



(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)



1-0036896

(51)<sup>7</sup> H04H 60/32

(13) B

(21) 1-2019-04504

(22) 18/12/2013

(62) 1-2015-02388

(86) PCT/US2013/076176 18/12/2013

(87) WO 2014/107309 A1 10/07/2014

(30) 61/748,893 04/01/2013 US; 14/035,711 24/09/2013 US

(45) 25/09/2023 426

(43) 25/12/2019 381A

(73) SONY CORPORATION (JP)

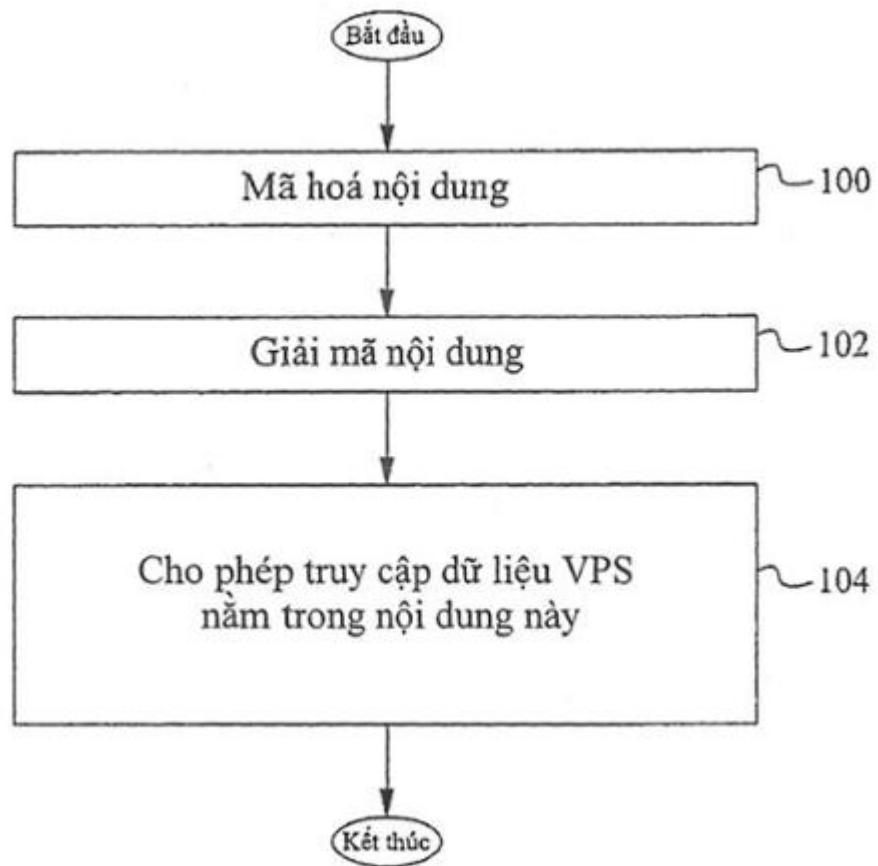
1-7-1 Konan Minato-ku, Tokyo 108-0075, Japan

(72) HAQUE, Munsir (US); TABATABAI, Ali (US).

(74) Công ty TNHH một thành viên Sở hữu trí tuệ VCCI (VCCI-IP CO.,LTD)

(54) THIẾT BỊ XỬ LÝ HÌNH ẢNH, PHƯƠNG PHÁP XỬ LÝ HÌNH ẢNH VÀ BỘ NHỚ KHÔNG TẠM THỜI

(57) Sáng chế đề cập đến phương pháp và thiết bị xử lý hình ảnh. Phương pháp này bao gồm các bước: tạo ra dòng bit bao gồm dữ liệu hình ảnh và cú pháp tập thông số video (video parameter set, viết tắt là VPS); và truyền dòng bit được tạo ra, trong đó cú pháp đồng chỉnh byte trong cú pháp VPS là theo điều kiện của cờ mở rộng VPS, trong đó cú pháp đồng chỉnh byte liên quan đến việc đồng chỉnh byte, trong đó giá trị cờ mở rộng VPS được xác định, trong đó việc đồng chỉnh byte được thực hiện phụ thuộc vào trị số của cờ mở rộng VPS, và trong đó chức năng mở rộng VPS được thực hiện dựa vào việc đồng chỉnh byte.



### **Lĩnh vực kỹ thuật được đề cập**

Sáng chế đề cập đến lĩnh vực mã hóa video. Cụ thể hơn, sáng chế đề cập đến phương pháp mã hóa video hiệu quả cao.

### **Tình trạng kỹ thuật của sáng chế**

Tập thông số video (Video Parameter set - VPS) đã được thêm vào dưới dạng siêu dữ liệu để mô tả các đặc điểm tổng quát của các chuỗi video được mã hóa, bao gồm những sự phụ thuộc giữa các lớp con theo thời gian. Mục đích chính của việc này là để cho phép khả năng mở rộng tương thích của tiêu chuẩn xét về mặt báo hiệu tại lớp hệ thống, ví dụ, khi lớp cơ sở của luồng bit khả biến mở rộng hoặc luồng bit đa điểm quan sát trong tương lai cần phải có thể được giải mã bằng bộ giải mã kế thừa, nhưng thông tin bổ sung về cấu trúc luồng bit mà chỉ liên quan đến bộ giải mã cải tiến sẽ bị bỏ qua.

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

Các cấu trúc cú pháp VPS và vps\_extension( ) được cập nhật với một số thao tác dọn sạch đối với phần mở rộng kỹ thuật mã hóa video hiệu suất cao (HEVC Extensions) trong các vùng mã hóa video khả biến, mã hóa đa điểm quan sát và mã hóa video 3D. Ngoài ra, bốn tùy chọn bổ sung các cú pháp để hỗ trợ các chuỗi video hỗn hợp trong các lớp khác nhau đối với phần mở rộng VPS cũng được mô tả.

Một khía cạnh của sáng chế đề xuất phương pháp được lập trình trong bộ nhớ bất biến của thiết bị, phương pháp này bao gồm bước giải mã nội dung và truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp. Hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video theo điều kiện của cờ mở rộng tập thông số video. Các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video theo điều kiện của cờ mở rộng tập thông số video. Dữ liệu tập thông số video này bao gồm giá trị

các bit đuôi phần tải hữu ích của chuỗi byte thô. Dữ liệu tập thông số video này được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte. Dữ liệu tập thông số video này được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ mã hóa giải mã (codec) hỗn hợp nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cờ hiển thị video hỗn hợp nguồn. Phương pháp này còn bao gồm bước mã hóa nội dung. Thiết bị nêu trên bao gồm máy tính cá nhân, máy tính xách tay, máy tính trạm, máy chủ, máy tính lớn, máy tính cầm tay, máy trợ giúp cá nhân kỹ thuật số, điện thoại di động, đồ dùng thông minh, máy chơi trò chơi, máy ảnh số, máy quay phim số, điện thoại tích hợp máy ảnh, điện thoại thông minh, máy chơi nhạc cầm tay, máy tính bảng, thiết bị di động, máy phát video, đầu ghi/phát đĩa video, đầu ghi/phát đĩa độ phân giải cao, đầu ghi/phát đĩa độ phân giải siêu cao, máy thu hình, hệ thống giải trí gia đình, hoặc đồng hồ thông minh.

Khía cạnh khác của sáng chế đề xuất phương pháp được lập trình trong bộ nhớ bất biến của thiết bị, phương pháp này bao gồm bước cung cấp nội dung và cho phép truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp. Hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video. Các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video theo điều kiện của cờ mở rộng tập thông số video. Dữ liệu tập thông số video này bao gồm giá trị các bit đuôi phần tải hữu ích của chuỗi byte thô. Dữ liệu tập thông số video này được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte. Dữ liệu tập thông số video này được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ codec hỗn hợp nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp

nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cờ hiển thị video hỗn hợp nguồn. Phương pháp này còn bao gồm bước mã hóa nội dung. Thiết bị nêu trên bao gồm máy tính cá nhân, máy tính xách tay, máy tính trạm, máy chủ, máy tính lớn, máy tính cầm tay, máy trợ giúp cá nhân kỹ thuật số, điện thoại di động, đồ dùng thông minh, máy chơi trò chơi, máy ảnh số, máy quay phim số, điện thoại tích hợp máy ảnh, điện thoại thông minh, máy chơi nhạc cầm tay, máy tính bảng, thiết bị di động, máy phát video, đầu ghi/phát đĩa video, đầu ghi/phát đĩa độ phân giải cao, đầu ghi/phát đĩa độ phân giải siêu cao, máy thu hình, hệ thống giải trí gia đình, hoặc đồng hồ thông minh.

Khía cạnh khác của sáng chế đề xuất thiết bị bao gồm bộ nhớ bất biến để chứa ứng dụng, ứng dụng này để giải mã nội dung và truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp, và thành phần xử lý được ghép vào bộ nhớ này, thành phần xử lý này được tạo cấu hình để xử lý ứng dụng này. Hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video. Các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video theo điều kiện của cờ mở rộng tập thông số video. Dữ liệu tập thông số video này bao gồm giá trị các bit đuôi phần tải hữu ích của chuỗi byte thô. Dữ liệu tập thông số video này được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte. Dữ liệu tập thông số video này được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ codec hỗn hợp nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp nguồn. Dữ liệu tập thông số video này được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cờ hiển thị video hỗn hợp nguồn. Thiết bị này còn bao gồm thành phần để mã hóa nội dung.

