 12

seq_parameter_set_rbsp() {	Bộ mô tả
...	
log2_luma_min_BT_size_minus2	ue(v)
log2_diff_luma_min_QT_min_BT_size	ue(v)
log2_diff_luma_max_BT_min_QT_size	ue(v)
log2_diff_luma_max_QT_max_BT_size	ue(v)
...	

Như được suy ra từ các phần tử cú pháp trên đây, kích thước thực (MinBtSizeY) của khối luma BT nhỏ nhất được nêu bởi phương trình 5, kích thước thực (MinQtSizeY) của khối QT nhỏ nhất được nêu bởi phương trình 6, và kích thước thực (MaxBtSizeY) của khối BT lớn nhất được nêu bởi phương trình 7, và kích thước CTU thực (CtbSizeY) được nêu bởi phương trình 8.

$$\text{MinBtLog2SizeY} = \text{log2_luma_min_BT_size_minus2} + 2$$

$$\text{MinQtLog2SizeY} = \text{MinBtLog2SizeY} + \text{log2_diff_luma_min_QT_min_BT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MinQtLog2SizeY} +$$

$$\text{log2_diff_luma_max_BT_min_QT_size}$$

$$\text{CtbLog2SizeY} = \text{MaxBtLog2SizeY} + \log2_diff_luma_max_QT_max_BT_size$$

$$\text{MinBtSizeY} = 1 \ll \text{MinBtLog2SizeY} \quad (5)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (6)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (7)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (8)$$

Phương án thực hiện thứ hai (HLS)

Các phần tử cú pháp đề xuất theo phương án thực hiện được nêu dưới đây.

1) **log2_sym_luma_min_BT_size_minus2**: Định nghĩa kích thước khối nhỏ nhất (kích thước khối BT nhỏ nhất đối xứng) được phép trong BT dựa trên khối luma tách đối xứng.

2) **log2_asym_luma_min_BT_size_minus1**: Định nghĩa kích thước khối nhỏ nhất (kích thước khối BT nhỏ nhất bất đối xứng) được phép trong BT dựa trên khối luma tách bất đối xứng.

3) **log2_diff_luma_min_QT_min_BT_size**: Định nghĩa Hiệu số giữa kích thước khối BT nhỏ nhất và kích thước khối QT nhỏ nhất (kích thước khối QT nhỏ nhất - kích thước khối BT nhỏ nhất đối xứng).

5) **log2_diff_luma_max_BT_min_QT_size**: Định nghĩa hiệu số (kích thước khối BT lớn nhất - kích thước khối QT nhỏ nhất) giữa kích thước khối lớn nhất (kích thước khối BT lớn nhất) của nút gốc được phép trong BT và kích thước khối nhỏ nhất (kích thước khối QT nhỏ nhất) của các nút lá được phép trong QT, dựa trên khối luma.

6) **log2_diff_luma_max_QT_max_BT_size**: Định nghĩa hiệu số (kích thước khối QT lớn nhất - kích thước khối BT lớn nhất: tức là, kích thước của khối CTU) giữa kích thước khối lớn nhất (kích thước khối QT

lớn nhất) được phép trong QT và kích thước khối lớn nhất (kích thước khối BT lớn nhất) của nút gốc được phép trong BT, dựa trên khối luma.

Bảng 13 thể hiện dưới dạng ví dụ các phần tử cú pháp được định nghĩa trong SPS. Như được mô tả trên đây, các phần tử cú pháp có thể được định nghĩa trong PPS hoặc tương tự.

Bảng 13

<code>seq_parameter_set_rbsp() {</code>	Bộ mô tả
<code>...</code>	
<code>log2_sym_luma_min_BT_size_minus2</code>	<code>ue(v)</code>
<code>log2_asym_luma_min_BT_size_minus1</code>	<code>ue(v)</code>
<code>log2_diff_luma_min_QT_min_BT_size</code>	<code>ue(v)</code>
<code>log2_diff_luma_max_BT_min_QT_size</code>	<code>ue(v)</code>
<code>log2_diff_luma_max_QT_max_BT_size</code>	<code>ue()</code>
<code>...</code>	

Như được suy ra từ các phần tử cú pháp trên đây, kích thước thực (`SymMinBtSizeY`) của khối luma BT nhỏ nhất đối xứng được nêu bởi phương trình 9, kích thước (`AsymMinBtSizeY`) của khối luma BT nhỏ nhất bất đối xứng được nêu bởi phương trình 10, kích thước (`MinQtSizeY`) của khối QT nhỏ nhất được nêu bởi phương trình 11, kích thước (`MaxBtSizeY`) của khối BT lớn nhất được nêu bởi phương trình 12, và kích thước CTU (`CtbSizeY`) được nêu bởi phương trình 13.

$$\text{SymMinBtLog2SizeY} = \text{log2_sym_luma_min_BT_size_minus2} + 2$$

$$\text{AsymMinBtLog2SizeY} = \text{log2_asym_luma_min_BT_size_minus1} + 1$$

$$\text{MinQtLog2SizeY} = \text{SymMinBtLog2SizeY} + \text{log2_diff_luma_min_QT_min_BT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MinQtLog2SizeY} + \text{log2_diff_luma_max_BT_min_QT_size}$$

$$\text{CtbLog2SizeY} = \text{MaxBtLog2SizeY} + \text{log2_diff_luma_max_QT_max_BT_size}$$

$$\text{SymMinBtSizeY} = 1 \ll \text{SymMinBtLog2SizeY} \quad (9)$$

$$\text{AsymMinBtSizeY} = 1 \ll \text{AsymMinBtLog2SizeY} \quad (10)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (11)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (12)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (13)$$

Phương án thực hiện thứ ba (HLS)

Các phần tử cú pháp đề xuất theo phương án thực hiện được nêu dưới đây.

log2_asym_luma_min_BT_size_minus1: Định nghĩa kích thước khối nhỏ nhất (kích thước khối BT nhỏ nhất bất đối xứng) được phép trong BT, dựa trên khối luma tách bất đối xứng.

2) **log2_diff_luma_min_QT_min_BT_size:** Định nghĩa hiệu số (kích thước khối QT nhỏ nhất đối xứng - kích thước khối BT nhỏ nhất) giữa kích thước khối nhỏ nhất (kích thước khối QT nhỏ nhất) của các nút lá được phép trong QT dựa trên khối luma và kích thước khối nhỏ nhất (kích thước khối BT nhỏ nhất đối xứng) được phép trong BT dựa trên khối luma tách đối xứng.

3) **log2_diff_luma_max_BT_min_QT_size:** Hiệu số giữa kích thước khối BT lớn nhất và kích thước khối QT nhỏ nhất (kích thước khối BT lớn nhất-kích thước khối QT nhỏ nhất)

4) **log2_diff_luma_max_QT_max_BT_size:** Hiệu số giữa kích thước khối QT lớn nhất và kích thước khối BT lớn nhất (kích thước khối QT lớn nhất - kích thước khối BT lớn nhất: tức là, kích thước của khối CTU)

Bảng 13 thể hiện dưới dạng ví dụ các phần tử cú pháp được định nghĩa trong SPS. Như được mô tả trên đây, các phần tử cú pháp có thể

được định nghĩa trong PPS hoặc tương tự.

