



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/102 (2006.01); H04N 19/44 (2006.01); H04N 19/172 (2006.01); H04N 19/70 (2006.01); H04N 19/46 (2006.01); H04N 19/103 (2006.01); H04N 19/107 (2006.01); G06T 9/00 (2006.01)

(21)(22) Заявка: 2015102596, 19.06.2013

(24) Дата начала отсчета срока действия патента:
19.06.2013

Дата регистрации:
02.03.2018

Приоритет(ы):

(30) Конвенционный приоритет:
28.06.2012 US 61/665,862;
29.06.2012 US 61/666,688;
04.03.2013 US 13/784,006

(43) Дата публикации заявки: 20.08.2016 Бюл. № 23

(45) Опубликовано: 02.03.2018 Бюл. № 7

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 28.01.2015

(86) Заявка РСТ:
US 2013/046529 (19.06.2013)

(87) Публикация заявки РСТ:
WO 2014/004201 (03.01.2014)

Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

РАМАСУБРАМОНИАН Адарш Кришнан
(US),
ВАН Е-Куй (US),
ДЖОШИ Раджан Лаксман (US),
ЧЭНЬ Ин (US)

(73) Патентообладатель(и):

КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(56) Список документов, цитированных в отчете
о поиске: GARY J. SULLIVAN, "CRA
pictures with broken links", Joint Collaborative
Team on Video Coding (JCT-VC) of ITU-T
SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 9th
Meeting: Geneva, CH, 27 April - 7 May 2012,
Document: JCTVC-I0404 (rev. 1), 3 л., опубл.
27.04.2012. HENDRY, "Undiscardable leading
pictures for CRA", Joint Collaborative Team
on Video Coding (см. прод.)

(54) ПРОИЗВОЛЬНЫЙ ДОСТУП И СИГНАЛИЗАЦИЯ ДОЛГОСРОЧНЫХ ОПОРНЫХ КАРТИНОК
ПРИ КОДИРОВАНИИ ВИДЕО

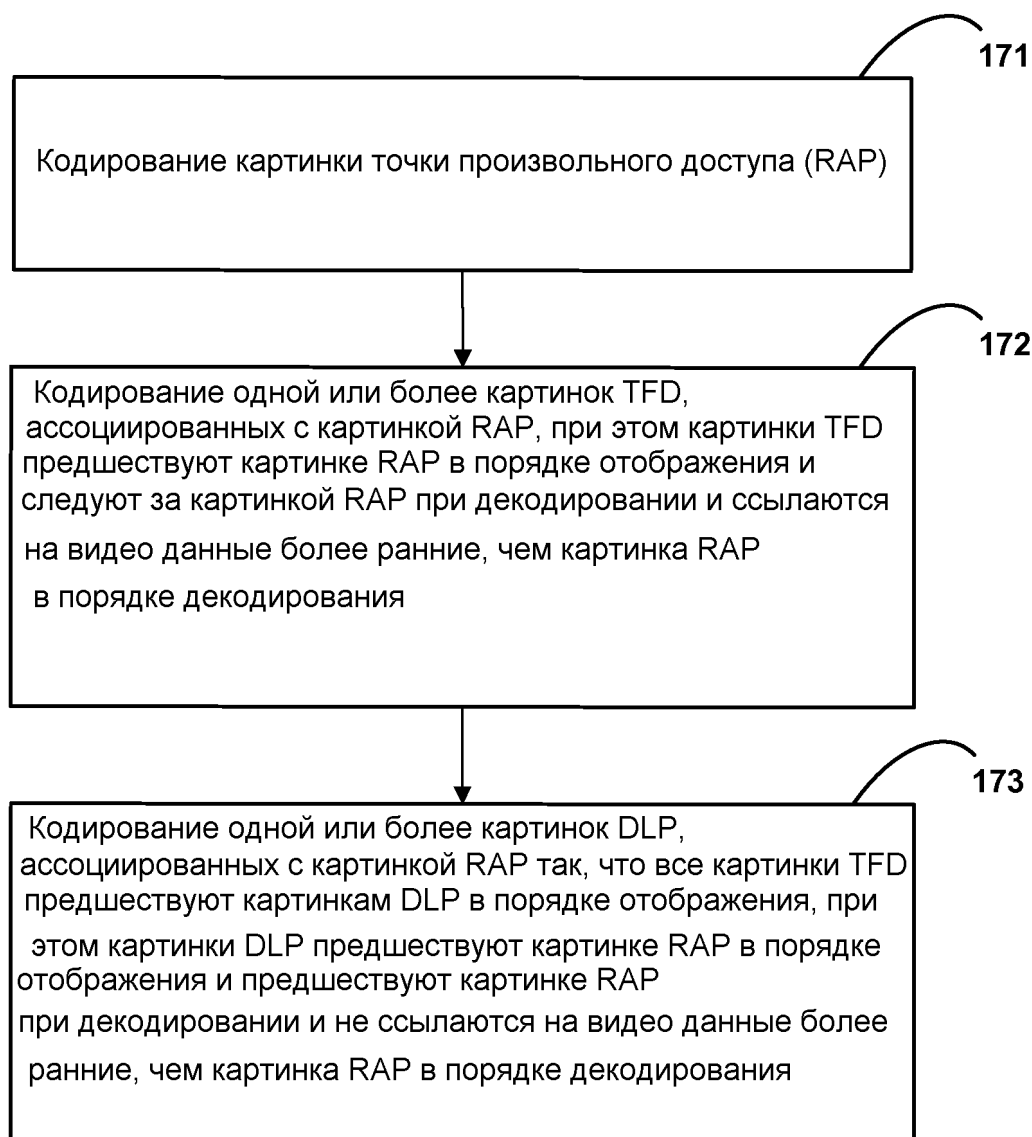
(57) Реферат:

Изобретение относится к области
кодирования/декодирования видеоданных.
Технический результат – предотвращение
неравномерности в скорости передачи кадров.
Способ декодирования видеоданных содержит:
декодирование, устройством декодирования
видео, содержащим один или несколько
процессоров, картинки точки произвольного

доступа (RAP); идентификацию одной или
нескольких картинок первого типа,
ассоциированных с картинкой RAP, причем одна
или несколько картинок первого типа
предшествуют картинке RAP в значении порядка
вывода, следуют за картинкой RAP в порядке
декодирования и ссылаются на видеоданные,
более ранние, чем картинка RAP в порядке

декодирования; отбрасывание одной или нескольких картинок первого типа; декодирование одной или нескольких декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все

из одной или нескольких картинок первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода; и вывод декодированной картинки RAP. 8 н. и 49 з.п. ф-лы, 11 ил., 8 табл.



ФИГ.7

(56) (продолжение):

(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 7th Meeting: Geneva, CH, 21 - 30 November, 2011, Document: JCTVC-G158, 7 л., опубли. 21.11.2011. US 2006/0230162 A1, 12.10.2006. US 2009/0238268 A1, 24.09.2009. WO 2010/086501 A1, 05.08.2010. US 2009/0220012 A1, 03.09.2009. RU 2419245 C2, 20.05.2011.



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

H04N 19/102 (2014.01)*H04N 19/44* (2014.01)*H04N 19/172* (2014.01)(12) **ABSTRACT OF INVENTION**

(52) CPC

H04N 19/102 (2006.01); *H04N 19/44* (2006.01); *H04N 19/172* (2006.01); *H04N 19/70* (2006.01); *H04N 19/46* (2006.01); *H04N 19/103* (2006.01); *H04N 19/107* (2006.01); *G06T 9/00* (2006.01)

(21)(22) Application: 2015102596, 19.06.2013

(24) Effective date for property rights:
19.06.2013

Registration date:
02.03.2018

Priority:

(30) Convention priority:
28.06.2012 US 61/665,862;
29.06.2012 US 61/666,688;
04.03.2013 US 13/784,006

(43) Application published: 20.08.2016 Bull. № 23

(45) Date of publication: 02.03.2018 Bull. № 7

(85) Commencement of national phase: 28.01.2015

(86) PCT application:
US 2013/046529 (19.06.2013)

(87) PCT publication:
WO 2014/004201 (03.01.2014)

Mail address:
129090, Moskva, ul. B. Spasskaya, 25, stroenie 3,
OOO "Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):

RAMASUBRAMONIAN Adarsh Krishnan
(US),
VAN E-Kuj (US),
DZHOSHI Radzhan Laksman (US),
CHEN In (US)

(73) Proprietor(s):

KVELKOMM INKORPOREJTED (US)

(54) **RANDOM ACCESS AND SIGNALING OF LONG-TERM REFERENCE PICTURES IN VIDEO CODING**

(57) Abstract:

FIELD: data processing.

SUBSTANCE: invention relates to the field of encoding / decoding video data. Video data decoding method comprises: decoding, with a video decoding device comprising one or more processors, random access point (RAP) pictures; identifying one or more pictures of a first type associated with the RAP picture, wherein one or more pictures of the first type precede the RAP picture in the output order value, follow the RAP picture in the decoding order and refer to video data earlier than the RAP picture in decoding order;

discarding one or more pictures of the first type; decoding one or more decoded initial pictures for RAP picture in accordance with a constraint, which forces the given condition to be true, the predetermined condition requiring that all of one or more pictures of the first type precede one or more decoded initial pictures associated with the RAP picture in the output order; and outputting the decoded RAP picture.

EFFECT: preventing unevenness in frame rate.

57 cl, 11 dwg, 8 tbl



ФИГ.7

[0001] Настоящая заявка испрашивает приоритет предварительной заявки на патент № 61/665,862, поданной 28 июня 2012, и предварительной заявки на патент № 61/666,688, поданной 29 июня 2012, каждая из которых тем самым включена по ссылке в их полноте.

ОБЛАСТЬ ТЕХНИКИ

[0002] Настоящее раскрытие в целом относится к кодированию видео.

УРОВЕНЬ ТЕХНИКИ

[0003] Цифровые возможности видео могут быть включены в широкий диапазон устройств, включающий в себя цифровые телевизоры, цифровые системы прямого вещания, беспроводные системы вещания, персональные цифровые помощники (PDA), ноутбуки или настольные компьютеры, планшетные компьютеры, электронные книги, цифровые камеры, цифровые устройства регистрации, цифровые медиаплееры, видеоигровые устройства, пультах видеоигр, сотовые или спутниковые радиотелефоны, так называемые «смартфоны», видеоустройства организации телеконференций, устройства потоковой передачи видео, и т.п. Эти цифровые видеоустройства реализуют способы кодирования видео, такие как описаны в стандартах, определенных посредством MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Часть 10, Усовершенствованное кодирование видео (AVC), стандарт высокоэффективного кодирования видео (HEVC), развивающийся в настоящее время, и расширениях таких стандартов. Недавний проект развивающегося стандарта HEVC, называемого «HEVC Working Draft 7», или «WD7», описан в документе HCTVC-I1003, Bross et al., «High Efficiency Video Coding (HEVC) Text Specification Draft 7», Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 9th Meeting: Geneva, Switzerland, 27 апреля 2012-7 мая 2012, который, на 5 февраля 2013, доступен по адресу http://phenix.it-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v5.zip. Видеоустройства могут передавать, принимать, кодировать, декодировать, и/или хранить цифровую информацию видео более эффективно, реализовывая такие способы кодирования видео.

[0004] Способы кодирования видео включают в себя пространственное (внутри картинки) предсказание и/или временное (между картинками) предсказание, чтобы уменьшить или удалить избыточность, присущую видеопоследовательностям. Для основанного на блоке кодирования видео видеовырезка (то есть видеокадр или часть видеокадра) может быть разделена на блоки видео, которые могут также упоминаться как блоки дерева кодирования (CTU), единицы кодирования (CU) и/или узлы кодирования. Блоки видео во внутренне кодированной (I) вырезке картинки закодированы, используя пространственное предсказание относительно опорных выборок в соседних блоках в одной и той же картинке. Блоки видео во внешне кодированной (P или B) вырезке картинки могут использовать пространственное предсказание относительно опорных выборок в соседних блоках в одной и той же картинке или временное предсказание относительно опорных выборок в других опорных картинках. Картинки могут упоминаться как кадры, и опорные картинки могут упоминаться как опорные кадры.

[0005] Пространственное или временное предсказание приводит к предсказываемому блоку для блока, который должен быть закодирован. Остаточные данные представляют пиксельные разности между первоначальным блоком, который должен быть закодирован, и предсказываемым блоком. Внешне кодированный блок кодируют согласно вектору движения, который указывает на блок опорных выборок, формирующих предсказывающий блок, и остаточным данным, указывающим разность

между закодированным блоком и предсказывающим блоком. Внутренне кодированный блок кодируют согласно режиму внутреннего кодирования и остаточным данным. Для дальнейшего сжатия остаточные данные могут быть преобразованы из пиксельной области в область преобразования, приводя к остаточным коэффициентам преобразования, которые затем могут быть квантованы. Квантованные коэффициенты преобразования, первоначально скомпонованные в двумерном массиве, могут быть сканированы, чтобы сформировать одномерный вектор коэффициентов преобразования, и энтропийное кодирование может быть применено, чтобы достигнуть еще большего сжатия.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0006] В целом, настоящее раскрытие описывает способы для поддержания произвольного доступа и сигнализации долгосрочных опорных картинок при кодировании видео. Настоящее раскрытие предлагает несколько ограничений для поддержания произвольного доступа на основании различных типов картинок. В одном примере все тэгированные для отбрасывания (TFD) картинки для картинки точки произвольного доступа (RAP) могут иметь значение порядка отображения, которое является более ранним, чем значение порядка отображения для всех декодируемых начальных картинок (DLPs) для этой RAP. Настоящее раскрытие также предлагает ограничение, в котором предотвращено перемежение начальных картинок с “находящимися сзади картинками” в порядке декодирования. То есть, в соответствии со способами настоящего раскрытия, видеокодировщики могут гарантировать, что все начальные картинки (включая и картинки TFD и DLPs) имеют значения порядка декодирования ранее, чем значения порядка декодирования картинок, которые имеют и значения порядка отображения, большие, чем значение порядка отображения соответствующей картинки RAP, и значения порядка декодирования, большие, чем значения порядка декодирования соответствующей картинки RAP. Порядок отображения может также упоминаться как порядок вывода. Картинки TFD могут также упоминаться как пропущенные начальные картинки произвольного доступа (RASL), и картинки DLP могут также упоминаться как декодируемые начальные картинки произвольного доступа (RADL).

[0007] В одном примере способ кодирования видеоданных включает в себя кодирование картинки точки произвольного доступа (RAP); и кодирование одной или более декодируемых начальных картинок (DLPs) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0008] В другом примере устройство для кодирования видеоданных включает в себя видеокодировщика, конфигурируемого, чтобы закодировать картинку точки произвольного доступа (RAP); и кодировать одну или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0009] В другом примере устройство для кодирования видеоданных включает в себя средство для кодирования картинки точки произвольного доступа (RAP); и средство для кодирования одной или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0010] В другом примере считываемый компьютером запоминающий носитель,

хранящий инструкции, которые, когда выполняются одним или более процессорами, заставляют один или более процессоров кодировать картинку точки произвольного доступа (RAP); и кодировать одну или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким образом, что все картинки, которые

5 предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0011] В другом примере способ кодирования видеоданных включает в себя кодирование картинки точки произвольного доступа (RAP) и кодирование одной или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким

10 образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0012] В другом примере устройство для кодирования видеоданных включает в себя видеокодер, конфигурируемый, чтобы закодировать картинку точки произвольного доступа (RAP); и кодировать одну или более декодируемых начальных картинок

15 (картинок DLP) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0013] В другом примере устройство для кодирования видеоданных включает в себя средство для кодирования картинки точки произвольного доступа (RAP) и средство

20 для кодирования одной или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения.

[0014] В другом примере считываемый компьютером запоминающий носитель хранит

25 инструкции, которые, когда выполняются одним или более процессорами заставляют один или более процессоров кодировать картинку точки произвольного доступа (RAP); и кодировать одну или более декодируемых начальных картинок (картинок DLP) для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в

30 порядке отображения.

[0015] Подробности одного или более примеров сформулированы в сопроводительных чертежах и описании ниже. Другие признаки, объекты, и преимущества будут очевидны из описания и чертежей и из формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0016] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему кодирования и декодирования видео, которая может использовать способы для

35 сигнализации долгосрочных опорных картинок в заголовках вырезок.

[0017] Фиг. 2 является блок-схемой, иллюстрирующей пример видеокодера, который может реализовать способы для сигнализации долгосрочных опорных картинок в

40 заголовках вырезок.

[0018] Фиг. 3 является блок-схемой, иллюстрирующей пример видеodeкодера, который может реализовать способы для сигнализации долгосрочных опорных картинок в заголовках вырезок.

[0019] Фиг. 4 является концептуальной диаграммой, иллюстрирующей

45 последовательность закодированных видео картинок.

[0020] Фиг. 5 является блок-схемой, иллюстрирующей примерный набор устройств, которые формируют часть сети.

[0021] Фиг. 6 является последовательностью операций, иллюстрирующей примерную

операцию для кодирования картинки точки произвольного доступа (RAP) согласно способам настоящего раскрытия.

[0022] Фиг. 7 является последовательностью операций, иллюстрирующей примерную операцию для кодирования картинок RAP согласно способам настоящего раскрытия.

[0023] Фиг. 8 является последовательностью операций, иллюстрирующей примерную операцию для кодирования картинок RAP согласно способам настоящего раскрытия.

[0024] Фиг. 9 является последовательностью операций, иллюстрирующей примерную операцию для кодирования картинок RAP согласно способам настоящего раскрытия.

[0025] Фиг. 10 является последовательностью операций, иллюстрирующей примерную операцию для кодирования картинок RAP согласно способам настоящего раскрытия.

[0026] Фиг. 11 является последовательностью операций, иллюстрирующей примерную операцию для кодирования картинок RAP согласно способам настоящего раскрытия.

ПОДРОБНОЕ ОПИСАНИЕ

[0027] Обычно видеоданные представляют последовательностью картинок, которые захватываются или отображаются в быстрой последовательности. Должно быть понятно, что в некоторых примерах картинки или части картинок могут генерироваться, например, используя компьютерную графику, вместо (или в дополнение к) захвата.

Порядок, в котором должны быть отображены картинки (который может быть или может не быть тем же самым как порядком, в котором были захвачены или сгенерированы картинки), может отличаться от порядка, в котором закодированные картинки включены в поток битов. Порядок отображения картинок в пределах закодированной видеопоследовательности может быть представлен значением счета по порядку картинки (POC), в то время как порядок кодирования картинок может быть представлен значениями количества кадров (frame_num).

[0028] Кодирование картинок обычно вовлекает преимущественное использование избыточных данных, имеющих место в картинках. Например, пространственное кодирование, также называемое внутреннее предсказывающее кодирование, использует преимущество избыточных данных, имеющих место между пространственно соседними блоками пиксельных значений общей картинки. В качестве другого примера, временное кодирование, также называемое внешнее предсказывающее кодирование, использует преимущество избыточных данных, имеющих место между различными картинками, которые захвачены, сгенерированы или отображены в различные моменты времени. Более подробно, картинка может быть разделена на вырезки, которые могут быть назначены для внутреннего или внешнего предсказания. Кроме того, каждая вырезка может быть разделена на блоки (например, наибольшие единицы кодирования (единицы LCU), также называемые как единицы дерева кодирования), и каждый из блоков может быть или далее разделен или закодирован на основании того, назначена ли соответствующая вырезка для внутреннего или внешнего предсказания.

[0029] Относительно примера внешнего предсказывающего кодирования блоки картинки, кодируемой в настоящее время, (также называемой “текущей картинкой”), могут быть предсказаны из опорной картинки. WD7 определяет “опорную картинку” как картинку с nal_ref_flag равным 1. nal_ref_flag является элементом синтаксиса единицы уровня абстракции сети (NAL), указывающей, нужно ли данные, включенные в единицу NAL, рассматривать как опорную картинку. WD7 также обеспечивает, что опорная картинка содержит выборки (то есть пиксельные значения), которые могут быть использованы для внешнего предсказания в процессе декодирования последующих картинок в порядке декодирования.

[0030] Как введено выше, обычно видеоданные включают в себя различные типы

картинок. Например, видеоданные включают в себя случайные точки доступа (точки RAP), которые могут быть использованы для начала потока видеоданных. Точки RAP являются картинками, которые независимо кодируются, например, как закодированные картинки с внутренним предсказанием (I-картинки). Точки RAP включают в себя

5 картинки мгновенного обновления декодера (IDR), картинки доступа с разорванной ссылкой (BLA), картинки чистого произвольного доступа (CRA), и картинки постепенного обновления при декодировании (GDR). Другие типы картинок включают в себя картинки, которые предназначены для отбрасывания (TFD), декодируемые начальные картинки (картинки DLP), и находящиеся сзади картинки. Вообще, картинки

10 DLP (как все начальные картинки) являются картинками, которые имеют порядок вывода, который является более ранним, чем порядок вывода соответствующей RAP, но порядок декодирования, который является более поздним, чем порядок декодирования соответствующей RAP. Однако, картинки DLP являются начальными картинками, которые являются декодируемыми без извлечения данных, которые

15 предшествуют соответствующей RAP в порядке декодирования. Находящиеся сзади картинки, с другой стороны, имеют порядок вывода, который является более поздним, чем порядок вывода соответствующей RAP, так же как порядок декодирования, который является более поздним, чем порядок декодирования соответствующей RAP. Картинки, которые являются TFD, не могут быть должным образом декодированы при выполнении

20 произвольного доступа от соответствующей RAP. Это происходит обычно из-за картинки TFD, предсказываемой исходя из данных, которые предшествуют соответствующей RAP в порядке декодирования. Поскольку произвольный доступ от RAP опускает извлечение данных ранее, чем RAP в порядке декодирования, картинки TFD, которые зависят от данных ранее, чем RAP в порядке декодирования, не являются

25 правильно декодируемыми, когда RAP используется как точка произвольного доступа, и могут поэтому быть отброшены без декодирования. Картинки TFD "зависят" от более ранних данных в том смысле, что они предсказываются относительно одной или более картинок, имеющих место в порядке декодирования, чем RAP.

[0031] Единицам NAL можно назначить конкретные значения типа единицы NAL, например, в заголовке единицы NAL, чтобы указать тип данных, включенных в

30 соответствующую единицу NAL. Единицы NAL могут обычно включать в себя данные уровня кодирования видео (VCL), также называемые единицы NAL VCL, или данные не-VCL, также называемые единицы NAL не-VCL. Данные не-VCL включают в себя, например, наборы параметров, такие как наборы параметров картинки (PPS), наборы

35 параметров последовательности (SPS), и наборы параметров адаптации (APS), так же как сообщения информации дополнительного расширения (SEI). Данные VCL обычно включают в себя закодированные видеоданные. Тип единицы NAL может обеспечить индикацию типа данных, включенных в единицу NAL, включая, соответствуют ли RAP закодированные видеоданные в единице NAL, и если это так, соответствуют ли эти

40 закодированные видеоданные картинке CRA, BLA или IDR. Тип единицы NAL может указывать, включает ли единица NAL в себя другие типы данных также. Таблица 1 ниже обеспечивает пример некоторых значений типа единицы NAL, и данных, включенных в соответствующую единицу NAL, как было предоставлено в WD7:

ТАБЛИЦА 1		
nal_unit_type	Контент единицы NAL и структура синтаксиса RBSP	Класс типа единицы NAL
1	Кодированная вырезка не-RAP, не-TFD and не-TLA картинки slice_layer_rbsp()	VCL
2	Кодированная вырезка картинки TFD slice_layer_rbsp()	VCL

3	Кодированная вырезка картинки TLA не-TFD slice_layer_rbsp()	VCL
4, 5	Кодированная вырезка картинки CRA slice_layer_rbsp()	VCL
6,7	Кодированная вырезка картинки BLA slice_layer_rbsp()	VCL
8	Кодированная вырезка картинки IDR slice_layer_rbsp()	VCL

[0032] Кроме того, WD7 различает долгосрочные опорные картинки от краткосрочных опорных картинок. Например, WD7 определяет долгосрочную опорную картинку как опорную картинку, которая маркирована как “используется для долгосрочной ссылки». WD7 обеспечивает, что флаг в наборе параметров последовательности (SPS) указывает, сигнализированы ли долгосрочные опорные картинки обычно для закодированной видеопоследовательности. В соответствии с WD7, видеокодировщики сигнализируют младшие значащие биты (LSB) значений POC долгосрочных опорных картинок в заголовках вырезок в вырезках. Сигнализация LSB может привести к экономии битов относительно сигнализации полного значения POC. Видеокодировщики, в соответствии с WD7, могут сигнализировать старшие значащие биты (MSB) значений POC долгосрочной опорной картинки, если имеется больше, чем одна опорная картинка в буфере декодированных картинок (DPB), которая имеет те же самые биты LSB как долгосрочная опорная картинка. Кроме того, в соответствии с WD7, видеокодировщики могут использовать флаг, чтобы указать, могут ли сигнализированные долгосрочные опорные картинки использоваться текущей картинкой для ссылки.

[0033] Одно предположение для конструкции набора опорных картинок (RPS) в HEVC является то, что выведение RPS должно быть робастным к потерям картинок, пока относительные значения POC могут быть корректно установлены как для краткосрочных опорных картинок (STRP) так и для долгосрочных опорных картинок (LTRP). Чтобы достигнуть этого, выведение RPS для любой картинки может быть автономным, то есть не в зависимости от статуса буфера декодированных картинок (DPB). С этим принципом конструкции, в соответствии с WD7, STRP сигнализируются, используя значения их дельты POC относительно текущей картинки. Следовательно, декодер может быть в состоянии вывести POC этих опорных картинок, даже когда некоторые картинки были потеряны, и следовательно проверить, присутствует ли каждая из STRP.

[0034] Настоящее раскрытие распознает некоторые проблемы в текущей сигнализации заголовка вырезки для LTRP в присутствии потерь картинок в WD7. Ошибочное выведение RPS возможно в присутствии потерь. Это происходит из-за сигнализации только величины LSB для POC для LTRP, которая является той же самой, как для сигнализации значения POC для каждой текущей картинки (то есть `pic_order_cnt_lsb`), хотя в некоторых случаях полные значения POC, используя элемент синтаксиса `delta_poc_msb_cycle_lsb` сигнализируются для LTRPs. Если есть картинка в DPB, которая имеет тот же самый LSB, как тот что сигнализирован для LTRP, то может быть двусмысленность в выведении RPS, и неправильный результат RPS может иметь место, как показано в примерах, представленных ниже. В примерах, описанных ниже, "MaxPicOrderCntLsb", как предполагается, равен 256, или другими словами, `log2_max_pic_order_cnt_lsb_minus4` равняется 4.

[0035] В качестве первого примера, который иллюстрирует проблемы, описанные выше, рассмотрим три картинки n-1, n, и n+1, которые являются последовательными в порядке декодирования. Предположим, что картинка n-1 имеет опорные картинки

со значениями POC 0 и 256 в качестве LTRP, и картинки n и n+1 имеют только опорную картинку с POC 256 в качестве LTRP. Значения POC картинок n-1, n, и n+1 находятся все в диапазоне 257-511, включительно. Также предположим, что ни одна из картинок n-1, n, и n+1 или любая другая картинка в DPB не имеет значения LSB, равного 0. При сигнализации HEVC согласно WD7, релевантные элементы синтаксиса соответствующего потока битов являются такими, как показано в Таблице 2.