### **Mô tả vắn tắt các hình vẽ**

Fig.1 là hình vẽ thể hiện lưu đồ của phương pháp vận dụng cú pháp VPS đã được sửa đổi, theo một số phương án của sáng chế.

Fig.2 là hình thể hiện sơ đồ khối của thiết bị tính toán ví dụ được tạo cấu hình để thực hiện phương pháp VPS đã được sửa đổi theo một số phương án của sáng chế.

Fig.3 là hình vẽ thể hiện sơ đồ tổng quát của bộ mã hóa HEVC theo một số phương án của sáng chế.

Fig.4 là hình vẽ thể hiện sơ đồ tổng quát của bộ giải mã EDEVC theo một số phương án của sáng chế.

### Mô tả chi tiết sáng chế

Cấu trúc cú pháp của tập thông số video (Video Parameter set - VPS) trong đặc tả dự thảo tiêu chuẩn quốc tế (Draft International Standard - DIS) hiện tại dành cho mã hóa video hiệu quả cao (High Efficiency Video Coding - HEVC) được thể hiện dưới đây.

| video_parameter_set_rbsp() {                          | Ký hiệu mô tả |
|-------------------------------------------------------|---------------|
| vps_video_parameter_set_id                            | u(4)          |
| vps_reserved_three_2bits                              | u(2)          |
| vps_reserved_zero_6bits                               | u(6)          |
| vps_max_sub_layers_minus1                             | u(3)          |
| vps_temporal_id_nesting_flag                          | u(1)          |
| vps_reserved_Oxffff_16bits                            | u(16)         |
| profile_tier_level(1, vps_max_sub_layers_minus1)      |               |
| bit_rate_pic_rate_info( 0, vps_max_sub_layers_minus1) |               |
| vps_sub_layer_ordering_info_present_flag              | u(1)          |

|                                                                                                                                |       |
|--------------------------------------------------------------------------------------------------------------------------------|-------|
| for (i= (vps_sub_laver_ordering_info_present_flag ?<br>0: vps_max_sub_layers_minus1);<br>1<= vps_max_sub_layers_minus1; i++) { |       |
| vps_max_dec_pic_buffering[i]                                                                                                   | ue(v) |
| vps_max_num_reorder_pics [i]                                                                                                   | ue(v) |
| vps_max_latency_increase[i]                                                                                                    | ue(v) |
| }                                                                                                                              |       |
| vps_max_nuh_reserved_zero_layer_id                                                                                             | u(6)  |
| vps_num_op_sets_minus1                                                                                                         |       |
| for(i=1;i<vps_num_op_sets_minus1;i++)                                                                                          |       |
| operation_point_set(i)                                                                                                         |       |
| vps_num_hrd_parameters                                                                                                         | ue(v) |
| for(i=0;i<num_hrd_parameters;i++) {                                                                                            |       |
| hrd_op_set_idx[i]                                                                                                              | ue(v) |
| if(i>0)                                                                                                                        |       |
| cprms_present_flag[i]                                                                                                          | u(1)  |
| hrd_parameters(cprms_present_flag[i], vps_max_sub_lavers_minus1)                                                               |       |
| }                                                                                                                              |       |
| vps_extension_flag                                                                                                             | u(1)  |
| if(vps_extension_flag)                                                                                                         |       |
| while(more_rbsp_data())                                                                                                        |       |
| vps_extension_data_flag                                                                                                        | u(1)  |
| rbsp_trailing_bits()                                                                                                           |       |
| }                                                                                                                              |       |

Phần if(vps\_extension\_flag) nêu trên được cập nhật dưới đây để hỗ trợ các

phần mở rộng HEVC khác nhau.

| video_parameter_set_rbspO {                                                                                                     | Ký hiệu mô tả |
|---------------------------------------------------------------------------------------------------------------------------------|---------------|
| vps_video_parameter_set_id                                                                                                      | u(4)          |
| vps_reserved_three_2bits                                                                                                        | u(2)          |
| vps_max_layers_minus1//vps_reserved_zero_6bits in version-1                                                                     | u(6)          |
| vps_max_sub_layers_minus1                                                                                                       | u(3)          |
| vps_temporal_id_nesting_flag                                                                                                    | u(1)          |
| next_essential_info_byte_offset//vps_reserved_Oxffff_16bits in                                                                  | u(16)         |
| profile_tier_level(1,vps_max_sub_layers_minus1)                                                                                 |               |
| bit_rate_pic_rate_info( 0, vps_max_sub_layers_minus1)                                                                           |               |
| vps_sub_layer_ordering_info_present_flag                                                                                        | u(1)          |
| for (i= (vps_sub_layer_ordering_info_present_flag ?<br>0 : vps_max_sub_layers_minus1);<br>1 <= vps_max_sub_layers_minus1;i++) { |               |
| vps_max_dec_pic_buffering[i]                                                                                                    | ue(v)         |
| vps_max_num_reorder_pics [i]                                                                                                    | ue(v)         |
| vps_max_latency_increase[i]                                                                                                     | ue(v)         |
| }                                                                                                                               |               |
| vps_max_nuh_reserved_zero_layer_id                                                                                              | u(6)          |
| vps_num_op_sets_minus1                                                                                                          |               |
| for(i=1;i<vps_num_op_sets_minus1;i++)                                                                                           |               |
| operation_point_set(i)                                                                                                          |               |
| vps_num_hrd_parameters                                                                                                          | ue(v)         |
| for(i=0;i<num_hrd_parameters;i++) {                                                                                             |               |
| hrd_op_set_idx[i]                                                                                                               | ue(v)         |
| if(i>O)                                                                                                                         |               |

|                                                                  |      |
|------------------------------------------------------------------|------|
| cprms_present_flag[i]                                            | u(1) |
| hrd_parameters(cprms_present_flag[i], vps_max_sub_lavers_minus1) |      |
| }                                                                |      |
| vps_extension_flag /bit_equal_to_one in JCTVC_K1007_v1           | u(1) |
| if(vps_extension_flag) {                                         |      |
| while(!byte_aligned())                                           |      |
| vps_extension_byte_alignment_reserved_one_bit                    | u(1) |
| vps_extension( )                                                 |      |
| }                                                                |      |
| rbsp_trailing_bits()                                             |      |
| }                                                                |      |

Các bản cập nhật này bao gồm:

1) “vps\_reserved\_zero\_6bits in version-1” được thay đổi thành “vps\_max\_num\_layers\_minus1”. Tuy nhiên, trong cấu trúc cú pháp vps\_extension( ) thì thông số này được sử dụng dưới dạng vps\_max\_layers\_minus1. Điều này cũng thống nhất với thông số tương tự “vps\_max\_sub\_layers\_minus1 in version-1”. Do đó, vps\_max\_layers\_minus1 được sử dụng trong VPS.

2) Đối với các phần mở rộng HEVC, thì cấu trúc cú pháp “vps\_extension( )” như được khai báo trước đây được chèn vào trong VPS theo điều kiện của vps\_extension\_flag.

3) “vps\_reserved\_0xffff\_16bits in version-1” được thay bằng “next\_essential\_info\_byte\_offset”. Thông số cú pháp mới này có thể được dùng để định vị thông tin thiết yếu khả dụng trong cấu trúc cú pháp “vps\_extension( )”

4) Thông số cú pháp next\_essential\_info\_byte\_offset cũng bao gồm các cú pháp liên quan đến việc đồng chỉnh byte gần đỉnh của cấu trúc cú pháp “vps\_extension( )”. Các cú pháp đồng chỉnh byte này được lấy ra và được đặt bên trong VPS theo điều kiện của vps\_extension\_flag, ngay trước cấu trúc

“vps\_syntax( )” cho sạch.

5) “rbsp\_trailing\_bits” chứa một lượng bit đồng chỉnh byte đối với VPS\_rbsp( ), và lượng bit này có thể được xác định như sau:  $nByteAlignmentBits = ((vps\_extension\_length + 7) / 8) * 8 - vps\_extension\_length$ , trong đó  $vps\_extension\_length$  là tổng số bit trong  $vps\_extension( )$ , và con số này có thể tính được dễ dàng vì cấu trúc cú pháp  $vps\_extension( )$  có chứa các bit số nguyên không dấu có độ dài cố định.

