



(51) МПК
H04N 19/70 (2014.01)
H04N 19/44 (2014.01)
H04N 19/184 (2014.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
 ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/70 (2006.01); *H04N 19/44* (2006.01); *H04N 19/184* (2006.01); *H04N 19/136* (2006.01); *H04N 19/30* (2006.01); *H04N 19/152* (2006.01); *H04N 19/196* (2006.01); *H04N 19/12* (2006.01)

(21)(22) Заявка: 2015115519, 18.09.2013

(24) Дата начала отсчета срока действия патента:
18.09.2013

Дата регистрации:
30.03.2018

Приоритет(ы):

(30) Конвенционный приоритет:
24.09.2012 US 61/705,102;
14.06.2013 US 13/918,041

(43) Дата публикации заявки: 20.11.2016 Бюл. № 32

(45) Опубликовано: 30.03.2018 Бюл. № 10

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 24.04.2015

(86) Заявка РСТ:
US 2013/060412 (18.09.2013)

(87) Публикация заявки РСТ:
WO 2014/047183 (27.03.2014)

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

(72) Автор(ы):

ВАН Е-Куй (US)

(73) Патентообладатель(и):

КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(56) Список документов, цитированных в отчете о поиске: BENJAMIN BROSS et al., "High efficiency video coding (HEVC) text specification draft 8", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, 11-20 July, 2012, 260 с., Document: JCTVC-J1003_d7, опубл. 28.07.2012. YE-KUI WANG, "BoG on high-level syntax for extension planning", (см. прод.)

(54) ПАРАМЕТРЫ ГИПОТЕТИЧЕСКОГО ОПОРНОГО ДЕКОДЕРА ПРИ КОДИРОВАНИИ ВИДЕО

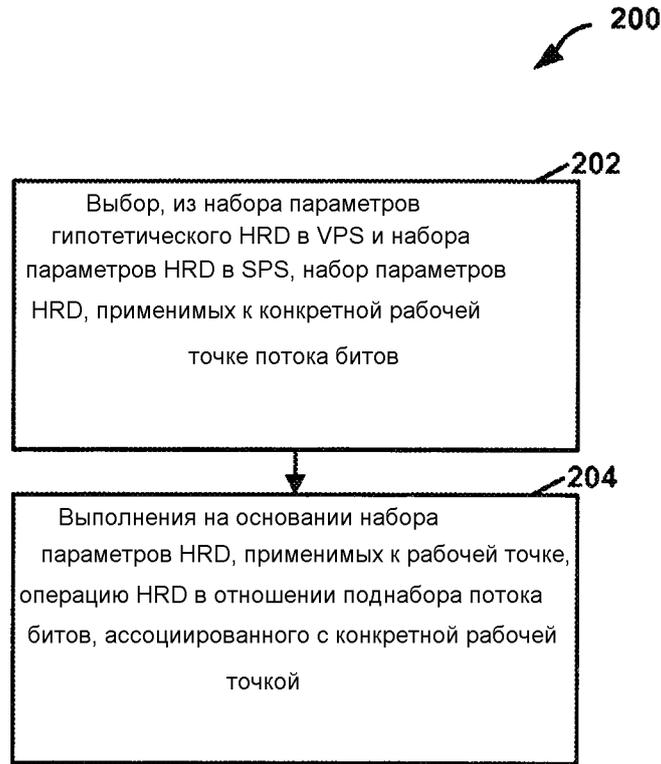
(57) Реферат:

Изобретение относится к области кодирования/декодирования видеоданных. Технический результат – сокращение траты вычислительных ресурсов. Способ обработки видеоданных содержит: прием набора параметров видео (VPS) и набора параметров последовательности (SPS), причем VPS содержит множество синтаксических структур параметров гипотетического эталонного декодера (HRD), содержащих соответствующие наборы параметров HRD, причем VPS дополнительно

включает в себя элементы синтаксиса, указывающие соответствующий набор идентификаторов уровня для соответствующей синтаксической структуры параметров HRD, и SPS содержит синтаксическую структуру параметров HRD, содержащую набор параметров HRD; выбор набора параметров HRD, применимых к некоторой конкретной рабочей точке, при этом набор параметров HRD в наборе синтаксической структуры параметров HRD в SPS применим к упомянутой конкретной рабочей

точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки идентичен набору из всех идентификаторов уровня, присутствующих в потоке битов, который подвергается тестированию с помощью теста

соответствия потока битов; и выполнение операции HRD с использованием выбранного набора параметров HRD. 4 н. и 14 з.п. ф-лы, 6 ил., 8 табл.



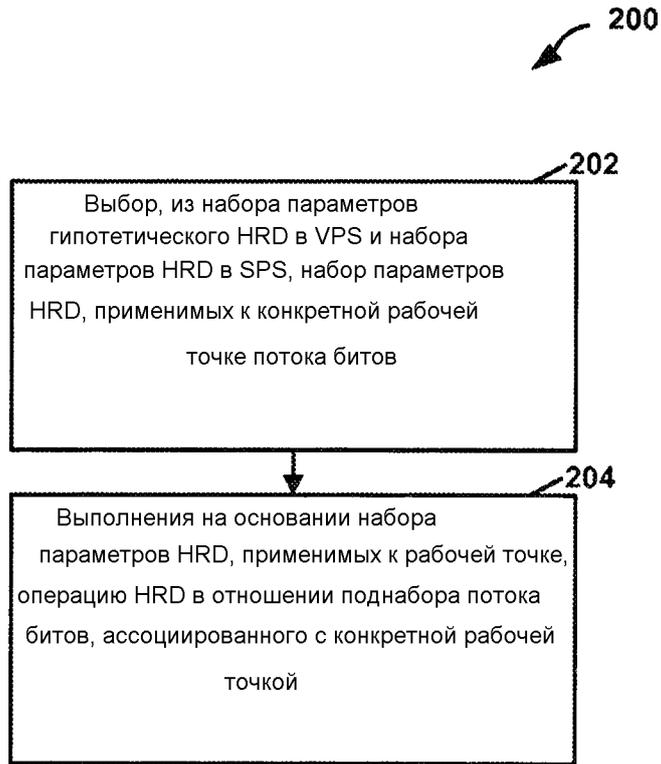
ФИГ.4

(56) (продолжение):

