



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/12 (2021.02); H04N 19/157 (2021.02); H04N 19/176 (2021.02)

(21)(22) Заявка: 2019119077, 15.12.2017

(24) Дата начала отсчета срока действия патента:  
15.12.2017

Дата регистрации:  
09.11.2021

Приоритет(ы):

(30) Конвенционный приоритет:  
28.12.2016 JP 2016-256732

(43) Дата публикации заявки: 21.12.2020 Бюл. № 36

(45) Опубликовано: 09.11.2021 Бюл. № 31

(85) Дата начала рассмотрения заявки РСТ на  
национальной фазе: 19.06.2019

(86) Заявка РСТ:  
JP 2017/045042 (15.12.2017)

(87) Публикация заявки РСТ:  
WO 2018/123644 (05.07.2018)

Адрес для переписки:  
101000, Москва, ул. Мясницкая, 13, стр. 5, ООО  
"Союзпатент"

(72) Автор(ы):

ЦУКУБА Такеси (JP)

(73) Патентообладатель(и):

СОНИ КОРПОРЕЙШН (JP)

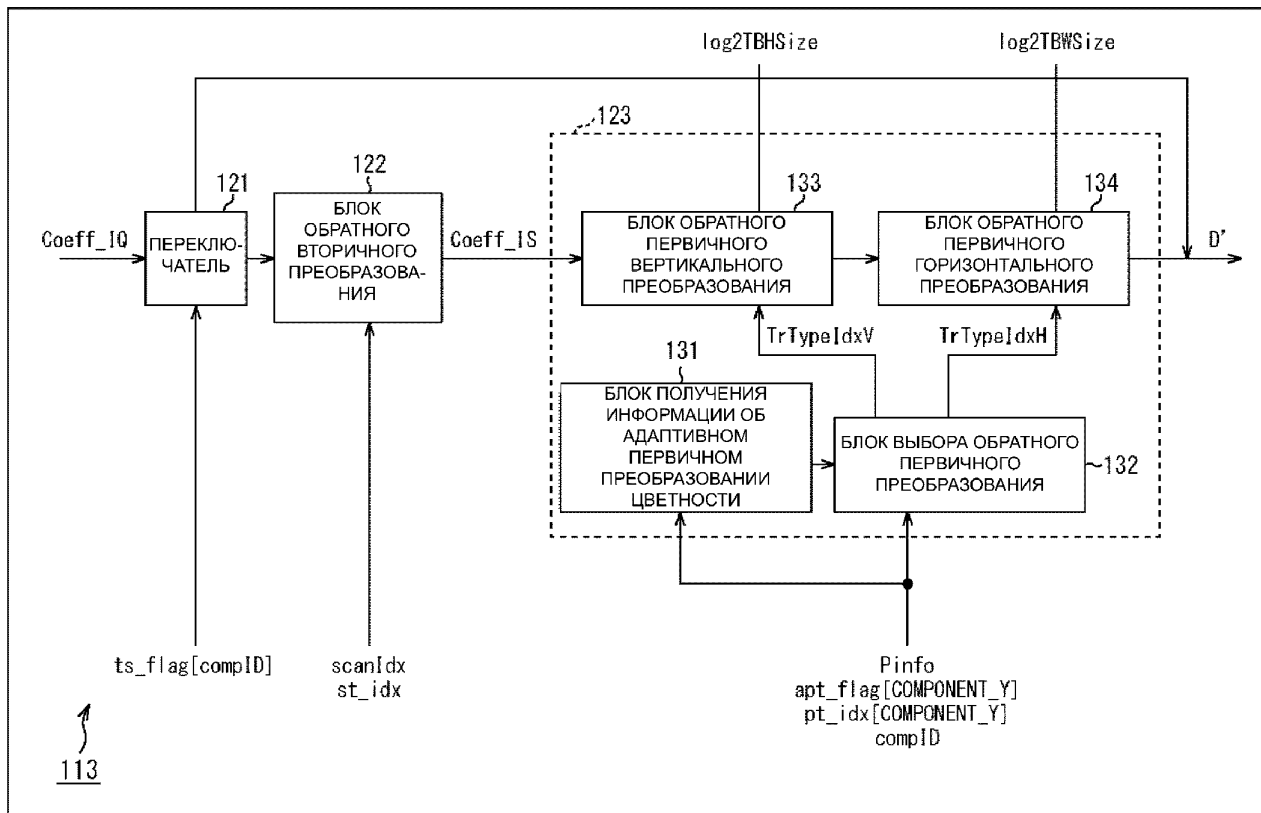
(56) Список документов, цитированных в отчете  
о поиске: US 8098940 B2, 17.01.2012. WO 2013/  
008800 A1, 17.01.2013. WO 2013/088687 A1,  
20.06.2013. WO 2011/155332 A1, 15.12.2011. US  
9167265 B2, 20.10.2015.

(54) УСТРОЙСТВО И СПОСОБ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

(57) Реферат:

Изобретение относится к средствам для обработки изображений. Технический результат заключается в повышении эффективности кодирования. Обратное ортогональное преобразование цветности выполняется с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости. Например, информация, касающаяся ортогонального преобразования, включает в себя флаг адаптивного первичного преобразования,

указывающий, должно ли быть применено к обрабатываемому блоку преобразования адаптивное первичное преобразование с адаптивным выбором одного из множества различных ортогональных преобразований и с использованием выбранного преобразования в качестве первичного преобразования. Настоящее раскрытие может быть применено, например, к устройству обработки изображений, устройству кодирования изображений, устройству декодирования изображений или тому подобному. 4 н. и 16 з.п. ф-лы, 77 ил.



Фиг. 10



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

*H04N 19/12* (2014.01)*H04N 19/157* (2014.01)*H04N 19/176* (2014.01)(12) **ABSTRACT OF INVENTION**

(52) CPC

*H04N 19/12 (2021.02); H04N 19/157 (2021.02); H04N 19/176 (2021.02)*(21)(22) Application: **2019119077, 15.12.2017**(24) Effective date for property rights:  
**15.12.2017**Registration date:  
**09.11.2021**

Priority:

(30) Convention priority:  
**28.12.2016 JP 2016-256732**(43) Application published: **21.12.2020 Bull. № 36**(45) Date of publication: **09.11.2021 Bull. № 31**(85) Commencement of national phase: **19.06.2019**(86) PCT application:  
**JP 2017/045042 (15.12.2017)**(87) PCT publication:  
**WO 2018/123644 (05.07.2018)**Mail address:  
**101000, Moskva, ul. Myasnitskaya, 13, str. 5, OOO  
"Soyuzpatent"**

(72) Inventor(s):

**TSUKUBA Takeshi (JP)**

(73) Proprietor(s):

**SONY CORPORATION (JP)**(54) **DEVICE AND METHOD FOR IMAGE PROCESSING**

(57) Abstract:

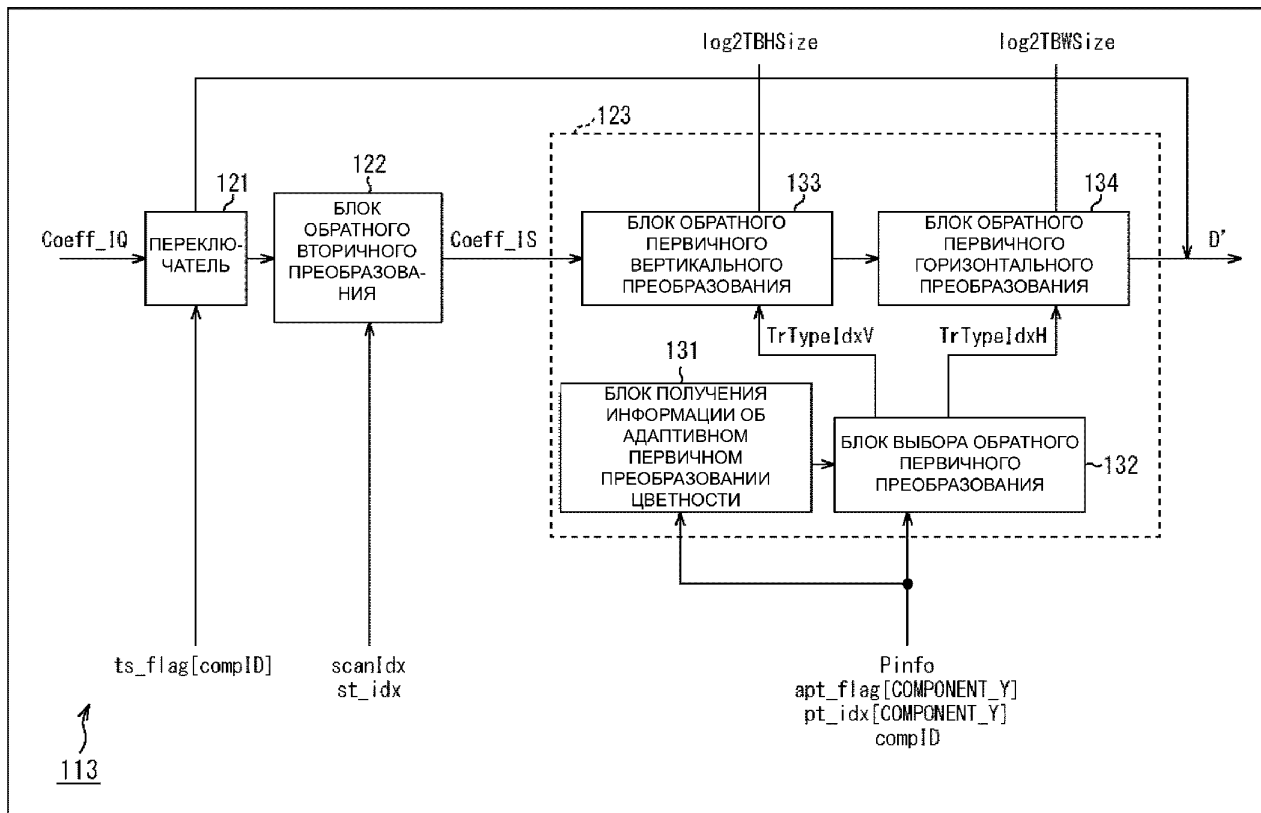
FIELD: image processing.

SUBSTANCE: invention relates to means for image processing. The inverse orthogonal chroma transformation is performed using information about the inverse orthogonal chroma transformation obtained based on information about the inverse orthogonal brightness transformation. For example, information related to the orthogonal transformation includes a flag of adaptive primary transformation indicating, whether adaptive primary transformation should be applied to

the processed transformation block with adaptive selection of one of many different orthogonal transformations and using the selected transformation as the primary transformation. The present disclosure may be applied, for example, to an image processing device, an image encoding device, an image decoding device, and so forth.

EFFECT: increase in the coding efficiency.

20 cl, 77 dwg



Фиг. 10

Область техники, к которой относится изобретение

Настоящее раскрытие относится к устройству и способу обработки изображений и, в частности, к устройству и способу обработки изображений для ограничения снижения эффективности кодирования.

5 Уровень техники

В уровне техники описано адаптивное первичное преобразование (адаптивное многоядерное преобразование или АМТ) для адаптивного выбора первичного преобразования из множества различных ортогональных преобразований для каждого первичного преобразования в горизонтальном направлении PThor (которое также упоминается как первичное горизонтальное преобразование) и первичного преобразования в вертикальном направлении PTver (которое также упоминается как первичное вертикальное преобразование) каждой единицы преобразования (TU) для яркости (например, смотри непатентную литературу 1).

Перечень цитируемой литературы

15 Непатентная литература

Непатентная литература 1: Jianle Chen, Elena Alshina, Gary J. Sullivan, Jens-Rainer, and Jill Boyce, "Algorithm Description of Joint Exploration Test Model 4" JVET-D1001\_v3, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 4th Meeting at Chengdu, China, October 15th to 21st, 2016

20 Раскрытие сущности изобретения

Техническая задача

Однако в непатентной литературе 1, хотя адаптивное первичное преобразование может применяться только к яркости (Y), оно не применяется к цветностям (Cb и Cr). Таким образом, существует проблема эффективности кодирования первичного преобразования для цветностей, которые ухудшаются в большей степени по сравнению с первичным преобразованием для яркости.

Настоящее раскрытие учитывает вышеупомянутые обстоятельства и нацелено на ограничение снижения эффективности кодирования.

Решение технической задачи

30 Устройство обработки изображений первого аспекта настоящей технологии представляет собой устройство обработки изображений, включающее в себя: блок обратного ортогонального преобразования, выполненный с возможностью выполнения обратного ортогонального преобразования цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

35 Информация, касающаяся обратного ортогонального преобразования, может включать в себя флаг адаптивного первичного преобразования, указывающий то, следует ли применять адаптивное обратное первичное преобразование для адаптивного выбора одного из множества различных обратных ортогональных преобразований и использования выбранного преобразования в качестве обратного первичного преобразования.

Значение флага адаптивного первичного преобразования цветности может быть установлено равным значению флага адаптивного первичного преобразования яркости.

45 Информация, касающаяся обратного ортогонального преобразования, может включать в себя идентификатор первичного преобразования, указывающий на то, какое обратное первичное преобразование следует применить к обратным первичным преобразованиям в вертикальном направлении и в горизонтальном направлении.

Значение идентификатора первичного преобразования цветности может быть

установлено равным значению идентификатора первичного преобразования яркости в случае, когда флаг адаптивного первичного преобразования цветности имеет значение "истина", и может быть установлен в заданное значение в случае, когда, флаг адаптивного первичного преобразования цветности имеет значение "ложь".

5 Блок обратного ортогонального преобразования может выполнить обратное первичное преобразование в качестве обратного ортогонального преобразования.

Блок обратного ортогонального преобразования может выполнить обратное первичное горизонтальное преобразование, которое представляет собой обратное первичное преобразование в горизонтальном направлении, и обратное первичное  
10 вертикальное преобразование, которое представляет собой обратное первичное преобразование в вертикальном направлении, в качестве обратного ортогонального преобразования.

Значение флага адаптивного первичного преобразования цветности может быть установлено равным значению флага адаптивного первичного преобразования яркости  
15 в случае, когда тип предсказания блока кодирования, к которому принадлежит обрабатываемый блок преобразования, представляет собой внутриблочное предсказание.

Значение флага адаптивного первичного преобразования цветности может быть установлено равным значению флага адаптивного первичного преобразования яркости  
20 в случае, когда типом предсказания блока кодирования, к которому принадлежит обрабатываемый блок преобразования, является внутриблочное предсказание, или в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование.

Значение флага адаптивного первичного преобразования цветности может быть  
25 установлено равным значению флага адаптивного первичного преобразования яркости в случае, когда типом предсказания блока кодирования, к которому принадлежит обрабатываемый блок преобразования, является внутриблочное предсказание, или в случае, когда типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом.

30 Значение флага адаптивного первичного преобразования цветности может быть установлено равным значению флага адаптивного первичного преобразования яркости в случае, когда флаг вывода информации об адаптивном первичном преобразовании цветности, указывающий на то, следует ли выводить флаг адаптивного первичного преобразования цветности на основании флага адаптивного первичного преобразования  
35 яркости, имеет значение "истина".

Значение флага адаптивного первичного преобразования цветности может быть установлено равным значению флага адаптивного первичного преобразования яркости в случае, когда размер короткой стороны обрабатываемого блока преобразования цветности больше или равен значению заданного порогового значения.

40 Блок обратного ортогонального преобразования может быть выполнен с возможностью выполнения обратного первичного горизонтального преобразования, которое представляет собой обратное первичное преобразование в горизонтальном направлении, и обратного первичного вертикального преобразования, которое представляет собой обратное первичное преобразование в вертикальном направлении,  
45 в качестве обратного ортогонального преобразования, тип преобразования обратного первичного горизонтального преобразования может быть установлен на основании набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования в случае, когда размер ширины картинки

обрабатываемого блока преобразования цветности больше значения заданного порогового значения, и тип преобразования обратного первичного вертикального преобразования может быть установлен на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования в случае, когда размер высоты картинки обрабатываемого блока преобразования цветности больше значения заданного порогового значения.

Информация, касающаяся обратного ортогонального преобразования, может включать в себя флаг пропуска преобразования, указывающий на то, должен ли быть пропущен процесс обратного ортогонального преобразования.

Значение флага пропуска преобразования цветности может быть установлено равным значению флага пропуска преобразования яркости.

Информация, касающаяся обратного ортогонального преобразования, может включать в себя идентификатор вторичного преобразования, указывающий на то, какое обратное вторичное преобразование должно быть применено.

Значение идентификатора вторичного преобразования цветности может быть установлено равным значению идентификатора вторичного преобразования яркости.

Способ обработки изображений первого аспекта настоящей технологии представляет собой способ обработки изображений, включающий в себя: выполнение обратного ортогонального преобразования цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

Устройство обработки изображений второго аспекта настоящей технологии представляет собой устройство обработки изображений, включающее в себя: блок ортогонального преобразования, выполненный с возможностью выполнения ортогонального преобразования цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

Способ обработки изображений второго аспекта настоящей технологии представляет собой способ обработки изображений, включающий в себя: выполнение ортогонального преобразования цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

В устройстве и способе обработки изображений первого аспекта настоящей технологии обратное ортогональное преобразование цветности выполняется с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

В устройстве и способе обработки изображений второго аспекта настоящей технологии ортогональное преобразование цветности выполняется с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

Преимущественные эффекты изобретения

Согласно настоящему раскрытию изображения могут быть обработаны. В частности, можно ограничивать снижение эффективности кодирования.

Краткое описание чертежей

Фиг.1 – схема, иллюстрирующая отношение соответствия между наборами преобразования и выбранными ортогональными преобразованиями.

Фиг.2 – схема, иллюстрирующая отношение соответствия между типами

ортогонального преобразования и функциями, которые должны использоваться.

Фиг.3 – схема, иллюстрирующая отношение соответствия между наборами преобразования и режимами предсказания.

Фиг.4 – схема, иллюстрирующая пример синтаксиса и семантики единицы преобразования.

Фиг.5 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.6 – схема, иллюстрирующая примеры параметров цветности, полученных с использованием значений яркости.

Фиг.7 – схема, иллюстрирующая пример способа получения флагов адаптивного первичного преобразования цветности.

Фиг.8 – схема, иллюстрирующая формы CU, PU и TU.

Фиг.9 – блок-схема, иллюстрирующая пример основной конфигурации устройства декодирования изображения.

Фиг.10 – блок-схема, иллюстрирующая пример основной конфигурации блока обратного преобразования.

Фиг.11 – блок-схема, иллюстрирующая пример основной конфигурации блока получения информации об адаптивном первичном преобразовании цветности.

Фиг.12 – блок-схема, описывающая пример последовательности операций процесса декодирования изображения.

Фиг.13 – блок-схема, описывающая пример последовательности операций процесса декодирования информации о первичном преобразовании.

Фиг.14 – блок-схема, описывающая пример последовательности операций процесса обратного преобразования.

Фиг.15 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.16 – блок-схема, описывающая пример последовательности операций процесса выбора обратного первичного преобразования.

Фиг.17 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.18 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.19 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.20 – схема, иллюстрирующая пример синтаксиса и семантики единицы преобразования.

Фиг.21 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.22 – блок-схема, описывающая пример последовательности операций процесса декодирования флага вывода информации адаптивного первичного преобразования цветности.

Фиг.23 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.24 – блок-схема, описывающая пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности.

Фиг.25 – блок-схема, описывающая пример последовательности операций процесса получения типа первичного горизонтального преобразования.

Фиг.26 – блок-схема, описывающая пример последовательности операций процесса получения типа первичного вертикального преобразования.

Фиг.27 – схема, иллюстрирующая пример способа получения флага пропуска

преобразования цветности.

Фиг.28 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.29 – блок-схема, описывающая пример последовательности операций процесса получения флага пропуска преобразования.

5 Фиг.30 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.31 – блок-схема, описывающая пример последовательности операций процесса получения флага пропуска преобразования.

Фиг.32 – схема, иллюстрирующая пример синтаксиса residual\_coding.

10 Фиг.33 – блок-схема, описывающая пример последовательности операций процесса получения флага пропуска преобразования.

Фиг.34 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.35 – блок-схема, описывающая пример последовательности операций процесса получения флага пропуска преобразования.

15 Фиг.36 – схема, иллюстрирующая пример синтаксиса и семантики единицы преобразования.

Фиг.37 – схема, иллюстрирующая пример синтаксиса residual\_coding.

Фиг.38 – блок-схема, описывающая пример последовательности операций процесса получения флага пропуска преобразования.

20 Фиг.39 – схема, иллюстрирующая пример способа получения идентификатора вторичного преобразования цветности.

Фиг.40 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

Фиг.41 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

Фиг.42 – блок-схема, описывающая пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности.

25 Фиг.43 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

Фиг.44 – блок-схема, описывающая пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности.

Фиг.45 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

30 Фиг.46 – блок-схема, описывающая пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности.

Фиг.47 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

Фиг.48 – блок-схема, описывающая пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности.

Фиг.49 – схема, иллюстрирующая пример синтаксиса единицы преобразования.

35 Фиг.50 – блок-схема, описывающая пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности.

Фиг.51 – блок-схема, иллюстрирующая пример основной конфигурации устройства кодирования изображения.

40 Фиг.52 – блок-схема, иллюстрирующая пример основной конфигурации единицы преобразования.

Фиг.53 – блок-схема, иллюстрирующая пример основной конфигурации блока получения информации об адаптивном первичном преобразовании цветности.

Фиг.54 – блок-схема, описывающая пример последовательности операций процесса декодирования изображения.

45 Фиг.55 – блок-схема, описывающая пример последовательности операций процесса преобразования.

Фиг.56 – блок-схема, описывающая пример последовательности операций процесса выбора первичного преобразования.

Фиг.57 – блок-схема, описывающая пример последовательности операций процесса кодирования информации о первичном преобразовании.

Фиг.58 – блок-схема, описывающая пример последовательности операций процесса кодирования флага вывода информации об адаптивном первичном преобразовании цветности.

Фиг.59 – блок-схема, описывающая пример последовательности операций процесса кодирования флага пропуска преобразования.

Фиг.60 – блок-схема, описывающая пример последовательности операций процесса кодирования флага пропуска преобразования.

Фиг.61 – блок-схема, описывающая пример последовательности операций процесса кодирования флага пропуска преобразования.

Фиг.62 – блок-схема, описывающая пример последовательности операций процесса кодирования флага пропуска преобразования.

Фиг.63 – блок-схема, описывающая пример последовательности операций процесса кодирования флага пропуска преобразования.

Фиг.64 – блок-схема, описывающая пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности.

Фиг.65 – блок-схема, описывающая пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности.

Фиг.66 – блок-схема, описывающая пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности.

Фиг.67 – блок-схема, описывающая пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности.

Фиг.68 – блок-схема, описывающая пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности.

Фиг.69 – блок-схема, иллюстрирующая пример основной конфигурации компьютера.

Фиг.70 – блок-схема, иллюстрирующая пример схематичной конфигурации телевизионного устройства.

Фиг.71 – блок-схема, иллюстрирующая пример схематичной конфигурации мобильного телефона.

Фиг.72 – блок-схема, иллюстрирующая пример схематичной конфигурации устройства записи/воспроизведения.

Фиг.73 – блок-схема, иллюстрирующая пример схематичной конфигурации устройства формирования изображения.

Фиг.74 – блок-схема, иллюстрирующая один пример схематичной конфигурации видеонабора.

Фиг.75 – блок-схема, иллюстрирующая один пример схематичной конфигурации видеопроцессора.

Фиг.76 – блок-схема, иллюстрирующая другой пример схематичной конфигурации видеопроцессора.

Фиг.77 – блок-схема, иллюстрирующая один пример схематичной конфигурации сетевой системы.

Осуществление изобретения

Ниже будут описаны примерные варианты осуществления (которые будут упоминаться ниже как варианты осуществления) для реализации настоящего раскрытия. Следует отметить, что описание будет предоставлено в следующем порядке.

1. Ортогональное преобразование цветности

2. Первый вариант осуществления (устройство декодирования изображения, apt\_flag

и pt\_idx)

3. Второй вариант осуществления (устройство декодирования изображения и ts\_flag)

4. Третий вариант осуществления (устройство декодирования изображения и st\_idx)

5. Четвертый вариант осуществления (устройство кодирования изображения, apt\_flag

и pt\_idx)

6. Пятый вариант осуществления (устройство кодирования изображения и ts\_flag)

7. Шестой вариант осуществления (устройство кодирования изображения и st\_idx)

8. Прочее

1. Ортогональное преобразование цветности

10 Первичное преобразование

В тестовой модели, описанной в непатентной литературе 1 (совместная экспериментальная модель исследования 4 (JEM 4)), с целью повышения эффективности кодирования изображения с высоким разрешением 4К или подобным, максимальный размер блока дерева кодирования (CTU) увеличен с 128×128 до 256×256. Кроме того, в качестве структуры блочного разбиения, в дополнение к разбиению квадродерева согласно уровню техники, были введены двоичные деревья в горизонтальном/вертикальном направлениях и, в дополнение к квадратным блокам преобразования, были также введены, соответственно, прямоугольные блоки преобразования.

Кроме того, в JEM 4 раскрыто адаптивное первичное преобразование (адаптивные многоядерные преобразования (AMT)) адаптивного выбора первичного преобразования из множества различных ортогональных преобразований для каждого первичного преобразования в горизонтальном направлении PThor (которое также упоминается как первичное горизонтальное преобразование) и первичного преобразования PTver в вертикальном направлении (которое также упоминается как первичное вертикальное преобразование) каждого блока преобразования для яркости.

Более конкретно, в случае, когда флаг apt\_flag адаптивного первичного преобразования (который упоминается также как amt\_flag, cu\_pt\_flag или emt\_flag), указывающий на то, должно или нет адаптивное первичное преобразование для яркости исполняться в единице блока преобразования яркости (который также упоминается как блок преобразования яркости), равен 0 (ложь), дискретное косинусное преобразование (DCT)-II или дискретное синусное преобразование (DST)-VII (однозначно) определяется с использованием информации о режиме в качестве ортогонального преобразования, применяемого к горизонтальному первичному преобразованию и вертикальному первичному преобразованию.

С другой стороны, в случае, когда, например, флаг apt\_flag адаптивного первичного преобразования яркости равен 1 (истина), как в таблице (LUT\_TrSetToTrTypeIdx), показанной на фиг.1, набор TrSet преобразований, включающий в себя ортогональные преобразования, которые служат в качестве кандидатов для первичного преобразования в каждом горизонтальном направлении (направлении x) и вертикальном направлении (направлении y), выбирается из четырех кандидатов (набор Idx преобразований = 0-3). DST-VII, DCT-VIII и т.п., показанные на фиг.1, указывают типы ортогонального преобразования, и каждая из функций, показанных в таблице на фиг.2, используется для них.

Следует отметить, что в таблице, показанной на фиг.2, определены идентификаторы, соответствующие ортогональным преобразованиям соответствующих типов (идентификатор TrTypeIdx типа преобразования). Например, если типом ортогонального преобразования (тип преобразования) является DCT-II, 0 присваивается в качестве значения идентификатора TrTypeIdx типа преобразования.

Выбор (определение) набора TransformSet преобразований варьируется в зависимости от типа предсказания единицы кодирования (CU) (который также упоминается как единица кодирования), к которому принадлежит обрабатываемый блок преобразования. Например, если типом предсказания является межблочное предсказание, выбор

5 выполняется на основании режима (IntraPredMode) межблочного предсказания, как показано в таблице на фиг.3 (LUT\_IntraModeToTrSet). Например, выбор выполняется для установки идентификатора TrSetIdx набора преобразований для назначения набора TrSet преобразований, соответствующего набору (TrSetH или TrSetV) преобразований каждого направления, как показано в следующих формулах (1) и (2).

$$10 \quad \text{TrSetH} = \text{LUT\_IntraModeToTrSet} [\text{IntraPredMode}] [\text{H} (=0)] \dots (1)$$

$$\text{TrSetV} = \text{LUT\_IntraModeToTrSet} [\text{IntraPredMode}] [\text{V} (=1)] \dots (2)$$

В данном случае TrSetH представляет набор преобразований первичного горизонтального преобразования PThor (который также упоминается как набор первичных горизонтальных преобразований), и TrSetV представляет набор

15 преобразований первичного вертикального преобразования PTver (который также упоминается как набор первичных вертикальных преобразований). Кроме того, таблица LUT\_IntraModeToTrSet поиска представляет таблицу соответствия, показанную на фиг.3. Первый массив таблицы LUT\_IntraModeToTrSet[][] поиска имеет режим IntraPredMode межблочного предсказания в качестве аргумента, и второй массив имеет {H=0, V=1} в

20 качестве аргумента.

В случае, когда режим межблочного предсказания имеет номер 18 (IntraPredMode == 18), набор преобразований идентификатора TrSetIdx=2 набора преобразований, указанного в таблице (LUT\_TrSetToTrTypeIdx), показанной на фиг.1, выбран в качестве

набора TrSetH первичных горизонтальных преобразований, и набор преобразований

25 идентификатора TrSetIdx=0 набора преобразований, указанного в таблице на фиг.1, выбран в качестве набора TrSetV первичных вертикальных преобразований.

Если типом предсказания является внутриблочное предсказание, идентификатор TrSetIdx набора преобразований для обозначения набора TrSet преобразований для внутриблочного предсказания (=InterTrSetIdx) устанавливается для наборов

30 преобразований в каждом из направлений (TrSetH и TrSetV), как в следующих формулах (3) и (4). Например, значение InterTrSet равно 3 в случае таблицы, показанной на фиг.1.

$$\text{TrSetH} = \text{InterTrSetIdx} \dots (3)$$

$$\text{TrSetV} = \text{InterTrSetIdx} \dots (4)$$

Кроме того, то, какое ортогональное преобразование в выбранном наборе TrSet преобразований должно быть применено к первичному горизонтальному преобразованию, выбирается с использованием флага pt\_hor\_flag точного определения

35 первичного горизонтального преобразования. Кроме того, то, какое ортогональное преобразование в выбранном наборе TrSet преобразований должно быть применено к первичному вертикальному преобразованию, выбирается с использованием флага

40 pt\_ver\_flag точного определения первичного вертикального преобразования. Например, они получаются из таблицы определений наборов (LUT\_TrSetToTrTypeIdx) преобразований, показанных на фиг.1, с использованием набора TrSet {H, V} первичных {горизонтального, вертикального} преобразований и флага pt\_{hor, ver}\_flag точного определения первичного {горизонтального, вертикального} преобразования в качестве

45 аргументов, как показано в следующих формулах (5) и (6).

$$\text{TrTypeIdxH} = \text{LUT\_TrSetToTrTypeIdx} [\text{TrSetH}] [\text{pt\_hor\_flag}] \dots (5)$$

$$\text{TrTypeIdxV} = \text{LUT\_TrSetToTrTypeIdx} [\text{TrSetV}] [\text{pt\_ver\_flag}] \dots (6)$$

В случае, когда режим межблочного предсказания имеет номер 18 (IntraPredMode ==

18), так как значение идентификатора TrSetIdx набора преобразований из набора TrSetH первичных горизонтальных преобразований равно 2 на основании таблицы, показанный на фиг.3, ортогональное преобразование, которое будет применяться к первичному горизонтальному преобразованию, выбирается (назначается) из набора преобразований, имеющего идентификатор TrSetIdx==2 набора преобразований в таблице LUT\_TrSetToTrTypeIdx определения набора преобразований, показанной на фиг.1. То есть в случае, когда флаг pt\_hor\_flag точного определения первичного горизонтального преобразования равен 0, значение "4" идентификатора TrTypeIdx типа преобразования, указывающего DST-VII, установлен для идентификатора TrTypeIdxH типа горизонтального преобразования, указывающего тип ортогонального преобразования первичного горизонтального преобразования PThor, как показано на фиг.1, и в случае, когда флаг pt\_hor\_flag точного определения первичного горизонтального преобразования равен 1, значение "1" идентификатора TrTypeIdx типа преобразования, указывающего DCT-V, устанавливается для идентификатора TrTypeIdxH типа горизонтального преобразования.

Следует отметить, что идентификатор pt\_idx первичного преобразования получен из флага pt\_hor\_flag точного определения первичного горизонтального преобразования и флага pt\_ver\_flag точного определения первичного вертикального преобразования с использованием следующей формулы (7).

$$pt\_idx = (pt\_ver\_flag < 1) + pt\_hor\_flag \dots (7)$$

То есть старший 1 разряд идентификатора pt\_idx первичного преобразования соответствует флагу точного определения первичного вертикального преобразования, и младший 1 разряд соответствует значению флага точного определения первичного горизонтального преобразования. Кодирование выполняется путем применения арифметического кодирования к битовой строке полученного идентификатора pt\_idx первичного преобразования и выработки битовой строки. Следует отметить, что старший 1 разряд pt\_idx может быть установлен в качестве флага точного определения первичного горизонтального преобразования, и младший 1 разряд может быть установлен в качестве флага точного определения первичного вертикального преобразования.

С другой стороны, адаптивное первичное преобразование не применяется для преобразования блоков цветности (Cb и Cr), и DCT-II всегда выбирается в качестве типа ортогонального преобразования, такого как первичное горизонтальное преобразование и первичное вертикальное преобразование.

Поз.А на фиг.4 представляет собой пример синтаксической таблицы единицы преобразования (TU), и поз.В на фиг.4 представляет собой пример семантики, соответствующей ей. В синтаксисе, показанном на фиг.4, флаг cbf[x0][y0][compID] наличия остаточных данных (coded\_block\_flag), показанный в строке, обозначенной опорным символом SYN11, является флагом, указывающим на то, имеются ли один или более коэффициентов, отличных от нуля, в блоке преобразования цветового сигнала, обозначенного идентификатором compID цветового сигнала, как показано поз.В на фиг.4. В случае, когда флаг равен 1 (истина), он указывает, что в соответствующем блоке преобразования имеются один или более коэффициентов, отличных от нуля, и в случае, когда флаг равен 0 (ложь), он указывает, что в блоке преобразования отсутствует коэффициент, отличный от нуля. Следует отметить, что имеется также случай, когда флаг cbf[x0][y0][COMPONENT\_Y] наличия остаточных данных для яркости упоминается ради удобства как cbf\_luma (cbf\_luma = cbf[x0][y0][COMPONENT\_Y]).

Кроме того, флаг apt\_enabled\_flag разрешения адаптивного первичного

преобразования, показанный поз.А на фиг.4, представляет собой информацию, касающуюся разрешения для адаптивного первичного преобразования, как показано поз.В на фиг.4. В случае, когда значение флага равно 1 (истина), это указывает на то, что информация относительно адаптивного первичного преобразования присутствует в кодированных данных. Кроме того, в случае, когда значение флага равно 0 (ложь), это указывает на то, что в кодированных данных отсутствует информация, касающаяся адаптивного первичного преобразования.

Кроме того, флаг `apt_flag` адаптивного первичного преобразования блока преобразования яркости, показанный в строке, обозначенной опорным символом, SYN12 в синтаксисе, показанном на фиг.4, является флагом, указывающим на то, следует ли применить адаптивное первичное преобразование к блоку преобразования яркости, как показано поз.В на фиг.4. В случае, когда значение флага равно 1 (истина), применяется адаптивное первичное преобразование, и в случае, когда значение флага равно 0 (ложь), первичное преобразование приложения не применяется. Как показано в синтаксической таблице, показанной поз.А на фиг.4, в случае, когда флаг наличия остаточных данных `cbf_lum` яркости равен "1" (то есть имеет значение "истина"), и флаг `apt_enabled_flag` разрешения адаптивного первичного преобразования равен "1" (то есть имеет значение "истина"), флаг `apt_flag` адаптивного первичного преобразования яркости кодируется (декодируется). В случае, когда `apt_flag` отсутствует в кодированных данных, значение `apt_flag` интерпретируется как 0.

Кроме того, в синтаксисе, показанном поз.А на фиг.4, остаточные данные `residual_coding ()` блока преобразования цветового сигнала, обозначенного идентификатором `compID` цветового сигнала, показанным в строке, обозначенной опорным символом SYN13, кодируется (декодируется) в случае, когда флаг наличия остаточных данных соответствующего цветового сигнала равен 1 (истина).

На фиг.5 показан пример синтаксиса `residual_coding`. В синтаксической таблице, показанной на фиг.5, идентификатор `pt_idx` первичного преобразования для яркости, обозначенной опорным символом SYN22, кодируется (декодируется) в случае, когда флаг `apt_flag` адаптивного первичного преобразования яркости равен 1 (истина), флаг `transquant_bypass_flag` обхода квантования преобразования равен 0 (ложь), флаг `ts_flag` пропуска преобразования равен 0 (ложь), и идентификатор `compID` цветового сигнала указывает яркость (`compID == COMPONENT_Y`), и в случае, когда общее количество ненулевых коэффициентов, присутствующих в блоке `numSig` преобразования, больше или равно заданному пороговому значению `ptNumSigTH`, и длинная сторона блока преобразования ( $\max(\log_2 \text{TBWSize}, \log_2 \text{TBHSize})$ ) равна или меньше заданного порогового значения `maxPTSize`, как показано в синтаксической таблице. В случае, когда идентификатор `pt_idx` первичного преобразования для яркости отсутствует в кодированных данных, значение `pt_idx` интерпретируется как 0.

В непатентной литературе 1, хотя адаптивное первичное преобразование может быть применено только к яркости (Y), оно не применяется к цветности (Cb и Cr). Таким образом, существует проблема, связанная со снижением эффективности кодирования первичного преобразования для цветности в большей степени, чем для яркости.

С другой стороны, можно явно кодировать, например, флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования для каждого блока преобразования цветности (Cb или Cr) аналогично яркости (Y). Однако в случае этого способа, так как флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования для цветности кодируются, существует проблема, связанная с увеличением объема кода и снижением эффективности

кодирования.

Следует также отметить, что в этом случае, на стороне кодирования необходимо определить режим для выбора/не выбора адаптивного первичного преобразования и определения идентификатора первичного преобразования для каждого из цветных сигналов (Y, Cb и Cr), и, таким образом, существует проблема, связанная с увеличением объема обработки. Кроме того, на стороне декодирования флаг apt\_flag адаптивного первичного преобразования и идентификатор pt\_idx первичного преобразования должны быть также декодированы для каждого из цветных сигналов (Y, Cb и Cr), и, таким образом, существует проблема, связанная с увеличением объема обработки.

#### Использование параметра яркости

Таким образом, информация относительно (обратного) ортогонального преобразования для цветности получается на основании информации относительно (обратного) ортогонального преобразования для яркости. То есть, используя информацию относительно (обратного) ортогонального преобразования для цветности, полученную на основании (обратного) ортогонального преобразования для яркости, выполняется (обратное) ортогональное преобразование для цветности. Например, при использовании информации относительно (обратного) ортогонального преобразования для цветности, полученную на основании (обратного) ортогонального преобразования для яркости, в устройстве обработки изображений предусмотрен блок (обратного) ортогонального преобразования, который выполняет (обратное) ортогональное преобразование для цветности.

Таким образом, можно исключить кодирование и декодирование информации относительно (обратного) ортогонального преобразования для цветности, и, таким образом, можно сдерживать увеличение объема кода, и можно предотвратить снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование. Следует отметить, что в данном описании ортогональное преобразование и обратное ортогональное преобразование являются обратными процессами по отношению друг к другу, и предполагается, что данные, которые не были ортогонально преобразованы, могут быть восстановлены, например, путем выполнения обратного ортогонального преобразования над ортогонально преобразованными данными. Кроме того, хотя информацией об ортогональном преобразовании является информация, которая будет использоваться при ортогональном преобразовании, существует также случай, когда информация может использоваться при обратном ортогональном преобразовании. Кроме того, хотя информацией относительно обратного ортогонального преобразования является информация, которая будет использоваться при обратном ортогональном преобразовании, также существует случай, когда информация может быть использована при ортогональном преобразовании. То есть в данном описании существует случай, в котором информация относительно ортогонального преобразования и информация относительно обратного ортогонального преобразования относятся к одной и той же информации (информация относительно обоих преобразований может включать в себя одну и ту же информацию). То же самое относится и к соотношениям между первичным преобразованием и обратным первичным преобразованием, и вторичным преобразованием и обратным вторичным преобразованием.

Содержание информации относительно (обратного) ортогонального преобразования является произвольным. Например, как показано в таблице на фиг.6, информация может включать в себя флаг apt\_flag адаптивного первичного преобразования, указывающий на то, применяется ли адаптивное (обратное) первичное преобразование,

которое будет использоваться в качестве (обратного) первичного преобразования путем адаптивного выбора любого из множества различных обратных ортогональных преобразований, к блоку преобразования, подлежащему обработке. Кроме того, информация может также включать в себя идентификатор `pt_idx` первичного преобразования, указывающий на то, какое (обратное) первичное преобразование должно быть применено к (обратным) первичным преобразованиям в вертикальном направлении и горизонтальном направлении.

Например, получая каждый из флага `apt_flag` адаптивного первичного преобразования [Cb] цветности (Cb) и флага `apt_flag` адаптивного первичного преобразования [Cr] цветности (Cr) на основании флага `apt_flag` [Y] адаптивного первичного преобразования яркости, можно сдерживать снижение эффективности кодирования цветности (Cb/Cr). Кроме того, можно также сдерживать увеличение издержек для объема кода. Кроме того, например, получая каждый из идентификатора первичного преобразования цветности (Cb) `pt_idx`[Cb] и идентификатора первичного преобразования цветности (Cr) `pt_idx`[Cr] на основании идентификатора первичного преобразования яркости `pt_idx` [Y], можно сдерживать снижение эффективности кодирования цветности (Cb/Cr). Кроме того, можно также сдерживать увеличение издержек для объема кода. Следует отметить, что идентификатор первичного преобразования цветности может быть общим для (совместно использоваться) Cb и Cr.

Дальнейшее описание будет предоставлено со ссылкой на таблицу, показанную на фиг.7. В уровне техники адаптивное (обратное) первичное преобразование не используется для цветности, и флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования для цветности всегда опускаются, как указано в строке под номером #0. Поскольку в таких случаях адаптивное (обратное) первичное преобразование не используется, существует проблема, связанная со снижением эффективности кодирования, как описано выше.

С другой стороны, адаптивное (обратное) первичное преобразование может быть установлено таким образом, чтобы оно применялось к цветности, и флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования для цветности могут быть дополнительно выведены из флага `apt_flag` адаптивного первичного преобразования и идентификатора `pt_idx` первичного преобразования для яркости в любой момент времени, как указано в строке под номером #1. Например, значение флага `apt_flag` [Cb/Cr] адаптивного первичного преобразования цветности может быть установлено равным значению флага `apt_flag` [Y] адаптивного первичного преобразования яркости. Кроме того, например, в случае, когда флаг `apt_flag` [Cb/Cr] адаптивного первичного преобразования цветности имеет значение "истина", значение идентификатора `pt_idx` первичного преобразования [Cb/Cr] цветности может быть установлено равным значению идентификатора `pt_idx`[Y] первичного преобразования яркости, и в случае, когда флаг `apt_flag` [Cb/Cr] адаптивного первичного преобразования цветности имеет значение "ложь", значение идентификатора `pt_idx` первичного преобразования [Cb/Cr] цветности может быть установлено равным заданному значению. Таким образом, адаптивное (обратное) первичное преобразование может быть применено не только к цветности, а передачу (кодирование и декодирование) флага `apt_flag` адаптивного первичного преобразования и идентификатора `pt_idx` первичного преобразования цветности можно также опустить, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

Кроме того, в случае, когда адаптивное (обратное) первичное преобразование

устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание ( $CuPredMode == MODE\_INTER$ ), значение флага  $apt\_flag [Cb/Cr]$

5 адаптивного первичного преобразования цветности может быть установлено равным значению флага  $apt\_flag [Y]$  адаптивного первичного преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, флаг  $apt\_flag [Cb/Cr]$  адаптивного первичного преобразования цветности может быть установлен равным значению 0 (ложь), как указано в строке под номером  
10 #2. Таким образом, к цветности может применяться не только адаптивное (обратное) первичное преобразование, а также может использоваться флаг адаптивного первичного преобразования яркости, но только в случае режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования.

15 Кроме того, в случае, когда адаптивное (обратное) первичное преобразование устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и  
20 цветности совпадают друг с другом, значение флага  $apt\_flag [Cb/Cr]$  адаптивного первичного преобразования цветности может быть установлено равным значению флага  $apt\_flag [Y]$  адаптивного первичного преобразования яркости, как указано в строке под номером #3, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности  
25 не совпадают друг с другом, флаг  $apt\_flag [Cb/Cr]$  адаптивного первичного преобразования цветности может быть установлен равным значению 0 (ложь). Таким образом, к цветности может применяться не только адаптивное (обратное) первичное преобразование, а также может использоваться флаг адаптивного первичного преобразования яркости, но только в случае режима предсказания, в котором тенденции  
30 остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда адаптивное (обратное) первичное преобразование устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, типом предсказания блока кодирования, к которому  
35 принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, как указано в строке под номером #4, значение флага  $apt\_flag [Cb/Cr]$  адаптивного первичного преобразования цветности может быть установлено равным значению флага  $apt\_flag [Y]$  адаптивного первичного  
40 преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование (IntraBC) (которое также упоминается как "компенсация движения на экране"), значение флага  $apt\_flag [Cb/Cr]$  адаптивного первичного преобразования цветности может быть установлено равным значению 0 (ложь). Таким образом, к  
45 цветности может применяться не только адаптивное (обратное) первичное преобразование, а также может использоваться флаг адаптивного первичного преобразования яркости, но только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать

снижение эффективности кодирования.

Кроме того, в случае, когда адаптивное (обратное) первичное преобразование устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, флаг `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности, указывающий на то, выведен ли флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности на основании флага `apt_flag [Y]` адаптивного первичного преобразования яркости, равен 1 (истина), как указано в строке под номером #5, флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности может быть установлен равным значению флага `apt_flag [Y]` адаптивного первичного преобразования яркости, и в случае, когда флаг `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности равен 0 (ложь), значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности может быть установлено равным значению 0 (ложь). Таким образом, можно однозначно управлять тем, следует ли использовать значение флага адаптивного первичного преобразования яркости, используя флаг `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности. Поэтому можно сделать так, чтобы значение флага адаптивного первичного преобразования яркости использовалось только в случае, когда можно достаточно легко получить достаточно большие эффекты, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда адаптивное (обратное) первичное преобразование устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, размер короткой стороны блока преобразования, подлежащего обработке, больше или равен значению заданного порогового значения, как указано в строке под номером #6, значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности может быть установлено равным значению флага `apt_flag [Y]` адаптивного первичного преобразования яркости, и в случае, когда размер короткой стороны блока преобразования меньше порогового значения, значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности может быть установлено равным значению 0 (ложь). Таким образом, так как можно не применять адаптивное (обратное) первичное преобразование к блоку преобразования, который не позволяет получить достаточно большие эффекты, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования, и можно избежать ненужного увеличения масштаба схемы в результате применения адаптивного (обратного) первичного преобразования к небольшому блоку преобразования до такой степени, чтобы не получались достаточно большие эффекты.

Кроме того, адаптивное (обратное) первичное преобразование устанавливается таким образом, чтобы быть применимым к цветности в той же степени, и в дополнение к этому, например, размер ширины картинки блока преобразования с цветностью, подлежащей обработке, равен или меньше заданного порогового значения, как указано в строке под номером #7, тип преобразования (обратного) первичного горизонтального преобразования может быть установлен в заданный тип преобразования, и в случае, когда размер ширины картинки блока преобразования больше порогового значения, тип преобразования (обратного) первичного горизонтального преобразования может быть установлен на основании набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования. Аналогичным образом, в случае, когда размер высоты картинки блока преобразования с цветностью, подлежащей обработке, равен или меньше заданного порогового значения, тип преобразования

(обратного) первичного вертикального преобразования может быть установлен в заданный тип преобразования, и в случае, когда размер высоты картинки блока преобразования больше порогового значения, тип преобразования (обратного) первичного вертикального преобразования может быть установлен на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования. Таким образом, можно не применять адаптивное (обратное) первичное преобразование к блоку преобразования, который является узким до такой степени, что не получаются достаточно большие эффекты ((обратное) первичное преобразование в направлении ширины), и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования, и можно избежать ненужного увеличения масштаба схемы в результате применения адаптивного (обратного) первичного преобразования к узкому блоку преобразования до такой степени, чтобы не получались достаточно большие эффекты.

Следует отметить, что можно также произвольным образом объединить каждый из случаев, описанных выше. Например, случай в строке под номером #2 можно объединить со случаем в строке под номером #7, аналогично случаю в строке под номером #8. Таким образом, можно продемонстрировать эффекты, полученные в каждом из случаев. Кроме того, каждый из вышеописанных случаев можно также объединить с другим случаем, который не описан выше. Например, в случае, когда типом предсказания обрабатываемого блока кодирования является межблочное предсказание в случае строки под номером #2, может быть передан (просигнализирован) флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности. То есть в этом случае выполняется кодирование и декодирование флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности. Кроме того, в случае, когда значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности равно 1 (истина), дополнительно передается (сигнализируется) идентификатор `pt_idx` адаптивного первичного преобразования `[Cb/Cr]` цветности. То есть в этом случае выполняется кодирование и декодирование идентификатора `pt_idx [Cb/Cr]` адаптивного первичного преобразования цветности.

#### Вид CU, PU и TU

Ниже будут описаны виды единицы кодирования (CU), единицы предсказания (PU) и TU, которые являются частичными областями (блоками обработки) движущегося изображения в битовом потоке (кодированных данных). CU, PU и единицы преобразования (TU), показанные на фиг.8, представляют собой CU, PU и TU квадродерева плюс двоичного дерева (QTTBT), описанного в JVET-C0024, "EE2.1: Интеграция структуры квадродерева плюс двоичного дерева с инструментами JEM". В частности, при разбиении на блоки для CU, один блок разделен не только на четыре ( $=2 \times 2$ ) подблока, но также и на два ( $=1 \times 2$  или  $2 \times 1$ ) подблока. Другими словами, путем рекурсивного повторяющегося разбиения одного блока на четыре или два подблока при разбиении на блоки для CU, последовательно формируется структура дерева в форме квадродерева (Quad-Tree) или в форме двоичного дерева (Binary-Tree) в горизонтальном направлении или в вертикальном направлении.

В результате, существует вероятность того, что CU имеет как квадратную, так и прямоугольную форму. Например, в случае, когда размер наибольшей единицы кодирования (LCU) равен  $128 \times 128$ , существует вероятность того, что размер CU (размер в горизонтальном направлении  $w \times$  размер в вертикальном направлении  $h$ ) равен размеру квадрата, такого как  $128 \times 128$ ,  $64 \times 64$ ,  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$  или  $4 \times 4$ , а также размеру прямоугольника, такого как  $128 \times 64$ ,  $128 \times 32$ ,  $128 \times 16$ ,  $128 \times 8$ ,  $128 \times 4$ ,  $64 \times 128$ ,  $32 \times 128$ ,

16×128, 8×128, 4×128, 64×32, 64×16, 64×8, 64×4, 32×64, 16×64, 8×64, 4×64, 32×16, 32×8, 32×4, 16×32, 8×32, 4×32, 16×8, 16×4, 8×16, 4×16, 8×4 или 4×8, как показано на фиг.8. Следует отметить, что PU и TU имеют одинаковую вероятность, как и CU.

Кроме того, TU включает в себя блок преобразования яркости (Y) и блок преобразования цветности (Cb/Cr). В случае, когда цветовой формат составляет 4:2:0 (например, YUV420), соотношение размера картинки цветности к размеру картинки яркости по отношению к высоте картинки и ширине картинки составляет 1/2. Таким образом, если размер блока преобразования яркости равен 8×4, размер соответствующего блока преобразования цветности составляет 4×2. Кроме того, в случае, когда цветовой формат составляет 4:2:2 (например, YUV422), соотношение размера картинки цветности к размеру картинки яркости составляет 1/2 по отношению к высоте картинки и 1 по отношению к ширине картинки. Таким образом, если размер блока преобразования яркости равен 8×4, размер блока преобразования для соответствующей цветности равен 8×2. Кроме того, в случае, когда цветовой формат составляет 4:4:4 (например, YUV444), соотношение размера картинки цветности к размеру картинки яркости равно 1 по отношению к высоте картинки и 1 по отношению к ширине картинки. Таким образом, если размер блока преобразования яркости равен 8×4, размер блока преобразования соответствующей цветности равен 8×4.

Следует отметить, что по отношению к I-слайсам яркость (Y) и цветность (Cb/Cr) могут кодироваться как разные CU. Поскольку в этом случае яркость и цветность могут иметь разные структуры разбиения CU, проявляется эффект повышения эффективности кодирования I-слайсов. Хотя ниже будет представлено описание при предположении, что информация о яркости и цветности включена в один и тот же CU для удобства, изобретение не ограничивается этим.

## 2. Первый вариант осуществления

### Устройство декодирования изображения

На фиг.9 показана блок-схема, иллюстрирующая пример конфигурации устройства декодирования изображения, которое представляет собой аспект устройства обработки изображений, к которому была применена настоящая технология. Устройство декодирования изображения, показанное на фиг.9, является устройством, которое декодирует кодированные данные, полученные в результате кодирования предсказуемого остатка изображения и предсказуемого изображения, как в AVC или HEVC. Устройство декодирования изображения реализует, например, технологию, предложенную HEVC или группой совместного исследования видео (JVET).

На фиг.9 устройство декодирования изображения имеет блок 111 декодирования, блок 112 обратного квантования, блок 113 обратного преобразования, блок 114 арифметических операций, память 115 кадров и блок 116 предсказания. Следует отметить, что блок 116 предсказания имеет блок межблочного предсказания и блок внутриблочного предсказания, которые не показаны. Устройство декодирования изображения является устройством для выработки движущегося изображения #2 посредством декодирования кодированных данных #1 (битового потока).

Блок 111 декодирования принимает ввод кодированных данных #1 и выполняет декодирование с переменной длиной кода в отношении значения синтаксиса каждого синтаксического элемента из битовой строки кодированных данных #1 в соответствии с решением синтаксической таблицы. Кроме того, каждый синтаксический элемент включает в себя информацию, такую как информация Hinfo заголовка, информацию PInfo о режиме предсказания, информацию Tinfo о преобразовании и остаточную информацию Rinfo.

Информация Hinfo заголовка, такая как VPS/SPS/PPS/заголовок SH слайса включает в себя информацию, которая определяет размеры изображения (ширину PicWidth картинки и высоту PicHeight картинки), битовые глубины (яркость bitDepthY и цветность bitDepthC), максимальное значение размера MaxCUsSize CU/минимальное значение размера MinCUsSize CU, максимальную глубину MaxQTDdepth разбиения квадродерева (которая также упоминается как разбиение квадродерева)/минимальную глубину MinQTDdepth разбиения квадродерева, максимальную глубину разбиения MaxBTDepth двоичного дерева (которая также упоминается как разбиение двоичного дерева)/минимальную глубину MinBTDepth двоичного дерева, максимальное значение MaxTSSize блока пропуска преобразования (которое также упоминается как максимальный размер блока пропуска преобразования), флаг включения/выключения каждого инструмента кодирования (который также упоминается как флаг разрешения) и т.п.

Далее будет описан флаг включения/выключения инструмента кодирования, включенный в информацию Hinfo заголовка, например, флаг включения/выключения для преобразования и процессов квантования, которые будут представлены ниже. Следует отметить, что флаг включения/выключения инструмента кодирования можно также интерпретировать как флаг, указывающий на то, присутствует ли синтаксис для инструмента кодирования в кодированных данных. Кроме того, в случае, когда значение флага включения/выключения равно 1 (истина), это указывает на то, что инструмент кодирования доступен, и в случае, когда значение флага включения/выключения равно 0 (ложь), это указывает на то, что инструмент кодирования недоступен. Следует отметить, что интерпретация значения флага может быть изменена.

Флаг apt\_enabled\_flag разрешения адаптивного первичного преобразования (который также упоминается как adaptive\_primary\_transform\_enabled\_flag, adaptive\_pt\_enabled\_flag, amt\_enabled\_flag) представляет собой флаг, указывающий на то, доступен ли инструмент кодирования, который может выбрать адаптивное первичное преобразование (которое также упоминается как адаптивное первичное преобразование) в качестве одного из процесса преобразования и обратного процесса по отношению к нему.

Флаг st\_enabled\_flag разрешения вторичного преобразования является флагом, указывающим на то, доступен или нет инструмент кодирования, который выполняет вторичное преобразования/обратное вторичное преобразование в качестве одного из процесса преобразования и обратного процесса по отношению к нему.

Флаг transquant\_bypass\_enabled\_flag разрешения обхода квантования преобразования является флагом, указывающим на то, доступен или нет инструмент кодирования, который пропускает преобразование, и квантование/обратное преобразование и обратное квантование в качестве одного из преобразования/квантования и процессов, которые являются обратными по отношению к ним.

Флаг ts\_enabled\_flag разрешения пропуска преобразования является флагом, указывающим на то, доступен или нет пропуск двумерного преобразования или пропуск одномерного преобразования в качестве одного из процесса преобразования и обратного процесса по отношению к нему. Пропуск двумерного преобразования является инструментом кодирования, который пропускает ортогональное преобразование и обратный процесс (обратное ортогональное преобразование) по отношению к нему, в том числе первичное преобразование и вторичное преобразование. Кроме того, пропуск одномерного преобразования является инструментом кодирования, который пропускает первичное преобразование в горизонтальном направлении или вертикальном направлении и обратное преобразование (обратное первичное преобразование), соответствующее ему, среди первичных преобразований, и вторичного

преобразования и обратного вторичного преобразования по отношению к нему.

Информация Pinfo о режиме предсказания дополнительно включает в себя размер PU (размер блока предсказания), который будет обрабатывать PUSize, информацию IPinfo о режиме межблочного предсказания (например, prev\_intra\_luma\_pred\_flag, mpm\_idx, rem\_intra\_pred\_mode в синтаксисе единицы кодирования 7.3.8.5 JCTVC-W1005, режим IntraPredModeY межблочного предсказания яркости и режим IntraPredModeC межблочного предсказания цветности, полученный из синтаксиса и т.д.), информацию MVinfo предсказания движения (например, merge\_idx, merge\_flag, inter\_pred\_idc, ref\_idx\_LX, mvr\_LX\_flag, X = {0,1} и mvd со ссылкой на синтаксис единицы предсказания 7.3.8.6 JCTVC-W1005) и т.п.