Cấu trúc cú pháp của  $vps\_extension( )$  được thể hiện dưới đây:

| $vps\_extension( ) \{$                                      | Ký hiệu mô tả |
|-------------------------------------------------------------|---------------|
| $while(!byte\_aligned())$                                   |               |
| $vps\_extension\_byte\_alignment\_reserved\_one\_bit$       | u(1)          |
| $avc\_base\_codec\_flag$                                    | u(1)          |
| $scalability\_mask$                                         | u(16)         |
| $for(i=0; i < NumScalabilityTypes; i++) \{$                 |               |
| $dimension\_id\_len\_minus1[i]$                             | u(3)          |
| $\}$                                                        |               |
| $vps\_nuh\_layer\_id\_present\_flag$                        | u(1)          |
| $//layer\ specific\ information$                            |               |
| $for\ (i=1; i \leq vps\_max\_layers\_minus1; i++) \{$       |               |
| $//mapping\ of\ layer\ id\ to\ scalability\ dimension\ IDs$ |               |
| $if(vps\_nuh\_layer\_id\_present\_flag)$                    |               |
| $layer\_id\_in\_nuh[i]$                                     | u(6)          |
| $num\_dimensions\_minus1[i]$                                | u(4)          |
| $for\ (j=0; j \leq num\_dimensions\_minus1; j++) \{$        |               |
| $dimension\_type[i][j]$                                     | u(4)          |
| $dimension\_id[i][j]$                                       | u(v8)         |

|                                                |      |
|------------------------------------------------|------|
| }                                              |      |
| }                                              |      |
| for(i=1;i<vps_max_layers_minus1;i++) {         |      |
| profile_tier_level(1,vps_max_sublayers_minus1) |      |
| for(i=1;i<vps_max_layers_minus1;i++) {         |      |
| //layer dependency                             |      |
| num_direct_ref_layers[i]                       | u(6) |
| for(j=0;j<num_direct_ref_lavers[i];j++)        |      |
| ref_layer_id[i][j]                             | u(6) |
| }                                              |      |

Trong khi xem xét cấu trúc cú pháp và phần mô tả ngữ nghĩa nêu trên, thì một số vấn đề được đề cập dưới đây dưới dạng hoạt động dọn sạch hoặc nhằm mục đích biên tập.

Đối với thông số cú pháp scalability\_mask thì bảng thông số sau đây được liệt kê, trong khi các mục nhập tương ứng thì thể hiện vị trí bit đối với dữ liệu kiểu thanh ghi của “scalability\_mask”, nhưng nó không được mô tả rõ.

| Scalability_mask | Chiều khả biến        |
|------------------|-----------------------|
| 0                | không có (HEVC cơ sở) |
| 1                | không gian            |
| 2                | chất lượng            |
| 3                | chiều sâu             |
| 4                | đa điểm quan sát      |
| 5                | chưa xác định         |
| 6-15             | Để dành               |

Thay cho bảng nêu trên đối với scalability\_mask, thì phần mô tả kiểu thanh ghi sau đây sẽ cho phép hiểu rõ hơn:

|               |                  |           |            |            |          |                |   |   |   |
|---------------|------------------|-----------|------------|------------|----------|----------------|---|---|---|
| 5             | 4                | 3         | 2          | 1          | 0        | Các vị trí bit |   |   |   |
| chưa xác định | đa điểm quan sát | chiều sâu | chất lượng | không gian | không có | chiều khả biến |   |   |   |
| 15            | 14               | 13        | 12         | 11         | 10       | 9              | 8 | 7 | 6 |
| để dành       |                  |           |            |            |          |                |   |   |   |

Ngoài ra, cấu trúc cú pháp đã được sửa đổi của vps\_extension( ) dưới đây biểu thị một số hoạt động dọn dẹp có ích.

|                                                     |               |
|-----------------------------------------------------|---------------|
| vps_extension( ) {                                  | Ký hiệu mô tả |
| avc_base_codec_flag                                 | u(1)          |
| scalability_mask                                    | u(16)         |
| for(i=0;i<NumScalabilityTypes;i++) {                |               |
| dimension_id_len_minus1[i]                          | u(3)          |
| }                                                   |               |
| vps_nuh_layer_id_present_flag                       | u(1)          |
| //layer specific information                        |               |
| for (i=0; i<=vps_max_layers_minus1;i++) {           |               |
| //mapping of layer id to scalabililty dimension IDs |               |
| if (vps_nuh_layer_id_present_flag)                  |               |
| layer_id_in_nuh[i]                                  | u(6)          |
| for (j=0; j<=NumScalabilityTypes; j++) {            |               |
| v = dimension_id_len_minus1[j]+1                    |               |
| dimension_id[i] [j ]                                | u(v)          |
| }                                                   |               |
| }                                                   |               |
| for(i=1;i<vps_max_layers_minus1;i++)                |               |

|                                                |      |
|------------------------------------------------|------|
| profile_tier_level(1,vps_max_sublayers_minus1) |      |
| for(i=0;i<vps_max_layers_minus1;i++) {         |      |
| //layer dependency                             |      |
| num_direct_ref_layers[i]                       | u(6) |
| for(j=0; j<num_direct_ref_lavers[i];j++)       |      |
| ref_layer_id[i][j]                             | u(6) |
| }                                              |      |
| }                                              |      |

1) Các cú pháp liên quan đến việc đồng chỉnh byte được loại bỏ.

2) Thông số chưa xác định “num\_dimensions\_minus1” được thay bằng “NumScalabilityTypes”.

3) Hai trong số các vòng lặp “for” đã được sửa đổi để bắt đầu từ 0 thay vì 1. Ví dụ, đối với “layer specific info”, thì là vòng lặp “for(i=1; i<=vps\_max\_layers\_minus1; i++){...}”, nếu chuỗi video đã mã hóa có chứa lớp hình nổi với lớp chiều sâu (2+1=3 lớp), thì là có 3 lớp. Do đó, vps\_max\_layers\_minus1=3 - 1=2. Nếu theo vòng lặp “for” hiện tại, thì

$i = 1 \Rightarrow \text{Layer-1} = \text{depth-view};$

$i = 2 \text{ (vps\_max\_layers\_minus1)} \Rightarrow \text{Layer-2} = \text{View-1}$

where is  $i=3 \Rightarrow \text{Layer-3} = \text{View-2?}$

Sau khi hiệu chỉnh thì vòng lặp này là “for(i=0; i<=vps\_max\_layers\_minus1; i++){...}” đối với hai trường hợp, ngoại trừ “profile\_tier\_level(...)”.

3) Phần mô tả ngữ nghĩa cho thông số cú pháp “dimension\_id[i][j]” cũng không khả dụng đối với “chiều khả biến” và “dimension\_id\_len\_minus1[j]”. Một bảng ví dụ được thể hiện dưới đây để thể hiện mối quan hệ giữa những thông số này:

| Chiều khả biến | dimension_id | dimension_idjen(example) |
|----------------|--------------|--------------------------|
|----------------|--------------|--------------------------|

|                       |               |   |
|-----------------------|---------------|---|
| Không có (HEVC cơ sở) | -             | - |
| Không gian            | dependency_id | 4 |
| Chất lượng            | quality_id    | 4 |
| Chiều sâu             | depth_flag    | 1 |
| Đa điểm quan sát      | view_idx      | 4 |
| Chưa xác định         | -             | - |
| Để dành               | -             | - |

Các cấu trúc cú pháp VPS và vps\_extension( ) được cập nhật đối với phần HEVC Extensions trong các vùng mã hóa video khả biến, mã hóa đa điểm quan sát và mã hóa video 3D.

VPS chứa siêu dữ liệu để mô tả các đặc điểm tổng quát của video được mã hóa, cho phép tính tương thích mở rộng tiêu chuẩn xét về mặt báo hiệu lớp hệ thống để bộ giải mã kế thừa có thể bỏ qua thông tin bổ sung về cấu trúc luồng bit mở rộng, cấu trúc cú pháp VPS hiện tại và phần mở rộng của nó chứa thông tin dành cho các chuỗi video đa lớp, trong đó một lớp có thể là lớp cơ sở (dành cho HEVC và các phần mở rộng của nó), hoặc có thể là lớp tăng cường khả biến (về mặt không gian hoặc chất lượng) dành cho SHVC, hoặc lớp quan sát (kết cấu hoặc chiều sâu) để mã hóa đa điểm quan sát phần mở rộng HEVC (MHVC). Tất cả các lớp này cũng có thể có các lớp con của chúng (về mặt thời gian). Mỗi lớp video có thể phụ thuộc vào các lớp thấp hơn lân cận của nó, hoặc có thể là độc lập mà không có sự dự đoán liên lớp nào.

Động lực đằng sau “khả năng hỗ trợ báo hiệu các chuỗi hỗn hợp” là để phát triển các thuộc tính lớp cụ thể mới trong cấu trúc cú pháp mở rộng VPS, như được đề xuất trong “JCTVC-K\_notes”. Cú pháp phân lớp dành cho các lớp độc lập trong VPS có thể được khai thác một cách thích hợp để hỗ trợ nhiều loại nội dung hỗn hợp (các loại nguồn, các kiểu mã hóa, v.v.) trong cấu trúc cú pháp mức cao dành cho các phần mở rộng HEVC.

Một số ví dụ ứng dụng khả thi khi phát sóng hỗn hợp hoặc truyền qua mạng dành cho VPS với 2 lớp là:

Video có cấu trúc khung ở lớp 1 và video có cấu trúc trường ở lớp 2: mỗi lớp

video có SPS tương ứng của nó và hơn nữa. Các bộ giải mã hình chìm (2-D) kế thừa có thể sử dụng mỗi trong số 2 lớp này cho ứng dụng lớp cơ sở của chúng.

Video có cấu trúc khung (2-D, hình chìm) ở lớp 1 và video tương thích khung (đóng gói khung) (3D, hình nổi) ở lớp 2: bộ giải mã hình chìm (2-D) kế thừa sử dụng video lớp 1, còn bộ giải mã video tương thích khung hình nổi (3D) cải tiến thì có thể sử dụng lớp 2 đối với ứng dụng video hình nổi 3D.