ТАБЛИЦА 2				
POC (в порядке декодирования)	LTRP	LSB LTRP	delta_poc_msb_present_flag	delta_poc_msb_cycle_lt
...
n-1	0,256	0,0	1,1	1,0
n	256	0	1	0
n+1	256	0	0	-
...

[0036] Для картинки n, даже при том, что только картинка 256 сигнализирована как LTRP, когда RPS картинки n выводится, картинка 0 все еще была в DPB. Следовательно, даже для картинки n, delta_poc_msb_present_flag устанавливается в 1, и значение delta_poc_msb_cycle_lt посылается. После вывода RPS для картинки n, картинка 0 маркируется как “неиспользуемая для ссылки» и может быть удалена из DPB. Для картинки n+1, однако, так как только одна LTRP присутствует в DPB, delta_poc_msb_present_flag устанавливается в 0, и delta_poc_msb_cycle_lt не присутствует.

[0037] При структуре синтаксиса WD7 вышеупомянутая таблица будет соответствовать соответствующему потоку битов. Однако, если картинка n потеряна, то сигнализация LTRP для картинки n+1, будет недостаточной, чтобы решить, какая картинка из DPB должна быть выбрана как LTRP. И 0 и 256 имеют LSB равный 0, и декодер может не быть в состоянии решить проблему. Декодер нуждается в delta_poc_msb_present_flag, равным 1 для картинки n+1, чтобы уникально идентифицировать картинку, которая должна быть использоваться как единственная LTRP.

[0038] Второй пример проблем, описанных выше, относящихся к первому примеру выше, является следующим. Описание Таблицы 3 подобно описанию Таблицы 2, за исключением того, что картинка 0 сигназируется как LTRP, и картинка 256 сигназируется как STRP для картинки n-1.

ТАБЛИЦА 3					
POC (в порядке декодирования)	LTRPs	STRPs	LTRP LSBs	delta_poc_msb_present_flag	delta_poc_msb_cycle_lt
...
n-1	0	256	0	1	1
n	256	-	0	1	0
n+1	256	-	0	0	-
...

[0039] Здесь снова, если бы картинка n потеряна, декодер может не быть в состоянии правильно идентифицировать эту LTRP, сигнализированную как картинка 256, и декодер будет нуждаться в delta_poc_msb_present_flag, равным 1, чтобы уникально идентифицировать картинку, которая должна быть использоваться как единственная LTRP.

[0040] Третий пример проблем, описанных выше, описан со ссылками на Таблицу 4.

ТАБЛИЦА 4

POC (в порядке декодирования)	LTRPs	STRP	LTRP LSBs	delta_poc_msb_present_flag	delta_poc_msb_cycle_lt
...
255	0	-	0	0	
256	0	-	0	0	
257	256	-	0	1	0
258	256	-	0	0	-
...

[0041] Относительно примера Таблицы 4, предположим, что картинка с POC 255 принята, с последующей картинкой 258 (другими словами, картинки 256 и 257 были потеряны). После декодирования заголовка вырезки для картинки 258, картинка 0 (вместо картинки 256) будет маркирована как LTRP. Вместо того, чтобы указать, что опорная картинка 256 не присутствует для картинки 258, декодер может некорректно вывести, что картинка 0 является сигнализированной LTRP и продолжить декодирование (обеспечив другие аспекты работы процесса декодирования для этой потери).

[0042] Из вышеупомянутых примеров обманчиво простое решение для решения вышеупомянутых проблем может оказаться сигнализировать фактическое полное значение POC этих LTRP. Также может казаться, что выведение RPS будет автономным с полной сигнализацией POC. Однако, возможность, что картинки CRA могут или начать поток битов, или что картинки CRA могут быть преобразованы в картинку BLA с помощью средством стыковки, исключает сигнализацию полного значения POC. Например, рассмотрим картинку CRA в потоке битов со значением POC равным 256 (все еще принимая, что MaxPicOrderCntLsb = 256), и пусть некоторые картинки, которые следуют за картинкой CRA в порядке декодирования, имеют картинку CRA в качестве LTRP. Если полное POC картинки CRA, которое равно 256, сигнализируется, и картинка CRA преобразуется в картинку BLA или начинает поток битов, то LTRP может быть неправильно выведена, и поток битов станет несогласованным. Когда картинка CRA преобразуется в картинку BLA или начинает поток битов, декодер выводит свою POC только на основе его LSB, в соответствии со способами WD7. В вышеупомянутом примере POC картинки CRA (которая является теперь картинкой BLA) будет настроено равным 0. Когда декодер просматривает LTRP с POC 256, DPB не будет содержать какую-либо подобную картинку, потому что теперь картинка CRA имеет выведенный POC равный 0. Это будет приводить к «нет опорной картинки» в RPS, и если текущая картинка не является картинкой TFD, ассоциированной с картинкой CRA, результирующий поток битов будет несогласованным.

[0043] Настоящее раскрытие описывает способы для поддержания произвольного доступа для видеоданных. В некоторых случаях эти способы могут быть использованы для преодоления вышеупомянутых проблем в WD7. Более подробно, настоящее раскрытие предлагает несколько ограничений для поддержания произвольного доступа на основании различных типов картинок. Во-первых, согласно одному способу настоящего раскрытия, все картинки TFD для картинки RAP должны иметь значение порядка отображения, которое является более ранним, чем значение порядка отображения для всех картинок DLP для RAP. Это ограничение может улучшить опыт просмотра зрителем, избегая зыби в скорости передачи кадров. Таким образом, если есть одна или более картинок, которые не могут быть корректно декодированы (то есть картинки TFD) и если эти картинки имеют значения порядка отображения, смешанные с картинками DLP, то скорость передачи кадров может казаться изменчивой, что может ухудшить пользовательский опыт. Согласно другому способу настоящее

раскрытие также предлагает ограничение, в котором предотвращено перемежение начальных картинок с “находящимися сзади картинками” в порядке декодирования. Таким образом, в соответствии со способами настоящего раскрытия, видеокодировщики могут гарантировать, что все начальные картинки (включая как картинки TFD так и

5 картинки DLP) имеют значения порядка декодирования более ранние, чем значения порядка декодирования картинок, которые имеют значения порядка отображения, большие чем значение порядка отображения соответствующей картинки RAP. Как отмечено выше, “находящиеся сзади картинки” являются картинками, которые следуют за картинкой RAP и в порядке декодирования и в порядке отображения.

10 [0044] Как введено выше, обычно видеоданные включают в себя различные типы картинок. Например, видеоданные включают в себя точки RAP, которые могут быть использованы для начала потока видеоданных. Точки RAP являются картинками, которые являются независимо кодированными, например, в качестве кодированных картинок с внутренним предсказанием (I-картинки). Точки RAP включает в себя

15 картинки IDR, картинки BLA и картинки CRA.

[0045] Другие типы картинок включают в себя TFD, картинки DLP, и находящиеся сзади картинки. Обычно, картинки DLP (как и все начальные картинки) являются картинками, которые имеют порядок вывода, который является более ранним, чем порядок вывода соответствующей картинки RAP, но порядок декодирования, который

20 является более поздним, чем порядок декодирования соответствующей RAP. Однако, картинки DLP являются начальными картинками, которые являются декодируемыми без извлечения данных, которые предшествуют соответствующей картинке RAP в порядке декодирования. Другими словами, картинки DLP являются начальными картинками, которые предсказываются относительно данных одной или более картинок,

25 включая и/или следующие после соответствующей RAP. Находящиеся сзади картинки, с другой стороны, имеют порядок вывода, который является более поздним, чем порядок вывода соответствующей RAP, так же как порядок декодирования, который является более поздним, чем порядок декодирования соответствующей картинки RAP.

[0046] Картинки, которые являются картинками TFD, не могут быть должным

30 образом декодированы при выполнении произвольного доступа от соответствующей картинки RAP. Это происходит обычно из-за картинки TFD, предсказываемой из данных, которые предшествуют соответствующей картинке RAP в порядке декодирования. Поскольку произвольный доступ из картинки RAP пропускает извлечение данных ранее, чем картинка RAP, картинки TFD в зависимости от данных, более ранних, чем

35 картинка RAP, не будет правильно декодирована, и может поэтому быть отклонена без декодирования.

[0047] Настоящее раскрытие предлагает несколько ограничений для поддержания произвольного доступа на основании этих различных типов картинок. Во-первых, все

40 картинки TFD картинки RAP должны иметь значение порядка отображения, которое является более ранним, чем значение порядка отображения для всех картинок DLP этой картинки RAP. Это ограничение может улучшить опыт зрительского просмотра, потому что оно избегает зыби в скорости передачи кадров. Таким образом, если имеется одна или более картинок, которые не могут быть корректно декодированы (такие как

45 картинки TFD), если эти картинки имеют значения порядка отображения, смешанные с картинками DLP, то скорость передачи кадров может казаться изменчивой, что может ухудшить пользовательский опыт.

[0048] Настоящее раскрытие также предлагает ограничение, в котором предотвращено перемежение начальных картинок с “находящимися сзади картинками”

в порядке декодирования. Таким образом, в соответствии со способами настоящего раскрытия видеокодировщики могут гарантировать, что все начальные картинки (включая как картинки TFD так и картинки DLP) имеют значения порядка декодирования ранние, чем значения порядка декодирования картинок, которые имеют значения порядка отображения, большие, чем значение порядка отображения соответствующей картинки RAP. Как отмечено выше, «находящиеся сзади картинки» являются картинками, которые следуют за картинкой RAP и в порядке декодирования и в порядке отображения. Реализуя способы настоящего раскрытия, видеокодер может быть сконфигурирован, чтобы закодировать картинку RAP и закодировать одну или более картинок DLP для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке вывода. Аналогично, видеodeкодер не должен пытаться декодировать картинки TFD, следующие после картинки RAP, которая используется как точка произвольного доступа. Вместо этого видеodeкодер может просто синтаксически разобрать данные для таких картинок TFD, не пытаясь декодировать данные. Это может обеспечить экономию ресурсов для устройства, такого как видеodeкодер. Например, эти способы могут сэкономить мощность батареи и/или улучшить эффективность обработки. Не перемежая начальные и находящиеся сзади картинки в порядке декодирования, система (такая как MANE или видеodeкодер) может легко идентифицировать начальные картинки, ассоциированные с картинкой RAP, не имея потребности выполнять синтаксический разбор вне первой находящейся сзади картинки, которая следует за ассоциированной картинкой RAP. Идентификация начальных картинок может быть полезной в принятии решения о выводе при случайном доступе, потому что начальная картинка имеет место перед RAP в порядке отображения и может быть полезной для удаления начальных картинок, если промежуточная система конфигурируется, чтобы делать это. Кроме того, как отмечено выше, эти способы могут привести к улучшенному пользовательскому опыту, избегая зыби в скорости передачи кадров.

[0049] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему 10 кодирования и декодирования видео, которая может использовать способы для сигнализации долгосрочных опорных картинок в заголовках вырезок. Как показано на фиг. 1, система 10 включает в себя исходное устройство 12, которая обеспечивает закодированные видеоданные, которые должны быть декодированы в более позднее время устройством 14 назначения. В частности, исходное устройство 12 выдает видеоданные устройству 14 назначения через считываемый компьютером носитель 16. Исходное устройство 12 и устройство 14 назначения могут содержать любое из широкого диапазона устройств, включая настольные компьютеры, портативные компьютеры (то есть ноутбуки), планшетные компьютеры, телевизионные приставки, телефонные трубки, такие как так называемые "интеллектуальные" телефоны, так называемые "интеллектуальные" клавиатуры, телевизоры, камеры, устройства отображения, цифровые медиаплееры, пульты видеоигр, устройство потоковой передачи видео или подобное. В некоторых случаях исходное устройство 12 и устройство 14 назначения могут быть оборудованы для беспроводной связи.

[0050] Устройство 14 назначения может принять закодированные видеоданные, которые должны быть декодированы, через считываемый компьютером носитель 16. Считываемый компьютером носитель 16 может содержать любой тип носителя или устройства, способного к перемещению закодированных видеоданных от исходного устройства 12 к устройству 14 назначения. В одном примере считываемый компьютером

носитель 16 может содержать коммуникационный носитель, чтобы позволить исходному устройству 12 передавать закодированные видеоданные непосредственно к устройству 14 назначения в реальном времени. Закодированные видеоданные могут модулироваться согласно стандарту связи, такому как протокол беспроводной связи, и передаваться к устройству 14 назначения. Коммуникационный носитель может содержать любой беспроводной или проводной коммуникационный носитель, такой как радиочастотный (RF, РЧ) спектр или одна или более физических линий передачи. Коммуникационный носитель может явиться частью основанной на пакетной передаче сети, такой как локальная сеть, региональная сеть, или глобальная сеть, такая как Интернет.

Коммуникационный носитель может включать в себя маршрутизаторы, коммутаторы, базовые станции, или любое другое оборудование, которое может быть полезным, чтобы облегчить связь от исходного устройства 12 к устройству 14 назначения.

[0051] В некоторых примерах закодированные данные могут быть выведены из интерфейса 22 вывода на устройство хранения. Аналогично, к закодированным данным может получить доступ от устройства хранения интерфейс ввода. Устройство хранения может включать в себя любой из множества распределенных или локально доступных запоминающих носителей данных, таких как накопитель на жестких дисках, диски Blu-ray, DVD, CD-ROM, флэш-память, энергозависимая или энергонезависимая память, или любые другие подходящие цифровые запоминающие носители для того, чтобы хранить закодированные видеоданные. В еще одном примере устройство хранения может соответствовать файловому серверу или другому промежуточному устройству хранения, которое может хранить закодированное видео, генерируемое исходным устройством 12. Устройство 14 назначения может получить доступ к сохраненным видеоданным от устройства хранения с помощью потоковой передачи или загрузки. Файловый сервер может быть любым типом сервера, способного к тому, чтобы хранить закодированные видеоданные и передавать закодированные видеоданные к устройству 14 назначения. Примерные файловые серверы включают в себя web-сервер (например, для вебсайта), сервер FTP, устройства сетевого хранения (NAS), или локальный дисковый накопитель. Устройство 14 назначения может получить доступ к закодированным видеоданным через любое стандартное соединение данных, включая интернет-соединение. Оно может включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, DSL, кабельный модем, и т.д.), или комбинацию обоих, которая является подходящей для того, чтобы получить доступ к закодированным видеоданным, хранящимся на файловом сервере. Передача закодированных видеоданных от устройства хранения может быть потоковой передачей, передачей загрузки, или их комбинацией.

[0052] Способы настоящего раскрытия не обязательно ограничены беспроводными приложениями или параметрами настройки. Эти способы могут быть применены к кодированию видео в поддержку любого из множества мультимедийных приложений, таких как эфирное телевидение, передачи кабельного телевидения, передачи спутникового телевидения, Интернет, потоковой передачи видео, такой как динамическая адаптивная потоковая передача по HTTP (DASH), цифровое видео, которое закодировано на запоминающем носителе данных, декодирование цифрового видео, сохраненного на запоминающем носителе данных, или других приложений. В некоторых примерах система 10 может быть сконфигурирована, чтобы поддерживать одностороннюю или двухстороннюю передачу видео, чтобы поддерживать приложения, такую как потоковая передача видео, воспроизведение видео, вещание видео, и/или видеотелефония.

[0053] В примере на Фиг. 1 исходное устройство 12 включает в себя видеоисточник

18, видеокодер 20 и интерфейс 22 вывода. Устройство 14 назначения включает в себя интерфейс 28 ввода, видеодекодер 30, и устройство 32 отображения. В соответствии с настоящим раскрытием, видеокодер 20 из исходного устройства 12 может быть сконфигурирован, чтобы применять способы для сигнализации долгосрочных опорных картинок в заголовках вырезов. В других примерах исходное устройство и устройство назначения могут включать в себя другие компоненты или компоновки. Например, исходное устройство 12 может принять видеоданные из внешнего источника 18 видео, такого как внешняя камера. Аналогично, устройство 14 назначения может взаимодействовать с внешним устройством отображения вместо включения интегрированного устройства отображения.

[0054] Иллюстрированная система 10 на Фиг. 1 является просто одним примером. Способы сигнализации долгосрочных опорных картинок в заголовках вырезов могут быть выполнены любым цифровым устройством кодирования и/или декодирования видео. Хотя обычно способы настоящего раскрытия выполняются устройством кодирования видео, эти способы могут также быть выполнены кодером/декодером видео, типично называемым как "кодек". Кроме того, способы настоящего раскрытия могут также быть выполнены препроцессором видео. Исходное устройство 12 и устройство 14 назначения является просто примерами таких устройств кодирования, в которых исходное устройство 12 генерирует закодированные видеоданные для передачи к устройству 14 назначения. В некоторых примерах устройства 12, 14 могут работать по существу симметричным способом таким образом, что каждое из устройств 12, 14 включают в себя компоненты кодирования и декодирования видео. Следовательно, система 10 может поддерживать одностороннюю или двухстороннюю передачу видео между видеоустройствами 12, 14, например, для потоковой передачи видео, воспроизведения видео, вещания видео или видеотелефонии.

[0055] Видеоисточник 18 из исходного устройства 12 может включать в себя устройство захвата видео, такое как видеокамера, видеоархив, содержащий ранее захваченное видео, и/или интерфейс подачи видео, чтобы принять видео от поставщика видеоконтента. В качестве дальнейшей альтернативы, видеоисточник 18 может генерировать основанные на компьютерной графике данные в качестве исходного видео, или комбинацию «живого» видео, заархивированного видео, и машинно-генерируемого видео. В некоторых случаях, если видеоисточник 18 является видеокамерой, исходное устройство 12 и устройство 14 назначения может сформировать так называемые камерофоны или видеотелефоны. Как упомянуто выше, однако, способы, описанные в этом раскрытии, могут быть применимыми к кодированию видео в целом, и могут быть применены к беспроводным и/или проводным приложениям. В каждом случае захваченное, предварительно захваченное или машинно-генерируемое видео может быть закодировано видеокодером 20. Закодированная видеoinформация затем может быть выведена интерфейсом 22 вывода на считываемый компьютером носитель 16.

[0056] Считываемый компьютером носитель 16 может включать в себя временный носитель, такой как беспроводное вещание или передача по проводной сети, или запоминающие носители (то есть невременные запоминающие носители), такие как жесткий диск, накопитель на флэш-памяти, компакт-диск, цифровой видеодиск, диск Blu-ray, или другой считываемый компьютером носитель. В некоторых примерах сетевой сервер (не показан) может принять закодированные видеоданные от исходного устройства 12 и выдать закодированные видеоданные к устройству 14 назначения, например, через сетевую передачу. Аналогично, вычислительное устройство фабрики

производства носителей, такого как устройство штамповки дисков, может принять закодированные видеоданные от исходного устройства 12 и изготовить диск, содержащий закодированные видеоданные. Поэтому считываемый компьютером носитель 16 может быть понят как включающий в себя один или более считываемых компьютером носителей различных форм, в различных примерах.

[0057] Интерфейс 28 ввода из устройства 14 назначения принимает информацию от считываемого компьютером носителя 16. Информация считываемого компьютером носителя 16 может включать в себя информацию синтаксиса, определенную видеокодером 20, которая также используется видеодекодером 30, которая включает в себя элементы синтаксиса, которые описывают характеристики и/или обработку блоков и других закодированных единиц, например, GOP (групп картинок). Устройство 32 отображения отображает декодированные видеоданные пользователю и может содержать любое из множества устройств отображения, таких как электронно-лучевая трубка (CRT), жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих (OLED) диодах, или другой тип устройства отображения.

[0058] Видеокодер 20 и видеодекодер 30 могут работать согласно стандарту кодирования видео, такому как стандарт высокоэффективного кодирования видео (HEVC), развивающийся в настоящее время, и могут соответствовать Тестовой Модели HEVC (НМ). Альтернативно, видеокодер 20 и видеодекодер 30 могут работать согласно другим составляющим собственность стандартам или стандартам промышленности, таким как стандарт ITU-T H.264, альтернативно называемый MPEG-4, Часть 10, Усовершенствованное кодирование видео (AVC), или расширениям таких стандартов. Способы настоящего раскрытия, однако, не ограничены никаким конкретным стандартом кодирования. Другие примеры стандартов кодирования видео включают в себя MPEG-2 и ITU-T H.263. Хотя не показано на фиг. 1, в некоторых аспектах видеокодер 20 и видеодекодер 30 могут каждый интегрироваться с аудио кодером и декодером, и могут включать в себя соответствующие блоки MUX-DEMUX (мультиплексор-демультиплексор), или другое аппаратное обеспечение и программное обеспечение, чтобы осуществлять кодирование как аудио так и видео в общем потоке данных или отдельных потоках данных. Если применимо, блоки MUX-DEMUX могут соответствовать протоколу ITU H.223 мультиплексора, или другим протоколам, таким как протокол дейтаграмм пользователя (UDP).

[0059] Стандарт ITU-T H.264/MPEG-4 (AVC) был сформулирован Группой Экспертов по Кодированию видео ITU-T (VCEG) вместе с Группой экспертов по движущимся изображениям (MPEG) ISO/IEC как продукт коллективного партнерства, известного как объединенная команда по видео (JVT). В некоторых аспектах способы, описанные в настоящем раскрытии, могут быть применены к устройствам, которые обычно соответствуют стандарту H.264. Стандарт H.264 описан в Рекомендации H.264 ITU-T, Усовершенствованное кодирование видео для общих аудиовизуальных услуг, Группой по изучению ITU-T, и датированной мартом 2005, которая может быть упомянута здесь как стандарт H.264 или спецификация H.264, или стандарт H.264/AVC или спецификация. Объединенная команда по видео (JVT) продолжает работу над расширениями к H.264/MPEG-4 AVC.

[0060] Видеокодер 20 и видеодекодер 30 каждый может быть реализован как любое из множества подходящих схем кодера, таких как один или более микропроцессоров, цифровых сигнальных процессоров (DSPs), специализированных интегральных схем (ASICs), программируемых пользователем вентильных матриц (FPGAs), дискретной логики, программного обеспечения, аппаратного обеспечения, программно-аппаратных

средств или любых их комбинаций. Когда способы реализуются частично в программном обеспечении, устройство может сохранить инструкции для программного обеспечения в подходящем невременном считываемом компьютером носителе и выполнять инструкции в аппаратном обеспечении, используя один или более процессоров, чтобы

5 выполнить способы настоящего раскрытия. Каждый из видеокодера 20 и видеodeкодера 30 может быть включен в один или более кодеров или декодеров, любой из которых может интегрироваться как часть объединенного кодера/декодера (кодек) в соответствующем устройстве.

[0061] JCT-VC работает над развитием стандарта HEVC. Усилия по стандартизации

10 HEVC основаны на развивающейся модели устройства кодирования видео, называемой Тестовой Моделью HEVC (НМ). НМ предполагает несколько дополнительных способностей устройств кодирования видео относительно существующих устройств согласно, например, ITU-T H.264/AVC. Например, тогда как H.264 обеспечивает девять режимов кодирования с внутренним предсказанием, НМ может обеспечить целых

15 тридцать три режима кодирования с внутренним предсказанием. Хотя настоящее раскрытие может время от времени полагаться на WD7 HEVC в целях объяснения, способы настоящего раскрытия никоим образом не ограничены WD7. Стандарт HEVC продолжает развиваться, и рассматривается, что способы настоящего раскрытия могут быть совместимыми с будущими версиями HEVC. Например, способы в настоящем

20 раскрытии могут быть использованы в соединении с "HEVC Working Draft 10" или "WD10", который описан в документе JCTVC-L1003_v18, Bross и et al., "High Efficiency Video Coding (HEVC) Text Specification Draft 10», Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, Switzerland, 14-23 January 2013, который на 5 февраля 2013 может быть загружен с http://phenix.it-sudparis.eu/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v18.zip.

25

[0062] В общем рабочая модель НМ описывает, что видеокадрилы картинка могут быть разделены на последовательность блоков дерева или наибольших единиц кодирования (LCU), которые включают в себя выборки как яркости так и насыщенности

цвета. Данные синтаксиса в пределах потока битов могут определить размер для LCU, которая является наибольшей единицей кодирования в терминах количества пикселей.

30 Вырезка включает в себя совокупность последовательных блоков дерева в порядке кодирования. Видеокадрилы картинка могут быть разделены на одну или более вырезок. Каждый блок дерева может быть разделен на единицы кодирования (CU) согласно квадродереву. Обычно, структура данных квадродерева включает в себя один узел для

35 каждой CU, с корневым узлом, соответствующим блоку дерева. Если CU разделена на четыре суб-CU, узел, соответствующий CU, включает в себя четыре листовых узла, каждый из которых соответствует одной из суб-CU.

[0063] Каждый узел структуры данных квадродерева может обеспечить данные синтаксиса для соответствующей CU. Например, узел в квадродереве может включать

40 в себя флаг разделения, указывающий, разделена ли CU, соответствующая узлу, на суб-CU. Элементы синтаксиса для CU могут быть определены рекурсивно, и могут зависеть от того, разделена ли CU на единицы суб-CU. Если CU не разделяется далее, она называется листовой CU. В настоящем раскрытии четыре единицы суб-CU листовой CU будут также упоминаться как листовые CU, даже если не будет явного разделения

45 первоначальной листовой CU. Например, если CU с размером 16×16 не будет разделена далее, то четыре 8×8 единицы суб-CU будут также упоминаться как листовые CU, хотя 16×16 CU никогда не разделялась.

[0064] CU имеет аналогичную цель как макроблок стандарта H.264, за исключением

того, что CU не имеет различия в размерах. Например, блок дерева может быть разделен на четыре дочерних узла (также называемых единицы суб-CU), и каждый дочерний узел может в свою очередь быть родительским узлом и быть разделен еще на четыре дочерних узла. Заключительный неразделенный дочерний узел, называемый листовой узел

5 квадродерева, содержит узел кодирования, также называемый листовой CU. Данные синтаксиса, ассоциированные с закодированным потоком битов, могут определить максимальное количество раз, сколько блок дерева может быть разделен, называемое максимальной глубиной CU, и может также определить минимальный размер узлов кодирования. Соответственно, поток битов может также определить наименьшую

10 единицу кодирования (SCU). Настоящее раскрытие использует термин "блок", чтобы ссылаться на любую из CU, PU или TU, в контексте HEVC, или подобные структуры данных в контексте других стандартов (например, макроблоки и их суб-блоки в H.264/AVC).