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, 11-20 July, 2012, 17 с., Document: JCTVC-J00574r1, опублик. 19.07.2012. US 2004/0005007 A1, 08.01.2004. MISKA M. HANNUKSELA, "AHG10 Hooks for Scalable Coding: Sequence Parameter Set Desing", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting: Stockholm, SE, 11-20 July, 2012, 11 с., Document: JCTVC-J0074, опублик. 02.07.2012. US 2010/0091837 A1, 15.04.2010. US 2006/0120463 A1, 08.06.2006. US 2009/0003534 A1, 01.01.2009. RU 2406254 C2, 10.12.2010.

RU 2 6 4 9 2 9 7 C 2

RU 2 6 4 9 2 9 7 C 2



ФИГ.4

RU 2649297 C2

RU 2649297 C2

[0001] Настоящая заявка испрашивает приоритет предварительной заявки на патент США № 61/705,102, поданной 24 сентября 2012, все содержимое которой включено здесь по ссылке.

ОБЛАСТЬ ТЕХНИКИ

5 [0002] Настоящее раскрытие относится к кодированию и декодированию видео (то есть, кодированию и/или декодированию видео данных).

УРОВЕНЬ ТЕХНИКИ

[0003] Цифровые способности видео могут быть включены в широкий диапазон устройств, включая цифровые телевизоры, цифровые системы прямого вещания,
10 беспроводные системы вещания, персональные цифровые помощники (PDAs), ноутбуки или настольные компьютеры, планшетные компьютеры, считыватели электронных книг, цифровые камеры, устройства цифровой записи, цифровые медиаплееры, устройства видео игр, пульты видеоигр, сотовые или спутниковые радио-телефоны, так называемые «смартфоны,» устройства организации видео телеконференций,
15 устройства потоковой передачи видео, и т.п. Цифровые видео устройства реализуют способы сжатия видео, такие как описанные в стандартах, определенных посредством MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Часть 10, Усовершенствованное видео кодирование (AVC), стандарт высокоэффективного кодирования видео (HEVC), развиваемый в настоящее время, и расширениях таких стандартов. Видео устройства
20 могут передавать, принимать, кодировать, декодировать, и/или хранить цифровую видео информацию более эффективно, реализовывая такие способы сжатия видео.

[0004] Способы сжатия видео выполняют пространственное (внутри картинки) предсказание и/или временное (между картинками) предсказание, чтобы уменьшить или удалить избыточность, присущую видео последовательностям. Для основанного
25 на блоках кодирования видео, видео вырезка (то есть, видео кадр или часть видео кадра) может быть разделена на блоки видео. Блоки видео во внутренне кодированной (I) вырезке картинки кодируют, используя пространственное предсказание относительно опорных выборок в соседних блоках в той же самой картинке. Блоки видео во внешне кодированной (P или B) вырезке картинки могут использовать пространственное
30 предсказание относительно опорных выборок в соседних блоках в той же самой 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 предсказывающему
35 блоку для блока, который должен быть закодирован. Остаточные данные представляют пиксельные разности между первоначальным блоком, который должен быть закодирован, и предсказывающим блоком. Внешне кодированный блок кодируют согласно вектору движения, который указывает на блок опорных выборок, формирующий предсказывающий блок, и остаточные данные указывают различие
40 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, которые затем могут быть квантованы. Квантованные коэффициенты, первоначально
45 размещенные в двумерном массиве, могут сканироваться, чтобы сформировать одномерный вектор коэффициентов, и энтропийное кодирование может быть применено, чтобы достигнуть даже большей степени сжатия.

[0006] Поток битов кодирования множественных видов может быть сгенерирован

посредством кодирования видов, например, с множественных перспектив. Некоторые стандарты трехмерного (3D) видео были развиты, которые используют кодирование аспектов множественных видов. Например, различные виды могут передать виды левого и правого глаз, чтобы поддерживать 3D видео. Альтернативно, некоторые процессы кодирования 3D видео могут применять так называемое кодирование множественных видов плюс глубина. В кодировании множественных видов плюс глубина поток битов 3D видео может содержать не только компоненты вида текстуры, но также и компоненты вида глубины. Например, каждый вид может содержать один компонент вида текстуры и один компонент вида глубины.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0007] В целом настоящее раскрытие описывает сигнализацию и выбор параметров гипотетического опорного декодера (HRD) при кодировании видео. Более конкретно, устройство выполняет операцию гипотетического опорного декодера (HRD), которая определяет соответствие потока битов стандарту кодирования видео или определяет соответствие видео декодера стандарту кодирования видео. В качестве части выполнения операции HRD, устройство определяет наивысший временный идентификатор поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Кроме того, в качестве части операции HRD, устройство определяет, на основании наивысшего временного идентификатора, конкретный элемент синтаксиса из числа массива элементов синтаксиса. Устройство затем использует этот конкретный элемент синтаксиса в операции HRD.

[0008] В одном примере настоящее раскрытие описывает способ обработки видео данных. Способ содержит выполнение операции HRD. Операция HRD определяет соответствие потока битов стандарту кодирования видео или определяет соответствие видео декодера стандарту кодирования видео. Выполнение операции HRD содержит определение наивысшего временного идентификатора поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Выполнение операции HRD также содержит определение, на основании наивысшего временного идентификатора, конкретного элемента синтаксиса из числа массива элементов синтаксиса. Кроме того, выполнение операции HRD содержит использование конкретного элемента синтаксиса в операции HRD.