Sự kết hợp của các loại video hỗn hợp sẽ có mặt trong các luồng bit video để hỗ trợ hai bộ giải mã này (bộ giải mã kế thừa và bộ giải mã cải tiến).

Những sự kết hợp đó giữa nội dung video đã mã hóa với nhiều loại, nguồn và kiểu mã hóa có thể cùng tồn tại trong các ứng dụng hiện tại và tương lai mà trong đó các bộ giải mã kế thừa hoặc các bộ giải mã cải tiến sẽ loại bỏ luồng bit tương ứng để giải mã.

Những đặc điểm dựa trên lớp vốn có của VPS có thể được mở rộng để hỗ trợ các ứng dụng này, ít nhất là đối với các giải pháp mức hệ thống (chèn quảng cáo, ghép nối, v.v.).

Có bốn tùy chọn khả thi để bổ sung cú pháp cho việc báo hiệu các chuỗi hỗn hợp trong phần mở rộng VPS.

Tùy chọn 1:2 cú pháp để hỗ trợ báo hiệu các chuỗi hỗn hợp

Tùy chọn 2: Tùy chọn 1 cộng với thông số cú pháp `source_mixed_codec_flag`

Tùy chọn 3: Tùy chọn 2 cộng với thông số cú pháp `mixed_video_present_flag`

Tùy chọn 4: hỗ trợ các ứng dụng cụ thể cộng với thông số cú pháp `mixed_video_present_flag` và dùng nó làm điều kiện cho cú pháp/ngữ nghĩa được bổ sung đối với tùy chọn này.

Tùy chọn 1

|                                  |                   |
|----------------------------------|-------------------|
| <code>vps_extension( ) {</code>  | Ký hiệu mô tả     |
| <code>avc_base_codec_flag</code> | <code>u(1)</code> |

|                                                    |       |
|----------------------------------------------------|-------|
| source_mixed_video_type                            | u(4)  |
| scalability_mask                                   | u(16) |
| for(i=0;i<NumScalabilityTypes;i++) {               |       |
| dimension_id_len_minus1[i]                         | u(3)  |
| }                                                  |       |
| vps_nuh_layer_id_present_flag                      | u(1)  |
| //layer specific information                       |       |
| for (i=0; i<=vps_max_layers_minus1;i++) {          |       |
| //mapping of layer id to scalability dimension IDs |       |
| if (vps_nuh_layer_id_present_flag)                 |       |
| layer_id_in_nuh[i]                                 | u(6)  |
| for (j=0; j<=NumScalabilityTypes; j++) {           |       |
| v = dimension_id_len_minus1[j]+1                   |       |
| dimension_id[i] [j ]                               | u(v)  |
| }                                                  |       |
| if(source_mixed_video_type > 0)                    |       |
| source_mixed_video_idc[i]                          | u(4)  |
| }                                                  |       |
| for(i=0;i<vps_max_layers_minus1;i++)               |       |
| profile_tier_level(1,vps_max_sublayers_minus1)     |       |
| for(i=0;i<vps_max_layers_minus1;i++) {             |       |
| //layer dependency                                 |       |
| num_direct_ref_layers[i]                           | u(6)  |
| for(j=0; j<num_direct_ref_layers[i];j++)           |       |
| ref_layer_id[i][j]                                 | u(6)  |

|   |  |
|---|--|
| } |  |
| } |  |

Phần mô tả ngữ nghĩa đối với hai thông số cú pháp được bổ sung này được thể hiện trong các bảng dưới đây:

| Source_mixed_video_type | Các nguồn kết hợp video hỗn hợp (thứ nhất/thứ hai/thứ ba...) |
|-------------------------|--------------------------------------------------------------|
| 0                       | Không có                                                     |
| 1                       | Khung / Trường                                               |
| 2                       | Khung / Tương thích khung                                    |
| 3                       | Trường / Tương thích khung                                   |
| 4                       | Tương thích khung / Hình nổi                                 |
| 5                       | Khung / Trường / Tương thích khung                           |
| 6                       | Khung / Tương thích khung / Hình nổi                         |
| 7...15                  | Chưa xác định                                                |

| source_mixed_video_idc | Thứ tự kết hợp video hỗn hợp                                                |
|------------------------|-----------------------------------------------------------------------------|
| 0                      | Loại video nguồn thứ nhất có mặt trong lớp i                                |
| 1                      | Loại video nguồn thứ hai có mặt trong lớp i                                 |
| 2                      | Hỗn hợp các loại video nguồn thứ nhất và thứ hai có mặt trong lớp i         |
| 3                      | Loại video nguồn thứ ba có mặt trong lớp i                                  |
| 4                      | Hỗn hợp các loại video nguồn thứ nhất, thứ hai và thứ ba có mặt trong lớp i |
| 5...15                 | Chưa xác định                                                               |

Tùy chọn 2

|                         |               |
|-------------------------|---------------|
| vps_extension( ) {      | Ký hiệu mô tả |
| avc_base_codec_flag     | u(1)          |
| source_mixed_video_type | u(4)          |
| scalability_mask        | u(16)         |

|                                                    |      |
|----------------------------------------------------|------|
| for(i=0;i<NumScalabilityTypes;i++) {               |      |
| dimension_id_len_minus1[i]                         | u(3) |
| }                                                  |      |
| vps_nuh_layer_id_present_flag                      | u(1) |
| //layer specific information                       |      |
| for (i=0; i<=vps_max_layers_minus1;i++) {          |      |
| //mapping of layer id to scalability dimension IDs |      |
| if (vps_nuh_layer_id_present_flag)                 |      |
| layer_id_in_nuh[i]                                 | u(6) |
| for (j=0; j<=NumScalabilityTypes; j++) {           |      |
| v = dimension_id_len_minus1[j]+1                   |      |
| dimension_id[i] [j ]                               | u(v) |
| }                                                  |      |
| if(source_mixed_video_type > 0) {                  |      |
| source_mixed_video_idc[i]                          | u(4) |
| source_mixed_codec_flag[i]                         | u(1) |
| }                                                  |      |
| }                                                  |      |
| for(i=0;i<vps_max_layers_minus1;i++)               |      |
| profile_tier_level(1,vps_max_sublayers_minus1)     |      |
| for(i=0;i<vps_max_layers_minus1;i++) {             |      |
| //layer dependency                                 |      |
| num_direct_ref_layers[i]                           | u(6) |
| for(j=0; j<num_direct_ref_layers[i];j++)           |      |
| ref_layer_id[i][j]                                 | u(6) |

Phần mô tả ngữ nghĩa đối với cú pháp mới được bổ sung là giống như `avc_base_codec_flag` như sau:

| <code>avc_base_codec_flag / source_mixed_codec_flag</code> | Kiểu mã hóa video hỗn hợp |
|------------------------------------------------------------|---------------------------|
| 0                                                          | HEVC                      |
| 1                                                          | AYC                       |

### Tùy chọn 3

| <code>vps_extension( ) {</code>                                 | Ký hiệu mô tả      |
|-----------------------------------------------------------------|--------------------|
| <code>avc_base_codec_flag</code>                                | <code>u(1)</code>  |
| <code>mixed_video_present_flag</code>                           | <code>u(1)</code>  |
| <code>if(mixed_video_present_flag)</code>                       |                    |
| <code>source_mixed_video_type</code>                            | <code>u(4)</code>  |
| <code>scalability_mask</code>                                   | <code>u(16)</code> |
| <code>for(i=0;i&lt;NumScalabilityTypes;i++) {</code>            |                    |
| <code>dimension_id_len_minus1[i]</code>                         | <code>u(3)</code>  |
| <code>}</code>                                                  |                    |
| <code>vps_nuh_layer_id_present_flag</code>                      | <code>u(1)</code>  |
| <code>//layer specific information</code>                       |                    |
| <code>for (i=0; i&lt;=vps_max_layers_minus1;i++) {</code>       |                    |
| <code>//mapping of layer id to scalability dimension IDs</code> |                    |
| <code>if (vps_nuh_layer_id_present_flag)</code>                 |                    |
| <code>layer_id_in_nuh[i]</code>                                 | <code>u(6)</code>  |
| <code>for (j=0; j&lt;=NumScalabilityTypes; j++) {</code>        |                    |
| <code>v = dimension_id_len_minus1[j]+1</code>                   |                    |
| <code>dimension_id[i] [j ]</code>                               | <code>u(v)</code>  |

|                                                |      |
|------------------------------------------------|------|
| }                                              |      |
| if(mixed_video_present_flag) {                 |      |
| source_mixed_video_idc[i]                      | u(4) |
| source_mixed_codec_flag[i]                     | u(1) |
| }                                              |      |
| }                                              |      |
| for(i=0;i<vps_max_layers_minus1;i++)           |      |
| profile_tier_level(1,vps_max_sublayers_minus1) |      |
| for(i=0;i<vps_max_layers_minus1;i++) {         |      |
| //layer dependency                             |      |
| num_direct_ref_layers[i]                       | u(6) |
| for(j=0; j<num_direct_ref_layers[i];j++)       |      |
| ref_layer_id[i][j]                             | u(6) |
| }                                              |      |
| }                                              |      |

Thông số cú pháp mixed\_video\_present\_flag bằng 1 cho thấy sự hiện diện của video hỗn hợp trong các lớp VPS. Nếu thông số này bằng 0 thì các chuỗi hỗn hợp này không có mặt trong các lớp này.

