



(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)



1-0043296

(51)^{2020.01} H04N 19/91; H04N 19/70

(13) B

(21) 1-2021-05479

(22) 12/03/2020

(86) PCT/US2020/022355 12/03/2020

(87) WO2020/186038 A1 17/09/2020

(30) 62/817,475 12/03/2019 US; 62/834,297 15/04/2019 US; 62/871,519 08/07/2019 US;
16/816,016 11/03/2020 US

(45) 25/02/2025 443

(43) 25/11/2021 404

(73) QUALCOMM INCORPORATED (US)

ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, CA
92121-1714, United States of America

(72) DONG, Jie (CN); SAID, Amir (US); SEREGIN, Vadim (US); KARCZEWICZ,
Marta (US).

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP MÃ HÓA ENTROPY VÀ GIẢI MÃ ENTROPY DỮ LIỆU
VIDEO, THIẾT BỊ MÃ HÓA VÀ GIẢI MÃ VIDEO, VÀ PHƯƠNG TIỆN LƯU
TRỮ BẤT BIẾN ĐỌC ĐƯỢC BẰNG MÁY TÍNH

(21) 1-2021-05479

(57) Sáng chế đề cập đến phương pháp mã hóa entropy và giải mã entropy dữ liệu video, thiết bị mã hóa và giải mã video, và phương tiện lưu trữ bất biến đọc được bằng máy tính. Phương pháp giải mã entropy dữ liệu video làm ví dụ bao gồm các bước: truy hồi giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước và trong miền tuyến tính, trạng thái xác suất ban đầu của ngữ cảnh; và giải mã entropy, từ dòng bit và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, giá trị bin cho phần tử cú pháp.

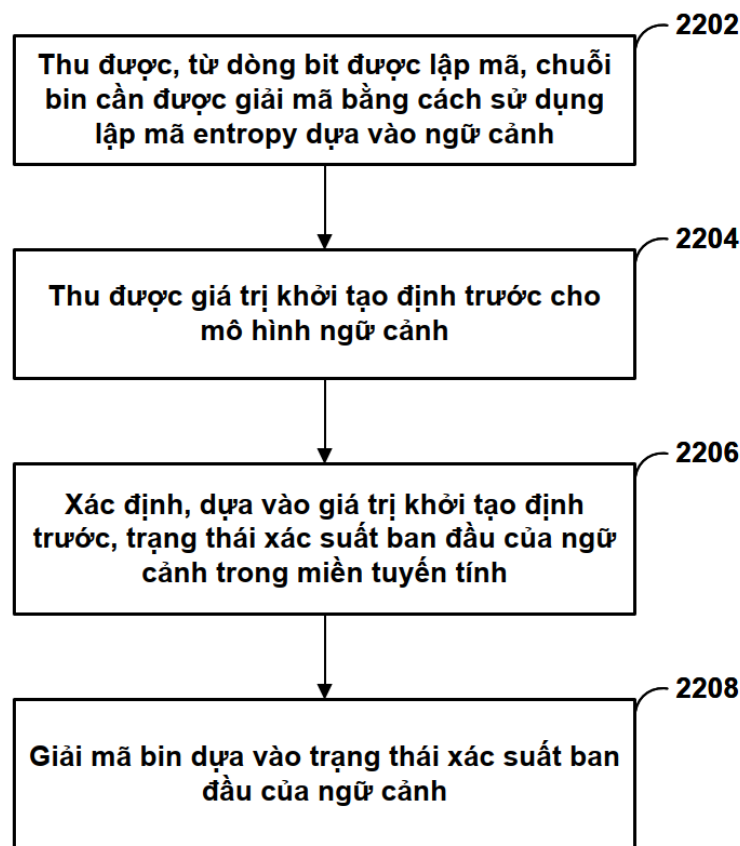


Fig.22

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến kỹ thuật mã hóa và giải mã video.

Tình trạng kỹ thuật của sáng chế

Các tính năng video kỹ thuật số có thể được đưa vào nhiều loại thiết bị bao gồm máy thu hình kỹ thuật số, hệ thống phát rộng trực tiếp kỹ thuật số, hệ thống phát rộng không dây, thiết bị kỹ thuật số hỗ trợ cá nhân (personal digital assistant - PDA), máy tính xách tay hoặc máy tính để bàn, máy tính bảng, máy đọc sách điện tử, máy ảnh kỹ thuật số, thiết bị ghi số, máy phát phương tiện kỹ thuật số, thiết bị trò chơi điện tử, bàn giao tiếp trò chơi điện tử, điện thoại di động hoặc điện thoại vô tuyến vệ tinh, thiết bị được gọi là “điện thoại thông minh”, thiết bị hội nghị truyền hình từ xa, thiết bị truyền phát trực tiếp video, và thiết bị tương tự. Các thiết bị video kỹ thuật số thực hiện các kỹ thuật lập mã video, chẳng hạn như các kỹ thuật được mô tả theo các chuẩn được quy định bởi MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Phần 10, lập mã video nâng cao (Advanced Video Coding - AVC), lập mã video hiệu suất cao (High Efficiency Video Coding - HEVC), và các phiên bản mở rộng của các chuẩn này. Các thiết bị video có thể truyền, nhận, mã hóa, giải mã và/hoặc lưu trữ thông tin video kỹ thuật số hiệu quả hơn bằng cách thực hiện các kỹ thuật lập mã video như vậy.

Các kỹ thuật lập mã video bao gồm dự đoán không gian (nội ảnh) và/hoặc dự đoán thời gian (liên ảnh) để giảm hoặc loại bỏ dữ liệu dư vốn có trong các chuỗi video. Đối với kỹ thuật lập mã video dựa trên khối, lát video (ví dụ, ảnh video hoặc một phần ảnh video) có thể được chia thành các khối video, các khối này còn có thể được gọi là đơn vị cây lập mã (coding tree unit - CTU), đơn vị lập mã (coding unit - CU) và/hoặc nút lập mã. Các khối video trong lát được lập mã nội ảnh (I) của ảnh được mã hóa bằng cách sử dụng kỹ thuật dự đoán không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng một ảnh. Các khối video trong lát được lập mã liên ảnh (P hoặc B) của ảnh có thể sử dụng kỹ thuật dự đoán không gian đối với các mẫu tham chiếu trong các khối lân cận trong cùng một ảnh, hoặc dự đoán thời gian đối với các mẫu tham chiếu

trong các ảnh tham chiếu khác. Ảnh có thể được gọi là khung, và ảnh tham chiếu có thể được gọi là khung tham chiếu.

Bản chất kỹ thuật của sáng chế

Nói chung, sáng chế mô tả các kỹ thuật khởi tạo xác suất dùng cho việc lập mã số học để nén video. Ví dụ, bộ lập mã video có thể thiết lập (các) trạng thái xác suất ban đầu của mỗi mô hình ngữ cảnh được sử dụng trong bộ lập mã số học nhị phân. Quy trình này có thể được yêu cầu khi bắt đầu mã hóa entropy hoặc giải mã đơn vị có thể giải mã độc lập, chẳng hạn như lát hoặc ô.

Trong một ví dụ, phương pháp lập mã entropy dữ liệu video bao gồm các bước: thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước và trong miền tuyến tính, trạng thái xác suất ban đầu của ngữ cảnh; và lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Trong ví dụ khác, thiết bị bao gồm bộ nhớ để lưu trữ dữ liệu video; và một hoặc nhiều bộ xử lý được triển khai trong mạch và được tạo cấu hình để: thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước và trong miền tuyến tính, trạng thái xác suất ban đầu của ngữ cảnh; và lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Trong ví dụ khác, thiết bị bao gồm phương tiện để thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; phương tiện để xác định, dựa vào giá trị khởi tạo định trước và trong miền tuyến tính, trạng thái xác suất ban đầu của ngữ cảnh; và phương tiện để lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Trong ví dụ khác, phương tiện lưu trữ đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi sẽ khiến cho một hoặc nhiều bộ xử lý của bộ lập mã video thực hiện: thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước và trong miền tuyến tính, trạng thái xác suất ban đầu của ngữ cảnh; và lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Chi tiết về một hoặc nhiều khía cạnh của sáng chế được mô tả trong các hình vẽ kèm theo và phần mô tả chi tiết dưới đây. Các dấu hiệu, đối tượng và ưu điểm khác của các kỹ thuật được mô tả trong sáng chế sẽ trở nên rõ ràng qua phần mô tả, các hình vẽ và bộ yêu cầu bảo hộ.

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối minh họa một ví dụ về hệ thống mã hóa và giải mã video có thể thực hiện các kỹ thuật của sáng chế.

Các hình vẽ Fig.2A và Fig.2B là các sơ đồ khái niệm minh họa quy trình cập nhật phạm vi trong lập mã số học nhị phân.

Fig.3 là sơ đồ khái niệm minh họa quy trình xuất ra trong lập mã số học nhị phân.

Các hình vẽ Fig.4A và Fig.4B là các sơ đồ khái niệm minh họa ví dụ về cấu trúc cây nhị phân tứ phân (quadtree binary tree - QTBT) và đơn vị cây lập mã (coding tree unit - CTU) tương ứng.

Fig.5 là sơ đồ khối minh họa một ví dụ về bộ mã hóa video có thể thực hiện các kỹ thuật của sáng chế.

Fig.6 là sơ đồ khối minh họa bộ lập mã số học nhị phân thích ứng theo ngữ cảnh (context adaptive binary arithmetic coding - CABAC) trong bộ mã hóa video.

Fig.7 là sơ đồ khối minh họa một ví dụ về bộ giải mã video có thể thực hiện các kỹ thuật của sáng chế.

Fig.8 là sơ đồ khối minh họa bộ lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC) trong bộ giải mã video.

Fig.9 là đồ thị minh họa các trạng thái xác suất để biểu diễn các xác suất bằng 1, theo chuẩn HEVC.

Các hình vẽ Fig.10A và Fig.10B là đồ thị minh họa các trạng thái xác suất để biểu diễn các xác suất bằng 1, theo chuẩn VVC, trong đó Fig.10A có 1024 trạng thái xác suất (độ chính xác 10 bit) và Fig.10B có 16384 trạng thái xác suất (độ chính xác 14 bit).

Các hình vẽ Fig.11A và Fig.11B là đồ thị minh họa ánh xạ từ InitProbState đến trạng thái xác suất có thể được sử dụng trong máy lập mã số học của lập mã video đa năng (Versatile Video Coding - VVC), trong đó Fig.11A có 1024 trạng thái xác suất (độ chính xác 10 bit) và Fig.11B có 16384 trạng thái xác suất (độ chính xác 14 bit).

Fig.12 là đồ thị minh họa kiến thức đã biết về phân phối xác suất của phần tử cú pháp SaoMergeFlag (1 bin) được sử dụng trong lát I.

Fig.13A là đồ thị minh họa kiến thức đã biết về phân phối xác suất trên Fig.12 được biểu diễn bởi trạng thái xác suất trong miền lôgarit, và Fig.13B là đồ thị minh họa trạng thái xác suất của mô hình và SliceQP_Y với hàm tuyến tính.

Fig.14A là đồ thị minh họa kiến thức đã biết về phân phối xác suất trên Fig.12 được biểu diễn bởi trạng thái xác suất trong miền tuyến tính; Fig.14B là đồ thị minh họa trạng thái xác suất của mô hình và SliceQP_Y với hàm tuyến tính mà không tăng phạm vi của độ dốc và độ lệch được xác định trong HEVC và VVC hiện thời, và Fig.14C là đồ thị minh họa trạng thái xác suất của mô hình và SliceQP_Y với hàm tuyến tính có tăng phạm vi của độ dốc và độ lệch.

Fig.15A là đồ thị minh họa kiến thức đã biết về phân phối xác suất của phần tử cú pháp SplitFlag được sử dụng trong lát B; Fig.15B là đồ thị minh họa trạng thái xác suất lôgarit của mô hình và SliceQP_Y với hàm tuyến tính; Fig.15C là đồ thị minh họa trạng thái xác suất tuyến tính của mô hình và SliceQP_Y với hàm tuyến tính bằng cách sử dụng các độ dốc và các độ lệch được xác định trong HEVC và VVC hiện thời, và Fig.15D là đồ thị minh họa trạng thái xác suất tuyến tính của mô hình và SliceQP_Y với hàm tuyến tính có đường giao nhau ở SliceQP_Y bằng 32.

Các hình vẽ từ Fig.16A đến Fig.16C là đồ thị minh họa việc sử dụng hàm tuyến tính từng mảnh được xác định bởi nhiều đường giao nhau thích hợp với các phân phối xác suất.

Các hình vẽ Fig.17A và Fig.17B là đồ thị minh họa ví dụ về ánh xạ miền bậc hai InitProbState đến trạng thái xác suất có thể được sử dụng trong máy lập mã số học của

VVC, trong đó Fig.17A có 1024 trạng thái xác suất (độ chính xác 10 bit) và Fig.17B có 16384 trạng thái xác suất (độ chính xác 14 bit).

Các hình vẽ Fig.18A và Fig.18B là đồ thị minh họa các ví dụ về các phân phối xác suất cực trị.

Fig.19 là lưu đồ minh họa một ví dụ về phương pháp mã hóa khối hiện thời.

Fig.20 là lưu đồ minh họa một ví dụ về quy trình thực hiện mã hóa entropy dựa vào ngữ cảnh, theo một hoặc nhiều kỹ thuật của sáng chế.

Fig.21 là lưu đồ minh họa một ví dụ về phương pháp giải mã khối hiện thời.

Fig.22 là lưu đồ minh họa một ví dụ về quy trình thực hiện giải mã entropy dựa vào ngữ cảnh, theo một hoặc nhiều kỹ thuật của sáng chế.

Mô tả chi tiết sáng chế

Nói chung, sáng chế mô tả các kỹ thuật khởi tạo xác suất dùng cho việc lập mã số học để nén video. Ví dụ, bộ lập mã video có thể thiết lập (các) trạng thái xác suất ban đầu của mỗi mô hình ngữ cảnh được sử dụng trong bộ lập mã số học nhị phân. Quy trình này có thể được yêu cầu khi bắt đầu mã hóa hoặc giải mã entropy đơn vị có thể giải mã độc lập, chẳng hạn như lát.

Để thực hiện quá trình khởi tạo, bộ lập mã video có thể thu được giá trị khởi tạo định trước (ví dụ, `initValue`) cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho một lát dữ liệu video. Bộ lập mã video có thể xác định, dựa vào giá trị khởi tạo định trước và trong miền lôgarit, trạng thái xác suất ban đầu của ngữ cảnh (ví dụ, `pStateIdx`). Bộ lập mã video có thể lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Khi lập mã dữ liệu video theo một số chuẩn video, bộ lập mã video có thể duy trì các trạng thái xác suất trong miền tuyến tính. Ví dụ, như được thảo luận chi tiết hơn ở phần sau, bộ lập mã số học nhị phân thích ứng theo ngữ cảnh (Context Adaptive Binary Arithmetic Coder - CABAC) được ứng dụng trong chuẩn lập mã video đa năng (Versatile Video Coding - VVC) có thể duy trì các trạng thái xác suất của các ngữ cảnh trong miền tuyến tính. Khi trạng thái xác suất ban đầu được xác định trong miền lôgarit, các bộ lập mã video có thể thực hiện các hoạt động khác nhau để chuyển đổi trạng thái xác suất ban đầu được xác định trong miền tuyến tính. Ví dụ, bộ lập mã video có thể sử

dụng bảng tìm kiếm (look-up table - LUT) để chuyển đổi trạng thái xác suất ban đầu từ miền lôgarit đến miền tuyến tính.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ lập mã video có thể trực tiếp xác định trạng thái xác suất ban đầu trong miền tuyến tính. Ví dụ, bộ mã hóa video có thể xác định, dựa vào giá trị khởi tạo định trước, trạng thái xác suất ban đầu của ngữ cảnh (ví dụ, pStateIdx) trong miền tuyến tính mà không sử dụng LUT để chuyển đổi trạng thái xác suất ban đầu từ miền lôgarit sang miền tuyến tính. Theo cách này, bộ lập mã video có thể tránh được việc phải lưu trữ và/hoặc truy cập LUT mà chuyển đổi giữa các trạng thái xác suất ban đầu trong miền lôgarit và miền tuyến tính.

Fig.1 là sơ đồ khối minh họa ví dụ về hệ thống mã hóa và giải mã video 100 có thể thực hiện các kỹ thuật của sáng chế. Các kỹ thuật của sáng chế nói chung hướng đến việc lập mã (mã hóa và/hoặc giải mã) dữ liệu video. Nói chung, dữ liệu video bao gồm dữ liệu bất kỳ để xử lý video. Do đó, dữ liệu video có thể bao gồm video thô, chưa được lập mã, video được mã hóa, video được giải mã (ví dụ, được tái tạo), và siêu dữ liệu video, chẳng hạn như dữ liệu báo hiệu.

Như được thể hiện trên Fig.1, hệ thống 100 bao gồm thiết bị nguồn 102 cung cấp dữ liệu video được mã hóa cần được giải mã và hiển thị bởi thiết bị đích 116 trong ví dụ này. Cụ thể, thiết bị nguồn 102 cung cấp dữ liệu video cho thiết bị đích 116 qua phương tiện đọc được bằng máy tính 110. Thiết bị nguồn 102 và thiết bị đích 116 có thể là thiết bị bất kỳ trong số rất nhiều thiết bị, bao gồm máy tính để bàn, máy tính notebook (tức là, máy tính xách tay), máy tính bảng, đầu thu giải mã, các máy điện thoại cầm tay như điện thoại thông minh, ti vi, máy ảnh, thiết bị hiển thị, trình phát phương tiện kỹ thuật số, bàn điều khiển trò chơi điện tử, thiết bị truyền phát trực tiếp dữ liệu video hoặc các thiết bị tương tự khác. Trong một số trường hợp, thiết bị nguồn 102 và thiết bị đích 116 có thể được trang bị để truyền thông không dây, và do đó có thể được gọi là các thiết bị truyền thông không dây.

Theo ví dụ trên Fig.1, thiết bị nguồn 102 bao gồm nguồn video 104, bộ nhớ 106, bộ mã hóa video 200 và giao diện đầu ra 108. Thiết bị đích 116 bao gồm giao diện đầu vào 122, bộ giải mã video 300, bộ nhớ 120 và thiết bị hiển thị 118. Theo sáng chế, bộ mã hóa video 200 của thiết bị nguồn 102 và bộ giải mã video 300 của thiết bị đích 116 có thể được tạo cấu hình để ứng dụng các kỹ thuật khởi tạo xác suất cho việc lập mã số học để nén video. Do đó, thiết bị nguồn 102 là một ví dụ về thiết bị mã hóa video, còn

thiết bị đích 116 là một ví dụ của thiết bị giải mã video. Theo các ví dụ khác, thiết bị nguồn và thiết bị đích có thể bao gồm các thành phần hoặc các cách sắp xếp khác. Ví dụ, thiết bị nguồn 102 có thể nhận dữ liệu video từ nguồn video bên ngoài, như camera ngoài. Tương tự, thiết bị đích 116 có thể tương tác với thiết bị hiển thị bên ngoài, chứ không phải bao gồm thiết bị hiển thị tích hợp.

Hệ thống 100 được thể hiện trên Fig.1 chỉ là một ví dụ. Nói chung, thiết bị mã hóa và/hoặc giải mã dữ liệu video kỹ thuật số bất kỳ có thể thực hiện các kỹ thuật khởi tạo xác suất cho việc lập mã số học để nén video. Thiết bị nguồn 102 và thiết bị đích 116 chỉ là các ví dụ về các thiết bị lập mã như vậy, trong đó thiết bị nguồn 102 tạo ra dữ liệu video được lập mã để truyền đến thiết bị đích 116. Sáng chế đề cập đến thiết bị “lập mã” là thiết bị thực hiện lập mã (mã hóa và/hoặc giải mã) dữ liệu. Do đó, bộ mã hóa video 200 và bộ giải mã video 300 lần lượt là các ví dụ của thiết bị lập mã, cụ thể là bộ mã hóa video và bộ giải mã video. Theo một số ví dụ, các thiết bị 102, 116 có thể hoạt động theo cách về cơ bản là đối xứng sao cho mỗi thiết bị 102, 116 bao gồm các thành phần mã hóa và giải mã video. Do vậy, hệ thống 100 có thể hỗ trợ truyền video một chiều hoặc hai chiều giữa các thiết bị video 102, 116, ví dụ, để truyền phát trực tiếp video, phát lại video, phát rộng video, hoặc điện thoại truyền hình.

Nói chung, nguồn video 104 là nguồn dữ liệu video (tức là, dữ liệu video thô, chưa được lập mã) và cung cấp chuỗi các ảnh liên tiếp (còn được gọi là "khung") của dữ liệu video cho bộ mã hóa video 200 để mã hóa dữ liệu cho các ảnh. Nguồn video 104 của thiết bị nguồn 102 có thể bao gồm thiết bị quay video, như máy quay video, kho trữ video chứa video thô đã quay trước đó, và/hoặc giao diện nạp video để nhận video từ nhà cung cấp nội dung video. Theo một phương án khác, nguồn video 104 có thể tạo ra dữ liệu dựa trên đồ họa máy tính dưới dạng video nguồn, hoặc kết hợp của video trực tiếp, video đã lưu trữ, và video được tạo ra bằng máy tính. Trong mỗi trường hợp, bộ mã hóa video 200 mã hóa dữ liệu video được quay, được quay trước hoặc được tạo ra bằng máy tính. Bộ mã hóa video 200 có thể sắp xếp lại các ảnh từ trình tự nhận được (đôi khi được gọi là “trình tự hiển thị”) thành trình tự lập mã để lập mã. Bộ mã hóa video 200 có thể tạo ra dòng bit bao gồm dữ liệu video được mã hóa. Thiết bị nguồn 102 sau đó có thể xuất ra dữ liệu video được mã hóa qua giao diện đầu ra 108 lên phương tiện đọc được bằng máy tính 110 để nhận và/hoặc truy hồi bằng, ví dụ, giao diện đầu vào 122 của thiết bị đích 116.

Bộ nhớ 106 của thiết bị nguồn 102 và bộ nhớ 120 của thiết bị đích 116 là các bộ nhớ đa năng. Theo một số ví dụ, các bộ nhớ 106, 120 có thể lưu trữ dữ liệu video thô, ví dụ, video thô từ nguồn video 104 và dữ liệu video thô, được giải mã từ bộ giải mã video 300. Ngoài ra hoặc theo cách khác, các bộ nhớ 106, 120 có thể lưu trữ các lệnh phần mềm có thể thực thi được lần lượt bằng, ví dụ, bộ mã hóa video 200 và bộ giải mã video 300. Mặc dù được thể hiện tách biệt với bộ mã hóa video 200 và bộ giải mã video 300 trong ví dụ này, nhưng cần phải hiểu rằng bộ mã hóa video 200 và bộ giải mã video 300 cũng có thể bao gồm các bộ nhớ trong dùng cho các mục đích tương tự hoặc tương đương về mặt chức năng. Hơn nữa, các bộ nhớ 106, 120 có thể lưu trữ dữ liệu video được mã hóa, ví dụ, được xuất ra từ bộ mã hóa video 200 và nhập vào bộ giải mã video 300. Theo một số ví dụ, các phần của các bộ nhớ 106, 120 có thể được phân bổ dưới dạng một hoặc nhiều bộ đệm video, ví dụ, để lưu trữ dữ liệu video thô, được giải mã và/hoặc được mã hóa.

Phương tiện đọc được bằng máy tính 110 có thể là kiểu phương tiện hoặc thiết bị bất kỳ có khả năng truyền dữ liệu video được mã hóa từ thiết bị nguồn 102 đến thiết bị đích 116. Trong một số ví dụ, phương tiện đọc được bằng máy tính 110 là phương tiện truyền thông cho phép thiết bị nguồn 102 truyền dữ liệu video được mã hóa trực tiếp đến thiết bị đích 116 theo thời gian thực, ví dụ, qua mạng tần số vô tuyến hoặc mạng dựa trên máy tính. Giao diện đầu ra 108 có thể điều biến tín hiệu truyền bao gồm dữ liệu video được mã hóa, và giao diện đầu vào 122 có thể giải điều biến tín hiệu truyền thu được, theo chuẩn truyền thông, chẳng hạn như giao thức truyền thông không dây. Phương tiện truyền thông có thể bao gồm phương tiện truyền thông không dây hoặc có dây bất kỳ, như phổ tần số vô tuyến (Radio Frequency - RF) hoặc một hay nhiều đường truyền vật lý. Phương tiện truyền thông có thể tạo nên một phần mạng dựa trên gói, như mạng cục bộ, mạng diện rộng, hoặc mạng toàn cầu như Internet. Phương tiện truyền thông có thể bao gồm các bộ định tuyến, bộ chuyển mạch, trạm gốc, hoặc thiết bị khác bất kỳ có thể có ích để hỗ trợ truyền thông từ thiết bị nguồn 102 đến thiết bị đích 116.

Theo một số ví dụ, phương tiện đọc được bằng máy tính 110 có thể bao gồm thiết bị lưu trữ 112. Thiết bị nguồn 102 có thể xuất ra dữ liệu được mã hóa từ giao diện xuất 108 cho thiết bị lưu trữ 112. Tương tự, thiết bị đích 116 có thể truy cập dữ liệu được mã hóa từ thiết bị lưu trữ 112 thông qua giao diện đầu vào 122. Thiết bị lưu trữ 112 có thể bao gồm phương tiện bất kỳ trong số các phương tiện lưu trữ dữ liệu truy cập phân tán

hoặc cục bộ như ổ cứng, đĩa Blu-ray, DVD, CD-ROM, bộ nhớ flash, bộ nhớ khả biến hoặc không khả biến hoặc phương tiện lưu trữ kỹ thuật số thích hợp khác bất kỳ để lưu trữ dữ liệu video được mã hóa.

Theo một số ví dụ, phương tiện đọc được bằng máy tính 110 có thể bao gồm máy chủ tập tin 114 hoặc thiết bị lưu trữ trung gian khác có thể lưu trữ dữ liệu video được mã hóa do thiết bị nguồn 102 tạo ra. Thiết bị nguồn 102 có thể xuất ra dữ liệu video được mã hóa cho máy chủ tập tin 114 hoặc một thiết bị lưu trữ trung gian khác có thể lưu trữ dữ liệu video được mã hóa do thiết bị nguồn 102 tạo ra. Thiết bị đích 116 có thể truy cập dữ liệu video đã lưu trữ từ máy chủ tập tin 114 qua truyền phát trực tiếp hoặc tải xuống. Máy chủ tập tin 114 có thể là kiểu thiết bị máy chủ bất kỳ có khả năng lưu trữ dữ liệu video được mã hóa và truyền dữ liệu video được mã hóa đến thiết bị đích 116. Máy chủ tập tin 114 có thể là máy chủ web (ví dụ, cho trang web), máy chủ giao thức truyền tập tin (File Transfer Protocol - FTP), thiết bị mạng phân phối nội dung, hoặc thiết bị lưu trữ gắn với mạng (network attached storage - NAS). Thiết bị đích 116 có thể truy cập dữ liệu video được mã hóa từ máy chủ tập tin 114 qua kết nối dữ liệu chuẩn bất kỳ, bao gồm kết nối Internet. Kết nối này có thể bao gồm kênh không dây (ví dụ, kết nối Wi-Fi), kết nối có dây (ví dụ, DSL, modem cáp, v.v.), hoặc kết hợp cả hai phù hợp với việc truy cập dữ liệu video được mã hóa lưu trữ trên máy chủ tập tin 114. Máy chủ tập tin 114 và giao diện đầu vào 122 có thể được tạo cấu hình để hoạt động theo giao thức truyền phát trực tiếp, giao thức truyền tải xuống hoặc kết hợp của chúng.

Giao diện đầu ra 108 và giao diện đầu vào 122 có thể là bộ phát/bộ thu không dây, modem, các thành phần mạng có dây (ví dụ, thẻ Ethernet), các thành phần truyền thông không dây hoạt động theo chuẩn bất kỳ trong số các chuẩn IEEE 802.11, hoặc các thành phần vật lý khác. Trong các ví dụ trong đó giao diện đầu ra 108 và giao diện đầu vào 122 bao gồm các thành phần không dây, giao diện đầu ra 108 và giao diện đầu vào 122 có thể được tạo cấu hình để truyền dữ liệu, chẳng hạn như dữ liệu video được mã hóa, theo chuẩn truyền thông dạng ô, như 4G, 4G-LTE (Long-Term Evolution: Tiến hóa dài hạn), LTE cải tiến, 5G hoặc các chuẩn tương tự. Theo một số ví dụ trong đó giao diện đầu ra 108 bao gồm bộ phát không dây, giao diện đầu ra 108 và giao diện đầu vào 122 có thể được tạo cấu hình để truyền dữ liệu, như dữ liệu video được mã hóa, theo các chuẩn không dây khác, chẳng hạn như đặc tả kỹ thuật IEEE 802.11, đặc tả kỹ thuật IEEE 802.15 (ví dụ, ZigBee™), chuẩn Bluetooth™ hoặc các chuẩn tương tự. Theo một số

ví dụ, thiết bị nguồn 102 và/hoặc thiết bị đích 116 có thể bao gồm các thiết bị có hệ thống trên chip (system-on-a-chip - SoC) tương ứng. Ví dụ, thiết bị nguồn 102 có thể bao gồm thiết bị SoC để thực hiện chức năng được quy cho bộ mã hóa video 200 và/hoặc giao diện đầu ra 108, và thiết bị đích 116 có thể bao gồm thiết bị SoC để thực hiện chức năng được quy cho bộ giải mã video 300 và/hoặc giao diện đầu vào 122.

Các kỹ thuật của sáng chế có thể được áp dụng để lập mã video nhằm hỗ trợ các ứng dụng đa phương tiện bất kỳ, như phát sóng truyền hình qua không trung, truyền hình cáp, truyền hình vệ tinh, truyền video trực tiếp qua mạng internet, như truyền phát trực tiếp thích ứng động trên HTTP (DASH), video kỹ thuật số được mã hóa trên phương tiện lưu trữ dữ liệu, giải mã video kỹ thuật số được lưu trữ trên phương tiện lưu trữ dữ liệu, hoặc các ứng dụng khác.

Giao diện đầu vào 122 của thiết bị đích 116 nhận dòng bit video được mã hóa từ phương tiện đọc được bằng máy tính 110 (ví dụ, thiết bị lưu trữ 112, máy chủ tập tin 114 hoặc tương tự). Dòng bit video được mã hóa có thể bao gồm thông tin báo hiệu được xác định bởi bộ mã hóa video 200, thông tin này cũng được bộ giải mã video 300 sử dụng, chẳng hạn như các phần tử cú pháp có các giá trị mô tả các đặc điểm và/hoặc việc xử lý các khối video hoặc các đơn vị được lập mã khác (ví dụ, lát, ảnh, nhóm ảnh, chuỗi hoặc tương tự). Thiết bị hiển thị 118 hiển thị ảnh được giải mã của dữ liệu video được giải mã cho người dùng. Thiết bị hiển thị 118 có thể là thiết bị bất kỳ trong số nhiều thiết bị hiển thị như ống tia catot (Cathode Ray Tube - CRT), màn hình tinh thể lỏng (Liquid Crystal Display - LCD), màn hình plasma, màn hình điốt phát sáng hữu cơ (Organic Light Emitting Diode - OLED), hoặc loại thiết bị hiển thị khác.

Mặc dù không được thể hiện trên Fig.1, nhưng theo một số ví dụ, mỗi bộ mã hóa video 200 và bộ giải mã video 300 có thể được tích hợp với bộ mã hóa âm thanh và/hoặc bộ giải mã âm thanh, và có thể bao gồm các bộ MUX-DEMUX thích hợp, hoặc phần cứng và/hoặc phần mềm khác, để xử lý các dòng được ghép kênh bao gồm cả âm thanh và video trong dòng dữ liệu chung. Nếu có thể, các bộ MUX-DEMUX có thể tuân theo giao thức ghép kênh ITU H.223, hoặc các giao thức khác như giao thức gói dữ liệu người dùng (user datagram protocol - UDP).

Mỗi bộ mã hóa video 200 và bộ giải mã video 300 có thể được triển khai dưới dạng bất kỳ trong số rất nhiều mạch mã hóa và/hoặc mạch giải mã thích hợp, như một hoặc nhiều bộ vi xử lý, bộ xử lý tín hiệu số (digital signal processor - DSP), mạch tích

hợp chuyên dụng (Application Specific Integrated Circuit - ASIC), mảng công lập trình được theo trường (Field Programmable Gate Array - FPGA), mạch logic rời rạc, phần mềm, phần cứng, firmware hoặc kết hợp bất kỳ của các loại trên. Khi các kỹ thuật này được triển khai một phần trong phần mềm, thiết bị có thể lưu trữ các lệnh cho phần mềm trên phương tiện bất biến đọc được bằng máy tính thích hợp và thực thi các lệnh trong phần cứng bằng cách sử dụng một hoặc nhiều bộ xử lý để thực hiện các kỹ thuật của sáng chế. Mỗi bộ mã hóa video 200 và bộ giải mã video 300 có thể được bao gồm trong một hoặc nhiều bộ mã hóa hoặc bộ giải mã, mỗi trong số chúng có thể được tích hợp như là một phần của bộ mã hóa/bộ giải mã (encoder/decoder - CODEC) kết hợp trong thiết bị tương ứng. Thiết bị bao gồm bộ mã hóa video 200 và/hoặc bộ giải mã video 300 có thể bao gồm mạch tích hợp, bộ vi xử lý và/hoặc thiết bị truyền thông không dây, như điện thoại di động.

Bộ mã hóa video 200 và bộ giải mã video 300 có thể hoạt động theo chuẩn lập mã video, chẳng hạn như ITU-T H.265, còn được gọi là lập mã video hiệu suất cao (High Efficiency Video Coding - HEVC) hoặc các phiên bản mở rộng của nó, chẳng hạn như các phiên bản mở rộng lập mã video đa cảnh nhìn và/hoặc có thể định tỷ lệ. Theo cách khác, bộ mã hóa video 200 và bộ giải mã video 300 cũng có thể hoạt động theo các chuẩn độc quyền hoặc công nghiệp khác, như mô hình thử nghiệm khảo sát chung (Joint Exploration Test Model - JEM) hoặc ITU-T H.266, còn được gọi là lập mã video đa năng (Versatile Video Coding - VVC). Dự thảo gần đây của chuẩn VVC được mô tả trong tài liệu của Bross và các cộng sự có tên là “Versatile Video Coding (Draft 4)” nhóm chuyên gia hợp tác chung về kỹ thuật video (Joint Video Experts Team - JVET) của ITU-T SG 16. WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 13 tại Marrakech, Maroc, từ ngày 9-18/1/2019, JVET-M1001-v6 (dưới đây gọi là “VVC Bản dự thảo 4”). Tuy nhiên, các kỹ thuật của sáng chế không bị giới hạn ở bất kỳ chuẩn lập mã cụ thể nào.

Nói chung, bộ mã hóa video 200 và bộ giải mã video 300 có thể thực hiện lập mã các ảnh dựa trên khối. Thuật ngữ “khối” thường đề cập đến cấu trúc bao gồm dữ liệu cần xử lý (ví dụ, mã hóa, giải mã, hoặc nếu không thì được sử dụng trong quy trình mã hóa và/hoặc giải mã). Ví dụ, khối có thể bao gồm ma trận hai chiều của mẫu dữ liệu độ chói và/hoặc sắc độ. Nói chung, bộ mã hóa video 200 và bộ giải mã video 300 có thể lập mã dữ liệu video được biểu diễn ở định dạng YUV (ví dụ, Y, Cb, Cr). Tức là, thay

vì lập mã dữ liệu màu đỏ, xanh lục và xanh dương (red, green, and blue - RGB) cho các mẫu của ảnh, bộ mã hóa video 200 và bộ giải mã video 300 có thể lập mã các thành phần độ chói và sắc độ, trong đó các thành phần sắc độ có thể bao gồm cả thành phần sắc độ màu đỏ và sắc độ màu xanh dương. Theo một số ví dụ, bộ mã hóa video 200 chuyển đổi dữ liệu định dạng RGB nhận được thành dạng biểu diễn YUV trước khi mã hóa, và bộ giải mã video 300 chuyển đổi dạng biểu diễn YUV thành định dạng RGB. Theo cách khác, các đơn vị xử lý trước và sau (không được thể hiện trên hình vẽ) có thể thực hiện các quy trình chuyển đổi này.

Nói chung sáng chế có thể đề cập đến việc lập mã (ví dụ, mã hóa và giải mã) các ảnh để bao gồm quy trình mã hóa hoặc giải mã dữ liệu của ảnh. Tương tự, sáng chế có thể đề cập đến việc lập mã các khối của ảnh để bao gồm quy trình mã hóa hoặc giải mã dữ liệu cho các khối, ví dụ, dự đoán và/hoặc lập mã dữ liệu dư. Dòng bit video được mã hóa thường bao gồm một loạt giá trị cho các phần tử cú pháp biểu diễn các quyết định lập mã (ví dụ, chế độ lập mã) và chia các ảnh thành các khối. Do đó, nói đến lập mã ảnh hoặc khối thường được hiểu là lập mã các giá trị cho các phần tử cú pháp tạo thành ảnh hoặc khối.

HEVC định nghĩa một số khối, bao gồm các đơn vị lập mã (coding unit - CU), đơn vị dự đoán (prediction unit - PU) và đơn vị biến đổi (transform unit - TU). Theo HEVC, bộ lập mã video (chẳng hạn như bộ mã hóa video 200) phân chia đơn vị cây lập mã (coding tree unit - CTU) thành các CU theo cấu trúc cây tứ phân. Tức là, bộ lập mã video phân chia các CTU và CU thành bốn ô vuông bằng nhau không chồng lấn, và mỗi nút của cây tứ phân có không hoặc bốn nút con. Các nút không có nút con có thể được gọi là “các nút lá” và các CU của các nút lá như vậy có thể bao gồm một hoặc nhiều PU và/hoặc một hoặc nhiều TU. Bộ lập mã video có thể phân chia tiếp các PU và các TU. Ví dụ, trong HEVC, các cây tứ phân dư (residual quadtree - RQT) biểu diễn sự phân chia các TU. Trong HEVC, các PU biểu diễn dữ liệu dự đoán liên ảnh, còn các TU biểu diễn dữ liệu dư. Các CU được dự đoán nội ảnh chứa thông tin dự đoán nội ảnh, như chỉ báo chế độ nội ảnh.

