



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ
(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11) 
CỤC SỞ HỮU TRÍ TUỆ
1-0053909
(51)^{2021.01} H04N 19/60; H04N 19/70; H04N 19/176; H04N 19/46 (13) B

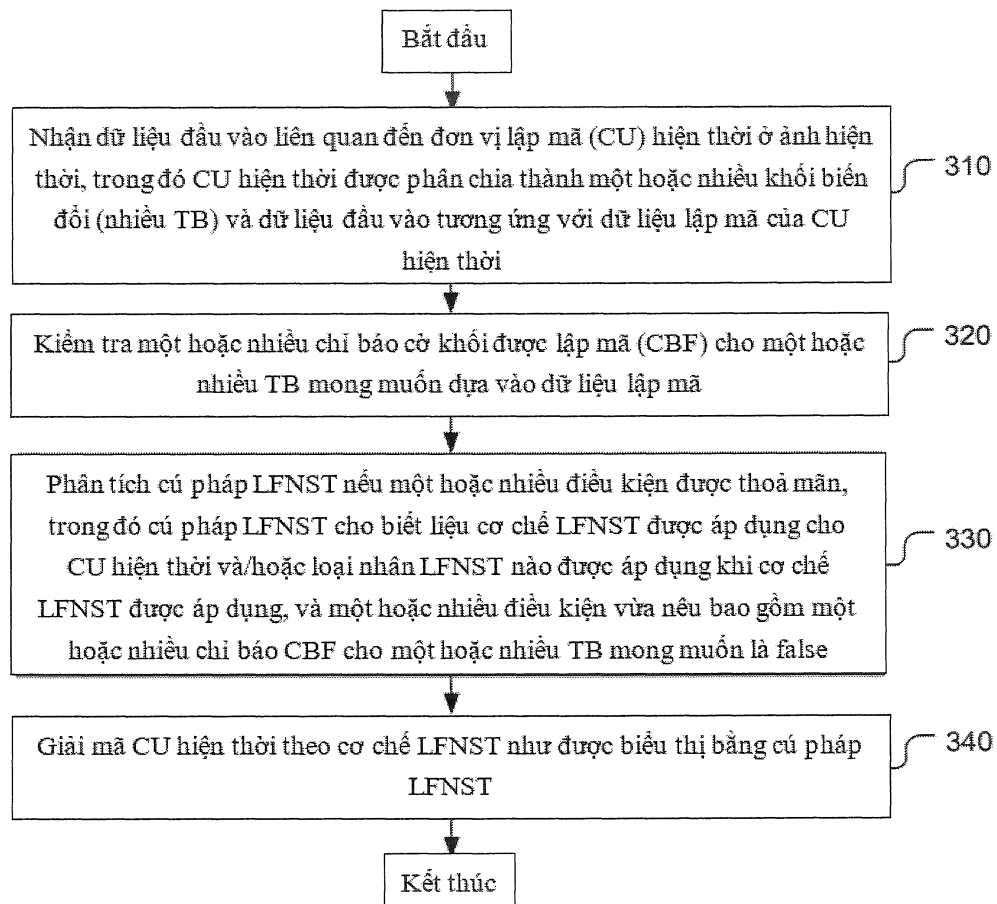
(21) 1-2022-05900 (22) 24/02/2021
(86) PCT/CN2021/077654 24/02/2021 (87) WO2021/169994 02/09/2021
(30) 62/981,066 25/02/2020 US; 62/988,423 12/03/2020 US
(45) 25/11/2025 452 (43) 25/11/2022 416A
(73) HFI INNOVATION INC. (CN)
3F.-7, No.5, Taiyuan 1st St., Zhubei City, Hsinchu County 302, Taiwan
(72) CHIANG, Man-Shu (TW); HSU, Chih-Wei (TW); CHUANG, Tzu-Der (TW);
CHEN, Ching-Yeh (TW).
(74) Công ty TNHH Trường Xuân (AGELESS CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ GIẢI MÃ CHUỖI VIDEO, PHƯƠNG PHÁP VÀ
THIẾT BỊ MÃ HÓA CHUỖI VIDEO

(21) 1-2022-05900

(57) Sáng chế đề cập đến phương pháp và thiết bị giải mã chuỗi video, phương pháp và thiết bị mã hóa chuỗi video. Dữ liệu đầu vào tương ứng với dữ liệu được biến đổi chính tại bộ mã hoá và dữ liệu đầu vào tương ứng với dữ liệu lập mã của CU hiện thời tại bộ giải mã. CU được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB). Cú pháp LFNST được xác định tại bộ mã hoá hoặc tại bộ giải mã nếu một hoặc nhiều điều kiện được thoả mãn. Cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng. Các điều kiện vừa nêu bao gồm các chỉ báo CBF cho các TB mong muốn là false. CU hiện thời được mã hoá hoặc được giải mã theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST.

Fig. 3



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến lập mã video. Cụ thể hơn là, sáng chế đề cập đến phương pháp và thiết bị giải mã chuỗi video, phương pháp và thiết bị mã hóa chuỗi video.

Tình trạng kỹ thuật của sáng chế

Chuẩn lập mã video hiệu suất cao HEVC (High-Efficiency Video Coding - HEVC) là một chuẩn lập video quốc tế mới được phát triển bởi nhóm hợp tác chung về mã hóa video JCT-VC (Joint Collaborative Team on Video Coding - JCT-VC). HEVC dựa trên cấu trúc mã hóa biến đổi tương tự với phép biến đổi DCT bù chuyển động dựa vào khối mạch lai. Đơn vị cơ bản để nén, về mặt thuật ngữ được gọi là đơn vị lập mã CU (Coding Unit - CU), là khối vuông có kích thước $2N \times 2N$, và mỗi CU có thể được chia đệ quy thành bốn CU nhỏ hơn cho đến khi kích cỡ nhỏ nhất định trước đạt được. Mỗi CU bao gồm một hoặc nhiều đơn vị dự đoán PU (Prediction Unit - PU).

Để đạt được hiệu suất lập mã tốt nhất cho kiến trúc lập mã lai trong HEVC, có hai dạng cơ chế dự đoán cho mỗi PU, đó là dự đoán nội ảnh và dự đoán liên ảnh. Đối với các cơ chế dự đoán nội ảnh, các điểm ảnh được tái tạo liên tiếp theo không gian có thể được sử dụng để tạo ra các dự đoán có hướng. Có tới 35 hướng trong HEVC. Đối với các cơ chế dự đoán liên ảnh, các khung hình tham chiếu được khôi phục theo thời gian có thể được sử dụng để tạo ra các dự đoán được bù chuyển động. Có ba loại cơ chế dự đoán liên ảnh, bao gồm cơ chế dự đoán vectơ tiên tiến AMVP (Advanced Motion Vector Prediction - AMVP), rẽ nhánh (Skip), trộn (Merge) và liên ảnh (Inter).

Phép biến đổi

Sau khi dự đoán, các phần dư dự đoán cho một CU được chia thành các đơn vị biến đổi TU (Transform Unit - TU) và được lập mã bằng phép biến đổi và lượng tử hóa. Giống như nhiều tiêu chuẩn lập mã tiền trước khác, HEVC sử dụng phép biến đổi cosin rời rạc dạng II (DCT-II) làm phép biến đổi nhân (biến đổi chính) bởi vì nó có đặc tính “nén năng lượng mạnh”. Để nâng cao khả năng biến đổi, phép biến đổi sin rời rạc DST

(Discrete Sine Transform - DST) cũng được đề xuất để được sử dụng thay phép DCT đối với những phương vị nội ảnh theo phương chéo. Đối với phần dư được dự đoán liên ảnh, DCT-II là phép biến đổi duy nhất được sử dụng ở phiên bản HEVC hiện tại. Tuy nhiên, DCT-II không phải là phép biến đổi tối ưu cho mọi trường hợp. Phép biến đổi sin rời rạc dạng VII (DST-VII) và phép biến đổi cosin rời rạc dạng VIII (DCT-VIII) được đề xuất để thay cho DCT-II trong một số trường hợp. Mô hình chọn phép biến đổi (MTS) cũng được sử dụng để lập mã phần dư cho cả khối được lập mã nội ảnh và liên ảnh. Nó sử dụng nhiều phép biến đổi được chọn từ họ các phép DCT/DST khác với các phép biến đổi hiện tại ở HEVC. Các ma trận biến đổi mới nhất được giới thiệu là DCT-VIII. Theo VVC, mô hình chọn phép biến đổi (MTS) cho biến đổi nhân được mô tả như sau.

Ngoài DCT-II đã được sử dụng trong HEVC, mô hình chọn phép biến đổi (MTS) được sử dụng để lập mã phần dư cho các khối được lập mã nội ảnh và/hoặc liên ảnh. Nó sử dụng nhiều phép biến đổi được chọn từ DCT8 (DCT-VIII)/DST7 (DST-VII). Các ma trận biến đổi mới nhất được giới thiệu là DST-VII và DCT-VIII. Bảng dưới đây thể hiện các hàm cơ bản của DST/DCT được chọn.

Bảng 1. Các hàm biến đổi cơ bản của DCT-II/ VIII và DSTVII cho đầu vào N điểm

Dạng biến đổi	Hàm cơ bản $T_i(j)$, $i, j = 0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ trong đó, $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

Để giữ tính trực giao của ma trận biến đổi, các ma trận biến đổi được lượng tử hóa chính xác hơn các ma trận biến đổi trong HEVC. Để giữ các giá trị trung gian của các hệ số biến đổi nằm trong phạm vi 16 bit, sau khi biến đổi ngang và sau khi biến đổi đứng, thì tất cả các hệ số đều phải có 10 bit.

Để điều khiển mô hình MTS, các cờ kích hoạt riêng được định ra tại phân mức

SPS cho cơ chế nội ảnh và liên ảnh, một cách tương ứng. Khi MTS được sử dụng tại SPS, chỉ số phân mức CU được báo hiệu để biểu thị cơ chế biến đổi có các dạng biến đổi theo phương ngang và đứng cho CU hiện thời. Ở đây, MTS chỉ được áp dụng cho mẫu sáng (luma). Chỉ số phân mức CU MTS (tức là, `mts_idx`) có thể được báo hiệu khi cả chiều rộng và chiều cao nhỏ hơn hoặc bằng 32 và cờ khối được lập mã CBF (Coded Block Flag – CBF) bằng 1.

Nếu chỉ số CU MTS bằng 0, thì DCT2 được áp dụng theo cả hai phương. Tuy nhiên, nếu chỉ số CU MTS lớn hơn 0, thì các dạng biến đổi theo phương ngang và đứng được xác định theo bảng 2.

Bảng 2. Bảng ánh xạ biến đổi và báo hiệu

mts_idx	Nội ảnh/liên ảnh	
	Ngang	Đứng
0	DCT2	
1	DST7	DST7
2	DCT8	DST7
3	DST7	DCT8
4	DCT8	DCT8

Để giảm độ phức tạp của DST-7 và DCT-8 kích cỡ lớn, các hệ số biến đổi tần số cao được đặt bằng không đối với các khối DST-7 và DCT-8 có kích cỡ (chiều rộng hoặc chiều cao, hoặc cả chiều rộng và chiều cao) bằng 32. Chỉ các hệ số nằm trong vùng tần số thấp hơn 16x16 được giữ lại.

Biến đổi không phân tách tần số thấp LFNST (LFNST - Low-Frequency Non-Separable Transform)

Theo VVC, LFNST thuận (biến đổi không phân tách tần số thấp) 120, còn được gọi là biến đổi phụ suy giảm, được áp dụng giữa phép biến đổi chính thuận 110 và phép lượng tử hóa 130 (tại bộ mã hoá) và LFNST ngược 150 được áp dụng giữa phép giải lượng tử 140 và phép biến đổi chính ngược 160 (tại bộ giải mã) như được thể hiện trên Fig. 1. Theo LFNST, phép biến đổi không phân tách 4x4 hoặc phép biến đổi không phân tách 8x8 được áp dụng theo kích cỡ khối. Ví dụ, LFNST 4x4 được áp dụng cho các khối nhỏ (tức là, $\min(\text{chiều rộng}, \text{chiều cao}) < 8$) và LFNST 8x8 được áp dụng cho các khối lớn (tức là, $\min(\text{chiều rộng}, \text{chiều cao}) > 4$). Trên Fig. 1, vùng điền dấu chấm 122 tương ứng với 16 hệ số đầu vào cho LFNST thuận 4x4 hoặc 48 hệ số đầu vào cho LFNST

thuận 8x8. Vùng điền dấu chấm 152 tương ứng với 8 hoặc 16 hệ số đầu vào cho LFNST ngược 4x4 hoặc 8 hoặc 16 hệ số đầu vào cho LFNST ngược 8x8. Trong trường hợp này, đầu vào cho phép biến đổi chính thuận là các tín hiệu dư dự đoán và đầu ra từ phép biến đổi chính ngược là tín hiệu dư được khôi phục.

Việc áp dụng phép biến đổi không phân tách, mà được sử dụng trong LFNST, được mô tả ở ví dụ dưới đây. Để áp dụng LFNST 4x4, khối đầu vào X 4x4,

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

trước tiên được biểu diễn dưới dạng vectơ $\vec{X}$:

$$\vec{X} = [X_{00} X_{01} X_{02} X_{03} X_{10} X_{11} X_{12} X_{13} X_{20} X_{21} X_{22} X_{23} X_{30} X_{31} X_{32} X_{33}]^T$$

Biến đổi không phân tách được tính là $\vec{F} = T \cdot \vec{X}$, trong đó $\vec{F}$ biểu thị vectơ hệ số biến đổi, và T là ma trận biến đổi 16x16. Hệ số vectơ $\vec{F}$ 16x1 được tái tạo lại thành khối 4x4 bằng cách sử dụng lệnh quét (tức là, ngang, đứng hoặc chéo) đối với khối đó. Các hệ số có chỉ số nhỏ hơn sẽ được thay bằng chỉ số quét nhỏ hơn trong khối hệ số 4x4.

Biến đổi không phân tách suy giảm

LFNST (biến đổi không phân tách tần số thấp) dựa theo phương pháp nhân ma trận trực tiếp để áp dụng phép biến đổi không phân tách sao cho nó được thực hiện trong một lần duy nhất mà không lặp lại nhiều lần. Tuy nhiên, kích thước ma trận biến đổi không phân tách cần được giảm để giảm thiểu mức độ phức tạp tính toán và không gian bộ nhớ để lưu các hệ số biến đổi. Vì vậy, phương pháp biến đổi không phân tách suy giảm RST (Reduced non-Separate Transform - RST) được sử dụng trong LFNST. Ý tưởng chính của phép biến đổi không phân tách suy giảm là ánh xạ vectơ N chiều thành vectơ R chiều ở không gian khác, trong đó N/R ($R < N$) là hệ số suy giảm và N thường bằng 64 đối với các biến đổi phụ không phân tách NSST 8x8 (Non-Separable Secondary Transform - NSST). Vì vậy, thay cho ma trận $N \times N$, ma trận RST trở thành ma trận $R \times N$ như sau:

$$T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \dots & t_{RN} \end{bmatrix}$$

trong đó R hàng của ma trận biến đổi là R chiều của không gian N chiều. Ma trận biến đổi ngược cho RT là chuyển vị của phép biến đổi thuận. Đối với LFNST 8×8 , hệ số suy giảm bằng 4 được áp dụng. Trong trường hợp này, ma trận trực tiếp 64×64 , mà thường được sử dụng cho ma trận biến đổi không phân tách 8×8 , được giảm xuống thành ma trận trực tiếp 16×48 . Do đó, ma trận RST ngược 48×16 được sử dụng tại bộ giải mã để tạo ra các hệ số biến đổi nhân (chính) ở các vùng 8×8 trên cùng-bên trái. Khi các ma trận 16×48 được áp dụng thay cho 16×64 với cùng cấu hình thiết lập biến đổi, mỗi chúng lấy 48 dữ liệu đầu vào từ ba khối 4×4 ở khối 8×8 trên cùng-bên trái mà không chứa khối 4×4 bên phải-dưới cùng.