Sự hiện diện của cờ này có thể cho phép tiết kiệm một số bit được bổ sung trong trường hợp “không có loại video hỗn hợp”.

| Source_mixed_video_type | Các nguồn kết hợp video hỗn hợp (thứ nhất/thứ hai/thứ ba...) |
|-------------------------|--------------------------------------------------------------|
| 0                       | Khung / Trường                                               |
| 1                       | Khung / Tương thích khung                                    |
| 2                       | Trường / Tương thích khung                                   |
| 3                       | Tương thích khung / Hình nổi                                 |
| 4                       | Khung / Trường / Tương thích khung                           |

|        |                                      |
|--------|--------------------------------------|
| 5      | Khung / Tương thích khung / Hình nổi |
| 6...15 | Chưa xác định                        |

## Tùy chọn 4

|                                                     |               |
|-----------------------------------------------------|---------------|
| vps_extension( ) {                                  | Ký hiệu mô tả |
| avc_base_codec_flag                                 | u(1)          |
| mixed_video_present_flag                            | u(1)          |
| scalability_mask                                    | u(16)         |
| for(i=0;i<NumScalabilityTypes;i++) {                |               |
| dimension_id_len_minus1[i]                          | u(3)          |
| }                                                   |               |
| vps_nuh_layer_id_present_flag                       | u(1)          |
| //layer specific information                        |               |
| for (i=0; i<=vps_max_layers_minus1;i++) {           |               |
| //mapping of layer id to scalabililty dimension IDs |               |
| if (vps_nuh_layer_id_present_flag)                  |               |
| layer_id_in_nuh[i]                                  | u(6)          |
| for (j=0; j<=NumScalabilityTypes; j++) {            |               |
| v = dimension_id_len_minus1[j]+1                    |               |
| dimension_id[i] [j ]                                | u(v)          |
| }                                                   |               |
| if(mixed_video_present_flag) {                      |               |
| source_scan_type_info_idc[i]                        | u(4)          |
| source_2d_3d_info_idc [i]                           | u(1)          |
| }                                                   |               |
| }                                                   |               |

|                                                |      |
|------------------------------------------------|------|
| for(i=0;i<vps_max_layers_minus1;i++)           |      |
| profile_tier_level(1,vps_max_sublayers_minus1) |      |
| for(i=0;i<vps_max_layers_minus1;i++) {         |      |
| //layer dependency                             |      |
| num_direct_ref_layers[i]                       | u(6) |
| for(j=0; j<num_direct_ref_layers[i];j++)       |      |
| ref_layer_id[i][j]                             | u(6) |
| }                                              |      |
| }                                              |      |

Thông số cú pháp `mixed_video_present_flag` bằng 1 cho thấy sự hiện diện của video hỗn hợp trong các lớp VPS. Nếu thông số này bằng 0 thì các chuỗi hỗn hợp này không có mặt trong các lớp này.

Trong trường hợp “`mixed_video_present_flag = 1`”, thì hai cú pháp mới có mặt cho mỗi lớp và hai thông số cú pháp này thích hợp với hai ví dụ ứng dụng cụ thể với các nguồn video hỗn hợp:

Kiểu quét cấu trúc khung / trường và khung đối với video 2D, và

Video cấu trúc đóng gói khung (Frame-Packing Arrangement - FPA) đối với video 3D.

Phần mô tả ngữ nghĩa đối với hai cú pháp mới này được thể hiện trong các bảng dưới đây:

| source_scan_type_info_idc[i] | Các đặc điểm lớp video                               |
|------------------------------|------------------------------------------------------|
| 0                            | Tất cả các ảnh trong lớp i đều có kiểu quét đan xen  |
| 1                            | Tất cả các ảnh trong lớp i đều có kiểu quét liên tục |
| 2                            | Các ảnh trong lớp i có kiểu quét chưa biết           |

|   |                                                                           |
|---|---------------------------------------------------------------------------|
| 3 | Các ảnh trong lớp i có kiểu quét hỗn hợp là quét đan xen và quét liên tục |
|---|---------------------------------------------------------------------------|

| source_2d_3d_info_idc[i] | Các đặc điểm lớp video                                                                                        |
|--------------------------|---------------------------------------------------------------------------------------------------------------|
| 0                        | Tất cả các hình ảnh trong lớp i là các khung 2D                                                               |
| 1                        | Tất cả các hình ảnh trong lớp i đều được đóng gói thành định dạng 3D tương thích khung hoặc cấu trúc đóng gói |
| 2                        | Các hình ảnh trong lớp i có FPA chưa biết                                                                     |
| 3                        | Các hình ảnh trong lớp i có hỗn hợp các khung 3D và 2D tương thích khung                                      |

Sự sửa đổi cú pháp được mô tả ở đây sẽ mở rộng các thuộc tính dựa trên lớp của VPS để tương thích với các ứng dụng khác nhau đang xuất hiện trong các lớp hệ thống (biên tập nội dung, ghép nối, chèn quảng cáo) cũng như trong việc giải mã tổng quát để truyền thông tốt hơn và tích hợp hệ thống tốt hơn.

Fig.1 là hình vẽ thể hiện lưu đồ của phương pháp vận dụng cú pháp VPS đã được sửa đổi, theo một số phương án. Ở bước 100, nội dung (ví dụ, video) được mã hóa. Dữ liệu VPS tạo ra được nhờ sử dụng cú pháp VPS đã được sửa đổi này cũng được bao gồm với nội dung đã mã hóa. Ở bước 102, quá trình giải mã được thực hiện. Ví dụ, video được giải mã bằng cách sử dụng bộ giải mã bất kì. Ở bước 104, dữ liệu VPS có thể được truy cập. Trong khi giải mã thì dữ liệu VPS có thể được sử dụng cho hoạt động xử lý bổ sung. Dữ liệu VPS này chứa các thông tin HEVC Extension khác nhau trong các vùng mã hóa video khả biến, mã hóa đa điểm quan sát và mã hóa video 3D. Theo một số phương án, dữ liệu VPS này bao gồm thông tin báo hiệu hỗn hợp. Theo một số phương án, có thêm các bước bổ sung, hoặc ít bước hơn, được thực hiện. Theo một số phương án, thứ tự của các bước được thay đổi. Fig.2 là hình thể hiện sơ đồ khối của thiết bị tính toán ví dụ được tạo cấu hình để thực hiện phương pháp VPS đã được sửa đổi, theo một số phương án. Thiết bị tính toán 200 có thể được sử dụng để thu thập, lưu trữ, tính toán, xử lý truyền thông và/hoặc hiển thị các thông tin, chẳng hạn hình ảnh và

video. Nói chung, cấu trúc phần cứng phù hợp cho việc thực hiện thiết bị tính toán 200 bao gồm giao diện mạng 202, bộ nhớ 204, bộ xử lý 206, (các) thiết bị vào/ra (Input/Output - I/O) 208, bus 210 và thiết bị lưu trữ 212. Việc lựa chọn bộ xử lý là không bị giới hạn, miễn là bộ xử lý có đủ tốc độ. Bộ nhớ 204 có thể là bộ nhớ máy tính thông thường đã biết. Thiết bị lưu trữ 212 có thể bao gồm đĩa cứng, đĩa CDROM, đĩa CDRW, đĩa DVD, đĩa DVDRW, đĩa Blu-ray®, thẻ nhớ flash hoặc thiết bị lưu trữ bất kì khác. Thiết bị tính toán 200 có thể bao gồm một hoặc nhiều giao diện mạng 202. Ví dụ về giao diện mạng này bao gồm các mạng được nối với mạng Ethernet hoặc loại mạng LAN khác. (Các) thiết bị I/O 208 có thể bao gồm một hoặc nhiều thiết bị trong số các thiết bị sau đây: bàn phím, chuột, màn hình, máy in, môđem, màn hình cảm ứng, giao diện nút bấm, và các thiết bị khác. (Các) ứng dụng VPS 230 đã sửa đổi, vốn được sử dụng để thực hiện phương pháp YPS đã sửa đổi, có thể được lưu trong thiết bị lưu trữ 212 và bộ nhớ 204, và được xử lý như các ứng dụng thông thường được xử lý. Nhiều hoặc ít thành phần hơn so với như được thể hiện trên Fig.2 có thể được sử dụng trong thiết bị tính toán 200. Theo một số phương án thực hiện, phần cứng VPS đã sửa đổi 220 được sử dụng. Mặc dù thiết bị tính toán 200 trên Fig.2 bao gồm các ứng dụng 230 và phần cứng 220 dành cho phương pháp VPS đã sửa đổi, nhưng phương pháp VPS đã sửa đổi này cũng có thể được thực hiện trên thiết bị tính toán bằng phần cứng, phần sụn, phần mềm, hoặc sự kết hợp bất kì giữa phần cứng, phần sụn và phần mềm. Ví dụ, theo một số phương án, các ứng dụng YPS đã sửa đổi 230 được lập trình trong bộ nhớ và được thực thi bằng bộ xử lý. Theo các phương án khác, phần cứng VPS đã sửa đổi 220 là phần cứng được lập trình, bao gồm các cổng logic được thiết kế chuyên để thực hiện phương pháp VPS đã sửa đổi.