Bảng 14

seq_parameter_set_rbsp() {	Bộ mô tả
...	
log2_asym_luma_min_BT_size_minus1	ue(v)
log2_diff_luma_min_QT_min_BT_size	ue(v)
log2_diff_luma_max_BT_min_QT_size	ue(v)
log2_diff_luma_max_QT_max_BT_size	ue(v)
...	

Như được suy ra từ các phần tử cú pháp trên đây, kích thước thực (SymMinBtSizeY) của khối luma BT nhỏ nhất đối xứng được nêu bởi phương trình 14, kích thước (AsymMinBtSizeY) của khối luma BT nhỏ nhất bất đối xứng được nêu bởi phương trình 15, kích thước (MinQtSizeY) của khối QT nhỏ nhất được nêu bởi phương trình 16, kích thước (MaxBtSizeY) của khối BT lớn nhất được nêu bởi phương trình 17, và kích thước CTU (CtbSizeY) được nêu bởi phương trình 18.

$$\text{SymMinBtLog2SizeY} = \text{log2_asym_luma_min_BT_size_minus1} + 2$$

$$\text{AsymMinBtLog2SizeY} = \text{log2_asym_luma_min_BT_size_minus1} + 1$$

$$\text{MinQtLog2SizeY} = \text{SymMinBtLog2SizeY} +$$

$$\text{log2_diff_luma_min_QT_min_BT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MinQtLog2SizeY} +$$

$$\text{log2_diff_luma_max_BT_min_QT_size}$$

$$\text{CtbLog2SizeY} = \text{MaxBtLog2SizeY} + \text{log2_diff_luma_max_QT_max_BT_size}$$

$$\text{SymMinBtSizeY} = 1 \ll \text{SymMinBtLog2SizeY} \quad (14)$$

$$\text{AsymMinBtSizeY} = 1 \ll \text{AsymMinBtLog2SizeY} \quad (15)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (16)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (17)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (18)$$

Trong một số ví dụ, như được mô tả trên đây liên quan đến phương án thực hiện thứ hai, khi chiều sâu QT chung của QT được biểu diễn ở một trong các tập SPS, PPS, và tiêu đề lát, tập chiều sâu QT chung ở mức cao hơn CTU (tức là, tập trong SPS, PPS, hoặc tiêu đề lát) có thể được tham chiếu cho chiều sâu QT chung về độ chói và màu sắc trong CTU. Trong một số ví dụ khác, chiều sâu QT chung cho độ chói và màu sắc trong CTU có thể được biểu diễn như là hiệu số từ chiều sâu QT chung được thiết lập trên CTU.

Theo các phương án thực hiện từ thứ nhất đến thứ ba nêu trên, các tham số liên quan đến kích thước khối có thể được định nghĩa như là mức video, mức chuỗi, hoặc mức ảnh. Theo các phương án thực hiện nêu từ thứ tư đến thứ sáu dưới đây, các tham số bổ sung chỉ báo hiệu số (tức là, các độ lệch) từ các tham số nêu trên được định nghĩa trong tiêu đề dưới tiêu đề được sử dụng theo các phương án thực hiện từ thứ nhất đến thứ ba để đề xuất phương pháp điều chỉnh các tham số ở các mức thấp hơn. Tức là, đề xuất rằng các phân tử cú pháp theo các phương án thực hiện từ thứ nhất đến thứ ba được biểu diễn trong tiêu đề trên và các phân tử cú pháp để định nghĩa các độ lệch cho các phân tử cú pháp được biểu diễn trong tiêu đề trên được biểu diễn trong tiêu đề dưới. Ở đây, tiêu đề trên có thể là tiêu đề của mức chuỗi hoặc mức ảnh. Khi tiêu đề trên là tiêu đề mức chuỗi, tiêu đề dưới có thể là tiêu đề mức ảnh hoặc tiêu đề lát. Khi tiêu đề trên là tiêu đề mức ảnh, tiêu đề dưới có thể là tiêu đề lát.

Theo phương pháp này, bằng cách sử dụng giá trị cú pháp được biểu diễn trong tiêu đề dưới (chẳng hạn, tiêu đề lát), kích thước khối BT nhỏ

nhất, kích thước khối QT nhỏ nhất, kích thước khối BT lớn nhất, và/hoặc kích thước khối QT lớn nhất có thể được điều chỉnh ở mức tương ứng (chẳng hạn, mức lát).

Phương án thực hiện thứ tư (HLS)

■ Các phần tử cú pháp của tiêu đề trên

Các phần tử cú pháp được bao gồm trong tiêu đề trên giống như các phần tử theo phương án thực hiện thứ nhất.

- 1) `log2_luma_min_BT_size_minus1`
- 2) `log2_diff_luma_min_QT_min_BT_size`
- 3) `log2_diff_luma_max_BT_min_QT_size`
- 4) `log2_diff_luma_max_QT_max_BT_size`

■ Các phần tử cú pháp của tiêu đề dưới

Các phần tử cú pháp sau được bao gồm trong tiêu đề dưới.

1) `delta_log2_luma_min_BT_size`: Định nghĩa độ lệch (delta kích thước khối BT nhỏ nhất) được thêm vào kích thước khối BT nhỏ nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên dựa trên khối luma.

2) `delta_log2_luma_min_QT_size`: Định nghĩa độ lệch (kích thước khối QT nhỏ nhất delta) được thêm vào kích thước khối QT nhỏ nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

3) `delta_log2_luma_max_BT_size`: Định nghĩa độ lệch (kích thước khối BT lớn nhất delta) được thêm vào kích thước khối BT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

4) **delta_log2_luma_max_QT_size**: Định nghĩa độ lệch (kích thước khối QT lớn nhất delta) được thêm vào kích thước khối QT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

Như được suy ra từ các phần tử cú pháp của tiêu đề trên và tiêu đề dưới, kích thước thực (MinBtSizeY) của khối luma BT nhỏ nhất được nêu bởi phương trình 19, kích thước thực (MinQtSizeY) của khối QT nhỏ nhất được nêu bởi phương trình 20, kích thước thực (MaxBtSizeY) của kích thước khối BT lớn nhất được nêu bởi phương trình 21, và kích thước CTU thực (CtbSizeY) được nêu bởi phương trình 22.