Nhờ kích thước được giảm, nên việc sử dụng bộ nhớ để lưu toàn bộ các ma trận LFNST được giảm từ 10KB xuống 8KB với độ sụt hiệu suất chấp nhận được. Để giảm độ phức tạp tính toán, LFNST được ràng buộc sao cho được áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên đều không có nghĩa. Do đó, toàn bộ các hệ số mà chỉ áp dụng biến đổi chính phải được cho bằng 0 khi LFNST được áp dụng. Điều này cho phép thiết đặt điều kiện đối với báo hiệu chỉ số LFNST ở vị trí có nghĩa cuối. Do vậy, nó giúp tránh phải quét hệ số phụ thêm trong cấu trúc LFNST hiện tại, vốn là việc cần thiết để kiểm tra các hệ số có nghĩa chỉ ở các vị trí cụ thể. Phép xử lý trường hợp xấu nhất của LFNST, khi thực hiện nhiều phép nhân trên điểm ảnh, sẽ ràng buộc các biến đổi không phân tách cho các khối 4×4 và 8×8 thành các biến đổi 8×16 và 8×48 , một cách tương ứng. Đối với những trường hợp này, vị trí quét có nghĩa cuối cùng phải nhỏ hơn 8 khi LFNST được áp dụng cho các kích cỡ khác nhỏ hơn 16. Đối với các khối có hình dạng $4 \times N$ và $N \times 4$ trong đó $N \geq 4$, việc ràng buộc đã nêu có nghĩa là LFNST được áp dụng chỉ một lần và được áp dụng chỉ cho vùng 4×4 trên cùng-bên trái. Đối với các khối có hình dạng $8 \times N$ và $N \times 8$ trong đó $N \geq 8$, việc ràng buộc đã nêu có nghĩa là LFNST được áp dụng chỉ một lần và được áp dụng chỉ cho vùng 8×8 trên cùng-bên trái. Vì toàn bộ các hệ số mà chỉ áp dụng biến đổi chính phải được cho bằng 0 khi LFNST được áp dụng, nên số lượng các thuật toán dùng cho các biến đổi chính được giảm ở những trường hợp vừa nêu. Xét từ bộ mã hóa, việc lượng tử hóa các hệ số được đơn giản hóa đáng kể khi các biến đổi LFNST được kiểm tra. Phép lượng tử tối ưu độ méo phải được thực hiện nhiều nhất cho 8 hoặc 16 hệ số đầu tiên theo thứ tự quét, và các hệ số còn lại được thiết đặt bằng 0.

Chọn biến đổi cho LFNST

Có tổng cộng 4 thiết lập biến đổi và 2 ma trận (nhân) biến đổi không phân tách cho một thiết lập biến đổi được sử dụng ở LFNST. Ánh xạ từ cơ chế dự đoán nội ảnh thành thiết lập biến đổi được định trước như được thể hiện ở bảng dưới đây. Nếu một trong số ba cơ chế mô hình tuyến tính thành phần chéo CCLM (Cross-Component Linear Model - CCLM) (tức là, INTRA_LT_CCLM, INTRA_T_CCLM hoặc INTRA_L_CCLM khi được biểu diễn bằng $81 \leq \text{predModeIntra} \leq 83$) được sử dụng cho khối hiện thời, thì thiết lập biến đổi 0 hoặc cơ chế dự đoán nội ảnh được chọn cho khối màu (chroma) hiện thời. Đối với mỗi thiết lập biến đổi, ứng viên biến đổi phụ không phân tách (hay còn được gọi là ma trận biến đổi không phân tách) được xác định tiếp bằng chỉ số LFNST được báo hiệu hiện. Chỉ số thiết lập biến đổi được báo hiệu trong dòng bit một lần cho mỗi CU nội ảnh sau các hệ số biến đổi.

Bảng 3. Bảng chọn biến đổi

IntraPredMode	Chỉ số thiết lập biến đổi
$\text{IntraPredMode} < 0$	1
$0 \leq \text{IntraPredMode} \leq 1$	0
$2 \leq \text{IntraPredMode} \leq 12$	1
$13 \leq \text{IntraPredMode} \leq 23$	2
$24 \leq \text{IntraPredMode} \leq 44$	3
$45 \leq \text{IntraPredMode} \leq 55$	2
$56 \leq \text{IntraPredMode} \leq 80$	1
$81 \leq \text{IntraPredMode} \leq 83$	0

Báo hiệu chỉ số LFNST và tương tác với các công cụ khác

Vì LFNST được ràng buộc sao cho được áp dụng chỉ khi toàn bộ các hệ số nằm ngoài nhóm con hệ số đầu tiên đều không có nghĩa, nên lập mã chỉ số LFNST (phân mức CU) phụ thuộc vào vị trí của hệ số có nghĩa cuối cùng. Ngoài ra, chỉ số LFNST được lập mã thuộc tính. Tuy nhiên, chỉ số LFNST không phụ thuộc vào cơ chế dự đoán nội ảnh và ít nhất một vùng trống được lập mã thuộc tính. Ngoài ra, LFNST được áp dụng cho CU nội ảnh ở cả mảng (slice) nội ảnh và liên ảnh cho cả luma và/hoặc chroma. Nếu cây lưỡng được sử dụng, thì các tham số LFNST cho luma và chroma được báo hiệu riêng. Đối với slice liên ảnh (tức là, cây lưỡng không được sử dụng), một chỉ số LFNST duy nhất được báo hiệu và được sử dụng cho cả luma và/hoặc chroma.

Giả sử rằng CU lớn hơn 64x64 được phân chia ẩn (phân lát TU) do giới hạn kích

cỡ biến đổi lớn nhất hiện có (tức là, kích cỡ 64x64 hoặc kích cỡ được thiết lập theo cấu hình), phép tìm kiếm chỉ số LFNST có thể làm tăng bộ đệm dữ liệu lên bốn lần đối với số lượng các trạng thái truyền giải mã nhất định. Vì vậy, kích cỡ lớn nhất mà LFNST được phép được giới hạn là 64x64 hoặc kích cỡ biến đổi lớn nhất. Chú ý rằng MTS được sử dụng chỉ khi LFNST không được dùng.

Như được đề xuất trong JVET-P0058 (T. Tsukuba, và các cộng sự, “CE8-2.1: Rẽ nhánh biến đổi cho chroma bằng cách giới hạn số lượng lớn nhất vùng trống được lập mã thuộc tính khi lập mã phần dư TS,” ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 16: Geneva, CH, Ngày 1 đến 11 tháng 10 năm 2019, Tài liệu: JVET-P0058), tài liệu này giới thiệu phép rẽ nhánh biến đổi TS (Transform Skip - TS) dùng cho chroma và áp dụng lập mã phần dư TS cho khối chroma được rẽ nhánh biến đổi. Ví dụ, TS được sử dụng cho chroma theo mọi định dạng mẫu chroma. Ngoài ra, vì phép điều chế mã xung vi sai dựa vào khối BDPCM (Block-based Delta Pulse Code Modulation - BDPCM) sử dụng TS, nên BDPCM có thể được dùng chỉ khi điều kiện sử dụng TS được thoả mãn. Điều kiện sử dụng TS chứa điều kiện ràng buộc kích cỡ mà nghĩa là điều kiện khi chiều rộng khối nhỏ hơn hoặc bằng kích cỡ rẽ nhánh biến đổi lớn nhất (MaxTsSize) và chiều cao khối nhỏ hơn hoặc bằng MaxTsSize. Nếu điều kiện vừa nêu được thoả mãn, TS có thể được dùng. MaxTsSize là số nguyên hoặc biến cố định bằng $1 \ll (\log_2_transform_skip_max_size_minus2 + 2)$, trong đó $\log_2_transform_skip_max_size_minus2$ xác định kích cỡ khối lớn nhất được sử dụng cho phép rẽ nhánh biến đổi. $\log_2_transform_skip_max_size_minus2$ sẽ nằm trong khoảng từ 0 đến 3 và được xem là bằng 0 khi không xuất hiện.

Theo VVC, điều kiện ràng buộc kích cỡ đối với TS cho luma là nếu $tbWidth \leq MaxTsSize \ \&\& \ tbHeight \leq MaxTsSize$, thì TS có thể được sử dụng.

Theo VVC, điều kiện ràng buộc kích cỡ đối với TS cho chroma là nếu $wC \leq MaxTsSize \ \&\& \ hC \leq MaxTsSize$, thì TS có thể được sử dụng.

Ở những điều kiện ràng buộc được nêu ở trên, $wC = tbWidth / SubWidthC$ và $hC = tbHeight / SubHeightC$. $TbWidth$ là chiều rộng khối cho luma và $TbHeight$ là chiều cao khối cho luma. Các biến $SubWidthC$ và $SubHeightC$ được xác định ở bảng dưới đây tùy theo cấu trúc mẫu định dạng chroma, mà được xác định bằng $chroma_format_idc$ và $separate_colour_plane_flag$. Các trị số khác của $chroma_format_idc$, $SubWidthC$ và $SubHeightC$ có thể được xác định sau.

Bảng 4. Mô tả các biến SubWidthC và SubHeightC

chroma_format_ idc	separate_colour_plane_ flag	Định dạng chroma	SubWidth C	SubHeight C
0	0	Đơn sắc	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

Điều kiện báo hiệu chi tiết cho cơ chế rẽ nhánh biến đổi cho từng thành phần được thể hiện ở bảng dưới đây.

Bảng 5. Điều kiện báo hiệu cho cơ chế rẽ nhánh biến đổi cho từng thành phần

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Biến mô tả
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions - 1) {	
xC = CbPosX[chType][x0][y0]	
yC = CbPosY[chType][x0][y0]	
wC = CbWidth[chType][x0][y0] / SubWidthC	
hC = CbHeight[chType][x0][y0] / SubHeightC	
} else {	
xC = x0	
yC = y0	
wC = tbWidth / SubWidthC	
hC = tbHeight / SubHeightC	
}	
chromaAvailable = treeType != DUAL_TREE_LUMA && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1))	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag))) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex == NumIntraSubPartitions - 1))) {	
tu_cbf_cbf[xC][yC]	ae(v)

tu_cbf_cr[xC][yC]	ae(v)
}	
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag))) && (CuPredMode[chType][x0][y0] == MODE_INTRA (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC])) CbWidth[chType][x0][y0] > MaxTbSizeY CbHeight[chType][x0][y0] > MaxTbSizeY)) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfLuma)))	
tu_cbf_luma[x0][y0]	ae(v)
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT)	
InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[x0][y0]	
}	
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y0] > 64 tu_cbf_luma[x0][y0] (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC]))) && treeType != DUAL_TREE_CHROMA) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
}	
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y0] > 64 (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC]))) && treeType != DUAL_TREE_LUMA) {	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	

cu_chroma_qp_offset_idx	ae(v)
}	
}	
if(sps_joint_cbr_enabled_flag && ((CuPredMode[chType][x0][y0] == MODE_INTRA && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC])) (tu_cbf_cb[xC][yC] && tu_cbf_cr[xC][yC])) && chromaAvailable)	
tu_joint_cbr_residual_flag[xC][yC]	ae(v)
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && !cu_sbt_flag)	
transform_skip_flag[x0][y0][0]	ae(v)
if(!transform_skip_flag[x0][y0][0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
if(tu_cbf_cb[xC][yC] && treeType != DUAL_TREE_LUMA) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][1]	ae(v)
if(!transform_skip_flag[xC][yC][1])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 1)	
else	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 1)	
}	
if(tu_cbf_cr[xC][yC] && treeType != DUAL_TREE_LUMA && !(tu_cbf_cb[xC][yC] && tu_joint_cbr_residual_flag[xC][yC])) {	
if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][2]	ae(v)
if(!transform_skip_flag[xC][yC][2])	
residual_coding(xC, yC, Log2(wC), Log2(hC), 2)	
else	

<code>residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 2)</code>	
<code>}</code>	
<code>}</code>	

Ở bảng cú pháp ở trên, `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` xác định liệu phép biến đổi được áp dụng cho khối biến đổi liên quan hay không. Các chỉ số mảng `x0`, `y0` xác định vị trí (`x0`, `y0`) của mẫu luma trên cùng-bên trái của khối biến đổi được xét so với mẫu luma trên cùng-bên trái của ảnh. Chỉ số mảng `cIdx` xác định chỉ số cho thành phần màu thành phần màu; chỉ số đó bằng 0 cho Y, 1 cho Cb, và 2 cho Cr. `Transform_skip_flag[ x0 ][ y0 ][ cIdx ]` bằng 1 xác định rằng không có phép biến đổi nào được áp dụng cho khối biến đổi liên quan. `Transform_skip_flag[ x0 ][ y0 ][ cIdx ]` bằng 0 xác định rằng quyết định trong đó liệu phép biến đổi nào được áp dụng cho khối biến đổi liên quan hay không phụ thuộc vào các thành phần cú pháp khác.

Khi `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` không xuất hiện, nó được suy diễn như sau:

- Nếu `BdpcmFlag[ x0 ][ y0 ][ cIdx ]` bằng 1, `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` được xem là bằng 1.
- Nếu không thì (`BdpcmFlag[ x0 ][ y0 ][ cIdx ]` bằng 0), `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` được xem là bằng 0.

Theo trên, `BdpcmFlag[ x0 ][ y0 ][ cIdx ]` là biến tương ứng với cờ BDPCM nội ảnh cho thành phần luma (tức là, `cIdx = 0`) hoặc cho thành phần chroma (tức là, `cIdx = 1` hoặc `2`).

Phép điều chế mã xung vi sai dựa vào khối (BDPCM)

Phương pháp BDPCM cũ được đề xuất trong JVET-M0057 (“CE8: BDPCM có biến dự đoán ngang/đứng và các vùng có thể giải mã độc lập (kiểm nghiệm 8.3.1b)”, JVET-M0057, Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Marrakech, MA, Tháng 1 năm 2019) sử dụng các mẫu được khôi phục để dự đoán lần lượt các hàng hoặc các cột. Chiều BDPCM được báo hiệu sẽ cho biết liệu dự đoán phương đứng hay phương ngang được sử dụng. Các điểm ảnh tham chiếu được sử dụng là các mẫu chưa được lọc. Sai số dự đoán được lượng tử hoá theo miền không gian. Các điểm ảnh được khôi phục bằng cách cộng sai số dự đoán được giải lượng tử vào phép dự đoán.

Theo JVET-N0413 (T. Tsukuba, và các cộng sự, “CE8-2.1: Rẽ nhánh biến đổi cho chroma bằng cách giới hạn số lượng lớn nhất vùng trông được lập mã thuộc tính khi lập mã phần dư TS,” Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 16: Geneva, CH, Ngày 1 đến 11 tháng 10 năm 2019, Tài liệu: JVET-P0058), là mô hình thay thế cho BDPCM cũ, phép điều chế mã xung miền dư được lượng tử, được gọi là RDPCM, được đề xuất. Các chiều dự đoán và báo hiệu được sử dụng là tương tự như mô hình BDPCM cũ như được mô tả trong JVET-M0057 (F. Henry, và các cộng sự, “CE8: BDPCM có biến dự đoán ngang/đứng và các vùng có thể giải mã độc lập (kiểm nghiệm 8.3.1b)”, Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Marrakech, MA, Tháng 1 năm 2019, Tài liệu: JVET-M0057). Dự đoán nội ảnh cho BDPCM được thực hiện ở toàn bộ khối bằng cách sao chép mẫu theo chiều dự đoán (dự đoán phương ngang hoặc phương đứng) tương tự như dự đoán nội ảnh thông thường. Phần dư được lượng tử hoá và giá trị delta giữa phần dư được lượng tử hoá vừa nêu và trị số lượng tử hoá biến dự đoán (ngang hoặc đứng) của phần dư đó được lập mã. Điều này có thể được mô tả như sau.

Đối với khối M (hàng) $\times N$ (cột), cho $r_{i,j}$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$, là phần dư dự đoán sau khi thực hiện dự đoán nội ảnh theo phương ngang (tức là, sao chép trị số điểm ảnh liền kề bên trái ở khắp khối được dự đoán theo từng hàng) hoặc theo phương đứng (tức là, sao chép hàng liền kề phía trên thành từng hàng trong khối được dự đoán) bằng cách sử dụng các mẫu chưa được lọc từ các mẫu biên giới phía trên hoặc bên trái. Cho $Q(r_{i,j})$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$, là giá trị lượng tử hoá của phần dư $r_{i,j}$, trong đó phần dư là sai khác giữa các giá trị khối gốc ban đầu và giá trị khối được dự đoán. BDPCM sau đó được áp dụng cho các mẫu được lượng tử hoá phần dư, tạo ra mảng $\tilde{R}$ $M \times N$ được sửa đổi có các phần tử $\tilde{r}_{i,j}$. Khi BDPCM đứng được báo hiệu, các mẫu được lượng tử hoá phần dư được tạo ra bằng:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \leq j \leq (N - 1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \end{cases}$$

Đối với dự đoán phương ngang, các quy tắc tương tự được áp dụng, và các mẫu được lượng tử hoá phần dư được tạo ra bằng:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), j = 0 \\ Q(r_{i,j}) - Q(r_{i,j-1}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

Các mẫu được lượng tử hoá phần dư $\tilde{r}_{i,j}$ được lập mã và được truyền đến bộ giải mã.