[0065] CU включает в себя узел кодирования и единицы предсказания (PU), и единицы преобразования (TU), ассоциированные с узлом кодирования. Размер CU соответствует размеру узла кодирования и должен быть квадратным по форме. Размер CU может ранжироваться от 8×8 пикселей до размера блока дерева с максимумом 64×64 пикселей или больше. Каждая CU может содержать одну или более единиц PU и одну или более единиц TU. Данные синтаксиса, ассоциированные с CU, могут описывать, например,

20 разделение CU на одну или более единиц PU. Режимы разделения могут отличаться между тем, является ли CU закодированной в режиме пропуска или прямом, режиме внутреннего предсказания, или закодированной в режиме внешнего предсказания. Единицы PU могут быть разделены так, чтобы быть неквадратными по форме. Данные синтаксиса, ассоциированные с CU, могут также описывать, например, разделение CU

25 в одну или более единиц TU согласно квадродереву. TU может быть квадратной или неквадратной (например, прямоугольной) по форме

[0066] Стандарт HEVC обеспечивает преобразования согласно единицам TU, которые могут быть различными для различных единиц CU. Единицы TU обычно имеет размеры на основании размера единиц PU в пределах заданной CU, определенной для разделенной LCU, хотя это может не всегда иметь место. Единицы TU типично имеют тот же размер

30 или меньший, как единицы PU. В некоторых примерах остаточные выборки, соответствующие CU, могут быть подразделены на меньшие блоки, используя структуру квадродерева, известную как "остаточное квадродерево" (RQT). Листовые узлы RQT могут упоминаться как единицы преобразования (единицы TU). Значения пиксельной разности, ассоциированные с единицами TU, могут быть преобразованы, чтобы

35 сформировать коэффициенты преобразования, которые могут быть квантованы.

[0067] Листовая CU может включать в себя одну или более единиц предсказания (единиц PU). Обычно PU представляет пространственную область, соответствующую всем или части соответствующей CU, и может включать в себя данные для того, чтобы

40 извлечь опорную выборку для PU. Кроме того, PU включает в себя данные, относящиеся к предсказанию. Например, когда PU закодирована во внутреннем режиме, данные для PU могут быть включены в остаточное квадродерево (RQT), которое может включать в себя данные, описывающие режим внутреннего предсказания для TU, соответствующей PU. В качестве другого примера, когда PU является закодированной

45 во внешнем режиме, PU может включать в себя данные, определяющие один или более векторов движения для PU. Данные, определяющие вектор движения для PU, могут описывать, например, горизонтальный компонент вектора движения, вертикальный компонент вектора движения, разрешение для вектора движения (например, пиксельную

точность в одну четверть или пиксельную точность в одну восьмую), опорную картинку, на которую вектор движения указывает, и/или список опорных картинок (например, Список 0, Список 1 или Список C) для вектора движения.

[0068] Листовая CU, имеющая одну или более единиц PU, может также включать в себя одну или более единиц преобразования (единиц TU). Единицы преобразования могут быть заданы, используя RQT (также называемое структурой квадродерева TU), как описано выше. Например, флаг разделения может указывать, разделена ли листовая CU на четыре единицы преобразования. Затем каждая единица преобразования может быть разделена далее в дополнительные единицы суб-TU. Когда TU не разделена далее, она может упоминаться как листовая TU. Обычно для внутреннего кодирования все листовые единицы TU, принадлежащие листовой CU, совместно использует один и тот же режим внутреннего предсказания. Таким образом, один и тот же режим внутреннего предсказания обычно применяется, чтобы вычислить предсказанные значения для всех единиц TU листовой CU. Для внутреннего кодирования видеокодер может вычислить остаточное значение для каждой листовой TU, используя режим внутреннего предсказания, как разность между частью CU, соответствующей этой TU, и первоначальным блоком. TU не обязательно ограничена размером PU. Таким образом, единицы TU могут быть большими или меньшими, чем PU. Для внутреннего кодирования PU может быть совмещена с соответствующей листовой TU для той же самой CU. В некоторых примерах максимальный размер листовой TU может соответствовать размеру соответствующей листовой CU.

[0069] Кроме того, единицы TU листовых CU могут быть также ассоциированы с соответствующими структурами данных квадродерева, называемых остаточными квадродеревами (RQTs). Таким образом, листовая CU может включать в себя квадродерево, указывающее, как листовая CU разделена на единицы TU. Корневой узел квадродерева TU обычно соответствует листовой CU, в то время как корневой узел квадродерева CU обычно соответствует блоку дерева (или LCU). Единицы TU в RQT, которые не разделены, упоминаются как листовые TU. В целом настоящее раскрытие использует термины CU и TU, чтобы ссылаться на листовые CU и листовые TU, соответственно, если не отмечено иначе.

[0070] Видеопоследовательность типично включает в себя последовательность кадров или картинок видео. Группа картинок (GOP) обычно содержит последовательность из одной или более картинок видео. GOP может включать в себя данные синтаксиса в заголовке GOP, заголовке одной или более картинок, или в другом месте, которые описывают совокупность картинок, включенных в GOP. Каждая вырезка картинки может включать в себя данные синтаксиса вырезки, которые описывают режим кодирования для соответствующей вырезки. Видеокодер 20 типично оперирует над блоками видео в пределах индивидуальных видео вырезок, чтобы закодировать видеоданные. Блок видео может соответствовать узлу кодирования внутри CU. Блоки видео могут иметь фиксированный или переменный размеры, и могут отличаться по размеру согласно заданному стандарту кодирования.

[0071] В качестве примера НМ поддерживает предсказание в PU различных размеров. Предполагая, что размер конкретной CU равен $2N \times 2N$, НМ поддерживает внутреннее предсказание в PU размером $2N \times 2N$ или $N \times N$, и внешнее предсказание в PU симметричных размеров, равных $2N \times 2N$, $2N \times N$, $N \times 2N$ или $N \times N$. НМ также поддерживает асимметричное разделение для внешнего предсказания в PU размером $2N \times nU$, $2N \times nD$, $nL \times 2N$, и $nR \times 2N$. При асимметричном разделении одно направление CU не разделено, в то время как другое направление разделено на 25% и 75%. Часть CU, соответствующая

25%-ому разделению, обозначена “n”, сопровождаемым индикацией «верхний», «нижний», «левый» или «правый». Таким образом, например, “2N×nU” относится к 2N×2N CU, которая разделена горизонтально с 2N×0,5N PU сверху и 2N×1,5N PU внизу.

[0072] В настоящем раскрытии “N×N” и “N на N” могут быть использованы

5 взаимозаменяемо, чтобы ссылаться на пиксельные измерения блока видео в терминах вертикального и горизонтального измерений, например, 16×16 пикселей или 16 на 16 пикселей. Обычно, блок 16×16 имеет 16 пикселей в вертикальном направлении (y=16) и 16 пикселей в горизонтальном направлении (x=16). Аналогично, блок N×N обычно имеет N пикселей в вертикальном направлении и N пикселей в горизонтальном
10 направлении, где N представляет неотрицательное целочисленное значение. Пиксели в блоке могут быть размещены в рядах и колонках. Кроме того, блоки не обязательно должны иметь то же количество пикселей в горизонтальном направлении, как в вертикальном направлении. Например, блоки могут содержать N×M пикселей, где M не обязательно равно N.

15 [0073] После кодирования с внутренним предсказанием или внешним предсказанием, используя единицы PU из CU, видеокодер 20 может вычислить остаточные данные для единиц TU в этой CU. Единицы PU могут содержать данные синтаксиса, описывающие способ или режим генерирования предсказывающих пиксельных данных в пространственной области (также называемой пиксельной областью), и единицы TU
20 могут содержать коэффициенты в области преобразования после применения преобразования, например, дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлет преобразования или концептуально подобного преобразования к остаточным видео данным. Остаточные данные могут соответствовать пиксельным разностям между пикселями незакодированной картинки и значениями
25 предсказания, соответствующими единицам PU. Видеокодер 20 может сформировать единицы TU, включающие остаточные данные для CU, и затем преобразовать единицы TU, чтобы сформировать коэффициенты преобразования для этой CU.

[0074] После любого преобразования, чтобы сформировать коэффициенты преобразования, видеокодер 20 может выполнить квантование коэффициентов
30 преобразования. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются, чтобы возможно уменьшить величину данных, использованных для представления коэффициентов, обеспечивая дальнейшее сжатие. Процесс квантования может уменьшить битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Например, значение n-битов может быть
35 округлено в меньшую сторону к m-битовому значению во время квантования, где n больше, чем m.

[0075] Вслед за квантованием видеокодер может сканировать коэффициенты преобразования, формируя одномерный вектор из двумерной матрицы, включающей в себя квантованные коэффициенты преобразования. Сканирование может быть
40 разработано, чтобы поместить коэффициенты более высокой энергии (и поэтому более низкой частоты) впереди массива и поместить коэффициенты более низкой энергии (и поэтому более высокой частоты) в конце массива. В некоторых примерах видеокодер 20 может использовать заранее заданный порядок сканирования, чтобы сканировать квантованные коэффициенты преобразования, чтобы сформировать преобразованный
45 в последовательную форму вектор, который может быть энтропийно кодирован. В других примерах видеокодер 20 может выполнить адаптивное сканирование. После просмотра квантованных коэффициентов преобразования, чтобы сформировать одномерный вектор, видеокодер 20 может энтропийно кодировать одномерный вектор,

например, согласно контекстно-адаптивному кодированию с переменной длиной кода (CAVLC), контекстно-адаптивному двоичному арифметическому кодированию (CABAC), основанному на синтаксисе контекстно-адаптивному двоичному арифметическому кодированию (SBAC), энтропийному кодированию с разделением интервала вероятности (PIPE) или другой методологии энтропийного кодирования. Вideoкодер 20 может также энтропийно кодировать элементы синтаксиса, ассоциированные с закодированными видео данными, для использования видеодекодером 30 при декодировании видеоданных.

[0076] Чтобы выполнить CABAC, видеокодер 20 может назначить контекст в контекстной модели символу, который должен быть передан. Контекст может относиться к тому, например, являются ли соседние значения символа ненулевыми или нет. Чтобы выполнить CAVLC, видеокодер 20 может выбрать код с переменной длиной слова для символа, который должен быть передан. Кодовые слова в VLC могут быть построены таким образом, что относительно более короткие коды соответствуют более вероятным символам, в то время как более длинные коды соответствуют менее вероятным символам. Таким образом, использование VLC может достигнуть экономии битов, например, используя ключевые слова равной длины для каждого символа, который должен быть передан. Определение вероятности может быть на основании на контексте, назначенном на символ.

[0077] В соответствии со способами настоящего раскрытия, видеокодер 20 и видеодекодер 30 могут быть сконфигурированы, чтобы выполнить примерную функцию «GetLSB (poc, len)», где poc - целое число, и len - положительное целое число, следующим образом.

$$\text{GetLSB (poc, len)} = \text{poc} - (1 \ll \text{len}) * \text{Floor}\left(\frac{\text{poc}}{1 \ll \text{len}}\right) \quad (1)$$

В примере этой функции (1) « \ll » представляет оператор побитового сдвига влево, и Floor() возвращает значение аргумента, переданного ему, которое округлено в меньшую сторону.

[0078] Видеокодер 20 и видеодекодер 30 могут быть сконфигурированы, чтобы выполнить один или более различных способов для вывода наборов опорных картинок (RPS) и сигнализации долгосрочных опорных картинок (LTRP), которые должны быть включены в RPS закодированной картинки. Обычно видеокодер 20 и видеодекодер 30 могут быть сконфигурированы, чтобы выполнить любой или все из примеров, описанных в настоящем раскрытии, по одному или в любой комбинации. В целом настоящее раскрытие описывает способы, относящиеся к следующим способам кодирования и декодирования, где видеокодер 20 и видеодекодер 30 могут быть сконфигурированы, чтобы выполнить любой из этих способов в комбинации с другими способами WD7 или другими структурами кодирования видео.

[0079] В одном примере информация, сообщенная для конкретной LTRP, которая должны быть включена в RPS текущей картинки, может быть значением дельты POC или между текущей картинкой и LTRP или между LTRP и предыдущей картинкой RAP в порядке декодирования. В качестве одной альтернативы, сигнализация LTRP и вывод могут остаться теми же самыми как в WD7, за исключением того, что элемент синтаксиса delta_poc_msb_cycle_lf[i] всегда сигнализируется для каждой LTRP в заголовке вырезки, или посредством удаления элемента синтаксиса delta_poc_msb_present_flag[i], или посредством обеспечения что delta_poc_msb_present_flag[i] равен 1 для всех LTRP. В качестве другой альтернативы, информация, сигнализируемая для конкретной

LTRP, которая должна быть включена в RPS текущей картинки, может или быть значением дельты РОС между LTRP и предыдущей картинкой RAP в порядке декодирования, или вышеупомянутый альтернативный способ может быть применен.

[0080] В одной альтернативе картинкам после картинки CRA или BLA в порядке декодирования может быть запрещено использовать картинки, предшествующие картинкам CRA или BLA в порядке декодирования, в качестве картинок LTRP. Следовательно, начальным картинкам может быть запрещено ссылаться на картинки, которые предшествуют ассоциированной картинке CRA или BLA, в порядке декодирования, в качестве картинок LTRP. Для не-TFD картинок количество битов, требуемых для сигнализации LSB в LTRP, может быть определено картинкой с наименьшим РОС, которая не маркирована как TFD, и которая предшествует, в порядке декодирования, текущей картинке, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования включая предыдущую картинку RAP в порядке декодирования. Для картинок TFD количество битов, требуемых для сигнализации LSB LTRP, может быть определено картинкой с наименьшим РОС, которая предшествует, в порядке декодирования, текущей картинке, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования включая предыдущую картинку RAP в порядке декодирования.

[0081] В одной альтернативе картинкам TFD, ассоциированным с картинкой RAP, может быть запрещено следовать за, в порядке декодирования, любой картинкой DLP, ассоциированной с одной и той же картинкой RAP. Для не-TFD картинок количество битов, требуемых для сигнализации LSB LTRP, может быть определено картинкой с наименьшим РОС, которая не маркирована как картинка TFD, и которая предшествует текущей картинке, в порядке декодирования, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования, включая предыдущую картинку RAP в порядке декодирования. Для картинок TFD количество битов, требуемых для сигнализации LSB LTRP, может быть определено картинкой с наименьшим РОС, которая предшествует, в порядке декодирования, текущей картинке, и следует за, в порядке вывода, RAP перед предыдущей картинкой RAP в порядке декодирования включая RAP перед предыдущей картинкой RAP в порядке декодирования.

[0082] В одной альтернативе начальным картинкам, ассоциированным с картинкой CRA или BLA, может быть запрещено ссылаться на ассоциированную картинку CRA или BLA или другие начальные картинки, ассоциированные с той же картинкой CRA или BLA, в качестве картинок LTRP. Количество битов, требуемых для сигнализации LSB LTRP, может быть определено картинкой RAP, которая предшествует, в порядке вывода, текущей картинке.

[0083] В одной альтернативе может быть наложено следующее ограничение. Любая картинка, предшествующая картинке CRA или BLA в порядке декодирования, должна предшествовать любая картинка DLP, ассоциированная с картинкой CRA или BLA в порядке вывода. Кроме того, может также быть наложено следующее ограничение, что любой картинке TFD, ассоциированной с картинкой CRA или BLA, должна предшествовать любая картинка DLP, ассоциированная с картинкой CRA или BLA в порядке вывода. Эти примерные ограничения запрещают случаи, в которых картинки DLP перемежаются, в порядке вывода, с картинками TFD или картинками ранее, в порядке декодирования, чем ассоциированная картинка CRA или BLA. В тех случаях, когда случайный доступ или выполнение переключение потока от картинки CRA или BLA, скорость передачи начальной картинки может быть меньшей, чем после того, как выводятся картинки DLP. При потоковой передаче с адаптацией с частым

переключением потока, например, в контексте динамической адаптивной потоковой передачи по HTTP (DASH) такая флуктуация в скорости передачи картинки может вызвать нежелательный пользовательский опыт. Для картинки CRA или BLA, WD7 предусматривает, что не должно быть картинки, предшествующей картинке CRA или BLA в порядке декодирования и после картинки CRA или BLA в порядке вывода. Включение для этого ограничения было по подобной причине.

[0084] В одной альтернативе начальным картинкам (более конкретно, картинкам TFD), ассоциированным с первой картинкой, разрешено использовать картинки DLP, ассоциированные со второй картинкой, для ссылки с внешним предсказанием в качестве картинок STRP или LTRP, при этом первая картинка является картинкой CRA, вторая картинка является картинкой CRA или BLA, и вторая картинка и первые картинки являются двумя смежными картинками RAP в порядке декодирования, с первой картинкой, следующей за второй картинкой в порядке декодирования. Такая ссылка с внешним предсказанием картинок DLP с помощью начальных картинок может быть разрешено посредством разрешения таким картинкам DLP присутствовать в наборе опорных картинок таких начальных картинок, и может обеспечить более гибкое и более высокоэффективное кодирование начальных картинок. Соответственно, определения картинок CRA и BLA могут быть изменены, чтобы разрешить такое внешнее предсказание.

[0085] В одной альтернативе перемежение порядка декодирования картинок TFD, ассоциированных с картинкой CRA или BLA и картинками, следующими за картинкой CRA или BLA и в порядке декодирования и в порядке вывода, то есть находящиеся сзади картинки для картинки CRA или BLA, могут быть запрещены. Другими словами, любые картинки TFD, ассоциированные с картинкой CRA или BLA, могут быть обязаны предшествовать любой находящейся сзади картинки для картинки CRA или BLA, в порядке декодирования.

[0086] Альтернативно, перемежение порядка декодирования начальных картинок, ассоциированных с картинкой CRA или BLA и картинками, следующими за картинкой CRA или BLA и в порядке декодирования и в порядке вывода, может быть запрещен. Другими словами, любая начальная картинка, ассоциированная с картинкой CRA или BLA, может быть обязана предшествовать любой картинке, в порядке декодирования, которая следует за картинкой CRA или BLA и в порядке декодирования и в порядке вывода.

[0087] Следующие примеры реализаций обеспечивают дополнительные детали для примеров, описанных выше. Должно быть понятно, что различные способы, описанные в примерных реализациях, могут быть использованы по одному или в любой комбинации.

[0088] Первая примерная реализация обеспечивает способ для попытки решить проблему ошибки устойчивости и гарантировать, что отклонение RPS является автономным (то есть не зависящим от статуса DPB), и является в то же самое время свободным от проблем, когда произвольный доступ выполняется от картинки CRA или когда картинка CRA преобразуется в картинку BLA. В этой первой примерной реализации информация, сигнализируемая для конкретной LTRP, которая должна быть включена в RPS текущей картинки, является значением дельты POC, или между текущей картинкой и LTRP (режим 1) или между LTRP и предыдущей картинкой RAP в порядке декодирования (режим 2). Эта первая примерная реализация также согласуется с текущей сигнализацией RPS и вывода для картинок STRP в WD7. Кодеры, такие как видеокодер 20, могут выбрать один из этих двух режимов для каждой LTRP, которая

должна быть сигнализирована, который может привести к меньшему количеству битов, которые должны использоваться. Например, если LTRP ближе к предыдущему RAP в порядке декодирования, чем текущая картинка в расстоянии порядка вывода/отображения, то видеокодер 20 может быть сконфигурирован, чтобы использовать режим 2. Иначе, видеокодер 20 может использовать режим 1. Аналогично, видеодекодер 30 может принять данные синтаксиса, описывающие режим для использования, и использовать режим, сигнализированный данными синтаксиса.

[0089] Чтобы достигнуть первой примерной реализации, следующие изменения синтаксиса, семантики, и процесса кодирования относительно WD7 могут быть использованы, где подчеркнутый текст представляет изменение относительно WD7. Таблица 5 ниже обеспечивает примерный набор синтаксиса для набора параметров картинки (PPS).

ТАБЛИЦА 5	
	Дескриптор
pic_parameter_set_rbsp() {	
pic_parameter_set_id	ue(v)
...	
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
poc_lt_idc	u(2)
...	
rbp_trailing_bits()	
}	

[0090] В этой первой примерной реализации, как показано в Таблице 5, синтаксис PPS включает в себя дополнительный элемент синтаксиса, poc_lt_idc, относительно WD7. Семантика для других элементов синтаксиса может остаться той же самой. В этом примере poc_lt_idc может быть использован для указания присутствия delta_poc_lt_curr_pic_flag[i] в заголовке вырезки, и, когда delta_poc_lt_curr_pic_flag[i] не присутствует в заголовке вырезки, для определения логически выведенного значения delta_poc_lt_curr_pic_flag[i] в заголовке вырезки. Если poc_lt_idc равно 0 или 1, delta_poc_lt_curr_pic_flag[i] не присутствует в заголовке вырезки, и значение логически выводится, чтобы быть равным poc_lt_idc, в этом примере. Когда poc_lt_idc равно 2, delta_poc_lt_curr_pic_flag[i] сигнализируется в заголовке вырезки, в этом примере. Значение 3 для poc_lt_idc сохранено для будущего использования, в этом примере.

[0091] Таблица 6 обеспечивает примерный набор синтаксиса для заголовка вырезки в соответствии с первой примерной реализацией.

ТАБЛИЦА 6	
	Дескриптор
slice_header() {	
...	
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i=0; i<num_long_term_pics; i++) {	
if(poc_lt_idc==2)	
delta_poc_lt_curr_pic_flag[i]	u(1)
delta_poc_lt_len[i]	ue(v)
if(delta_poc_lt_len[i] !=0.)	
delta_poc_lt[i]	u(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
...	

}	
---	--

[0092] В этом примере заголовок вырезки включает в себя дополнительные элементы синтаксиса `delta_poc_lt_curr_pic_flag[i]`, `delta_poc_lt_len[i]`, и в некоторых случаях `delta_poc_lt[i]`. Примерный набор семантики описан для этих элементов синтаксиса

ниже. Семантика для других элементов синтаксиса может остаться той же самой как в WD7.

[0093] В этой первой примерной реализации `delta_poc_lt_curr_pic_flag[i]` равно 1, может определить, что `delta_poc_lt[i]` представляет разность счета по порядку картинок между текущей картинкой и *i*-th долгосрочной опорной картинкой.

`delta_poc_lt_curr_pic_flag[i]`, равно 0, может задавать, что `delta_poc_lt[i]` представляет разность счета по порядку картинок между предыдущей картинкой RAP в порядке декодирования и *i*-th долгосрочной опорной картинкой. Если `poc_lt_idc` (например, из Таблицы 5) равно 0 или 1, значение `delta_poc_lt_curr_pic_flag[i]` может быть логически выведено, чтобы быть равным `poc_lt_idc`.

[0094] В этой первой примерной реализации `delta_poc_lt_len[i]` может быть использовано для задания количества битов, чтобы представить `delta_poc_lt[i]`. Если `delta_poc_lt_curr_pic_flag[i]` равно 0, значение `delta_poc_lt_len[i]` может быть в диапазоне от 0 до 24, включительно. Иначе, (`delta_poc_lt_curr_pic_flag[i]` равно 1), значение `delta_poc_lt_len[i]` может быть в диапазоне 1-24, включительно.

[0095] Переменная `DeltaPocLtLen[i]` может быть выведена следующим образом:

```
if(i==0)
```

```
DeltaPocLtLen[i]=delta_poc_lt_len[i]
```

```
else
```

```
DeltaPocLtLen[i]=DeltaPocLtLen[i-1]+delta_poc_lt_len[i]
```

[0096] Альтернативно, `delta_poc_lt_len[i]` может быть изменено на `poc_lt_len[i]`, которое является закодированной с фиксированной длиной кода, например, как *u* (N), с N равным 5. В этом случае, семантика может быть следующий. `poc_lt_len[i]` может задавать количество битов, чтобы представить `delta_poc_lt[i]`. Если `delta_poc_lt_curr_pic_flag[i]` равно 0, значение `poc_lt_len[i]` может быть в диапазоне от 0 до 24, включительно, Иначе, (`delta_poc_lt_curr_pic_flag[i]` равно 1), значение `poc_lt_len[i]` может быть в диапазоне 1-24, включительно. Переменная `DeltaPocLtLen[i]` может быть установлена равной `poc_lt_len[i]`.

[0097] В этой первой примерной реализации `delta_poc_lt[i]` может задавать значение дельты счета по порядку картинки *i*-th долгосрочной опорной картинки, которая включена в набор долгосрочных опорных картинок текущей картинки. Длина `delta_poc_lt[i]` может быть равна `DeltaPocLtLen[i]`.

[0098] Видеодекодер 30 может быть сконфигурирован, чтобы выполнить процесс декодирования для набора опорных картинок в соответствии с этой первой примерной реализацией следующим образом. Видеодекодер 20 может выполнить аналогичный процесс, в то же время кодируя видеоданные, чтобы генерировать набор опорных картинок. Видеодекодер 30 может вызвать этот процесс декодирования однократно для картинки, после декодирования заголовка вырезки, но до декодирования любой единицы кодирования и до процесса декодирования для конструирования списка опорных картинок вырезки, как определено в подпункте 8.3.3 в WD7. Процесс может привести к маркировке одной или более опорных картинок как «неиспользуемая для ссылки». Цитирования из WD7, начинающиеся с "ПРИМЕЧАНИЯ", должны рассматриваться просто как цитирования, взятые непосредственно из WD7, и не необходимые для способов настоящего раскрытия.

ПРИМЕЧАНИЕ 1 - Набор опорных картинок является абсолютным описанием опорных картинок, используемых в процессе декодирования текущих и будущих закодированных картинок. Сигнализация набора опорных картинок является явной в том смысле, что все опорные картинки, включенные в набор опорных картинок, перечисляются явно и не имеется процесса конструирования набора опорных картинок по умолчанию в декодере, который зависит от статуса буфера декодированных картинок.

[0099] Опорные картинки могут быть идентифицированы их значениями PicOrderCntVal.

[0100] Пять списков значений счета по порядку картинки могут быть сконструированы, чтобы вывести набор опорных картинок - PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll с NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и количество NumPocLtFoll элементов, соответственно.

[0101] Если текущая картинка является картинкой CRA, которая является первой закодированной картинкой в потоке битов, картинкой IDR, или картинкой BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll все устанавливаются в «пусто», и NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и NumPocLtFoll могут быть все установлены в 0.

[0102] Иначе, следующее можно применить для вывода пяти списков значений счета по порядку картинки и количеств записей.

```

for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS0[ StRpsIdx ][ i ] )
        PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
5    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
NumPocStCurrBefore = j

for( i = 0, j = 0; i < NumPositivePics[ StRpsIdx ]; i++ )
10    if( UsedByCurrPicS1[ StRpsIdx ][ i ] )
        PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
NumPocStCurrAfter = j
15    NumPocStFoll = k
(8-5)

for( i = 0, j = 0, k = 0; i < num_long_term_pics; i++ )
    if( used_by_curr_pic_lt_flag[ i ] ) {
20        if( delta_poc_lt_curr_pic_flag[ i ] )
            PocLtCurr[ j ] = PicOrderCntVal - delta_poc_lt[ i ]
        else
            PocLtCurr[ j ] = PrevRapPicPoc + delta_poc_lt[ i ]
        j++
25    }
    else {
        if( delta_poc_lt_curr_pic_flag[ i ] )
            PocLtFoll[ k ] = PicOrderCntVal - delta_poc_lt[ i ]
        else
30            PocLtFoll[ k ] = PrevRapPicPoc + delta_poc_lt[ i ]
        k++
    }
    NumPocLtCurr = j
35    NumPocLtFoll = k

```

где PicOrderCntVal и PrevRapPicPoc являются значениями счета по порядку картинки текущей картинки и предыдущей картинки RAP в порядке декодирования, соответственно, как определено в подпункте 8.2.1.

