



УКРАЇНА

(19) **UA** (11) **129651** (13) **C2**  
(51) МПК  
**H04N 19/593** (2014.01)  
**H04N 19/159** (2014.01)

НАЦІОНАЛЬНИЙ ОРГАН  
ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ  
ДЕРЖАВНА ОРГАНІЗАЦІЯ  
"УКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ  
ОФІС ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ ТА ІННОВАЦІЙ"

**(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД**

<p>(21) Номер заявки: <b>а 2021 05884</b></p> <p>(22) Дата подання заявки: <b>23.03.2020</b></p> <p>(24) Дата, з якої є чинними права інтелектуальної власності: <b>26.06.2025</b></p> <p>(31) Номер попередньої заявки відповідно до Паризької конвенції: <b>62/822,865, 62/824,282, 62/824,360</b></p> <p>(32) Дата подання попередньої заявки відповідно до Паризької конвенції: <b>23.03.2019, 26.03.2019, 27.03.2019</b></p> <p>(33) Код держави-учасниці Паризької конвенції, до якої подано попередню заявку: <b>US, US, US</b></p> <p>(41) Публікація відомостей про заявку: <b>26.01.2022, Бюл.№ 4</b></p> <p>(46) Публікація відомостей про державну реєстрацію: <b>25.06.2025, Бюл.№ 26</b></p> <p>(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ: <b>PCT/CN2020/080674, 23.03.2020</b></p>	<p>(72) Винахідник(и): <b>Ван Бяо (DE), Есенлік Семіг (DE), Котра Ананд Меер (DE), Гао Хань (DE), Чен Цзянле (US)</b></p> <p>(73) Володілець (володільці): <b>ХУАВЕЙ ТЕКНОЛОДЖИЗ КО., ЛТД., Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong 518129, China (CN)</b></p> <p>(74) Представник: <b>Бочаров Максим Анатолійович, реєстр. №367</b></p> <p>(56) Перелік документів, взятих до уваги експертизою: WO 2018232676 A1, 27.12.2018 EP 2916545 A2, 09.09.2015 CN 102685474 A, 19.09.2012 WO 2013000324 A1, 03.01.2013 WO 2018074812 A1, 26.04.2018 CN 105357535 A, 24.02.2016 US 2016373782 A1, 22.12.2016 US 2017332084 A1, 16.11.2017 WANG (HUAWEI) B et al. "CE3-3.1.1: Unified MPM list generation", 14. JVET MEETING, 17.03.2019, GENEVA, (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16, XP030255224 [знайдено 2023-09-20] BOSSON (SHARPLABS) F ET AL: "CE3-3.4-related: unified MPM list construction", 126. MPEG MEETING, 20.03.2019, GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), XP030210739 [знайдено 2023-09-20] WANG (HUAWEI) B et al. "CE3-related: A unified MPM list for intra mode coding", 14 JVET MEETING, 04.04.2019, GENEVA, THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16, XP030205166 [знайдено 2023-09-20]</p>
---	---

**(54) КОДЕР, ДЕКОДЕР І ВІДПОВІДНІ СПОСОБИ ІНТРАПРОГНОЗУВАННЯ**

**(57) Реферат:**

Запропоновані пристрої і способи для інтра-прогнозування. Спосіб включає етапи, на яких: отримують значення інформації вказівки поточного блока. Коли значення інформації вказівки

UA 129651 C2

вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш імовірних режимів, виводять режим інтра-прогнозування  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  поточного блока за допомогою таких впорядкованих етапів: i  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  встановлюють таким, що дорівнює  $\text{intra\_luma\_trm\_remainder}[x_{Cb}][y_{Cb}]$ , ii значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  збільшують на одиницю. Цей спосіб може підвищити ефективність визначення режиму інтра-прогнозування поточного блока.

Перехресне посилання на споріднені заявки

Дана заявка заявляє пріоритет попередньої заявки [США № 62/822865, поданої 23 березня 2019 р., попередньої заявки США № 62/824282, поданої 26 березня 2019 р., і попередньої заявки США № 62/824360, поданої 27 березня 2019 р.], всі з яких включені в даний документ за посиланням.

Галузь техніки

Варіанти здійснення даної заявки (розкриття) загалом стосуються галузі обробки зображень і, зокрема, інтра-(intra)-прогнозування.

Рівень техніки

Відеокодування (кодування і декодування відео) використовується в широкому діапазоні застосувань цифрового відео, наприклад, в широкомовному цифровому телебаченні, передачі відео через Інтернет і мобільні мережі, в діалогових застосунках реального часу, таких як відеочат, відеоконференцзв'язок, DVD і Blu-Ray диски, системи збору і редагування відеоконтенту і камкодери програм безпеки.

Об'єм відеоданих, необхідних для надання навіть відносно короткого відео, може бути значним, що може спричинити труднощі, коли ці дані необхідно передавати у потоковому режимі або у інший спосіб передавати через мережу зв'язку з обмеженою пропускну здатністю. Отже, відеодані, як правило, стискають перед тим, як передавати через сучасні телекомунікаційні мережі. Розмір відео також може бути проблемою, коли відео зберігають на запам'ятовувальній пристрої, оскільки ресурси пам'яті можуть бути обмежені. Пристрої стиснення відео часто використовують програмне забезпечення і/або апаратні засоби в джерелі для кодування відеоданих перед передачею або зберіганням, тим самим зменшуючи кількість даних, необхідних для представлення цифрових відеозображень. Стиснені дані потім приймають у пункті призначення пристроєм декомпресії відео, який декодує відеодані. З обмеженими мережевими ресурсами і постійно зростаючими вимогами до більш високої якості відео бажаними є покращені методи стиснення і декомпресії, які покращують ступінь стиснення з мінімальними втратами якості зображення або взагалі без таких втрат.

Зокрема, існує постійна потреба у зниженні навантаження на процесор в контексті обробки інтра-прогнозування. У даній галузі техніки відоме використання списку найбільш імовірних режимів (Most Probable Mode, MPM) для кодування режиму інтра-прогнозування. Список MPM зменшує кількість бітів, необхідних для кодування режиму інтра-прогнозування поточного блока. Коли режим інтра-прогнозування поточного блока відповідає запису в списку MPM, кодується індекс, а не фактичний режим, внаслідок чого потрібно менше бітів. Однак, процес виведення режиму інтра-прогнозування впливає на ефективність кодування режиму інтра-прогнозування, коли режим інтра-прогнозування не міститься в списку MPM.

У зв'язку з цим задача даного винаходу полягає у наданні способу отримання режиму інтра-прогнозування, який сприяє більш ефективному кодуванню порівняно з рівнем техніки.

Суть винаходу

Згадані вище і інші цілі досягаються заявленими об'єктами незалежних пунктів формули винаходу. Додаткові варіанти здійснення є очевидними із залежних пунктів формули винаходу, опису винаходу і креслень.

Згідно з першим аспектом винахід стосується способу, який здійснюють за допомогою пристрою декодування. Спосіб включає етапи, на яких: отримують значення інформації вказівки поточного блока, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш імовірних режимів. Коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш імовірних режимів, виводять режим інтра-прогнозування  $IntraPredModeY[xCb][yCb]$  поточного блока за допомогою таких впорядкованих етапів, на яких:

i.  $IntraPredModeY[xCb][yCb]$  встановлюють таким, що дорівнює  $intra\_luma\_mpm\_remainder[xCb][yCb]$ ,

ii. значення  $IntraPredModeY[xCb][yCb]$  збільшується на одиницю; де  $intra\_luma\_mpm\_remainder[xCb][yCb]$  є режимом інтра-прогнозування залишку, місцеположення яскравості (xCb, yCb) визначає верхній лівий дискретний відлік поточного блока щодо верхнього лівого дискретного відліку яскравості поточного зображення.

У можливого варіанті здійснення способу згідно з першим аспектом після етапу ii виведення режиму інтра-прогнозування  $IntraPredModeY[xCb][yCb]$  поточного блока додатково містить етап, на якому:

iii. Для i, що дорівнює від 0 до 4 включно, коли  $IntraPredModeY[xCb][yCb]$  є більшим або дорівнює  $sandModeList[i]$ , значення  $IntraPredModeY[xCb][yCb]$  збільшується на одиницю.

У можливого варіанті здійснення способу згідно з першим аспектом набір найбільш

імовірних режимів містить 5 можливих режимів інтра-прогнозування і планарний режим. Потенційно придатний режим інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш імовірних режимів отримують у такий спосіб:

коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0\dots3$  і, для кожного  $i, j = (i+1)\dots4$ , обидва значення міняються місцями (swap) так:

$(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .

Згідно з другим аспектом винахід стосується способу, який здійснюється пристроєм кодування. Спосіб включає виведення змінною (наприклад,  $\text{intra\_luma\_mpm\_remainder}$ ) з використанням режиму інтра-прогнозування  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  поточного блока за допомогою таких впорядкованих етапів, коли режим інтра-прогнозування  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  поточного блока не міститься в наборі найбільш імовірних режимів:

i. Для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  є більшим, ніж  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  зменшується на одиницю, де  $\text{candModeList}[i]$  є потенційно придатним режимом інтра-прогнозування в наборі найбільш імовірних режимів;

ii. значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  зменшується на одиницю;

де місцеположення яскравості ( $x\text{Cb}$ ,  $y\text{Cb}$ ) визначає верхній лівий дискретний відлік поточного блока щодо верхнього лівого дискретного відліку яскравості поточного зображення. Спосіб додатково включає етап, на якому відправляють потік бітів в декодер, причому потік бітів містить інформацію згаданої змінної.

У можливішому варіанті здійснення способу згідно з другим аспектом набір найбільш імовірних режимів містить 5 можливих режимів інтра-прогнозування і планарний режим. Потенційно придатний режим інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш імовірних режимів отримують у такий спосіб:

коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0\dots3$  і, для кожного  $i, j = (i+1)\dots4$ , обидва значення міняються місцями так:

$(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .

Спосіб згідно з першим аспектом винаходу може бути здійснений за допомогою пристрою декодування згідно з третім аспектом винаходу. Пристрій декодування містить блок отримання і блок виведення. При цьому блок отримання сконфігурований з можливістю отримувати значення інформації вказівки поточного блока, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш імовірних режимів. Блок виведення сконфігурований з можливістю виводити режим інтра-прогнозування  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  поточного блока за допомогою таких впорядкованих етапів, коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш імовірних режимів:

i.  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  встановлюється еквівалентним  $\text{intra\_luma\_mpm\_remainder}[x\text{Cb}][y\text{Cb}]$ ,

ii. значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  збільшується на одиницю;

де  $\text{intra\_luma\_mpm\_remainder}[x\text{Cb}][y\text{Cb}]$  є режимом інтра-прогнозування залишку, місцеположення яскравості ( $x\text{Cb}$ ,  $y\text{Cb}$ ) визначає верхній лівий дискретний відлік поточного блока щодо верхнього лівого дискретного відліку яскравості поточного зображення.

Додаткові ознаки і варіанти здійснення способу згідно з третім аспектом винаходу відповідають ознакам і варіантам здійснення пристрою згідно з першим аспектом винаходу.

Спосіб згідно з другим аспектом винаходу може бути здійснений за допомогою пристрою кодування згідно з четвертим аспектом винаходу. Пристрій кодування містить блок виведення і блок передачі. Блок виведення сконфігурований з можливістю виводити змінну (наприклад,  $\text{intra\_luma\_mpm\_remainder}$ ) з використанням режиму інтра-прогнозування  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  поточного блока за допомогою таких впорядкованих етапів, коли режим інтра-прогнозування  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  поточного блока не міститься в наборі найбільш імовірних режимів:

i. Для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  є більшим, ніж  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  зменшується на одиницю, де  $\text{candModeList}[i]$  є потенційно придатним режимом інтра-прогнозування в наборі найбільш імовірних режимів;

ii. значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  зменшується на одиницю;

Блок передачі сконфігурований з можливістю відправляти потік бітів у декодер, причому потік бітів містить інформацію згаданої змінної.

Додаткові ознаки і варіанти здійснення способу згідно з четвертим аспектом винаходу відповідають ознакам і варіантам здійснення пристрою згідно з другим аспектом винаходу.

Згідно з п'ятим аспектом винахід стосується пристрою для декодування відеопотоку, який містить процесор і пам'ять. У пам'яті зберігаються інструкції, які наказують процесору здійснювати спосіб згідно з першим аспектом.

5 Згідно з шостим аспектом винахід стосується пристрою для кодування відеопотоку, який містить процесор і пам'ять. У пам'яті зберігаються інструкції, які наказують процесору здійснювати спосіб згідно з другим аспектом.

Згідно з сьомим аспектом запропонований машинозчитуваний носій, на якому зберігаються інструкції, які при їхньому виконанні наказують одному або декільком процесорам кодувати відеодані. Інструкції наказують одному або декільком процесорам здійснювати спосіб згідно з першим або другим аспектом, або будь-яким можливим варіантом здійснення першого або другого аспекту.

Згідно з восьмим аспектом винахід стосується комп'ютерної програми, яка містить програмний код для здійснення способу згідно з першим або другим аспектом, або будь-яким можливим варіантом здійснення першого або другого аспекту при виконанні на комп'ютері.

15 Варіанти здійснення даного винаходу спрощують процес виведення режиму без MPM, коли планарний режим завжди розглядається в наборі найбільш імовірних режимів (або неявно, або явно). Оскільки планарний відповідає найменшому значенню режимів інтра-прогнозування, процес виведення для режиму без MPM може бути спрощений. Для етапу порівняння і приручування порівняння з планарним режимом може бути збережене, оскільки планарний завжди присутній в списку MPM, отже, початкове значення режиму інтра-прогнозування залишку може бути збільшене на одиницю безпосередньо, що відповідає етапу ii, описаному вище. Відповідно, варіанти здійснення даного винаходу пропонують метод для отримання режиму інтра-прогнозування, який сприяє більш ефективному кодуванню порівняно з рівнем техніки.

20 Більш детально один або більше варіантів здійснення викладені на доданих кресленнях і в наведеному нижче описі. Інші ознаки, задачі і переваги будуть очевидні з опису, креслень і формули винаходу.

25 Короткий опис креслень

У наведених нижче варіантах здійснення винаходи описані більш детально з посиланням на додані фігури і креслення, на яких:

30 на фіг. 1A представлена блок-схема, яка демонструє приклад системи відеокодування, виконаної з можливістю реалізації варіантів здійснення винаходу;

на фіг. 1B представлена блок-схема, яка демонструє інший приклад системи відеокодування, виконаної з можливістю реалізації варіантів здійснення винаходу;

35 на фіг. 2 представлена блок-схема, яка демонструє приклад відеокодера, виконаного з можливістю реалізації варіантів здійснення винаходу;

на фіг. 3 представлена блок-схема, яка демонструє приклад структури відеодекодера, виконаного з можливістю реалізації варіантів здійснення винаходу;

40 на фіг. 4 представлена блок-схема, яка ілюструє приклад пристрою кодування або пристрої декодування;

на фіг. 5 представлена блок-схема, яка ілюструє інший приклад пристрою кодування або пристрої декодування;

на фіг. 6 представлена блок-схема, яка ілюструє приклад інструмента кодування з множинними опорними рядками (MRL);

на фіг. 7 представлена блок-схема, яка ілюструє приклад розділення блоків  $4 \times 8$  і  $8 \times 4$ ;

45 на фіг. 8 представлена блок-схема, яка ілюструє приклад розділення всіх блоків, крім  $4 \times 8$ ,  $8 \times 4$  і  $4 \times 4$ ;

на фіг. 9 представлена блок-схема, яка ілюструє приклад контекстного моделювання/кодування CABAC;

50 на фіг. 10 представлена блок-схема, яка ілюструє приклад лівого (L) і верхнього (A) сусідніх блоків поточного блока;

на фіг. 11 представлена блок-схема, яка ілюструє режими інтра-прогнозування;

фіг. 12 ілюструє варіанти здійснення способу 1200 інтра-прогнозування поточного блока, який реалізовується пристроєм декодування згідно з даним винаходом;

55 фіг. 13 ілюструє варіанти здійснення способу 1300 інтра-прогнозування поточного блока, який реалізовується пристроєм кодування згідно з даним винаходом;

фіг. 14 ілюструє варіанти здійснення пристрою 1400 декодування для використання у декодері зображення згідно з даним винаходом;

фіг. 15 ілюструє варіанти здійснення пристрою 1500 кодування для використання у кодері зображення згідно з даним винаходом;

на фіг. 16 представлена блок-схема, яка демонструє зразкову структуру системи 3100 доставки контенту, яка реалізовує послугу доставки контенту; і

на фіг. 17 представлена блок-схема, яка демонструє структуру зразкового термінального пристрою.

5 У наведеному нижче описі однакові посилальний позиції стосуються ідентичних або щонайменше функціонально еквівалентних ознак, якщо явно не вказано інше.

Детальний опис варіантів здійснення винаходу

10 Наведений нижче опис посилається на додані креслення, які становлять частину опису і на яких показані як ілюстрація конкретні аспекти варіантів здійснення винаходу або конкретні аспекти, в яких варіанти здійснення даного винаходу можуть бути використані. Зрозуміло, що варіанти здійснення винаходу можуть бути використані і в інших аспектах і містити структурні або логічні зміни, не зображені на кресленнях. Тому наведений нижче детальний опис не слід сприймати в обмежувальному сенсі, а об'єм даного винаходу визначається доданою формулою винаходу.

15 Так, наприклад, повинно бути зрозуміло, що розкриття у зв'язку з описаним способом також може стосуватися і відповідного пристрою або системи, виконаної з можливістю здійснення способу, і навпаки. Якщо, наприклад, описаний один або множина конкретних етапів способу, відповідний пристрій може містити один або множину блоків, наприклад, функціональних блоків, для здійснення описаного одного або множини етапів способу (наприклад, один блок, який

20 здійснює один або множину етапів або множина блоків, кожний з яких здійснює один або декілька із множини етапів), навіть якщо такий один або декілька блоки явно не описані або не проілюстровані на кресленнях. З іншого боку, наприклад, якщо конкретний пристрій описаний на основі одного або множини блоків, наприклад, функціональних блоків, відповідний спосіб може включати один етап для реалізації функціональних можливостей одного або множини блоків

25 (наприклад, один етап реалізації функціональних можливостей одного або множини блоків або множина етапів, кожний з яких реалізує функціональні можливості одного або декількох із множини блоків), навіть якщо такий один або множина етапів явно не описані або не проілюстровані на кресленнях. Крім того, зрозуміло, що ознаки різних зразкових варіантів здійснення і/або аспектів, описаних в даному документі, можуть бути об'єднані один з одним,

30 якщо спеціально не вказано інше.

Відеокодування зазвичай стосується обробки послідовності зображень, які утворюють відео або відеопослідовність. Замість терміну "зображення" можна використовувати термін "кадр" або "знімок" як синоніми в галузі відеокодування. Відеокодування (або кодування в загальному значенні) містить дві частини: кодування відео і декодування відео. Кодування відео

35 виконується з боку джерела, зазвичай містить обробку (наприклад, за допомогою стиснення) вихідних відеозображень для зменшення об'єму даних, необхідних для представлення відеозображень (для більш ефективного зберігання і/або передачі). Декодування відео виконується з боку одержувача і зазвичай включає зворотну обробку порівняно з кодером для відновлення відеозображень. Варіанти здійснення, які стосуються "кодування" відеозображень

40 (або зображень загалом), слід розуміти як такі, що стосуються або "кодування", або "декодування" відеозображень або відповідних відеопослідовностей. Комбінацію кодувальної частини і декодувальної частини також називається CODEC (кодування і декодування).

У випадку кодування відео без втрат вихідні відеозображення можуть бути відновлені, тобто відновлені відеозображення мають ту саму якість, що і вихідні відеозображення (за умови

45 відсутності втрат при передаванні або інших втрат даних під час зберігання або передавання). У випадку кодування відео з втратами виконується подальше стиснення, наприклад, за допомогою квантування, щоб зменшити об'єм даних, що представляють відеозображення, які не можуть бути повністю відновлені в декодері, тобто якість відновлених відеозображень є нижчою або гіршою порівняно з якістю вихідних відеозображень.

50 Декілька стандартів відеокодування належать до групи "гібридних відеокодеків з втратами" (тобто поєднують просторове і часове прогнозування в галузі дискретних відліків і кодування з двовимірним перетворенням для застосування квантування в галузі перетворення). Кожне зображення відеопослідовності зазвичай розділяється на набір блоків, які не перекриваються, і кодування зазвичай виконується на рівні блоків. Іншими словами, в кодері відео зазвичай обробляється, тобто кодується, на рівні блока (відеоблока), наприклад, з використанням просторового (всередині зображення) (інтра-) прогнозування і/або часового (між зображеннями) (інтер-) прогнозування для генерування блока прогнозу, віднімаючи блок прогнозування від поточного блока (блок, який в даний момент обробляється/повинен бути оброблений) для отримання залишкового блока, перетворення залишкового блока і квантування залишкового

60 блока в галузі перетворення для зменшення об'єму даних, які повинні бути передані

(стиснення), у той час як в декодері зворотна обробка порівняно з кодером застосовується до кодованого або стисненого блока, щоб відновити поточний блок для представлення. Крім того, кодер дублює цикл обробки декодера, так що обидва будуть генерувати ідентичні прогнози (наприклад, інтра- і інтер-прогнозування) і/або реконструкцію для обробки, тобто кодування, подальших блоків.

У наведених нижче варіантах здійснення системи 10 відеокодування кодер 20 і декодер 30 описуються на основі фіг. 1-3.

Фіг. 1А є схематичною блок-схемою, яка ілюструє зразкову систему 10 кодування, наприклад, систему 10 відеокодування (або коротко систему 10 кодування), яка може використовувати методики даної заявки. Відеокодер 20 (або коротко кодер 20) і відеодекодер 30 (або коротко декодер 30) системи 10 кодування відео є прикладами пристроїв, які можуть бути виконані з можливістю виконання методик згідно з різними прикладами, описаними в даній заявці.

Як показано на фіг. 1А, система 10 кодування містить пристрій-джерело 12, виконаний з можливістю надання кодованих даних 21 зображення, наприклад, в пристрій-одержувач 14 для декодування кодованих даних 13 зображення.

Пристрій-джерело 12 містить кодер 20 і може додатково, тобто необов'язково, містити джерело 16 зображення, препроцесор (або модуль попередньої обробки) 18, наприклад, препроцесор 18 зображення, і інтерфейс або модуль 22 зв'язку.

Джерело 16 зображення може містити або бути пристроєм захоплення зображення будь-якого типу, наприклад, камерою для захоплення реального зображення і/або пристроєм генерування зображення будь-якого вигляду, наприклад, процесор комп'ютерної графіки для генерування комп'ютерного анімованого зображення, або будь-який інший пристрій для отримання і/або надання реального зображення, згенерованого комп'ютером зображення (наприклад, вміст екрана, зображення віртуальної реальності (VR)) і/або будь-яка їхня комбінація (наприклад, зображення доповненої реальності (AR)). Джерелом зображення може бути будь-який тип пам'яті або сховища, в якому зберігається будь-яке із зазначених вище зображень.

На відміну від препроцесора 18 і обробки, яка виконується модулем 18 попередньої обробки, зображення або дані 17 зображення також можуть називатися необробленим зображенням або даними 17 необроблених зображення.

Препроцесор 18 виконаний з можливістю прийому даних 17 (необробленого) зображення і виконання попередньої обробки даних 17 зображення для отримання заздалегідь обробленого зображення 19 або даних 19 заздалегідь оброблених зображення. Попередня обробка, яка виконується препроцесором 18, може, наприклад, включати в себе обрізання, перетворення колірного формату (наприклад, з RGB на YCbCr), корекцію кольору або зменшення шуму. Можна зрозуміти, що модуль 18 попередньої обробки може бути необов'язковим компонентом.

Відеокодер 20 виконаний з можливістю прийому даних 19 заздалегідь обробленого зображення і надання даних 21 кодованого зображення (додаткова інформація буде надана нижче, наприклад, на основі фіг. 2).

Інтерфейс 22 зв'язку пристрою-джерела 12 може бути виконаний з можливістю прийому даних 21 кодованого зображення і передачі даних 21 кодованого зображення (або будь-якої їхньої подальшої обробленої версії) по каналу 13 зв'язку на інший пристрій, наприклад, на пристрій-одержувач 14 або будь-який інший пристрій для зберігання або прямого відновлення.

Пристрій-одержувач 14 містить декодер 30 (наприклад, відеодекодер 30) і може додатково, тобто необов'язково, містити інтерфейс або модуль 28 зв'язку, постпроцесор 32 і пристрій 34 відображення.

Інтерфейс 28 зв'язку пристрою-одержувача 14 виконаний з можливістю прийому даних 21 кодованого зображення (або будь-якої їхньої подальшої обробленої версії), наприклад, безпосередньо від пристрою-джерела 12 або з будь-якого іншого джерела, наприклад, запам'ятовувального пристрою, наприклад, пристрій зберігання даних кодованого зображення, і надання даних 21 кодованого зображення в декодер 30.

Інтерфейс 22 зв'язку і інтерфейс 28 зв'язку можуть бути виконані з можливістю передачі або прийому даних 21 кодованого зображення або кодованих даних 13 через пряму лінію зв'язку між пристроєм-джерелом 12 і пристроєм-одержувачем 14, наприклад, пряме дротове або бездротове з'єднання, або через мережу будь-якого типу, наприклад, дротову або бездротову мережу або будь-яку їхню комбінацію, або будь-яку приватну і загальнодоступну мережу, або будь-яку їхню комбінацію.

Інтерфейс 22 зв'язку може, наприклад, бути виконаний з можливістю упакування даних 21 кодованого зображення у відповідний формат, наприклад, в пакети, і/або обробки даних

кодованого зображення з використанням будь-якого типу кодування передачі або обробки для передачі по каналу зв'язку або через мережу зв'язку.

Інтерфейс 28 зв'язку, який утворює аналог інтерфейсу 22 зв'язку, може, наприклад, бути виконаний з можливістю прийому переданих даних і обробки даних передачі з використанням 5 будь-якого виду відповідного декодування або обробки передачі і/або розпаковування для отримання даних 21 кодованого зображення.

І інтерфейс 22 зв'язку, і інтерфейс 28 зв'язку можуть бути виконані як однонаправлені інтерфейси зв'язку, як показано стрілкою для каналу 13 зв'язку на фіг. 1А, від пристрою-джерела 12 до пристрою-одержувача 14, або як інтерфейси двонаправленого зв'язку, і може 10 бути сконфігурований, наприклад, для відправки і прийому повідомлень, наприклад, для встановлення з'єднання, для підтвердження і обміну будь-якою іншою інформацією, яка стосується каналу зв'язку і/або передачі даних, наприклад, передачі даних кодованого зображення.

Декодер 30 виконаний з можливістю прийому даних 21 кодованого зображення і надання 15 даних 31 декодованого зображення або декодованого зображення 31 (додаткова інформація буде надана нижче, наприклад, на основі фіг. 3 або фіг. 5).