Tại bộ giải mã, các phép tính nêu trên được lấy nghịch đảo để tạo ra $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$. Đối với trường hợp dự đoán phương đứng,

$$Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1).$$

Đối với trường hợp dự đoán phương ngang,

$$Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1).$$

Các phần dư được lượng tử hoá ngược, $Q^{-1}(Q(r_{i,j}))$, được cộng vào các trị số dự đoán khối nội ảnh để tạo ra các giá trị mẫu được khôi phục.

Cú pháp của RDPCM được báo hiệu tại phân mức CU/CB, khi CU/CB là CU/CB nội ảnh luma và chiều rộng và/hoặc chiều cao CB nhỏ hơn hoặc bằng giá trị ngưỡng định trước (ví dụ, 16, 32, 64, 128, 256, 512, hoặc 1024), một cờ (ví dụ, `bdpcm_flag`) được báo hiệu để biểu thị liệu có sử dụng RDPCM hay không. Nếu `bdpcm_flag` là đúng (`true`), một cờ phụ (`bdpcm_dir_flag`) được báo hiệu để cho biết chiều dự đoán được sử dụng ở RDPCM. Ví dụ, nếu `bdpcm_dir_flag` bằng 0, phương ngang được sử dụng; nếu không thì nếu `bdpcm_dir_flag` bằng 1, phương đứng được sử dụng.

BDPCM có thể được áp dụng cho luma và chroma. Bảng cú pháp cho BDPCM được thể hiện trên ở bảng dưới đây. Chi tiết hơn có thể được tìm thấy trong JVET-N0413 (M. Karczewicz, và các cộng sự, “CE8: Phép BDPCM phần dư được lượng tử hoá,” Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 14: Geneva, CH, Ngày 19 đến 27 tháng 3 năm 2019, Tài liệu: JVET-N0413).

đư chroma chung ($\text{resJointC}[x][y]$ ở **Error! Reference source not found.**) được báo hiệu, và khối dư cho Cb (resCb) và khối dư cho Cr (resCr) được suy dẫn bằng cách xét đến thông tin chẳng hạn tu_cbf_cb , tu_cbf_cr , và CSign , là giá trị dấu được xác định ở tiêu đề mảng.

Tại bộ mã hoá, các thành phần chroma chung được suy dẫn như được diễn giải dưới đây. Phụ thuộc vào cơ chế (được liệt kê ở các bảng nêu trên), $\text{resJointC}\{1,2\}$ được tạo ra bởi bộ mã hoá như sau:

– Nếu cơ chế bằng 2 (phần dư riêng có phép khôi phục $\text{Cb} = \text{C}$, $\text{Cr} = \text{CSign} * \text{C}$), phần dư chung được xác định theo

$$\text{resJointC}[x][y] = (\text{resCb}[x][y] + \text{CSign} * \text{resCr}[x][y]) / 2,$$

– Nếu không thì, nếu cơ chế bằng 1 (phần dư riêng có phép khôi phục $\text{Cb} = \text{C}$, $\text{Cr} = (\text{CSign} * \text{C}) / 2$), phần dư chung được xác định theo

$$\text{resJointC}[x][y] = (4 * \text{resCb}[x][y] + 2 * \text{CSign} * \text{resCr}[x][y]) / 5,$$

– Nếu không thì (cơ chế bằng 3, tức là, phần dư riêng có phép khôi phục $\text{Cr} = \text{C}$, $\text{Cb} = (\text{CSign} * \text{C}) / 2$), phần dư chung được xác định theo

$$\text{resJointC}[x][y] = (4 * \text{resCr}[x][y] + 2 * \text{CSign} * \text{resCb}[x][y]) / 5.$$

Trong quá trình khôi phục các phần dư chroma, trị số CSign là giá trị dấu (+1 hoặc -1), mà được xác định ở tiêu đề mảng, $\text{resJointC}[]$ là phần dư được truyền đến.

Bảng 7. Bảng cú pháp cho BDPCM

tu_cbf_cb	tu_cbf_cr	Phép khôi phục các phần dư Cb và Cr	Cơ chế
1	0	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$	1
1	1	$\text{resCb}[x][y] = \text{resJointC}[x][y]$ $\text{resCr}[x][y] = \text{CSign} * \text{resJointC}[x][y]$	2
0	1	$\text{resCb}[x][y] = (\text{CSign} * \text{resJointC}[x][y]) \gg 1$ $\text{resCr}[x][y] = \text{resJointC}[x][y]$	3

Ba cơ chế phụ lập mã chroma chung được mô tả ở trên ở **Error! Reference source not found.** chỉ được hỗ trợ ở các slice I. Ở các slice P và B, chỉ cơ chế 2 được hỗ trợ. Do đó, ở các slice P và B, thành phần cú pháp $\text{tu_joint_cbr_residual_flag}$ chỉ xuất hiện nếu toàn bộ các CBF chroma bằng 1.

Cơ chế JCCR có thể được kết hợp với cơ chế rẽ nhánh biến đổi TS chroma. Nhằm

tăng tốc độ quyết định của bộ mã hoá, lựa chọn biến đổi JCCR phụ thuộc vào liệu phép lập mã độc lập các thành phần Cb và Cr chọn DCT-2 hay TS làm biến đổi tốt nhất, và liệu có các hệ số khác không khi lập mã chroma độc lập. Cụ thể là, nếu một thành phần chroma chọn DCT-2 (hoặc TS) và thành phần khác đều bằng 0, hoặc toàn bộ các thành phần chroma chọn DCT-2 (hoặc TS), thì chỉ DCT-2 (hoặc TS) sẽ được xét khi mã hoá JCCR. Nếu không thì, nếu một thành phần chọn DCT-2 và thành phần khác chọn TS, thì cả DCT-2 và TS sẽ được xét khi mã hoá JCCR. Chi tiết hơn có thể được tìm thấy ở JVET-N0054 (J. Lainema, “CE7: Lập mã chung các phần dư màu (CE7-1),” Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 14: Geneva, CH, Ngày 19 đến 27 tháng 3 năm 2019, Tài liệu: JVET-N0413).

Các phân vùng con nội ảnh ISP (ISP - Intra sub-partitions) trong VVC

Các phân vùng con nội ảnh (ISP) chia các khối được dự đoán nội ảnh luma theo phương đứng hoặc phương ngang thành 2 hoặc 4 phân vùng con tùy theo kích cỡ khối. Ví dụ, kích cỡ khối nhỏ nhất cho ISP để phân chia một khối là 4×8 (hoặc 8×4). Nếu kích cỡ khối lớn hơn 4×8 (hoặc 8×4) thì khối tương ứng đó được chia thành 4 phân vùng con. Cần được lưu ý rằng các khối ISP $M \times 128$ (với $M \leq 64$) và $M \times 128$ (với $N \leq 64$) có thể gây ra lỗi tiềm năng với VDPUs 64×64 . Ví dụ, CU $M \times 128$ ở trường hợp cây đơn có TB luma $M \times 128$ và hai TB chroma tương ứng $(M/2) \times 64$. Nếu CU sử dụng ISP, thì TB luma sẽ được chia thành bốn TB $M \times 32$ (chỉ phân chia ngang là có thể), mỗi chúng nhỏ hơn khối 64×64 . Tuy nhiên, ở cấu trúc hiện tại của ISP các khối chroma không được chia. Vì vậy, toàn bộ các thành phần chroma sẽ có kích cỡ lớn hơn khối 32×32 . Tương tự, trường hợp tương tự có thể xuất hiện với CU $126 \times N$ sử dụng ISP. Do đó, hai trường hợp vừa nêu là một vấn đề đối với đường truyền giải mã 64×64 . Vì lý do này, các kích cỡ CU mà có thể sử dụng ISP bị giới hạn lớn nhất bằng 64×64 . **Error! Reference source not found.** 2A và Fig. 2B thể hiện các ví dụ về hai trường hợp vừa nêu. Toàn bộ các phân vùng con thoả mãn điều kiện có ít nhất 16 mẫu. Fig. 2A minh hoạ trường hợp cho kích cỡ khối bằng 4×8 hoặc 8×4 . Trong trường hợp này, khối 210 được phân chia ngang thành hai khối con 220 hoặc được phân chia đứng thành hai khối con 230. Fig. 2B minh hoạ trường hợp cho kích cỡ khối không phải là 4×8 và 8×4 . Trong trường hợp này, khối 240 được phân chia ngang thành bốn khối con 250 hoặc được phân chia đứng thành bốn

khối con 260.

Theo ISP, sự phụ thuộc của dự đoán khối con $1 \times N$ hoặc $2 \times N$ vào các giá trị khôi phục của các khối con $1 \times N$ hoặc $2 \times N$ được giải mã trước đó của khối lập mã là không được chấp nhận để cho chiều rộng nhỏ nhất của dự đoán cho các khối con bằng bốn mẫu. Ví dụ, khối lập mã $8 \times N$ ($N > 4$) mà được lập mã bằng ISP có phân chia đứng được phân chia thành hai vùng dự đoán mỗi vùng có kích cỡ $4 \times N$ và bốn biến đổi có kích cỡ $2 \times N$. Tương tự, khối lập mã $4 \times N$ mà được lập mã bằng ISP có phân chia đứng được dự đoán bằng cách sử dụng toàn bộ khối $4 \times N$; bốn biến đổi mỗi biến đổi có kích cỡ $1 \times N$ được sử dụng. Mặc dù các kích cỡ biến đổi $1 \times N$ và $2 \times N$ là được phép, nhưng cần khẳng định rằng phép biến đổi các khối vừa nêu ở các vùng $4 \times N$ có thể được thực hiện song song. Ví dụ, khi vùng dự đoán $4 \times N$ chứa bốn biến đổi $1 \times N$, thì không có biến đổi theo phương ngang; biến đổi theo phương đứng có thể được thực hiện dưới dạng một biến đổi $4 \times N$ duy nhất theo phương đứng. Tương tự, khi vùng dự đoán $4 \times N$ chứa hai khối biến đổi $2 \times N$, thuật toán biến đổi của hai khối $2 \times N$ theo từng phương (ngang và đứng) có thể được thực hiện song song. Do đó, không có độ trễ cộng dồn vào khi xử lý những khối nhỏ hơn vừa nêu so với khi xử lý các khối nội ảnh 4×4 được lập mã thông thường.

Đối với mỗi phân vùng con, các mẫu được khôi phục được tạo ra bằng cách cộng tín hiệu dư vào tín hiệu dự đoán. Ở đây, tín hiệu dư được tạo ra bởi các phép xử lý chẳng hạn giải mã entropy, lượng tử hoá ngược và biến đổi ngược. Vì vậy, các giá trị mẫu được khôi phục của mỗi phân vùng con có thể sử dụng để tạo ra dự đoán phân vùng con kế tiếp, và mỗi phân vùng con được xử lý lặp lại. Ngoài ra, phân vùng con đầu tiên sẽ được xử lý là phân vùng chứa mẫu trên cùng-bên trái của CU và sau đó tiếp tục xử lý hướng xuống (phân chia ngang) hoặc hướng sang phải (phân chia đứng). Kết quả là, các mẫu tham chiếu được sử dụng để tạo ra tín hiệu dự đoán các phân vùng con chỉ được phân bố tại mép bên trái và phía trên của các đường phân chia. Tất cả các phân vùng con đều dùng chung cơ chế nội ảnh giống nhau. Những điều nêu dưới đây là tổng hợp những tương tác của ISP với các công cụ lập mã khác.

– Đa đường tham chiếu MRL (Multiple Reference Line - MRL): nếu khối có chỉ số MRL khác 0, thì cơ chế lập mã ISP sẽ được suy diễn bằng 0 và vì vậy thông tin cơ chế ISP sẽ không được truyền đến bộ giải mã.

– Kích cỡ nhóm hệ số lập mã entropy: các kích cỡ của các khối con lập mã entropy được sửa đổi để chúng có 16 mẫu ở mọi trường hợp, như được thể hiện trên **Error!**

Reference source not found.. Chú ý rằng các kích cỡ mới chỉ ảnh hưởng đến các khối được tạo ra bởi ISP trong đó một trong số các kích thước nhỏ hơn 4 mẫu. Ở toàn bộ các trường hợp khác, các nhóm hệ số giữ nguyên các kích thước 4×4 .

– Lập mã CBF: giả sử có ít nhất một trong số các phân vùng con có CBF khác không. Do đó, nếu n là số lượng các phân vùng con và $n - 1$ phân vùng con đầu tiên đã có CBF bằng không, thì CBF của n -th phân vùng con được xem là 1.

– Cách dùng MPM: cờ MPM sẽ được suy diễn bằng 1 ở khối được lập mã bằng cơ chế ISP, và danh mục MPM được sửa đổi để không chứa cơ chế DC và ưu tiên các cơ chế nội ảnh theo phương ngang cho phân chia ngang ISP và các cơ chế nội ảnh theo phương đứng cho phân chia đứng ISP.

– Giới hạn kích cỡ biến đổi: toàn bộ biến đổi ISP có chiều dài lớn hơn 16 điểm sử dụng DCT-II.

– PDPC: khi CU sử dụng cơ chế lập mã ISP, các bộ lọc PDPC sẽ không được áp dụng cho các phân vùng con thành quả.

– Cờ MTS: nếu CU sử dụng cơ chế lập mã ISP, cờ CU MTS sẽ được thiết lập bằng 0 và nó sẽ không được truyền đến bộ giải mã. Vì vậy, bộ mã hoá sẽ không thực hiện các kiểm nghiệm RD đối với các biến đổi hiện có khác nhau cho từng phân vùng con thành quả. Lựa chọn phép biến đổi cho cơ chế ISP sẽ không được cố định mà theo cơ chế nội ảnh, thứ tự xử lý và kích cỡ khối được dùng. Do đó, không cần báo hiệu. Ví dụ, cho t_H và t_V là các biến đổi ngang và đứng được chọn tương ứng cho phân vùng con $w \times h$, trong đó w là chiều rộng và h là chiều cao. Khi đó biến đổi được chọn theo các quy tắc sau:

- Nếu $w = 1$ hoặc $h = 1$, thì không có biến đổi ngang hoặc đứng tương ứng.
- Nếu $w = 2$ hoặc $w > 32$, $t_H = \text{DCT-II}$
- Nếu $h = 2$ hoặc $h > 32$, $t_V = \text{DCT-II}$
- Nếu không thì, biến đổi được chọn như ở Bảng 9.

Bảng 8. Kích cỡ nhóm hệ số lập mã entropy

Kích cỡ khối	Kích cỡ nhóm hệ số
$1 \times N, N \geq 16$	1×16
$N \times 1, N \geq 16$	16×1
$2 \times N, N \geq 8$	2×8
$N \times 2, N \geq 8$	8×2
Mọi trường hợp $M \times N$ khác	4×4

Bảng 9. Chọn biến đổi theo cơ chế nội ảnh

Bảng 8. Kích cỡ nhóm hệ số lập mã entropy

Kích cỡ khối	Kích cỡ nhóm hệ số
$1 \times N, N \geq 16$	1×16
$N \times 1, N \geq 16$	16×1
$2 \times N, N \geq 8$	2×8
$N \times 2, N \geq 8$	8×2
Mọi trường hợp $M \times N$ khác	4×4

Bảng 9. Chọn biến đổi theo cơ chế nội ảnh

Cơ chế nội ảnh	t_H	t_V
Phẳng Góc 31,32,34,36,37	DST-VII	DST-VII
DC Góc 33, 35	DCT-II	DCT-II
Góc 2, 4, 6...28,30 Góc 39,41,43...63,65	DST-VII	DCT-II
Góc 3,5,7...27,29 Góc 38,40,42...64,66	DCT-II	DST-VII

Theo cơ chế ISP, toàn bộ 67 phương vị nội ảnh đều được chấp nhận. PDPC cũng được áp dụng nếu chiều rộng và chiều cao tương ứng bằng ít nhất chiều dài 4 mẫu. Ngoài ra, điều kiện để chọn phép lọc nội suy nội ảnh không còn tồn tại nữa, và phép lọc bậc ba (DCT-IF) luôn được áp dụng cho phép nội suy vị trí phân đoạn theo cơ chế ISP. Chi tiết hơn có thể được tìm thấy ở JVET-M0102 (S. De-Luxán-Hernández, và các cộng sự, “CE3: Cơ chế lập mã các phân vùng con nội ảnh (Các kiểm nghiệm 1.1.1 và 1.1.2)”, Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 13: Marrakech, MA, Ngày 9 đến 18 tháng 1 năm 2019, Tài liệu: JVET-M0102).