Кроме того, информация Tinfo о преобразовании может включать в себя синтаксис, например, размер TBWidth ширины картинки и размер TBHeight высоты картинки блока преобразования, подлежащего обработке, флаг transquant\_bypass\_flag обхода квантования преобразования, указывающий на то, следует или нет пропускать процессы (обратного) преобразования и (обратного) квантования, флаг ts\_flag пропуска преобразования, указывающий на то, следует или нет применять пропуск двумерного преобразования, флаг apt\_flag адаптивного первичного преобразования, указывающий на то, должно ли применяться адаптивное первичное преобразование к целевому блоку преобразования, идентификатор pt\_idx первичного преобразования, указывающий на то, какое (обратное) первичное преобразование должно применяться между (обратными) первичными преобразованиями в вертикальном направлении и горизонтальном направлении, идентификатор st\_idx вторичного преобразования (который также упоминается как dnsst\_idx, nsst\_idx или rot\_idx), указывающий на то, какое должно применяться (обратное) вторичное преобразование, идентификатор scanIdx сканирования, параметр qr квантования, матрица scaling\_matrix квантования и т.п. Следует отметить, что вместо размера TBWidth ширины картинки и размера TBHeight высоты картинки блока преобразования, подлежащего обработке, можно использовать логарифмическое значение каждого из TBWidth и TBHeight, имеющее 2 в качестве основания log2TBWidth (которое также упоминается как log2TBWSize или log2TBW) или log2TBHeight (которое также упоминается как log2TBHSize или log2TBH).

Остаточная информация Rinfo может включать в себя, например, флаг (cbf (coded\_block\_flag)) наличия/отсутствия остаточных данных, координату (last\_sig\_coeff\_x\_pos) последнего ненулевого коэффициента X, координату (last\_sig\_coeff\_y\_pos) последнего ненулевого коэффициента Y, флаг (coded\_sub\_block\_flag) наличия/отсутствия ненулевого коэффициента подблока, флаг (sig\_coeff\_flag) наличия/отсутствия ненулевого коэффициента, флаг, указывающий на то, больше ли 1 уровень ненулевого коэффициента (который также упоминается как флаг GR1), флаг, указывающий на то, больше ли 2 уровень ненулевого коэффициента флаг (gr2\_flag) (который также упоминается как флаг GR2), флаг знака (sign\_flag), указывающий положительный/отрицательный знак ненулевого коэффициента (который также упоминается как символ знака), оставшийся уровень ненулевого коэффициента (coeff\_abs\_level\_remaining) (который также упоминается как оставшийся уровень ненулевого коэффициента) и т.п.

Блок 111 декодирования получает уровень квантованного по уровню коэффициента преобразования каждого положения коэффициента в каждом блоке преобразования со ссылкой на остаточную информацию Rinfo. Блок 111 декодирования подает информацию Pinfo о режиме предсказания, уровень квантованного по уровню коэффициента преобразования и информацию Tinfo о преобразовании, полученную из

декодирования, в каждый блок. Например, блок 111 декодирования подает информацию PInfo о режиме предсказания в блок 113 обратного преобразования и блок 116 предсказания, уровень квантованного по уровню коэффициента преобразования в блок 112 обратного квантования и информацию Tinfo о преобразовании в блок 113 обратного преобразования и блок 112 обратного квантования.

Блок 112 обратного квантования принимает ввод информации Tinfo о преобразовании и уровень квантованного по уровню коэффициента преобразования, выполняет масштабирование (обратное квантование) над значением уровня квантованного по уровню коэффициента преобразования на основании информации Tinfo о преобразовании и выводит коэффициент преобразования, который был подвергнут обратному квантованию Coeff\_IQ, в блок 113 обратного преобразования.

Блок 113 обратного преобразования принимает ввод коэффициента Coeff\_IQ преобразования, информации Tinfo о преобразовании и информации PInfo о режиме предсказания, применяет обратное преобразование к коэффициенту Coeff\_IQ преобразования на основании информации Tinfo о преобразовании, информации PInfo о режиме предсказания и т.п., получает предсказанный остаток D' и выводит предсказанный остаток в блок 114 арифметических операций. Следует отметить, что обратное преобразование является обратным процессом по отношению к процессу преобразования, выполняемому на стороне кодирования и т.п. Например, обратное ортогональное преобразование, такое как процесс, обратный по отношению к ортогональному преобразованию, выполняемому на стороне кодирования или т.п., включен в обратное преобразование. Кроме того, в случае, когда первичное преобразование или вторичное преобразование выполняется в качестве ортогонального преобразования на стороне кодирования, например, обратное первичное преобразование, которое выполняет обратный процесс по отношению к первичному преобразованию, обратное вторичное преобразование, которое выполняет обратный процесс по отношению к вторичному преобразованию и т.п., включено в обратное преобразование. Детали блока 113 обратного преобразования будут описаны ниже.

Блок 114 арифметических операций принимает ввод предсказанного остатка D', и предсказанное изображение P, поданное из блока 116 предсказания, добавляет предсказанный остаток D' в предсказанное изображение P (предсказанный сигнал), соответствующее предсказанному остатку D', как показано в следующей формуле (8), получает локальное декодированное изображение Res и подает локальное декодированное изображение в память 115 кадров или выводит за пределы устройства 100 декодирования изображения.

$$Res = D' + P \dots (8)$$

Память 115 кадров принимает ввод локального декодированного изображения Res, поданного из блока 114 арифметических операций, восстанавливает декодированное изображение в каждом блоке картинки и затем сохраняет декодированное изображение в памяти 115 кадров. Память 115 кадров считывает декодированное изображение, точно определенное в информации PInfo о режиме предсказания блока 116 предсказания в качестве опорного изображения из буфера, и подает декодированное изображение в блок 116 предсказания. Кроме того, память 115 кадров может хранить информацию Hinfo о заголовке, информацию PInfo о режиме предсказания, информацию Tinfo о преобразовании и т.п., которые относятся к выработке декодированного изображения в буфере, находящемся внутри памяти кадров.

Блок 116 предсказания принимает ввод информации PInfo о режиме предсказания, вырабатывает предсказанное изображение P, используя в качестве опорного

изображения декодированное изображение, хранящееся в памяти 115 кадров, которое точно определено в информации PInfo о режиме предсказания с использованием способа предсказания, точно определенного в информации PInfo о режиме предсказания, и выводит предсказанное изображение в блок 114 арифметических операций.

#### 5 Блок обратного преобразования

На фиг.10 показана блок-схема, иллюстрирующая пример основной конфигурации блока 113 обратного преобразования устройства 100 декодирования изображения, показанного на фиг.9. Как показано на фиг.9, блок 113 обратного преобразования включает в себя переключатель 121, блок 122 обратного вторичного преобразования  
10 и блок 123 обратного первичного преобразования.

Переключатель 121 принимает коэффициент Coeff\_IQ преобразования и флаг ts\_flag [compID] пропуска преобразования. В случае, когда значение флага ts\_flag[compID] пропуска преобразования представляет собой NO\_TS (=0), то есть в случае, когда пропуск преобразования не применяется, переключатель 121 выводит коэффициент  
15 Coeff\_IQ преобразования в блок 122 обратного вторичного преобразования. Кроме того, в случае, когда значение флага ts\_flag[compID] пропуска преобразования представляет собой 2D\_TS (=1), то есть в случае, когда флаг указывает на применение пропуска двумерного преобразования, переключатель 121 пропускает блок 122 обратного вторичного преобразования и блок 123 обратного первичного  
20 преобразования и выводит коэффициент Coeff\_IQ преобразования в виде предсказанного остатка D'.

Блок 122 обратного вторичного преобразования выполняет процесс, относящийся к обратному вторичному преобразованию, который является обратным процессом по отношению к вторичному преобразованию, который представляет собой заданный  
25 процесс преобразования, например, ортогональное преобразование или т.п. Например, блок 122 обратного вторичного преобразования принимает ввод идентификатора st\_idx вторичного преобразования, идентификатора scanIdx сканирования, указывающего способ сканирования коэффициента преобразования, и коэффициент Coeff\_IQ преобразования получает коэффициент Coeff\_IS преобразования (который также  
30 упоминается как коэффициент Coeff\_P первичного преобразования), который был подвергнут обратному вторичному преобразованию, и подает коэффициент преобразования в блок 123 обратного первичного преобразования. Более конкретно, в случае, когда идентификатор st\_idx вторичного преобразования указывает на применение обратного вторичного преобразования (st\_idx > 0), блок 122 обратного  
35 вторичного преобразования выполняет процесс обратного вторичного преобразования, соответствующего идентификатору st\_idx вторичного преобразования, над коэффициентом Coeff\_IQ преобразования и выводит коэффициент Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию. В случае, когда идентификатор st\_idx вторичного преобразования указывает на  
40 неприменение обратного вторичного преобразования (st\_idx == 0), блок 122 обратного вторичного преобразования пропускает обратное вторичное преобразование и выводит коэффициент Coeff\_IQ преобразования в виде коэффициента Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию.

Блок 123 обратного первичного преобразования выполняет процесс, относящийся  
45 к обратному первичному преобразованию, который представляет собой обратный процесс по отношению к первичному преобразованию, который представляет собой заданный процесс преобразования, например, ортогональное преобразование. Например, блок 123 обратного первичного преобразования принимает вводы

идентификатора compID цветового сигнала, флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования яркости, идентификатора pt\_idx[COMPONENT\_Y] первичного преобразования яркости, информации PInfo о режиме предсказания, размеров блока преобразования (логарифмического значения ширины log2TBWSize картинки и логарифмического значения высоты log2TBHSize картинки) и коэффициента Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию. Блок 123 обратного первичного преобразования выбирает тип TrTypeIdxH обратного первичного горизонтального преобразования и тип TrTypeIdxV обратного первичного вертикального преобразования цветового сигнала, точно

определенного идентификатором compID цветового сигнала по отношению к информации PInfo о режиме предсказания, идентификатор compID цветового сигнала, флаг apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования яркости, идентификатор pt\_idx[COMPONENT\_Y] первичного преобразования яркости и идентификатор pt\_idx[COMPONENT\_Y] первичного преобразования яркости, выполняет обратное первичное вертикальное преобразование, определенное типом TrTypeIdxV обратного первичного вертикального преобразования и высотой log2TBHSize картинки блока преобразования, и обратное первичное горизонтальное преобразование, определенное типом TrTypeIdxH обратного первичного горизонтального преобразования и шириной log2TBWSize картинки блока преобразования, над коэффициентом Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию, и затем получает и выводит предсказанный остаток D', который был подвергнут обратному первичному преобразованию.

Блок обратного первичного преобразования

Как показано на фиг.11, блок 123 обратного первичного преобразования имеет блок 131 получения информации об адаптивном первичном преобразовании цветности, блок 132 выбора обратного первичного преобразования, блок 133 обратного первичного вертикального преобразования и блок 134 обратного первичного горизонтального преобразования.

Блок 131 получения информации об адаптивном первичном преобразовании цветности получает информацию (apt\_flag[compID] и pt\_idx[compID]) (compID=COMPONENT\_Cb или COMPONENT\_Cr)) об адаптивном первичном преобразовании цветности, используя информацию (apt\_flag[COMPONENT\_Y], pt\_idx[COMPONENT\_Y]) об адаптивном первичном преобразовании блока преобразования яркости, соответствующей блоку преобразования цветности в случае, когда идентификатор compID цветового сигнала указывает цветность (compID == COMPONENT\_Cb или COMPONENT\_Cr) и подает результат в блок 132 выбора обратного первичного преобразования.

Блок получения информации об адаптивном первичном преобразовании цветности

На фиг.11 показана функциональная блок-схема, иллюстрирующая пример основной конфигурации функций блока 131 получения информации об адаптивном первичном преобразовании цветности. Как показано на фиг.11, блок 131 получения информации об адаптивном первичном преобразовании цветности имеет блок 151 получения apt\_flag и блок 152 получения pt\_idx. Блок 151 получения apt\_flag выполняет процесс, относящийся к получению флага apt\_flag адаптивного первичного преобразования цветности.

Например, блок 151 получения apt\_flag получает флаг apt\_flag[compID] адаптивного первичного преобразования (compID=COMPONENT\_Cb или COMPONENT\_Cr), касающийся блока преобразования цветности, подлежащего обработке, как показано в следующей формуле (9), на основании флага apt\_flag[COMPONENT\_Y] адаптивного

первичного преобразования, относящегося к блоку преобразования яркости, соответствующей блоку преобразования цветности.

$\text{apt\_flag}[\text{compID}] = \text{apt\_flag}[\text{COMPONENT\_Y}]$

(здесь  $\text{compID} = \text{COMPONENT\_Cb}$  или  $\text{COMPONENT\_Cr}$ ) ... (9)

- 5 То есть блок 151 получения  $\text{apt\_flag}$  устанавливает значение флага  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  адаптивного первичного преобразования яркости, соответствующей флагу  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования цветности ( $\text{compID} = \text{COMPONENT\_Cb}$  или  $\text{COMPONENT\_Cr}$ ).

- 10 Следует отметить, что в качестве модифицированного примера формулы (9), блок 151 получения  $\text{apt\_flag}$  может получить флаг  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования цветности, как показано в следующей формуле (10), со ссылкой на флаг  $\text{cbf\_luma}$  наличия остаточных данных яркости.

$\text{apt\_flag}[\text{compID}] = \text{cbf\_luma} ? \text{apt\_flag}[\text{COMPONENT\_Y}] : 0$  ... (10)

- 15 То есть блок 151 получения  $\text{apt\_flag}$  устанавливает значение флага  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  адаптивного первичного преобразования яркости, соответствующей значению флага  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования цветности ( $\text{compID} = \text{COMPONENT\_Cb}$  или  $\text{COMPONENT\_Cr}$ ) в случае, когда флаг  $\text{cbf\_luma}$  наличия остаточных данных яркости равен 1 (истина), и устанавливает 0 в других случаях (в случае, когда флаг  $\text{cbf\_luma}$  наличия остаточных данных яркости равен 0 (ложь)).

- 20 Следует отметить, что в случае, когда  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  не присутствует в кодированных данных, и значение флага установлено в 0, нет необходимости ссылаться на флаг наличия остаточных данных яркости. Однако в случае, когда отсутствует операция установки  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  в 0, значение  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  становится неопределенным, и, таким образом, желательно управлять тем, следует ли  
25 ссылаться на значение флага  $\text{apt\_flag}[\text{COMPONENT\_Y}]$  адаптивного первичного преобразования яркости, со ссылкой на флаг  $\text{cbf\_luma}$  наличия остаточных данных яркости.

- Кроме того, блок 152 получения  $\text{pt\_idx}$  выполняет процесс, который относится к получению идентификатора  $\text{pt\_idx}$  первичного преобразования цветности. Например,  
30 блок 152 получения  $\text{pt\_idx}$  получает идентификатор  $\text{pt\_idx}[\text{compID}]$  первичного преобразования блока преобразования цветности, как показано в следующей формуле (11), на основании флага  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования блока преобразования цветности ( $\text{compID} = \text{COMPONENT\_Cb}$  или  $\text{COMPONENT\_Cr}$ ), и идентификатор  $\text{pt\_idx}[\text{COMPONENT\_Y}]$  первичного преобразования для блока  
35 преобразования яркости, соответствующей блоку преобразования цветности.

$\text{pt\_idx}[\text{compID}] = \text{apt\_flag}[\text{compID}] ? \text{pt\_idx}[\text{COMPONENT\_Y}] : \text{заданное значение}$  ... (11)

- То есть в случае, когда флаг  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования блока преобразования цветности, равен 1 (истина), блок 152 получения  $\text{pt\_idx}$  дополнительно устанавливает идентификатор  $\text{pt\_idx}[\text{compID}]$  первичного преобразования  
40 блока преобразования цветности в значение идентификатора первичного преобразования для блока  $\text{pt\_idx}[\text{COMPONENT\_Y}]$  преобразования яркости, соответствующей блоку преобразования цветности. В других случаях устанавливается заданное значение.

- Как описано выше, блок 131 получения информации об адаптивном первичном  
45 преобразовании цветности позволяет получить информацию об адаптивном первичном преобразовании блока преобразования цветности на основании информации об адаптивном первичном преобразовании яркости, соответствующей блоку преобразования цветности. Поэтому можно уменьшить объем обработки декодирования

без декодирования информации об адаптивном первичном преобразовании блока преобразования цветности из кодированных данных.

Блок обратного первичного преобразования

Возвращаясь к фиг.10, блок 132 выбора обратного первичного преобразования принимает ввод информации PInfo о режиме предсказания, идентификатора compID цветного сигнала, флага apt\_flag[compID] адаптивного первичного преобразования цветного сигнала, соответствующего идентификатору compID цветного сигнала, и идентификатора pt\_idx[compID] первичного преобразования цветного сигнала, соответствующего идентификатору compID цветного сигнала. Блок 132 выбора обратного первичного преобразования получает тип TrTypeIdxH преобразования обратного первичного горизонтального преобразования и тип TrTypeIdxV преобразования обратного первичного вертикального преобразования цветного сигнала, точно определенного идентификатором compID цветного сигнала со ссылкой на информацию PInfo о режиме предсказания, идентификатор compID цветного сигнала, флаг apt\_flag[compID] адаптивного первичного преобразования цветного сигнала, соответствующий идентификатору compID цветного сигнала, и идентификатор pt\_idx[compID] первичного преобразования цветного сигнала, соответствующий идентификатору compID цветного сигнала, и подает результаты в блок 134 обратного первичного горизонтального преобразования и блок 133 первичного обратного вертикального преобразования.

Блок 133 обратного первичного вертикального преобразования принимает ввод коэффициента Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию, тип TrTypeIdxV преобразования обратного первичного вертикального преобразования и размер log2TBHSize (логарифмическое значение высоты картинки), указывающий высоту картинки блока преобразования для каждого блока преобразования каждого цветного сигнала. Блок 133 обратного первичного вертикального преобразования выполняет обратное первичное вертикальное преобразование IPver, определенное типом TrTypeIdxV преобразования и высотой картинки блока преобразования, над коэффициентом Coeff\_IS преобразования, который был подвергнут обратному вторичному преобразованию, и подает результат в блок 134 обратного первичного горизонтального преобразования в виде коэффициента Coeff\_IPver преобразования, который был подвергнут обратному первичному вертикальному преобразованию. Следует отметить, что обратное первичное вертикальное преобразование IPver является обратным преобразованием по отношению к первичному вертикальному преобразованию Pver.

Блок 134 обратного первичного горизонтального преобразования принимает ввод коэффициента Coeff\_IPver преобразования, который был подвергнут обратному первичному вертикальному преобразованию, тип TrTypeIdxH преобразования обратного первичного горизонтального преобразования и размер log2TBWSize (логарифмическое значение ширины картинки), указывающий высоту картинки блока преобразования для каждого блока преобразования каждого цветного сигнала. Блок 134 обратного первичного горизонтального преобразования выполняет обратное первичное горизонтальное преобразование IPhor, определенное типом TrTypeIdxH преобразования и шириной картинки блока преобразования, над коэффициентом Coeff\_IPver преобразования, который был подвергнут обратному первичному вертикальному преобразованию, и выводит результат за пределы блока 113 обратного преобразования в виде предсказанного остатка D' (подает результат в блок 114 арифметических операций). Следует отметить, что обратное первичное горизонтальное преобразование

IPhog является обратным преобразованием по отношению к первичному горизонтальному преобразованию Phog.

Процесс, выполняемый блоком выбора обратного первичного преобразования

Далее будет подробно описано получение типа TrTypeIdxH обратного первичного горизонтального преобразования и типа TrTypeIdxV обратного первичного вертикального преобразования цветового сигнала, точно определенного идентификатором compID цветового сигнала с помощью блока 132 выбора обратного первичного преобразования.

В частности, в случае, когда флаг apt\_flag[compID] адаптивного первичного преобразования цветового сигнала, точно определенного идентификатором compID цветового сигнала, равен 1 (истина), блок 132 выбора обратного первичного преобразования выбирает набор TrSet преобразований, включающий в себя ортогональное преобразование, которое служит в качестве кандидата для первичного преобразования из четырех наборов TrSet преобразований с идентификаторами TrSetSetIdx=0... 3 наборов преобразований, показанными в таблице на фиг.1 в каждом из горизонтального направления и вертикального направлении со ссылкой на информацию PInfo о режиме предсказания. Следует отметить, что соответствие между типами ортогонального преобразования и соответствующими значениями идентификаторов TrTypeIdx не ограничивается таблицей, показанной на фиг.2, и может быть изменено в выполняемом диапазоне.

Следует отметить, что в случае, когда флаг apt\_flag[compID] адаптивного первичного преобразования равен 0 (ложь), это указывает на то, что адаптивное первичное преобразование не будет выполнено, и, таким образом, блок 132 выбора обратного первичного преобразования устанавливает значение идентификатора типа преобразования, указывающее тип заданного (например, DCT-II) ортогонального преобразования для типа TrTypeIdxH преобразования обратного первичного горизонтального преобразования и тип TrTypeIdxV преобразования обратного первичного вертикального преобразования (TrIdxTypeH = TrIdxTypeV = заданное значение).

Выбор набора преобразований

В случае, когда типом предсказания CU, к которому принадлежит обрабатываемый блок преобразования CuPredMode, является межблочным предсказанием (CuPredMode == MODE\_INTRA), установка выполняется на основании режима межблочного предсказания (IntraPredMode) в виде таблицы (LUT\_IntraModeToTrSet), показанной на фиг.3. Например, установка идентификатора TrSetIdx набора преобразований, точно определяющего соответствующий набор TrSet преобразований, выполняется для набора преобразований в каждом из направлений (TrSetH и TrSetV) с использованием следующих формул (12) и (13). Кроме того, в случае, когда типом предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, идентификатор TrSetIdx набора преобразований, точно определяющий набор TrSet преобразований для внутриблочного предсказания (=InterTrSetIdx), устанавливается в соответствии со следующей формулой (14). Следует отметить, что если режим IntraPredMode межблочного предсказания предназначен для яркости, то предполагается, что значение для режима IntraPredModeY межблочного предсказания яркости будет установлено, и если режим межблочного предсказания предназначен для цветности, то предполагается, что значение цветности для режима IntraPredModeC межблочного предсказания будет установлено.

if (CuPredMode == MODE\_INTRA) {

TrSetH = LUT\_IntraModeToTrSet [ IntraPredMode ] [ H (=0) ] ... (12)

TrSetV = LUT\_IntraModeToTrSet [ IntraPredMode ] [ V (=1) ] ... (13)

} else { // CuPredMode==MODE\_INTER

TrSetH = InterTrSetIdx

5 TrSetV = InterTrSetIdx ... (14)

}

В данном случае TrSetH представляет собой набор преобразований первичного горизонтального преобразования PThor (которое упоминается как набор первичных горизонтальных преобразований), и TrSetV представляет собой набор преобразований первичного вертикального преобразования PTver (который также упоминается как набор первичных вертикальных преобразований). Кроме того, таблица поиска LUT\_IntraModeToTrSet представляет собой таблицу соответствия, показанную на фиг.3. Первый массив таблицы LUT\_IntraModeToTrSet[][] поиска имеет режим IntraPredMode межблочного предсказания в качестве аргумента, и второй массив имеет {H=0, V=1} в качестве аргумента.

В случае, когда режим межблочного предсказания имеет номер 18 (IntraPredMode == 18), набор преобразований идентификатора TrSetIdx=2 набора преобразований, указанного в таблице (LUT\_TrSetToTrTypeIdx), показанной на фиг.1, выбирается в качестве набора TrSetH первичных горизонтальных преобразований, и набор преобразований идентификатора TrSetIdx=0 набора преобразований, указанного в таблице на фиг.1, выбирается в качестве набора TrSetV первичных вертикальных преобразований.

Следует отметить, что в случае, когда режим IntraPredMode межблочного предсказания указывает внутриблочное копирование (IBC или IntraBC) (которое также упоминается как "компенсация движения на экране") (в таблице, показанной на фиг.3, IntraPredMode == 67 соответствует IntraBC), показанное на фиг.3, можно выделить идентификатор набора преобразований, выделенный внутриблочному предсказанию. Так как внутриблочное предсказание и внутриблочное копирование имеют, в общем, тенденцию иметь схожий остаток, разумно выделить один и тот же набор преобразований. Кроме того, можно выделить идентификатор набора преобразований, выделенный межблочному копированию.

Получение флага точного определения первичного (горизонтального/вертикального) преобразования

Кроме того, блок 132 выбора обратного первичного преобразования получает флаг pt\_hor\_flag точного определения первичного горизонтального преобразования и флаг pt\_ver\_flag точного определения первичного вертикального преобразования из идентификатора pt\_idx[compID] первичного преобразования (compID=COMPONENT\_Y, COMPONENT\_Cb, COMPONENT\_Cr) цветового сигнала, точно определенного идентификатором compID цветового сигнала, например, в соответствии со следующей формулой (15).

pt\_hor\_flag = pt\_dx[compID] & 0x01

pt\_ver\_flag = pt\_idx[compID]>> 1 (= (pt\_idx[compID] & 0x10)>> 1) ... (15)

То есть идентификатор pt\_idx[compID] первичного преобразования имеет 2-битовое значение, при этом его 1 бит старшего разряда соответствует флагу pt\_ver\_flag точного определения первичного вертикального преобразования, и его 1 бит младшего разряда соответствует флагу pt\_hor\_flag точного определения первичного горизонтального преобразования.

Выбор типа преобразования обратного первичного (горизонтального/вертикального)

преобразования

Кроме того, блок 132 выбора обратного первичного преобразования выбирает тип преобразования ортогонального преобразования, которое будет использоваться при обратном первичном преобразовании из наборов TrSetH и TrSetV преобразований, выбранных в каждом горизонтальном/вертикальном направлении с использованием каждого из: флага pt\_hor\_flag точного определения первичного горизонтального преобразования и флага pt\_ver\_flag точного определения первичного вертикального преобразования.

В частности, блок 132 выбора обратного первичного преобразования определяет тип TrTypeIdxH преобразования ортогонального преобразования, которое будет применено к обратному первичному горизонтальному преобразованию на основании таблицы соответствия для наборов преобразований, и типы (LUT\_TrSetToTrTypeIdx) преобразования, показанные на фиг.1, набор TrSetH первичных горизонтальных преобразований и флаг pt\_hor\_flag точного определения первичного горизонтального преобразования, как показано в следующей формуле (16).

$$\text{TrTypeIdxH} = \text{LUT\_TrSetToTrTypeIdx} [\text{TrSetH}] [\text{pt\_hor\_flag}] \dots (16)$$

Аналогичным образом, блок 132 выбора обратного первичного преобразования определяет тип TrTypeIdxV преобразования ортогонального преобразования, которое будет применено к обратному первичному вертикальному преобразованию на основании таблицы соответствия для наборов преобразований, и типы (LUT\_TrSetToTrTypeIdx) преобразований, показанные на фиг.1, набор TrSetVx первичных вертикальных преобразований и флаг pt\_ver\_flag точного определения первичного вертикального преобразования, как показано в следующей формуле (17).

$$\text{TrTypeIdxV} = \text{LUT\_TrSetToTrTypeIdx} [\text{TrSetV}] [\text{pt\_ver\_flag}] \dots (17)$$

Если значение идентификатора TrSetIdx набора преобразований, указанное набором TrSetH первичных горизонтальных преобразований, равно 2, ортогональное преобразование, которое будет применено к первичному горизонтальному преобразованию, выбирается из набора преобразований, имеющего идентификатор TrSetIdx==2 набора преобразований в таблице LUT\_TrSetToTrTypeIdx определения набора преобразований, показанной на фиг.1. То есть в случае, когда флаг pt\_hor\_flag точного определения первичного горизонтального преобразования равен 0, значение "4" идентификатора TrTypeIdx типа преобразования, указывающего DST-VII, устанавливается для идентификатора TrTypeIdxH типа горизонтального преобразования, назначающего тип ортогонального преобразования первичного горизонтального преобразования PThor, как показано на фиг.1, и в случае, когда флаг pt\_hor\_flag точного определения первичного горизонтального преобразования равен 1, значение "1" идентификатора TrTypeIdx типа преобразования, указывающего DCT-V, устанавливается для идентификатора TrTypeIdxH типа горизонтального преобразования.

Следует отметить, что тип преобразования, определенный с помощью значений идентификатора TrSetIdx набора преобразований таблицы, показанной на фиг.1, флаг точного определения первичного горизонтального преобразования и флаг точного определения первичного вертикального преобразования могут быть установлены с возможностью изменения в пределах исполняемого диапазона.

После этого блок 132 выбора обратного первичного преобразования подает тип TrTypeIdxH преобразования обратного первичного горизонтального преобразования IPThor и тип TrTypeIdxV преобразования обратного первичного вертикального преобразования IPTver цветового сигнала, точно определенного идентификатором compID цветового сигнала, в блок 133 первичного обратного вертикального

преобразования и блок 134 обратного первичного горизонтального преобразования каждый по отдельности.

Как описано выше, блок 123 обратного первичного преобразования может применить обратное адаптивное первичное преобразование, выбранное для яркости, к блоку преобразования цветности в случае, когда остаточный сигнал цветового сигнала показывает тенденцию, аналогичную остаточному сигналу яркости. Таким образом, процесс обратного первичного преобразования для остаточного сигнала цветности можно выполнить с более высокой эффективностью кодирования, чем в предшествующем уровне техники.

Кроме того, объем обработки кодера можно уменьшить, не допуская снижения эффективности кодирования для цветности по сравнению со случаем, в котором флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования явно декодируются для каждой из яркости и цветности.

Последовательность операций процесса декодирования изображения

Далее будет описана последовательность операций каждого процесса, выполняемого устройством 100 декодирования изображения, которое описано выше. Сначала, со ссылкой на блок-схему последовательности операций, показанную на фиг.12, будет описан пример последовательности операций процесса декодирования изображения.

Когда начинается процесс декодирования изображения, на этапе S101 блок 111 декодирования декодирует битовый поток (кодированные данные), поданный в устройство 100 декодирования изображения и, таким образом, получает информацию такую как информация `Hinfo` о заголовке, информация `PInfo` о режиме предсказания, информация `Tinfo` о преобразовании, остаточная информация `Rinfo` и уровень квантованного по уровню коэффициента преобразования.

На этапе S102 блок 112 обратного квантования выполняет обратное квантование в отношении уровня квантованного по уровню коэффициента преобразования, полученного из процесса на этапе S101 и, таким образом, получает коэффициент `Coeff_IQ` преобразования. Обратное квантование представляет собой обратный процесс по отношению к квантованию, выполняемому в процессе кодирования изображения, который будет описан ниже и является аналогичным процессом по отношению к обратному квантованию, выполняемому в процессе кодирования изображения.

На этапе S103 блок 113 обратного преобразования выполняет обратное преобразование коэффициента `Coeff_IQ` преобразования, полученного в процессе на этапе S102, и, таким образом, получает предсказанный остаток `D'`. Обратное преобразование представляет собой обратный процесс по отношению к процессу преобразования, выполняемому в процессе кодирования изображения, который будет описан ниже, и является аналогичным процессом по отношению к обратному преобразованию, выполняемому в процессе кодирования изображения.

На этапе S104 блок 116 предсказания выполняет предсказание в таком же режиме предсказания, как и предсказание, выполняемое во время кодирования на основании информации `PInfo` о режиме предсказания, и вырабатывает предсказанное изображение.

На этапе S105 блок 114 арифметических операций добавляет предсказанное изображение, полученное в процессе на этапе S104, в предсказанный остаток `D'`, полученный в процессе на этапе S103, и, таким образом, получает декодированное изображение.

На этапе S106 блок 114 арифметических операций выводит декодированное изображение, полученное в процессе на этапе S105 за пределы устройства 100 декодирования изображения.

На этапе S107 память 115 кадров сохраняет декодированное изображение, полученное в процессе на этапе S105.

Когда процесс на этапе S107 заканчивается, процесс декодирования изображения заканчивается.

5 Процесс декодирования информации о первичном преобразовании

Различные типы информации, включенные в кодированные данные #1, декодируются на этапе S101, показанном на фиг.12, как описано выше. В это время блок 111 декодирования декодирует также надлежащим образом информацию, такую как флаг apt\_flag адаптивного первичного преобразования, идентификатор pt\_idx первичного преобразования и т.п. Блок 111 декодирования декодирует, например, флаг apt\_flag [COMPONENT\_Y] адаптивного первичного преобразования яркости и идентификатор pt\_idx[COMPONENT\_Y] первичного преобразования яркости. Однако кодированные данные #1 не включают в себя флаг apt\_flag[compID] адаптивного первичного преобразования цветности (compID=COMPONENT\_Cb или COMPONENT\_Cr) и идентификатор pt\_idx[compID] первичного преобразования цветности (compID=COMPONENT\_Cb или COMPONENT\_Cr), как описано выше. Таким образом, блок 111 декодирования исключает декодирование информации элементов.

Для того, чтобы выполнить декодирование так, как описано выше, блок 111 декодирования выполняет процесс декодирования информации о первичном преобразовании на этапе S101. Пример последовательности операций процесса декодирования информации о первичном преобразовании будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.13.

Когда начинается процесс декодирования информации о первичном преобразовании, на этапе S111 блок 111 декодирования определяет, является ли компонент, подлежащий обработке, яркостью (compID == COMPONENT\_Y). В случае, когда определяется, что компонентом является яркость, процесс переходит к этапу S112.

На этапе S112 блок 111 декодирования может декодировать флаг apt\_flag [COMPONENT\_Y] адаптивного первичного преобразования яркости, включенный в кодированные данные #1. На этапе S113 блок 111 декодирования может декодировать идентификатор pt\_idx[COMPONENT\_Y] первичного преобразования яркости, включенный в кодированные данные #1. Когда процесс на этапе S113 заканчивается, процесс декодирования информации о первичном преобразовании заканчивается.

Кроме того, в случае, когда на этапе S111 компонент, подлежащий обработке определяется как цветность (compID != COMPONENT\_Y), процессы на этапе S112 и на этапе S113 опускаются, и процесс декодирования информации о первичном преобразовании заканчивается. То есть в случае цветности, декодирование флага apt\_flag адаптивного первичного преобразования и идентификатора pt\_idx первичного преобразования опускается. Поэтому можно сдерживать увеличение нагрузки декодирования.

40 Последовательность операций процесса обратного преобразования

Далее, со ссылкой на блок-схему последовательности операций, показанную на фиг.14, будет описан пример последовательности операций процесса обратного преобразования, выполняемого на этапе S103, который показан на фиг.12. Когда начинается процесс обратного преобразования, переключатель 121 на этапе S121 определяет, представляет ли собой флаг ts\_flag пропуска преобразования 2D\_TS (режим пропуска двумерного преобразования), или равен ли 1 (истина) флаг transquant\_bypass\_flag обхода квантования преобразования. В случае, когда определяется, что идентификатор ts\_idx 2D\_TS пропуска преобразования или флаг обхода квантования

преобразования равен 1 (истина), переключатель 121 выводит коэффициент  $\text{Coeff\_IQ}$  преобразования наружу в виде предсказанного остатка  $D'$  (который подается в блок 114 арифметических операций), процесс обратного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.12.

5 Кроме того, в случае, когда на этапе S121 определяется, что идентификатор  $\text{ts\_idx}$  пропуска преобразования не представляет собой 2D\_TS (режим, отличный от пропуска двумерного преобразования), и флаг обхода квантования преобразования равен 0 (ложь), переключатель 121 подает коэффициент  $\text{Coeff\_IQ}$  преобразования в блок 122 обратного вторичного преобразования, и процесс переходит к этапу S122.

10 На этапе S122 блок 122 обратного вторичного преобразования выполняет обратное вторичное преобразование над входным коэффициентом  $\text{Coeff\_IQ}$  преобразования на основании идентификатора  $\text{st\_idx}$  вторичного преобразования, и получает и выводит коэффициент  $\text{Coeff\_IS}$  преобразования, который был подвергнут обратному вторичному преобразованию.

15 На этапе S123 блок 132 выбора обратного первичного преобразования определяет, указывает ли идентификатор  $\text{compID}$  цветового сигнала яркость или цветность. В случае, когда идентификатор  $\text{compID}$  цветового сигнала указывает цветность ( $\text{compID} = \text{COMPONENT\_Y}$ ), процесс переходит к этапу S124. На этапе S124 блок 131 получения информации об адаптивном первичном преобразовании цветности получает

20 идентификатор  $\text{pt\_idx}[\text{compID}]$  первичного преобразования для блока преобразования цветности на основании флага  $\text{apt\_flag}[\text{compID}]$  ( $\text{compID} = \text{COMPONENT\_Cb}$  или  $\text{COMPONENT\_Cr}$ ) адаптивного первичного преобразования блока преобразования цветности, и идентификатор  $\text{pt\_idx}[\text{COMPONENT\_Y}]$  первичного преобразования для блока преобразования яркости, соответствующей блоку преобразования цветности.

25 Когда процесс на этапе S124 заканчивается, процесс переходит к этапу S125.

Кроме того, на этапе S123, в случае, когда идентификатор  $\text{compID}$  цветового сигнала указывает яркость ( $\text{compID} = \text{COMPONENT\_Y}$ ), процесс переходит к этапу S125.

На этапе S125 блок 132 выбора обратного первичного преобразования получает тип  $\text{TrTypeIdxH}$  преобразования обратного первичного горизонтального преобразования  
30 и тип  $\text{TrTypeIdxV}$  преобразования обратного первичного вертикального преобразования цветового сигнала, точно определенного идентификатором  $\text{compID}$  цветового сигнала со ссылкой на информацию  $\text{PInfo}$  о режиме предсказания, идентификатор  $\text{compID}$  цветового сигнала, флаг  $\text{apt\_flag}[\text{compID}]$  адаптивного первичного преобразования цветового сигнала, соответствующего идентификатору  $\text{compID}$  цветового сигнала, и  
35 идентификатор  $\text{pt\_idx}[\text{compID}]$  первичного преобразования цветового сигнала, соответствующего идентификатору  $\text{compID}$  цветового сигнала.

На этапе S126 блок 133 обратного первичного вертикального преобразования выполняет обратное первичное вертикальное преобразование  $\text{IPver}$ , определенное типом  $\text{TrTypeIdxV}$  преобразования обратного первичного вертикального  
40 преобразования и высотой картинки блока преобразования, над коэффициентом  $\text{Coeff\_IS}$  преобразования, который был подвергнут обратному вторичному преобразованию блока преобразования для каждого блока преобразования, точно определенного идентификатором  $\text{compID}$  цветового сигнала, и выводит результат в виде коэффициента  $\text{Coeff\_IPver}$  преобразования, который был подвергнут обратному первичному  
45 вертикальному преобразованию.

На этапе S127 блок 134 обратного первичного горизонтального преобразования выполняет обратное первичное горизонтальное преобразование  $\text{IPhor}$ , определенное типом  $\text{TrTypeIdxH}$  преобразования обратного первичного горизонтального

преобразования и высотой картинки блока преобразования, над коэффициентом Coeff\_IPver преобразования, который подвергся обратному первичному вертикальному преобразованию блока преобразования для каждого блока преобразования, точно определенного идентификатором compID цветового сигнала, и выводит результат в виде предсказанного остатка D'. Когда процесс на этапе S127 заканчивается, процесс обратного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.12.

Последовательность операций процесса получения информации об адаптивном первичном преобразовании цветности

Далее, со ссылкой на блок-схему последовательности операций, показанную на фиг.15, будет описан пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности, выполняемого на этапе S124, показанном на фиг.14. Когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S131 блок 151 получения apt\_flag из блока 131 получения информации об адаптивном первичном преобразовании цветности устанавливает значение флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования яркости для значения флага apt\_flag[compID] адаптивного первичного преобразования цветности.

На этапе S132 блок 152 получения pt\_idx блока 131 получения информации об адаптивном первичном преобразовании цветности определяет, имеет ли флаг apt\_flag[compID] адаптивного первичного преобразования цветности значение "истина". Когда определяется, что он имеет значение "истина", процесс переходит к этапу S133. На этапе S133 блок 152 получения pt\_idx устанавливает значение идентификатора pt\_idx[COMPONENT\_Y] первичного преобразования яркости для идентификатора pt\_idx[compID] первичного преобразования цветности. Когда процесс на этапе S133 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

Кроме того, в случае, когда на этапе S132, показанном на фиг.15, определяется, что флаг apt\_flag[compID] адаптивного первичного преобразования цветности имеет значение "ложь", процесс переходит к этапу S134. На этапе S134 блок 152 получения pt\_idx устанавливает заданное значение для идентификатора pt\_idx[compID] первичного преобразования цветности. Когда процесс на этапе S134 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

Последовательность операций процесса выбора обратного первичного преобразования

Далее, пример последовательности операций процесса выбора обратного первичного преобразования, выполняемого на этапе S125, показанном на фиг.14, будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.16.

Когда начинается процесс выбора обратного первичного преобразования, на этапе S141 блок 132 выбора обратного первичного преобразования определяет, равен ли 1 (истина) флаг apt\_flag[compID] адаптивного первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала. В случае, когда определяется, что флаг apt\_flag[compID] адаптивного первичного преобразования равен 1 (истина), процесс переходит к этапу S142.

На этапе S142 блок 132 выбора обратного первичного преобразования выбирает набор TrSetV преобразований (набор первичных горизонтальных преобразований)

обратного первичного вертикального преобразования и набор TrSetH преобразований (набор первичных вертикальных преобразований) обратного первичного горизонтального преобразования на основании информации PInfo о режиме предсказания.

5 На этапе S143 блок 132 выбора обратного первичного преобразования получает флаг pt\_hor\_flag точного определения первичного горизонтального преобразования и флаг pt\_ver\_flag точного определения первичного вертикального преобразования из идентификатора pt\_idx[compID] первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала.

10 На этапе S144 блок 132 выбора обратного первичного преобразования выбирает тип TrTypeIdxH преобразования ортогонального преобразования, применяемого в качестве обратного первичного горизонтального преобразования IPThor со ссылкой на набор TrSetH первичных горизонтальных преобразований и флаг pt\_hor\_flag точного определения первичного горизонтального преобразования.

15 На этапе S145 блок 132 выбора обратного первичного преобразования выбирает тип TrTypeIdxV преобразования ортогонального преобразования, применяемого в качестве обратного первичного вертикального преобразования IPTver по отношению к набору TrSetV первичных вертикальных преобразований, и флаг pt\_ver\_flag точного определения первичного вертикального преобразования. Когда процесс на этапе S145  
20 заканчивается, процесс выбора обратного первичного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

Кроме того, на этапе S141 в случае, когда определяется, что флаг apt\_flag[compID] адаптивного первичного преобразования равен 0 (ложь), процесс переходит к этапу S146. На этапе S146 блок 132 выбора обратного первичного преобразования выбирает  
25 заданное ортогональное преобразование в качестве типа TrTypeIdxH преобразования обратного первичного горизонтального преобразования IPThor (TrTypeIdxH = заданное значение).

На этапе S147 блок 132 выбора обратного первичного преобразования выбирает заданное ортогональное преобразование в качестве типа TrTypeIdxV преобразования  
30 обратного первичного вертикального преобразования IPTver (TrTypeIdxV = заданное значение). Когда процесс на этапе S147 заканчивается, процесс выбора обратного первичного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

То есть блок 132 выбора обратного первичного преобразования получает тип  
35 TrTypeIdxH преобразования обратного первичного горизонтального преобразования IPThor и тип TrTypeIdxV преобразования обратного первичного вертикального преобразования IPTver, используя способ в соответствии со значением флага apt\_flag[compID] адаптивного первичного преобразования цветности.

Как описано выше, блок 113 обратного преобразования, включенный в устройство  
40 100 декодирования изображения, может применить обратное адаптивное первичное преобразование, выбранное для яркости, к блоку преобразования цветности в случае, когда, остаточный сигнал цветности показывает тенденцию, аналогичную остаточному сигналу яркости. Поэтому обратный процесс первичного преобразования может быть выполнен над остаточным сигналом цветности с еще более увеличенной эффективностью  
45 кодирования по сравнению с уровнем техники. Кроме того, объемы обработки кодирования и декодирования могут быть уменьшены при уменьшении ухудшения эффективности кодирования для цветности по сравнению со случаем, когда флаг apt\_flag адаптивного первичного преобразования и идентификатор pt\_idx первичного

преобразования явно декодируются для каждой из яркости и цветности.

#### Модифицированный пример #1

Следует отметить, что получение флага адаптивного первичного преобразования цветности может быть ограничено случаем, когда типом предсказания является интер-  
 5 CU. Например, флаг `apt_flag[compID]` (`compID=COMPONENT_Cb` или `COMPONENT_Cr`) адаптивного первичного преобразования по отношению к блоку преобразования цветности, подлежащему обработке, может быть получен на основании флага `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку преобразования цветности и типу  
 10 `CuPredMode` предсказания CU (единицы кодирования), к которому принадлежит блок преобразования цветности, как показано в следующей формуле (18).

```

    if ( CuPredMode == MODE_INTER ) {
        apt_flag[compID] = apt_flag[COMPONENT_Y]
    } else {
15 apt_flag[compID] = 0
    }
    ... (18)
  
```

В случае, когда типом `CuPredMode` предсказания CU, к которому принадлежит блок преобразования цветности, является внутриблочное предсказание (`CuPredMode ==`  
 20 `MODE_INTER`), например, флаг `apt_flag[compID]` адаптивного первичного преобразования по отношению к блоку преобразования цветности установлен на основании флага `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования блока преобразования яркости, соответствующей блоку преобразования цветности. С другой стороны, в случае, когда типом `CuPredMode` предсказания CU, к которому  
 25 принадлежит блок преобразования цветности, не является внутриблочное предсказание (но является межблочное предсказание), значение флага `apt_flag[compID]` адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается в 0 (ложь).

Пример последовательности операций процесса получения информации об  
 30 адаптивном первичном преобразовании цветности такого случая будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.17. Блок-схема последовательности операций соответствует блок-схеме последовательности операций, показанной на фиг.15. В этом случае, когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S151 блок  
 35 151 получения `apt_flag` определяет, является ли типом `CUPredMode` предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, внутриблочное предсказание (`MODE_INTER`). В случае, когда определяется, что типом предсказания является внутриблочное предсказание, процесс переходит к этапу S152.

На этапе S152 блок 151 получения `apt_flag` устанавливает значение флага `apt_flag`  
 40 `[COMPONENT_Y]` адаптивного первичного преобразования яркости для значения флага `apt_flag[compID]` адаптивного первичного преобразования цветности, как и в случае этапа S131, показанного на фиг.15. Когда процесс на этапе S152 заканчивается, процесс переходит к этапу S154.

Кроме того, в случае, когда на этапе S151 определяется, что типом `CUPredMode`  
 45 предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, является межблочное предсказание, процесс переходит к этапу S153. На этапе S153 блок 151 получения `apt_flag` устанавливает флаг `apt_flag[compID]` адаптивного первичного преобразования цветности в 0 (ложь). Когда процесс на этапе S153

заканчивается, процесс переходит к этапу S154.

Каждый из процессов этапа S154 – этапа S156 выполняется аналогично каждому из процессов этапа S132 – этапа S134, показанных на фиг.15. Таким образом, когда процесс на этапе S155 или этапе S156 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

В случае, когда типом предсказания является межблочное предсказание, возможен случай, когда режимы межблочного предсказания отличаются друг от друга по яркости и цветности. В это время, так как остаточные сигналы яркости и цветности проявляют различные тенденции, если информация об адаптивном первичном преобразовании яркости применяется к цветности, существует вероятность снижения эффективности. Поэтому за счет применения информации об адаптивном первичном преобразовании яркости к цветности только в случае, когда выполняется процесс получения информации об адаптивном первичном преобразовании цветности, как показано на фиг.17, а типом предсказания является внутриблочное предсказание, можно дополнительно сдерживать снижение эффективности кодирования, чем в случае, показанном на фиг.15.

#### Модифицированный пример #2

Кроме того, получение флага адаптивного первичного преобразования цветности может быть ограничено интер-CU или интра-CU и в том случае, когда режимом межблочного предсказания яркости является внутриблочное копирование (компенсация движения на экране). Например, флаг apt\_flag[compID] (compID=COMPONENT\_Cb или COMPONENT\_Cr) адаптивного первичного преобразования блока преобразования цветности, подлежащего обработке, может быть установлен на основании флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования блока преобразования яркости, соответствующей блоку преобразования цветности и типу CUPredMode предсказания CU (единицы кодирования), к которому принадлежит блок преобразования цветности, и информации PInfo о режиме предсказания, как показано в следующей формуле (19).

```

if ( CuPredMode == MODE_INTER ) {
    apt_flag[compID] = apt_flag[COMPONENT_Y]
} else if (IntraPredModeC == "IntraBC") { // CuPredMODE == MODE_INTRA
    apt_flag[compID] = apt_flag[COMPONENT_Y]
} else {
    apt_flag[compID] = 0
}
... (19)

```

То есть в случае, когда типом CuPredMode предсказания CU, к которому принадлежит блок преобразования цветности, является внутриблочное предсказание, или в случае, когда типом предсказания CU является межблочное предсказанием, и режимом IntraPredModeC межблочного предсказания цветности является внутриблочное копирование (IntraBC), флаг apt\_flag[compID] адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается на основании флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку преобразования цветности. С другой стороны, в других случаях (в случае, когда типом CUPredMode предсказания CU является межблочное предсказание, и режимом IntraPredModeC межблочного предсказания не является внутриблочное копирование) значение флага apt\_flag[compID] адаптивного первичного преобразования по отношению к блоку преобразования

цветности устанавливается в 0 (ложь). Следует отметить, что в случае, когда информация о яркости и цветности включена в один и тот же CU вместо режима межблочного предсказания IntraPredModeC цветности, условное определение может быть выполнено с использованием режима IntraPredModeY межблочного предсказания яркости в

5 вышеописанной формуле (19).

Пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности такого случая будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.18. Блок-схема последовательности операций соответствует блок-схеме последовательности

10 операций, показанной на фиг.15. В этом случае, когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S171 блок 151 получения apt\_flag определяет, является ли типом CUPredMode предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, внутриблочное предсказание (MODE\_INTER). В случае, когда определяется, что типом предсказания

15 является межблочное предсказание, процесс переходит к этапу S172.

На этапе S172 блок 151 получения apt\_flag определяет, является ли режимом IntraPredModeC межблочного предсказания цветности внутриблочное копирование (IntraBC). В случае, когда определяется, что режимом является внутриблочное копирование (IntraBC), процесс переходит к этапу S173. Кроме того, когда на этапе

20 S171 определяется, что типом CUPredMode предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание (MODE\_INTER), процесс переходит к этапу S173.

На этапе S173 блок 151 получения apt\_flag устанавливает значение флага apt\_flag [COMPONENT\_Y] адаптивного первичного преобразования яркости для значения

25 флага apt\_flag[compID] адаптивного первичного преобразования цветности, как и в случае этапа S131, показанного на фиг.15. Когда процесс на этапе S173 заканчивается, процесс переходит к этапу S175.

Кроме того, в случае, когда на этапе S172 определяется, что режимом IntraPredModeC межблочного предсказания цветности не является внутриблочное копирование (IntraBC),

30 процесс переходит к этапу S174. На этапе S174 блок 151 получения apt\_flag устанавливает значение флага apt\_flag[compID] адаптивного первичного преобразования цветности в 0 (ложь). Когда процесс на этапе S174 заканчивается, процесс переходит к этапу S175.

Каждый из процессов этапа S175 – этапа S177 выполняется аналогично каждому из процессов этапа S132 – этапа S134, показанных на фиг.15. Таким образом, когда процесс

35 на этапе S176 или этапе S177 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

В случае, когда типом предсказания является межблочное предсказание, и режимом межблочного предсказания IntraPredMode является внутриблочное копирование,

40 тенденции остатка яркости и остатка цветности являются аналогичными, как и в случае внутриблочного предсказания. Поэтому, так как адаптивное первичное преобразование яркости может быть применено к цветности в случае, когда типом предсказания является межблочное предсказание, и режимом IntraPredMode межблочного предсказания является IntraBC, эффективность кодирования можно дополнительно повысить по сравнению с

45 модифицированным примером #1.

Модифицированный пример #3

Кроме того, получение флага адаптивного первичного преобразования цветности может быть ограничено интер-CU или интра-CU и в том случае, когда режим

межблочного предсказания яркости эквивалентен режиму межблочного предсказания цветности. Например, флаг `apt_flag[compID]` (`compID=COMPONENT_Cb` или `COMPONENT_Cr`) адаптивного первичного преобразования блока преобразования цветности, подлежащего обработке, может быть установлен на основании флага `apt_flag` [COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку преобразования цветности, типу `CUPredMode` предсказания CU (единицы кодирования), к которому принадлежит блок преобразования цветности, режиму `IntraPredModeY` межблочного предсказания яркости CU и режиму `IntraPredModeC` межблочного предсказания цветности CU, как показано

в следующей формуле (20).

```

if ( CuPredMode == MODE_INTER ) {
    apt_flag[compID] = apt_flag[COMPONENT_Y]
} else if (IntraPredModeY == IntraPredModeC) { // CuPredMode==MODE_INTRA
    apt_flag[compID] = apt_flag[COMPONENT_Y]
} else {
    apt_flag[compID] = 0
}
... (20)

```

То есть в случае, когда типом `CUPredMode` предсказания CU, к которому принадлежит блок преобразования цветности, является внутриблочное предсказание, или типом предсказания CU является межблочное предсказание, а режим `IntraPredModeC` межблочного предсказания цветности совпадает с режимом `IntraPredModeY` межблочного предсказания яркости, флаг `apt_flag[compID]` адаптивного первичного преобразования блока преобразования цветности устанавливается на основании флага `apt_flag` [COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку преобразования цветности. С другой стороны, в других случаях (в случае, когда типом `CUPredMode` предсказания CU является межблочное предсказание, и режим межблочного предсказания цветности отличается от режима межблочного предсказания яркости) значение флага `apt_flag[compID]` адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается в 0 (ложь).

Пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности такого случая будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.19. Блок-схема последовательности операций соответствует блок-схеме последовательности операций, показанной на фиг.15. В этом случае, когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S181 блок 151 получения `apt_flag` определяет, является ли типом `CUPredMode` предсказания CU, к которому принадлежит блок преобразования, подлежащий обработке, внутриблочное предсказание (`MODE_INTER`). В случае, когда определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S182.

На этапе S182 блок 151 получения `apt_flag` определяет, совпадает или нет режим `IntraPredModeC` межблочного предсказания цветности с режимом `IntraPredModeY` межблочного предсказания яркости. В случае, когда режимы совпадают, процесс переходит к этапу S183. Кроме того, на этапе S182 определяется, что режим `IntraPredModeC` межблочного предсказания цветности не совпадает с режимом `IntraPredModeY` межблочного предсказания яркости, процесс переходит к этапу S184.

Каждый из процессов этапа S183 – этапа S187 выполняется аналогично каждому из

процессов этапа S173 – этапа S177, показанных на фиг.18. Таким образом, когда процесс на этапе S186 или этапе S187 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

5 В случае, когда типом предсказания CU является межблочное предсказание, то есть яркость и цветность имеют один и тот же режим межблочного предсказания, остаточные сигналы яркости и цветности имеют аналогичные тенденции. Таким образом, за счет применения информации об адаптивном первичном преобразовании яркости к цветности только в том случае, когда режим межблочного предсказания яркости имеет такое же  
10 значение, как и режим межблочного предсказания цветности, адаптивное первичное преобразование яркости может быть применено к цветности в случае, когда типом предсказания является межблочное предсказание, и режим межблочного предсказания яркости имеет такое же значение, как и режим межблочного предсказания цветности, и, таким образом, можно еще больше повысить эффективность кодирования, чем в  
15 случае модифицированного примера #1.

#### Модифицированный пример #4

Кроме того, можно добавить флаг, указывающий на то, установлена или нет информация относительно адаптивного первичного преобразования цветности на основании адаптивного первичного преобразования яркости, и установку информации  
20 относительно адаптивного первичного преобразования цветности можно контролировать с использованием флага. Например, флаг apt\_flag[compID] (compID=COMPONENT\_Cb или COMPONENT\_Cr) адаптивного первичного преобразования по отношению к блоку преобразования цветности, подлежащему обработке, может быть установлен на основании флага chroma\_apt\_infer\_flag вывода информации об адаптивном  
25 первичном преобразовании цветности, указывающего на то, выполнить или нет установку (вывод) с использованием информации об адаптивном первичном преобразовании блока преобразования яркости, соответствующей блоку преобразования цветности, и флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку  
30 преобразования цветности, как показано в следующей формуле (21).

```

    if ( chroma_apt_info_infer_flag [compID] ) {
        apt_flag[compID] = apt_flag[COMPONENT_Y] : 0
    } else {
        apt_flag[compID] = 0
    }
    ... (21)

```

То есть в случае, когда флаг chroma\_apt\_info\_infer\_flag [compID] вывода информации об адаптивном первичном преобразовании цветности равен 1 (истина), флаг apt\_flag [compID] адаптивного первичного преобразования по отношению к блоку  
40 преобразования цветности устанавливается на основании флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости блока преобразования цветности. С другой стороны, в других случаях (в случае, когда флаг chroma\_apt\_info\_infer\_flag [compID] вывода информации об адаптивном первичном преобразовании цветности имеет значение 0 (ложь)), значение флага apt\_flag  
45 [compID] адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается в 0 (ложь).

Пример синтаксиса TU в этом случае показан поз.А на фиг.20. Семантика каждого параметра показана поз.В на фиг.20. Кроме того, пример синтаксиса остаточного

кодирования, включенного в синтаксис А на фиг.20, показан на фиг.21. Флаг `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности декодируется блоком 111 декодирования в соответствии с синтаксической таблицей, показанной на фиг.21.