Постпроцесор 32 пристрою-одержувача 14 виконаний з можливістю постобробки даних 31 декодованого зображення (які також називають даними відновленого зображення), наприклад, 20 декодованого зображення 31, для отримання даних 33 зображення з постобробкою, наприклад, зображення 33 з постобробкою. Постобробка, яка виконується модулем 32 постобробки, може включати, наприклад, перетворення колірного формату (наприклад, з YCbCr в RGB), корекцію кольору, обрізання або передискретизацію, або будь-яку іншу обробку, наприклад, для підготовки даних декодованого зображення 31 для відображення, наприклад, за допомогою пристрою 34 відображення.

Пристрій 34 відображення пристрою-одержувача 14 виконаний з можливістю прийому даних 25 33 зображення з постобробкою для відображення зображення, наприклад, користувачу або глядачу. Пристрій 34 відображення може бути або містити будь-який вид дисплея для представлення відновленого зображення, наприклад, інтегрований або зовнішній дисплей або монітор. Дисплеї можуть, наприклад, включати рідкокристалічні дисплеї (LCD), дисплеї на 30 органічних світлодіодах (OLED), плазмові дисплеї, проектори, мікро-світлодіодні дисплеї, рідкі кристали на кремнії (LCoS), цифровий світловий процесор (DLP) або будь-які інші дисплеї.

Хоча на фіг. 1А представлений пристрій-джерело 12 і пристрій-одержувач 14 як окремі пристрої, варіанти здійснення пристроїв також можуть містити обидві функціональні можливості, 35 пристрій-джерело 12 або відповідні функціональні можливості і пристрій-одержувач 14 або відповідні функціональні можливості. У таких варіантах здійснення пристрій-джерело 12 або відповідні функціональні можливості і пристрій-одержувач 14 або відповідні функціональні можливості можуть бути реалізовані з використанням одних і тих самих апаратних засобів і/або програмного забезпечення або за допомогою окремих апаратних засобів і/або програмного забезпечення або будь-якої їхньої комбінації.

Як буде очевидно для фахівця на основі опису, наявність і (точно) розділення 40 функціональних можливостей різних модулів або функцій в пристрої-джерелі 12 і/або пристрої-одержувачі 14, як показано на фіг. 1А, може варіюватися залежно від фактичного пристрою і застосування.

Кодер 20 (наприклад, відеокодер 20) або декодер 30 (наприклад, відеодекодер 30), або як 45 кодер 20, так і декодер 30 можуть бути реалізовані за допомогою схем обробки, як показано на фіг. 1В, таких як один або більше мікропроцесорів, процесорів цифрових сигналів (DSP), прикладних програм, спеціальні інтегральні схеми (ASIC), програмовані користувачем вентильні матриці (FPGA), дискретна логіка, обладнання, призначене для відеокодування, або будь-які їхні комбінації. Кодер 20 може бути реалізований через схему 46 обробки для втілення різних 50 модулів, як обговорювалося у випадку кодера 20 на фіг. 2, і/або будь-яку іншу систему або підсистему кодування, описану в даному винаході. Декодер 30 може бути реалізований через схему 46 обробки для втілення різних модулів, як обговорювалося у випадку декодера 30 на фіг. 3, і/або будь-яку іншу систему або підсистему декодера, описану в даному документі. Схема обробки може бути виконана з можливістю здійснення різних операцій, як описано нижче. Як 55 показано на фіг. 5, якщо методики частково реалізовані в програмному забезпеченні, пристрій може зберігати інструкції для програмного забезпечення на відповідно енергонезалежному машинозчитуваному носії даних і може виконувати інструкції в апаратних засобах, використовуючи один або більше процесорів для виконання методик згідно з даним винаходом. Як відеокодер 20, так і відеодекодер 30 може бути інтегрований як частина об'єднаного 60 кодера/декодера (CODEC) в одному пристрої, наприклад, як показано на фіг. 1В.

Пристрій-джерело 12 і пристрій-одержувач 14 можуть містити будь-який пристрій з широкого діапазону пристроїв, включаючи будь-які види портативних або стаціонарних пристроїв, наприклад, портативні або переносні комп'ютери, мобільні телефони, смартфони, планшети або планшетні комп'ютери, камери, настільні комп'ютери, набори-приставки, телевізори, пристрої відображення, цифрові медіаплеєри, ігрові приставки, пристрої потокової передачі відео (наприклад, сервери служб контенту або сервери доставки контенту), пристрій ширококомовного приймача, пристрій ширококомовного передавача тощо, і можуть використовувати або не використовувати будь-який вид операційної системи. У деяких випадках пристрій-джерело 12 і пристрій-одержувач 14 можуть бути обладнані для бездротового зв'язку. Отже, пристрій-джерело 12 і пристрій-одержувач 14 можуть бути пристроями бездротового зв'язку.

У деяких випадках система 10 відеокодування, проілюстрована на фіг. 1А, є просто прикладом, і методи даної заявки можуть бути застосовані до налаштувань кодування відео (наприклад, кодування відео або декодування відео), які не обов'язково включають будь-який обмін даними між пристроями кодування і декодування. В інших прикладах дані можуть бути отримані з локальної пам'яті, передаються через мережу тощо. Пристрій кодування відео може кодувати і зберігати дані в пам'яті, і/або пристрій декодування відео може витягувати і декодувати дані з пам'яті. У деяких прикладах кодування і декодування здійснюються пристроями, які не обмінюються даними один з одним, а просто кодують дані в пам'ять і/або витягують і декодують дані з пам'яті.

Для зручності опису варіанти здійснення винаходу описані у даному винаході, наприклад, з посиланням на високоефективне відеокодування (HEVC) або на опорне програмне забезпечення універсального кодування відео (VVC), стандарту кодування відео наступного покоління, розробленого Об'єднаною групою співпраці з кодуванням відео (JCT-VC) Групи експертів ITU-T з кодування відео (VCEG) і Групи експертів ISO/IEC з рухомих зображень (MPEG). Фахівець в даній галузі техніки зрозуміє, що варіанти здійснення винаходу не обмежуються HEVC або VVC.

#### Кодер і спосіб кодування

На фіг. 2 представлена блок-схема прикладу відеокодера 20, який виконаний з можливістю реалізації способів даної заявки. У прикладі на фіг. 2 відеокодер 20 містить вхід 201 (або інтерфейс 201 входу), модуль 204 обчислення залишку, модуль 206 обробки перетворення, модуль 208 квантування, модуль 210 зворотного квантування і модуль 212 обробки зворотного перетворення, модуль 214 відновлення, модуль 220 контурного фільтра, буфер 230 (DPB) декодованих зображень, модуль 260 вибору режиму, модуль 270 ентропійного кодування і вихід 272 (або інтерфейс 272 виходу). Модуль 260 вибору режиму може містити модуль 244 інтер-прогнозування, модуль 254 інтра-прогнозування і модуль 262 розбивання. Модуль 244 інтер-прогнозування може містити модуль оцінки руху і модуль компенсації руху (не показані). Відеокодер 20, показаний на фіг. 2, також може називатися гібридним відеокодером або відеокодером згідно з гібридним відеокодером.

Модуль 204 обчислення залишку, модуль 206 обробки перетворення, модуль 208 квантування, модуль 260 вибору режиму можуть стосуватися формування прямого шляху сигналу кодера 20, у той час як модуль 210 зворотного квантування, модуль 212 обробки зворотного перетворення, модуль 214 відновлення, буфер 216, контурний фільтр 220, буфер 230 (DPB) декодованих зображень, модуль 244 інтер-прогнозування і модуль 254 інтра-прогнозування можуть стосуватися формування зворотного шляху сигналу відеокодера 20, при цьому зворотний шлях сигналу відеокодера 20 відповідає шляху сигналу декодера (див. відеокодер 30 на фіг. 3). Модуль 210 зворотного квантування, модуль 212 обробки зворотного перетворення, модуль 214 відновлення, контурний фільтр 220, буфер 230 (DPB) декодованих зображень, модуль 244 інтер-прогнозування і модуль 254 інтра-прогнозування також стосуються формування "вбудованого декодера" відеокодера 20.

#### Зображення і розбивання зображень (зображення і блоки)

Кодер 20 може бути виконаний з можливістю прийому, наприклад, через вхід 201 зображення 17 (або даних 17 зображення), наприклад зображення з послідовності зображень, які утворюють відео або відеопослідовність. Прийняте зображення або дані зображення також можуть бути заздалегідь обробленим зображенням 19 (або даними 19 заздалегідь обробленого зображення). Для спрощення наведений нижче опис посилається на зображення 17. Зображення 17 також може бути назване поточним зображенням або зображенням, яке підлягає кодуванню (зокрема, при відеокодуванні, щоб відрізнити поточне зображення від інших зображень, наприклад, раніше закодованих і/або декодованих зображень тієї самої відеопослідовності, тобто відеопослідовності, яка також містить поточне зображення).

(Цифрове) зображення є або може розглядатися як двовимірний масив або матриця дискретних відліків зі значеннями інтенсивності. Дискретний відлік в масиві також може називатися пікселем (скорочена форма елемента зображення) або *pel*. Кількість дискретних відліків в горизонтальному і вертикальному напрямку (або по осі) масиву або зображення визначає розмір і/або роздільну здатність зображення. Для представлення кольору зазвичай використовують три колірні компоненти, тобто зображення може бути представлене або містити три масиви дискретних відліків. У форматі RGB або колірному просторі зображення містить відповідний масив дискретних відліків червоного, зеленого і синього кольорів. Однак, при кодуванні відео кожний піксель зазвичай представлений у форматі яскравості і кольоровості або колірному просторі, наприклад, YCbCr, який містить компонент яскравості, позначений Y (іноді замість нього також використовується L), і два компоненти кольоровості, позначені Cb і Cr. Компонент Y яскравості (або короткої яскравості) є яскравістю або інтенсивністю рівня сірого (наприклад, як у напівтоновому зображенні), у той час як два компоненти кольоровості (або короткої кольоровості) Cb і Cr є компонентами кольоровості або інформації про колір. Відповідно, зображення в форматі YCbCr містить масив дискретних відліків яскравості значень дискретних відліків яскравості (Y) і два масиви дискретних відліків кольоровості значень кольоровості (Cb і Cr). Зображення в форматі RGB можуть бути перетворені або перетворені в формат YCbCr, і навпаки, цей процес також відомий як перетворення або конверсія кольору. Якщо зображення є монохромним, то воно може містити лише масив дискретних відліків яскравості. Відповідно, зображення може, наприклад, бути масивом дискретних відліків яскравості в монохромному форматі або масивом дискретних відліків яскравості і двома відповідними масивами дискретних відліків кольоровості у колірному форматі 4:2:0, 4:2:2 і 4:4:4.

Варіанти здійснення відеокодера 20 можуть містити модуль розбивання зображення (не показаний на фіг. 2), виконаний з можливістю розбивання зображення 17 на безліч (які зазвичай не перекриваються) блоків 203 зображення. Ці блоки також можуть називатися кореневими блоками, макроблоками (H.264/AVC) або блоками дерева кодування (CTB) або одиницями дерева кодування (CTU) (H.265/HEVC і VVC). Блок розбивання зображення може бути пристосований для використання одного і того самого розміру блока для всіх зображень відеопослідовності і відповідної сітки, що визначає розмір блока, або для зміни розміру блока між зображеннями або підмножинами або групами зображень, а також для розбивання кожного зображення на відповідні блоки.

У додаткових варіантах здійснення відеокодера може бути виконаний з можливістю прийому безпосередньо блока 203 зображення 17, наприклад, одного, декількох або всіх блоків, які утворюють зображення 17. Блок 203 зображення також може називатися блоком поточного зображення або блоком зображення, який повинен бути кодований.

Як і зображення 17, блок 203 зображення, в свою чергу, є або може розглядатися як двовимірний масив або матриця дискретних відліків зі значеннями інтенсивності (значеннями дискретних відліків), хоча і меншого розміру, ніж зображення 17. Іншими словами, блок 203 може містити, наприклад, один масив дискретних відліків (наприклад, масив яскравості у випадку монохромного зображення 17 або масив яскравості або кольоровості кольорового зображення) або три масиви дискретних відліків (наприклад, масив яскравості і два масиви кольоровості у випадку кольорового зображення 17) або будь-яку іншу кількість і/або вид масивів залежно від застосовуваного колірному формату. Кількість дискретних відліків у горизонтальному і вертикальному напрямку (або по осі) блока 203 визначає розмір блока 203. Відповідно, блок може, наприклад, бути масивом дискретних відліків  $M \times N$  ( $M$  колонок на  $N$  рядків) або масив  $M \times N$  коефіцієнтів перетворення.

Варіанти здійснення відеокодера 20, як показано на фіг. 2, можуть бути виконані з можливістю кодування зображення 17 блок за блоком, наприклад, кодування і прогнозування здійснюється для кожного блока 203.

Варіанти здійснення відеокодера 20, як показано на фіг. 2, можуть бути додатково виконані з можливістю розбивання і/або кодування зображення з використанням зрізів (які також називають відео зрізами (слайсами)), при цьому зображення може бути розділене або закодоване з використанням одного або більше слайсів (зазвичай без перекриття), і кожний слайс може містити один або більше блоків (наприклад, CTU).

Варіанти здійснення відеокодера 20, як показано на фіг. 2, можуть бути додатково виконані з можливістю розбивання і/або кодування зображення з використанням груп комірок (плиток, *tile*) (які також називають групами комірок) і/або комірок (які також називають відео комірками), в яких зображення може бути розділене або закодоване з використанням однієї або більше груп комірок (які зазвичай не перекриваються), і кожна група комірок може містити, наприклад, один або більше блоків (наприклад, CTU) або одну або більше комірок, при цьому кожна комірка,

наприклад, може мати прямокутну форму і може містити один або більше блоків (наприклад, СТУ), наприклад, повні або дробові блоки.

#### Обчислення залишку

5 Модуль 204 обчислення залишку може бути виконаний з можливістю обчислення залишкового блока 205 (який також називають залишком 205) на основі блока 203 зображення і блока 265 прогнозування (додаткова інформація про блок 265 прогнозування буде надана нижче), наприклад, за допомогою віднімання значень дискретних відліків блока 265 прогнозу від значень дискретних відліків блока 203 зображення, дискретний відлік за дискретним відліком (піксель за пікселем), щоб отримати залишковий блок 205 в зоні дискретних відліків.

#### 10 Перетворення

Модуль 206 обробки перетворення може бути виконаний з можливістю застосування перетворення, наприклад дискретного косинусного перетворення (DCT) або дискретного синусного перетворення (DST), до значень дискретних відліків залишкового блока 205 для отримання коефіцієнтів 207 перетворення в зоні перетворення. Коефіцієнти 207 перетворення 15 можуть також називатися залишковими коефіцієнтами перетворення і представляти залишковий блок 205 в зоні перетворення.

Модуль 206 обробки перетворення може бути виконаний з можливістю застосування цілочислових апроксимацій DCT/DST, таких як перетворення, вказані для H.265/HEVC. Порівняно з ортогональним перетворенням DCT, такі цілочислові апроксимації зазвичай 20 масштабуються з певним коефіцієнтом. Щоб зберегти норму залишкового блока, який обробляється прямим і зворотним перетвореннями, в процесі перетворення застосовують додаткові коефіцієнти масштабування. Коефіцієнти масштабування зазвичай вибирають на основі певних обмежень, таких як коефіцієнти масштабування, які є степеню двійки для операцій зсуву, розрядність коефіцієнтів перетворення, компроміс між точністю і витратами на реалізацію тощо. Конкретні коефіцієнти масштабування, наприклад, задаються для зворотного 25 перетворення, наприклад, модулем 212 обробки зворотного перетворення (і відповідним зворотним перетворенням, наприклад, модулем 312 обробки зворотного перетворення у відеокодері 30), і відповідні коефіцієнти масштабування для прямого перетворення, наприклад, за допомогою модуля 206 обробки перетворення, в кодері 20 можуть бути визначені відповідно.

30 Варіанти здійснення відеокодера 20 (відповідно модуль 206 обробки перетворення) можуть бути виконані з можливістю виведення параметрів перетворення, наприклад, типу перетворення або перетворень, наприклад, безпосередньо або у кодованому або стисненому вигляді через модуль 270 ентропійного кодування, так що, наприклад, відеодекодер 30 може приймати і використовувати параметри перетворення для декодування.

#### 35 Квантування

Модуль 208 квантування може бути виконаний з можливістю квантування коефіцієнтів 207 перетворення для отримання квантованих коефіцієнтів 209, наприклад, за допомогою застосування скалярного квантування або векторного квантування. Квантовані коефіцієнти 209 40 також можуть називатися квантованими коефіцієнтами 209 перетворення або квантованими залишковими коефіцієнтами 209.

Процес квантування може зменшити бітову глибину, пов'язану з деякими або всіма коефіцієнтами 207 перетворення. Так, наприклад,  $n$ -бітовий коефіцієнт перетворення може бути округлений до  $m$ -бітового коефіцієнта перетворення під час квантування, де  $n$  є більшим, ніж  $m$ . Ступінь квантування може бути змінений за допомогою регулювання параметра квантування 45 (QP). Так, наприклад, для скалярного квантування може бути застосоване інше масштабування для досягнення більш за точного або більш грубого квантування. Менші розміри кроків квантування відповідають більш точному квантуванню, у той час як великі розміри кроків квантування відповідають більш грубому квантуванню. Застосовний розмір кроку квантування може бути вказаний параметром квантування (QP). Параметр квантування може, наприклад, 50 бути індексом для заздалегідь визначеного набору застосовних розмірів кроку квантування. Так, наприклад, малі параметри квантування можуть відповідати точному квантуванню (малі розміри кроків квантування), а великі параметри квантування можуть відповідати грубому квантуванню (великі розміри кроків квантування) або навпаки. Квантування може включати ділення на розмір кроку квантування, і відповідне  $i$ /або зворотне деквантування, наприклад, за допомогою модуля 55 210 зворотного квантування, може включати множення на розмір кроку квантування. Варіанти здійснення згідно з деякими стандартами, наприклад, HEVC, можуть бути виконані з можливістю використання параметра квантування для визначення розміру кроку квантування. Зазвичай розмір кроку квантування може бути обчислений на основі параметра квантування з використанням апроксимації з фіксованою точкою рівняння, що включає ділення. Додаткові 60 коефіцієнти масштабування можуть бути введені для квантування і деквантування, щоб

відновити норму залишкового блока, який може бути змінений через масштабування, використовуване в наближенні з фіксованою точкою рівняння для розміру кроку квантування і параметра квантування. У одному з прикладів реалізації масштабування зворотного перетворення і деквантування можуть бути об'єднані. Як альтернатива можуть бути використані оптимізовані таблиці квантування, і кодер може передавати їх в декодер, наприклад, в потоці бітів. Квантування є операцією з втратами, де втрати зростають зі збільшенням розмірів етапів квантування.

Варіанти здійснення відеокодера 20 (відповідно модуль 208 квантування) можуть бути виконані з можливістю виведення квантування (QP), наприклад, безпосередньо або у кодованому вигляді через модуль 270 ентропійного кодування, так що, наприклад, відеодекодер 30 може приймати і застосовувати параметри квантування для декодування.

#### Зворотне квантування

Модуль 210 зворотного квантування виконаний з можливістю застосування зворотного квантування блока 208 квантування до квантованих коефіцієнтів для отримання деквантованих коефіцієнтів 211, наприклад, за допомогою застосування зворотної схеми квантування, що застосовується блоком 208 квантування, на основі або з використанням того самого розміру кроку квантування, що і у випадку модуля 208 квантування. Деквантовані коефіцієнти 211 також можуть називатися деквантованими залишковими коефіцієнтами 211 і відповідати – хоча зазвичай вони не ідентичні коефіцієнтам перетворення через втрату при квантуванні – коефіцієнтам 207 перетворення.

#### Зворотне перетворення

Модуль 212 обробки зворотного перетворення виконаний з можливістю застосування зворотного перетворення, застосовуваного блоком 206 обробки перетворення, наприклад, зворотного дискретного косинусного перетворення (DCT) або зворотного дискретного синусного перетворення (DST) або інших зворотних перетворення, щоб отримати відновлений залишковий блок 213 (або відповідні коефіцієнти 213 деквантування) в зоні дискретних відліків. Відновлений залишковий блок 213 також може називатися блоком 213 перетворення.

#### Відновлення

Модуль 214 відновлення (наприклад, суматор 214) виконаний з можливістю додавання блока 213 перетворення (тобто відновленого залишкового блока 213) до блока 265 прогнозування, щоб отримати відновлений блок 215 в зоні дискретних відліків, наприклад, за допомогою додавання (дискретний відлік за дискретним відліком) значень дискретних відліків відновленого залишкового блока 213 і значення дискретних відліків блока 265 прогнозування.

#### Фільтрування

Модуль 220 контурного фільтра (або скорочено "контурний фільтр" 220) виконаний з можливістю фільтрування відновленого блока 215, щоб отримати відфільтрований блок 221, або, загалом, для фільтрування відновлених дискретних відліків для отримання відфільтрованих дискретних відліків. Модуль контурного фільтра, наприклад, виконаний з можливістю згладжування переходів пікселів або іншого покращення якості відео. Модуль 220 контурного фільтра може містити один або більше контурних фільтрів, таких як фільтр усунення блоковості, фільтр з адаптивним зміщенням дискретних відліків (SAO), або один або більше інших фільтрів, таких як двосторонній фільтр, адаптивний контурний фільтр (ALF), фільтри згладжування, підвищення різкості або спільні фільтри, або будь-яку їхню комбінацію. Хоча модуль 220 контурного фільтра показаний на фіг. 2 як контурний фільтр, в інших конфігураціях модуль 220 контурного фільтра може бути реалізований як постконтурний фільтр. Відфільтрований блок 221 також може називатися відфільтрованим відновленим блоком 221.

Варіанти здійснення відеокодера 20 (відповідно модуля 220 контурного фільтра) можуть бути виконані з можливістю виведення параметрів контурного фільтра (таких як інформація адаптивного зміщення дискретних відліків), наприклад, безпосередньо або у кодованому вигляді через модуль 270 ентропійного кодування, так що, наприклад, декодер 30 може приймати і застосовувати аналогічні параметри контурного фільтра або відповідні контурні фільтри для декодування.

#### Буфер декодованих зображень

Буфер 230 декодованих зображень (DPB) може бути пам'яттю, в якій зберігаються опорні зображення або, загалом, дані опорних зображень для кодування відеоданих за допомогою відеокодера 20. DPB 230 може бути утворений будь-яким із множини запам'ятовувальних пристроїв, таких як динамічна пам'ять з довільним доступом (DRAM), у тому числі синхронна DRAM (SDRAM), магніторезистивна RAM (MRAM), резистивна RAM (RRAM) або запам'ятовувальні пристрої інших типів. Буфер 230 (DPB) декодованих зображень може бути виконаний з можливістю зберігання одного або більше фільтрованих блоків 221. Буфер 230

декодованих зображень може бути додатково виконаний з можливістю зберігання інших раніше відфільтрованих блоків, наприклад, раніше відновлених і відфільтрованих блоків 221, того самого поточного зображення або різних зображень, наприклад, раніше відновлених зображень, і може надавати повні раніше відновлені, тобто декодовані, зображення (і відповідні опорні блоки і дискретні відліки) і/або частково відновлене поточне зображення (і відповідні опорні блоки і дискретні відліки), наприклад, для інтер-прогнозування. Буфер 230 декодованих зображень (DPB) також може бути виконаний з можливістю зберігання одного або більше нефільтрованих відновлених блоків 215 або, загалом, нефільтрованих відновлених дискретних відліків, наприклад, якщо відновлений блок 215 не фільтрується модулем 220 контурного фільтра, або будь-якої іншої додатково обробленої версії відновлених блоків або дискретних відліків.

Вибір режиму (Розбивання і Прогнозування)

Модуль 260 вибору режиму містить модуль 262 розбивання, модуль 244 інтер-прогнозування і модуль 254 інтра-прогнозування і виконаний з можливістю прийому або отримання даних вихідного зображення, наприклад, вихідного блока 203 (поточного блока 203 поточного зображення 17), і даних відновленого зображення, наприклад, відфільтрованих і/або нефільтрованих відновлених дискретних відліків або блоків того самого (поточного) зображення і/або з одного або множини попередньо декодованих зображень, наприклад, з буфера 230 декодованих зображень або інших буферів (наприклад, лінійного (рядкового) буфера, не показаний). Дані відновленого зображення використовуються як дані опорного зображення для прогнозування, наприклад, інтер-прогнозування або інтра-прогнозування, щоб отримати блок 265 прогнозування або предиктор 265.

Модуль 260 вибору режиму може бути виконаний з можливістю визначення або вибору розбивання для режиму прогнозування поточного блока (включаючи відсутність розбивання) і режиму прогнозування (наприклад, режиму інтра- або інтер-прогнозування) і генерування відповідного блока 265 прогнозування, який використовується для обчислення залишкового блока 205 і для відновлення відновленого блока 215.

Варіанти здійснення модуля 260 вибору режиму можуть бути виконані з можливістю вибору розбивання і режиму прогнозування (наприклад, з тих, які підтримуються або доступні для модуля 260 вибору режиму), які забезпечують найкращий збіг або, іншими словами, мінімальний залишок (мінімальний залишок означає краще стиснення для передачі або зберігання), або мінімальні витрати на сигналізування (мінімальні витрати на сигналізування означають краще стиснення для передачі або зберігання), або який враховує або врівноважує обидва параметри. Модуль 260 вибору режиму може бути виконаний з можливістю визначення режиму розбивання і прогнозування на основі оптимізації співвідношення спотворення-швидкість (RDO), тобто вибору режиму прогнозування, який забезпечує мінімальне співвідношення спотворення-швидкість. Такі терміни, як "кращий", "мінімальний", "оптимальний" тощо в даному контексті не обов'язково стосуються загального значення "кращий", "мінімальний", "оптимальний" тощо, а й можуть стосуватися також критерію припинення або вибору, такого як значення, що перевищує або є меншим, ніж порогове значення, або інші характеристики, які потенційно приводять до "неоптимального вибору", але зменшують складність і час обробки.

Іншими словами, модуль 262 розбивання може бути виконаний з можливістю розбивання блока 203 на більш дрібні блокові розділи або підблоки (які знов утворюють блоки), наприклад, ітеративно з використанням розбивання на основі квадратичного дерева (QT), двійкового розбивання (BT) або розбивання на основі трійкового дерева (TT) або будь-якої їхньої комбінації, і для здійснення, наприклад, прогнозування для кожного з розбивань блока або підблоків, при цьому вибір режиму включає вибір деревоподібної структури розділеного блока 203, і режими прогнозу застосовуються до кожного з розбивань або підблоків блока.

Далі більш детально пояснюється розбивання (наприклад, за допомогою модуля 260 розбивання) і обробка прогнозування (за допомогою модуля 244 інтер-прогнозування і модуля 254 інтра-прогнозування), які здійснюються наведеним як приклад відеокодером 20.