$$\text{MinBtLog2SizeY} = \text{log2_luma_min_BT_size_minus2} + 2$$

$$\text{MinQtLog2SizeY} = \text{MinBtLog2SizeY} + \text{log2_diff_luma_min_QT_min_BT_size}$$

$$\begin{aligned} \text{MaxBtLog2SizeY} &= \text{MinQtLog2SizeY} + \\ &\text{log2_diff_luma_max_BT_min_QT_size} \end{aligned}$$

$$\text{CtbLog2SizeY} = \text{MaxBtLog2SizeY} + \text{log2_diff_luma_max_QT_max_BT_size}$$

$$\text{MinBtLog2SizeY} = \text{MinBtLog2SizeY} + \text{delta_log2_luma_min_BT_size}$$

$$\text{MinQtLog2SizeY} = \text{MinQtLog2SizeY} + \text{delta_log2_luma_min_QT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MaxBtLog2SizeY} + \text{delta_log2_luma_max_BT_size}$$

$$\text{CtbLog2SizeY} = \text{CtbLog2SizeY} + \text{delta_log2_luma_max_QT_size}$$

$$\text{MinBtSizeY} = 1 \ll \text{MinBtLog2SizeY} \quad (19)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (20)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (21)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (22)$$

Phương án thực hiện thứ năm (HLS)

■ Các phần tử cú pháp của tiêu đề trên

Các phần tử cú pháp được bao gồm trong tiêu đề trên giống như các phần tử theo phương án thực hiện thứ hai.

1) `log2_sym_luma_min_BT_size_minus2`

2) `log2_asym_luma_min_BT_size_minus1`

3) `log2_diff_luma_min_QT_min_BT_size`

4) `log2_diff_luma_max_BT_min_QT_size`

5) `log2_diff_luma_max_QT_max_BT_size`

■ Các phần tử cú pháp của tiêu đề dưới

Các phần tử cú pháp sau đây được bao gồm trong tiêu đề dưới.

1) `delta_log2_sym_luma_min_BT_size`: Định nghĩa độ lệch (kích thước khối BT nhỏ nhất đối xứng delta) được thêm vào kích thước khối BT nhỏ nhất đối xứng được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên dựa trên khối luma tách đối xứng.

2) `delta_log2_asym_luma_min_BT_size`: Định nghĩa độ lệch (kích thước khối BT nhỏ nhất bất đối xứng delta) được thêm vào kích thước khối BT nhỏ nhất đối xứng được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên dựa trên khối luma tách bất đối xứng.

3) `delta_log2_luma_min_QT_size`: Định nghĩa độ lệch (kích thước khối QT nhỏ nhất delta) được thêm vào kích thước khối QT nhỏ nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

4) `delta_log2_luma_max_BT_size`: Định nghĩa độ lệch (kích thước khối BT lớn nhất delta) được thêm vào kích thước khối BT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề

trên.

5) **delta_log2_luma_max_QT_size**: Định nghĩa độ lệch (kích thước khối QT lớn nhất delta) được thêm vào kích thước khối QT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

Như được suy ra từ các phần tử cú pháp của tiêu đề trên và tiêu đề dưới, kích thước thực (SymMinBtSizeY) của khối luma BT nhỏ nhất đối xứng được nêu bởi phương trình 23, kích thước thực (AsymMinBtSizeY) của khối luma BT nhỏ nhất bất đối xứng được nêu bởi phương trình 24, kích thước thực (MinQtSizeY) của khối QT nhỏ nhất được nêu bởi phương trình 25, kích thước thực (MaxBtSizeY) của khối BT lớn nhất được nêu bởi phương trình 26, và kích thước CTU thực (CtbSizeY) được nêu bởi phương trình 27.

$$\text{SymMinBtLog2SizeY} = \log2_sym_luma_min_BT_size_minus2 + 2$$

$$\text{AsymMinBtLog2SizeY} = \log2_asym_luma_min_BT_size_minus1 + 1$$

$$\text{MinQtLog2SizeY} = \text{SymMinBtLog2SizeY} +$$

$$\log2_diff_luma_min_QT_min_BT_size$$

$$\text{MaxBtLog2SizeY} = \text{MinQtLog2SizeY} +$$

$$\log2_diff_luma_max_BT_min_QT_size$$

$$\text{CtbLog2SizeY} = \text{MaxBtLog2SizeY} + \log2_diff_luma_max_QT_max_BT_size$$

$$\text{SymMinBtLog2SizeY} = \text{SymMinBtLog2SizeY} +$$

$$\text{delta_log2_sym_luma_min_BT_size}$$

$$\text{AsymMinBtLog2SizeY} =$$

$$\text{AsymMinBtLog2SizeY} + \text{delta_log2_asym_luma_min_BT_size}$$

$$\text{MinQtLog2SizeY} = \text{MinQtLog2SizeY} + \text{delta_log2_luma_min_QT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MaxBtLog2SizeY} + \text{delta_log2_luma_max_BT_size}$$

$$\text{CtbLog2SizeY} = \text{CtbLog2SizeY} + \text{delta_log2_luma_max_QT_size}$$

$$\text{SymMinBtSizeY} = 1 \ll \text{SymMinBtLog2SizeY} \quad (23)$$

$$\text{AsymMinBtSizeY} = 1 \ll \text{AsymMinBtLog2SizeY} \quad (24)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (25)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (26)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (27)$$

Phương án thực hiện thứ sáu (HLS)

Các phần tử cú pháp của tiêu đề trên

Các phần tử cú pháp được bao gồm trong tiêu đề trên giống như các phần tử theo phương án thực hiện thứ ba.

1) **log2_asym_luma_min_BT_size_minus1**

2) **log2_diff_luma_min_QT_min_BT_size**

3) **log2_diff_luma_max_BT_min_QT_size**

4) **log2_diff_luma_max_QT_max_BT_size**

■ Các phần tử cú pháp của tiêu đề dưới

Các phần tử cú pháp sau được bao gồm trong tiêu đề dưới.

1) **delta_log2_asym_luma_min_BT_size**: Định nghĩa độ lệch (kích thước khối BT nhỏ nhất bất đối xứng delta) được thêm vào kích thước khối BT nhỏ nhất bất đối xứng được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên dựa trên khối luma.

2) **delta_log2_luma_min_QT_size**: Định nghĩa độ lệch (kích thước khối QT nhỏ nhất delta) được thêm vào kích thước của khối QT nhỏ nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

3) **delta_log2_luma_max_BT_size**: Định nghĩa độ lệch (kích thước

khối BT lớn nhất delta) được thêm vào kích thước khối BT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

4) **delta_log2_luma_max_QT_size**: Định nghĩa độ lệch (kích thước khối QT lớn nhất delta) được thêm vào kích thước khối QT lớn nhất được xác định bởi các phần tử cú pháp được định nghĩa trong tiêu đề trên.