5 То есть в случае, когда идентификатор `compID` цветового сигнала указывает цветность, флаг `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости равен 1 (истина), флаг `ts_flag[compID]` пропуска преобразования цветового сигнала, соответствующего идентификатору `compID` цветового сигнала, равен 0 (ложь), и флаг `transquant_bypass_flag` обхода квантования преобразования равен 0 (ложь), декодируется флаг  
10 `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности. В случае других условий предполагается, что значение флага `chroma_apr_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности равно 0.

Блок 111 декодирования выполняет такое декодирование на этапе S101, показанном  
15 на фиг.12 путем выполнения процесса декодирования флага вывода информации адаптивного первичного преобразования цветности. Пример последовательности операций процесса декодирования флага вывода информации адаптивного первичного преобразования цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.22.

20 Когда начинается процесс декодирования флага вывода информации адаптивного первичного преобразования цветности, на этапе S191 блок 111 декодирования определяет, является ли идентификатором `compID` цветового сигнала яркость. В случае, когда определяется, что идентификатором `compID` цветового сигнала является цветность (`compID != COMPONENT_Y`), процесс переходит к этапу S192. На этапе S192 блок 111  
25 декодирования определяет, равен ли флаг `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости 1 (истина). В том случае, когда определяется, что флаг `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости равен 1 (истина), процесс переходит к этапу S193.

На этапе S193 блок 111 декодирования определяет, равен ли флаг  
30 `transquant_bypass_flag` обхода квантования преобразования 1 (истина). В случае, когда определяется, что флаг обхода квантования преобразования, равен 0 (ложь) (! `transquant_bypass_flag`), процесс переходит к этапу S194. На этапе S194 блок 111 декодирования определяет, равен ли флаг `ts_flag[compID]` пропуска преобразования цветового сигнала, соответствующего идентификатору `compID` цветового сигнала, 1  
35 (истина). В случае, когда определяется, что флаг `ts_flag[compID]` пропуска преобразования равен 0 (ложь) (! `ts_flag[compID]`), процесс переходит к этапу S195.

На этапе S195 блок 111 декодирования декодирует флаг `chroma_apr_info_infer_flag` [ `compID` ] вывода информации об адаптивном первичном преобразовании цветности, соответствующий идентификатору `compID` цветового сигнала из битовой строки  
40 кодированных данных #1, и выводит результат как часть информации `Tinfo` о преобразовании. Когда процесс на этапе S195 заканчивается, процесс декодирования флага вывода информации адаптивного первичного преобразования цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.12.

Кроме того, в случае, когда на этапе S191 идентификатор `compID` цветового сигнала  
45 представляет собой яркость (`compID == COMPONENT_Y`), процесс переходит к этапу S196. Кроме того, в случае, когда на этапе S192 определяется, что флаг `apt_flag` [ `COMPONENT_Y` ] адаптивного первичного преобразования яркости равен 0 (ложь) (! `apt_flag[COMPONENT_Y]`), процесс переходит к этапу S196. Кроме того, в случае, когда

на этапе S193 определяется, что флаг обхода квантования преобразования равен 1 (истина) (`transquant_bypass_flag`), процесс переходит к этапу S196. Кроме того, в случае, когда на этапе S194 флаг `ts_flag[compID]` пропуска преобразования равен 1 (истина) (`ts_flag[compID]`), процесс переходит к этапу S196.

5 На этапе S196 блок 111 декодирования исключает декодирование флага `chroma_apr_info_infer_flag [compID]` вывода информации об адаптивном первичном преобразовании цветности, соответствующего идентификатору `compID` цветового сигнала, и устанавливает значение флага в 0 (ложь) (`chroma_apr_info_infer_flag[compID] = 0`). Когда процесс на этапе S196 заканчивается, процесс декодирования флага вывода информации адаптивного первичного преобразования цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.12.

Блок 131 получения информации об адаптивном первичном преобразовании цветности выполняет процесс получения информации об адаптивном первичном преобразовании цветности с использованием флага `chroma_apr_info_infer_flag [compID]` вывода информации об адаптивном первичном преобразовании цветности, декодированного или установленного так, как описано выше. Пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности случая будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.23. Эта блок-схема последовательности операций соответствует блок-схеме последовательности операций, показанной на фиг.15. В этом случае, когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S201 блок 151 получения `apt_flag` определяет, имеет ли значение флага `chroma_apr_info_infer_flag [compID]` вывода информации об адаптивном первичном преобразовании цветности значение "истина". В случае, когда результатом определения имеет значение "истина", процесс переходит к этапу S202. Кроме того, в случае, когда определяется, что значение флага `chroma_apr_info_infer_flag [compID]` вывода информации об адаптивном первичном преобразовании цветности имеет значение "ложь" на этапе S201, процесс переходит к этапу S203.

Каждый из процессов этапа S202 – этапа S206 выполняется аналогично каждому из процессов этапа S183 – этапа S187, показанных на фиг.19. Таким образом, когда процесс на этапе S205 или этапе S206 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

То, следует ли применять информацию адаптивного первичного преобразования яркости к цветности, можно однозначно контролировать на основании флага вывода информации об адаптивном первичном преобразовании цветности по отношению к сигналам, показывающим различные тенденции по остатку яркости и остатку цветности. Поэтому адаптивное первичное преобразование может быть применено к цветности с более высокой эффективностью, чем в случае, показанном на фиг.15, и, таким образом, можно повысить эффективность кодирования.

#### Модифицированный пример #5

Кроме того, получение флага адаптивного первичного преобразования цветности может быть ограничено случаем, когда размер короткой стороны блока преобразования цветности больше или равен значению заданного порогового значения. Например, флаг `apt_flag[compID]` (`compID=COMPONENT_Cb` или `COMPONENT_Cr`) адаптивного первичного преобразования по отношению к блоку преобразования цветности, подлежащему обработке, может быть установлен на основании флага `apt_flag [COMPONENT_Y]` адаптивного первичного преобразования по отношению к блоку

преобразования яркости, соответствующей блоку преобразования цветности и размеру короткой стороны блока преобразования цветности, как показано в следующей формуле (22).

```

    if ( min (log2TBWSize, log2TBHSize) >= TH ) {
5      apt_flag[compID] = apt_flag[COMPONENT_Y]
    } else {
      apt_flag[compID] = 0
    }
    ... (22)

```

То есть в случае, когда размер короткой стороны блока преобразования цветности больше или равен пороговому значению ( $\min(\log_2TBWSize, \log_2TBHSize) \geq TH$ ), флаг apt\_flag[compID] адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается на основании флага apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования по отношению к блоку преобразования яркости, соответствующей блоку преобразования цветности. С другой стороны, в других случаях (в случае, когда размер короткой стороны блока преобразования цветности меньше порогового значения) значение флага apt\_flag[compID] адаптивного первичного преобразования по отношению к блоку преобразования цветности устанавливается в 0 (ложь).

Пример последовательности операций процесса получения информации об адаптивном первичном преобразовании цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.24. Эта блок-схема последовательности операций соответствует блок-схеме последовательности операций, показанной на фиг.15. В этом случае, когда начинается процесс получения информации об адаптивном первичном преобразовании цветности, на этапе S211 блок 151 получения apt\_flag определяет, больше или равен размер ( $\min(\log_2TBWSize, \log_2TBHSize)$ ) короткой стороны блока преобразования, подлежащего обработке, заданному пороговому значению TH. В случае, когда определяется, что размер короткой стороны блока преобразования, подлежащего обработке, больше или равен значению заданного порогового значения ( $\min(\log_2TBWSize, \log_2TBHSize) \geq TH$ ), процесс переходит к этапу S212. Кроме того, в случае, когда определяется, что размер короткой стороны блока преобразования, подлежащего обработке, меньше заданного порогового значения ( $\min(\log_2TBWSize, \log_2TBHSize) < TH$ ) на этапе S211, процесс переходит к этапу S213.

Каждый из процессов этапа S212 – этапа S216 выполняется аналогично каждому из процессов этапа S202 – этапа S206, показанных на фиг.23. Таким образом, когда процесс на этапе S215 или этапе S216 заканчивается, процесс получения информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.14.

В случае, когда размер короткой стороны блока преобразования цветности меньше заданного значения (например, 4), даже в том случае, если адаптивное первичное преобразование DST-7/DST-1/DCT-5/DCT-8 или т.п. для размера, равного размеру короткой стороны, применяется к короткой стороне, диапазон улучшений эффективности кодирования является маленьким. По этой причине, не применяя адаптивное первичное преобразование в случае, когда размер короткой стороны блока преобразования цветности меньше заданного значения, масштаб схемы по отношению к преобразованию, используемому при адаптивном первичном преобразовании для размера, меньшего, чем заданное значение, можно уменьшить при подавлении снижения

эффективности кодирования.

Модифицированный пример #6

Кроме того, можно выбрать тип преобразования первичного горизонтального преобразования IP<sub>hor</sub> и обратного первичного вертикального преобразования IP<sub>ver</sub>, используя размеры высоты изображения и ширины изображения блока преобразования цветности. Например, хотя выше было описано, что блок 132 выбора обратного первичного преобразования определяет тип ортогонального преобразования, которое будет применено к обратному преобразованию каждого направления на основании таблицы соответствия наборов преобразований и типов (LUT\_TrSetToTrTypeIdx) преобразований, показанных на фиг.1, набор TrSet{H,V} первичных {горизонтального, вертикального} преобразований и флаг pt\_{hor,ver}\_flag точного определения набора первичных {горизонтального, вертикального} преобразований, технология не ограничивается этим. Например, в случае блока преобразования цветности тип преобразования каждого направления может быть определен дополнительно со ссылкой на размер блока преобразования цветности.

В случае, когда на этапе S144, показанном на фиг.16, ширина картинка блока преобразования цветности (в данном случае логарифмическое значение ширины log2TBWSize картинка) равна или меньше порогового значения TH, как показано в следующей формуле (23), например, тип TrTypeIdxH преобразования (обратного) первичного горизонтального преобразования может быть установлен в заданный тип (DCT-2) преобразования, и в других случаях тип TrTypeIdxH преобразования (обратного) первичного горизонтального преобразования может быть установлен на основании набора TrSetH горизонтальных преобразований и флага pt\_hor\_flag точного определения первичного горизонтального преобразования. Следует отметить, что в формуле (23), в конкретном примере пороговое значение TH может быть установлено, например, на 1 или 2. Следует отметить, что в формуле (23), логарифмическое значение log2TBWSize ширины изображения блока преобразования можно заменить на ширину TBW изображения. В этом случае пороговое значение TH' устанавливается на  $1 < TH$ .

```

if (compID==COMPONENT_Y ) {
    TrTypeIdxH = LUT_TrSetToTrTypeIdx[ TrSetH ][ pt_hor_flag ]
} else if (log2TBWSize<=TH) {
    TrTypeIdxH = predetermined value (set to the value of the transform type indicating DCT-II)
} else {
    TrTypeIdxH = LUT_TrSetToTrTypeIdx[ TrSetH ][ pt_hor_flag ]
}
... (23)

```

Пример последовательности операций процесса получения типа первичного горизонтального преобразования, выполняемого на этапе S144, показанном на фиг.16, как описано выше, будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.25. Когда начинается процесс получения типа первичного горизонтального преобразования, на этапе S221 блок 132 выбора обратного первичного преобразования определяет, является ли объектом обработки блок преобразования яркости. В случае, когда определяется, что объектом обработки является блок преобразования цветности, процесс переходит к этапу S222. На этапе S222 блок 132 выбора обратного первичного преобразования определяет, равна или меньше ширина картинка блока преобразования цветности (логарифмическое значение log2TBWSize ширины картинка) порогового значения TH. В случае, когда определяется, что ширина картинка блока преобразования цветности больше порогового значения (log2TBWSize >

ТН), процесс переходит к этапу S223. Кроме того, в случае, когда на этапе S221 определяется, что объектом обработки является блок преобразования яркости, процесс переходит к этапу S223.

На этапе S223 блок 132 выбора обратного первичного преобразования устанавливает  
 5 тип TrTypeIdxH преобразования (обратного) первичного горизонтального преобразования на основании таблицы соответствия наборов преобразований и типов (LUT\_TrSetToTrTypeIdx) преобразования, показанной на фиг.1, набора TrSetH первичных горизонтальных преобразований и флага pt\_hor\_flag точного определения первичного горизонтального преобразования. Когда процесс на этапе S223 заканчивается, процесс  
 10 получения типа первичного горизонтального преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.16.

Кроме того, в случае, когда на этапе S222 ширина картинки блока преобразования цветности определяется равной или меньшей порогового значения ( $\log_2 \text{TBWSize} \leq \text{TH}$ ), процесс переходит к этапу S224. На этапе S224 блок 132 выбора обратного  
 15 первичного преобразования устанавливает тип TrTypeIdxH преобразования (обратного) первичного горизонтального преобразования на заданное значение. Когда процесс на этапе S224 заканчивается, процесс получения типа первичного горизонтального преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.16.

То же самое относится и к вертикальному направлению. В случае, когда на этапе S145, показанном на фиг.16, высота картинки блока преобразования цветности (в данном случае логарифмическое значение высоты  $\log_2 \text{TBHSize}$  картинки) равна или меньше порогового значения ТН, как показано в следующей формуле (24), например, заданный тип (DCT-2) преобразования может быть установлен как тип TrTypeIdxV  
 25 преобразования (обратного) первичного вертикального преобразования, и в других случаях тип TrTypeIdxV преобразования (обратного) первичного вертикального преобразования может быть установлен на основании вертикального идентификатора TrSetV набора преобразований TrSetV и флага pt\_ver\_flag точного определения первичного вертикального преобразования. Следует отметить, что в формуле (24), в  
 30 конкретном примере пороговое значение ТН может быть установлено, например, на 1 или 2. Кроме того, в формуле (24) логарифмическое значение  $\log_2 \text{TBHSize}$  высоты картинки блока преобразования можно заменить на высоту картинки ТВН. В этом случае, пороговое значение ТН' устанавливается на  $1 \ll \text{TH}$ .

```

    if (compID==COMPONENT_Y ) {
  35   TrTypeIdxV = LUT_TrSetToTrTypeIdx[ TrSetV ][ pt_ver_flag ]
    } else if (log2TBHSize<=TH) {
      TrTypeIdxV = predetermined value (a value of the transform type indicating DCT-II is set)
    } else {
      TrTypeIdxV = LUT_TrSetToTrTypeIdx[ TrSetV ][ pt_ver_flag ]
  40   }
    ... (24)
  
```

Пример последовательности операций процесса получения типа первичного вертикального преобразования, выполняемого на этапе S145, показанном на фиг.16, который описан выше, будет описан со ссылкой на блок-схему последовательности  
 45 операций, показанную на фиг.26. Когда начинается процесс получения типа первичного вертикального преобразования, на этапе S231 блок 132 выбора обратного первичного преобразования определяет, является ли объектом обработки блок преобразования яркости. В случае, когда определяется, что объектом обработки является блок

преобразования цветности, процесс переходит к этапу S232. На этапе S232 блок 132 выбора обратного первичного преобразования определяет, равна или меньше высота картинки блока преобразования цветности (логарифмическое значение  $\log_2 TBHSize$  высоты картинки) порогового значения  $TH$ . В случае, когда определяется, что высота картинки блока преобразования цветности больше порогового значения ( $\log_2 TBHSize > TH$ ), процесс переходит к этапу S233. Кроме того, в случае, когда на этапе S231 определяется, что объектом обработки является блок преобразования яркости, процесс переходит к этапу S233.

На этапе S233 блок 132 выбора обратного первичного преобразования устанавливает тип  $TrTypeIdxV$  преобразования (обратного) первичного вертикального преобразования на основании таблицы соответствия наборов преобразований и типов ( $LUT\_TrSetToTrTypeIdx$ ) преобразования, показанной на фиг.1, набора  $TrSetV$  первичных вертикальных преобразований и флага  $pt\_ver\_flag$  точного определения первичного вертикального преобразования. Когда процесс на этапе S233 заканчивается, процесс получения типа первичного вертикального преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.16.

Кроме того, в случае, когда на этапе S232 ширина картинки блока преобразования цветности определяется равной или меньшей порогового значения ( $\log_2 TBHSize \leq TH$ ), процесс переходит к этапу S234. На этапе S234 блок 132 выбора обратного первичного преобразования устанавливает тип  $TrTypeIdxV$  преобразования (обратного) первичного вертикального преобразования на заданное значение. Когда процесс на этапе S234 заканчивается, процесс получения типа первичного вертикального преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.16.

В случае, когда ширина картинки (высота картинки) блока преобразования цветности меньше заданного значения (например, 4), существует небольшая разница в эффективности кодирования между  $DST-7/DST-1/DCT-5/DCT-8$  и  $DCT-2$  в качестве (обратного) первичного горизонтального (вертикального) преобразования. Поэтому путем выбора заданного ортогонального преобразования ( $DCT-2$ ) в качестве (обратного) первичного горизонтального (вертикального) преобразования в случае, когда ширина картинки (высота картинки) блока преобразования цветности меньше заданного значения (например, 4), масштаб схемы по отношению к ортогональному преобразованию, используемому в адаптивном первичном преобразовании для размера, меньше заданного значения, может быть уменьшен.

### 3. Второй вариант осуществления

$ts\_flag$

Следует отметить, что хотя в первом варианте осуществления примеры флага  $apt\_flag$  адаптивного первичного преобразования и идентификатора  $pt\_idx$  первичного преобразования были описаны в виде информации, касающейся (обратного) ортогонального преобразования, детали информации, касающейся (обратного) ортогонального преобразования, являются произвольными и не ограничиваются параметрами. Например, может быть включен флаг  $ts\_flag$  пропуска преобразования, указывающий на то, должен ли быть пропущен процесс (обратного) ортогонального преобразования, как показано в таблице на фиг.6. Например, получая каждый из флага  $ts\_flag[Cb]$  пропуска преобразования цветности ( $Cb$ ) и флага  $ts\_flag[Cr]$  пропуска преобразования цветности ( $Cr$ ) на основании флага  $ts\_flag$  пропуска преобразования  $[Y]$  яркости, можно также сдерживать увеличение издержек, связанных с объемом кода синтаксиса, сигнализация которого осуществляется цветностью ( $Cb/Cr$ ).

Описание флага будет дополнительно предоставлено со ссылкой на таблицу, показанную на фиг.27. В уровне техники флаг `ts_flag` пропуска преобразования цветности передается (сигнализируется) со стороны кодирования на сторону декодирования, как указано в строке под номером #0. Если флаги `ts_flag` пропуска преобразования не только яркости, но также и цветности истолкованы как кодированные, как описано выше, существует проблема, связанная с увеличением объема кода и снижением эффективности кодирования. В этом случае сторона кодирования также должна выполнить настройки и кодирование флага `ts_flag` пропуска преобразования для каждого из сигналов (`Y`, `Cb` и `Cr`), и, таким образом, существует также проблема, связанная с увеличением объема обработки. Кроме того, сторона декодирования должна декодировать флаг `ts_flag` пропуска преобразования для каждого из сигналов (`Y`, `Cb` и `Cr`) и, таким образом, существует проблема, связанная с увеличением объема обработки.

Таким образом, флаг `ts_flag` пропуска преобразования цветности получается в любом случае из флага `ts_flag` пропуска преобразования яркости, например, как указано в строке под номером #1. Например, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости. Таким образом, передача (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности может быть опущена, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, например, как указано в строке под номером #2, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение флага `ts_flag` пропуска преобразования яркости можно использовать только в случае режима внутриблочного предсказания, в котором его остаточные сигналы показывают аналогичные тенденции, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, например, как указано в строке под номером #3 или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение флага `ts_flag` пропуска преобразования яркости можно использовать только в случае режима предсказания, в котором его остаточные сигналы показывают аналогичные тенденции, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, например, как указано в строке под номером #4, или межблочное предсказание, а режимом предсказания является внутриблочное копирование, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение флага `ts_flag` пропуска преобразования яркости можно использовать только в случае режима предсказания, в котором его остаточные сигналы показывают аналогичные тенденции, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности, указывающий на то, должен ли быть получен флаг `ts_flag[compID]` пропуска преобразования цветности на основании флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, равен 1 (истина), например, как указано в строке под номером #5, значение флага `ts_flag` пропуска преобразования `[compID]` цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности равен 0 (ложь), флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных). Таким образом, можно однозначно управлять тем, следует ли использовать значение флага пропуска преобразования яркости, используя флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности. Поэтому можно легко установить значение флага пропуска преобразования яркости, используемого только в случае, когда может быть получен достаточно большой эффект, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Следует отметить, что каждый из вышеописанных случаев может быть произвольно объединен аналогичным образом к случаю, показанному на фиг.7. Таким образом, могут быть продемонстрированы эффекты, полученные в каждом из случаев. Кроме того, каждый из вышеописанных случаев может быть также объединен с другими случаями, которые не описаны выше.

#### Основной пример 1

Флаг `ts_flag` пропуска преобразования цветности может быть получен из флага `ts_flag` пропуска преобразования яркости в любом случае, как описано со ссылкой на строку под номером #1, показанную на фиг.27. Пример синтаксиса `residual_coding` для этого случая показан на фиг.28. В этом случае "`compID == COMPONENT_Y`" включен в условие сигнализации флага `ts_flag` пропуска преобразования, как показано на фиг.28. Другими словами, в случае цветности (`compID != COMPONENT_Y`) установка выполняется с использованием флага пропуска преобразования яркости без сигнализации флага `ts_flag` пропуска преобразования.

В таком случае, блок 111 декодирования устанавливает флаг `ts_flag[compID]` пропуска преобразования цветности путем выполнения процесса получения флага пропуска преобразования. Пример последовательности операций процесса получения флага

пропуска преобразования будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.29.

Когда начинается процесс получения флага пропуска преобразования, на этапе S241 блок 111 декодирования определяет, имеет ли флаг `ts_enabled_flag` разрешения флага пропуска преобразования значение "истина". В случае, когда определяется, что флаг `ts_enabled_flag` разрешения флага пропуска преобразования имеет значение "истина", процесс переходит к этапу S242. На этапе S242 блок 111 декодирования определяет, имеет ли флаг `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования значение "ложь". В случае, когда флаг `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования имеет значение "ложь", процесс переходит к этапу S243. На этапе S243 блок 111 декодирования определяет, равен или меньше размер блока преобразования, подлежащего обработке, максимального размера блока пропуска преобразования. В случае, когда определяется, что размер равен или меньше максимального размера блока пропуска преобразования, процесс переходит к этапу S244.

На этапе S244 блок 111 декодирования определяет, является ли объектом обработки цветность (`compID != COMPONENT_Y`). В случае, когда определяется, что объектом обработки является цветность (`compID != COMPONENT_Y`), процесс переходит к этапу S245. На этапе S245 блок 111 декодирования устанавливает значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости для значения флага `ts_flag[compID]` пропуска преобразования цветности. Когда процесс на этапе S245 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Кроме того, в случае, когда на этапе S244 определяется, что объектом обработки является яркость (`compID == COMPONENT_Y`), процесс переходит к этапу S246. Так как флаг `ts_flag[compID]` пропуска преобразования яркости сигнализируется, блок 111 декодирования декодирует кодированные данные и, таким образом, получает флаг `ts_flag[compID]` пропуска преобразования яркости на этапе S246. Когда процесс на этапе S246 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Следует отметить, что в случае, когда на этапе S241 определяется, что флаг `ts_enabled_flag` разрешения флага пропуска преобразования имеет значение "ложь", на этапе S242 определяется, что флаг `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования имеет значение "истина", или на этапе S243 определяется, что размер блока преобразования объекта обработки превышает максимальный размер блока пропуска преобразования, процесс получения флага пропуска преобразования заканчивается.

Таким образом, передача (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности может быть опущен, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно сдерживать увеличение нагрузки кодирования и декодирования.

#### Модифицированный пример 1

Как описано со ссылкой на строку под номером #2, показанную на фиг.27, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, флаг `ts_flag[compID]` пропуска преобразования цветности может быть

просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных).

Пример синтаксиса `residual_coding` в этом случае показан на фиг.30. В этом случае `"&& (CuPreMode == MODE_INTRA || CuPreMode == MODE_INTER && compID == COMPONENT_Y)"` включается в условие сигнализации флага `ts_flag` пропуска преобразования, как показано на фиг.30. То есть даже в том случае, если режимом предсказания является межблочное предсказание, или режимом предсказания является внутриблочное предсказание, флаг `ts_flag` пропуска преобразования сигнализируется в случае яркости. Другими словами, флаг `ts_flag` пропуска преобразования не сигнализируется только в том случае, когда режимом предсказания для цветности является внутриблочное предсказание. То есть в этом случае флаг `ts_flag` пропуска преобразования устанавливается с использованием флага `ts_flag` пропуска преобразования яркости.

Пример последовательности операций процесса получения флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.31. Каждый из процессов этапа S251 – этапа S254 выполняется аналогично каждому из процессов этапа S241 – этапа S244, показанных на фиг.29.

На этапе S255 блок 111 декодирования дополнительно определяет, является ли режимом предсказания внутриблочное предсказание. В случае, когда определяется, что режимом предсказания является внутриблочное предсказание (`CuPredMode == MODE_INTER`), процесс переходит к этапу S256. Кроме того, в случае, когда определяется, что режимом предсказания является межблочное предсказание (`CuPredMode == MODE_INTRA`), процесс переходит к этапу S257.

Процесс на этапе S256 выполняется аналогично процессу на этапе S245, показанному на фиг.29. То есть блок 111 декодирования устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости. Когда процесс на этапе S256 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Кроме того, процесс на этапе S257 выполняется аналогично процессу на этапе S246, показанному на фиг.29. То есть блок 111 декодирования декодирует кодированные данные и, таким образом, получает просигнализированный флаг `ts_flag[compID]` пропуска преобразования. Когда процесс на этапе S257 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить за счет использования значения флага `ts_flag` пропуска преобразования яркости только в случае режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

## Модифицированный пример 2

В случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, как описано со ссылкой на строку под номером #3, показанную на фиг.27 или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости в случае, когда типом предсказания блока кодирования

является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных).

- 5 Пример синтаксиса `residual_coding` в этом случае показан на фиг.32. Как показано на фиг.32, в этом случае информация "`&& (CuPreMode == MODE_INTRA && compID == COMPONENT_Y || CuPreMode == MODE_INTRA && compID != COMPONENT_Y && IntraPredModeY != IntraPredModeC CuPredMode == MODE_INTER && compID == COMPONENT_Y)`" включена в условие сигнализации флага `ts_flag` пропуска преобразования. То есть независимо от того, является ли флаг яркостью или цветностью, флаг `ts_flag` пропуска преобразования сигнализируется в случае, когда типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом. Другими словами, только в случае, когда
- 10 типом предсказания является внутриблочное предсказание для цветности или межблочное предсказание для цветности, а режимы предсказания для яркости и цветности совпадают друг с другом, флаг `ts_flag` пропуска преобразования не сигнализируется. То есть в этом случае флаг `ts_flag` пропуска преобразования устанавливается с использованием флага `ts_flag` пропуска преобразования яркости.

- Пример последовательности операций процесса получения флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.33. Каждый из процессов этапа S261 – этапа S265 выполняется аналогично каждому из процессов этапа S251 – этапа S255, показанных на фиг.31.

- В случае, когда на этапе S265 определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S266. На этапе S266 блок 111 декодирования определяет, совпадают ли режимы предсказания для яркости и цветности друг с другом. В случае, когда определяется, что режимы предсказания совпадают друг с другом (`IntraPredModeY == IntraPredModeC`), процесс переходит к этапу S267. Кроме того, в случае, когда определяется, что режимы предсказания не совпадают друг с
- 25 другом, процесс переходит к этапу S268.

- Процесс на этапе S267 выполняется аналогично процессу на этапе S256, показанному на фиг.31. То есть блок 111 декодирования устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости. Когда процесс на этапе S267 заканчивается, процесс получения флага пропуска преобразования заканчивается.

- Кроме того, процесс на этапе S268 выполняется аналогично процессу на этапе S257, показанному на фиг.31. То есть блок 111 декодирования декодирует кодированные данные и, таким образом, получает просигнализованный флаг `ts_flag[compID]` пропуска преобразования. Когда процесс на этапе S268 заканчивается, процесс получения флага пропуска преобразования заканчивается.

- Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить за счет использования значения флага `ts_flag` пропуска преобразования яркости только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом,
- 45 можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

Модифицированный пример 3

В случае, когда типом предсказания блока кодирования, к которому принадлежит

блок преобразования, подлежащий обработке, является внутриблочное предсказание, как описано со ссылкой на строку под номером #4, показанную на фиг.27, или межблочное предсказание, а режимом предсказания является внутриблочное копирование, значение флага `ts_flag[compID]` пропуска преобразования цветности может  
 5 быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности  
 10 может быть декодирован из кодированных данных).

Пример синтаксиса `residual_coding` в этом случае показан на фиг.34. Как показано на фиг.34, в этом случае информация "`&& (CuPreMode == MODE_INTRA && compID == COMPONENT_Y || CuPreMode == MODE_INTRA && compID != COMPONENT_Y && IntraPredModeY != IntraBC CuPredMode == MODE_INTER && compID == COMPONENT_Y)`"  
 15 включена в условие сигнализации флага `ts_flag` пропуска преобразования. То есть независимо от того, является ли флаг яркостью или цветностью, флаг `ts_flag` пропуска преобразования сигнализируется в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование. Другими словами, только в случае, когда типом предсказания является  
 20 внутриблочное предсказание для цветности или межблочное предсказание для цветности, а режимом предсказания является внутриблочное копирование, флаг `ts_flag` пропуска преобразования не сигнализируется. То есть флаг `ts_flag` пропуска преобразования устанавливается с использованием флага `ts_flag` пропуска преобразования яркости в этом случае.

Пример последовательности операций процесса получения флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.35. Каждый из процессов этапа S271 – этапа S275 выполняется аналогично каждому из процессов этапа S261 – этапа S265, показанных на фиг.33.

В случае, когда на этапе S275 определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S276. На этапе S276 блок 111 декодирования определяет, является ли режимом предсказания внутриблочное копирование. В случае, когда определяется, что режимом предсказания является внутриблочное копирование (`IntraPredModeY == IntraBC`), процесс переходит к этапу  
 35 S277. Кроме того, в случае, когда определяется, что режимом предсказания является внутриблочное копирование, процесс переходит к этапу S278.

Процесс на этапе S277 выполняется аналогично процессу на этапе S267, показанному на фиг.33. То есть блок 111 декодирования устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости. Когда процесс на этапе S277 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Кроме того, процесс на этапе S278 выполняется аналогично процессу на этапе S268, показанному на фиг.33. То есть блок 111 декодирования декодирует кодированные данные и, таким образом, получает просигнализированный флаг `ts_flag[compID]`  
 45 пропуска преобразования. Когда процесс на этапе S278 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить за счет использования значения флага

ts\_flag пропуска преобразования яркости только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### 5 Модифицированный пример 4

В случае, когда флаг chroma\_ts\_info\_infer\_flag вывода информации пропуска преобразования цветности равен 1 (истина), как описано со ссылкой на строку под номером #5, показанную на фиг.27, значение флага ts\_flag пропуска преобразования [compID] цветности может быть установлено равным значению флага ts\_flag [COMPONENT\_Y] пропуска преобразования яркости, и в случае, когда флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности равен 0 (ложь), флаг ts\_flag[compID] пропуска преобразования цветности может быть просигнализирован (то есть флаг ts\_flag[compID] пропуска преобразования цветности может быть декодирован из кодированных данных).

15 Пример синтаксиса единицы преобразования в этом случае показан на фиг.36. Как показано на фиг.36, флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности установлен. Кроме того, пример синтаксиса residual\_coding показан на фиг.37. Как показано на фиг.37, информация "&& compID != COMPONENT\_Y &&! chroma\_ts\_info\_infer\_flag" включена в условие сигнализации флага ts\_flag пропуска преобразования в этом случае. То есть независимо от того, является ли флаг яркостью или цветностью, флаг ts\_flag пропуска преобразования сигнализируется в случае, когда 20 флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности имеет значение "ложь". Другими словами, только в том случае, когда флаг имеет цветность, и флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности имеет значение "истина", флаг ts\_flag пропуска преобразования не сигнализируется. То есть флаг ts\_flag пропуска преобразования устанавливается с использованием флага ts\_flag пропуска преобразования яркости в этом случае.

Пример последовательности операций процесса получения флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему 30 последовательности операций, показанную на фиг.38. Каждый из процессов этапа S281 – этапа S283 выполняется аналогично каждому из процессов этапа S241 – этапа S245, показанных на фиг.29.

На этапе S284 блок 111 декодирования определяет, является ли объектом обработки 35 цветность, и флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности имеет значение "истина". В случае, когда определяется, что флаг определяет цветность и имеет значение "истина", процесс переходит к этапу S285. Кроме того, в случае, когда на этапе S284 определяется, что объектом обработки является яркость, или определяется, что флаг chroma\_ts\_info\_infer\_flag вывода информации о пропуске преобразования цветности имеет значение "ложь", процесс 40 переходит к этапу S286.

Процесс на этапе S285 выполняется аналогично процессу на этапе S245, показанному на фиг.29. То есть блок 111 декодирования устанавливает значение флага ts\_flag[compID] пропуска преобразования цветности в значение флага ts\_flag[COMPONENT\_Y] пропуска преобразования яркости. Когда процесс на этапе S285 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Кроме того, процесс на этапе S286 выполняется аналогично процессу на этапе S246, показанному на фиг.29. То есть блок 111 декодирования декодирует кодированные

данные и, таким образом, получает просигнализированный флаг `ts_flag[compID]` пропуска преобразования. Когда процесс на этапе S286 заканчивается, процесс получения флага пропуска преобразования заканчивается.

Таким образом, то, следует или нет использовать значение флага пропуска преобразования яркости, можно однозначно контролировать, используя флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности. Поэтому можно легко установить значение флага пропуска преобразования яркости, используемого только в случае, когда может быть получен достаточно большой эффект, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### 4. Третий вариант осуществления `st_idx`

Кроме того, в качестве информации, касающейся (обратного) ортогонального преобразования, например, может быть включен идентификатор `st_idx` вторичного преобразования, который должен применять такое (обратное) вторичное преобразование, как показано в таблице на фиг.6. Например, получая идентификатор `chroma_st_idx` вторичного преобразования цветности (общей для `Cb` и `Cr`) на основании идентификатора `st_idx [Y]` вторичного преобразования яркости, можно также сдерживать увеличение издержек, связанных с объемом кода синтаксиса, с помощью которого осуществляется сигнализация цветности (`Cb` или `Cr`).

Дальнейшее описание будет предоставлено со ссылкой на таблицу, показанную на фиг.39. В уровне техники идентификатор `st_idx` вторичного преобразования цветности также передается (сигнализируется) на сторону декодирования со стороны кодирования, как указано в строке под номером #0. На фиг.40 показан пример синтаксиса TU согласно уровню техники. Как указано в синтаксисе на фиг.40, вычисляется также сигнализация идентификатора `chroma_st_idx` вторичного преобразования цветности. Если идентификатор `st_idx` вторичного преобразования является явно кодированным, но только не для яркости, а только для цветности, существует проблема, связанная со снижением эффективности кодирования из-за увеличения объема кода. Кроме того, в этом случае сторона кодирования должна выполнить настройки и кодирование идентификатора `st_idx` вторичного преобразования для каждого из сигналов (`Y`, `Cb` и `Cr`), и, таким образом, существует проблема, связанная с увеличением объема обработки. Кроме того, сторона декодирования должна декодировать идентификатор `st_idx` вторичного преобразования для каждого из сигналов (`Y`, `Cb` и `Cr`), и, таким образом, существует проблема, связанная с увеличением объема обработки.

Таким образом, идентификатор `chroma_st_idx` вторичного преобразования цветности получается из идентификатора `st_idx` вторичного преобразования яркости в любом случае, например, как указано в строке под номером #1. Например, значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости. Таким образом, передачу (кодирование и декодирование) идентификатора `chroma_st_idx` вторичного преобразования цветности можно опустить, и, таким образом, можно уменьшить издержки, связанные с объемом кода, и можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное

предсказание, как указано в строке под номером #2, например, значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, и в случае, когда

5 идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение идентификатора `st_idx` вторичного преобразования яркости можно использовать только в случае режима внутриблочного предсказания, когда тенденции остаточных сигналов  
10 являются аналогичными, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, например, как указано в строке под номером #3, или межблочное  
15 предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с  
20 другом, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение идентификатора `st_idx` вторичного преобразования яркости можно использовать только в случае режима предсказания, в котором тенденции остаточных  
25 сигналов являются аналогичными, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, например, как указано в строке под номером #4, или межблочное  
30 предсказание, а режимом предсказания является внутриблочное копирование, значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование,  
35 идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных). Таким образом, значение идентификатора `st_idx` вторичного преобразования яркости можно использовать только в случае режима предсказания, в котором тенденции остаточных сигналов являются  
40 аналогичными, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Кроме того, например, в случае, когда флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности, указывающий на то, должен ли быть получен идентификатор `chroma_st_idx` вторичного преобразования цветности  
45 на основании идентификатора `st_idx` вторичного преобразования яркости, равен 1 (истина), значение идентификатора `st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, и в случае, когда флаг `chroma_st_idx_infer_flag` вывода идентификатора

вторичного преобразования цветности равен 0 (ложь), идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных), как указано в строке под номером #5. Таким образом, можно однозначно контролировать, следует или нет использовать значение идентификатора `st_idx` вторичного преобразования яркости, используя флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности. Поэтому можно легко установить значение флага вывода идентификатора `chroma_st_idx_infer_flag` вторичного преобразования цветности яркости, используемого только в случае, когда может быть получен достаточно большой эффект, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования.

Следует отметить, что каждый из вышеописанных случаев может быть произвольно объединен аналогичным образом со случаем, показанным на фиг.7. Таким образом, можно продемонстрировать эффекты, полученные в каждом из этих случаев. Кроме того, каждый из вышеописанных случаев может быть также объединен с другими случаями, которые не описаны выше.

#### Основной пример 1

Идентификатор `chroma_st_idx` вторичного преобразования цветности может быть получен в любом случае из идентификатора `st_idx` вторичного преобразования яркости, как описано со ссылкой на строку под номером #1, показанную на фиг.39. Пример синтаксиса TU в этом случае показан на фиг.41. В этом случае установка выполняется с использованием идентификатора `st_idx` вторичного преобразования яркости, без сигнализации идентификаторе `chroma_st_idx` вторичного преобразования цветности, как показано на фиг.41.

В таком случае блок 111 декодирования выполняет процесс получения идентификатора вторичного преобразования для цветности и, таким образом, устанавливает идентификатор `chroma_st_idx` вторичного преобразования цветности. Пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.42.

Когда начинается процесс получения идентификатора вторичного преобразования для цветности, на этапе S291 блок 111 декодирования определяет, имеет или нет флаг `st_enabled_flag` вторичного преобразования значение "истина". В случае, когда определяется, что флаг `st_enabled_flag` разрешения вторичного преобразования имеет значение "истина", процесс переходит к этапу S292. На этапе S292 блок 111 декодирования определяет, имеет или нет флаг `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования значение "ложь". В случае, когда определяется, что флаг `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования имеет значение "ложь", процесс переходит к этапу S293. На этапе S293 блок 111 декодирования определяет, больше или равно количество ненулевых коэффициентов (`numNonZeroCoeffTH`) цветности заданному пороговому значению (`stNumZeroCoeffTH`). В случае, когда определяется, что количество ненулевых коэффициентов цветности больше или равно заданному пороговому значению, процесс переходит к этапу S294.

На этапе S294 блок 111 декодирования устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx` вторичного преобразования яркости. Когда процесс на этапе S294 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Следует отметить, что в случае, когда на этапе S291 определяется, что флаг `st_enabled_flag` разрешения вторичного преобразования имеет значение "ложь", процесс получения идентификатора вторичного преобразования для цветности заканчивается. Кроме того, в случае, когда на этапе S292 определяется, что флаг

5 `transquant_bypass_enabled_flag` разрешения обхода квантования преобразования имеет значение "истина", процесс получения идентификатора вторичного преобразования для цветности заканчивается. Кроме того, в случае, когда на этапе S293 определяется, что количество ненулевых коэффициентов цветности меньше заданного порогового значения, процесс получения идентификатора вторичного преобразования для цветности  
10 заканчивается.

Таким образом, передачу (кодирование и декодирование) идентификатора `chroma_st_idx` вторичного преобразования цветности можно опустить, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### 15 Модифицированный пример 1

В случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования  
20 яркости, как описано со ссылкой на строку под номером #2, показанную на фиг.39, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных).

25 Пример синтаксиса TU в этом случае показан на фиг.43. Как показано на фиг.43, в случае, когда режимом предсказания является межблочное предсказание (`CuPreMode == MODE_INTRA`), то сигнализируется идентификатор `chroma_st_idx` вторичного преобразования цветности. Кроме того, в случае, когда режимом предсказания является межблочное предсказание (`CuPreMode == MODE_INTER`), идентификатор `chroma_st_idx`  
30 вторичного преобразования цветности устанавливается с использованием идентификатора `st_idx` вторичного преобразования яркости.

Пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.44. Каждый из  
35 процессов этапа S301 – этапа S303 выполняется аналогично каждому из процессов этапа S291 – этапа S293, показанных на фиг.42.

На этапе S304 блок 111 декодирования дополнительно определяет, является ли режимом предсказания межблочное предсказание. В случае, когда режимом предсказания является внутриблочное предсказание (`CuPredMode == MODE_INTER`),  
40 процесс переходит к этапу S305.

Процесс на этапе S305 выполняется аналогично процессу на этапе S294, показанному на фиг.42. То есть блок 111 декодирования устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx` вторичного преобразования яркости. Когда процесс на этапе S305 заканчивается,  
45 процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Кроме того, в случае, когда определяется, что режимом предсказания является межблочное предсказание (`CuPredMode == MODE_INTRA`), процесс переходит к этапу

S306. На этапе S306 блок 111 декодирования декодирует кодированные данные и, таким образом, получает просигнализованный идентификатор `st_idx` вторичного преобразования. Когда процесс на этапе S306 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

5 Таким образом, передачу (кодирование и декодирование) идентификатора `chroma_st_idx` вторичного преобразования цветности можно опустить, используя значение идентификатора `st_idx` вторичного преобразования яркости только в случае режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

Модифицированный пример 2

В случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание  
15 или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, как описано со ссылкой на строку под номером #3, показанную на фиг.39, блок 111 декодирования устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx` вторичного преобразования яркости. Кроме того, в случае, когда типом предсказания  
20 блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных).

25 Пример синтаксиса TU в этом случае показан на фиг.45. Как показано на фиг.45, в случае, когда режим предсказания является режимом межблочного предсказания (`CuPredMode == MODE_INTRA`), и режимы предсказания для яркости и цветности не совпадают друг с другом, осуществляется сигнализация идентификатора `chroma_st_idx` вторичного преобразования. В других случаях, значение идентификатора `chroma_st_idx`  
30 вторичного преобразования цветности устанавливается на значение идентификатора `st_idx` вторичного преобразования яркости.

Пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.46. Каждый из  
35 процессов этапа S311 – этапа S314 выполняется аналогично каждому из процессов этапа S301 – этапа S304, показанных на фиг.44.

В случае, когда на этапе S314 определяется, что режимом предсказания является межблочное предсказание, процесс переходит к этапу S315. Кроме того, в случае, когда определяется, что режимом предсказания является внутриблочное предсказание, процесс  
40 переходит к этапу S316. На этапе S315 блок 111 декодирования определяет, совпадают ли режимы предсказания для яркости и цветности друг с другом. В случае, когда режимы предсказания совпадают друг с другом (`IntraPredModeY == IntraPredModeC`), процесс переходит к этапу S316. Кроме того, в случае, когда режимы предсказания не совпадают друг с другом (`IntraPredModeY != IntraPredModeC`), процесс переходит к этапу S317.

45 Процесс на этапе S316 выполняется аналогично процессу на этапе S305, показанному на фиг.44. Кроме того, процесс на этапе S317 выполняется аналогично процессу на этапе S306, показанному на фиг.44.

Таким образом, передачу (кодирование и декодирование) идентификатора

chroma\_st\_idx вторичного преобразования цветности можно опустить, используя значение идентификатора st\_idx вторичного преобразования яркости только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

### Модифицированный пример 3

В случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, как описано со ссылкой на строку под номером #4, показанную на фиг.39, значение идентификатора chroma\_st\_idx вторичного преобразования цветности устанавливается в значение идентификатора st\_idx вторичного преобразования яркости. Кроме того, в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, идентификатор chroma\_st\_idx вторичного преобразования цветности может быть просигнализирован (то есть идентификатор chroma\_st\_idx вторичного преобразования цветности может быть декодирован из кодированных данных).

Пример синтаксиса TU в этом случае показан на фиг.47. Как показано на фиг.47, флаг ts\_flag пропуска преобразования и идентификатор st\_idx вторичного преобразования сигнализируются в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование (CuPredMode == MODE\_INTRA && IntraPredModeY != IntraBC), независимо от того, представляет ли собой идентификатор яркость или цветность. Другими словами, сигнализация идентификатора st\_idx вторичного преобразования не осуществляется только в случае, когда идентификатор имеет цветность, и типом предсказания является внутриблочное предсказание, или в случае, когда идентификатор имеет цветность, и типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование. То есть в этом случае идентификатор chroma\_st\_idx вторичного преобразования устанавливается с использованием идентификатора st\_idx вторичного преобразования яркости.

Пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.48. Каждый из процессов этапа S321 – этапа S324 выполняется аналогично каждому из процессов этапа S311 – этапа S314, показанных на фиг.46.

В случае, когда на этапе S324 определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S325. Кроме того, в случае, когда определяется, что типом предсказания является внутриблочное предсказание, процесс переходит к этапу S326.

На этапе S325 блок 111 декодирования определяет, является ли режимом предсказания внутриблочное копирование. В случае, когда определяется, что режимом предсказания является внутриблочное копирование (IntraPredModeY == IntraBC), процесс переходит к этапу S326. Кроме того, в случае, когда определяется, что режимом предсказания является внутриблочное копирование, процесс переходит к этапу S327.

Процесс на этапе S326 выполняется аналогично процессу на этапе S316, показанному на фиг.46. То есть блок 111 декодирования устанавливает значение идентификатора chroma\_st\_idx вторичного преобразования цветности в значение идентификатора st\_idx

вторичного преобразования яркости. Когда процесс на этапе S326 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Кроме того, процесс на этапе S327 выполняется аналогично процессу на этапе S317, показанному на фиг.46. То есть блок 111 декодирования может декодировать кодированные данные и, таким образом, получает просигнализированный идентификатор `st_idx` вторичного преобразования. Когда процесс на этапе S327 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Таким образом, передачу (кодирование и декодирование) идентификатора `st_idx` вторичного преобразования цветности можно опустить, используя значение идентификатора `st_idx` вторичного преобразования яркости только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 4

Как описано со ссылкой на строку под номером #5, показанную на фиг.39, в случае, когда флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности равен 1 (истина), значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, и в случае, когда флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности равен 0 (ложь), идентификатор `st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных).

Пример синтаксиса TU в этом случае показан на фиг.49. Как показано на фиг.49, сначала устанавливается флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности, и затем осуществляется сигнализация идентификатора `chroma_st_idx` вторичного преобразования цветности в соответствии со значением или осуществляется управление идентификатором, который получается из идентификатора `st_idx` вторичного преобразования яркости.

Пример последовательности операций процесса получения идентификатора вторичного преобразования для цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.50. Каждый из процессов этапа S331 – этапа S333 выполняется аналогично каждому из процессов этапа S321 – этапа S323, показанных на фиг.48.

На этапе S334 блок 111 декодирования декодирует и получает флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности. На этапе S335 блок 111 декодирования определяет, имеет ли полученный флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности значение "истина". В случае, когда определяется, что флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности имеет значение "истина", процесс переходит к этапу S336. Кроме того, когда определяется, что флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности имеет значение "ложь", процесс переходит к этапу S337.

Процесс на этапе S336 выполняется аналогично процессу на этапе S326, показанному на фиг.48. То есть блок 111 декодирования устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx`

вторичного преобразования яркости. Когда процесс на этапе S336 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Кроме того, процесс на этапе S337 выполняется аналогично процессу на этапе S327, показанному на фиг.48. То есть блок 111 декодирования может декодировать кодированные данные и, таким образом, получает просигнализированный идентификатор `chroma_st_idx` вторичного преобразования цветности. Когда процесс на этапе S337 заканчивается, процесс получения идентификатора вторичного преобразования для цветности заканчивается.

Таким образом, можно однозначно управлять тем, следует ли использовать значение идентификатора `st_idx` вторичного преобразования яркости, используя флаг `chroma_st_idx_infer_flag` вывода идентификатора вторичного преобразования цветности. Поэтому можно легко установить значение идентификатора вторичного преобразования яркости, используемой только в случае, когда достигается достаточный большой эффект, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### 5. Четвертый вариант осуществления

##### Устройство кодирования изображения

Далее будет описано кодирование для выработки кодированных данных, которые будут декодироваться так, как описано выше. На фиг.51 показана блок-схема, иллюстрирующая пример основной конфигурации устройства кодирования изображения, которая является аспектом устройства обработки изображений, к которому применима настоящая технология. Устройство 400 кодирования изображения, показанное на фиг.51, представляет собой устройство кодирования изображения, соответствующее устройству 100 декодирования изображения, показанному на фиг.9, и вырабатывает кодированные данные (битовый поток), которые были декодированы устройством 100 декодирования изображения путем кодирования изображения с использованием способа кодирования, соответствующего способу декодирования, который используется устройством 100 декодирования изображения. Устройство 400 кодирования изображения реализует, например, технологию, предложенную HEVC или JVET.

Следует отметить, что на фиг.51 показаны основные конфигурации, такие как последовательности операций обработки блоков и данных и т.п., и на фиг.51 не показаны все конфигурации. То есть в устройстве 400 кодирования изображения могут быть блоки обработки, которые не показаны как блоки на фиг.51, или последовательности операций процессов и данных, которые не указаны стрелками и т.п. на фиг.51.

Как показано на фиг.51, устройство 400 кодирования изображения имеет блок 411 управления, блок 412 арифметических операций, блок 413 преобразования, блок 414 квантования, блок 415 кодирования, блок 416 обратного квантования, блок 417 обратного преобразования, блок 418 арифметических операций, память 419 кадров и блок 420 предсказания.

Блок 411 управления разбивает движущееся изображение #2 на блоки, которые являются блоками обработки (CU, PU, блоки преобразования и т.д.) на основании размера блока единиц обработки точно определенных извне или заранее, и выводит изображение I, соответствующее разделенным блокам, в блок 412 арифметических операций. Кроме того, блок 411 управления определяет параметры кодирования (информацию `Hinfo` заголовка, информация `PInfo` о режиме предсказания, информация `Tinfo` о преобразовании и т.д.), которые будут подаваться в каждый из блоков, например,

на основе оптимизации по скорости-искажению (RDO). Определенные параметры кодирования подаются в каждый из блоков. Детали параметров представлены следующим образом.

Информация  $P_{info}$  заголовка подается в каждый блок. Информация  $P_{info}$  о режиме предсказания подается в блок 413 преобразования, блок 415 кодирования и блок 420 предсказания. Информация  $T_{info}$  о преобразовании подается в блок 415 кодирования, блок 413 преобразования, блок 414 квантования, блок 416 обратного квантования и блок 417 обратного преобразования.

Блок 412 арифметических операций принимает ввод изображения  $I$ , соответствующего блокам, которые являются блоками обработки, и предсказанного изображения  $P$ , поданного из блока 420 предсказания, вычитает предсказанное изображение  $P$  из изображения  $I$ , которое указано в формуле (25), таким образом, получает предсказанный остаток  $D$  и подает предсказанный остаток в блок 413 преобразования.

$$D = I - P \dots (25)$$

Блок 413 преобразования является блоком обработки, который выполняет обратное преобразование, которое является обратным процессом по отношению к процессу преобразования, выполняемому блоком 417 обратного преобразования, принимает ввод предсказанного остатка  $D$ , информации  $P_{info}$  о режиме предсказания и информации  $T_{info}$  о преобразовании, применяет преобразование к предсказанному остатку  $D$  на основании информации  $P_{info}$  о режиме предсказания и информации  $T_{info}$  о преобразовании, в результате чего получается коэффициент  $Coeff$  преобразования, и подает коэффициент преобразования в блок 414 квантования.

Блок 414 квантования выполняет обратный процесс по отношению к блоку 416 обратного квантования, принимает ввод информации  $T_{info}$  о преобразовании и коэффициента  $Coeff$  преобразования, масштабирует (квантует) коэффициент  $Coeff$  преобразования на основании информации  $T_{info}$  о преобразовании и подает квантованный коэффициент преобразования, то есть уровень квантованного уровня коэффициента преобразования, в блок 415 кодирования.

Блок 415 кодирования выполняет обратный процесс по отношению блоку 111 декодирования (фиг.9), преобразует параметры кодирования (информацию заголовка, информацию  $P_{info}$  о режиме предсказания и информацию  $T_{info}$  о преобразовании), подаваемые из блока 411 управления, и уровень квантованного уровня коэффициента преобразования, поданный из блока 414 квантования, в синтаксические значения каждого синтаксического элемента в соответствии с определениями синтаксической таблицы, преобразует каждое из значений синтаксиса в коды переменной длины (например, арифметический код) и, таким образом, вырабатывает битовые строки.

Следует отметить, что блок 415 кодирования извлекает остаточную информацию  $R_{info}$  из уровня квантованного уровня коэффициента преобразования, преобразует остаточную информацию  $R_{info}$  в коды переменной длины и вырабатывает битовые строки. Кроме того, блок 415 кодирования мультиплексирует битовые строки каждого синтаксического элемента, которые были преобразованы в коды переменной длины, и затем вырабатывает и выводит кодированные данные #1.

Блок 416 обратного квантования представляет собой блок обработки, аналогичный блоку 112 обратного квантования (фиг.9), и выполняет процесс, аналогичный процессу, выполняемому блоком 112 обратного квантования в устройстве 400 кодирования изображения. Блок 417 обратного преобразования представляет собой блок обработки, аналогичный блоку 113 обратного преобразования (фиг.9), и выполняет процесс, аналогичный процессу, выполняемому блоком 113 обратного преобразования в

устройстве 400 кодирования изображения. Память 419 кадров представляет собой блок обработки, аналогичный памяти 115 кадров (фиг.9), и выполняет процесс, аналогичный процессу, выполняемому памятью 115 кадров в устройстве 400 кодирования изображения. Блок 420 предсказания представляет собой блок обработки, аналогичный блоку 116 предсказания (фиг.9), и выполняет процесс, аналогичный процессу, выполняемому блоком 116 предсказания в устройстве 400 кодирования изображения.

#### Блок преобразования

На фиг.52 показана блок-схема, иллюстрирующая пример основной конфигурации блока 413 преобразования. Как показано на фиг.52, блок 413 преобразования имеет переключатель 431, блок 432 первичного преобразования и блок 433 вторичного преобразования.

Переключатель 431 принимает ввод предсказанного остатка D и флага `ts_flag[compID]` пропуска преобразования цветового сигнала, соответствующего идентификатору `compID` цветового сигнала, и выводит предсказанный остаток D в блок 432 первичного преобразования в случае, когда, значение флага `ts_flag[compID]` пропуска преобразования представляет собой `NO_TS (=0)` (случай, когда пропуск преобразования не применяется). Кроме того, в случае, когда значение флага `ts_flag[compID]` пропуска преобразования `2D_TS (=1)` (случай, когда флаг указывает, что применяется пропуск двумерного преобразования), блок 432 первичного преобразования и блок 433 вторичного преобразования пропускаются, и предсказанный остаток D выводится в виде коэффициента `Coeff` преобразования.

Блок 432 первичного преобразования выполняет процесс, относящийся к первичному преобразованию, которое представляет собой заданный процесс преобразования, например, ортогональное преобразование. Первичное преобразование является обратным процессом обратного первичного преобразования, выполняемым блоком 123 обратного первичного преобразования и т.п. устройства 100 декодирования изображения. Блок 432 первичного преобразования принимает вводы, например, идентификатора `compID` цветового сигнала, флага `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости, идентификатора `pt_idx[COMPONENT_Y]` первичного преобразования яркости, информации `PInfo` о режиме предсказания, размеры блока преобразования (логарифмическое значение `log2TBWSize` ширины картинки и логарифмического значения `log2TBHSize` высоты картинки) и предсказанного остатка D. Блок 432 первичного преобразования выбирает тип `TrTypeIdxH` преобразования первичного горизонтального преобразования и тип `TrTypeIdxV` преобразования первичного вертикального преобразования цветового сигнала, точно определенного идентификатором `compID` цветового сигнала, со ссылкой на информацию `PInfo` о режиме предсказания, идентификатор `compID` цветового сигнала, флаг `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости, идентификатор `pt_idx[COMPONENT_Y]` первичного преобразования яркости и идентификатор `pt_idx[COMPONENT_Y]` первичного преобразования яркости, выполняет первичное горизонтальное преобразование, определенное типом `TrTypeIdxH` первичного горизонтального преобразования и шириной `log2TBWSize` картинки блока преобразования, и первичное вертикальное преобразование, определенное типом `TrTypeIdxV` первичного вертикального преобразования и высотой `log2TBHSize` картинки блока преобразования, над предсказанным остатком D и, таким образом, получает и выводит коэффициент `Coeff_P` преобразования, который был подвергнут первичному преобразованию (который также упоминается как коэффициент первичного преобразования)

Ниже будет описан более подробно блок 432 первичного преобразования. Как показано на фиг.52, блок 432 первичного преобразования имеет блок 441 получения информации об адаптивном первичном преобразовании цветности, блок 442 выбора первичного преобразования, блок 443 первичного горизонтального преобразования и блок 444 первичного вертикального преобразования.

Блок 441 получения информации об адаптивном первичном преобразовании цветности получает информацию об адаптивном первичном преобразовании цветности (apt\_flag [compID] и pt\_idx[compID] (compID=COMPONENT\_Cb или COMPONENT\_Cr)), используя информацию (apt\_flag[COMPONENT\_Y], pt\_idx[COMPONENT\_Y]) об адаптивном первичном преобразовании для блока преобразования яркости, соответствующей блоку преобразования цветности в случае, когда идентификатор compID цветового сигнала указывает цветность (compID == COMPONENT\_Cb или COMPONENT\_Cr), и подает результат в блок 442 выбора первичного преобразования.