Theo một số phương án, (các) ứng dụng VPS đã sửa đổi 230 bao gồm một số ứng dụng và/hoặc môđun. Theo một số phương án thực hiện, các môđun nêu trên cũng bao gồm một hoặc nhiều môđun con. Theo một số phương án thực hiện, có thể sử dụng ít hoặc nhiều môđun hơn.

Các ví dụ về các thiết bị tính toán phù hợp bao gồm máy tính cá nhân, máy tính xách tay, máy tính trạm, máy chủ, máy tính lớn, máy tính cầm tay, máy trợ

giúp cá nhân kỹ thuật số, điện thoại di động, đồ dùng thông minh, máy chơi trò chơi, máy ảnh số, máy quay phim số, điện thoại tích hợp máy ảnh, điện thoại thông minh, máy chơi nhạc cầm tay, máy tính bảng, thiết bị di động, máy phát video, đầu ghi/phát đĩa video (ví dụ, đầu ghi/phát đĩa DVD, đầu ghi/phát đĩa độ phân giải cao, đầu ghi/phát đĩa độ phân giải siêu cao), máy thu hình, hệ thống giải trí gia đình, đồ trang sức thông minh (ví dụ, đồng hồ thông minh) hoặc thiết bị tính toán phù hợp bất kì khác.

Fig.3 là hình vẽ thể hiện sơ đồ tổng quát của bộ mã hóa HEVC theo một số phương án. Bộ mã hóa 300 bao gồm thành phần điều khiển bộ mã hóa chung, thành phần biến đổi tỉ lệ và lượng tử hóa, thành phần thay đổi tỉ lệ và biến đổi ngược, thành phần ước lượng nội ảnh, thành phần phân tích điều khiển bộ lọc, thành phần dự đoán nội ảnh, thành phần bộ lọc tách khối và thành phần bộ lọc SAO (Sample Adaptive Offset - dịch thích ứng mẫu), thành phần bù chuyển động, thành phần ước lượng chuyển động, và thành phần định dạng mào đầu và CABAC (Context-Based Adaptive Binary Arithmetic Coding - mã hóa số học nhị phân thích ứng dựa trên ngữ cảnh). Tín hiệu video đầu vào được bộ mã hóa 300 nhận và được chia thành các đơn vị cây mã hóa (Coding Tree Unit - CTU). Các thành phần mã hóa HEVC xử lý dữ liệu video nhờ sử dụng VPS đã sửa đổi và tạo ra luồng bit được mã hóa.

Fig.4 là hình vẽ thể hiện sơ đồ tổng quát của bộ giải mã HEVC theo một số phương án. Bộ giải mã 400 bao gồm thành phần giải mã entropy, thành phần lượng tử hóa ngược, thành phần biến đổi ngược, thành phần khung hiện tại, thành phần nội dự đoán, thành phần khung trước đó, thành phần bù chuyển động, bộ lọc tách khối, thành phần SAO và bộ lọc vòng lặp thích ứng. Luồng bit vào (ví dụ, video đã mã hóa) được bộ giải mã 400 nhận, và luồng bit giải mã được tạo ra để hiển thị. Theo một số phương án, VPS đã sửa đổi được sử dụng trong lúc giải mã.

Để tận dụng phương pháp VPS đã sửa đổi, thì các thiết bị có thể truy cập các thông số trong VPS và phần mở rộng của nó đối với các chuỗi mã hóa video khả biến, các chuỗi mã hóa đa điểm quan sát, các chuỗi mã hóa video 3D và các chuỗi

video hỗn hợp. Phương pháp VPS đã sửa đổi này tự động được sử dụng khi thực hiện quá trình xử lý video hoặc những lúc khác. Phương pháp VPS đã sửa đổi này có thể được thực hiện một cách tự động mà không cần sự can thiệp của người dùng.

Trong quá trình hoạt động, thì các cấu trúc cú pháp VPS và `vps_extension()` được cập nhật với một số thao tác dọn sạch đối với phần HEVC Extensions trong các vùng mã hóa video khả biến, mã hóa đa điểm quan sát và mã hóa video 3D. Ngoài ra, bốn tùy chọn để bổ sung các cú pháp để hỗ trợ các chuỗi video hỗn hợp trong các lớp khác nhau đối với phần mở rộng VPS cũng được mô tả. VPS được tạo ra nhờ sử dụng cấu trúc cú pháp đã được sửa đổi này.

Một số phương án thực hiện sáng chế

1. Phương pháp được lập trình trong bộ nhớ bất biến của thiết bị, phương pháp này bao gồm các bước:
  - a. giải mã nội dung; và
  - b. truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp.
2. Phương pháp theo phương án 1, trong đó hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.
3. Phương pháp theo phương án 1, trong đó các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.
4. Phương pháp theo phương án 1, trong đó dữ liệu tập thông số video này bao gồm giá trị các bit đuôi phân tải hữu ích của chuỗi byte thô.
5. Phương pháp theo phương án 1, trong đó dữ liệu tập thông số video này được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte.
6. Phương pháp theo phương án 1, trong đó dữ liệu tập thông số video này

được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp.

7. Phương pháp theo phương án 6, trong đó dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ codec hỗn hợp nguồn.

8. Phương pháp theo phương án 7, trong đó dữ liệu tập thông số video này được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp nguồn.

9. Phương pháp theo phương án 1, trong đó dữ liệu tập thông số video này được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cờ hiển thị video hỗn hợp nguồn.

10. Phương pháp theo phương án 1, trong đó phương pháp này còn bao gồm bước mã hóa nội dung.

11. Phương pháp theo phương án 1, trong đó thiết bị nêu trên bao gồm máy tính cá nhân, máy tính xách tay, máy tính trạm, máy chủ, máy tính lớn, máy tính cầm tay, máy trợ giúp cá nhân kỹ thuật số, điện thoại di động, đồ dùng thông minh, máy chơi trò chơi, máy ảnh số, máy quay phim số, điện thoại tích hợp máy ảnh, điện thoại thông minh, máy chơi nhạc cầm tay, máy tính bảng, thiết bị di động, máy phát video, đầu ghi/phát đĩa video, đầu ghi/phát đĩa độ phân giải cao, đầu ghi/phát đĩa độ phân giải siêu cao, máy thu hình, hệ thống giải trí gia đình, hoặc đồng hồ thông minh.

12. Phương pháp được lập trình trong bộ nhớ bất biến của thiết bị, phương pháp này bao gồm các bước:

- a. cung cấp nội dung; và
- b. cho phép truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp.

13. Phương pháp theo phương án 12, trong đó hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.

14. Phương pháp theo phương án 12, trong đó các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.
15. Phương pháp theo phương án 12, trong đó dữ liệu tập thông số video này bao gồm giá trị các bit đuôi phần tải hữu ích của chuỗi byte thô.
16. Phương pháp theo phương án 12, trong đó dữ liệu tập thông số video này được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte.
17. Phương pháp theo phương án 12, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp.
18. Phương pháp theo phương án 17, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng thông số cú pháp cờ codec hỗn hợp nguồn.
19. Phương pháp theo phương án 18, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp nguồn.
20. Phương pháp theo phương án 12, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cơ hiển thị video hỗn hợp nguồn.
21. Phương pháp theo phương án 12, trong đó phương pháp này còn bao gồm bước mã hóa nội dung.
22. Phương pháp theo phương án 12, trong đó thiết bị nêu trên bao gồm máy tính cá nhân, máy tính xách tay, máy tính trạm, máy chủ, máy tính lớn, máy tính cầm tay, máy trợ giúp cá nhân kỹ thuật số, điện thoại di động, đồ dùng thông minh, máy chơi trò chơi, máy ảnh số, máy quay phim số, điện thoại tích hợp máy ảnh, điện thoại thông minh, máy chơi nhạc cầm tay, máy tính bảng, thiết bị di động, máy phát video, đầu ghi/phát đĩa video, đầu ghi/phát đĩa độ phân giải cao, đầu ghi/phát đĩa độ phân giải siêu cao, máy thu hình, hệ thống giải trí gia đình, hoặc đồng hồ thông minh.
23. Thiết bị bao gồm:
  - a. bộ nhớ bất biến để lưu giữ ứng dụng, trong đó ứng dụng này để:

i. giải mã nội dung; và

ii. truy cập thông tin liên quan đến nội dung này, trong đó thông tin này bao gồm dữ liệu tập thông số video, và trong đó dữ liệu tập thông số video này bao gồm thông tin báo hiệu hỗn hợp; và

b. bộ phận xử lý được ghép vào bộ nhớ, bộ phận xử lý này được cấu hình để xử lý ứng dụng nêu trên.

24. Thiết bị theo phương án 23, trong đó hàm tập thông số video được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.

25. Thiết bị theo phương án 23, trong đó các cú pháp đồng chỉnh byte được dùng khi xác định dữ liệu tập thông số video là theo điều kiện của cờ mở rộng tập thông số video.

26. Thiết bị theo phương án 23, trong đó dữ liệu tập thông số video này bao gồm giá trị các bit đuôi phần tải hữu ích của chuỗi byte thô.

27. Thiết bị theo phương án 23, trong đó dữ liệu tập thông số video được xác định mà không cần sử dụng các cú pháp đồng chỉnh byte.

28. Thiết bị theo phương án 23, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng hai cú pháp để hỗ trợ báo hiệu chuỗi hỗn hợp.