[0009] В другом примере настоящее раскрытие описывает устройство, содержащее один или более процессоров, сконфигурированных, чтобы выполнить операцию HRD. Операция HRD определяет соответствие потока битов стандарту кодирования видео или определяет соответствие видео декодера стандарту кодирования видео. Выполнение операции HRD содержит определение наивысшего временного идентификатора поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Выполнение операции HRD также содержит определение, на основании наивысшего временного идентификатора, конкретного элемента синтаксиса из числа массива элементов синтаксиса. Кроме того, выполнение операции HRD содержит использование конкретного элемента синтаксиса в операции HRD.

[0010] В другом примере настоящее раскрытие описывает устройство, содержащее средство для того, чтобы выполнить операцию HRD. Операция HRD определяет соответствие потока битов стандарту кодирования видео или определяет соответствие видео декодера стандарту кодирования видео. Выполнение операции HRD содержит определение наивысшего временного идентификатора поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Выполнение операции HRD также содержит определение, на основании наивысшего временного

идентификатора, конкретного элемента синтаксиса из числа массива элементов синтаксиса. Кроме того, выполнение операции HRD содержит использование конкретного элемента синтаксиса в операции HRD.

5 [0011] В другом примере настоящее раскрытие описывает считываемый компьютером запоминающий носитель, имеющий инструкции, сохраненные на нем, которые, когда выполняются одним или более процессорами устройства, конфигурируют это устройство, чтобы выполнить операцию HRD. Операция HRD определяет соответствие потока битов стандарту кодирования видео или определяет соответствие видео декодера стандарту кодирования видео. Выполнение операции HRD содержит определение
10 наивысшего временного идентификатора поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Кроме того, выполнение операции HRD содержит определение, на основании наивысшего временного идентификатора, конкретного элемента синтаксиса из числа массива элементов синтаксиса. Кроме того, выполнение операции HRD содержит использование конкретного элемента синтаксиса
15 в операции HRD.

[0012] Подробности одного или более примеров раскрытия сформулированы в сопроводительных чертежах и описании ниже. Другие признаки, задачи и преимущества будут очевидны из описания, чертежей, и формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

20 [0013] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему кодирования видео, которая может использовать способы, описанные в настоящем раскрытии.

[0014] Фиг. 2 является блок-схемой, иллюстрирующей примерный кодер видео, который может реализовать способы, описанные в настоящем раскрытии.

25 [0015] Фиг. 3 является блок-схемой, иллюстрирующей примерный декодер видео, который может реализовать способы, описанные в настоящем раскрытии.

[0016] Фиг. 4 является последовательностью операций, иллюстрирующей примерную операцию устройства, в соответствии с одним или более способами настоящего раскрытия.

30 [0017] Фиг. 5 является последовательностью операций, иллюстрирующей примерную операцию устройства, в соответствии с одним или более способами настоящего раскрытия.

[0018] Фиг. 6 является последовательностью операций, иллюстрирующей примерную операцию гипотетического опорного декодера (HRD) устройства, в соответствии с
35 одним или более способами настоящего раскрытия.

ПОДРОБНОЕ ОПИСАНИЕ

[0019] Видео кодер может генерировать поток битов, который включает в себя закодированные видео данные. Поток битов может содержать последовательность единиц уровня абстракции сети (NAL). Единицы NAL потока битов могут включать в
40 себя уровень кодирования видео (VCL) единицы NAL и единицы NAL не-VCL. Единицы NAL VCL могут включать в себя закодированные вырезки картинок. Единица NAL не-VCL может включать в себя набор параметров видео (VPS), набор параметров последовательности (SPS), набор параметров картинки (PPS), дополнительную информацию расширения (SEI), или другие типы данных. VPS является синтаксической
45 структурой, которая может содержать элементы синтаксиса, которые применяются к нулю или более закодированным видео последовательностям целиком. SPS является синтаксической структурой, которая может содержать элементы синтаксиса, которые применяются к нулю или более закодированным видео последовательностям целиком.

Единственный VPS может быть применимым ко множественным SPS. PPS является синтаксической структурой, которая может содержать элементы синтаксиса, которые применяются к нулю или более закодированным картинкам целиком. Единственный SPS может быть применимым ко множественным PPS. Различные аспекты VPS, SPS и PPS могут быть сформированы, в целом, как определено по стандарту HEVC.

[0020] Устройство, такое как устройство сети доставки содержимого (CDN), осведомленный о медиа сетевой элемент (MANE), или видео декодер, может извлечь подпоток битов из потока битов. Устройство может выполнить процесс извлечения подпотока битов, удаляя некоторые единицы NAL из потока битов. Результирующий подпоток битов включает в себя оставшиеся неудаленные единицы NAL потока битов. В качестве примеров, видео данные, декодированные из подпотока битов, могут иметь более низкую частоту кадров и/или могут представить меньше видов, чем первоначальный поток битов.

[0021] Стандарты кодирования видео могут включать в себя различные признаки, чтобы поддерживать процесс извлечения подпотока битов. Например, видео данные потока битов могут быть разделены на набор уровней. Для каждого из уровней данные в более низком уровне могут быть декодированы независимо от данных в любом более высоком уровне. Индивидуальная единица NAL только инкапсулирует данные единственного уровня. Таким образом, единицы NAL, инкапсулирующие данные наивысшего остающегося уровня потока битов, могут быть удалены из потока битов, не затрагивая декодируемость данных в оставшихся более низких уровнях потока битов. При масштабируемом кодировании видео (SVC) более высокие уровни могут включать в себя данные расширения, которые улучшают качество картинок в более низких уровнях (масштабируемость качества), увеличивают пространственный формат картинок в более низких уровнях (пространственная масштабируемость), или увеличивают временную скорость передачи картинок в более низких уровнях (временная масштабируемость). В кодировании множественных видов (MVC) и кодировании трехмерного видео (3DV) более высокие уровни могут включать в себя дополнительные виды.