40 ПРИМЕЧАНИЕ - 2 Значение StRpsIdx в диапазоне от 0 до num_short_term_ref_pic_sets-1, включительно, может указывать, что набор краткосрочных опорных картинок из активного набора параметров последовательности используется, где StRpsIdx - индекс набора краткосрочных опорных картинок на список наборов краткосрочных опорных картинок в порядке, в котором они сигнализируются в наборе параметров последовательности. StRpsIdx, равные num_short_term_ref_pic_sets, указывает, что
45 используется набор краткосрочных опорных картинок, явно сигнализированный в заголовке вырезки.

[0103] Набор опорных картинок может включать в себя пять списков опорных картинок - RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr и

RefPicSetLtFoll. Переменная NumPocTotalCurr может быть установлена равной $\text{NumPocStCurrBefore} + \text{NumPocStCurrAfter} + \text{NumPocLtCurr}$. При декодировании Р или В вырезки, WD7 требует, для согласования потока битов, что значение NumPocTotalCurr не должно быть равным 0.

5 ПРИМЕЧАНИЕ 3 - RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr могут содержать все опорные картинки, которые могут быть использованы во внешнем предсказании текущей картинки, и которые могут быть использованы во внешнем предсказании одной или более картинок после текущей картинки в порядке декодирования. RefPicSetStFoll и RefPicSetLtFoll могут включать в себя все опорные
10 картинки, которые не используются во внешнем предсказании текущей картинки, но могут быть использованы во внешнем предсказании одной или более картинок после текущей картинки в порядке декодирования.

[0104] Маркировка опорной картинки может быть «неиспользуемая для ссылки», «используемая для краткосрочной ссылки», или «используемая для долгосрочной
15 ссылки», но только одной среди этих трех, в WD7. Когда опорная картинка упоминается как маркированная как «используемая для ссылки», это все вместе относится к картинке, маркированной как «используемая для краткосрочной ссылки», или «используемая для долгосрочной ссылки» (но не обе в WD7). Опорная картинка, которая маркирована как «используемая для краткосрочной ссылки», упоминается как краткосрочная опорная
20 картинка. Опорная картинка, которая маркирована как «используемая для долгосрочной ссылки», упоминается как долгосрочная опорная картинка.

[0105] Процесс выведения для набора опорных картинок и маркировка картинок выполняют согласно следующим упорядоченным этапам, где DPB относится к буферу декодированных картинок, как описано в Приложении С WD7, где (#-#) относится к
25 номеру уравнения в WD7:

```

1. Следующее применяется:
   for(i=0; i<NumPocLtCurr; i++) {
     if(имеется картинка picX в DPB с PicOrderCntVal равным PocLtCurr[i])
       RefPicSetLtCurr[i]=picX
30   else
       RefPicSetLtCurr[i]=«нет опорной картинки»
   }
   for(i=0; i<NumPocLtFoll; i++) {
     if(имеется картинка picX в DPB с PicOrderCntVal равным PocLtFoll[i])
35   RefPicSetLtFoll[i]=picX
     else
       RefPicSetLtFoll[i]=«нет опорной картинки»
   }
2. Все опорные картинки, включенные в RefPicSetLtCurr и RefPicSetLtFoll, маркированы
40 как «используемая для долгосрочной ссылки»
3. Следующее применяется:
   for(i=0; i<NumPocStCurrBefore; i++)
     if(имеется картинка picX в DPB с PicOrderCntVal равным PocStCurrBefore[i])
       RefPicSetStCurrBefore[i]=picX
45   else
       RefPicSetStCurrBefore[i]=«нет опорной картинки»
   for(i=0; i<NumPocStCurrAfter; i++)
     if(имеется картинка picX in the DPB with PicOrderCntVal равным PocStCurrAfter[i])

```

(8-6)

RefPicSetStCurrAfter[i]=picX

else

RefPicSetStCurrAfter[i]=«нет опорной картинки» (8-7)

for(i=0; i<NumPocStFoll; i++)

5 if(имеется картинка picX в DPB с PicOrderCntVal равным PocStFoll[i])

RefPicSetStFoll[i]=picX

else

RefPicSetStFoll[i]=«нет опорной картинки»

10 4. Все опорные картинки, включенные в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetStFoll, маркированы как «используемая для краткосрочной ссылки».

[0106] Все опорные картинки в буфере декодированных картинок, которые не включены в RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, могут быть маркированы как «неиспользуемые для ссылки».

15 ПРИМЕЧАНИЕ 4 - Могут быть одна или более опорных картинок, которые включены в набор опорных картинок, но не присутствуют в буфере декодированных картинок. Должны быть проигнорированы записи в RefPicSetStFoll или RefPicSetLtFoll, которые не равны «нет опорной картинки». Если любое из следующих двух условий не верно, неумышленная потеря картинок должна быть логически выведена, в WD7, для каждой записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr, которая

20 равна «нет опорной картинки»: а) первая закодированная картинка в потоке битов является картинкой CRA и текущая закодированная картинка является картинкой TFD, ассоциированной с первой закодированной картинкой в потоке битов; б) предыдущая картинка RAP, предшествующая текущей закодированной картинке в порядке декодирования, является картинкой BLA, и текущая закодированная картинка является

25 картинкой TFD, ассоциированной с картинкой BLA.

[0107] Альтернативно, WD7 может потребовать, чтобы в вышеупомянутом каждая идентифицированная picX в DPB должна быть маркирована как «используемая для ссылки». Альтернативно, WD7 может ограничить, что картинка, маркированная как «неиспользуемая для ссылки», никогда не должна маркироваться как «используемая

30 для ссылки».

[0108] Альтернативно, вышеупомянутые пять этапов могут быть заменены следующим:

1. Следующее применяется:

for(i=0; i<NumPocLtCurr; i++) {

35 if(имеется картинка picX в DPB которая имеет PicOrderCntVal равное PocLtCurr[i] и которая

маркирована как «используемая для ссылки»)

RefPicSetLtCurr[i]=picX

else

40 RefPicSetLtCurr[i]=«нет опорной картинки»

}

(8-6)

for(i= 0; i< NumPocLtFoll; i++) {

if(имеется картинка picX в DPB которая имеет PicOrderCntVal равное PocLtFoll[i] и которая

45 маркирована как «используемая для ссылки»)

RefPicSetLtFoll[i]=pic

else

RefPicSetLtFoll[i]=«нет опорной картинки»

}

2. Все опорные картинки, включенные в RefPicSetLtCurr и RefPicSetLtFoll, маркируются как «используемая для долгосрочной ссылки»

3. Следующее применяется:

for(i=0; i<NumPocStCurrBefore; i++)

if(имеется картинка picX в DPB которая имеет PicOrderCntVal равное PocStCurrBefore [i]

и которая

маркирована как «используемая для ссылки»)

RefPicSetStCurrBefore[i]=picX

else

RefPicSetStCurrBefore[i]=«нет опорной картинки»

for(i=0; i<NumPocStCurrAfter; i++)

if(имеется картинка picX в DPB которая имеет PicOrderCntVal равное PocStCurrAfter

[i]

и которая

маркирована как «используемая для ссылки»)

RefPicSetStCurrAfter[i]=picX

else

RefPicSetStCurrAfter[i]=«нет опорной картинки»

(8-7)

for(i=0; i<NumPocStFoll; i++)

if(имеется картинка picX в DPB которая имеет PicOrderCntVal равное PocStFoll[i] и которая

маркирована как «используемая для ссылки»)

RefPicSetStFoll[i]=picX

else

RefPicSetStFoll[i]=«нет опорной картинки»

4. Все опорные картинки, включенные в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetStFoll, маркируются как «используемая для краткосрочной ссылки».

5. Все опорные картинки в буфере декодированных картинок, которые не включены в RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, маркируются как «неиспользуемые для ссылки».

ПРИМЕЧАНИЕ 4 - Могут быть одна или более опорных картинок, которые

включены в набор опорных картинок, но не присутствуют в буфере декодированных картинок. Записи в RefPicSetStFoll или RefPicSetLtFoll, которые равны «нет опорной картинки», должны быть проигнорированы, в соответствии с WD7. Если только любое из следующих двух условий не является верно, неумышленная картинок потеря должна быть логически выведена для каждой записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter

и RefPicSetLtCurr, которая не равна «нет опорной картинки» в WD7: а) первая

закодированная картинка в потоке битов является картинкой CRA и текущая

закодированная картинка является картинкой TFD, ассоциированной с первой

закодированной картинкой в потоке битов; б) предыдущая картинка RAP,

предшествующая текущей закодированной картинке в порядке декодирования, является

картинкой BLA, и текущая закодированная картинка является картинкой TFD,

ассоциированной с картинкой BLA.

[0109] Альтернативно, порядок вышеупомянутых этапов может быть изменен на 3, 4, 1, 2, и 5.

[0110] В некоторых примерах WD7 требует, для согласования потока битов, чтобы набор опорных картинок был ограничен следующим образом:

- Не должно быть опорной картинки с temporal_id большим, чем таковой текущей картинки, включенной в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr.

5 - Когда текущая картинка является картинкой TLA, не должно быть опорной картинки, включенной в набор опорных картинок с temporal_id большим чем или равным temporal_id текущей картинки.

10 - Не должно быть опорной картинки, включенной в набор опорных картинок, который предшествует, в порядок вывода, любой картинке RAP, которая предшествует текущей картинке и в порядке декодирования и в порядке вывода.

15 - Пока любое из следующих условий не будет верно, не должно быть записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetLtCurr, которая не равна «нет опорной картинки»: а) первая закодированная картинка в потоке битов является картинкой CRA и текущая закодированная картинка является картинкой TFD, ассоциированной с первой закодированной картинкой в потоке битов; б) предыдущая картинка RAP, предшествующая текущей закодированной картинке в порядке декодирования, является картинкой BLA, и текущая закодированная картинка является картинкой TFD, ассоциированной с картинкой BLA.

20 ПРИМЕЧАНИЕ 5 - опорная картинка не может быть включена в больше, чем один из пяти списков набора опорных картинок.

[0111] В некоторых примерах WD7 обеспечивает, что вышеупомянутыми ограничениями могут быть следующие:

- Не должно быть опорной картинки с temporal_id большим чем таковой текущей картинки, включенной в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetLtCurr.

25 - Когда текущая картинка является картинкой TLA, не должно быть опорной картинки, включенной в набор опорных картинок с temporal_id больше, чем или равный temporal_id текущей картинки.

30 - Пусть prevRapPic будет предыдущей картинкой RAP в порядке декодирования. Когда текущая картинка является картинкой RAP или предшествует, в порядке вывода, prevRapPic, не должно быть опорной картинки, включенной в набор опорных картинок, которая предшествует, в порядок вывода, любой картинке RAP, которая предшествует текущей картинке и в порядке декодирования и в порядке вывода.

35 - Пусть prevRapPic будет предыдущей картинкой RAP в порядке декодирования. Когда текущая картинка не является картинкой RAP и следует за prevRapPic в порядке вывода, не должно быть опорной картинки, включенной в набор опорных картинок, которая предшествует prevRapPic в порядке декодирования, и не должно быть опорной картинки, включенной в набор опорных картинок, которые являются картинками TFD.

40 - Если любое из следующих условий не будет верно, не должно быть записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetLtCurr, которая не равна «нет опорной картинки»: а) первая закодированная картинка в потоке битов является картинкой CRA и текущая закодированная картинка является картинкой TFD, ассоциированной с первой закодированной картинкой в потоке битов; б) предыдущая картинка RAP, предшествующая текущей закодированной картинке в порядке декодирования, является картинкой BLA, и текущая закодированная картинка является картинкой TFD, ассоциированной с картинкой BLA.

45 ПРИМЕЧАНИЕ 5 - Опорная картинка не может быть включена в больше, чем один из пяти списков набора опорных картинок, в соответствии с WD7.

[0112] Вторая примерная реализация, которая основана на структуре синтаксиса в

HEVC WD7, включает в себя способ, чтобы всегда сигнализировать цикл MSB LTRP и удалить элемент синтаксиса `delta_poc_msb_present_flag[i]`. Снова, должно быть понятно, что различные способы первой примерной реализации и второй примерной реализации могут быть объединены в любой комбинации. Видеокодер 20 и декодер 30 могут

быть сконфигурированы, чтобы выполнить любой или все из способов этой второй примерной реализации, одну или в любой комбинации с одним или более способов первой примерной реализации, описанной выше. Таблица 7 ниже обеспечивает примерный набор синтаксиса для заголовка вырезки в соответствии с этой второй примерной реализацией.

ТАБЛИЦА 7

slice_header() {	Дескриптор
...	u(1)
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i=0; i<num_long_term_pics; i++) {	
poc_lsb_lt[i]	u(v)
delta_poc_msb_cycle_lt[i]	ue(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
...	
byte_alignment()	
}	

[0113] В синтаксисе заголовка вырезки примера из Таблицы 7 элемент синтаксиса `delta_poc_msb_cycle_lt[i]` добавлен относительно WD7. Семантика для этого элемента синтаксиса может быть такой же как описано ниже, в то время как семантика для других элементов синтаксиса может остаться той же самой как в WD7, если не указано иначе. В этой второй примерной реализации `delta_poc_msb_cycle_lt[i]` может быть использован для определения значения старших значащих битов значения счета по порядку картинки *i*-th долгосрочной опорной картинки, которая включена в набор долгосрочных опорных картинок текущей картинки.

[0114] В этой второй примерной реализации переменная `DeltaPocMSBCycleLt[i]` может быть выведена следующим образом, где (#-#) относится к номеру уравнения WD7, и подчеркнутый текст представляет изменение относительно WD7:

if(i == 0)

DeltaPocMSBCycleLt[i] = delta_poc_msb_cycle_lt[i];

else

(7-37)

DeltaPocMSBCycleLt[i] = delta_poc_msb_cycle_lt[i] +

DeltaPocMSBCycleLt[i - 1];

[0115] Декодер 30 может выполнить следующий процесс декодирования для набора опорных картинок. Видеокодер 20 может быть сконфигурирован, чтобы выполнить по существу аналогичный процесс, в то же время кодируя видеоданные, чтобы сгенерировать набор опорных картинок. Этот процесс вызывается однократно для каждой картинки после декодирования заголовка вырезки, но до декодирования любой единицы кодирования и до процесса декодирования для конструирования списка опорных картинок вырезки, как определено в подпункте 8.3.3 в WD7. Процесс может привести к маркировке одной или более опорных картинок как «неиспользуемая для

ССЫЛКИ».

ПРИМЕЧАНИЕ 1 - Набор опорных картинок является абсолютным описанием опорных картинок, используемых в процессе декодирования текущей и будущей закодированных картинок. Сигнализация набора опорных картинок является явной в смысле, что все опорные картинки, включенные в набор опорных картинок, перечисляются явно и не имеется процесса конструирования набора опорных картинок по умолчанию в декодере, который зависит от статуса буфера декодированных картинок.

[0116] Опорные картинки идентифицируются их значениями PicOrderCntVal в этой второй примерной реализации.

[0117] Пять списков значений счета по порядку картинки могут быть сконструированы, чтобы вывести набор опорных картинок - PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll с NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и количество элементов NumPocLtFoll, соответственно.

- Если текущая картинка является картинкой CRA, которая является первой закодированной картинкой в потоке битов, картинкой IDR, или картинкой BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll все устанавливаются в «пусто», и NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и NumPocLtFoll все устанавливаются в 0.

- Иначе, следующее применяется для вывода пяти списков значений счета по порядку картинки и количеств записей.

```
for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS0[ StRpsIdx ][ i ] )
        PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
```

```
else
    PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
NumPocStCurrBefore = j
```

```
for( i = 0, j = 0; i < NumPositivePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS1[ StRpsIdx ][ i ] )
        PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
```

```
else
    PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
NumPocStCurrAfter = j
```

```
NumPocStFoll = k
```

(8-5)

```
for( i = 0, j = 0, k = 0; i < num_long_term_pics; i++ )
    if( used_by_curr_pic_lt_flag[ i ] )
        PocLtCurr[ j++ ] = PicOrderCntVal - DeltaPocMSBCycleLt[ i ] * MaxPicOrderCntLsb -
            pic_order_cnt_lsb + poc_lsb_lt[ i ]
```

```
else
    PocLtFoll[ k++ ] = PicOrderCntVal - DeltaPocMSBCycleLt[ i ] * MaxPicOrderCntLsb -
        pic_order_cnt_lsb + poc_lsb_lt[ i ]
```

```
NumPocLtCurr = j
```

```
NumPocLtFoll = k
```

где PicOrderCntVal - счет по порядку картинки текущей картинки, как определено в подпункте 8.2.1 в WD7.

ПРИМЕЧАНИЕ 2 - Значение `StRpsIdx` в диапазоне от 0 до `num_short_term_ref_pic_sets` - 1, включительно, в этом примере указывает, что используется набор краткосрочных опорных картинок из активного набора параметров последовательности, где `StRpsIdx` - индекс набора краткосрочных опорных картинок к списку наборов краткосрочных опорных картинок в порядке, в котором они сигнализируются в наборе параметров последовательности. `StRpsIdx`, равный `num_short_term_ref_pic_sets`, указывает, что используется набор краткосрочных опорных картинок, явно сигнализированный в заголовке вырезки.

[0118] Набор опорных картинок может включать в себя пять списков опорных картинок - `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter`, `RefPicSetStFoll`, `RefPicSetLtCurr` и `RefPicSetLtFoll`. Переменная `NumPocTotalCurr` может быть установлена равной `NumPocStCurrBefore` + `NumPocStCurrAfter` + `NumPocLtCurr`. При декодировании вырезки Р или В, WD7 требует, для согласования потока битов, чтобы значение `NumPocTotalCurr` не было равным 0.

ПРИМЕЧАНИЕ 3 - `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter` и `RefPicSetLtCurr` могут содержать все опорные картинки, которые могут быть использованы во внешнем предсказании текущей картинки, и которые могут использоваться во внешнем предсказании одной или более картинок после текущей картинки в порядке декодирования. `RefPicSetStFoll` и `RefPicSetLtFoll` могут включать в себя все опорные картинки, которые не используются во внешнем предсказании текущей картинки, но могут быть использованы во внешнем предсказании одной или более картинок после текущей картинки в порядке декодирования.

[0119] Маркировка опорной картинки может быть «неиспользуемой для ссылки», «используемой для краткосрочной ссылки», или «используемой для долгосрочной ссылки», но только одной среди этих трех в WD7. Когда опорная картинка упоминается как маркированная как «используемая для ссылки», это все вместе относится к картинке, отмечаемой как «используемая для краткосрочной ссылки», или «используемая для долгосрочной ссылки» (но не оба, в WD7). Опорная картинка, которая маркирована как «используемая для краткосрочной ссылки», упоминается как краткосрочная опорная картинка. Опорная картинка, которая маркирована как «используемая для долгосрочной ссылки», упоминается как долгосрочная опорная картинка.

[0120] Процесс выведения для набора опорных картинок и маркировки картинок могут быть выполнены для второй примерной реализации таким же образом, как в первой примерной реализации.

[0121] Третья примерная реализация обеспечивает упрощенный способ сигнализации LTRP, наряду с ограничением, что картинки, следующие после картинки CRA или BLA в порядке декодирования, запрещаются для использования картинок, предшествующих картинкам CRA или BLA в порядке декодирования в качестве картинок LTRP. Следовательно, начальные картинки запрещаются для ссылки на картинки, которые предшествуют ассоциированной картинке CRA или BLA в порядке декодирования в качестве картинок LTRP. Для не-TFD картинок количество битов, требуемых для сигнализации LSB LTRP, может быть определено посредством картинки с наименьшим POC, которая не маркирована как TFD, и которая предшествует в порядке декодирования текущей картинке, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования включая предыдущую картинку RAP в порядке декодирования.

[0122] Для картинок TFD количество битов, требуемых для сигнализации LSB LTRP, определяется картинкой с наименьшим POC, которая предшествует, в порядке

декодирования, текущей картинке, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования включая предыдущую картинку RAP в порядке декодирования. Структура синтаксиса основана на Wang и др., “Wang et al., “On Reference Picture Set,” JCTVC-I0342, contribution to JCTVC, 9th Meeting: Geneva, Switzerland, 27 апреля - 7 мая 2012, которая на 5 февраля 2013 доступна по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I0342-v1.zip, которая тем самым включена по ссылке, где сигнализируется длина LSB LTRP, с непосредственно последующими битами LSB. Изменения в семантике и процессе декодирования набора опорных картинок относительно HEVC WD7, подчеркнуты ниже в описании этой третьей примерной реализации. Кроме того, должно быть понятно, что способы третьей примерной реализации могут быть объединены со способами первой и/или второй примерных реализаций в любой комбинации.

[0123] Таблица 8 обеспечивает примерный набор синтаксиса для заголовка вырезки в соответствии с этой третьей примерной реализацией.

ТАБЛИЦА 8	
slice_header() {	Дескриптор
...	
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i=0; i<num_long_term_pics; i++) {	
poc_lsb_len_delta[i]	ue(v)
poc_lsb_lt[i]	u(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
...	
}	

[0124] В примере из Таблицы 8 заголовок вырезки включает в себя дополнительные элементы синтаксиса poc_lsb_len_delta[i] и poc_lsb_lt[i]. Семантика для этих элементов синтаксиса описана ниже, и семантика для некоторых элементов синтаксиса, которые изменены относительно WD7, также описаны ниже. Семантика для других элементов синтаксиса может остаться той же самой как в WD7, если не указано иначе.

[0125] В примере из Таблицы 8 poc_lsb_len_delta[i] может быть использован для определения количества битов, чтобы представить poc_lsb_lt[i].

[0126] Пусть CurrPicOrderCntVal будет PicOrderCntVal текущей картинки. Переменные PrevLtRefPicPoc и PrevLtRefPicPocForTFD выводятся следующим образом.

```

if( RapPicFlag || ( ( CurrPicOrderCntVal < PrevLtRefPicPoc ) && ( nal_unit_type != 2 ) ) )
    PrevLtRefPicPoc = CurrPicOrderCntVal
if( RapPicFlag || ( CurrPicOrderCntVal < PrevLtRefPicPoc ) )
    PrevLtRefPicPocForTFD = CurrPicOrderCntVal

```

[0127] Переменная PocLsbLtLen может быть выведена следующим образом.

```

if( nal_unit_type == 2 )
    MaxPocLsbLtLen = Log2( Ceil( CurrPicOrderCntVal - PrevLtRefPicPocForTFD ) )
else
    MaxPocLsbLtLen = Log2( Ceil( CurrPicOrderCntVal - PrevLtRefPicPoc ) )

```

[0128] Переменная PocLsbLtLen[i] может быть установлена равной MaxPocLsbLtLen.

[0129] Альтернативно, переменная PocLsbLtLen[i] может быть выведена следующим образом.

```

if( i == 0 )
    PocLsbLtLen[ i ] = log2_max_pic_order_cnt_lsb_minus4 + 4 + poc_lsb_len_delta[ i ]
else

```

```

    PocLsbLtLen[ i ] = PocLsbLtLen[ i - 1 ] + poc_lsb_len_delta[ i ]

```

[0130] Значение PocLsbLtLen[i] может быть в диапазоне log2_max_pic_order_cnt_lsb_minus4 + 4 до MaxPocLsbLtLen.

[0131] Альтернативно, элемент синтаксиса не должен быть сигнализирован, и переменная PocLsbLtLen[i] может быть установлена равной MaxPocLsbLtLen.

[0132] В примере из Таблицы 8 poc_lsb_lt[i] может задавать младшие значащие биты значения счета по порядку картинки i-th долгосрочной опорной картинки, которая включена в набор долгосрочных опорных картинок текущей картинки. poc_lsb_lt[i] должен быть в диапазоне от 0 до (1 << PocLsbLtLen[i]) - 1, включительно. Длина poc_lsb_lt[i] - PocLsbLtLen[i].

[0133] Видеодекодер 30 может выполнить следующий процесс декодирования для набора опорных картинок в соответствии с этой третьей примерной реализацией. Видеокодер 20 может быть сконфигурирован, чтобы выполнить по существу аналогичный процесс, в то же время кодируя видеоданные, чтобы генерировать набор опорных картинок. Этот процесс может быть вызван однократно для каждой картинки после декодирования заголовка вырезки, но до декодирования любой единицы кодирования и до процесса декодирования, для конструирования списка опорных картинок вырезки, как определено в подпункте 8.3.3 в WD7. Процесс может привести к маркировке одной или более опорных картинок как «неиспользуемая для ссылки».

ПРИМЕЧАНИЕ 1 - Набор опорных картинок может быть абсолютным описанием опорных картинок, используемых в процессе декодирования текущей и будущей закодированной картинок. Сигнализация набора опорных картинок может быть явной в том смысле, что все опорные картинки, включенные в набор опорных картинок, перечисляются явно и не имеется процесса конструирования набора опорных картинок по умолчанию в декодере, который зависит от статуса буфера декодированных картинок.

[0134] Краткосрочные опорные картинки могут быть идентифицированы их значениями PicOrderCntVal. Долгосрочные опорные картинки могут быть идентифицированы младшими значащими битами их значений PicOrderCntVal.

[0135] Пять списков значений счета по порядку картинки или младшие значащие биты значений счета по порядку картинки могут быть сконструированы, чтобы вывести набор опорных картинок - PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll с NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и количество элементов NumPocLtFoll, соответственно.

- Если текущая картинка является картинкой CRA, которая является первой закодированной картинкой в потоке битов, картинкой IDR, или картинкой BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll могут все установлены в «пусто», и NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и NumPocLtFoll могут все быть установлены в 0.

- Иначе, следующее может применяться для вывода пяти списков значений счета по порядку картинки и количеств записей.

```

for( i = 0, j = 0, k = 0; i < NumNegativePics[ StRpsIdx ]; i++ )
    if( UsedByCurrPicS0[ StRpsIdx ][ i ] )

```

```

        PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ StRpsIdx ][ i ]
5   NumPocStCurrBefore = j

    for( i = 0, j = 0; i < NumPositivePics[ StRpsIdx ]; i++ )
        if( UsedByCurrPicS1[ StRpsIdx ][ i ] )
            PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
10   else
            PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ StRpsIdx ][ i ]
        NumPocStCurrAfter = j
        NumPocStFoll = k
(8-5)

15   for( i = 0, j = 0, k = 0; i < num_long_term_pics; i++ )
        if( used_by_curr_pic_lt_flag[ i ] ) {
            PocLtCurr[ j ] = poc_lsb_lt[ i ]
            PocLsbLenCurr[ j ] = PocLsbLtLen[ i ]
20   j++
        }
        else {
            PocLtFoll[ k ] = poc_lsb_lt[ i ]
            PocLsbLenFoll[ k ] = PocLsbLtLen[ i ]
25   k++
        }
        NumPocLtCurr = j
        NumPocLtFoll = k

```

30 где PicOrderCntVal - счет по порядку картинки текущей картинки, как определено в подпункте 8.2.1 в WD7.