Блок получения информации об адаптивном первичном преобразовании цветности На фиг.53 показана функциональная блок-схема, иллюстрирующая пример основной конфигурации блока 441 получения информации об адаптивном первичном преобразовании цветности. Как показано на фиг.53, блок 441 получения информации об адаптивном первичном преобразовании цветности имеет блок 451 получения apt\_flag и блок 452 получения pt\_idx. Блок 451 получения apt\_flag выполняет процесс, относящийся к получению флага apt\_flag адаптивного первичного преобразования цветности. Кроме того, блок 452 получения pt\_idx выполняет процесс, относящийся к получению идентификатора pt\_idx первичного преобразования цветности. То есть каждый блок 451 получения apt\_flag и блок 452 получения pt\_idx имеет аналогичную конфигурацию и выполняет процессы, аналогичные тем, которые выполняются в блоке 151 получения apt\_flag и блоке 152 получения pt\_idx (фиг.11). То есть блок 441 получения информации об адаптивном первичном преобразовании цветности имеет аналогичную конфигурацию и выполняет процессы, аналогичные тем, которые выполняются в блоке 131 получения информации об адаптивном первичном преобразовании цветности. Поэтому подробное описание операций, выполняемых в блоке 441 получения информации об адаптивном первичном преобразовании цветности (блоке 151 получения apt\_flag и блоке 152 получения pt\_idx), аналогично подробному описанию операций, выполняемых блоком 131 получения информации об адаптивном первичном преобразовании цветности в первом варианте осуществления, и, таким образом, оно будет опущено.

Блок 441 получения информации об адаптивном первичном преобразовании цветности может получить информацию об адаптивном первичном преобразовании блока преобразования цветности на основании информации об адаптивном первичном преобразовании яркости, соответствующей блоку преобразования цветности. Поэтому объем обработки кодирования можно уменьшить за счет отсутствия кодирования информации об адаптивном первичном преобразовании блока преобразования цветности.

Блок 442 выбора первичного преобразования принимает ввод информации PInfo о режиме предсказания, идентификатора compID цветового сигнала, флага apt\_flag[compID] адаптивного первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала, и идентификатора pt\_idx[compID] первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала. Блок 442 выбора первичного преобразования получает тип TrTypeIdxH преобразования первичного горизонтального преобразования и тип

TrTypeIdxV преобразования первичного вертикального преобразования цветового сигнала, точно определенного идентификатором compID цветового сигнала со ссылкой на информацию PInfo о режиме предсказания, идентификатор compID цветового сигнала, флаг apt\_flag[compID] адаптивного первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала, и идентификатор pt\_idx [compID] первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала. Блок 442 выбора первичного преобразования подает полученный тип TrTypeIdxH преобразования первичного горизонтального преобразования в блок 443 первичного горизонтального преобразования. Кроме того, блок 442 выбора первичного преобразования подает полученный тип TrTypeIdxV преобразования первичного вертикального преобразования в блок 444 первичного вертикального преобразования.

Следует отметить, что операции, выполняемые блоком 442 выбора первичного преобразования, в основном аналогичны операциям, выполняемым блоком 132 выбора обратного первичного преобразования устройства 100 декодирования изображения. Таким образом, описание по отношению к блоку 132 выбора обратного первичного преобразования в первом варианте осуществления может представлять собой описание по отношению к блоку 442 выбора первичного преобразования, но при замене обратного первичного горизонтального преобразования IPThor на первичное горизонтальное преобразование PThor, замене обратного первичного вертикального преобразования IPTver на первичное вертикальное преобразование PTver и замене обратного первичного преобразования на первичное преобразование.

Блок 443 первичного горизонтального преобразования принимает ввод предсказанного остатка D, типа TrTypeIdxH преобразования первичного горизонтального преобразования и значение log2TBWSize, указывающее ширину картинки блока преобразования (логарифмическое значение ширины картинки) для каждого блока преобразования каждого цветового сигнала. Блок 443 первичного горизонтального преобразования выполняет первичное горизонтальное преобразование Phor, определенное типом TrTypeIdxH преобразования и шириной картинки блока преобразования, над предсказанным остатком D и выводит результат в виде коэффициента Coeff\_Phор преобразования, который был подвергнут первичному горизонтальному преобразованию.

Блок 444 первичного вертикального преобразования принимает ввод коэффициента Coeff\_Phор преобразования, который был подвергнут первичному горизонтальному преобразованию, типа TrTypeIdxV преобразования первичного вертикального преобразования и значения log2TBHSize, указывающего высоту картинки блока преобразования (логарифмическое значение высоты картинки) для каждого блока преобразования каждого цветового сигнала. Первичное вертикальное преобразование Pver, определенное типом TrTypeIdxV преобразования и высотой картинки блока преобразования, выполняется над коэффициентом Coeff\_Phор преобразования, который был подвергнут первичному горизонтальному преобразованию, и результат выводится в виде коэффициента Coeff\_P преобразования, который был подвергнут первичному преобразованию.

Как описано выше, блок 432 первичного преобразования может применить адаптивное первичное преобразование, выбранное для яркости, к блоку преобразования цветности в случае, когда остаточный сигнал цветового сигнала показывает тенденцию, аналогичную остаточному сигналу яркости. Таким образом, можно выполнять процесс первичного преобразования над остаточным сигналом цветности с более высокой

эффективностью кодирования по сравнению с уровнем техники.

Кроме того, объем обработки кодера можно уменьшить, не допуская снижения эффективности кодирования для цветности по сравнению со случаем, когда флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования явно декодируются для каждой из яркости и цветности.

Блок 433 вторичного преобразования выполняет процесс, относящийся к вторичному преобразованию, которое является заданным процессом преобразования, например, ортогональным преобразованием. Вторичное преобразование представляет собой обратный процесс по отношению к обратному вторичному преобразованию, выполняемому блоком 122 обратного вторичного преобразования и т.п. устройства 100 декодирования изображения. Блок 433 вторичного преобразования принимает ввод, например, идентификатора `st_idx` вторичного преобразования, идентификатора `scanIdx` сканирования, указывающего способ сканирования коэффициента преобразования, и коэффициента первичного преобразования `Coeff_P`, и получает и выводит коэффициент `Coeff` преобразования, который был подвергнут вторичному преобразованию (который также упоминается как вторичный коэффициент преобразования). Более конкретно, в случае, когда идентификатор `st_idx` вторичного преобразования указывает применение вторичного преобразования (`st_idx > 0`), блок 433 вторичного преобразования выполняет процесс вторичного преобразования, соответствующего идентификатору `st_idx` вторичного преобразования, над коэффициентом первичного преобразования `Coeff_P`, и выводит коэффициент преобразования `Coeff_S`, который был подвергнут вторичному преобразованию.

В случае, когда идентификатор `st_idx` вторичного преобразования указывает на неприменение вторичного преобразования (`st_idx == 0`), блок 433 вторичного преобразования пропускает вторичное преобразование и выводит коэффициент `Coeff_P` преобразования, который был подвергнут первичному преобразованию, в виде коэффициента преобразования `Coeff_S`, который был подвергнут вторичному преобразованию.

Последовательность операций процесса кодирования изображения

Далее будут описаны последовательности операций каждого из процессов, выполняемых устройством 400 кодирования изображения, как описано выше. Сначала будет описан пример последовательности операций процесса кодирования изображения со ссылкой на блок-схему последовательности операций, показанную на фиг.54.

Когда начинается процесс кодирования изображения, на этапе S401 блок 411

управления выполняет процесс управления кодированием и выполняет разбиение блоков, установку параметров кодирования и т.п.

На этапе S402 блок 420 предсказания выполняет процесс предсказания и вырабатывает предсказанное изображение и т.п. в оптимальном режиме предсказания. Например, в процессе предсказания блок 420 предсказания выполняет межблочное предсказание и вырабатывает предсказанное изображение и т.п. в оптимальном режиме межблочного предсказания, и выполняет внутриблочное предсказание и вырабатывает предсказанное изображение и т.п. в оптимальном режиме внутриблочного предсказания и выбирает оптимальный режим предсказания среди режимов на основе значения функции издержек и т.п.

На этапе S403 блок 412 арифметических операций вычисляет различия между входным изображением и предсказанным изображением в оптимальном режиме, выбранном в процессе предсказания на этапе S402. То есть блок 412 арифметических операций вырабатывает предсказанный остаток `D` между входным изображением и предсказанным

изображением. Объем данных предсказанного остатка D, полученного так, как описано выше, уменьшается в большей степени, чем объем данных первоначального изображения. Поэтому объем данных может быть более сжатым, чем в случае, когда изображение кодируется без изменений.

5 На этапе S404 блок 413 преобразования выполняет процесс преобразования в отношении предсказанного остатка D, выработанного в процессе на этапе S403, и получает коэффициент Coeff преобразования. Следует отметить, что процесс преобразования представляет собой обратный процесс по отношению к процессу обратного преобразования, выполняемому на этапе S407, и обратный процесс по  
10 отношению к процессу обратного преобразования, выполняемому в вышеописанном процессе декодирования изображения. Детали процесса на этапе S404 будут описаны ниже.

На этапе S405 блок 414 квантования выполняет квантование коэффициента Coeff преобразования, полученного в процесс на этапе S404 с использованием параметра  
15 квантования, вычисленного блоком 411 управления или тому подобного, и получает уровень квантованного по уровню коэффициента преобразования.

На этапе S406 блок 416 обратного квантования выполняет обратное квантование уровня квантованного по уровню коэффициента преобразования, выработанного в процессе на этапе S405, используя характеристики, соответствующие характеристикам  
20 квантования на этапе S405, и получает коэффициент Coeff\_IQ преобразования.

На этапе S407 блок 417 обратного преобразования выполняет обратное преобразование коэффициента Coeff\_IQ преобразования, полученного в процессе на этапе S406 с использованием способа, соответствующего процессу преобразования на этапе S404, и получает предсказанный остаток D'. Следует отметить, что процесс  
25 обратного преобразования выполняется аналогично процессу обратного преобразования, который выполняется в вышеописанном процессе декодирования изображения.

На этапе S408 блок 418 арифметических операций вырабатывает декодированное изображение, которое было локально декодировано путем добавления предсказанного  
30 изображения, полученного в процессе предсказания на этапе S402, к предсказанному остатку D', полученному из процесса на этапе S407.

На этапе S409 память 419 кадров сохраняет декодированное изображение, которое было локально декодировано и получено в процессе на этапе S408.

На этапе S410 блок 415 кодирования кодирует уровень квантованного по уровню  
35 коэффициента преобразования, полученный в процесс на этапе S405. Например, блок 415 кодирования кодирует уровень квантованного по уровню коэффициента преобразования, который представляет собой информацию относительно изображения, с использованием арифметического кодирования и т.п. и вырабатывает кодированные данные. Кроме того, в это время блок 415 кодирования кодирует различные параметры  
40 кодирования (информацию Hinfo заголовка, информацию PInfo о режиме предсказания и информацию Tinfo о преобразовании). Кроме того, блок 415 кодирования извлекает остаточную информацию RInfo из уровня квантованного по уровню коэффициента преобразования и кодирует остаточную информацию RInfo.

На этапе S411 блок 415 кодирования помещает кодированные данные различных  
45 видов информации, выработанной как описано выше, и выводит данные в виде битового потока за пределы устройства 400 кодирования изображения. Битовый поток передается на сторону декодирования, например, через канал передачи или носитель информации. Когда процесс на этапе S411 заканчивается, процесс кодирования изображения

заканчивается.

Последовательность операций процесса преобразования

Далее будет описан пример последовательности операций процесса преобразования, выполняемого на этапе S404, показанном на фиг.54, со ссылкой на блок-схему

5 последовательности операций, показанную на фиг.55. Когда начинается процесс преобразования, на этапе S421 переключатель 431 определяет, равен или нет флаг `ts_flag` пропуска преобразования `2D_TS` (случай, когда флаг указывает пропуск двумерного преобразования) или флаг `transquant_bypass_flag` обхода квантования преобразования 1 (истина). В случае, когда определяется, что флаг `ts_flag` пропуска преобразования  
10 равен `2D_TS`, или флаг обхода квантования преобразования равен 1 (истина), процесс преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.54. В этом случае переключатель 431 исключает процесс ортогонального преобразования (первичного преобразования или вторичного преобразования) и выводит входной предсказанный остаток `D` за пределы блока 413 преобразования  
15 (подает его в блок 414 квантования) в качестве коэффициента `Coeff` преобразования.

Кроме того, в случае, когда на этапе S421, показанном на фиг.55, определяется, что флаг `ts_flag` пропуска преобразования не представляет собой `2D_TS` (без указания пропуска двумерного преобразования), и флаг обхода квантования преобразования равен 0 (ложь), процесс переходит к этапу S422. В этом случае переключатель 431 подает  
20 входной предсказанный остаток `D` в блок 432 первичного преобразования.

Блок 432 первичного преобразования выполняет первичное преобразование над предсказанным остатком `D` на основании информации об адаптивном первичном преобразовании цветового сигнала, точно определенного идентификатором `compID` цветового сигнала, и выводит коэффициент `Coeff_P` преобразования, который был  
25 подвергнут первичному преобразованию. Более конкретно, на этапе S422 блок 432 первичного преобразования определяет, представляет ли собой идентификатор `compID` цветового сигнала яркость или цветность. В случае, когда определяется, что идентификатор `compID` цветового сигнала указывает цветность (`compID!` = `COMPONENT_Y`), процесс переходит к этапу S423.

30 На этапе S423 блок 441 получения информации об адаптивном первичном преобразовании цветности выполняет процесс получения информации об адаптивном первичном преобразовании цветности и получает идентификатор `pt_idx[compID]` первичного преобразования блока преобразования цветности на основании флага `apt_flag[compID]` (`compID`=`COMPONENT_Cb` или `COMPONENT_Cr`) адаптивного первичного  
35 преобразования блока преобразования цветности и идентификатора `pt_idx[COMPONENT_Y]` первичного преобразования блока преобразования яркости, соответствующей блоку преобразования цветности. Следует отметить, что, так как данный процесс аналогичен процессу получения информации об адаптивном первичном преобразовании цветности (фиг.15), выполняемому устройством 100 декодирования  
40 изображения (блоком 131 получения информации об адаптивном первичном преобразовании цветности) и описанному в первом варианте осуществления, его описание будет опущено. Другими словами, описание, представленное со ссылкой на блок-схему последовательности операций, показанную на фиг.15, можно также применить к описанию процесса получения информации об адаптивном первичном  
45 преобразовании цветности, выполняемому блоком 441 получения информации об адаптивном первичном преобразовании цветности.

Когда процесс на этапе S423 заканчивается, процесс переходит к этапу S424. Кроме того, в случае, когда идентификатор `compID` цветового сигнала определяется для

указания яркости (compID == COMPONENT\_Y) на этапе S422, процесс переходит к этапу S424.

На этапе S424 блок 442 выбора первичного преобразования выполняет процесс выбора первичного преобразования, получает тип TrTypeIdxH преобразования первичного горизонтального преобразования и тип TrTypeIdxV преобразования первичного вертикального преобразования цветového сигнала, точно определенного идентификатором compID цветového сигнала со ссылкой на информацию PInfo о режиме предсказания, идентификатор compID цветového сигнала, флаг apt\_flag[compID] адаптивного первичного преобразования цветového сигнала, соответствующего идентификатору compID цветového сигнала, и идентификатор pt\_idx[compID] первичного преобразования цветového сигнала, соответствующего идентификатору compID цветového сигнала, и выводит каждый из типов преобразования в блок 443 первичного горизонтального преобразования и блок 444 первичного вертикального преобразования.

На этапе S425 блок 443 первичного горизонтального преобразования выполняет первичное горизонтальное преобразование Phor, определенное типом TrTypeIdxH преобразования первичного горизонтального преобразования и высотой картинки блока преобразования, над предсказанным остатком D блока преобразования для каждого блока преобразования, точно определенного идентификатором compID цветového сигнала, и выводит результат в виде коэффициента Coeff\_Phор преобразования, который был подвергнут первичному горизонтальному преобразованию.

На этапе S426 блок 444 первичного вертикального преобразования выполняет первичное вертикальное преобразование Pver, определенное типом TrTypeIdxV преобразования первичного вертикального преобразования и высотой картинки блока преобразования, над коэффициентом Coeff\_Phор преобразования блока преобразования, который был подвергнут первичному горизонтальному преобразованию для каждого блока преобразования, точно определенного идентификатором compID цветového сигнала, и выводит результат в виде коэффициента Coeff\_P преобразования, который был подвергнут первичному преобразованию.

На этапе S427 блок 433 вторичного преобразования выполняет вторичное преобразование над входным коэффициентом первичного преобразования Coeff\_P на основании идентификатора st\_idx вторичного преобразования, и получает и выводит коэффициент Coeff преобразования. Когда процесс на этапе S427 заканчивается, процесс преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.54.

Последовательность операций процесса выбора первичного преобразования  
Далее будет описан пример последовательности операций процесса выбора первичного преобразования, выполняемого на этапе S424, показанном на фиг.55, со ссылкой на блок-схему последовательности операций, показанную на фиг.56.

Когда начинается процесс выбора первичного преобразования, блок 442 выбора первичного преобразования определяет, равен ли флаг apt\_flag[compID] адаптивного первичного преобразования цветového сигнала, соответствующего идентификатору compID цветového сигнала, 1 (истина) на этапе S431. В случае, когда определяется, что флаг apt\_flag[compID] адаптивного первичного преобразования равен 1 (истина), процесс переходит к этапу S432.

На этапе S432 блок 442 выбора первичного преобразования выбирает набор TrSetV преобразований (набор первичных горизонтальных преобразований) первичного вертикального преобразования и набор TrSetH преобразований (набор первичных вертикальных преобразований) первичного горизонтального преобразования на

основании информации PInfo о режиме предсказания.

На этапе S433 блок 442 выбора первичного преобразования получает флаг pt\_hor\_flag точного определения первичного горизонтального преобразования и флаг pt\_ver\_flag точного определения первичного вертикального преобразования из идентификатора pt\_idx[compID] первичного преобразования цветового сигнала, соответствующего идентификатору compID цветового сигнала.

На этапе S434 блок 442 выбора первичного преобразования выбирает тип TrTypeIdxH преобразования ортогонального преобразования, применяемого в качестве первичного горизонтального преобразования IPThor по отношению к набору TrSetH первичных горизонтальных преобразований, и флаг pt\_hor\_flag точного определения первичного горизонтального преобразования.

На этапе S435 блок 442 выбора первичного преобразования выбирает тип TrTypeIdxV преобразования ортогонального преобразования, применяемого в качестве обратного первичного вертикального преобразования IPTver по отношению к набору TrSetV первичных вертикальных преобразований, и флаг pt\_ver\_flag точного определения первичного вертикального преобразования. Когда процесс на этапе S435 заканчивается, процесс выбора первичного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.55.

Кроме того, на этапе S431 в случае, когда определяется, что флаг apt\_flag[compID] адаптивного первичного преобразования равен 0 (ложь), процесс переходит к этапу S436. На этапе S436 блок 442 выбора первичного преобразования выбирает заданное ортогональное преобразование в качестве типа TrTypeIdxH преобразования первичного горизонтального преобразования IPThor (TrTypeIdxH = заданное значение).

Кроме того, на этапе S437 блок 442 выбора первичного преобразования выбирает заданное ортогональное преобразование в качестве типа TrTypeIdxV преобразования первичного вертикального преобразования IPTver (TrTypeIdxV = заданное значение). Когда процесс на этапе S437 заканчивается, процесс выбора первичного преобразования заканчивается, и процесс возвращается к процессу, показанному на фиг.56.

То есть блок 432 первичного преобразования получает тип TrTypeIdxH преобразования обратного первичного горизонтального преобразования IPThor и тип TrTypeIdxV преобразования обратного первичного вертикального преобразования IPTver, используя способ в соответствии со значением флага apt\_flag[compID] адаптивного первичного преобразования цветности.

Последовательность операций процесса кодирования информации о первичном преобразовании

Блок 415 кодирования кодирует флаг apt\_flag адаптивного первичного преобразования и идентификатор pt\_idx первичного преобразования, выработанный блоком 413 преобразования. Однако блок 415 кодирования может кодировать флаг apt\_flag адаптивного первичного преобразования и идентификатор pt\_idx первичного преобразования яркости, и заставлять флаг и идентификатор осуществлять сигнализацию и может не кодировать флаг apt\_flag адаптивного первичного преобразования и идентификатор pt\_idx первичного преобразования цветности.

Пример последовательности операций процесса кодирования информации о первичном преобразовании, который выполняется блоком 415 кодирования с целью выполнения кодирования, описанного выше, будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.57. Когда начинается процесс кодирования информации о первичном преобразовании, на этапе S441 блок 415 кодирования определяет, представляет собой или нет объект обработки яркость. В

случае, когда определяется, что объектом обработки является яркость (`compID == COMPONENT_Y`), процесс переходит к этапу S442. На этапе S442 блок 415 кодирования кодирует флаг `apt_flag[COMPONENT_Y]` адаптивного первичного преобразования яркости (побуждает флаг к сигнализации). Кроме того, на этапе S443 блок 415 кодирования кодирует идентификатор `pt_idx[COMPONENT_Y]` первичного преобразования яркости (побуждает флаг к сигнализации). Когда процесс на этапе S443 заканчивается, процесс кодирования информации о первичном преобразовании заканчивается.

Следует отметить, что на этапе S441 блок 415 кодирования определяет, является ли объектом обработки цветность. В случае, когда определяется, что объектом обработки является цветность (`compID != COMPONENT_Y`), процессы на этапе S442 и на этапе S443 опускаются, то есть не осуществляется сигнализация флага `apt_flag` адаптивного первичного преобразования и идентификатора `pt_idx` первичного преобразования, и процесс кодирования информации о первичном преобразовании заканчивается.

Выполняя каждый процесс так, как описано выше, адаптивное первичное преобразование, выбранное для яркости, можно применить к блоку преобразования цветности в случае, когда остаточный сигнал цветности имеет тенденцию, аналогичную остаточному сигналу яркости. Поэтому процесс первичного преобразования с еще более увеличенной эффективностью кодирования можно выполнить над остаточным сигналом цветности по сравнению с технологиями уровня техники. Кроме того, так как нет необходимости кодировать информацию об адаптивном первичном преобразовании цветности, объем процесса кодирования может быть уменьшен в большей степени, чем в случае, когда флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования явно кодируются для каждой из яркости и цветности.

Процесс получения информации об адаптивном первичном преобразовании цветности

Следует отметить, что блок 441 получения информации об адаптивном первичном преобразовании цветности имеет аналогичную конфигурацию и выполняет процессы, аналогичные тем, которые выполняются блоком 131 получения информации об адаптивном первичном преобразовании цветности. То есть блок 441 получения информации об адаптивном первичном преобразовании цветности может выполнить процесс получения информации об адаптивном первичном преобразовании цветности, аналогично блоку 131 получения информации об адаптивном первичном преобразовании цветности. Поэтому устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности и дополнительно получить флаг `apt_flag` адаптивного первичного преобразования и идентификатор `pt_idx` первичного преобразования цветности из флага `apt_flag` адаптивного первичного преобразования и идентификатора `pt_idx` первичного преобразования яркости в любой момент времени, как указано в строке под номером #1 в таблице на фиг.7, и, таким образом, могут быть получены эффекты, аналогичные эффектам в случае устройства 100 декодирования изображения.

Аналогичным образом, устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности, дополнительно может установить значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в значение флага `apt_flag [Y]` адаптивного первичного преобразования яркости в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание (`CuPredMode == MODE_INTER`), например, и может установить флаг `apt_flag [Cb/Cr]`

адаптивного первичного преобразования цветности в 0 (ложь) в случае, когда типом предсказания блока кодирования является межблочное предсказание, как указано в строке под номером #2 в таблице на фиг.7. Таким образом, в этом случае могут быть также получены эффекты, аналогичные эффектам в случае устройства 100 декодирования изображения.

Аналогичным образом, устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности и дополнительно может установить значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в значение флага `apt_flag [Y]` адаптивного первичного преобразования яркости в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, например, и может установить флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в 0 (ложь) в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, как указано в строке под номером #3 в таблице на фиг.7. То есть в этом случае можно также получить аналогичные эффекты, как и в случае устройства 100 декодирования изображения.

Аналогичным образом, устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности, и дополнительно может установить значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в значение флага `apt_flag [Y]` адаптивного первичного преобразования яркости в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, например, и может установить значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в 0 (ложь) в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование (IntraBC) (которое также упоминается как компенсация движения на экране), как указано в строке под номером #4 в таблице на фиг.7. То есть в этом случае можно также получить аналогичные эффекты, как и в случае устройства 100 декодирования изображения.

Аналогичным образом, устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности, и дополнительно может установить флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в значение флага `apt_flag [Y]` адаптивного первичного преобразования яркости в случае, когда флаг `chroma_apt_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности, указывающий на то, получен ли флаг `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности на основании флага `apt_flag [Y]` адаптивного первичного преобразования яркости, имеет значение 1 (истина), например, и может установить значение флага `apt_flag [Cb/Cr]` адаптивного первичного преобразования цветности в 0 (ложь) в случае, когда флаг `chroma_apt_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности равен 0 (ложь), как указано в строке под номером #5 в таблице на фиг.7. Таким образом, можно однозначно управлять тем, следует ли использовать значение флага адаптивного первичного преобразования яркости, используя флаг `chroma_apt_info_infer_flag` вывода информации об адаптивном первичном преобразовании цветности. То есть в этом случае можно также получить аналогичные эффекты, как и в случае устройства 100

декодирования изображения.

Пример последовательности операций процесса кодирования флага вывода информации об адаптивном первичном преобразовании цветности для кодирования флага chroma\_apr\_info\_infer\_flag вывода информации об адаптивном первичном преобразовании цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.58.

Когда начинается процесс кодирования флага вывода информации об адаптивном первичном преобразовании цветности, на этапе S451 блок 415 кодирования определяет, является ли объектом обработки цветность. В случае, когда определяется, что объектом обработки является цветность, процесс переходит к этапу S452. На этапе S452 блок 415 кодирования определяет, имеет ли флаг apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования яркости значение "истина". В случае, когда определяется, что флаг имеет значение "истина", процесс переходит к этапу S453. На этапе S453 блок 415 кодирования определяет, имеет ли флаг transquant\_bypass\_enabled\_flag разрешения обхода квантования преобразования значение "ложь". В случае, когда определяется, что флаг имеет значение "ложь", процесс переходит к этапу S454. На этапе S454 блок 415 кодирования определяет, имеет или нет флаг ts\_flag пропуска преобразования цветности значение "ложь". В случае, когда определяется, что флаг ts\_flag пропуска преобразования цветности имеет значение "ложь", процесс переходит к этапу S455. На этапе S455 блок 415 кодирования кодирует флаг chroma\_apr\_info\_infer\_flag вывода информации об адаптивном первичном преобразовании цветности. Когда процесс на этапе S455 заканчивается, процесс кодирования флага вывода информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.54.

Следует отметить, что в случае, когда на этапе S451, показанном на фиг.58, определяется, что объектом обработки является яркость, процесс кодирования флага вывода информации об адаптивном первичном преобразовании заканчивается, и процесс возвращается к процессу, показанному на фиг.54. Кроме того, в случае, когда на этапе S452 определяется, что флаг apt\_flag[COMPONENT\_Y] адаптивного первичного преобразования яркости, имеет значение "ложь", процесс кодирования флага вывода информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.54. Кроме того, в случае, когда на этапе S453 определяется, что флаг transquant\_bypass\_enabled\_flag разрешения обхода квантования преобразования имеет значение "истина", процесс кодирования флага вывода информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.54. Кроме того, в случае, когда на этапе S454 определяется, что флаг ts\_flag пропуска преобразования цветности является истинным, процесс кодирования флага вывода информации об адаптивном первичном преобразовании цветности заканчивается, и процесс возвращается к процессу, показанному на фиг.54.

Кроме того, устройство 400 кодирования изображения может применить адаптивное первичное преобразование также к цветности, и дополнительно может установить значение флага apt\_flag [Cb/Cr] адаптивного первичного преобразования цветности в значение флага apt\_flag [Y] адаптивного первичного преобразования яркости в случае, когда размер короткой стороны обрабатываемого блока преобразования цветности больше или равен значению заданного порогового значения, например, и может установить значение флага apt\_flag [Cb/Cr] адаптивного первичного преобразования цветности в 0 (ложь) в случае, когда размер короткой стороны блока преобразования

меньше порогового значения, как указано в строке под номером #6 в таблице на фиг.7. То есть в этом случае можно также получить аналогичные эффекты, как и в случае устройства 100 декодирования изображения.

Аналогичным образом, устройство 400 кодирования изображения может применить  
 5 адаптивное (обратное) первичное преобразование также к цветности, и дополнительно может установить тип преобразования (обратного) первичного горизонтального преобразования в заданный тип преобразования в случае, когда размер ширины  
 картинки блока преобразования с цветностью, подлежащей обработке, равен или  
 10 меньше заданного порогового значения, например, и может установить тип преобразования (обратного) первичного горизонтального преобразования на основании  
 набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования в случае, когда размер ширины картинки блока  
 преобразования больше порогового значения, как указано в строке под номером #7 в  
 таблице на фиг.7. Кроме того, аналогичным образом, в случае, когда размер высоты  
 15 картинки блока преобразования с цветностью, подлежащей обработке, равен или меньше заданного порогового значения, тип преобразования (обратного) первичного  
 вертикального преобразования может быть установлен в заданный тип преобразования, и в случае, когда размер высоты картинки блока преобразования больше порогового  
 значения, тип преобразования (обратного) первичного вертикального преобразования  
 20 может быть установлен на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования. То есть в этом случае  
 можно также получить аналогичные эффекты, как в случае устройства 100 декодирования изображения.

Разумеется, множество вышеописанных случаев можно произвольно объединить,  
 25 как и в случае устройства 400 кодирования изображения. Кроме того, каждый из вышеописанных случаев может быть также объединен с другими случаями, которые не описаны выше. То есть в этом случае можно также получить аналогичные эффекты, как и в случае устройства 100 декодирования изображения.

#### 6. Пятый вариант осуществления

30 `ts_flag`

Информация относительно (обратного) ортогонального преобразования может  
 включать в себя флаг `ts_flag` пропуска преобразования, указывающий на то, должен  
 ли быть пропущен (обратный) процесс ортогонального преобразования, как и в случае  
 декодирования. Получая каждый флаг `ts_flag[Cb]` пропуска преобразования цветности  
 35 (`Cb`) и флаг `ts_flag[Cr]` пропуска преобразования цветности (`Cr`) на основании флага  
`ts_flag` пропуска преобразования [`Y`] яркости, например, можно также сдерживать  
 увеличение издержек объема кода синтаксиса, просигнализированного цветностью (`Cb/Cr`).

#### Основной пример 1

40 Флаг `ts_flag` пропуска преобразования цветности получается из флага `ts_flag` пропуска преобразования яркости в любом случае, например, как указано в строке под номером  
 #1 на фиг.27. Например, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]`  
 пропуска преобразования яркости.

45 В этом случае блок 411 управления выполняет процесс кодирования флага пропуска преобразования и управляет кодированием флага `ts_flag` пропуска преобразования.  
 Пример последовательности операций процесса кодирования флага пропуска преобразования будет описан со ссылкой на блок-схему последовательности операций,

показанную на фиг.59.

Когда начинается процесс кодирования флага пропуска преобразования, каждый из процессов этапа S461 – этапа S464, показанных на фиг.59, выполняется аналогично каждому из процессов этапа S241 – этапа S244 процесса получения флага пропуска преобразования, показанного на фиг.29.

Когда на этапе S464 определяется, что объектом обработки является цветность (`compID! = COMPONENT_Y`), процесс переходит к этапу S465. На этапе S465 блок 411 управления устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости (`ts_flag[compID] == ts_flag[COMPONENT_Y]`). Когда процесс на этапе S465 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае флаг `ts_flag[compID]` пропуска преобразования цветности может быть получен из флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости аналогично на стороне декодирования, и, таким образом, сигнализация флага не осуществляется.

Кроме того, в случае, когда на этапе S464 определяется, что объектом обработки является яркость (`compID == COMPONENT_Y`), процесс переходит к этапу S466. На этапе S466 блок 411 управления устанавливает произвольное значение для флага `ts_flag[compID]` пропуска преобразования яркости. Затем блок 411 управления подает флаг `ts_flag[compID]` пропуска преобразования яркости в блок 415 кодирования для воспроизведения флага, который будет кодироваться на этапе S467. Когда процесс на этапе S467 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае осуществляется сигнализация (передается) флага `ts_flag[compID]` пропуска преобразования яркости. Сторона декодирования получает флаг `ts_flag[compID]` пропуска преобразования яркости посредством извлечения и декодирования с использованием битового потока.

То есть в случае основного примера 1, осуществляется сигнализация всех флагов `ts_flag` пропуска преобразования яркости, и не осуществляется сигнализация всех флагов `ts_flag` пропуска преобразования цветности. Таким образом, можно опустить передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 1

Кроме того, как в строке под номером #2, показанной на фиг.27, например, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и в случае, когда типом предсказания блока кодирования является межблочное предсказание, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных).

Пример последовательности операций процесса кодирования флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.60. Каждый из процессов этапа S471 – этапа S475 выполняется аналогично каждому из процессов этапа S251 – этапа S255, показанных на фиг.31.

То есть в случае, когда на этапе S474 определяется, что объектом обработки является цветность, и на этапе S475 определяется, что типом предсказания является

внутриблочное предсказание, процесс переходит к этапу S476. На этапе S476 блок 411 управления устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости (`ts_flag[compID] == ts_flag[COMPONENT_Y]`). Когда процесс на этапе S476 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае сторона декодирования может аналогичным образом получить флаг `ts_flag[compID]` пропуска преобразования блока преобразования в случае цветности и внутриблочного предсказания из флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости, и, таким образом, флаг не сигнализируется.

С другой стороны, в случае, когда на этапе S474 определяется, что объектом обработки является яркость (`compID == COMPONENT_Y`), или в случае, когда на этапе S475 определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S477. На этапе S477 блок 411 управления устанавливает произвольное значение для флага `ts_flag[compID]` пропуска преобразования. Затем на этапе S478 блок 411 управления подает флаг `ts_flag[compID]` пропуска преобразования в блок 415 кодирования для воспроизведения флага, который будет кодироваться. Когда процесс на этапе S478 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае сигнализируется (передается) флаг `ts_flag[compID]` пропуска преобразования блока преобразования в случае яркости или межблочного предсказания. Сторона декодирования может получить флаг `ts_flag[compID]` пропуска преобразования посредством извлечения и декодирования с использованием битового потока.

То есть в случае модифицированного примера 1 сигнализируются флаг `ts_flag` пропуска преобразования яркости и флаг `ts_flag` пропуска преобразования блока преобразования, а режимом предсказания является межблочное предсказание, и не сигнализируется флаг `ts_flag` пропуска преобразования цветности, а режимом предсказания является внутриблочное предсказание. Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить только в случае режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 2

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, как показано в строке под номером #3 на фиг.27, или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, например, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных).

Пример последовательности операций процесса кодирования флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.61. Каждый из процессов этапа S481 – этапа S486 выполняется аналогично каждому из процессов этапа S261 – этапа S266,

показанных на фиг.33.

То есть в случае, когда на этапе S484 определяется, что объектом обработки является цветность, и на этапе S485 определяется, что типом предсказания является внутриблочное предсказание, или на этапе S485 определяется, что типом предсказания является межблочное предсказание, и в случае, когда на этапе S486 режимы предсказания для яркости и цветности совпадают друг с другом, процесс переходит к этапу S487. На этапе S487 блок 411 управления устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости (`ts_flag[compID] == ts_flag[COMPONENT_Y]`). Когда процесс на этапе S487 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае, как и в случае межблочного предсказания, в котором режимы предсказания яркости и цветности совпадают друг с другом, а также при соблюдении условия модифицированного примера 1, тенденции остаточных сигналов являются аналогичными, и, таким образом, сигнализация флага `ts_flag[compID]` пропуска преобразования не осуществляется.

С другой стороны, в случае, когда на этапе S484 определяется, что объектом обработки является яркость (`compID == COMPONENT_Y`), или в случае, когда на этапе S486 определяется, что типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, процесс переходит к этапу S488. На этапе S488 блок 411 управления устанавливает произвольное значение для флага `ts_flag[compID]` пропуска преобразования. Затем на этапе S489 блок 411 управления подает флаг `ts_flag[compID]` пропуска преобразования в блок 415 кодирования для воспроизведения флага, который будет кодироваться. Когда процесс на этапе S489 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае сигнализируются (передаются) флаг `ts_flag[compID]` пропуска преобразования яркости и флаг `ts_flag[compID]` пропуска преобразования блока преобразования для межблочного предсказания, а режимы предсказания для яркости и цветности не совпадают друг с другом. Сторона декодирования может получить флаг `ts_flag[compID]` пропуска преобразования посредством извлечения и декодирования с использованием битового потока.

То есть в случае модифицированного примера 2 флаг `ts_flag` пропуска преобразования яркости и флаг `ts_flag` пропуска преобразования блока преобразования для межблочного предсказания, в котором режимы предсказания для яркости и цветности не совпадают друг с другом, сигнализируются, и другие флаги `ts_flag` пропуска преобразования не сигнализируются. Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить только в случае режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

### Модифицированный пример 3

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, как в строке под номером #4, показанной на фиг.27, или межблочное предсказание, а режимом предсказания является внутриблочное копирование, например, значение флага `ts_flag[compID]` пропуска преобразования цветности может быть установлено равным значению флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости в случае, когда типом предсказания блока кодирования является межблочное

предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, флаг `ts_flag[compID]` пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag[compID]` пропуска преобразования цветности может быть декодирован из кодированных данных).

5 Пример последовательности операций процесса кодирования флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.62. Каждый из процессов этапа S491 – этапа S496 выполняется аналогично каждому из процессов этапа S271 – этапа S276, показанных на фиг.35.

10 То есть в случае, когда на этапе S494 и на этапе S495 определяется, что объектом обработки является цветность, и типом предсказания является внутриблочное предсказание, или в случае, когда на этапе S495 и этапе S496 определяется, что типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование, процесс переходит к этапу S497. На этапе S497 блок 411  
15 управления устанавливает значение флага `ts_flag[compID]` пропуска преобразования цветности в значение флага `ts_flag[COMPONENT_Y]` пропуска преобразования яркости (`ts_flag[compID] == ts_flag[COMPONENT_Y]`). Когда процесс на этапе S497 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае, как и в случае межблочного предсказания, для которого режимом предсказания  
20 является внутриблочное копирование, а также при соблюдении условия модифицированного примера 1, тенденции остаточных сигналов являются аналогичными, и таким образом, не осуществляется сигнализация флага `ts_flag[compID]` пропуска преобразования.

С другой стороны, в случае, когда на этапе S494 определяется, что объектом  
25 обработки является яркость (`compID == COMPONENT_Y`), или в случае, когда на этапе S495 и этапе S496 определяется, что типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, процесс переходит к этапу S498. На этапе S498 блок 411 управления устанавливает произвольное значение для флага `ts_flag[compID]` пропуска преобразования. Затем на этапе S499 блок  
30 411 управления подает флаг `ts_flag[compID]` пропуска преобразования в блок 415 кодирования для воспроизведения флага, который будет кодироваться. Когда процесс на этапе S489 заканчивается, процесс кодирования флага пропуска преобразования заканчивается. То есть в этом случае сигнализируются (передаются) флаг `ts_flag[compID]` пропуска преобразования яркости и флаг `ts_flag[compID]` пропуска преобразования  
35 блока преобразования для межблочного предсказания, а режимом предсказания не является внутриблочное копирование. Сторона декодирования может получить флаг `ts_flag[compID]` пропуска преобразования посредством извлечения и декодирования с использованием битового потока.

То есть в случае модифицированного примера 3 сигнализируются флаг `ts_flag`  
40 пропуска преобразования яркости и флаг `ts_flag` пропуска преобразования блока преобразования для межблочного предсказания, при этом режимом предсказания не является внутриблочное копирование, и другие флаги `ts_flag` пропуска преобразования не сигнализируются. Таким образом, передачу (кодирование и декодирование) флага `ts_flag` пропуска преобразования цветности можно опустить только в случае режима  
45 внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 4

Кроме того, в случае, когда флаг `chroma_ts_info_infer_flag` вывода информации пропуска преобразования цветности равен 1 (истина), как в строке под номером #5, показанной на фиг.27, например, значение флага `ts_flag` пропуска преобразования [compID] цветности может быть установлено равным значению флага `ts_flag` [COMPONENT\_Y] пропуска преобразования яркости, и в случае, когда флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности, равен 0 (ложь), флаг `ts_flag`[compID] пропуска преобразования цветности может быть просигнализирован (то есть флаг `ts_flag`[compID] пропуска преобразования цветности может быть декодирован из кодированных данных).

Пример последовательности операций процесса кодирования флага пропуска преобразования в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.63. Каждый из процессов этапа S501 – этапа S504 выполняется аналогично каждому из процессов этапа S281 – этапа S284, показанных на фиг.38.

То есть в случае, когда на этапе S504 определяется, что флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности имеет значение "истина", процесс переходит к этапу S505. На этапе S505 блок 411 управления устанавливает значение флага `ts_flag`[compID] пропуска преобразования цветности в значение флага `ts_flag`[COMPONENT\_Y] пропуска преобразования яркости (`ts_flag`[compID] == `ts_flag` [COMPONENT\_Y]). Когда процесс на этапе S505 заканчивается, процесс кодирования флага пропуска преобразования заканчивается.

С другой стороны, когда на этапе S504 определяется, что флаг `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности имеет значение "ложь", процесс переходит к этапу S506. На этапе S506 блок 411 управления устанавливает произвольное значение для флага `ts_flag`[compID] пропуска преобразования. Затем на этапе S507 блок 411 управления подает флаг `ts_flag`[compID] пропуска преобразования в блок 415 кодирования и вызывает кодирование флага. Когда процесс на этапе S507 заканчивается, процесс кодирования флага пропуска преобразования заканчивается.

То есть в случае модифицированного примера 4, явное управление тем, следует или нет сигнализировать флаг `ts_flag`[compID] пропуска преобразования, выполняется с использованием флага `chroma_ts_info_infer_flag` вывода информации о пропуске преобразования цветности. Поэтому можно легко установить значение флага пропуска преобразования яркости, используемого только в случае, когда может быть получен достаточно большой эффект, и, таким образом, можно дополнительно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### 7. Шестой вариант осуществления

##### `st_idx`

Как и в случае декодирования, идентификатор `st_idx` вторичного преобразования, указывающий на то, какое (обратное) вторичное преобразование должно быть применено, может быть включен в информацию относительно (обратного) ортогонального преобразования. Получая идентификатор `chroma_st_idx` вторичного преобразования цветности (характерный для Cb и Cr) на основании идентификатора `st_idx` [Y] вторичного преобразования яркости, например, можно также сдерживать увеличение издержек объема кода синтаксиса, с помощью которого осуществляется сигнализация цветности (Cb или Cr).

#### Основной пример 1

Например, идентификатор `st_idx` вторичного преобразования цветности может быть получен в любой момент времени из идентификатора `st_idx` вторичного преобразования яркости, как указано в строке под номером #1, показанной на фиг.39. Например, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть

5 установлен равным значению идентификатора `st_idx` вторичного преобразования яркости.

В этом случае блок 411 управления выполняет процесс кодирования идентификатора вторичного преобразования для цветности и управляет кодированием идентификатора `chroma_st_idx` вторичного преобразования для цветности. Пример последовательности

10 операций процесса кодирования идентификатора вторичного преобразования для цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.64.

Когда начинается процесс кодирования идентификатора вторичного преобразования для цветности, каждый из процессов этапа S511 – этапа S514, показанных на фиг.64,

15 выполняется аналогично каждому из процессов этапа S291 – этапа S294 процесса кодирования идентификатора вторичного преобразования цветности, показанного на фиг.42.

Таким образом, передачу (кодирование и декодирование) идентификатора вторичного преобразования `chroma_ts_flag` цветности можно опустить, и, таким образом, можно

20 сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 1

В случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание,

25 значение идентификатора `chroma_st_idx` вторичного преобразования цветности может быть установлено равным значению идентификатора `st_idx` вторичного преобразования яркости, как описано со ссылкой на строку под номером #2, показанную на фиг.39, и в случае, когда типом предсказания блока кодирования является межблочное

30 предсказание, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных).

В этом случае блок 411 управления выполняет процесс кодирования идентификатора вторичного преобразования для цветности и управляет кодированием идентификатора `chroma_st_idx` вторичного преобразования цветности. Пример последовательности

35 операций процесса кодирования идентификатора вторичного преобразования для цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.65.

Когда начинается процесс кодирования идентификатора вторичного преобразования для цветности, каждый из процессов этапа S521 – этапа S524, показанных на фиг.65,

40 выполняется аналогично каждому из процессов этапа S301 – этапа S304 процесса кодирования идентификатора вторичного преобразования цветности, показанного на фиг.44.

То есть в случае, когда на этапе S524 определяется, что типом предсказания является внутриблочное предсказание, процесс переходит к этапу S525. На этапе S525 блок 411

45 управления устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx` вторичного преобразования яркости (`chroma_st_idx == st_idx`). Когда процесс на этапе S525 заканчивается, процесс кодирования идентификатора вторичного преобразования для цветности заканчивается.

То есть в этом случае сторона декодирования может аналогичным образом получить идентификатор `chroma_st_idx` вторичного преобразования блока преобразования в случае цветности и внутриблочного предсказания из идентификатора `st_idx` вторичного преобразования яркости, и, таким образом, идентификатор не сигнализируется.

- 5 С другой стороны, в случае, когда на этапе S524 определяется, что типом предсказания является межблочное предсказание, процесс переходит к этапу S526. На этапе S526 блок 411 управления устанавливает произвольное значение для идентификатора `chroma_st_idx` вторичного преобразования цветности. Затем на этапе S527 блок 411 управления подает идентификатор `chroma_st_idx` вторичного преобразования цветности в блок 415 кодирования для воспроизведения идентификатора, который будет кодироваться. Когда процесс на этапе S527 заканчивается, процесс кодирования идентификатора вторичного преобразования для цветности заканчивается. То есть в этом случае сигнализируется (передается)
- 10 идентификатор `chroma_st_idx` вторичного преобразования цветности блока преобразования для межблочного предсказания. Сторона декодирования может получить идентификатор `chroma_st_idx` вторичного преобразования цветности посредством извлечения и декодирования с использованием битового потока.
- 15

Таким образом, передачу (кодирование и декодирование) идентификатора `chroma_st_idx` вторичного преобразования цветности можно опустить только в случае

20 режима внутриблочного предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 2

- 25 Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, как в строке под номером #3, показанной на фиг.39, например, значение идентификатора `chroma_st_idx` вторичного преобразования цветности
- 30 устанавливается в значение идентификатора `st_idx` вторичного преобразования яркости. Кроме того, в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом, идентификатор `chroma_st_idx` вторичного преобразования цветности может быть просигнализирован (то есть идентификатор `chroma_st_idx` вторичного преобразования цветности может быть декодирован из кодированных данных).
- 35

Пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.66. Каждый из процессов этапа S531 – этапа S535 выполняется аналогично каждому из процессов этапа S311 – этапа S315,

40 показанных на фиг.46.

- То есть в случае, когда на этапе S534 определяется, что типом предсказания является внутриблочное предсказание, или в случае, когда на этапе S534 и S535 определяется, что типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, процесс переходит к этапу S536. На
- 45 этапе S536, блок 411 управления устанавливает значение идентификатора `chroma_st_idx` вторичного преобразования цветности в значение идентификатора `st_idx` вторичного преобразования яркости (`chroma_st_idx == st_idx`). Когда процесс на этапе S536 заканчивается, процесс кодирования идентификатора вторичного преобразования для

цветности заканчивается. То есть, как и в случае межблочного предсказания, режимы предсказания для яркости и цветности совпадают друг с другом, а также при соблюдении условия модифицированного примера 1, тенденции остаточных сигналов являются аналогичными, и, таким образом, идентификатор chroma\_st\_idx вторичного преобразования цветности может не сигнализироваться.

С другой стороны, в случае, когда определяется, что типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом на этапе S535 (compID! = COMPONENT\_Y), процесс переходит к этапу S537. На этапе S537 блок 411 управления устанавливает произвольное значение для идентификатора chroma\_st\_idx вторичного преобразования цветности. Затем на этапе S538, блок 411 управления подает идентификатор chroma\_st\_idx вторичного преобразования цветности в блок 415 кодирования для воспроизведения идентификатора, который будет кодироваться. Когда процесс на этапе S538 заканчивается, процесс кодирования идентификатора вторичного преобразования для цветности заканчивается.

Таким образом, передачу (кодирование и декодирование) идентификатора chroma\_st\_idx вторичного преобразования цветности можно опустить только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

### Модифицированный пример 3

Кроме того, в случае, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, как в строке под номером #4, показанной на фиг.39, например, значение идентификатора chroma\_st\_idx вторичного преобразования цветности устанавливается в значение идентификатора st\_idx вторичного преобразования яркости. Кроме того, в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, идентификатор chroma\_st\_idx вторичного преобразования цветности может быть просигнализирован (то есть идентификатор chroma\_st\_idx вторичного преобразования цветности может быть декодирован из кодированных данных).

Пример последовательности операций процесса кодирования идентификатора вторичного преобразования для цветности будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.67. Каждый из процессов этапа S541 – этапа S545 выполняется аналогично каждому из процессов этапа S321 – этапа S325, показанных на фиг.48.

То есть в случае, когда на этапе S544 определяется, что типом предсказания является внутриблочное предсказание, или в случае, когда на этапах S544 и S545 определяется, что типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование, процесс переходит к этапу S546. На этапе S546 блок 411 управления устанавливает значение идентификатора chroma\_st\_idx вторичного преобразования цветности в значение идентификатора st\_idx вторичного преобразования яркости (chroma\_st\_idx == st\_idx). Когда процесс на этапе S546 заканчивается, процесс кодирования идентификатора вторичного преобразования для цветности заканчивается. То есть, как и в случае межблочного предсказания, для которого режимом предсказания является внутриблочное копирование, а также при соблюдении условия

модифицированного примера 1, тенденции остаточных сигналов являются аналогичными, и, таким образом, идентификатор chroma\_st\_idx вторичного преобразования цветности может не сигнализироваться.

С другой стороны, в случае, когда на этапе S545 определяется, что типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование, процесс переходит к этапу S547. На этапе S547 блок 411 управления устанавливает произвольное значение для идентификатора chroma\_st\_idx вторичного преобразования цветности. Затем на этапе S548 блок 411 управления подает идентификатор chroma\_st\_idx вторичного преобразования цветности в блок 415 кодирования для воспроизведения идентификатора, который будет кодироваться. Когда процесс на этапе S548 заканчивается, процесс кодирования идентификатора вторичного преобразования для цветности заканчивается.

Таким образом, передачу (кодирование и декодирование) идентификатора chroma\_st\_idx вторичного преобразования цветности можно опустить только в случае режима предсказания, в котором тенденции остаточных сигналов являются аналогичными, и, таким образом, можно сдерживать снижение эффективности кодирования. Кроме того, можно также сдерживать увеличение нагрузки на кодирование и декодирование.

#### Модифицированный пример 5

Кроме того, как в строке под номером #5, показанной на фиг.39, например, в случае, когда флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности равен 1 (истина), значение идентификатора chroma\_st\_idx вторичного преобразования цветности может быть установлено равным значению идентификатора st\_idx вторичного преобразования яркости, и в случае, когда флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности равен 0 (ложь), идентификатор chroma\_st\_idx вторичного преобразования цветности может быть просигнализирован (то есть идентификатор chroma\_st\_idx вторичного преобразования цветности может быть декодирован из кодированных данных).

Пример последовательности операций процесса кодирования флага вывода идентификатора вторичного преобразования цветности в данном случае будет описан со ссылкой на блок-схему последовательности операций, показанную на фиг.68. Каждый из процессов этапа S551 – этапа S553 выполняется аналогично каждому из процессов этапа S331 – этапа S333, показанных на фиг.50.

То есть на этапе S554 блок 411 управления устанавливает флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности и подает флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности в блок 415 кодирования для того, чтобы осуществить его кодирование на этапе S555.

На этапе S556 блок 411 управления определяет, имеет ли флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности значение "истина". В случае, когда определяется, что флаг chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности имеет значение "истина", процесс переходит к этапу S557. На этапе S557 блок 411 управления применяет идентификатор st\_idx вторичного преобразования для яркости к идентификатору chroma\_st\_idx вторичного преобразования цветности.

Кроме того, в случае, когда на этапе S556 определяется, что значение флага chroma\_st\_idx\_infer\_flag вывода идентификатора вторичного преобразования цветности имеет значение "ложь", процесс переходит к этапу S558. На этапе S558 блок 411 управления устанавливает идентификатор chroma\_st\_idx вторичного преобразования

цветности. Кроме того, на этапе S559, блок 411 управления подает идентификатор в блок 415 кодирования для воспроизведения идентификатора, который будет кодироваться. Когда процесс на этапе S559 заканчивается, процесс кодирования флага пропуска преобразования заканчивается.

- 5 То есть в случае модифицированного примера 5 явное управление тем, осуществлять или нет сигнализацию идентификатора chroma\_st\_idx вторичного преобразования цветности, выполняется с использованием флага вывода идентификатора вторичного преобразования цветности. Поэтому легко и надежным образом можно выбрать более эффективный способ. Таким образом, можно дополнительно сдерживать снижение
- 10 эффективности кодирования. Кроме того, можно также управлять увеличением нагрузки кодирования и декодирования.

## 8. Прочее

### Блок данных информации

- Блок данных (или блок данных целевых данных), с помощью которого
- 15 устанавливаются информация относительно изображения и информация относительно кодирования и декодирования изображения, описанного выше, является произвольным в каждом случае и не ограничивается вышеописанными примерами. Например, информацию можно установить в каждой из единицы преобразования (TU), блока преобразования (TB), единицы предсказания (PU), блока предсказания (PB), единицы
- 20 кодирования (CU), наибольшей единицы кодирования (LCU), подблока, блока, ячейки, слайса, картинки, последовательности или компонента, или данные блоков данных могут быть целевыми. Конечно, блок данных устанавливается для каждого фрагмента информации. То есть все фрагменты информации нельзя установить (или нацелить) в одном и том же блоке данных. Следует отметить, что место хранения информации
- 25 является произвольным, и информация может храниться в заголовке вышеописанного блока данных, набора параметров или т.п. Кроме того, информация может храниться во множестве мест.

### Управляющая информация

- Управляющая информация, которая относится к настоящей технологии, описанной
- 30 в каждом из приведенных выше вариантов осуществления, может передаваться со стороны кодирования на сторону декодирования. Например, может передаваться управляющая информация, которая управляет, разрешить (или запретить) или нет применение настоящей технологии, описанной выше. Кроме того, например, может передаваться управляющая информация, которая точно определяет верхний предел,
- 35 нижний предел или и то и другое размера блока, который разрешает (или запрещает) применение настоящей технологии, описанной выше.

### Кодирование и декодирование

- Настоящая технология может быть применена к кодированию и декодированию произвольного изображения, при котором выполняются первичное преобразование и
- 40 вторичное преобразование (обратное вторичное изменение и обратное первичное преобразование). То есть спецификации преобразования (обратного преобразования), квантования (обратного квантования), кодирования (декодирования), предсказания и т.п. являются произвольными и не ограничиваются вышеописанными примерами. Например, по отношению к преобразованию (обратному преобразованию) могут
- 45 выполняться (обратные) преобразования (то есть три или более (обратных), преобразований), которые отличаются от (обратного) первичного преобразования и (обратного) вторичного преобразования. Кроме того, кодирование (декодирование) может представлять собой обратимый способ или необратимый способ. Кроме того,

квантование (обратное квантование), предсказание и т.п. могут быть опущены. Кроме того, может быть выполнен процесс, который не описан выше, например, процесс фильтрации.

Области применения настоящей технологии

5 Система, устройство, процессор и т.п., к которым применима настоящая технология, могут использоваться в любых областях, таких как дорожное движение, медицинское обслуживание, предупреждение преступности, сельское хозяйство, животноводство, горнодобывающая промышленность, косметология, промышленные предприятия, бытовая техника, метеорологические наблюдения, наблюдения за природой или т.п.

10 В одном примере настоящая технология применима к системам и устройствам, которые передают изображение, предоставленное для просмотра. В дополнение к этому, в одном примере настоящая технология применима к системам и устройствам, предназначенным для дорожного движения. Кроме того, в одном примере настоящая технология применима к системам и устройствам, предназначенным для обеспечения  
15 безопасности. В дополнение к этому, в одном примере настоящая технология применима к системам и устройствам, предназначенным для спортивных мероприятий. Кроме того, в одном примере настоящая технология применима к системам и устройствам, предназначенным для сельского хозяйства. В дополнение к этому, в одном примере настоящая технология применима к системам и устройствам, предназначенным для  
20 промышленного животноводства. Кроме того, в одном примере настоящая технология также применима к системам и устройствам, предназначенным для мониторинга природных условий, таких как вулканы, леса, океаны и т.п. В дополнение к этому, в одном примере настоящая технология применима к системам наблюдения за погодой и устройствам наблюдения за погодой для наблюдения за погодой, температурой,  
25 влажностью, скоростью ветра, продолжительностью солнечного света и т.п. Кроме того, в одном примере настоящая технология также применима к системам, устройствам и т.п. для наблюдения за экологией дикой природы, такой как птицы, рыбы, рептилии, амфибии, млекопитающие, насекомые, растения и т.п.