“Khởi” ở sáng chế này có thể có nghĩa là TU/TB/CU/CB/PU/PB.

Bản chất kỹ thuật của sáng chế

Phương pháp và thiết bị mã hoá và giải mã video bằng cách sử dụng cơ chế biến đổi không phân tách tần số thấp (LFNST) được bộc lộ. Theo sáng chế, trong quá trình giải mã, dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời được nhận, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu lập mã của CU hiện thời. Một hoặc nhiều chỉ báo cờ khối được lập mã (cờ CBF) cho một hoặc nhiều TB mong muốn được kiểm tra dựa vào dữ liệu lập mã. Cú pháp LFNST được phân tích nếu một hoặc nhiều điều kiện được thoả mãn. Cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng

cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng. Các điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo CBF cho một hoặc nhiều TB mong muốn là sai (false). CU hiện thời được giải mã tại bộ giải mã theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST.

Trong quá trình mã hoá, dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời được nhận, trong đó dữ liệu đầu vào tương ứng với dữ liệu được biến đổi chính. Phép LFNST được áp dụng dựa vào nhân LFNST để suy dẫn dữ liệu đầu ra tạm thời. Cú pháp LFNST được xác định và được báo hiệu nếu một hoặc nhiều điều kiện được thoả mãn. Các điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo cờ khởi được lập mã (CBF) cho một hoặc nhiều khối biến đổi (nhiều TB) mong muốn là false. CU hiện thời được mã hoá theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST được xác định.

Theo một phương án, các TB mong muốn tương ứng với một hoặc nhiều TB có các cờ rẽ nhánh biến đổi TS (cờ TS) không bằng 0.

Theo một phương án, ở cây phân chia luma, CU hiện thời tương ứng với khối lập mã luma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma. Theo phương án khác, ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB chroma. Theo phương án khác nữa, ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma.

Theo một phương án, các TB mong muốn tương ứng với TB được định trước cho từng khối lập mã ở CU hiện thời. Ví dụ, TB được định trước tương ứng với TB đầu tiên cho từng khối lập mã ở CU hiện thời.

Theo một phương án, nếu toàn bộ các chỉ báo CBF cho các TB mong muốn là false, cơ chế LFNST được chấp nhận cho CU hiện thời. Theo phương án khác, nếu toàn bộ các chỉ báo CBF cho các TB mong muốn có các cờ TS không bằng 0 là false, cơ chế LFNST được chấp nhận cho CU hiện thời. Theo phương án khác nữa, nếu bất kỳ một trong số các chỉ báo CBF cho các TB mong muốn có các cờ TS không bằng 0 là true, cơ chế LFNST không được chấp nhận cho CU hiện thời.

Mô tả vắn tắt các hình vẽ

Nhiều phương án khác nhau của sáng chế được trình bày dưới dạng các ví dụ sẽ được mô tả chi tiết dựa vào các hình vẽ dưới đây, trong đó các số chỉ dẫn giống nhau biểu diễn các phần tử giống nhau, và trong đó:

Fig. 1 minh hoạ ví dụ về phép LFNST (biến đổi không phân tách tần số thấp);

Error! Reference source not found. 2A và Fig. 2B thể hiện các ví dụ về các phân vùng con nội ảnh (ISP);

Fig. 3 minh hoạ lưu đồ của phép phân tích hệ thống giải mã được lấy làm ví dụ có tích hợp LFNST theo phương án của sáng chế; và

Fig. 4 minh hoạ lưu đồ của phép phân tích hệ thống mã hoá được lấy làm ví dụ có tích hợp LFNST theo phương án của sáng chế.

Mô tả chi tiết sáng chế

Sẽ dễ dàng được hiểu rằng các phần tử của sáng chế, như được mô tả và được minh hoạ một cách tổng quát trên các hình vẽ ở bản mô tả này, có thể được bố trí và được thiết kế theo nhiều dạng cấu hình khác nhau. Do đó, phần mô tả chi tiết hơn dưới đây về các phương án của các hệ thống và các phương pháp của sáng chế, như được biểu diễn trên các hình vẽ, là không nhằm giới hạn phạm vi của sáng chế, như được yêu cầu bảo hộ, mà chỉ nhằm minh hoạ các phương án được chọn của sáng chế. Tham chiếu toàn bộ bản mô tả này theo “một phương án”, “một số phương án”, hoặc cách diễn đạt tương tự có nghĩa là một dấu hiệu, cấu trúc, hoặc đặc tính cụ thể nào đó được mô tả theo các phương án có thể thuộc ít nhất một phương án của sáng chế. Do đó, sự xuất hiện của các cụm từ “theo một phương án” hoặc “theo một số phương án” ở những phần khác nhau trong toàn bộ bản mô tả này không có nghĩa là nhất thiết phải đề cập đến phương án giống nhau.

Ngoài ra, các dấu hiệu, các cấu trúc, hoặc các đặc tính đã được mô tả có thể được kết hợp theo bất kỳ cách thức phù hợp nào theo một hoặc nhiều phương án. Tuy nhiên, một người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương tự này sẽ biết rằng sáng chế có thể được thực hiện mà không có một hoặc nhiều hơn một trong số các phương án chi tiết cụ thể, hoặc được thực hiện với các phương pháp khác, các phần tử khác v.v. Mặt khác, các cấu trúc đã biết rõ, hoặc các cách thức đã biết rõ không được thể hiện hoặc được mô tả chi tiết nhằm tránh làm rối các khía cạnh của sáng chế. Các phương án

được minh họa của sáng chế sẽ được hiểu rõ nhất bằng cách tham chiếu đến các hình vẽ, trong đó các bộ phận giống nhau được gán các ký tự số giống nhau. Phần mô tả dưới đây được dự định chỉ làm ví dụ, và minh họa giản lược các phương án được chọn nhất định của thiết bị và các phương pháp của sáng chế như được yêu cầu bảo hộ ở đây.

Các kết hợp giữa LFNST với phép rẽ nhánh biến đổi sẽ không được chấp nhận bởi vì khi phép rẽ nhánh biến đổi được áp dụng, không có phép biến đổi nào (biến đổi nhân/chính và/hoặc biến đổi phụ) sẽ được sử dụng. Theo VVC phiên bản 7 (B. Bross, và các cộng sự, “Lập mã video đa năng (Phiên bản 7)”, Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 16: Geneva, CH, Ngày 1 đến 11 tháng 10 năm 2019, Tài liệu: JVET-P2001), cú pháp cho cơ chế rẽ nhánh biến đổi được báo hiệu/được phân tích tại phân mức TB. Mặt khác, cú pháp cho LFNST được báo hiệu/được phân tích tại phân mức CU sau khi toàn bộ các TU/TB nằm trong các CU/CB đó được báo hiệu/được phân tích. Vì vậy, theo dự thảo VVC hiện tại (như được thể hiện ở bảng 10A), các điều kiện báo hiệu/phân tích cho LFNST sẽ xét đến cờ rẽ nhánh biến đổi cho luma như sau. Như được thể hiện ở bảng cú pháp dưới đây, các điều kiện hiện có bao gồm kiểm tra phép rẽ nhánh biến đổi luma (tức là, $\text{transform_skip_flag}[x_0][y_0][0] == 0$) để ngăn không cho kết hợp đã nêu. Đối với phép kiểm tra vừa nêu, mô hình kiểm nghiệm VVC phiên bản 7 (VTM7, J. Chen, và các cộng sự, “Mô tả thuật toán cho lập mã video đa năng và mô hình kiểm nghiệm 7 (VTM 7)”, Nhóm chuyên gia video chung (JVET) của ITU-T SG 16 WP 3 và ISO/IEC JTC 1/SC 29/WG 11, Hội nghị lần thứ 16: Geneva, CH, Ngày 1 đến 11 tháng 10 năm 2019, Tài liệu: JVET-P2002) đưa ra các mã tương thích với VVC phiên bản 7. Bảng cú pháp dùng để lập mã phần dư theo JVET-P2001 được thể hiện ở bảng 10B.

Bảng 10A. Các điều kiện báo hiệu/phân tích cho LFNST theo VVC phiên bản 7

<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	Biến mô tả
<code>chType = treeType == DUAL_TREE_CHROMA ? 1 : 0</code>	
<code>...</code>	
<code>if(CuPredMode[chType][x0][y0] == MODE_INTRA CuPredMode[chType][x0][y0] == MODE_PLT) {</code>	
<code>if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {</code>	
<code>if(pred_mode_plt_flag) {</code>	
<code>palette_coding(x0, y0, cbWidth, cbHeight, treeType)</code>	
<code>} else {</code>	

if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
...	
}	
}	
}	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, treeType)	
else {	
if(!cu_act_enabled_flag) {	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER hoặc MODE_IBC */	
...	
}	
if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
...	
LfstDcOnly = 1	
LfstZeroOutSigCoeffFlag = 1	

MtsZeroOutSigCoeffFlag = 1	
transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)	
lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType == ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : ((IntraSubPartitionsSplitType == ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight)	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && transform_skip_flag[x0][y0][0] == 0 && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)	
lfnst_idx	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && lfnst_idx == 0 && transform_skip_flag[x0][y0][0] == 0 && Max(cbWidth, cbHeight) <= 32 && IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT && cu_sbt_flag == 0 && MtsZeroOutSigCoeffFlag == 1 && tu_cbf_luma[x0][y0]) {	
if(((CuPredMode[chType][x0][y0] == MODE_INTER && sps_explicit_mts_inter_enabled_flag) (CuPredMode[chType][x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)))	
mts_idx	ae(v)
}	
}	

Bảng 10B. Bảng cú pháp dùng để lập mã phân dư theo VVC phiên bản 7

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Biến mô tả
if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth == 5 && log2TbHeight < 6)	
log2ZoTbWidth = 4	
else	
log2ZoTbWidth = Min(log2TbWidth, 5)	

if(sps_mts_enabled_flag && cu_sbt_flag && cIdx == 0 && log2TbWidth < 6 && log2TbHeight == 5)	
log2ZoTbHeight = 4	
else	
log2ZoTbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth + log2TbHeight > 3) {	
if(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} else if(log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH))) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH] [lastSubBlock][1]	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	

$yC = (yS \ll \log2SbH) +$ $DiagScanOrder[\log2SbW][\log2SbH][lastScanPos][1]$	
$\} \text{ while}((xC \neq LastSignificantCoeffX) \mid \mid (yC \neq LastSignificantCoeffY$ $\}))$	
$\text{if}(lastSubBlock == 0 \ \&\& \log2TbWidth \geq 2 \ \&\& \log2TbHeight \geq 2$ $\&\&$ $\text{!transform_skip_flag}[x0][y0][cIdx] \ \&\& \text{lastScanPos} > 0)$	
$LfstDcOnly = 0$	
$\text{if}((lastSubBlock > 0 \ \&\& \log2TbWidth \geq 2 \ \&\& \log2TbHeight \geq 2) \mid \mid$ $(lastScanPos > 7 \ \&\& (\log2TbWidth == 2 \mid \mid \log2TbWidth == 3) \ \&\&$ $\log2TbWidth == \log2TbHeight))$	
$LfstZeroOutSigCoeffFlag = 0$	
$\text{if}((LastSignificantCoeffX > 15 \mid \mid LastSignificantCoeffY > 15) \ \&\& cIdx$ $== 0)$	
$MtsZeroOutSigCoeffFlag = 0$	
$\dots$	
$\}$	
$\}$	
$\}$	

Theo điều kiện báo hiệu/phân tích LFNST hiện đang có, hai vấn đề được phát hiện. Một vấn đề là khi luma và chroma sử dụng các cây phân chia khác nhau, thì không thể tìm ra cờ rẽ nhánh biến đổi cho luma (tức là, `transform_skip_flag[x0][y0][0]`) khi CU hiện thời nằm ở cây phân chia chroma (tức là, trường hợp có `treeType == DUAL_TREE_CHROMA`). Một vấn đề khác gây ra bởi sự mở rộng của phép rẽ nhánh biến đổi cho chroma như được bộc lộ trong JVET-P0058. Phép kiểm tra được nêu ở trên cần được mở rộng để chứa các phép kiểm tra Cb và Cr. Một số phương pháp được đề xuất để giải quyết các vấn đề vừa nêu.

Phép kiểm tra được đề xuất sẽ tiến hành xem xét điều kiện của cờ (các cờ) rẽ nhánh biến đổi của M TB trong CU. Đối với TB có một hoặc nhiều mức hệ số biến đổi không bằng 0, cờ rẽ nhánh biến đổi cho TB được sử dụng để biểu thị liệu các thuật toán biến đổi được áp dụng cho TB hay không và phép kiểm tra được đề xuất được sử dụng để ngăn không cho TB có các kết hợp giữa LFNST với phép rẽ nhánh biến đổi. Như đã nêu trước đó, ở cây phân chia tương ứng mà có thể là cây phân chia luma (`DUAL_TREE_LUMA`), cây phân chia chroma (`DUAL_TREE_CHROMA`), hoặc cây phân chia đơn (`SINGLE_TREE`), có một hoặc nhiều TB ở CU hiện thời. M TB tương ứng với bộ các TB được chọn, được gọi là các TB mong muốn. Điều kiện của cờ (các cờ) rẽ nhánh biến đổi của bộ TB mong muốn được kiểm tra. Việc đạt yêu cầu kiểm tra nghĩa là cờ (các cờ) rẽ nhánh biến đổi cho toàn bộ các M TB là false (tức là, cờ (các cờ)

rẽ nhánh biến đổi cho toàn bộ các M TB bằng 0); nói cách khác, việc đạt yêu cầu kiểm tra nghĩa là điều kiện mong muốn (tương ứng với tất cả TB mong muốn ở bộ TB mong muốn có chỉ báo cơ chế TS là false) được thoả mãn. Nói cách khác, điều kiện cờ (các cờ) cơ chế rẽ nhánh biến đổi được thoả mãn nếu không có TB (các TB) được chọn nào sử dụng cơ chế rẽ nhánh biến đổi. Sau khi đạt yêu cầu kiểm tra (tức là, điều kiện cờ (các cờ) cơ chế rẽ nhánh biến đổi được thoả mãn), các điều kiện báo hiệu/phân tích cho LFNST liên quan đến cơ chế rẽ nhánh biến đổi được thoả mãn và cú pháp cho LFNST có thể được báo hiệu/được phân tích nếu các điều kiện báo hiệu/phân tích LFNST khác cũng được thoả mãn.

Theo một phương án, M TB chỉ chứa thành phần đầu tiên ở từng cây phân chia luma/chroma. Ví dụ của bảng cú pháp được đề xuất được thể hiện như sau.

Bảng 11. Bảng cú pháp mẫu về điều kiện báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> If(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && transform_skip_flag[x0][y0][chType] == 0 && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
lfnst_idx	ae(v)
}	

Theo phương án khác, để cây phân chia đơn được sử dụng cho cả thành phần luma và chroma, thì M TB chứa một hoặc nhiều thành phần.

Theo một phương án phụ, M TB tham chiếu đến một thành phần được chọn. Ví dụ, M TB tham chiếu đến thành phần đầu tiên. Theo ví dụ khác, M TB tham chiếu đến thành phần đầu tiên Y (tức là, thành phần luma). Bảng cú pháp mẫu theo một phương án được thể hiện như sau. Theo ví dụ khác, M TB có thể là một thành phần bất kỳ trong cây phân chia.

Bảng 12. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == SINGLE_TREE && transform_skip_flag[x0][y0][0] == 0) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
lfnst_idx	ae(v)
}	

Theo phương án khác, khi cây phân chia không là cây chroma (tức là, cây phân chia chứa thành phần Y (tức là, luma)), M TB tham chiếu đến TB (các TB) Y (tức là, luma). Bảng cú pháp mẫu theo một phương án được thể hiện như sau.