29. Thiết bị theo phương án 28, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng thông số cú pháp cờ codec hỗn hợp nguồn.

30. Thiết bị theo phương án 29, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng thông số cú pháp cờ hiển thị video hỗn hợp nguồn.

31. Thiết bị theo phương án 23, trong đó dữ liệu tập thông số video được xác định bằng cách sử dụng sự hỗ trợ của ứng dụng cụ thể và thông số cú pháp cờ hiển thị video hỗn hợp nguồn.

32. Thiết bị theo phương án 23, trong đó thiết bị này còn bao gồm thành phần để mã hóa nội dung.

Sáng chế đã được mô tả dựa vào các phương án thực hiện chi tiết cụ thể để làm rõ các nguyên lý, cấu trúc và sự hoạt động của phương pháp và thiết bị theo sáng chế. Phần mô tả chi tiết các phương án thực hiện này chỉ nhằm minh họa chứ không nhằm giới hạn phạm vi của các điểm yêu cầu bảo hộ kèm theo. Chuyên gia trong lĩnh vực kỹ thuật này dễ dàng nhận thấy rằng có thể có những phương án cải biến khác nhau đối với các phương án vốn được chọn nhằm mục đích minh họa nêu trên mà không vượt quá tinh thần và phạm vi của sáng chế, vốn được xác định theo các điểm yêu cầu bảo hộ này.

## YÊU CẦU BẢO HỘ

1. Phương pháp xử lý hình ảnh, trong đó phương pháp này bao gồm:

trong thiết bị xử lý hình ảnh bao gồm ít nhất một bộ xử lý:

tạo ra dòng bit bao gồm dữ liệu hình ảnh và cú pháp tập thống số video (video parameter set, viết tắt là VPS); và

truyền dòng bit được tạo ra,

trong đó cú pháp đồng chỉnh byte trong cú pháp VPS là theo điều kiện của cờ mở rộng VPS,

trong đó cú pháp đồng chỉnh byte liên quan đến việc đồng chỉnh byte,

trong đó giá trị cờ mở rộng VPS được xác định,

trong đó việc đồng chỉnh byte được thực hiện phụ thuộc vào trị số của cờ mở rộng VPS, và

trong đó chức năng mở rộng VPS được thực hiện dựa vào việc đồng chỉnh byte.

2. Phương pháp xử lý hình ảnh theo điểm 1,

trong đó cú pháp đồng chỉnh byte được bao gồm trong cú pháp VPS dựa vào việc xác định rằng cờ mở rộng VPS là đúng.

3. Phương pháp xử lý hình ảnh theo điểm 1,

trong đó dòng bit được tạo ra với đơn vị cây mã hóa là đơn vị.

4. Phương pháp xử lý hình ảnh theo điểm 1, trong đó phương pháp này còn bao gồm việc lọc dữ liệu hình ảnh tham chiếu dùng cho sự bù chuyển động làm bộ lọc tách khối.

5. Phương pháp xử lý hình ảnh theo điểm 4, trong đó phương pháp này còn bao gồm việc lọc dữ liệu hình ảnh tham chiếu được áp dụng cho bộ lọc tách khối là việc xử lý dịch thích ứng mẫu (sample adaptive offset, viết tắt là SAO).

6. Phương pháp xử lý hình ảnh theo điểm bất kỳ trong số các điểm từ 1 đến 5,

trong đó chức năng mở rộng VPS được sử dụng để xác định dữ liệu VPS.

7. Thiết bị xử lý hình ảnh, trong đó thiết bị này bao gồm:

ít nhất một bộ xử lý được tạo cấu hình để:

tạo ra dòng bit bao gồm dữ liệu hình ảnh và cú pháp tập thông số video (video parameter set, viết tắt là VPS); và

truyền dòng bit được tạo ra,

trong đó cú pháp đồng chỉnh byte trong cú pháp VPS là theo điều kiện của cờ mở rộng VPS,

trong đó cú pháp đồng chỉnh byte liên quan đến việc đồng chỉnh byte,

trong đó giá trị cờ mở rộng VPS được xác định,

trong đó việc đồng chỉnh byte được thực hiện phụ thuộc vào trị số của cờ mở rộng VPS, và

trong đó chức năng mở rộng VPS được thực hiện dựa vào việc đồng chỉnh byte.

8. Thiết bị xử lý hình ảnh theo điểm 7,

trong đó cú pháp đồng chỉnh byte được bao gồm trong cú pháp VPS dựa vào việc xác định rằng cờ mở rộng VPS là đúng.

9. Thiết bị xử lý hình ảnh theo điểm 7, trong đó việc tạo ra dòng bit với đơn vị cây mã hóa là đơn vị.

10. Thiết bị xử lý hình ảnh theo điểm 7, trong đó ít nhất một bộ xử lý còn được tạo cấu hình để lọc dữ liệu hình ảnh tham chiếu dùng cho sự bù chuyển động là bộ lọc tách khối.

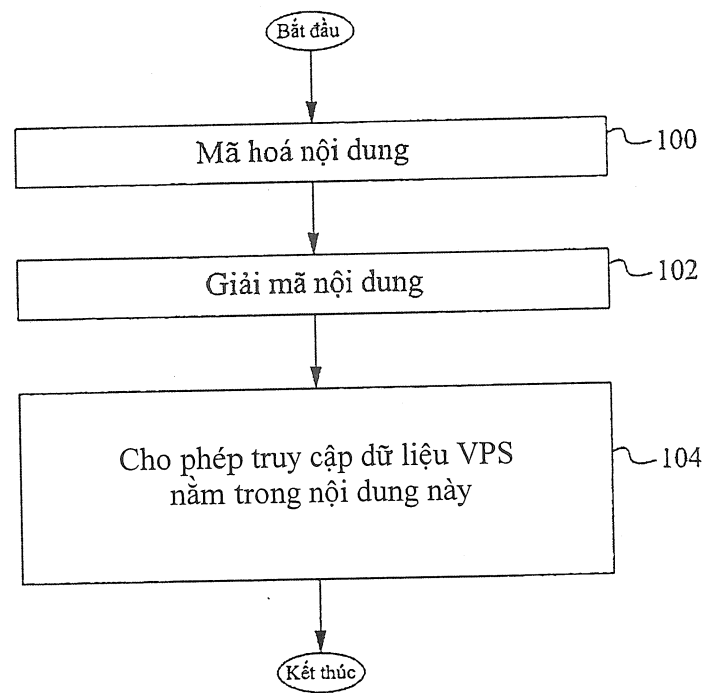
11. Thiết bị xử lý hình ảnh theo điểm 10, trong đó ít nhất một bộ xử lý còn được tạo cấu hình để lọc dữ liệu hình ảnh tham chiếu được áp dụng cho bộ lọc tách khối là việc xử lý dịch thích ứng mẫu (sample adaptive offset, viết tắt là SAO).

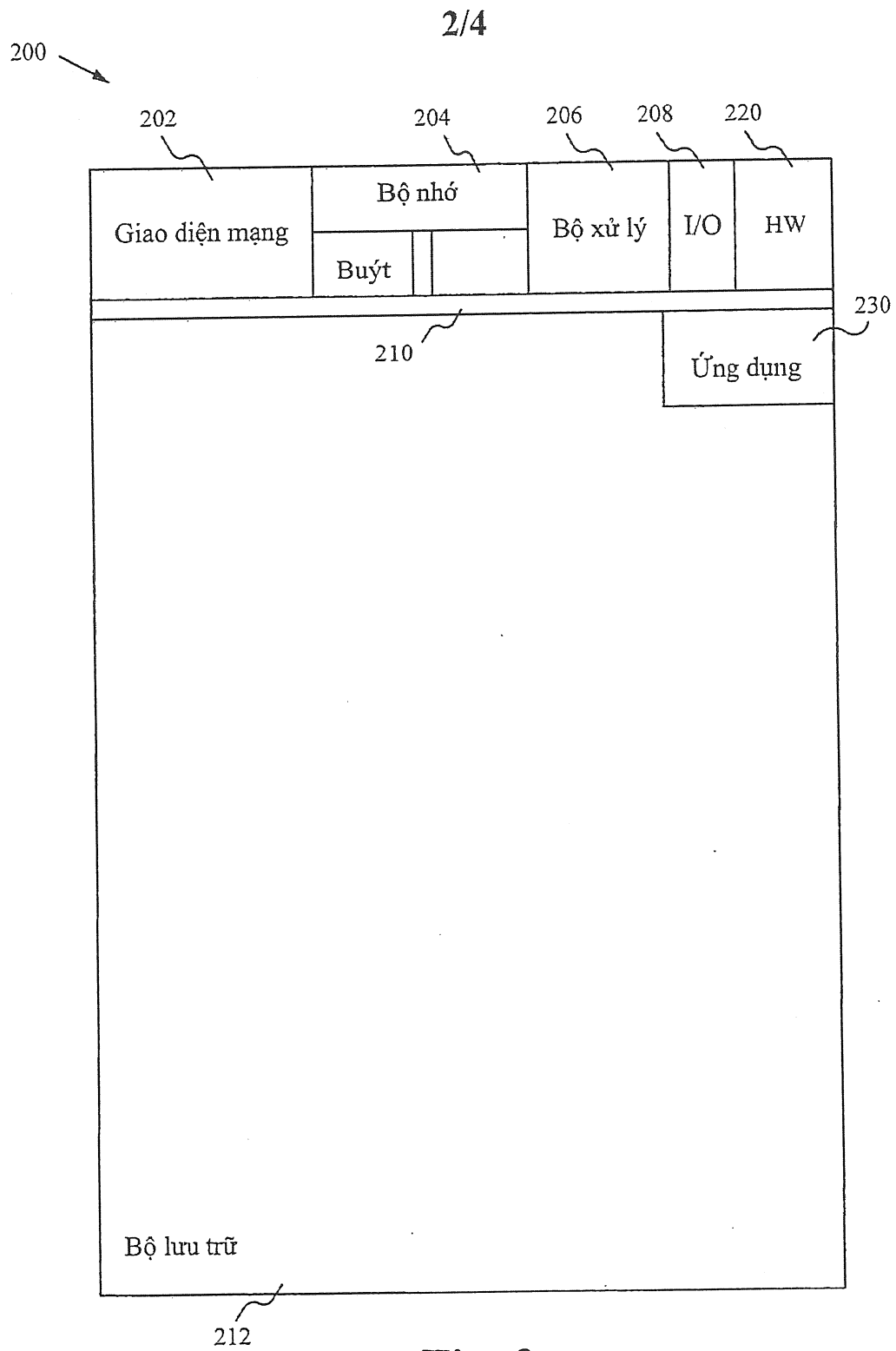
12. Thiết bị xử lý hình ảnh theo điểm bất kỳ trong số các điểm từ 7 đến 11,