Theo một ví dụ khác, bộ mã hóa video 200 và bộ giải mã video 300 có thể được tạo cấu hình để hoạt động theo JEM hoặc VVC. Theo JEM hoặc VVC, bộ lập mã video (như bộ mã hóa video 200) phân chia ảnh thành nhiều đơn vị cây lập mã (CTU). Bộ mã hóa video 200 có thể phân chia CTU theo cấu trúc cây, chẳng hạn như cấu trúc cây nhị

phân-cây tứ phân (quadtree-binary tree - QTBT) hoặc cấu trúc cây nhiều dạng (Multi-Type Tree - MTT). Cấu trúc QTBT loại bỏ các khái niệm về nhiều kiểu phân chia, như phân chia giữa các CU, PU và TU của HEVC. Cấu trúc QTBT bao gồm hai mức: mức thứ nhất được chia theo kỹ thuật phân chia cây tứ phân và mức thứ hai được phân chia theo kỹ thuật phân chia cây nhị phân. Nút gốc của cấu trúc QTBT tương ứng với CTU. Các nút lá của cây nhị phân tương ứng với các đơn vị lập mã (coding unit - CU).

Trong cấu trúc phân chia cây MTT, các khối có thể được phân chia bằng cách sử dụng phân chia cây tứ phân (quadtree - QT), phân chia cây nhị phân (binary tree - BT), và một hoặc nhiều kiểu phân chia cây tam phân (triple tree - TT). Phân chia cây tam phân là phân chia trong đó một khối được phân chia thành ba khối con. Theo một số ví dụ, phân chia cây tam phân chia một khối thành ba khối con mà không chia khối gốc qua tâm. Các loại phân chia trong MTT (ví dụ, QT, BT, và TT), có thể là đối xứng hoặc bất đối xứng.

Theo một số ví dụ, bộ mã hóa video 200 và bộ giải mã video 300 có thể sử dụng một cấu trúc QTBT hoặc MTT để biểu diễn mỗi thành phần trong số các thành phần độ chói và sắc độ, trong khi theo các ví dụ khác, bộ mã hóa video 200 và bộ giải mã video 300 có thể sử dụng hai hoặc nhiều cấu trúc QTBT hoặc MTT, như một cấu trúc QTBT/MTT cho thành phần độ chói và một cấu trúc QTBT/MTT khác cho cả hai thành phần sắc độ (hoặc hai cấu trúc QTBT/MTT cho các thành phần sắc độ tương ứng).

Bộ mã hóa video 200 và bộ giải mã video 300 có thể được tạo cấu hình để sử dụng phân chia cây tứ phân cho mỗi phân chia HEVC, QTBT, phân chia MTT, hoặc các cấu trúc phân chia khác. Để phục vụ mục đích giải thích, phần mô tả về các kỹ thuật của sáng chế được trình bày liên quan đến kỹ thuật phân chia QTBT. Tuy nhiên, cần phải hiểu rằng các kỹ thuật của sáng chế này cũng có thể được áp dụng cho các bộ lập mã video được tạo cấu hình để sử dụng kỹ thuật phân chia cây tứ phân hoặc các kiểu kỹ thuật phân chia khác.

Sáng chế có thể sử dụng “NxN” và “N nhân N” theo cách thay thế cho nhau để chỉ các kích thước mẫu của một khối (chẳng hạn như CU hoặc khối video khác) xét về các chiều dọc và chiều ngang, ví dụ, 16x16 mẫu hoặc 16 nhân 16 mẫu. Nói chung, CU 16x16 sẽ có 16 mẫu theo chiều dọc ($y = 16$) và 16 mẫu theo chiều ngang ($x = 16$). Tương tự, CU NxN thường có N mẫu theo chiều dọc và N mẫu theo chiều ngang, trong đó N là giá trị nguyên không âm. Các mẫu trong CU có thể được sắp xếp thành các hàng và

các cột. Hơn nữa, các CU không nhất thiết phải có số lượng mẫu theo chiều ngang bằng số lượng mẫu theo chiều dọc. Ví dụ, các CU có thể bao gồm $N \times M$ mẫu, trong đó M không nhất thiết bằng N .

Bộ mã hóa video 200 mã hóa dữ liệu video cho các CU biểu diễn thông tin dự đoán và/hoặc thông tin dư, và thông tin khác. Thông tin dự đoán biểu thị cách thức CU sẽ được dự đoán để tạo ra khối dự đoán cho CU. Thông tin dư thường biểu diễn sự chênh lệch theo từng mẫu giữa các mẫu của CU trước khi mã hóa và khối dự đoán.

Để dự đoán đơn vị CU, bộ mã hóa video 200 có thể thường tạo ra khối dự đoán cho CU thông qua dự đoán liên ảnh hoặc dự đoán nội ảnh. Dự đoán liên ảnh thường được sử dụng để chỉ việc dự đoán CU từ dữ liệu của ảnh được lập mã trước đó, trong khi dự đoán nội ảnh thường được sử dụng để chỉ việc dự đoán CU từ dữ liệu được lập mã trước đó của cùng một ảnh. Để thực hiện dự đoán liên ảnh, bộ mã hóa video 200 có thể tạo ra khối dự đoán bằng cách sử dụng một hoặc nhiều vector chuyển động. Nhìn chung, bộ mã hóa video 200 có thể thực hiện tìm kiếm chuyển động để nhận dạng khối tham chiếu khớp nhất với CU, ví dụ, xét về sự khác biệt giữa CU và khối tham chiếu. Bộ mã hóa video 200 có thể tính toán số đo chênh lệch bằng cách sử dụng tổng chênh lệch tuyệt đối (sum of absolute difference - SAD), tổng chênh lệch bình phương (sum of squared differences - SSD), chênh lệch tuyệt đối trung bình (mean absolute difference - MAD), chênh lệch bình phương trung bình (mean squared differences - MSD) hoặc các phép tính chênh lệch như vậy khác để xác định xem khối tham chiếu có khớp nhất với CU hiện thời không. Trong một số ví dụ, bộ mã hóa video 200 có thể dự đoán CU hiện thời bằng cách sử dụng kỹ thuật dự đoán một hướng hoặc dự đoán hai hướng.

Một số ví dụ của JEM và VVC còn cung cấp chế độ bù chuyển động afin, chế độ này có thể được xem là chế độ dự đoán liên ảnh. Trong chế độ bù chuyển động afin, bộ mã hóa video 200 có thể xác định hai hoặc nhiều vector chuyển động biểu diễn chuyển động không tịnh tiến, chẳng hạn như phóng to hoặc thu nhỏ, xoay, chuyển động phối cảnh, hoặc các kiểu chuyển động bất quy tắc khác.

Để thực hiện dự đoán nội ảnh, bộ mã hóa video 200 có thể lựa chọn chế độ dự đoán nội ảnh để tạo ra khối dự đoán. Một số ví dụ của JEM và VVC cung cấp sáu mươi bảy chế độ dự đoán nội ảnh, bao gồm các chế độ định hướng khác nhau, cũng như chế độ phẳng và chế độ DC. Nói chung, bộ mã hóa video 200 chọn chế độ dự đoán nội ảnh để mô tả các mẫu lân cận cho khối hiện thời (ví dụ, khối của CU) để từ đó dự đoán các

mẫu của khối hiện thời. Các mẫu như vậy có thể thường ở trên, bên trên và bên trái hoặc bên trái của khối hiện thời trong cùng ảnh giống như khối hiện thời, giả sử bộ mã hóa video 200 lập mã các CTU và các CU theo thứ tự quét mảnh (từ trái sang phải, từ trên xuống dưới).

Bộ mã hóa video 200 mã hóa dữ liệu biểu diễn chế độ dự đoán cho khối hiện thời. Ví dụ, đối với các chế độ dự đoán liên ảnh, bộ mã hóa video 200 có thể mã hóa dữ liệu biểu diễn chế độ dự đoán liên ảnh nào trong số các chế độ dự đoán liên ảnh có sẵn, cũng như thông tin chuyển động cho chế độ tương ứng. Ví dụ, đối với dự đoán liên ảnh một chiều hoặc hai chiều, bộ mã hóa video 200 có thể mã hóa các vector chuyển động bằng cách sử dụng chế độ dự đoán vector chuyển động nâng cao (advanced motion vector prediction - AMVP) hoặc chế độ hợp nhất. Bộ mã hóa video 200 có thể sử dụng các chế độ tương tự để mã hóa các vector chuyển động cho chế độ bù chuyển động afin.

Sau dự đoán, chẳng hạn như dự đoán nội ảnh hoặc dự đoán liên ảnh các khối, bộ mã hóa video 200 có thể tính toán dữ liệu dư cho khối. Dữ liệu dư, chẳng hạn như khối dư, biểu diễn các chênh lệch mẫu-mẫu giữa khối này và khối dự đoán cho khối này, được tạo ra bằng cách sử dụng chế độ dự đoán tương ứng. Bộ mã hóa video 200 có thể áp dụng một hoặc nhiều phép biến đổi cho khối dư, để tạo ra dữ liệu được biến đổi trong miền biến đổi thay vì miền mẫu. Ví dụ, bộ mã hóa video 200 có thể áp dụng phép biến đổi cosin rời rạc (discrete cosine transform - DCT), biến đổi số nguyên, biến đổi sóng con, hoặc biến đổi tương tự về mặt khái niệm cho dữ liệu video dư. Ngoài ra, bộ mã hóa video 200 có thể áp dụng phép biến đổi thứ cấp sau phép biến đổi thứ nhất, chẳng hạn như biến đổi thứ cấp không tách biệt phụ thuộc vào chế độ (mode-dependent non-separable secondary transform - MDNSST), biến đổi phụ thuộc vào tín hiệu, biến đổi Karhunen-Loeve (Karhunen-Loeve transform - KLT) hoặc tương tự. Bộ mã hóa video 200 tạo ra các hệ số biến đổi sau khi áp dụng một hoặc nhiều phép biến đổi.

Như nêu ở trên, sau phép biến đổi bất kỳ để tạo ra các hệ số biến đổi, bộ mã hóa video 200 có thể thực hiện lượng tử hóa các hệ số biến đổi. Nói chung, lượng tử hóa chỉ quy trình trong đó các hệ số biến đổi được lượng tử hóa để có thể giảm lượng dữ liệu dùng để biểu diễn các hệ số, cho phép nén thêm. Bằng cách thực hiện quy trình lượng tử hóa, bộ mã hóa dữ liệu video 200 có thể giảm độ sâu bit liên quan đến một số hoặc tất cả các hệ số. Ví dụ, bộ mã hóa video 200 có thể làm tròn giá trị n -bit xuống đến giá trị m -bit trong quá trình lượng tử hóa, trong đó n lớn hơn m . Theo một số ví dụ, để thực

hiện lượng tử hóa, bộ mã hóa video 200 có thể thực hiện dịch phải theo bit đối với giá trị cần được lượng tử hóa.

Sau khi lượng tử hóa, bộ mã hóa video 200 có thể quét các hệ số biến đổi, tạo ra vector một chiều từ ma trận hai chiều bao gồm các hệ số biến đổi được lượng tử hóa. Quy trình quét được thiết kế để đặt các hệ số năng lượng cao hơn (và do đó tần số thấp hơn) ở phía trước vector và để đặt các hệ số biến đổi năng lượng thấp hơn (và do đó tần số cao hơn) ở phía sau vector. Theo một số ví dụ, bộ mã hóa video 200 có thể sử dụng thứ tự quét định trước để quét các hệ số biến đổi được lượng tử hóa để tạo ra vector nối tiếp, và sau đó mã hóa entropy các hệ số biến đổi được lượng tử hóa của vector. Trong các ví dụ khác, bộ mã hóa video 200 có thể thực hiện quy trình quét thích ứng. Sau khi quét các hệ số biến đổi được lượng tử hóa để tạo vector một chiều, bộ mã hóa video 200 có thể mã hóa entropy vector một chiều, ví dụ, theo kỹ thuật lập mã số học nhị phân thích ứng ngữ cảnh (context-adaptive binary arithmetic coding - CABAC). Bộ mã hóa video 200 cũng có thể mã hóa entropy các giá trị cho các phần tử cú pháp mô tả siêu dữ liệu liên quan đến dữ liệu video được mã hóa để bộ giải mã video 300 sử dụng để giải mã dữ liệu video.

Để thực hiện CABAC, bộ mã hóa video 200 có thể gán ngữ cảnh trong mô hình ngữ cảnh cho ký hiệu cần được truyền. Ngữ cảnh có thể liên quan tới, ví dụ, việc các giá trị lân cận của ký hiệu có giá trị bằng không hay không. Việc xác định xác suất có thể được dựa vào ngữ cảnh được gán cho ký hiệu.

Bộ mã hóa video 200 còn có thể tạo ra dữ liệu cú pháp, như dữ liệu cú pháp dựa trên khối, dữ liệu cú pháp dựa trên ảnh, và dữ liệu cú pháp dựa trên chuỗi, cho bộ giải mã video 300, ví dụ, trong phần đầu ảnh, phần đầu khối, phần đầu lát, hoặc dữ liệu cú pháp khác, như tập hợp thông số chuỗi (sequence parameter set - SPS), tập hợp thông số ảnh (picture parameter set - PPS), hoặc tập hợp thông số video (video parameter set - VPS). Tương tự, bộ giải mã video 300 có thể giải mã dữ liệu cú pháp như vậy để xác định cách giải mã dữ liệu video tương ứng.

Theo cách này, bộ mã hóa video 200 có thể tạo ra dòng bit bao gồm dữ liệu video được mã hóa, ví dụ, các phần tử cú pháp mô tả việc phân chia ảnh thành các khối (ví dụ, các CU) và thông tin dự đoán và/hoặc thông tin dự cho các khối. Cuối cùng, bộ giải mã video 300 có thể nhận dòng bit và giải mã dữ liệu video được mã hóa.

Phần sau đây sẽ mô tả chi tiết hơn về các kỹ thuật lập mã số học nhị phân (binary arithmetic coding - BAC) và CABAC. Nói chung, BAC là quy trình chia nhỏ theo khoảng đệ quy. BAC được sử dụng để mã hóa các bin trong quy trình CABAC theo các chuẩn lập mã video H.264/AVC và H.265/HEVC. Đầu ra của bộ lập mã BAC là dòng nhị phân biểu diễn giá trị hoặc con trỏ đến xác suất trong khoảng xác suất được lập mã cuối cùng. Khoảng xác suất được định rõ bằng phạm vi ('range') và giá trị đầu dưới ('giá trị thấp'). Phạm vi là phần mở rộng của khoảng xác suất. Giá trị đầu dưới là giới hạn dưới của khoảng xác suất/lập mã.

Việc áp dụng lập mã số học để lập mã video được mô tả trong tài liệu của D. Marpe, H. Schwarz, và T. Wiegand "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard" của IEEE Trans. Circuits and Systems for Video Technology, tập 13, số 7, tháng 7/2003. CABAC gồm ba chức năng chính là nhị phân hóa, mô hình hóa ngữ cảnh và lập mã số học. Nhị phân hóa là chức năng ánh xạ các phân tử cú pháp thành các ký hiệu nhị phân (hoặc "các bin"). Các ký hiệu nhị phân có thể còn được gọi là "các chuỗi bin". Mô hình hóa ngữ cảnh là chức năng ước lượng xác suất của các bin khác nhau. Lập mã số học là chức năng tiếp theo để nén các bin thành bit, dựa vào xác suất ước lượng được. Các thiết bị và/hoặc module khác nhau của nó, chẳng hạn như bộ lập mã số học nhị phân, có thể thực hiện chức năng lập mã số học.

Các quy trình nhị phân hóa khác nhau được sử dụng trong HEVC, bao gồm đơn phân (unary - U), đơn phân cắt bớt (truncated unary - TU), Exp-Golomb bậc k (kth-order Exp-Golomb - EGk), và độ dài cố định (fixed length - FL). Chi tiết về các quy trình nhị phân hóa này được mô tả trong tài liệu của V. Sze và M. Budagavi, "High throughput CABAC coding entropy in HEVC," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), tập 22, số 12, trang 1778–1791, tháng 12/2012.

Mỗi ngữ cảnh (tức là, mô hình xác suất) trong CABAC được biểu diễn bởi một trạng thái. Mỗi trạng thái (σ) ngầm biểu diễn xác suất (p_σ) của một ký hiệu cụ thể (ví dụ, bin) là ký hiệu có xác suất thấp nhất (Least Probable Symbol - LPS). Một ký hiệu có thể là LPS hoặc ký hiệu có xác suất cao nhất (Most Probable Symbol - MPS). Các ký hiệu thuộc nhị phân, và do đó, MPS và LPS có thể là 0 hoặc 1. Xác suất được ước lượng cho

ngữ cảnh tương ứng và được sử dụng (ngầm) để lập mã entropy cho ký hiệu bằng cách sử dụng bộ lập mã số học.

Quy trình BAC được thực hiện bởi máy trạng thái để làm thay đổi các giá trị bên trong của nó là ‘phạm vi’ và ‘giá trị thấp’ tùy theo ngữ cảnh để lập mã và giá trị của bin đang được lập mã. Tùy theo trạng thái của ngữ cảnh (tức là xác suất của nó), phạm vi được phân chia thành $rangeMPS_{\sigma}$ (phạm vi của ký hiệu có xác suất cao nhất ở trạng thái $_{\sigma}$) và $rangeLPS_{\sigma}$ (phạm vi của ký hiệu có xác suất thấp nhất ở trạng thái $_{\sigma}$). Về lý thuyết, giá trị $rangeLPS_{\sigma}$ của trạng thái xác suất $_{\sigma}$ được suy ra từ phép nhân:

$$rangeLPS_{\sigma} = range \times p_{\sigma},$$

trong đó p_{σ} là xác suất lựa chọn LPS. Tất nhiên là xác suất của MPS là $1-p_{\sigma}$. Tương đương như thế, $rangeMPS_{\sigma}$ bằng $range$ trừ $rangeLPS_{\sigma}$. BAC lập lại cập nhật phạm vi tùy theo trạng thái của ngữ cảnh bin để lập mã, phạm vi hiện thời, và giá trị của bin đang được lập mã (tức là, bin bằng LPS hoặc MPS).

Các hình vẽ Fig.2A và Fig.2B thể hiện các ví dụ của quy trình này tại bin n . Trong ví dụ 201 trên Fig.2A, phạm vi tại bin n bao gồm RangeMPS và RangeLPS được xác định bởi xác suất của LPS (p_{σ}) dựa vào trạng thái ngữ cảnh nhất định (σ). Ví dụ 201 thể hiện việc cập nhật phạm vi tại bin $n+1$ khi giá trị bin n bằng MPS. Trong ví dụ này, giá trị thấp vẫn giữ nguyên, nhưng giá trị của phạm vi tại bin $n+1$ được giảm xuống bằng giá trị RangeMPS tại bin n . Ví dụ 203 trên Fig.2B thể hiện việc cập nhật phạm vi tại bin $n+1$ khi giá trị bin n không bằng MPS (tức là bằng LPS). Trong ví dụ này, giá trị thấp được chuyển đến giá trị phạm vi thấp hơn RangeLPS tại bin n . Ngoài ra, giá trị của phạm vi tại bin $n+1$ được giảm xuống bằng giá trị RangeLPS tại bin n .

Theo HEVC, phạm vi được biểu diễn bằng 9 bit và giá trị thấp có 10 bit. Có một quy trình chuẩn hóa lại để duy trì phạm vi và các giá trị thấp có độ chính xác đủ cao. Quy trình chuẩn hóa lại diễn ra mỗi khi phạm vi nhỏ hơn 256. Do đó, phạm vi luôn bằng hoặc lớn hơn 256 sau khi chuẩn hóa lại. Tùy thuộc vào các giá trị của phạm vi và giá trị thấp, BAC xuất ra dòng bit, '0' hoặc '1' hoặc cập nhật biến bên trong (gọi là BO: các bit nổi bật) để giữ các giá trị đầu ra tương lai. Fig.3 thể hiện các ví dụ về đầu ra BAC tùy thuộc vào phạm vi. Ví dụ, '1' được xuất ra dòng bit khi phạm vi và giá trị thấp cao hơn ngưỡng nhất định (ví dụ, 512). '0' được xuất ra dòng bit khi phạm vi và giá trị thấp thấp hơn ngưỡng nhất định (ví dụ, 512). Không có gì được xuất ra dòng bit khi phạm vi và

giá trị thấp hơn nằm giữa một số ngưỡng nhất định. Thay vào đó, giá trị BO được tăng lên, và bin tiếp theo được mã hóa.

Các kỹ thuật được mô tả trong sáng chế có thể được thực hiện, ví dụ, trong bộ mã hóa video, bộ giải mã video, hoặc bộ mã hóa-giải mã video kết hợp (CODEC). Cụ thể là, các kỹ thuật này có thể được thực hiện trong đơn vị mã hóa entropy của bộ mã hóa video và/hoặc đơn vị giải mã entropy của bộ giải mã video. Các kỹ thuật có thể được thực hiện, ví dụ, trong quy trình CABAC, có thể được tạo cấu hình để hỗ trợ lập mã video, chẳng hạn như lập mã video theo các khía cạnh của chuẩn HEVC. Đơn vị mã hóa và giải mã entropy có thể áp dụng các quy trình lập mã theo cách xuôi hoặc ngược, ví dụ, để mã hóa hoặc giải mã các dữ liệu video bất kỳ, chẳng hạn như các hệ số biến đổi được lượng tử hóa liên quan đến dữ liệu video dư, thông tin vector chuyển động, các phần tử cú pháp, và các loại thông tin khác có thể có ích trong quy trình mã hóa và/hoặc giải mã video.

Nói chung, bộ giải mã video 300 thực hiện quy trình thuận nghịch với quy trình được thực hiện bởi bộ mã hóa video 200 để giải mã dữ liệu video đã mã hóa của dòng bit. Ví dụ, bộ giải mã video 300 có thể giải mã các giá trị cho các phần tử cú pháp của dòng bit bằng cách sử dụng kỹ thuật CABAC theo cách về cơ bản tương tự như, mặc dù thuận nghịch với, quy trình mã hóa CABAC của bộ mã hóa video 200. Các phần tử cú pháp có thể xác định thông tin phân chia ảnh thành các CTU, và phân chia mỗi CTU theo cấu trúc phân chia tương ứng, chẳng hạn như cấu trúc QTBT, để xác định các CU của CTU. Các phần tử cú pháp còn có thể xác định thông tin dự đoán và thông tin dư cho các khối (ví dụ, các CU) của dữ liệu video.

Thông tin dư có thể được biểu diễn bằng, ví dụ, các hệ số biến đổi được lượng tử hóa. Bộ giải mã video 300 có thể lượng tử hóa ngược và biến đổi ngược các hệ số biến đổi được lượng tử hóa của khối để tái tạo khối dư cho khối. Bộ giải mã video 300 sử dụng chế độ dự đoán được báo hiệu (dự đoán nội ảnh hoặc dự đoán liên ảnh) và thông tin dự đoán liên quan (ví dụ, thông tin chuyển động của dự đoán liên ảnh) để tạo ra khối dự đoán cho khối. Sau đó, bộ giải mã video 300 có thể kết hợp khối dự đoán và khối dư (trên cơ sở từng mẫu) để tái tạo khối gốc. Bộ giải mã video 300 có thể thực hiện xử lý bổ sung, chẳng hạn như thực hiện quy trình tách khối để giảm các xảo ảnh trực quan dọc theo các biên của khối.

Theo các kỹ thuật của sáng chế, bộ mã hóa video 200 và/hoặc bộ giải mã video 300 có thể thu được chỉ số độ dốc định trước (ví dụ, SlopeIdx) và chỉ số độ lệch định trước (ví dụ, OffsetIdx) cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp trong lát dữ liệu video; xác định, dựa vào chỉ số độ dốc định trước và chỉ số độ lệch định trước, trạng thái xác suất ban đầu của ngữ cảnh cho lát dữ liệu video trong miền tuyến tính; và mã hóa entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Nói chung, sáng chế có thể liên quan đến việc "báo hiệu" thông tin nhất định như các phần tử cú pháp. Nói chung, thuật ngữ "báo hiệu" có thể chỉ sự truyền thông các giá trị cho phần tử cú pháp và/hoặc dữ liệu khác được sử dụng để giải mã dữ liệu video được mã hóa. Tức là, bộ mã hóa video 200 có thể báo hiệu các giá trị cho các phần tử cú pháp trong dòng bit. Nói chung, việc báo hiệu là tạo ra giá trị trong dòng bit. Như đã lưu ý ở trên, thiết bị nguồn 102 có thể truyền dòng bit đến thiết bị đích 116 về cơ bản theo thời gian thực, hoặc không theo thời gian thực, chẳng hạn có thể xảy ra khi lưu trữ các phần tử cú pháp vào thiết bị lưu trữ 112 để thiết bị đích 116 truy hồi sau.

Các kỹ thuật của sáng chế thích hợp cho các bộ lập mã số học nhị phân trong đó trạng thái xác suất biểu diễn xác suất thực trong miền tuyến tính, chẳng hạn như bộ lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC) được ứng dụng trong chuẩn lập mã video đa năng (VVC).

Trong lập mã số học nhị phân, chuỗi video trước tiên được chuyển đổi thành các phần tử dữ liệu (hoặc *các phần tử cú pháp*) với dữ liệu dư theo không gian-thời gian được loại bỏ; các phần tử cú pháp sau đó được chuyển đổi không hao hụt thành các dạng biểu diễn nhị phân (hoặc *các dòng bit*) bằng kỹ thuật lập mã entropy. Các chuẩn lập mã video gần đây (ví dụ, HEVC và VVC) có thể sử dụng lập mã số học nhị phân để thực hiện lập mã entropy.

Quy trình lập mã số học nhị phân có ba giai đoạn chính là nhị phân hóa, ước lượng xác suất thích ứng và lập mã số học. Trong giai đoạn nhị phân hóa, mỗi phần tử cú pháp không nhị phân cần được lập mã được chuyển đổi thành chuỗi các ký hiệu dữ liệu nhị phân (hoặc *các bin*).

Trong giai đoạn ước lượng xác suất thích ứng, bất kể là phần tử cú pháp nhị phân hay một phần tử trong chuỗi nhị phân được chuyển đổi từ phần tử cú pháp không nhị

phân, thì mỗi bin đều có ước lượng phân phối xác suất của nó (tức là, xác suất bằng 0 hoặc 1). Các phân phối có thể được chia thành hai loại: (1) phân phối dừng và đều (tức là, luôn luôn $p = 0,5$) và (2) phân phối thay đổi theo thời gian hoặc không đều. Bin có phân phối loại (2) có thể được gán theo dõi mô hình xác suất (hoặc *mô hình ngữ cảnh*) mà phân phối theo thời gian thực của bin dựa vào các giá trị bin trước của nó và các số liệu thống kê ngữ cảnh khác.

Trong giai đoạn lập mã số học, bin có phân phối loại (1) có thể được lập mã trong chế độ bỏ qua, chế độ có thể song song hóa ở mức cao và có độ phức tạp thấp; bin có phân phối loại (2) được lập mã trong chế độ thông thường, trong đó giá trị bin và xác suất của nó được ước lượng bằng mô hình ngữ cảnh liên quan được sử dụng.

Khi được sử dụng trong các bộ lập mã số học nhị phân để lập mã video, xác suất, về lý thuyết là có giá trị thực và nằm trong khoảng từ 0 đến 1, được số hóa, và do đó thường được gọi là trạng thái xác suất. Ví dụ, theo HEVC, xác suất có độ chính xác 7 bit, tương ứng với 128 trạng thái xác suất. Fig.9 thể hiện ánh xạ giữa xác suất và trạng thái xác suất. Như có thể thấy, trạng thái xác suất theo HEVC biểu diễn xác suất thực trong miền lôgarit. Trong ví dụ khác, theo VVC, ước lượng xác suất của bin cụ thể là giá trị trung bình của hai xác suất được theo dõi trong mô hình ngữ cảnh liên quan của bin và được cập nhật lần lượt ở tốc độ cao và tốc độ thấp. Xác suất được cập nhật ở tốc độ cao có độ chính xác 10 bit, tương ứng với 1024 trạng thái xác suất; xác suất khác được cập nhật ở tốc độ thấp có độ chính xác 14 bit, tương ứng với 16384 trạng thái xác suất. Khác với HEVC, VVC chấp nhận ánh xạ tuyến tính giữa trạng thái xác suất và xác suất (xem Fig.10A và Fig.10B).

Theo HEVC và VVC, dòng bit video bao gồm hoặc gồm có nhiều đơn vị có thể giải mã độc lập (ví dụ, lát), suy ra rằng ở đầu đơn vị này, các trạng thái xác suất của tất cả các mô hình ngữ cảnh phải được thiết lập lại thành một số giá trị định trước. Thông thường, trước đây không có hiểu biết kỹ thuật nào về bản chất thống kê của nguồn, mỗi mô hình ngữ cảnh phải giả định có phân phối đều ($p = 0,5$). Tuy nhiên, để kết nối các giai đoạn hiểu biết về ước lượng xác suất thích ứng và để có sự thích ứng ban đầu ở các điều kiện lập mã khác nhau, nghiên cứu cho thấy sẽ có lợi khi cung cấp một số trạng thái xác suất ban đầu (*quy trình khởi tạo*) có tính phù hợp hơn trạng thái có xác suất ngang nhau cho mỗi mô hình xác suất.

CABAC trong HEVC có quy trình khởi tạo phụ thuộc vào tham số lượng tử hóa (quantization parameter - QP) được yêu cầu ở đầu mỗi lát. Biết giá trị ban đầu của QP độ chói cho lát, SliceQP_Y , trạng thái xác suất ban đầu của mô hình ngữ cảnh nhất định, được biểu thị là InitProbState , được tạo ra bởi các phương trình (1) đến (3),

$$m = \text{SlopeIdx} * 5 - 45 \quad (1)$$

$$n = (\text{OffsetIdx} \ll 3) - 16 \quad (2)$$

$$\text{InitProbState} = \text{Clip3}(1, 127, ((m * \text{SliceQP}_Y) \gg 4) + n) \quad (3)$$

trong đó SlopeIdx và OffsetIdx (cả hai số nguyên nằm trong khoảng từ 0 đến 15 tính cả hai giá trị đầu mút) là các tham số khởi tạo được định trước và được lưu trữ cho mỗi mô hình ngữ cảnh. Phương trình (3) có nghĩa là InitProbState được mô hình hóa bởi hàm tuyến tính của SliceQP_Y , có độ dốc xấp xỉ $m \gg 4$ và đường giao nhau n ở $\text{SliceQP}_Y = 0$. Ánh xạ từ SlopeIdx đến độ dốc và từ OffsetIdx đến đường giao nhau có thể thấy lần lượt trong Bảng 1 và 2.

Bảng 1 - Ánh xạ từ SlopeIdx đến độ dốc

SlopeIdx	0	1	2	3	4	5	6	7
độ dốc	-2,81	-2,50	-2,19	-1,88	-1,56	-1,25	-0,94	-0,63
SlopeIdx	8	9	10	11	12	13	14	15
độ dốc	-0,31	0	0,31	0,63	0,94	1,25	1,56	1,88

Bảng 2 - Ánh xạ từ OffsetIdx đến đường giao nhau

OffsetIdx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
đường giao nhau	- 16	- 8	0	8	16	24	32	40	48	56	64	72	80	88	96	104

Nói cách khác, mô hình ngữ cảnh có thể không lưu trữ trạng thái xác suất ban đầu trực tiếp; thay vào đó, nó lưu trữ hai tham số khởi tạo cùng xác định hàm tuyến tính mà, ở đầu mỗi lát, sử dụng SliceQP_Y làm đối số (argument) để suy ra trạng thái xác suất.

SlopeIdx và OffsetIdx , cả hai có độ chính xác 4 bit, được đóng gói thành một giá trị khởi tạo 8 bit, trong đó 4 bit cao và thấp lần lượt là SlopeIdx và OffsetIdx .

CABAC của VVC hiện thời sử dụng phương thức về cơ bản là tương tự để suy ra InitProbState theo chuẩn HEVC, ngoại trừ trong phương trình (3) phần cắt là từ 0 đến 127. Tuy nhiên, sau khi suy ra InitProbState, biểu diễn xác suất trong miền lôgarit (xem Fig.9), quy trình khởi tạo theo VVC cần thêm một bước để chuyển đổi InitProbState thành trạng thái xác suất biểu diễn xác suất trong miền tuyến tính (xem các hình vẽ Fig.10A và Fig.10B), để được sử dụng trong máy lập mã số học của VVC. Việc chuyển đổi (hoặc ánh xạ), được thể hiện trên Fig.11A và Fig.11B, được thực hiện bằng cách sử dụng bảng tìm kiếm (LUT) (xem Bảng 3), dưới đây:

1. Sử dụng InitProbState làm chỉ số tìm kiếm để tìm giá trị probability_state tương ứng trong LUT.

2. - Trạng thái xác suất có độ chính xác thấp hơn (10-bit), được biểu thị là ProbabilityStateL, được suy ra trong phương trình (4).

$$\text{ProbabilityStateL} = \text{probability_state} \gg 5 \quad (4)$$

- Trạng thái xác suất có độ chính xác cao hơn (14-bit), được biểu thị là ProbabilityStateH, được suy ra trong phương trình (5).

$$\text{ProbabilityStateH} = \text{probability_state} \gg 1 \quad (5)$$

Bảng 3 - Ánh xạ từ InitProbState đến probability_state

InitProbState	0	1	2	3	4	5	6	7
probability_state	614	647	681	718	756	797	839	884
InitProbState	8	9	10	11	12	13	14	15
probability_state	932	982	1034	1089	1148	1209	1274	1342
InitProbState	16	17	18	19	20	21	22	23
probability_state	1414	1490	1569	1653	1742	1835	1933	2037
InitProbState	24	25	26	27	28	29	30	31
probability_state	2146	2261	2382	2509	2643	2785	2934	3091
InitProbState	32	33	34	35	36	37	38	39
probability_state	3256	3430	3614	3807	4011	4225	4452	4690
InitProbState	40	41	42	43	44	45	46	47
probability_state	4941	5205	5483	5777	6086	6412	6755	7116

InitProbState	48	49	50	51	52	53	54	55
probability_state	7497	7898	8320	8766	9235	9729	10249	10798
InitProbState	56	57	58	59	60	61	62	63
probability_state	11375	11984	12625	13300	14012	14762	15551	16384
InitProbState	64	65	66	67	68	69	70	71
probability_state	16384	17216	18005	18755	19467	20142	20783	21392
InitProbState	72	73	74	75	76	77	78	79
probability_state	21969	22518	23038	23532	24001	24447	24869	25270
InitProbState	80	81	82	83	84	85	86	87
probability_state	25651	26012	26355	26681	26990	27284	27562	27826
InitProbState	88	89	90	91	92	93	94	95
probability_state	28077	28315	28542	28756	28960	29153	29337	29511
InitProbState	96	97	98	99	100	101	102	103
probability_state	29676	29833	29982	30124	30258	30385	30506	30621
InitProbState	104	105	106	107	108	109	110	111
probability_state	30730	30834	30932	31025	31114	31198	31277	31353
InitProbState	112	113	114	115	116	117	118	119
probability_state	31425	31493	31558	31619	31678	31733	31785	31835
InitProbState	120	121	122	123	124	125	126	127
probability_state	31883	31928	31970	32011	32049	32086	32120	32153

Như đã nêu trên, CABAC của VVC hiện thời chấp nhận các trạng thái xác suất biểu diễn các xác suất trong miền tuyến tính (xem Fig.10A và Fig.10B), nhưng vẫn sử dụng quy trình khởi tạo kế thừa của HEVC, các trạng thái xác suất ban đầu được suy ra biểu diễn các xác suất trong miền lôgarit. Để kết nối đầu ra của quy trình khởi tạo và đầu vào của máy lập mã số học, LUT 128 mục nhập được sử dụng để ánh xạ.

Theo một hoặc nhiều kỹ thuật của sáng chế, quy trình khởi tạo có thể được biến đổi để dùng cho VVC (ví dụ, các kỹ thuật này có thể cải thiện quy trình tìm kiếm theo bảng này bằng cách biến đổi thích hợp quy trình khởi tạo trong VVC). Sáng chế đề xuất hai phương pháp biến đổi. Trong một phương pháp, trạng thái xác suất đầu ra InitProbState biểu diễn xác suất trong miền tuyến tính. Bằng cách này, quy trình ánh xạ

được giảm xuống thành các phép dịch, chỉ nhằm mục đích điều chỉnh độ chính xác. Trong phương pháp còn lại, trạng thái xác suất đầu ra `InitProbState` biểu diễn xác suất trong miền bậc hai, và ánh xạ được nhận ra bằng cách sử dụng các phương trình, chứ không phải LUT.

Tăng phạm vi đường gốc và đường giao nhau. Để tìm ra các tham số khởi tạo thích hợp cho mô hình ngữ cảnh cho trước, cần phải hiểu được các kỹ thuật đã biết về phân phối xác suất của các bin sử dụng mô hình ngữ cảnh đó, bằng nhiều cách, chẳng hạn như phân tích thống kê nguồn. Fig.12 thể hiện ví dụ về bin của phần tử cú pháp `SaoMergeFlag` được sử dụng trong lát I.

Theo HEVC và VVC hiện nay, các kỹ thuật đã biết về xác suất được chuyển đổi thành trạng thái xác suất trong miền lôgarit (Fig.13A được chuyển đổi từ Fig.12), và các tham số khởi tạo tối ưu, mà qua đó hàm tuyến tính được xác định phù hợp nhất với sự tương quan giữa trạng thái xác suất và `SliceQPY`, được xác định qua thuật toán tìm kiếm brute-force hoặc các phương thức khác thông minh hơn. Theo ví dụ tương tự, `SlopeIdx` và `OffsetIdx` lần lượt là 13 và 6 (độ dốc xấp xỉ 1,25 và đường giao nhau 32), và hàm tuyến tính tương ứng phù hợp nhất được biểu diễn bằng đồ thị trên Fig.13B.