ПРИМЕЧАНИЕ 2 - Значение StRpsIdx в диапазоне от 0 до num_short_term_ref_pic_sets - 1, включительно, может указывать, что используется набор краткосрочных опорных картинок из активного набора параметров последовательности, где StRpsIdx - индекс набора краткосрочных опорных картинок к списку наборов краткосрочных опорных картинок в порядке, в котором они сигнализируются в наборе параметров последовательности. StRpsIdx, равный num_short_term_ref_pic_sets, может указывать, что используется набор краткосрочных опорных картинок, который явно сигнализируется в заголовке вырезки.

40 [0136] Набор опорных картинок может включать в себя пять списков опорных картинок - RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr и RefPicSetLtFoll. Переменная NumPocTotalCurr устанавливается равной NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. При декодировании вырезки Р или В, WD7 требует, для согласования потока битов, что значение NumPocTotalCurr не должно быть равным 0.

45 ПРИМЕЧАНИЕ 3 - RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr могут содержать все опорные картинки, которые могут быть использованы во внешнем предсказании текущей картинки, и которые могут использоваться во внешнем

предсказании одной или более картинок после текущей картинки в порядке декодирования. RefPicSetStFoll и RefPicSetLtFoll могут включать в себя все опорные картинки, которые не используются во внешнем предсказании текущей картинки, но могут быть использованы во внешнем предсказании одной или более картинок после текущей картинки в порядке декодирования.

[0137] Маркировка опорной картинки может быть «неиспользуемая для ссылки», «используемая для краткосрочной ссылки», или «используемая для долгосрочной ссылки», но только одной среди этих трех в WD7. Когда опорная картинка упоминается как маркированная как «используемая для ссылки», это все вместе относится к картинке, отмечаемой как «используемая для краткосрочной ссылки», или «используемая для долгосрочной ссылки» (но не обе в WD7). Опорная картинка, которая маркирована как «используемая для краткосрочной ссылки», упоминается как краткосрочная опорная картинка. Опорная картинка, которая маркирована как «используемая для долгосрочной ссылки», упоминается как долгосрочная опорная картинка.

[0138] Процесс выведения для набора опорных картинок и маркировка картинок выполняются согласно следующим упорядоченным этапам, где DPB относится к буферу декодированных картинок, как описано в Приложении C WD7:

1. Следующее применяется:

```
for(i=0; i<NumPocLtCurr; i++) {
```

```
    if(имеется долгосрочная опорная картинка picX в DPB
    с PicOrderCntVal для которой GetLSB(PicOrderCntVal, PocLsbLenCurr[i]) равно PocLtCurr
    [i])
```

```
        RefPicSetLtCurr[i]=picX
```

```
    else if(имеется краткосрочная опорная картинка picY в DPB
```

```
    с PicOrderCntVal для которой GetLSB(PicOrderCntVal, PocLsbLenCurr[i]) равно PocLtCurr
    [i])
```

```
        RefPicSetLtCurr[i]=picY
```

```
    else
```

```
        RefPicSetLtCurr[i]=«нет опорной картинки»
```

```
    }
```

(8-6)

```
for(i=0; i<NumPocLtFoll; i++) {
```

```
    if( имеется долгосрочная опорная картинка picX в DPB
    с PicOrderCntVal для которой GetLSB(PicOrderCntVal, PocLsbLenFoll[i]) равно PocLtFoll
    [i])
```

```
        RefPicSetLtFoll[i]=picX
```

```
    else if( имеется краткосрочная опорная картинка picY в DPB
```

```
    с PicOrderCntVal для которой GetLSB(PicOrderCntVal, PocLsbLenFoll[i]) равно PocLtFoll
    [i])
```

```
        RefPicSetLtFoll[i]=picY
```

```
    else
```

```
        RefPicSetLtFoll[i]=«нет опорной картинки»
```

```
    }
```

2. Все опорные картинки, включенные в RefPicSetLtCurr и RefPicSetLtFoll, маркированы как «используемая для долгосрочной ссылки»

3. Следующее применяется:

```
for(i=0; i<NumPocStCurrBefore; i++)
```

```
    if(имеется краткосрочная опорная картинка picX в DPB
    с PicOrderCntVal равным PocStCurrBefore[i])
```

```

RefPicSetStCurrBefore[i]=picX
else
RefPicSetStCurrBefore[i]=«нет опорной картинки»
for(i=0; i<NumPocStCurrAfter; i++)
5 if( имеется краткосрочная опорная картинка picX в DPB
с PicOrderCntVal равным PocStCurrAfter[i])
RefPicSetStCurrAfter[i]=picX
else
RefPicSetStCurrAfter[i]=«нет опорной картинки» (8-7)
10 for(i=0; i<NumPocStFoll; i++)
if(имеется краткосрочная опорная картинка picX в DPB
с PicOrderCntVal равным PocStFoll[i])
RefPicSetStFoll[i]=picX
else
15 RefPicSetStFoll[i]=«нет опорной картинки»

```

4. Все опорные картинки, включенные в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetStFoll, маркируются как «используемая для краткосрочной ссылки».

5. Все опорные картинки в буфере декодированных картинок, которые не включены в RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, маркируются как «неиспользуемые для ссылки».

ПРИМЕЧАНИЕ 4 - Могут быть одна или более опорных картинок, которые включены в набор опорных картинок, но не присутствуют в буфере декодированных картинок. Записи в RefPicSetStFoll или RefPicSetLtFoll, которые не равны «нет опорной картинки», должны быть проигнорированы в соответствии с WD7. Если только любое из следующих двух условий не является верным, неумышленная потеря картинок может быть логически выведена для каждой записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr, которая не равна «нет опорной картинки»: а) первая закодированная картинка в потоке битов является картинкой CRA и текущая закодированная картинка, является картинкой TFD, ассоциированной с первой закодированной картинкой в потоке битов; б) предыдущая картинка RAP, предшествующая текущей закодированной картинке в порядке декодирования, является картинкой BLA, и текущая закодированная картинка является картинкой TFD, ассоциированной с картинкой BLA.

[0139] WD7 требует, для согласования потока битов, чтобы набор опорных картинок был ограничен следующим образом, как пересмотрено этой третьей примерной реализацией:

- Не должно быть опорной картинки с temporal_id большим, чем таковой текущей картинки, включенной в RefPicSetStCurrBefore, RefPicSetStCurrAfter и RefPicSetLtCurr.
- Когда текущая картинка является картинкой TLA, не должно быть опорной картинки, включенной в набор опорных картинок с temporal_id большим чем или равным temporal_id текущей картинки.
- Не должно быть опорной картинки, включенной в набор опорных картинок, которая предшествует, в порядок вывода, любой картинке RAP, которая предшествует текущей картинке и в порядке декодирования и в порядке вывода.
- Если только любое одно из следующих условий не будет верно, не должно быть записи в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetLtCurr, которая не равна «нет опорной картинки»: а) первая закодированная картинка в потоке битов является картинкой CRA, и текущая закодированная картинка является картинкой TFD, ассоциированной с первой закодированной картинкой в потоке битов; б) предыдущая

картинка RAP, предшествующая текущей закодированной картинке в порядке декодирования, является картинкой BLA, и текущая закодированная картинка является картинкой TFD, ассоциированной с картинкой BLA.

ПРИМЕЧАНИЕ 5 - опорная картинка не может быть включена в больше, чем один из пяти списков набора опорных картинок.

- Для каждого значения i в диапазоне от 0 до NumPocLtCurr - 1, включительно, должна быть не больше, чем одна опорная картинка в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenCurr[i]) равен PocLtCurr [i]. Для каждого значения i в диапазоне от 0 до NumPocLtFoll - 1, включительно, не должно быть больше, чем одной опорной картинки в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenFoll[i]) равен PocLtFoll [i].

- Пусть prevRapPic является предыдущей картинкой RAP в порядке декодирования. Когда текущая картинка является картинкой RAP или предшествует, в порядке вывода, prevRapPic, не должно быть опорной картинки, включенной в или RefPicSetLtCurr или RefPicSetLtFoll, которая предшествует, в порядок декодирования, prevRapPic.

[0140] Четвертая примерная реализация обеспечивает способ сигнализации LTRP на основании структуры синтаксиса в третьей примерной реализации, описанной выше, где длина LSB LTRP сигнализируется непосредственно с последующими битами LSB. Ограничение на начальные картинки, не ссылающиеся на картинки перед ассоциированным RAP, удалено в этой четвертой примерной реализации. Картинкам TFD, ассоциированным с картинкой RAP, запрещено следовать за, в порядке декодирования, любой картинкой DLP, ассоциированной с одной и той же картинкой RAP, в этой четвертой примерной реализации. Для не-TFD картинок количество битов, требуемых для сигнализации LSB LTRP, может быть определено с помощью картинки с наименьшим POC, которая не маркирована как картинка TFD, и которая предшествует текущей картинке, в порядке декодирования, и следует за, в порядке декодирования, предыдущей картинкой RAP в порядке декодирования, включая предыдущую картинку RAP в порядке декодирования. Для картинок TFD количество битов, требуемых для сигнализации LSB LTRP, может быть определено с помощью картинки с наименьшим POC, которая предшествует, в порядке декодирования, текущей картинке, и следует за, в порядке вывода, RAP перед предыдущей картинкой RAP в порядке декодирования включительно RAP перед предыдущей картинкой RAP в порядке декодирования. Только изменения к семантике и процессу выведения RPS из третьей примерной реализации представлены ниже. Должно быть понятно, что способы четвертой примерной реализации могут быть объединены с любым из способов первой, второй, и/или третьей примерных реализаций в любой комбинации. Подчеркнутый текст представляет изменения относительно третьей примерной реализации, описанной выше.

[0141] В этой четвертой примерной реализации poc_lsb_len_delta[i] может быть использован для определения количества битов, чтобы представить poc_lsb_lt[i].

[0142] Пусть CurrPicOrderCntVal будет PicOrderCntVal текущей картинки. Пусть PrevRapPicPoc будет PicOrderCntVal картинки RAP, которая предшествует, в порядке декодирования, текущей картинке. Переменные PrevPrevLtRefPicPoc и PrevLtRefPicPoc вывятся следующим образом.

if(RapPicFlag)

PrevPrevLtRefPicPoc = PrevRapPicPoc

if(RapPicFlag || ((CurrPicOrderCntVal < PrevLtRefPicPoc) && (nal_unit_type != 2)))

PrevLtRefPicPoc = CurrPicOrderCntVal

[0143] Переменная MaxPocLsbLtLen может быть выведена следующим образом.

if((nal_unit_type == 4) || (nal_unit_type == 5) || (nal_unit_type == 2))

MaxPocLsbLtLen = Log2(Ceil(CurrPicOrderCntVal – PrevPrevLtRefPicPoc))

else

MaxPocLsbLtLen = Log2(Ceil(CurrPicOrderCntVal – PrevLtRefPicPoc))

[0144] Процесс выведения для PocLsbLtLen[i] может быть выполнен таким же образом как в третьей примерной реализации, как описано выше.

[0145] Видеодекодер 30 может выполнить следующий процесс декодирования для набора опорных картинок. Видеодекодер 20 может быть сконфигурирован, чтобы выполнить по существу аналогичный процесс, в то же время кодируя видеоданные, чтобы генерировать набор опорных картинок. Этот процесс может быть вызван однократно для каждой картинки, после декодирования заголовка вырезки, но до декодирования любой единицы кодирования и до процесса декодирования для конструирования списка опорных картинок вырезки, как определено в подпункте 8.3.3 в WD7. Процесс может привести к маркировке одной или более опорных картинок как «неиспользуемая для ссылки».

ПРИМЕЧАНИЕ 1 - Набор опорных картинок может быть абсолютным описанием опорных картинок, используемых в процессе декодирования текущей и будущей закодированной картинок. Сигнализация набора опорных картинок является явной в том смысле, что все опорные картинки, включенные в набор опорных картинок, перечисляются явно и не имеется процесса конструирования набора опорных картинок по умолчанию в декодере, который зависит от статуса буфера декодированных картинок.

[0146] В этой четвертой примерной реализации опорные картинки идентифицируются их значениями PicOrderCntVal.

[0147] Пять списков значений счета по порядку картинки могут быть сконструированы, чтобы вывести набор опорных картинок - PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll с NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и количество элементов NumPocLtFoll, соответственно.

[0148] Многоточие ниже представляет текст, который может остаться таким же как в третьей примерной реализации и/или в текущей версии WD7:

- Если текущая картинка является картинкой CRA, которая является первой закодированной картинкой в потоке битов, картинкой IDR, или картинкой BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll все устанавливаются в «пусто», и NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и NumPocLtFoll все устанавливаются в 0.

- ...

- Для каждого значения i в диапазоне от 0 до NumPocLtCurr - 1, включительно, должна быть не больше, чем одна опорная картинка в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenCurr[i]) равен PocLtCurr [i]. Для каждого значения i в диапазоне от 0 до NumPocLtFoll - 1, включительно, не должно быть больше одной опорной картинки в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет

PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenFoll[i]) равен PocLtFoll[i].

- Любая картинка TFD, ассоциированная с картинкой CRA или BLA, должна предшествовать любой картинке, которая не является маркированной TFD, предшествует, в порядке вывода, картинке CRA или BLA и следует за, в порядке декодирования, картинкой CRA или BLA.

[0149] Пятая примерная реализация также в целом основана на третьей примерной реализации. Эта пятая примерная реализация раскрывает способ, который не разрешает начальным картинкам, ассоциированным с картинкой CRA или BLA, ссылаться на ассоциированную картинку CRA или BLA, или другие начальные картинки, ассоциированные с той же самой картинкой CRA или BLA, в качестве картинок LTRP. Ограничение в третьей примерной реализации в отношении начальных картинок, не ссылающихся на картинки перед ассоциированным RAP в качестве картинок LTRP, удалено в этой пятой примерной реализации. Количество битов, требуемых для сигнализации LSB LTRP, может быть определено с помощью картинки RAP, которая предшествует, в порядке вывода, текущей картинке в этой пятой примерной реализации. Кроме того, должно быть понятно, что способы этой пятой примерной реализации могут быть объединены с любым из способов первой, второй, третьей и/или четвертой примерных реализаций в любой комбинации.

[0150] В этой пятой примерной реализации семантика заголовка вырезки третьей примерной реализации может быть изменена, как обозначено подчеркнутым текстом ниже:

[0151] Пусть CurrPicOrderCntVal будет PicOrderCntVal текущей картинки. Пусть PrevRapPicPoc будет PicOrderCntVal картинки RAP, которая предшествует, в порядке вывода, текущей картинке.

Переменная MaxPocLsbLtLen может быть выведена следующим образом.

$$\text{MaxPocLsbLtLen} = \text{Log2}(\text{Ceil}(\text{CurrPicOrderCntVal} - \text{PrevRapPicPoc}))$$

[0152] Процесс выведения для PocLsbLtLen[i] может быть выполнен таким же образом как в третьей примерной реализации.

[0153] Видеодекодер 30 может выполнить следующий процесс декодирования для набора опорных картинок. Видеодекодер 20 может быть сконфигурирован, чтобы выполнить по существу аналогичный процесс, в то же время кодируя видеоданные, чтобы генерировать набор опорных картинок. Этот процесс может быть вызван однократно для каждой картинки, после декодирования заголовка вырезки, но до декодирования любой единицы кодирования и до процесса декодирования, для конструирования списка опорных картинок вырезки как определено в подпункте 8.3.3 в WD7. Процесс может привести к маркировке одной или более опорных картинок как «неиспользуемая для ссылки».

ПРИМЕЧАНИЕ 1 - Набор опорных картинок может быть абсолютным описанием опорных картинок, используемых в процессе декодирования текущей и будущей закодированной картинок. Сигнализация набора опорных картинок является явной в том смысле, что все опорные картинки, включенные в набор опорных картинок, перечисляются явно и не имеется процесса конструирования набора опорных картинок по умолчанию в декодере, который зависит от статуса буфера декодированных картинок.

[0154] В этой пятой примерной реализации опорные картинки идентифицируются их значениями PicOrderCntVal.

[0155] Пять списков значений счета по порядку картинки могут быть сконструированы, чтобы вывести набор опорных картинок - PocStCurrBefore,

PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll с NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и количество элементов NumPocLtFoll, соответственно.

[0156] Многоточие ниже представляет, что текст может остаться таким же как в третьей примерной реализации и/или в текущей версии WD7:

5 - Если текущая картинка является картинкой CRA, которая является первой закодированной картинкой в потоке битов, картинкой IDR, или картинкой BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, и PocLtFoll все устанавливаются в «пусто», и NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr, и NumPocLtFoll все устанавливаются в 0.

10 - ...

- Для каждого значения i в диапазоне от 0 до NumPocLtCurr - 1, включительно, должна быть не больше, чем одна опорная картинка в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenCurr[i]) равен PocLtCurr [i]. Для каждого значения i в диапазоне от 0 до NumPocLtFoll - 1, включительно, не должно быть больше одной опорной картинки в DPB, которая не включена в RefPicSetStCurrBefore, RefPicSetStCurrAfter или RefPicSetStFoll, и которая имеет PicOrderCntVal, для которого GetLSB (PicOrderCntVal, PocLsbLenFoll[i]) равен PocLtFoll [i].

20 - Пусть prevRapPic будет предыдущей картинкой RAP в порядке декодирования. Когда текущая картинка предшествует, в порядке вывода, prevRapPic, не должно быть опорной картинки, включенной или в RefPicSetLtCurr или в RefPicSetLtFoll, которая является или prevRapPic, или любой картинкой, которая предшествует, в порядке вывода, prevRapPic, но следует за prevRapPic, в порядке декодирования.

25 [0157] Видеокодер 20 и видеodeкодер 30 каждый может быть реализован как любое из множества подходящих схем кодера или декодера, как является подходящим, таким как один или более микропроцессоров, цифровых сигнальных процессоров (DSP), специализированных интегральных схем (ASICs), программируемых пользователем вентильных матриц (FPGAs), дискретных логических схем, программное обеспечение, аппаратное обеспечение, программно-аппаратные средства или любые их комбинации. Каждый видеокодер 20 и видеodeкодер 30 может быть включен в один или более кодеров или декодеров, любой из которых может интегрироваться как часть объединенного видеокодера/декодера (кодек). Устройство, включающее в себя видеокодер 20 и/или видеodeкодер 30, может содержать интегральную схему, микропроцессор, и/или устройство беспроводной связи, такое как сотовый телефон.

[0158] Фиг. 2 является блок-схемой, иллюстрирующей пример видеокодера 20, который может реализовать способы для сигнализации долгосрочных опорных картинок в заголовках вырезов. Видеокодер 20 может выполнять внутреннее и внешнее кодирование блоков видео в пределах видео вырезов. Внутреннее кодирование полагается на пространственное предсказание, чтобы уменьшить или удалить пространственную избыточность в видео в пределах заданного кадра или картинки видео. Внешнее кодирование полагается на временное предсказание, чтобы уменьшить или удалить временную избыточность в видео в пределах смежных кадров или картинок последовательности видео. Внутренний режим (I режим) может ссылаться на любой из нескольких пространственно основанных режимов кодирования. Внешние режимы, такие как однонаправленное предсказание (P режим) или bi-предсказание (B режим), могут ссылаться на любой из нескольких основанных на времени режимов кодирования.

[0159] Как показано на фиг. 2, видеокодер 20 принимает текущий блок видео в

пределах видео кадра, который должен быть закодирован. В примере на Фиг. 2 видеокодер 20 включает в себя модуль 40 выбора режима, память 64 опорных картинок, сумматор 50, модуль 52 обработки преобразования, модуль 54 квантования, и модуль 56 энтропийного кодирования. Модуль 40 выбора режима, в свою очередь, включает в себя модуль 44 компенсации движения, модуль 42 оценки движения, модуль 46 обработки внутреннего предсказания, и модуль 48 разделения. Для реконструкции блока видео, видеокодер 20 также включает в себя модуль 58 обратного квантования, модуль 60 обработки обратного преобразования, и сумматор 62. Фильтр удаления блочности (не показан на фиг. 2) может также быть включен, чтобы фильтровать границы блока, чтобы удалить артефакты блочности из восстановленного видео. Если желательно, фильтр удаления блочности может типично фильтровать вывод сумматора 62. Дополнительные фильтры (в контуре или после контура) могут также использоваться в дополнение к фильтру удаления блочности. Такие фильтры не показаны для краткости, но если желательно, могут фильтровать выходной сигнал сумматора 50 (как фильтр в контуре).

[0160] Во время процесса кодирования видеокодер 20 принимает видеокадр или вырезку, которая должна быть закодирована. Кадр или вырезка могут быть разделены на множественные блоки видео. Модуль 42 оценки движения и модуль 44 компенсации движения выполняют внешнее предсказывающее кодирование принятого блока видео относительно одного или более блоков в одном или более опорных кадрах, чтобы обеспечить временное предсказание. Модуль 46 обработки внутреннего предсказания может альтернативно выполнить внутреннее предсказывающее кодирование принятого блока видео относительно одного или более соседних блоков в том же самом кадре или вырезке в качестве блока, который должен быть закодирован, чтобы обеспечить пространственное предсказание. Видеокодер 20 может выполнить множественные проходы кодирования, например, чтобы выбрать соответствующий режим кодирования для каждого блока видеоданных.

[0161] Кроме того, модуль 48 разделения может разделить блоки видеоданных на суб-блоки, на основании оценки предыдущих схем разделения в предыдущих проходах кодирования. Например, модуль 48 разделения может первоначально разделить кадр или вырезку на единицы LCU, и разделить каждую из единиц LCU на единицы суб-CU, на основании анализа «скорость передачи - искажение» (например, оптимизации «скорость передачи - искажение»). Модуль 40 выбора режима может далее сформировать структуру данных квадродрева, указывающую разделения LCU в единицы суб-CU. Листовые единицы CU квадродрева могут включать в себя одну или более единиц PU и один или более единиц TU.

[0162] Модуль 40 выбора режима может выбрать один из режимов кодирования, внутренний или внешний, например, на основании результатов ошибки, и выдавать результирующий внутренний или внешне закодированный блок к сумматору 50, чтобы генерировать остаточные данные блока, и к сумматору 62, чтобы восстановить закодированный блок для использования в качестве опорного кадра. Модуль 40 выбора режима также обеспечивает элементы синтаксиса, такие как вектора движения, индикаторы внутреннего режима, информацию разделения, и другую такую информацию синтаксиса, к модулю 56 энтропийного кодирования.

[0163] Модуль 42 оценки движения и модуль 44 компенсации движения могут быть высоко интегрированными, но иллюстрированы отдельно в концептуальных целях. Оценка движения, выполненная модулем 42 оценки движения, является процессом генерирования векторов движения, которые оценивают движение для блоков видео.

Вектор движения, например, может указывать смещение PU блока видео в пределах текущего видео кадра или картинки относительно предсказывающего блока в пределах опорного кадра (или другой закодированной единицы) относительно текущего блока, закодированного в пределах текущего кадра (или другой закодированной единицы).

- 5 Предсказывающий блок является блоком который обнаружен как близко соответствующий блоку, который должен быть закодирован, в терминах пиксельной разности, которая может быть определена суммой абсолютных разностей (SAD), суммой разностей квадратов (SSD), или других метрик различия. В некоторых примерах видеокодер 20 может вычислить значения для суб-целочисленных пиксельных позиций
- 10 опорных картинок, сохраненных в памяти 64 опорных картинок. Например, видеокодер 20 может интерполировать значения пиксельных позиций с одной четвертью, пиксельных позиций с одной восьмой, или другими фракционными пиксельными позициями опорной картинки. Поэтому модуль 42 оценки движения может выполнить поиск движения относительно полных пиксельных позиций и фракционных пиксельных позиций и
- 15 вывести вектор движения с фракционной пиксельной точностью.

[0164] Модуль 42 оценки движения вычисляет вектор движения для PU блока видео во внешне кодированной вырезке, сравнивая позицию PU с позицией предсказывающего блока опорной картинки. Опорная картинка может быть выбрана из первого списка опорных картинок (Список 0) или второго списка опорных картинок (Список 1), каждый

20 из которых идентифицирует одну или более опорных картинок, сохраненных в памяти 64 опорных картинок. Модуль 42 оценки движения посылает вычисленный вектор движения в модуль 56 энтропийного кодирования и модуль 44 компенсации движения.

[0165] Компенсация движения, выполненная модулем 44 компенсации движения, может вовлечь в себя установку или генерирование предсказывающего блока на

25 основании вектора движения, определенного модулем 42 оценки движения. Снова, модуль 42 оценки движения и модуль 44 компенсации движения могут быть функционально интегрированными в некоторых примерах. После приема вектора движения для PU текущего блока видео модуль 44 компенсации движения может задавать местонахождение предсказывающего блока, на который вектор движения указывает

30 в одном из списков опорных картинок. Сумматор 50 формирует остаточный блок видео, вычитая пиксельные значения предсказывающего блока из пиксельных значений текущего закодированного блока видео, формируя значения пиксельной разности, как описано ниже. Обычно модуль 42 оценки движения выполняет оценку движения относительно компонентов яркости, и модуль 44 компенсации движения использует

35 вектора движения, вычисленные на основании компонентов яркости как для компонентов насыщенности цвета так и для компонентов яркости. Модуль 40 выбора режима может также генерировать элементы синтаксиса, ассоциированные с блоками видео и видео вырезкой для использования видеодекодером 30 при декодировании блоков видео видео вырезки.

40 [0166] Модуль 40 выбора режима может реализовать ограничения настоящего раскрытия, выбирая опорные картинки из памяти 64 опорных картинок. Например, модуль 40 выбора режима режим может определять, следует ли конкретная картинка за картинкой, которая была выбрана в качестве RAP. Если это так, модуль 40 выбора режима может определять, является ли картинка начальной картинкой и следует за

45 картинкой DLP, которая ассоциирована с той же самой RAP в порядке вывода. Если это так, модуль 40 выбора режима выбирает режимы кодирования для картинки таким образом, что картинку обрабатывают как картинку DLP. Таким образом, модуль 40 выбора режима гарантирует, что текущая картинка не предсказывается ни из каких

картинок, предшествующих RAP, или любых ассоциированных картинок TFD, и таким образом, все картинки TFD предшествуют всем картинкам DLP в порядке вывода.

Другими словами, картинка, которая имеет иначе могла быть закодирована как картинка TFD, кодируется как картинка DLP, если эта картинка следует за другой картинкой DLP в порядке кодирования. Аналогично модуль 40 выбора режима может выбрать режимы кодирования, чтобы гарантировать, что все начальные картинки предшествуют в порядке декодирования всем находящимся сзади картинкам для RAP.

[0167] Модуль 46 обработки внутреннего предсказания может внутреннее предсказать текущий блок, как альтернатива внешнему предсказанию, выполненному модулем 42 оценки движения и модулем 44 компенсации движения, как описано выше. В частности, модуль 46 обработки внутреннего предсказания может задавать режим внутреннего предсказания для использования, чтобы закодировать текущий блок. В некоторых примерах модуль 46 обработки внутреннего предсказания может закодировать текущий блок, используя различные режимы внутреннего предсказания, например, во время отдельных проходов кодирования, и модуль 46 обработки внутреннего предсказания (или модуль 40 выбора режима, в некоторых примерах) может выбрать для использования соответствующий режим внутреннего предсказания из протестированных режимов.