trong đó chức năng mở rộng VPS được sử dụng để xác định dữ liệu VPS.

13. Bộ nhớ không tạm thời của thiết bị bao gồm mã được tạo cấu hình để khiến bộ xử lý thực hiện phương án theo điểm bất kỳ trong số các điểm từ 1 đến 6 khi mã được thực hiện bởi bộ xử lý.

1/4

**Fig. 1**

**Fig. 2**

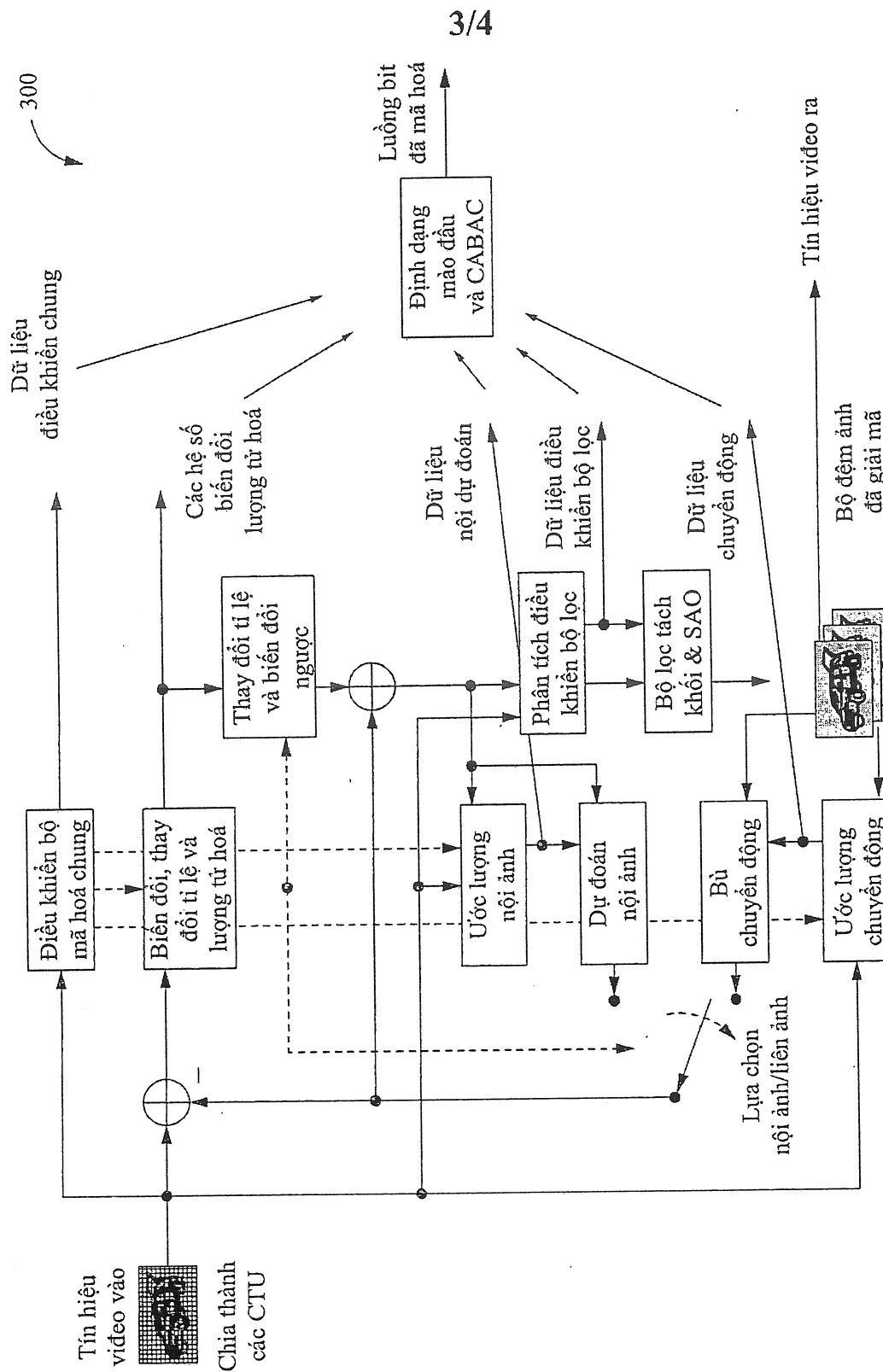


Fig. 3

4/4

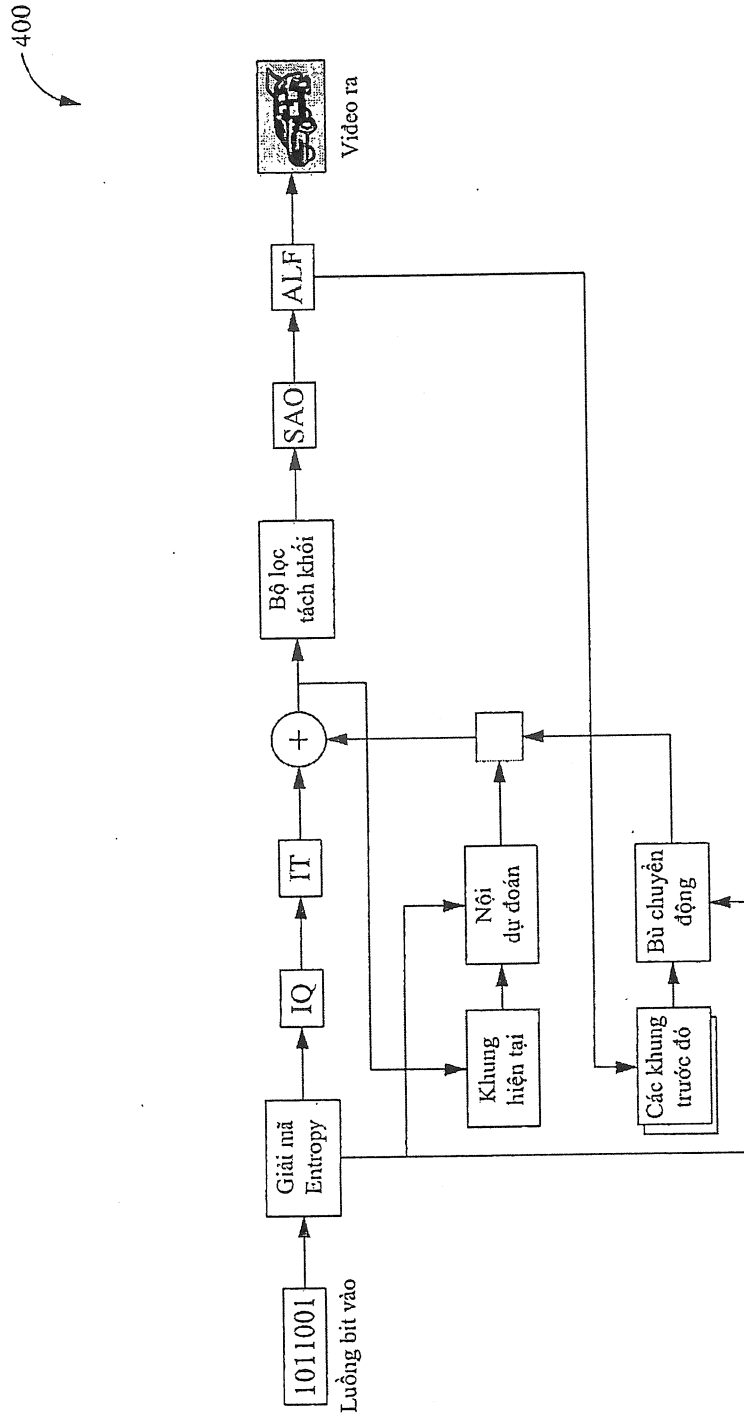


Fig. 4