Theo các kỹ thuật của sáng chế, trạng thái xác suất ban đầu nằm trong miền tuyến tính, cần được làm hài hòa với máy lập mã số học. Trong ví dụ tương tự, các kỹ thuật đã biết được biểu diễn bởi trạng thái xác suất trong miền tuyến tính được thể hiện trên Fig.14A, đây đơn giản chỉ là phiên bản có tỷ lệ khác của Fig.12. Lưu ý rằng, để đơn giản và rõ ràng thì độ chính xác 7 bit được sử dụng cho cả trạng thái xác suất lôgarit và tuyến tính. Tuy nhiên, trong các ví dụ khác, các độ chính xác bit khác có thể được sử dụng.

Sự so sánh Fig.13A và Fig.14A thể hiện rằng trạng thái xác suất thay đổi ít hơn nhiều theo `SliceQPY` trong miền lôgarit so với trong miền tuyến tính, điều này có nghĩa là đường thẳng khớp với các trạng thái xác suất trong miền lôgarit có thể trở nên khá là dốc trong miền tuyến tính. Về trạng thái xác suất được thể hiện trên Fig.14A, không có sự thay đổi quy trình khởi tạo được xác định trong HEVC và VVC hiện thời (xem các phương trình (1) đến (3)), các tham số khởi tạo tốt nhất có thể tìm thấy bằng thuật toán tìm kiếm brute-force là `SlopeIdx` bằng 15 và `OffsetIdx` bằng 4, mà qua đó đạt được độ dốc có giá trị lớn nhất là 1,88, như được thể hiện trên Fig.14B. Tuy nhiên, đường này có thể vẫn chưa đủ dốc để đạt được sự thích hợp tốt. Khi phạm vi tìm kiếm độ dốc và chỗ giao nhau tăng lên, có thể tìm được sự phù hợp tốt hơn, như được thể hiện trên

Fig.14C, trong đó độ dốc là 3,25 và chỗ giao nhau là -23, cả hai đều vượt qua các giới hạn mà HEVC và VVC hiện thời cho phép.

Lưu ý rằng ví dụ (các hình vẽ Fig.12 đến Fig.14C) đều có tính tiêu chuẩn chứ không phải khác thường – số lượng bin ổn có kiểu phân phối xác suất như vậy. Do đó, phạm vi của độ dốc và độ lệch cần tăng lên để sử dụng mô hình tuyến tính để phù hợp với sự tương quan của SliceQP_Y và trạng thái xác suất đang nằm trong miền tuyến tính.

Trong một ví dụ, số lượng các giá trị độ dốc và chỗ giao nhau có thể có thì không thay đổi (tức là, SlopeIdx và OffsetIdx vẫn nằm trong khoảng từ 0 đến 15), nhưng khoảng cách của các bước giữa chúng tăng lên, có nghĩa là hằng số a trong phương trình (6) lớn hơn 5 trong phương trình (1) và b trong phương trình (7) lớn hơn 3 trong phương trình 2.

$$m = \text{SlopeIdx} * a - c \quad (6)$$

$$n = (\text{OffsetIdx} \ll b) - d \quad (7)$$

Trong ví dụ khác, số lượng các giá trị độ dốc và chỗ giao nhau có thể có tăng lên. Ví dụ, SlopeIdx có thể nằm trong khoảng từ 0 đến 31, tạo ra 32 giá trị độ dốc có thể có. Hai ví dụ trên đây có thể cùng được sử dụng, hoặc có thể được sử dụng độc lập.

Xác định chỗ giao nhau tại SliceQP_Y không bằng 0. Bên cạnh việc cung cấp trạng thái xác suất tuyến tính có phạm vi của độ dốc và độ lệch tăng, việc xác định chỗ giao nhau tại SliceQP_Y không bằng 0 là một phương thức khác hoặc phương thức bổ sung. Ví dụ, Fig.15A thể hiện sự phân phối xác suất của SplitFlag được sử dụng trong lát B. Trong miền lôgarit, như được thể hiện trên Fig.15B, trạng thái xác suất theo SliceQP_Y có thể được mô hình hóa tốt bằng hàm tuyến tính có độ dốc bằng -0,63 và chỗ giao nhau bằng 88 (tức là, SlopeIdx 7 và OffsetIdx 13 như được xác định trong HEVC và VVC hiện thời). Tuy nhiên, trong miền tuyến tính, trạng thái xác suất không thể được mô hình hóa tốt bằng sự kết hợp đường dốc/ đường giao nhau có giá trị được xác định trong HEVC và VVC hiện thời. Fig.15C thể hiện kết quả thích hợp nhất mà thuật toán tìm kiếm brute-force tìm ra, trong đó độ dốc bằng -0,94 và chỗ giao nhau bằng 104 (tức là, SlopeIdx 6 và OffsetIdx 15). Như được thể hiện trên Fig.15D, độ dốc có giá trị khác là -1,88 (SlopeIdx 3) không đạt được sự phù hợp hơn, nhưng chiếu đến trục y (tức là, chỗ giao nhau tại SliceQP_Y bằng 0) là 130, cao hơn nhiều so với giới hạn trên là 104.

Như được thể hiện trong phương trình (3), trong một số ví dụ, trạng thái xác suất ban đầu có thể được xác định một phần bằng cách nhân biến m với SliceQP_Y . Trong ví dụ của phương trình (3), mô hình tuyến tính của trạng thái xác suất được xác định ở điểm mà ở đó SliceQP_Y bằng không.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ lập mã video có thể sử dụng độ lệch sao cho mô hình tuyến tính của trạng thái xác suất được xác định ở giá trị của SliceQP_Y chứ không phải ở giá trị không. Điểm trên mô hình tuyến tính mà tại đó trạng thái xác suất được xác định có thể được gọi là điểm neo tham số lượng tử hóa ($\text{QP}_{\text{anchor}}$). Ví dụ, bộ lập mã video có thể lấy SliceQP trừ đi độ lệch, và nhân kết quả với biến số m . Trong một ví dụ, điều này có thể được hoàn thành bằng cách biến đổi phương trình (3) thành phương trình (8) như được thể hiện dưới đây.

$$\text{InitProbState} = \text{Clip3}(1, 127, ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg 4) + n) \quad (8)$$

trong đó $\text{QP}_{\text{anchor}}$ là giá trị QP mà ở đó đường giao nhau được xác định. Theo cách này, đường giao nhau có thể không vượt quá phạm vi giá trị của các trạng thái xác suất (từ 1 đến 127 trong ví dụ này), và việc xem xét phép chiếu lớn ở trục y được loại bỏ, đặc biệt là khi đường dốc rất dốc. Trên Fig.15D, SlopeIdx bằng 3, $\text{QP}_{\text{anchor}}$ bằng 32, và chỗ giao nhau tại $\text{QP}_{\text{anchor}}$ bằng 72. Các giá trị khác cho $\text{QP}_{\text{anchor}}$ bao gồm, nhưng không chỉ giới hạn ở, 8, 16, 64, 128, v.v..

Sử dụng dạng biểu diễn khác cho hàm tuyến tính. Trong các ví dụ nêu trên, hàm tuyến tính được sử dụng để mô hình hóa trạng thái xác suất bằng SliceQP_Y được xác định bằng cặp đường dốc và đường giao nhau, tương tự như HEVC và VVC hiện thời. Phần này đề xuất các dạng biểu diễn thay thế cho các hàm tuyến tính, trong đó hàm dạng đường thẳng được xác định bởi hai đường giao nhau ở hai QP khác nhau. So với dạng biểu diễn đường dốc/đường giao nhau, dạng biểu diễn hai đường giao nhau này có tính linh hoạt hơn khi so khớp, vì nó cho phép nhiều giá trị độ dốc hơn nhiều mà không ở dạng chuỗi số học.

Biểu thị hai đường giao nhau và hai QP tương ứng lần lượt là int1 , int2 , QP1 và QP2 . QP1 và QP2 có thể tùy ý, miễn là chúng nằm trong phạm vi có giá trị được xác định trong VVC hiện thời (từ 0 đến 63). Tuy nhiên, vì InitProbState được tính toán bằng phương trình (9) hoặc phương trình (10) như dưới đây (lưu ý rằng để đơn giản và rõ ràng thì các phép cắt và dịch chuyển trung gian không được xem xét ở đây),

$$\frac{\text{int1}-\text{int2}}{\text{QP1}-\text{QP2}} (\text{SliceQP}_Y - \text{QP1}) + \text{int1} \quad (9)$$

$$\frac{\text{int1}-\text{int2}}{\text{QP1}-\text{QP2}} (\text{SliceQP}_Y - \text{QP2}) + \text{int2} \quad (10)$$

Chênh lệch giữa QP1 và QP2 tốt hơn là một số nguyên lũy thừa 2, để các phép chia trong phương trình (9) và phương trình (10) có thể được thay thế bằng phép dịch phải.

Dạng biểu diễn hai đường giao nhau có thể được còn mở rộng thành dạng biểu diễn nhiều đường giao nhau xác định các hàm tuyến tính từng mảnh. Một ví dụ như vậy là bin từ SigFlag độ chói, xem các hình vẽ từ Fig.16A đến Fig.16C, trong đó điểm ngoặt diễn ra tại QP khoảng 30 hoặc 31 và đường thẳng được xác định bởi cặp đường dốc/đường giao nhau hoặc hai đường giao nhau không đủ tốt cho việc so khớp (xem Fig.16A). Trong trường hợp này, dạng biểu diễn ba đường giao nhau được đề xuất, sử dụng ba tham số khởi tạo: đường giao nhau tại QP bằng 31 và hai đường giao nhau khác tại QP lần lượt nhỏ hơn và lớn hơn 31 (xem Fig.16B). Để phù hợp với sự phân phối xác suất một cách chính xác hơn nữa, dạng biểu diễn bốn đường giao nhau có thể được sử dụng, trong đó tham số khởi tạo bổ sung biểu diễn đường giao nhau tại QP bằng 37 được sử dụng để thu điểm ngoặt nhỏ khác (xem Fig.16C).

Giữ độ chính xác trung gian cao hơn. Các ví dụ trên đây giới thiệu các cách biến đổi các phương trình (1) đến (3), và suy ra InitProbState trong miền tuyến tính. InitProbState vẫn có độ chính xác 7 bit (0 đến 127), do kế thừa từ HEVC, và cần phải dịch trái 3 bit và 7 bit, để thẳng hàng với các độ chính xác của ProbabilityStateL và ProbabilityStateH được xác định trong VVC hiện thời, như trong các phương trình (11) và (12).

$$\text{ProbabilityStateL} = \text{InitProbState} \ll 3 \quad (11)$$

$$\text{ProbabilityStateH} = \text{InitProbState} \ll 7 \quad (12)$$

Lưu ý rằng bước tìm kiếm theo bảng ngay trước các phương trình (4) và (5) được lưu trong kỹ thuật này.

Việc áp dụng phép dịch phải để suy ra InitProbState (xem phương trình (3)) theo sau là các phép dịch trái (xem các phương trình (11) và (12)) chắc chắn làm mất nhiều thông tin hơn là rời tất cả (hoặc một phần của) phép dịch phải đến bước cuối cùng, vì

như thế sẽ bảo toàn được độ chính xác trung gian cao hơn. Hai ví dụ sau đây đã rời tất cả phép dịch phải đến bước cuối cùng.

Trong một ví dụ:

$$\text{InitProbState} = \text{Clip3}(1, 2047, ((m * \text{SliceQP}_Y) + (n \ll 4))) \quad (13)$$

$$\text{ProbabilityStateL} = \text{InitProbState} \gg 1 \quad (14)$$

$$\text{ProbabilityStateH} = \text{InitProbState} \ll 3 \quad (15)$$

Trong ví dụ khác:

$$\text{InitProbState} = \text{Clip3}(0, 2047, ((m * \text{SliceQP}_Y) + (n \ll 4))) \quad (16)$$

$$\text{ProbabilityStateL} = \text{InitProbState} \gg 1 \quad (17)$$

$$\text{ProbabilityStateH} = (\text{InitProbState} \ll 3) + 4 \quad (18)$$

Liên quan đến CE1: Việc đơn giản hóa JVET-O0191 bằng cách sử dụng 4 hoặc 6 bit trên một giá trị khởi tạo, JVET-O0946, Gothenburg, Thụy sĩ, từ ngày 3 - 12/7/2019 (sau đây gọi là “JVET-O0946”) đề xuất kỹ thuật khởi tạo sau đây:

$$q = \text{Clip3}(9, 23, \text{SliceQP}_Y \gg 1) - 8$$

$$a = (\text{initValue} \gg 3) * 9$$

$$b = (\text{initValue} \& 7) * 9$$

$$\text{ProbabilityStateL} = 16 * a + (b - a) * q + 8$$

$$\text{ProbabilityStateH} = \text{ProbabilityStateL} \ll 4$$

Tuy nhiên, kỹ thuật được đề xuất trong JVET-O0946 có thể có một hoặc nhiều nhược điểm. Ví dụ, kỹ thuật được đề xuất trong JVET-O0946 có thể làm hao hụt một số độ chính xác trung gian.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ lập mã video có thể thực hiện quá trình khởi tạo như sau:

$$q = \text{Clip3}(18, 46, \text{SliceQP}_Y) - 16$$

$$a = (\text{initValue} \gg 3) * 9$$

$$b = (\text{initValue} \& 7) * 9$$

$$\text{ProbabilityStateL} = 16 * a + ((b - a) * q \gg 1) + 8$$

$$\text{ProbabilityStateH} = \text{ProbabilityStateL} \ll 4$$

Như có thể thấy trên đây, bằng cách rời phép dịch phải đến bước cuối cùng (tức là bước tính toán ProbabilityStateL), kỹ thuật này cho phép độ chính xác trung gian cao.

Khởi tạo trong miền bậc hai. Các ví dụ trên đây có thể dành cho các trường hợp mà đầu ra của quy trình khởi tạo biểu diễn trạng thái xác suất trong miền tuyến tính. Tuy nhiên, đối với một số nguồn đầu vào, có thể có mong muốn khiến cho trạng thái xác suất biểu diễn xác suất trong miền lôgarit, tương ứng với biểu diễn entropy trong miền tuyến tính. Để đạt được dạng biểu diễn trong miền lôgarit, sáng chế đề xuất sử dụng các trạng thái xác suất trong miền bậc hai. Các hình vẽ Fig.17A và Fig.17B thể hiện ví dụ về ánh xạ từ miền bậc hai InitProbState đến các xác suất tuyến tính được sử dụng trong máy lập mã số học của VVC, có vẻ như ánh xạ cho InitProbState lôgarit (xem các hình vẽ Fig.11A và Fig.11B), nhưng có thể được thực hiện bằng các phép nhân (lưu ý rằng LUT lưu trữ tất cả kết quả phép nhân có thể có có thể được tạo ra từ trước để sử dụng nếu có phương án thực hiện nào ưu tiên điều này). Dưới đây là một ví dụ,

```
slope      = int( initId & 0x0F) - 8;
offset     = int( initId & 0xF0) - 128;
shp        = offset + ((slope * (qp - 32)) >> 1);
idx        = Clip3( -127, 127, shp );
val        = 128 - abs(idx);
p1         = (idx < 0 ? 32768 - val2 : val2);
ProbabilityStateL = p1 >> 1
ProbabilityStateH = p1 >> 5
```

trong đó initId là giá trị khởi tạo có 4 bit cao và thấp lần lượt biểu diễn độ dốc và độ lệch, và idx nghĩa là trạng thái xác suất có độ chính xác 8 bit và chính tâm tại 0.

Lưu ý rằng phương pháp được đề xuất trong phần này có thể được kết hợp với việc sử dụng độ chính xác trung gian cao hơn như đã giới thiệu trên đây.

Khởi tạo cho các phân phối xác suất cực trị. Một số lượng lớn các bin có thể có các phân phối xác suất cực trị, nghĩa là các bin này có xác suất rất cao là bằng 1 hoặc 0,

bất kể giá trị của SliceQP_Y. Các hình vẽ Fig.18A và Fig.18B là đồ thị minh họa các ví dụ về các phân phối xác suất cực trị.

Để mô hình hóa cho các phân phối xác suất cực trị như vậy, độ dốc có thể phải bằng 0, và trạng thái xác suất ban đầu có thể chỉ phụ thuộc vào n (xem các phương trình (2) và (7)). Như vậy, trạng thái xác suất ban đầu có thể không thay đổi theo tham số lượng tử hóa (QP). Công thức suy ra n, như trong phương trình (2) hoặc phương trình (7), có thể được thiết kế sao cho các xác suất tiệm cận 0 và 1 đều có thể được biểu diễn một cách có hiệu quả. Các ví dụ sau đây dựa vào dạng biểu diễn 7 bit (0 đến 127) của các trạng thái xác suất và có thể được mở rộng đến các độ chính xác khác.

Lấy phương trình (2) làm ví dụ. Khi OffsetIdx nằm trong khoảng từ 0 đến 15, giá trị n lớn nhất có thể đạt được là 104, biểu diễn xác suất 0,8189 (tức là, 104/128), khác xa so với 1,0.

Trong tài liệu “Simplification of the initialization process for context variables”, JVET-N0301, tại Geneva, Thụy Sĩ, từ ngày 19-27/3/2019, n được suy ra với độ chính xác 10 bit, có OffsetIdx nằm trong khoảng từ 0 đến 7, như trong phương trình (19).

$$n = ((\text{OffsetIdx} * 5) \ll 5) + 32 \quad (19)$$

Dạng biểu diễn 7 bit tương đương của nó của n được thể hiện trong phương trình (20),

$$n = ((\text{OffsetIdx} * 5) \ll 2) + 4 \quad (20)$$

trong đó xác suất gần 0 nhất mà nó có thể đạt được là 0,0313 (OffsetIdx bằng 0 và n bằng 4, $4/128 = 0,0313$) và xác suất gần 1,0 nhất mà nó có thể đạt được là 0,9688 (OffsetIdx bằng 6 và n bằng 124, $124/128 = 0,9688$). Tuy nhiên, các con số gần đúng này có thể không đủ chính xác để biểu diễn các phân phối xác suất cực trị.

Theo một hoặc nhiều kỹ thuật của sáng chế, bộ lập mã video có thể sử dụng một hoặc nhiều trong số các công thức sau đây để suy ra tham số khởi tạo (ví dụ, n).

Nếu OffsetIdx là 3 bit (từ 0 đến 7), n có thể được suy ra như được thể hiện trong phương trình (21).

$$n = (\text{OffsetIdx} * 18) + 1 \quad (21)$$

Nếu OffsetIdx là 4 bit (từ 0 đến 15), n có thể được suy ra như được thể hiện trong phương trình (22).

$$n = (\text{OffsetIdx} * 9) + 1 \quad (22)$$

Như có thể thấy, trong cả hai trường hợp, n có thể đạt đến 1 và 127, nghĩa là có thể đạt được các xác suất là 0,0078 (tức là, $1/128$) và 0,9922 (tức là, $127/128$). Các xác suất này có thể đủ gần 0 và 1. Theo cách này, các kỹ thuật của sáng chế cho phép bộ lập mã video khởi tạo các phân phối xác suất cực trị chính xác hơn, điều này có thể làm giảm tốc độ của dữ liệu video được lập mã.

Trong một số ví dụ, m , n , và/hoặc initState có thể được xác định theo một trong các phương trình sau:

(23)

$$m = \text{slopeIdx} * 4 - 36$$

$$n = \text{OffsetIdx} * 9 + 1$$

$$\text{initState} = (m * (\text{qp} - 24) \gg 4) + n$$

(24)

$$m = \text{slopeIdx} * 8 - 32$$

$$n = \text{OffsetIdx} * 18 + 1$$

$$\text{initState} = (m * (\text{qp} - 16) \gg 4) + n$$

(25)

$$m = \text{slopeIdx} * 5 - 45$$

$$n = \text{OffsetIdx} * 8 + 7$$

$$\text{initState} = (m * (\text{qp} - 32) \gg 4) + n$$

Các hình vẽ Fig.4A và Fig.4B là sơ đồ khái niệm minh họa một ví dụ về cấu trúc cây nhị phân tứ phân (QTBT) 130, và đơn vị cây lập mã (CTU) 132 tương ứng. Các đường nét liền biểu thị việc tách cây tứ phân và các đường nét đứt biểu thị việc tách cây nhị phân. Trong mỗi nút tách (tức là, nút không phải nút lá) của cây nhị phân, một cờ được báo hiệu để chỉ báo kiểu tách nào (tức là, tách ngang hay tách dọc) được sử dụng, trong đó 0 biểu thị tách ngang và 1 biểu thị tách dọc trong ví dụ này. Đối với việc tách

cây tứ phân, không cần thiết biểu thị kiểu tách, vì các nút cây tứ phân chia khối theo chiều ngang và theo chiều dọc thành 4 khối con có kích thước bằng nhau. Do đó, bộ mã hóa video 200 có thể mã hóa, và bộ giải mã video 300 có thể giải mã, các phần tử cú pháp (ví dụ như thông tin tách) cho mức cây khu vực (tức là, mức thứ nhất) của cấu trúc QTBT 130 (tức là, các đường nét liền) và các phần tử cú pháp (ví dụ như thông tin tách) cho mức cây dự đoán (tức là, mức thứ hai) của cấu trúc QTBT 130 (tức là, các đường nét đứt). Bộ mã hóa video 200 có thể mã hóa, và bộ giải mã video 300 có thể giải mã, dữ liệu video, như dữ liệu dự đoán và biến đổi, cho các CU được biểu diễn bởi các nút lá đầu cuối của cấu trúc QTBT 130.

Nói chung, CTU 132 trên Fig.4B có thể được kết hợp với các tham số xác định kích thước của các khối tương ứng với các nút của cấu trúc QTBT 130 ở mức thứ nhất và thứ hai. Các tham số này có thể bao gồm kích thước CTU (biểu diễn kích thước CTU 132 trong các mẫu), kích thước cây tứ phân tối thiểu (MinQTSIZE, biểu diễn kích thước nút lá cây tứ phân tối thiểu được phép), kích thước cây nhị phân tối đa (MaxBTSIZE, biểu diễn kích thước nút gốc cây nhị phân tối đa được phép), độ sâu cây nhị phân tối đa (MaxBTDepth, biểu diễn độ sâu cây nhị phân tối đa được phép) và kích thước cây nhị phân tối thiểu (MinBTSize, biểu diễn kích thước nút lá cây nhị phân tối thiểu được phép).

Nút gốc của cấu trúc QTBT tương ứng với CTU có thể có bốn nút con ở mức thứ nhất của cấu trúc QTBT, mỗi nút có thể được phân chia theo kỹ thuật phân chia cây tứ phân. Tức là, các nút của mức thứ nhất là các nút lá (không có nút con) hoặc có bốn nút con. Ví dụ về cấu trúc QTBT 130 biểu diễn các nút như vậy bao gồm nút cha và các nút con có các đường nét liền cho các nhánh. Nếu các nút của mức thứ nhất không lớn hơn kích thước nút gốc cây nhị phân lớn nhất cho phép (MaxBTSIZE), thì chúng có thể được cây nhị phân tương ứng phân chia tiếp. Việc phân chia cây nhị phân của một nút có thể được lặp lại cho đến khi các nút tạo ra từ quy trình phân tách đạt đến kích thước nút lá cây nhị phân tối thiểu được phép (MinBTSize) hoặc độ sâu cây nhị phân tối đa được phép (MaxBTDepth). Ví dụ về cấu trúc QTBT 130 biểu diễn các nút như vậy dưới dạng có các đường nét đứt cho các nhánh. Nút lá cây nhị phân được gọi là một đơn vị lập mã (CU), đơn vị này được sử dụng để dự đoán (ví dụ, dự đoán nội ảnh hoặc dự đoán liên ảnh) và biến đổi, mà không phân chia thêm. Như mô tả ở trên, các CU cũng có thể được gọi là các “khối dữ liệu video” hoặc “các khối”.

Trong một ví dụ của cấu trúc phân chia QTBT, kích thước CTU được thiết lập là 128×128 (mẫu độ chói và hai mẫu sắc độ 64×64 tương ứng), MinQTSIZE được thiết lập là 16×16 , MaxBTSIZE được thiết lập là 64×64 , MinBTSIZE (cho cả chiều rộng lẫn chiều cao) được thiết lập là 4 và MaxBTDepth được thiết lập là 4. Phân chia cây tứ phân được áp dụng cho CTU trước tiên để tạo ra các nút lá cây tứ phân. Các nút lá cây tứ phân có thể có kích thước từ 16×16 (tức là MinQTSIZE) đến 128×128 (tức là, kích thước CTU). Nếu nút lá cây tứ phân là 128×128 , nó sẽ không được phân chia tiếp bởi cây nhị phân bởi vì kích thước này vượt quá MaxBTSIZE (tức là 64×64 trong ví dụ này). Cách khác, nút lá cây tứ phân sẽ được phân chia tiếp bởi cây nhị phân. Do đó, nút lá cây tứ phân cũng là nút gốc cho cây nhị phân và có độ sâu cây nhị phân bằng 0. Khi độ sâu cây nhị phân đạt đến MaxBTDepth (bằng 4 trong một ví dụ), thì không được phép tách tiếp nữa. Khi nút cây nhị phân có độ rộng bằng MinBTSIZE (bằng 4 trong ví dụ này), việc này ngụ ý rằng không được phép tách theo chiều dọc nữa. Tương tự, nút cây nhị phân có chiều cao bằng MinBTSIZE ngụ ý rằng không được phép tách theo chiều ngang nữa cho nút cây nhị phân đó. Như nêu trên đây, các nút lá của cây nhị phân được gọi là các CU, và được xử lý thêm theo kỹ thuật dự đoán và biến đổi mà không phân chia thêm nữa.

Fig.5 là sơ đồ khối minh họa một ví dụ về bộ mã hóa video 200 có thể thực hiện các kỹ thuật của sáng chế. Fig.5 được đưa ra để giải thích và không nên được xem là làm giới hạn các kỹ thuật như được minh họa và được mô tả rộng rãi trong sáng chế. Với mục đích giải thích, sáng chế mô tả bộ mã hóa video 200 trong ngữ cảnh các chuẩn lập mã video như chuẩn lập mã video HEVC và chuẩn lập mã video H.266 đang phát triển. Tuy nhiên, các kỹ thuật của sáng chế không bị giới hạn ở các chuẩn lập mã video này, và được áp dụng cho quy trình mã hóa và giải mã video nói chung.

Theo ví dụ trên Fig.5, bộ mã hóa video 200 bao gồm bộ nhớ dữ liệu video 230, đơn vị lựa chọn chế độ 202, đơn vị tạo dữ liệu dư 204, đơn vị xử lý biến đổi 206, đơn vị lượng tử hóa 208, đơn vị lượng tử hóa ngược 210, đơn vị xử lý biến đổi ngược 212, đơn vị tái tạo 214, đơn vị lọc 216, bộ đệm ảnh được giải mã (decoded picture buffer - DPB) 218 và đơn vị mã hóa entropy 220. Bất kỳ hoặc tất cả trong số bộ nhớ dữ liệu video 230, đơn vị lựa chọn chế độ 202, đơn vị tạo dữ liệu dư 204, đơn vị xử lý biến đổi 206, đơn vị lượng tử hóa 208, đơn vị lượng tử hóa ngược 210, đơn vị xử lý biến đổi ngược 212, đơn vị tái tạo 214, đơn vị lọc 216, DPB 218 và đơn vị mã hóa entropy 220 có thể được triển khai trong một hoặc nhiều bộ xử lý hoặc trong mạch xử lý. Hơn nữa, bộ mã hóa video

200 có thể bao gồm bộ xử lý hoặc mạch xử lý bổ sung hoặc thay thế để thực hiện các chức năng này và các chức năng khác.

Bộ nhớ dữ liệu video 230 có thể lưu trữ dữ liệu video cần được mã hóa bởi các thành phần của bộ mã hóa video 200. Bộ mã hóa video 200 có thể nhận dữ liệu video lưu trữ trong bộ nhớ dữ liệu video 230 từ, ví dụ, nguồn video 104 (Fig.1). DPB 218 có thể hoạt động như bộ nhớ ảnh tham chiếu để lưu trữ dữ liệu video tham chiếu dùng cho việc dự đoán dữ liệu video tiếp theo bằng bộ mã hóa video 200. Bộ nhớ dữ liệu video 230 và DPB 218 có thể được tạo ra bởi thiết bị bất kỳ trong số nhiều thiết bị nhớ, như bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM), bao gồm DRAM đồng bộ (synchronous DRAM - SDRAM), RAM từ điện trở (magnetoresistive RAM - MRAM), phép dịch (resistive RAM - RRAM), hoặc các loại thiết bị nhớ khác. Bộ nhớ dữ liệu video 230 và DPB 218 có thể được cung cấp bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Theo các ví dụ khác nhau, bộ nhớ dữ liệu video 230 có thể nằm trên chip cùng các thành phần khác của bộ mã hóa video 200, như được minh họa, hoặc ngoài chip so với các thành phần này.

Theo sáng chế này, việc tham chiếu đến bộ nhớ dữ liệu video 230 không được hiểu là bị giới hạn ở bộ nhớ bên trong bộ mã hóa video 200, trừ khi được mô tả cụ thể như vậy, hoặc bộ nhớ bên ngoài bộ mã hóa video 200, trừ khi được mô tả cụ thể như vậy. Thay vào đó, bộ nhớ dữ liệu video 230 nên được hiểu là bộ nhớ tham chiếu lưu trữ dữ liệu video mà bộ mã hóa video 200 nhận để mã hóa (ví dụ, dữ liệu video cho khối hiện thời sẽ được mã hóa). Bộ nhớ 106 trên Fig.1 có thể cũng cung cấp việc lưu trữ tạm thời các dữ liệu đầu ra từ các đơn vị khác nhau của bộ mã hóa video 200.

Một số đơn vị trên Fig.5 được minh họa để giúp hiểu các hoạt động được thực hiện bởi bộ mã hóa video 200. Các đơn vị có thể được thực hiện như các mạch chức năng cố định, các mạch lập trình được, hoặc kết hợp của chúng. Các mạch chức năng cố định là các mạch có chức năng cụ thể, và được thiết lập trước trên các thao tác mà có thể được thực hiện. Các mạch lập trình được là các mạch mà có thể được lập trình để thực hiện một số nhiệm vụ khác nhau, và cung cấp chức năng linh hoạt trong các hoạt động mà có thể được thực hiện. Ví dụ, các mạch lập trình được có thể thực thi phần mềm hoặc firmware khiến cho các mạch lập trình được hoạt động theo cách thức được xác định bởi các lệnh của phần mềm hoặc firmware. Các mạch chức năng cố định có thể thực thi các lệnh phần mềm (ví dụ, để thu các tham số hoặc xuất ra các tham số), nhưng

các kiểu hoạt động mà các mạch chức năng cố định thực hiện nói chung là không thể thay đổi. Theo một số ví dụ, một hoặc nhiều trong số các đơn vị có thể là các khối mạch riêng (chức năng cố định hoặc có thể lập trình được), và theo một số ví dụ, một hoặc nhiều đơn vị có thể là các mạch được tích hợp.

Bộ mã hóa video 200 có thể bao gồm đơn vị logic số học (arithmetic logic unit - ALU), đơn vị chức năng cơ bản (elementary function unit - EFU), mạch kỹ thuật số, mạch tương tự và/hoặc lõi có thể lập trình, được tạo ra từ các mạch lập trình được. Trong các ví dụ mà các hoạt động của bộ mã hóa video 200 được thực hiện bằng cách sử dụng phần mềm được thực thi bởi các mạch lập trình được, bộ nhớ 106 (Fig.1) có thể lưu trữ mã đối tượng của phần mềm mà bộ mã hóa video 200 thu và thực thi, hoặc một bộ nhớ khác trong bộ mã hóa video 200 (không được thể hiện trên hình vẽ) có thể lưu trữ các lệnh như vậy.

Bộ nhớ dữ liệu video 230 được tạo cấu hình để lưu trữ dữ liệu video nhận được. Bộ mã hóa video 200 có thể truy hồi ảnh của dữ liệu video từ bộ nhớ dữ liệu video 230 và cung cấp dữ liệu video cho đơn vị tạo dữ liệu dư 204 và đơn vị lựa chọn chế độ 202. Dữ liệu video trong bộ nhớ dữ liệu video 230 có thể là dữ liệu video thô cần được mã hóa.

Đơn vị lựa chọn chế độ 202 bao gồm đơn vị ước lượng chuyển động 222, đơn vị bù chuyển động 224 và đơn vị dự đoán nội ảnh 226. Đơn vị lựa chọn chế độ 202 có thể bao gồm các đơn vị có chức năng bổ sung để thực hiện dự đoán video theo các chế độ dự đoán khác. Ví dụ, đơn vị lựa chọn chế độ 202 có thể bao gồm đơn vị bảng màu, đơn vị sao chép khối nội ảnh (có thể là một phần của đơn vị ước lượng chuyển động 222 và/hoặc đơn vị bù chuyển động 224), đơn vị afin, đơn vị mô hình tuyến tính (linear model - LM), hoặc tương tự.

Đơn vị lựa chọn chế độ 202 thường phối hợp nhiều lượt mã hóa để kiểm tra các kết hợp của các tham số mã hóa và các giá trị biến dạng tốc độ thu được cho các tổ hợp như vậy. Các tham số mã hóa có thể bao gồm chia các CTU thành các CU, các chế độ dự đoán cho các CU, các kiểu biến đổi cho dữ liệu dư của các CU, các tham số lượng tử hóa cho dữ liệu dư của các CU, v.v.. Đơn vị lựa chọn chế độ 202 có thể cuối cùng chọn tổ hợp của các tham số mã hóa có giá trị biến dạng tốc độ tốt hơn các tổ hợp đã thử nghiệm khác.

Bộ mã hóa video 200 có thể phân chia ảnh được truy hồi từ bộ nhớ dữ liệu video 230 thành một loạt các CTU, và đóng gói một hoặc nhiều CTU trong lát. Đơn vị lựa chọn chế độ 202 có thể chia CTU của ảnh theo cấu trúc cây, chẳng hạn như cấu trúc QTBT hoặc cấu trúc cây tứ phân của HEVC được mô tả ở trên. Như mô tả ở trên, bộ mã hóa video 200 có thể tạo ra một hoặc nhiều CU từ quy trình chia CTU theo cấu trúc cây. CU như vậy cũng có thể được gọi chung là “khối video” hoặc “khối”.

Nói chung, đơn vị lựa chọn chế độ 202 cũng điều khiển các thành phần của chúng (ví dụ, đơn vị ước lượng chuyển động 222, đơn vị bù chuyển động 224, và đơn vị dự đoán nội ảnh 226) để tạo ra khối dự đoán cho khối hiện thời (ví dụ, CU hiện thời, hoặc trong HEVC, phần chồng lấn của PU và TU). Để dự đoán liên ảnh khối hiện thời, đơn vị ước lượng chuyển động 222 có thể thực hiện tìm kiếm chuyển động để xác định một hoặc nhiều khối tham chiếu phù hợp trong một hoặc nhiều ảnh tham chiếu (ví dụ, một hoặc nhiều ảnh được lập mã trước đó được lưu trữ trong DPB 218). Cụ thể, đơn vị ước lượng chuyển động 222 có thể tính toán giá trị thể hiện khối tham chiếu tiềm năng là tương tự như thế nào với khối hiện thời, ví dụ, theo tổng chênh lệch tuyệt đối (sum of absolute difference - SAD), tổng chênh lệch bình phương (sum of squared differences - SSD), chênh lệch tuyệt đối trung bình (mean absolute difference - MAD), chênh lệch bình phương trung bình (mean squared difference - MSD), hoặc tương tự. Đơn vị ước lượng chuyển động 222 nói chung có thể thực hiện các tính toán này bằng cách sử dụng các giá trị chênh lệch từng mẫu giữa khối hiện thời và khối tham chiếu đang được xem xét. Đơn vị ước lượng chuyển động 222 có thể xác định khối tham chiếu có giá trị thấp nhất thu được từ các phép tính này, biểu thị khối tham chiếu mà phù hợp nhất với khối hiện thời.

Đơn vị ước lượng chuyển động 222 có thể tạo ra một hoặc nhiều vector chuyển động (MV) xác định các vị trí của các khối tham chiếu trong ảnh tham chiếu so với vị trí của khối hiện thời trong ảnh hiện thời. Sau đó, đơn vị ước lượng chuyển động 222 có thể cung cấp các vector chuyển động cho đơn vị bù chuyển động 224. Ví dụ, đối với dự đoán liên ảnh một chiều, đơn vị ước lượng chuyển động 222 có thể cung cấp một vector chuyển động, trong khi đối với dự đoán liên ảnh hai chiều, đơn vị ước lượng chuyển động 222 có thể cung cấp hai vector chuyển động. Sau đó, đơn vị bù chuyển động 224 có thể tạo ra khối dự đoán bằng cách sử dụng các vector chuyển động. Ví dụ, đơn vị bù chuyển động 224 có thể truy hồi dữ liệu của khối tham chiếu bằng cách sử dụng vector

chuyển động. Một ví dụ khác, nếu vector chuyển động có độ chính xác mẫu phân số, thì đơn vị bù chuyển động 224 có thể nội suy các giá trị cho khối dự đoán theo một hoặc nhiều bộ lọc nội suy. Hơn nữa, để dự đoán liên ảnh hai chiều, đơn vị bù chuyển động 224 có thể truy hồi dữ liệu cho hai khối tham chiếu được xác định bởi các vector chuyển động tương ứng và kết hợp dữ liệu đã truy hồi, ví dụ, qua phép lấy trung bình từng mẫu hoặc lấy trung bình có trọng số.