Розбивання

Модуль 262 розбивання може розбивати (або розділяти) поточний блок 203 на більш дрібні елементи розбивання, наприклад, менші блоки квадратного або прямокутного розміру. Ці менші блоки (які також можуть називатися підблоками) можуть бути додатково розділені на ще менші розділи. Це також стосується розбивання дерева або ієрархічного розбивання дерева, в якому кореневий блок, наприклад, на рівні кореневого дерева 0 (рівень ієрархії 0, глибина 0), може бути рекурсивно розділений, наприклад, розділений на два або більше блоків наступного нижчого рівня дерева, наприклад, вузли на рівні дерева 1 (рівень ієрархії 1, глибина 1), при

цьому ці блоки можуть бути знову розділені на два або більше блоків наступного нижчого рівня, наприклад, рівень дерева 2 (рівень ієрархії 2, глибина 2) тощо, доти, доки розбивання не буде завершено, наприклад, оскільки виконаний критерій завершення, наприклад, досягнута максимальна глибина дерева або мінімальний розмір блока. Блоки, які не розділяються далі, також називають листовими блоками або кінцевими вузлами дерева. Дерево, яке використовує розбивання на два елементи розбивання, називається двійковим деревом (BT), дерево, яке використовує розбивання на три елементи розбивання, називається трійковим деревом (TT), а дерево, яке використовує розбивання на чотири елементи розбивання, називається квадродеревом (QT).

Як згадувалося раніше, використовуваний у даному винаході термін "блок" може бути частиною, зокрема квадратною або прямокутною частиною зображення. Щодо, наприклад, HEVC і VVC, блок може бути або відповідати одиничному модулю дерева кодування (CU), одиничному модулю кодування (CU), одиничному модулю прогнозування (PU) і одиничному модулю перетворення (TU) і/або відповідним блокам, наприклад, блоку дерева кодування (CTB), блоку кодування (CB), блоку перетворення (TB) або блоку прогнозування (PB).

Так, наприклад, одиничний модуль дерева кодування (CU) може бути або містити СТВ дискретних відліків яскравості, два відповідних СТВ дискретних відліків кольоровості зображення, яке має три масиви дискретних відліків, або СТВ дискретних відліків монохромного зображення або зображення, яке кодується з використанням трьох окремих колірних площин і синтаксичних структур, які використовуються для кодування дискретних відліків. Відповідно, блок дерева кодування (CTB) може бути блоком  $N \times N$  дискретних відліків для деякого значення  $N$ , так що розділення компонента на СТВ є розбиванням. Одиничний модуль кодування (CU) може бути або містити блок кодування дискретних відліків яскравості, два відповідних блоки кодування дискретних відліків кольоровості зображення, яке має три масиви дискретних відліків, або блок кодування дискретних відліків монохромного зображення або зображення, яке кодується з використанням трьох окремих колірних площин і синтаксичних структур, які використовуються для кодування дискретних відліків. Відповідно, блок кодування (CB) може бути блоком  $M \times N$  дискретних відліків для деяких значень  $M$  і  $N$ , так що розділення СТВ на блоки кодування є розбиванням.

У варіантах здійснення, наприклад, згідно з HEVC, одиниця дерева кодування (CU) може бути розділена на CU з використанням структури квадродерева, позначеної як дерево кодування. Рішення про те, чи слід кодувати ділянку зображення з використанням проміжного (інтер-) (часового) або внутрішнього (інтра-) (просторового) прогнозування, приймається на рівні CU. Кожний CU може бути додатково розділений на один, два або чотири PU відповідно до типу розбивання PU. Всередині одного PU застосовується один і той самий процес прогнозування, і відповідна інформація передається в декодер на основі PU. Після отримання залишкового блока за допомогою застосування процесу прогнозування на основі типу розбивання PU, CU може бути розділений на блоки перетворення (TU) відповідно до іншої структури квадродерева, аналогічної дереву кодування для CU.

У варіантах здійснення, наприклад згідно з останнім найновішим розроблюваним стандартом відеокодування, який називається універсальним відеокодуванням (VVC), для розділення блока кодування використовують, наприклад, розділення комбінованого квадродерева і двійкового дерева (QTBT). У структурі блока QTBT CU може мати або квадратну, або прямокутну форму. Так, наприклад, одиничний блок дерева кодування (CU) спочатку розділяється на структуру квадродерева. Кінцеві вузли квадродерева додатково розділяються за допомогою структури двійкового дерева або трійкового дерева. Кінцеві вузли дерева розділення називаються одиничними модулями кодування (CU), і ця сегментація використовується для обробки прогнозування і перетворення без будь-якого подальшого розбивання. Це означає, що CU, PU і TU мають однаковий розмір блока в структурі блока кодування QTBT. Паралельно, разом зі структурою блока QTBT можна використовувати множинне розбивання, наприклад, розбивання трійкового дерева.

У одному прикладі модуль 260 вибору режиму відеокодера 20 може бути виконаний з можливістю здійснення будь-якої комбінації способів розбивання, описаних в даному документі.

Як описано вище, відеокодер 20 виконаний з можливістю визначення або вибору найкращого або оптимального режиму прогнозування з набору (наприклад, заздалегідь визначених) режимів прогнозування. Набір режимів прогнозування може містити, наприклад, режими інтра-прогнозування і/або режими інтер-прогнозування.

Інтра-прогнозування

Набір режимів інтра-прогнозування може містити 35 різних режимів інтра-прогнозування, наприклад ненаправлені режими, такі як режим постійного (або середнього) і планарний режим,

або направлені режими, наприклад, як визначено в HVEC, або може містити 67 різних режимів інтра-прогнозування, наприклад, ненаправлені режими, такі як режим DC (або середнього) і планарний режим, або направлені режими, наприклад, як визначено для VVC.

5 Модуль 254 інтра-прогнозування виконаний з можливістю використання відновлених дискретних відліків сусідніх блоків одного і того самого поточного зображення для генерування блока 265 інтра-прогнозування згідно з режимом інтра-прогнозування набору режимів інтра-прогнозування.

10 Модуль 254 інтра-прогнозування (або, загалом, модуль 260 вибору режиму) додатково виконаний з можливістю виведення інтра-прогнозування (або, загалом, інформації, яка вказує вибраний режим інтра-прогнозування для блока) в модуль 270 ентропійного кодування у формі елементів 266 синтаксису для включення в дані 21 закодованого зображення, щоб, наприклад, відеодекодер 30 міг приймати і використовувати параметри прогнозування для декодування.

Інтер-прогнозування

15 Набір (можливих) режимів інтер-прогнозування залежить від доступних опорних зображень (тобто попередніх, щонайменше частково декодованих зображень, які, наприклад, зберігаються в DBP 230) і інших параметрів інтер-прогнозування, наприклад, залежно від того, чи використовується все опорне зображення або лише частина, наприклад, ділянка пошуку вікна навколо ділянки поточного блока, опорного зображення для пошуку найкращої відповідності опорного блока, і/або, наприклад, від того, чи застосовуються інтерполяції пікселів, наприклад, 20 інтерполяція половини/напів-пікселя і/або чверті пікселя, чи ні.

На додаток до зазначених вище режимів прогнозування можуть бути застосовані режим пропуску і/або прямиї режим.

25 Модуль 244 інтер-прогнозування може містити модуль оцінки руху (ME) (не показаний на фіг. 2) і модуль компенсації руху (MC) (обидва не показані на фіг. 2). Модуль оцінки руху може бути виконаний з можливістю прийому або отримання блока 203 зображення (поточного блока 203 поточного зображення 17) і декодованого зображення 231 або щонайменше одного або множини попередньо відновлених блоків, наприклад, відновлених блоків одного або множини інших/відмінних попередньо декодованих зображень 231 для оцінки руху. Так, наприклад, 30 відеопослідовність може містити поточне зображення і попередньо декодовані зображення 231, або, іншими словами, поточне зображення і попередньо декодовані зображення 231 можуть бути частиною або утворювати послідовність зображень, які утворюють відеопослідовність.

35 Кодер 20 може, наприклад, бути виконаний з можливістю вибору опорного блока з множини опорних блоків однакових або різних зображень з множини інших зображень і надання опорного зображення (або індексу опорного зображення) і/або зміщення (просторового зміщення) між положенням (X, Y) опорного блока і положенням поточного блока як взаємопов'язані параметри прогнозування в модуль оцінки руху. Це зміщення також називається вектором руху (MV).

40 Модуль компенсації руху виконаний з можливістю отримання, наприклад, прийому параметра інтер-прогнозування і виконання інтер-прогнозування на основі або з використанням параметра інтер-прогнозування для отримання блока 265 інтер-прогнозування. Компенсація руху, що виконується блоком компенсації руху, може включати отримання або генерування блока прогнозування на основі вектора руху/блока, визначеного за допомогою оцінки руху, можливе виконання інтерполяції до субпіксельної точності. Інтерполяційне фільтрування може генерувати додаткові дискретні відліки пікселів з дискретних відліків відомих пікселів, у такий 45 спосіб потенційно збільшуючи кількість блоків прогнозування потенційно придатних варіантів, які можуть бути використані для кодування блока зображення. Після прийому вектора руху для PU поточного блока зображення модуль компенсації руху може знайти блок прогнозування, на який вказує вектор руху, в одному зі списків опорних зображень. Модуль компенсації руху також може генерувати елементи синтаксису, пов'язані з блоками і відеослайсами, для використання відеодекодером 30 при декодування блоків зображення відеослайса. На додаток до або як 50 альтернатива слайсам і відповідним елементам синтаксису можуть генеруватися або використовуватися групи комірок і/або комірки і відповідні елементи синтаксису.

Ентропійне кодування

60 Модуль 270 ентропійного кодування виконаний з можливістю застосування, наприклад, алгоритму або схеми ентропійного кодування (наприклад, схеми кодування зі змінною довжиною (VLC), схеми контекстно-адаптивного VLC (CAVLC), схеми арифметичного кодування, бінаризації, контекстно-адаптивного двійкового арифметичного кодування (CABAC), заснованого на синтаксисі контекстно-адаптивного двійкового арифметичного кодування (SBAC), ентропійного кодування з розбиванням інтервалу імовірності (PIPE) або іншого способу або методики ентропійного кодування) або пропуску (без стиснення) щодо квантованих коефіцієнтів 209, параметрів інтер-прогнозування, параметрів інтра-прогнозування, параметрів

контурного фільтра і/або інших елементів синтаксису для отримання даних 21 кодованого зображення, які можуть виводитися через вихід 272, наприклад, у формі кодованого бітового потоку 21, так що, наприклад, відеодекодер 30 може приймати і використовувати ці параметри для декодування. Кодований бітовий потік 21 може бути переданий на відеодекодер 30 або збережений в пам'яті для подальшої передачі або зчитування відеодекодером 30.

Для кодування відеопотоку можуть бути використані інші зміни в структурі відеокодера 20. Так, наприклад, кодер 20, не заснований на перетворенні, може квантувати залишковий сигнал безпосередньо без блока 206 обробки перетворення. У іншій реалізації кодер 20 може мати модуль 208 квантування і модуль 210 зворотного квантування, об'єднані в один блок.

Декодер і спосіб декодування

На фіг. 3 показаний приклад відеокодера 30, який виконаний з можливістю реалізації способів даної заявки. Відеодекодер 30 виконаний з можливістю прийому даних 21 кодованого зображення (наприклад, кодованого бітового потоку 21), наприклад, кодованими кодером 20, для отримання декодованого зображення 331. Дані кодованого зображення або кодований бітовий потік містять інформацію для декодування даних кодованого зображення, наприклад, даних, які представляють блоки зображення кодованого слайса відео (і/або комірок або груп комірок) і пов'язані елементи синтаксису.

У прикладі на фіг. 3 декодер 30 містить модуль 304 ентропійного декодування, модуль 310 зворотного квантування, модуль 312 обробки зворотного перетворення, модуль 314 відновлення (наприклад, суматор 314), контурний фільтр 320, буфер 330 (DPB) декодованих зображень, модуль 360 застосування режиму, модуль 344 інтер-прогнозування і модуль 354 інтер-прогнозування. Модуль 344 інтер-прогнозування може бути або містить модуль компенсації руху. Відеодекодер 30 може, в деяких прикладах, виконувати етап декодування, загалом зворотний щодо етапу кодування, описаного у контексті відеокодера 100 на фіг. 2.

Як описано щодо кодера 20, модуль 210 зворотного квантування, модуль 212 обробки зворотного перетворення, модуль 214 відновлення, контурний фільтр 220, буфер 230 (DPB) декодованих зображень, модуль 344 інтер-прогнозування і модуль 354 інтра-прогнозування також стосуються формування "вбудованого декодера" відеокодера 20. Відповідно, модуль 310 зворотного квантування може бути ідентичний за функцією модулю 110 зворотного квантування, модуль 312 обробки зворотного перетворення може бути ідентичний за функцією модулю 212 обробки зворотного перетворення, модуль 314 відновлення може бути ідентичний за функцією модулю 214 відновлення, контурний фільтр 320 може бути ідентичний за функцією контурному фільтру 220, а буфер 330 декодованих зображень може бути ідентичний за функцією буферу 230 декодованих зображень. Отже, пояснення, надані для відповідних блоків і функцій відеокодера 20, можуть бути відповідно застосовні і до відповідних блоків і функцій відеокодера 30.

Ентропійне декодування

Модуль 304 ентропійного декодування виконаний з можливістю синтаксичного аналізу (розбору) бітового потоку 21 (або, загалом, даних 21 кодованого зображення) і здійснення, наприклад, ентропійного декодування щодо даних 21 кодованого зображення для отримання, наприклад, квантованих коефіцієнтів 309 і/або параметрів декодованого кодування (не показані на фіг. 3), наприклад, будь-яких або всіх з параметрів інтер-прогнозування (наприклад, індекс опорного зображення і вектор руху), параметра інтра-прогнозування (наприклад, індекс або режим інтра-прогнозування), параметрів перетворення, параметрів квантування, параметрів контурного фільтра і/або інших елементів синтаксису. Модуль 304 ентропійного декодування може бути виконаний з можливістю застосування алгоритмів або схем декодування, які відповідають схемам кодування, як описано у випадку модуля 270 ентропійного кодування кодера 20. Модуль 304 ентропійного декодування може бути додатково виконаний з можливістю надання параметрів інтер-прогнозування, параметра інтра-прогнозування і/або інших елементів синтаксису модулю 360 застосування режиму і інших параметрів іншим модулям декодера 30. Відеодекодер 30 може приймати елементи синтаксису на рівні відеослайса і/або рівні відеоблока. На додаток до або як альтернатива слайсам і відповідним елементам синтаксису можуть бути прийняті і/або використані групи комірок і/або комірки і відповідні елементи синтаксису.

Зворотне квантування

Модуль 310 зворотного квантування може бути виконаний з можливістю прийому параметрів квантування (QP) (або, загалом, інформації, що стосується зворотного квантування) і квантованих коефіцієнтів з даних 21 кодованого зображення (наприклад, за допомогою синтаксичного аналізу і/або декодування, наприклад, за допомогою модуля 304 ентропійного декодування) і застосування на основі параметрів квантування зворотного квантування до

декодованих квантованих коефіцієнтів 309 для отримання деквантованих коефіцієнтів 311, які також можуть називатися коефіцієнтами 311 перетворення. Процес зворотного квантування може включати використання параметра квантування, визначеного відеокодером 20 для кожного відеоблока у відеослайсі (або комірці або групі комірок), для визначення ступеня квантування  $i$ , аналогічно, ступеня зворотного квантування, які повинні бути застосовані.

#### Зворотне перетворення

Модуль 312 обробки зворотного перетворення може бути виконаний з можливістю прийому деквантованих коефіцієнтів 311, які також називають коефіцієнтами 311 перетворення, і застосування перетворення до деквантованих коефіцієнтів 311 для отримання відновлених залишкових блоків 213 в зоні дискретних відліків. Відновлені залишкові блоки 213 також можуть називатися блоками 313 перетворення. Перетворення може бути зворотним перетворенням, наприклад, зворотним DCT, зворотним DST, зворотним цілочисловим перетворенням або принципово аналогічним процесом зворотного перетворення. Модуль 312 обробки зворотного перетворення може бути додатково виконаний з можливістю прийому параметрів перетворення або відповідної інформації з даних 21 кодового зображення (наприклад, за допомогою синтаксичного аналізу і/або декодування, наприклад, за допомогою блока 304 ентропійного декодування) для визначення перетворення, яке необхідно застосовувати до деквантованих коефіцієнтів 311.

#### Відновлення

Модуль 314 відновлення (наприклад, суматор 314) може бути виконаний з можливістю додавання відновленого залишкового блока 313 до блока 365 прогнозування для отримання відновленого блока 315 в зоні дискретних відліків, наприклад, за допомогою додавання значень дискретних відліків відновленого залишкового блока 313 і значень дискретних відліків блока 365 прогнозування.

#### Фільтрування

Модуль 320 контурного фільтра (або в контурі кодування, або після контуру кодування) виконаний з можливістю фільтрування відновленого блока 315 для отримання відфільтрованого блока 321, наприклад, для згладжування переходів пікселів або іншого покращення якості відео. Модуль 320 контурного фільтра може містити один або більше контурних фільтрів, таких як фільтр зняття блоковості, фільтр з адаптивним зміщенням дискретних відліків (SAO), або один або більше інших фільтрів, таких як двосторонній фільтр, адаптивний контурний фільтр (ALF), фільтри згладжування, підвищення різкості або спільні фільтри, або будь-яку їхню комбінацію. Хоча модуль 320 контурного фільтра показаний на фіг. 3 як контурний фільтр, в інших конфігураціях модуль 320 контурного фільтра може бути реалізований як постконтурний фільтр.

#### Буфер декодованих зображень

Декодовані відеоблоки 321 зображення потім зберігаються в буфері 330 декодованих зображень, який зберігає декодовані зображення 331 як опорні зображення для подальшої компенсації руху для інших зображень і/або для виведення, відповідно, відображення.

Декодер 30 виконаний з можливістю виведення декодованого зображення 311, наприклад, через вихід 312 для представлення або перегляду користувачем.

#### Прогнозування

Модуль 344 інтер-прогнозування може бути ідентичний модулю 244 інтер-прогнозування (зокрема, модулю компенсації руху), а модуль 354 інтра-прогнозування може бути ідентичний модулю 254 інтер-прогнозування за функцією, він приймає рішення щодо розбивання або розділення і виконує прогнозування на основі параметрів розбивання і/або прогнозування або відповідної інформації, яка приймається з кодованих даних 21 зображення (наприклад, за допомогою аналізу і/або декодування, наприклад, за допомогою блока 304 ентропійного декодування). Модуль 360 застосування режиму може бути виконаний з можливістю здійснення прогнозування (інтра- або інтер-прогнозування) для кожного блока на основі відновлених зображень, блоків або відповідних дискретних відліків (фільтрованих або нефільтрованих) для отримання блока 365 прогнозування.

Коли відеослайс кодується як слайс з інтра-кодуванням (I), модуль 354 інтра-прогнозування модуля 360 застосування режиму конфігурується для генерації блока 365 прогнозування для блока зображення поточного відеослайса на основі визначеного режиму інтра-прогнозування і даних із попередньо декодованих блоків поточного зображення. Коли відеозображення кодується як слайс, закодований інтер-кодуванням (тобто B або P), модуль 344 інтер-прогнозування (наприклад, модуль компенсації руху) модуля 360 застосування режиму конфігурується для створення блоків 365 прогнозування для відеоблока поточного відео слайса на основі векторів руху і інших елементів синтаксису, отриманих від модуля 304 ентропійного

декодування. Для інтер-прогнозування блоки прогнозування можуть бути створені з одного із опорних зображень в одному зі списків опорних зображень. Відеодекодер 30 може створювати списки опорних кадрів, список 0 і список 1, використовуючи способи побудови за замовчуванням на основі опорних зображень, які зберігаються в DPB 330. Те саме або подібне

5 може бути застосоване для або за допомогою варіантів здійснення з використанням груп комірок (наприклад, груп відеокомірок) і/або комірок (наприклад, відеокомірок) на додаток або альтернативно до слайсів (наприклад, відеослайсів), наприклад, відео може бути кодоване з використанням груп I, P або B комірок і/або комірок.

Модуль 360 застосування режиму виконаний з можливістю визначення інформації прогнозування для відеоблока поточного відеослайса за допомогою синтаксичного аналізу векторів руху або пов'язаної інформації і інших елементів синтаксису, і використання інформації прогнозування для створення блоків прогнозування для поточного декодованого відеоблока. Так, наприклад, модуль 360 застосування режиму використовує деякі з прийнятих елементів синтаксису для визначення режиму прогнозування (наприклад, інтра- або інтер-прогнозування),

15 що використовується для кодування відеоблоків відеослайса, типу слайса інтер-прогнозування (наприклад, B-слайс, P-слайс або GPB-слайс), інформацію про конструкцію для одного або більше списків опорних зображень для слайса, вектори руху для кожного відеоблока з інтер-кодуванням слайса, статус інтер-прогнозування для кожного інтер-кодованого відеоблока слайса і іншу інформацію для декодування відеоблоків у поточному відеослайсі. Те саме або

20 подібне може бути застосоване для варіантів здійснення або за допомогою варіантів здійснення з використанням груп комірок (наприклад, груп відеокомірок) і/або комірок (наприклад, відеокомірок) на додаток або альтернативно до слайсів (наприклад, відеослайсів), наприклад, відео може бути кодоване з використанням груп I, P або B комірок і/або комірок.

Варіанти здійснення відеодекодера 30, як показано на фіг. 3, можуть бути виконані з

25 можливістю розбивання і/або декодування зображення з використанням слайсів (які також називають відео слайсами), при цьому зображення може бути розділене на або декодоване з використанням одного або більше слайсів (зазвичай без перекриття), і кожний слайс може містити один або більше блоків (наприклад, STU).

Варіанти здійснення відеодекодера 30, як показано на фіг. 3, можуть бути виконані з

30 можливістю розбивання і/або декодування зображення з використанням груп комірок (які також називають групами відеокомірок) і/або комірок (які також називають відеокомірками), при цьому зображення може бути розділене на або декодоване з використанням однієї або більше груп комірок (які зазвичай не перекриваються), і кожна група комірок може містити, наприклад, один або більше блоків (наприклад, STU) або одну або більше комірок, при цьому кожна комірка,

35 наприклад, може мати прямокутну форму і може містити один або більше блоків (наприклад, STU), наприклад, повні або дробові блоки.

Інші варіанти відеодекодера 30 можуть бути використані для декодування даних 21 закодованого зображення. Так, наприклад, декодер 30 може створювати вихідний відеопотік без блока 320 контурного фільтрування. Наприклад, декодер 30, не заснований на перетворенні,

40 може виконувати зворотне квантування залишкового сигналу напряму без модуля 312 обробки зворотного перетворення для певних блоків або кадрів. В іншій реалізації відеодекодер 30 може мати модуль 310 зворотного квантування і модуль 312 обробки зворотного перетворення, об'єднані в один модуль.

Слід розуміти, що в кодері 20 і декодері 30 результат обробки поточного етапу може бути

45 додатково оброблений, а потім виведений на наступний етап. Так, наприклад, після інтерполяційного фільтрування, виведення руху або контурного фільтрування може бути виконана додаткова операція, така як обрізання або зсув, щодо результату обробки інтерполяційним фільтруванням, виведенням руху або контурним фільтруванням.

Слід зазначити, що до виведених векторів руху поточного блока можуть бути застосовані

50 додаткові операції (в тому числі, але не обмежуючись ними, до векторів руху контрольної точки афінного режиму, векторів руху підблока в афінному, планарному режимах, режимі ATMVP, часових векторів руху тощо). Так, наприклад, значення вектора руху обмежується попередньо визначеним діапазоном відповідно до його бітового представлення. Якщо бітовим представленням вектора руху є  $\text{bitDepth}$  (бітова глибина), тоді діапазон становить  $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}-1$ , де « $\wedge$ » означає піднесення до степені. Так, наприклад, якщо значення  $\text{bitDepth}$  дорівнює 16, то діапазон становить  $-32768 \sim 32767$ ; якщо  $\text{bitDepth}$  дорівнює 18, то діапазон становить  $-131072 \sim 131071$ . Так, наприклад, значення виведеного вектора руху (наприклад, MV чотирьох підблоків  $4 \times 4$  в межах одного блока  $8 \times 8$ ) обмежується так, щоб

55 максимальна різниця між цілими частинами MV чотирьох підблоків  $4 \times 4$  не перевищувала N

пікселів, наприклад, була не більше ніж 1 піксель. Нижче представлені два способи обмеження вектора руху згідно з bitDepth.

Спосіб 1: видалення переповнення MSB (найстаршого біта) за допомогою таких операцій:

$ux=(mvx+2^{bitDepth}) \% 2^{bitDepth}$	(1)
$mvx=(ux \geq 2^{bitDepth-1})? (ux-2^{bitDepth}): ux$	(2)
$uy=(mvy+2^{bitDepth}) \% 2^{bitDepth}$	(3)
$mvy=(uy \geq 2^{bitDepth-1})? (uy-2^{bitDepth}): uy$	(4)

5

де mvx є горизонтальним компонентом вектора руху блока зображення або підблока, mvy є вертикальним компонентом вектора руху блока зображення або підблока, а ux і uy позначають проміжне значення;

10 Так, наприклад, якщо значення mvx дорівнює -32769, після застосування формул (1) і (2) отримане значення дорівнює 32767. У комп'ютерній системі десяткові числа зберігаються як додатковий двійковий код. Додатковий двійковий код числа -32769 дорівнює 1,0111,1111,1111,1111 (17 біт), тоді MSB відкидається, отже, отриманий додатковий двійковий код числа дорівнює 0111,1111,1111,1111 (десятьковим числом є 32767), що співпадає з вихідними даними при застосуванні формул (1) і (2).

15

$ux=(mvpx+mvd_x+2^{bitDepth}) \% 2^{bitDepth}$	(5)
$mvx=(ux \geq 2^{bitDepth-1})? (ux-2^{bitDepth}): ux$	(6)
$uy=(mvpy+mvd_y+2^{bitDepth}) \% 2^{bitDepth}$	(7)
$mvy=(uy \geq 2^{bitDepth-1})? (uy-2^{bitDepth}): uy$	(8)

Операції можуть бути застосовані при додаванні mvr до mvd, як показано у формулах (5) - (8).

Спосіб 2: видалення переповнення MSB шляхом відсікання значення

20

$$vx=Clip3(-2^{bitDepth-1}, 2^{bitDepth-1}-1, vx)$$

$$vy=Clip3(-2^{bitDepth-1}, 2^{bitDepth-1}-1, vy)$$

де vx є горизонтальним компонентом вектора руху блока зображення або підблока, vy є вертикальним компонентом вектора руху блока зображення або підблока; x, y і z відповідно відповідають трьом вхідним значенням процесу відсікання MV, а визначення функції Clip3 є

25

$$Clip3(x, y, z) = \begin{cases} x, & z < x \\ y, & z > y \\ z, & \text{інше} \end{cases}$$

На фіг. 4 представлена блок-схема пристрою 400 відеокодування згідно з варіантом здійснення винаходу. Пристрій 400 відеокодування придатний для реалізації розкритих варіантів здійснення, як описано в даному документі. У варіанті здійснення пристрій 400 відеокодування може бути декодером, таким як відеодекодер 30 з фіг. 1А, або кодером, таким як відеокодер 20 з фіг. 1А.