Bảng 13. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType != DUAL_TREE_CHROMA && transform_skip_flag[x0][y0][0] == 0) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
lfnst_idx	ae(v)
}	

Theo phương án khác, đối với cây phân chia chroma, M TB chứa toàn bộ các thành phần chroma (ví dụ Cb và Cr). Nếu cờ rẽ nhánh biến đổi bất kỳ cho các thành phần chroma là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt.

Theo phương án khác, đối với cây phân chia luma, M TB chứa tất cả các thành phần (ví dụ Y). Nếu cờ rẽ nhánh biến đổi bất kỳ cho các thành phần này là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp mẫu theo phương án này được thể hiện ở bảng 14.

Theo phương án khác, đối với cây đơn được sử dụng cho các thành phần luma và chroma, M TB chứa tất cả các thành phần (ví dụ Y, Cb, và Cr). Nếu cờ rẽ nhánh biến đổi bất kỳ cho các thành phần này là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp mẫu theo phương án này được thể hiện ở bảng 14.

Bảng 14. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == DUAL_TREE_CHROMA? (transform_skip_flag[x0][y0][1] == 0 transform_skip_flag[x0][y0][2] == 0) : (treeType == DUAL_TREE_LUMA ? transform_skip_flag[x0][y0][0] == 0: (transform_skip_flag[x0][y0][0] == 0 transform_skip_flag[x0][y0][1] == 0 transform_skip_flag[x0][y0][2] == 0))) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfnst_idx </pre>	ae(v)
<pre> } </pre>	

Theo phương án khác, đối với cây phân chia chroma, M TB chứa toàn bộ các thành phần chroma (ví dụ Cb và Cr). Nếu toàn bộ các cờ rẽ nhánh biến đổi cho các thành phần chroma là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt.

Theo phương án khác, đối với cây phân chia luma, M TB chứa toàn bộ các thành phần luma (ví dụ Y). Nếu toàn bộ các cờ rẽ nhánh biến đổi cho các thành phần này là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp mẫu

theo phương án này được thể hiện ở bảng 15.

Theo phương án khác, đối với cây đơn được sử dụng cho các thành phần luma và chroma, M TB chứa tất cả các thành phần (ví dụ Y, Cb, và Cr). Nếu toàn bộ các cờ rẽ nhánh biến đổi cho các thành phần này là false (tức là, cờ rẽ nhánh biến đổi bằng 0), thì phép kiểm tra đó đạt. Bảng cú pháp mẫu theo phương án này được thể hiện ở bảng 15.

Theo phương án khác nữa, hai hoặc nhiều hơn hai trong số ba phương án nêu trên có thể được kết hợp. Ví dụ, phương án kết hợp vừa nêu có thể chỉ kiểm tra transform_skip_flag luma khi cây phân chia không là cây phân chia chroma (ví dụ không là DUAL_TREE_CHROMA) và chỉ kiểm tra transform_skip_flag chroma khi cây phân chia không là cây phân chia luma (ví dụ không là DUAL_TREE_LUMA). Bảng cú pháp mẫu theo phương án này được thể hiện ở bảng 15.

Bảng 15. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == DUAL_TREE_CHROMA? (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0) : (treeType == DUAL_TREE_LUMA ? transform_skip_flag[x0][y0][0] == 0 : (transform_skip_flag[x0][y0][0] == 0 && transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0))) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfstWidth, lfstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfstDcOnly == 0) && LfstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfst_idx </pre>	ae(v)
<pre> } </pre>	

Bảng cú pháp mẫu khác để kết hợp ba phương án nêu trên được thể hiện ở bảng 16.

Bảng 16. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 &&CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfnst_idx </pre>	ae(v)
<pre> } </pre>	

Theo phương án khác, khi phép kiểm tra không đạt, thì cú pháp cho LFNST (ví dụ, chỉ số LFNST) không được báo hiệu/được phân tích.

Theo một phương án phụ, cú pháp cho LFNST (chỉ số LFNST) được xem là bằng 0 (tức là, LFNST không được áp dụng).

Theo phương án khác, một tương hợp dòng bit được yêu cầu để xử lý trường hợp khi phép kiểm tra không đạt. Tương hợp dòng bit vừa nêu là như sau. Có yêu cầu tương hợp dòng bit đó là trị số của lfnst_index sẽ không lớn hơn 0 khi phép kiểm tra không đạt. Dưới đây, ví dụ về tương hợp dòng bit được minh hoạ cho trường hợp kiểm tra “cờ (các cờ) rẽ nhánh biến đổi cho M TB, trong đó M TB chỉ chứa thành phần đầu tiên ở từng cây phân chia luma/chroma”. Yêu cầu tương hợp dòng bit tương ứng với trị số của lfnst_index sẽ không lớn hơn 0 khi trị số của cờ rẽ nhánh biến đổi cho thành phần đầu tiên ở từng cây phân chia luma/chroma (ví dụ, transform_skip_flag[x0][y0][chType], trong đó nếu treeType == DUAL_TREE_CHROMA, chType biểu thị 1 (tức là, Cb); nếu không thì, chType biểu thị 0 (tức là, Y)) lớn hơn 1.

Theo phương án khác, một biến có thể được tạo ra ở câu lệnh hoặc phần mềm để biểu thị liệu có báo hiệu/phân tích cú pháp cho LFNST. Trị số của biến này được cập nhật theo một hoặc nhiều điều kiện báo hiệu/phân tích LFNST hiện có và/hoặc một hoặc nhiều phép kiểm tra được đề xuất ở sáng chế này. Ví dụ, biến này được khởi tạo bằng 1

và nếu bất kỳ điều kiện báo hiệu/phân tích LFNST và/hoặc một hoặc nhiều phép kiểm tra được đề xuất ở sáng chế này không được thoả mãn, thì biến này được chuyển thành 0 và cú pháp cho LFNST không được báo hiệu/được phân tích.

Theo phương án khác, cơ chế kiểm tra hợp nhất được sử dụng cho các cây phân chia khác nhau cho luma và chroma. Ví dụ, khi luma và chroma sử dụng các cây lưỡng (tức là, các cây phân chia riêng), CU luma tuân theo cây phân chia luma và CU chroma tuân theo cây phân chia chroma. Cơ chế kiểm tra hợp nhất vừa nêu là cơ chế mà LFNST được huỷ bỏ nếu bất kỳ một trong số các cờ rẽ nhánh biến đổi cho tất cả các thành phần ở CU hiện thời đang sử dụng phép rẽ nhánh biến đổi.

Do điều kiện ràng buộc kích cỡ hiện tại đối với LFNST, LFNST có thể được áp dụng khi một CU/CB chứa một TU/TB. Phép kiểm tra có thể xem xét cờ rẽ nhánh biến đổi cho một TU/TB thay vì nhiều TU/TB. Khi một CU/CB chứa nhiều TU/TB, phép kiểm tra được đề xuất ở trên tuân theo một hoặc nhiều TU/TB ở CU/CB đó. Theo một phương án, phép kiểm tra được đề xuất ở trên tuân theo toàn bộ các TU/TB ở CU/CB đó. Theo phương án khác, phép kiểm tra tuân theo bất kỳ một trong số các TU/TB ở CU/CB đó (ví dụ TU/TB đầu tiên hoặc TU/TB cuối). Ví dụ, ở cây phân chia luma, bộ TB mong muốn bao gồm TB luma đầu tiên cho CB luma ở CU hiện thời. Ở ví dụ khác, ở cây phân chia chroma, bộ TB mong muốn bao gồm TB Cb đầu tiên cho CB Cb ở CU hiện thời và TB Cr đầu tiên cho CB Cr ở CU hiện thời. Ở ví dụ khác, ở cây phân chia đơn, bộ TB mong muốn bao gồm TB luma đầu tiên cho CB luma ở CU hiện thời, TB Cb đầu tiên cho CB Cb ở CU hiện thời, và TB Cr đầu tiên cho CB Cr ở CU hiện thời. Theo phương án khác, phép kiểm tra tuân theo tập con bất kỳ tập con trong số các TU/TB ở CU/CB đó.

Ngoài ra, cách sử dụng LFNST có thể bị giới hạn theo một số điều kiện. Ở cấu trúc hiện tại, LFNST được áp dụng cho CU nội ảnh ở cả slice nội ảnh và liên ảnh, và luma và/hoặc chroma. Nếu cây lưỡng được sử dụng, thì các tham số LFNST cho luma và chroma được báo hiệu/được phân tích riêng. Đối với slice liên ảnh, trong đó cây lưỡng được huỷ bỏ, chỉ số LFNST duy nhất được báo hiệu/được phân tích và được sử dụng cho Luma và/hoặc chroma. Ở sáng chế này, LFNST chroma được huỷ bỏ theo một số trường hợp.

Theo một phương án, đối với cây đơn, LFNST chroma được huỷ bỏ.

Theo một phương án phụ, khi LFNST chroma được huỷ bỏ, chỉ số LFNST vẫn

được báo hiệu/được phân tích và có thể được sử dụng cho luma.

Theo phương án khác, LFNST chroma được huỷ bỏ.

Theo một phương án phụ, khi LFNST chroma được huỷ bỏ, chỉ số LFNST không được báo hiệu/được phân tích ở cây lưỡng chroma.

Theo phương án khác, LFNST không thể được sử dụng cho TB mặc dù chỉ số LFNST cho CU chứa TB lớn hơn 0. Biến, `applyLfnstFlag`, được tạo ra để biểu thị liệu LFNST có thể được sử dụng hay không. Nếu `applyLfnstFlag` bằng 0, LFNST không thể được sử dụng. Nếu `applyLfnstFlag` bằng 1, LFNST có thể được sử dụng.

Ví dụ, đối với cây đơn, LFNST chroma được huỷ bỏ. Biến `applyLfnstFlag` được suy dẫn như sau: (trong đó `xTbY` và `yTbY` là vị trí mẫu luma tương ứng cho TB, `cIdx` biểu thị thành phần cho TB (ví dụ `cIdx` bằng 0 biểu thị thành phần luma, `cIdx` bằng 1 biểu thị thành phần Cb, và `cIdx` bằng 2 biểu thị thành phần Cr), `lfnst_idx` là chỉ số LFNST cho CU, và `nTbW` và `nTbH` biểu thị chiều rộng và chiều cao của TB)

- Nếu (1) `treeType` bằng `SINGLE_TREE`, (2) `lfnst_idx` không bằng 0, (3) `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` bằng 0, (4) `cIdx` bằng 0 và (5) cả `nTbW` và `nTbH` lớn hơn hoặc bằng 4, `applyLfnstFlag` được thiết lập bằng 1. (Tập con bất kỳ trong số từ (1) đến (5) có thể được sử dụng theo điều kiện vừa nêu)

- Nếu không thì, nếu (1) `treeType` không bằng `SINGLE_TREE`, (2) `lfnst_idx` không bằng 0, (3) `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` bằng 0 và (4) cả `nTbW` và `nTbH` lớn hơn hoặc bằng 4, `applyLfnstFlag` được thiết lập bằng 1. (Tập con bất kỳ trong số từ (1) đến (4) có thể được sử dụng theo điều kiện vừa nêu)

- Nếu không thì, `applyLfnstFlag` được thiết lập bằng 0.

Theo ví dụ khác, LFNST chroma được huỷ bỏ. Biến `applyLfnstFlag` được suy dẫn như sau:

- Nếu (1) `lfnst_idx` không bằng 0, (2) `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` bằng 0, (3) `cIdx` bằng 0 và (4) cả `nTbW` và `nTbH` lớn hơn hoặc bằng 4, `applyLfnstFlag` được thiết lập bằng 1. (Tập con bất kỳ trong số từ (1) đến (4) có thể được sử dụng theo điều kiện vừa nêu)

- Nếu không thì, `applyLfnstFlag` được thiết lập bằng 0.

Theo phương án phụ khác, `applyLfnstFlag` có thể được sử dụng ở một hoặc nhiều phân mục liên quan đến LFNST. Ví dụ, chỉ số LFNST được tham chiếu ở phân mục tương ứng ở tiêu chuẩn dự thảo.

8.7.4 Phép biến đổi cho các hệ số biến đổi được phóng tỷ lệ

....Khi `applyLfnstFlag` bằng 1 //`lfnst_idx` không bằng 0 và `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` bằng 0 và cả `nTbW` và `nTbH` lớn hơn hoặc bằng 4//, điều sau được áp dụng:...

Ở các câu lệnh được sửa đổi ở trên dựa vào tiêu chuẩn dự thảo, các ký tự chữ nằm giữa cặp dấu “//” biểu thị các ký tự chữ được xoá.

8.7.3 Phép phóng tỷ lệ cho các hệ số biến đổi

...Để suy dẫn các hệ số biến đổi được phóng tỷ lệ $d[x][y]$ với $x = 0..nTbW - 1$, $y = 0..nTbH - 1$, điều sau được áp dụng:

- Hệ số phóng tỷ lệ trung gian $m[x][y]$ được suy dẫn như sau:
- Nếu một hoặc nhiều trong số những điều kiện dưới đây là đúng, $m[x][y]$ được thiết lập bằng 16:

- `sps_scaling_list_enabled_flag` bằng 0.
- `pic_scaling_list_present_flag` bằng 0.
- `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` bằng 1.
- `scaling_matrix_for_lfnst_disabled_flag` bằng 1 và `applyLfnstFlag` bằng 1 //`lfnst_idx[ xTbY ][ yTbY ]` không bằng 0//....

Ở các câu lệnh được sửa đổi ở trên dựa vào tiêu chuẩn dự thảo, các ký tự chữ nằm giữa cặp dấu “//” biểu thị các ký tự chữ được xoá.

Theo phương án khác, khi LFNST chroma được huỷ bỏ ở một số trường hợp, `LfnstDcOnly`, mà được khởi tạo bằng 1 trước khi phân tích từng TB trong một CU và được chuyển thành 0 nếu bất kỳ TB trong CU đó có các hệ số có nghĩa (hoặc hệ số có nghĩa cuối) được nằm tại vị trí lớn hơn vị trí DC, không được cập nhật ở các TB không LFNST. Ví dụ, LFNST chroma được huỷ bỏ cho cây đơn. Thì khi đó, các TB không LFNST chứa các TB chroma đối với cây đơn. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện như sau.

Bảng 17. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

Residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) { ...	Biến mô tả
<pre> if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0 && ((cIdx == 0) (treeType != SINGLE_TREE))) LfntDcOnly = 0 </pre>	
<pre> if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight)) LfntZeroOutSigCoeffFlag = 0 </pre>	
...}	

Theo ví dụ khác, LFNST chroma được huỷ bỏ và các TB không LFNST chứa các TB chroma. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện như sau.

Bảng 18. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) { ...	Biến mô tả
<pre> if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0 && (cIdx = 0)) LfntDcOnly = 0 </pre>	
<pre> if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight)) LfntZeroOutSigCoeffFlag = 0 </pre>	
...}	

Khi TB không có cbf, không có phép biến đổi nào cần được áp dụng. Tuy nhiên, chỉ số LFNST có thể vẫn được báo hiệu/được phân tích trong trường hợp này. Ví dụ, đối với cây đơn, khi luma không chứa cbf, nhưng chroma thoả mãn điều kiện báo hiệu/phân tích LFNST (ví dụ chroma không là rẽ nhánh biến đổi và có các hệ số nằm tại vị trí không DC), chỉ số LFNST có thể được báo hiệu/được phân tích. Trong trường hợp này, chỉ số LFNST được báo hiệu/được phân tích và luôn có trị số bằng 0, vì LFNST

được áp dụng cho luma đối với cây đơn và chỉ số này là độ dư thừa.

Theo một phương án, LfnstDcOnlyFlag được cập nhật bởi Y chỉ (TB sẽ được áp dụng LFNST) đối với cây đơn.

Theo phương án khác, LfnstDcOnlyFlag được tách thành LfnstDcOnlyLumaFlag và LfnstDcOnlychromaFlag. LfnstDcOnlyLumaFlag được cập nhật bởi TB Y và LfnstDcOnlychromaFlag được cập nhật bởi TB (các TB) Cb hoặc Cr.

Theo một phương án phụ, đối với cây đơn, chỉ LfnstDcOnlyLumaFlag được xét cho báo hiệu/phân tích LFNST.

Theo phương án phụ khác, đối với cây lưỡng luma, chỉ LfnstDcOnlyLumaFlag được xét cho báo hiệu/phân tích LFNST.

Theo phương án phụ khác, đối với cây lưỡng chroma, chỉ LfnstDcOnlychromaFlag được xét cho báo hiệu/phân tích LFNST.