Như được suy ra từ các phần tử cú pháp của tiêu đề trên và tiêu đề dưới, kích thước thực (SymMinBtSizeY) của khối luma BT nhỏ nhất đối xứng được nêu bởi phương trình 28, kích thước thực (AsymMinBtSizeY) của khối luma BT nhỏ nhất bất đối xứng được nêu bởi phương trình 29, kích thước thực (MinQtSizeY) của khối QT nhỏ nhất được nêu bởi phương trình 30, kích thước thực (MaxBtSizeY) của khối BT lớn nhất được nêu bởi phương trình 31, và kích thước CTU thực (CtbSizeY) được nêu bởi phương trình 32.

$$\text{SymMinBtLog2SizeY} = \text{log2_asym_luma_min_BT_size_minus1} + 2$$

$$\text{AsymMinBtLog2SizeY} = \text{log2_asym_luma_min_BT_size_minus1} + 1$$

$$\begin{aligned} \text{MinQtLog2SizeY} &= \text{SymMinBtLog2SizeY} + \\ &\text{log2_diff_luma_min_QT_min_BT_size} \end{aligned}$$

$$\begin{aligned} \text{MaxBtLog2SizeY} &= \text{MinQtLog2SizeY} + \\ &\text{log2_diff_luma_max_BT_min_QT_size} \end{aligned}$$

$$\begin{aligned} \text{CtbLog2SizeY} &= \text{MaxBtLog2SizeY} + \text{log2_diff_luma_max_QT_max_BT_size} \\ &\text{SymMinBtLog2SizeY} \end{aligned}$$

$$= \text{SymMinBtLog2SizeY} + \text{delta_log2_asym_luma_min_BT_size} +$$

1

$$\text{AsymMinBtLog2SizeY} = \text{AsymMinBtLog2SizeY} +$$

$\text{delta_log2_luma_min_BT_size}$

$$\text{MinQtLog2SizeY} = \text{MinQtLog2SizeY} + \text{delta_log2_luma_min_QT_size}$$

$$\text{MaxBtLog2SizeY} = \text{MaxBtLog2SizeY} + \text{delta_log2_luma_max_BT_size}$$

$$\text{CtbLog2SizeY} = \text{CtbLog2SizeY} + \text{delta_log2_luma_max_QT_size}$$

$$\text{SymMinBtSizeY} = 1 \ll \text{SymMinBtLog2SizeY} \quad (28)$$

$$\text{AsymMinBtSizeY} = 1 \ll \text{AsymMinBtLog2SizeY} \quad (29)$$

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (30)$$

$$\text{MaxBtSizeY} = 1 \ll \text{MaxBtLog2SizeY} \quad (31)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (32)$$

Bằng cách sử dụng thông tin về HLS nêu trên, thiết bị giải mã video có thể tính toán các tham số chẳng hạn MinBtSizeY , SymMinBtSizeY , AsymMinBtSizeY , MinQtSizeY , MaxBtSizeY , và CtbSizeY , và có thể suy ra, dựa trên các tham số được tính toán, liệu tách QT hoặc tách BT được áp dụng cho một số khối (phụ) khi thực hiện xác định cấu trúc phân vùng khối QTBT của CTU hoặc CTB cụ thể, mà không giải mã QT_split_flag hoặc BT_split_flag . Thiết bị mã hóa video không báo hiệu QT_split_flag hoặc BT_split_flag cho một số khối (phụ) mà thiết bị giải mã video có thể suy ra từ các tham số.

Mặc dù các phương án thực hiện lấy làm ví dụ đã được mô tả nhằm minh họa, song người có hiểu biết trung bình trong lĩnh vực sẽ hiểu rằng một số chỉnh sửa và thay đổi là có thể, mà không xa rời tinh thần và phạm vi của các phương án thực hiện. Các phương án thực hiện lấy làm ví dụ đã được mô tả cho ngắn gọn và rõ ràng. Do đó, người có hiểu biết trung bình trong lĩnh vực sẽ hiểu phạm vi của các phương án thực hiện không bị giới hạn bởi các phương án thực hiện được mô tả tường minh trên đây nhưng bao gồm các điểm yêu cầu bảo hộ và các

điểm tương đương của nó.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã dữ liệu video bao gồm các bước:

nhận dòng bit chứa dữ liệu được mã hóa cho khối dữ liệu video;

phân tách các phần tử cú pháp để phân vùng khối được định nghĩa ở thông tin tiêu đề của mức cao từ dòng bit, khối của dữ liệu video thuộc mức cao, trong đó việc phân vùng khối dựa trên cấu trúc phân vùng trong đó cây đa loại có gốc từ nút lá của cây tứ phân (quad tree, QT), và cây đa loại bao gồm cây nhị phân (binary tree, BT) trong đó khối tương ứng với nút cha mẹ được phép phân chia thành hai khối phụ tương ứng với hai nút con và cây tam phân (triple tree, TT) trong đó nút cha mẹ được phép phân chia thành ba khối phụ tương ứng với ba nút con;

tính toán các tham số để phân vùng khối từ các phần tử cú pháp, trong đó các tham số để phân vùng khối bao gồm tham số thứ nhất chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong BT, tham số thứ hai chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong QT, tham số thứ ba chỉ báo kích thước khối lớn nhất của nút gốc được phép trong BT, và tham số thứ tư chỉ báo kích thước của khối của dữ liệu video;

dẫn xuất cấu trúc phân vùng cho khối của dữ liệu video từ dòng bit theo các giới hạn được đặt ra dựa trên các tham số được tính toán; và

giải mã các khối phụ tương ứng với các nút lá của cấu trúc phân vùng cho khối của dữ liệu video từ dòng bit,

trong đó các phần tử cú pháp xác định:

kích thước của khối của dữ liệu video;

kích thước khối nhỏ nhất của nút lá được phép trong BT;

khác biệt giữa kích thước khối nhỏ nhất của nút lá được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT; và

khác biệt giữa kích thước khối lớn nhất của nút gốc được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT.

2. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khí chiều rộng và chiều cao của khối phụ cụ thể bằng kích thước được chỉ báo bởi tham số thứ nhất, suy ra rằng khối phụ cụ thể không còn được tách BT nữa.

3. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khí chiều cao của khối phụ cụ thể bằng kích thước được chỉ báo bởi tham số thứ nhất và khối phụ cụ thể được tách BT, suy ra rằng khối phụ cụ thể được tách theo chiều dọc.

4. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khí chiều rộng của khối phụ cụ thể bằng kích thước được chỉ báo bởi tham số thứ nhất and khối phụ cụ thể được tách BT, suy ra rằng khối phụ cụ thể được tách theo chiều ngang.

5. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khi khối phụ cụ thể bằng kích thước được chỉ báo bởi tham số thứ hai, suy ra rằng khối phụ cụ thể không được tách QT nữa.

6. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khi kích thước của nút lá của QT lớn hơn kích thước được chỉ báo bởi tham số thứ ba, suy ra rằng nút lá của QT là nút lá của QTBT.

7. Phương pháp theo điểm 1, trong đó việc dẫn xuất cấu trúc phân vùng bao gồm bước:

khi kích thước được chỉ báo bởi tham số thứ tư lớn hơn kích thước biến đổi lớn nhất định trước, suy ra rằng khối của dữ liệu video được tách QT đệ quy thành kích thước biến đổi lớn nhất.