Применение в системе кодирования и декодирования многоракурсных изображений

30 Ряд процессов, описанных выше, можно применить к системе кодирования и декодирования многоракурсных изображений, которая выполняет кодирование и декодирование многоракурсных изображений, в том числе и изображений с множеством ракурсов (видов). В этом случае настоящая технология может быть применена к кодированию и декодированию каждого ракурса (вида).

35 Применение в системе кодирования и декодирования иерархических изображений  
Помимо всего прочего, ряд процессов, описанных выше, может быть применен к системе кодирования (масштабируемого кодирования) и декодирования иерархических изображений, которая выполняет кодирование и декодирование иерархических изображений, которые были разбиты на множество слоев (иерархии), чтобы иметь  
40 функцию масштабируемости по отношению к заданному параметру. В этом случае настоящая технология может применяться для кодирования и декодирования каждой из иерархий (уровней).

Компьютер

Ряд процессов, описанных выше, может выполняться с помощью аппаратных средств,  
45 а также с помощью программного обеспечения. В случае выполнения ряда процессов с помощью программного обеспечения на компьютер устанавливается программа, принимающая форму программного обеспечения. В данном документе термин "компьютер" включает в себя компьютер, встроенный в аппаратные средства

специального назначения, компьютер, способный выполнять различные функции после установки на него различных программ, такой, например, как персональный компьютер общего назначения и т.п.

На фиг.69 показана блок-схема, иллюстрирующая примерную аппаратную конфигурацию компьютера, который выполняет ряд процессов, описанных выше в соответствии с программой.

В компьютере 800, показанном на фиг.69, центральное процессорное устройство (CPU) 801, постоянное запоминающее устройство (ROM) 802 и оперативное запоминающее устройство (RAM) 803 взаимодействуют между собой через шину 804.

Кроме того, интерфейс 810 ввода/вывода также подсоединен к шине 804. Блок 811 ввода, блок 812 вывода, блок 813 хранения данных, блок 814 связи и накопитель 815 подсоединены к интерфейсу 810 ввода/вывода.

Блок 811 ввода включает в себя, например, клавиатуру, мышь, микрофон, сенсорную панель, терминал ввода и т.п. Блок 812 вывода включает в себя, например, дисплей, громкоговоритель, терминал вывода и т.п. Блок 813 хранения данных включает в себя, например, жесткий диск, диск RAM, энергонезависимую память и т.п. Блок 814 связи включает в себя, например, сетевой интерфейс. Накопитель 815 управляет съемным носителем 821, таким как магнитный диск, оптический диск, магнитооптический диск или полупроводниковая память.

В компьютере, сконфигурированном так, как изложено выше, ряд процессов, описанных выше, выполняется, например, путем предписания CPU 801 загружать программу, хранящуюся в блоке 813 хранения, в RAM 803 через интерфейс 810 ввода/вывода и шину 804 и исполнять программу. Кроме того, данные, необходимые для того, чтобы CPU 801 выполняло различные процессы и т.п., также при необходимости сохраняются в RAM 803.

Программа, исполняемая компьютером (CPU 801), может применяться, будучи записанной на съемный носитель 821, например, в виде экземпляра упакованного носителя и т.п. В этом случае программу можно установить в блоке 813 хранения через интерфейс 810 ввода/вывода путем вставки съемного носителя 821 в накопитель 815.

Кроме того, программа может также предоставляться через проводную или беспроводную среду передачи данных, такую как локальная сеть, Интернет или цифровое спутниковое вещание. В этом случае программа может быть принята блоком 814 связи и установлена в блоке 813 хранения.

В иных случаях, программу можно также предварительно установить в ROM 802 или в блоке 813 хранения.

#### Применение настоящей технологии

Устройство 400 кодирования изображения и устройство 100 декодирования изображения согласно вышеупомянутым вариантам осуществления может применяться, например, в различных электронных устройствах, таких как передатчики или приемники для спутникового вещания, проводного вещания, такого как кабельное телевидение, распределение по Интернет и распределение по терминалам через сотовую связь; записывающие устройства, которые записывают изображения на носители, такие как оптические диски, магнитные диски и флэш-память; или устройства воспроизведения, которые воспроизводят изображения с вышеупомянутых носителей информации.

#### Первый пример применения: телевизионный приемник

На фиг.70 показан пример схематичной конфигурации телевизионного устройства, к которому применим вышеописанный вариант осуществления. Телевизионное устройство 900 имеет антенну 901, тюнер 902, демультимплексор 903, декодер 904, блок

905 обработки видеосигнала, блок 906 отображения, блок 907 обработки аудиосигналов, громкоговоритель 908, блок 909 внешнего интерфейса (I/F), блок 910 управления, блок 911 пользовательского интерфейса (I/F) и шину 912.

Тюнер 902 извлекает сигнал желаемого канала из широковещательного сигнала, принятого через антенну 901, и демодулирует извлеченный сигнал. Затем тюнер 902 выводит кодированный битовый поток, полученный в результате демодуляции, в демультимплексор 903. То есть тюнер 902 играет роль секции передачи телевизионного устройства 900, которое принимает кодированный поток, в котором изображения являются кодированными.

Демультимплексор 903 демультимплексирует видеопоток и аудиопоток программы, подлежащей просмотру, из кодированного потока и выводит демультимплексированные потоки в декодер 904. Кроме того, демультимплексор 903 извлекает вспомогательные данные, такие как электронная программа передач (EPG), из кодированного битового потока и подает извлеченные данные в блок 910 управления. Следует отметить, что в случае, когда кодированный битовый поток был скремблирован, демультимплексор 903 может выполнить дескремблирование.

Декодер 904 декодирует видеопоток и аудиопоток, введенные из демультимплексора 903. Затем декодер 904 выводит видеоданные, выработанные в процессе декодирования, в блок 905 обработки видеосигнала. Кроме того, декодер 904 выводит аудиоданные, выработанные в процессе декодирования, в блок 907 обработки аудиосигналов.

Блок 905 обработки видеосигнала воспроизводит видеоданные, введенные из декодера 904, чтобы заставить блок 906 отображения отображать видео. Кроме того, модуль 905 обработки видеосигнала может заставить модуль 906 отображения отображать на экране приложение, полученное через сеть. Кроме того, блок 905 обработки видеосигнала может выполнить дополнительный процесс, например, шумоподавление, над видеоданными в соответствии с настройкой. Более того, блок 905 обработки видеосигнала может вырабатывать изображение графического интерфейса пользователя (GUI), например, меню, кнопки или курсора, и накладывать выработанное изображение на выходное изображение.

Блок 906 отображения приводится в действие с помощью управляющего сигнала, поступающего из блока 905 обработки видеосигнала, и отображает видео или изображение на видеоплоскости устройства отображения (например, жидкокристаллического дисплея, плазменного дисплея, органического светоизлучающего дисплея (OLED) и т.д.).

Блок 907 обработки аудиосигнала выполняет процесс воспроизведения, включающий в себя цифро-аналоговое преобразование и усиление аудиоданных, введенных из декодера 904, и заставляет громкоговоритель 908 выводить звук. Кроме того, блок 907 обработки аудиосигнала может выполнять дополнительный процесс, такой как устранение шумов в аудиоданных.

Блок 909 внешнего интерфейса является интерфейсом для подключения телевизионного устройства 900 к внешнему устройству или сети. Например, видеопоток или аудиопоток, принятый через блок 909 внешнего интерфейса, может быть декодирован декодером 904. Другими словами, блок 909 внешнего интерфейса также играет роль секций передачи телевизионного устройства 900, которое принимает закодированный поток, в котором закодированы изображения.

Блок 910 управления имеет процессор, такой как CPU, и память, такую как RAM и ROM. В памяти хранятся программа, исполняемая CPU, данные программы, данные EPG и данные, полученные через сеть. Программа, хранящаяся в памяти, считывается

и исполняется CPU во время, например, запуска телевизионного устройства 900. CPU управляет операциями телевизионного устройства 900 путем исполнения программы в ответ, например, на сигналы операций, вводимые из секции 911 пользовательского интерфейса.

5 Секция 911 пользовательского интерфейса подсоединена к блоку 910 управления. Секция 911 пользовательского интерфейса включает в себя, например, кнопки и переключатели, с помощью которых пользователь управляет телевизионным устройством 900, блок приема для сигналов дистанционного управления и т.п. Секция 911 пользовательского интерфейса вырабатывает сигнал операции путем обнаружения  
10 операции пользователем через любой вышеупомянутый составляющий элемент и выводит выработанный сигнал операции в блок 910 управления.

Шина 912 соединяет тюнер 902, демультимплексор 903, декодер 904, блок 905 обработки видеосигнала, блок 907 обработки аудиосигналов, блок 909 внешнего интерфейса и блок 910 управления друг с другом.

15 В телевизионном устройстве 900, сконфигурированном таким образом, декодер 904 может также включать в себя функции устройства 100 декодирования изображения, описанного выше. Другими словами, декодер 904 можно выполнить с возможностью декодирования кодированных данных согласно способу, описанному в каждом из  
20 вышеупомянутых вариантов осуществления. Таким образом, телевизионное устройство 900 позволяет получить эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

Кроме того, в телевизионном устройстве 900, сконфигурированном таким образом, блок 905 обработки видеосигнала может иметь возможность кодировать данные изображения, предоставленные из декодера 904, и заставлять полученные кодированные  
25 данные подаваться извне в телевизионное устройство 900 через блок 909 внешнего интерфейса. Кроме того, блок 905 обработки видеосигнала может также включать в себя функции устройства 400 кодирования изображения, описанного выше. Другими словами, блок 905 обработки видеосигнала можно выполнить с возможностью кодирования данных изображения, поступающих из декодера 904, согласно способу,  
30 описанному в каждом из вышеупомянутых вариантов осуществления. При такой компоновке телевизионное устройство 900 становится способным дополнительно уменьшить объем передачи, связанный с передачей (отправкой) параметров. Таким образом, телевизионное устройство 900 позволяет получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1  
35 – фиг.68.

Второй пример применения: мобильный телефон

На фиг.71 показан пример схематичной конфигурации мобильного телефона, к которому применимы вышеописанные варианты осуществления. Мобильный телефон 920 включает в себя антенну 921, блок 922 связи, аудиокодек 923, громкоговоритель  
40 924, микрофон 925, блок 926 камеры, блок 927 обработки изображения, блок 928 мультимплексирования/демультимплексирования, блок 929 записи/воспроизведения, блок 930 отображения, блок 931 управления, блок 932 оперирования и шину 933.

Антенна 921 подсоединена к блоку 922 связи. Громкоговоритель 924 и микрофон 925 подсоединены к аудиокодеку 923. Блок 932 оперирования соединен с блоком 931  
45 управления. Шина 933 соединяет между собой блок 922 связи, аудиокодек 923, блок 926 камеры, блок 927 обработки изображения, блок 928 мультимплексирования/демультимплексирования, блок 929 записи/воспроизведения, блок 930 отображения и блок 931 управления.

Мобильный телефон 920 выполняет действия, такие как передача/прием аудиосигнала, передача/прием сообщений электронной почты или данных изображения, захват изображения и запись данных в различных режимах работы, включая режим аудиовызова, режим передачи данных, режим фотографирования и режим видеофона.

5 В режиме аудиовызова аналоговый аудиосигнал, выработанный микрофоном 925, подается в аудиокодек 923. Аудиокодек 923 затем преобразует аналоговый аудиосигнал в аудиоданные, выполняет аналого-цифровое преобразование над преобразованными аудиоданными и сжимает данные. Затем аудиокодек 923 выводит сжатые аудиоданные в блок 922 связи. Блок 922 связи кодирует и модулирует аудиоданные для выработки  
10 сигнала передачи. Затем блок 922 связи передает выработанный сигнал передачи в базовую станцию (не показана) через антенну 921. Кроме того, блок 922 связи усиливает радиосигнал, принятый через антенну 921, выполняет преобразование частоты и получает принятый сигнал. После того блок 922 связи демодулирует и декодирует принятый сигнал для выработки аудиоданных и выводит выработанные аудиоданные  
15 в аудиокодек 923. Аудиокодек 923 расширяет аудиоданные, выполняет цифро-аналоговое преобразование над данными и вырабатывает аналоговый аудиосигнал. Затем аудиокодек 923 подает выработанный аудиосигнал в громкоговоритель 924, чтобы заставить его выводить звук.

В режиме передачи данных, например, блок 931 управления вырабатывает  
20 символьные данные, конфигурирующие электронную почту, в соответствии с операцией пользователя, обнаруженной с помощью блока 932 оперирования. Блок 931 управления дополнительно отображает символы на экране блока 930 отображения. Более того, блок 931 управления вырабатывает данные электронной почты в соответствии с инструкцией для их отправки, полученные от пользователя, через блок 932 оперирования  
25 и выводит выработанные данные электронной почты в блок 922 связи. Блок 922 связи кодирует и модулирует данные электронной почты для выработки сигнала передачи. Затем блок 922 связи передает выработанный сигнал передачи в базовую станцию (не показана) через антенну 921. Блок 922 связи дополнительно усиливает радиосигнал, принятый через антенну 921, выполняет преобразование частоты и получает принятый  
30 сигнал. После этого блок 922 связи демодулирует и декодирует принятый сигнал, восстанавливает данные электронной почты и выводит восстановленные данные электронной почты в блок 931 управления. Блок 931 управления отображает содержание электронной почты на экране блока 930 отображения, а также подает данные электронной почты на носитель информации блока 929 записи/воспроизведения, чтобы  
35 обеспечить запись данных на носитель.

Блок 929 записи/воспроизведения включает в себя произвольный носитель информации, который выполнен с возможностью считывания и записи. Например, носитель информации может быть встроенным носителем информации, таким как RAM или флэш-память, или может быть устанавливаемым извне носителем информации,  
40 таким как жесткий диск, магнитный диск, магнитооптический диск, оптический диск, USB-память или карта памяти.

В режиме фотографии, например, блок 926 камеры формирует изображение объекта для выработки данных изображения и выводит выработанные данные изображения в блок 927 обработки изображения. Блок 927 обработки изображения кодирует данные  
45 изображения, введенные из блока 926 камеры, и подает кодированный поток на носитель информации блока 929 записи/воспроизведения, чтобы обеспечить запись кодированного потока на носитель.

Кроме того, в режиме отображения изображения блок 929 записи/воспроизведения

считывает кодированный поток, записанный на носителе информации, и выводит в блок 927 обработки изображения. Блок 927 обработки изображения декодирует кодированный поток, введенный из блока 929 записи/воспроизведения, подает данные изображения в блок 930 отображения и обеспечивает отображение изображения.

5 В режиме видеотелефона, например, блок 928 мультиплексирования/демультиплексирования мультиплексирует видеопоток, кодированный блоком 927 обработки изображения, и аудиопоток, введенный из аудиокодека 923, и выводит мультиплексированный поток в блок 922 связи. Блок 922 связи кодирует и модулирует поток для генерирования сигнала передачи. Затем блок 922 связи передает  
10 выработанный сигнал передачи в базовую станцию (не показана) через антенну 921. Более того, блок 922 связи усиливает радиосигнал, принятый через антенну 921, выполняет преобразование частоты и получает принятый сигнал. Сигнал передачи и принятый сигнал могут включать в себя кодированный битовый поток. Таким образом, блок 922 связи демодулирует и декодирует принятый сигнал для восстановления потока  
15 и выводит восстановленный поток в блок 928 мультиплексирования/демультиплексирования. Блок 928 мультиплексирования/демультиплексирования демультиплексирует видеопоток и аудиопоток из входного потока и выводит видеопоток и аудиопоток в блок 927 обработки изображения и аудиокодек 923, соответственно. Блок 927 обработки изображения декодирует видеопоток для выработки видеоданных.  
20 Затем видеоданные подаются в блок 930 отображения, который отображает ряд изображений. Аудиокодек 923 расширяет и выполняет цифро-аналоговое преобразование над аудиопотоком для выработки аналогового аудиосигнала. Затем аудиокодек 923 подает выработанный аудиосигнал в громкоговоритель 924, чтобы заставить его выводить звук.

25 В мобильном телефоне 920, сконфигурированном таким образом, блок 927 обработки изображения может включать в себя функции, например, устройства 400 кодирования изображения, описанного выше. Другими словами, блок 927 обработки изображения можно выполнить с возможностью кодирования данных изображения согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом,  
30 мобильный телефон 920 позволяет получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

Кроме того, в мобильном телефоне 920, сконфигурированном таким образом, блок 927 обработки изображения может включать в себя функции, например, устройства  
35 100 декодирования изображения, описанного выше. Другими словами, блок 927 обработки изображения можно выполнить с возможностью декодирования кодированных данных согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом, мобильный телефон 920 позволяет получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных  
40 выше со ссылкой на фиг.1 – фиг.68.

40 Третий пример применения: устройство записи/воспроизведения

На фиг.72 показан пример схематичной конфигурации устройства записи/воспроизведения, к которому применимы вышеописанные варианты осуществления. Устройство 940 записи/воспроизведения кодирует аудиоданные и видеоданные  
45 принимаемой широковещательной программы и записывает данные на носитель записи, например. Устройство 940 записи/воспроизведения может также кодировать аудиоданные и видеоданные, полученные из другого устройства, и записывать данные на носитель записи, например. Устройство 940 записи/воспроизведения воспроизводит данные, записанные на носителе записи, используя монитор и громкоговоритель,

например, в ответ на инструкцию пользователя. В этом случае устройство 940 записи/воспроизведения декодирует аудиоданные и видеоданные.

Устройство 940 записи/воспроизведения включает в себя тюнер 941, блок 942 внешнего интерфейса, кодер 943, накопитель 944 на жестком диске (HDD), дисковый накопитель 945, селектор 946, декодер 947, блок 948 отображения на экране (OSD), блок 949 управления и блок 950 пользовательского интерфейса.

Тюнер 941 извлекает сигнал желаемого канала из широковещательного сигнала, принятого через антенну (не показана), и демодулирует извлеченный сигнал. Затем тюнер 941 выводит кодированный битовый поток, полученный в результате демодуляции, в селектор 946. То есть тюнер 941 играет роль блока передачи в устройстве 940 записи/воспроизведения.

Блок 942 внешнего интерфейса представляет собой интерфейс, который соединяет устройство 940 записи/воспроизведения с внешним устройством или сетью. Блок 942 внешнего интерфейса может представлять собой, например, интерфейс 1394 Института инженеров по электротехнике и радиоэлектронике (IEEE), сетевой интерфейс, USB-интерфейс или интерфейс флэш-памяти. Видеоданные и аудиоданные, принятые через блок 942 внешнего интерфейса, вводятся, например, в кодер 943, например. То есть блок 942 внешнего интерфейса играет роль блока передачи в устройстве 940 записи/воспроизведения.

Кодер 943 кодирует видеоданные и аудиоданные в случае, когда видеоданные и аудиоданные, введенные из блока 942 внешнего интерфейса, не являются кодированными. После этого кодер 943 выводит кодированный битовый поток в селектор 946.

Блок 944 HDD записывает на внутренний жесткий диск кодированный битовый поток, на котором сжимаются данные содержания, такие как видео и аудио, различные программы и другие данные. Блок 944 HDD считывает эти данные с жесткого диска при воспроизведении видео и аудио.

Дисковый накопитель 945 записывает и считывает данные на/с носителя записи, вставленного в дисковый накопитель. Носитель записи, вставленный в дисковый накопитель 945, может быть, например, цифровым универсальным диском (DVD) (таким как DVD для видео (DVD-Video), оперативным запоминающим устройством на основе DVD (DVD-RAM), записываемым DVD (DVD-R), перезаписываемым DVD (DVD-RW), записываемым DVD+ (DVD+R) или перезаписываемым DVD+ (DVD+RW)) или диском Blu-ray (зарегистрированный торговый знак).

При записи видео и аудио селектор 946 выбирает кодированный битовый поток, введенный из тюнера 941 или кодера 943, и выводит выбранный кодированный битовый поток в блок 944 HDD или дисковый накопитель 945. С другой стороны, при воспроизведении видео и аудио селектор 946 выводит кодированный битовый поток из блока 944 HDD или дискового накопителя 945 в декодер 947.

Декодер 947 декодирует кодированный битовый поток для выработки видеоданных и аудиоданных. Затем декодер 947 выводит выработанные видеоданные в блок 948 OSD и подает выработанные аудиоданные во внешний громкоговоритель.

Блок 948 OSD воспроизводит видеоданные, введенные из декодера 947, и отображает видео. Блок 948 OSD может также накладывать на отображенное видео изображение GUI, такое как меню, кнопки или курсор.

Блок 949 управления включает в себя процессор, такой как CPU, и память, такую как RAM и ROM. В памяти хранится программа, исполняемая CPU, а также данные программы. Программа, хранящаяся в памяти, считывается CPU при запуске устройства

940 записи/воспроизведения и исполняется, например. При выполнении программы CPU управляет работой устройства 940 записи/воспроизведения в соответствии с сигналом операции, который вводится, например, из блока 950 пользовательского интерфейса.

5 Блок 950 пользовательского интерфейса подключен к блоку 949 управления. Блок 950 пользовательского интерфейса включает в себя кнопку и переключатель для пользователя, который эксплуатирует устройство 940 записи/воспроизведения, а также приемную часть, которая принимает, например, сигнал дистанционного управления. Блок 950 пользовательского интерфейса обнаруживает операцию пользователя  
10 посредством этих компонентов для выработки сигнала операции, и выводит выработанный сигнал операции в блок 949 управления.

В устройстве 940 записи/воспроизведения, сконфигурированном таким образом, кодер 943 может включать в себя функции, например, устройства 400 кодирования изображения, описанного выше. Другими словами, кодер 943 можно выполнить с  
15 возможностью кодирования данных изображения согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом, устройство 940 записи/воспроизведения позволяет получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

Кроме того, в устройстве 940 записи/воспроизведения, сконфигурированном таким образом, декодер 947 может включать в себя функции, например, устройства 100 декодирования изображения, описанного выше. Другими словами, декодер 947 можно  
20 выполнить с возможностью декодирования кодированных данных согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом, устройство 940 записи/воспроизведения позволяет получать эффекты, аналогичные  
25 эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

Четвертый пример применения: устройство формирования изображения

На фиг.73 показан пример схематичной конфигурации устройства формирования изображения, к которому применимы вышеописанные варианты осуществления.  
30 Устройство 960 формирования изображения формирует изображение объекта для выработки изображения, кодирует данные изображения и записывает данные на носитель записи.

Устройство 960 формирования изображения включает в себя оптический блок 961, блок 962 формирования изображения, блок 963 обработки сигналов, блок 964 обработки  
35 изображения, блок 965 отображения, блок 966 внешнего интерфейса, блок 967 памяти, медиа-накопитель 968, блок 969 OSD, блок 970 управления, блок 971 пользовательского интерфейса и шину 972.

Оптический блок 961 подсоединен к блоку 962 формирования изображения. Блок 962 формирования изображения подсоединен к блоку 963 обработки сигналов. Блок  
40 965 отображения подсоединен к блоку 964 обработки изображения. Блок 971 пользовательского интерфейса подсоединен к блоку 970 управления. Шина 972 соединяет между собой блок 964 обработки изображения, блок 966 внешнего интерфейса, блок 967 памяти, медиа-накопитель 968, блок 969 OSD и блок 970 управления.

Оптический блок 961 включает в себя фокусирующую линзу и диафрагменный  
45 механизм. Оптический блок 961 формирует оптическое изображение объекта в плоскости изображения блока 962 формирования изображения. Блок 962 формирования изображения включает в себя светочувствительную матрицу, такую как прибор с зарядовой связью (CCD) или комплементарный металлооксидный полупроводник

(CMOS), и выполняет фотоэлектрическое преобразование для преобразования оптического изображения, сформированного в плоскости изображения, в сигнал изображения в виде электрического сигнала. Затем блок 962 формирования изображения выводит сигнал изображения в блок 963 обработки сигналов.

5 Блок 963 обработки сигналов выполняет различные процессы обработки сигналов камеры, такие как коленная коррекция, гамма-коррекция и цветовая коррекция сигнала изображения, подаваемого из блока 962 формирования изображения. Блок 963 обработки сигналов выводит данные изображения, в отношении которых были выполнены процессы обработки сигналов камеры, в блок 964 обработки изображения.

10 Блок 964 обработки изображения кодирует данные изображения, поступающие из блока 963 обработки сигналов, и вырабатывает кодированные данные. Затем блок 964 обработки изображения выводит выработанные кодированные данные в блок 966 внешнего интерфейса или медиа-накопитель 968. Блок 964 обработки изображения также декодирует кодированные данные, поступающие из блока 966 внешнего  
15 интерфейса или медиа-накопителя 968, для выработки данных изображения. Затем блок 964 обработки изображения выводит выработанные данные изображения в блок 965 отображения. Более того, блок 964 обработки изображения может выводить в блок 965 отображения данные изображения, введенные из блока 963 обработки сигнала, для того, чтобы заставить блок 965 отображения отображать изображение. Кроме того,  
20 блок 964 обработки изображения может накладывать отображаемые данные, полученные из блока 969 OSD, на изображение, которое выводится на экран блока 965 отображения.

Блок 969 OSD генерирует изображение, такое как GUI, меню, кнопки или курсор, и выводит сгенерированное изображение в блок 964 обработки изображения.

25 Блок 966 внешнего интерфейса сконфигурирован, например, в виде USB терминала ввода/вывода. Блок 966 внешнего интерфейса соединяет устройство 960 формирования изображения с принтером, например, в случае печати изображения. Более того, при необходимости накопитель подключается к блоку 966 внешнего интерфейса. Съёмный носитель, такой как магнитный диск или оптический диск, вставляется в накопитель,  
30 например, таким образом, чтобы можно было установить программу, считываемую со съёмного носителя, в устройстве 960 формирования изображения. Блок 966 внешнего интерфейса может быть также сконфигурирован как сетевой интерфейс, который подключен к сети, такой как LAN или Интернет. То есть блок 966 внешнего интерфейса играет роль блока передачи в устройстве 960 формирования изображения.

35 Носитель записи, вставленный в медиа-накопитель 968, может быть произвольным съёмным носителем, который имеет возможность считывания и записи, таким как магнитный диск, магнитооптический диск, оптический диск или полупроводниковая память. Кроме того, носитель записи, может вставляться в медиа-накопитель 968 фиксированным образом с тем, чтобы был сконфигурирован, например, переносимый  
40 блок хранения данных, такой как встроенный накопитель на жестком диске или твердотельный накопитель (SSD).

Блок 970 управления включает в себя процессор, такой как CPU, и память, такую как RAM и ROM. В памяти хранятся программа, исполняемая CPU, а также данные программы. Программа, хранящаяся в памяти, считывается CPU при запуске устройства  
45 960 формирования изображения и затем выполняется. При выполнении программы CPU управляет работой устройства 960 формирования изображения в соответствии с сигналом операции, который вводится, например, из блока 971 пользовательского интерфейса.

Блок 971 пользовательского интерфейса подсоединен к блоку 970 управления. Блок 971 пользовательского интерфейса включает в себя кнопки и переключатели для пользователя, чтобы управлять, например, устройством 960 формирования изображения. Блок 971 пользовательского интерфейса обнаруживает операцию пользователя через  
 5 эти компоненты для того, чтобы выработать сигнал операции, и выводит выработанный сигнал операции в блок 970 управления.

В устройстве 960 формирования изображения, сконфигурированном таким образом, блок 964 обработки изображения может включать в себя, например, функции устройства 400 кодирования изображения, описанные выше. Другими словами, блок 964 обработки  
 10 изображения можно выполнить с возможностью кодирования данных изображения согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом, устройство 960 формирования изображения позволяет получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

Кроме того, в устройстве 960 формирования изображения, сконфигурированном таким образом, блок 964 обработки изображения может включать в себя, например, функции устройства 100 декодирования изображения, описанные выше. Другими  
 15 словами, блок 964 обработки изображения можно выполнить с возможностью декодирования кодированных данных согласно способу, описанному в каждом из вышеупомянутых вариантов осуществления. Таким образом, устройство 960 формирования изображения позволяет получать эффекты, аналогичные эффектам  
 20 каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1 – фиг.68.

#### Пятый пример применения: видеонабор

В дополнение к этому, настоящая технология может быть также реализована в  
 25 конфигурацией любого вида, установленной в любом устройстве или устройстве, включенном в систему, таком как процессор, представленный в виде микросхемы с высоким уровнем интеграции (LSI) или т.п., модуль, который использует многочисленные процессоры или т.п., блок, который использует многочисленные модули или т.п., набор, который дополнительно добавляет другие функции в блок (то есть в конфигурацию  
 30 части устройства) или т.п. На фиг.74 показан один пример схематичной конфигурации видеонабора, в котором применяется настоящая технология.

В последнее время электронные устройства становятся все более многофункциональными, и при разработке и изготовлении таких электронных устройств, в случае реализации их частичной конфигурации для продажи, предложения или т.п.,  
 35 стало обычным делом не только осуществлять реализацию в виде конфигурации, которая включает в себя одну функцию, но также объединять многочисленные конфигурации, которые включают в себя связанные между собой функции и осуществляют реализацию в виде единого набора, включающего в себя многочисленные функции.

Видеонабор 1300, показанный на фиг.74, имеет такую многофункциональную  
 40 конфигурацию и представляет собой объединение устройства, которое включает в себя функции, связанные с кодированием и декодированием изображения (либо с одним, либо и с тем и с другим), с устройством, которое включает в себя другие функции, связанные с такими функциями.

Как показано на фиг.74, видеонабор 1300 включает в себя группу модулей, такую  
 45 как видеомодуль 1311, внешняя память 1312, модуль 1313 управления электропитанием и модуль 1314 внешнего интерфейса, и устройство, которое включает в себя связанные функции, такие как блок 1321 подключений, камера 1322 и датчик 1323.

Модуль представляет собой часть, которая объединяет несколько взаимосвязанных

частичных функций в единую функцию. Конкретная физическая конфигурация может представлять собой любую конфигурацию, но, например, можно размещать и объединять многочисленные процессоры с соответствующими функциями, элементами электронных схем, такими как резисторы и конденсаторы, другими устройствами и т.п. на печатной плате или т.п. Можно также объединить модуль с другим модулем, процессором или т.п. для создания нового модуля.

В случае примера, показанного на фиг.74, видеомодуль 1311 является комбинацией конфигураций, которые включают в себя функции, которые относятся к обработке изображений, и включает в себя процессор приложений, видеопроцессор, широкополосный модем 1333 и РЧ модуль 1334.

Процессор представляет собой объединение конфигураций, имеющих заранее определенные функции, в полупроводниковую микросхему, такую как система на одной микросхеме (SoC), и может также обозначаться, например, как микросхема с большой степенью интеграции (LSI) или т.п. Конфигурации, имеющие заранее определенные функции, могут представлять собой логические схемы (аппаратные конфигурации), но могут также представлять собой CPU, ROM, RAM и т.п., а также программу, исполняемую с их использованием (программные конфигурации), и могут также представлять собой их сочетание. Например, процессор может включать в себя логические схемы и CPU, ROM, RAM и т.п. и может быть выполнен с возможностью реализации подмножества функций с логическими схемами (аппаратными конфигурациями) при реализации других функций с программами (программными конфигурациями), исполняемыми на CPU.

Процессор 1331 приложений, показанный на фиг.74, представляет собой процессор, который исполняет приложение, связанное с обработкой изображения. Чтобы реализовать заданную функцию, приложение, исполняемое в процессоре 1331 приложений, может не только исполнять вычислительную обработку, но при необходимости может также управлять, например, конфигурациями внутри и снаружи видеомодуля 1311, такого как видеопроцессор 1332.

Видеопроцессор 1332 представляет собой процессор, который включает в себя функции, которые относятся к кодированию/декодированию изображений (либо к одному, либо и к тому и другому).

Широкополосный модем 1333 выполняет цифровую модуляцию и т.п. для преобразования данных (цифрового сигнала), передаваемых посредством проводной или беспроводной (или и то и другое) широкополосной связи, выполняемой по широкополосному соединению, такому как Интернет или телефонная сеть общего пользования, в аналоговый сигнал, а также выполняет демодуляцию для преобразования аналогового сигнала, принятого с помощью такой широкополосной связи, в данные (цифровой сигнал). Широкополосный модем 1333 обрабатывает любой вид информации, такой, например, как данные изображения, обработанные видеопроцессором 1332, в поток, в котором данные изображения являются кодированными, прикладные программы и настройки данных.

РЧ модуль 1334 представляет собой модуль, который выполняет процессы преобразования частоты, модуляции/демодуляции, усиления, фильтрации и т.п. над радиочастотными (РЧ) сигналами, переданными и принятыми через антенну. Например, РЧ модуль 1334 вырабатывает РЧ сигнал путем выполнения преобразования частоты и т.п. над видеосигналом, выработанным широкополосным модемом 1333. Кроме того, например, РЧ модуль 1334 вырабатывает видеосигнал путем выполнения преобразования частоты и т.п. над РЧ сигналом, принятым через модуль 1314 внешнего

интерфейса.

Следует отметить, что, как показано пунктирной линией 1341 на фиг.74, процессор 1331 приложений и видеопроцессор 1332 могут быть также унифицированы и сконфигурированы как один процессор.

5 Внешняя память 1312 представляет собой модуль, выполненный снаружи видеомодуля 1311, который включает в себя запоминающее устройство, используемое видеомодулем 1311. Запоминающее устройство внешней памяти 1312 может быть реализовано с помощью физической конфигурации любого вида, но так как запоминающее устройство, как правило, используется для хранения больших объемов данных, таких как данные  
10 изображения в блоках кадров, желательно реализовать запоминающее устройство с относительно недорогой полупроводниковой памятью большой емкости, такой, например, как динамическое оперативное запоминающее устройство (DRAM).

Модуль 1313 управления электропитанием управляет и контролирует подачей питания на видеомодуль 1311 (каждую конфигурацию внутри видеомодуля 1311).

15 Модуль 1314 внешнего интерфейса представляет собой модуль, который выполняет функцию внешнего интерфейса (схема приемопередающего порта на стороне антенны) в РЧ модуле 1334. Как показано на фиг.74, модуль 1314 внешнего интерфейса включает в себя, например, антенный блок 1351, фильтр 1352 и блок 1353 усиления.

Антенный блок 1351 включает в себя антенну, которая передает и принимает сигналы  
20 беспроводной связи и их периферийную конфигурацию. Антенный блок 1351 передает сигнал, подаваемый из блока 1353 усиления в виде сигнала беспроводной связи, и подает принятый сигнал беспроводной связи в фильтр 1352 в виде электрического сигнала (РЧ сигнала). Фильтр 1352 выполняет процесс фильтрации и т.п. над РЧ сигналом, принятым через антенный блок 1351, и подает обработанный РЧ сигнал в РЧ модуль 1334. Блок  
25 1353 усиления усиливает и подает РЧ сигнал, поданный из РЧ модуля 1334, в антенный блок 1351.

Блок 1321 подключений представляет собой модуль, который включает в себя функции, которые относятся к внешним подключениям. Физическая конфигурация блока 1321 подключений может иметь любую конфигурацию. Например, блок 1321  
30 подключений включает в себя конфигурацию, имеющую функцию связи, отличную от стандарта связи, поддерживаемого широкополосным модемом 1333, внешним терминалом ввода/вывода и т.п.

Например, блок 1321 подключений может включать в себя модуль, имеющий функцию связи, соответствующую стандарту беспроводной связи, такому как стандарт Bluetooth  
35 (зарегистрированный торговый знак), стандарт IEEE 802.11 (например, стандарт беспроводной связи Wireless Fidelity (Wi-Fi) (зарегистрированный торговый знак))), стандарт беспроводной связи ближнего радиуса действия (NFC) или стандарт Ассоциации по средствам передачи данных в инфракрасном диапазоне (IrDA), и антенну или т.п., которая передает и принимает сигналы, соответствующие стандарту. Кроме того,  
40 например, блок 1321 подключений может включать в себя модуль, имеющий функцию связи, соответствующую функции беспроводной связи, такой как универсальная последовательная шина (USB) или высококачественный мультимедийный интерфейс (HDMI) (зарегистрированный торговый знак) и порт, соответствующий стандарту. Кроме того, блок 1321 подключений может включать в себя, например, функцию  
45 передачи данных (сигнала) другого вида, например, аналоговый терминал ввода/вывода.

Следует отметить, что блок 1321 подключений может включать в себя устройство пункта назначения передачи данных (сигнала). Например, блок 1321 подключений

может включать в себя накопитель (не только накопитель для съемных носителей, но также и жесткий диск, твердотельный накопитель (SSD), запоминающее устройство, подключаемое к сети (NAS), и т.п.), который считывает и записывает данные по отношению к носителю записи, такому как магнитный диск, оптический диск, магнитооптический диск или полупроводниковая память. Кроме того, блок 1321 подключений может включать в себя устройства (такие как монитор и громкоговоритель), которые выводят изображения и звук.

Камера 1322 представляет собой модуль, который имеет функцию формирования изображения объекта и получения данных изображения объекта. Данные изображения, полученные в результате формирования изображения камерой 1322, подаются, например, в видеопроцессор 1332 и кодируются.

Датчик 1323 представляет собой модуль, имеющий функции датчика любого типа, такого, например, как датчик звука, ультразвуковой датчик, датчик света, датчик освещенности, инфракрасный датчик, датчик изображения, датчик вращения, датчик угла, датчик угловой скорости, датчик скорости, датчик ускорения, датчик наклона, датчик магнитного поля, датчик удара или датчик температуры. Данные, обнаруженные датчиком 1323, подаются, например, в процессор 1331 приложений и используются приложением и т.п.

Конфигурации, описанные выше в виде модуля, могут быть реализованы в виде процессора, и, наоборот, конфигурации, описанные в виде процессора, могут быть также реализованы в виде модуля.

В видеонаборе 1300 с конфигурацией, подобной выше, настоящая технология может быть применена к видеопроцессору 1332, как будет описано позже. Следовательно, видеонабор 1300 может быть выполнен в виде набора, в котором применяется настоящая технология.

Примерная конфигурация видеопроцессора

На фиг.75 показан один пример схематичной конфигурации видеопроцессора 1332 (фиг.74), в котором применяется настоящая технология.

В случае примера, показанного на фиг.75, видеопроцессор 1332 включают в себя функцию приема ввода видеосигнала и аудиосигнала и кодирования этих сигналов в соответствии с заданным способом, и функцию декодирования кодированных видеоданных и аудиоданных, и воспроизведение и вывод видеосигнала и аудиосигнала.

Как показано на фиг.75, видеопроцессор 1332 включают в себя блок 1401 обработки входного видеосигнала, первый блок 1402 увеличения/уменьшения изображения, второй блок 1403 увеличения/уменьшения изображения, блок 1404 обработки выходного видеосигнала, память 1405 кадров и блок 1406 управления памятью. Кроме того, видеопроцессор 1332 включает в себя механизм 1407 кодирования/декодирования, буферы 1408А и 1408В видео элементарных потоков (ES) и буферы 1409А и 1409В аудио ES. Кроме того, видеопроцессор 1332 включает в себя аудиокодер 1410, аудиодекодер 1411, мультиплексор (MUX) 1412, демультимплексор (DMUX) 1413 и буфер 1414 потока.

Блок 1401 обработки входного видеосигнала получает входной видеосигнал, например, из блока 1321 подключений (фиг.74) и т.п., и преобразует видеосигнал в данные цифрового изображения. Первый блок 1402 увеличения/уменьшения изображения выполняет преобразование формата, процесс увеличения/уменьшения изображения и т.п. над данными изображения. Второй блок 1403 увеличения/уменьшения изображения выполняет процесс увеличения или уменьшения изображения в соответствии с форматом в пункте назначения, в который должен выводиться формат преобразования через блок 1404 обработки выходного видеосигнала, и процесс увеличения/уменьшения

изображения, аналогичный процессу, выполняемому в первом блоке 1402 увеличения/уменьшения изображения и т.п., над данными изображения. Блок 1404 обработки выходного видеосигнала выполняет преобразование формата, преобразование в аналоговый сигнал и т.п. над данными изображения и выводит результат в блок 1321

5 подключений, например, в виде воспроизводимого видеосигнала.

Память 1405 кадров представляет собой память для данных изображения, совместно используемых блоком 1401 обработки входного видеосигнала, первым блоком 1402 увеличения/уменьшения изображения, вторым блоком 1403 увеличения/уменьшения изображения, блоком 1404 обработки выходного видеосигнала и механизмом 1407

10 кодирования/декодирования. Память 1405 кадров реализована в виде полупроводниковой памяти, такой, например, как DRAM.

Блок 1406 управления памятью принимает сигнал синхронизации из механизма 1407 кодирования/декодирования и управляет доступом к, записью в и считыванием из памяти 1405 кадров в соответствии с графиком доступа к памяти 1405 кадров,

15 записанным в таблице 1406А управления доступом. Таблица 1406А управления доступом обновляется блоком 1406 управления памятью в соответствии с процессами, исполняемыми механизмом 1407 кодирования/декодирования, первым блоком 1402 увеличения/уменьшения изображения, вторым блоком 1403 увеличения/уменьшения изображения и т.п.

Механизм 1407 кодирования/декодирования выполняет процесс кодирования данных изображения, а также процесс декодирования видеопотока, который представляет собой данные, в которых данные изображения являются кодированными. Например, механизм 1407 кодирования/декодирования кодирует данные изображения, считанные из памяти 1405 кадров, и последовательно записывает кодированные данные в буфере

25 1408А видео ES в виде видеопотока. Кроме того, например, механизм 1407 кодирования/декодирования последовательно считывает и декодирует видеопоток из буфера 1409В видео ES и записывает декодированные данные в памяти 1405 кадров в виде данных изображения. Во время этого кодирования и декодирования механизм 1407 кодирования/декодирования использует память 1405 кадров в качестве рабочей области. Кроме того,

30 механизм 1407 кодирования/декодирования выводит сигнал синхронизации в блок 1406 управления памятью при таймировании начала процесса для каждого макроблока, например.

Буфер 1408А видео ES буферизует и подает видеопоток, выработанный механизмом 1407 кодирования/декодирования, в мультиплексор (MUX) 1412. Буфер 1408В видео

35 ES буферизует и подает видеопоток, поданный из демультимплексора (DMUX) 1413, в механизм 1407 кодирования/декодирования.

Буфер 1409А аудио ES буферизует и подает аудиопоток, выработанный аудиокодером 1410, в мультиплексор (MUX) 1412. Буфер 1409В аудио ES буферизует и подает аудиопоток, поданный из демультимплексора (DMUX) 1413, в аудиодекодер 1411.

40 Аудиокодер 1410, например, преобразует цифровым образом аудиосигнал, введенный, например, из блока 1321 подключений или т.п., и кодирует аудиосигнал согласно заданному способу, такому, например, как способ MPEG Аудио или способ AudioCode номер 3 (AC3). Аудиокодер 1410 последовательно записывает аудиопоток, который представляет собой данные, в которых аудиосигнал является кодированным, в буфер

45 1409А аудио ES. Аудиодекодер 1411 декодирует аудиопоток, поданный из буфера 1409В аудио ES, выполняет преобразование в аналоговый сигнал и т.п., например, и подает результат в блок 1321 подключений и т.п., например, в виде воспроизводимого аудиосигнала.

Мультиплексор (MUX) 1412 мультиплексирует видеопоток и аудиопоток. Способ мультиплексирования (то есть формат битового потока, выработанного в результате мультиплексирования) может быть любым способом. В дополнение к этому, во время этого мультиплексирования мультиплексор (MUX) 1412 позволяет также добавлять заданную информацию заголовка или т.п. в битовый поток. Другими словами, мультиплексор (MUX) 1412 может преобразовывать формат потоков путем мультиплексирования. Например, путем мультиплексирования видеопотока и аудиопотока мультиплексор (MUX) 1412 преобразует потоки в транспортный поток, который представляет собой битовый поток в формате для передачи. Кроме того, например, путем мультиплексирования видеопотока и аудиопотока мультиплексор (MUX) 1412 преобразует потоки в данные (данные файла) в формате файла для записи.

Демультимплексор (DMUX) 1413 демультимплексирует битовый поток, в котором видеопоток и аудиопоток являются мультиплексированными согласно способу, соответствующему мультиплексированию мультиплексором (MUX) 1412. Другими словами, демультимплексор (DMUX) 1413 извлекает видеопоток и аудиопоток (разделяет видеопоток и аудиопоток) из битового потока, считанного из буфера 1414 потока. Другими словами, демультимплексор (DMUX) 1413 может преобразовывать формат потока путем демультимплексирования (обратного преобразования по отношению к преобразованию, выполняемому мультиплексором (MUX) 1412). Например, демультимплексор (DMUX) 1413 имеет возможность получать транспортный поток, подаваемый из блока 1321 подключений, широкополосного модема 1333 или т.п., например, через буфер 1414 потока, и за счет демультимплексирования имеет возможность преобразовывать транспортный поток в видеопоток и аудиопоток. Кроме того, например, демультимплексор (DMUX) 1413 имеет возможность получать данные файла, считанные из любых носителей записи различных типов с помощью блока 1321 подключений, например, через буфер 1414 потока, и за счет демультимплексирования имеет возможность преобразовывать данные файла в видеопоток и аудиопоток.

Буфер 1414 потока буферизует битовый поток. Например, буфер 1414 потока буферизует транспортный поток, поданный из мультиплексора (MUX) 1412, и при заданном таймировании или на основе внешнего запроса и т.п. подает транспортный поток, например, в блок 1321 подключений, широкополосный модем 1333 и т.п.

Кроме того, например, буфер 1414 потока буферизует данные файла, поданные из мультиплексора (MUX) 1412, и, при заданном таймировании или на основе внешнего запроса и т.п., подает данные файла в блок 1321 подключений и т.п., например, и обеспечивает запись данных файла на любых носителях записи различных типов.

Кроме того, буфер 1414 потока буферизует транспортный поток, полученный через блок 1321 подключений, широкополосный модем 1333 и т.п., например, и, при заданном таймировании или на основе внешнего запроса и т.п., подает транспортный поток в демультимплексор (DMUX) 1413.

Кроме того, буфер 1414 потока буферизует данные файла, считанные из любых носителей записи различных типов в блоке 1321 подключений и т.п., например, и, при заданном таймировании или на основе внешнего запроса и т.п., подает данные файла в демультимплексор (DMUX) 1413.

Далее будет описан пример работы видеопроцессора 1332 с такой конфигурацией. Например, видеосигнал, введенный в видеопроцессор 1332 из блока 1321 подключений и т.п., преобразуется в цифровые данные изображения заданного формата, такого как формат 4:2:2 Y/Cb/Cr в блоке 1401 обработки входного видеосигнала, и последовательно записывается в памяти 1405 кадров. Данные цифрового изображения считываются в

первом блоке 1402 увеличения/уменьшения изображения или втором блоке 1403 увеличения/уменьшения изображения, подвергаются преобразованию формата в заданный формат, такой как 4:2:0 Y/Cb/Cr или т.п., и процессу увеличения/уменьшения и снова записываются в памяти 1405 кадров. Данные изображения кодируются с помощью механизма 1407 кодирования/декодирования и записываются в буфер 1408A видео ES в виде видеопотока.

Кроме того, аудиосигнал, введенный в видеопроцессор 1332 из блока 1321 подключений и т.п., кодируется аудиокодером 1410 и записывается в буфер 1409A аудио ES в виде аудиопотока.

Видеопоток в буфере 1408A видео ES и аудиопоток в буфере 1409A аудио ES считываются и мультиплексируются с помощью мультиплексора (MUX) 1412 и преобразуются в транспортный поток, данные файла или т.п. Транспортный поток, выработанный мультиплексором (MUX) 1412, буферизуется в буфере 1414 потока и затем выводится во внешнюю сеть через блок 1321 подключений, широкополосный модем 1333 или т.п., например. Кроме того, данные файла, выработанные мультиплексором (MUX) 1412, буферизуются в буфере 1414 потока и затем выводятся в блок 1321 подключений или т.п., например, и записываются на любых носителях записи различных типов.

Кроме того, транспортный поток, вводимый в видеопроцессор 1332 из внешней сети через блок 1321 подключений, широкополосный модем 1333 или т.п., например, буферизуется в буфере 1414 потока, и затем демультимплексируется демультимплексором (DMUX) 1413. Кроме того, данные файла, считанные с любого из носителей записи различных типов в блоке 1321 подключений или т.п., например, и введенные в видеопроцессор 1332, буферизуются в буфере 1414 потока и затем демультимплексируются демультимплексором (DMUX) 1413. Другими словами, транспортный поток или данные файла, поступающие в видеопроцессор 1332, разделяются на видеопоток и аудиопоток демультимплексором (DMUX) 1413.

Аудиопоток подается в аудиодекодер 1411 через буфер 1409B аудио ES и декодируется, и аудиосигнал воспроизводится. Кроме того, видеопоток после записи в буфер 1408B видео ES последовательно считывается и декодируется механизмом 1407 кодирования/декодирования и записывается в память 1405 кадров. Декодированные данные изображения подвергаются процессу увеличения/уменьшения вторым блоком 1403 увеличения/уменьшения изображения и записываются в памяти 1405 кадров. Затем декодированные данные изображения считываются в блоке 1404 обработки выходного видеосигнала, преобразуются по формату в заданный формат, такой как формат 4:2:2 Y/Cb/Cr, дополнительно преобразуются в аналоговый сигнал, и видеосигнал воспроизводится и выводится.

В случае применения настоящей технологии к видеопроцессору 1332, сконфигурированному таким образом, достаточно применить настоящую технологию согласно вариантам осуществления, описанным выше, к механизмам 1407 кодирования/декодирования. Другими словами, например, механизм 1407 кодирования/декодирования может включать в себя функции устройства 400 кодирования изображения или функции устройства 100 декодирования изображения, которые были описаны выше, или и те и другие. При такой компоновке видеопроцессор 1332 способен получать эффекты, аналогичные каждому из вариантов осуществления, описанных выше со ссылкой на фиг. 1-68.

Следует отметить, что в механизме 1407 кодирования/декодирования настоящая технология (то есть функции устройства 400 кодирования изображения, функции

устройства 100 декодирования изображения или одновременно и те и другие) может быть реализована аппаратными средствами, такими как логическая схема и т.п., может быть реализована с помощью программного обеспечения, такого как встроенная программа, или может быть реализована одновременно аппаратными средствами и программным обеспечением.

Другая примерная конфигурация видеопроцессора

На фиг.76 показан другой пример схематичной конфигурации видеопроцессора 1332, в котором применяется настоящая технология. В случае примера, показанного на фиг.76, видеопроцессор 1332 включает в себя функцию кодирования/декодирования видеоданных в соответствии с заданным способом.

Более конкретно, как показано на фиг.76, видеопроцессор 1332 включает в себя блок 1511 управления, интерфейс 1512 дисплея, механизм 1513 отображения, механизм 1514 обработки изображения и внутреннюю память 1515. Кроме того, видеопроцессор 1332 включает в себя механизм 1516 кодека, интерфейс 1517 памяти, мультиплексор/демультиплексор (MUX DMUX) 1518, сетевой интерфейс 1519 и видеоинтерфейс 1520.

Блок 1511 управления управляет работой каждого блока обработки в видеопроцессоре 1332, таком как интерфейс 1512 дисплея, механизм 1513 отображения, механизм 1514 обработки изображения и механизм 1516 кодека.

Как показано на фиг.76, блок 1511 управления включает в себя, например, основное CPU 1531, вспомогательное CPU 1532 и системный контроллер 1533. Основное CPU 1531 исполняет программу и т.п. для управления работой каждого блока обработки в видеопроцессоре 1332. Основное CPU 1531 вырабатывает управляющие сигналы в соответствии с программой и т.п. и подает управляющие сигналы в каждый блок обработки (другими словами, управляет работой каждого блока обработки).

Вспомогательное CPU 1532 выполняет вспомогательную роль по отношению к основному CPU 1531. Например, вспомогательное CPU 1532 выполняет дочерние процессы, подпрограммы и т.п. программы и т.п., исполняемые основным CPU 1531. Системный контроллер 1533 управляет операциями основного CPU 1531 и вспомогательного CPU 1532, например, предписывающими основному CPU 1531 и вспомогательному CPU 1532 исполнять программы.

Интерфейс 1512 дисплея, управляемый блоком 1511 управления, выводит данные изображения, например, в блок 1321 подключений и т.п. Например, интерфейс 1512 дисплея преобразует данные цифрового изображения в аналоговый сигнал и выводит аналоговый сигнал или выводит данные цифрового изображения непосредственно в виде воспроизводимого видеосигнала в устройство контроля и т.п. блока 1321 подключений.

Механизм 1513 отображения, управляемый блоком 1511 управления, выполняет различные процессы преобразования, такие как преобразование формата, преобразование размера и преобразование цветовой гаммы над данными изображения, чтобы соответствовать спецификациям аппаратных средств монитора или т.п., который должен отображать изображение.

Механизм 1514 обработки изображения, управляемый блоком 1511 управления, выполняет заданную обработку изображения над данными изображения, такую как процесс фильтрации, например, для повышения качества изображения.

Внутренняя память 1515 представляет собой память, которая находится внутри видеопроцессора 1332 и совместно используется механизмом 1513 отображения, механизмом 1514 обработки изображения и механизмом 1516 кодека. Например, внутренняя память 1515 используется для обмена данными между механизмом 1513

отображения, механизмом 1514 обработки изображения и механизмом 1516 кодека. Например, внутренняя память 1515 хранит данные, подаваемые из механизма 1513 отображения, механизма 1514 обработки изображения или механизма 1516 кодека и при необходимости (например, в ответ на запрос) подает данные в механизм 1513

отображения, механизм 1514 обработки изображения или механизм 1516 кодека. Внутреннюю память 1515 можно реализовать с помощью запоминающего устройства любого вида, но так как запоминающее устройство, как правило, используется для хранения маленьких объемов данных, такие как данные изображения в единицах блоков, параметров и т.п., желательно реализовать запоминающее устройство с помощью полупроводниковой памяти, которая является относительно маленькой по объему (например, по сравнению с внешней памятью 1312), но имеет высокое быстродействие, такое, например, как статическое оперативное запоминающее устройство (SRAM).

Механизм 1516 кодека выполняет процессы, связанные с кодированием и декодированием данных изображения. Способ кодирования/декодирования, поддерживаемый механизмом 1516 кодека, может представлять собой любой способ, и может существовать один или множество таких способов. Например, механизм 1516 кодека может быть снабжен функцией кодека для многочисленных способов кодирования/декодирования и может быть выполнен с возможностью кодирования или декодирования данных изображения путем выбора среди многочисленных способов.

В примере, показанном на фиг.76, механизм 1516 кодека включает в себя, например, блок 1541 MPEG-2 Video, блок 1542 AVC/H.264, блок 1543 HEVC/H.265, блок 1544 HEVC/H.265 (масштабируемый), блок 1545 HEVC/H.265 (многоакурсный) и блок 1551 MPEG-DASH в виде функциональных блоков обработки, связанной с кодеками.

Блок 1541 MPEG-2 Video является функциональным блоком, который кодирует и декодирует данные изображения согласно способу MPEG-2. Блок 1542 AVC/H.264 является функциональным блоком, который кодирует и декодирует данные изображения согласно способу AVC. Блок 1543 HEVC/H.265 является функциональным блоком, который кодирует и декодирует данные изображения согласно способу HEVC. Блок 1544 HEVC/H.265 (масштабируемый) является функциональным блоком, который кодирует с масштабированием и декодирует с масштабированием данные изображения согласно способу HEVC. Блок 1545 HEVC/H.265 (многоакурсный) является функциональным блоком, который кодирует и декодирует данные многоакурсного изображения согласно способу HEVC.

Блок 1551 MPEG-DASH является функциональным блоком, который передает и принимает данные изображения согласно способу динамической адаптивной потоковой передачи MPEG по HTTP (MPEG-DASH). MPEG-DASH является технологией, в которой используется протокол передачи гипертекста (HTTP) для потоковой передачи видео, одной особенностью которого является то, что соответствующие кодированные данные выбираются и передаются в единицах сегментов из множества наборов кодированных данных, имеющих разное разрешение или тому подобное, подготовленных заранее. Блок 1551 MPEG-DASH выполняет выработку, управление передачей и т.п. потока, соответствующего стандарту, в то время как для кодирования/декодирования данных изображения используются блок 1541 MPEG-2 Video – блок 1545 HEVC/H.265 (многоакурсный).

Интерфейс 1517 памяти представляет собой интерфейс для внешней памяти 1312. Данные, поданные из механизма 1514 обработки изображения и механизма 1516 кодека, подаются во внешнюю память 1312 через интерфейс 1517 памяти. Кроме того, данные, считанные из внешней памяти 1312, поступают в видеопроцессор 1332 (механизм 1514

обработки изображения или механизм 1516 кодека) через интерфейс 1517 памяти.

Мультиплексор/демультиплексор (MUX DMUX) 1518 мультиплексирует и демультиплексирует различные данные, которые относятся к изображению, такие как битовый поток кодированных данных, данные изображения, видеосигнал и т.п. Способ мультиплексирования/демультиплексирования может быть любым способом. Например, при мультиплексировании мультиплексор/демультиплексор (MUX DMUX) 1518 может не только осуществлять сбор многочисленных фрагментов данных в одном фрагменте данных, но также добавлять заданную информацию заголовка и т.п. в данные. Кроме того, при демультиплексировании мультиплексор/демультиплексор (MUX DMUX) 1518 может не только разбивать один фрагмент данных на многочисленные фрагменты данных, но также добавлять заданную информацию заголовка и т.п. в каждый разбитый фрагмент данных. Другими словами, мультиплексор/демультиплексор (MUX DMUX) 1518 может преобразовывать формат данных путем мультиплексирования/демультиплексирования. Например, при мультиплексировании битового потока, мультиплексор/демультиплексор (MUX DMUX) 1518 может преобразовывать битовый поток в транспортный поток, который является битовым потоком в формате для передачи, или в данные в формате файла (данных файла) для записи. Очевидно, что при демультиплексировании также возможно обратное преобразование.

Сетевой интерфейс 1519 представляет собой, например, интерфейс для широкополосного модема 1333, блока 1321 подключений и т.п. Видеоинтерфейс 1520 представляет собой, например, интерфейс для блока 1321 подключений, камеры 1322 и т.п.