Theo phương án khác, phép kiểm tra được thêm cho báo hiệu LFNST như sau. Nếu luma không có Cbf, chỉ số LFNST không được báo hiệu/được phân tích. Ví dụ về bảng cú pháp được thể hiện như sau.

Bảng 19. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) </pre>
<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>
<pre> if (treeType == DUAL_TREE_CHROMA tu_cbf_luma[x0][y0]) lfnst_idx </pre>
<pre> } </pre>

Theo một phương án phụ, phép kiểm tra vừa nêu được thêm vào tại phân mức CU.

Theo phương án phụ khác, phép kiểm tra vừa nêu được thêm vào tại phân mức TB.

Theo phương án phụ khác, phép kiểm tra vừa nêu được thực hiện đối với cây đơn như được thể hiện ở ví dụ dưới đây.

Bảng 20. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) </pre>
<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>
<pre> if (treeType != SINGLE_TREE tu_cbf_luma[x0][y0]) lfnst_idx </pre>
<pre> } </pre>

Theo phương án phụ khác, phép kiểm tra vừa nêu không bị giới hạn đối với cơ chế ISP. Lý do được nêu như sau. Khi ISP được áp dụng, một CB luma được phân chia thành nhiều TB (ví dụ 4 TB) và ít nhất một hệ số có nghĩa có mặt bên trong TB (các TB chứa các hệ số có nghĩa có thể là bất kỳ một hoặc nhiều TB bên trong CU). Các ví dụ về bảng cú pháp được thể hiện như sau.

Bảng 21A. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) </pre>
<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1) </pre>
<pre> if (treeType != SINGLE_TREE (tu_cbf_luma[x0][y0] IntraSubPartitionsSplitType != ISP_NO_SPLIT)) lfnst_idx </pre>
<pre> } </pre>

Bảng 21B. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) </pre>
<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT (LfnstDcOnly == 0 && (treeType != SINGLE_TREE tu_cbf_luma[x0][y0])) && LfnstZeroOutSigCoeffFlag == 1) </pre>
<pre> lfnst_idx </pre>
<pre> } </pre>

Bảng 21C. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) </pre>
<pre> if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfnstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT (LfnstDcOnly == 0 && (!(treeType == SINGLE_TREE && tu_cbf_luma[x0][y0]))) && LfnstZeroOutSigCoeffFlag == 1) </pre>
<pre> lfnst_idx </pre>
<pre> } </pre>

Ở các bảng 20 và 21A-C ở trên, lfnstNotTsFlag là cờ, trong đó nếu lfnstNotTsFlag bằng 1, thì cơ chế LFNST có thể được áp dụng (giả thiết rằng các điều kiện kích hoạt khác của LFNST được thoả mãn). Nếu không thì, (tức là, lfnstNotTsFlag bằng 0), cơ chế LFNST không thể được áp dụng (tức là, cú pháp LFNST được suy diễn là huỷ bỏ).

Theo phương án phụ khác, phép kiểm tra vừa nêu được thực hiện cho cây lưỡng luma.

Theo một phương án, khi chỉ số LFNST > 0 (tức là, lfnst được thực hiện), LFNST được áp dụng cho thành phần xuất hiện đầu tiên ở cây phân chia. LFNST không được áp dụng cho các thành phần còn lại.

Theo một phương án phụ, thành phần xuất hiện đầu tiên là thành phần đầu tiên mà chứa cbf. Ví dụ, đối với cây lưỡng chroma, nếu cb chứa các hệ số có nghĩa, thì thành phần xuất hiện đầu tiên là Cb; nếu không thì nếu Cb cbf = 0 và Cr chứa các hệ số có nghĩa, thành phần xuất hiện đầu tiên là Cr; nếu không thì nếu Cb và Cr không chứa hệ số có nghĩa, lfnst không được áp dụng (tức là, chỉ số LFNST được suy diễn bằng 0 mà không báo hiệu). Theo ví dụ khác, lfnst được áp dụng cho luma, đối với cây đơn. Trong trường hợp này, nếu luma chứa các hệ số có nghĩa, thành phần xuất hiện đầu tiên là luma; nếu không thì, lfnst không được áp dụng.

Theo phương án khác, khi cơ chế JCCR được kích hoạt, một khối dư chroma chung ($\text{resJointC}[x][y]$ ở **Error! Reference source not found.**7) được báo hiệu/được phân tích, do đó lnst có thể được áp dụng cho khối đơn đó (tức là, LFNST tác động đến cả Cb và Cr).

Theo phương án khác, lnst được báo hiệu/được phân tích tại TB xuất hiện đầu tiên.

Theo một phương án phụ, TB xuất hiện đầu tiên là TB đầu tiên mà chứa cbf . Ví dụ, đối với cây lưỡng chroma, nếu cb chứa các hệ số có nghĩa, TB xuất hiện đầu tiên là Cb; nếu không thì nếu $\text{Cb cbf} = 0$ và Cr chứa các hệ số có nghĩa, TB xuất hiện đầu tiên là Cr; nếu không thì nếu Cb và Cr không chứa hệ số có nghĩa, lnst không được áp dụng (tức là, chỉ số LFNST được suy dẫn bằng 0 và không báo hiệu). Theo ví dụ khác, lnst được áp dụng cho luma. Đối với cây đơn, nếu luma chứa các hệ số có nghĩa, thành phần xuất hiện đầu tiên là luma; nếu không thì, lnst không được báo hiệu/được phân tích.

Theo phương án phụ khác, khi lập mã chung các phần dư màu (JCCR) được áp dụng, đối với cây lưỡng chroma, TB xuất hiện đầu tiên là TB chroma mà các phần dư của nó được sử dụng để suy dẫn các phần dư cho thành phần chroma khác. LFNST có thể được xem là được áp dụng cho toàn bộ các thành phần chroma. Ví dụ, đối với cây lưỡng chroma, nếu $\text{tu_cbf_cb}[x_0][y_0] > 0$ và $\text{tu_cbf_cr}[x_0][y_0] = 0$ (các phần dư của Cr được suy dẫn từ Cb và thông tin liên quan đến phần dư của Cb được báo hiệu), thì TB xuất hiện đầu tiên là Cb. Theo ví dụ khác, đối với cây lưỡng chroma, nếu $\text{tu_cbf_cb}[x_0][y_0] = 0$ và $\text{tu_cbf_cr}[x_0][y_0] > 0$ (các phần dư của Cb được suy dẫn từ Cr và thông tin liên quan đến phần dư của Cr được báo hiệu), thì TB xuất hiện đầu tiên là Cr. Theo ví dụ khác, đối với cây lưỡng chroma, nếu $\text{tu_cbf_cb}[x_0][y_0] > 0$ và $\text{tu_cbf_cr}[x_0][y_0] > 0$ (các phần dư của Cr được suy dẫn từ Cb và thông tin liên quan đến phần dư của Cb được báo hiệu), thì TB xuất hiện đầu tiên là Cb.

Theo phương án phụ khác, chỉ số LFNST được báo hiệu/được phân tích ở cuối TB xuất hiện đầu tiên. Ví dụ, chỉ số LFNST được phân tích sau khi phân tích các phần dư cho từng khối con (nhóm lập mã hoặc khối con 4×4) trong TB đó.

Theo phương án phụ khác, chỉ số LFNST được báo hiệu/được phân tích sau khi báo hiệu/phân tích cờ có nghĩa của TB đó. Các điều kiện báo hiệu/phân tích cho LFNST (ví dụ LfnstDcOnlyFlag hoặc $\text{LfnstZeroOutSigCoeffFlag}$) chỉ phụ thuộc vào thông tin của TB xuất hiện đầu tiên.

Theo phương án khác, đối với cây đơn, lfnst chỉ được áp dụng cho luma. Khi luma không thể áp dụng lfnst (ví dụ luma không chứa cbf), chỉ số LFNST không được báo hiệu/được phân tích ngay cả khi chroma xuất hiện cho báo hiệu/phân tích lfnst (ví dụ chroma chứa cbf hoặc JCCR được sử dụng cho chroma).

Khi TB không có cbf, không cần phép biến đổi nào. Tuy nhiên, cờ TS có thể bằng 1 trong trường hợp này. Ví dụ, đối với khối BDPCM, cờ rẽ nhánh biến đổi được xem là bằng 1. Tuy nhiên, khối BDPCM vừa nêu có thể không chứa cbf. Khi xét đến báo hiệu/phân tích LFNST, TB không có cbf được xem là khối TS và chỉ số LFNST không thể được báo hiệu/được phân tích vì LFNST không được chấp nhận cho khối TS. Một số phương pháp được đề xuất nhằm ngăn những trường hợp không mong muốn vừa nêu.

Theo một phương án, khi xét đến báo hiệu/phân tích chỉ số LFNST, chỉ báo cờ khối được lập mã (CBF) được dùng để ngăn sự không chấp nhận LFNST cho CU chứa một hoặc nhiều TB mà có một hoặc nhiều cờ TS không bằng 0 nhưng không được thực hiện hoàn toàn bằng phép TS. Trước tiên, chỉ báo cờ khối được lập mã (CBF) cho CU hiện thời được kiểm tra. Chỉ báo CBF của CU hiện thời được biểu diễn dưới dạng `cu_coded_flag` (hoặc `cu_cbf`).

`Cu_coded_flag` bằng 1 xác định rằng cấu trúc cú pháp `transform_tree()` xuất hiện cho đơn vị lập mã hiện thời. `Cu_coded_flag` bằng 0 xác định rằng cấu trúc cú pháp `transform_tree()` không xuất hiện cho đơn vị lập mã hiện thời.

Khi `cu_coded_flag` không xuất hiện, nó được suy diễn như sau:

Nếu `cu_skip_flag[x0][y0]` bằng 1 hoặc `pred_mode_plt_flag` bằng 1, `cu_coded_flag` được xem là bằng 0.

Nếu không thì, `cu_coded_flag` được xem là bằng 1.

Nếu chỉ báo CBF cho CU hiện thời là true, các phép kiểm tra một hoặc nhiều chỉ báo dưới đây cho một hoặc nhiều TB mong muốn là cần thiết; nếu không thì, cú pháp LFNST được suy diễn là huỷ bỏ.

Theo một phương án, khi xét đến báo hiệu/phân tích chỉ số LFNST, ngoài kiểm tra chỉ báo CBF cho CU hiện thời, `tu_cbf` cũng được kiểm tra. Vì vậy, đối với TB có cờ TS bằng 1, nếu TB này không chứa cbf, nó sẽ không được xem là khối TS cho báo hiệu/phân tích LFNST. `lfnstNotTsFlag` được cập nhật như sau:

```
lfnstNotTsFlag = ( treeType == DUAL_TREE_CHROMA ||
  (transform_skip_flag[x0][y0][0] == 0 || !tu_cbf_luma[x0][y0]) ) &&
```

```
( treeType == DUAL_TREE_LUMA ||
  ( (transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 || !tu_cbf_cb[x0][y0] ) &&
    (transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0 || !tu_cbf_cr[x0][y0]) ) )
```

Phép suy dẫn `lnstNotTsFlag` như được thể hiện ở trên là dựa vào ít nhất một trong hai yếu tố: chỉ báo cơ chế TS là false và chỉ báo CBF là false. Cơ chế LFNST được chấp nhận cho CU hiện thời nếu mỗi TB mong muốn thoả mãn ít nhất một trong số yếu tố 1 và yếu tố 2. Việc chấp nhận cơ chế LFNST cho CU hiện thời có nghĩa là nếu các điều kiện kích hoạt LFNST khác được thoả mãn, cú pháp LFNST được báo hiệu/được phân tích để biểu thị liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng. Chỉ báo cơ chế TS phụ thuộc vào cờ TS. Chỉ báo CBF phụ thuộc vào cờ khối được lập mã (`cbf`) cho TB mong muốn. `Cbf` cho Y, Cb, và Cr có thể được biểu diễn bằng `tu_y_coded_flag` (hoặc `tu_cbf_luma`), `tu_cb_coded_flag` (hoặc `tu_cbf_cb`) và `tu_cr_coded_flag` (hoặc `tu_cbf_cr`).

`Tu_cb_coded_flag[ x0 ][ y0 ]` bằng 1 xác định rằng khối biến đổi Cb chứa một hoặc nhiều mức hệ số biến đổi không bằng 0. Các chỉ số mảng `x0` và `y0` xác định vị trí (`x0`, `y0`) của mẫu luma trên cùng-bên trái của khối biến đổi được xét so với mẫu luma trên cùng-bên trái của ảnh.

Khi `tu_cb_coded_flag[ x0 ][ y0 ]` không xuất hiện, trị số của nó được xem là bằng 0.

`Tu_cr_coded_flag[ x0 ][ y0 ]` bằng 1 xác định rằng khối biến đổi Cr chứa một hoặc nhiều mức hệ số biến đổi không bằng 0. Các chỉ số mảng `x0`, `y0` xác định vị trí (`x0`, `y0`) của mẫu luma trên cùng-bên trái của khối biến đổi được xét so với mẫu luma trên cùng-bên trái của ảnh.

Khi `tu_cr_coded_flag[ x0 ][ y0 ]` không xuất hiện, trị số của nó được xem là bằng 0.

`Tu_y_coded_flag[ x0 ][ y0 ]` bằng 1 xác định rằng khối biến đổi luma chứa một hoặc nhiều mức hệ số biến đổi không bằng 0. Các chỉ số mảng `x0`, `y0` xác định vị trí (`x0`, `y0`) của mẫu luma trên cùng-bên trái của khối biến đổi được xét so với mẫu luma trên cùng-bên trái của ảnh.

Khi `tu_y_coded_flag[ x0 ][ y0 ]` không xuất hiện và `treeType` không bằng `DUAL_TREE_CHROMA`, trị số của nó được suy diễn như sau:

- Nếu `cu_sbt_flag` bằng 1 và một trong số những điều kiện dưới đây là true, `tu_y_coded_flag[ x0 ][ y0 ]` được xem là bằng 0:

- `subTuIndex` bằng 0 và `cu_sbt_pos_flag` bằng 1;
- `subTuIndex` bằng 1 và `cu_sbt_pos_flag` bằng 0.
- Nếu không thì, `tu_y_coded_flag[ x0 ][ y0 ]` được xem là bằng 1.

Ví dụ, ở cây phân chia luma:

- Nếu cờ TS cho TB luma mong muốn bằng 0, `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu TB luma mong muốn không chứa dữ liệu lập mã có nghĩa (các chỉ báo CBF là false), `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu không thì: `lnstNotTsFlag` được thiết lập bằng 0.

Theo ví dụ khác, ở cây phân chia chroma:

- Nếu các cờ TS cho toàn bộ các TB Cb và Cr mong muốn đều bằng 0, thì `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu “toàn bộ các TB mong muốn không chứa dữ liệu lập mã có nghĩa”, thì `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu “mỗi TS TB mong muốn (TS TB mong muốn = TB mong muốn có cờ TS không bằng 0) không chứa dữ liệu lập mã có nghĩa”, `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu không thì: `lnstNotTsFlag` được thiết lập bằng 0.

Theo ví dụ khác, ở cây phân chia đơn:

- Nếu các cờ TS cho toàn bộ các TB luma, Cb, và Cr mong muốn đều bằng 0, `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu “toàn bộ các TB mong muốn không chứa dữ liệu lập mã có nghĩa”, `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu “mỗi TS TB mong muốn (TS TB mong muốn = TB mong muốn có cờ TS không bằng 0) không chứa dữ liệu lập mã có nghĩa”, `lnstNotTsFlag` được thiết lập bằng 1.

- Nếu không thì, `lnstNotTsFlag` được thiết lập bằng 0.

Theo một phương án, các TB mong muốn tương ứng với một hoặc nhiều TB có các cờ TS không bằng 0.

Theo một phương án, ở cây phân chia luma, CU hiện thời tương ứng với khối lập

mã luma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma. Theo phương án khác, ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB chroma. Ví dụ, các khối lập mã chroma là các khối lập mã Cb và Cr, và các TB chroma là các TB Cb và Cr. Theo phương án khác nữa, ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma. Ví dụ, các khối lập mã chroma là các khối lập mã Cb và Cr, và các TB chroma là các TB Cb và Cr.