[0022] Единицы NAL могут включать в себя заголовки и полезные данные. Заголовки единиц NAL включают в себя элементы синтаксиса `nuh_reserved_zero_6bits`. Элемент синтаксиса `nuh_reserved_zero_6bits` единицы NAL равен 0, если единица NAL относится к базовому уровню при кодировании множественных видов, кодировании 3DV, или SVC. Данные в базовом уровне потока битов могут быть декодированы независимо от данных в любом другом уровне потока битов. Если единица NAL не относится к базовому уровню при кодировании множественных видов, 3DV, или SVC, элемент синтаксиса `nuh_reserved_zero_6bits` может иметь ненулевое значение. В частности, если единица NAL не относится к базовому уровню при кодировании множественных видов, 3DV, или SVC, элемент синтаксиса `nuh_reserved_zero_6bits` единицы NAL задает идентификатор уровня единицы NAL.

[0023] Кроме того, некоторые картинки в пределах уровня могут быть декодированы независимо от других картинок в пределах того же самого уровня. Таким образом, единицы NAL, инкапсулирующие данные некоторых картинок уровня, могут быть удалены из потока битов, не затрагивая декодируемость других картинок в этом уровне. Например, картинки с четным значением счета по порядку картинки (POC) могут быть декодируемыми независимо от картинок с нечетными значениями POC. Удаление единиц NAL, инкапсулирующих данные таких картинок, может уменьшить частоту кадров потока битов. Поднабор картинок в пределах уровня, который может быть декодирован

независимо от других картинок в пределах этого уровня, может быть упомянут здесь как подуровень.

[0024] Единицы NAL могут включать в себя элементы синтаксиса `temporal_id`. Элемент синтаксиса `temporal_id` единицы NAL определяет временный идентификатор единицы NAL. Если временный идентификатор первой единицы NAL меньше, чем временный идентификатор второй единицы NAL, данные, инкапсулированные первой единицей NAL, могут быть декодированы независимо от данных, инкапсулированных второй единицей NAL.

[0025] Каждая рабочая точка потока битов ассоциирована с набором идентификаторов уровня (то есть, набором значений `nuh_reserved_zero_6bits`) и временным идентификатором. Набор идентификаторов уровня может быть обозначен как `OpLayerIdSet`, и временный идентификатор может быть обозначен как `TemporalID`. Если идентификатор уровня единицы NAL находится в наборе идентификаторов уровня рабочей точки, и временный идентификатор единицы NAL меньше чем или равен временному идентификатору рабочей точки, единица NAL ассоциирована с этой рабочей точкой. Представлением рабочей точки является поднабор потока битов, который ассоциирован с рабочей точкой. Представление рабочей точки может включать в себя каждую единицу NAL, которая ассоциирована с рабочей точкой. Представление рабочей точки не включает в себя единицы NAL VCL, которые не ассоциированы с рабочей точкой.

[0026] Внешний источник может задавать набор целевых идентификаторов уровня для рабочей точки. Например, устройство, такое как устройство CDN или MANE, может задавать набор целевых идентификаторов уровня. В этом примере устройство может использовать набор целевых идентификаторов уровня, чтобы идентифицировать рабочую точку. Устройство может затем извлечь представление рабочей точки для рабочей точки и отправить представление рабочей точки вместо первоначального потока битов к клиентскому устройству. Извлечение и отправление представления рабочей точки клиентскому устройству могут уменьшить частоту следования битов потока битов.

[0027] Кроме того, стандарты кодирования видео задают модели буферизации видео. Модель буферизации видео может также упоминаться как «гипотетический опорный декодер» или «HRD». HRD описывает, как данные должны быть буферизованы для декодирования и как декодированные данные буферизуются для вывода. Например, HRD описывает операцию буфера кодированных картинок («CPB») и буфера декодированных картинок («DPB») в видео декодере. CPB является буфером «первый вошел - первый вышел», содержащим единицы доступа в порядке декодирования, заданном посредством HRD. DPB является буфером, хранящим декодированные картинки для ссылки, переупорядочения вывода, или задержки вывода, заданных посредством HRD.

[0028] Видео кодер может сигнализировать набор параметров HRD. Параметры HRD управляют различными аспектами HRD. Параметры HRD включают в себя начальную задержку удаления из CPB, размер CPB, частоту следования битов, начальную задержку вывода из DPB, и размер DPB. Эти параметры HRD закодированы в синтаксической структуре `hrd_parameters()`, заданной в VPS и/или SPS. Параметры HRD могут также быть заданы в сообщении дополнительной информации расширения (SEI) периода буферизации или сообщении SEI тактирования картинок.

[0029] Как объяснено выше, представление рабочей точки может иметь другую частоту кадров и/или частоту следования битов, чем первоначальный поток битов. Это

имеет место потому, что представление рабочей точки может не включать в себя некоторые картинки и/или некоторые из данных первоначального потока битов. Следовательно, если видео декодер должен был удалить данные из CPB и/или DPB при конкретной скорости передачи при обработке первоначального потока битов, и если
 5 видео декодер должен был удалить данные из CPB и/или DPB при одной и той же скорости передачи при обработке представления рабочей точки, видео декодер может удалить слишком много или слишком мало данных из CPB и/или DPB. Соответственно, видео кодер может сигнализировать различные наборы параметров HRD для различных рабочих точек. В развиваемом стандарте высокоэффективного кодирования видео
 10 (HEVC) видео кодер может сигнализировать наборы параметров HRD в VPS, или видео кодер может сигнализировать наборы параметров HRD в SPS. Проект развивающегося стандарта HEVC, названного «HEVC, Рабочий Проект 8» описан в Bross et al., «High Efficiency Video Coding (HEVC) text specification draft 8,» Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting, Stockholm,
 15 Sweden, июль 2012, который на 8 мая 2013 доступен по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Sweden/wg11/JCTVC-J1003-v8.zip.