30

Пристрій 400 відеокодування містить вхідні порти 410 (або порти 410 введення) і модулі 420 приймача (Rx) для прийому даних; процесор, логічний модуль або центральний процесор (CPU) 430 для обробки даних; модулі 440 передавача (Tx) і вихідні порти 450 (або порти 450 виведення) для передачі даних; і пам'ять 460 для зберігання даних. Пристрій 400 відеокодування може також містити оптико-електричні (OE) компоненти і електрично-оптичні (EO) компоненти, підключені до вхідних портів 410, модулів 420 приймача, модулів 440 передавача і вихідних портів 450 для виходу або входу оптичних або електричних сигналів.

35

Процесор 430 може бути реалізований апаратно і програмно. Процесор 430 може бути реалізований як одна або більше мікросхем ЦП (CPU), ядра (наприклад, як багатоядерний процесор), FPGA, ASIC і DSP. Процесор 430 підтримує зв'язок із вхідними портами 410, модулями 420 приймача, модулями 440 передавача, вихідними портами 450 і пам'яттю 460. Процесор 430 містить модуль 470 кодування. Модуль 470 кодування реалізує розкриті варіанти здійснення, описані вище. Так, наприклад, модуль 470 кодування реалізує, обробляє, готує або надає різні операції кодування. Отже, включення модуля 470 кодування забезпечує істотне покращення функціональних можливостей пристрою 400 відеокодування і забезпечує переведення пристрою 400 відеокодування в інший стан. Як альтернатива модуль 470 кодування реалізований як інструкції, які зберігаються в пам'яті 460 і виконуються процесором 430.

45

Пам'ять 460 може містити один або більше дисків, стрічкових накопичувачів і твердотільних накопичувачів і може бути використана як пристрій зберігання даних з переповненням, для зберігання програм, коли такі програми вибрані для виконання, і для зберігання інструкцій і даних, які зчитуються під час виконання програми. Пам'ять 460 може, наприклад, бути енергозалежною і/або енергонезалежною, а також може бути постійною пам'яттю (ROM), оперативною пам'яттю (RAM), трійковою пам'яттю з адресацією за вмістом (TCAM) і/або статичною пам'яттю з довільним доступом (SRAM)).

На фіг. 5 представлена спрощена блок-схема пристрою 500, який може бути використаний як одне або обидва з пристрою-джерела 12 і пристрою-одержувача 14 за фіг. 1 згідно з прикладом варіанта здійснення.

Процесор 502 в пристрої 500 може бути центральним процесором. Альтернативно процесор 502 може бути пристроєм будь-якого іншого типу або множиною пристроїв, здатних керувати або обробляти інформацію, яка існує в цей момент або буде розроблена в майбутньому. Хоча описані варіанти здійснення можуть бути реалізовані з використанням одного процесора, як показано, наприклад, процесора 502, переваги у швидкості і ефективності можуть бути досягнуті з використанням понад одного процесора.

У варіанті реалізації пам'ять 504 в пристрої 500 може бути постійним запам'ятовувальним пристроєм (ПЗП, ROM) або пристроєм оперативної пам'яті (ОЗП, RAM) в реалізації. Як пам'ять 504 може бути використаний будь-який інший придатний тип запам'ятовувального пристрою. Пам'ять 504 може містити код і дані 506, до яких звертається процесор 502, використовуючи канал 512. Пам'ять 504 може додатково містити операційну систему 508 і прикладні програми 510, а прикладні програми 510 містять щонайменше одну програму, яка дозволяє процесору 502 здійснювати описані у даному винаході способи. Так, наприклад, прикладні програми 510 можуть містити застосунки з 1 по N, які додатково містять застосунок кодування відео, який здійснює описані у даному винаході способи.

Пристрій 500 також може містити один або більше пристроїв виведення, наприклад дисплей 518. Дисплей 518 може бути, в одному прикладі, сенсорним дисплеєм, який поєднує дисплей з сенсорним елементом, здатним сприймати сенсорне введення (дотик). Дисплей 518 може бути з'єднаний з процесором 502 через канал 512.

Хоч у даному випадку йдеться про один канал, канал 512 пристрою 500 може складатися із множини каналів. Крім того, вторинне сховище 514 може бути безпосередньо з'єднане з іншими компонентами пристрою 500 або може бути доступне через мережу і може містити один вбудований блок, такий як карта пам'яті, або множину блоків, таких як множина карт пам'яті. Отже, пристрій 500 може бути реалізований в найрізноманітніших конфігураціях.

Вихідні дані про кодування режиму інтра-прогнозування

У поточній версії VTM 4.0 для кодування режиму інтра-прогнозування створюється список з 6-MPM (найбільш імовірних режимів). Список MPM вводиться для зменшення кількості бітів, необхідних для кодування режиму інтра-прогнозування поточного блока. Список 6-MPM будується на основі режимів інтра-прогнозування сусідніх блоків поточного блока. Коли режим інтра-прогнозування поточного блока потрапляє в список MPM, кодується індекс, а не фактичний режим, який може використовувати менше бітів. Коли режим інтра-прогнозування поточного блока не потрапляє в список MPM, для кодування режиму інтра-прогнозування поточного блока використовується усічена бінаризація. Список MPM забезпечує хороше прогнозування для режиму інтра-прогнозування поточного блока.

Список MPM для множинного опорного рядка (MRL)

У VTM 4.0 інструмент кодування з множинними опорними рядками (MRL) може використовувати один із множини сусідніх опорних рядків для прогнозування дискретних відліків поточного блока. Коли значення індексу рядка MRL дорівнює 0 (тобто використовується найближчий сусідній опорний рядок, як показано на фіг. 6), використовується звичайний список з 6 MPM, який включає планарний режим і режим DC. Коли значення індексу рядка MRL не дорівнює 0, використовується список з 6-MPM, за винятком планарного режиму (тобто значення 0) і режиму DC (тобто значення 1).

Список MPM для режиму кодування всередині підрозділу (ISP)

Режим кодування всередині підрозділу (ISP) – це інструмент, нещодавно прийнятий в VTM4.0 (JVET-M0102). Блоки з інтра-прогнозуванням яскравості діляться по вертикалі або горизонталі на 2 або 4 частини, залежно від розмірів блока, як показано в таблиці 1. На фіг. 7 і 8 показані приклади двох можливостей. Всі підрозділи задовольняють умові наявності не менше ніж 16 дискретних відліків.

Таблиця 1

Кількість підрозділів залежно від розміру блока

Розмір блока	Кількість підрозділів
4×4	Не розділений
4×8 і 8×4	2
Всі інші випадки	4

5 Коли включений режим кодування всередині підрозділу, в VTM 4.0 використовується інший список MPM, в якому виключений режим DC. Режим всередині підрозділу може бути застосований, коли значення індексу множинного опорного рядка дорівнює 0 (тобто MRL не застосовується до поточного блока інтра-прогнозування). Всі підрозділи використовують один режим інтра-прогнозування, тому список MPM створюється один раз для інтра-блока і спільно використовується усіма підрозділами.

10 Побудова MPM може залежати від режиму розбивання ISP. Визначені два режими розбивання: горизонтальний і вертикальний. Вони показані на фіг. 7 і 8, де горизонтальне розділення/розбивання показане вгорі, а вертикальне розділення/розбивання показане внизу.

У наведеній нижче таблиці 2 вказані характеристики списку 3 MPM, використовуваного в VTM 4.0.

Таблиця 2

різні 6-MPM, використовувані в VTM 4.0 за різних обставин

Конфігурація MRL	Конфігурація ISP	Характеристика списку MPM
MRL! = 0	-	6 MPM, за винятком планарного режиму і режиму DC
MRL = = 0	1	6 MPM, за винятком режиму DC
MRL = = 0	0	6 MPM, включаючи планарний режим і режим DC

15

Вихідні дані про контекстне моделювання/кодування CABAC

20 Як показано на фіг. 9, кодування CABAC складається з бінаризації, контекстного моделювання і двійкового арифметичного кодування. Бінаризація відображає елементи синтаксису в двійкові символи (елементи дискретизації, біни). Контекстне моделювання оцінює імовірність кожного непропущеного (тобто регулярно кодованого) біна на основі певного конкретного контексту. Зрештою, двійкове арифметичне кодування стискає біни в біти відповідно до оцінної імовірності.

25 За допомогою розкладання кожного недвійкового значення елемента синтаксису в послідовність бінів, подальша обробка кожного значення біна в CABAC залежить від відповідного рішення щодо режиму кодування, який може бути вибраний як регулярний або режим з пропуском. Останній вибирається для бінів, які, як передбачається, розподілені рівномірно і для яких, відповідно, весь процес регулярного двійкового арифметичного кодування (і декодування) просто пропускається. У режимі регулярного кодування кожне значення біна кодується з використанням механізму звичайного двійкового арифметичного кодування, де відповідна імовірнісна модель визначається або за допомогою фіксованого вибору, на основі типу елемента синтаксису і позиції біна або індексу біна (binIdx) в бінаризованому представленні елемента синтаксису, або адаптивно вибирається з двох або більше імовірнісних моделей залежно від відповідної додаткової інформації.

35 Вибір імовірнісної моделі називається контекстним моделюванням. Кожна імовірнісна модель в CABAC адресується з використанням унікального індексу контексту (ctxIdx), який визначається за допомогою фіксованого присвоєння або обчислюється логікою виведення контексту, за допомогою якої, в свою чергу, задається дана контекстна модель.

Інформація щодо можливості визначення сусідів зліва і зверху

40 Кодування режиму інтра-прогнозування щодо VVC залежить від режимів інтра-прогнозування його сусідніх блоків, а саме лівого і верхнього блоків поточного блока. Їхнє положення показане на фіг. 10.

Сигналізування індексу MPM

Мітка (тобто мітка MPM) використовується, щоб вказувати, чи знаходиться режим інтра-

прогнозування поточного блока в списку MPM, чи ні. Коли мітка MPM має значення "істина" (тобто значення 1), режим інтра-прогнозування може бути визначений з використанням індексу MPM. Індекс MPM кодується з використанням усіченого унарного коду, як показаний в наведеній нижче таблиці 3, коли довжина списку MPM дорівнює 6. Коли мітка MPM – не істина, режим інтра-прогнозування поточного блока відсутній в списку MPM, і режим кодується з використанням усіченого двійкового коду. Інший приклад індексу MPM, коли список MPM складається з 5 режимів, показаний в таблиці 4. Рядок бінів для індексу кодування називається усіченим унарним кодом (неусіченим двійковим кодом для кодування режимів без MPM). Слід звернути увагу, що для усіченого унарного коду біни 0 і 1 взаємозамінні без втрати спільності. В одному прикладі індекс MPM, коли список MPM складається з 5 режимів, також може бути закодований, як показано в таблиці 5.

Таблиця 3

Кодування мітки MPM і індексу MPM при 6 записах в списку MPM

Режими інтра-прогнозування	Мітка MPM	Індекс MPM	Рядок бінів індексу MPM
Режими MPM (6)	1	0	0
		1	10
		2	110
		3	1110
		4	11110
		5	11111
	0		Усічений двійковий код (ТВ)

Таблиця 4

Кодування мітки MPM і індексу MPM, коли списком MPM містить 5 записів

Режими інтра-прогнозування	Мітка MPM	Індекс MPM	Рядок бінів індексу MPM
Режими MPM (5)	1	0	0
		1	10
		2	110
		3	1110
		4	1111
	0		Усічений двійковий код (ТВ)

Таблиця 5

Кодування мітки MPM і індексу MPM, коли в списку MPM 5 записів, і альтернативне кодове слово для індексу MPM

Режими інтра-прогнозування	Мітка MPM	Індекс MPM	Рядок бінів індексу MPM
Режими MPM (5)	1	0	1
		1	01
		2	001
		3	0001
		4	0000
	0		Усічений двійковий код (ТВ)

15

Вихідні дані про планарний режим, режим DC, горизонтальний, вертикальний режими і кутові режими: Режими інтра-прогнозування: планарний (значення 0) і DC (значення 1). Інші режими інтра-прогнозування називаються кутовими режимами прогнозування.

У VTM4.0 використовується така структура синтаксису, як показано в таблиці 6, для передачі інформації про режим інтер-прогнозування.

Таблиця 6

Розбір синтаксису в VTM4.0

якщо (intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
якщо (intra_luma_mpm_flag[x0][y0]) {	
intra_luma_mpm_idx[x0][y0]	ae(v)
}	
інакше	
intra_luma_mpm_remainder[x0][y0]	ae(v)

5 intra\_luma\_ref\_idx[x0][y0] може мати три значення: 0, 1 або 2, які вказують, який опорний рядок використовується. Коли intra\_luma\_ref\_idx[x0][y0] відсутній, передбачається, що він дорівнює 0.

intra\_subpartitions\_mode\_flag[x0][y0], що дорівнює 1, вказує, що поточна одиниця інтра-кодування додатково розділена (тобто включений режим ISP).  
 10 intra\_subpartitions\_mode\_flag[x0][y0], що дорівнює 0, вказує, що поточна одиниця інтра-кодування не розділена на прямокутні підрозділи блока перетворення. Коли intra\_subpartitions\_mode\_flag[x0][y0] відсутній, передбачається, що він дорівнює 0.

Елементи синтаксису intra\_luma\_mpm\_flag[x0][y0], intra\_luma\_mpm\_idx[x0][y0] і  
 15 intra\_luma\_mpm\_remainder[x0][y0] визначають режим інтра-прогнозування для дискретних відліків яскравості.

intra\_luma\_mpm\_flag[x0][y0] вказує на те, чи отриманий режим інтра-прогнозування поточного блока зі списку MPM, чи ні. Якщо значення intra\_luma\_mpm\_flag[x0][y0] дорівнює 1, це вказує на те, що режим інтра-прогнозування поточного блока отриманий зі списку MPM; Якщо значення intra\_luma\_mpm\_flag[x0][y0] дорівнює 0, це вказує, що режим інтра-прогнозування  
 20 поточного блока не отриманий зі списку MPM. Коли intra\_luma\_mpm\_flag[x0][y0] відсутній, передбачається, що він дорівнює 1. intra\_luma\_mpm\_flag[x0][y0] відповідає мітці MPM в таблиці 3.

intra\_luma\_mpm\_idx відповідає індексу MPM в таблиці 3 і використовує усічений унарний код, як показано в таблиці 3.

25 intra\_luma\_mpm\_remainder відображає усічений двійковий код режиму, який відсутній в списку MPM.

У варіанті 1 здійснення розкритий елемент синтаксису intra\_luma\_planar\_flag і ctx.

Згідно з одним варіантом здійснення винаходу вводиться мітка intra\_luma\_planar\_flag. Якщо значення intra\_luma\_mpm\_flag – істина (наприклад, значення дорівнює 1), декодер шляхом  
 30 аналізу отримує intra\_luma\_planar\_flag, щоб визначити, чи є режим інтра-прогнозування поточного блока планарним режимом, чи ні. Коли значення intra\_luma\_planar\_flag дорівнює 1, режим інтра-прогнозування поточного блока є планарним режимом. Коли значення intra\_luma\_planar\_flag дорівнює 0, режим інтра-прогнозування поточного блока не є планарним режимом. Якщо intra\_luma\_planar\_flag відсутній (наприклад, ця мітка не може бути отримана  
 35 аналізом з потоку бітів у випадку, якщо intra\_luma\_mpm\_flag дорівнює 0), значення intra\_luma\_planar\_flag виводиться як 0.

При написанні в стилі специфікації робочого проекту VVC, режим інтра-прогнозування поточного блока

IntraPredModeY[xCb][yCb] виводиться шляхом застосування такої процедури:

- 40 - якщо intra\_luma\_mpm\_flag[xCb][yCb] дорівнює 1,
- якщо intra\_luma\_planar\_flag[xCb][yCb] дорівнює 1,
- intraPredModeY[xCb][yCb] встановлюється таким, що дорівнює INTRA\_PLANAR.
- В іншому випадку IntraPredModeY[xCb][yCb] встановлюється таким, що дорівнює sandModeList[intra\_luma\_mpm\_idx[xCb][yCb]].

45 Якщо режим інтра-прогнозування не є планарним режимом (тобто значення intra\_luma\_planar\_flag дорівнює 0), декодер додатково аналізує елемент синтаксису intra\_luma\_mpm\_idx.

В одному прикладі значення `intra_luma_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодоване кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок звичайним інтра-блоком, чи ні.

5 В одному прикладі значення `intra_luma_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодовано кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком із включеним множинним опорним рядком (MRL), чи ні.

10 В одному прикладі значення `intra_luma_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодовано кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком із включеним режимом кодування всередині підрозділу (ISP), чи ні.

У прикладі позиція `intra_luma_planar_flag` розміщена, як показано в таблиці 7.

Таблиця 7

синтаксис з `intra_luma_planar_flag`

якщо ( <code>intra_luma_ref_idx[x0][y0] == 0 &amp;&amp; intra_subpartitions_mode_flag[x0][y0] == 0</code> )	
<code>intra_luma_mpm_flag[x0][y0]</code>	<code>ae(v)</code>
якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {	
<code>intra_luma_planar_flag[x0][y0]</code>	<code>ae(v)</code>
якщо ( <code>!intra_luma_planar_flag[x0][y0]</code> )	
<code>intra_luma_mpm_idx[x0][y0]</code>	<code>ae(v)</code>
}	
інакше	
<code>intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>

15 Варіант здійснення 2: `intra_luma_not_planar_flag` і `ctx`

Згідно з одним варіантом здійснення винаходу вводиться елемент синтаксису `intra_luma_not_planar_flag`.

20 Якщо значення `intra_luma_mpm_flag` – істина (наприклад, значення дорівнює 1), декодер аналізує `intra_luma_not_planar_flag`, щоб визначити, чи є режим інтра-прогнозування поточного блока планарним режимом, чи ні. Коли значення `intra_luma_not_planar_flag` дорівнює 1, режим інтра-прогнозування поточного блока не є планарним режимом; коли значення `intra_luma_not_planar_flag` дорівнює 0, режим інтра-прогнозування поточного блока є планарним режимом. Якщо `intra_luma_not_planar_flag` відсутній в потоці бітів, значення `intra_luma_not_planar_flag` виводиться як 1. У прикладі позиція `intra_luma_not_planar_flag`

25 розміщена, як показано в таблиці 8.

Якщо режим інтра-прогнозування поточного блока не є планарним режимом (тобто значення `intra_luma_not_planar_flag` дорівнює 1), декодер додатково аналізує елемент синтаксису `intra_luma_mpm_idx`.

30 При написанні в стилі специфікації робочого проекту VVC, режим інтра-прогнозування поточного блока

`IntraPredModeY[xCb][yCb]` виводиться шляхом застосування такої процедури:

- якщо `intra_luma_mpm_flag[xCb][yCb]` дорівнює 1,

- якщо `intra_luma_not_planar_flag[xCb][yCb]` дорівнює,

`intraPredModeY[xCb][yCb]` встановлюється таким, що дорівнює `INTRA_PLANAR`.

35 - В іншому випадку `IntraPredModeY[xCb][yCb]` встановлюється таким, що дорівнює `sandModeList[intra_luma_mpm_idx[xCb][yCb]]`.

В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодовано кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок звичайним інтра-блоком, чи ні.

40 В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодовано кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком із включеним множинним опорним рядком (MRL), чи ні.

45 В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодовано кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком із включеним режимом кодування всередині підрозділу (ISP), чи ні.

Таблиця 8

синтаксис з `intra_luma_not_planar_flag`

якщо ( <code>intra_luma_ref_idx[x0][y0] == 0 &amp;&amp; intra_subpartitions_mode_flag[x0][y0] == 0</code> )	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {	
<code>intra_luma_not_planar_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_not_planar_flag[x0][y0]</code> )	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
інакше	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

Варіант здійснення 3: `intra_luma_planar_flag` і MRL, і декодування з пропуском

Згідно з одним варіантом здійснення винаходу вводиться мітка `intra_luma_planar_flag`.

5 Якщо значення `intra_luma_mpm_flag` – істина (наприклад, значення дорівнює 1), декодер аналізує, чи дорівнює значення індексу опорного рядка (`intra_luma_ref_idx`) 0, чи ні. Якщо значення індексу опорного рядка дорівнює 0, значення `intra_luma_planar_flag` аналізується, щоб визначити, чи є режим інтра-прогнозування поточного блока планарним режимом, чи ні. Коли значення `intra_luma_planar_flag` дорівнює 1, режим інтра-прогнозування поточного блока є планарним режимом. Коли значення `intra_luma_planar_flag` дорівнює 0, режим інтра-прогнозування поточного блока не є планарним режимом. Якщо `intra_luma_planar_flag` відсутній, значення `intra_luma_planar_flag` виводиться як 0.

Якщо режим інтра-прогнозування поточного блока не є планарним режимом (тобто значення `intra_luma_planar_flag` дорівнює 0), декодер додатково аналізує `intra_luma_mpm_idx`.

15 В одному прикладі значення `intra_luma_planar_flag` закодоване кодуванням САВАС з пропуском (тобто не закодоване контекстним кодуванням САВАС), і декодер аналізує мітку без використання будь-якого контексту САВАС.

Позиція `intra_luma_planar_flag` розміщена, як показано в Таблиці 9.

Таблиця 9

синтаксис з `intra_luma_planar_flag`, і він закодований з пропуском

якщо ( <code>intra_luma_ref_idx[x0][y0] == 0 &amp;&amp; intra_subpartitions_mode_flag[x0][y0] == 0</code> )	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {	
якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> )	
<code>intra_luma_planar_flag[x0][y0]</code>	ae(v)
якщо ( <code>!intra_luma_planar_flag[x0][y0]</code> )	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
інакше	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

20 Варіант 4 здійснення: `intra_luma_not_planar_flag` і MRL, і декодування з пропуском

Згідно з одним варіантом здійснення винаходу вводиться мітка `intra_luma_not_planar_flag`.

25 Якщо значення `intra_luma_mpm_flag` – істина (наприклад, значення дорівнює 1), декодер аналізує, чи дорівнює значення індексу опорного рядка 0, чи ні. Якщо значення індексу опорного рядка дорівнює 0, `intra_luma_not_planar_flag` аналізується, щоб визначити, чи є режим інтра-прогнозування поточного блока планарним режимом, чи ні. Коли значення `intra_luma_not_planar_flag` дорівнює 1, режим інтра-прогнозування поточного блока не є планарним режимом. Коли значення `intra_luma_planar_flag` дорівнює 0, режим інтра-прогнозування поточного блока є планарним режимом. Якщо `intra_luma_not_planar_flag` відсутній, значення `intra_luma_not_planar_flag` виводиться як 1.

30 Якщо режим інтра-прогнозування поточного блока не є планарним режимом (тобто значення `intra_luma_not_planar_flag` дорівнює 1), декодер додатково аналізує `intra_luma_mpm_idx`.

В одному прикладі значення `intra_luma_not_planar_flag` закодоване кодуванням САВАС з пропуском (тобто не закодоване контекстним кодуванням САВАС), і декодер аналізує мітку без використання будь-якого контексту САВАС.

Позиція `intra_not_luma_planar_flag` розміщена, як показано в таблиці 10.

Таблиця 10

синтаксис з `intra_luma_not_planar_flag`, і він закодований з пропуском

якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> && <code>intra_subpartitions_mode_flag[x0][y0] == 0</code> )	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {	
якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> )	
<code>intra_luma_not_planar_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_not_planar_flag[x0][y0]</code> )	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
інакше	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

5

Варіант здійснення 5: `intra_luma_not_planar_flag` і MRL, і контекстне кодування САВАС (відмінність від варіанта здійснення 4 полягає в тому, що `intra_luma_not_planar_flag` кодується контекстним кодуванням САВАС замість кодування з пропуском).

Згідно з одним варіантом здійснення винаходу вводиться мітка `intra_luma_not_planar_flag`.

10

Якщо значення `intra_luma_mpm_flag` – істина (наприклад, значення дорівнює 1), декодер аналізує, чи дорівнює значення індексу опорного рядка 0, чи ні. Якщо значення індексу опорного рядка дорівнює 0, `intra_luma_not_planar_flag` аналізується, щоб визначити, чи є режим інтра-прогнозування поточного блока планарним режимом, чи ні. Коли значення `intra_luma_not_planar_flag` дорівнює 1, режим інтра-прогнозування поточного блока НЕ є планарним режимом. Коли значення `intra_luma_planar_flag` дорівнює 0, режим інтра-прогнозування поточного блока є планарним режимом. Якщо `intra_luma_not_planar_flag` відсутній, значення `intra_luma_not_planar_flag` виводиться як 1.

15

Якщо режим інтра-прогнозування поточного блока не є планарним режимом (тобто значення `intra_luma_not_planar_flag` дорівнює 1), декодер додатково аналізує `intra_luma_mpm_idx`.

20

В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодоване кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок звичайним інтра-блоком, чи ні.

25

В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодоване кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком з включеним множинним опорним рядком (MRL), чи ні.

30

В одному прикладі значення `intra_luma_not_planar_flag` закодоване контекстним кодуванням САВАС (тобто не закодоване кодуванням САВАС з пропуском), і декодер аналізує мітку з використанням контексту на основі того, чи є поточний блок інтра-блоком з включеним режимом кодування всередині підрозділу (ISP), чи ні.

Позиція `intra_not_luma_planar_flag` розміщена, як показано в Таблиці 11.

Таблиця 11

синтаксис з `intra_luma_not_planar_flag`, і він закодований контекстним кодуванням САВАС

якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> && <code>intra_subpartitions_mode_flag[x0][y0] == 0</code> )	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {	
якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> )	
<code>intra_luma_not_planar_flag[x0][y0]</code>	ae(v)
якщо ( <code>intra_luma_not_planar_flag[x0][y0]</code> )	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
інакше	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

У прикладі,  
Додати `intra_luma_not_planar_flag`

якщо ( <code>intra_luma_mpm_flag[x0][y0]</code> ) {		
якщо ( <code>intra_luma_ref_idx[x0][y0] == 0</code> )		
<code>intra_luma_not_planar_flag[x0][y0]</code>		<code>ae(v)</code>
якщо ( <code>intra_luma_not_planar_flag[x0][y0]</code> )		
<code>intra_luma_mpm_idx[x0][y0]</code>		<code>ae(v)</code>
} інакше		
<code>intra_luma_mpm_remainder[x0][y0]</code>		<code>ae(v)</code>
}		
<code>intra_luma_mpm_flag [] []</code>	FL	<code>cMax=1</code>
<code>intra_luma_not_planar_flag [] []</code>	FL	<code>cMax=1</code>
<code>intra_luma_mpm_idx [] []</code>	Tr	<code>cMax=4, cRiceParam=0</code>

5

Коли `intra_luma_not_planar_flag[x0][y0]` відсутній, передбачається, що він дорівнює 1.  
У наведеній нижче Таблиці 12 показаний приклад присвоєння `ctxInc` елементам синтаксису з контекстно-кодованими бінами.