Theo một phương án, các TB mong muốn tương ứng với TB được định trước cho từng khối lập mã ở CU hiện thời. Ví dụ, TB được định trước tương ứng với TB đầu tiên cho từng khối lập mã ở CU hiện thời. Vị trí của TB đầu tiên có thể là vị trí của mẫu luma trên cùng-bên trái của khối biến đổi được xét. Vị trí của TB đầu tiên có thể là vị trí của mẫu luma trên cùng-bên trái của khối lập mã được xét (CB được xét ở CU hiện thời) so với mẫu luma trên cùng-bên trái của ảnh.

Theo một phương án, nếu toàn bộ các chỉ báo CBF cho các TB mong muốn là false, cơ chế LFNST được chấp nhận cho CU hiện thời (không phụ thuộc vào phép kiểm tra TS). Theo phương án khác, nếu toàn bộ các chỉ báo CBF cho các TB mong muốn có các cờ TS không bằng 0 là false, cơ chế LFNST được chấp nhận cho CU hiện thời. Theo phương án khác nữa, nếu bất kỳ một trong số các chỉ báo CBF cho các TB mong muốn có các cờ TS không bằng 0 là true, cơ chế LFNST không được chấp nhận cho CU hiện thời.

Theo phương án khác, khi khối không chứa cbf (cbf là false), cờ TS sẽ không bằng 1. Ví dụ, nếu khối BDPCM không có cbf, cờ TS của nó sẽ không được suy diễn bằng 1. Ví dụ của các ngôn ngữ được sửa đổi theo một phương án của sáng chế được thể hiện dưới đây.

Khi `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` không xuất hiện, nó được suy diễn như sau:

- Nếu `BdpcmFlag[ x0 ][ y0 ][ cIdx ]` bằng 1,
`transform_skip_flag[ x0 ][ y0 ][ cIdx ]` được suy diễn như sau
- Nếu `cIdx = 0` và `tu_cbf_luma[x0][y0] = 1`, `transform_skip_flag[ x0 ][ y0 ][ cIdx ]` được xem là bằng 1.

- Nếu không thì nếu $cIdx = 1$ và $tu_cbf_cb[x0][y0] = 1$, $transform_skip_flag[x0][y0][cIdx]$ được xem là bằng 1.
- Nếu không thì nếu $cIdx = 1$ và $tu_cbf_cr[x0][y0] = 1$, $transform_skip_flag[x0][y0][cIdx]$ được xem là bằng 1.
- Nếu không thì, $transform_skip_flag[x0][y0][cIdx]$ được xem là bằng 0.

Theo phương án khác, khi LFNST chroma được huỷ bỏ ở một số trường hợp, $LfnstZeroOutSigCoeffFlag$, mà được khởi tạo bằng 1 trước khi phân tích từng TB trong một CU và được chuyển thành 0 nếu bất kỳ TB trong CU đó có các hệ số có nghĩa (hoặc hệ số có nghĩa cuối) được nằm tại vùng LFNST bằng không, không được cập nhật ở các TB không LFNST. Ví dụ, LFNST chroma được huỷ bỏ đối với cây đơn. Thì khi đó, các TB không LFNST chứa các TB chroma cho cây đơn. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện như sau.

Bảng 22. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) { ...</code>	Biến mô tả
<code>if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)</code>	
<code>LfnstDcOnly = 0</code>	
<code>if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight) && ((cIdx == 0) (treeType != SINGLE_TREE)))</code>	
<code>LfnstZeroOutSigCoeffFlag = 0</code>	
<code>...}</code>	

Theo ví dụ khác, LFNST chroma được huỷ bỏ và các TB không LFNST chứa các TB chroma. Ví dụ về những thay đổi tương ứng ở bảng cú pháp được thể hiện như sau.

Bảng 23. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<code>Residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) { ...</code>	Biến mô tả
---	------------

if(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfnstDcOnly = 0	
if((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight) && (cIdx = 0))	
LfnstZeroOutSigCoeffFlag = 0	
...}	

Dựa vào Bảng 15, là bảng dùng để chỉ kiểm tra transform_skip_flag luma khi lập mã các TU luma (ví dụ không là DUAL_TREE_CHROMA) và chỉ kiểm tra transform_skip_flag chroma khi lập mã các TU chroma (ví dụ không là DUAL_TREE_LUMA), LFNST chroma được huỷ bỏ theo một số trường hợp. Ví dụ, đối với cây đơn, LFNST chroma được huỷ bỏ. Ví dụ về bảng cú pháp được đề xuất được thể hiện như sau.

Bảng 24. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == DUAL_TREE_CHROMA? (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0) : (transform_skip_flag[x0][y0][0] == 0)) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)	
lfnst_idx	ae(v)
}	

Ví dụ khác của bảng cú pháp được đề xuất dựa vào Bảng 16 còn được thể hiện như sau.

Bảng 25. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType == DUAL_TREE_CHROMA transform_skip_flag[x0][y0][0] == 0) && (treeType != DUAL_TREE_CHROMA (transform_skip_flag[x0][y0][1] == 0 && transform_skip_flag[x0][y0][2] == 0)) && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfstWidth, lfstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfstDcOnly == 0) && LfstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfst_idx </pre>	ae(v)
<pre> } </pre>	

Theo ví dụ khác, LFNST chroma được huỷ bỏ. Bảng cú pháp được đề xuất được thể hiện như sau. Ví dụ của bảng cú pháp được đề xuất được thể hiện như sau.

Bảng 26. Bảng cú pháp mẫu về báo hiệu/phân tích LFNST theo một phương án của sáng chế.

<pre> if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (treeType != DUAL_TREE_CHROMA && (transform_skip_flag[x0][y0][0] == 0)) && (treeType != DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfstWidth, lfstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfstDcOnly == 0) && LfstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfst_idx </pre>	ae(v)
<pre> } </pre>	

Theo VVC phiên bản 7, các chỉ số LFNST và MTS được lập mã ở cuối của CU, điều này gây ra các vấn đề về độ trễ và bộ nhớ đệm mà bộ giải mã cần để nhớ đệm tất cả các hệ số của toàn bộ ba thành phần màu trước khi nhận các chỉ số LFNST và MTS. Để giảm các vấn đề về độ trễ và bộ nhớ đệm, đã có đề xuất bỏ đi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS (ví dụ 0 là huỷ bỏ, 1 và 2 là kích hoạt sử dụng. 1 và 2 nghĩa là sử dụng phép biến đổi chính khác nhau hoặc ma trận LFNST khác nhau) ở cuối TB đầu tiên trong CU hoặc ở cuối của một hoặc nhiều TB của thành phần màu đầu tiên, hoặc ở

cuối TB khác không đầu tiên (và TB không rẽ nhánh biến đổi) ở CU hoặc ở cuối của một hoặc nhiều TB của thành phần màu đầu tiên (và TB không rẽ nhánh biến đổi).

Theo phương án khác, đã có đề xuất bỏ đi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS ở cuối TB khác không đầu tiên (và TB không rẽ nhánh biến đổi) ở CU hoặc ở cuối của một hoặc nhiều TB của thành phần màu đầu tiên (và TB không rẽ nhánh biến đổi). Theo một ví dụ, nó có thể được áp dụng chỉ cho cây đơn. Ở cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau TB luma (hoặc trước các TB chroma). Nếu cây đơn và ISP được áp dụng, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau TB luma cuối (hoặc trước các TB chroma). Ví dụ, subTuIndex có thể được sử dụng. Khi subTuIndex bằng NumIntraSubPartitions – 1, TB hiện thời là TB luma, và dạng cây hiện thời là cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích (nếu một hoặc nhiều điều kiện được thỏa mãn).

Theo phương án khác, đã có đề xuất bỏ đi cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS ở cuối của một hoặc nhiều TB luma ở CU (hoặc trước các TB chroma) ở trường hợp cây đơn; trong khi ở cây lưỡng luma, chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích ở cuối của một hoặc nhiều TB luma ở CU (hoặc ở cuối CU); trong khi ở cây lưỡng chroma, chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích sau phần cuối của các TB Cr trong CU (hoặc ở cuối CU). Nếu cây đơn và ISP được áp dụng, các chỉ số LFNST và MTS được báo hiệu/được phân tích sau TB luma cuối (hoặc trước các TB chroma). Ví dụ, subTuIndex có thể được sử dụng. Khi subTuIndex bằng NumIntraSubPartitions – 1, TB hiện thời là TB luma và dạng cây hiện thời là cây đơn, các chỉ số LFNST và MTS được báo hiệu/được phân tích (nếu một hoặc nhiều điều kiện được thỏa mãn).

Theo phương án khác, cờ kích hoạt hoặc chỉ số LFNST và/hoặc MTS được báo hiệu/được phân tích tại TB đầu tiên (ví dụ ở cuối TB đầu tiên) khi cơ chế ISP được sử dụng. Phương pháp được đề xuất có thể chỉ được áp dụng cho cây đơn (ví dụ vẫn báo hiệu/phân tích chỉ số MTS/LFNST ở cuối CU trong cây lưỡng luma hoặc cây lưỡng chroma).

Ở phương pháp đã nêu ở trên, chỉ số MTS có thể được báo hiệu/được phân tích sau chỉ số LFNST. Nếu LFNST được sử dụng (ví dụ chỉ số LFNST không bằng 0), chỉ số MTS được xem là bằng 0. Theo phương án khác, LFNST có thể được báo hiệu/được phân tích sau khi báo hiệu/phân tích chỉ số MTS. Nếu MTS được sử dụng (ví dụ chỉ số

MTS không bằng 0), chỉ số LFNST được xem là bằng 0.

Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được kết hợp.

Bất kỳ biến thể kết hợp của các phương pháp nêu trên có thể được quyết định ảnh dựa vào chiều rộng khối hoặc chiều cao khối hoặc diện tích khối, hoặc được quyết định hiện bằng cờ được báo hiệu/được phân tích tại phân mức CU, CTU, mảng ảnh, lát ảnh, nhóm lát ảnh, SPS, PPS, hoặc phân mức ảnh. “Khối” ở sáng chế này có thể có nghĩa là TU/TB/CU/CB/PU/PB.

Bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện ở các bộ mã hoá và/hoặc các bộ giải mã. Ví dụ, bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện ở môđun lập mã liên ảnh/nội ảnh/biến đổi của bộ mã hoá, môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã. Ngoài ra, bất kỳ trong số các phương pháp được đề xuất ở trên có thể được thực hiện dưới dạng mạch được liên kết với môđun lập mã liên ảnh/nội ảnh/biến đổi của bộ mã hoá và/hoặc môđun bù chuyển động, môđun suy dẫn ứng viên merge của bộ giải mã.

Quá trình giải mã mà tích hợp phương án của sáng chế có thể được hiểu dựa vào phần mô tả được nêu ở trên. Đối với quá trình giải mã cho hệ thống tích hợp LFNST, dữ liệu đầu vào là dữ liệu lập mã mà chứa CU đang được giải mã. Quá trình giải mã sau đó sẽ kiểm tra các chỉ báo CBF dựa vào dữ liệu lập mã. Cú pháp LFNST được phân tích theo kết quả kiểm tra. CU sau đó được giải mã theo cú pháp LFNST. Fig. 3 minh hoạ lưu đồ của phép phân tích hệ thống giải mã được lấy làm ví dụ có tích hợp LFNST (biến đổi không phân tách tần số thấp) theo phương án của sáng chế. Các bước được thể hiện trên lưu đồ có thể được thực hiện dưới dạng các mã chương trình có thể xử lý bằng một hoặc nhiều bộ xử lý (ví dụ, một hoặc nhiều CPU) tại bộ mã hoá. Các bước được thể hiện trên lưu đồ có thể còn được thực hiện dựa vào phần cứng chẳng hạn một hoặc nhiều thiết bị điện tử hoặc mạch xử lý được bố trí để thực hiện các bước trên lưu đồ.

Như được thể hiện trên Fig. 3, dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời được nhận ở bước 310, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu lập mã của CU hiện thời. Một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều TB mong muốn được kiểm tra dựa vào dữ liệu lập mã ở bước 320. Cú pháp LFNST được phân tích nếu một hoặc nhiều điều kiện được thoả mãn ở bước 330, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc

loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều TB mong muốn là false. CU hiện thời được giải mã theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST ở bước 340.

Quá trình mã hoá mà tích hợp phương án của sáng chế có thể được hiểu dựa vào phần mô tả được nêu ở trên. Đối với quá trình mã hoá cho hệ thống tích hợp LFNST, dữ liệu đầu vào cho LFNST là dữ liệu được biến đổi chính. Quá trình mã hoá sau đó áp dụng LFNST dựa vào nhân LFNST để suy dẫn dữ liệu đầu ra tạm thời. Ví dụ, nếu phép kiểm tra đạt, cú pháp LFNST được báo hiệu. Theo ví dụ khác, nếu phép kiểm tra không đạt (và kiểm tra TS không đạt), cú pháp LFNST được thiết đặt bằng 0. Cú pháp LFNST được xác định và được báo hiệu theo kết quả kiểm tra. CU hiện thời sau đó được mã hoá theo cú pháp LFNST được xác định.

Fig. 4 minh hoạ lưu đồ của phép phân tích hệ thống mã hoá được lấy làm ví dụ có tích hợp LFNST (biến đổi không phân tách tần số thấp) theo phương án của sáng chế. Như được thể hiện trên Fig. 4, dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời được nhận ở bước 410, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu được biến đổi chính. Phép LFNST được áp dụng dựa vào nhân LFNST để suy dẫn dữ liệu đầu ra tạm thời ở bước 420. Cú pháp LFNST được xác định và được báo hiệu nếu một hoặc nhiều điều kiện được thoả mãn ở bước 430, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều khối biến đổi (nhiều TB) mong muốn là false. CU hiện thời được mã hoá theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST được xác định ở bước 440.

Lưu đồ được thể hiện là nhằm minh hoạ ví dụ về hệ thống mã hoá/giải mã theo sáng chế. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này có thể sửa đổi từng bước, sắp xếp lại các bước, phân chia bước, hoặc kết hợp các bước để thực hiện sáng chế mà không trệch khỏi nội dung kỹ thuật của sáng chế. Ở bản mô tả này, cú pháp và các ngôn ngữ cụ thể được sử dụng để minh hoạ các ví dụ thực hiện các phương án của sáng chế. Người có hiểu biết trung bình có thể thực hiện sáng chế bằng cách thay thế cú pháp và các ngôn ngữ vừa nêu bằng cú pháp và các ngôn ngữ tương đương mà không trệch khỏi nội dung kỹ thuật của sáng chế.

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

Phương án của sáng chế như được mô tả ở trên có thể được thực hiện theo nhiều lập trình phần cứng, phần mềm khác nhau hoặc kết hợp cả phần cứng và phần mềm. Ví dụ, phương án của sáng chế có thể là một hoặc nhiều vi mạch điện tử được tích hợp vào chip nén video hoặc mã chương trình được tích hợp vào phần mềm nén video để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Phương án của sáng chế có thể còn là mã chương trình được xử lý trên bộ xử lý tín hiệu số DSP (Digital Signal Processor - DSP) để thực hiện quá trình xử lý đã được mô tả ở bản mô tả này. Sáng chế có thể còn liên quan đến số lượng chức năng được thực hiện bởi bộ xử lý máy tính, bộ xử lý tín hiệu số, bộ vi xử lý, hoặc vi mạch tích hợp FPGA (Field Programmable Gate Array-FPGA). Những bộ xử lý này có thể được cấu hình để thực hiện những nhiệm vụ cụ thể theo sáng chế, bằng cách xử lý mã phần mềm hoặc mã vi chương trình có thể đọc được bằng máy mà định ra các phương pháp thực hiện của sáng chế. Mã phần mềm hoặc mã vi chương trình có thể được phát triển theo các ngôn ngữ lập trình khác và các định dạng hoặc cách thức khác. Mã phần mềm có thể còn được tương thích với nhiều nền tảng khác nhau. Tuy nhiên, các định dạng, cách thức và ngôn ngữ mã khác nhau của các mã chương trình và các phương thức cấu hình mã khác để thực hiện những nhiệm vụ theo sáng chế sẽ không trệch khỏi nội dung và phạm vi của sáng chế. Phương án của sáng chế như được mô tả ở trên có thể được thực hiện ở bộ mã hoá video và bộ giải mã video. Các thành phần của bộ mã hoá video và bộ giải mã video có thể được xử lý bằng các phần cứng, một hoặc nhiều bộ xử lý được cấu hình để xử lý các lệnh chương trình được lưu ở bộ nhớ, hoặc kết hợp giữa phần cứng và bộ xử lý. Ví dụ, bộ xử lý xử lý các lệnh

chương trình để điều khiển việc nhận dữ liệu đầu vào liên quan đến chuỗi video chứa khôi hiện thời ở ảnh hiện thời. Bộ xử lý được trang bị một hoặc nhiều nhân xử lý. Ở một số ví dụ, bộ xử lý xử lý các lệnh chương trình để thực hiện các chức năng ở một số bộ phận bộ mã hoá và bộ giải mã, và bộ nhớ được kết nối điện với bộ xử lý được sử dụng để lưu các lệnh chương trình, thông tin tương ứng với các ảnh được khôi phục của các khối, và/hoặc dữ liệu trung gian trong quá trình mã hoá hoặc giải mã. Bộ nhớ theo một số phương án bao gồm phương tiện có thể đọc bằng máy tính không chuyển tiếp, chẳng hạn bộ nhớ bán dẫn hoặc bộ nhớ trạng thái rắn, bộ nhớ truy cập ngẫu nhiên (RAM), bộ nhớ chỉ đọc (ROM), ổ cứng, ổ quang, hoặc phương tiện lưu thích hợp khác. Bộ nhớ vừa nêu có thể còn là kết hợp giữa hai hoặc nhiều hơn hai trong số các phương tiện có thể đọc bằng máy tính không chuyển tiếp vừa nêu ở trên.