[0168] Например, модуль 46 обработки внутреннего предсказания может вычислить значения «скорость передачи - искажение», используя анализ «скорость передачи - искажение» для различных протестированных режимов внутреннего предсказания, и выбрать режим внутреннего предсказания, имеющий лучшие характеристики «скорость передачи - искажение» среди протестированных режимов. Анализ «скорость передачи - искажение» обычно определяет величину искажения (или ошибку) между закодированным блоком и первоначальным незакодированным блоком, который был закодирован, чтобы сформировать закодированный блок, а так же скорость передачи в битах (то есть количество битов), использованных для получения закодированного блока. Модуль 46 обработки внутреннего предсказания может вычислить отношения из искажений и скоростей передачи для различных закодированных блоков, чтобы определить, какой режим внутреннего предсказания показывает лучшее значение «скорость передачи - искажение» для блока.

[0169] После выбора режима внутреннего предсказания для блока модуль 46 обработки внутреннего предсказания может предоставить информацию, указывающую выбранный режим внутреннего предсказания для блока, к модулю 56 энтропийного кодирования. Модуль 56 энтропийного кодирования может закодировать информацию, указывающую выбранный режим внутреннего предсказания. Вideoкодер 20 может включать в переданный поток битов данные конфигурации, которые могут включать в себя множество таблиц индекса режима внутреннего предсказания, и множество модифицированных таблиц индекса режима внутреннего предсказания (также называемых таблицы отображения кодовых слов), определения контекстов кодирования для различных блоков, и индикации самого вероятного режима внутреннего предсказания, таблицы индекса режима внутреннего предсказания, и модифицированную таблицу индекса режима внутреннего предсказания, чтобы использовать для каждого из контекстов.

[0170] Вideoкодер 20 формирует остаточный блок видео, вычитая данные предсказания из модуля 40 выбора режима из первоначального закодированного блока видео. Сумматор 50 представляет компонент или компоненты, которые выполняют эту операцию вычитания. Модуль 52 обработки преобразования применяет

преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально аналогичное преобразование, к остаточному блоку, формируя блок видео, содержащий остаточные значения коэффициента преобразования. Модуль 52 обработки преобразования может выполнить другие преобразования, которые концептуально аналогичны DCT. Вейвлет преобразования, целочисленные преобразования, преобразование частотных поддиапазонов или другие типы преобразования могут также использоваться. В любом случае модуль 52 обработки преобразования применяет преобразование к остаточному блоку, формируя блок остаточных коэффициентов преобразования. Преобразование может преобразовать остаточную информацию из области пиксельных значений к области преобразования, такой как частотная область. Модуль 52 обработки преобразования может послать получающиеся коэффициенты преобразования в модуль 54 квантования. Модуль 54 квантования квантует коэффициенты преобразования, чтобы далее уменьшить частоту следования битов. Процесс квантования может уменьшить битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Степень квантования может быть модифицирована, регулируя параметр квантования. В некоторых примерах модуль 54 квантования может затем выполнить сканирование матрицы, включающей в себя квантованные коэффициенты преобразования. Альтернативно, модуль 56 энтропийного кодирования может выполнить сканирование.

[0171] Вслед за квантованием модуль 56 энтропийного кодирования энтропийно кодируют квантованные коэффициенты преобразования. Например, модуль 56 энтропийного кодирования может выполнить контекстно-адаптивное кодирование с переменной длиной кода (CAVLC), контекстно-адаптивное двоичное арифметическое кодирование (CABAC), основанное на синтаксисе контекстно-адаптивное двоичное арифметическое кодирование (SBAC), энтропийное кодирование с разделением интервала вероятности (PIPE) или другой способ энтропийного кодирования. В случае основанного на контексте энтропийного кодирования контекст может быть основан на соседних блоках. После энтропийного кодирования блоком 56 энтропийного кодирования закодированный поток битов может быть передан на другое устройство (например, видеodeкодер 30) или заархивирован для более поздней передачи или поиска.

[0172] Модуль 58 обратного квантования и модуль 60 обработки обратного преобразования применяет обратное квантование и обратное преобразование, соответственно, чтобы восстановить остаточный блок в пиксельной области, например, для более позднего использования в качестве опорного блока. Модуль 44 компенсации движения может вычислить опорный блок посредством суммирования остаточного блока с предсказывающим блоком одного из кадров памяти 64 опорных картинок. Модуль 44 компенсации движения может также применить один или более фильтров интерполяции к восстановленному остаточному блоку, чтобы вычислить суб-целочисленные пиксельные значения для использования при оценке движения. Сумматор 62 суммирует восстановленный остаточный блок с блоком предсказания со скомпенсированным движением, сформированному модулем 44 компенсации движения, чтобы сформировать восстановленный блок видео для сохранения в памяти 64 опорных картинок. Восстановленный блок видео может использоваться модулем 42 оценки движения и модулем 44 компенсации движения в качестве опорного блока, чтобы внешне кодировать блок в последующем видео кадре.

[0173] Видеокодер 20 на Фиг. 2 представляют пример видеокодера, который может быть сконфигурирован, чтобы выполнить любой из способов настоящего раскрытия. Например, видеокодер 20 может быть сконфигурирован, чтобы закодировать картинку

RAP и кодировать одну или более декодируемых начальных картинок DLP для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения. Картины DLP может включать в себя одну или более картинок, имеющих значения порядка отображения раньше, чем значение порядка отображения картинки RAP, и которые не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования. Видеокодер 20 может также закодировать одну или более начальных картинок относительно картинки RAP таким образом, что все начальные картинки для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, где находящиеся сзади картинки включают в себя картинки, имеющие значения порядка отображения, которые больше, чем значение порядка отображения картинки RAP. Картинка RAP может включать в себя, например, одну из картинки CRA и картинки BLA. Любая картинка, предшествующая картинке CRA или BLA в порядке декодирования, может предшествовать любой картинке DLP, ассоциированной с картинкой CRA или картинкой BLA в порядке отображения. Видеокодер 20 может также закодировать одну или более начальных картинок, ассоциированных с картинкой RAP, где начальные картинки предшествуют картинке RAP в значении порядка оценивают и следуют за картинкой RAP в порядке декодирования, и видеокодер 20 может закодировать одну или более находящихся сзади картинок, ассоциированных с картинкой RAP, таким образом, что все начальные картинки предшествуют всем находящимся сзади картинкам в порядке декодирования, где находящиеся сзади картинки следуют за картинкой RAP и в значении порядка отображения и в порядке декодирования.

[0174] Видеокодер 20 может также закодировать одну или более тегированных для отбрасывания (TFD) картинок, ассоциированных с картинкой RAP. Одна или более картинок TFD могут предшествовать картинке RAP в значении порядка отображения и следовать за картинкой RAP в порядке декодирования и ссылаться на видеоданные ранее чем картинка RAP в порядке декодирования. Видеокодер 20 может закодировать одну или более картинок DLP для картинки RAP, кодируя одну или более картинок DLP, ассоциированных с картинкой RAP таким образом, что все картинки TFD предшествуют картинкам DLP в порядке отображения, где картинки DLP предшествуют картинке RAP в порядке отображения и следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, ранние чем картинка RAP в порядке декодирования.

[0175] Видеокодер 20 может также закодировать одну или более картинок, которые предшествуют картинке RAP в порядке декодирования. Видеокодер 20 может закодировать одну или более картинок DLP, кодируя одну или более картинок DLP, ассоциированных с картинкой RAP таким образом, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем картинкам DLP в порядке отображения. Картины DLP могут предшествовать картинке RAP в порядке отображения и следовать за картинкой RAP в порядке декодирования и не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования.

[0176] Фиг. 3 является блок-схемой, иллюстрирующей пример видеodeкодера 30, который может реализовывать способы для сигнализации долгосрочных опорных картинок в заголовках вырезов. В примере на Фиг. 3 видеodeкодер 30 включает в себя модуль 70 энтропийного декодирования, модуль 72 компенсации движения, модуль 74 обработки внутреннего предсказания, модуль 76 обратного квантования, модуль 78

обработки обратного преобразования, память 82 опорных картинок и сумматор 80. Вicodeодекодер 30 в некоторых примерах может выполнить проход декодирования, в целом обратный проходу кодирования, описанному относительно вicodeокодера 20 (Фиг. 2). Модуль 72 компенсации движения может генерировать данные предсказания на основании векторов движения, принятых от модуля 70 энтропийного декодирования, в то время как модуль 74 обработки внутреннего предсказания может генерировать данные предсказания на основании индикаторов режима внутреннего предсказания, принятых от модуля 70 энтропийного декодирования.

[0177] Во время процесса декодирования вicodeодекодер 30 принимает закодированный видео поток битов, который представляет блоки видео закодированной видео вырезки и ассоциированные элементы синтаксиса от вicodeокодера 20. Модуль 70 энтропийного декодирования из вicodeокодера 30 энтропийно декодирует поток битов, чтобы сгенерировать квантованные коэффициенты, вектора движения или индикаторы режима внутреннего предсказания, и другие элементы синтаксиса. Модуль 70 энтропийного декодирования направляет вектора движения и другие элементы синтаксиса к модулю 72 компенсации движения. Вicodeодекодер 30 может принять элементы синтаксиса на видео уровне вырезки и/или уровне блока видео.

[0178] Когда видео вырезка закодирована как внутренне кодированная (I) вырезка, модуль 74 обработки внутреннего предсказания может генерировать данные предсказания для блока видео текущей видео вырезки на основании сигнализированного режима внутреннего предсказания и данные от ранее декодированных блоков текущего кадра или картинки. Когда вicodeокадр закодирован как внешне кодированная (то есть В, Р или GРВ) вырезка, модуль 72 компенсации движения формирует предсказывающие блоки для блока видео текущей видео вырезки на основании вектора движения и других элементов синтаксиса, принятых от модуля 70 энтропийного декодирования. Предсказывающие блоки могут быть сформированы из одной из опорных картинок в пределах одного из списков опорных картинок. Вicodeодекодер 30 может построить списки опорных кадров, Список 0 и Список 1, используя способы конструирования (построения) по умолчанию, на основании опорных картинок, сохраненных в памяти 92 опорных кадров. Модуль 72 компенсации движения определяет информацию предсказания для блока видео текущей видео вырезки посредством синтаксического разбора векторов движения и других элементов синтаксиса, и использует информацию предсказания, чтобы сформировать предсказывающие блоки для текущего декодируемого блока видео. Например, модуль 72 компенсации движения использует некоторые из принятых элементов синтаксиса, чтобы определить режим предсказания (например, внутреннего или внешнего предсказания), использованный для кодирования блоков видео для видео вырезки, тип вырезки внешнего предсказания (например, В вырезка, Р вырезка, или GРВ вырезка), информацию конструирования для одного или более списков опорных картинок для вырезки, вектора движения для каждого внешне кодированного блока видео вырезки, статус внешнего предсказания для каждого внешне кодированного блока видео вырезки, и другую информацию, чтобы декодировать блоки видео в текущей вырезке видео.

[0179] Модуль 72 компенсации движения может также выполнить интерполяцию на основании фильтров интерполяции. Модуль 72 компенсации движения может использовать фильтры интерполяции, который используется вicodeокодером 20 во время кодирования блоков видео, чтобы вычислить интерполированные значения для суб-целочисленных пикселей опорных блоков. В этом случае модуль 72 компенсации движения может задавать фильтры интерполяции, используемые вicodeокодером 20, из

принятых элементов синтаксиса и использовать эти фильтры интерполяции, чтобы сформировать предсказывающие блоки.

[0180] Модуль 76 обратного квантования обратно квантует, то есть деквантует, квантованные коэффициенты преобразования, предоставленные в потоке битов и декодированные модулем 80 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра квантования QP_Y , вычисленного видеodeкодером 30 для каждого блока видео в видео вырезке, чтобы определить степень квантования и, аналогично, степень обратного квантования, которое должно быть применено.

[0181] Блок 78 обработки обратного преобразования применяет обратное преобразование, например, обратное DCT, обратное целочисленное преобразование, или концептуально аналогичный процесс обратного преобразования, к коэффициентам преобразования, чтобы сформировать остаточные блоки в пиксельной области.

[0182] После того, как модуль 72 компенсации движения генерирует предсказывающий блок для текущего блока видео на основании вектора движения и других элементов синтаксиса, видеodeкодер 30 формирует декодированный блок видео, суммируя остаточные блоки от блока 78 обработки обратного преобразования с соответствующими предсказывающими блоками, генерируемыми блоком 72 компенсации движения. Сумматор 90 представляет компонент или компоненты, которые выполняют эту операцию суммирования. Если желательно, фильтр удаления блочности может также быть применен, чтобы фильтровать декодированные блоки, чтобы удалить артефакты блочности. Другие контурные фильтры (или в контуре кодирования или после контура кодирования) могут также быть использованы для сглаживания пиксельных переходов, или иным образом улучшить качество видео. Декодированные блоки видео в заданном кадре или картинке затем сохраняются в памяти 92 опорных картинок, которая хранит опорные картинки, используемые для последующей компенсации движения. Память 82 опорных картинок также хранит декодированное видео для более позднего представления на устройстве отображения, таком как устройство 32 отображения на Фиг. 1.

[0183] Реализовывая способы настоящего раскрытия, видеodeкодер 30 может принять картинку RAP и определить, что картинка RAP используется как точка произвольного доступа для потока битов. Определение, что картинка RAP должна использоваться видеodeкодером 30 как точка произвольного доступа, может, например, быть основано на сигнализации в потоке битов. Для картинок после картинки RAP в порядке кодирования видеodeкодер может определять, являются ли картинки картинками TFD. Если картинка является картинкой TFD, то видеodeкодер 30 может синтаксически разобрать синтаксис, ассоциированный с картинкой TFD, без необходимости декодирования картинки TFD. Например, на основании типа единицы NAL видеodeкодер может идентифицировать картинки TFD и не полностью декодировать картинку TFD. Если видеodeкодер 30 определяет, что картинка является картинкой DLP, то видеodeкодер 30 может декодировать картинку.

[0184] Видеodeкодер 30 на Фиг. 3 представляют пример видеodeкодера, который может быть сконфигурирован, чтобы выполнить любой из способов настоящего раскрытия. Например, видеodeкодер 20 может быть сконфигурирован, чтобы закодировать картинку RAP и кодировать одну или более декодируемых начальных картинок DLP для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения. Картинки DLP могут включать в себя одну или более картинок, имеющих

значения порядка отображения, более раннее, чем значение порядка отображения картинки RAP, и которые не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования. Видеодекодер 30 может также закодировать одну или более начальных картинок относительно картинки RAP таким образом, что все

5 начальные картинки для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, где находящиеся сзади картинки включают в себя картинки, имеющие значения порядка отображения, которые больше, чем значение порядка отображения картинки RAP. Картинка RAP может включать в себя, например, одну из картинки CRA и картинки BLA. Любая картинка,

10 предшествующая картинке CRA или BLA в порядке декодирования, может предшествовать любой картинке DLP, ассоциированной с картинкой CRA или картинкой BLA в порядке отображения. Видеокодер 20 может также закодировать одну или более начальных картинок, ассоциированных с картинкой RAP, где начальные картинки предшествуют картинке RAP в значении порядка отображения и следуют за картинкой

15 RAP в порядке декодирования, и видеодекодер 30 может закодировать одну или более находящихся сзади картинок, ассоциированных с картинкой RAP, таким образом, что все начальные картинки предшествуют всем находящимся сзади картинкам в порядке декодирования, где находящиеся сзади картинки следуют за картинкой RAP и в значении порядка отображения и в порядке декодирования.

20 [0185] Видеодекодер 30 может также закодировать одну или более тегированных для отбрасывания (TFD) картинок, ассоциированных с картинкой RAP. Одна или более картинок TFD могут предшествовать картинке RAP в значении порядка отображения и следовать за картинкой RAP в порядке декодирования и ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования. Видеокодер 20 может

25 закодировать одну или более картинок DLP для картинки RAP посредством кодирования одной или более картинок DLP, ассоциированных с картинкой RAP таким образом, что все картинки TFD предшествуют картинкам DLP в порядке отображения, где картинки DLP предшествуют картинке RAP в порядке отображения и следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные более ранние

30 чем картинка RAP в порядке декодирования.

[0186] Видеодекодер 30 может также закодировать одну или более картинок, которые предшествуют картинке RAP в порядке декодирования. Видеодекодер 30 может закодировать одну или более картинок DLP посредством кодирования одной или более

35 картинок DLP, ассоциированных с картинкой RAP таким образом, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем картинкам DLP в порядке отображения. Картинки DLP могут предшествовать картинке RAP в порядке отображения и следовать за картинкой RAP в порядке декодирования и не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования.

40 [0187] Фиг. 4 является концептуальной диаграммой, иллюстрирующей последовательность закодированных видеокартинки 100-132. Фиг. 4 иллюстрирует отношения между точками RAP, начальными картинками, картинками TFD и находящимися сзади картинками. Картинки заштрихованы по-разному, чтобы указать позиции в пределах иерархической структуры предсказания. Например, картинки 100,

45 116 и 132 заштрихованы черным, чтобы представить, что картинки 100, 116, 132 находятся сверху иерархической структуры предсказания. Картинки 100, 116, 132 могут содержать, например, внутренне закодированные картинки или внешне закодированные картинки, которые предсказаны из других картинок в единственном направлении

(например, Р-картинок). Когда внутренне кодированы, картинки 100, 116, 132 предсказываются исключительно из данных в пределах той же самой картинки. Когда внешне кодированы, картинка 116, например, может быть закодирована относительно данных картинки 100, как обозначено пунктирной стрелкой от картинки 116 к картинке 100. Картинки 116, 132 формируют ключевые картинки групп картинок (GOP) 134, 136, соответственно.

[0188] Картинки 108, 124 заштрихованы темным, чтобы указать, что они являются следующими в иерархии кодирования после картинок 100, 116 и 132. Картинки 108, 124 могут содержать двунаправленные, предсказанные во внешнем режиме закодированные картинки. Например, картинка 108 может быть предсказана из данных картинок 100 и 116, в то время как картинка 124 может быть предсказана из картинок 116 и 132. Картинки 104, 112, 120, и 128 заштрихованы светлым, чтобы указать, что они являются следующими в иерархии кодирования после картинок 108 и 124. Картинки 104, 112, 120 и 128 могут также содержать двунаправленные, предсказанные во внешнем режиме закодированные картинки. Например, картинка 104 может быть предсказана из картинок 100 и 108, картинка 112 может быть предсказана из картинок 108 и 116, картинка 120 может быть предсказана из картинки 116 и 124, и картинка 128 может быть предсказана из картинки 124 и 132. Обычно, картинки, которые ниже в иерархии, могут быть закодированы из любых опорных картинок, которые выше в иерархии, предполагая, что опорные картинки все еще буферизованы в буфере декодированных картинок, и предполагая, что опорные картинки были закодированы ранее, чем картинка в настоящее время кодируемая.

[0189] Наконец, картинки 102, 106, 110, 114, 118, 122, 126 и 130 заштрихованы белым, чтобы указать, что эти картинки являются последними в иерархии кодирования.

Картинки 102, 106, 110, 114, 118, 122, 126 и 130 могут быть двунаправленными, предсказанными во внешнем режиме закодированными картинками. Картинка 102 может быть предсказана из картинок 100 и 104, картинка 106 может быть предсказана из картинок 104 и 108, картинка 110 может быть предсказана из картинок 108 и 112, картинка 114 может быть предсказана из картинок 112 и 116, картинка 118 может быть предсказана из картинки 116 и 120, картинка 122 может быть предсказана из картинок 120 и 124, картинка 126 может быть предсказана из картинок 124 и 128, и картинка 130 может быть предсказана из картинок 128 и 132. Снова, должно быть понятно, что картинки ниже в иерархии кодирования могут быть закодированы из других картинок, которые выше в иерархии кодирования. Например, любые из картинок 102, 106, 110 или 114 могут быть предсказаны относительно любой из картинок 100, 116, или 108, в дополнение или в альтернативе.

[0190] Картинки 100-132 иллюстрированы в порядке отображения. То есть, следуя декодированию, картинка 100 показана перед картинкой 102, картинка 102 показана перед картинкой 104, и так далее. Как описано выше, значения РОС обычно описывают порядок отображения относительно картинок, который является также по существу таким же как порядок, в котором необработанные картинки были захвачены или сгенерированы до кодирования. Однако, из-за иерархии кодирования, картинки 100-132 могут быть декодированы в различном порядке. Кроме того, будучи закодированным, картинки 100-132 могут быть размещены в порядке декодирования в потоке битов, включая закодированные данные для картинок 100-132. Например, картинка 116 может быть показана последней среди картинок GOP 134. Однако, из-за иерархии кодирования, картинка 116 может быть декодирована первой из GOP 134. Таким образом, чтобы должным образом декодировать картинку 108, например, может

быть необходимо, чтобы картинка 116 была декодирована сначала, чтобы действовать в качестве опорной картинки для картинки 108. Аналогично, картинка 108 может действовать в качестве опорной картинки для картинок 104, 106, 110 и 112, и поэтому может быть необходимо, чтобы она была декодирована перед картинками 104, 106, 110 и 112.

[0191] Кроме того, некоторые картинки можно рассматривать как долгосрочные опорные картинки, в то время как другие картинки можно рассматривать как краткосрочные опорные картинки. Предположим, например, что картинки 100 и 116 представляют долгосрочные опорные картинки, в то время как картинки 108, 104 и 112 представляют краткосрочные опорные картинки. Может иметь место, в этом примере, что картинки 102 и 106 могут быть предсказаны относительно любой из картинок 100, 116, 108 или 104, но что картинки 110 и 114 могут быть предсказаны относительно любой из картинок 100, 116, 108 или 112. Другими словами, картинка 104 могут не быть доступной для ссылки при кодировании картинок 110 и 114. В качестве другого примера, предполагая, что картинки 100 и 116 представляют долгосрочные опорные картинки и картинки 108, 104 и 112 представляют краткосрочные опорные картинки, картинки 108, 104 и 112 могут не быть доступны для ссылки при кодировании картинок 118, 122, 126 и 130. В соответствии со способами настоящего раскрытия, данные относительно долгосрочных опорных картинок могут быть сигнализированы в заголовках вырезов для вырезов для любых из картинок 100-132.

[0192] В примере на Фиг. 4 стрелки представляют потенциальные отношения предсказания, где оконечная часть каждой стрелки представляет картинку, которая может быть предсказана, и головная часть каждой стрелки представляет картинку, из которой эта картинка может быть предсказана (то есть головная часть стрелки указывает на потенциальную опорную картинку). Хотя картинка 116 имеет стрелку, указывающую на картинку 100, в целях примера, предположим, что картинка 116 является закодированной внутренним предсказанием (то есть закодированной независимо от любой другой картинки). Кроме того, предположим, что картинка 108 закодирована как Р-картинка со ссылкой на картинку 116. Кроме того, предположим, что картинка 116 используется как точка произвольного доступа, например, как картинка IDR, BLA или CRA. Таким образом, поток битов, извлеченный, начиная с картинки 116, будет включать в себя каждую из картинок 102-132 в порядке декодирования, но не будет включать в себя картинку 100.

[0193] В этом примере, согласно вышеизложенным предположениям, картинки 102, 104 и 106 представляют картинки TFD, потому что картинки 102, 104, и 106 зависят, прямо или косвенно, от картинки 100. Картинки 108-114 представляют примеры картинок DLP, потому что картинки 108-114 не зависят прямо или косвенно от картинки 100, и имеют порядок отображения, более ранний чем картинка 116, но порядок декодирования, который является более поздним, чем картинки 116. Картинки 118-132 представляют находящиеся сзади картинки в этом примере, потому что порядок отображения и порядок декодирования картинок 118-132 оба являются более поздними, чем картинка 116. Согласно способам настоящего раскрытия, при кодировании одной или более картинок DLP для картинки RAP, картинки DLP могут быть закодированы таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке отображения. Например, как описано выше относительно Фиг. 2, при кодировании картинки 108, модуль 40 выбора режима может определить кодировать картинку 108 как Р-картинку относительно, например, картинки 116. В этом способе картинка 108

может быть классифицирована как DLP. Соответственно, модуль 40 выбора режима может определить кодировать каждую из картинок 110, 112, 114 как картинки DLP также, а не как картинки TFD. Аналогично, видеodeкодер 30 после определения, что картинка 108 является DLP, определяет, что последующие картинки в порядке вывода являются декодируемыми. С другой стороны, видеodeкодер 30 может просто синтаксически разобрать, без декодирования, данные для любых картинок, которые являются TFD. В этом примере видеodeкодер 30 может определить, что картинки 102, 104 и 106 являются картинками TFD, и поэтому просто синтаксически разбирает данные для этих картинок, не пытаясь декодировать картинки. Для картинок, которые имеют порядок вывода позже, чем картинка 108, видеodeкодер 30 может начать декодирование как нормальное, не определяя, являются ли картинки картинками TFD или DLP, потому что, в соответствии с ограничениями, предложенными в настоящем раскрытии, все картинки TFD предшествуют всем картинкам DLP в порядке вывода.

[0194] Как описано выше, настоящее раскрытие предлагает несколько ограничений для поддержания произвольного доступа на основании этих различных типов картинок. Во-первых, все картинки TFD для картинки RAP должны иметь значение порядка отображения, которое является более ранним, чем значение порядка отображения для всех картинок DLP для RAP. Это ограничение может улучшить опыт зрительского просмотра, потому что оно избегает «зыби» (неравномерности) в скорости передачи кадров. Таким образом, если имеются одна или более картинок, которые не могут быть корректно декодированы (то есть картинки TFD), если эти картинки имеют значения порядка отображения, смешиваемые с картинками DLP, то скорость передачи кадров может казаться изменчивой, что может ухудшить пользовательское восприятие. Настоящее раскрытие также предлагает ограничение, в котором предотвращено перемежение начальных картинок с «находящимися сзади картинками» в порядке декодирования. Таким образом, в соответствии со способами настоящего раскрытия, видеodeкодировщики могут гарантировать, что все начальные картинки (включая как картинки TFD так и картинки DLP) имеют значения порядка декодирования более ранние, чем значения порядка декодирования картинок, которые имеют значения порядка отображения больше, чем значение порядка отображения соответствующей картинки RAP. Как отмечено выше, «находящиеся сзади картинки» являются картинками, которые следуют за картинкой RAP и в порядке декодирования и в порядке отображения.

[0195] Фиг. 5 является блок-схемой, иллюстрирующей примерный набор устройств, которые формируют часть сети 150. В этом примере сеть 150 включает в себя устройства 154А, 154В маршрутизации (устройства 154 маршрутизации) и устройство 156 транскодирования. Устройства 154 маршрутизации и устройство 156 транскодирования предназначены, чтобы представить небольшое количество устройств, которые могут явиться частью сети 150. Другие сетевые устройства, такие как коммутаторы, концентраторы, шлюзы, брандмауэры, мосты, и другие такие устройства могут также быть включены в пределы сети 150. Кроме того, дополнительные сетевые устройства могут быть предоставлены вдоль сетевого пути между серверным устройством 152 и клиентским устройством 158. Серверное устройство 152 может соответствовать исходному устройству 12 (Фиг. 1), в то время как клиентское устройство 158 может соответствовать устройству 14 назначения (Фиг. 1), в некоторых примерах.