[0030] В некоторых версиях HEVC только наборы параметров HRD в VPS выбирают для операций HRD. Таким образом, хотя параметры HRD могут быть предоставлены в SPS, наборы параметров HRD в SPS не выбирают видео декодерами HEVC для
 20 операций HRD. Видео декодеры всегда синтаксически разбирают и декодируют VPS потока битов. Следовательно, видео декодеры всегда синтаксически разбирают и декодируют наборы параметров HRD для VPS. Это верно независимо от того, включает ли поток битов в себя единицы NAL не базового уровня. Следовательно, если поток битов включает в себя единицы NAL не базового уровня, может быть тратой
 25 вычислительных ресурсов синтаксически разбирать и обращаться с наборами параметров HRD в SPS. Кроме того, если наборы параметров HRD присутствуют в VPS, наборы параметров HRD в SPS могут быть потраченными впустую битами.

[0031] В соответствии со способами настоящего раскрытия, видео кодер может генерировать поток битов, который включает в себя SPS, который применим к
 30 последовательности картинок. SPS включает в себя набор параметров HRD. Набор параметров HRD применим к каждой рабочей точке потока битов, который имеет набор идентификаторов уровня, которые соответствуют набору целевых идентификаторов уровня. Таким образом, наборы параметров HRD в SPS не расходуются впустую, а вместо этого могут использоваться для операций HRD.
 35 Например, устройство может выбрать, из числа набора параметров HRD в VPS и набора параметров HRD в SPS, набор параметров HRD, применимых к конкретной рабочей точке. Устройство может выполнять, на основании, по меньшей мере частично, набора параметров HRD, применимых к конкретной рабочей точке, тест соответствия потока битов, который проверяет, соответствует ли поднабор потока битов, ассоциированный
 40 с конкретной рабочей точкой, стандарту кодирования видео.

[0032] Устройство, такое как видео кодер, видео декодер, или другой тип устройства, такое как устройство CDN или MANE, может выполнить тест соответствия потока битов в отношении представления рабочей точки для рабочей точки. Тест соответствия
 45 потока битов может верифицировать, что представление рабочей точки соответствует стандарту кодирования видео, такому как HEVC. Как упомянуто выше, набор целевых идентификаторов уровня и временный идентификатор могут быть использованы для идентификации рабочей точки. Набор целевых идентификаторов уровня может быть обозначен как «TargetDecLayerIdSet». Временный идентификатор может быть обозначен

как «TargetDecHighestTid». Проблематично, Working Draft 8 HEVC не задает, как TargetDecLayerIdSet или TargetDecHighestTid устанавливаются, выполняя тест соответствия потока битов.

[0033] В соответствии с одним или более способами настоящего раскрытия, устройство может выполнить процесс декодирования в качестве части выполнения теста соответствия потока битов. Выполнение процесса декодирования содержит выполнение процесса извлечения потока битов, чтобы декодировать из потока битов представление рабочей точки для рабочей точки, определенной целевым набором идентификаторов уровня, и целевой наивысший временный идентификатор. Целевой набор идентификаторов уровня (то есть, TargetDecLayerIdSet) содержит значения элементов синтаксиса идентификатора уровня (например, элементы синтаксиса nuh_reserved_zero_6bits), присутствующие в представлении рабочей точки. Целевой набор идентификаторов уровня является поднабором значений элементов синтаксиса идентификатора уровня потока битов. Целевой наивысший временный идентификатор (то есть, TargetDecHighestTid) равен наибольшему временному идентификатору, присутствующему в представлении рабочей точки. Целевой наивысший временный идентификатор меньше чем или равен наибольшему временному идентификатору, присутствующему в потоке битов. Выполнение процесса декодирования может также содержать декодирование единиц NAL представления рабочей точки.

[0034] В HEVC SPS может включать в себя массив элементов синтаксиса, обозначенных как sps_max_dec_pic_buffering [i], где i ранжируется от 0 до максимального количества временных уровней в потоке битов. sps_max_dec_pic_buffering [i] указывает максимальный требуемый размер DPB, когда наивысший временный идентификатор (HighestTid) равен i. sps_max_dec_pic_buffering [i] указывает требуемый размер в терминах единиц буферов хранения картинок.

[0035] Кроме того, в HEVC, SPS может включать в себя массив элементов синтаксиса, обозначенный как sps_max_num_reorder_pics [i], где i ранжируется от 0 до максимального количества временных уровней в потоке битов. sps_max_num_reorder_pics [i] указывает максимальное разрешенное количество картинок, предшествующих любой картинке в порядке декодирования и следующих за этой картинкой в порядке вывода, когда наивысший временный идентификатор (HighestTid) равен i.

[0036] В HEVC набор параметров HRD может включать в себя массив элементов синтаксиса, обозначенный cpb_cnt_minus1 [i], где i ранжируется от 0 до максимального количества временных уровней в потоке битов. cpb_cnt_minus1 [i] задает количество альтернативных спецификаций CPB в потоке битов закодированной видео последовательности, когда наивысший временный идентификатор (HighestTid) равен i, в котором одна альтернативная спецификация CPB относится к одной конкретной операции CPB с конкретным набором параметров CPB.

[0037] В HEVC Working Draft 8, sps_max_dec_pic_buffering [i], sps_max_num_reorder_pics [i], и cpb_cnt_minus1 [i] должны образом не выбираться в операциях HRD, операциях соответствия (согласования) потока битов, и ограничениях уровня. Это, по меньшей мере частично, является следствием того что HEVC Working Draft 8 не задает то, что подразумевается под наивысшим временным идентификатором (HighestTid).

[0038] В соответствии с одним или более способами настоящего раскрытия, устройства, такие как видео кодер, видео декодер, или другое устройство, могут определить наивысший временный идентификатор поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов. Кроме того, устройство может определить, на основании наивысшего временного идентификатора, конкретный

элемент синтаксиса из числа массива элементов синтаксиса (например, `sps_max_dec_pic_buffering []`, `sps_max_num_reorder_pics []`, или `cpb_cnt_minus1 []`). Устройство может выполнить операцию, которая использует этот конкретный элемент синтаксиса, чтобы определить соответствие потока битов стандарту кодирования видео или определить соответствие видео декодера стандарту кодирования видео.