8. Phương pháp mã hóa dữ liệu video bao gồm các bước:

xác định các tham số để phân vùng khối ở mức cao, trong đó việc phân vùng khối dựa trên cấu trúc phân vùng trong đó cây đa loại có gốc từ nút lá của QT, và cây đa loại bao gồm BT trong đó khối tương ứng với nút cha mẹ được phép phân chia thành hai khối phụ tương ứng với hai nút con và TT trong đó nút cha mẹ được phép phân chia thành ba khối phụ tương ứng với ba nút con;

mã hóa các phần tử cú pháp thứ nhất để phân vùng khối trong thông tin tiêu đề của mức cao;

mã hóa các phần tử cú pháp thứ hai xác định cấu trúc phân vùng cho khối dữ liệu video theo các giới hạn được đặt ra dựa trên các tham số được xác định, khối của dữ liệu video thuộc mức cao; và

mã hóa các khối phụ tương ứng với các nút lá của cấu trúc phân vùng cho khối của dữ liệu video,

trong đó các tham số để phân vùng khối bao gồm tham số thứ nhất chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong BT, tham số thứ hai chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong QT, tham số thứ ba chỉ báo kích thước khối lớn nhất của nút gốc được phép trong BT, và tham số thứ tư chỉ báo kích thước của khối của dữ liệu video, và

trong đó các phân tử cú pháp thứ nhất xác định:

kích thước của khối của dữ liệu video;

kích thước khối nhỏ nhất của nút lá được phép trong BT;

khác biệt giữa kích thước khối nhỏ nhất của nút lá được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT; và

khác biệt giữa kích thước khối lớn nhất của nút gốc được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT.

9. Vật ghi máy tính đọc được bất biến lưu trữ dòng bit chứa dữ liệu được mã hóa cho khối dữ liệu video, dòng bit được tạo cấu hình được giải mã bằng các quá trình:

phân tách các phân tử cú pháp để phân vùng khối được định nghĩa ở thông tin tiêu đề của mức cao từ dòng bit, khối của dữ liệu video thuộc mức cao, trong đó việc phân vùng khối dựa trên cấu trúc phân vùng trong đó cây đa loại có gốc từ nút lá của QT, và cây đa loại bao gồm BT trong đó khối tương ứng với nút cha mẹ được phép phân chia thành hai khối phụ tương ứng với hai nút con và TT trong đó nút cha mẹ được phép phân chia thành ba khối phụ tương ứng với ba nút con;

tính toán các tham số để phân vùng khối từ các phần tử cú pháp, trong đó các tham số để phân vùng khối bao gồm tham số thứ nhất chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong BT, tham số thứ hai chỉ báo kích thước khối nhỏ nhất của nút lá được phép trong QT, tham số thứ ba chỉ báo kích thước khối lớn nhất của nút gốc được phép trong BT, và tham số thứ tư chỉ báo kích thước của khối của dữ liệu video;

dẫn xuất cấu trúc phân vùng cho khối của dữ liệu video từ dòng bit theo các giới hạn được đặt ra dựa trên các tham số được tính toán; và

giải mã các khối phụ tương ứng với các nút lá của cấu trúc phân vùng cho khối của dữ liệu video dựa trên dòng bit,

trong đó các phần tử cú pháp xác định:

kích thước của khối của dữ liệu video;

kích thước khối nhỏ nhất của nút lá được phép trong BT;

khác biệt giữa kích thước khối nhỏ nhất của nút lá được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT; và

khác biệt giữa kích thước khối lớn nhất của nút gốc được phép trong BT và kích thước khối nhỏ nhất của nút lá được phép trong QT.

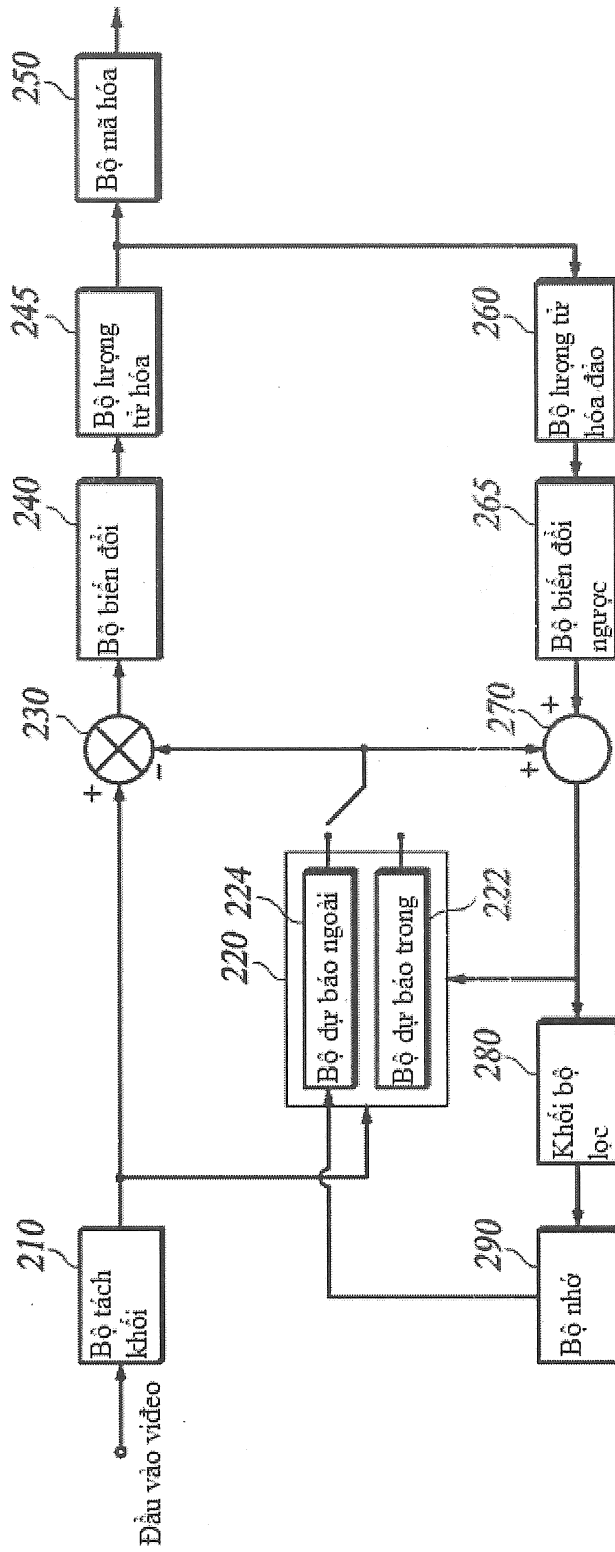
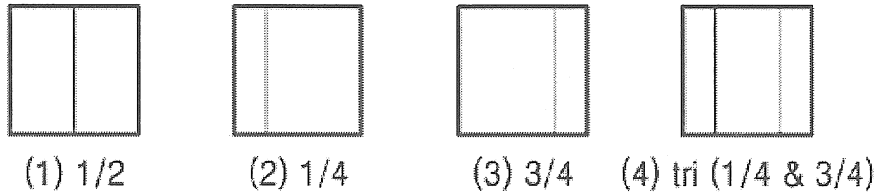
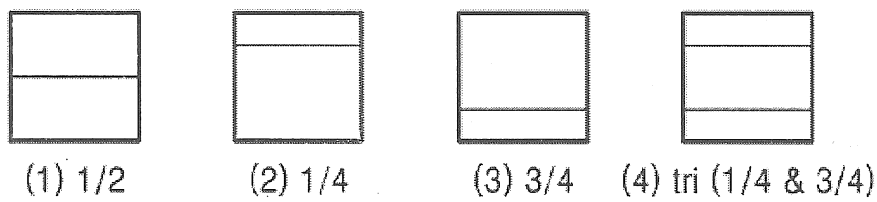