Một ví dụ khác, để dự đoán nội ảnh hoặc lập mã dự đoán nội ảnh, đơn vị dự đoán nội ảnh 226 có thể tạo ra khối dự đoán từ các mẫu lân cận với khối hiện thời. Ví dụ, đối với các chế độ định hướng, đơn vị dự đoán nội ảnh 226 có thể thường kết hợp toán học các giá trị của các mẫu lân cận và gán các giá trị được tính toán này theo hướng xác định trên khối hiện thời để tạo ra khối dự đoán. Một ví dụ khác, đối với chế độ DC, đơn vị dự đoán nội ảnh 226 có thể tính giá trị trung bình của các mẫu lân cận với khối hiện thời và tạo ra khối dự đoán để bao gồm giá trị trung bình thu được này cho mỗi mẫu của khối dự đoán.

Đơn vị lựa chọn chế độ 202 cung cấp khối dự đoán cho đơn vị tạo dữ liệu dư 204. Đơn vị tạo dữ liệu dư 204 nhận phiên bản thô, chưa lập mã của khối hiện thời từ bộ nhớ dữ liệu video 230 và khối dự đoán từ đơn vị lựa chọn chế độ 202. Đơn vị tạo dữ liệu dư 204 tính toán các giá trị chênh lệch theo từng mẫu giữa khối hiện thời và khối dự đoán. Giá trị chênh lệch theo từng mẫu thu được xác định khối dư cho khối hiện thời. Theo một số ví dụ, đơn vị tạo dữ liệu dư 204 cũng có thể xác định giá trị chênh lệch giữa các giá trị mẫu trong khối dư để tạo ra khối dư bằng cách sử dụng kỹ thuật điều biến mã xung chênh lệch dư (residual differential pulse code modulation - RDPCM). Theo một số ví dụ, đơn vị tạo dữ liệu dư 204 có thể được tạo bằng cách sử dụng một hoặc nhiều mạch trừ mà thực hiện phép trừ nhị phân.

Trong các ví dụ mà đơn vị lựa chọn chế độ 202 chia các CU thành các PU, mỗi PU có thể được kết hợp với một đơn vị dự đoán độ chói và các đơn vị dự đoán sắc độ tương ứng. Bộ mã hóa video 200 và bộ giải mã video 300 có thể hỗ trợ các PU có các kích thước khác nhau. Như nêu trên đây, kích thước của CU có thể chỉ kích thước của khối lập mã độ chói của CU và kích thước của PU có thể chỉ kích thước của đơn vị dự đoán độ chói của PU. Giả sử kích thước của một CU cụ thể là $2N \times 2N$, bộ mã hóa video 200 có thể hỗ trợ các kích thước PU $2N \times 2N$ hoặc $N \times N$ để dự đoán nội ảnh, và các kích thước PU đối xứng $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, hoặc tương tự để dự đoán liên ảnh. Bộ

mã hóa video 200 và bộ giải mã video 300 cũng có thể hỗ trợ phân chia bất đối xứng các kích thước PU $2N \times nU$, $2N \times nD$, $nL \times 2N$, và $nR \times 2N$ để dự đoán liên ảnh.

Trong các ví dụ mà đơn vị lựa chọn chế độ không chia tiếp CU thành các PU, mỗi CU có thể được kết hợp với một khối lập mã độ chói và các khối lập mã sắc độ tương ứng. Như nêu trên đây, kích thước của CU có thể là kích thước của khối lập mã độ chói của CU. Bộ mã hóa video 200 và bộ giải mã video 300 có thể hỗ trợ các kích thước CU $2N \times 2N$, $2N \times N$ hoặc $N \times 2N$.

Đối với các kỹ thuật lập mã dữ liệu video khác, như lập mã chế độ sao chép khối nội ảnh, lập mã chế độ afin, và lập mã chế độ mô hình tuyến tính (linear model - LM), chẳng hạn, đơn vị lựa chọn chế độ 202, qua các đơn vị tương ứng liên quan đến các kỹ thuật lập mã, tạo ra khối dự đoán cho khối hiện thời đang mã hóa. Theo một số ví dụ, như kỹ thuật lập mã chế độ bảng màu, đơn vị lựa chọn chế độ 202 có thể không tạo ra khối dự đoán, và thay vào đó tạo ra các phần tử cú pháp biểu thị cách thức tái tạo khối dựa trên bảng màu đã chọn. Trong các chế độ như vậy, đơn vị lựa chọn chế độ 202 có thể cung cấp các phần tử cú pháp này cho đơn vị mã hóa entropy 220 cần được mã hóa.

Như mô tả trên đây, đơn vị tạo dữ liệu dư 204 nhận dữ liệu video cho khối hiện thời và khối dự đoán tương ứng. Sau đó, đơn vị tạo dữ liệu dư 204 tạo ra khối dư cho khối hiện thời. Để tạo ra khối dư, đơn vị tạo dữ liệu dư 204 tính toán các giá trị chênh lệch theo từng mẫu giữa khối hiện thời và khối dự đoán.

Đơn vị xử lý biến đổi 206 áp dụng một hoặc nhiều phép biến đổi cho khối dư để tạo ra khối gồm các hệ số biến đổi (ở đây được gọi là "khối hệ số biến đổi"). Đơn vị xử lý biến đổi 206 có thể áp dụng các phép biến đổi khác nhau cho khối dư để tạo ra khối hệ số biến đổi. Ví dụ, đơn vị xử lý biến đổi 206 có thể áp dụng phép biến đổi cosin rời rạc (discrete cosine transform - DCT), biến đổi có hướng, biến đổi Karhunen-Loeve (Karhunen-Loeve transform - KLT) hoặc biến đổi tương tự về mặt khái niệm cho khối dư. Theo một số ví dụ, đơn vị xử lý biến đổi 206 có thể thực hiện nhiều phép biến đổi cho khối dư, ví dụ, biến đổi chính và biến đổi phụ, như biến đổi quay chẳng hạn. Trong một số ví dụ, đơn vị xử lý biến đổi 206 không áp dụng các phép biến đổi cho khối biến đổi.

Đơn vị lượng tử hóa 208 có thể lượng tử hóa các hệ số biến đổi trong khối hệ số biến đổi, để tạo ra khối hệ số biến đổi lượng tử hóa. Đơn vị lượng tử hóa 208 có thể lượng tử hóa các hệ số biến đổi của khối hệ số biến đổi theo giá trị tham số lượng tử hóa

(quantization parameter - QP) kết hợp với khối hiện thời. Bộ mã hóa video 200 (ví dụ, qua đơn vị lựa chọn chế độ 202) có thể điều chỉnh mức lượng tử hóa áp dụng cho các khối hệ số biến đổi kết hợp với khối hiện thời bằng cách điều chỉnh giá trị QP kết hợp với CU. Quy trình lượng tử hóa có thể làm mất thông tin, và do đó, các hệ số biến đổi lượng tử hóa có thể có độ chính xác thấp hơn các hệ số biến đổi gốc được tạo ra bởi đơn vị xử lý biến đổi 206.

Đơn vị lượng tử hóa ngược 210 và đơn vị xử lý biến đổi ngược 212 có thể lần lượt áp dụng lượng tử hóa ngược và biến đổi ngược cho khối hệ số biến đổi được lượng tử hóa để tái tạo khối dư từ khối hệ số biến đổi. Đơn vị tái tạo 214 có thể tạo ra một khối tái tạo tương ứng với khối hiện thời (mặc dù có khả năng bị biến dạng ở mức độ nào đó) dựa trên khối dư được tái tạo và khối dự đoán được tạo ra bởi đơn vị lựa chọn chế độ 202. Ví dụ, đơn vị tái tạo 214 có thể thêm các mẫu của khối dư tái tạo vào các mẫu tương ứng từ khối dự đoán được tạo ra bởi đơn vị lựa chọn chế độ 202 để tạo ra khối được tái tạo.

Đơn vị lọc 216 có thể thực hiện một hoặc nhiều hoạt động lọc trên các khối được tái tạo. Ví dụ, đơn vị lọc 216 có thể thực hiện các hoạt động tách khối để giảm các thành phần lạ dạng khối dọc theo các cạnh của CU. Theo một số ví dụ, các hoạt động của đơn vị lọc 216 có thể được bỏ qua.

Bộ mã hóa video 200 lưu trữ các khối được tái tạo vào DPB 218. Chẳng hạn, trong các ví dụ mà không cần đến các hoạt động của đơn vị lọc 216, đơn vị tái tạo 214 có thể lưu trữ các khối được tái tạo vào DPB 218. Trong các ví dụ mà cần đến các hoạt động của đơn vị lọc 216, đơn vị tái tạo 216 có thể lưu trữ các khối được tái tạo đã lọc vào DPB 218. Đơn vị ước lượng chuyển động 222 và đơn vị bù chuyển động 224 có thể truy hồi ảnh tham chiếu từ DPB 218, được tạo ra từ các khối được tái tạo (và có thể được lọc), để dự đoán liên ảnh các khối của các ảnh được mã hóa sau đó. Ngoài ra, đơn vị dự đoán nội ảnh 226 có thể sử dụng các khối được tái tạo trong DPB 218 của ảnh hiện thời để dự đoán nội ảnh các khối khác trong ảnh hiện thời.

Nói chung, đơn vị mã hóa entropy 220 có thể mã hóa entropy các phần tử cú pháp thu được từ các thành phần chức năng khác của bộ mã hóa video 200. Ví dụ, đơn vị mã hóa entropy 220 có thể mã hóa entropy các khối hệ số biến đổi lượng tử hóa từ đơn vị lượng tử hóa 208. Theo ví dụ khác, đơn vị mã hóa entropy 220 có thể mã hóa entropy các phần tử cú pháp dự đoán (ví dụ, thông tin chuyển động để dự đoán liên ảnh hoặc

thông tin chế độ nội ảnh để dự đoán nội ảnh) từ đơn vị lựa chọn chế độ 202. Đơn vị mã hóa entropy 220 có thể thực hiện một hoặc nhiều thao tác mã hóa entropy trên các phần tử cú pháp, mà là một ví dụ khác về dữ liệu video, để tạo ra dữ liệu được mã hóa entropy. Ví dụ, đơn vị mã hóa entropy 220 có thể thực hiện thao tác lập mã có độ dài thay đổi thích ứng theo ngữ cảnh (context-adaptive variable length coding - CAVLC), thao tác CABAC, thao tác lập mã độ dài biến đổi đến biến đổi (variable-to-variable - V2V), thao tác lập mã số học nhị phân thích ứng theo ngữ cảnh dựa trên cú pháp (syntax-based context-adaptive binary arithmetic coding - SBAC), thao tác mã hóa entropy phân chia khoảng xác suất (Probability Interval Partitioning Entropy - PIPE), thao tác mã hóa hàm mũ- Golomb, hoặc một kiểu thao tác mã hóa entropy khác trên dữ liệu. Theo một số ví dụ, đơn vị mã hóa entropy 220 có thể hoạt động ở chế độ bỏ qua trong đó các phần tử cú pháp không được mã hóa entropy.

Fig.6 là sơ đồ khối của một ví dụ về đơn vị mã hóa entropy 220 mà có thể được tạo cấu hình để thực hiện CABAC theo các kỹ thuật của sáng chế. Phần tử cú pháp 1180 được nhập vào đơn vị mã hóa entropy 220. Nếu phần tử cú pháp đã là phần tử cú pháp giá trị nhị phân (ví dụ, có hoặc phần tử cú pháp khác chỉ có giá trị 0 và 1), thì bước nhị phân hóa có thể được bỏ qua. Nếu phần tử cú pháp là phần tử cú pháp không có giá trị nhị phân (ví dụ, phần tử cú pháp có thể có giá trị khác 1 hoặc 0), thì phần tử cú pháp không có giá trị nhị phân này được nhị phân hóa bằng bộ nhị phân 1200. Bộ nhị phân 1200 thực hiện ánh xạ phần tử cú pháp không có giá trị nhị phân thành chuỗi quyết định nhị phân. Các quyết định nhị phân này thường được gọi là các "bin". Ví dụ, đối với các mức hệ số biến đổi, giá trị của mức có thể được chia thành các bin liên tiếp, mỗi bin biểu thị liệu giá trị tuyệt đối của mức hệ số có lớn hơn một số giá trị hay không. Ví dụ, bin 0 (đôi khi được gọi là cờ có nghĩa) biểu thị liệu giá trị tuyệt đối của mức hệ số biến đổi có lớn hơn 0 hay không. Bin 1 biểu thị liệu giá trị tuyệt đối của mức hệ số biến đổi có lớn hơn 1 hay không, v.v.. Phép ánh xạ duy nhất có thể được triển khai cho mỗi phần tử cú pháp không có giá trị nhị phân.

Mỗi bin được tạo ra bởi bộ nhị phân 1200 được cấp cho phía lập mã số học nhị phân của đơn vị mã hóa entropy 220. Tức là, đối với tập hợp định trước gồm các phần tử cú pháp không có giá trị nhị phân, mỗi loại bin (ví dụ, bin 0) được lập mã trước loại bin tiếp theo (ví dụ, bin 1). Bước lập mã có thể được thực hiện ở chế độ thông thường hoặc chế độ bỏ qua. Ở chế độ bỏ qua, máy mã hóa bỏ qua 1260 thực hiện lập mã số học

bằng cách sử dụng mô hình xác suất cố định, ví dụ, sử dụng kỹ thuật lập mã Golomb-Rice hoặc Golomb lũy thừa. Chế độ bỏ qua thường được dùng cho các phần tử cú pháp để dự đoán hơn.

lập mã ở chế độ thông thường bao gồm bước thực hiện kỹ thuật CABAC. CABAC ở chế độ thông thường dùng để lập mã các giá trị bin mà trong đó xác suất của giá trị bin có thể dự đoán được nếu biết các giá trị của các bin lập mã trước đó. Xác suất của bin là LPS được xác định bởi bộ mô hình hóa ngữ cảnh 1220. Bộ mô hình hóa ngữ cảnh 1220 xuất giá trị bin và trạng thái xác suất cho ngữ cảnh (ví dụ, trạng thái xác suất σ , bao gồm giá trị của LPS và xác suất của việc xuất hiện LPS). Mô hình ngữ cảnh có thể là mô hình ngữ cảnh ban đầu cho một loạt các bin, hoặc có thể được xác định dựa trên các giá trị lập mã của các bin lập mã trước đó. Như đã mô tả trên đây, bộ mô hình hóa ngữ cảnh 1220 có thể cập nhật trạng thái dựa vào việc bin thu được có phải là MPS hoặc LPS hay không. Sau khi trạng thái xác suất và ngữ cảnh σ được xác định bởi bộ mô hình hóa ngữ cảnh 1220, máy mã hóa thông thường 1240 thực hiện kỹ thuật BAC trên giá trị bin.

Quay trở lại Fig.5, bộ mã hóa video 200 có thể xuất ra dòng bit bao gồm các phần tử cú pháp mã hóa entropy cần thiết để tái tạo các khối của lát hoặc ảnh. Cụ thể, đơn vị mã hóa entropy 220 có thể xuất ra dòng bit

Các hoạt động mô tả ở trên được mô tả liên quan đến khối. Sự mô tả như vậy nên được hiểu là các hoạt động đối với khối lập mã độ chói và/hoặc các khối lập mã sắc độ. Như được mô tả trên đây, trong một số ví dụ, khối lập mã độ chói và các khối lập mã sắc độ là các thành phần độ chói và sắc độ của CU. Trong một số ví dụ, khối lập mã độ chói và các khối lập mã sắc độ là các thành phần độ chói và sắc độ của PU.

Trong một số ví dụ, các hoạt động được thực hiện đối với khối lập mã độ chói không cần phải lặp lại đối với các khối lập mã sắc độ. Ví dụ, các thao tác để xác định vectơ chuyển động (MV) và ảnh tham chiếu cho khối lập mã độ chói không cần phải lặp lại để xác định MV và ảnh tham chiếu cho các khối sắc độ. Thay vào đó, MV cho khối lập mã độ chói có thể được định tỷ lệ để xác định MV cho các khối sắc độ, và ảnh tham chiếu có thể giống nhau. Theo ví dụ khác, quy trình dự đoán nội ảnh có thể giống nhau đối với các khối lập mã độ chói và các khối lập mã sắc độ.

Bộ mã hóa video 200 là một ví dụ về thiết bị được tạo cấu hình để mã hóa dữ liệu video bao gồm bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video, và một hoặc nhiều

đơn vị xử lý được triển khai trong mạch và được tạo cấu hình để thu được chỉ số độ dốc định trước (ví dụ, SlopeIdx) và chỉ số độ lệch định trước (ví dụ, OffsetIdx) cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp trong lát dữ liệu video; xác định, dựa vào chỉ số độ dốc định trước và chỉ số độ lệch định trước, trạng thái xác suất ban đầu của ngữ cảnh cho lát dữ liệu video; và mã hóa entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Fig.7 là sơ đồ khối minh họa ví dụ về bộ giải mã video 300 có thể thực hiện các kỹ thuật của sáng chế. Fig.7 được đưa ra nhằm giải thích và không giới hạn ở các kỹ thuật như được minh họa và mô tả rộng rãi trong sáng chế. Để cho mục đích giải thích, sáng chế mô tả bộ giải mã video 300 theo các kỹ thuật của chuẩn JEM, VVC, và HEVC. Tuy nhiên, các kỹ thuật của sáng chế này có thể được thực hiện bởi các thiết bị lập mã video mà được tạo cấu hình theo các chuẩn lập mã video khác.

Trong ví dụ trên Fig.7, bộ giải mã video 300 bao gồm bộ đệm ảnh được lập mã (coded picture buffer - CPB) 320, đơn vị giải mã entropy 302, đơn vị xử lý dự đoán 304, đơn vị lượng tử hóa ngược 306, đơn vị xử lý biến đổi ngược 308, đơn vị tái tạo 310, đơn vị lọc 312 và bộ đệm ảnh được giải mã (DPB) 314. Bất kỳ hoặc tất cả bộ nhớ CPB 320, đơn vị giải mã entropy 302, đơn vị xử lý dự đoán 304, đơn vị lượng tử hóa ngược 306, đơn vị xử lý biến đổi ngược 308, đơn vị tái tạo 310, đơn vị lọc 312, và DPB 314 có thể được triển khai trong một hoặc nhiều bộ xử lý hoặc trong bảng mạch xử lý. Hơn nữa, bộ giải mã video 300 có thể bao gồm bộ xử lý hoặc bảng mạch xử lý bổ sung hoặc thay thế để thực hiện các chức năng này và các chức năng khác.

Đơn vị xử lý dự đoán 304 bao gồm đơn vị bù chuyển động 316 và đơn vị dự đoán nội ảnh 318. Đơn vị xử lý dự đoán 304 có thể bao gồm các đơn vị bổ sung để thực hiện dự đoán theo các chế độ dự đoán khác. Ví dụ, đơn vị xử lý dự đoán 304 có thể bao gồm đơn vị bảng màu, đơn vị sao chép khối nội ảnh (có thể tạo thành một phần của đơn vị bù chuyển động 316), đơn vị afin, đơn vị mô hình tuyến tính (linear model - LM), hoặc tương tự. Trong các ví dụ khác, bộ giải mã dữ liệu video 300 có thể bao gồm nhiều thành phần chức năng hơn, ít thành phần chức năng hơn hoặc các thành phần chức năng khác nhau.

Bộ nhớ CPB 320 có thể lưu trữ dữ liệu video, ví dụ như dòng bit video được mã hóa, cần được giải mã bởi các thành phần của bộ giải mã video 300. Dữ liệu video lưu

trữ trong bộ nhớ CPB 320 có thể thu được, ví dụ, từ phương tiện đọc được bằng máy tính 110 (Fig.1). Bộ nhớ CPB 320 có thể bao gồm CPB lưu trữ dữ liệu video được mã hóa (ví dụ, các phần tử cú pháp) từ dòng bit video được mã hóa. Ngoài ra, bộ nhớ CPB 320 có thể lưu trữ dữ liệu video ngoài các phần tử cú pháp của ảnh được lập mã, chẳng hạn như dữ liệu tạm thời biểu diễn các đầu ra từ các đơn vị khác nhau của bộ giải mã video 300. DPB 314 thường lưu trữ các ảnh đã giải mã mà bộ giải mã video 300 có thể xuất ra và/hoặc sử dụng làm dữ liệu video tham chiếu khi giải mã dữ liệu hoặc các ảnh tiếp theo của dòng bit video được mã hóa. Bộ nhớ CPB 320 và DPB 314 có thể được tạo bởi thiết bị bất kỳ trong số nhiều thiết bị bộ nhớ, chẳng hạn như bộ nhớ truy cập ngẫu nhiên động (DRAM), bao gồm DRAM đồng bộ (SDRAM), RAM từ trở (MRAM), RAM điện trở (RRAM), hoặc các loại thiết bị bộ nhớ khác. Bộ nhớ CPB 320 và DPB 314 có thể được tạo ra bởi cùng một thiết bị nhớ hoặc các thiết bị nhớ riêng. Trong các ví dụ khác nhau, bộ nhớ CPB 320 có thể nằm trên chip cùng các thành phần khác của bộ giải mã video 300, hoặc không nằm trên chip so với các thành phần đó.

Ngoài ra hoặc theo cách khác, trong một số ví dụ, bộ giải mã video 300 có thể truy hồi dữ liệu video được lập mã từ bộ nhớ 120 (Fig.1). Tức là, bộ nhớ 120 có thể lưu trữ dữ liệu như mô tả ở trên với bộ nhớ CPB 320. Tương tự, bộ nhớ 120 có thể lưu trữ các lệnh cần được thực thi bởi bộ giải mã video 300, khi một số hoặc tất cả các chức năng của bộ giải mã video 300 được thực hiện trong phần mềm được thực thi bằng mạch xử lý của bộ giải mã video 300.

Các đơn vị khác nhau được thể hiện trên Fig.7 được minh họa để hỗ trợ hiệu các hoạt động được thực hiện bởi bộ giải mã video 300. Các đơn vị có thể được triển khai dưới dạng các mạch chức năng cố định, các mạch lập trình được, hoặc kết hợp của chúng. Tương tự với Fig.5, các mạch chức năng cố định là các mạch có chức năng cụ thể, và được cài đặt trước trong các hoạt động mà có thể được thực hiện. Các mạch lập trình được là các mạch mà có thể được lập trình để thực hiện một số nhiệm vụ khác nhau, và cung cấp chức năng linh hoạt trong các hoạt động mà có thể được thực hiện. Ví dụ, các mạch lập trình được có thể thực thi phần mềm hoặc firmware khiến cho các mạch lập trình được hoạt động theo cách thức được xác định bởi các lệnh của phần mềm hoặc firmware. Các mạch chức năng cố định có thể thực thi các lệnh phần mềm (ví dụ, để thu các tham số hoặc xuất ra các tham số), nhưng các kiểu hoạt động mà các mạch chức năng cố định thực hiện nói chung là không thể thay đổi. Theo một số ví dụ, một

hoặc nhiều trong số các đơn vị có thể là các khối mạch riêng (chức năng cố định hoặc có thể lập trình được), và theo một số ví dụ, một hoặc nhiều đơn vị có thể là các mạch được tích hợp.

Bộ giải mã video 300 có thể bao gồm các ALU, EFU, mạch kỹ thuật số, mạch tương tự và/hoặc các lõi lập trình được tạo ra từ mạch có thể lập trình. Trong các ví dụ mà các hoạt động của bộ giải mã video 300 được thực hiện bởi phần mềm thực thi trên các mạch lập trình được, bộ nhớ trên chip hoặc ngoài chip có thể lưu trữ các lệnh (ví dụ, mã đối tượng) của phần mềm mà bộ giải mã video 300 nhận và thực thi.

Đơn vị giải mã entropy 302 có thể nhận dữ liệu video được mã hóa từ CPB và giải mã entropy dữ liệu video để tái tạo các phần tử cú pháp. Đơn vị xử lý dự đoán 304, đơn vị lượng tử hóa ngược 306, đơn vị xử lý biến đổi ngược 308, đơn vị tái tạo 310, và đơn vị lọc 312 có thể tạo ra dữ liệu video đã giải mã dựa vào các phần tử cú pháp trích xuất từ dòng bit.

Fig.8 là sơ đồ khối của đơn vị giải mã entropy ví dụ 302 mà có thể được tạo cấu hình để thực hiện kỹ thuật CABAC theo các kỹ thuật của sáng chế. Đơn vị giải mã entropy 302 trên Fig.8 thực hiện kỹ thuật CABAC theo cách ngược với cách của đơn vị mã hóa entropy 220 được mô tả trên Fig.6. Các bit lập mã từ dòng bit 2180 được đưa vào đơn vị giải mã entropy 302. Các bit lập mã được nạp vào bộ mô hình hóa ngữ cảnh 2200 hoặc máy giải mã bỏ qua 2220 dựa vào việc chúng đã được lập mã entropy bằng cách sử dụng chế độ bỏ qua hay chế độ thông thường. Nếu các bit lập mã được lập mã ở chế độ bỏ qua, thì máy giải mã bỏ qua sẽ sử dụng kỹ thuật giải mã Golomb-Rice hoặc kỹ thuật giải mã Golomb lũy thừa, ví dụ, để truy hồi các phần tử cú pháp có giá trị nhị phân hoặc các bin chứa các phần tử cú pháp không phải nhị phân.

Nếu các bit lập mã được lập mã ở chế độ thông thường, thì bộ mô hình hóa ngữ cảnh 2200 có thể xác định mô hình xác suất cho các bit lập mã và máy giải mã thông thường 2240 có thể giải mã các bit lập mã để tạo ra các bin chứa các phần tử cú pháp không có giá trị nhị phân (hoặc chính các phần tử cú pháp nếu có giá trị nhị phân). Sau khi trạng thái xác suất và ngữ cảnh σ được xác định bởi bộ mô hình hóa ngữ cảnh 2200, máy giải mã thông thường 2240 thực hiện kỹ thuật BAC để giải mã giá trị bin. Nói cách khác, máy giải mã thông thường 2240 có thể xác định trạng thái xác suất của ngữ cảnh, và giải mã giá trị bin dựa vào các bin được lập mã trước đó và phạm vi hiện thời. Sau

khi giải mã bin, bộ mô hình hóa ngữ cảnh 2200 có thể cập nhật trạng thái xác suất của ngữ cảnh dựa vào kích thước cửa sổ và giá trị của bin được giải mã.

Quay trở lại Fig.7, nói chung, bộ giải mã video 300 tái tạo ảnh trên cơ sở từng khối. Bộ giải mã video 300 có thể thực hiện thao tác tái tạo trên từng khối riêng (trong đó khối hiện đang được tái tạo, tức là được giải mã, có thể được gọi là “khối hiện thời”).

Đơn vị giải mã entropy 302 có thể giải mã entropy các phần tử cú pháp xác định các hệ số biến đổi được lượng tử hóa của khối hệ số biến đổi được lượng tử hóa, cũng như thông tin biến đổi, như tham số lượng tử hóa (quantization parameter - QP) và/hoặc (các) chỉ báo chế độ biến đổi. Đơn vị lượng tử hóa ngược 306 có thể sử dụng QP liên quan đến khối hệ số biến đổi được lượng tử hóa để xác định mức lượng tử hóa và, tương tự, mức lượng tử hóa ngược cho đơn vị lượng tử hóa ngược 306 để áp dụng. Ví dụ, đơn vị lượng tử hóa ngược 306 có thể thực hiện phép dịch trái theo từng bit để lượng tử hóa ngược các hệ số biến đổi được lượng tử hóa. Nhờ đó, đơn vị lượng tử hóa ngược 306 có thể tạo ra khối hệ số biến đổi bao gồm các hệ số biến đổi.

Sau khi đơn vị lượng tử hóa ngược 306 tạo ra khối hệ số biến đổi, đơn vị xử lý biến đổi ngược 308 có thể áp dụng một hoặc nhiều phép biến đổi ngược cho khối hệ số biến đổi để tạo ra khối dự liên quan đến khối hiện thời. Ví dụ, đơn vị xử lý biến đổi ngược 308 có thể áp dụng phép DCT ngược, biến đổi số nguyên ngược, biến đổi Karhunen-Loeve (Karhunen-Loeve transform - KLT) ngược, biến đổi quay ngược, biến đổi hướng ngược hoặc một biến đổi ngược khác cho khối hệ số biến đổi.

Hơn nữa, đơn vị xử lý dự đoán 304 tạo ra khối dự đoán theo các phần tử cú pháp thông tin dự đoán đã được giải mã entropy bởi đơn vị giải mã entropy 302. Ví dụ, nếu các phần tử cú pháp thông tin dự đoán biểu thị rằng khối hiện thời được dự đoán liên ảnh, thì đơn vị bù chuyển động 316 có thể tạo ra khối dự đoán. Trong trường hợp này, các phần tử cú pháp thông tin dự đoán có thể biểu thị ảnh tham chiếu trong DPB 314 mà truy hồi khối tham chiếu từ đó, cũng như vector chuyển động xác định vị trí của khối tham chiếu trong ảnh tham chiếu liên quan đến vị trí của khối hiện thời trong ảnh hiện thời. Đơn vị bù chuyển động 316 có thể thường thực hiện quy trình dự đoán liên ảnh theo cách thức về cơ bản tương tự như quy trình được mô tả liên quan đến đơn vị bù chuyển động 224 (Fig.5).

Theo ví dụ khác, nếu các phần tử cú pháp thông tin dự đoán biểu thị rằng khối hiện thời được dự đoán nội ảnh, thì đơn vị dự đoán nội ảnh 318 có thể tạo ra khối dự

đoán theo chế độ dự đoán nội ảnh được biểu thị bởi các phần tử cú pháp thông tin dự đoán. Mặt khác, đơn vị dự đoán nội ảnh 318 có thể thường thực hiện quy trình dự đoán nội ảnh theo cách thức về cơ bản tương tự như quy trình được mô tả liên quan đến đơn vị dự đoán nội ảnh 226 (Fig.5). Đơn vị dự đoán nội ảnh 318 có thể truy hồi dữ liệu của các mẫu lân cận cho khối hiện thời từ DPB 314.

Đơn vị tái tạo 310 có thể tái tạo khối hiện thời bằng cách sử dụng khối dự đoán và khối dư. Ví dụ, đơn vị tái tạo 310 có thể thêm các mẫu của khối dư vào các mẫu tương ứng của khối dự đoán để tái tạo khối hiện thời.

Đơn vị lọc 312 có thể thực hiện một hoặc nhiều hoạt động lọc trên các khối được tái tạo. Ví dụ, đơn vị lọc 312 có thể thực hiện các hoạt động tách khối để giảm các thành phần lạ dạng khối dọc theo các cạnh của khối được tái tạo. Các hoạt động của đơn vị lọc 312 không cần thiết được thực hiện trong tất cả các ví dụ.

Bộ giải mã video 300 có thể lưu trữ các khối được tái tạo vào DPB 314. Ví dụ, trong các ví dụ mà không thực hiện hoạt động của đơn vị lọc 312, đơn vị tái tạo 310 có thể lưu trữ các khối được tái tạo vào DPB 314. Trong các ví dụ mà thực hiện các hoạt động của đơn vị lọc 312, đơn vị tái tạo 312 có thể lưu trữ các khối được tái tạo và lọc vào DPB 314. Như đã mô tả ở trên, DPB 314 có thể cung cấp thông tin tham chiếu cho đơn vị xử lý dự đoán 304, chẳng hạn như các mẫu của ảnh hiện thời để dự đoán nội ảnh và các ảnh đã được giải mã trước đó để bù chuyển động tiếp theo. Hơn nữa, bộ giải mã video 300 có thể xuất ra ảnh được giải mã từ DPB để trình diễn sau trên thiết bị hiển thị, chẳng hạn như thiết bị hiển thị 118 trên Fig.1.

Theo cách này, bộ giải mã video 300 biểu diễn ví dụ về thiết bị giải mã video bao gồm bộ nhớ được tạo cấu hình để lưu trữ dữ liệu video, và một hoặc nhiều đơn vị xử lý được triển khai trong mạch và được tạo cấu hình để thu được chỉ số độ dốc định trước (ví dụ, SlopeIdx) và chỉ số độ lệch định trước (ví dụ, OffsetIdx) cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp trong lát dữ liệu video; xác định, dựa vào chỉ số độ dốc định trước và chỉ số độ lệch định trước, trạng thái xác suất ban đầu của ngữ cảnh cho lát dữ liệu video; và giải mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Fig.19 là lưu đồ minh họa một ví dụ về phương pháp mã hóa khối hiện thời. Khối hiện thời có thể bao gồm CU hiện thời. Mặc dù được mô tả liên quan đến bộ mã hóa

video 200 (Fig.1 và Fig.5), nhưng cần hiểu rằng các thiết bị khác có thể được tạo cấu hình để thực hiện phương pháp tương tự như phương pháp trên Fig.19.

Trong ví dụ này, bộ mã hóa video 200 ban đầu dự đoán khối hiện thời (350). Ví dụ, bộ mã hóa video 200 có thể tạo ra khối dự đoán cho khối hiện thời. Sau đó, bộ mã hóa video 200 có thể tính toán khối dư cho khối hiện thời (352). Để tính toán khối dư, bộ mã hóa video 200 có thể tính toán sự chênh lệch giữa khối gốc, khối chưa lập mã và khối dự đoán cho khối hiện thời. Sau đó, bộ mã hóa video 200 có thể biến đổi và lượng tử hóa các hệ số của khối dư (354). Tiếp theo, bộ mã hóa video 200 có thể quét các hệ số biến đổi được lượng tử hóa của khối dư (356). Trong khi quét, hoặc sau khi quét, bộ mã hóa video 200 có thể mã hóa entropy các hệ số (358). Ví dụ, bộ mã hóa video 200 có thể mã hóa các hệ số, và/hoặc các phần tử cú pháp khác, sử dụng CAVLC hoặc CABAC bằng cách sử dụng các kỹ thuật khởi tạo xác suất nêu trên và theo Fig.20. Sau đó, bộ mã hóa video 200 có thể xuất ra dữ liệu lập mã entropy của khối (360).

Fig.20 là lưu đồ minh họa một ví dụ về quy trình thực hiện mã hóa entropy dựa vào ngữ cảnh, theo một hoặc nhiều kỹ thuật của sáng chế. Các kỹ thuật trên Fig.20 có thể được thực hiện bởi bộ mã hóa video, ví dụ như bộ mã hóa video 200 minh họa trên Fig.1, Fig.5 và Fig.6. Nhằm mục đích minh họa, các kỹ thuật trên Fig.20 được mô tả trong ngữ cảnh của bộ mã hóa video 200 trên Fig.1, Fig.5 và Fig.6, tuy nhiên, các bộ mã hóa video có cấu hình khác với bộ mã hóa video 200 có thể thực hiện các kỹ thuật trên Fig.20.

Bộ mã hóa video 200 có thể thu được chuỗi bin (ví dụ, vector nhị phân một chiều) cần mã hóa bằng cách sử dụng lập mã entropy dựa vào ngữ cảnh (ví dụ, CABAC) (2002). Ví dụ, đơn vị mã hóa entropy 220 của bộ mã hóa video 200 có thể thu được chuỗi bin bằng cách nhị phân hóa phần tử cú pháp thu được từ đơn vị lựa chọn chế độ 202 của bộ mã hóa video 200.

Bộ mã hóa video 200 có thể thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh (2004). Ví dụ, đơn vị mã hóa entropy 220 của bộ mã hóa video 200 có thể thu được giá trị initValue, có thể là biến số 6 bit.

Bộ mã hóa video 200 có thể xác định, dựa vào giá trị khởi tạo định trước, trạng thái xác suất ban đầu của ngữ cảnh cho đơn vị có thể lập mã độc lập (ví dụ, lát, ô, v.v.) của dữ liệu video trong miền tuyến tính (2006). Ví dụ, đơn vị mã hóa entropy 220 có thể xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác

định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit. Trong một số ví dụ, đơn vị mã hóa entropy 220 có thể xác định trạng thái xác suất ban đầu mà không sử dụng LUT để chuyển đổi trạng thái xác suất ban đầu từ miền lôgarit đến miền tuyến tính.

Để xác định trạng thái xác suất ban đầu, đơn vị mã hóa entropy 220 có thể thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập (ví dụ, SliceQP_Y), và thu được giá trị của điểm neo tham số lượng tử hóa khác không (ví dụ, $\text{QP}_{\text{anchor}}$). Đơn vị mã hóa entropy 220 có thể xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa cho lát và giá trị của điểm neo tham số lượng tử hóa. Ví dụ, đơn vị mã hóa entropy 220 có thể xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg \text{rshift}) + n.$$

trong đó InitProbState là trạng thái xác suất ban đầu, SliceQP_Y là giá trị ban đầu của tham số lượng tử hóa, $\text{QP}_{\text{anchor}}$ là điểm neo tham số lượng tử hóa, và rshift là giá trị dịch phải.

Như đã nêu trên, trong một số ví dụ, đơn vị mã hóa entropy 220 có thể được tạo cấu hình để thực hiện quá trình khởi tạo có độ chính xác tăng đối với các phân phối xác suất cực trị. Ví dụ, đơn vị mã hóa entropy 220 có thể xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch. Đơn vị mã hóa entropy 220 có thể xác định, dựa vào giá trị chỉ số độ dốc, giá trị của m ; và xác định, dựa vào giá trị chỉ số độ lệch, giá trị của n . Trong một số ví dụ, đơn vị mã hóa entropy 220 có thể xác định giá trị của n , theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch. Như được thể hiện trên đây, trong một số ví dụ, đơn vị mã hóa entropy 220 có thể xác định trạng thái xác suất ban đầu dựa vào giá trị của m và giá trị của n .

Bộ mã hóa video 200 có thể mã hóa, trong dòng bit video và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, một bin trong chuỗi bin (2008). Ví dụ, đơn vị mã hóa entropy 220 có thể xuất dòng nhị phân mà biểu diễn giá trị hoặc con trỏ đến xác suất trong khoảng xác suất lập mã cuối cùng của ngữ cảnh.