Далее будет описан пример работы такого видеопроцессора 1332. Например, когда транспортный поток принимается из внешней сети через блок 1321 подключений, широкополосный модем 1333 и т.п., транспортный поток подается в мультиплексор/демультиплексор (MUX DMUX) 1518 через сетевой интерфейс 1519 и демультиплексируется и декодируется механизмом 1516 кодека. Данные изображения, полученные путем декодирования, выполняемого механизмом 1516 кодека, подвергаются, например, заданной обработке изображения механизмом 1514 обработки изображения, подвергаются заданному преобразованию механизмом 1513 отображения, подаются в блок 1321 подключений или т.п., например, через интерфейс 1512 дисплея, и изображение отображается на мониторе. Кроме того, например, данные изображения, полученные путем декодирования механизма 1516 кодека, повторно кодируются механизмом 1516 кодека, мультиплексируются и преобразуются в данные файла мультиплексором/демультиплексором (MUX DMUX) 1518, выводятся в блок 1321 подключений или т.п., например, через видеоинтерфейс 1520, и записываются на любом из носителей записи различных типов.

Кроме того, например, данные файла кодированных данных, в которых кодируются данные изображения, которые считываются с непоказанного носителя записи блоком 1321 подключений или т.п., подаются в мультиплексор/демультиплексор (MUX DMUX) 1518 через видеоинтерфейс 1520 и демультиплексируются и декодируются с помощью механизма 1516 кодека. Данные изображения, полученные путем декодирования механизмом 1516 кодека, подвергаются заданной обработке изображения механизмом 1514 обработки изображения, подвергаются заданному преобразованию механизмом 1513 отображения, подаются в блок 1321 подключений или т.п., например, через интерфейс 1512 дисплея, и изображение отображается на мониторе. Кроме того, например, данные изображения, полученные путем декодирования механизмом 1516 кодека, повторно кодируются механизмом 1516 кодека, мультиплексируются и

преобразуются в транспортный поток мультиплексором/демультиплексором (MUX/DMUX) 1518, подаются в блок 1321 подключений, широкополосный модем 1333 и т.п., например, через сетевой интерфейс 1519, и передаются в другое устройство, которое не показано.

5 Следует отметить, что обмен данными изображения и другими данными между каждым из блоков обработки в видеопроцессоре 1332 выполняется путем использования внутренней памяти 1515 и внешней памяти 1312, например. Кроме того, модуль 1313 управления электропитанием управляет подачей питания на блок 1511 управления, например.

10 В случае применения настоящей технологии к видеопроцессору 1332, сконфигурированному таким образом, достаточно применить настоящую технологию согласно вариантам осуществления, описанным выше, к механизму 1516 кодека. Другими словами, например, достаточно, чтобы механизм 1516 кодека включал в себя функции устройства 400 кодирования изображения или функции устройства 100 декодирования  
15 изображения, описанные выше, или и то и другое. При такой компоновке видеопроцессор 1332 способен получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1-68.

Следует отметить, что в механизме 1516 кодека настоящая технология (то есть функции устройства 400 кодирования изображения) может быть реализована  
20 аппаратными средствами, такими как логическая схема и т.п., может быть реализована с помощью программного обеспечения, такого как встроенная программа, или может быть реализована одновременно с помощью аппаратных средств и программного обеспечения.

Вышеизложенное иллюстрирует две конфигурации видеопроцессора 1332 в качестве  
25 примеров, но конфигурация видеопроцессора 1332 может иметь любую конфигурацию и может иметь конфигурацию, отличную от двух примеров, описанных выше. Кроме того, видеопроцессор 1332 может быть сконфигурирован в виде одной полупроводниковой интегральной схемы, но может быть также сконфигурирован в виде многочисленных полупроводниковых интегральных схем. Например, возможна  
30 трехмерная ламинированная LSI интегральная схема, в которой ламинированы многочисленные полупроводники. Кроме того, возможна конфигурация, реализованная с помощью многочисленных LSI интегральных схем.

#### Пример применения в устройстве

Видеонабор 1300 может быть встроен в любое из устройств различных типов, которые  
35 обрабатывают данные изображения. Например, видеонабор 1300 может быть включен в телевизионное устройство 900 (фиг.70), мобильный телефон 920 (фиг.71), устройство 940 записи/воспроизведения (фиг.72), устройство 960 формирования изображения (фиг.73) и т.п. Встраивая видеонабор 1300, устройство способно получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со  
40 ссылкой на фиг.1-68.

Следует отметить, что до тех пор, пока видеонабор 1332 включен в устройство, даже часть каждой конфигурации видеонабора 1300, описанного выше, может быть выполнена в виде конфигурации, в которой применяется настоящая технология. Например, можно выполнить только видеопроцессор 1332 в виде видеопроцессора, в  
45 котором применяется настоящая технология. Кроме того, например, процессор, показанный пунктирной линией 1341, как описано выше, видеомодуль 1311 и т.п. можно выполнить в виде процессора, модуля или т.п., которые применяют настоящую технологию. Кроме того, например, видеомодуль 1311, внешняя память 1312, модуль

1313 управления электропитанием и модуль 1314 внешнего интерфейса можно также объединить и выполнить в виде видеоблока 1361, к которому применима настоящая технология. Используя любую из этих конфигураций, можно получить эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1-68.

Другими словами, до тех пор, пока видеопроцессор 1332 включен в устройство, любой тип конфигурации можно встроить в любое устройство различного типа, которое вырабатывает данные изображения, аналогично случаю с видеонабором 1300. Например, видеопроцессор 1332, процессор, показанный пунктирной линией 1341, видеомодуль 1311 или видеоблок 1361 могут быть включены в телевизионное устройство 900 (фиг.70), мобильный телефон 920 (фиг.71), устройство 940 записи/воспроизведения (фиг.72), устройство 960 формирования изображения (фиг.73) и т.п. Кроме того, встраивая и конфигурируя применение настоящей технологии, устройство способно получать эффекты, аналогичные эффектам каждого из вариантов осуществления, описанных выше со ссылкой на фиг.1-68, аналогично видеонабору 1300.

Шестой пример применения: сетевая система

Кроме того, настоящая технология также применима к сетевой системе, которая включает в себя множество устройств. На фиг.77 показан один пример схематичной конфигурации сетевой системы, к которой применима настоящая технология.

Сетевая система 1600, показанная на фиг.77, представляет собой систему, в которой устройства обмениваются друг с другом информацией, касающейся изображений (движущихся изображений), по сети. Облачный сервис 1601 сетевой системы 1600 представляет собой систему, которая предоставляет услугу, связанную с изображениями (движущимися изображениями), в терминалы, такие как компьютер 1611, аудиовизуальное (AV) оборудование 1612, мобильный терминал 1613 обработки информации и устройство 1614 Интернета вещей (IoT), коммуникативно связанное с облачным сервисом 1601. Например, облачный сервис 1601 предоставляет услугу доставки содержания изображения (движущегося изображения) в терминалы такого типа, которые называются потоковой передачей видео (по запросу или прямая трансляция). В качестве другого примера, облачный сервис 1601 предоставляет услугу резервного копирования, которое принимает и хранит содержание изображения (движущегося изображения), полученных из терминалов. В качестве другого примера, облачный сервис 1601 предоставляет услугу посредничества при обмене содержанием изображения (движущегося изображения) между терминалами.

Физическая конфигурация облачного сервиса 1601 может иметь любую конфигурацию. Например, облачный сервис 1601 может включать в себя различные серверы, такие как сервер, который хранит и управляет движущимися изображениями, сервер, который доставляет движущиеся изображения в терминалы, сервер, который получает движущиеся изображения из терминалов, и сервер, который управляет пользователями (терминалами) и платежами, а также любой тип сети, такой как Интернет или LAN.

Компьютер 1611 включает в себя устройство обработки информации, такое, например, как персональный компьютер, сервер или автоматизированное рабочее место. AV-оборудование 1612 включает в себя устройство обработки изображений, такое, например, как телевизионный приемник, устройство записи на жесткий диск, игровую приставку или камеру. Мобильный терминал 1613 обработки информации включает в себя мобильное устройство обработки информации, такое, например, как персональный компьютер типа ноутбук, терминал на основе планшетного компьютера,

мобильный телефон или смартфон. IoT-устройство 1614 включает в себя любой объект, который выполняет обработку, связанную с изображением, такую, например, как машину, бытовой электроприбор, предмет мебели, какую-то другую вещь, IC-тег или устройство в виде карты. Все эти терминалы включают в себя функцию связи и способны

5 подключаться (устанавливать сеанс с) к облачной службе 1601 и обмениваться информацией с облачной службой 1601 (то есть поддерживать с ней связь). Кроме того, каждый терминал может также поддерживать связь с другим терминалом. Связь между терминалами, может выполняться путем просмотра облачного сервиса 1601 или может выполняться без просмотра облачного сервиса 1601.

10 Когда настоящая технология применяется в сетевой системе 1600, как указано выше, и между терминалами или между терминалом и облачным сервисом 1601 осуществляется обмен данными изображения (движущегося изображения), данные изображения могут кодироваться/декодироваться, как описано выше в каждом из вариантов осуществления. Другими словами, терминалы (от компьютера 1611 к IoT-устройству 1614) и облачный

15 сервис 1601 каждый по отдельности может включать в себя функции устройства 400 кодирования изображения и устройства 100 декодирования изображения, которые описаны выше. Таким образом, терминалы (от компьютера 1611 до IoT-устройства 1614), которые обмениваются данными изображения, и облачный сервис 1601 позволяют получать эффекты, аналогичные эффектам каждого из вариантов осуществления,

20 описанных выше со ссылкой на фиг.1 – фиг.68.

#### Дополнение

Следует отметить, что в настоящем описании "флаг" представляет собой информацию для идентификации каждого из множества состояний, включающих в себя не только информацию, которая будет использоваться для идентификации двух состояний "истина"

25 (1) или "ложь" (0), но также информацию, которая позволяет идентифицировать три или более состояния. Поэтому значение, которое может иметь "флаг", может быть, например, двоичным значением 1 или 0, или троичным или иметь более высокое значение. То есть количество битов, образующих "флаг", является произвольным и может составлять один или более битов. Кроме того, предполагается, что

30 идентификационная информация (также включающая в себя флаг) имеет форму, в которой идентификационная информация включена в битовый поток, и форму, в которой отличительная информация между некоторой опорной информацией и идентификационной информацией включена в битовый поток, и, таким образом, в настоящем описании "флаг" и "идентификационная информация" включают в себя не

35 только информацию, но также отличительную информацию, которая служит в качестве ссылки.

В дополнение к этому, различные виды информации (метаданные и т.д.) по отношению к кодированным данным (битовому потоку) могут передаваться или записываться в любой форме до тех пор, пока они ассоциированы с кодированными

40 данными. Используемый в данном документе термин "ассоциированный с" означает в одном примере то, что при обработке одних данных могут использоваться другие данные (связанные). Другими словами, данные, ассоциированные друг с другом, могут быть собраны в виде одних данных или могут быть отдельными данными. В одном примере информация, ассоциированная с кодированными данными (изображением)

45 может передаваться по каналу передачи, который отличается от канала передачи кодированных данных (изображения). В дополнение к этому, в одном примере, информация, ассоциированная с кодированными данными (изображением), может быть записана на носителе записи (или других зонах записи того же самого носителя записи),

который отличается от носителей записи кодированных данных (изображения). Более того, этот термин "ассоциированный с" может означать часть данных, а не все данные. В одном примере изображение и информация, соответствующая изображению, могут ассоциироваться друг с другом в любых единицах, таких как множество кадров, один кадр, часть в пределах кадра или т.п.

Кроме того, используемые в данном документе термины "объединять", "мультиплексировать", "присоединять", "объединять", "включать в себя", "сохранять", "добавлять в", "включать в", "вставлять" и т.п. означают объединение множества объектов в одно целое, например, объединение кодированных данных и метаданных в один элемент данных, что означает одно употребление термина "ассоциированный с", описанного выше.

Кроме того, вариант осуществления настоящей технологии не ограничивается вариантами осуществления, описанными выше, и различные изменения и модификации могут быть сделаны без отклонения от объема настоящей технологии.

Кроме того, например, настоящая технология может быть также реализована посредством любой из конфигураций, образующих устройство или систему, например, посредством процессора с большой степенью интеграции (LSI) и т.п., модуля, использующего множество процессоров и т.п., блока, использующего множество модулей и т.п., набора, в котором дополнительные функции дополнительно добавлены в набор или т.п. (то есть посредством частичной конфигурации устройства).

Следует отметить, что в данном описании под системой подразумевается набор из множества составляющих элементов (например, устройств или модулей (частей)) независимо от того, находятся или нет все составляющие элементы в одном корпусе. Соответственно, множество устройств, которые содержатся в разных корпусах и соединены через сеть, и одно устройство, в котором множество модулей содержится в одном корпусе, представляют собой обе системы.

Кроме того, например, элемент, описанный как одно устройство (или блок обработки), может быть разделен и сконфигурирован как множество устройств (или блоков обработки). Наоборот, элементы, описанные выше как множество устройств (или блоков обработки), могут быть совместно сконфигурированы как одно устройство (или блок обработки). Кроме того, элемент, отличный от описанных выше, может быть добавлен в конфигурацию каждого устройства (или блока обработки). Кроме того, часть конфигурации данного устройства (или блока обработки) может быть включена в конфигурацию другого устройства (или другого блока обработки), если конфигурация или работа системы в целом является по существу одинаковой.

Кроме того, например, настоящая технология может принимать конфигурацию облачных вычислений, которая выполняет обработку путем выделения и совместного использования одной функции с помощью множества устройств через сеть.

Кроме того, например, описанная выше программа может быть выполнена в любом устройстве. В этом случае достаточно, если устройство имеет необходимую функцию (функциональный блок или т.п.) и может получать необходимую информацию.

Кроме того, например, каждый этап, описанный с помощью представленных выше блок-схем последовательности операций, может быть выполнен одним устройством или выполнен путем выделения множества устройств. Кроме того, в случае, когда множество процессов включено в один этап, множество процессов, включенных в этот один этап, может быть выполнено одним устройством или выполнено путем выделения множества устройств. Другими словами, множество процессов, включенных в один этап, может быть также выполнено в виде процесса из множества этапов. И наоборот,

процесс, описанный как множество этапов, может быть выполнен совместно как один этап.

Следует отметить, что в программе, выполняемой компьютером, обработка на этапах, описывающих программу, может выполняться в хронологическом порядке в соответствии с порядком, описанным в данном описании, или может выполняться одновременно или по отдельности в необходимый момент времени, например, когда выполняется вызов. Другими словами, при отсутствии противоречий обработка на этапах может выполняться в порядке, отличном от порядка, описанного выше. Кроме того, обработка на этапах, описывающих программу, может выполняться одновременно с обработкой другой программы или может выполняться в сочетании с обработкой другой программы.

Следует отметить, что при отсутствии противоречий множество существующих технологий, описанных в данном описании, может быть выполнено по отдельности независимо друг от друга. Разумеется, любое множество настоящих технологий может быть выполнено в комбинации. В одном примере часть или вся настоящая технология, описанная в любом из вариантов осуществления, может быть выполнена в сочетании с частью или всей настоящей технологией, описанной в другом варианте осуществления. Кроме того, любая из частей или вся настоящая технология, описанная выше, может быть выполнена в сочетании с другой технологией, которая не описана выше.

Кроме того, настоящая технология может быть также сконфигурирована следующим образом.

(1) Устройство обработки изображений, включающее в себя:

блок обратного ортогонального преобразования, выполненный с возможностью осуществления обратного ортогонального преобразования цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

(2) Устройство обработки изображений по (1), в котором информация относительно обратного ортогонального преобразования, включает в себя флаг адаптивного первичного преобразования, указывающий, следует ли применять адаптивное обратное первичное преобразование для адаптивного выбора одного из множества различных обратных ортогональных преобразований и использования выбранного преобразования в качестве обратного первичного преобразования, в блоке преобразования, подлежащем обработке.

(3) Устройство обработки изображений по (2), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости.

(4) Устройство обработки изображений по (3), в котором информация относительно обратного ортогонального преобразования, включает в себя идентификатор первичного преобразования, указывающий, какое обратное первичное преобразование следует применить к обратным первичным преобразованиям в вертикальном направлении и горизонтальном направлении.

(5) Устройство обработки изображений по (4), в котором значение идентификатора первичного преобразования цветности устанавливается в значение идентификатора первичного преобразования яркости в случае, когда флаг адаптивного первичного преобразования цветности имеет значение "истина", и устанавливается в заданное значение, когда флаг адаптивного первичного преобразования цветности имеет значение "ложь".

(6) Устройство обработки изображений по (5), в котором блок обратного

ортогонального преобразования выполнен с возможностью осуществления обратного первичного преобразования в качестве обратного ортогонального преобразования.

(7) Устройство обработки изображений по (6), в котором блок обратного ортогонального преобразования выполнен с возможностью осуществления обратного первичного горизонтального преобразования, представляющего собой обратное первичное преобразование в горизонтальном направлении, и обратного первичного вертикального преобразования, представляющего собой обратное первичное преобразование в вертикальном направлении, в качестве обратного ортогонального преобразования.

(8) Устройство обработки изображений по (7), в котором тип преобразования обратного первичного горизонтального преобразования цветности и тип преобразования обратного первичного вертикального преобразования цветности получается каждый по отдельности с использованием способа в соответствии со значением флага адаптивного первичного преобразования цветности.

(9) Устройство обработки изображений по (8), в котором, когда флаг адаптивного первичного преобразования цветности имеет значение "истина",

набор преобразований обратного первичного горизонтального преобразования и набор преобразований обратного первичного вертикального преобразования выбираются на основании информации о режиме предсказания,

флаг точного определения первичного горизонтального преобразования и первичный вертикальный флаг преобразования получают из идентификатора первичного преобразования цветности,

тип преобразования обратного первичного горизонтального преобразования цветности выбирается на основании набора первичных горизонтальных преобразований и флага точного определения первичного горизонтального преобразования, а

тип преобразования обратного первичного вертикального преобразования цветности выбирается на основании набора первичных вертикальных преобразований и первичного вертикального преобразования.

(10) Устройство обработки изображений по (8) или (9), в котором, когда флаг адаптивного первичного преобразования цветности имеет значение "ложь", заданный тип преобразования выбирается в качестве типа преобразования обратного первичного горизонтального преобразования цветности, и заданный тип преобразования выбирается в качестве типа преобразования обратного первичного вертикального преобразования цветности.

(11) Устройство обработки изображений по любому из (2)-(10), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, и флаг устанавливается в значение "ложь", когда типом предсказания блока кодирования является межблочное предсказание.

(12) Устройство обработки изображений по любому из (2)-(11), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, или в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование, и флаг установлен в значение "ложь" в случае, когда

типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочным копированием.

(13) Устройство обработки изображений по любому из пп.(2)-(12), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочным предсказанием, или когда типом предсказания является межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, и устанавливается в значение "ложь" в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом.

(14) Устройство обработки изображений по любому из (2)-(13), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда флаг вывода информации об адаптивном первичном преобразовании цветности, указывающий на то, должен ли быть получен флаг адаптивного первичного преобразования цветности на основании флага адаптивного первичного преобразования яркости, имеет значение "истина", и устанавливается в значение "ложь" в случае, когда флаг вывода информации об адаптивном первичном преобразовании цветности имеет значение "ложь".

(15) Устройство обработки изображений по (14), в котором флаг вывода информации об адаптивном первичном преобразовании цветности декодируется из битового потока, когда компонентом объекта обработки является цветность, флаг адаптивного первичного преобразования яркости имеет значение "истина", флаг обхода квантования преобразования имеет значение "ложь", и флаг пропуска преобразования имеет значение "ложь", а в других случаях значение вывода информации об адаптивном первичном преобразовании цветности устанавливается в значение "ложь".

(16) Устройство обработки изображений по любому из (2)-(15), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости в случае, когда размер короткой стороны обрабатываемого блока преобразования цветности больше или равен значению заданного порогового значения, а значение устанавливается в значение "ложь" в случае, когда размер короткой стороны блока преобразования меньше порогового значения.

(17) Устройство обработки изображений по любому из (1)-(16), в котором блок обратного ортогонального преобразования выполнен с возможностью осуществления обратного первичного горизонтального преобразования, являющегося обратным первичным преобразованием в горизонтальном направлении, и обратного первичного вертикального преобразования, являющегося обратным первичным преобразованием в вертикальном направлении, в качестве обратного ортогонального преобразования,

тип преобразования обратного первичного горизонтального преобразования устанавливается в заданный тип преобразования, когда размер ширины картинки обрабатываемого блока преобразования цветности равен или меньше значения заданного порогового значения, и тип преобразования устанавливается на основании набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования, когда размер ширины картинки блока преобразования больше порогового значения, и

тип преобразования обратного первичного вертикального преобразования устанавливается в заданный тип преобразования, когда размер высоты картинки

обрабатываемого блока преобразования цветности равен или меньше порогового значения, и тип преобразования устанавливается на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования, когда размер высоты картинки блока преобразования больше

5 порогового значения.

(18) Устройство обработки изображений по любому из (1)-(17), в котором информация, относящаяся к обратному ортогональному преобразованию, включает в себя флаг пропуска преобразования, указывающий, должен ли быть пропущен процесс обратного ортогонального преобразования.

10 (19) Устройство обработки изображений по (18), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости.

(20) Устройство обработки изображений по (18) или (19), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является межблочное предсказание, а значение устанавливается в значение флага пропуска преобразования цветности, которое получается путем декодирования битового потока, когда типом предсказания блока кодирования является межблочное предсказание.

20 (21) Устройство обработки изображений по любому из (18)-(20), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и

25 цветности совпадают друг с другом, и значение устанавливается в значение флага пропуска преобразования цветности, полученное посредством декодирования битового потока, когда типом предсказания блока кодирования является межблочное предсказание, и режимы предсказания для яркости и цветности не совпадают друг с другом.

30 (22) Устройство обработки изображений по любому из (18)-(21), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является

35 внутриблочное копирование, и значение устанавливается в значение флага пропуска преобразования цветности, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование.

(23) Устройство обработки изображений по любому из (18)-(22), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда флаг вывода информации о пропуске преобразования цветности, указывающий на то, должен ли быть получен флаг пропуска преобразования цветности на основании флага пропуска преобразования яркости, имеет значение "истина", и значение устанавливается в значение флага пропуска преобразования

40 цветности, полученное путем декодирования битового потока, когда флаг вывода информации о пропуске преобразования цветности имеет значение "ложь".

(24) Устройство обработки изображений по любому из (1)-(23), в котором информация, касающаяся обратного ортогонального преобразования, включает в себя

идентификатор вторичного преобразования, указывающий на то, какое обратное вторичное преобразование должно быть применено.

(25) Устройство обработки изображений по (24), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости.

(26) Устройство обработки изображений по (24) или (25), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, и значение устанавливается в значение идентификатора вторичного преобразования цветности, полученное путем декодирования битового потока, когда типом предсказания блока кодирования является межблочное предсказание.

(27) Устройство обработки изображений по любому из (24)-(26), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, и значение устанавливается в значение идентификатора вторичного преобразования цветности, полученное путем декодирования битового потока, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом.

(28) Устройство обработки изображений по любому из (24)-(27), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, и значение устанавливается в значение идентификатора вторичного преобразования цветности, полученное путем декодирования битового потока в случае, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование.

(29) Устройство обработки изображений по любому из (24)-(28), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда флаг вывода идентификационной информации о вторичном преобразовании цветности, указывающий на то, должен ли быть получен идентификатор вторичного преобразования цветности на основании идентификатора вторичного преобразования яркости, имеет значение "истина", и значение устанавливается в значение идентификатора вторичного преобразования цветности, полученное путем декодирования битового потока в случае, когда флаг вывода идентификационной информации о вторичном преобразовании цветности имеет значение "ложь".

(30) Способ обработки изображений, содержащий этапы, на которых: осуществляют обратное ортогональное преобразование цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

(31) Устройство обработки изображений, включающее в себя:

блок ортогонального преобразования, выполненный с возможностью осуществления ортогонального преобразования цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

5 (32) Устройство обработки изображений по (31), в котором информация об ортогональном преобразовании включает в себя флаг адаптивного первичного преобразования, указывающий, следует ли применять адаптивное первичное преобразование при адаптивном выборе одного из множества различных ортогональных преобразований и использовании выбранного преобразования в качестве первичного преобразования к блоку преобразования, подлежащего обработке.

10 (33) Устройство обработки изображений по (32), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости.

15 (34) Устройство обработки изображений по (33), в котором информация, касающаяся ортогонального преобразования, включает в себя идентификатор первичного преобразования, указывающий на то, какое первичное преобразование должно быть применено к первичным преобразованиям в вертикальном направлении и горизонтальном направлении.

20 (35) Устройство обработки изображений по (34), в котором значение идентификатора первичного преобразования цветности устанавливается в значение идентификатора первичного преобразования яркости, когда флаг адаптивного первичного преобразования цветности имеет значение "истина", и устанавливается в заданное значение, когда флаг адаптивного первичного преобразования цветности имеет значение "ложь".

25 (36) Устройство обработки изображений по (35), в котором блок ортогонального преобразования выполняет первичное преобразование в качестве ортогонального преобразования.

30 (37) Устройство обработки изображений по (36), в котором блок ортогонального преобразования выполняет первичное горизонтальное преобразование, которое является первичным преобразованием в горизонтальном направлении, и первичное вертикальное преобразование, которое является первичным преобразованием в вертикальном направлении в качестве ортогонального преобразования.

35 (38) Устройство обработки изображений по (37), в котором тип преобразования первичного горизонтального преобразования цветности и тип преобразования первичного вертикального преобразования цветности получается каждый по отдельности с использованием способа в соответствии со значением флага адаптивного первичного преобразования цветности.

40 (39) Устройство обработки изображений по (38), в котором когда флаг адаптивного первичного преобразования цветности имеет значение "истина",

набор преобразований первичного горизонтального преобразования и набор преобразований первичного вертикального преобразования выбираются на основании информации о режиме предсказания,

45 флаг точного определения первичного горизонтального преобразования и первичный вертикальный флаг преобразования получают из идентификатора первичного преобразования цветности,

тип преобразования первичного горизонтального преобразования цветности выбирается на основании набора первичных горизонтальных преобразований и флага

точного определения первичного горизонтального преобразования, и

тип преобразования первичного вертикального преобразования цветности выбирается на основании набора первичных вертикальных преобразований и флага точного определения первичного вертикального преобразования.

5 (40) Устройство обработки изображений по (38) или (39), в котором когда флаг адаптивного первичного преобразования цветности имеет значение "ложь",

заданный тип преобразования выбирается из типа преобразования первичного горизонтального преобразования цветности, и

10 заданный тип преобразования выбирается в качестве типа преобразования первичного вертикального преобразования цветности.

(41) Устройство обработки изображений по любому из (32)-(40), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания 15 блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, а значение устанавливается в значение "ложь", когда типом предсказания блока кодирования является межблочное предсказание.

(42) Устройство обработки изображений по любому из (32)-(41), в котором значение 20 флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, а значение 25 устанавливается в значение "ложь", когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное копирование.

(43) Устройство обработки изображений по любому из (32)-(42), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение 30 флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, и значение устанавливается в значение "ложь" в случае, когда типом предсказания блока 35 кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом.

(44) Устройство обработки изображений по любому из (32)-(43), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда флаг вывода информации 40 об адаптивном первичном преобразовании цветности, указывающий, должен ли быть получен флаг адаптивного первичного преобразования цветности на основании флага адаптивного первичного преобразования яркости, имеет значение "истина", и значение устанавливается в значение "ложь", когда флаг вывода информации об адаптивном первичном преобразовании цветности имеет значение "ложь".

45 (45) Устройство обработки изображений по (44), в котором, когда компонентом объекта обработки является цветность, флаг адаптивного первичного преобразования яркости имеет значение "истина", флаг обхода квантования преобразования имеет значение "ложь", и флаг пропуска преобразования имеет значение "ложь", флаг вывода

информации об адаптивном первичном преобразовании цветности кодируется.

(46) Устройство обработки изображений по любому из (32)-(45), в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда размер короткой стороны обрабатываемого блока преобразования цветности больше или равен значению заданного порогового значения, и значение устанавливается в значение "ложь", когда размер короткой стороны блока преобразования меньше порогового значения.

(47) Устройство обработки изображений, по любому из (31)-(46), в котором блок ортогонального преобразования выполнен с возможностью осуществления первичного горизонтального преобразования, которое является первичным преобразованием в горизонтальном направлении, и первичного вертикального преобразования, которое является первичным преобразованием в вертикальном направлении, в качестве ортогонального преобразования,

тип преобразования первичного горизонтального преобразования устанавливается в заданный тип преобразования, когда размер ширины картинки обрабатываемого блока преобразования цветности равен или меньше значения заданного порогового значения, и тип преобразования устанавливается на основании набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования, когда размер ширины картинки блока преобразования больше

порогового значения, и

тип преобразования первичного вертикального преобразования устанавливается в заданный тип преобразования, когда размер высоты картинки обрабатываемого блока преобразования цветности равен или меньше заданного порогового значения, и тип преобразования устанавливается на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования, когда размер высоты картинки блока преобразования больше порогового значения.

(48) Устройство обработки изображений по любому из (31)-(47), в котором информация, относящаяся к обратному ортогональному преобразованию, включает в себя флаг пропуска преобразования, указывающий, должен ли быть пропущен процесс обратного ортогонального преобразования.

(49) Устройство обработки изображений по (48), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости.

(50) Устройство обработки изображений по (48) или (49), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, и значение устанавливается в произвольное значение и кодируется в случае, когда типом предсказания блока кодирования является межблочное предсказание.

(51) Устройство обработки изображений по любому из (48)-(50), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, и значение устанавливается в произвольное значение и кодируется, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с

другом.

(52) Устройство обработки изображений по любому из (48)-(51), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, и значение устанавливается в произвольное значение и кодируется в случае, когда типом предсказания является межблочное предсказание, а режимом предсказания не является внутриблочное копирование.

(53) Устройство обработки изображений по любому из (48)-(52), в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования яркости, когда флаг вывода информации о пропуске преобразования цветности, указывающий, должен ли быть получен флаг пропуска преобразования цветности на основании флага пропуска преобразования яркости, имеет значение "истина", и значение устанавливается в произвольное значение и кодируется в случае, когда флаг вывода информации о пропуске преобразования цветности имеет значение "ложь".

(54) Устройство обработки изображений по любому из (31)-(53), в котором информация, относящаяся к ортогональному преобразованию, включает в себя идентификатор вторичного преобразования, указывающий, какое вторичное преобразование должно быть применено.

(55) Устройство обработки изображений по (54), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости.

(56) Устройство обработки изображений по (54) или (55), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, и значение устанавливается в произвольное значение и кодируется в случае, когда типом предсказания блока кодирования является межблочное предсказание.

(57) Устройство обработки изображений по любому из (54)-(56), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом, и значение устанавливается в произвольное значение и кодируется, когда типом предсказания блока кодирования является межблочное предсказание, а режимы предсказания для яркости и цветности не совпадают друг с другом.

(58) Устройство обработки изображений по любому из (54)-(57), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание или межблочное предсказание, а режимом предсказания является внутриблочное копирование, и значение устанавливается в произвольное значение и кодируется, когда типом предсказания блока кодирования является межблочное предсказание, а режимом предсказания не является внутриблочное

копирование.

(59) Устройство обработки изображений по любому из (54)-(58), в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости, когда флаг вывода

5 идентификационной информации о вторичном преобразовании цветности, указывающий, должен ли быть получен идентификатор вторичного преобразования цветности на основании идентификатора вторичного преобразования яркости, имеет значение "истина", и значение устанавливается в произвольное значение и кодируется в случае, когда флаг вывода идентификационной информации о вторичном преобразовании  
10 цветности имеет значение "ложь".

(60) Способ обработки изображений, содержащий этап, на котором:

осуществляют ортогональное преобразование цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

15 Перечень ссылочных позиций

100 – устройство декодирования изображения

111 – блок декодирования

112 – блок обратного квантования

113 – блок обратного преобразования

20 114 – блок арифметических операций

115 – память кадров

116 – блок предсказания

121 – переключатель

122 – блок обратного вторичного преобразования

25 123 – блок обратного первичного преобразования

131 – блок получения информации об адаптивном первичном преобразовании  
цветности

132 – блок выбора обратного первичного преобразования

133 – блок обратного первичного вертикального преобразования

30 134 – блок обратного первичного горизонтального преобразования

151 – блок получения apt\_flag

152 – блок получения pt\_idx

400 – устройство кодирования изображения

411 – блок управления

35 412 – блок арифметических операций

413 – блок преобразования

414 – блок квантования

415 – блока кодирования

416 – блок обратного квантования

40 417 – блок обратного преобразования

418 – блок арифметических операций

419 – память кадров

420 – блок предсказания

431 – переключатель

45 432 – блок первичного преобразования

433 – блок вторичного преобразования

441 – блок получения информации об адаптивном первичном преобразовании  
цветности

- 442 – блок выбора первичного преобразования
- 443 – блок первичного горизонтального преобразования
- 444 – блок первичного вертикального преобразования
- 451 – блок получения apt\_flag
- 5 452 – блок получения pt\_idx

# (57) Формула изобретения

1. Устройство обработки изображений, содержащее:

10 блок обратного ортогонального преобразования, выполненный с возможностью осуществления обратного ортогонального преобразования цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

2. Устройство обработки изображений по п.1, в котором информация, относящаяся к обратному ортогональному преобразованию, включает в себя флаг адаптивного  
15 первичного преобразования, указывающий, следует ли применять адаптивное обратное первичное преобразование для адаптивного выбора одного из множества различных обратных ортогональных преобразований и использования выбранного преобразования в качестве обратного первичного преобразования.

3. Устройство обработки изображений по п.2, в котором значение флага адаптивного  
20 первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости.

4. Устройство обработки изображений по п.3, в котором информация, относящаяся к обратному ортогональному преобразованию, включает в себя идентификатор  
25 первичного преобразования, указывающий, какое обратное первичное преобразование следует применить в качестве обратного первичного преобразования в вертикальном направлении и горизонтальном направлении.

5. Устройство обработки изображений по п.4, в котором значение идентификатора  
30 первичного преобразования цветности устанавливается в значение идентификатора первичного преобразования яркости, когда флаг адаптивного первичного преобразования цветности имеет значение "истина", и устанавливается в заданное значение, когда флаг адаптивного первичного преобразования цветности имеет значение "ложь".

6. Устройство обработки изображений по п.5, в котором блок обратного  
35 ортогонального преобразования выполнен с возможностью осуществления обратного первичного преобразования в качестве обратного ортогонального преобразования.

7. Устройство обработки изображений по п.6, в котором блок обратного  
40 ортогонального преобразования выполнен с возможностью обратного первичного горизонтального преобразования, являющегося обратным первичным преобразованием в горизонтальном направлении, и обратного первичного вертикального преобразования, являющегося обратным первичным преобразованием в вертикальном направлении, в качестве обратного ортогонального преобразования.

8. Устройство обработки изображений по п.2, в котором значение флага адаптивного  
45 первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание.

9. Устройство обработки изображений по п.2, в котором значение флага адаптивного  
первичного преобразования цветности устанавливается в значение флага адаптивного

первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, или когда типом предсказания является межблочное предсказание, а режимом предсказания является внутриблочное копирование.

5 10. Устройство обработки изображений по п.2, в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда типом предсказания блока кодирования, к которому принадлежит блок преобразования, подлежащий обработке, является внутриблочное предсказание, или когда типом предсказания является межблочное  
10 предсказание, а режимы предсказания для яркости и цветности совпадают друг с другом.

11. Устройство обработки изображений по п.2, в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда флаг вывода информации об адаптивном первичном преобразовании цветности, указывающий, должен ли быть получен флаг  
15 адаптивного первичного преобразования цветности на основании флага адаптивного первичного преобразования яркости, имеет значение "истина".

12. Устройство обработки изображений по п.2, в котором значение флага адаптивного первичного преобразования цветности устанавливается в значение флага адаптивного первичного преобразования яркости, когда размер короткой стороны обрабатываемого  
20 блока преобразования цветности больше или равен значению заданного порогового значения.

13. Устройство обработки изображений по п.1, в котором блок обратного ортогонального преобразования выполнен с возможностью осуществления обратного первичного горизонтального преобразования, являющегося  
25 обратным первичным преобразованием в горизонтальном направлении, и обратного первичного вертикального преобразования, являющегося обратным первичным преобразованием в вертикальном направлении, в качестве обратного ортогонального преобразования,

тип преобразования обратного первичного горизонтального преобразования  
30 устанавливается на основании набора горизонтальных преобразований и флага точного определения первичного горизонтального преобразования, когда размер ширины картинки обрабатываемого блока преобразования цветности больше значения заданного порогового значения, и

тип преобразования обратного первичного вертикального преобразования  
35 устанавливается на основании набора вертикальных преобразований и флага точного определения первичного вертикального преобразования, когда размер высоты картинки обрабатываемого блока преобразования цветности больше значения заданного порогового значения.

14. Устройство обработки изображений по п.1, в котором информация, относящаяся  
40 к обратному ортогональному преобразованию, включает в себя флаг пропуска преобразования, указывающий, должен ли быть пропущен процесс обратного ортогонального преобразования.

15. Устройство обработки изображений по п.14, в котором значение флага пропуска преобразования цветности устанавливается в значение флага пропуска преобразования  
45 яркости.

16. Устройство обработки изображений по п.1, в котором информация, относящаяся к обратному ортогональному преобразованию, включает в себя идентификатор вторичного преобразования, указывающий, какое обратное вторичное преобразование

должно быть применено.

17. Устройство обработки изображений по п.16, в котором значение идентификатора вторичного преобразования цветности устанавливается в значение идентификатора вторичного преобразования яркости.

5 18. Способ обработки изображений, содержащий этап, на котором:  
осуществляют обратное ортогональное преобразование цветности с использованием информации об обратном ортогональном преобразовании цветности, полученной на основании информации об обратном ортогональном преобразовании яркости.

10 19. Устройство обработки изображений, содержащее:  
блок ортогонального преобразования, выполненный с возможностью осуществления ортогонального преобразования цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

15 20. Способ обработки изображений, содержащий этап, на котором:  
осуществляют ортогональное преобразование цветности с использованием информации об ортогональном преобразовании цветности, полученной на основании информации об ортогональном преобразовании яркости.

20

25

30

35

40

45

1

Фиг. 1

| Четыре набора заданных преобразований-кандидатов (LUT_TrSetToTrTypeIdx) |                          |                       |
|-------------------------------------------------------------------------|--------------------------|-----------------------|
| TrSetIdx                                                                | Преобразования-Кандидаты |                       |
|                                                                         | pt_{hor, ver}_flag==0    | pt_{hor, ver}_flag==1 |
| 0                                                                       | 4 (DST-VII)              | 2 (DCT-VIII)          |
| 1                                                                       | 4 (DST-VII)              | 3 (DST-I)             |
| 2                                                                       | 4 (DST-VII)              | 1 (DCT-V)             |
| 3                                                                       | 2 (DCT-VIII)             | 4 (DST-VI)            |
| 4                                                                       | DCT-II                   |                       |

для межблокового  
предсказания

для внутриблокового  
предсказания

Набор преобразований

случаи, в которых адаптивное первичное преобразование  
НЕ ПРИМЕНЯЕТСЯ

★ TrTypeIdx==0 соответствует DCT-II

2

Фиг. 2

| TrType<br>Idx | Тип<br>преобразования | Базисная функция $T_i(j), i, j=0, 1, \dots, N-1$                                                                                                                                                                                                                                                          |
|---------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0             | DCT-II                | $T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right), \text{ where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i=0 \\ 1 & i \neq 0 \end{cases}$                                                                                                     |
| 1             | DCT-V                 | $T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right), \text{ where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i=0, \omega_1 = \\ 1 & i \neq 0 \end{cases}$<br>$\begin{cases} \sqrt{\frac{2}{N}} & j=0 \\ 1 & j \neq 0 \end{cases}$ |
| 2             | DCT-VIII              | $T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$                                                                                                                                                                                                        |
| 3             | DST-I                 | $T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$                                                                                                                                                                                                            |
| 4             | DST-VII               | $T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$                                                                                                                                                                                                         |

$N=2, 4, 8, 16, \dots, 2^{**M}$

С ТОЧКИ ЗРЕНИЯ РЕАЛИЗАЦИИ ЗНАЧЕНИЯ КАЖДОГО КОЭФФИЦИЕНТА  $T(i)$  ОРТОГОНАЛЬНОГО ПРЕОБРАЗОВАНИЯ ПОДДЕРЖИВАЕТСЯ С ТОЧНОСТЬЮ ЦЕЛОГО ЧИСЛА, РАВНОГО К БИТАМ

Фиг. 3

| Выбранные наборы горизонтального (H) и вертикального (V) преобразований для каждого режима межблокового преобразования (LUT_IntraModeToTrSet) |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                    |    |    |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------|----|----|----|
| Режим межблокового преобразования                                                                                                             | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14                 | 15 | 16 | 17 |
| V                                                                                                                                             | 2  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0                  | 0  | 0  | 0  |
| H                                                                                                                                             | 2  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 2                  | 2  | 2  | 2  |
| Режим межблокового преобразования                                                                                                             | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32                 | 33 | 34 |    |
| V                                                                                                                                             | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0                  | 1  | 0  |    |
| H                                                                                                                                             | 2  | 2  | 2  | 2  | 2  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0                  | 1  | 0  |    |
| Режим межблокового преобразования                                                                                                             | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49                 | 50 | 51 | 52 |
| V                                                                                                                                             | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 2  | 2  | 2  | 2                  | 2  | 2  | 2  |
| H                                                                                                                                             | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0                  | 0  | 0  | 0  |
| Режим межблокового преобразования                                                                                                             | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 (=IntraBC)      |    |    |    |
| V                                                                                                                                             | 2  | 2  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 4 (=InterTrSetIdx) |    |    |    |
| H                                                                                                                                             | 0  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 4 (=InterTrSetIdx) |    |    |    |

A

|                                                                            | Decompmap |
|----------------------------------------------------------------------------|-----------|
| $\{ transform\_in(\{ -d_1 y_0, 1, g_2^2 360\} z_{el}, log2360\} z_{el}, )$ |           |
| $\{$                                                                       |           |
| for case [0 : 0], comp [0, max(a) : 1] name : comp [n : 3]                 |           |
| $\frac{g_2^2}{360} \text{ sort } [0 : 0] \text{ comp } [0 : 1]$            |           |
| if [0 : 0] [0 : 0] comp [0 : 1]                                            |           |
| if [0 : 0] [0 : 0] comp [0 : 1] & & a[n] end else : f ag 1:                |           |
| $\frac{g_2^2}{360} \text{ sort } [0 : 0] [0 : 2]$                          |           |
| 1                                                                          |           |
| log2360, 220 - comp [0 : 0] comp [0 : 1] log2360, 265 : log2360, 265       |           |
| log2360, 265 - comp [0 : 0] comp [0 : 1] log2360, 265 : log2360, 265       |           |
| residual log2360, 265 : log2360, 265                                       |           |
| }                                                                          |           |
| $\}$ SYN11                                                                 | ae(v)     |
| $\}$ SYN12                                                                 | ae(v)     |
| $\}$ SYN13                                                                 |           |
| $\}$                                                                       |           |

A

флаг `is_kodir` [y0] [compnd]. флаг кодирования блока.  
флаг `is_kodir` указывает, является ли один или более ненулевых коэффициентов в транспортном блоке сигналом цветности. Цветность пространства, точно определенного `compnd`.  
в случае 1 (истина), один или более ненулевых коэффициентов находятся в соответствующем блоке преобразования.  
в случае 0 (ложь), ненулевой коэффициент отсутствует в соответствующем транспортном блоке.  
флаг `is_adapt` [y0] [compnd]. флаг разрешения адаптивного преобразования.  
флаг `is_adapt` указывает разрешение адаптивного преобразования.  
в случае 1 (истина), флаг указывает, что информация, касающаяся адаптивного преобразования, может быть представлена в кодированных данных.  
в случае 0 (ложь), флаг указывает, что информация, касающаяся адаптивного преобразования, отсутствует в кодированных данных.  
флаг `is_adapt` [y0] [compnd]. флаг адаптивного преобразования.  
флаг `is_adapt` указывает, следует ли применять адаптивное преобразование к блоку преобразования яркости.  
в случае 1 (истина), флаг указывает возможность применения адаптивного преобразования.  
в случае 0 (ложь), флаг указывает на невозможность применения адаптивного преобразования.  
флаг `is_adapt` [y0] [compnd]. флаг адаптивного преобразования.  
флаг `is_adapt` указывает, когда флаг `is_adapt` отсутствует в кодированных данных, значение `is_adapt` устанавливается на 0.



Фиг. 5

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| residual_coding( x0, y0, log2TBSize, log2TBSize, compID ) {<br>if( !ts.enabled_flag && !transquant_bypass_flag &&<br>( log2TBSize <= log2MaxTSSize ) )<br><b>ts_flag</b> [x0][y0][compID]<br>... syntaxes related to last_pos [x, y],<br>coeff_sblk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ...<br>if( !apt_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]<br>&& compID==COMPONENT_Y && numSig >= ptNumSigTH &&<br>max( log2TBSize, log2TBSize ) <= maxPTSize )<br><b>pt_idx</b> [x0][y0]<br>...<br>if( ! ( !ts.enabled_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]<br>&& compID==COMPONENT_Y && numSig >= stNumSigTH ) )<br><b>st_idx</b> [x0][y0]<br>} | Дескриптор    |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>ae</b> (v) |
| } SYN21                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |
| } SYN22                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>ae</b> (v) |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |
| } SYN23                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>ae</b> (v) |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |

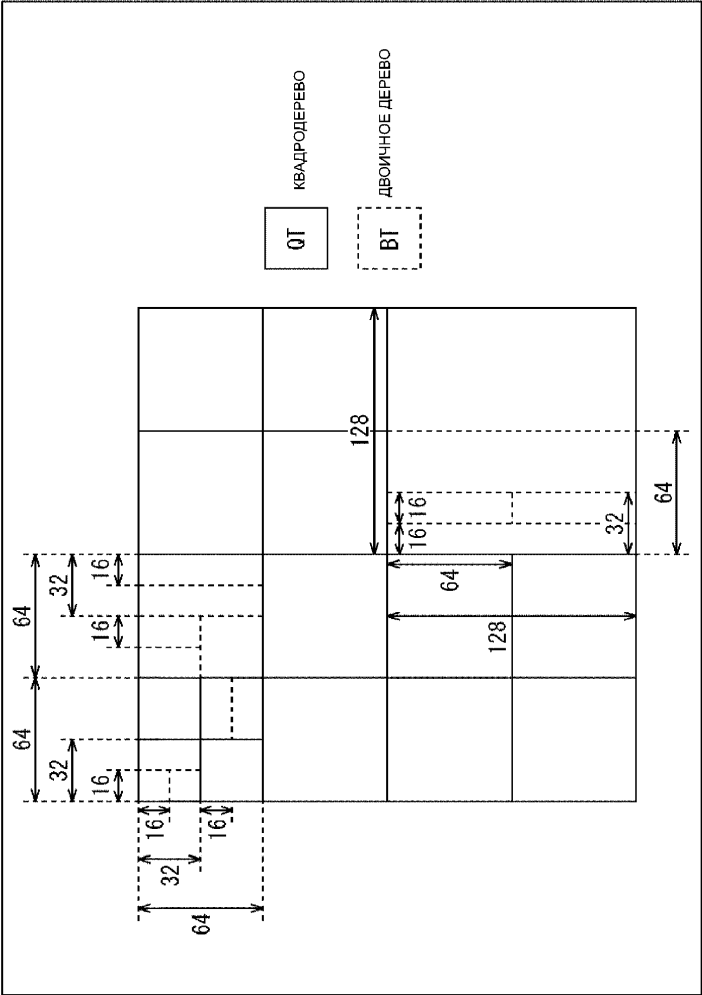
Фиг. 6

| яркость                                                | цветность                         |                                         | результат                                                                                     |
|--------------------------------------------------------|-----------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------|
|                                                        | Cb                                | Cr                                      |                                                                                               |
| флаг apt_flag[Y] АДАПТИВНОГО ПЕРВИЧНОГО ПРЕОБРАЗОВАНИЯ | apt_flag[Cb]                      | apt_flag[Cr]                            | •ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ ЦВЕТНОСТИ (Cb/Cr)<br>- УМЕНЬШЕНИЕ ИЗДЕРЖЕК               |
| ИДЕНТИФИКАТОР pt_idx[Y] ПЕРВИЧНОГО ПРЕОБРАЗОВАНИЯ      | pt_idx[Cb]                        | pt_idx [Cr] *МОЖЕТ БЫТЬ ОБЩЕЙ ДЛЯ Cb/Cr | •ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ ЦВЕТНОСТИ (Cb/Cr)<br>- УМЕНЬШЕНИЕ ИЗДЕРЖЕК               |
| ИДЕНТИФИКАТОР st_idx[Y] ВТОРИЧНОГО ПРЕОБРАЗОВАНИЯ      | chroma_st_idx<br>*ОБЩАЯ ДЛЯ Cb/Cr |                                         | •УМЕНЬШЕНИЕ ИЗДЕРЖЕК СООТВЕТСТВУЮЩЕГО СИНТАКСИСА, ПРОСИГНАЛИЗИРОВАННОГО ДЛЯ ЦВЕТНОСТИ (Cb/Cr) |
| флаг ts_flag ПРЕОБРАЗОВАНИЯ (transform_skip_flag)      | ts_flag[Cb]                       | ts_flag[Cr]                             | •УМЕНЬШЕНИЕ ИЗДЕРЖЕК СООТВЕТСТВУЮЩЕГО СИНТАКСИСА, ПРОСИГНАЛИЗИРОВАННОГО ДЛЯ ЦВЕТНОСТИ (Cb/Cr) |

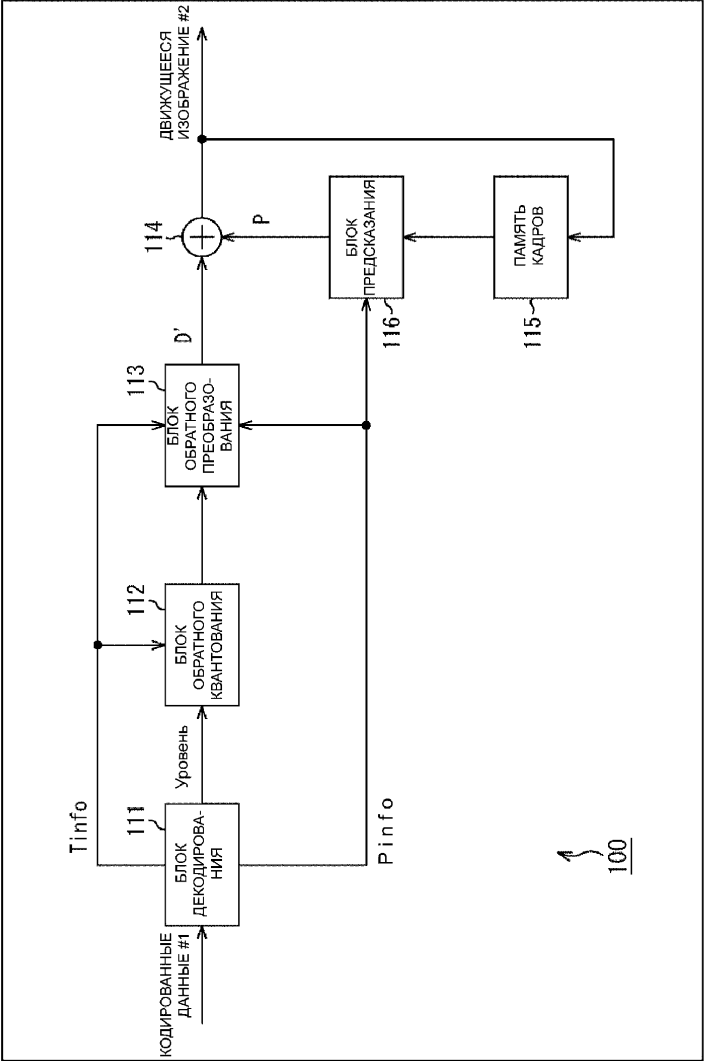
Фиг. 7

| ФЛАГ apt_flag[compID]pt_idx[compID] АДАПТИВНОГО ПРЕОБРАЗОВАНИЯ ЦВЕТНОСТИ |                                     |                                  |                                  |          |                                                               |                                                                               |   |                                        |                                                                 | РЕЗУЛЬТАТ |
|--------------------------------------------------------------------------|-------------------------------------|----------------------------------|----------------------------------|----------|---------------------------------------------------------------|-------------------------------------------------------------------------------|---|----------------------------------------|-----------------------------------------------------------------|-----------|
| №                                                                        | ВЫВОД apt_flag ЦВЕТНОСТИ ИЗ ЯРКОСТИ | IntraPredModeY == IntraPredModeC |                                  |          |                                                               | ДРУГОЕ УСЛОВИЕ                                                                |   | ПРИМЕЧАНИЕ                             |                                                                 |           |
|                                                                          |                                     | ИНТРА-CU                         |                                  | ИНТЕР-CU | ФЛАГ, УКАЗЫВАЮЩИЙ НА ТО, ЧТО СЛЕДУЕТ ПРЕОБРАЗОВАНИЕ ЦВЕТНОСТИ | РАЗМЕР, КОТОРЫЙ СТОИТ ПРЕОБРАЗОВАТЬ, ИЛИ БОЛЬШЕ ИЛИ РАВНО ПОРОГОВОМУ ЗНАЧЕНИЮ |   |                                        |                                                                 |           |
|                                                                          |                                     | БЕЗ ДОПОЛНИТЕЛЬНОГО УСЛОВИЯ      | IntraPredModeY == IntraPredModeC |          |                                                               |                                                                               |   |                                        |                                                                 |           |
| #0                                                                       |                                     | X                                |                                  |          | X                                                             |                                                                               |   | ЦВЕТНОСТЬ ОТНОСИТЕЛЬНО МОМЕНТА ВРЕМЕНИ |                                                                 |           |
| #1                                                                       | X                                   |                                  |                                  |          |                                                               |                                                                               |   | ВЫВОДИЗ ЯРКОСТИ                        | ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ                             |           |
| #2                                                                       | X                                   |                                  |                                  |          | X                                                             |                                                                               |   | ОПУСКАЕТСЯ ТОЛЬКО ПРИ ВВОДЕ            | ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ                             |           |
| #3                                                                       | X                                   |                                  | X                                |          | X                                                             |                                                                               |   |                                        | УМЕНЬШЕНИЕ ОБЪЕМА КОДА                                          |           |
| #4                                                                       | X                                   |                                  |                                  | X        | X                                                             |                                                                               |   |                                        | УМЕНЬШЕНИЕ ОБЪЕМА КОДА                                          |           |
| #5                                                                       | X                                   |                                  |                                  |          |                                                               | X                                                                             |   |                                        | УМЕНЬШЕНИЕ ОБЪЕМА КОДА                                          |           |
| #6                                                                       | X                                   |                                  |                                  |          |                                                               |                                                                               | X |                                        | УМЕНЬШЕНИЕ РАЗМЕРОВ СХЕМЫ                                       |           |
| #7                                                                       | X                                   |                                  |                                  |          |                                                               |                                                                               |   | X                                      | УМЕНЬШЕНИЕ РАЗМЕРОВ СХЕМЫ                                       |           |
| #8                                                                       | X                                   |                                  |                                  |          | X                                                             |                                                                               |   | X                                      | ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ - УМЕНЬШЕНИЕ РАЗМЕРОВ СХЕМЫ |           |