[0196] Обычно устройства 154 маршрутизации реализуют один или более протоколов маршрутизации, чтобы обмениваться информацией сети через сеть 150. Обычно устройства 154 маршрутизации выполняют протоколы маршрутизации, чтобы

обнаружить маршруты через сеть 150. Выполняя такие протоколы маршрутизации, устройство 154B маршрутизации может обнаружить маршрут сети от себя к серверному устройству 152 через устройство 154A маршрутизации.

[0197] Фиг. 6 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы согласно Фиг. 6 ниже описаны в отношении общего кодировщика видео. Этот общий кодировщик видео может, например, соответствовать или видеокодеру 20 или видеодекодеру 30. Кодировщик видео кодирует картинку RAP (161). Картинка RAP может включать в себя одну из картинок чисто произвольного доступа и картинки доступа с разорванной ссылкой. Кодировщик видео кодирует одну или более картинок DLP для картинки RAP таким образом, что все картинки, которые предназначены для отбрасывания, предшествуют картинкам DLP, ассоциированным с картинкой RAP в порядке (162) отображения. Картинки DLP могут включать в себя одну или более картинок, имеющих значения порядка отображения, более ранние, чем значение порядка отображения картинки RAP, и которые не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования.

[0198] Фиг. 7 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы на Фиг. 7 будут описаны со ссылками на общий кодировщик видео. Этот общий кодировщик видео может, например, соответствовать или видеокодеру 20 или видеодекодеру 30. Видеокодировщик кодирует картинку RAP (171). Видеокодировщик кодирует одну или более картинок TFD ассоциированных с картинкой RAP, в которых картинки TFD предшествуют картинке RAP в порядке отображения и следуют за картинкой RAP при декодировании и ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования (172). Видеокодировщик кодирует одну или более картинок DLP, ассоциированных с картинкой RAP таким образом, что все картинки TFD предшествуют картинкам DLP в порядке отображения, при этом картинки DLP предшествуют картинке RAP в порядке отображения и предшествуют картинке RAP при декодировании и не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования (173).

[0199] Фиг. 8 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы Фиг. 8 будут описаны в отношении общего кодировщика видео. Этот общий кодировщик видео может, например, соответствовать или видеокодеру 20 или видеодекодеру 30. Видеокодировщик кодирует картинку RAP (181). Видеокодировщик кодирует одну или более начальных картинок, ассоциированных с картинкой RAP, причем начальные картинки предшествуют картинке RAP в порядке отображения и следуют за картинкой RAP в порядке декодирования (182). Видеокодировщик кодирует одну или более находящихся сзади картинок, ассоциированных с картинкой RAP, таким образом, что все начальные картинки предшествуют всем находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади картинки следуют за картинкой RAP и в порядке отображения и в порядке декодирования (183).

[0200] Фиг. 9 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы Фиг. 9 будут описаны в отношении общего кодировщика видео. Этот общий кодировщик видео может, например, соответствовать или видеокодеру 20 или видеодекодеру 30. Видеокодировщик кодирует одну или более картинок, которые предшествуют картинке RAP в порядке декодирования (191). Видеокодировщик кодирует картинку RAP (192). Видеокодировщик кодирует одну или более картинок DLP, ассоциированных с картинкой RAP таким образом, что все

картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем картинкам DLP в порядке отображения, причем картинки DLP предшествуют картинке RAP в порядке отображения и следуют за картинкой RAP при декодировании и не ссылаются на видеоданные более ранние, чем картинка RAP в

5 порядке декодирования (193).

[0201] Фиг. 10 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы Фиг. 10 могут, например, быть выполнены модулем 40 выбора режима из видеокодера 20. Модуль 40 выбора режима может закодировать картинку RAP (202). Для следующей картинки, после картинки RAP,

10 модуль 40 выбора режима может определять, закодировать ли следующую картинку как TFD или DLP (204). Если модуль 40 выбора режима кодирует следующую картинку TFD (206), то для последующей картинки модуль 40 выбора режима может еще раз определить, закодировать ли картинку как TFD или DLP (204). Как только модуль 40 выбора режима кодирует картинку, следующую после картинки RAP, как DLP (208),

15 затем модуль выбора режима кодирует последующие картинки как картинки DLP (210). В примере на Фиг. 10, “следующая картинка” обычно относится к картинке, которая является последующей в порядке вывода. Перемежение порядка декодирования картинок DLP и TFD все еще разрешено, в то время как перемежение порядка вывода может быть запрещено.

[0202] Фиг. 11 является последовательностью операций, иллюстрирующей примерный способ настоящего раскрытия. Способы согласно Фиг. 11 могут, например, быть выполнены видеодекодером 30. Видеодекодер 30 может декодировать картинку RAP и определить, что картинка должна использоваться для произвольного доступа к потоку битов. Для следующей картинки в потоке битов видеодекодер 30 может определять,

25 например, на основании типа единицы NAL для картинки, является ли эта картинка картинкой TFD или DLP (214). Если картинка является TFD, то видеодекодер 30 может пропустить декодирование картинки TFD (216). Если картинка является DLP, то видеодекодер 30 может декодировать DLP (218).

[0203] Хотя способы согласно Фиг. 6, 7, 8, 9, 10 и 11 были представлены отдельно,

30 предполагается, что аспекты способов, представленных на фиг. 6, 7, 8, 9, 10 и 11, могут быть реализованы одновременно. Должно быть также понятно, что в зависимости от примера, некоторые действия или события любого из способов, описанных здесь, могут быть выполнены в иной последовательности, могут быть добавлены, слиты или пропущены в целом (например, не все описанные действия или события необходимы

35 для реализации способов). Кроме того, в некоторых примерах действия или события могут быть выполнены одновременно, например, с помощью многопоточной обработки, обработки с прерываниями или множественных процессоров, а не последовательно.

[0204] В одном или более примерах описанные функции могут быть реализованы в аппаратном обеспечении, программном обеспечении, программно-аппаратных

40 средствах, или любой их комбинации. Если реализовано в программном обеспечении, функции могут быть сохранены на или переданы по как одна или более инструкций или код по считываемому компьютером носителю и выполнены основанным на аппаратном обеспечении блоком обработки. Считываемый компьютером носитель может включать в себя считываемый компьютером запоминающий носитель, который

45 соответствует материальному носителю, такому как запоминающие носители данных, или коммуникационные носители, включая любые носители, которые облегчают передачу компьютерной программы от одного места к другому, например, согласно протоколу связи. В этом способе считываемый компьютером носитель обычно может

соответствовать (1) материальным считываемым компьютером запоминающим носителям, которые являются невременными, или (2) коммуникационным носителям, таким как сигнал или несущая. Запоминающие носители данных могут быть любым доступным носителем, к которому могут получить доступ один или более компьютеров или один или более процессоров, чтобы извлечь инструкции, код и/или структуры данных для реализации способов, описанных в настоящем раскрытии. Компьютерный программный продукт может включать в себя считываемый компьютером носитель.

[0205] Посредством примера, а не ограничения, такие считываемые компьютером запоминающие носители могут содержать RAM, ROM, EEPROM, CD-ROM или другое запоминающее устройство на оптических дисках, запоминающее устройство на магнитных дисках, или другие магнитные запоминающие устройства хранения, флэш-память, или любой другой носитель, который может быть использован для хранения желаемого программного кода в форме инструкций или структур данных, и к которому может получить доступ компьютер. Кроме того, любое соединение должным образом называют считываемым компьютером носителем. Например, если инструкции переданы от вебсайта, сервера, или другого удаленного источника, используя коаксиальный кабель, волоконно-оптический кабель, витую пару, цифровую абонентскую линию (DSL), или беспроводные технологии, такие как инфракрасное излучение, радиоизлучение, и микроволновое излучение, то эти коаксиальный кабель, волоконно-оптический кабель, витая пара, DSL, или беспроводные технологии, такие как инфракрасное излучение, радиоизлучение, и микроволновое излучение включены в определение носителя. Нужно подразумевать, однако, что считываемые компьютером запоминающие носители и запоминающие носители данных не включают в себя соединения, несущие, сигналы, или другие временные носители, но вместо этого направлены на невременные, материальные запоминающие носители. Диск и диск, как используется здесь, включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), дискета и диск Blu-ray, где диски (disks) обычно воспроизводят данные магнитно, в то время как диски (discs) воспроизводят данные оптически с помощью лазеров. Комбинации вышеупомянутого должны также быть включены в рамки считываемого компьютером носителя.

[0206] Инструкции могут быть выполнены одним или более процессорами, такими как одним или более цифровыми сигнальными процессорами (DSPs), микропроцессорами общего назначения, специализированными интегральными схемами (ASICs), программируемыми пользователем вентильными матрицами (FPGAs), или другими эквивалентными интегрированными или дискретными логическими схемами. Соответственно, термин "процессор", как используется здесь, может относиться к любой предшествующей структуре или любой другой структуре, подходящей для реализации способов, описанных в настоящем описании. Кроме того, в некоторых аспектах функциональные возможности, описанные здесь, могут быть предоставлены в пределах специализированного аппаратного обеспечения и/или программных модулей, конфигурируемых для кодирования и декодирования, или объединены в объединенный кодек. Кроме того, способы могли быть полностью реализованы в одной или более схемах или логических элементах.

[0207] Способы настоящего раскрытия могут быть реализованы в широком разнообразии устройств или аппаратов, включая беспроводную телефонную трубку, интегральную схему (IC) или набор IC (например, микропроцессорный набор). Различные компоненты, модули, или блоки описаны в настоящем раскрытии, чтобы подчеркнуть функциональные аспекты устройств, конфигурируемых, чтобы выполнить

раскрытые способы, но не обязательно требовать реализации различными блоками аппаратного обеспечения. Вместо этого, как описано выше, различные блоки могут быть объединены в блоке аппаратного обеспечения кодека или предоставлены коллекцией взаимодействующих блоков аппаратного обеспечения, включая один или более процессоров, как описано выше, в соединении с подходящим программным обеспечением и/или программно-аппаратными средствами.

[0208] Были описаны различные примеры. Эти и другие примеры находятся в рамках нижеследующей формулы изобретения.

(57) Формула изобретения

1. Способ декодирования видеоданных, причем способ содержит:

декодирование, устройством декодирования видео, содержащим один или несколько процессоров, картинки точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

идентификацию, устройством декодирования видео, содержащим один или несколько процессоров, одной или нескольких картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

отбрасывание одной или нескольких картинок первого типа;

декодирование, устройством декодирования видео, содержащим один или несколько процессоров, одной или нескольких декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все из одной или нескольких картинок первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и

вывод, устройством декодирования видео, содержащим один или несколько процессоров, декодированной картинки RAP.

2. Способ по п. 1, дополнительно содержащий декодирование, устройством декодирования видео, содержащим один или несколько процессоров, одной или нескольких начальных картинок относительно картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

3. Способ по п. 1, в котором любая картинка, предшествующая картинке CRA или картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке вывода.

4. Способ по п. 1, причем способ дополнительно содержит:

декодирование, устройством декодирования видео, содержащим один или несколько

процессоров, одной или нескольких начальных картинок, ассоциированных с картинкой RAP, при этом начальные картинки предшествуют картинке RAP в значении порядка вывода и следуют за картинкой RAP в порядке декодирования, и

декодирование, устройством декодирования видео, содержащим один или несколько процессоров, одной или нескольких находящихся сзади картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все из начальных картинок предшествуют всем находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

5. Способ по п. 1, в котором декодирование одной или нескольких декодируемых начальных картинок для картинки RAP содержит декодирование одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

6. Способ по п. 1, причем способ дополнительно содержит:

декодирование, устройством декодирования видео, содержащим один или несколько процессоров, одной или нескольких картинок, которые предшествуют картинке RAP в порядке декодирования, и

при этом декодирование одной или нескольких декодируемых начальных картинок содержит декодирование одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

7. Способ по п. 1, в котором декодирование картинки RAP выполняется как часть процесса кодирования видео.

8. Способ по п. 1, в котором одна или несколько картинок первого типа содержат одну или несколько пропущенных начальных картинок произвольного доступа.

9. Способ по п. 1, в котором одна или несколько декодируемых начальных картинок содержат одну или несколько декодируемых начальных картинок произвольного доступа.

10. Способ по п. 1, в котором одна или несколько картинок первого типа содержат одну или несколько картинок, которые зависят от данных от одной или нескольких картинок, имеющих место раньше, в порядке декодирования, чем картинка RAP.

11. Устройство для декодирования видеоданных, причем устройство содержит: память, выполненную с возможностью хранения видеоданных; видеodeкодер, содержащий один или несколько процессоров, выполненных с возможностью:

декодировать картинку точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

идентифицировать одну или несколько картинок первого типа, ассоциированных с

картинкой RAP, причем одна или несколько картинок первого типа предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

5 отбрасывать одну или несколько картинок первого типа;

декодировать одну или несколько декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все из одной или нескольких картинок первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или
10 несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP,
15 и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и
выводить декодированную картинку RAP

12. Устройство по п. 11, в котором видеodeкодер дополнительно выполнен с возможностью декодировать одну или несколько начальных картинок относительно
20 картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

25 13. Устройство по п. 11, в котором любая картинка, предшествующая картинке CRA или картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке вывода.

14. Устройство по п. 11, в котором видеodeкодер дополнительно выполнен с
30 возможностью декодировать одну или несколько начальных картинок, ассоциированных с картинкой RAP, причем начальные картинки предшествуют картинке RAP в значении порядка вывода и следуют за картинкой RAP в порядке декодирования; и декодировать одну или несколько находящихся сзади картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все из начальных картинок предшествуют всем
35 находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

15. Устройство по п. 11, в котором видеodeкодер выполнен с возможностью декодировать одну или несколько декодируемых начальных картинок для картинки
40 RAP посредством декодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP
45 в порядке декодирования и не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования.

16. Устройство по п. 11, в котором видеodeкодер дополнительно выполнен с возможностью декодировать одну или несколько картинок, которые предшествуют

картинке RAP в порядке декодирования; и в котором упомянутый видеodeкодер выполнен с возможностью декодировать одну или несколько декодируемых начальных картинок посредством декодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все
 5 картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в
 10 порядке декодирования.

17. Устройство по п. 11, причем устройство содержит по меньшей мере одно из: интегральной схемы, микропроцессора и устройства беспроводной связи, которое включает в себя видеodeкодер.

15 18. Устройство по п. 11, в котором одна или несколько картинок первого типа содержат одну или несколько пропущенных начальных картинок произвольного доступа.

19. Устройство по п. 11, в котором одна или несколько декодируемых начальных картинок содержат одну или несколько декодируемых начальных картинок
 20 произвольного доступа.

20. Устройство по п. 11, в котором одна или несколько картинок первого типа содержат одну или несколько картинок, которые зависят от данных от одной или нескольких картинок, имеющих место раньше, в порядке декодирования, чем картинка RAP.

21. Устройство для декодирования видеоданных, причем устройство содержит: средство для декодирования картинки точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или
 25 картинки доступа с разорванной ссылкой (BLA);

средство для идентификации одной или нескольких картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа
 30 предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

средство для отбрасывания одной или нескольких картинок первого типа;

35 средство для декодирования одной или нескольких декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все из одной или нескольких картинок первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке
 40 вывода, причем одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных
 45 картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и

средство для вывода декодированной картинки RAP.

22. Устройство по п. 21, дополнительно содержащее средство для декодирования

одной или нескольких начальных картинок относительно картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

23. Устройство по п. 21, в котором любая картинка, предшествующая картинке CRA или картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке вывода.

24. Устройство по п. 21, причем устройство дополнительно содержит: средство для декодирования одной или нескольких начальных картинок, ассоциированных с картинкой RAP, при этом начальные картинки предшествуют картинке RAP в значении порядка вывода и следуют за картинкой RAP в порядке декодирования, и

средство для декодирования одной или нескольких находящихся сзади картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все из начальных картинок предшествуют всем находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

25. Устройство по п. 21, в котором средство для декодирования одной или нескольких декодируемых начальных картинок для картинки RAP содержит средство для декодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

26. Устройство по п. 21, причем устройство дополнительно содержит: средство для декодирования одной или нескольких картинок, которые предшествуют картинке RAP в порядке декодирования, и

при этом средство для декодирования одной или нескольких декодируемых начальных картинок содержит средство для декодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

27. Невременный считываемый компьютером запоминающий носитель, хранящий инструкции, которые, когда исполняются одним или несколькими процессорами, вынуждают упомянутые один или несколько процессоров:

декодировать картинку точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

идентифицировать одну или несколько картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют

картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

отбрасывать одну или несколько картинок первого типа;

- 5 декодировать одну или несколько декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все из одной или нескольких картинок первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или
10 несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на
15 видеоданные, более ранние, чем картинка RAP в порядке декодирования, и выводить декодированную картинку RAP.

28. Способ кодирования видеоданных, причем способ содержит:

- кодирование, устройством кодирования видео, содержащим один или несколько процессоров, картинки точки произвольного доступа (RAP), причем картинка RAP
20 содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

- кодирование, устройством кодирования видео, содержащим один или несколько процессоров, одной или нескольких картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют
25 картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

- кодирование, устройством кодирования видео, содержащим один или несколько процессоров, одной или нескольких декодируемых начальных картинок для картинки
30 RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих
35 значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и

- 40 вывод, устройством кодирования видео, содержащим один или несколько процессоров, картинки RAP.

29. Способ по п. 28, дополнительно содержащий кодирование одной или нескольких начальных картинок относительно картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади
45 картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

30. Способ по п. 28, в котором любая картинка, предшествующая картинке CRA или

картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке вывода.

31. Способ по п. 28, причем способ дополнительно содержит:

кодирование, устройством кодирования видео, содержащим один или несколько процессоров, одной или нескольких начальных картинок, ассоциированных с картинкой RAP, при этом начальные картинки предшествуют картинке RAP в значении порядка вывода и следуют за картинкой RAP в порядке декодирования, и

кодирование, устройством кодирования видео, содержащим один или несколько процессоров, одной или нескольких находящихся сзади картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все из начальных картинок предшествуют всем находящимся сзади картинкам в порядке декодирования, причем находящиеся сзади картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

32. Способ по п. 28, в котором кодирование одной или нескольких декодируемых начальных картинок для картинки RAP содержит кодирование одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

33. Способ по п. 28, причем способ дополнительно содержит:

кодирование, устройством кодирования видео, содержащим один или несколько процессоров, одной или нескольких картинок, которые предшествуют картинке RAP в порядке декодирования, и

при этом кодирование одной или нескольких декодируемых начальных картинок содержит кодирование одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

34. Способ по п. 28, в котором кодирование картинки RAP выполняется как часть процесса кодирования видео.

35. Способ по п. 28, в котором одна или несколько картинок первого типа содержат одну или несколько пропущенных начальных картинок произвольного доступа.

36. Способ по п. 28, в котором одна или несколько декодируемых начальных картинок содержат одну или несколько декодируемых начальных картинок произвольного доступа.

37. Способ по п. 28, в котором одна или несколько картинок первого типа содержат одну или несколько картинок, которые зависят от данных от одной или нескольких картинок, имеющих место раньше, в порядке декодирования, чем картинка RAP.

38. Устройство для кодирования видеоданных, причем устройство содержит:

память, выполненную с возможностью хранения видеоданных;

видеокодер, содержащий один или несколько процессоров, выполненных с возможностью:

кодировать картинку точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинок чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

5 кодировать одну или несколько картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

кодировать одну или несколько декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть
10 истинным, причем заданное условие требует того, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода более ранний, чем
15 значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и
выводить картинку RAP.

20 39. Устройство по п. 38, в котором видеокодер дополнительно сконфигурирован, чтобы закодировать одну или несколько начальных картинок относительно картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие
25 значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

40. Устройство по п. 38, в котором любая картинка, предшествующая картинке CRA или картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке
30 вывода.

41. Устройство по п. 38, в котором видеокодер дополнительно сконфигурирован, чтобы закодировать одну или несколько начальных картинок, ассоциированных с картинкой RAP, при этом начальные картинки предшествуют картинке RAP в значении
35 порядка вывода и следуют за картинкой RAP в порядке декодирования; и закодировать одну или несколько находящихся сзади картинок, ассоциированных с картинкой RAP в соответствии с ограничением, что все из начальных картинок предшествуют всем находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади
картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

40 42. Устройство по п. 38, в котором видеокодер сконфигурирован, чтобы закодировать одну или несколько декодируемых начальных картинок для картинки RAP, содержит кодирование одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки
первого типа предшествуют одной или нескольким декодируемым начальным картинкам
45 в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

43. Устройство по п. 38, в котором видеокодер дополнительно сконфигурирован, чтобы закодировать одну или несколько картинок, которые предшествуют картинке RAP в порядке декодирования, и при этом видеокодер сконфигурирован, чтобы закодировать одну или несколько декодируемых начальных картинок посредством кодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

44. Устройство по п. 38, причем устройство содержит по меньшей мере одно из: интегральной схемы,

микропроцессора и

устройства беспроводной связи, которое включает в себя видеокодер.

45. Устройство по п. 38, в котором одна или несколько картинок первого типа содержат одну или несколько пропущенных начальных картинок произвольного доступа.

46. Устройство по п. 38, в котором одна или несколько декодируемых начальных картинок содержат одну или несколько декодируемых начальных картинок произвольного доступа.

47. Устройство по п. 38, в котором одна или несколько картинок первого типа содержат одну или несколько картинок, которые зависят от данных от одной или нескольких картинок, имеющих место раньше, в порядке декодирования, чем картинка RAP.

48. Устройство для кодирования видеоданных, причем устройство содержит: средство для кодирования картинки точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

средство для кодирования одной или нескольких картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования;

средство для кодирования одной или нескольких декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, при этом одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и

вывода, устройством кодирования видео, содержащим один или несколько процессоров, картинки RAP.

49. Устройство по п. 48, дополнительно содержащее средство для того, чтобы закодировать одну или несколько начальных картинок относительно картинки RAP в соответствии с ограничением, что все из начальных картинок для картинки RAP предшествуют всем находящимся сзади картинкам для картинки RAP в порядке декодирования, при этом находящиеся сзади картинки содержат картинки, имеющие значения порядка вывода, которые больше, чем значение порядка вывода картинки RAP.

50. Устройство по п. 48, в котором любая картинка, предшествующая картинке CRA или картинке BLA в порядке декодирования, предшествует любой декодируемой начальной картинке, ассоциированной с картинкой CRA или картинкой BLA в порядке вывода.

51. Устройство по п. 48, причем устройство дополнительно содержит: средство для кодирования одной или нескольких начальных картинок, ассоциированных с картинкой RAP, при этом начальные картинки предшествуют картинке RAP в значении порядка вывода и следуют за картинкой RAP в порядке декодирования, и

средство для кодирования одной или нескольких находящихся сзади картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все из начальных картинок предшествуют всем находящимся сзади картинкам в порядке декодирования, при этом находящиеся сзади картинки следуют за картинкой RAP и в значении порядка вывода, и в порядке декодирования.

52. Устройство по п. 48, в котором средство для кодирования одной или нескольких декодируемых начальных картинок для картинки RAP содержит средство для кодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные более ранние, чем картинка RAP в порядке декодирования.

53. Устройство по п. 48, причем устройство дополнительно содержит: средство для кодирования одной или нескольких картинок, которые предшествуют картинке RAP в порядке декодирования, и

при этом средство для кодирования одной или нескольких декодируемых начальных картинок содержит средство для кодирования одной или нескольких декодируемых начальных картинок, ассоциированных с картинкой RAP, в соответствии с ограничением, что все картинки, которые предшествуют картинке RAP в порядке декодирования, также предшествуют всем из одной или нескольких декодируемых начальных картинок в порядке вывода, при этом одна или несколько декодируемых начальных картинок предшествуют картинке RAP в порядке вывода, следуют за картинкой RAP в порядке декодирования и не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования.

54. Устройство по п. 48, в котором одна или несколько картинок первого типа содержат одну или несколько пропущенных начальных картинок произвольного доступа.

55. Устройство по п. 48, в котором одна или несколько декодируемых начальных картинок содержат одну или несколько декодируемых начальных картинок произвольного доступа.

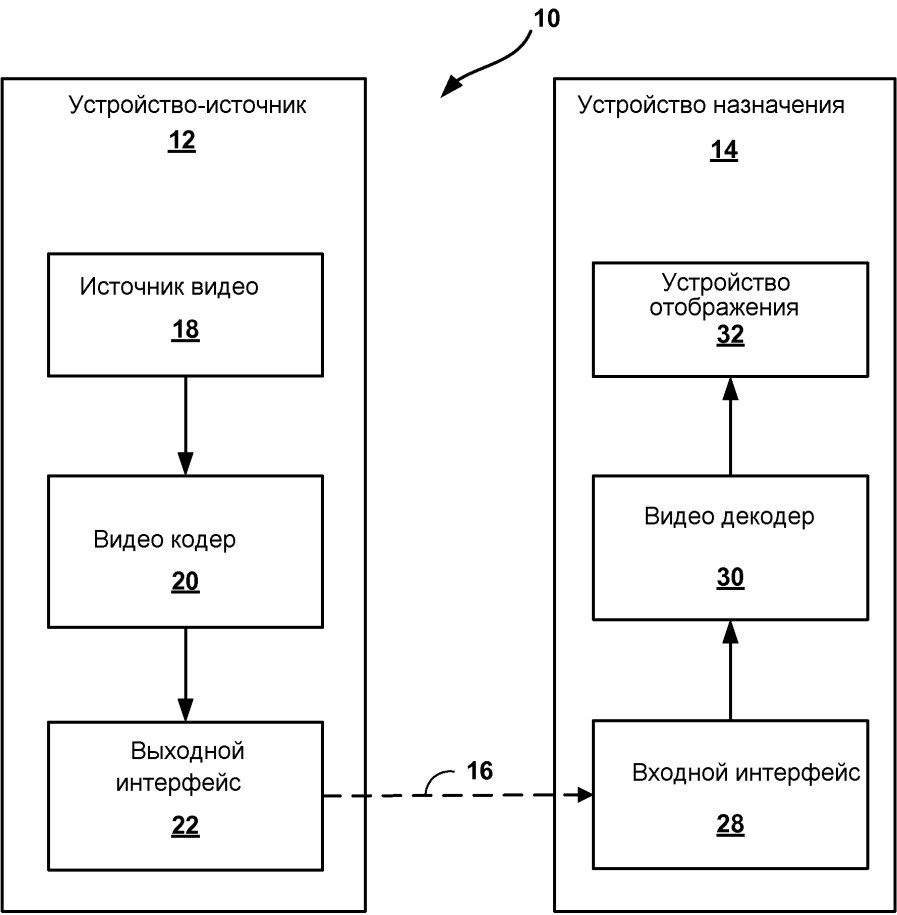
56. Устройство по п. 48, в котором одна или несколько картинок первого типа содержат одну или несколько картинок, которые зависят от данных от одной или нескольких картинок, имеющих место раньше, в порядке декодирования, чем картинка RAP.