[0039] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему 10 кодирования видео, которая может использовать способы настоящего раскрытия. Как используется в настоящем описании, термин «видео кодировщик» относится в общем и к кодерам видео и к декодерам видео. В настоящем раскрытии термины «кодирование видео» или «кодирование» могут относиться в целом к кодированию видео или декодированию видео.

[0040] Как показано на фиг. 1, система 10 кодирования видео включает в себя исходное устройство 12 и устройство 14 назначения. Исходное устройство 12 генерирует закодированные видео данные. Соответственно, исходное устройство 12 может упоминаться как устройство кодирования видео или аппарат кодирования видео. Устройство 14 назначения может декодировать закодированные видео данные, генерируемые исходным устройством 12. Соответственно, устройство 14 назначения может упоминаться как устройство декодирования видео или аппарат декодирования видео. Исходное устройство 12 и устройство 14 назначения может быть примерами устройств кодировки видео или аппаратов кодировки видео.

[0041] Исходное устройство 12 и устройство 14 назначения могут содержать широкий диапазон устройств, включая настольные компьютеры, мобильные вычислительные устройства, портативные (например, ноутбук) компьютеры, планшетные компьютеры, телевизионные приставки, телефонные трубки, такие как так называемые «смарт» телефоны, телевизоры, камеры, устройства отображения, цифровые медиаплееры, пульты видео игр, находящиеся внутри автомобиля компьютеры, или подобное.

[0042] Устройство 14 назначения может принять закодированные видео данные от исходного устройства 12 через канал 16. Канал 16 может содержать один или более носителей и устройств, способных к перемещению закодированных видео данных от исходного устройства 12 к устройству 14 назначения. В одном примере канал 16 может содержать один или более коммуникационных носителей, которые разрешают исходному устройству 12 передать закодированные видео данные непосредственно к устройству 14 назначения в реальном времени. В этом примере исходное устройство 12 может модулировать закодированные видео данные согласно стандарту связи, такому как протокол беспроводной связи, и может передать модулированные видео данные к устройству 14 назначения. Один или более коммуникационных носителей могут включать в себя беспроводные и/или проводные коммуникационные носители, такие как радиочастотный (RF) спектр или одну или более физических линий передачи. Один или более коммуникационных носителей могут являться частью основанной на пакетной передаче сети, такой как локальная сеть, региональная сеть, или глобальная сеть (например, Интернет). Один или более коммуникационных носителей могут включать в себя маршрутизаторы, коммутаторы, базовые станции, или другое оборудование, которое облегчает связь от исходного устройства 12 к устройству 14 назначения.

[0043] В другом примере канал 16 может включать в себя запоминающий носитель, который хранит закодированные видео данные, генерируемые исходным устройством 12. В этом примере устройство 14 назначения может получить доступ к запоминающему носителю, например, через доступ к диску или доступ к карте. Запоминающий носитель может включать в себя множество локально доступных запоминающих носителей

данных, таких как диски Blu-ray, DVD, CD-ROM, флэш-память, или другие подходящие цифровые запоминающие носители для того, чтобы хранить закодированные видео данные.

[0044] В другом примере канал 16 может включать в себя файловый сервер или другое промежуточное устройство хранения, которое хранит закодированные видео данные, генерируемые исходным устройством 12. В этом примере устройство 14 назначения может получить доступ к закодированным видео данным, хранящимся в файловом сервере или другом промежуточном устройстве хранения, с помощью потоковой передачи или загрузки. Файловый сервер может быть типом сервера, способного к тому, чтобы хранить закодированные видео данные и передавать закодированные видео данных к устройству 14 назначения. Примерные файловые серверы включают в себя web-серверы (например, для вебсайта), серверы согласно протоколу передачи файлов (FTP), устройства соединенных с сетью устройств хранения (NAS), и локальные дисководы. В примере согласно Фиг. 1 канал 16 включает в себя дополнительное устройство 21. В некоторых примерах дополнительное устройство 21 является устройством CDN, MANE, или другим типом устройства.

[0045] Устройство 14 назначения может получить доступ к закодированным видео данным через стандартное соединение данных, такое как интернет-соединение. Примерные типы соединений данных могут включать в себя беспроводные каналы (например, соединения Wi-Fi), проводные соединения (например, цифровая абонентская линия (DSL), кабельный модем, и т.д.), или комбинацию обоих, которые являются подходящими для того, чтобы получить доступ к закодированным видео данным, хранящимся на файловом сервере. Передача закодированных видео данных от файлового сервера может быть потоковой передачей, передачей загрузки, или комбинацией обеих.

[0046] Способы настоящего раскрытия не ограничены беспроводными приложениями или параметрами настройки. Способы могут быть применены к кодировке видео в поддержку множества мультимедийных приложений, таких как эфирное телевидение, передачи кабельного телевидения, передачи спутникового телевидения, потоковые передачи видео, например, через Интернет, кодирование видео данных для хранения на запоминающем носителе данных, декодирование видео данных, хранящихся на запоминающем носителе данных, или других приложениях. В некоторых примерах система 10 кодирования видео может быть сконфигурирована, чтобы поддерживать одностороннюю или двухстороннюю передачу видео, чтобы поддерживать приложения, такие как потоковая передача видео, воспроизведение видео, вещание видео, и/или видео телефония.

[0047] Фиг. 1 является просто примером, и способы настоящего раскрытия могут относиться к параметрам настройки кодирования видео (например, кодированию видео или декодированию видео), которые не обязательно включают в себя какую-либо передачу данных между устройствами кодирования и декодирования. В других примерах данные извлекают из локальной памяти, передают в виде потока по сети, или подобному. Устройство кодирования видео может закодировать и сохранять данные в памяти, и/или устройство декодирования видео может извлечь и декодировать данные из памяти. Во многих примерах кодирование и декодирование выполняются устройствами, которые не обмениваются друг с другом, но просто кодируют данные в память и/или извлекают и декодируют данные из памяти.