Fig.2

3/9



(a) Tách khối theo phương thẳng đứng



(b) Tách khối theo phương nằm ngang

Fig.3

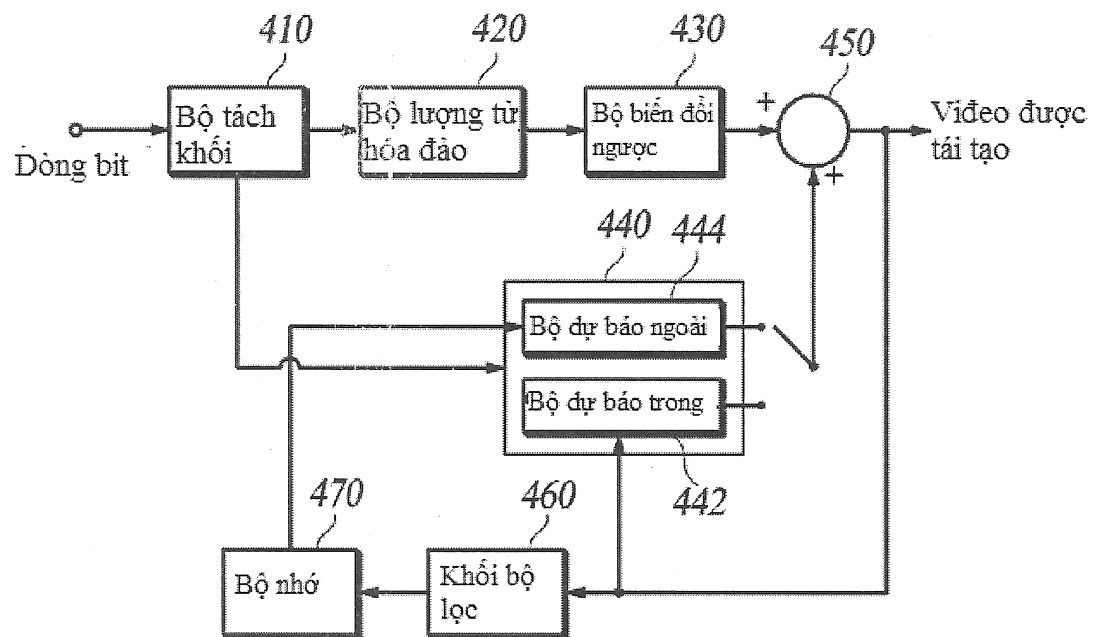
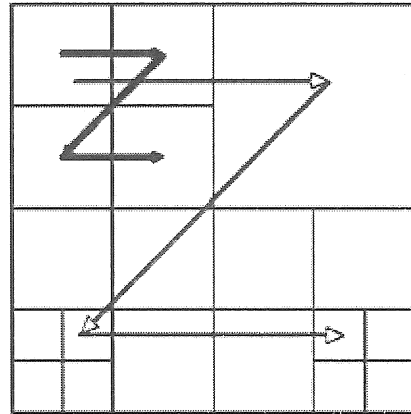
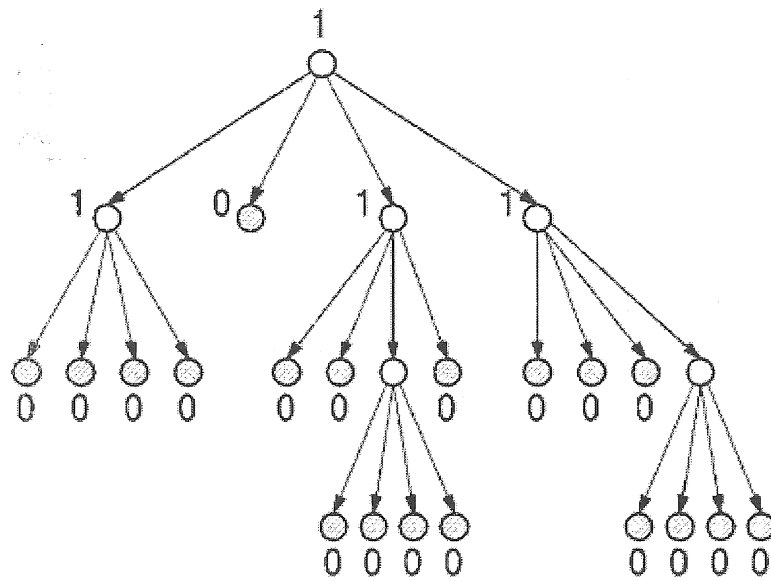


Fig.4

4/9



(a)



(b)