Fig.21 là lưu đồ minh họa ví dụ về phương pháp giải mã khối dữ liệu video hiện thời. Khối hiện thời có thể bao gồm CU hiện thời. Mặc dù được mô tả liên quan đến bộ

giải mã video 300 (Fig.1 và Fig.7), nhưng cần hiểu rằng các thiết bị khác có thể được tạo cấu hình để thực hiện phương pháp giống với phương pháp trên Fig.21.

Bộ giải mã video 300 có thể thu dữ liệu được lập mã entropy cho khối hiện thời, như thông tin dự đoán đã lập mã entropy và dữ liệu được lập mã entropy cho các hệ số của khối dư tương ứng với khối hiện thời (370). Bộ giải mã video 300 có thể giải mã entropy dữ liệu đã được lập mã entropy để xác định thông tin dự đoán cho khối hiện thời và để tái tạo các hệ số của khối dư (372). Ví dụ, bộ giải mã video 300 có thể giải mã các hệ số, và/hoặc các phần tử cú pháp khác, sử dụng CAVLC hoặc CABAC bằng cách sử dụng các kỹ thuật khởi tạo xác suất nêu trên và theo Fig.22. Bộ giải mã video 300 có thể dự đoán khối hiện thời (374), ví dụ, bằng cách sử dụng chế độ dự đoán nội ảnh hoặc liên ảnh như được chỉ báo bởi thông tin dự đoán cho khối hiện thời, để tính toán khối dự đoán cho khối hiện thời. Sau đó, bộ giải mã video 300 có thể quét ngược các hệ số được tái tạo (376), để tạo ra khối hệ số biến đổi được lượng tử hóa. Sau đó, bộ giải mã video 300 có thể lượng tử hóa ngược và biến đổi ngược các hệ số để tạo ra khối dư (378). Cuối cùng, bộ giải mã video 300 có thể giải mã khối hiện thời bằng cách kết hợp khối dự đoán và khối dư (380).

Fig.22 là lưu đồ minh họa một ví dụ về quy trình thực hiện giải mã entropy dựa vào ngữ cảnh, theo một hoặc nhiều kỹ thuật của sáng chế. Các kỹ thuật trên Fig.22 có thể được thực hiện bởi bộ giải mã video, ví dụ như bộ giải mã video 300 được minh họa trên Fig.1, Fig.7 và Fig.8. Nhằm mục đích minh họa, các kỹ thuật trên Fig.22 được mô tả trong ngữ cảnh của bộ giải mã video 300 trên Fig.1, Fig.7 và Fig.8, mặc dù các bộ giải mã video có cấu hình khác với bộ giải mã video 300 có thể thực hiện các kỹ thuật trên Fig.22.

Bộ giải mã video 300 có thể thu được, từ dòng bit video, chuỗi bin (ví dụ, vector nhị phân một chiều) cần được giải mã bằng cách sử dụng lập mã entropy dựa vào ngữ cảnh (2202). Ví dụ, đơn vị giải mã entropy 302 của bộ giải mã video 300 có thể thu được, từ bộ nhớ CPB 320, chuỗi bin. Trong một số ví dụ, chuỗi bin có thể biểu diễn giá trị hoặc con trỏ đến xác suất trong khoảng xác suất lập mã cuối cùng của ngữ cảnh. Trong một số ví dụ, lập mã entropy dựa vào ngữ cảnh có thể bao gồm lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC).

Bộ giải mã video 300 có thể thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh (2204). Ví dụ, đơn vị giải mã entropy 302 của bộ giải mã video 300 có thể thu được giá trị `initValue`, có thể là biến số 6 bit.

Bộ giải mã video 300 có thể xác định, dựa vào giá trị khởi tạo định trước, trạng thái xác suất ban đầu của ngữ cảnh cho đơn vị có thể lập mã độc lập của dữ liệu video (ví dụ, lát, ô, v.v.) trong miền tuyến tính (2206). Ví dụ, đơn vị giải mã entropy 302 có thể xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit. Trong một số ví dụ, đơn vị giải mã entropy 302 có thể xác định trạng thái xác suất ban đầu mà không sử dụng LUT để chuyển đổi trạng thái xác suất ban đầu từ miền lôgarit sang miền tuyến tính.

Để xác định trạng thái xác suất ban đầu, đơn vị giải mã entropy 302 có thể thu được giá trị ban đầu của tham số lượng tử hóa cho đơn vị có thể lập mã độc lập (ví dụ, `SliceQPY`), và thu được giá trị của điểm neo tham số lượng tử hóa khác không (ví dụ, `QPanchor`). Đơn vị giải mã entropy 302 có thể xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa cho đơn vị có thể lập mã độc lập và giá trị của điểm neo tham số lượng tử hóa. Ví dụ, đơn vị giải mã entropy 302 có thể xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg \text{rshift}) + n.$$

trong đó `InitProbState` là trạng thái xác suất ban đầu, `SliceQPY` là giá trị ban đầu của tham số lượng tử hóa, `QPanchor` là điểm neo tham số lượng tử hóa, và `rshift` là giá trị dịch phải.

Như đã nêu trên, trong một số ví dụ, đơn vị giải mã entropy 302 có thể được tạo cấu hình để thực hiện quá trình khởi tạo có độ chính xác tăng đối với các phân phối xác suất cực trị. Ví dụ, đơn vị giải mã entropy 302 có thể xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch. Đơn vị giải mã entropy 302 có thể xác định, dựa vào giá trị chỉ số độ dốc, giá trị của m ; và xác định, dựa vào giá trị chỉ số độ lệch, giá trị của n . Trong một số ví dụ, đơn vị giải mã entropy 302 có thể xác định giá trị của n , theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó `OffsetIdx` là giá trị chỉ số độ lệch. Như được thể hiện trên đây, trong một số ví dụ, đơn vị giải mã entropy 302 có thể xác định trạng thái xác suất ban đầu dựa vào giá trị của m và giá trị của n .

Bộ giải mã video 300 có thể giải mã, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, một bin trong chuỗi bin (2208). Bộ giải mã video 300 có thể xác định, dựa vào bin được giải mã và trạng thái xác suất ban đầu của ngữ cảnh, trạng thái xác suất được cập nhật của ngữ cảnh. Bộ giải mã video 300 có thể giải mã bin khác dựa vào trạng thái xác suất được cập nhật của ngữ cảnh (2206).

Các ví dụ được đánh số sau đây có thể minh họa một hoặc nhiều khía cạnh của sáng chế:

Ví dụ 1A. Phương pháp lập mã entropy dữ liệu video, phương pháp này bao gồm các bước: thu được chỉ số độ dốc định trước và chỉ số độ lệch định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp trong lát dữ liệu video; xác định, dựa vào chỉ số độ dốc định trước và chỉ số độ lệch định trước, trạng thái xác suất ban đầu của ngữ cảnh cho lát dữ liệu video; và lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Ví dụ 2A. Phương pháp theo ví dụ 1A, trong đó trạng thái xác suất ban đầu biểu diễn xác suất ban đầu trong miền tuyến tính.

Ví dụ 3A. Phương pháp theo ví dụ 1A, trong đó trạng thái xác suất ban đầu biểu diễn xác suất ban đầu trong miền bậc hai.

Ví dụ 4A. Phương pháp theo ví dụ 3A, trong đó bước xác định trạng thái xác suất ban đầu được thực hiện mà không sử dụng bảng tìm kiếm (LUT) để ánh xạ giữa chỉ số độ dốc định trước, chỉ số độ lệch định trước, và trạng thái xác suất ban đầu.

Ví dụ 5A. Phương pháp theo sự kết hợp bất kỳ giữa các ví dụ từ 1A–4A, trong đó trạng thái xác suất ban đầu được biểu diễn bởi `InitProbState`, chỉ số độ dốc được biểu diễn bởi `SlopeIdx`, và chỉ số độ lệch được biểu diễn bởi `OffsetIdx`.

Ví dụ 6A. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1A–5A, trong đó quy trình lập mã entropy thích ứng theo ngữ cảnh bao gồm quy trình lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC), hoặc quy trình lập mã có độ dài thay đổi thích ứng theo ngữ cảnh (CAVLC).

Ví dụ 7A. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1A–6A, trong đó bước lập mã bao gồm bước giải mã.

Ví dụ 8A. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1A–7A, trong đó bước lập mã bao gồm bước mã hóa.

Ví dụ 9A. Thiết bị lập mã dữ liệu video, thiết bị này bao gồm một hoặc nhiều phương tiện để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1A–8A.

Ví dụ 10A. Thiết bị theo ví dụ 9A, trong đó một hoặc nhiều phương tiện bao gồm một hoặc nhiều bộ xử lý được triển khai trong mạch.

Ví dụ 11A. Thiết bị theo ví dụ bất kỳ trong số các ví dụ 9A và 10A, trong đó thiết bị này còn bao gồm bộ nhớ để lưu trữ dữ liệu video.

Ví dụ 12A. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9A–11A, thiết bị này còn bao gồm màn hình được tạo cấu hình để hiển thị dữ liệu video được giải mã.

Ví dụ 13A. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9A–12A, trong đó thiết bị này bao gồm một hoặc nhiều trong số máy ảnh, máy tính, thiết bị di động, thiết bị thu phát rộng, hoặc hộp giải mã tín hiệu.

Ví dụ 14A. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9A–13A, trong đó thiết bị này bao gồm bộ giải mã video.

Ví dụ 15A. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9A–14A, trong đó thiết bị này bao gồm bộ mã hóa video.

Ví dụ 16A. Phương tiện lưu trữ đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi sẽ khiến cho một hoặc nhiều bộ xử lý thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1A–8A.

Ví dụ 1B. Phương pháp lập mã entropy dữ liệu video, phương pháp này bao gồm các bước: thu được chỉ số độ lệch định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp trong lát dữ liệu video; xác định, dựa vào chỉ số độ lệch định trước, trạng thái xác suất ban đầu của ngữ cảnh cho lát dữ liệu video; và lập mã entropy, dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp này.

Ví dụ 2B. Phương pháp theo ví dụ 1B, trong đó trạng thái xác suất ban đầu biểu diễn xác suất ban đầu trong miền tuyến tính.

Ví dụ 3B. Phương pháp theo ví dụ 1B, trong đó trạng thái xác suất ban đầu biểu diễn xác suất ban đầu trong miền bậc hai.

Ví dụ 4B. Phương pháp theo ví dụ 3B, trong đó bước xác định trạng thái xác suất ban đầu được thực hiện mà không sử dụng bảng tìm kiếm (LUT) để ánh xạ giữa chỉ số độ lệch định trước và trạng thái xác suất ban đầu.

Ví dụ 5B. Phương pháp theo sự kết hợp bất kỳ giữa các ví dụ từ 1B–4B, trong đó trạng thái xác suất ban đầu được biểu diễn bởi `InitProbState`, và chỉ số độ lệch được biểu diễn bởi `OffsetIdx`.

Ví dụ 6B. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1B–5B, trong đó quy trình lập mã entropy thích ứng theo ngữ cảnh bao gồm quy trình lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC), hoặc quy trình lập mã có độ dài thay đổi thích ứng theo ngữ cảnh (CAVLC).

Ví dụ 7B. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1B–6B, trong đó bước lập mã bao gồm bước giải mã.

Ví dụ 8B. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1B–7B, trong đó bước lập mã bao gồm bước mã hóa.

Ví dụ 9B. Thiết bị lập mã dữ liệu video, thiết bị này bao gồm một hoặc nhiều phương tiện để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1B–8B.

Ví dụ 10B. Thiết bị theo ví dụ 9B, trong đó một hoặc nhiều phương tiện bao gồm một hoặc nhiều bộ xử lý được triển khai trong mạch.

Ví dụ 11B. Thiết bị theo ví dụ bất kỳ trong số các ví dụ 9B và 10B, trong đó thiết bị này còn bao gồm bộ nhớ để lưu trữ dữ liệu video.

Ví dụ 12B. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9B–11B, thiết bị này còn bao gồm màn hình được tạo cấu hình để hiển thị dữ liệu video được giải mã.

Ví dụ 13B. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9B–12B, trong đó thiết bị này bao gồm một hoặc nhiều trong số máy ảnh, máy tính, thiết bị di động, thiết bị thu phát rộng, hoặc hộp giải mã tín hiệu.

Ví dụ 14B. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9B–13B, trong đó thiết bị này bao gồm bộ giải mã video.

Ví dụ 15B. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 9B–14B, trong đó thiết bị này bao gồm bộ mã hóa video.

Ví dụ 16B. Phương tiện lưu trữ đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi sẽ khiến cho một hoặc nhiều bộ xử lý thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1B–8B.

Ví dụ 1C. Phương pháp lập mã entropy dữ liệu video, phương pháp này bao gồm các bước: xác định, dựa vào giá trị của biến, giá trị trung gian thứ nhất; xác định, dựa vào giá trị trung gian thứ nhất, giá trị của trạng thái xác suất có độ chính xác thấp của

mô hình ngữ cảnh, trong đó bước xác định giá trị của trạng thái xác suất có độ chính xác thấp bao gồm bước dịch phải giá trị trung gian thứ nhất; và lập mã, dựa vào giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh, giá trị của ít nhất một bit của phần tử cú pháp.

Ví dụ 2C. Phương pháp theo ví dụ 1C, trong đó bước xác định giá trị trung gian thứ nhất bao gồm bước xác định giá trị trung gian thứ nhất theo phương trình sau: $q = \text{Clip3}(18, 46, \text{SliceQP}_Y) - 16$ trong đó q là giá trị trung gian thứ nhất, và SliceQP_Y là giá trị của biến số.

Ví dụ 3C. Phương pháp theo ví dụ 1C hoặc ví dụ 2C, trong đó bước xác định giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh bao gồm bước xác định giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh theo phương trình sau: $\text{ProbabilityStateL} = 16 * a + ((b - a) * q \gg 1) + 8$ trong đó q là giá trị trung gian thứ nhất, ProbabilityStateL là giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh, a là giá trị trung gian thứ hai, và b là giá trị trung gian thứ ba.

Ví dụ 4C. Phương pháp theo ví dụ 3C, phương pháp này còn bao gồm bước xác định các giá trị cho a và b theo các phương trình sau: $a = (\text{initValue} \gg 3) * 9$ $b = (\text{initValue} \& 7) * 9$ trong đó initValue là giá trị khởi tạo.

Ví dụ 5C. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–4C, phương pháp này còn bao gồm bước xác định, dựa vào giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh, giá trị của trạng thái xác suất có độ chính xác cao của mô hình ngữ cảnh.

Ví dụ 6C. Phương pháp theo ví dụ 5C, trong đó bước xác định giá trị của trạng thái xác suất có độ chính xác cao của mô hình ngữ cảnh bao gồm bước xác định giá trị của trạng thái xác suất có độ chính xác cao của mô hình ngữ cảnh theo phương trình sau: $\text{ProbabilityStateH} = \text{ProbabilityStateL} \ll 4$ trong đó ProbabilityStateH là giá trị của trạng thái xác suất có độ chính xác cao của mô hình ngữ cảnh, và ProbabilityStateL là giá trị của trạng thái xác suất có độ chính xác thấp của mô hình ngữ cảnh.

Ví dụ 7C. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–6C, trong đó quy trình lập mã entropy thích ứng theo ngữ cảnh bao gồm quy trình lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC), hoặc quy trình lập mã có độ dài thay đổi thích ứng theo ngữ cảnh (CAVLC).

Ví dụ 8C. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–7C, trong đó bước lập mã bao gồm bước giải mã.

Ví dụ 9C. Phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–7C, trong đó bước lập mã bao gồm bước giải mã.

Ví dụ 10C. Thiết bị lập mã dữ liệu video, thiết bị này bao gồm một hoặc nhiều phương tiện để thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–8C.

Ví dụ 11C. Thiết bị theo ví dụ 10C, trong đó một hoặc nhiều phương tiện bao gồm một hoặc nhiều bộ xử lý được triển khai trong mạch.

Ví dụ 12C. Thiết bị theo ví dụ bất kỳ trong số các ví dụ 10C và 11C, trong đó thiết bị này còn bao gồm bộ nhớ để lưu trữ dữ liệu video.

Ví dụ 13C. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 10C–12C, thiết bị này còn bao gồm màn hình được tạo cấu hình để hiển thị dữ liệu video giải mã.

Ví dụ 14C. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 10C–13C, trong đó thiết bị này bao gồm một hoặc nhiều trong số máy ảnh, máy tính, thiết bị di động, thiết bị thu phát rộng, hoặc hộp giải mã tín hiệu.

Ví dụ 15C. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 10C–14C, trong đó thiết bị này bao gồm bộ giải mã video.

Ví dụ 16C. Thiết bị theo ví dụ bất kỳ trong số các ví dụ từ 10C–15C, trong đó thiết bị này bao gồm bộ mã hóa video.

Ví dụ 17C. Phương tiện lưu trữ đọc được bằng máy tính lưu trữ các lệnh mà khi được thực thi sẽ khiến cho một hoặc nhiều bộ xử lý thực hiện phương pháp theo ví dụ bất kỳ trong số các ví dụ từ 1C–9C.

Cần hiểu rằng tùy thuộc vào ví dụ, các hành động hoặc sự kiện nhất định của kỹ thuật bất kỳ được mô tả ở đây có thể được thực hiện theo các trình tự khác nhau, có thể được bổ sung, hợp nhất hoặc được loại bỏ hoàn toàn (ví dụ, không phải tất cả các hành động hoặc sự kiện được mô tả đều cần thiết để thực hiện các kỹ thuật theo sáng chế). Ngoài ra, trong một số ví dụ, các hành động hoặc sự kiện có thể được thực hiện đồng thời, ví dụ, thông qua quy trình xử lý đa chuỗi, quy trình xử lý ngắt, hoặc nhiều bộ xử lý, chứ không phải chỉ có thực hiện tuần tự.

Trong một hoặc nhiều ví dụ, các chức năng đã mô tả có thể được thực hiện dưới dạng phần cứng, phần mềm, firmware, hoặc dạng kết hợp bất kỳ của chúng. Nếu được thực hiện bằng phần mềm, các chức năng có thể được lưu trữ hoặc truyền dưới dạng

một hoặc nhiều lệnh hoặc mã trên phương tiện đọc được bằng máy tính và được thực hiện bằng đơn vị xử lý dựa trên phần cứng. Phương tiện đọc được bằng máy tính có thể bao gồm các phương tiện lưu trữ đọc được bằng máy tính tương ứng với phương tiện hữu hình như phương tiện lưu trữ dữ liệu, hoặc phương tiện truyền thông có phương tiện bất kỳ hỗ trợ truyền chương trình máy tính từ nơi này đến nơi khác, ví dụ, theo giao thức truyền thông. Theo cách này, phương tiện đọc được bằng máy tính có thể thường tương ứng với (1) phương tiện lưu trữ hữu hình đọc được bằng máy tính là phương tiện bất khả biến hoặc (2) phương tiện truyền thông như tín hiệu hoặc sóng mang. Phương tiện lưu trữ dữ liệu có thể là phương tiện khả dụng bất kỳ mà có thể được truy cập bởi một hoặc nhiều máy tính hoặc một hoặc nhiều bộ xử lý để truy hồi các lệnh, mã và/hoặc các cấu trúc dữ liệu để thực hiện các kỹ thuật được mô tả trong sáng chế. Sản phẩm chứa chương trình máy tính có thể bao gồm phương tiện đọc được bằng máy tính.

Ví dụ, và không giới hạn, các phương tiện lưu trữ đọc được bằng máy tính như vậy có thể bao gồm bộ nhớ truy cập ngẫu nhiên (Random Access Memory - RAM), bộ nhớ chỉ đọc (Read Only Memory ROM), bộ nhớ chỉ đọc lập trình được xóa được bằng điện (Electrically Erasable Programmable Read Only Memory - EEPROM), đĩa compac-bộ nhớ chỉ đọc (CD-ROM) hoặc các thiết bị lưu trữ đĩa quang khác, thiết bị lưu trữ đĩa từ, hoặc các thiết bị lưu trữ từ tính khác, bộ nhớ flash, hoặc phương tiện khác bất kỳ mà có thể được sử dụng để lưu trữ mã chương trình mong muốn ở dạng các lệnh hoặc các cấu trúc dữ liệu và có thể được truy cập bởi máy tính. Ngoài ra, mọi dạng kết nối được gọi theo cách thích hợp là phương tiện đọc được bằng máy tính. Ví dụ, nếu các lệnh được truyền từ trang web, máy chủ hoặc nguồn từ xa khác bằng cách sử dụng cáp đồng trục, cáp quang sợi, cáp dây xoắn, đường thuê bao kỹ thuật số (digital subscriber line - DSL) hoặc các công nghệ không dây như hồng ngoại, vô tuyến, và vi sóng, thì cáp đồng trục, cáp quang sợi, cáp dây xoắn, đường DSL hoặc các công nghệ không dây như hồng ngoại, vô tuyến và vi sóng được bao gồm trong định nghĩa về phương tiện. Tuy nhiên, cần phải hiểu rằng phương tiện lưu trữ đọc được bằng máy tính và phương tiện lưu trữ dữ liệu không bao gồm kết nối, sóng mang, tín hiệu hoặc phương tiện chuyển tiếp khác, nhưng thay vào đó được hướng đến phương tiện lưu trữ hữu hình, bất biến. Đĩa từ và đĩa quang, như được sử dụng ở đây, bao gồm đĩa compac (compact disc - CD), đĩa laze, đĩa quang, đĩa số đa năng (DVD - Digital Versatile Disc), đĩa mềm và đĩa bluray, trong đó đĩa từ thường tái tạo dữ liệu bằng phương pháp từ tính, còn đĩa quang thì tái

tạo dữ liệu bằng phương pháp quang học sử dụng laze. Các dạng kết hợp của các loại phương tiện nêu trên cũng được bao gồm trong phạm vi của phương tiện đọc được bằng máy tính.

Các lệnh có thể được thực thi bởi một hoặc nhiều bộ xử lý, chẳng hạn như một hoặc nhiều bộ xử lý tín hiệu số (digital signal processor - DSP), các bộ vi xử lý đa năng, mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), mảng cổng lập trình được theo trường (field programmable gate array- FPGA), hoặc các hệ mạch logic tích hợp hoặc rời rạc tương đương khác. Do vậy, thuật ngữ “bộ xử lý” và “mạch xử lý”, như được sử dụng ở đây có thể chỉ cấu trúc bất kỳ trong số các cấu trúc nêu trên hoặc cấu trúc khác bất kỳ phù hợp để thực hiện các kỹ thuật được mô tả ở đây. Ngoài ra, theo một số khía cạnh, chức năng nêu trong sáng chế có thể được tạo ra trong modul phần cứng và/hoặc phần mềm chuyên dụng được tạo cấu hình để mã hóa và giải mã, hoặc được kết hợp thành bộ mã hoá-giải mã kết hợp. Hơn nữa, các kỹ thuật có thể được thực hiện hoàn toàn trong một hoặc nhiều mạch hoặc phần tử logic.

Các kỹ thuật của sáng chế có thể được triển khai trong nhiều loại thiết bị hoặc máy khác nhau, bao gồm thiết bị cầm tay không dây, mạch tích hợp (integrated circuit - IC) hoặc bộ IC (ví dụ, bộ chip). Một số thành phần, modul hoặc đơn vị được mô tả trong sáng chế để nhấn mạnh các khía cạnh chức năng của các thiết bị được tạo cấu hình để thực hiện các kỹ thuật đã bộc lộ, chứ không nhất thiết phải được thực hiện bằng các đơn vị phần cứng khác nhau. Thay vào đó, như được mô tả ở trên, một số đơn vị có thể được kết hợp trong đơn vị phần cứng codec hoặc được cung cấp bởi tập hợp các đơn vị phần cứng tương tác, bao gồm một hoặc nhiều bộ xử lý như được mô tả ở trên, kết hợp với phần mềm và/hoặc firmware thích hợp.

Một số ví dụ của sáng chế đã được mô tả. Các ví dụ này và các ví dụ khác đều nằm trong phạm vi của các điểm yêu cầu bảo hộ dưới đây.

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã entropy dữ liệu video, phương pháp này bao gồm các bước: truy hồi, từ bộ nhớ, giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m; xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; xác định, dựa vào giá trị khởi tạo định trước, tham số m, tham số n, và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó bước xác định trạng thái xác suất ban đầu còn dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và giải mã entropy, từ dòng bit video và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp.
2. Phương pháp theo điểm 1, trong đó bước xác định trạng thái xác suất ban đầu của ngữ cảnh bao gồm xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit.
3. Phương pháp theo điểm 1, trong đó giá trị ban đầu của tham số lượng tử hóa là SliceQP_Y.
4. Phương pháp theo điểm 1, trong đó giá trị xác định điểm neo tham số lượng tử hóa là 16.
5. Phương pháp theo điểm 1, trong đó bước xác định trạng thái xác suất ban đầu bao gồm xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = \text{clip3}(1, 127, ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg r\text{shift}) + n),$$

trong đó InitProbState là trạng thái xác suất ban đầu, SliceQP_Y là giá trị ban đầu của tham số lượng tử hóa, $\text{QP}_{\text{anchor}}$ là điểm neo tham số lượng tử hóa, và rshift là giá trị dịch phải.

6. Phương pháp theo điểm 1, trong đó bước giải mã entropy bao gồm giải mã entropy bằng cách sử dụng lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC).

7. Thiết bị giải mã video bao gồm: bộ nhớ lưu trữ dữ liệu video; và một hoặc nhiều bộ xử lý được triển khai trong mạch và được tạo cấu hình để: truy hồi, từ bộ nhớ, giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m ; xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; xác định, dựa vào giá trị khởi tạo định trước, tham số m , tham số n , và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó để xác định trạng thái xác suất ban đầu, các bộ xử lý còn được tạo cấu hình để xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và giải mã entropy, từ dòng bit video và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp.

8. Thiết bị theo điểm 7, trong đó, để xác định trạng thái xác suất ban đầu của ngữ cảnh, một hoặc nhiều bộ xử lý được tạo cấu hình để xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit.

9. Thiết bị theo điểm 7, trong đó giá trị ban đầu của tham số lượng tử hóa là SliceQP_Y .

10. Thiết bị theo điểm 7, trong đó giá trị của điểm neo tham số lượng tử hóa là 16.

11. Thiết bị theo điểm 7, trong đó, để xác định trạng thái xác suất ban đầu, một hoặc nhiều bộ xử lý được tạo cấu hình để xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = \text{clip3}(1, 127, ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg r\text{shift}) + n),$$

trong đó InitProbState là trạng thái xác suất ban đầu, SliceQP_Y là giá trị ban đầu của tham số lượng tử hóa, QP_{anchor} là điểm neo tham số lượng tử hóa, và rshift là giá trị dịch phải.

12. Thiết bị theo điểm 7, trong đó, để giải mã entropy bin, một hoặc nhiều bộ xử lý được tạo cấu hình để giải mã bin bằng cách sử dụng lập mã số học nhị phân thích ứng theo ngữ cảnh (CABAC).

13. Phương pháp mã hóa entropy dữ liệu video, phương pháp này bao gồm các bước: truy hồi, từ bộ nhớ, giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phân tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m; xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; xác định, dựa vào giá trị khởi tạo định trước, tham số m, tham số n, và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó bước xác định trạng thái xác suất ban đầu còn dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và mã hóa entropy, trong dòng bit video và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phân tử cú pháp.

14. Phương pháp theo điểm 13, trong đó bước xác định trạng thái xác suất ban đầu của ngữ cảnh bao gồm xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit.

15. Phương pháp theo điểm 13, trong đó giá trị ban đầu của tham số lượng tử hóa là SliceQP_Y .

16. Phương pháp theo điểm 13, trong đó giá trị của điểm neo tham số lượng tử hóa là 16.

17. Phương pháp theo điểm 13, trong đó bước xác định trạng thái xác suất ban đầu bao gồm xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = \text{clip3}(1, 127, ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg r\text{shift}) + n),$$

trong đó InitProbState là trạng thái xác suất ban đầu, SliceQP_Y là giá trị ban đầu của tham số lượng tử hóa, $\text{QP}_{\text{anchor}}$ là điểm neo tham số lượng tử hóa, và $r\text{shift}$ là giá trị dịch phải.

18. Thiết bị mã hóa video bao gồm: bộ nhớ lưu trữ dữ liệu video; và một hoặc nhiều bộ xử lý được triển khai trong mạch và được tạo cấu hình để: truy hồi, từ bộ nhớ, giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phân tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m ; xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; xác định, dựa vào giá trị khởi tạo định trước, tham số m , tham số n , và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó để xác định trạng thái xác suất ban đầu, các bộ xử lý còn được tạo cấu hình để xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và mã hóa entropy, trong dòng bit video và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phân tử cú pháp.

19. Thiết bị theo điểm 18, trong đó, để xác định trạng thái xác suất ban đầu của ngữ cảnh trong miền tuyến tính, một hoặc nhiều bộ xử lý được tạo cấu hình để xác định trạng thái

xác suất ban đầu của ngữ cảnh trong miền tuyến tính mà không xác định trung gian trạng thái xác suất ban đầu của ngữ cảnh trong miền lôgarit.

20. Thiết bị theo điểm 18, trong đó giá trị ban đầu của tham số lượng tử hóa là SliceQP_Y .

21. Thiết bị theo điểm 18, trong đó giá trị của điểm neo tham số lượng tử hóa là 16.

22. Thiết bị theo điểm 18, trong đó, để xác định trạng thái xác suất ban đầu, một hoặc nhiều bộ xử lý được tạo cấu hình để xác định trạng thái xác suất ban đầu theo phương trình sau:

$$\text{InitProbState} = \text{clip3}(1, 127, ((m * (\text{SliceQP}_Y - \text{QP}_{\text{anchor}})) \gg r\text{shift}) + n),$$

trong đó InitProbState là trạng thái xác suất ban đầu, SliceQP_Y là giá trị ban đầu của tham số lượng tử hóa, $\text{QP}_{\text{anchor}}$ là điểm neo tham số lượng tử hóa, và $r\text{shift}$ là giá trị dịch phải.

23. Thiết bị giải mã entropy dữ liệu video bao gồm: phương tiện để thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phân tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; phương tiện để xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; phương tiện để xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m ; phương tiện để xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; phương tiện để thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; phương tiện để xác định, dựa vào giá trị khởi tạo định trước, tham số m , tham số n , và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó phương tiện để xác định trạng thái xác suất ban đầu bao gồm phương tiện để xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và phương tiện để giải mã entropy, từ dòng bit và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phân tử cú pháp.

24. Phương tiện lưu trữ bất biến đọc được bằng máy tính lưu trữ các lệnh, khi được thực thi, khiến một hoặc nhiều bộ xử lý của bộ lập mã video: thu được giá trị khởi tạo định trước cho một ngữ cảnh trong số nhiều ngữ cảnh được sử dụng trong quy trình lập mã entropy thích ứng theo ngữ cảnh để lập mã entropy giá trị cho phần tử cú pháp cho đơn vị có thể lập mã độc lập của dữ liệu video; xác định, dựa vào giá trị khởi tạo định trước, giá trị chỉ số độ dốc và giá trị chỉ số độ lệch; xác định, dựa vào giá trị chỉ số độ dốc, giá trị của tham số m; xác định, dựa vào giá trị chỉ số độ lệch, giá trị của tham số n theo phương trình sau $n = (\text{OffsetIdx} * 18) + 1$ trong đó OffsetIdx là giá trị chỉ số độ lệch; thu được giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập; xác định, dựa vào giá trị khởi tạo định trước, tham số m, tham số n, và giá trị ban đầu của tham số lượng tử hóa, trạng thái xác suất ban đầu của ngữ cảnh, trong đó để xác định trạng thái xác suất ban đầu, các bộ xử lý còn được tạo cấu hình để xác định trạng thái xác suất ban đầu dựa vào chênh lệch giữa giá trị ban đầu của tham số lượng tử hóa liên quan đến đơn vị có thể lập mã độc lập và giá trị khác không xác định điểm neo tham số lượng tử hóa; và mã hóa entropy, trong dòng bit và dựa vào trạng thái xác suất ban đầu của ngữ cảnh, bin của giá trị cho phần tử cú pháp.