Таблиця 12

Присвоєння `ctxInc` елементам синтаксису с контекстно-кодованими бінами

Елемент синтаксису	binIdx					
	0	1	2	3	4	>=5
<code>intra_subpartition_split_flag</code>	0	нд	нд	нд	нд	нд
<code>intra_luma_mpm_flag [] []</code>	0	нд	нд	нд	нд	нд
<code>intra_luma_not_planar_flag [] []</code>	<code>!intra_subpartitions_mode_flag</code>	нд	нд	нд	нд	нд
<code>intra_luma_mpm_idx [] []</code>	обхід	пропуск	пропуск	пропуск	нд	нд
<code>intra_luma_mpm_remainder [] []</code>	обхід	пропуск	пропуск	пропуск	пропуск	пропуск

10

У варіанті здійснення список MPM створюється згідно з таким процесом.

8.4.2 Процес виведення для режиму інтра-прогнозування яскравості

Вхідними даними в даному процесі є такі:

15

- місцеположення яскравості (`xCb`, `yCb`), що вказує верхній лівий дискретний відлік поточного блока кодування яскравості щодо верхнього лівого дискретного відліку яскравості поточного зображення,

- змінна `cbWidth`, що вказує ширину поточного блока кодування в дискретних відліках яскравості,

20

- змінна `cbHeight`, що вказує висоту поточного блока кодування в дискретних відліках яскравості.

У даному процесі виводиться режим `IntraPredModeY[xCb][yCb]` інтра-прогнозування яскравості.

Таблиця 13 визначає значення для режиму інтра-прогнозування `IntraPredModeY[xCb][yCb]` і відповідні імена.

25

Визначення режиму інтра-прогнозування і відповідні імена

Режим інтра-прогнозування	Відповідні імена
0	INTRA_PLANAR
1	INTRA_DC
2...66	INTRA_ANGULAR2...INTRA_ANGULAR66
81...83	INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM

ПРИМІТКА: Режими інтра-прогнозування INTRA\_LT\_CCLM, INTRA\_L\_CCLM і INTRA\_T\_CCLM можуть бути застосовні лише до компонентів кольоровості.

5 IntraPredModeY[xCb][yCb] виводиться у такий спосіб:

- Якщо intra\_luma\_not\_planar\_flag[xCb][yCb] дорівнює 1, застосовуються такі впорядковані етапи:

1. Сусідні місцеположення (xNbA, yNbA) і (xNbB, yNbB) встановлюються однаковими (xCb – 1, yCb+cbHeight–1) і (xCb+cbWidth–1, yCb–1), відповідно.

10 2. Якщо X замінюється на A або B, змінні candIntraPredModeX виводяться у такий спосіб:

- Ініціюється процес виведення доступності для блока: процес перевірки доступності сусідніх блоків ініціюється з місцеположенням (xCurr, yCurr), встановленим таким, що дорівнює (xCb, yCb), і сусіднім місцеположенням (xNbY, yNbY), встановленим таким, що дорівнює (xNbX, yNbX) як вхідні дані, а виходу присвоюється availableX.

15 - Потенційно придатний режим інтра-прогнозування candIntraPredModeX виводиться у такий спосіб:

- Якщо істинною є одна або більше з таких умов, candIntraPredModeX встановлюється таким, що дорівнює INTRA\_PLANAR.

- Значення змінної availableX є ХИБНИМ (FALSE).

20 - CuPredMode[xNbX][yNbX] не дорівнює MODE\_INTRA, а ciip\_flag[xNbX][yNbX] не дорівнює 1.

- pcm\_flag[xNbX][yNbX] дорівнює 1.

- X дорівнює B і yCb – 1 є меншим, ніж ((yCb>>CtbLog2SizeY)<<CtbLog2SizeY).

- В іншому випадку CandIntraPredModeX встановлюється таким, що дорівнює IntraPredModeY[xNbX][yNbX],

25 де X замінюється або сусіднім блоком A, або B, місцеположення яскравості (xCb, yCb) визначає верхній лівий дискретний відлік поточного блока кодування яскравості щодо верхнього лівого дискретного відліку яскравості поточного зображення, місцеположення (xNbA, yNbA) сусіда A встановлюється таким, що дорівнює (xCb–1, yCb+cbHeight–1), місцеположення (xNbB, yNbB) сусіда B встановлюється рівним (xCb+cbWidth–1, yCb–1), змінна cbWidth задає ширину поточного блока кодування в дискретних відліках яскравості, змінна cbHeight задає висоту поточного блока кодування в дискретних відліках яскравості, CuPredMode відображає поточний режим прогнозування, SizeY відображає розмір Y компонентів блока дерева кодування.

3. CandModeList [x] з x=0...4 виводиться у такий спосіб:

35 - Якщо CandIntraPredModeB дорівнює candIntraPredModeA, а candIntraPredModeA є більшим, ніж INTRA\_DC, candModeList[x] з x=0...4 виводиться у такий спосіб:

candModeList[0]=candIntraPredModeA (8-10)

candModeList[1]=2+((candIntraPredModeA+61) % 64) (8-12)

candModeList[2]=2+((candIntraPredModeA–1) % 64) (8-13)

candModeList[3]=INTRA\_DC (8-11)

40 candModeList[4]=2+((candIntraPredModeA+60) % 64) (8-14).

- В іншому випадку, якщо candIntraPredModeB не дорівнює candIntraPredModeA, а candIntraPredModeA або candIntraPredModeB є більшим, ніж INTRA\_DC, застосовується таке:

- Змінні minAB і maxAB виводяться у такий спосіб:

45 minAB=Min(candIntraPredModeA, candIntraPredModeB) (8-24)

maxAB=Max(candIntraPredModeA, candIntraPredModeB) (8-25)

- Якщо candIntraPredModeA і candIntraPredModeB обидва є більшими, ніж INTRA\_DC, candModeList[x] з x=0...4 виводиться у такий спосіб:

candModeList[0]=candIntraPredModeA (8-27)

candModeList[1]=candIntraPredModeB (8-29)

50 candModeList[2]=INTRA\_DC (8-29)

- Якщо  $\max_{AB} - \min_{AB}$  знаходиться в діапазоні від 2 до 62 включно, застосовується таке:  
 $\text{candModeList}[3] = 2 + ((\max_{AB} + 61) \% 64)$  (8-30)  
 $\text{candModeList}[4] = 2 + ((\max_{AB} - 1) \% 64)$  (8-31)  
- Інакше застосовується таке:  
5  $\text{candModeList}[3] = 2 + ((\max_{AB} + 60) \% 64)$  (8-32)  
 $\text{candModeList}[4] = 2 + ((\max_{AB}) \% 64)$  (8-33)  
- В іншому випадку ( $\text{candIntraPredModeA}$  або  $\text{candIntraPredModeB}$  є більшим, ніж  $\text{INTRA\_DC}$ ),  $\text{candModeList}[x]$  з  $x=0 \dots 4$  виводиться у такий спосіб:  
10  $\text{candModeList}[0] = \max_{AB}$  (8-65)  
 $\text{candModeList}[1] = \text{INTRA\_DC}$  (8-66)  
 $\text{candModeList}[2] = 2 + ((\max_{AB} + 61) \% 64)$  (8-66)  
 $\text{candModeList}[3] = 2 + ((\max_{AB} - 1) \% 64)$  (8-67)  
 $\text{candModeList}[4] = 2 + ((\max_{AB} + 60) \% 64)$  (8-68)  
- Інакше застосовується таке:  
15  $\text{candModeList}[0] = \text{INTRA\_DC}$  (8-71)  
 $\text{candModeList}[1] = \text{INTRA\_ANGULAR50}$  (8-72)  
 $\text{candModeList}[2] = \text{INTRA\_ANGULAR18}$  (8-73)  
 $\text{candModeList}[3] = \text{INTRA\_ANGULAR46}$  (8-74)  
 $\text{candModeList}[4] = \text{INTRA\_ANGULAR54}$  (8-75)  
20 4.  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  отримують шляхом застосування такої процедури:  
- Якщо  $\text{intra\_luma\_mpm\_flag}[x_{Cb}][y_{Cb}]$  дорівнює 1,  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  встановлюється таким, що дорівнює  $\text{candModeList}[\text{intra\_luma\_mpm\_idx}[x_{Cb}][y_{Cb}]]$ .  
- В іншому випадку  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  отримують шляхом виконання таких впорядкованих етапів:  
25 1. Коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0 \dots 3$  і, для кожного  $i, j=(i+1) \dots 4$ , обидва значення міняються місцями так:  
 $(\text{candModeList}[i], \text{candModeList}[j]) = \text{Swap}(\text{candModeList}[i], \text{candModeList}[j])$  (8-94)  
2.  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  отримують шляхом виконання таких впорядкованих етапів:  
30 i.  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  встановлюється таким, що дорівнює  $\text{intra\_luma\_mpm\_remainder}[x_{Cb}][y_{Cb}]$ .  
ii. Значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  збільшується на одиницю.  
iii. Для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  є більшим ніж або дорівнює  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  збільшується на одиницю.  
- В іншому випадку ( $\text{intra\_luma\_not\_planar\_flag}[x_{Cb}][y_{Cb}]$  дорівнює 0)  
35  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  встановлюється таким, що дорівнює  $\text{INTRA\_PLANAR}$ .  
Змінна  $\text{IntraPredModeY}[x][y]$  з  $x=x_{Cb} \dots x_{Cb} + \text{cbWidth} - 1$  і  $y=y_{Cb} \dots y_{Cb} + \text{cbHeight} - 1$  встановлюється такою, що дорівнює  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ .  
В одному прикладі режим інтра-прогнозування поточного блока (наприклад, ім'я змінної  $\text{luma\_intra\_mode}$ ) не є ані планарним режимом, ані режимом інтра-прогнозування в списку 5-MPM, його значення виводиться, і прогнозування цього режиму інтра-прогнозування виконується у такий спосіб:  
40 1. Виводиться значення  $\text{intra\_luma\_mpm\_remainder}$  з потоку бітів, і  $\text{luma\_intra\_mode}$  встановлюється таким, що дорівнює значенню  $\text{intra\_luma\_mpm\_remainder}$ .  
2. Значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  збільшується на одиницю;  
45 3. Для  $i$ , що дорівнює від 0 до 4 (необхідно порівняти з кожним записом в списку 5-MPM), коли  $\text{luma\_intra\_mode}$  є більшим ніж або дорівнює  $\text{MPM}[i]$  (у даному випадку  $\text{MPM}[0]$  відповідає першому запису в списку MPM,  $\text{MPM}[1]$  відповідає другому запису в списку MPM і т.д.), значення  $\text{luma\_intra\_mode}$  збільшується на одиницю.  
4. Використання виведеного режиму інтра-прогнозування ( $\text{luma\_intra\_mode}$ ) і відповідних опорних дискретних відліків для виконання інтер-прогнозування.  
50 (З боку кодера)  
В одному прикладі режим інтра-прогнозування поточного блока (наприклад, ім'я змінної  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$ ) не є ані планарним режимом, ані режимом інтра-прогнозування в списку 5-MPM, і його значення кодується у такий спосіб:  
55 1. Коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0 \dots 3$  і, для кожного  $i, j=(i+1) \dots 4$ , обидва значення міняються місцями так:  
 $(\text{candModeList}[i], \text{candModeList}[j]) = \text{Swap}(\text{candModeList}[i], \text{candModeList}[j])$  (8-94)  
2. Для  $i$  від 4 до 0 включно, коли  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  є більшим, ніж  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  зменшується на одиницю.  
60 3. Значення  $\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$  зменшується на одиницю;

4. Значення  $\text{IntraPredModeY}[x\text{Cb}][y\text{Cb}]$  записується в потік бітів з кодовим словом з використанням процесу усіченої двійкової бінарizaції, і всі біни в кодовому слові кодуються з пропуском.

5 У кодері поточний режим вже відомий, виводиться кодове слово (тобто  $\text{intra\_luma\_mrm\_remainder}$ ).

Для специфікації процесу усіченої двійкової бінарizaції він визначається так (9.3.3.4 в JVET-Q2001-VE.docx): Процес бінарizaції усіченого двійкового коду (TB)

10 Входом в цей процес є запит на бінарizaцію TB для елемента синтаксису зі значенням  $\text{synVal}$  і  $\text{cMax}$ . Виходом з цього процесу є бінарizaція TB елемента синтаксису. Рядок бінів процесу бінарizaції TB елемента синтаксису  $\text{synVal}$  задається у такий спосіб:

$n = \text{cMax} + 1$   
 $k = \text{Floor}(\text{Log}_2(n))$  (1550)  
 $u = (1 \ll (k+1)) - n$

15 - Якщо  $\text{synVal}$  є меншим ніж  $u$ , рядок бінів TB отримують шляхом виклику процесу бінарizaції FL, вказаного в пункті 9.3.3.7 для  $\text{synVal}$ , зі значенням  $\text{cMax}$ , яке дорівнює  $(1 \ll k) - 1$ .

- В іншому випадку ( $\text{synVal}$  є більшим ніж або дорівнює  $u$ ) рядок бінів TB отримують шляхом виклику процесу бінарizaції FL, вказаного в розділі 9.3.3.7 для  $(\text{synVal} + u)$ , зі значенням  $\text{cMax}$ , яке дорівнює  $(1 \ll (k+1)) - 1$ .

20 В одній реалізації зазначених вище варіантів здійснення планарний режим може неявно розглядатися завжди як перший запис в неявному списку 6-MPM. Цей неявний список 6-MPM складається з планарного (завжди перший запис) і 5 інших режимів (які можуть бути явно описані як список 5-MPM).

25 Це пов'язано з тим, що якщо планарний режим завжди знаходиться в першій позиції неявного списку 6-MPM, він може бути відділений від неявного списку MPM і йому присвоюється спеціальна мітка (наприклад, мітка  $\text{intra\_luma\_planar\_flag}$  або мітка  $\text{intra\_luma\_not\_planar\_flag}$ ), щоб вказати, чи є режим інтра-прогнозування поточного блока планарним, чи ні.

30 В одному прикладі розглядається введення  $\text{intra\_luma\_not\_planar\_flag}[x0][y0]$  як спеціальної мітки, щоб вказати, чи є планарний режим режимом інтра-прогнозування поточного блока. Коли це планарний режим,  $\text{intra\_luma\_not\_planar\_flag}$  встановлюється таким, що дорівнює 0, в іншому випадку встановлюється таким, що дорівнює 1. В даному випадку перший бін  $\text{mrm\_idx}$  еквівалентний спеціальній мітці  $\text{intra\_luma\_not\_planar\_flag}$ .

Коли режим інтра-прогнозування поточного блока є планарним, індекс MPM дорівнює 0, а кодований бін дорівнює 0. Коли режим інтра-прогнозування поточного блока не є планарним, індекс MPM не дорівнює 0, а перший бін індексу MPM завжди дорівнює 1.

35 У прикладі  $\text{intra\_luma\_mrm\_idx}$  кодується з використанням усіченого унарного (TR) коду,

Значення TR  
 $\text{Mrm\_idx}$  0 0  
 $\text{Mrm\_idx}$  1 1 0  
 $\text{Mrm\_idx}$  2 1 1 0  
40  $\text{Mrm\_idx}$  3 1 1 1 0  
 $\text{Mrm\_idx}$  4 1 1 1 1.

Згідно з одним варіантом здійснення винаходу (некутова гілка):

45 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок кодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як  $\text{left\_mode}$ , а режим інтра-прогнозування верхнього сусіднього блока позначений як  $\text{above\_mode}$ .

50 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

-  $\text{Left\_mode}$  є не кутовим режимом і  $\text{above\_mode}$  є не кутовим режимом,

55 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає  $\text{MPM}[1]$ , 2-ий запис відповідає  $\text{MPM}[2]$  тощо):

$\text{MPM}[1]$ : DC  
 $\text{MPM}[2]$ : вертикальний режим (тобто 50)  
 $\text{MPM}[3]$ : горизонтальний режим (тобто 18)  
 $\text{MPM}[4]$ : вертикальний -4 (тобто 46)  
60  $\text{MPM}[5]$ : по вертикалі +4 (тобто 54)

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (некутова гілка):

5 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

10 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Left\_mode є не кутовим режимом і above\_mode є не кутовим режимом,

15 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

MPM[1]: DC

MPM[2]: вертикальний режим (тобто 50)

MPM[3]: горизонтальний режим (тобто 18)

20 MPM[4]: 66

MPM[5]: 2

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (некутова гілка):

25 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

30 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Left\_mode є не кутовим режимом і above\_mode є не кутовим режимом,

35 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

MPM[1]: DC

MPM[2]: вертикальний режим (тобто 50)

40 MPM[3]: горизонтальний режим (тобто 18)

MPM[4]: 2

MPM[5]: 34

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (некутова гілка):

45 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

50 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Left\_mode є некутовим режимом і above\_mode є некутовим режимом,

55 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

MPM[1]: DC

60 MPM[2]: вертикальний режим (тобто 50)

MPM[3]: горизонтальний режим (тобто 18)

MPM[4]: 66

MPM[5]: 34

5 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, один – кутовий, інший – некутовий):

10 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

15 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наведені нижче умови (марковані пункти) оцінені як істинні,

- Один з left\_mode і above\_mode є кутовим режимом, а інший – некутовим режимом,

20 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

задають кутовий режим як ang і отримують наступне:

якщо left\_mode є кутовим режимом, то  $ang = \text{left\_mode}$ ,

в іншому випадку (above\_mode є кутовим режимом)  $ang = \text{above\_mode}$

25 MPM[1]: ang

MPM[2]: DC

MPM[3]:  $2 + ((ang + 61) \% 64)$

MPM[4]:  $2 + ((ang - 1) \% 64)$

MPM[5]:  $2 + ((ang + 60) \% 64)$

30 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, один – кутовий, інший – некутовий):

35 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

40 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Один з left\_mode і above\_mode є кутовим режимом, а інший – некутовим режимом,

45 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

задають кутовий режим як ang і отримують наступне:

якщо left\_mode є кутовим режимом,  $ang = \text{left\_mode}$

в іншому випадку (above\_mode є кутовим режимом)  $ang = \text{above\_mode}$

50 MPM[1]: DC

MPM[2]: ang

MPM[3]:  $2 + ((ang + 61) \% 64)$

MPM[4]:  $2 + ((ang - 1) \% 64)$

MPM[5]:  $2 + ((ang + 60) \% 64)$

55 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, один – кутовий, інший – некутовий):

60 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення

або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

5 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Один з left\_mode і above\_mode є кутовим режимом, інший – некутовим режимом,

тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

задають кутовий режим як ang і отримують наступне:

якщо left\_mode є кутовим режимом,  $ang = \text{left\_mode}$ ,

в іншому випадку (above\_mode є кутовим режимом)  $ang = \text{above\_mode}$

MPM[1]: ang

15 MPM[2]:  $2 + ((ang + 61) \% 64)$

MPM[3]: DC

MPM[4]:  $2 + ((ang - 1) \% 64)$

MPM 5]:  $2 + ((ang + 60) \% 64)$

20 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, один – кутовий, інший – некутовий):

25 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

30 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Один з left\_mode і above\_mode є кутовим режимом, інший – некутовим режимом,

тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

задають кутовий режим як ang і отримують наступне:

якщо left\_mode є кутовим режимом,  $ang = \text{left\_mode}$ ,

в іншому випадку (above\_mode є кутовим режимом)  $ang = \text{above\_mode}$

MPM[0]: Планарний

40 MPM[1]: ang

MPM[2]:  $2 + ((ang + 61) \% 64)$

MPM[3]:  $2 + ((ang - 1) \% 64)$

MPM[4]: DC

MPM[5]:  $2 + ((ang + 60) \% 64)$

45 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, один – кутовий, інший – некутовий):

50 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.

55 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- Один з left\_mode і above\_mode є кутовим режимом, інший – некутовим режимом,

тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

задають кутовий режим як `ang` і отримують наступне:

5 якщо `left_mode` є кутовим режимом, `ang=left_mode`  
в іншому випадку (`above_mode` є кутовим режимом) `ang=above_mode`

MPM[1]: `ang`

MPM[2]:  $2+((ang+61) \% 64)$

MPM[3]:  $2+((ang-1) \% 64)$

10 MPM[4]:  $2+((ang+60) \% 64)$

MPM[5]: DC

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, обидва однаково кутові):

15 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

20 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- `Left_mode` і `above_mode` обидва є кутові режими і вони є однаковими,

25 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

MPM[1]: `left_mode`

MPM[2]:  $2+((left\_mode+61) \% 64)$

30 MPM[3]:  $2+((left\_mode-1) \% 64)$

MPM[4]: DC

MPM[5]:  $2+((left\_mode+60) \% 64)$

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

35 Згідно з одним варіантом здійснення винаходу (однокутова гілка, обидва однаково кутові):

- Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

40 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- `Left_mode` і `above_mode` обидва є кутовими режимами і вони є однаковими,

45 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий запис відповідає MPM[2] тощо):

MPM[1]: DC

50 MPM[2]: `left_mode`

MPM[3]:  $2+((left\_mode+61) \% 64)$

MPM[4]:  $2+((left\_mode-1) \% 64)$

MPM[5]:  $2+((left\_mode+60) \% 64)$

55 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, обидва однаково кутові):

60 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока

встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

- Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

5 - Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,  
 - `Left_mode` і `above_mode` обидва є кутовими режимами і вони є однаковими,  
 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає `MPM[1]`, 2-ий запис відповідає `MPM[2]` тощо):

10 `MPM[1]: left_mode`  
`MPM[2]: DC`  
`MPM[3]: 2+((left_mode+61) % 64)`  
`MPM[4]: 2+((left_mode-1) % 64)`  
`MPM[5]: 2+((left_mode+60) % 64)`

15 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс `MPM`) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (однокутова гілка, обидва однаково кутові):

20 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

25 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- `Left_mode` і `above_mode` обидва є кутовими режимами і вони є однаковими,

тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає `MPM[1]`, 2-ий запис відповідає `MPM[2]` тощо):

30 `MPM[1]: left_mode`  
`MPM[2]: 2+((left_mode+61) % 64)`  
`MPM[3]: DC`  
`MPM[4]: 2+((left_mode-1) % 64)`  
`MPM[5]: 2+((left_mode+60) % 64)`

35 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

40 - Етап 2: визначення того, чи є режими інтра-прогнозування сусідніх блоків кутовими режимами, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- `Left_mode` і `above_mode` обидва є кутовими режимами і вони є однаковими,

45 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає `MPM[1]`, 2-ий запис відповідає `MPM[2]` тощо):

50 `MPM[1]: left_mode`  
`MPM[2]: 2+((left_mode+61) % 64)`  
`MPM[3]: 2+((left_mode-1) % 64)`  
`MPM[4]: 2+((left_mode+60) % 64)`  
`MPM[5]: DC`

55 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс `MPM`) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (двокутова гілка):

60 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока

встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

- Етап 2: визначення того, чи є режим інтра-прогнозування сусідніх блоків кутовим режимом, чи ні.

5 - Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,  
 - `Left_mode` і `above_mode` є двома різними кутовими режимами,  
 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає `MPM[1]`, 2-ий запис відповідає `MPM[2]` тощо):

10 `MPM[1]: left_mode`  
`MPM[2]: above_mode`  
`MPM[3]: DC`

Змінні `minAB` і `maxAB` виводяться у такий спосіб:

15 `minAB=Min(above_mode, left_mode)`  
`maxAB=Max(above_mode, left_mode)`

Якщо `maxAB-minAB` знаходиться в діапазоні від 2 до 62 включно, застосовується таке:

`MPM[4]=2+((maxAB+61) % 64) (8-30)`

`MPM[5]=2+((maxAB-1) % 64) (8-31)`

Інакше застосовується таке:

20 `MPM[4]=2+((maxAB+60) % 64) (8-32)`

`MPM[5]=2+((maxAB) % 64) (8-33)`

- Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс `MPM`) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (двокутова гілка):

25 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

30 - Етап 2: визначення того, чи є режим інтра-прогнозування сусідніх блоків кутовим режимом, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,

- `Left_mode` і `above_mode` є двома різними кутовими режимами,

35 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає `MPM[1]`, 2-ий запис відповідає `MPM[2]` тощо):

`MPM[1]: above_mode`

`MPM[2]: left_mode`

40 `MPM[3]: DC`

Змінні `minAB` і `maxAB` виводяться у такий спосіб:

`minAB=Min(above_mode, left_mode)`

`maxAB=Max(above_mode, left_mode)`

Якщо `maxAB-minAB` знаходиться в діапазоні від 2 до 62 включно, застосовується таке:

45 `MPM[4]=2+((maxAB+61) % 64) (8-30)`

`MPM[5]=2+((maxAB-1) % 64) (8-31)`

Інакше застосовується таке:

`MPM[4]=2+((maxAB+60) % 64) (8-32)`

`MPM[5]=2+((maxAB) % 64) (8-33)`

50 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс `MPM`) і записів побудованого списку.

Згідно з одним варіантом здійснення винаходу (двокутова гілка):

55 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як `left_mode`, а режим інтра-прогнозування верхнього сусіднього блока позначений як `above_mode`.

60 - Етап 2: визначення того, чи є режим інтра-прогнозування сусідніх блоків кутовим режимом, чи ні.

- Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,  
 - Left\_mode і above\_mode є двома різними кутовими режимами,  
 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-  
 прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий  
 5 запис відповідає MPM[2] тощо):  
 MPM[1]: left\_mode  
 MPM[2]: DC  
 MPM[3]: above\_mode  
 Змінні minAB і maxAB виводяться у такий спосіб:  
 10 minAB=Min(above\_mode, left\_mode)  
 maxAB=Max(above\_mode, left\_mode)  
 Якщо maxAB-minAB знаходиться в діапазоні від 2 до 62 включно, застосовується таке:  
 MPM[4]=2+((maxAB+61) % 64) (8-30)  
 MPM[5]=2+((maxAB-1) % 64) (8-31)  
 15 Інакше застосовується таке:  
 MPM[4]=2+((maxAB+60) % 64) (8-32)  
 MPM[5]=2+((maxAB) % 64) (8-33)  
 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і  
 записів побудованого списку.  
 20 Згідно з одним варіантом здійснення винаходу (двокутова гілка):  
 - Етап 1: отримання режиму інтра-прогнозування лівого сусіднього блока і режиму інтра-  
 прогнозування верхнього сусіднього блока згідно з фіг. 10. Якщо режим інтра-прогнозування  
 сусіднього блока недоступний (наприклад, сусідній блок знаходиться поза межами зображення  
 або сусідній блок закодований інтер-кодуванням), режим інтра-прогнозування сусіднього блока  
 25 встановлюється як планарний. Режим інтра-прогнозування лівого сусіда позначений як  
 left\_mode, а режим інтра-прогнозування верхнього сусіднього блока позначений як above\_mode.  
 - Етап 2: визначення того, чи є режим інтра-прогнозування сусідніх блоків кутовим режимом,  
 чи ні.  
 - Етап 3: якщо всі наступні умови (марковані пункти) оцінені як істинні,  
 30 - Left\_mode і above\_mode є двома різними кутовими режимами,  
 тоді створюється список режимів інтра-прогнозування, який містить N записів режимів інтра-  
 прогнозування, у такий спосіб (наприклад, N дорівнює 5, де 1-ий запис відповідає MPM[1], 2-ий  
 запис відповідає MPM[2] тощо):  
 35 MPM[1]: above\_mode  
 MPM[2]: DC  
 MPM[3]: left\_mode  
 Змінні minAB і maxAB виводяться у такий спосіб:  
 minAB=Min(above\_mode, left\_mode)  
 maxAB=Max(above\_mode, left\_mode)  
 40 Якщо maxAB-minAB знаходиться в діапазоні від 2 до 62 включно, застосовується таке:  
 MPM[4]=2+((maxAB+61) % 64) (8-30)  
 MPM[5]=2+((maxAB-1) % 64) (8-31)  
 Інакше застосовується таке:  
 45 MPM[4]=2+((maxAB+60) % 64) (8-32)  
 MPM[5]=2+((maxAB) % 64) (8-33)  
 - Етап 4: прогнозування поточного блока відповідно до індексу (позначеного як індекс MPM) і  
 записів побудованого списку.  
 В одній реалізації варіанта здійснення N встановлене таким, що дорівнює 6.  
 В одній реалізації варіанта здійснення N встановлене таким, що дорівнює 6. В даному  
 50 випадку згаданий список режимів інтра-прогнозування містить перші 5 записів, як вказано вище,  
 а не шостий запис.  
 В одній реалізації варіанта здійснення N встановлене таким, що дорівнює 4. В даному  
 випадку згаданий список режимів інтра-прогнозування включає перші 4 записи, як вказано  
 вище, а не п'ятий запис і шостий запис.  
 55 В одній реалізації варіанта здійснення згаданий список режимів інтра-прогнозування  
 називається списком MPM.  
 В одній реалізації згаданий індекс списку (індекс MPM) сигналізується в потоці бітів як  
 індикатор. Переважно індекс MPM можна кодувати за допомогою контекстно-адаптивного  
 ентропійного кодера (CABAC). Індекс MPM може бути закодований згідно з різною кількістю  
 60 імовірнісних моделей (іншими словами, контекстом) за допомогою CABAC.

Так, наприклад, перший бін кодового слова для `trm_index` кодується контекстним кодуванням CABAC.

В одному прикладі його контекст визначається на основі того, чи застосовується поточний інтра-блок з множинними опорними рядками, ISP або звичайними інтра-блоками. Загалом створюється три контексти для контексту CABAC першого біна кодового слова для `trm_index`.

Якщо список MPM містить як планарний режим, так і режим DC, то список MPM, з виключенням планарного режиму і режиму DC зі створеного списку MPM, використовується інструментом кодування з множинними опорними рядками. У даному випадку список 4-MPM використовується при кодуванні в режимі інтра-прогнозування з множинними опорними рядками. В одному прикладі список MPM виглядає так: {Planar (значення 0), DC (значення 1), VER (значення 50), HOR (значення 18), VER-4 (значення 46), VER+4 (значення 54)} після побудови списку MPM, тоді список 4-MPM з {VER, HOR, VER-4, VER+4} використовується при кодуванні в режимі інтра-прогнозування, коли задіяний множинний опорний рядок.

Якщо список MPM містить планарний режим, але без режиму DC, то список MPM, з виключенням планарного режиму зі створеного списку MPM, використовується інструментом кодування з множинними опорними рядками. У даному випадку список 5-MPM використовується при кодуванні в режимі інтра-прогнозування з множинними опорними рядками. В одному прикладі список MPM виглядає так: {Planar, VER, HOR, 43, 60, 3} після побудови списку MPM, тоді список 5-MPM з {VER, HOR, 43, 60, 3} використовується при кодуванні в режимі інтра-прогнозування, коли задіяний множинний опорний рядок.

В іншому прикладі може бути необхідна модифікація синтаксису для списку MPM без режиму DC. Якщо список MPM не містить DC, вірогідно, режим DC буде дуже часто використаний у гілці без MPM (тобто мітка `trm` є хибною). У даному випадку вводиться новий синтаксис, `DC_mode`. `DC_mode`, який дорівнює 1, вказує, що режимом інтра-прогнозування поточного блока є режим DC. `DC_mode`, який дорівнює 0, вказує, що режим інтра-прогнозування поточного блока не є режимом DC.

А саме: старий синтаксис без MPM змінений з  
Якщо (`trm_flag`)

...  
інакше  
`intra_mode` (з використанням ТВ)  
на  
Якщо (`trm_flag`)

...  
інакше  
`DC_mode`  
Якщо (`DC_mode==0`)  
`intra_mode` (з використанням ТВ)

Синтаксис `DC_mode` може бути закодований за допомогою контекстно-адаптивного ентропійного кодера (Context Adaptive Entropy Coder, CABAC). `DC_mode` може бути закодований згідно з різною кількістю імовірнісних моделей (іншими словами, контекстом) за допомогою CABAC. З введенням `DC_Mode`, максимальне значення для усіченого двійкового об'єкта становить 60, тобто 67 (кількість всіх режимів)–6 (MPM)–1 (DC). У той час як без введення `DC_mode` максимальне значення для усіченого двійкового об'єкта становить 61, тобто 67 (кількість усіх режимів)–6 (MPM).

Зокрема, наведені нижче способи кодування з прогнозуванням поточного блока реалізуються пристроєм декодування. Пристрій декодування може бути відеодекодером 30, представленим на фіг. 1А, або декодером 30, представленим на фіг. 3.

Згідно з варіантом 1200 здійснення (див. фіг. 12) спосіб інтра-прогнозування поточного блока, який реалізується пристроєм декодування, включає такі етапи. На етапі 1201 пристрій отримує значення інформації вказівки поточного блока. Інформація вказівки позначається міткою, наприклад, `intra_luma_trm_flag`.

На етапі 1202 пристрій визначає, чи вказує значення інформації вказівки, що режим інтра-прогнозування поточного блока входить в набір найбільш імовірних режимів. Як описано вище, коли значення `intra_luma_trm_flag` дорівнює 1, режим інтра-прогнозування поточного блока міститься в наборі найбільш імовірних режимів. Коли значення `intra_luma_trm_flag` дорівнює 0, режим інтра-прогнозування поточного блока не міститься в наборі найбільш імовірних режимів.

На етапі 1203 пристрій виводить режим інтра-прогнозування `IntraPredModeY[xCb][yCb]` поточного блока за допомогою таких впорядкованих етапів, коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш

імовірних режимів:

i.  $IntraPredModeY[xCb][yCb]$  встановлюється таким, що дорівнює  $intra\_luma\_mpm\_remainder[xCb][yCb]$ .

ii. Значення  $IntraPredModeY[xCb][yCb]$  збільшується на одиницю.

5 iii. Для  $i$ , що дорівнює від 0 до 4 включно, коли  $IntraPredModeY[xCb][yCb]$  є більшим ніж або дорівнює  $sandModeList[i]$ , значення  $IntraPredModeY[xCb][yCb]$  збільшується на одиницю.

Потенційно придатний режим інтра-прогнозування  $sandModeList[i]$  в наборі найбільш імовірних режимів отримують у такий спосіб:

коли  $sandModeList[i]$  є більшим, ніж  $sandModeList[j]$  для  $i=0...3$  і, для кожного  $i$ ,  $j=(i+1)...4$ , обидва значення міняються місцями так:

$(sandModeList[i], sandModeList[j])=Swap(sandModeList[i], sandModeList[j])$ .

На етапі 1204 пристрій використовує виведений режим інтра-прогнозування ( $luma\_intra\_mode$ ) і відповідні опорні дискретні відліки для виконання інтер-прогнозування.

15 Детальна інформація про цей варіант 1200 здійснення представлена у наведених вище варіантах здійснення.

Згідно з варіантом здійснення 1300 (див. фіг. 13) спосіб інтра-прогнозування поточного блока, який реалізовується пристроєм кодування, включає такі етапи. Пристрій кодування може бути відеокодером 20, представленим на фіг. 1A, або кодером 20, представленим на фіг. 2.

20 На етапі 1301 пристрій визначає, чи входить режим інтра-прогнозування поточного блока в набір найбільш імовірних режимів. На етапі 1302 пристрій виводить змінну, використовуючи режим інтра-прогнозування  $IntraPredModeY[xCb][yCb]$  поточного блока, за допомогою таких впорядкованих етапів, коли режим інтра-прогнозування  $IntraPredModeY[xCb][yCb]$  поточного блока не міститься в наборі найбільш імовірних режимів:

25 i. Для  $i$  від 0 до 4 включно, коли  $IntraPredModeY[xCb][yCb]$  є більшим, ніж  $sandModeList[i]$ , значення  $IntraPredModeY[xCb][yCb]$  зменшується на одиницю;

ii. Значення  $IntraPredModeY[xCb][yCb]$  зменшується на одиницю.

Потенційно придатний режим інтра-прогнозування  $sandModeList[i]$  в наборі найбільш імовірних режимів отримують у такий спосіб:

30 коли  $sandModeList[i]$  є більшим, ніж  $sandModeList[j]$  для  $i=0...3$  і, для кожного  $i$ ,  $j=(i+1)...4$ , обидва значення міняються місцями так:

$(sandModeList[i], sandModeList[j])=Swap(sandModeList[i], sandModeList[j])$ .

На етапі 1303 пристрій кодує потік бітів, причому двійковий потік містить інформацію згаданої змінної, наприклад,  $intra\_luma\_mpm\_remainder[xCb][yCb]$ , і надсилає закодований потік бітів на пристрій декодування.

35 Детальна інформація про даний варіант 1300 здійснення представлена у наведених вище варіантах здійснення.

Фіг. 14 ілюструє варіанти здійснення пристрою 1400 декодування. Пристрій 1400 декодування може бути відеодекодером 30, представленим на фіг. 1A, або декодером 30, представленим на фіг. 3. Пристрій 1400 декодування може бути використаний для реалізації варіанта 1200 здійснення і інших варіантів здійснення, описаних вище.

40 Пристрій містить блок (1401) отримання і блок (1402) виведення. Блок (1401) отримання виконаний з можливістю отримання значення інформації вказівки поточного блока, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш імовірних режимів;

45 Блок (1402) виведення виконаний з можливістю виведення режиму інтра-прогнозування  $IntraPredModeY[xCb][yCb]$  поточного блока за допомогою таких впорядкованих етапів, коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш імовірних режимів:

50 i.  $IntraPredModeY[xCb][yCb]$  встановлюється таким, що дорівнює  $intra\_luma\_mpm\_remainder[xCb][yCb]$ ,

ii. Значення  $IntraPredModeY[xCb][yCb]$  збільшується на одиницю;

де  $intra\_luma\_mpm\_remainder[xCb][yCb]$  відображає режим інтра-прогнозування залишку, місцеположення яскравості ( $xCb$ ,  $yCb$ ) визначає верхній лівий дискретний відлік поточного блока щодо верхнього лівого дискретного відліку яскравості поточного зображення.

55 Пристрій 1400 додатково містить блок 1403 прогнозування (не показаний на фіг. 14). Блок 1403 прогнозування виконаний з можливістю використання виведеного режиму інтра-прогнозування ( $luma\_intra\_mode$ ) і відповідних опорних дискретних відліків для виконання інтер-прогнозування.

60 Фіг. 15 ілюструє варіанти здійснення пристрою 1500 кодування для інтер-прогнозування. Пристрій 1500 кодування може бути відеокодером 20, представленим на фіг. 1A, або кодером

20, представленим на фіг. 2. Пристрій 1500 кодування може бути використаний для реалізації варіанта 1300 здійснення і інших варіантів здійснення, описаних вище.

Пристрій 1500 містить блок 1501 виведення і блок 1502 передачі. Блок (1501) виведення виконаний з можливістю виведення змінною з використанням режиму інтра-прогнозування IntraPredModeY[xCb][yCb] поточного блока за допомогою таких впорядкованих етапів, коли режим інтра-прогнозування IntraPredModeY[xCb][yCb] поточного блока не міститься в наборі найбільш імовірних режимів:

i. Для  $i$ , що дорівнює від 0 до 4 включно, коли IntraPredModeY[xCb][yCb] є більшим, ніж candModeList[i], значення IntraPredModeY[xCb][yCb] зменшується на одиницю, де candModeList[i] є потенційно придатним режимом інтра-прогнозування в наборі найбільш імовірних режимів;

ii. значення IntraPredModeY[xCb][yCb] зменшується на одиницю;

при цьому місцеположення яскравості (xCb, yCb) визначає лівий верхній дискретний відлік поточного блока щодо верхнього лівого дискретного відліку яскравості поточного зображення.

Блок (1502) передачі виконаний з можливістю відправки потоку бітів в декодер, причому потік бітів містить інформацію згаданої змінної.

Варіанти здійснення даного винаходу спрощують процес виведення режиму без MPM, коли планарний режим завжди розглядається в списку MPM (або неявному, або явному) і вказується як мітка (intra\_luma\_not\_planar\_flag). Оскільки планарний відповідає найменшому значенню режимів інтра-прогнозування, процес виведення для режиму без MPM може бути спрощений. Для етапу порівняння і прирощування порівняння з планарним режимом може бути збережене, оскільки планарний завжди присутній в списку MPM, отже, початкове значення режиму інтра-прогнозування залишку може бути збільшене на одиницю безпосередньо, що відповідає етапу ii, описаному вище.

Нижче наведене пояснення застосувань способу кодування, а також способу декодування, як показано у наведених вище варіантах здійснення, і системи, яка їх використовує.

На фіг. 16 представлена блок-схема, яка демонструє систему 3100 доставки контенту для реалізації послуги розповсюдження контенту. Ця система 3100 доставки контенту включає пристрій 3102 захоплення, термінальний пристрій 3106 і необов'язково включає дисплей 3126. Пристрій 3102 захоплення обмінюється даними з термінальним пристроєм 3106 по лінії 3104 зв'язку. Лінія зв'язку може містити канал 13 зв'язку, описаний вище. Лінія 3104 зв'язку включає, але не обмежується ними, WIFI, Ethernet, кабель, бездротову мережу (3G/4G/5G), USB або будь-яку їхню комбінацію тощо.

Пристрій 3102 захоплення генерує дані і може кодувати дані способом кодування, як показано в наведених вище варіантах здійснення. Альтернативно пристрій 3102 захоплення може розподіляти дані на сервер потокової передачі (не показаний на кресленнях), а сервер кодує дані і передає кодовані дані на термінальний пристрій 3106. Пристрій 3102 захоплення включає, але не обмежується ними, камеру, смартфон або планшет, комп'ютер або портативний комп'ютер, систему відеоконференцзв'язку, КПК, пристрій, встановлений у транспортному засобі, або будь-яку їхню комбінацію тощо. Так, наприклад, пристрій 3102 захоплення може містити пристрій-джерело 12, як описано вище. Коли дані містять відео, відеокодер 20, включений в пристрій 3102 захоплення, може фактично виконувати обробку кодування відео. Коли дані містять звук (тобто голос), аудіокодер, включений в пристрій 3102 захоплення, може фактично виконувати обробку кодування звуку. Для деяких практичних сценаріїв пристрій 3102 захоплення розподіляє кодовані відео- і аудіодані шляхом їхнього мультиплексування разом. Для інших практичних сценаріїв, наприклад, у системі відеоконференцзв'язку, кодовані аудіодані і кодовані відеодані не мультиплекуються. Пристрій 3102 захоплення розподіляє окремо кодовані аудіодані і кодовані відеодані на термінальний пристрій 3106.

У системі 3100 доставки контенту термінальний пристрій 310 приймає і відтворює кодовані дані. Термінальний пристрій 3106 може бути пристроєм з можливістю прийому і відновлення даних, таким як смартфон або планшет 3108, комп'ютер або ноутбук 3110, мережевий відеореєстратор (NVR)/цифровий відеореєстратор (DVR) 3112, телевізор 3114, ресивер (STB) 3116 цифрового телебачення, система 3118 відеоконференцзв'язку, система 3120 відеоспостереження, кишеньковий персональний комп'ютер КПК (PDA) 3122, пристрій 3124, встановлений у транспортному засобі, або будь-яка їхня комбінація тощо, здатні декодувати зазначені вище кодовані дані. Так, наприклад, термінальний пристрій 3106 може містити пристрій-одержувач 14, як описано вище. Коли кодовані дані містить відео, відеодекодеру 30, включеному в термінальний пристрій, надається пріоритет для виконання декодування відео.

Коли кодовані дані містять звук, аудіодекодер, включений в термінальний пристрій, має пріоритет для виконання обробки декодування звуку.

У випадку термінального пристрою зі своїм дисплеєм, наприклад смартфона або планшета 3108, комп'ютера або ноутбука 3110, мережевого відеореєстратора (NVR)/цифрового відеореєстратора (DVR) 3112, телевізора 3114, кишенькового персонального комп'ютера КПК (PDA) 3122 або пристрою 3124, встановленого в транспортному засобі, термінальний пристрій може передавати декодовані дані на свій дисплей. У випадку термінального пристрою, не обладнаного дисплеєм, наприклад, ресивера 3116 цифрового телебачення (STB), системи 3118 відеоконференцз'язку або системи 3120 відеоспостереження, він контактує із зовнішнім дисплеєм 3126 для прийому і відображення декодованих даних.

Коли кожний пристрій у цій системі виконує кодування або декодування, може бути використаний пристрій кодування зображення або пристрій декодування зображення, як показано у наведених вище варіантах здійснення.

На фіг. 17 представлена схема, яка демонструє структуру прикладу термінального пристрою 3106. Після того, як термінальний пристрій 3106 приймає потік від пристрою 3102 захоплення, модуль 3202 обробки протоколу аналізує протокол передачі потоку. Протокол включає, але не обмежується ними, протокол потокової передачі в реальному часі (RTSP), протокол передачі гіпертексту (HTTP), протокол потокової передачі в реальному часі HTTP (HLS), MPEG-DASH, транспортний протокол в реальному часі (RTP), протокол обміну повідомленнями в реальному часі (RTMP) або будь-яку їхню комбінацію тощо.

Після того, як модуль 3202 обробки протоколу обробить потік, генерується файл потоку. Файл виводиться в модуль 3204 демультимплексування. Модуль 3204 демультимплексування може розділяти мультимплексовані дані на кодовані аудіодані і кодовані відеодані. Як описано вище, для деяких практичних сценаріїв, наприклад, у системі відеоконференцз'язку, кодовані аудіодані і кодовані відеодані не мультимплексуються. У цій ситуації кодовані дані передаються на відеодекодер 3206 і аудіодекодер 3208 без використання модуля 3204 демультимплексування.

За допомогою обробки демультимплексування генерується елементарний потік (ES) відео, ES аудіо і необов'язково субтитри. Відеодекодер 3206, який містить відеодекодер 30, як пояснено у наведених вище варіантах здійснення, декодує відео ES способом декодування, як показано у наведених вище варіантах здійснення, для генерування відеокадру і передає ці дані в модуль 3212 синхронізації. Аудіодекодер 3208 декодує аудіо ES для генерування аудіокадру і передає ці дані в модуль 3212 синхронізації. Альтернативно відеокадр може зберігатися в буфері (не показаний на фіг. 17) перед його подачею в модуль 3212 синхронізації. Аналогічно аудіокадр може зберігатися в буфері (не показаний на фіг. 17) перед його подачею в модуль 3212 синхронізації.

Модуль 3212 синхронізації синхронізує відеокадр і аудіокадр і передає відео/аудіо на відео/аудіо дисплей 3214. Так, наприклад, модуль 3212 синхронізації синхронізує представлення відео і аудіо інформації. Інформація може кодуватися в синтаксисі з використанням міток часу, що стосуються представлення кодованих аудіо- і візуальних даних, і міток часу, що стосується доставки самого потоку даних.

Якщо субтитри включені в потік, декодер 3210 субтитрів декодує субтитри і синхронізує їх з відеокадром і звуковим кадром і передає відео/аудіо/субтитри на дисплей 3216 відео/аудіо/субтитрів.

Даний винахід не обмежується описаною вище системою, і або пристрій кодування зображення, або пристрій декодування зображення у наведених вище варіантах здійснення може бути включений в іншу систему, наприклад, автомобільну систему.

#### Математичні оператори

Математичні оператори, використовувані в даній заявці, аналогічні тим, які використовуються в мові програмування C. Однак, результати операцій цілочислового ділення і арифметичного зсуву визначаються більш точно, крім того, визначаються додаткові операції, такі як піднесення до степені і ділення з дійсним знаком. Правила нумерації і підрахунку зазвичай починаються з 0, наприклад, "перший" еквівалентний 0-му, "другий" еквівалентний 1-му тощо.

55 Арифметичні оператори

Наведені нижче арифметичні оператори визначені так:	
+	Додавання
-	Віднімання (як оператор з двома аргументами) або заперечення (як унарний префіксний оператор)
*	Множення, в тому числі матричне множення
$x^y$	Піднесення до степені. Визначає $x$ в степені $y$ . В інших контекстах таке позначення використовується як нарядковий індекс, яке не слід інтерпретувати як піднесення до степені.
/	Цілочислове ділення з усиканням результату у бік нуля. Наприклад, $7/4$ і $-7/-4$ усикаються до 1, а $-7/4$ і $7/-4$ усикаються до -1.
$\frac{\div}{x}$	Використовується для позначення ділення в математичних рівняннях, де не передбачається усикання або округлення.
$\frac{y}{\sum_{i=x}^y f(i)}$	Використовується для позначення ділення в математичних рівняннях, де не передбачається усикання або округлення.
$\sum_{i=x}^y f(i)$	Підсумовування $f(i)$ , де $i$ приймає всі цілі значення від $x$ до $y$ включно.
$x\%y$	Операція по модулю. Залишок від розподілу $x$ на $y$ , визначена лише для цілих чисел $x$ і $y$ , де $x \geq 0$ і $y > 0$ .

#### Логічні оператори

Наведені нижче логічні оператори визначені так:

5  $x \&\& y$  Бульове логічне "і" для  $x$  і  $y$

$x \mid\mid y$  Бульове логічне "або" для  $x$  і  $y$

! Булеве логічне "не"

$x?y:z$  Якщо  $x$  дорівнює ІСТИНІ або не дорівнює 0, обчислюється значення  $y$ ; в іншому випадку обчислюється значення  $z$ .

#### Оператори відношення

10 Наведені нижче оператори відношення визначені так:

> Більше ніж

>= Більше ніж або дорівнює

< Менше ніж

<= Менше ніж або дорівнює

15 == Дорівнює

!= Не дорівнює

Коли оператор відношення застосовується до елемента або змінної синтаксису, якою присвоєне значення "на" (не застосовується), значення "на" обробляється як окреме значення для елемента або змінної синтаксису. Значення "на" не вважається таким, що дорівнює будь-якому іншому значенню.

20

#### Побітові оператори

Наведені нижче побітові оператори визначені так:

25 & Побітове "і". При роботі з цілочисловими аргументами оперує додатковим до двох представленням цілочислового значення. При роботі з двійковим аргументом, який містить менше бітів, ніж інший аргумент, коротший аргумент розширюється за допомогою додавання більш значущих бітів, які дорівнюють 0.

30 | Побітове "або". При роботі з цілочисловими аргументами оперує додатковим до двох представленням цілочислового значення. При роботі з двійковим аргументом, який містить менше бітів, ніж інший аргумент, коротший аргумент розширюється за допомогою додавання більш значущих бітів, які дорівнюють 0.

^ Побітове "виключне АБО". При роботі з цілочисловими аргументами оперує додатковим до двох представленням цілочислового значення. При роботі з двійковим аргументом, який містить менше бітів, ніж інший аргумент, коротший аргумент розширюється за допомогою додання більш значущих бітів, які дорівнюють 0.

35  $x \gg y$  Арифметичний зсув праворуч цілочислового представлення  $x$  з доповненням до двох на  $y$  двійкових цифр. Ця функція визначена лише для ненегативних цілих значень  $y$ . Біти, зсунені в старші значущі біти (MSB) внаслідок зсуву праворуч, мають значення, яке дорівнює MSB  $x$  до операції зсуву.

40  $x \ll y$  Арифметичний зсув ліворуч цілочислового представлення  $x$  з доповненням до двох на  $y$  двійкових цифр. Ця функція визначена лише для ненегативних цілих значень  $y$ . Біти, зсунені в молодші значущі біти (LSB) внаслідок зсуву ліворуч, мають значення, яке дорівнює 0.

Оператори присвоєння

Наведені нижче арифметичні оператори визначені так:

= Оператор присвоєння

5 ++ Приріст, тобто  $x++$  еквівалентне  $x=x+1$ ; при використанні в індексі масиву оцінює значення змінної перед операцією приросту.

-- Зменшення, тобто  $x--$  еквівалентне  $x=x-1$ ; при використанні в індексі масиву оцінює значення змінної перед операцією зменшення.

+= Приріст на вказану величину, тобто  $x+=3$  еквівалентне  $x=x+3$ , а  $x+=(-3)$  еквівалентне  $x=x+(-3)$ .

10 -= Зменшення на вказану величину, тобто  $x-=3$  еквівалентне  $x=x-3$ , а  $x-=(-3)$  еквівалентне  $x = x - (-3)$ .

Позначення діапазону

Наведене нижче позначення використовується для вказівки діапазону значень:

15  $x=y...z$   $x$  приймає цілочислові значення від  $y$  до  $z$  включно, де  $x$ ,  $y$  і  $z$  є цілими числами, і  $z$  є більшим ніж  $y$ .

Математичні функції

Визначені наведені нижче математичні функції:

$$\text{Abs}(x) = \begin{cases} x; & x \geq 0 \\ -x; & x < 0 \end{cases}$$

20  $\text{Asin}(x)$  – тригонометрична функція арксинуса, яка працює з аргументом  $x$ , що знаходиться в діапазоні від  $-1,0$  до  $1,0$  включно, з вихідним значенням в діапазоні від  $-\pi/2$  до  $\pi/2$  включно в одиницях числення радіан.