Fig.5

5/9

Fig.6

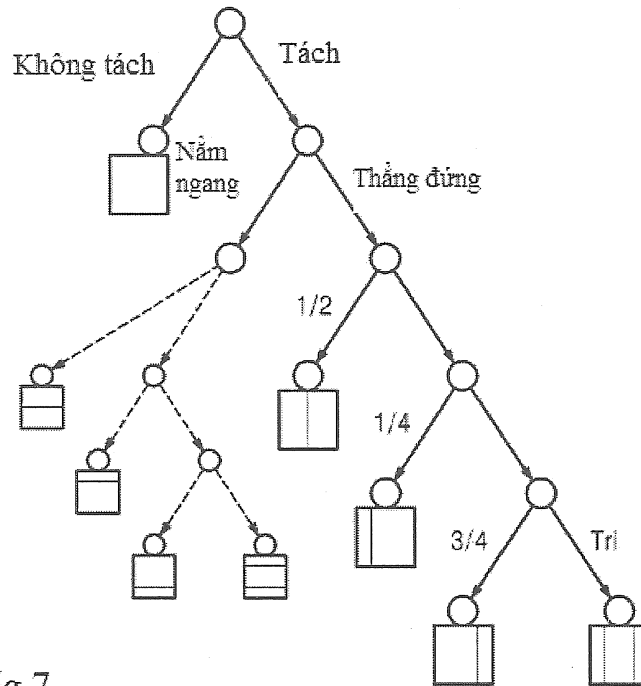
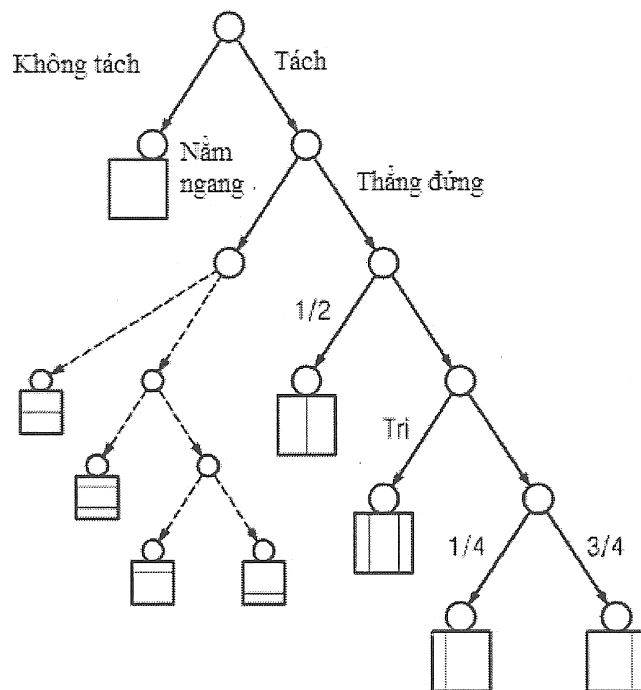


Fig.7



6/9

Fig.8

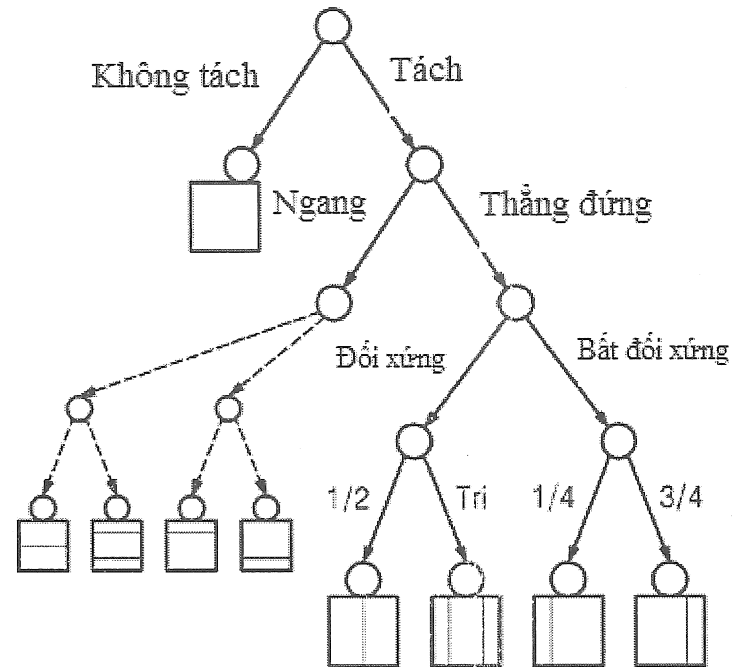
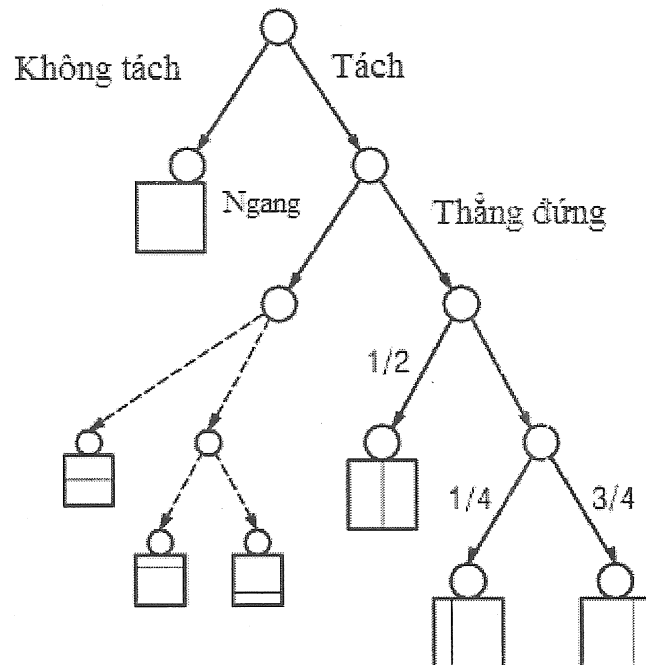


Fig.9



7/9

Fig.10

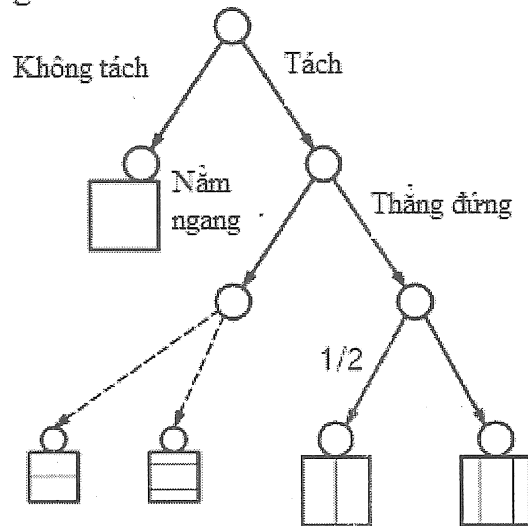
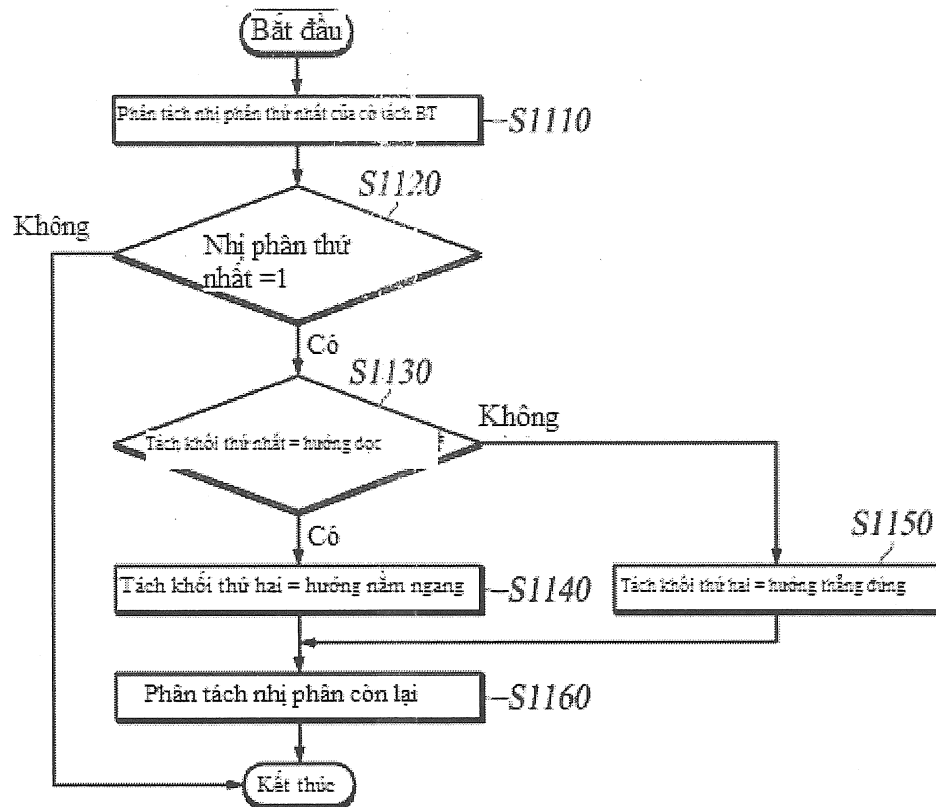