[0048] В примере согласно Фиг. 1 исходное устройство 12 включает в себя видео источник 18, видео кодер 20 и интерфейс 22 вывода. В некоторых примерах интерфейс

22 вывода может включать в себя модулятор/демодулятор (модем) и/или передатчик. Видео источник 18 может включать в себя устройство захвата видео, например, видео камеру, видео архив, содержащий ранее захваченные видео данные, интерфейс подачи видео, чтобы принять видео данные от поставщика видео контента, и/или систему компьютерной графики для того, чтобы генерировать видео данные, или комбинацию таких источников видео данных.

[0049] Видео кодер 20 может закодировать видео данные из видео источника 18. В некоторых примерах исходное устройство 12 непосредственно передает закодированные видео данные к устройству 14 назначения через интерфейс 22 вывода. В других примерах закодированные видео данные могут также храниться на запоминающем носителе или файловом сервере для более позднего доступа устройством 14 назначения для декодирования и/или воспроизведения.

[0050] В примере согласно Фиг. 1 устройство 14 назначения включает в себя интерфейс 28 ввода, видео декодер 30 и устройство 32 отображения. В некоторых примерах интерфейс 28 ввода включает в себя приемник и/или модем. Интерфейс 28 ввода может принять закодированные видео данные по каналу 16. Устройство 32 отображения может быть интегрировано с или может быть внешним к устройству 14 назначения. Вообще, устройство 32 отображения отображает декодированные видео данные. Устройство 32 отображения может содержать множество устройств отображения, таких как жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих (OLED) диодах, или другой тип устройства отображения.

[0051] Видео кодер 20 и видео декодер 30 каждый может быть реализован как любое из множества подходящих схем, таких как один или более микропроцессоров, цифровые сигнальные процессоры (DSPs), специализированные интегральные схемы (ASICs), программируемые пользователем вентильные матрицы (FPGAs), дискретная логика, аппаратное обеспечение, или любые их комбинации. Если способы реализованы частично в программном обеспечении, устройство может сохранить инструкции для программного обеспечения в подходящем невременном считываемом компьютером запоминающий носителе и может выполнить инструкции в аппаратном обеспечении, используя один или более процессоров, чтобы выполнить способы настоящего раскрытия. Любое из вышеописанного (включая аппаратное обеспечение, программное обеспечение, комбинацию аппаратного обеспечения и программного обеспечения, и т.д.), как могут полагать, является одним или более процессорами. Каждый из видео кодера 20 и видео декодера 30 может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть объединенного кодера/декодера (кодек) в соответствующем устройстве.

[0052] Настоящее раскрытие может в целом ссылаться на видео кодер 20, «сигнализирующий» некоторую информацию на другое устройство, такое как видео декодер 30 или дополнительное устройство 21. Термин «сигнализация» может вообще относиться к передаче элементов синтаксиса и/или других данных, используемых для декодирования сжатых видео данных. Такая передача может иметь место в реальном или почти реальном времени. Альтернативно, такая передача может иметь место в течение промежутка времени, например, может произойти при сохранении элементов синтаксиса в считываемом компьютером запоминающем носителе в закодированном потоке битов во время кодирования, которые затем могут быть извлечены устройством декодирования в любое время, будучи сохраненным на этом носителе.

[0053] В некоторых примерах видео кодер 20 и видео декодер 30 работают согласно стандарту сжатия видео, такому как ISO/IEC MPEG-4 Visual и ITU-T H.264 (также

известный как ISO/IEC MPEG-4 AVC), включая его расширение масштабируемого кодирования видео (SVC), расширение кодирования видео множественных видов (MVC), и/или основанное на MVC расширение 3DV. В некоторых случаях любой поток битов, соответствующий основанному на MVC расширению 3DV, всегда содержит подпоток битов, который совместим с профилем MVC, например, профилем высококачественного стерео. Кроме того, продолжают усиливаться усилия сгенерировать расширение кодирования трехмерного видео (3DV) для H.264/AVC, а именно, основанное на AVC 3DV. В других примерах видео кодер 20 и видео декодер 30 могут работать согласно ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 или ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, и ITU-T H.264, ISO/IEC Visual.

[0054] В других примерах видео кодер 20 и видео декодер 30 могут работать согласно стандарту высокоэффективного кодирования видео (HEVC), в настоящее время развиваемого объединенной командой сотрудничества по кодированию видео (JCT-VC) группы экспертов по кодированию видео ITU-T (VCEG) и ISO/IEC группы экспертов по движущимся изображениям (MPEG). Проект развивающегося стандарта HEVC, названного «HEVC, Рабочий Проект 9» описан в Bross et al., «High Efficiency Video Coding (HEVC) text specification draft 9,» Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting, Shanghai, China, октябрь 2012, который на 8 мая 2013, доступен по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zip. Кроме того, продолжают усиливаться усилия создать расширения SVC, кодирования множественных видов, и 3DV для HEVC. 3DV расширение HEVC может упоминаться Основанное на HEVC 3DV или 3D-HEVC.

[0055] В HEVC и других стандартах кодирования видео, видео последовательность типично включает в себя последовательность картинок. Картинки могут также упоминаться «как кадры». Картинка может включать в себя три массива выборок, обозначенных S_L , S_{Cb} и S_{Cr} . S_L является двумерным массивом (то есть, блоком) выборок яркости. S_{Cb} является двумерным массивом выборок цветности C_b . S_{Cr} является двумерным массивом выборок цветности C_r . Выборки цветности могут также быть упомянуты здесь как выборки «насыщенности цвета». В других случаях картинка может быть одноцветной и может только включать в себя массив выборок яркости.