$\text{Atan}(x)$  – тригонометрична функція арктангенса, яка працює з аргументом  $x$ , з вихідним значенням в діапазоні від  $-\pi/2$  до  $\pi/2$  включно в одиницях числення радіан.

$$\text{Atan2}(y,x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) + \pi; & x < 0 \ \& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi; & x < 0 \ \& \ y < 0 \\ \frac{\pi}{2}; & x == 0 \ \& \ y \geq 0 \\ -\frac{\pi}{2}; & \text{інакше} \end{cases}$$

25  $\text{Ceil}(x)$  найменше ціле число, яке є більшим ніж або дорівнює  $x$ .

$$\text{Clip1Y}(x) = \text{Clip3}(0, (1 \ll \text{BitDepthY}) - 1, x)$$

$$\text{Clip1C}(x) = \text{Clip3}(0, (1 \ll \text{BitDepthC}) - 1, x)$$

$$\text{Clip3}(x,y,z) = \begin{cases} x; & z < x \\ y; & z > y \\ z; & \text{інакше} \end{cases}$$

30  $\text{Cos}(x)$  – тригонометрична функція косинуса, яка працює з аргументом  $x$  в одиницях числення радіан.

$$\text{Floor}(x) - \text{найбільша ціла кількість, яка є меншою ніж або дорівнює } x.$$

$$\text{GetCurrMsb}(a,b,c,d) = \begin{cases} c+d; & b-a \geq d/2 \\ c-d; & a-b > d/2 \\ c; & \text{інакше} \end{cases}$$

$\text{Ln}(x)$  – натуральний логарифм  $x$  (логарифм за основою  $e$ , де  $e$  – постійна основи натурального логарифма 2,781 281 828...).

35  $\text{Log2}(x)$  – логарифм  $x$  за основою 2.

$\text{Log10}(x)$  – логарифм  $x$  за основою 10.

$$\text{Min}(x,y) = \begin{cases} x; & x < y \\ y; & x > y \end{cases}$$

$$\text{Max}(x,y) = \begin{cases} x; & x \geq y \\ y; & x < y \end{cases}$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$



Кожен вираз "Якщо ... Інакше, Якщо ... Інакше, ..." в тексті вводиться словами «... як зазначено нижче" або «... застосовується таке", за чим відразу йде "Якщо ...». Останньою умовою "Якщо ... Інакше, Якщо ... Інакше, ..." завжди є "Інакше ...». Чергування визначень "Якщо ... Інакше, Якщо ... Інакше ..." може бути ідентифіковано за допомогою зіставлення «... як зазначено нижче" або «... застосовується таке" із заключним "Інакше ...».

5 У тексті викладення логічних операцій, яке б математично описувалося в такій формі:

якщо(умова 0a && умова 0b)

твердження 0

10 інакше якщо(умова 1a || умова 1b)

твердження 1

інакше

твердження n

може бути описане у такий спосіб:

15 ... як зазначено нижче /... застосовується наступне:

-- Якщо всі з наведених нижче умов істинні, твердження 0:

- умова 0a

- умова 0b

- Інакше, якщо істинною є одна або більше з наведених нижче умов, твердження 1:

- умова 1a

20 - умова 1b

- Інакше, твердження n

У тексті викладення логічних операцій, яке б математично описувалося в такій формі:

якщо (умова 0)

твердження 0

25 якщо (умова 1)

твердження 1

може бути описано у такий спосіб:

Коли умова 0, твердження 0

Коли умова 1, твердження 1.

30 Хоч варіанти здійснення даного винаходу були в основному описані на основі відеокодування, слід зазначити, що варіанти здійснення системи 10 кодування, кодера 20 і декодера 30 (і, відповідно, системи 10), а також інші варіанти здійснення, описані в даному документі, також можуть бути виконані з можливістю обробки або кодування нерухомого зображення, тобто обробки або кодування окремого зображення незалежно від будь-якого попереднього або наступного зображення, як при відеокодуванні. Загалом, лише модулі 244 інтер-прогнозування (кодер) і 344 (декодер) можуть бути недоступні у випадку, якщо кодування для обробки зображення обмежене одним зображенням 17. Всі інші функціональні можливості (які також називають інструментами або технологіями) відеокодера 20 і відеодекодера 30 можуть бути рівною мірою використані для обробки нерухомих зображень, наприклад, обчислення 204/304 залишку, перетворення 206, квантування 208, зворотного квантування 210/310, (зворотного) перетворення 212/312, розбивання 262/362, інтра-прогнозування 254/354 і/або контурного фільтрування 220, 320 та ентропійного кодування 270 і ентропійного декодування 304.

45 Варіанти здійснення, наприклад, кодера 20 і декодера 30, а також описані в даному документі функції, наприклад, з посиланням на кодер 20 і декодер 30, можуть бути реалізовані апаратними засобами, програмним забезпеченням, вбудованим програмним забезпеченням або будь-якою їхньою комбінацією. Якщо функції реалізовані програмним забезпеченням, вони можуть зберігатися на машинозчитуваному носії або передаватися через засоби зв'язку у вигляді однієї або більше інструкцій, або код виконуватися апаратним блоком обробки. 50 Машинозчитуваний носій може містити машинозчитуваний носій даних, який відповідає матеріальному носію, наприклад, носію даних, або каналу передачі, включаючи будь-який носій, який полегшує передачу комп'ютерної програми з одного місця в інше, наприклад, згідно з протоколом зв'язку. Отже, машинозчитувані носії зазвичай можуть відповідати (1) матеріальним машинозчитуваним носіям даних, призначеним для тривалого зберігання інформації, або (2) 55 каналам передачі, таким як звичайний сигнал або електромагнітний сигнал передачі інформації. Носії даних можуть бути будь-якими доступними носіями, до яких може отримати доступ один або більше комп'ютерів або один або більше процесорів для вилучення інструкцій, коду і/або структури даних для здійснення способів, описаних у даному винаході. Комп'ютерний програмний продукт може містити машинозчитуваний носій.

Як приклад, але не обмежуючись ними, такі машинозчитувані носії даних можуть містити RAM, ROM, EEPROM, CD-ROM або інший запам'ятовувальний пристрій на оптичному диску, запам'ятовувальний пристрій на магнітному диску або інші магнітні запам'ятовувальні пристрої, флеш-пам'ять або будь-який інший носій, який може бути використаний для зберігання  
 5 бажаного програмного коду у вигляді інструкцій або структур даних і може бути доступний для комп'ютера. Крім того, будь-яке належним чином виконане з'єднання також називають машинозчитуваним носієм. Так, наприклад, якщо інструкції передаються з веб-сайту, сервера або іншого віддаленого джерела за допомогою коаксіального кабелю, оптоволоконного кабелю, крученої пари, цифрової абонентської лінії (DSL) або бездротових технологій, таких як  
 10 інфрачервоний порт, радіо і мікрохвильове випромінювання, тоді коаксіальний кабель, оптоволоконний кабель, кручена пара, DSL або бездротові технології, такі як інфрачервоний порт, радіо і мікрохвильовий зв'язок, включені у визначення носія. Однак, слід розуміти, що машинозчитувані носії даних і сховища даних не включають з'єднання, несучі хвилі, сигнали або інші перехідні носії, а натомість спрямовані на неперехідні, матеріальні носії даних. Диск у  
 15 даному контексті включає компакт-диск (CD), лазерний диск, оптичний диск, цифровий універсальний диск (DVD), дискету і диск Blu-ray, де одні диски зазвичай відтворюють дані магнітним способом, а інші диски відтворюють дані оптично за допомогою лазерів. Комбінації згаданих вище носіїв також повинні бути включені до обсягу машинозчитуваних носіїв даних.

Інструкції можуть виконуватися одним або більше процесорами, такими як один або більше  
 20 процесорів цифрових сигналів (DSP), мікропроцесори загального призначення, спеціалізовані інтегральні схеми (ASIC), здатні до програмування логічні матриці (FPGA) або інші еквівалентні інтегральні або дискретні логічні схеми. Отже, термін "процесор", використовуваний у даному документі, може стосуватися будь-якої із зазначених вище структури або будь-якої іншої структури, придатної для реалізації способів, описаних у даному винаході. Крім того, в деяких  
 25 аспектах описані у даному винаході функціональні можливості можуть бути надані в рамках виділених апаратних і/або програмних модулів, сконфігурованих для кодування і декодування або включених до об'єданого кодека. Крім того, способи можуть бути повністю реалізовані в одній або більше схемах або логічних елементах.

Способи згідно з даним винаходом можуть бути реалізовані у великій кількості пристроїв або  
 30 апаратів, включаючи бездротовий телефон, інтегральну схему (IC) або набір IC (наприклад, набір мікросхем). У даному винаході описані різні компоненти, модулі або блоки, щоб підкреслити функціональні аспекти пристроїв, сконфігурованих для виконання розкритих технологій, як, однак, не обов'язково вимагають реалізації різними апаратними блоками. Швидше, як описано вище, різні блоки можуть бути об'єдані в апаратний блок кодека або  
 35 надані у вигляді набору взаємодіючих апаратних блоків, включаючи один або більше процесорів, як описано вище, разом із відповідним програмним забезпеченням і/або вбудованим програмним забезпеченням.

#### ФОРМУЛА ВИНАХОДУ

40 1. Спосіб інтра-прогнозування, який реалізовується пристроєм декодування, включає етапи, на яких:

отримують значення інформації вказівки поточного блока, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш  
 45 ймовірних режимів, при цьому набір найбільш ймовірних режимів містить планарний режим і 5 потенційно прийнятних режимів інтра-прогнозування, причому значення планарного режиму є 0; коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів, виводять режим інтра-прогнозування  
 50 IntraPredModeY[xCb][yCb] поточного блока за допомогою наступних впорядкованих етапів, на яких:

i) IntraPredModeY[xCb][yCb] встановлюють таким, що дорівнює  
 intra\_luma\_mpm\_remainder[xCb][yCb],

ii) значення IntraPredModeY[xCb][yCb] збільшують на одиницю,

iii) для i, що дорівнює від 0 до 4 включно, коли IntraPredModeY[xCb][yCb] більше або дорівнює  
 55 candModeList[i], значення IntraPredModeY[xCb][yCb] збільшують на одиницю, причому candModeList[i] представляє потенційно прийнятний режим інтра-прогнозування в наборі найбільш ймовірних режимів, що відрізняється від планарного режиму,

при цьому intra\_luma\_mpm\_remainder[xCb][yCb] представляє режим інтра-прогнозування  
 60 залишку, місцеположення яскравості (xCb, yCb) задає верхній лівий дискретний відлік поточного блока відносно верхнього лівого дискретного відліку яскравості поточного зображення;

- здійснюють інтра-прогнозування на основі виведеного режиму інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  і відповідних опорних дискретних відліків, при цьому виведення потенційно прийняттого режиму інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш ймовірних режимів здійснюють за допомогою:
- 5 коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0\dots 3$  і, для кожного  $i, j=(i+1)\dots 4$ , обидва значення міняються місцями таким чином:  
 $(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .
2. Спосіб за п. 1, який **відрізняється** тим, що значення інформації вказівки вказується прапором  $\text{intra\_luma\_mpm\_flag}$ .
- 10 3. Спосіб за п. 2, який **відрізняється** тим, що, коли значення  $\text{intra\_luma\_mpm\_flag}$  не дорівнює 1, режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів.
4. Спосіб інтра-прогнозування, який реалізовується пристроєм кодування, включає етапи, на яких:
- 15 виводять режим інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  поточного блока за допомогою наступних впорядкованих етапів, коли режим інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  поточного блока не міститься в наборі найбільш ймовірних режимів, при цьому набір найбільш ймовірних режимів містить планарний режим і 5 потенційно прийнятних режимів інтра-прогнозування, причому значення планарного режиму є 0:
- 20 i)  $\text{IntraPredModeY}[xCb][yCb]$  встановлюють таким, що дорівнює  $\text{intra\_luma\_mpm\_remainder}[xCb][yCb]$ ,  
 ii) значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшують на одиницю;  
 iii) для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[xCb][yCb]$  більше або дорівнює  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшують на одиницю, де  $\text{candModeList}[i]$
- 25 представляє потенційно прийнятний режим інтра-прогнозування в наборі найбільш ймовірних режимів, що відрізняється від планарного режиму, при цьому місцеположення яскравості ( $xCb, yCb$ ) задає лівий верхній дискретний відлік поточного блока відносно верхнього лівого дискретного відліку яскравості поточного зображення;
- 30 передають потік бітів в декодер, при цьому потік бітів включає в себе значення інформації вказівки, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш ймовірних режимів, при цьому виведення потенційно прийняттого режиму інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш ймовірних режимів здійснюють за допомогою:
- 35 коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0\dots 3$  і, для кожного  $i, j=(i+1)\dots 4$ , обидва значення міняються місцями таким чином:  
 $(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .
5. Спосіб за п. 4, який **відрізняється** тим, що значення інформації вказівки вказується прапором  $\text{intra\_luma\_mpm\_flag}$ .
- 40 6. Спосіб за п. 5, який **відрізняється** тим, що, коли значення  $\text{intra\_luma\_mpm\_flag}$  не дорівнює 1, режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів.
7. Декодер (30), який містить схему обробки для виконання способу за будь-яким з пп. 1-3.
8. Кодер (20), який містить схему обробки для виконання способу за будь-яким з пп. 4-6.
- 45 9. Машиночитаний носій, який містить програмний код для виконання способу за будь-яким з пп. 1-6.
10. Декодер для інтра-прогнозування, який містить:  
 один або більше процесорів; і  
 довготривалий машиночитаний носій, який з'єднаний з процесорами і зберігає програмну
- 50 частину для виконання процесорами, причому програмна частина, коли виконується процесорами, конфігурує декодер для виконання способу за будь-яким з пп. 1-3.
11. Кодер для інтра-прогнозування, який містить:  
 один або більше процесорів; і  
 довготривалий машиночитаний носій, який з'єднаний з процесорами і зберігає програмну
- 55 частину для виконання процесорами, причому програмна частина, коли виконується процесорами, конфігурує кодер для виконання способу за будь-яким з пп. 4-6.
12. Пристрій декодування для інтра-прогнозування, який містить:  
 блок (1401) отримання, сконфігурований для отримання значення інформації вказівки поточного
- 60 блока, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш ймовірних режимів, при цьому набір найбільш ймовірних

режимів містить планарний режим і 5 потенційно прийнятних режимів інтра-прогнозування, причому значення планарного режиму є 0;

блок виведення (1402), сконфігурований для виведення режиму інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  поточного блока за допомогою наступних впорядкованих етапів, коли значення інформації вказівки вказує, що режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів:

i)  $\text{IntraPredModeY}[xCb][yCb]$  встановлюється таким, що дорівнює  $\text{intra\_luma\_mpm\_remainder}[xCb][yCb]$ ,

ii) значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшується на одиницю,

iii) для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[xCb][yCb]$  більше або дорівнює  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшується на одиницю, причому  $\text{candModeList}[i]$  представляє потенційно прийнятний режим інтра-прогнозування в наборі найбільш ймовірних режимів, що відрізняється від планарного режиму,

при цьому  $\text{intra\_luma\_mpm\_remainder}[xCb][yCb]$  представляє режим інтра-прогнозування залишку, місцеположення яскравості ( $xCb$ ,  $yCb$ ) задає верхній лівий дискретний відлік поточного блока відносно верхнього лівого дискретного відліку яскравості поточного зображення, при цьому блок виведення додатково сконфігурований виконувати інтра-прогнозування на основі виведеного режиму інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  і відповідних опорних дискретних відліків,

при цьому блок виведення додатково сконфігурований для виведення потенційно прийнятного режиму інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш ймовірних режимів за допомогою:

коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0...3$  і, для кожного  $i$ ,  $j=(i+1)...4$ , обидва значення міняються місцями таким чином:

$(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .

13. Пристрій за п. 12, який **відрізняється** тим, що значення інформації вказівки вказується прапором  $\text{intra\_luma\_mpm\_flag}$ .

14. Пристрій за п. 13, який **відрізняється** тим, що, коли значення  $\text{intra\_luma\_mpm\_flag}$  не дорівнює 1, режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів.

15. Пристрій кодування для реалізації інтра-прогнозування, який містить:

блок виведення (1501), сконфігурований для виведення режиму інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  поточного блока за допомогою наступних впорядкованих етапів, коли режим інтра-прогнозування  $\text{IntraPredModeY}[xCb][yCb]$  поточного блока не міститься в наборі найбільш ймовірних режимів, при цьому набір найбільш ймовірних режимів містить планарний режим і 5 потенційно прийнятних режимів інтра-прогнозування, причому значення планарного режиму є 0:

i)  $\text{IntraPredModeY}[xCb][yCb]$  встановлюється таким, що дорівнює  $\text{intra\_luma\_mpm\_remainder}[xCb][yCb]$ ,

ii) значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшується на одиницю;

iii) для  $i$ , що дорівнює від 0 до 4 включно, коли  $\text{IntraPredModeY}[xCb][yCb]$  більше або дорівнює  $\text{candModeList}[i]$ , значення  $\text{IntraPredModeY}[xCb][yCb]$  збільшується на одиницю, де  $\text{candModeList}[i]$  представляє потенційно прийнятний режим інтра-прогнозування в наборі найбільш ймовірних режимів, що відрізняється від планарного режиму,

при цьому місцеположення яскравості ( $xCb$ ,  $yCb$ ) задає лівий верхній дискретний відлік поточного блока відносно верхнього лівого дискретного відліку яскравості поточного зображення;

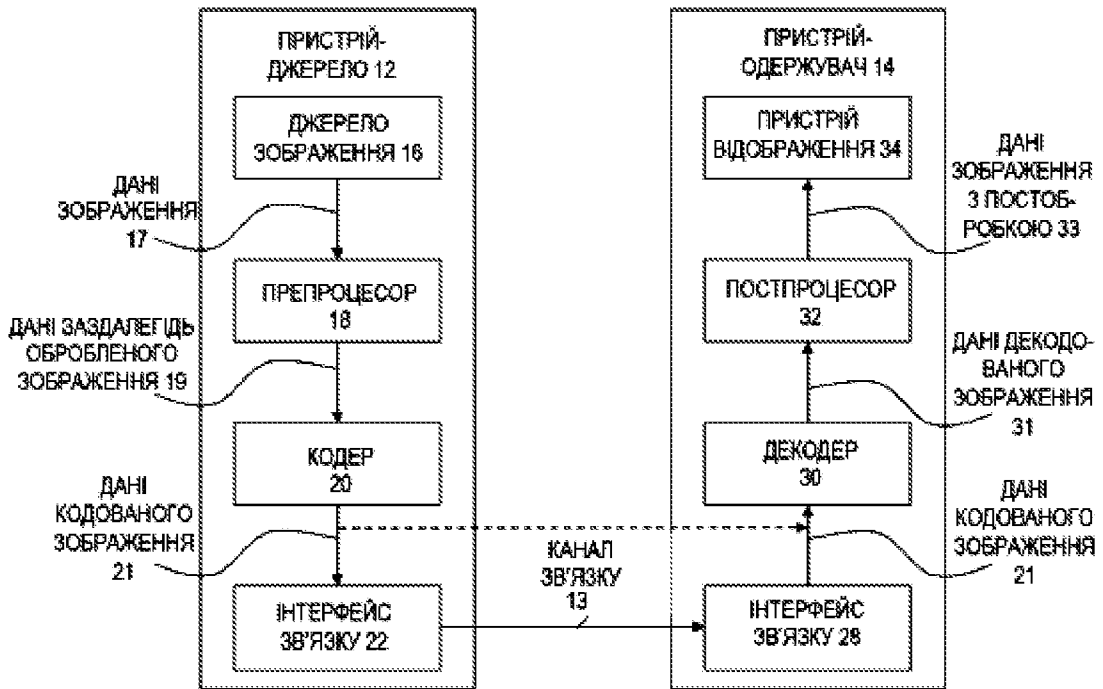
блок (1502) передачі, сконфігурований для передачі потоку бітів в декодер, при цьому потік бітів включає в себе значення інформації вказівки, причому значення інформації вказівки вказує, чи міститься режим інтра-прогнозування поточного блока в наборі найбільш ймовірних режимів, при цьому блок виведення додатково сконфігурований для виведення потенційно прийнятного режиму інтра-прогнозування  $\text{candModeList}[i]$  в наборі найбільш ймовірних режимів за допомогою:

коли  $\text{candModeList}[i]$  є більшим, ніж  $\text{candModeList}[j]$  для  $i=0...3$  і, для кожного  $i$ ,  $j=(i+1)...4$ , обидва значення міняються місцями таким чином:

$(\text{candModeList}[i], \text{candModeList}[j]) = \text{swap}(\text{candModeList}[i], \text{candModeList}[j])$ .

16. Пристрій за п. 15, який **відрізняється** тим, що значення інформації вказівки вказується прапором  $\text{intra\_luma\_mpm\_flag}$ .

17. Пристрій за п. 16, який **відрізняється** тим, що, коли значення `intra_luma_tpm_flag` не дорівнює 1, режим інтра-прогнозування поточного блока не міститься в наборі найбільш ймовірних режимів.



Фіг. 1А



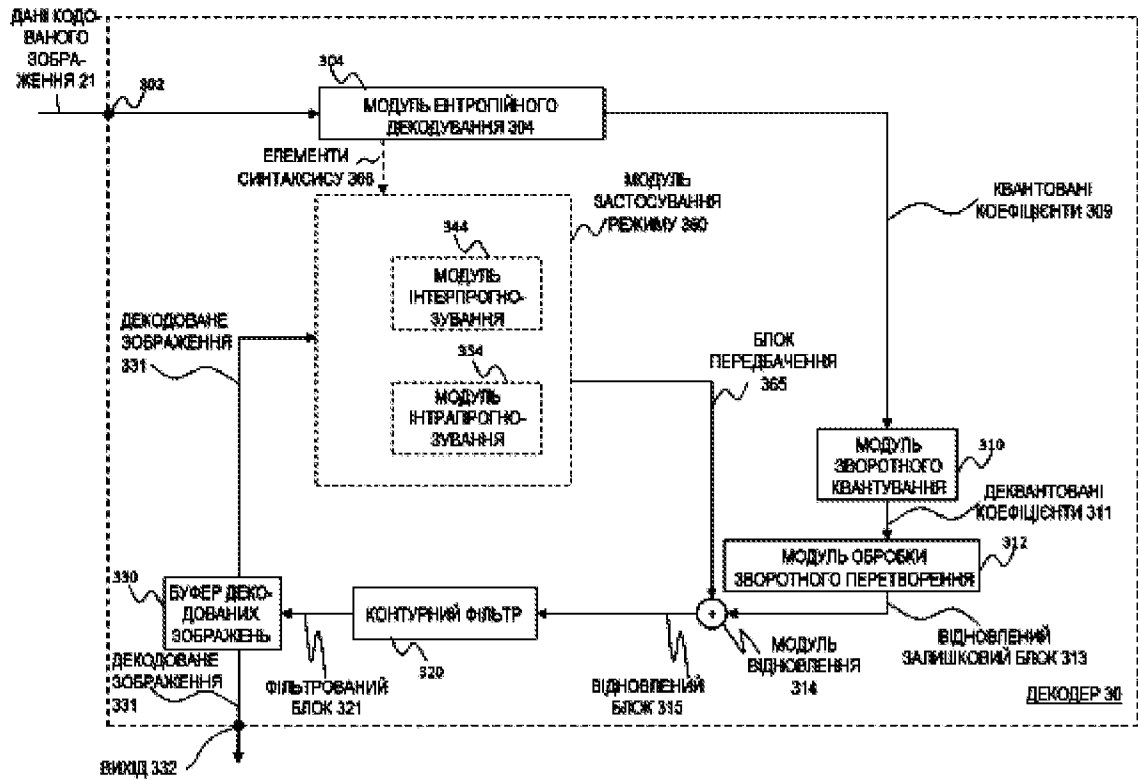


Fig. 3

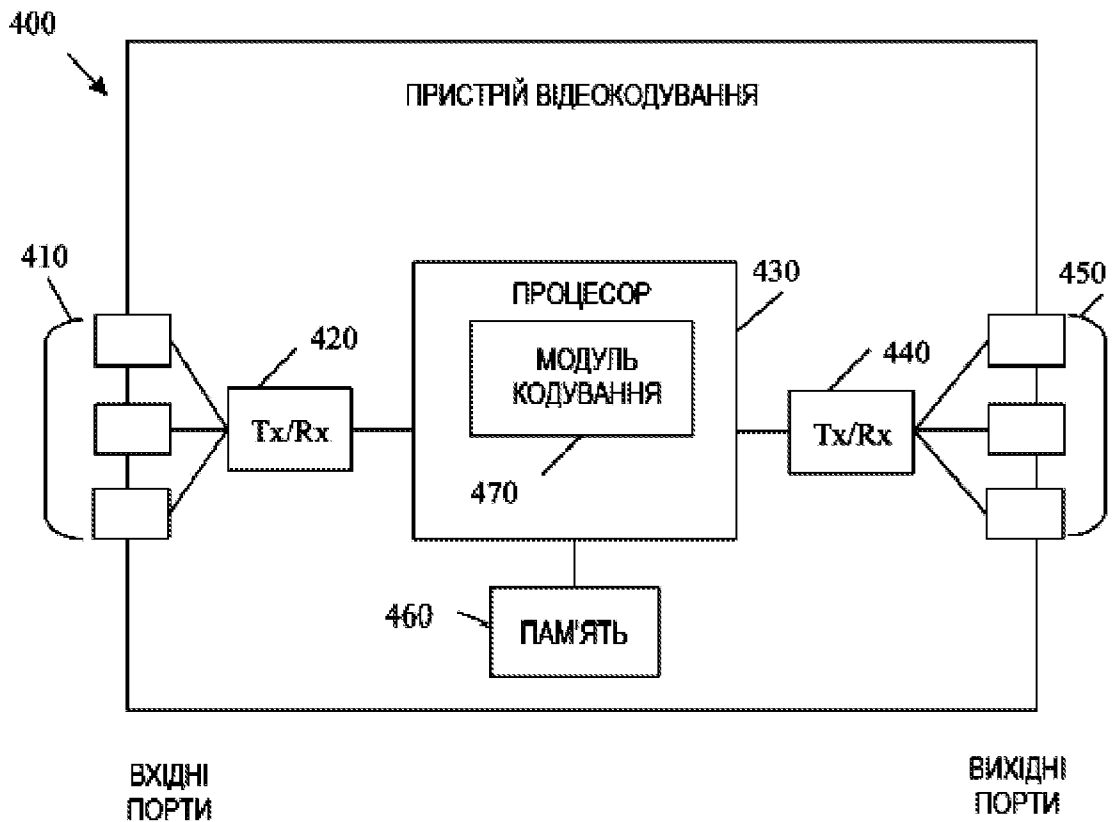
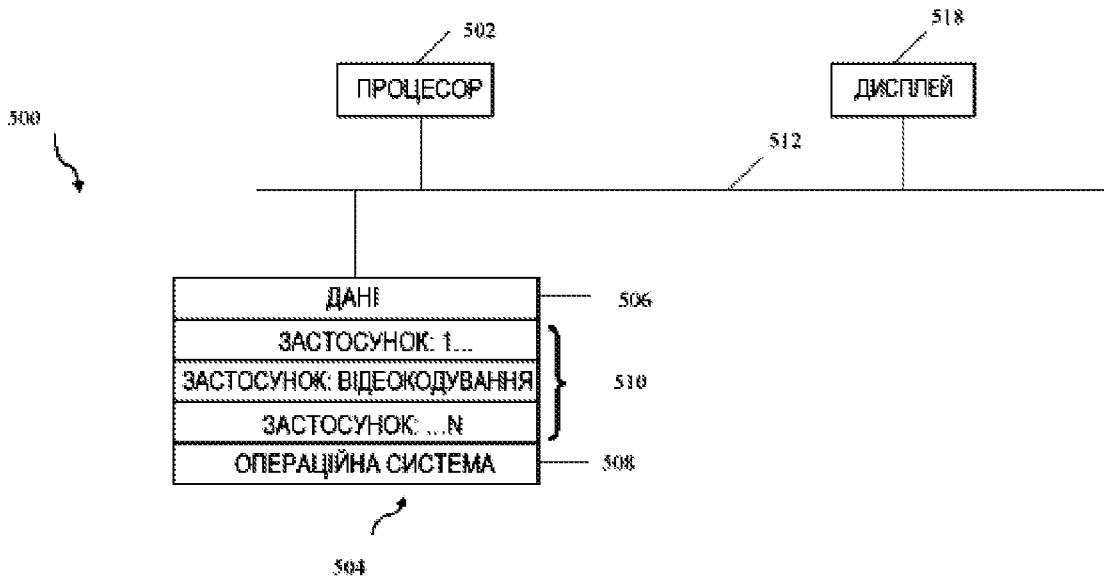
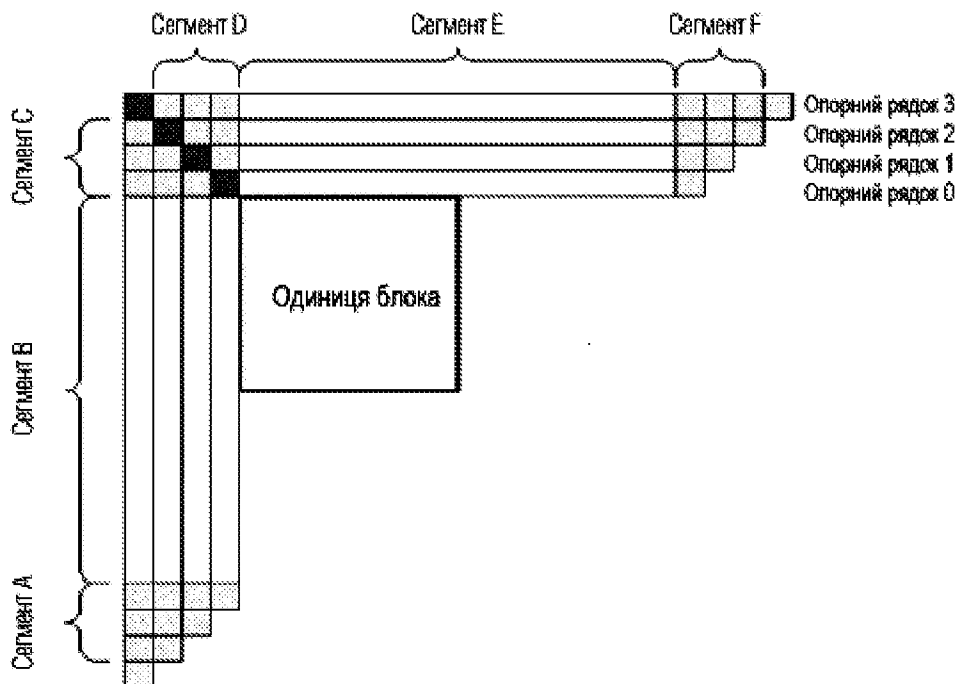