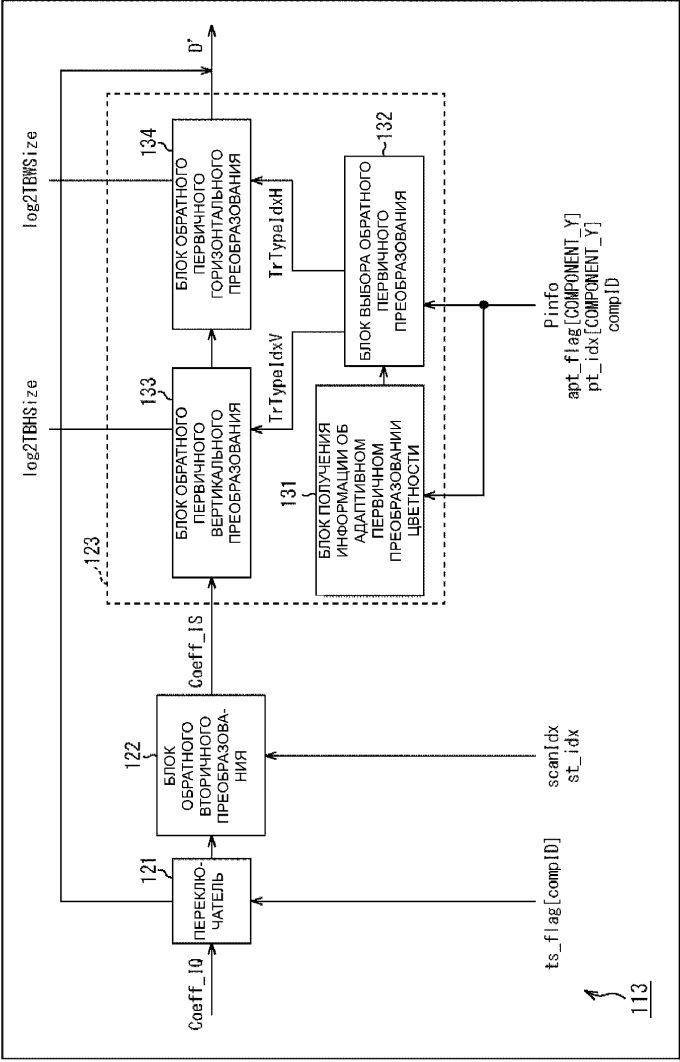
Фиг. 8



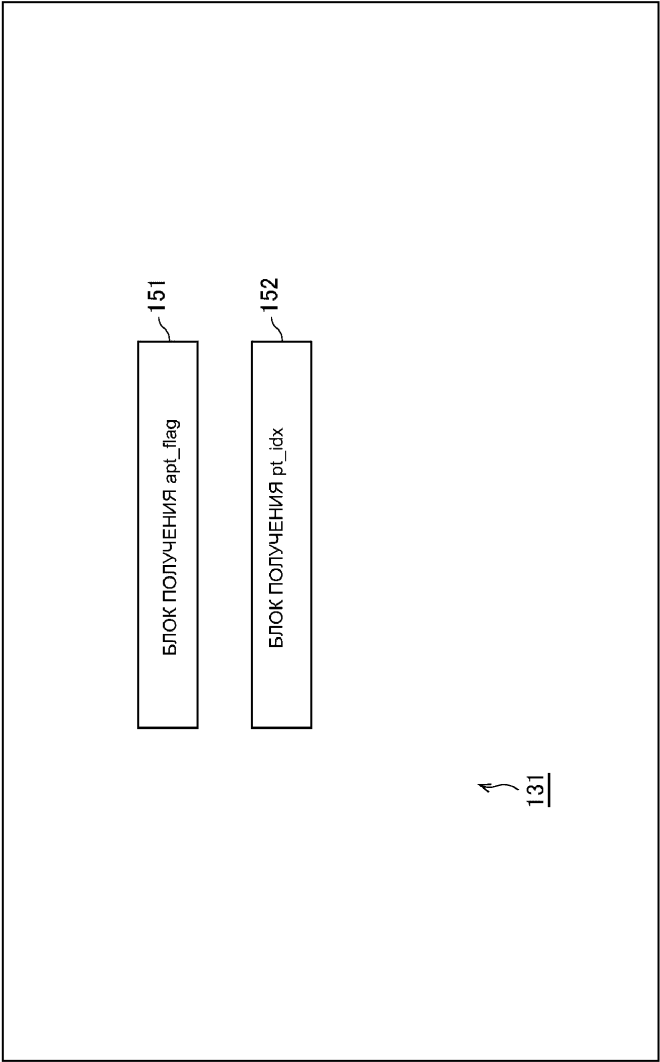
Фиг. 9



Фиг. 10



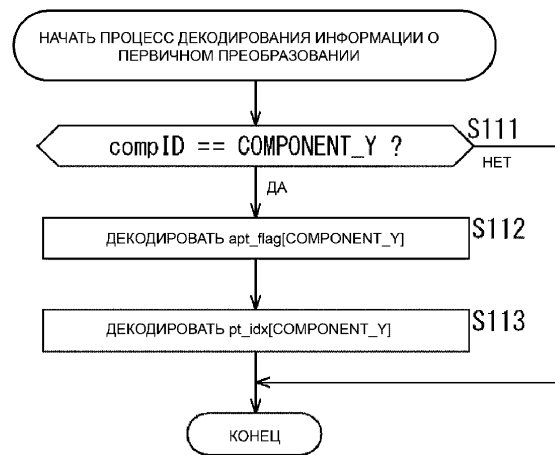
Фиг. 11



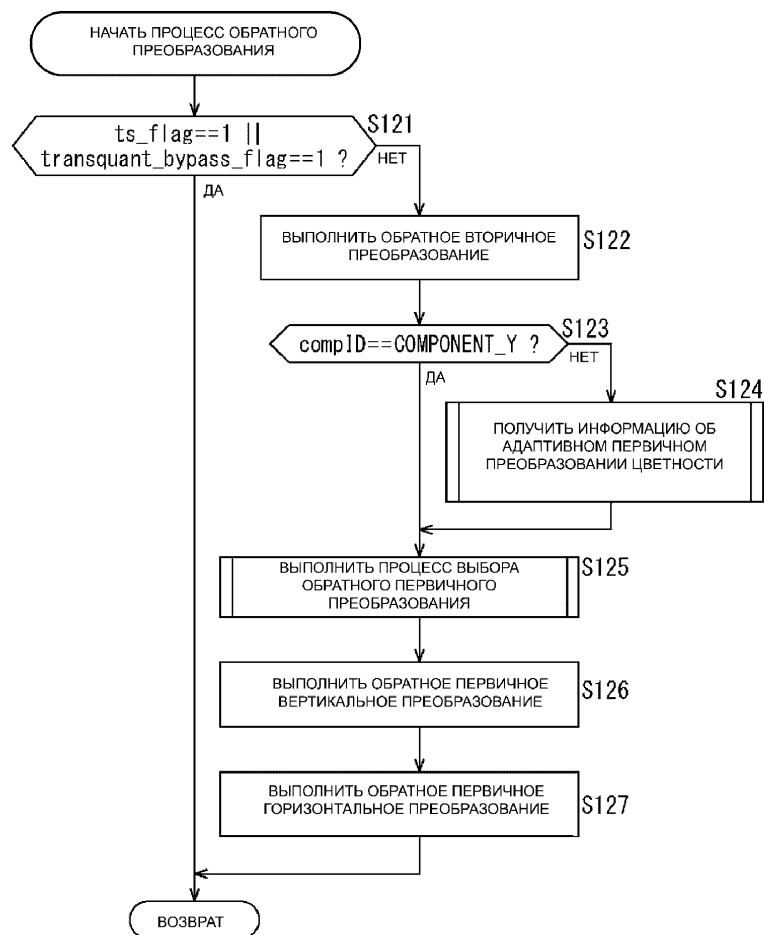
Фиг. 12



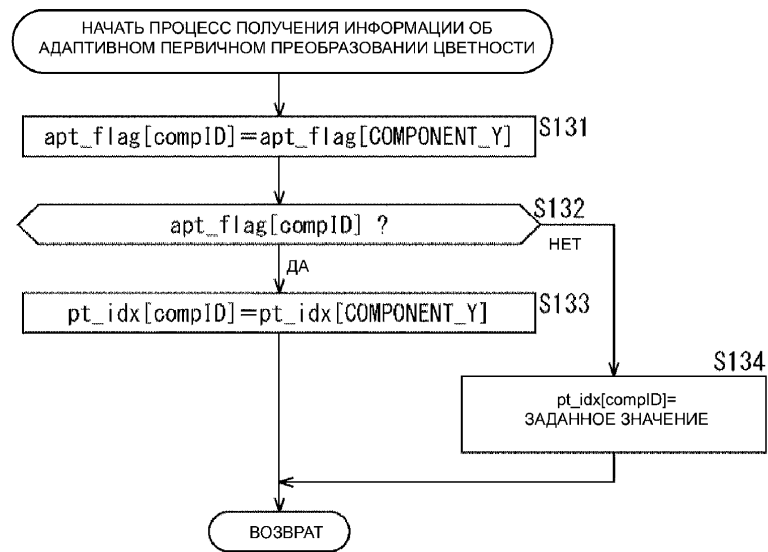
Фиг. 13



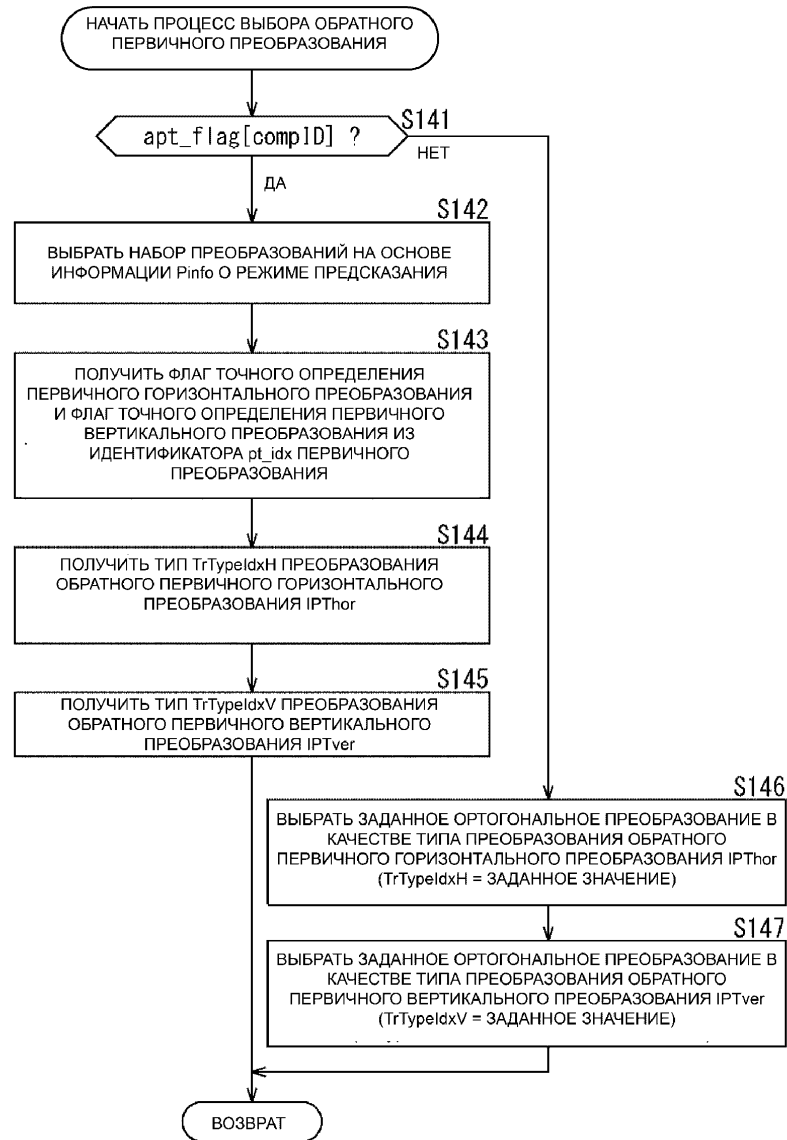
Фиг. 14



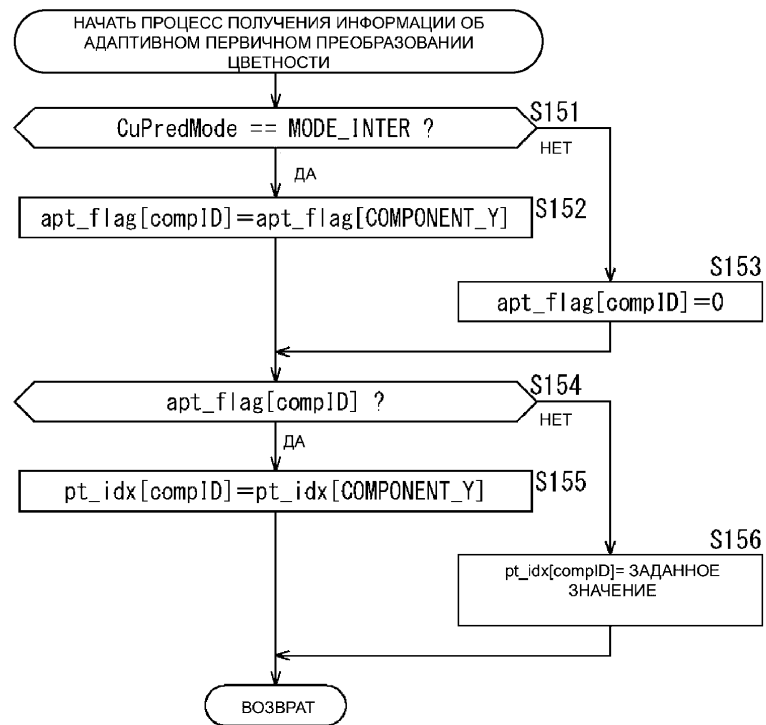
Фиг. 15



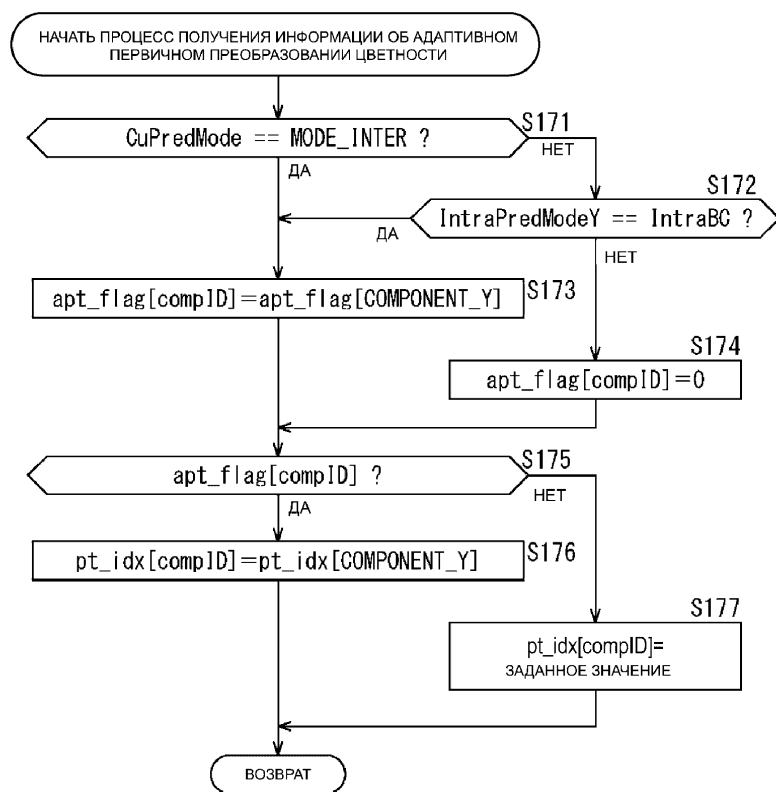
Фиг. 16



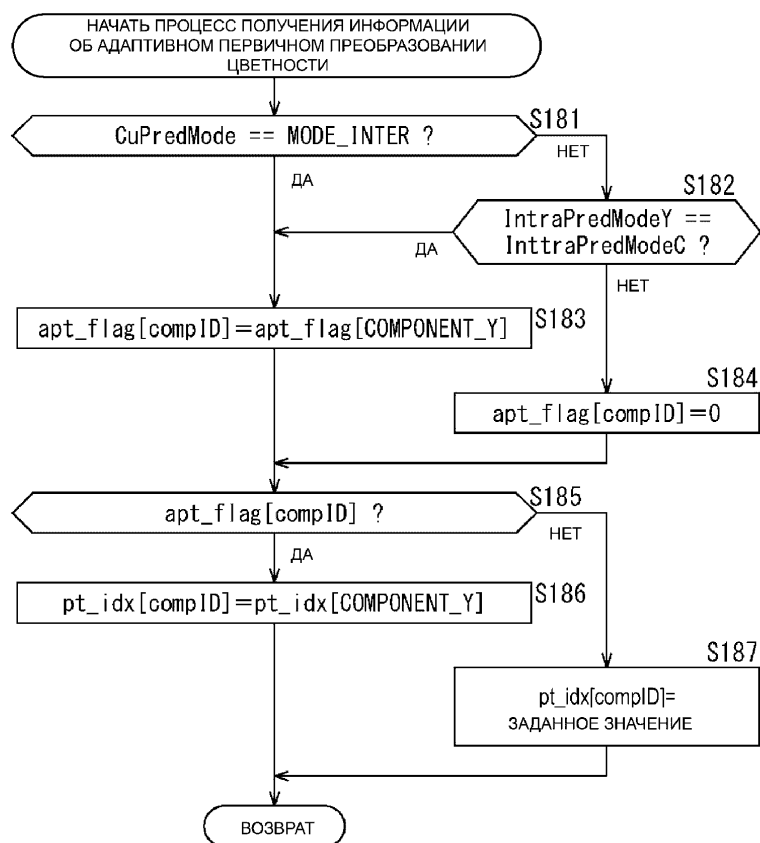
Фиг. 17



Фиг. 18



Фиг. 19



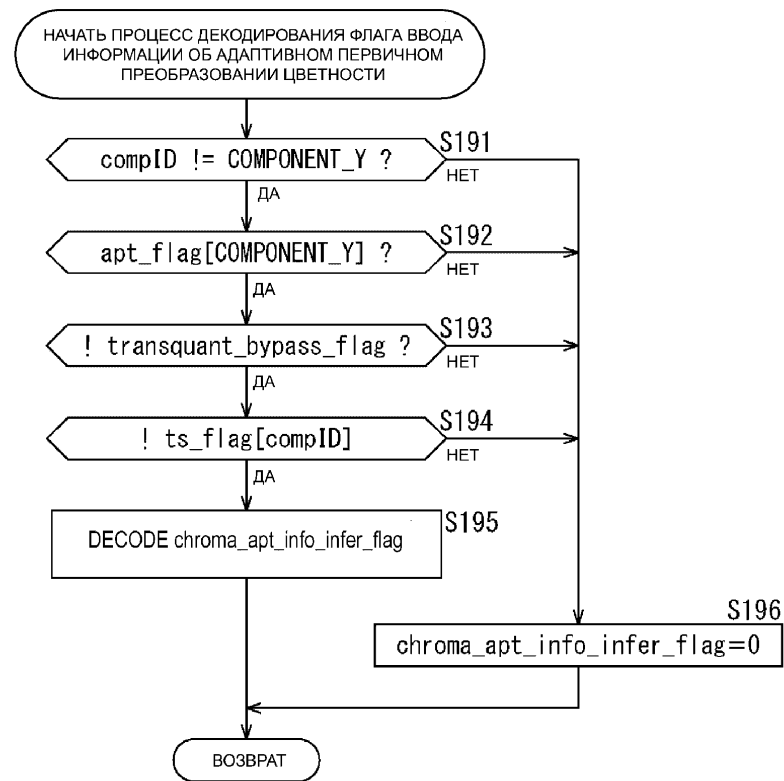
A

A

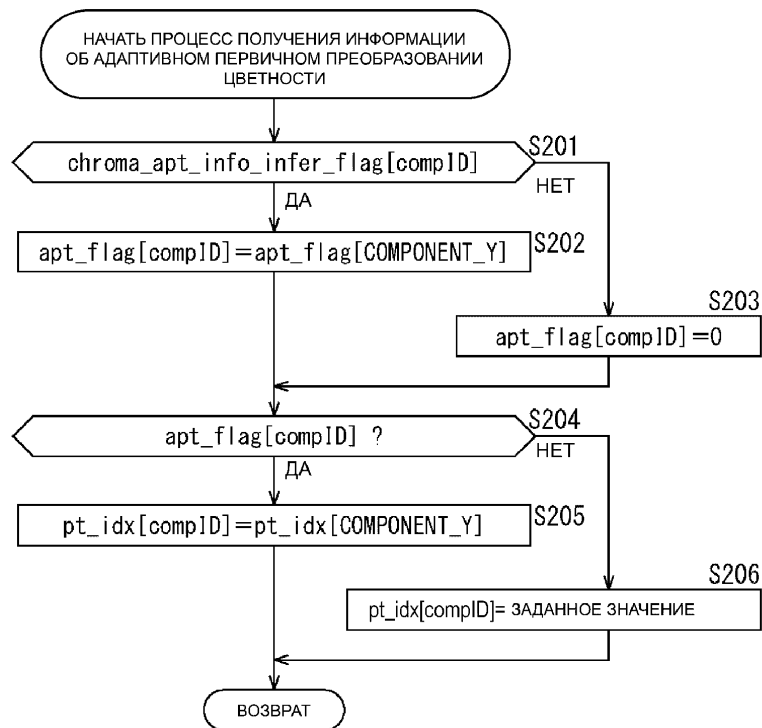
Фиг. 21

| Дескриптор               |                                                                                                                                |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|                          | <code>residual_coding ( x0 y0 log2TBSsize log2TBSsize compID ) {</code>                                                        |
|                          | <code>if ( ts_enabled_flag &amp;&amp; ! transient_bypass_flag &amp;&amp;</code>                                                |
|                          | <code>  ( log2TBSsize &lt;= log2MaxTSSize )</code>                                                                             |
| ts_flag [x0][y0][compID] | <code>  ts_flag [x0][y0][compID]</code>                                                                                        |
| ...                      | <code>  ...</code>                                                                                                             |
|                          | <code>if ( compID==COMPONENT_Y &amp;&amp; apt_flag[x0][y0][COMPONENT_Y] &amp;&amp; !ts_flag [x0][y0][compID] &amp;&amp;</code> |
|                          | <code>  !transquant_bypass_flag</code>                                                                                         |
|                          | <code>  chroma_apt_info_infer_flag [x0][y0][compID]</code>                                                                     |
|                          | <code>  }</code>                                                                                                               |
|                          | <code>  ... syntaxes related to last_pos {x, y},</code>                                                                        |
|                          | <code>    coeff_sbk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ...</code>                       |
|                          | <code>  if ( apt_flag[x0][y0][compID] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]</code>            |
|                          | <code>    &amp;&amp; compID==COMPONENT_Y ) &amp;&amp; numSig &gt;= pNumSigH )</code>                                           |
| pt_idx [x0][y0][compID]  | <code>    pt_idx [x0][y0][compID]</code>                                                                                       |
|                          | <code>  }</code>                                                                                                               |

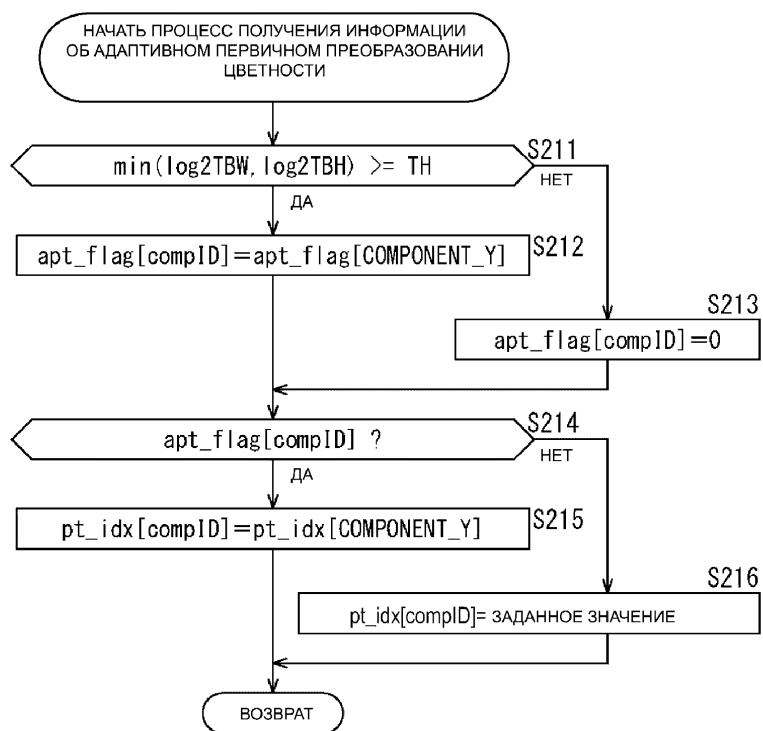
Фиг. 22



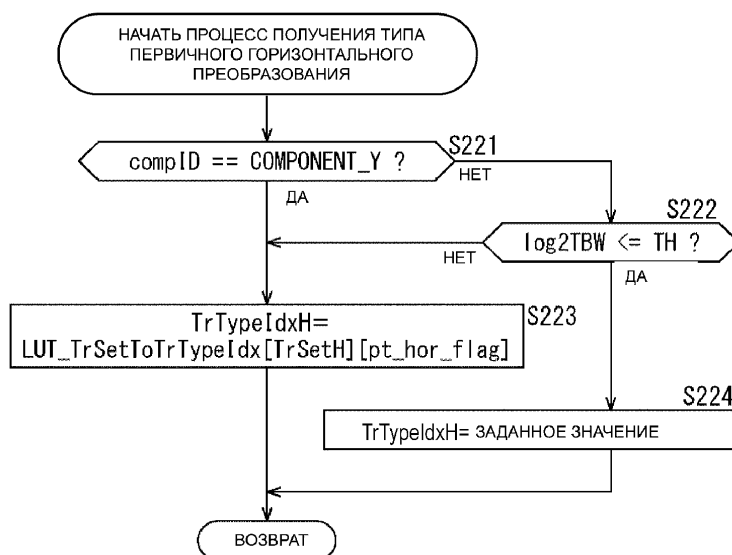
Фиг. 23



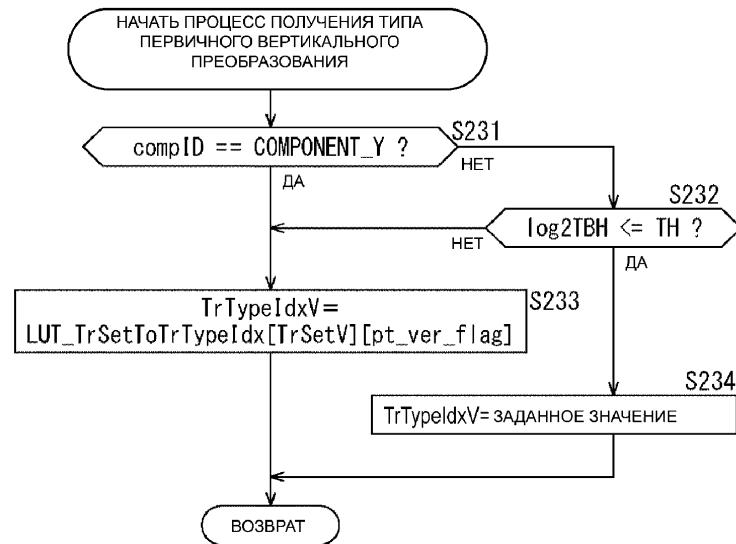
Фиг. 24



Фиг. 25



Фиг. 26



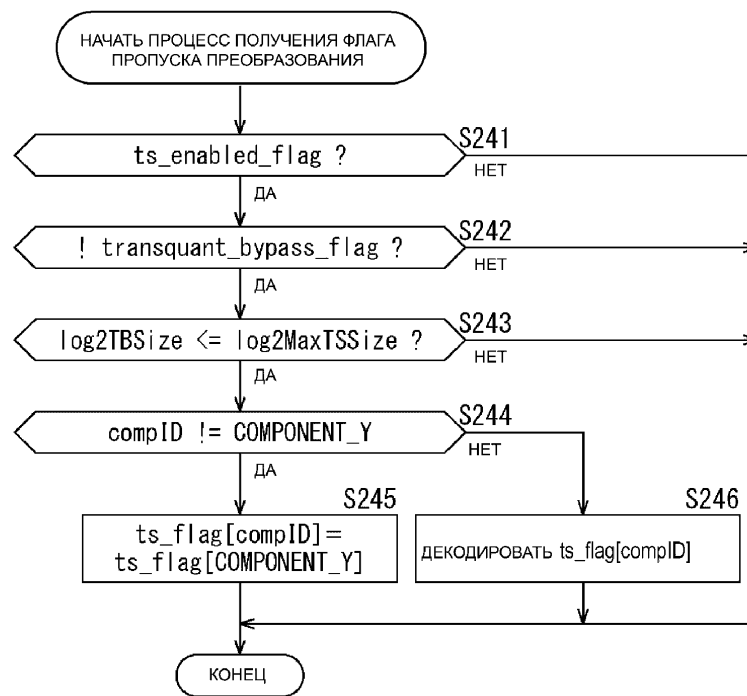
Фиг. 27

| ФЛАГ ts_flagCb/Cr] ПРОПУСКА ПРЕОБРАЗОВАНИЯ ЦВЕТНОСТИ |                           |                                                                |                                |          |                             |                                                                                | РЕЗУЛЬТАТ                                                 |
|------------------------------------------------------|---------------------------|----------------------------------------------------------------|--------------------------------|----------|-----------------------------|--------------------------------------------------------------------------------|-----------------------------------------------------------|
| №                                                    | ВВОД ЦВЕТНОСТИ ИЗ ЯРКОСТИ | УСЛОВИЕ ДЛЯ ИСКЛЮЧЕНИЯ ФЛАГА ПРОПУСКА ПРЕОБРАЗОВАНИЯ ЦВЕТНОСТИ |                                |          | ДРУГИЕ ХАРАКТЕРИСТИКИ ФЛАГА | ПРИМЕЧАНИЕ                                                                     |                                                           |
|                                                      |                           | ИНТРА-CU                                                       |                                | ИНТЕР-CU |                             |                                                                                |                                                           |
|                                                      |                           | БЕЗ ДОПОЛНИТЕЛЬНОГО УСЛОВИЯ                                    | IntraPredModeY==IntraPredModeC |          |                             |                                                                                |                                                           |
|                                                      |                           | IntraPredModeY==IntraPredModeC                                 | IntraPredModeY==IntraPredModeC |          |                             |                                                                                |                                                           |
| #0                                                   |                           |                                                                |                                |          |                             | СИГНАЛИЗАЦИЯ В ЛЮБОЙ МОМЕНТ ВРЕМЕНИ                                            |                                                           |
| #1                                                   | ×                         | ×                                                              |                                |          | ×                           | ИСКЛЮЧИТЬ КОДИРОВАНИЕ ПРОПУСКА ПРЕОБРАЗОВАНИЯ ЦВЕТНОСТИ В ЛЮБОЙ МОМЕНТ ВРЕМЕНИ | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                       |
| #2                                                   | ×                         |                                                                |                                |          | ×                           | ИСКЛЮЧИТЬ ТОЛЬКО ПРИ ИНТРА-ПРЕДСКАЗАНИИ                                        | ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ С ИНТРА-ПРЕДСКАЗАНИЕМ |
| #3                                                   | ×                         |                                                                | ×                              |          | ×                           |                                                                                | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                       |
| #4                                                   | ×                         |                                                                |                                | ×        | ×                           |                                                                                | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                       |
| #5                                                   | ×                         |                                                                |                                |          |                             | ×                                                                              | СИГНАЛИЗАЦИЯ В ОБЩЕМ ДЛЯ Cb/Cr                            |

Фиг. 28

|                                                                                          |            |
|------------------------------------------------------------------------------------------|------------|
| residual_coding( x0, y0, log2TBSize, log2TBSize, compID ) {                              | Дескриптор |
| if( !ts_enabled_flag && !transquant_bypass_flag &&                                       |            |
| ( log2TBSize <= log2MaxTSsize ) && compID==COMPONENT_Y )                                 |            |
| ts_flag [x0][y0][compID]                                                                 | ae(v)      |
| ... syntaxes related to last_pos [x, y],                                                 |            |
| coeff_sblk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ... |            |
| if( !apt_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]             |            |
| && compID==COMPONENT_Y && numSig >= ptNumSigTH &&                                        |            |
| max( log2TBSize, log2TBSize ) <= maxPTSize                                               |            |
| pt_idx[x0][y0]                                                                           | ae(v)      |
| ...                                                                                      |            |
| if( !st_enabled_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]      |            |
| && compID==COMPONENT_Y && numSig >= stNumSigTH )                                         |            |
| st_idx[x0][y0]                                                                           | ae(v)      |
| }                                                                                        |            |

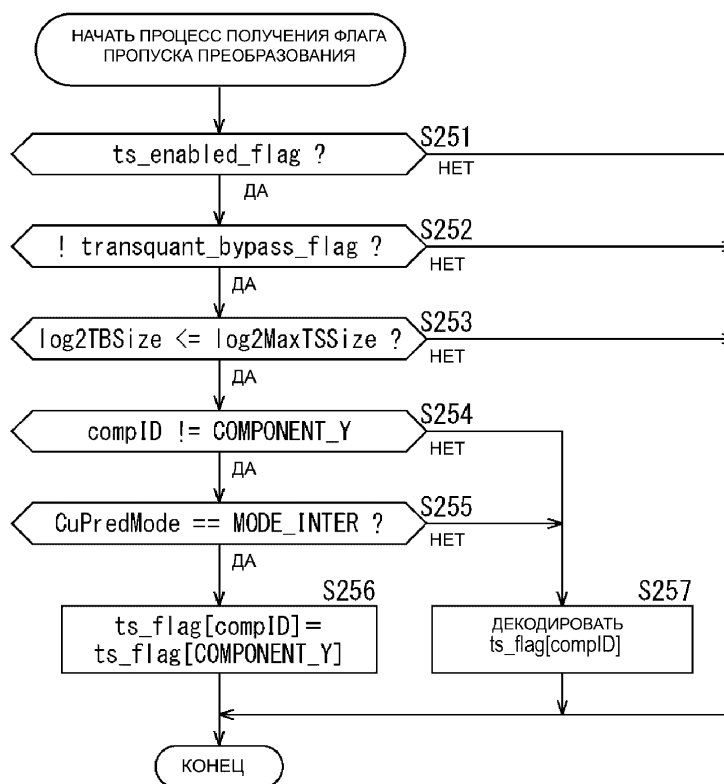
Фиг. 29



Фиг. 30

| Дескриптор |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <pre>residual_coding( x0, y0, log2IBSize, log2IBSize, compID ) {<br/>  if( !ts_enabled_flag &amp;&amp; !transquant_bypass_flag &amp;&amp;<br/>    ( log2IBSize &lt;= log2MaxTSSize ) &amp;&amp;<br/>    ( CuPredMode==MODE_INTRA    CuPredMode==MODE_INTER &amp;&amp; compID==COMPONENT_Y) )<br/>    ts_flag[x0][y0][compID]<br/>    ... syntaxes related to last pos. ix, y.<br/>    coeff_sbk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level/remaining/coeff_sign_flag ...<br/>  if(apt_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>    &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= ptNumSigTH &amp;&amp;<br/>    max(log2IBSize, log2IBSize)&lt;= maxPTSize)<br/>    pt_idx[x0][y0]<br/>    ...<br/>  if( (st_enabled_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>    &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= stNumSigTH ) )<br/>    st_idx[x0][y0]<br/>}</pre> |
| SYN21      | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| SYN22      | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| SYN22      | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

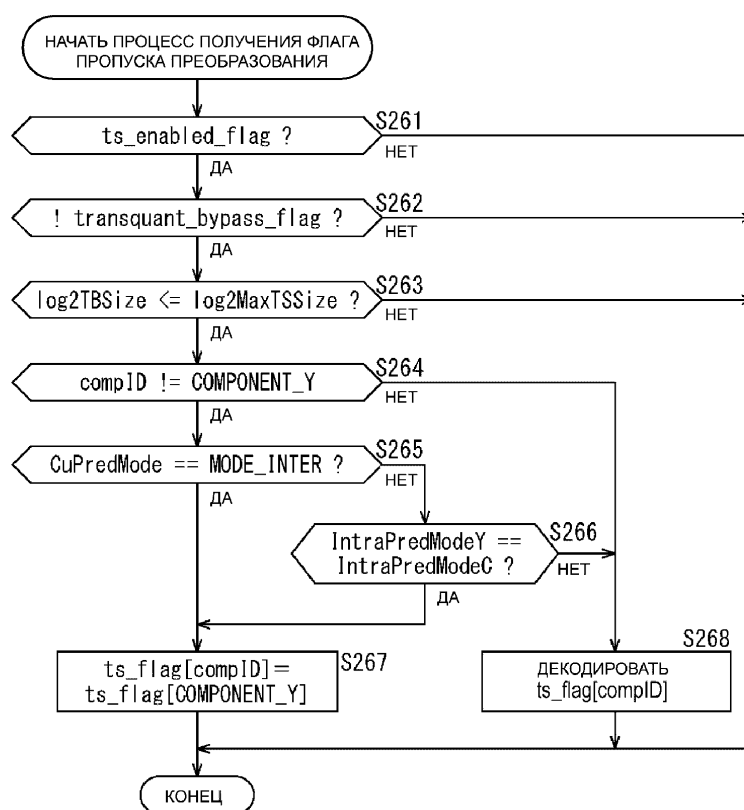
Фиг. 31



Фиг. 32

| Дескриптор |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <pre>residual_coding( x0 y0, log2TBSize, log2TBSize, compID ) {<br/>  if( !ts_enabled_flag &amp;&amp; !transquant_bypass_flag &amp;&amp;<br/>    ( log2TBSize &lt;= log2MaxTSSize ) &amp;&amp;<br/>    ( CuPredMode==MODE_INTRA &amp;&amp; compID==COMPONENT_Y   <br/>      CuPredMode==MODE_INTRA &amp;&amp; compID==COMPONENT_Y &amp;&amp; IntraPredModeC<br/>      CuPredMode==MODE_INTER &amp;&amp; compID==COMPONENT_Y ) )<br/>    ts_flag[x0][y0][compID]<br/>    ... syntaxes related to last_pos {x,y}.<br/>    coeff_blk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ...<br/>    if( !apt_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>      &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= ptNumSigTH &amp;&amp;<br/>      max(log2TBSize, log2TBSize)&lt;= maxPTSize )<br/>      pt_idx[x0][y0]<br/>    ...<br/>    if( !st_enabled_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>      &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= stNumSigTH ) )<br/>      st_idx[x0][y0]<br/>}</pre> |
| }-SYN21    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|            | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| }-SYN22    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|            | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| }-SYN22    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|            | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|            | ae(v)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

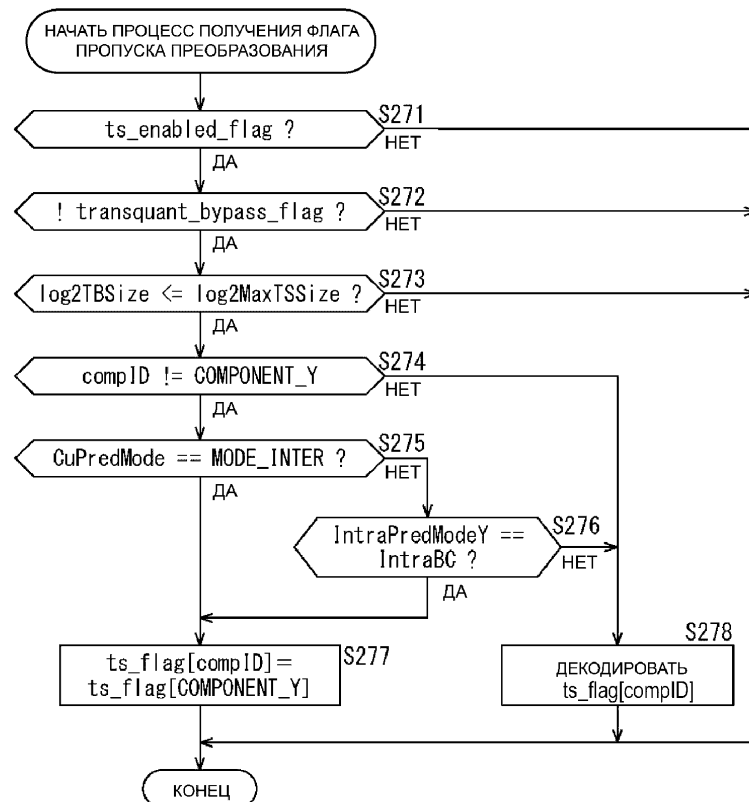
Фиг. 33



Фиг. 34

| Дескриптор |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <pre>residual_coding( x0 y0, log2TBSize, log2TBSize, compID ) {<br/>  if( !ts_enabled_flag &amp;&amp; !transquant_bypass_flag &amp;&amp;<br/>    ( log2TBSize &lt;= log2MaxTSSize ) &amp;&amp;<br/>    ( CuPredMode==MODE_INTRA &amp;&amp; compID==COMPONENT_Y   <br/>      CuPredMode==MODE_INTRA &amp;&amp; compID!=COMPONENT_Y &amp;&amp; IntraPredModeY!=Intra8C<br/>        CuPredMode==MODE_INTER &amp;&amp; compID==COMPONENT_Y ) )<br/>    ts_flag[x0][y0][compID]<br/>    ... syntaxes related to last_pos {x, y}.<br/>    coeff_blk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ...<br/>    if( !pt_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>      &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= ptNumSigTH &amp;&amp;<br/>        max(log2TBSize, log2TBSize)&lt;= maxPTSize )<br/>      pt_idx[x0][y0]<br/>    ...<br/>    if( !st_enabled_flag[x0][y0] &amp;&amp; !transquant_bypass_flag &amp;&amp; !ts_flag[x0][y0][compID]<br/>      &amp;&amp; compID==COMPONENT_Y &amp;&amp; numSig &gt;= stNumSigTH )<br/>      st_idx[x0][y0]<br/>  }</pre> |
| }-SYN21    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|            | <pre>ae(v)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| }-SYN22    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|            | <pre>ae(v)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| }-SYN22    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|            | <pre>ae(v)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|            | <pre>ae(v)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

Фиг. 35



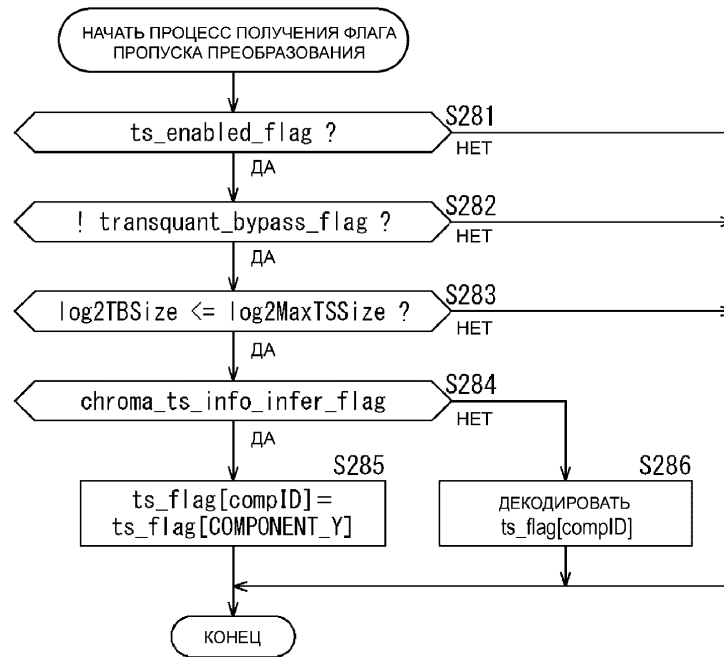
Фиг. 36

|                                                                   |            |
|-------------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBHSizeL, log2TBHSizeL)               | Дескриптор |
| ...                                                               |            |
| for (compID=0; compID<numValIdChannel; compID++) {                |            |
| cbf[x0][y0][compID]                                               | ae(v)      |
| if ( cbf[x0][y0][compID] ) {                                      |            |
| if ( cbf[x0][y0][COMPONENT_Y] && apt_enabled_flag ) {             |            |
| apt_flag[x0][y0]                                                  | ae(v)      |
| }                                                                 |            |
| if (ts_flag[x0][y0][COMPONENT_Y] && compID==COMPONENT_cb)         |            |
| chroma_ts_info_infer_flag[x0][y0]                                 | ae(v)      |
| log2TBHSize = compID==COMPONENT_Y? log2TBHSizeL>>1                |            |
| log2TBHSize = compID==COMPONENT_Y? log2TBHSizeL : log2TBHSizeL>>1 |            |
| residual_coding( x0, y0, log2TBHSize, log2TBHSize, compID)        |            |
| }                                                                 |            |
| }                                                                 |            |
| ...                                                               |            |
| }                                                                 |            |

Фиг. 37

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------|
| residual_coding ( x0 y0 log2TBSize log2TBSize compID ) {<br>if ( ts_enabled_flag && !transquant_bypass_flag &&<br>( log2TBSize <= log2MaxTSSize ) &&<br>( compID==COMPONENT_Y && !chroma_ts_info_infer_flag ) )<br>ts_flag [x0][y0][compID]<br>... syntaxes related to last_pos {x,y},<br>coeff_sbk_flag/sig_flag/gr1_flag/gr2_flag/coeff_abs_level_remaining/coeff_sign_flag ...<br>if ( !apt_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]<br>&& compID==COMPONENT_Y && numSig >= ptNumSigTH &&<br>max ( log2TBSize, log2TBSize ) <= maxPSize )<br>pt_idx [x0][y0]<br>...<br>if ( !st_enabled_flag[x0][y0] && !transquant_bypass_flag && !ts_flag[x0][y0][compID]<br>&& compID==COMPONENT_Y && numSig >= stNumSigTH )<br>st_idx [x0][y0]<br>}<br>} | Дескриптор | }-SYN21 |
| as {v}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |         |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            | }-SYN22 |
| as {v}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |         |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            | }-SYN22 |
| as {v}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |         |

Фиг. 38



Фиг. 39

| ИДЕНТИФИКАТОР st_idx(CbCr) ВТОРИЧНОГО ПРЕОБРАЗОВАНИЯ ЦВЕТНОСТИ |                           |                                                              |                                  |                                  |   |                            |                                                       |                                                                 |  |
|----------------------------------------------------------------|---------------------------|--------------------------------------------------------------|----------------------------------|----------------------------------|---|----------------------------|-------------------------------------------------------|-----------------------------------------------------------------|--|
| №                                                              | ВВОД ЦВЕТНОСТИ ИЗ ЯРКОСТИ | УСЛОВИЕ ДЛЯ ИСКЛЮЧЕНИЯ ФЛАГА ПРОПУСКА ПРЕДСКАЗАНИЯ ЦВЕТНОСТИ |                                  |                                  |   | ДРУГИЕ ХАРАКТЕРИСТИКИ ФЛАГ | ПРИМЕЧАНИЕ                                            | РЕЗУЛЬТАТ                                                       |  |
|                                                                |                           | ИНТРА-CU                                                     |                                  | ИНТЕР-CU                         |   |                            |                                                       |                                                                 |  |
|                                                                |                           | БЕЗ ДОПОЛНИТЕЛЬНОГО УСЛОВИЯ                                  | IntraPredModeY == IntraPredModeC | IntraPredModeY == IntraPredModeC |   |                            |                                                       |                                                                 |  |
|                                                                |                           |                                                              |                                  | IntraPredModeC                   |   |                            |                                                       |                                                                 |  |
| #0                                                             |                           |                                                              |                                  |                                  |   |                            | СИГНАЛИЗАЦИЯ ОДНОГО ИЗ st_idx(croma_st_idx) ДЛЯ Cb/Cr |                                                                 |  |
| #1                                                             | ×                         | ×                                                            |                                  |                                  |   |                            |                                                       | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                             |  |
| #2                                                             | ×                         |                                                              |                                  |                                  | × |                            |                                                       | ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОДИРОВАНИЯ С МЕЖКОЛОННЫМ ПРЕДСКАЗАНИЕМ |  |
| #3                                                             | ×                         |                                                              | ×                                |                                  | × |                            |                                                       | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                             |  |
| #4                                                             | ×                         |                                                              |                                  | ×                                | × |                            |                                                       | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                             |  |
| #5                                                             | ×                         |                                                              |                                  |                                  |   | ×                          |                                                       | УМЕНЬШЕНИЕ ИЗДЕРЖЕК                                             |  |

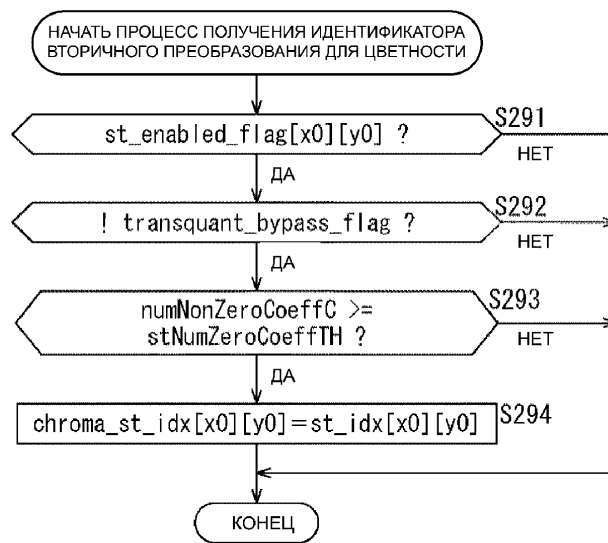
Фиг. 40

|                                                                  |            |
|------------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBSsizeL, log2TBSsizeL )             | Дескриптор |
| ...                                                              |            |
| for( compID=0; compID<numValidChannel; compID++) {               |            |
| obf[x0][y0][compID]                                              | ae(y)      |
| if( obf[x0][y0][compID] ) {                                      |            |
| if( obf[x0][y0][COMPONENT_Y] && apt_enabled_flag ) {             |            |
| apt_flag[x0][y0]                                                 | ae(y)      |
| }                                                                |            |
| log2TBSsize = compID==COMPONENT_Y? log2TBSsizeL : log2TBSsizeL>1 |            |
| log2TBSsize = compID==COMPONENT_Y? log2TBSsizeL : log2TBSsizeL>1 |            |
| residual_coding( x0, y0, log2TBSsize, log2TBSsize, compID )      |            |
| }                                                                |            |
| }                                                                |            |
| ...                                                              |            |
| numNonZeroCoeff = 0                                              |            |
| numNonZeroCoeffC = 0                                             |            |
| if ( !its_flag[COMPONENT_Y] )                                    |            |
| numNonZeroCoeff = countNonZeroCoeff (COMPONENT_Y)                |            |
| if( (st_enabled_flag[x0][y0] && 'transquant_bypass_flag &&       |            |
| numNonZeroCoeff >= stNumNonZeroCoeffTH ) {                       |            |
| st_idx[x0][y0]                                                   | ae(y)      |
| }                                                                |            |
| if ( !its_flag[x0][y0][COMPONENT_Cb] )                           |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)              |            |
| if ( !its_flag[x0][y0][COMPONENT_Cr] )                           |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)             |            |
| if( (st_enabled_flag[x0][y0] && 'transquant_bypass_flag &&       |            |
| numNonZeroCoeffC >= stNumNonZeroCoeffTH ) {                      |            |
| chroma_st_idx[x0][y0]                                            | ae(y)      |
| }                                                                |            |
| }                                                                |            |

Фиг. 41

|                                                                                                            |            |
|------------------------------------------------------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBWSizeL, log2TBHSizeL )                                                       | Дескриптор |
| ...                                                                                                        |            |
| numNonZeroCoeffY = 0                                                                                       |            |
| numNonZeroCoeffC = 0                                                                                       |            |
| if ( !its_flag[COMPONENT_Y] )                                                                              |            |
| numNonZeroCoeffY = countNonZeroCoeff (COMPONENT_Y)                                                         |            |
| if ( !st_enabled_flag[x0][y0] && !transquant_bypass_flag &&<br>numNonZeroCoeffY >= stNumNonZeroCoeffTH ) { |            |
| st_idx[x0][y0]                                                                                             | ae (v)     |
| }                                                                                                          |            |
| if ( !its_flag[x0][y0][COMPONENT_Cb] )                                                                     |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)                                                        |            |
| if ( !its_flag[x0][y0][COMPONENT_Cr] )                                                                     |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)                                                       |            |
| if ( !st_enabled_flag[x0][y0] && !transquant_bypass_flag &&<br>numNonZeroCoeffC >= stNumNonZeroCoeffTH ) { |            |
| chroma_st_idx[x0][y0] = st_idx[x0][y0]                                                                     |            |
| }                                                                                                          |            |
| }                                                                                                          |            |

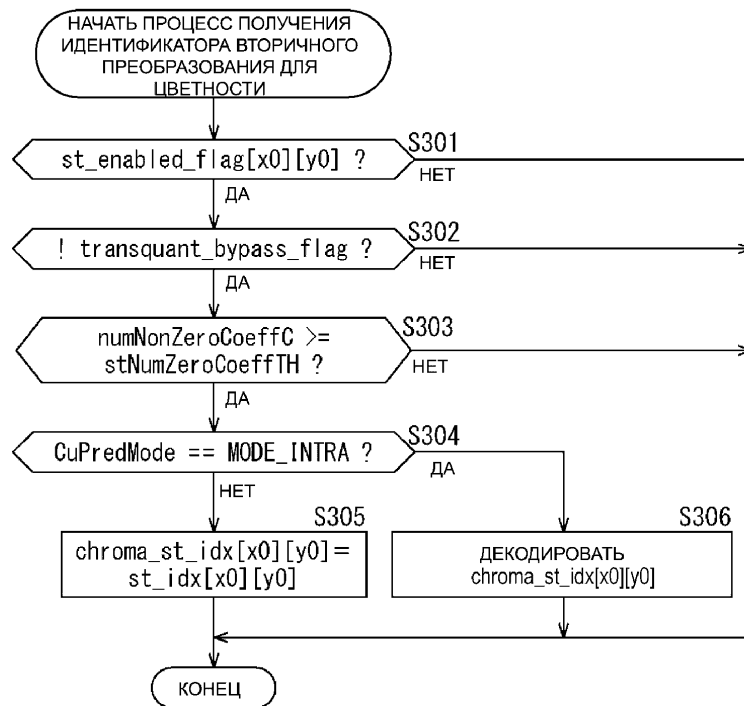
Фиг. 42



Фиг. 43

|                                                            |            |
|------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBWSizeL, log2TBHSizeL)        | Дескриптор |
| ...                                                        |            |
| numNonZeroCoeffY = 0                                       |            |
| numNonZeroCoeffC = 0                                       |            |
| if ( !its_flag[COMPONENT_Y]                                |            |
| numNonZeroCoeffY = countNonZeroCoeff (COMPONENT_Y)         |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffY >= stNumNonZeroCoeffTH ) ) {              |            |
| st_idx[x0][y0]                                             | ae (v)     |
| }                                                          |            |
| if ( !its_flag[x0][y0] [COMPONENT_Cb])                     |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)        |            |
| if ( !its_flag[x0][y0] [COMPONENT_Cr])                     |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)       |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffC >= stNumNonZeroCoeffTH ) ) {              |            |
| if (C_PredMode==MODE_INTRA)                                |            |
| chroma_st_idx[x0][y0]                                      | ae (v)     |
| else                                                       |            |
| chroma_st_idx[x0][y0] = st_idx[x0][y0]                     |            |
| }                                                          |            |
| }                                                          |            |

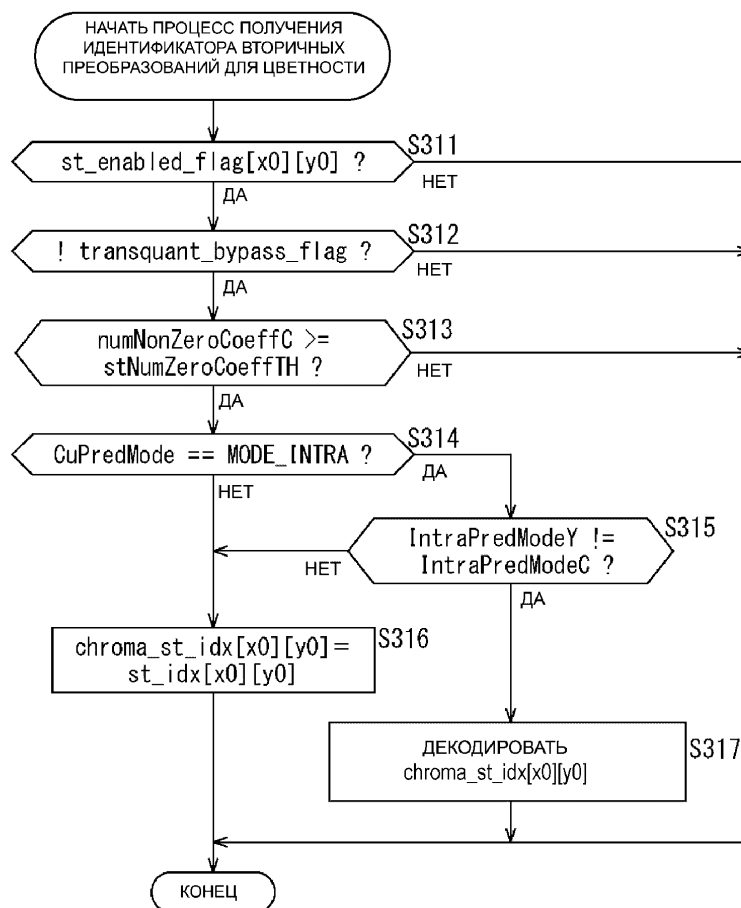
Фиг. 44



Фиг. 45

|                                                                  |            |
|------------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBWSizeL, log2TBHSizeL)              | Дескриптор |
| ...                                                              |            |
| numNonZeroCoeffY = 0                                             |            |
| numNonZeroCoeffC = 0                                             |            |
| if ( !its_flag[COMPONENT_Y]                                      |            |
| numNonZeroCoeffY = countNonZeroCoeff (COMPONENT_Y)               |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag &&       |            |
| numNonZeroCoeffY >= stNumNonZeroCoeffTH ) ) {                    |            |
| st_idx[x0][y0]                                                   | ae (v)     |
| }                                                                |            |
| if ( !its_flag[x0][y0][COMPONENT_Cb]                             |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)              |            |
| if ( !its_flag[x0][y0][COMPONENT_Cr]                             |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)             |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag &&       |            |
| numNonZeroCoeffC >= stNumNonZeroCoeffTH ) ) {                    |            |
| if ( (0,PredMode==MODE_INTRA && IntraPredModeY!=IntraPredModeC)) |            |
| chroma_st_idx[x0][y0]                                            | ae (v)     |
| else                                                             |            |
| chroma_st_idx[x0][y0] = st_idx[x0][y0]                           |            |
| }                                                                |            |
| }                                                                |            |