Fig.11



8/9

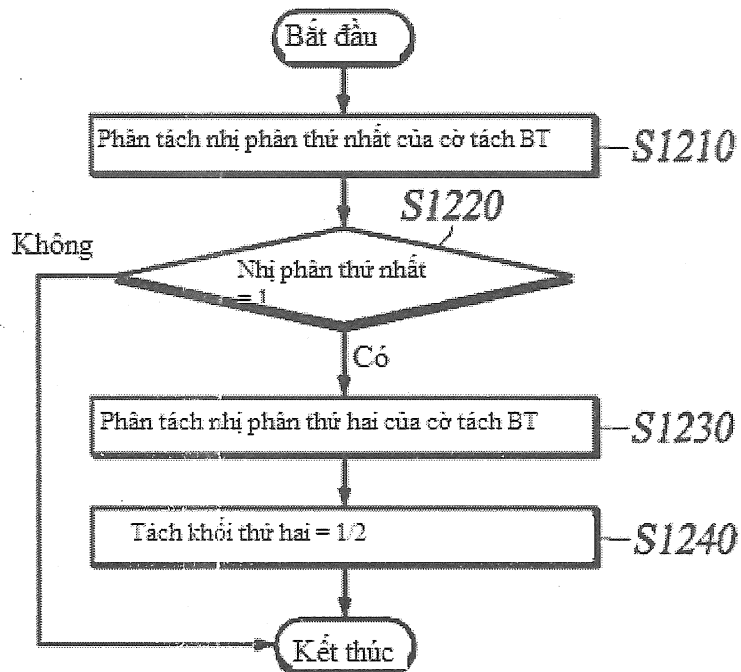
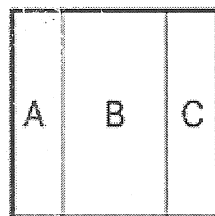
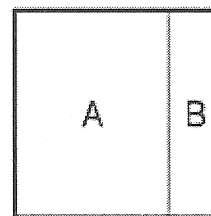


Fig.12



(a) Tách thành ba vùng



(b) Tách thành hai vùng

Fig.13

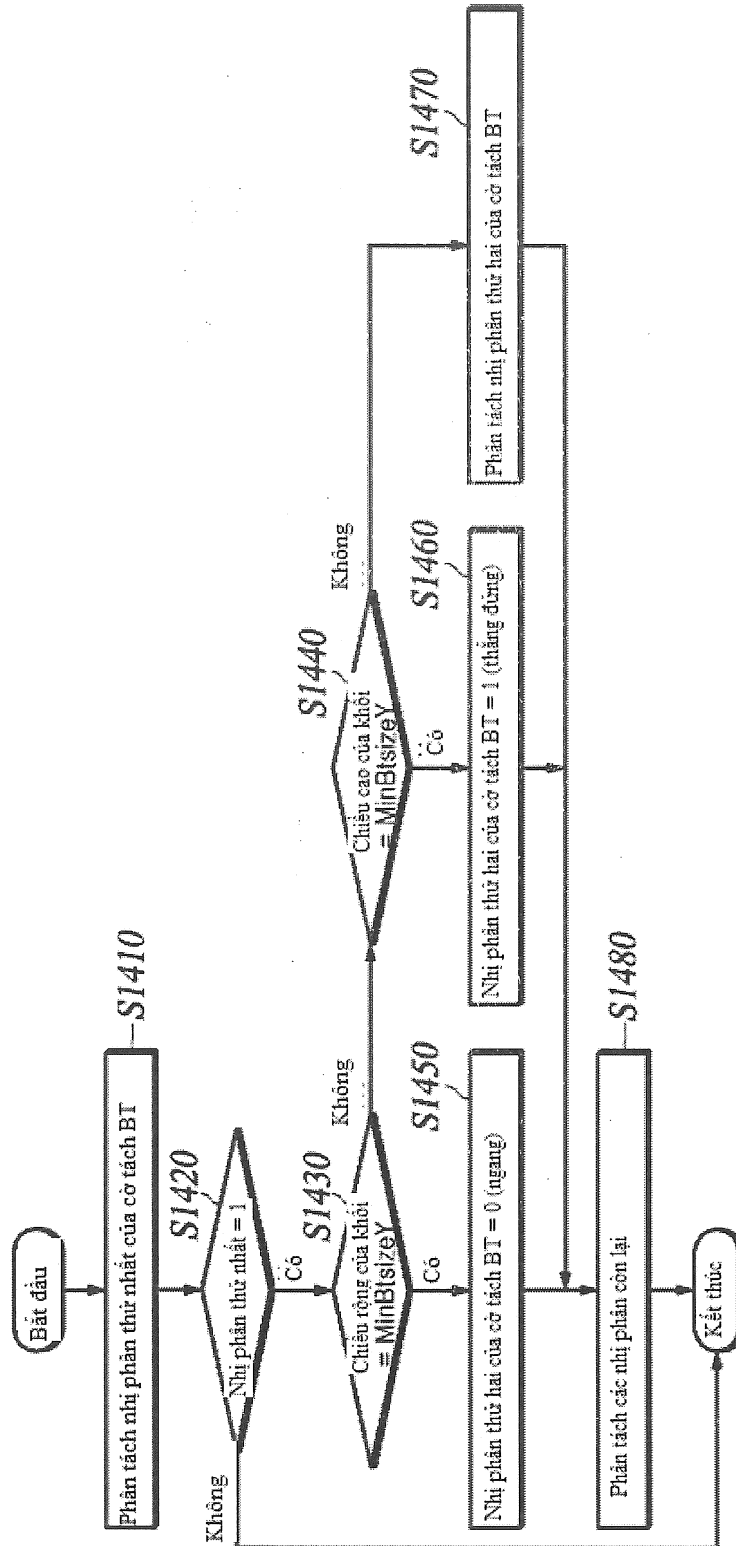


Fig.14