1 / 28

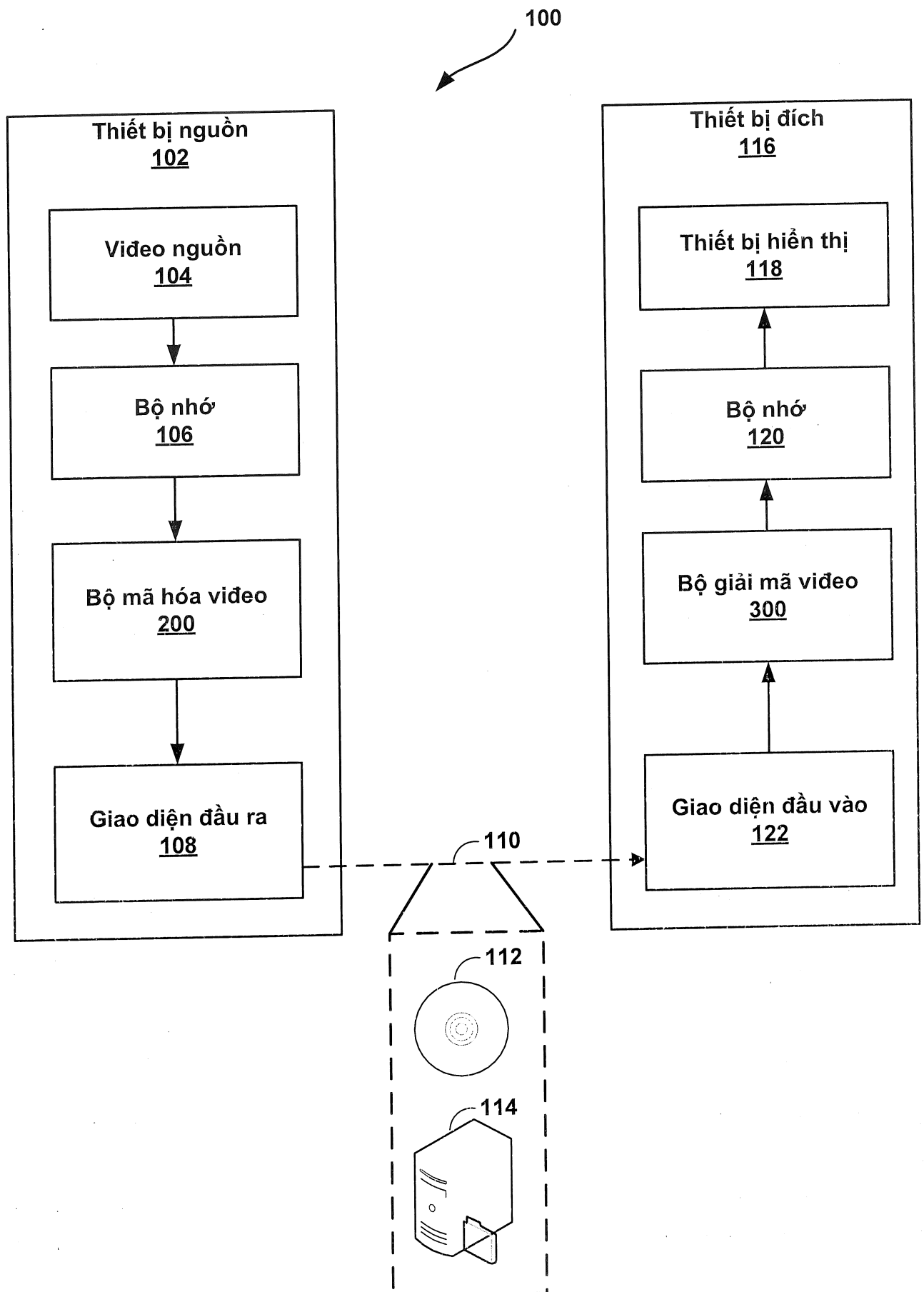


FIG. 1

2 / 28

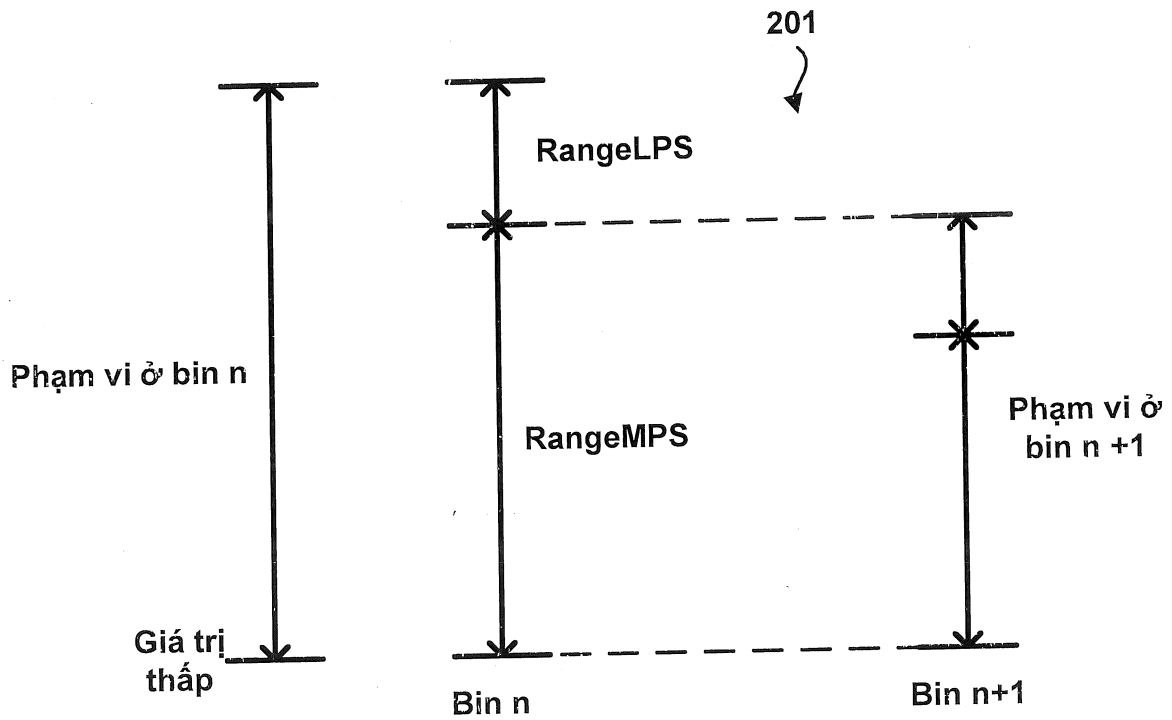


FIG. 2A

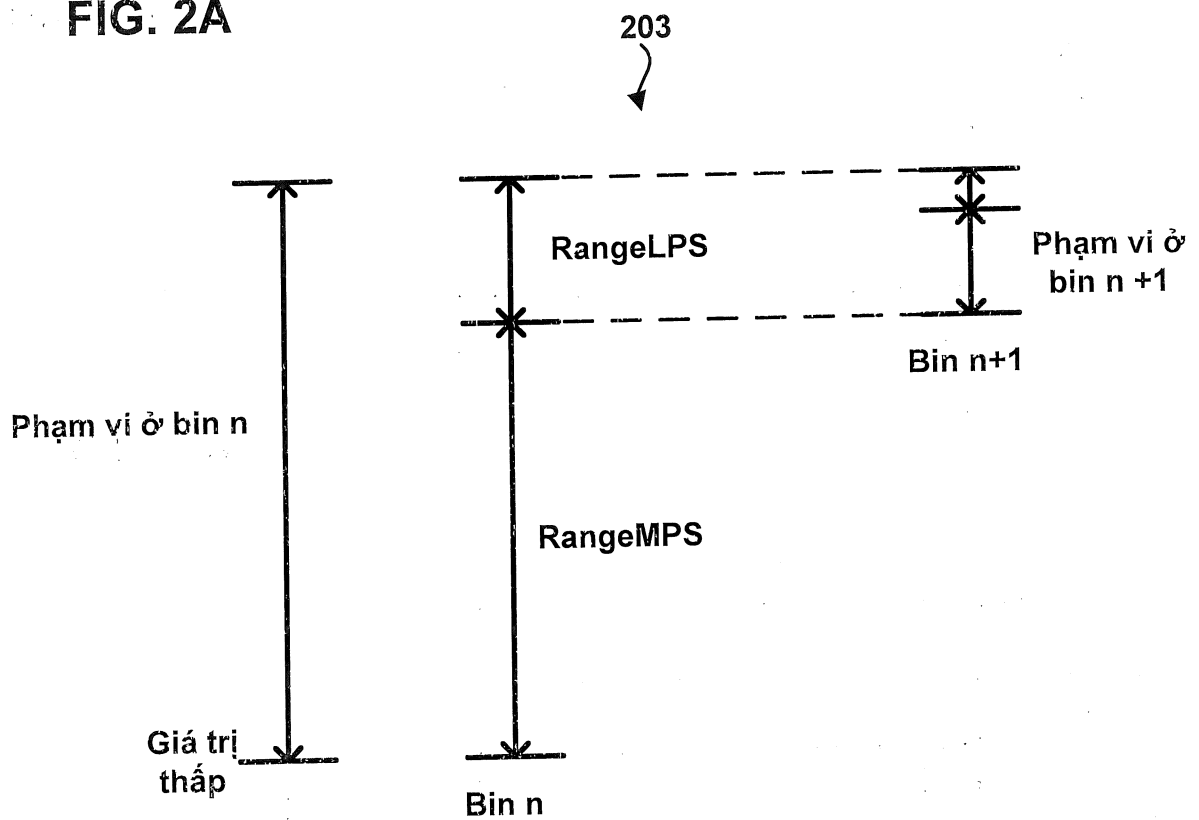


FIG. 2B

3 / 28

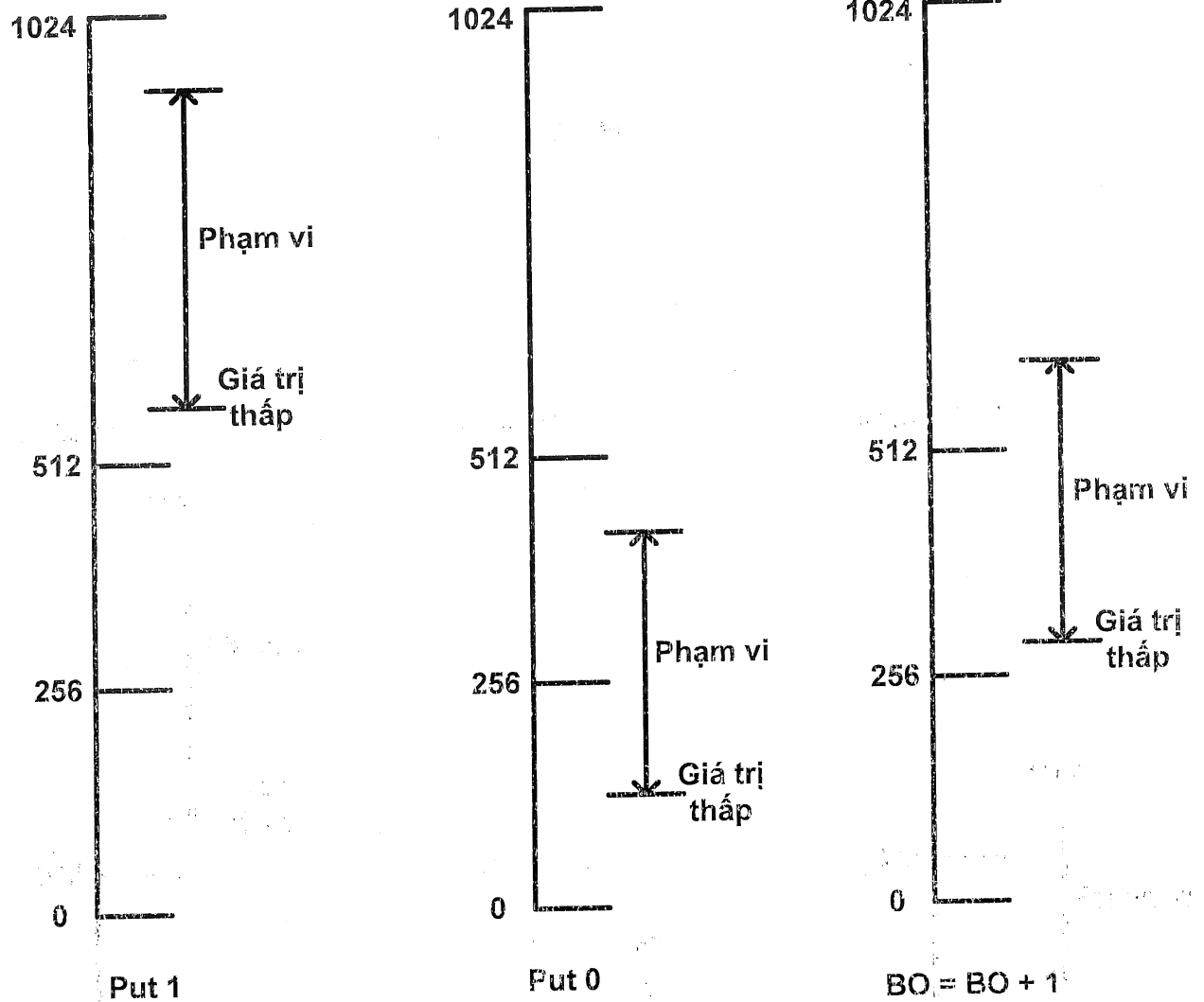


FIG. 3

4 / 28

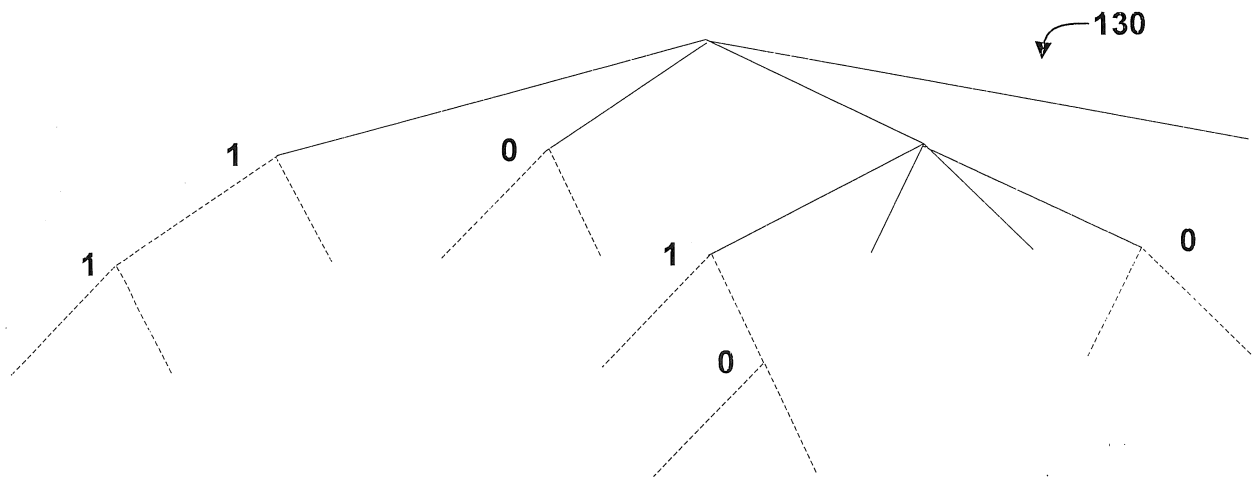


FIG. 4A

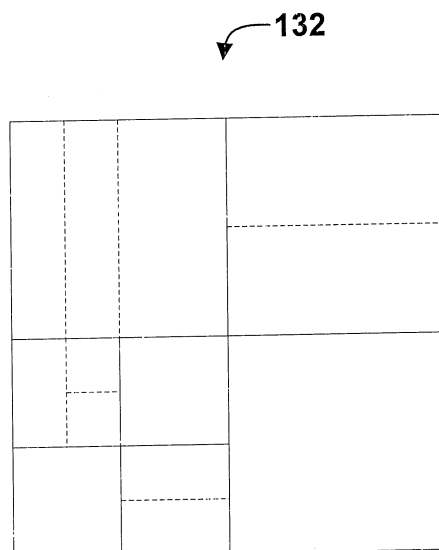


FIG. 4B

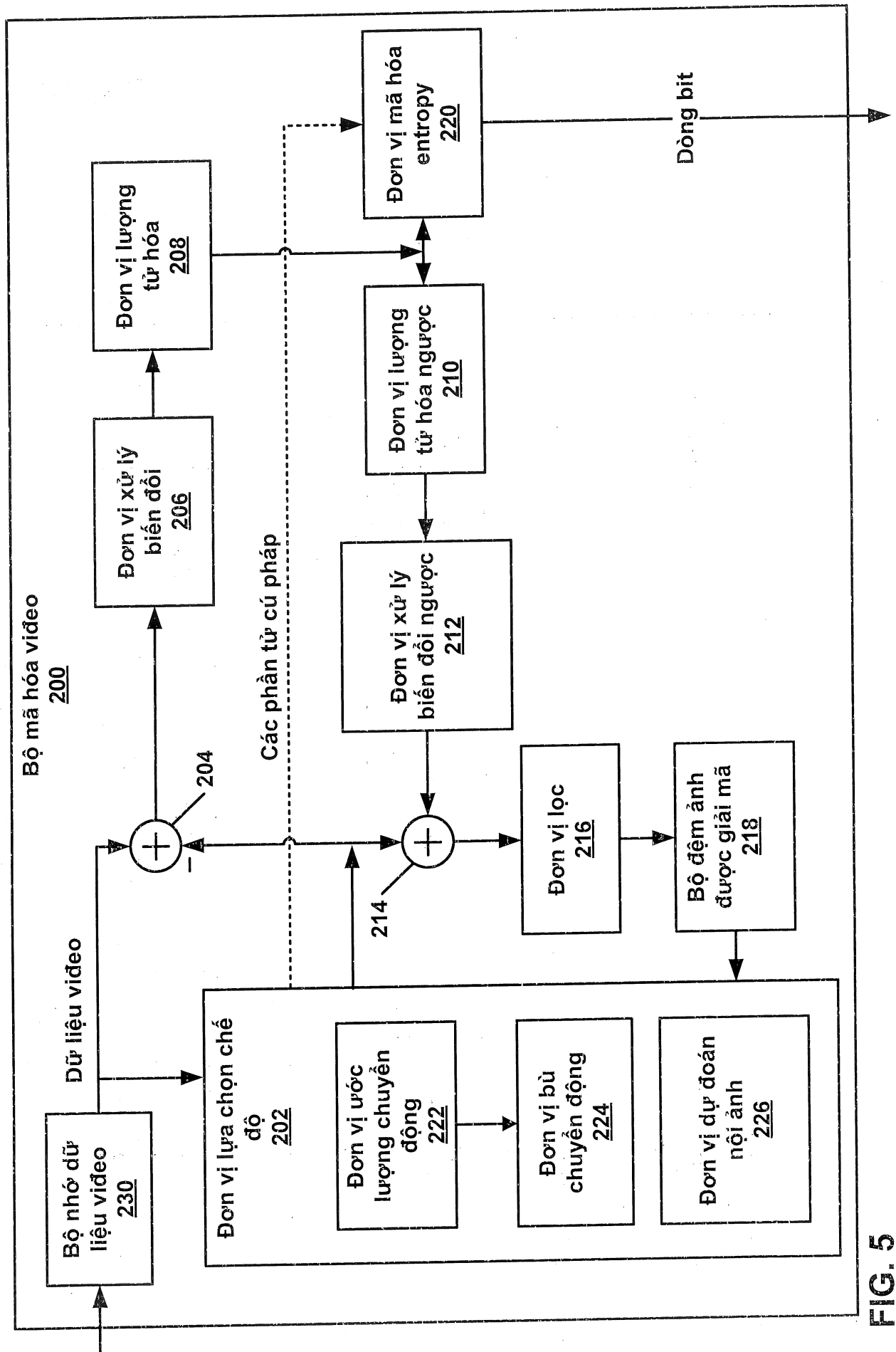
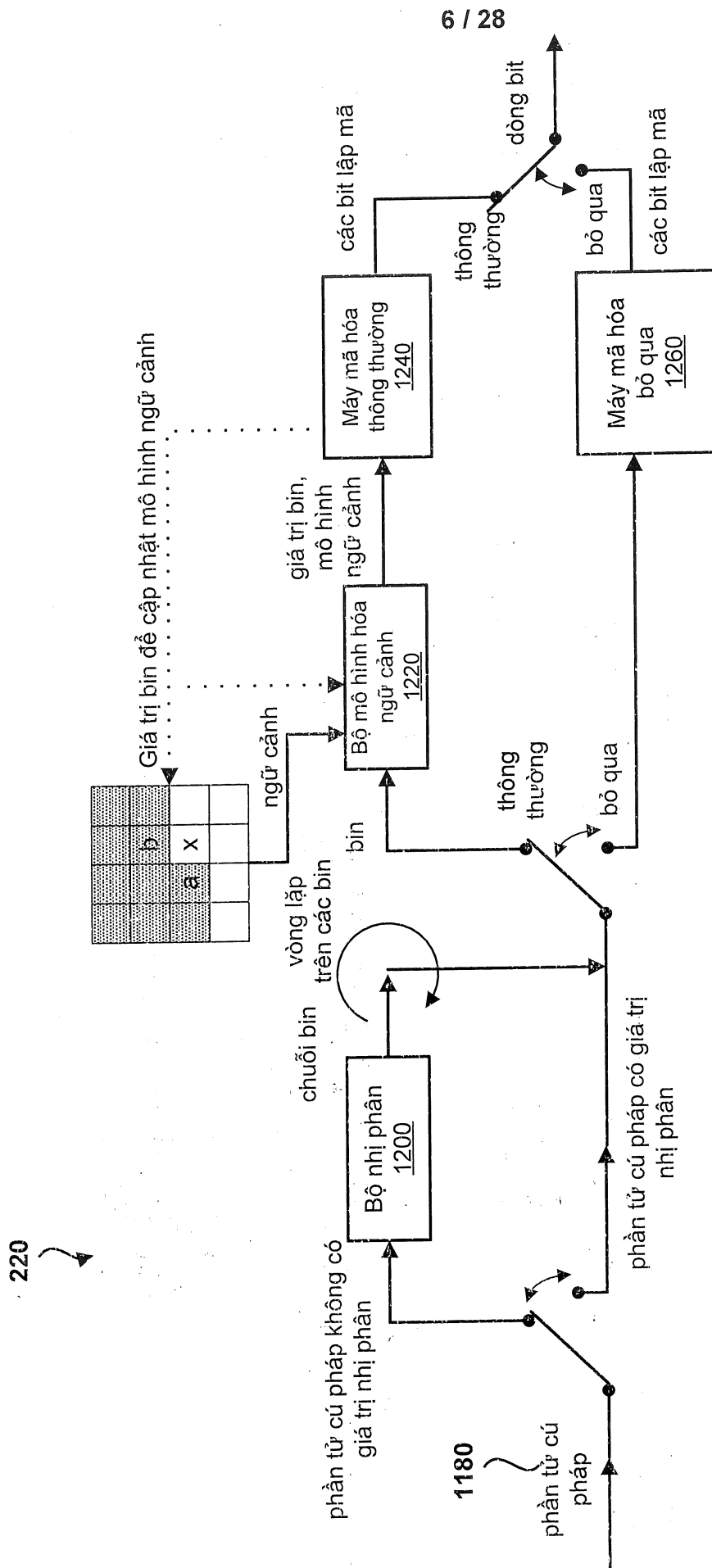


FIG. 5



65

7 / 28

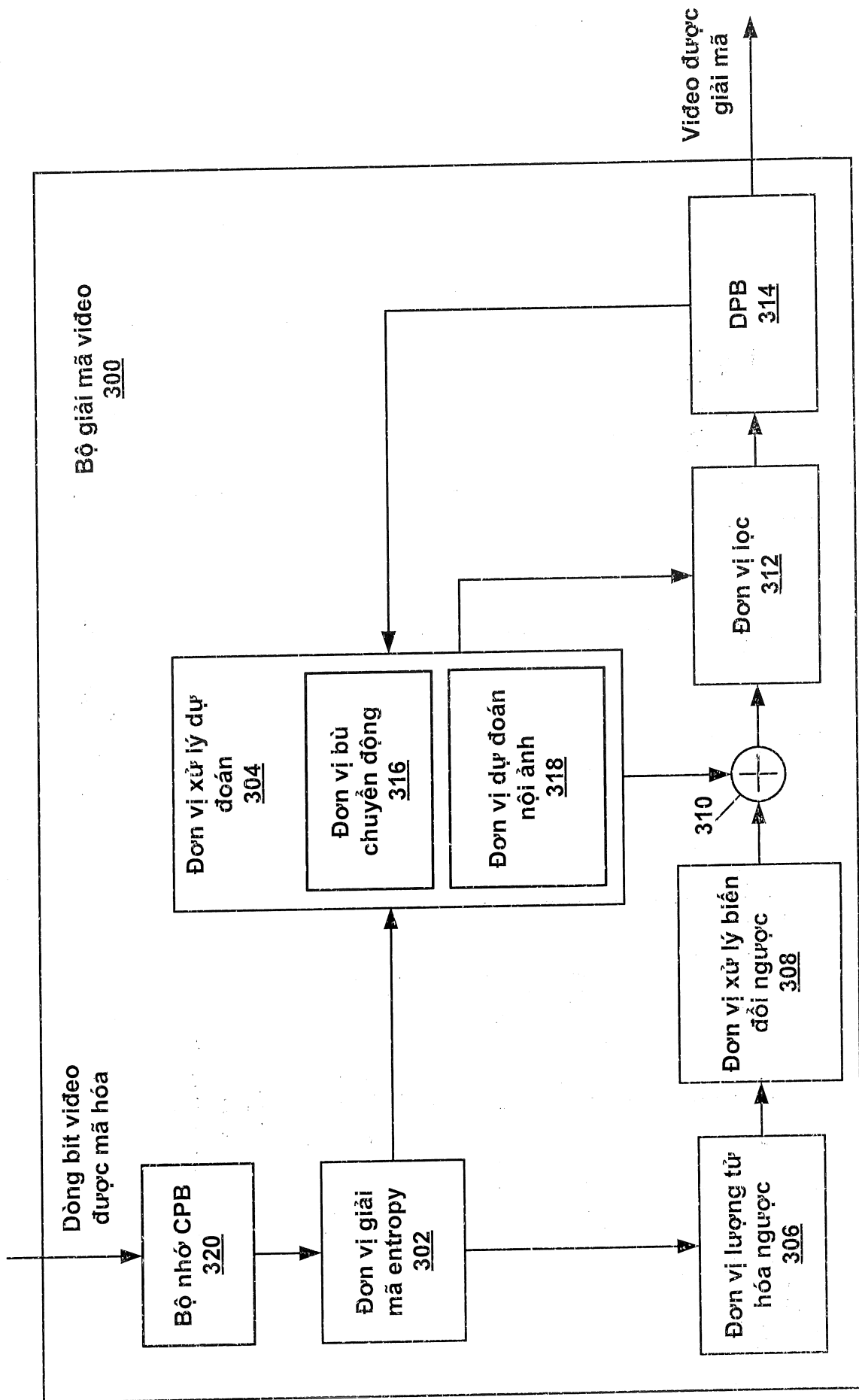


FIG. 7

8 / 28

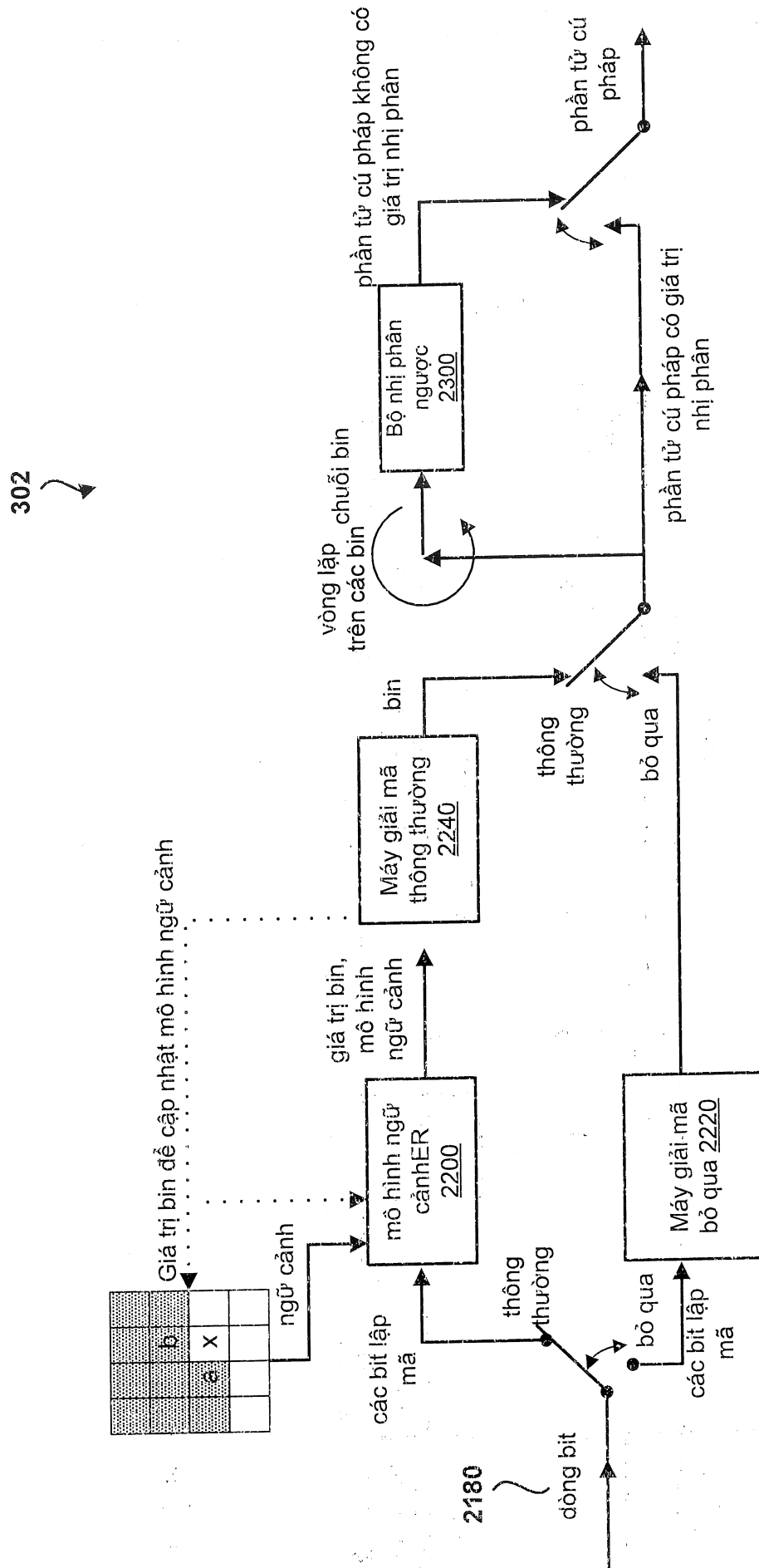


FIG. 8

9 / 28

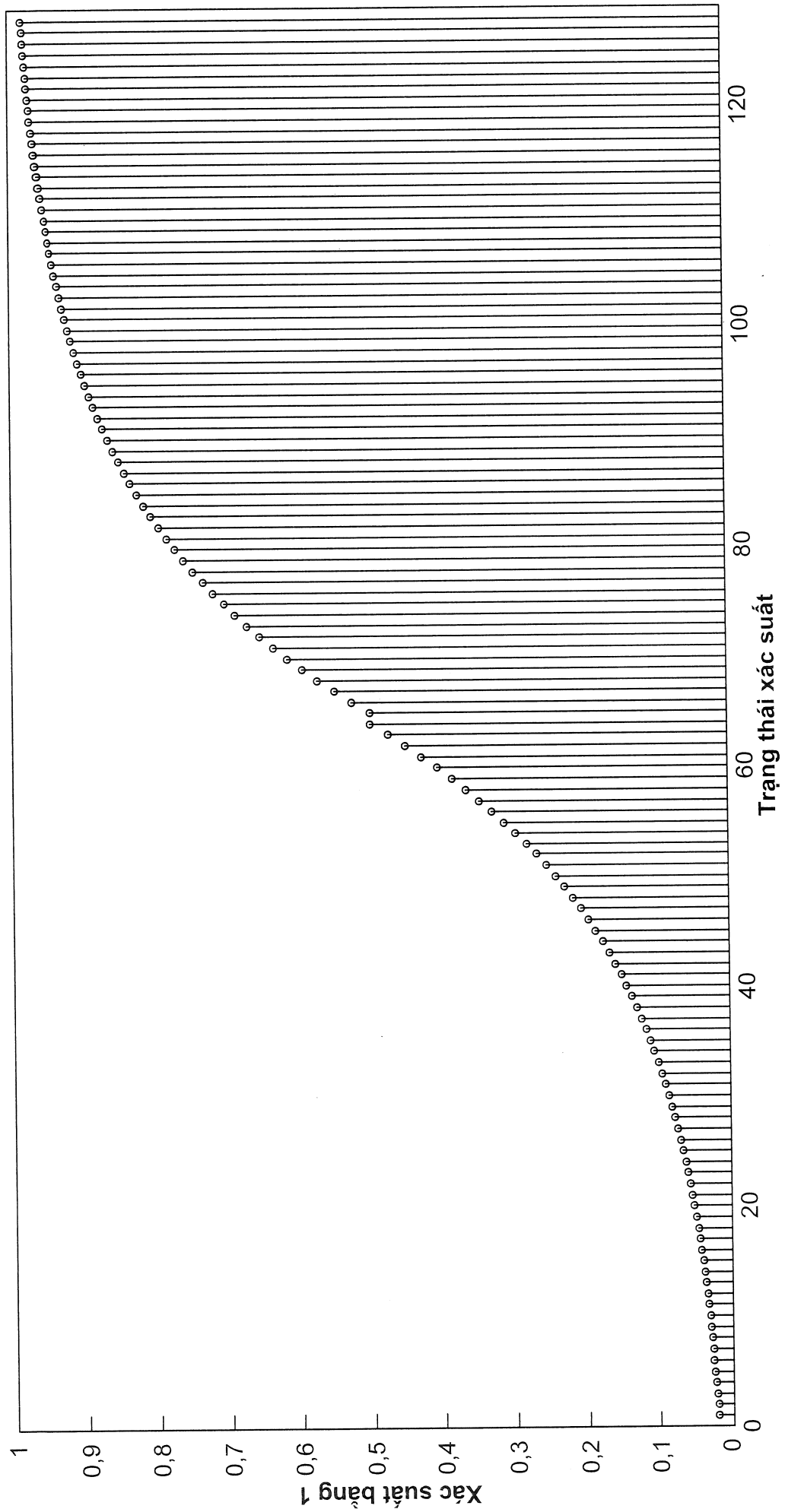
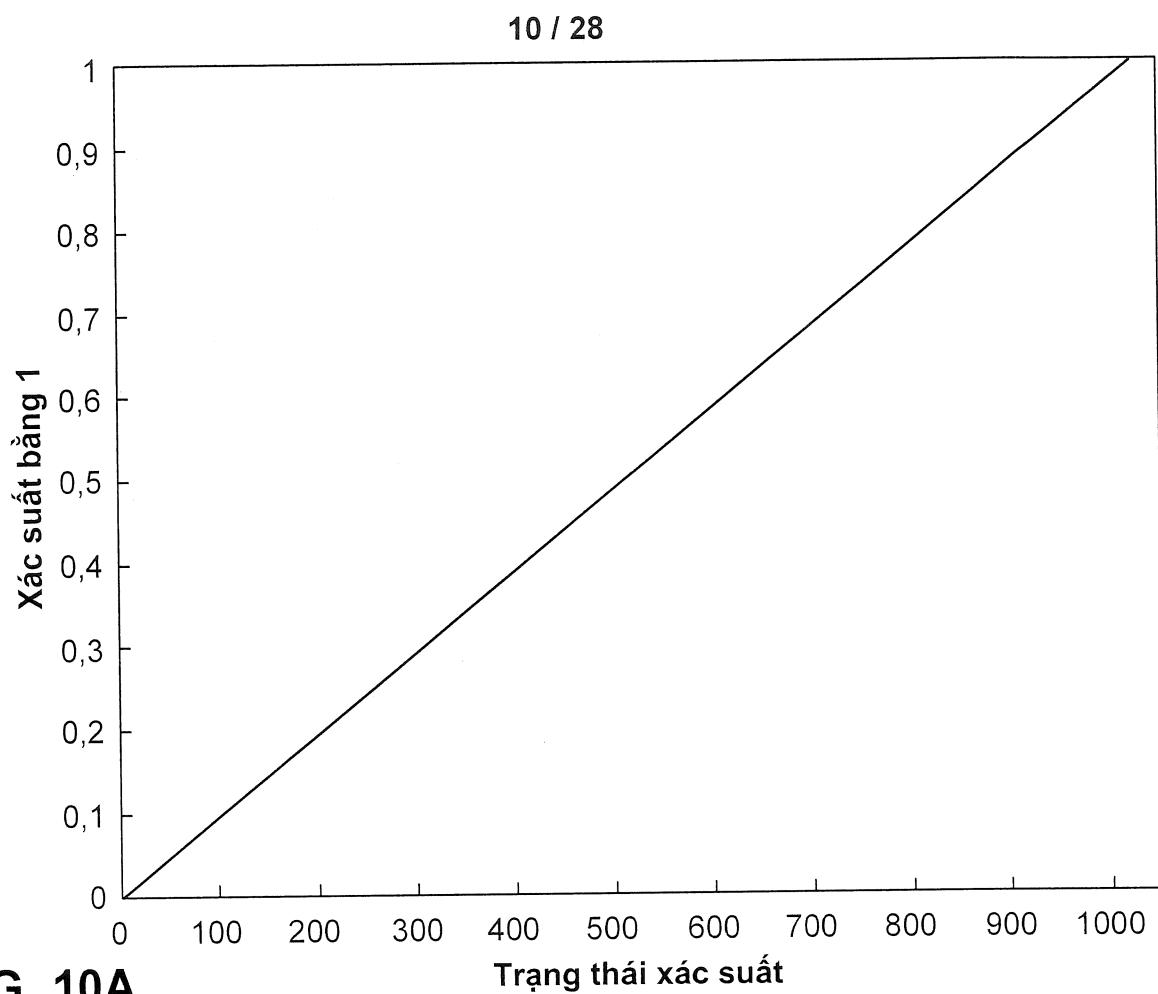
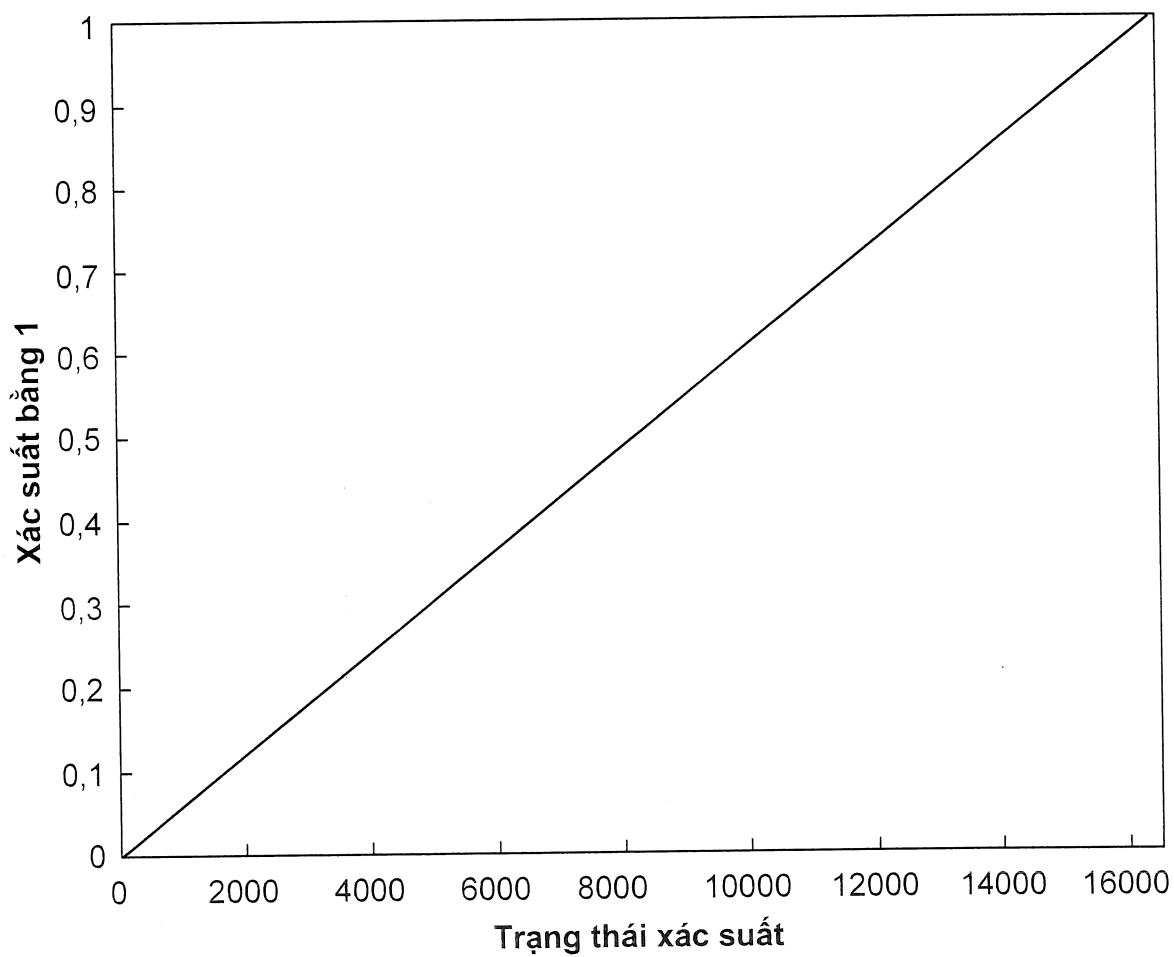