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

YÊU CẦU BẢO HỘ

1. Phương pháp giải mã chuỗi video, trong đó cơ chế biến đổi không phân tách tần số thấp (LFNST) được hỗ trợ, phương pháp này bao gồm:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu lập mã của CU hiện thời;

kiểm tra một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều TB mong muốn dựa vào dữ liệu lập mã;

phân tích cú pháp LFNST khi một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo CBF cho một hoặc nhiều TB mong muốn là false; và

giải mã CU hiện thời theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST.

2. Phương pháp theo điểm 1, trong đó một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB có các cờ TS không bằng 0.

3. Phương pháp theo điểm 1, trong đó, ở cây phân chia luma, CU hiện thời tương ứng với khối lập mã luma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma.

4. Phương pháp theo điểm 1, trong đó, ở cây phân chia chroma, CU hiện thời tương ứng với một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB chroma.

5. Phương pháp theo điểm 1, ở cây phân chia đơn, CU hiện thời tương ứng với một khối lập mã luma và một hoặc nhiều khối lập mã chroma, và một hoặc nhiều TB mong muốn tương ứng với một hoặc nhiều TB luma và một hoặc nhiều TB chroma.

6. Phương pháp theo điểm 1, trong đó một hoặc nhiều TB mong muốn tương ứng với TB được định trước cho từng khối lập mã ở CU hiện thời.

7. Phương pháp theo điểm 6, trong đó TB được định trước tương ứng với TB đầu tiên cho từng khối lập mã ở CU hiện thời.

8. Phương pháp theo điểm 1, trong đó nếu toàn bộ một hoặc nhiều chỉ báo CBF cho một hoặc nhiều TB mong muốn là false, cơ chế LFNST được chấp nhận cho CU hiện thời.

9. Phương pháp theo điểm 1, trong đó nếu toàn bộ một hoặc nhiều chỉ báo CBF cho một

hoặc nhiều TB mong muốn là false và một hoặc nhiều TB mong muốn biểu thị một hoặc nhiều TB có các cờ TS không bằng 0, cơ chế LFNST được chấp nhận cho CU hiện thời.

10. Phương pháp theo điểm 1, trong đó nếu bất kỳ một trong số một hoặc nhiều chỉ báo CBF cho một hoặc nhiều TB mong muốn có các cờ TS không bằng 0 là true, cơ chế LFNST không được chấp nhận cho CU hiện thời.

11. Thiết bị giải mã chuỗi video, trong đó cơ chế biến đổi không phân tách tần số thấp (LFNST) được hỗ trợ, thiết bị này bao gồm một hoặc nhiều mạch điện tử hoặc bộ xử lý được bố trí để:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu lập mã của CU hiện thời;

kiểm tra một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều TB mong muốn dựa vào dữ liệu lập mã;

phân tích cú pháp LFNST khi một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo CBF cho một hoặc nhiều TB mong muốn là false; và

giải mã CU hiện thời theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST.

12. Phương pháp mã hoá chuỗi video, trong đó cơ chế biến đổi không phân tách tần số thấp (LFNST) được hỗ trợ, phương pháp này bao gồm:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu được biến đổi chính;

áp dụng phép LFNST dựa vào nhân LFNST để suy dẫn dữ liệu đầu ra tạm thời;

xác định và báo hiệu cú pháp LFNST khi một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều khối biến đổi (nhiều TB) mong muốn là false; và

mã hoá CU hiện thời theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST được xác định.

13. Thiết bị mã hoá chuỗi video, trong đó cơ chế biến đổi không phân tách tần số thấp (LFNST) được hỗ trợ, thiết bị này bao gồm một hoặc nhiều mạch điện tử hoặc bộ xử lý được bố trí để:

nhận dữ liệu đầu vào liên quan đến đơn vị lập mã (CU) hiện thời ở ảnh hiện thời, trong đó CU hiện thời được phân chia thành một hoặc nhiều khối biến đổi (nhiều TB) và dữ liệu đầu vào tương ứng với dữ liệu được biến đổi chính;

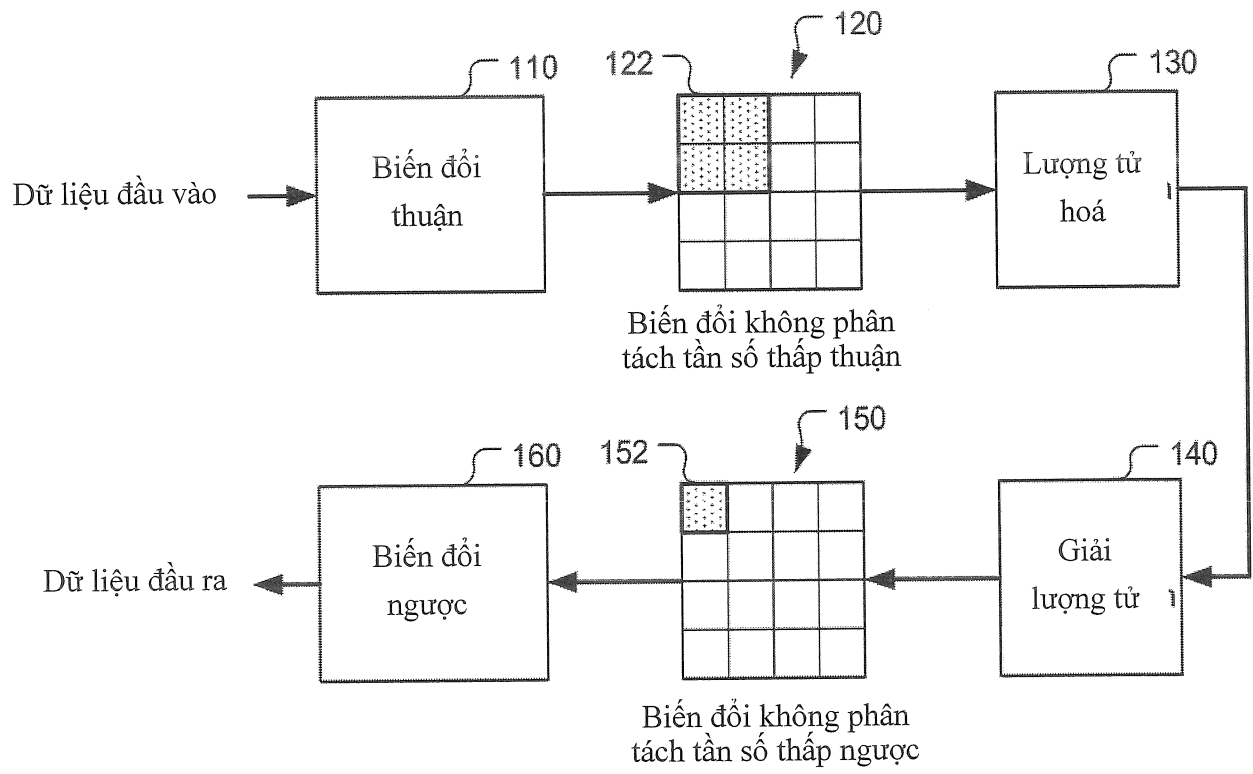
áp dụng phép LFNST dựa vào nhân LFNST để suy dẫn dữ liệu đầu ra tạm thời;

xác định và báo hiệu cú pháp LFNST khi một hoặc nhiều điều kiện được thoả mãn, trong đó cú pháp LFNST cho biết liệu cơ chế LFNST được áp dụng cho CU hiện thời và/hoặc loại nhân LFNST nào được áp dụng khi cơ chế LFNST được áp dụng, và một hoặc nhiều điều kiện vừa nêu bao gồm một hoặc nhiều chỉ báo cờ khối được lập mã (CBF) cho một hoặc nhiều khối biến đổi (nhiều TB) mong muốn là false; và

mã hoá CU hiện thời theo cơ chế LFNST như được biểu thị bằng cú pháp LFNST được xác định.

1/4

Fig. 1



2/4

Fig. 2A

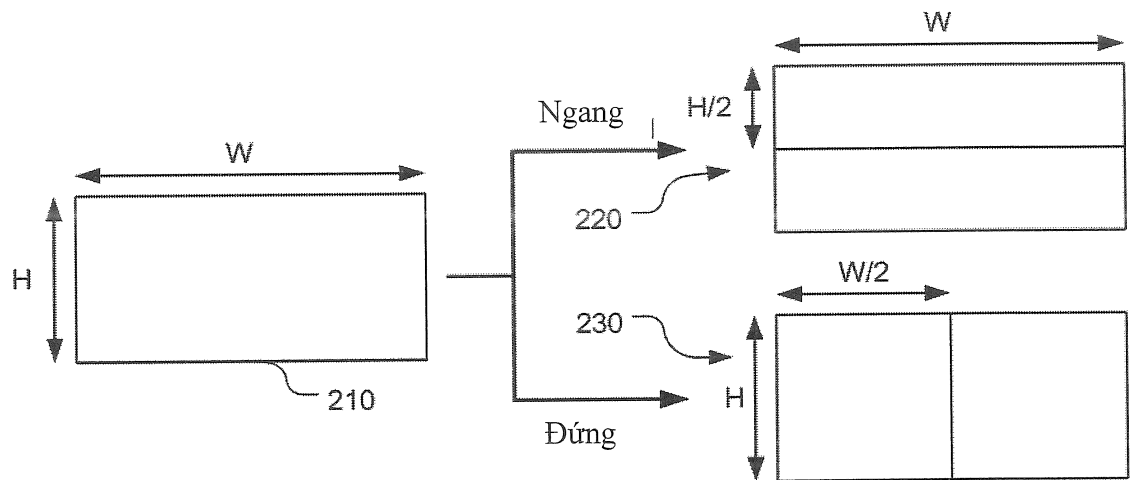
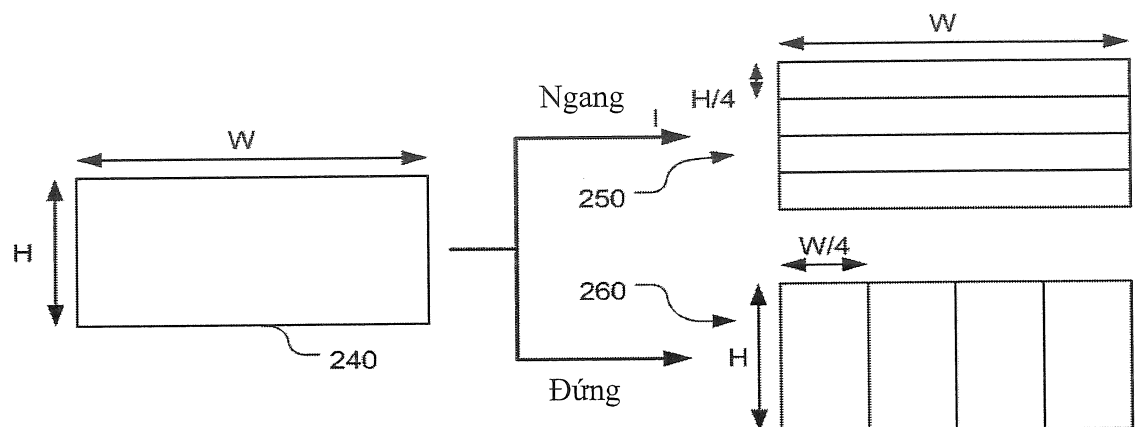
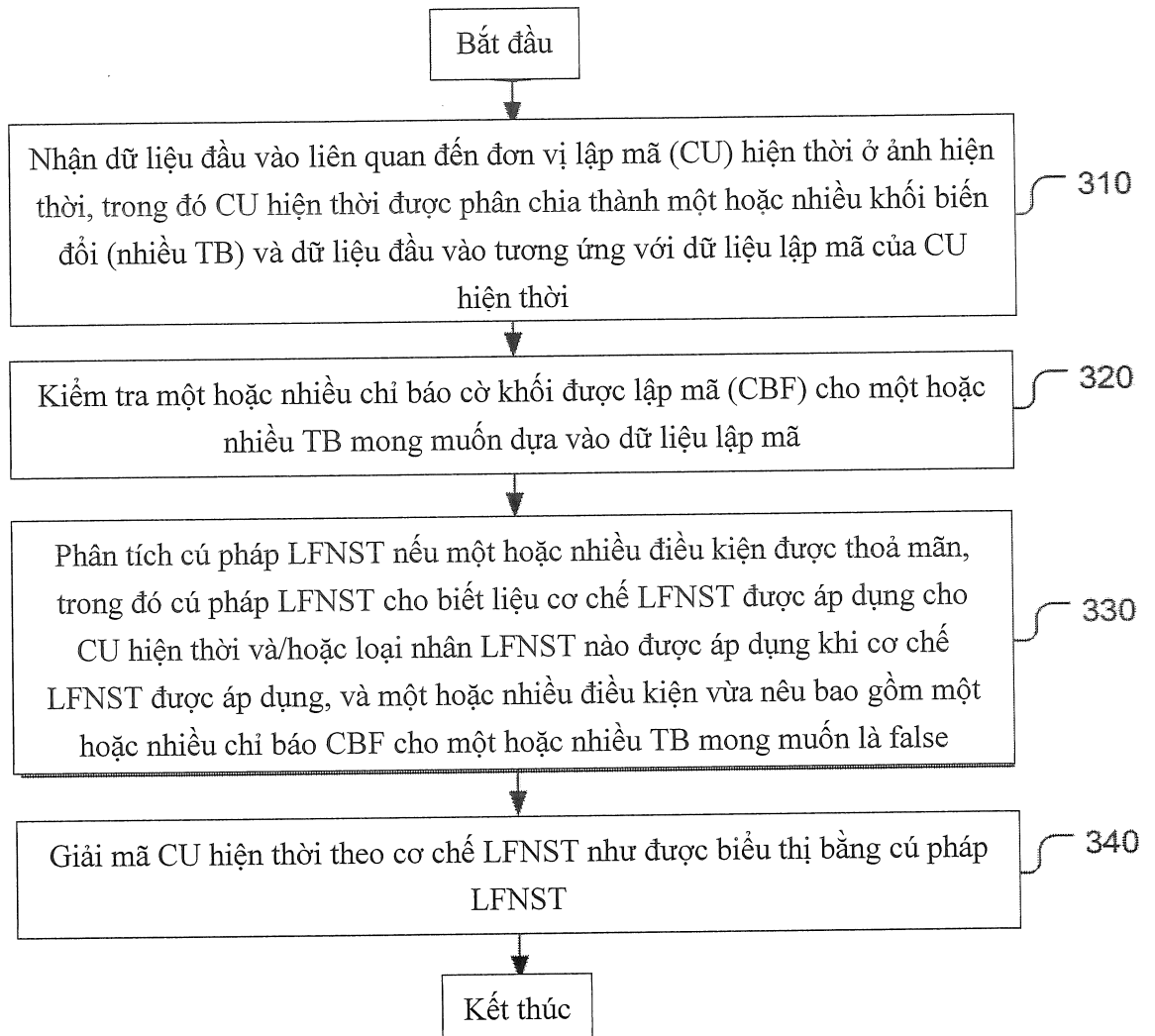


Fig. 2B



3/4

Fig. 3



4/4

Fig. 4