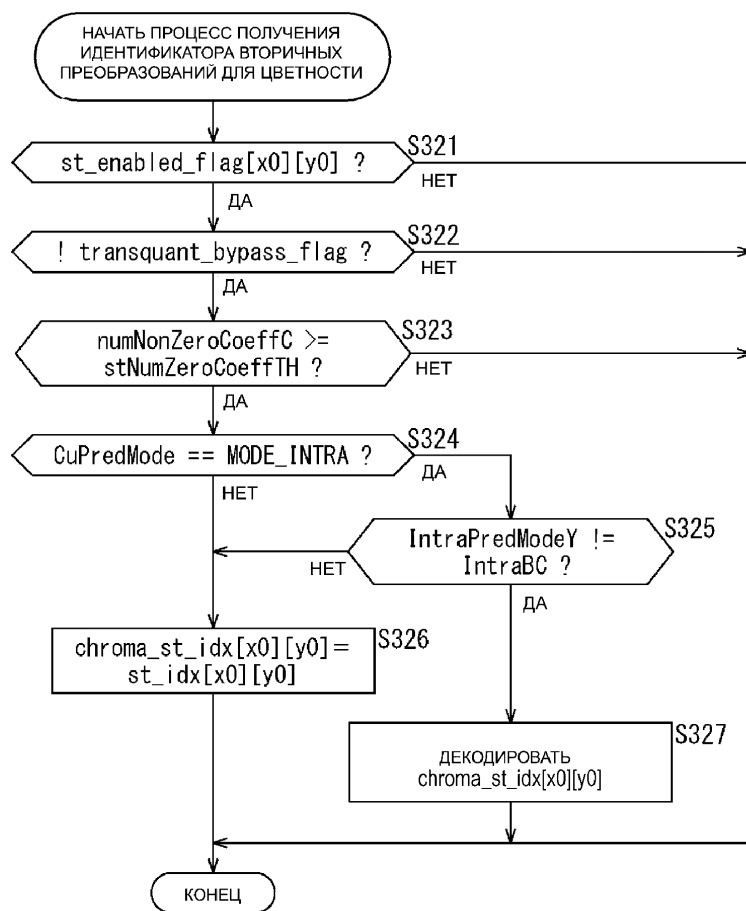
Фиг. 46



Фиг. 47

|                                                            |            |
|------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBWSizeL, log2TBHSizeL)        | Дескриптор |
| ...                                                        |            |
| numNonZeroCoeffY = 0                                       |            |
| numNonZeroCoeffC = 0                                       |            |
| if ( !its_flag[COMPONENT_Y]                                |            |
| numNonZeroCoeffY = countNonZeroCoeff (COMPONENT_Y)         |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffY >= stNumNonZeroCoeffTH ) ) {              |            |
| st_idx[x0][y0]                                             | ae (v)     |
| }                                                          |            |
| if ( !its_flag[x0][y0][COMPONENT_Cb]                       |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)        |            |
| if ( !its_flag[x0][y0][COMPONENT_Cr]                       |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)       |            |
| if( (st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffC >= stNumNonZeroCoeffTH ) ) {              |            |
| if ( (0,PredMode==MODE_INTRA && IntraPredModeY!=IntraBC))  |            |
| chroma_st_idx[x0][y0]                                      | ae (v)     |
| else                                                       |            |
| chroma_st_idx[x0][y0] = st_idx[x0][y0]                     |            |
| }                                                          |            |
| }                                                          |            |

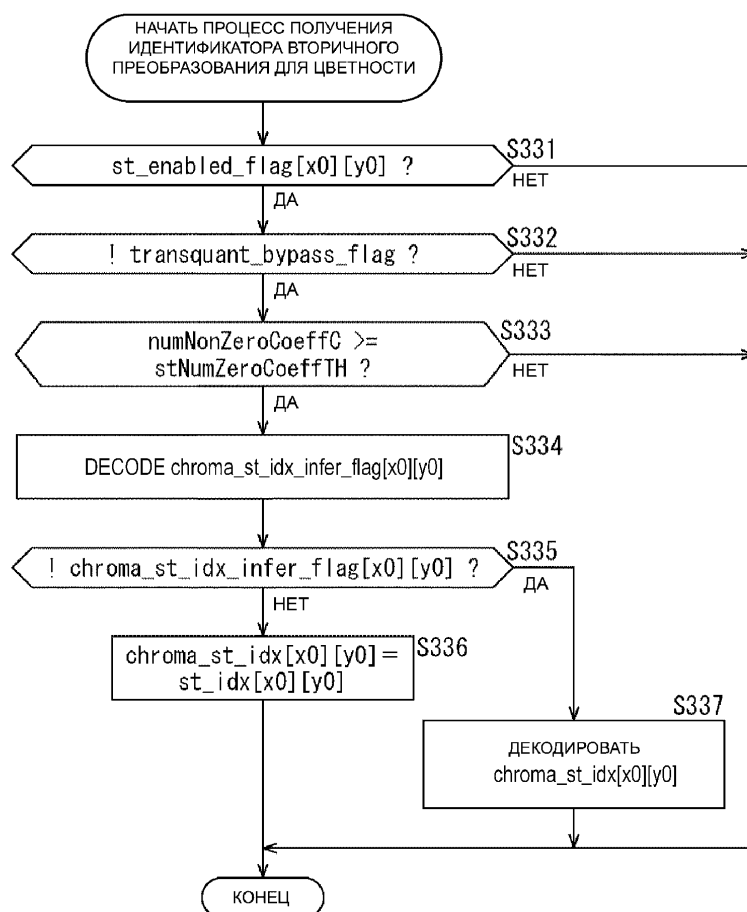
Фиг. 48



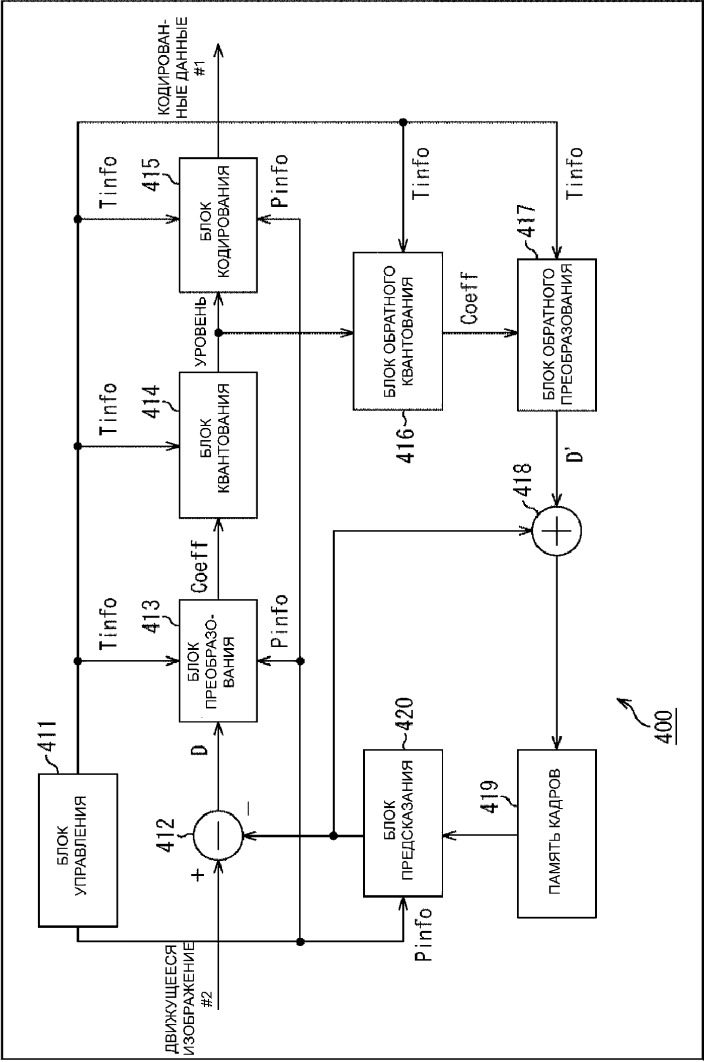
Фиг. 49

|                                                             |            |
|-------------------------------------------------------------|------------|
| transform_unit( x0, y0, log2TBWSizeL, log2TBSizeL)          | Дескриптор |
| ...                                                         |            |
| numNonZeroCoeffY = 0                                        |            |
| numNonZeroCoeffC = 0                                        |            |
| if ( !ts_flag[COMPONENT_Y]                                  |            |
| numNonZeroCoeffY = countNonZeroCoeff (COMPONENT_Y)          |            |
| if ( !st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffY >= stNumNonZeroCoeffTH ) {                 |            |
| st_idx[x0][y0]                                              | ae(v)      |
| }                                                           |            |
| if ( !ts_flag[x0][y0] [COMPONENT_Cb])                       |            |
| numNonZeroCoeffC = countNonZeroCoeff (COMPONENT_Cb)         |            |
| if ( !ts_flag[x0][y0] [COMPONENT_Cr])                       |            |
| numNonZeroCoeffC += countNonZeroCoeff (COMPONENT_Cr)        |            |
| if ( !st_enabled_flag[x0][y0] && !transquant_bypass_flag && |            |
| numNonZeroCoeffC >= stNumNonZeroCoeffTH ) {                 |            |
| chroma_st_idx_infer_flag[x0][y0]                            | ae(v)      |
| if ( !chroma_st_idx_infer_flag[x0][y0] )                    |            |
| chroma_st_idx[x0][y0]                                       | ae(v)      |
| else                                                        |            |
| chroma_st_idx[x0][y0] = st_idx[x0][y0]                      |            |
| }                                                           |            |
| }                                                           |            |

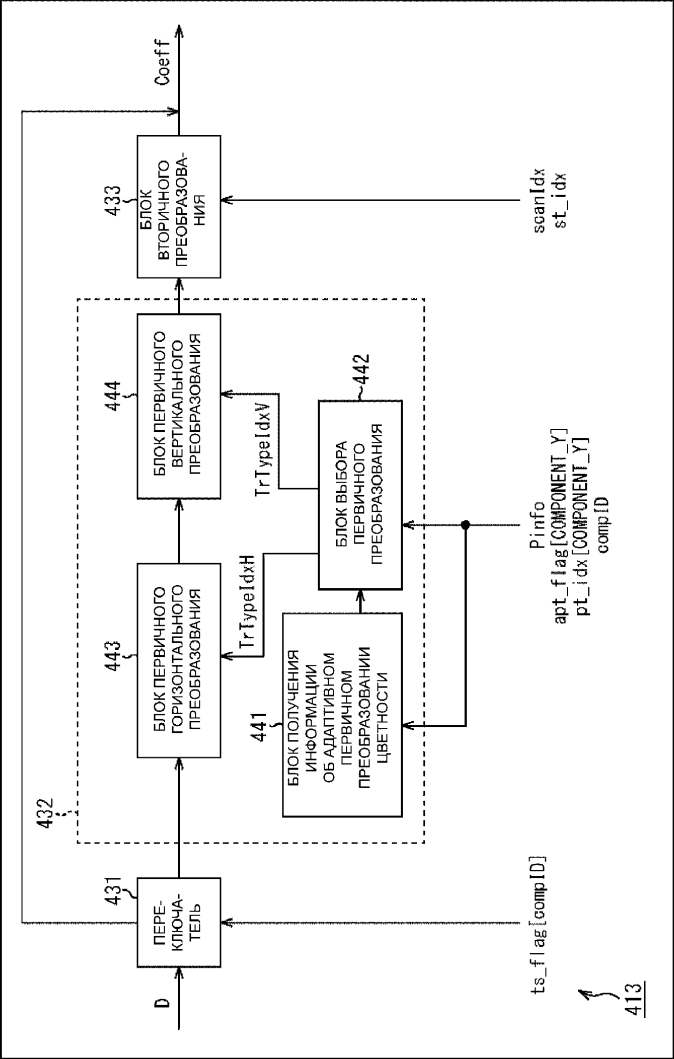
Фиг. 50



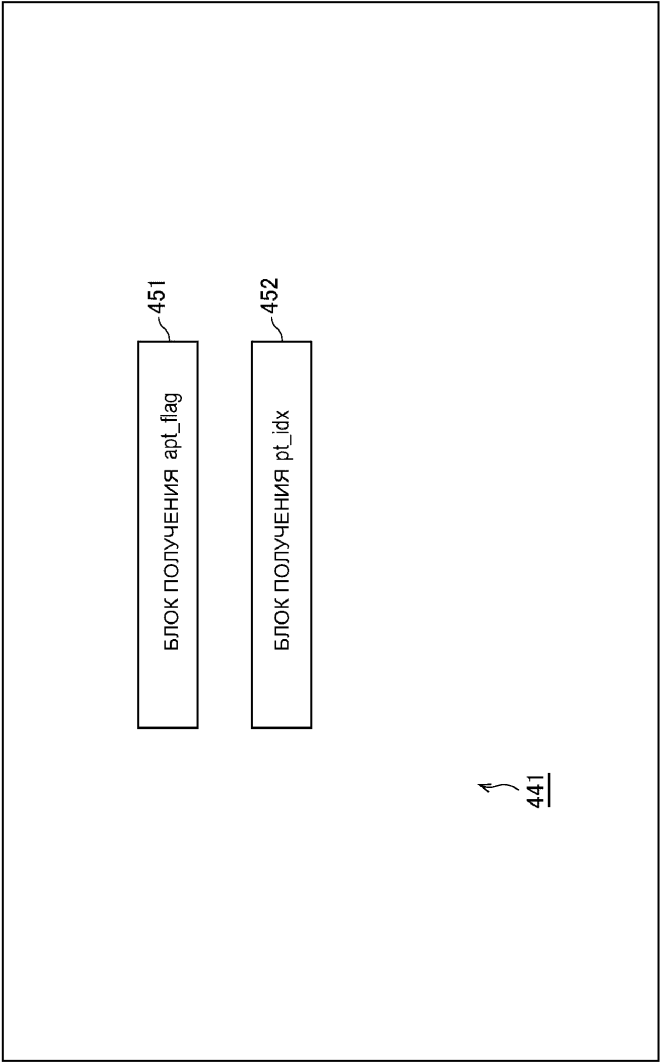
Фиг. 51



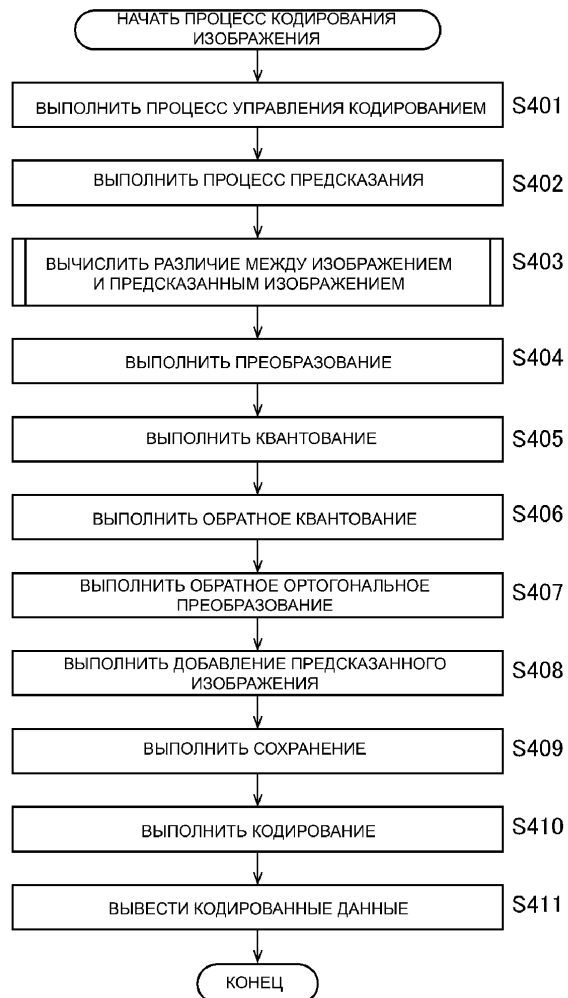
Фиг. 52



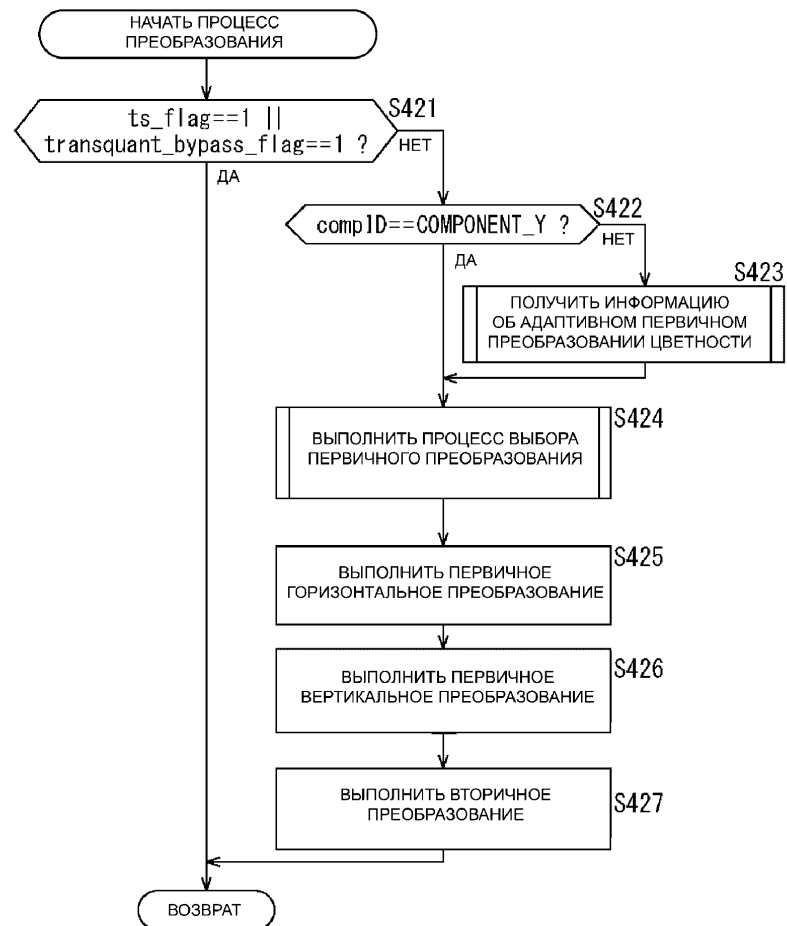
Фиг. 53



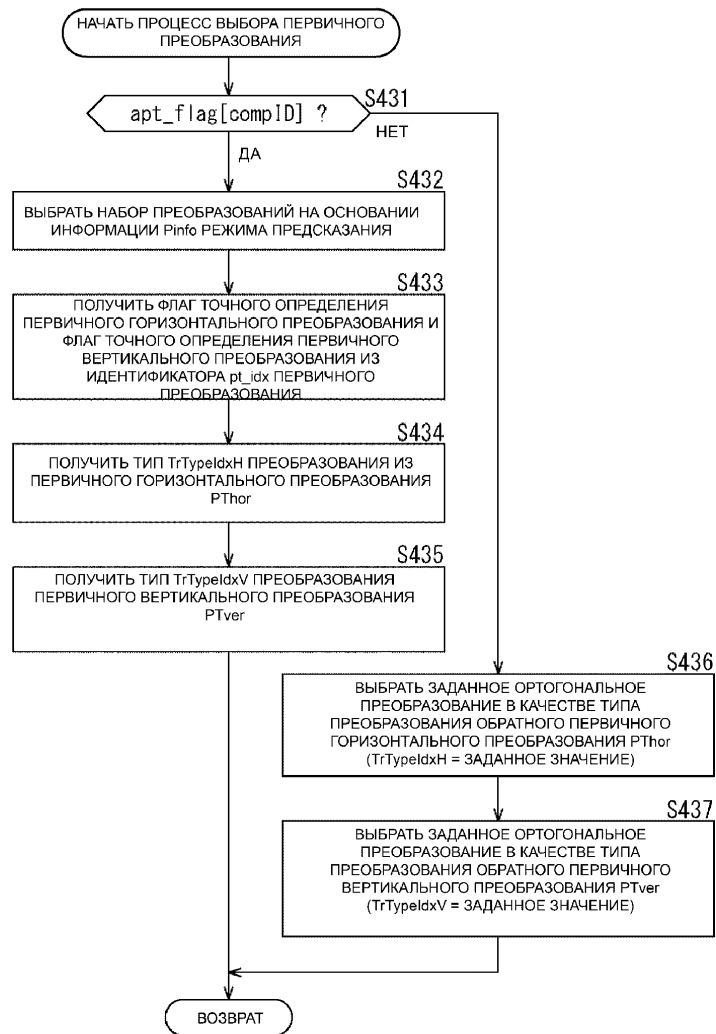
Фиг. 54



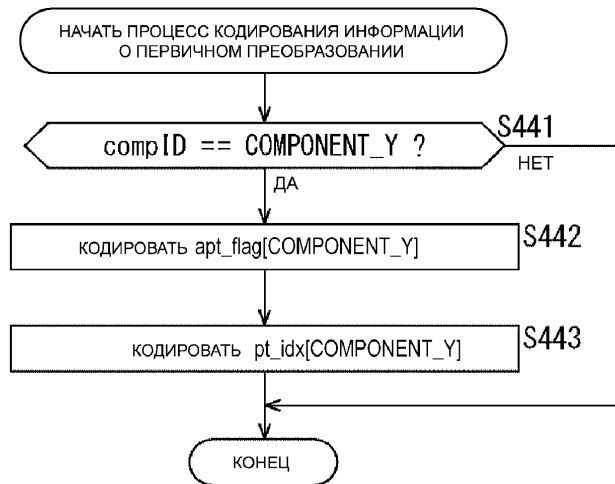
Фиг. 55



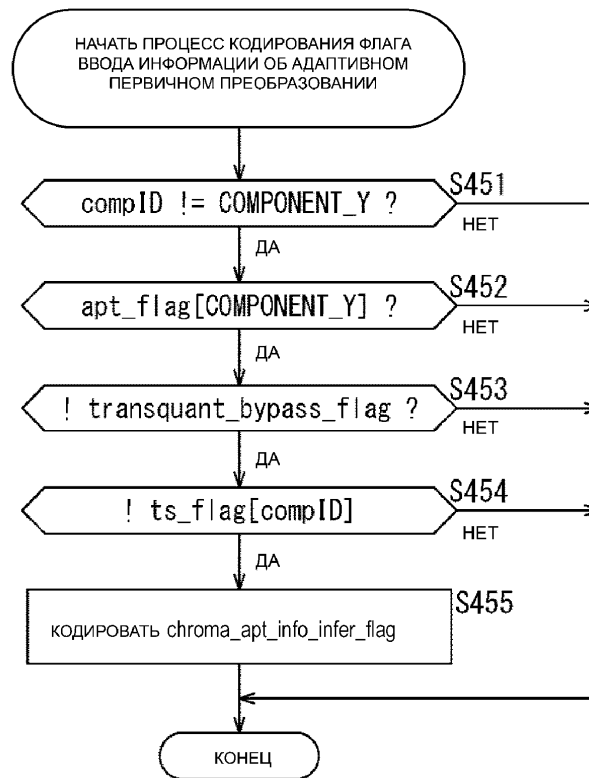
Фиг. 56



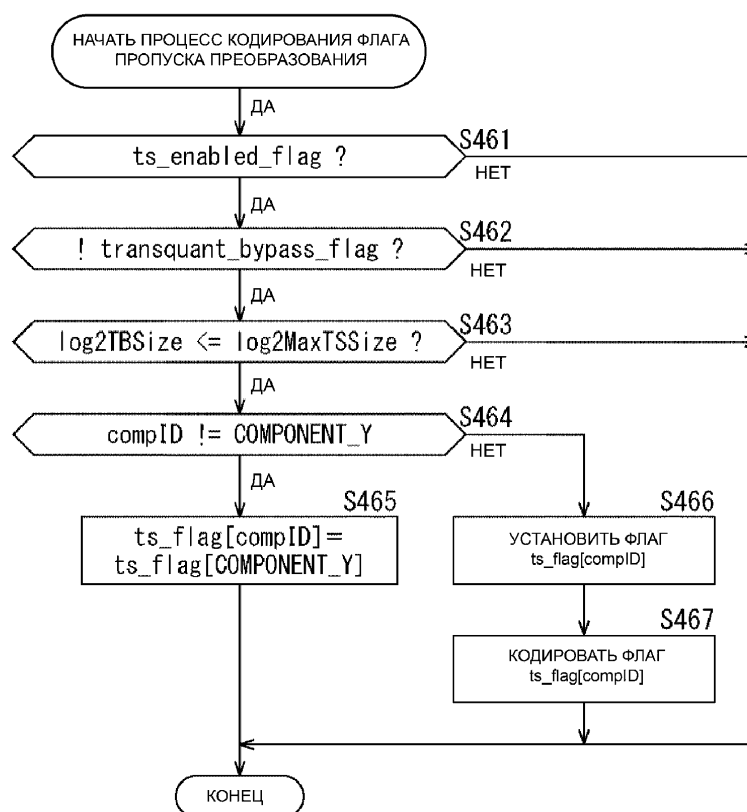
Фиг. 57



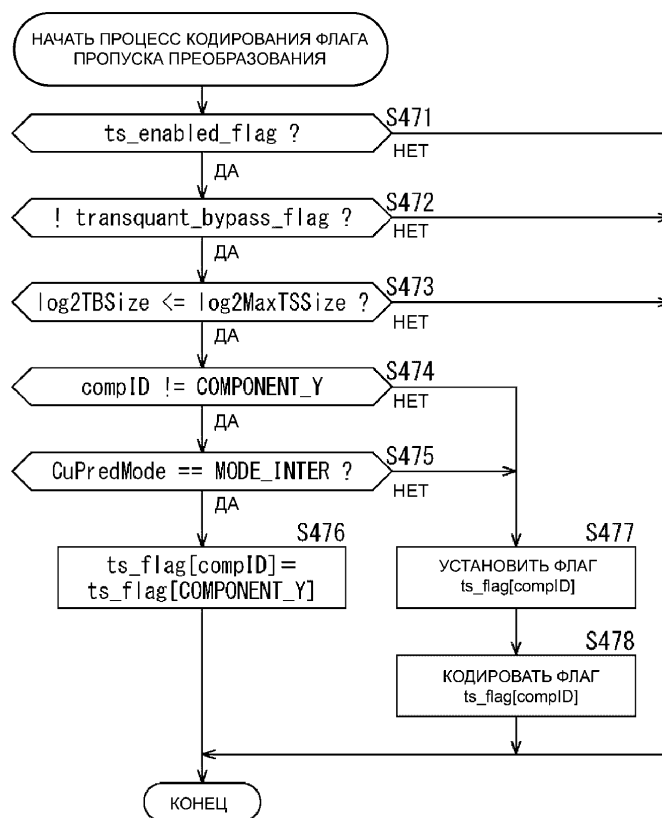
Фиг. 58



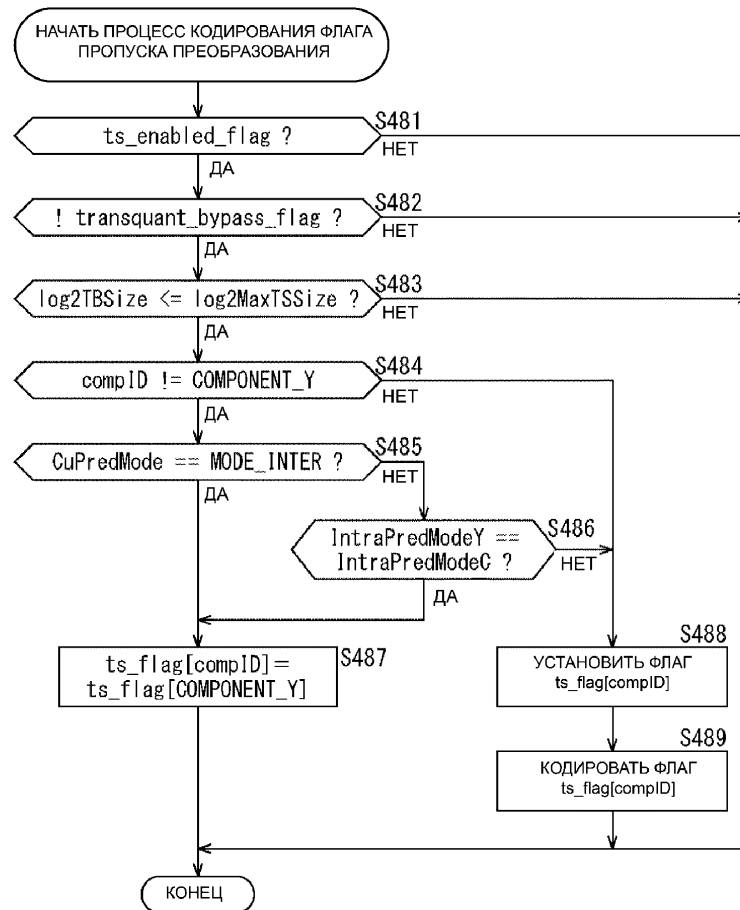
Фиг. 59



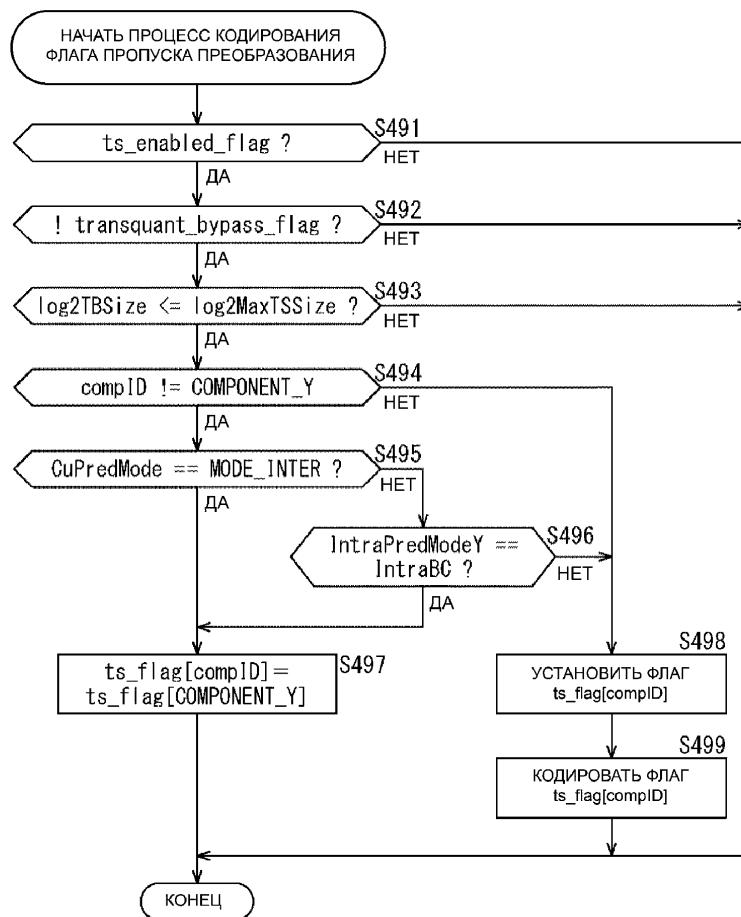
Фиг. 60



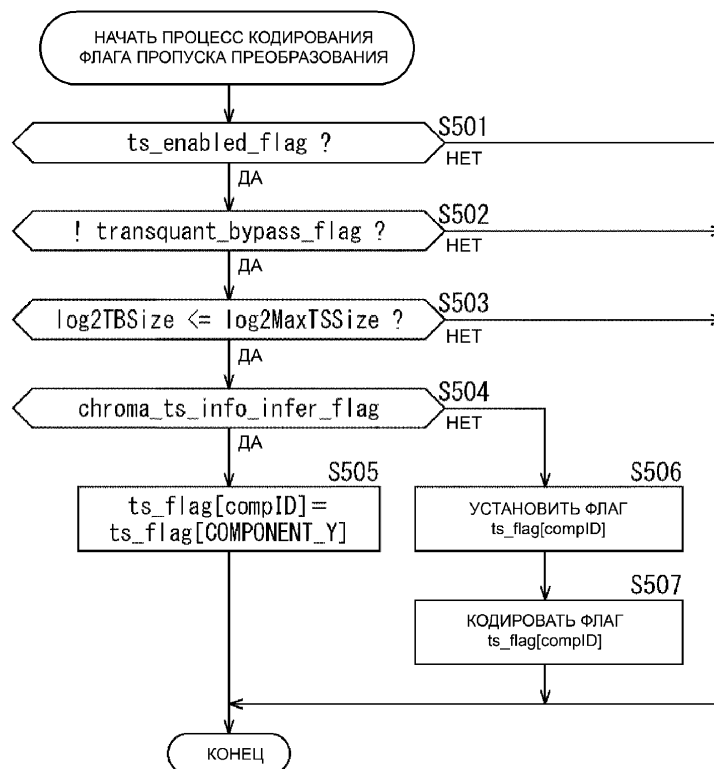
Фиг. 61



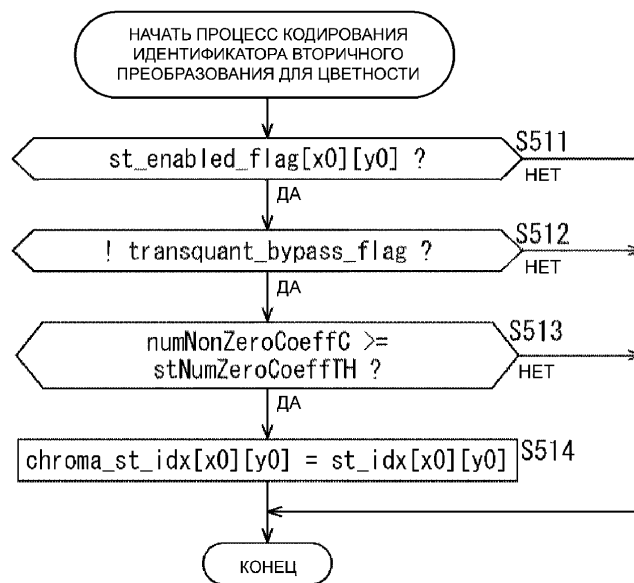
Фиг. 62



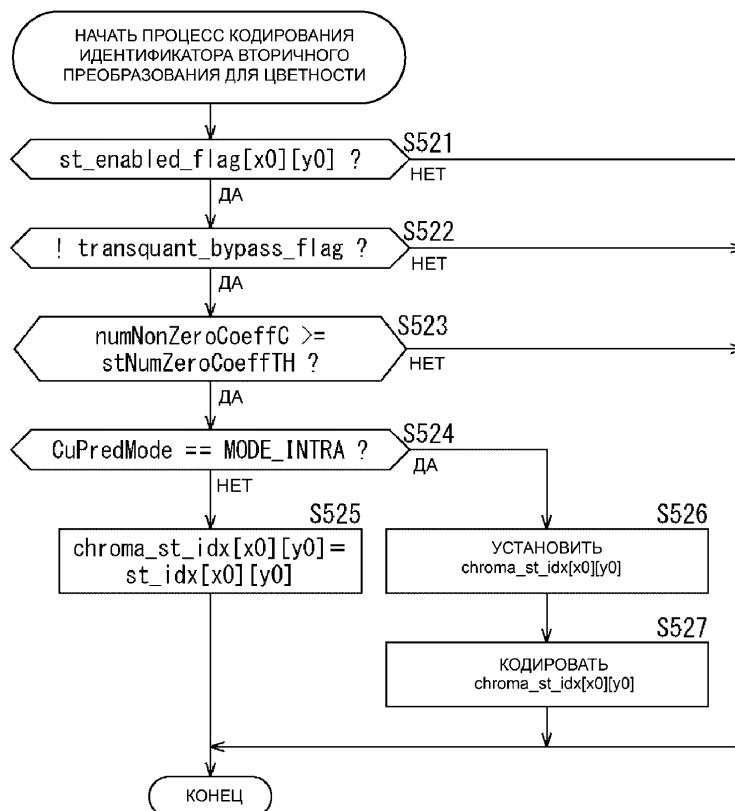
Фиг. 63



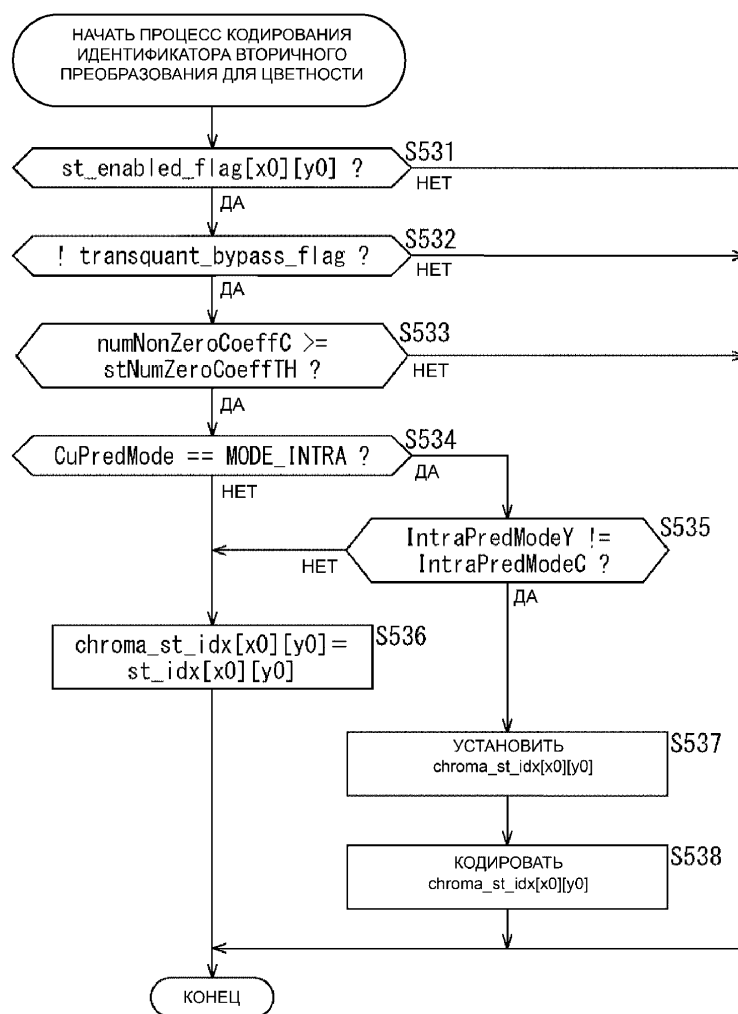
Фиг. 64



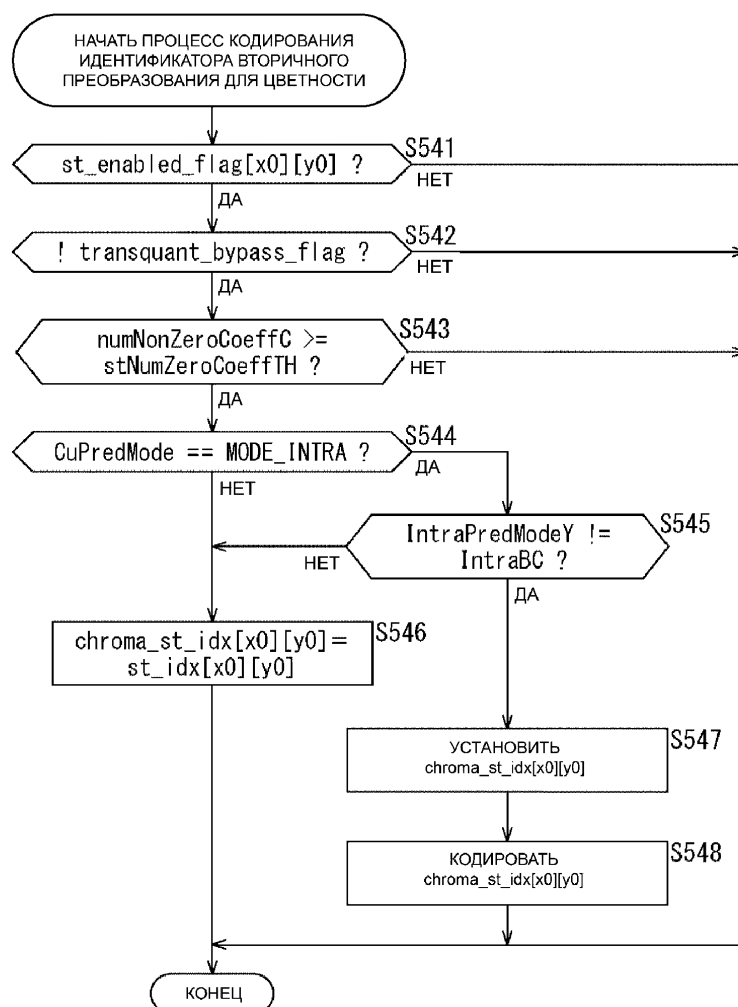
Фиг. 65



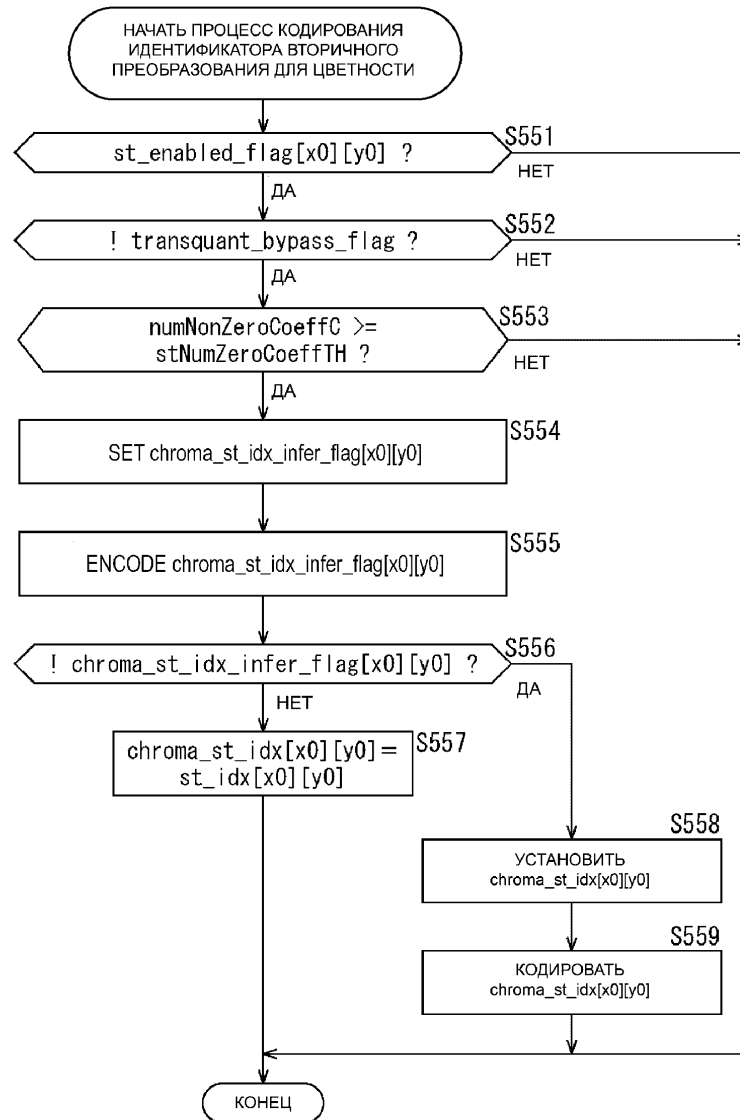
Фиг. 66



Фиг. 67

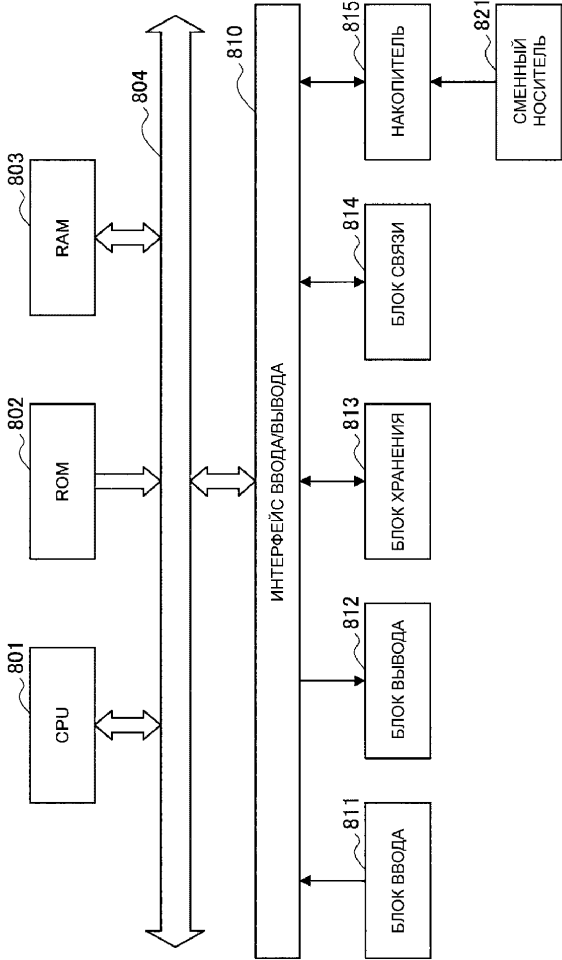


Фиг. 68

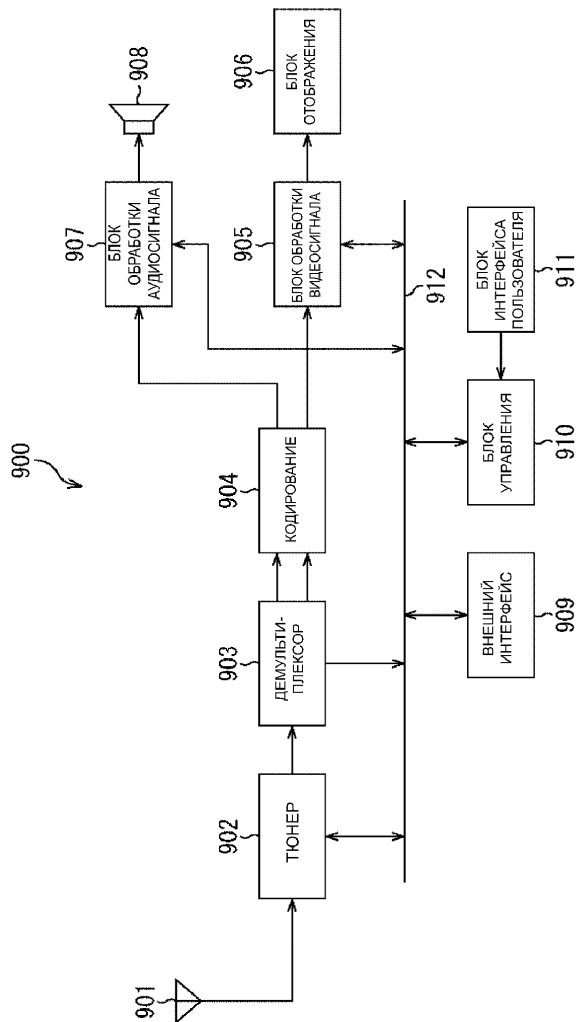


Фиг. 69

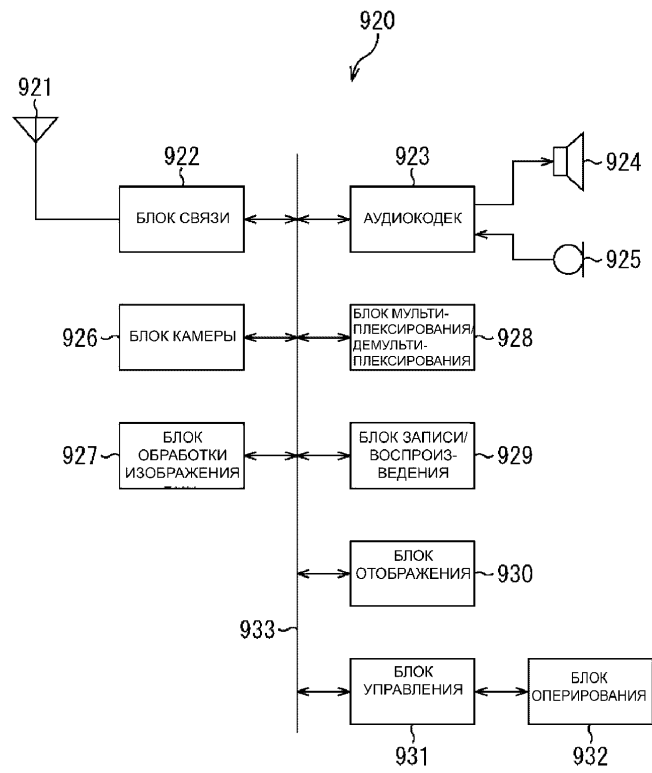
800



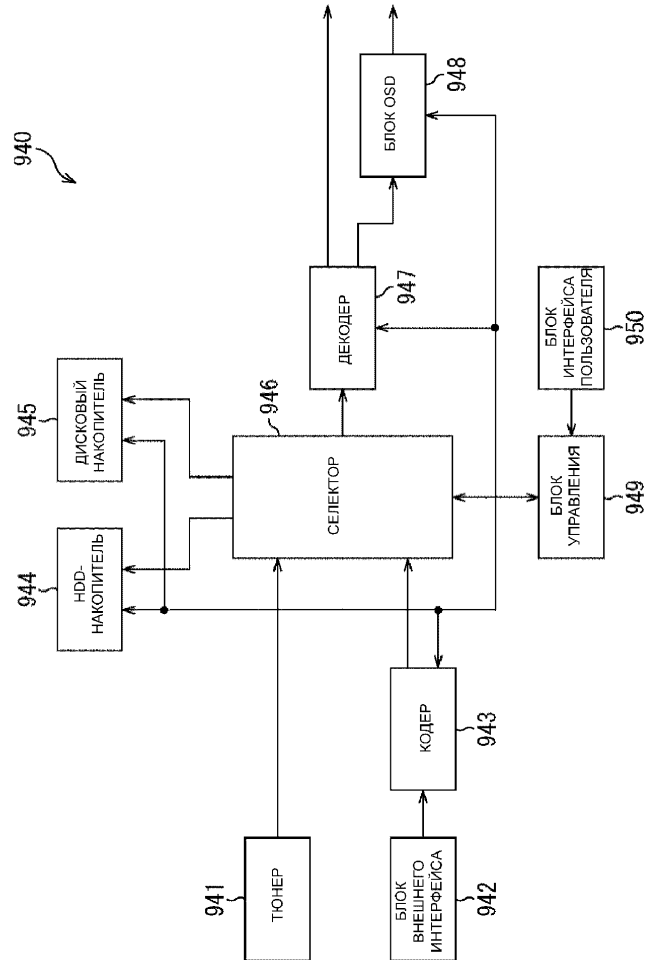
Фиг. 70



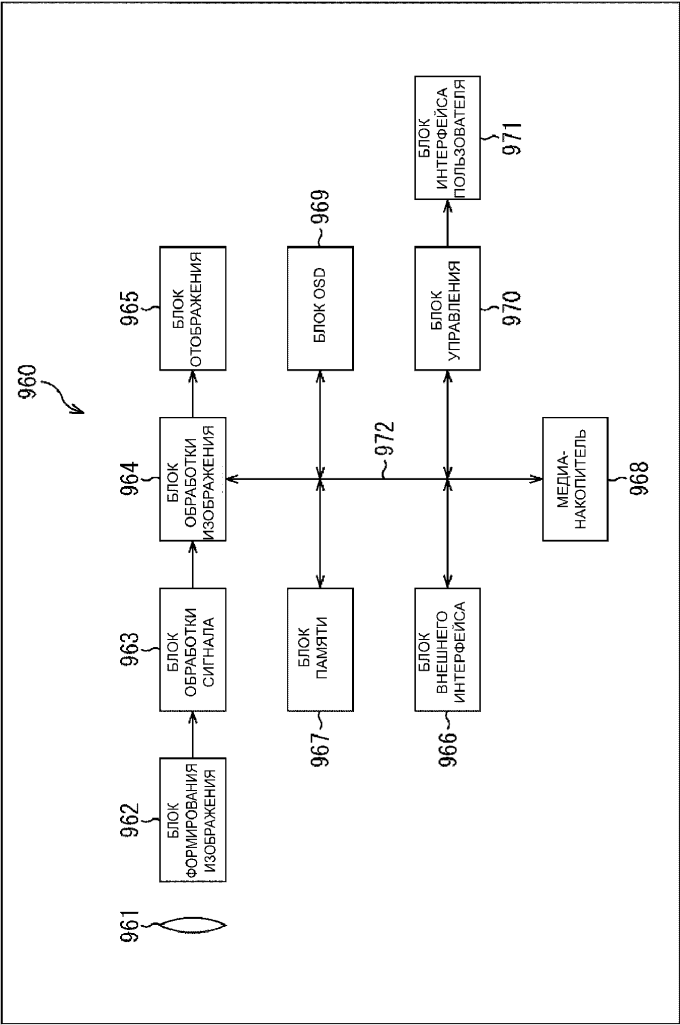
Фиг. 71



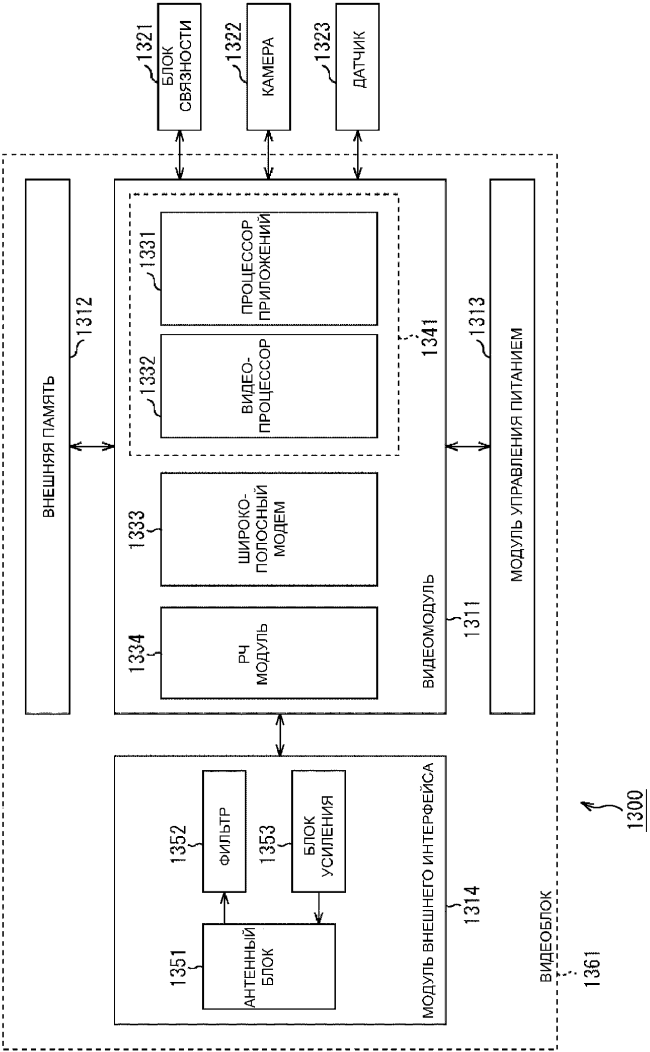
Фиг. 72



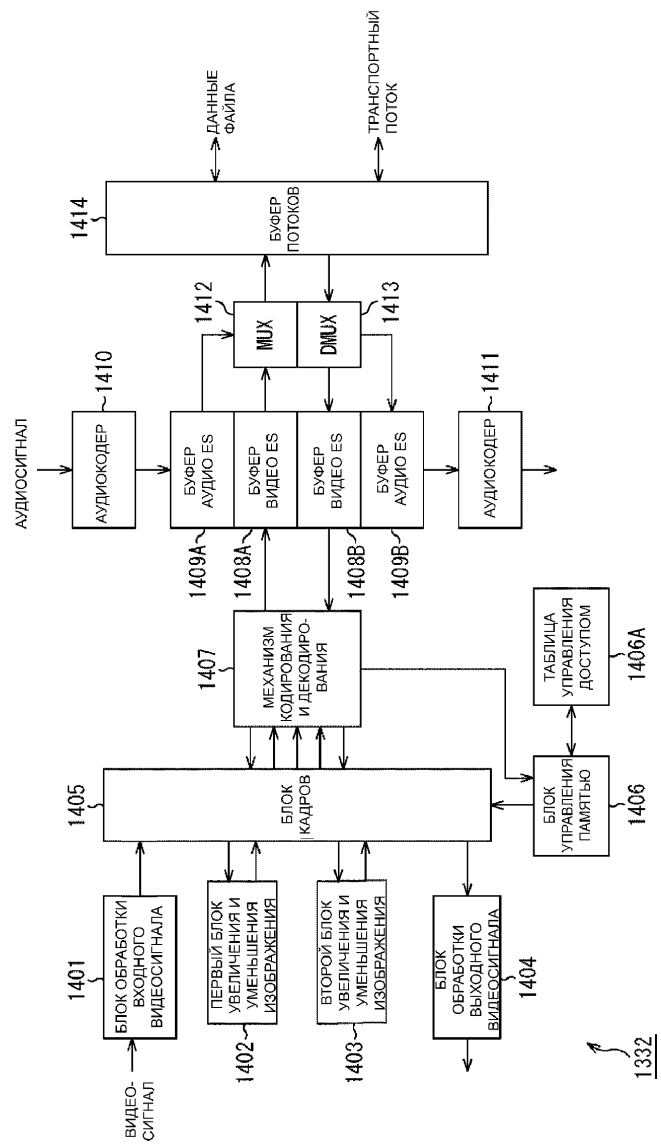
Фиг. 73



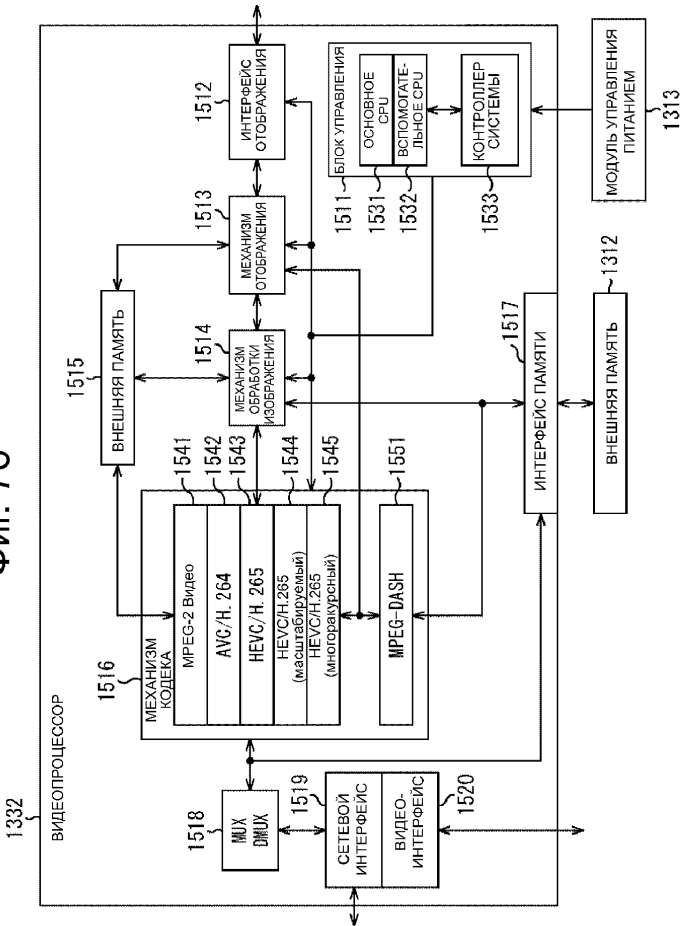
Фиг. 74



Фиг. 75



Фиг. 76



Фиг. 77