FIG. 9

**FIG. 10A****FIG. 10B**

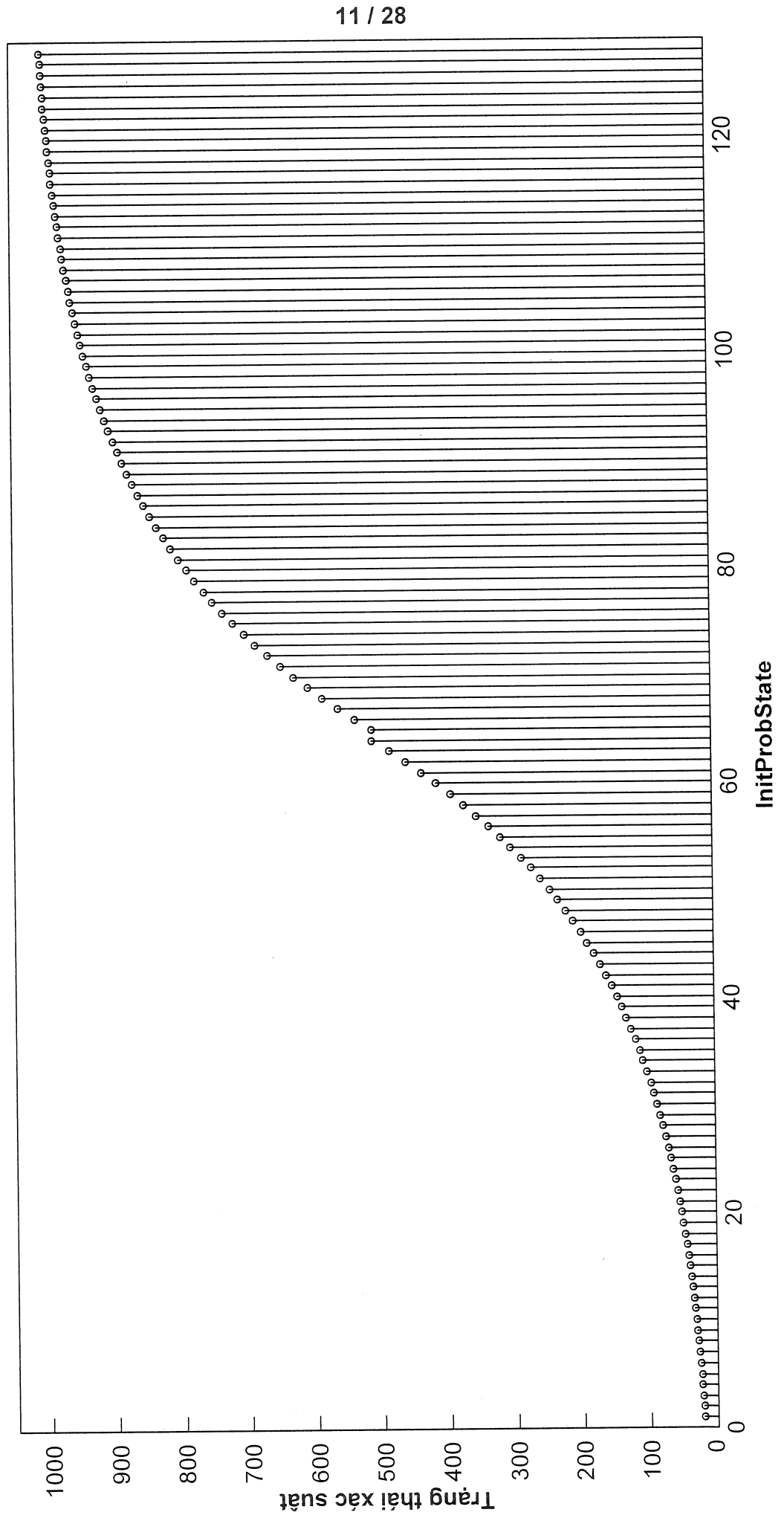


FIG. 11A

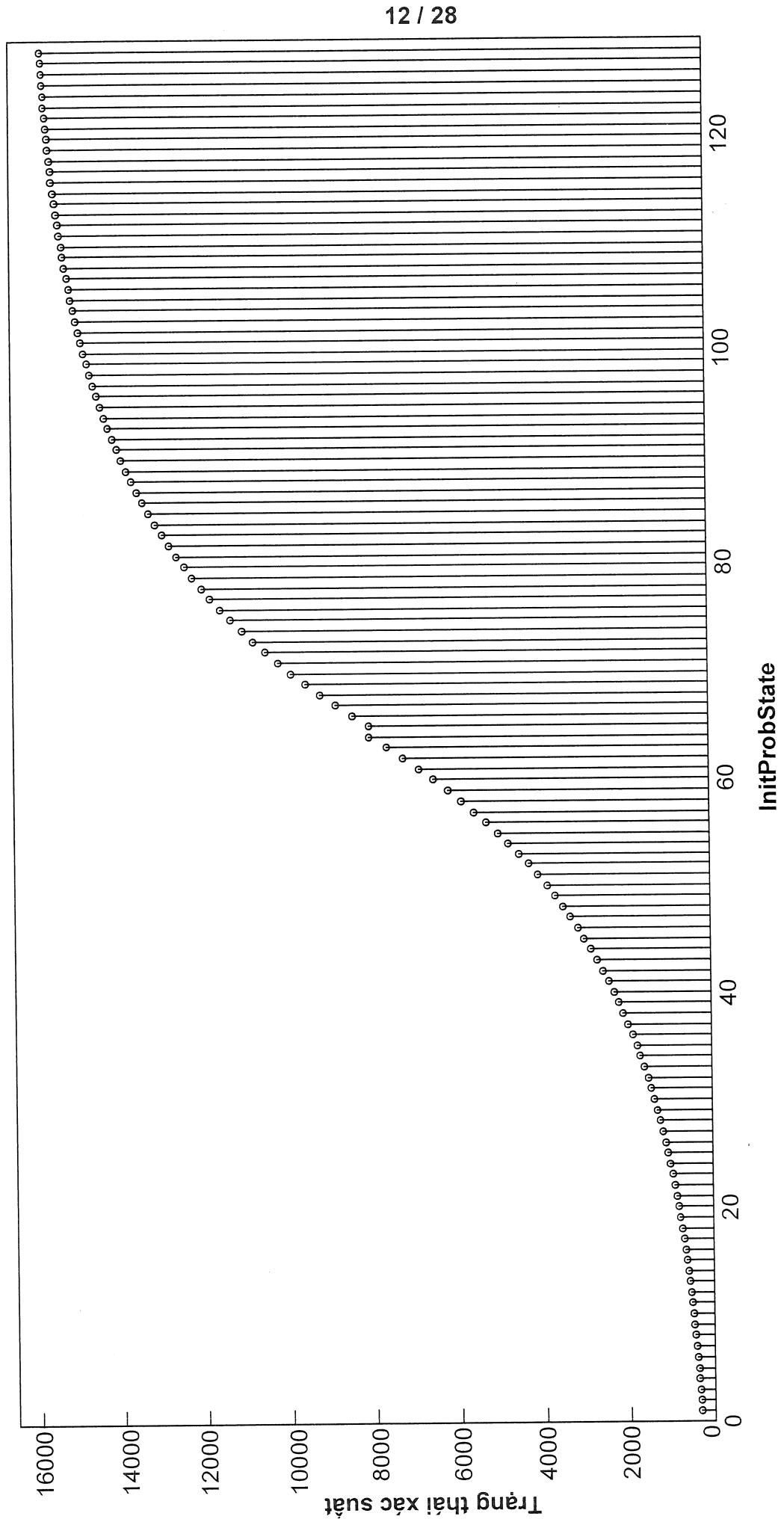


FIG. 11B

13 / 28

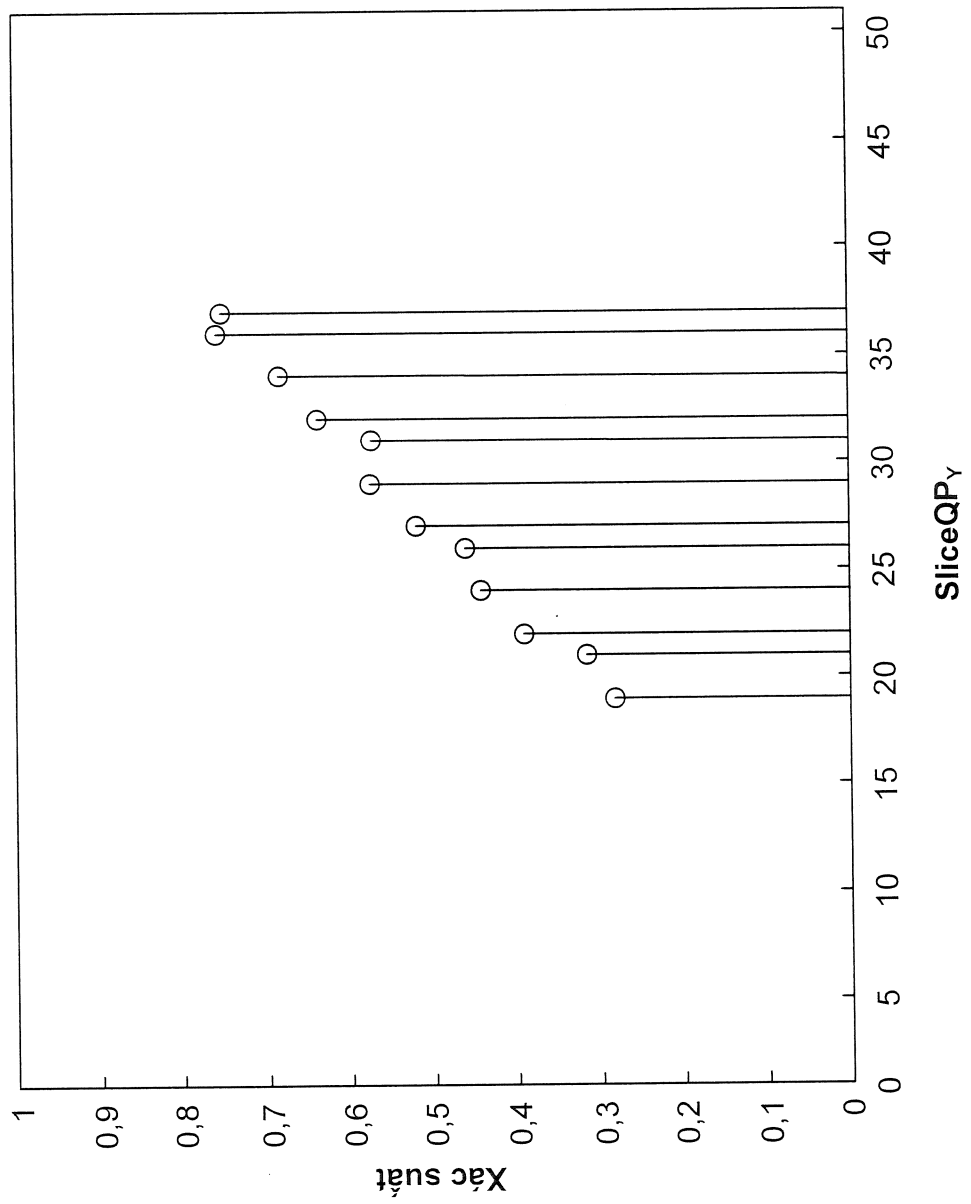


FIG. 12

14 / 28

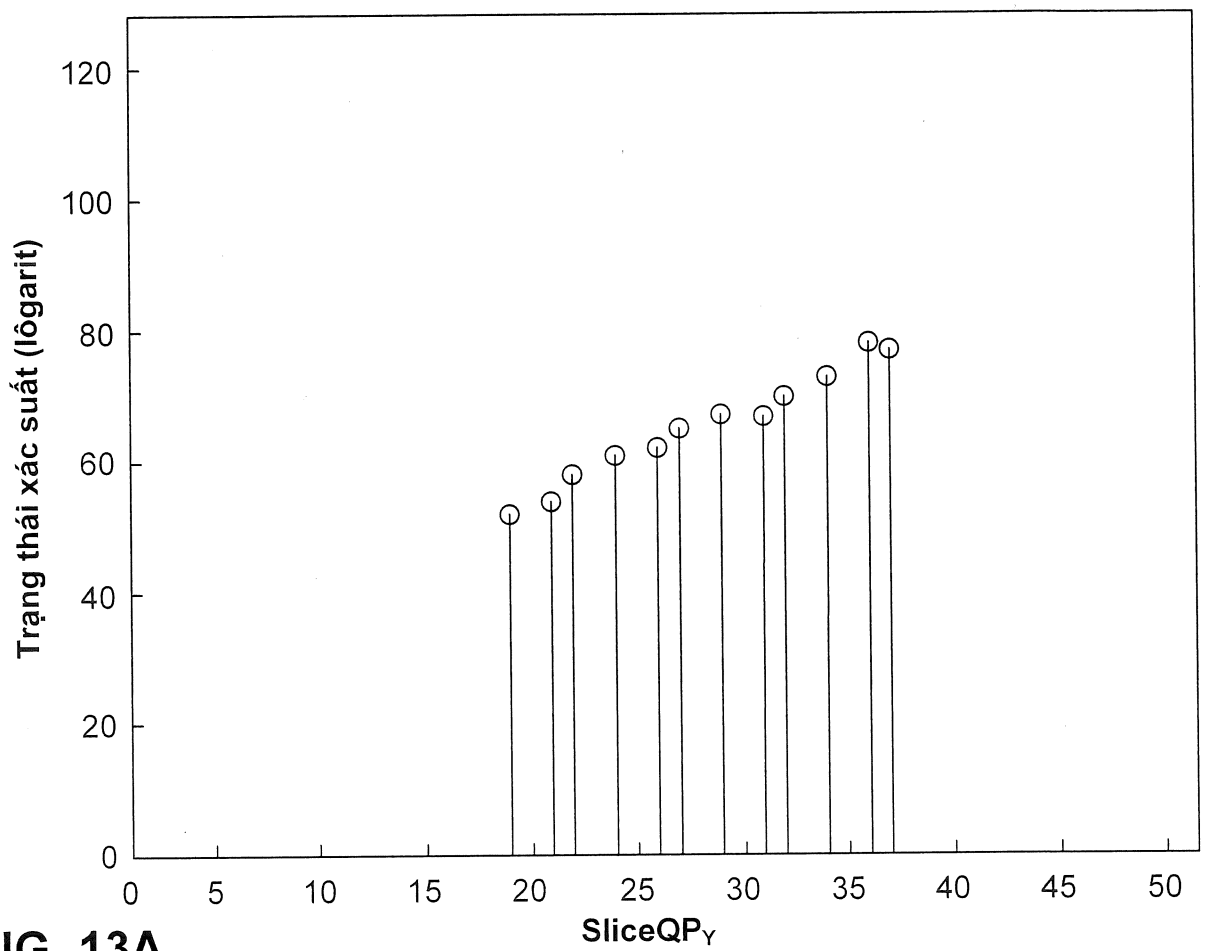


FIG. 13A

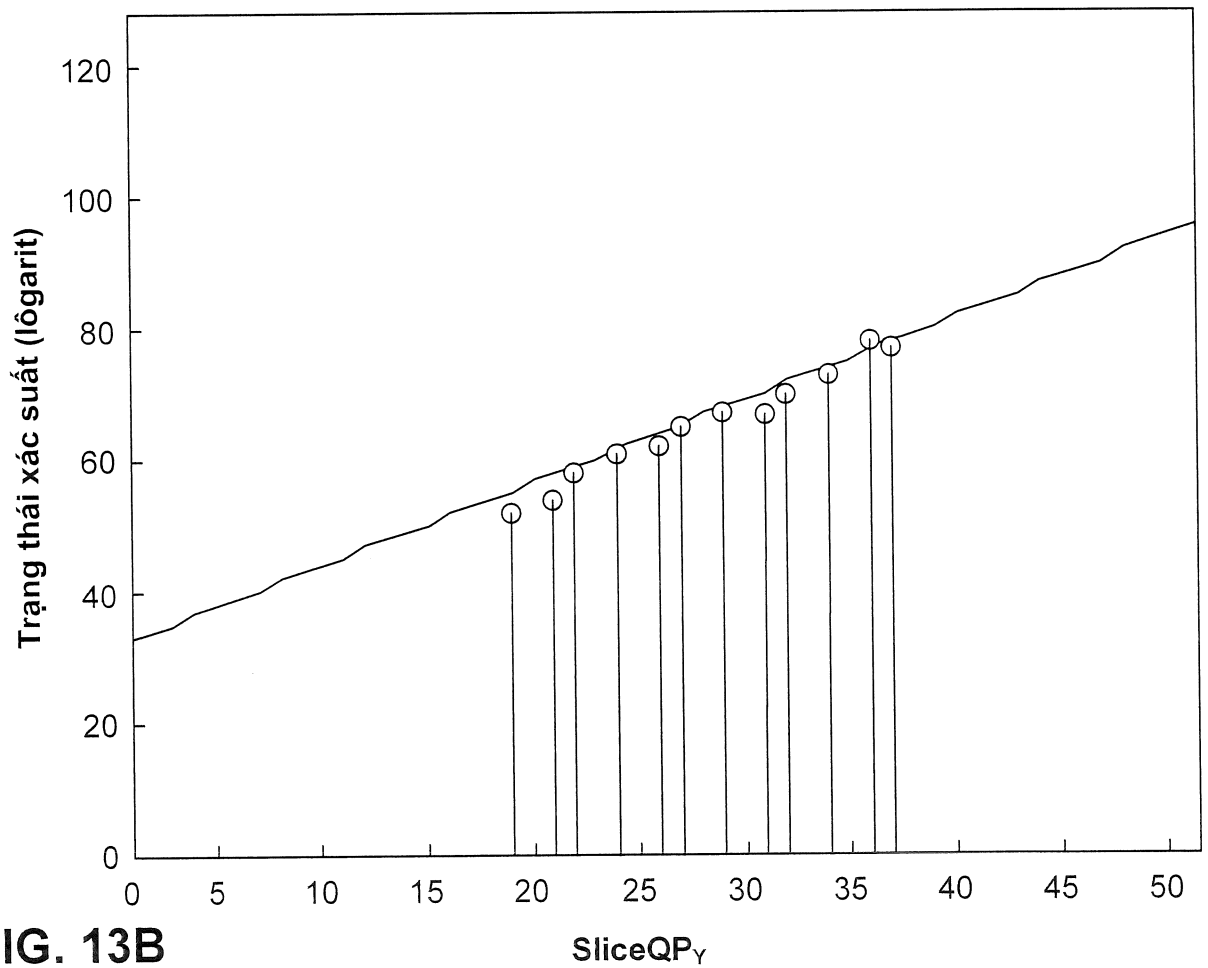


FIG. 13B

15 / 28

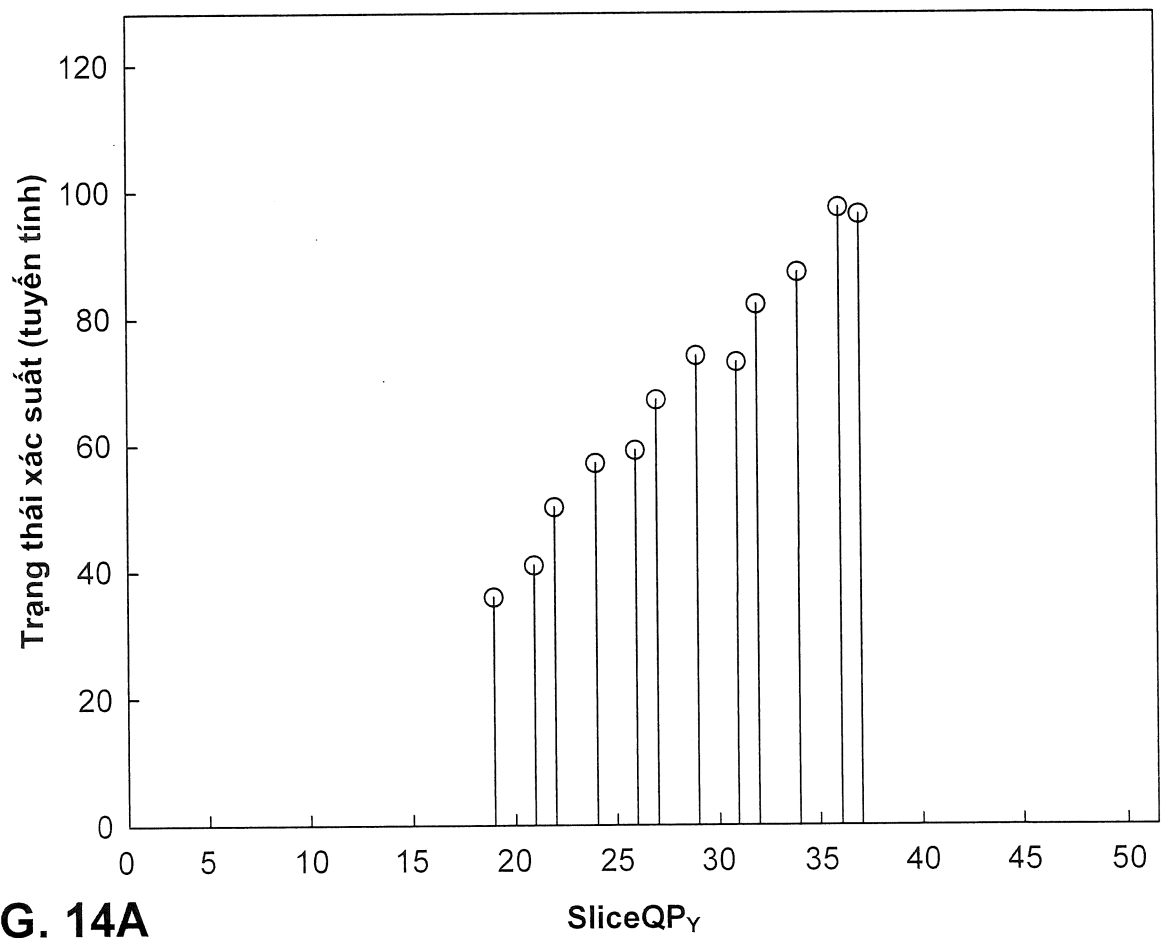


FIG. 14A

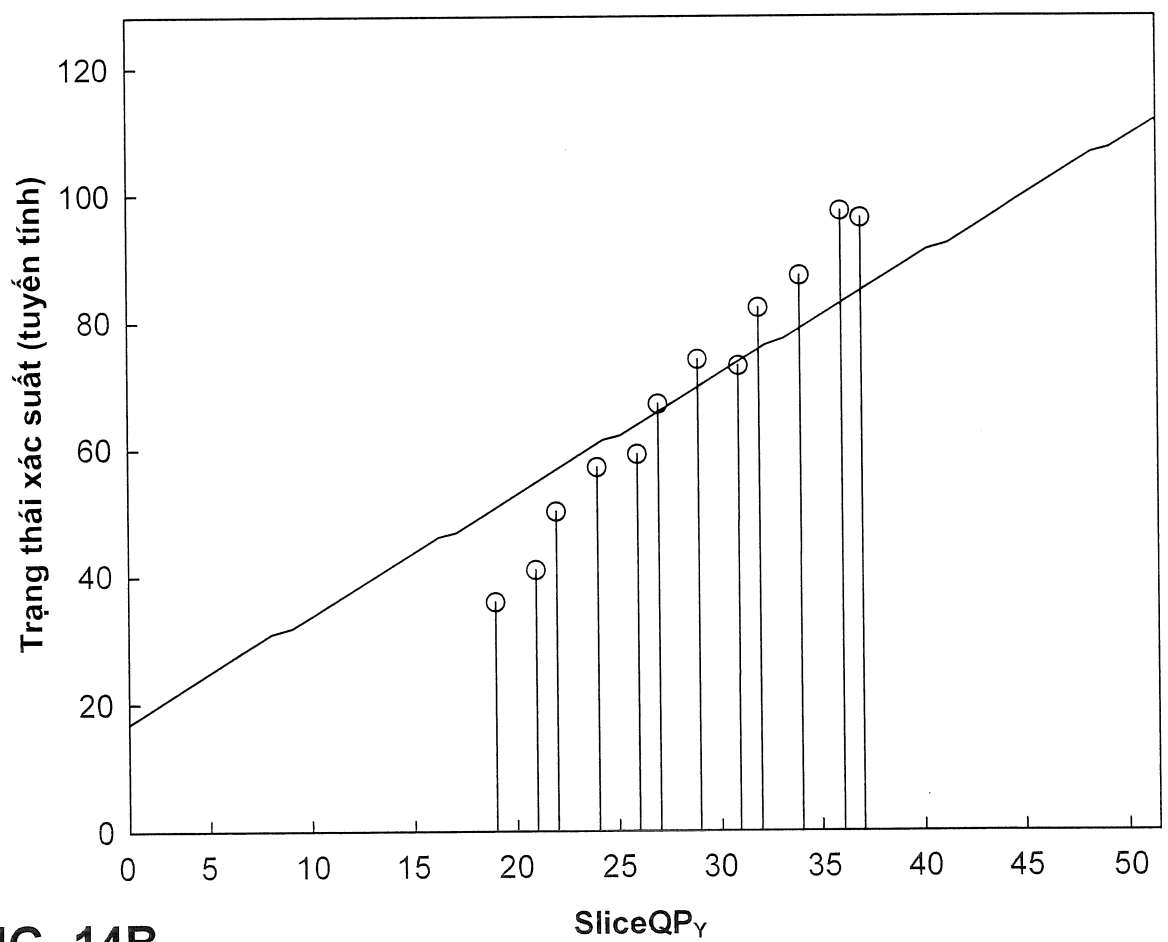


FIG. 14B

16 / 28

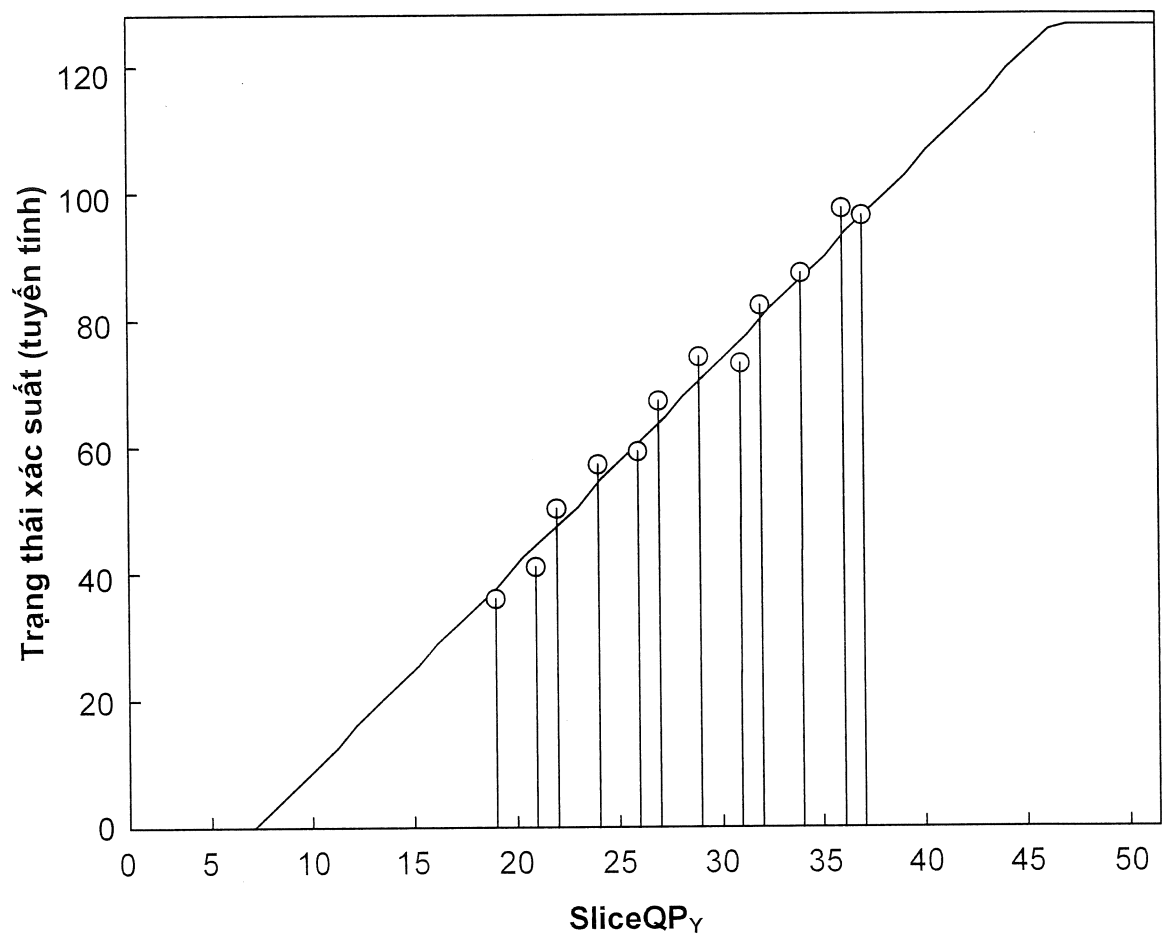


FIG. 14C

17 / 28

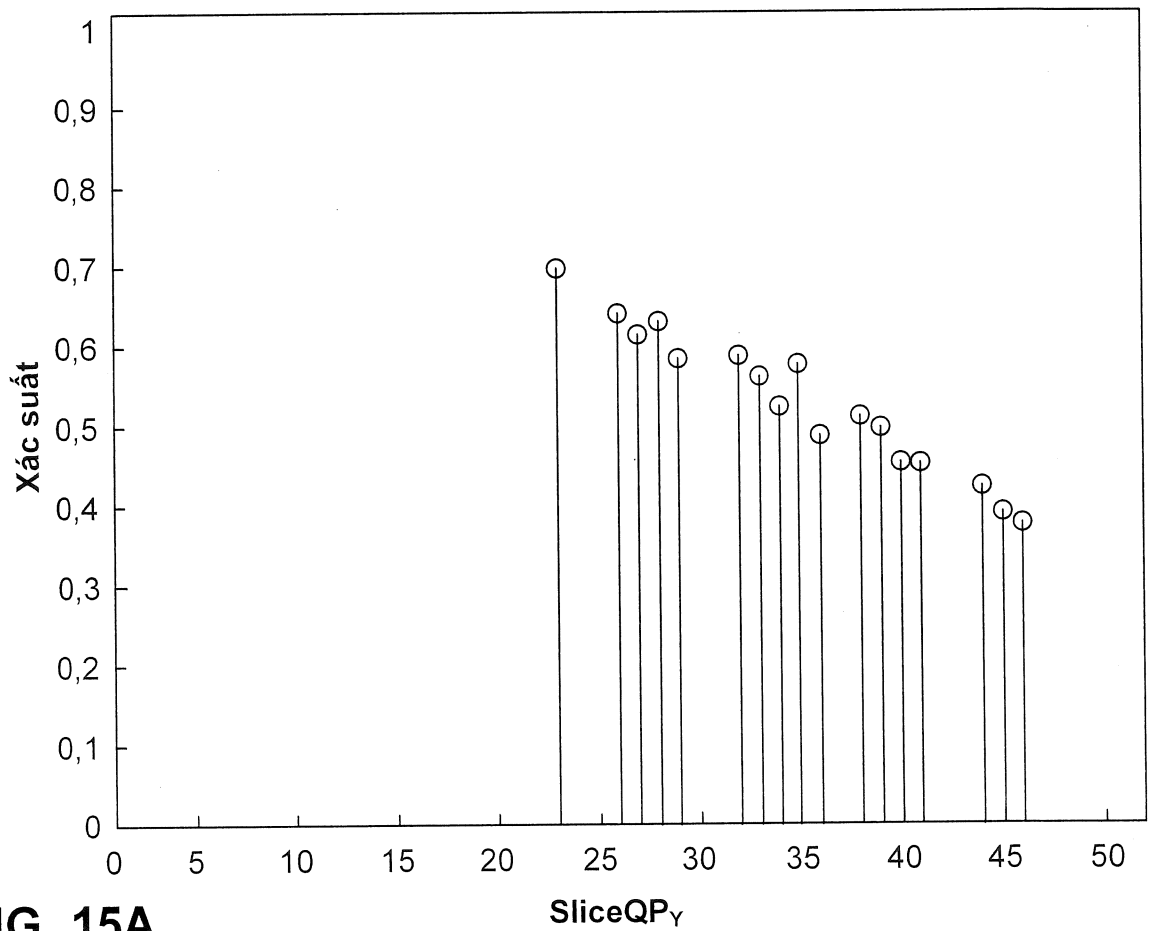


FIG. 15A

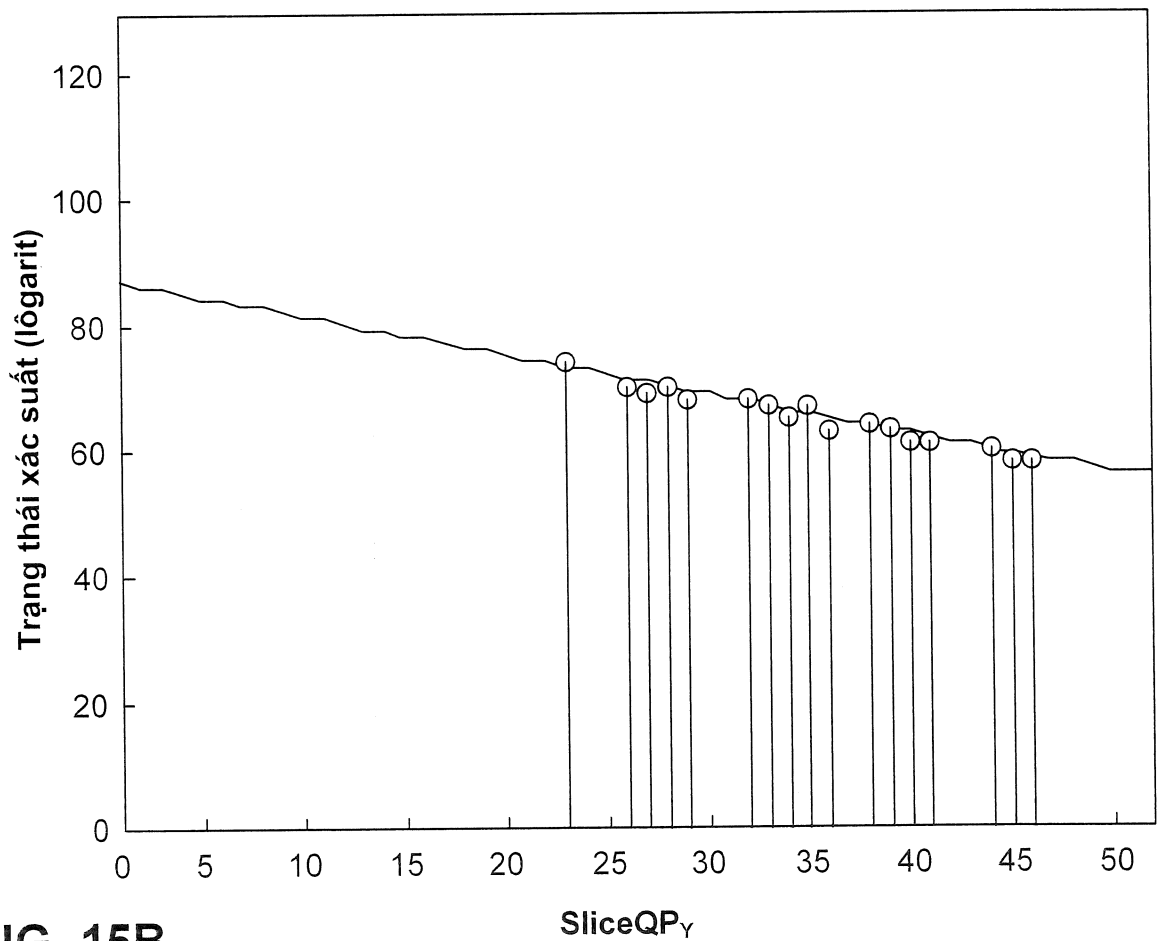


FIG. 15B

18 / 28

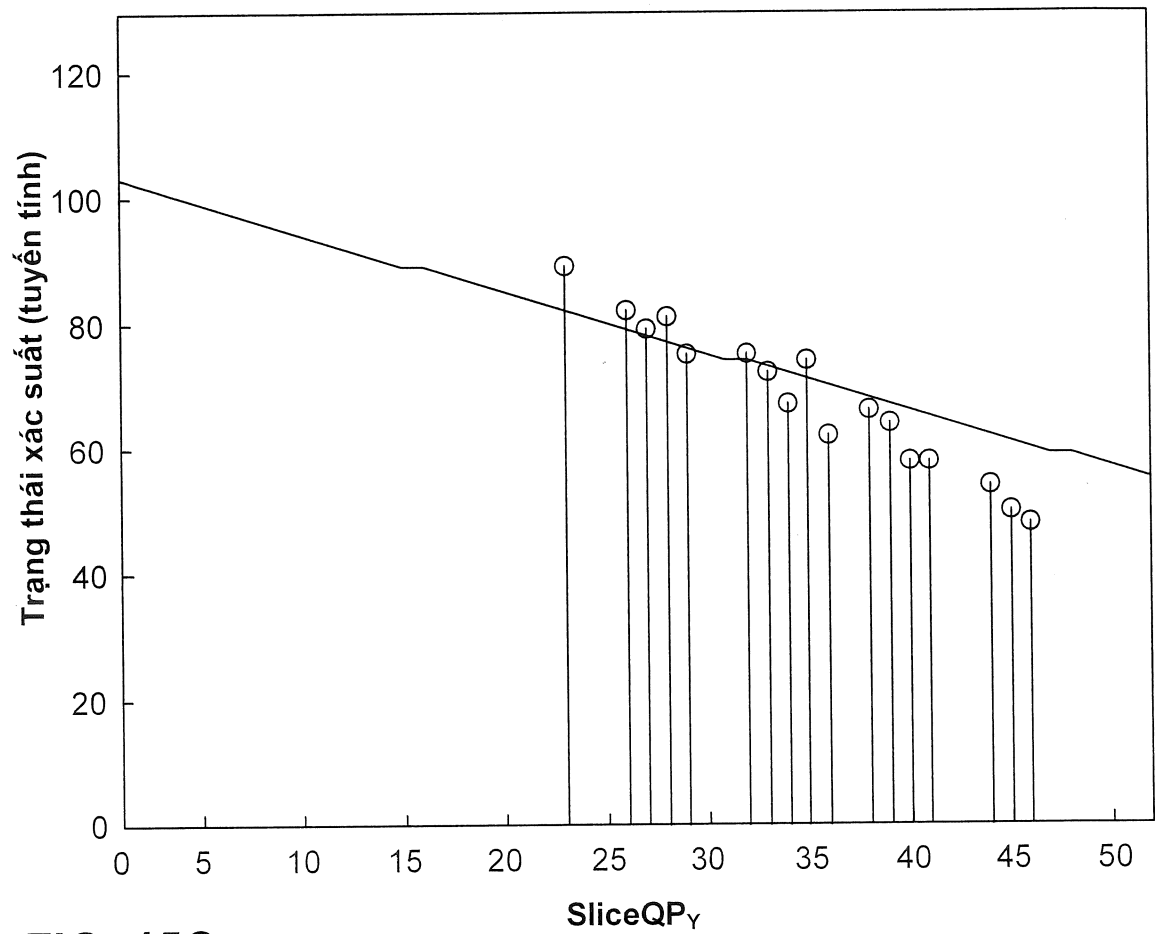


FIG. 15C

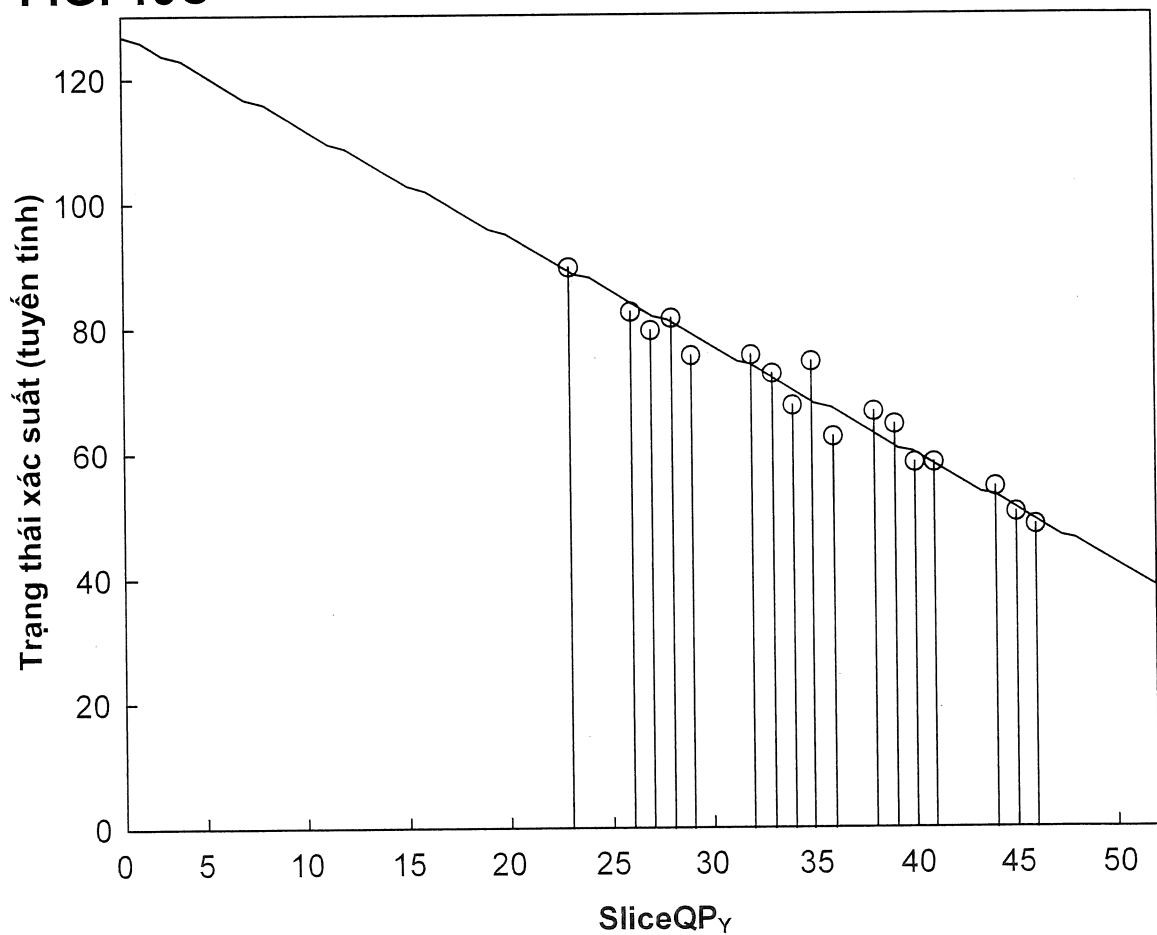