Fig. 4

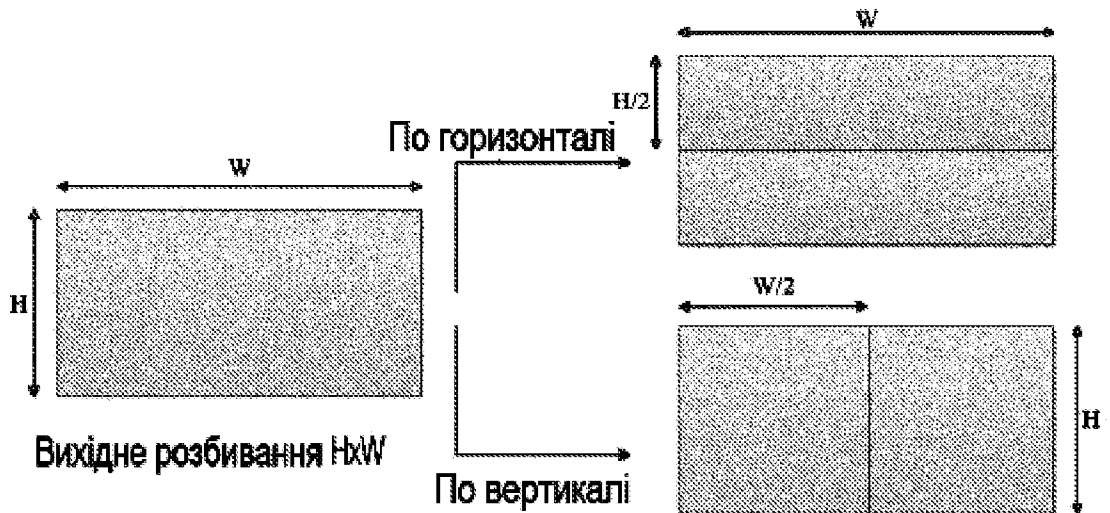


Фіг. 5



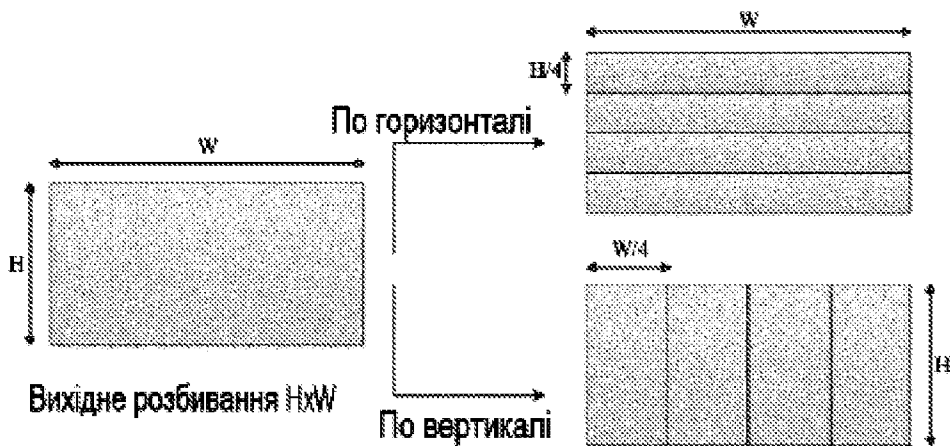
Інструмент кодування з множинними опорними рядками. Коли індекс опорного рядка дорівнює 0, використовується найближчий сусідній опорний рядок до поточного блока

Фіг. 6



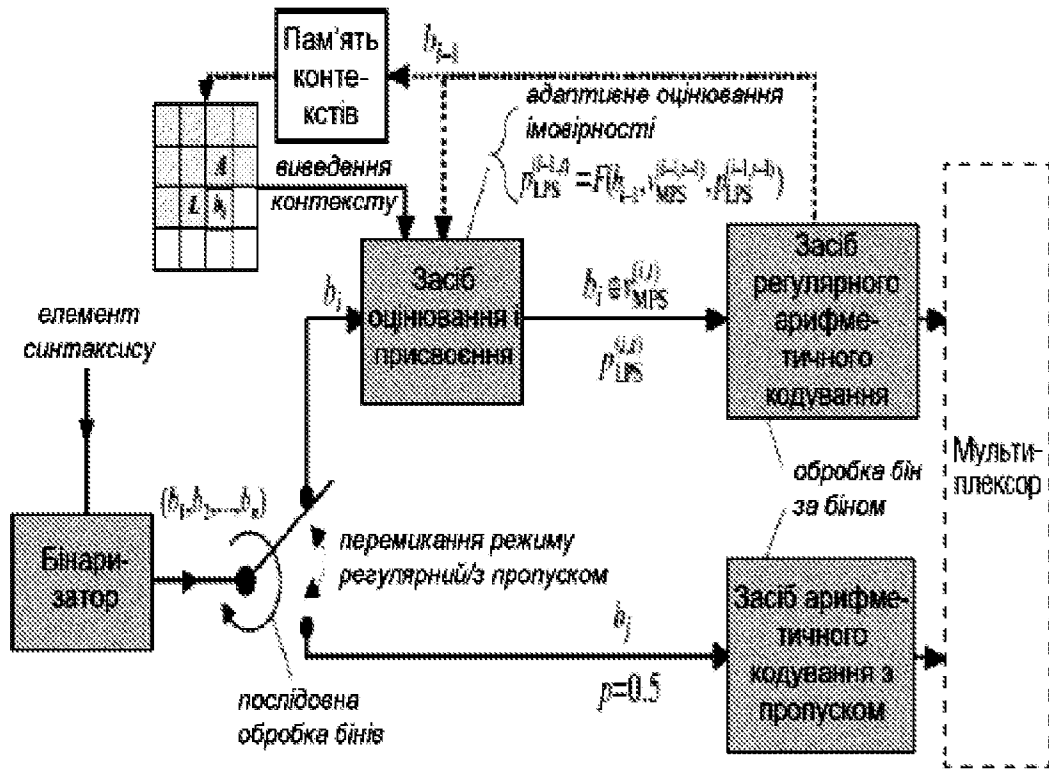
Приклад розділення блоків  $4 \times 8$ ,  $8 \times 4$  і  $4 \times 4$

Фіг. 7

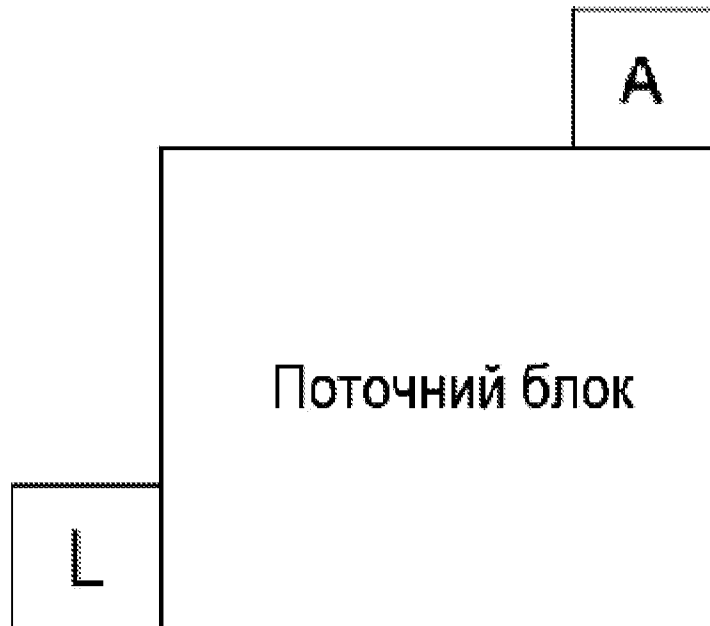


Приклад розділення всіх блоків, за винятком  $4 \times 8$ ,  $8 \times 4$  і  $4 \times 4$

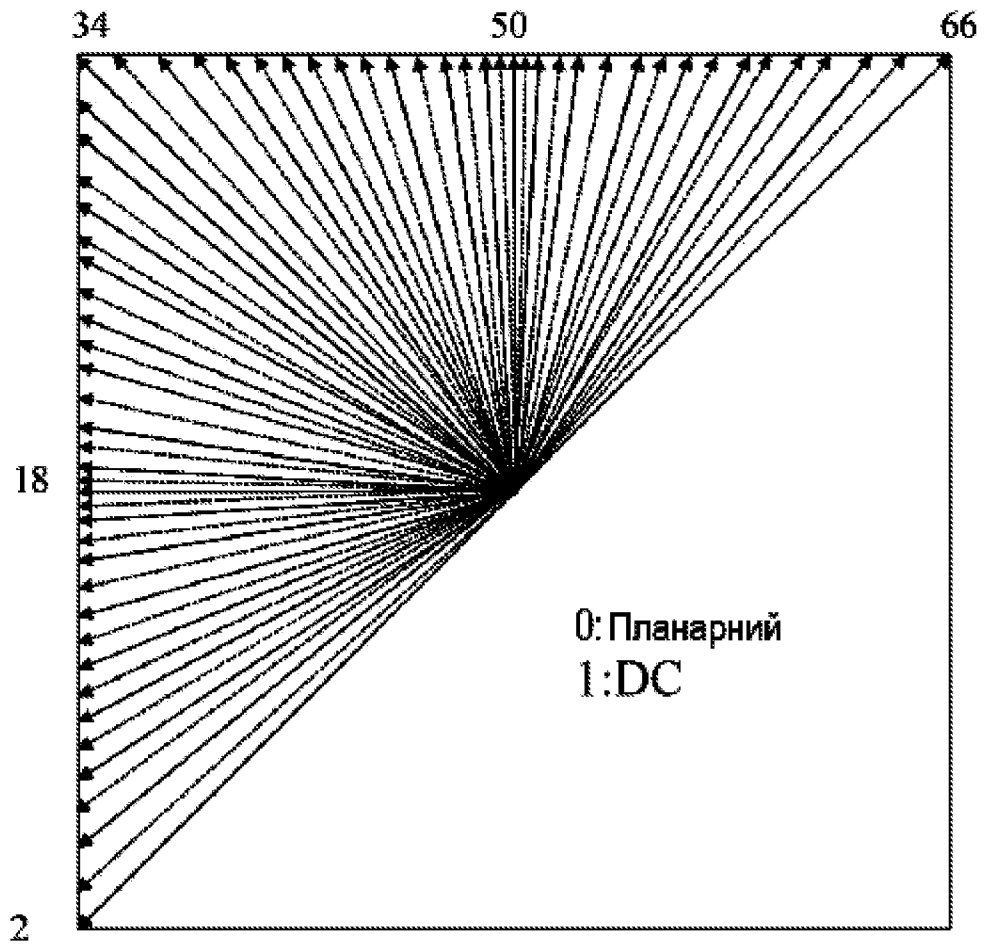
Фіг. 8



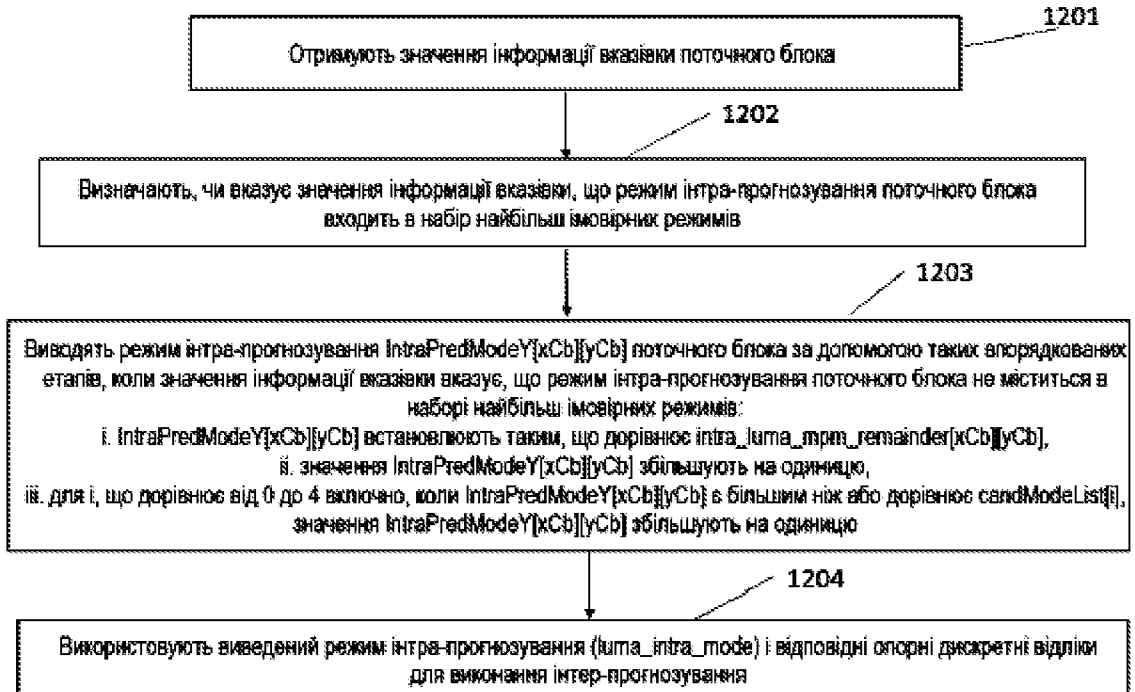
Фіг. 9



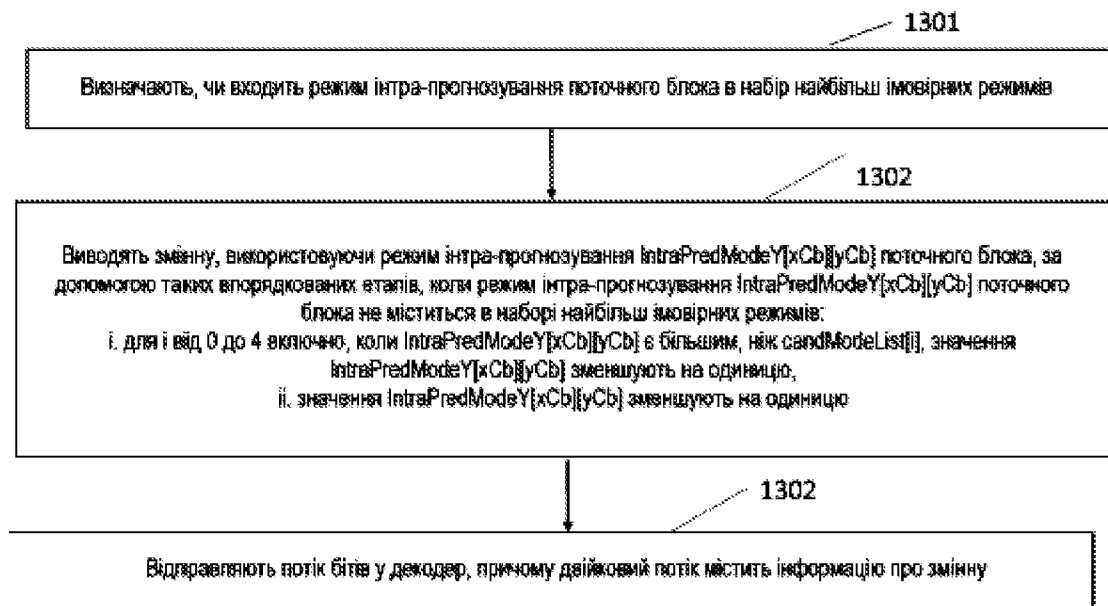
Фіг. 10



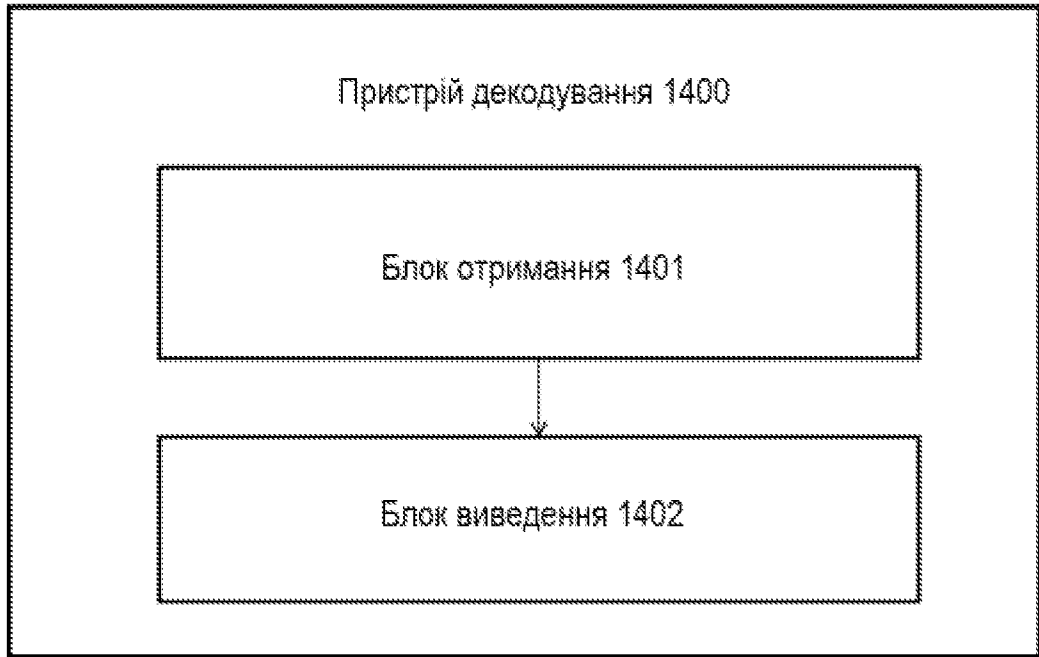
Фиг. 11



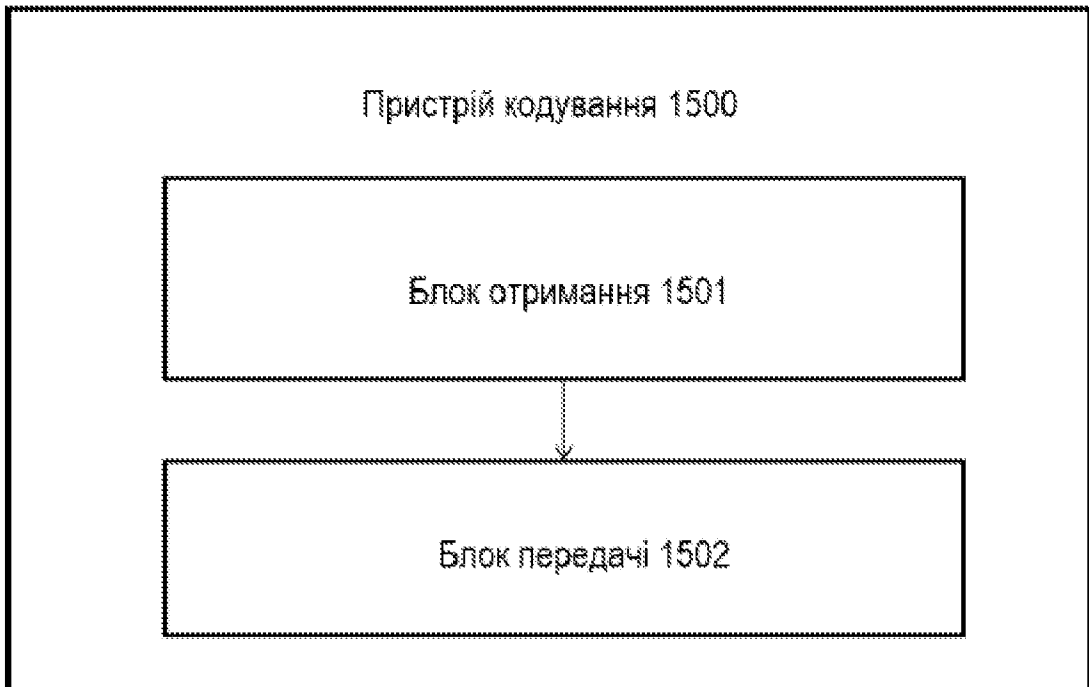
Фіг. 12



Фіг. 13



Фіг. 14



Фіг. 15

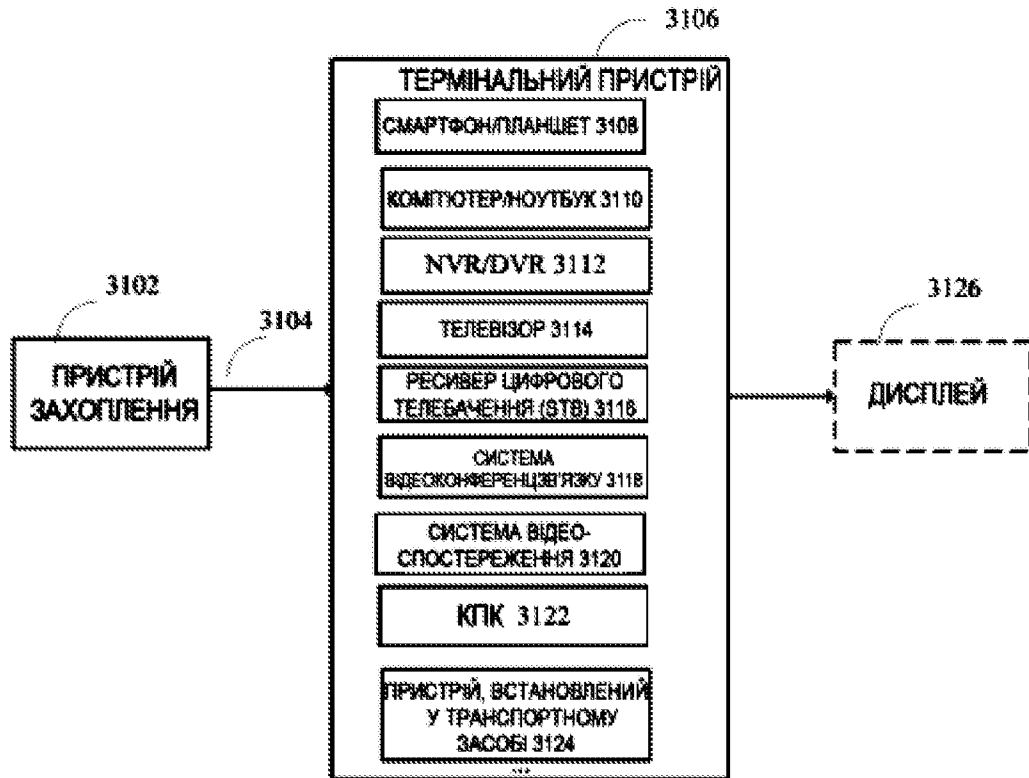


Fig. 16

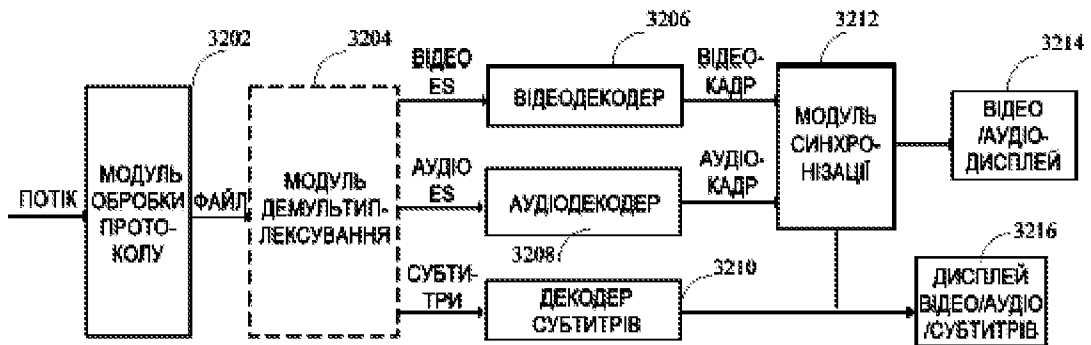


Fig. 17