57. Невременный считываемый компьютером запоминающий носитель, хранящий инструкции, которые, когда исполняются одним или несколькими процессорами, вынуждают эти один или несколько процессоров:

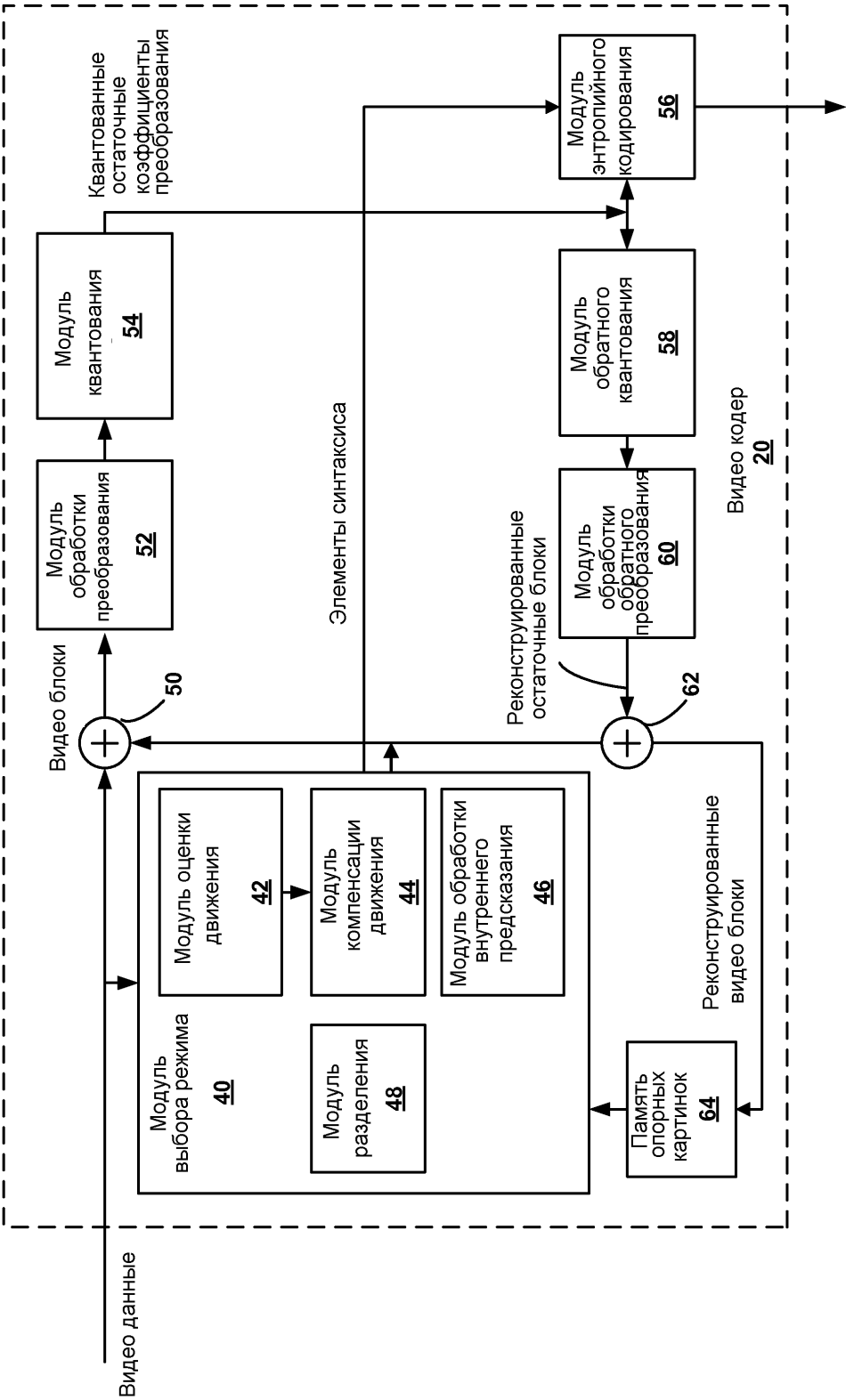
кодировать картинку точки произвольного доступа (RAP), причем картинка RAP содержит одну из картинки чистого произвольного доступа (CRA) или картинки доступа с разорванной ссылкой (BLA);

кодировать одну или несколько картинок первого типа, ассоциированных с картинкой RAP, причем одна или несколько картинок первого типа предшествуют картинке RAP в значении порядка вывода, следуют за картинкой RAP в порядке декодирования и ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и

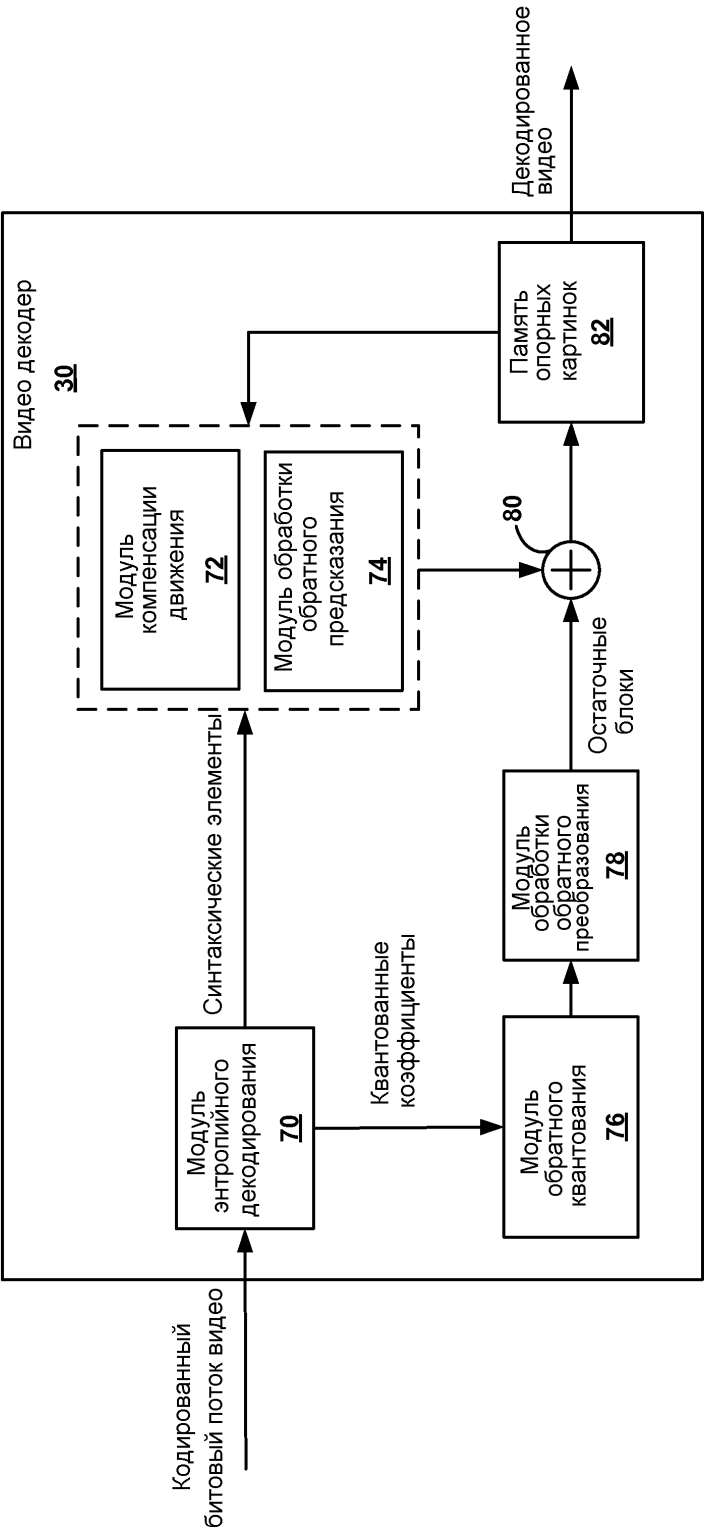
кодировать одну или несколько декодируемых начальных картинок для картинки RAP в соответствии с ограничением, которое вынуждает заданное условие быть истинным, причем заданное условие требует того, что все картинки первого типа предшествуют одной или нескольким декодируемым начальным картинкам, ассоциированным с картинкой RAP в порядке вывода, причем одна или несколько декодируемых начальных картинок содержат одну или несколько картинок, имеющих значения порядка вывода, которые указывают порядок вывода, более ранний, чем значение порядка вывода картинки RAP, и при этом одна или несколько декодируемых начальных картинок декодированы после декодирования картинки RAP, и при этом одна или несколько декодируемых начальных картинок не ссылаются на видеоданные, более ранние, чем картинка RAP в порядке декодирования, и выводить картинку RAP.



ФИГ.1

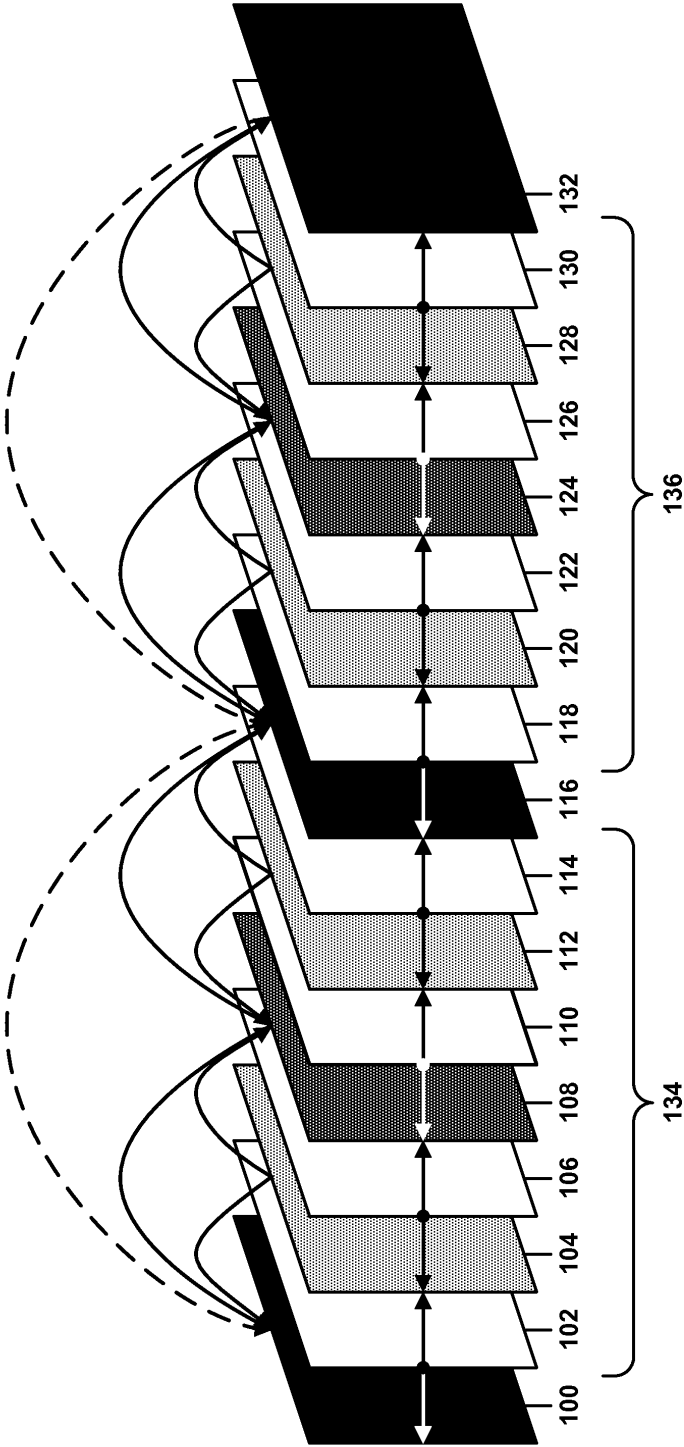


ФИГ.2

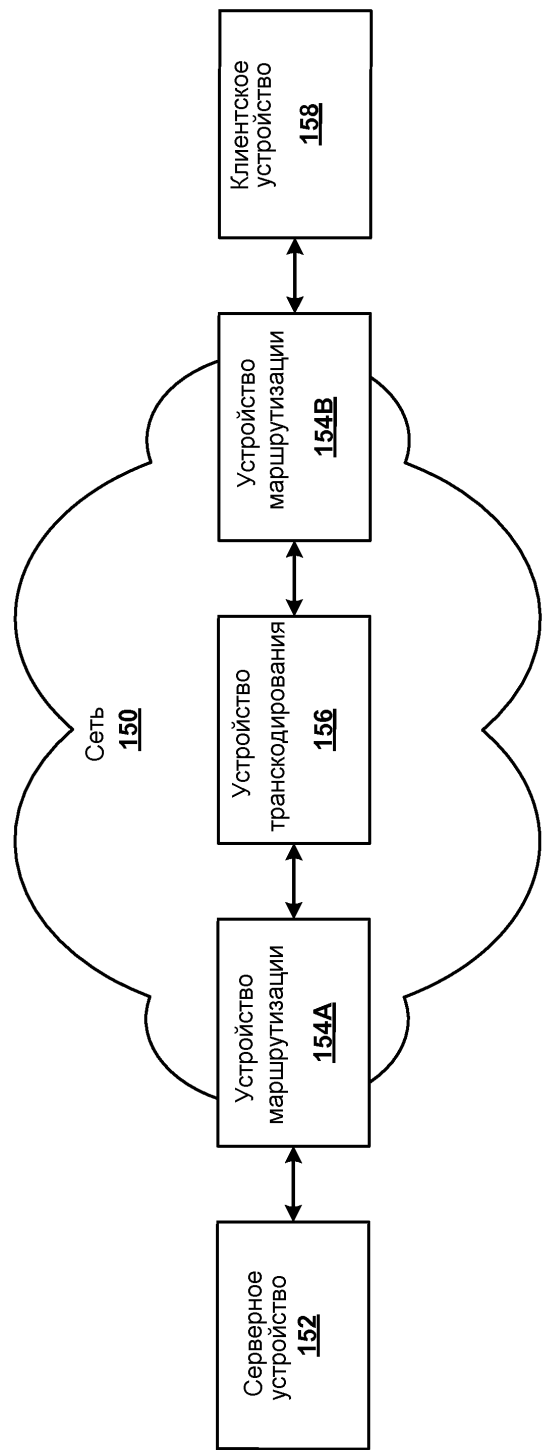


ФИГ.3

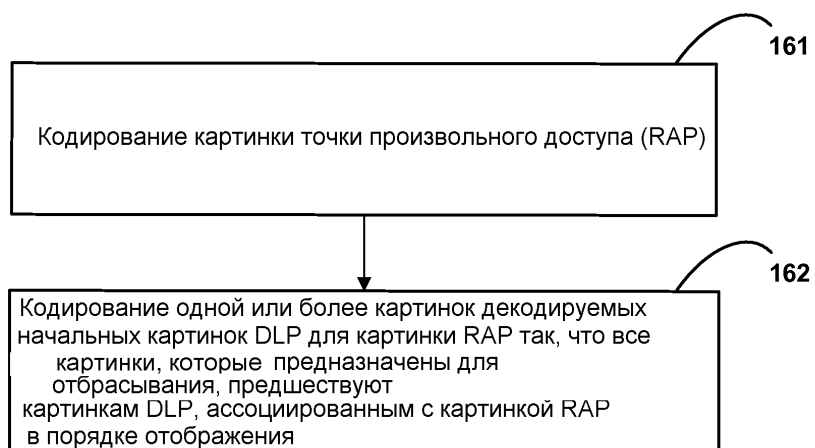
4/11



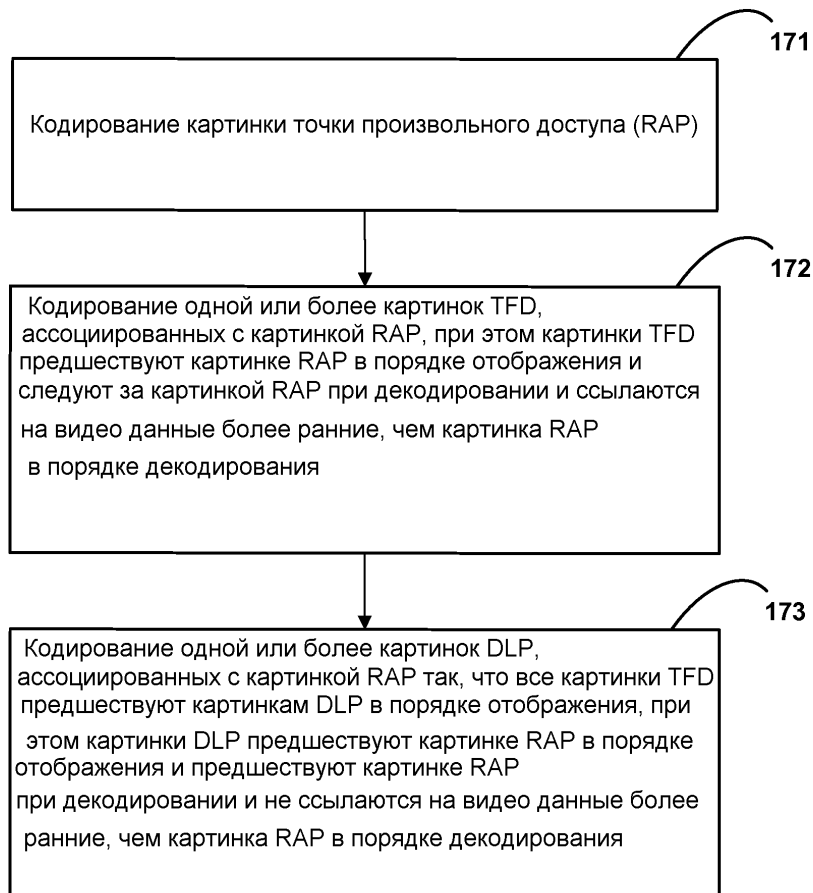
ФИГ.4



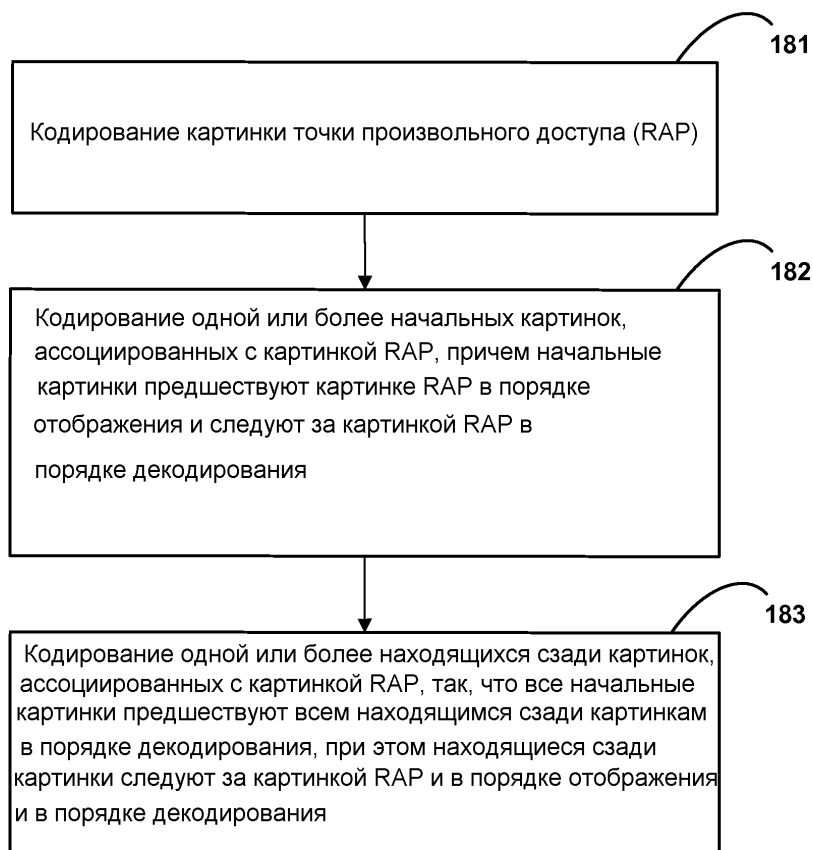
ФИГ.5



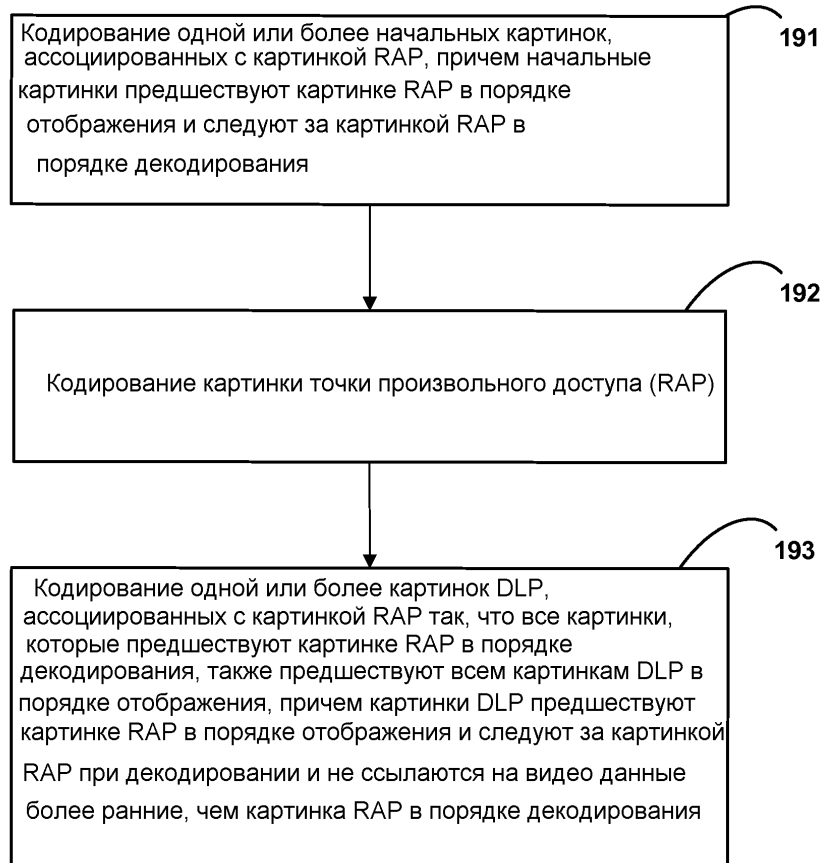
ФИГ.6



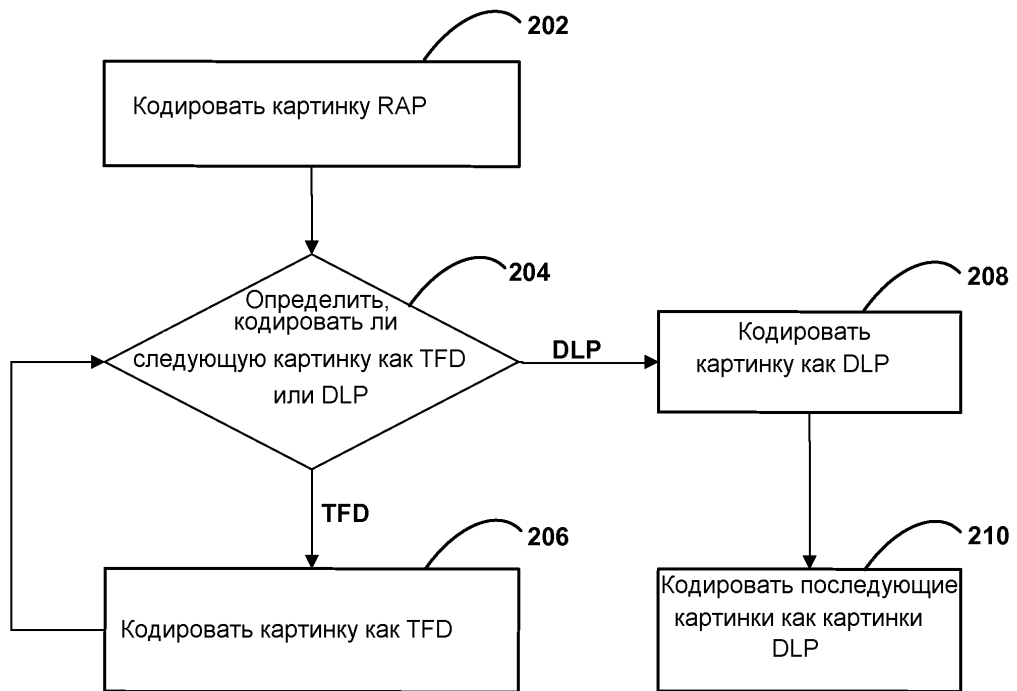
ФИГ.7



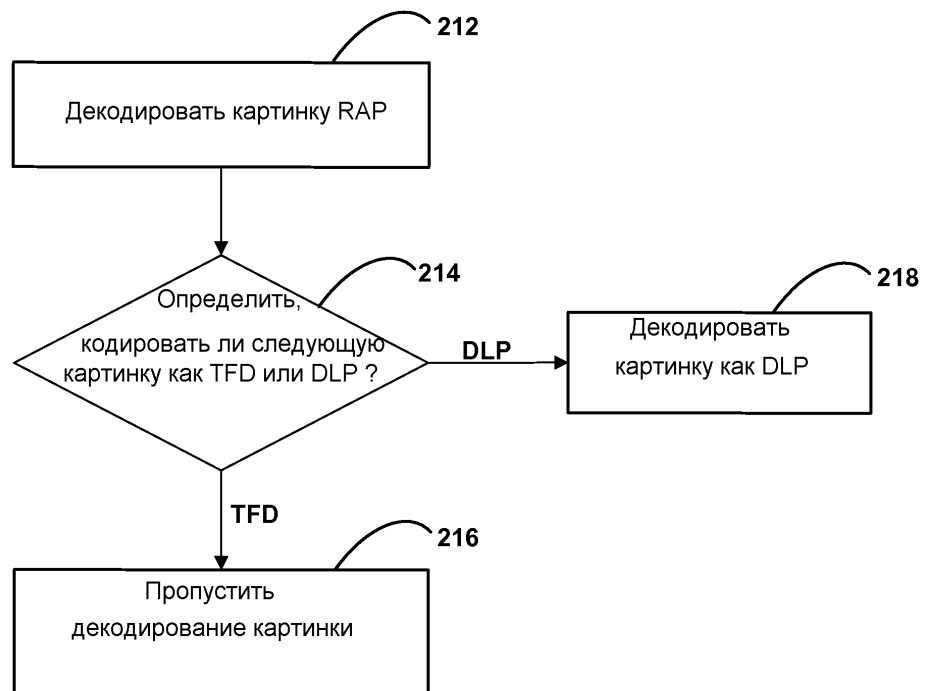
ФИГ.8



ФИГ.9



ФИГ.10



ФИГ.11