FIG. 15D

19 / 28

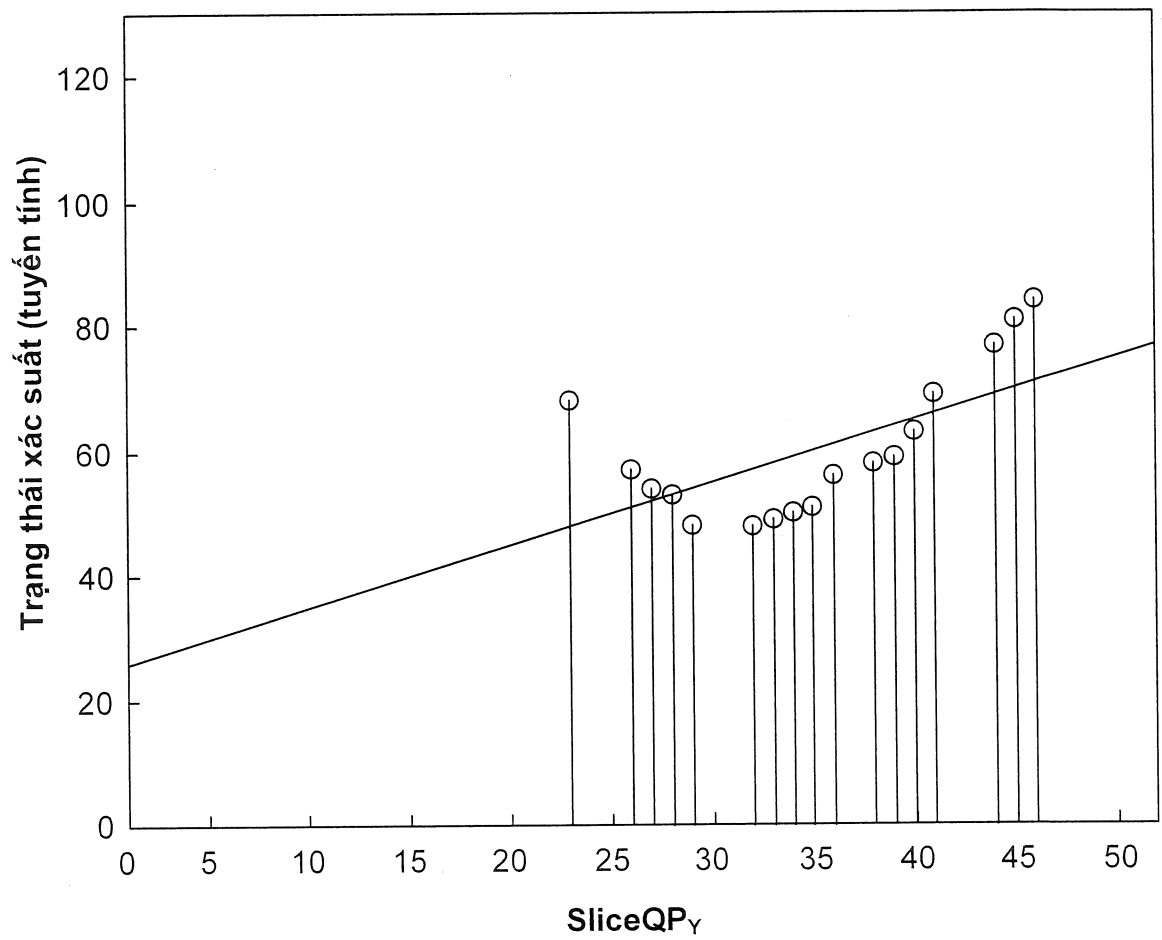


FIG. 16A

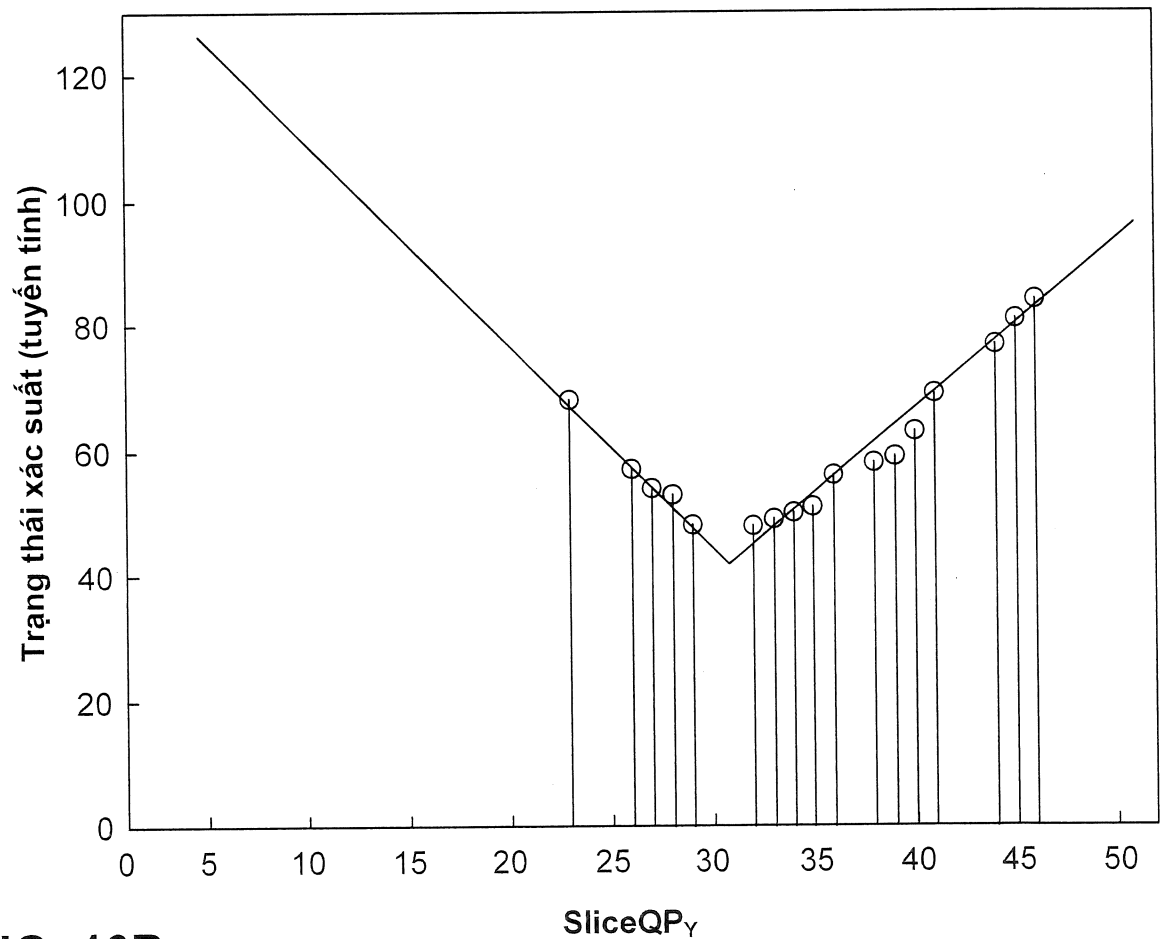


FIG. 16B

20 / 28

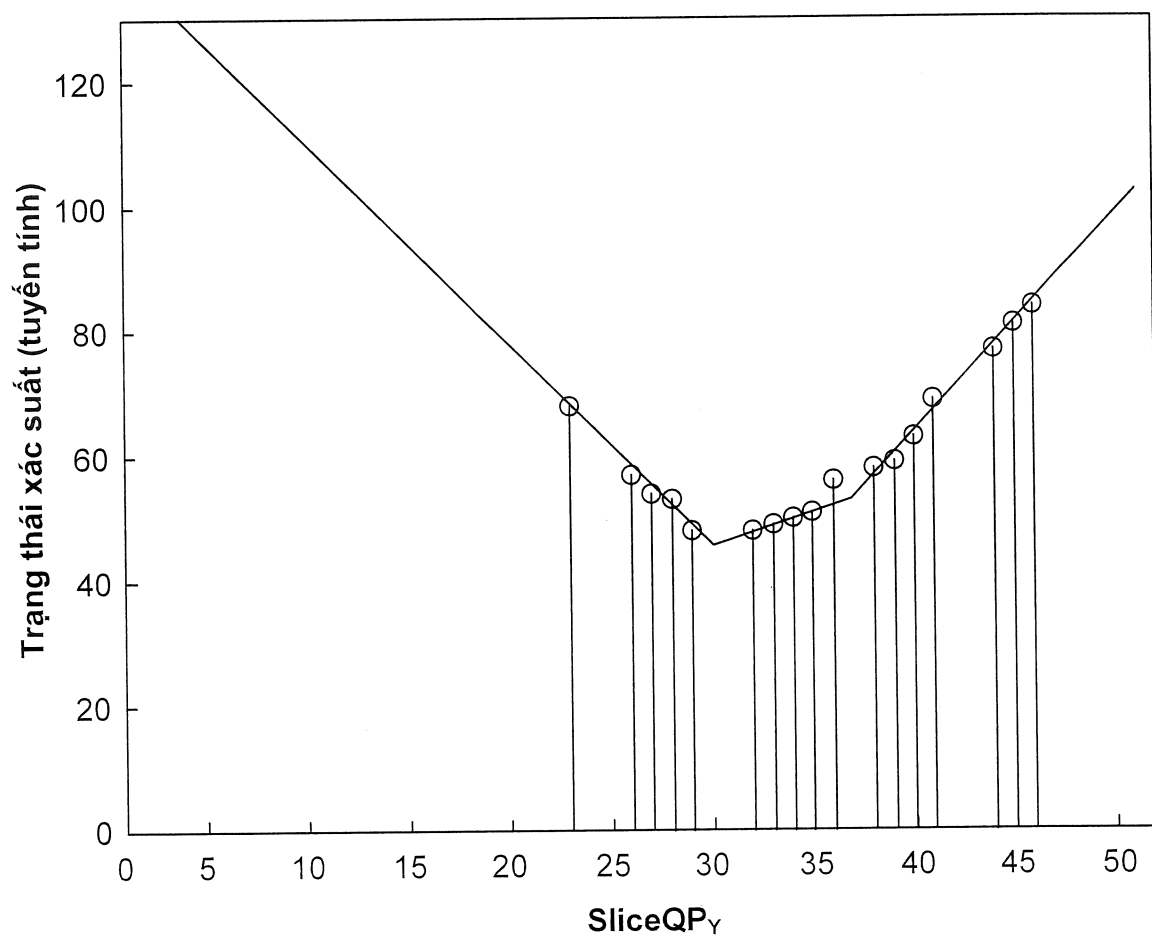


FIG. 16C

21 / 28

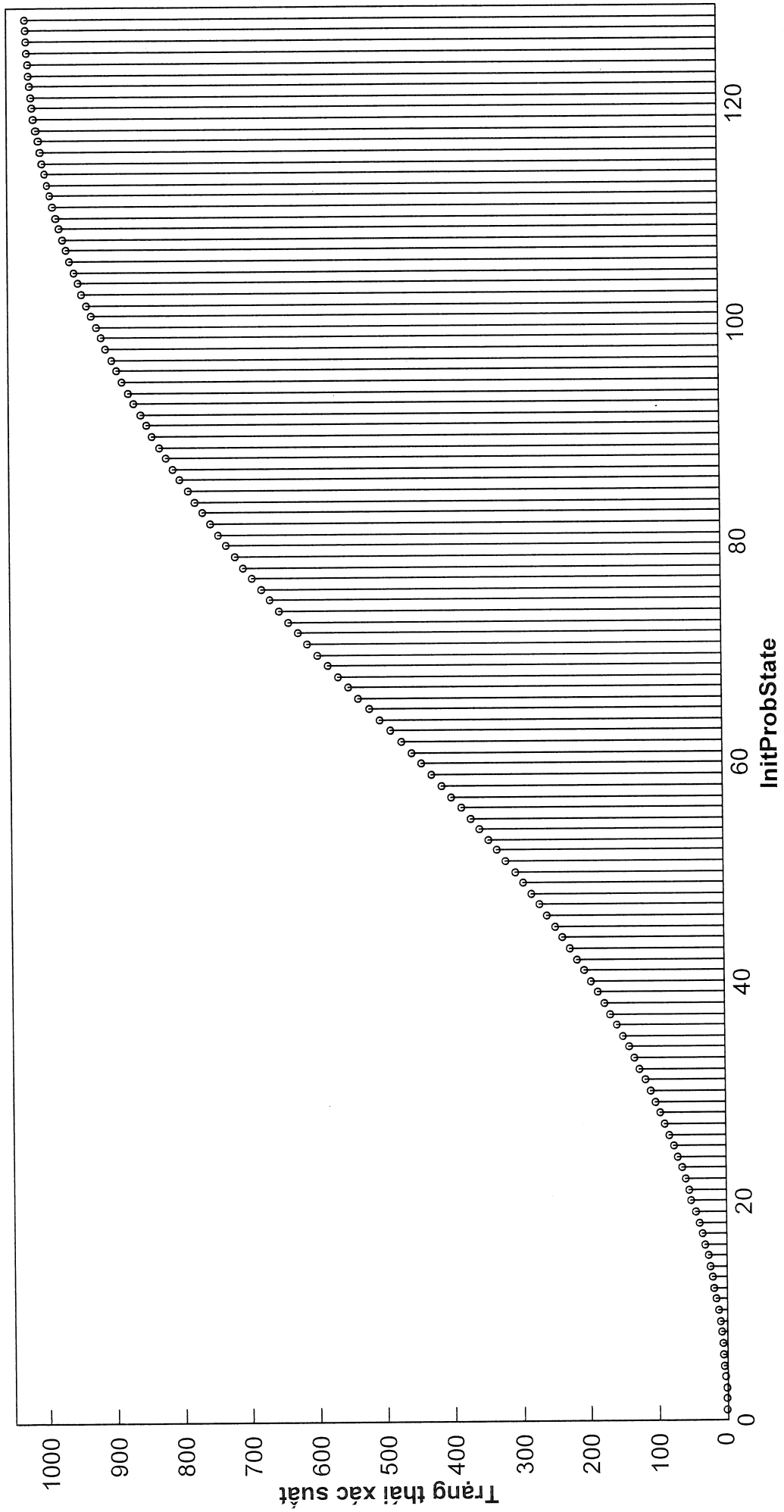


FIG. 17A

22 / 28

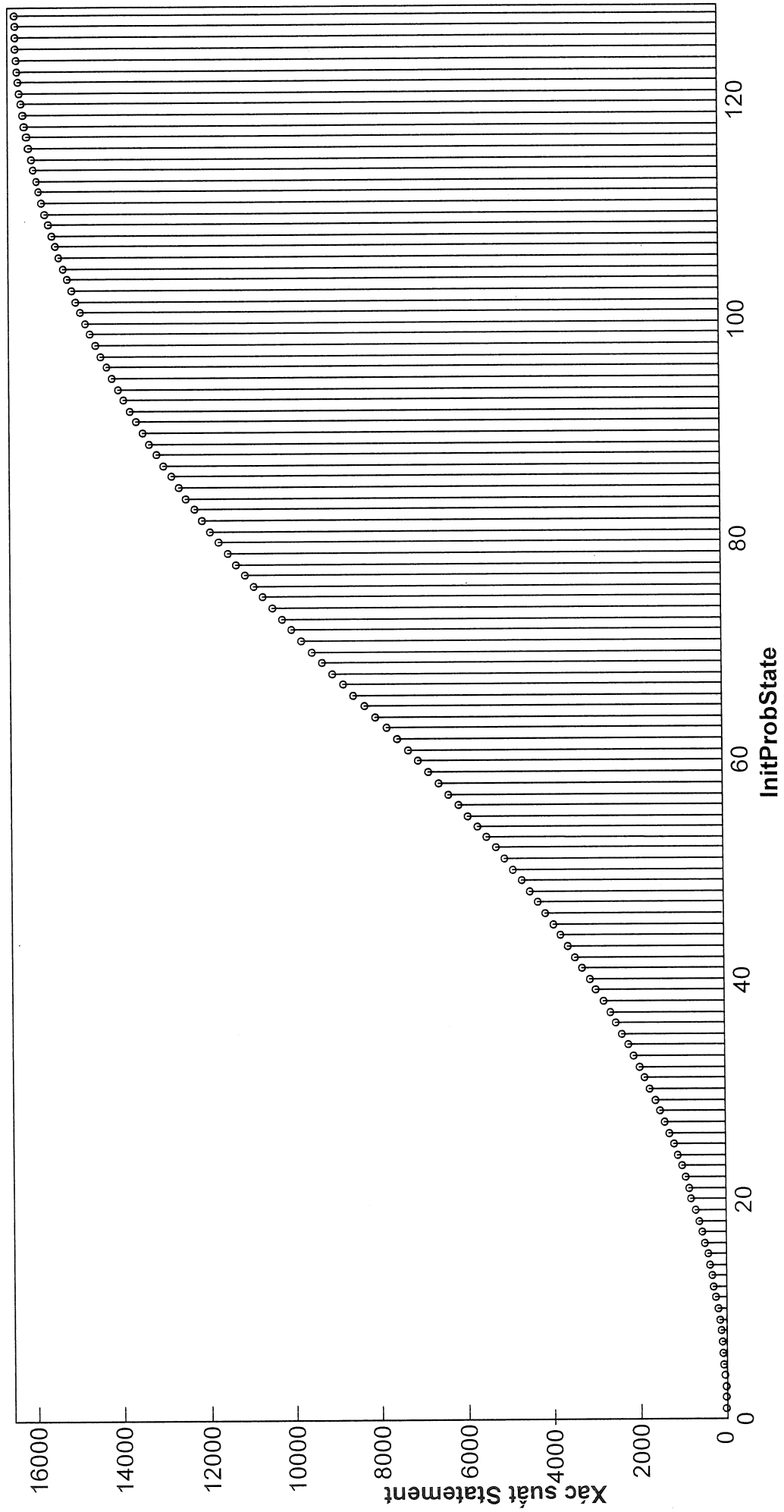


FIG. 17B

23 / 28

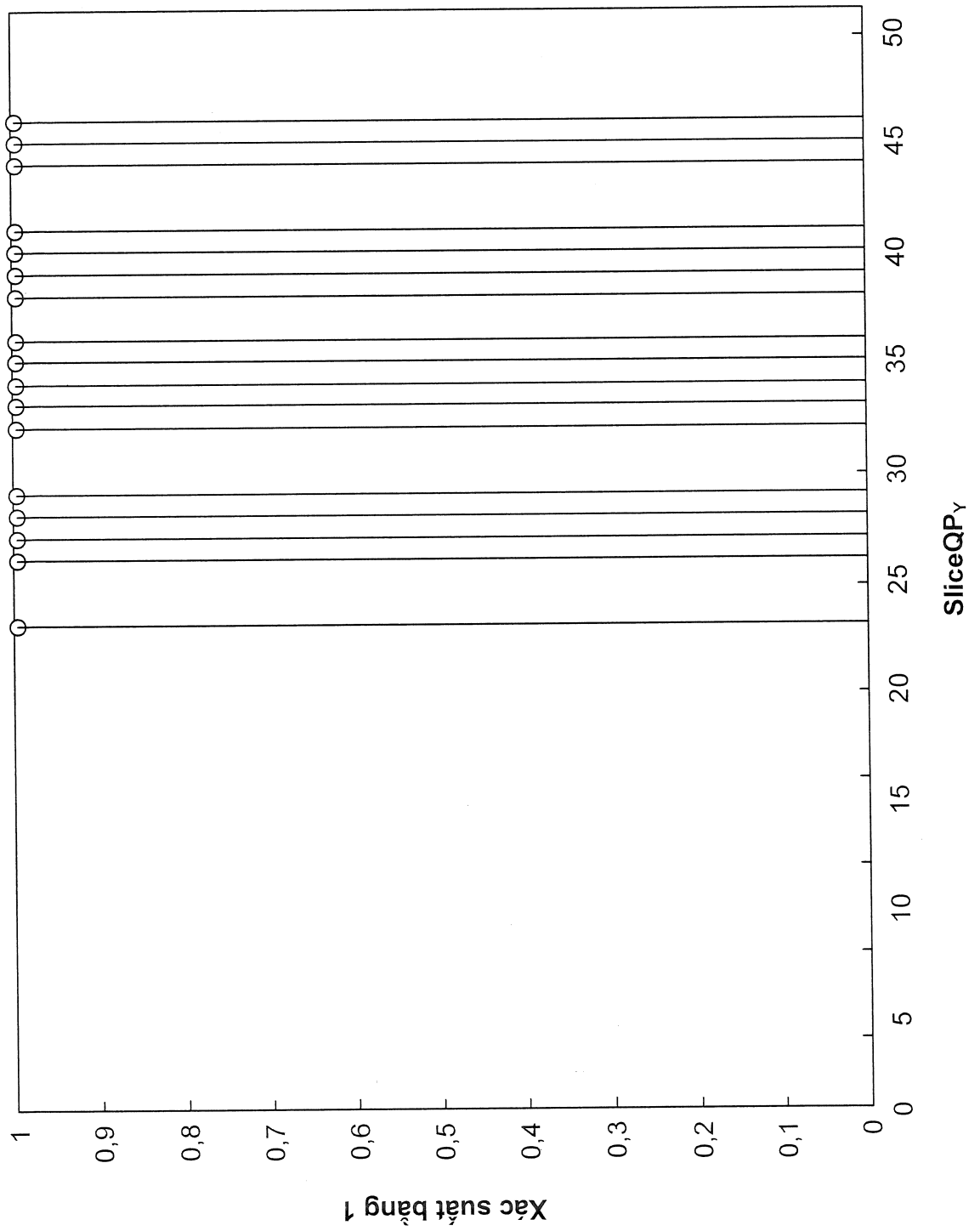


FIG. 18A

24 / 28

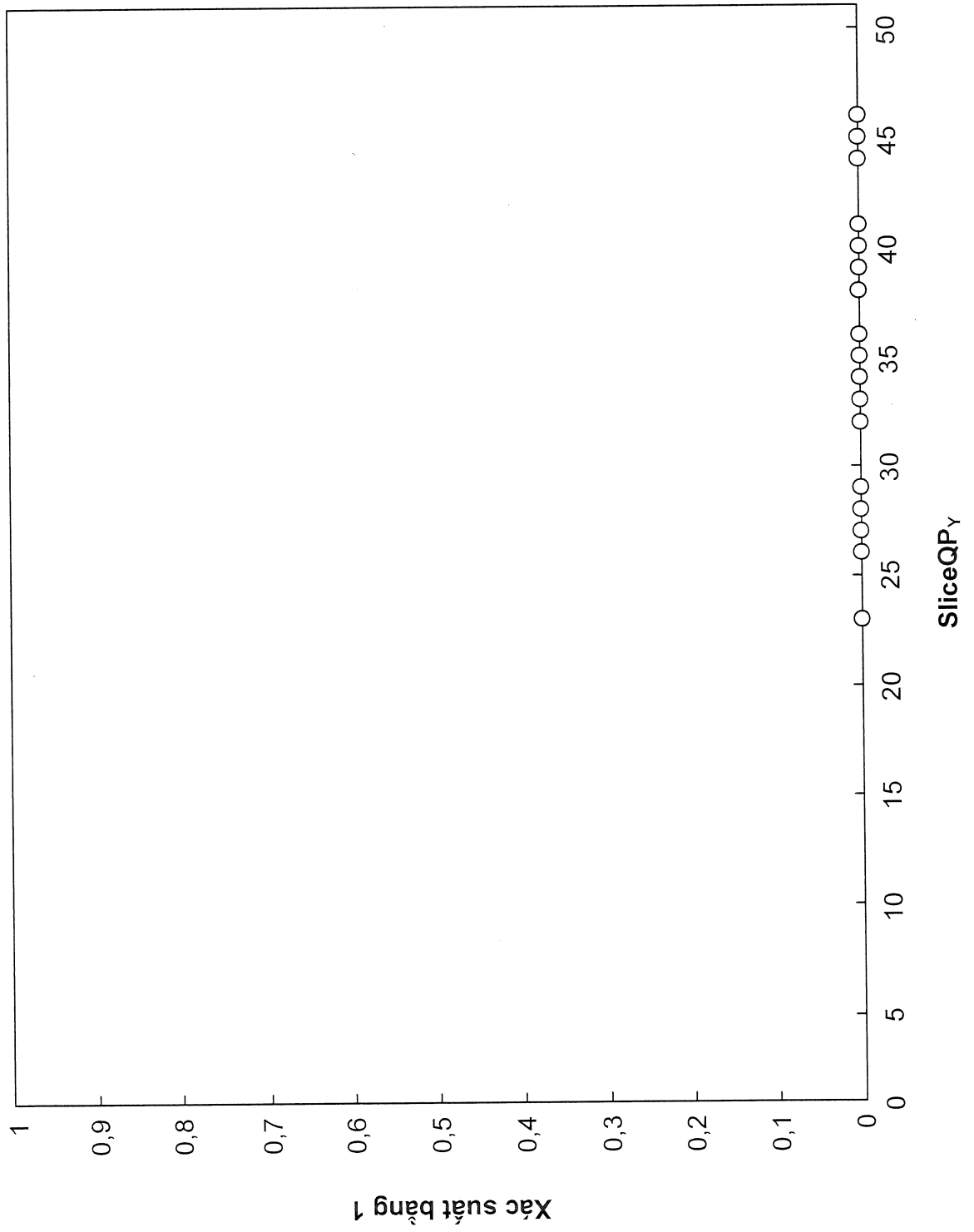


FIG. 18B

25 / 28

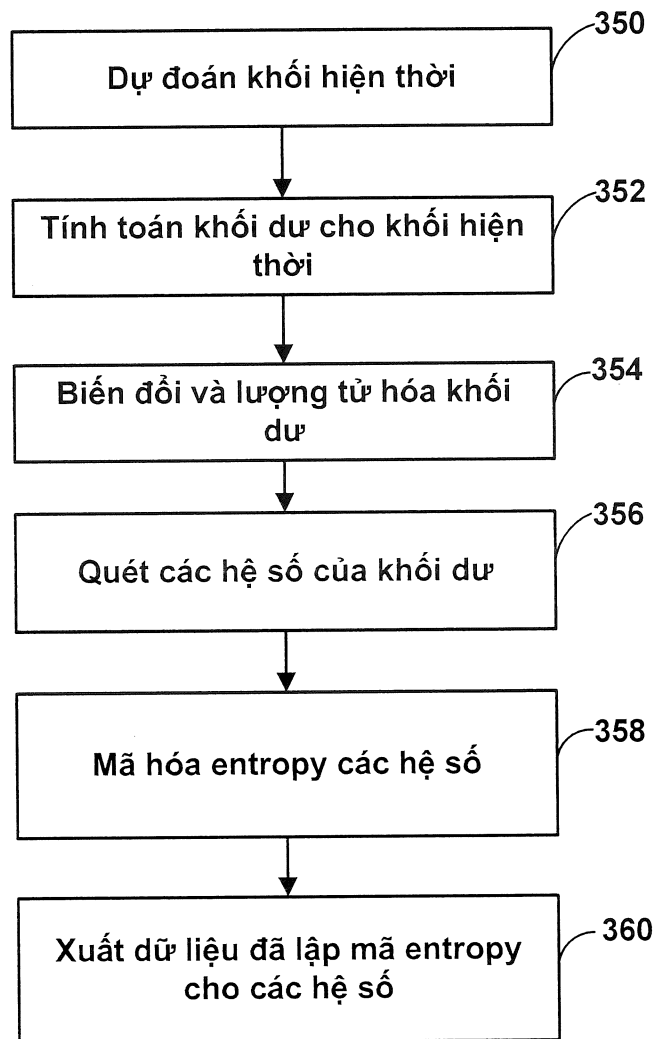


FIG. 19

26 / 28

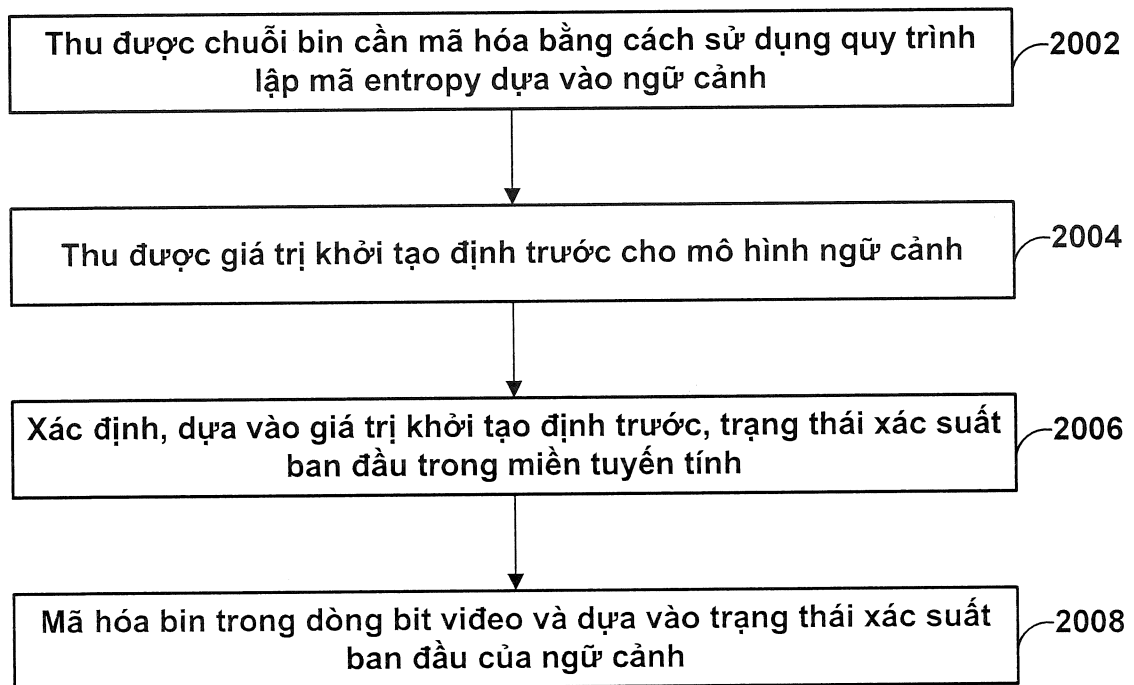


FIG. 20

27 / 28

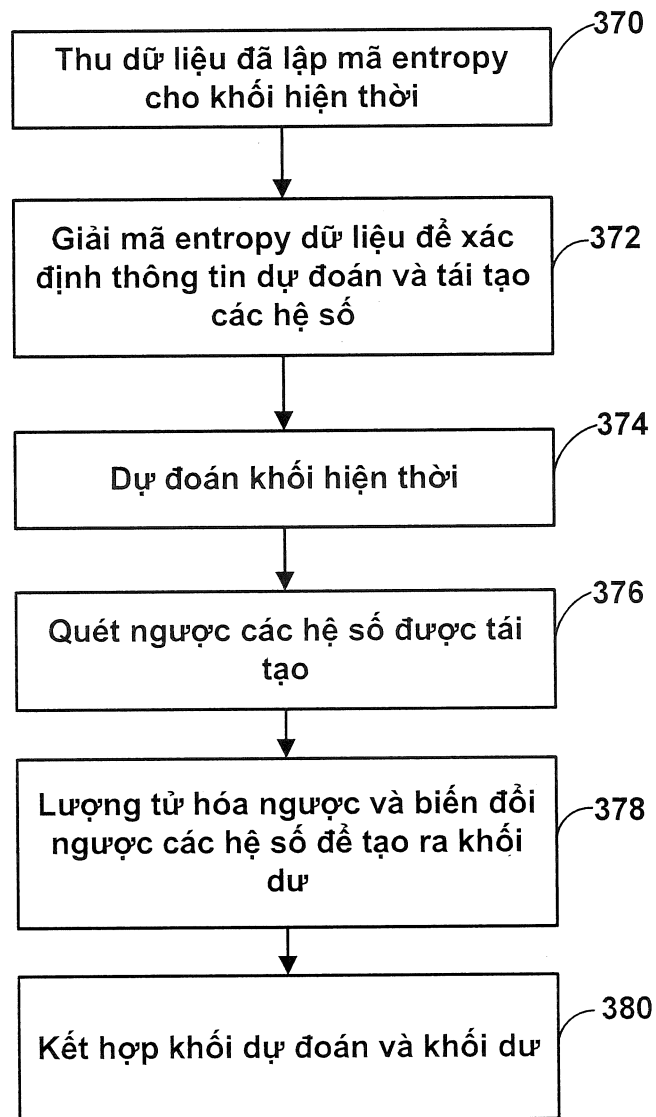


FIG. 21

28 / 28

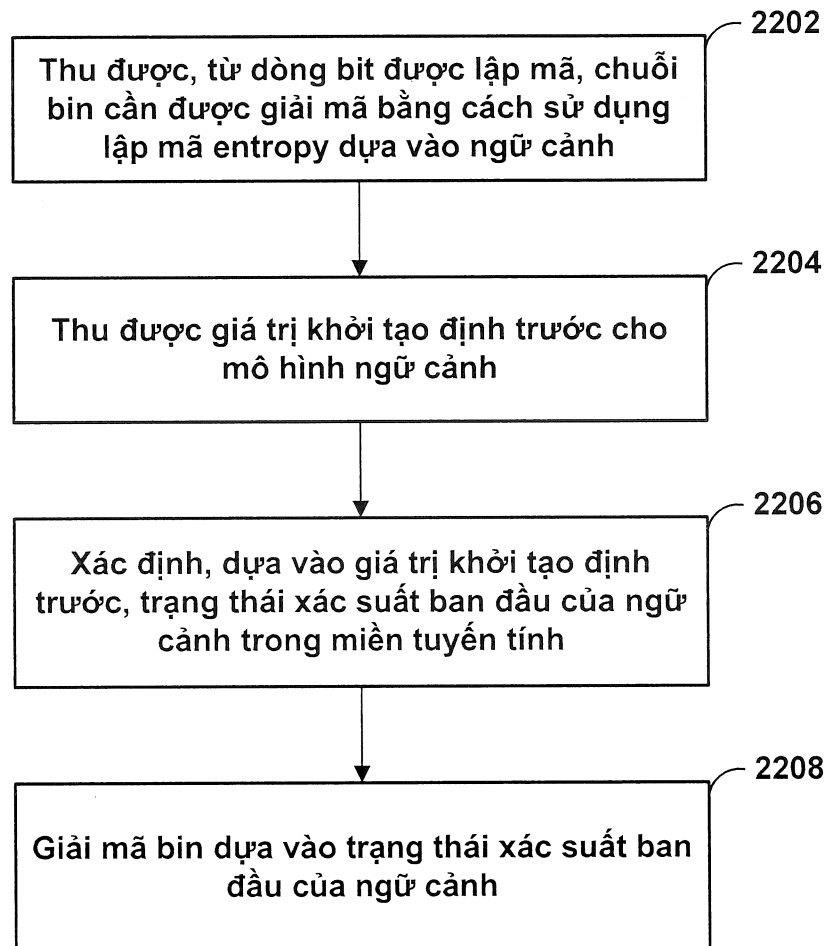


FIG. 22