[0056] Чтобы генерировать закодированное представление картинки, видео кодер 20 может генерировать набор единиц дерева кодирования (единиц STU). Каждая из единиц STU может быть блоком дерева кодирования выборок яркости, двумя соответствующими блоками дерева кодирования выборок насыщенности цвета, и синтаксической структурой, используемой для кодирования выборок блоков дерева кодирования. Блок дерева кодирования может быть блоком из $N \times N$ выборок. STU может также упоминаться как «блок дерева» или «наибольшая единица кодирования» (LCU). Единицы STU HEVC могут быть широким аналогом макроблоков других стандартов, таких как H.264/AVC. Однако, STU не обязательно ограничена конкретным размером и может включать в себя одну или более единиц кодирования (единиц CU). Вырезка может включать в себя целое количество единиц STU, упорядоченных последовательно в растровом сканировании.

[0057] Чтобы сгенерировать закодированную STU, видео кодер 20 может рекурсивно выполнять разделение квадродерева в отношении блоков дерева кодирования единицы STU, чтобы разделить блоки дерева кодирования на блоки кодирования, отсюда название «единицы дерева кодирования». Блок кодирования является блоком из $N \times N$ выборок. CU может быть блоком кодирования выборок яркости и двух соответствующих

блоков кодирования выборок насыщенности цвета картинки, которая имеет массив выборок яркости, массив выборок S_b и массив выборок S_r , и синтаксические структуры, используемые для кодирования выборок блоков кодирования. Видео кодер 20 может разделить блоки кодирования единицы CU в один или более блоков предсказания. Блок предсказания может быть прямоугольным (то есть, квадратным или неквадратным) блоком выборок, в отношении которых применяется одно и то же предсказание. Единица предсказания (PU) единицы CU может быть блоком предсказания выборок яркости, двумя соответствующими блоками предсказания выборок насыщенности цвета картинки, и синтаксическими структурами, использованными для предсказания выборок блока предсказания. Видео кодер 20 может генерировать предсказывающие блоки яркости, S_b и S_r для блоков предсказания яркости, S_b и S_r каждой PU в CU.

[0058] Видео кодер 20 может использовать внутреннее предсказание или внешнее предсказание, чтобы генерировать предсказывающие блоки для PU. Если видео кодер 20 использует внутреннее предсказание, чтобы сгенерировать предсказывающие блоки PU, видео кодер 20 может генерировать предсказывающие блоки PU на основании декодированных выборок картинки, ассоциированной с PU.

[0059] Если видео кодер 20 использует внешнее предсказание, чтобы генерировать предсказывающие блоки единицы PU, видео кодер 20 может генерировать предсказывающие блоки PU на основании декодированных выборок одной или более картинок, отличных от картинки, ассоциированной с PU. Видео кодер 20 может использовать однонаправленное предсказание или би-предсказание, чтобы генерировать предсказывающие блоки единицы PU. Когда видео кодер 20 использует однонаправленное предсказание, чтобы генерировать предсказывающие блоки для PU, PU может иметь единственный вектор движения. Когда видео кодер 20 использует би-предсказание, чтобы генерировать предсказывающие блоки для PU, PU может иметь два вектора движения.

[0060] После того, как видео кодер 20 генерирует предсказывающие блоки яркости, S_b и S_r для одной или более единиц PU в CU, видео кодер 20 может генерировать остаточный блок яркости для CU. Каждая выборка в остаточном блоке яркости в CU указывает разность между выборкой яркости в одном из предсказывающих блоков яркости CU и соответствующей выборкой в первоначальном блоке кодирования яркости в CU. Кроме того, видео кодер 20 может генерировать остаточный блок S_b для CU. Каждая выборка в остаточном блоке S_b в CU может указывать разность между выборкой S_b в одном из предсказывающих блоков S_b в CU и соответствующей выборкой в первоначальном блоке кодирования S_b в CU. Видео кодер 20 может также генерировать остаточный блок S_r для CU. Каждая выборка в остаточном блоке S_r в CU может указывать разность между выборкой S_r в одном из предсказывающих блоков S_r в CU и соответствующей выборкой в первоначальном блоке кодирования S_r в CU.

[0061] Кроме того, видео кодер 20 может использовать разделение квадродерева, чтобы выполнить разложение остаточных блоков яркости, S_b и S_r в CU в один или более блоков преобразования яркости, S_b и S_r . Преобразованный блок может быть прямоугольным блоком выборок, в отношении которого применяется одно и то же преобразование. Единица преобразования (TU) в CU может быть блоком преобразования выборок яркости, двумя соответствующими блоками преобразования выборок насыщенности цвета, и синтаксическими структурами, использованными для преобразования выборок блока преобразования. Таким образом, каждая TU в CU может быть ассоциирована с блоком преобразования яркости, блоком преобразования S_b , и блоком преобразования S_r . Блок преобразования яркости, ассоциированный с

TU, может быть суб-блоком остаточного блока яркости CU. Блок преобразования Sb может быть суб-блоком остаточного блока Sb в CU. Блок преобразования Sg может быть суб-блоком остаточного блока Sg в CU.

[0062] Видео кодер 20 может применять одно или более преобразований к блоку преобразования яркости в TU, чтобы генерировать блок коэффициентов яркости для TU. Блок коэффициентов может быть двумерным массивом коэффициентов преобразования. Коэффициент преобразования может быть скалярным числом. Видео кодер 20 может применять одно или более преобразований к блоку преобразования Sb в TU, чтобы генерировать блок коэффициентов Sb для TU. Видео кодер 20 может применять одно или более преобразований к блоку преобразования Sg в TU, чтобы генерировать блок коэффициентов Sg для TU.

[0063] После генерирования блока коэффициентов (например, блока коэффициентов яркости, блока коэффициентов Sb или блока коэффициентов Sg), видео кодер 20 может квантовать блок коэффициентов. Квантование вообще относится к процессу, в котором коэффициенты преобразования квантуются, чтобы возможно уменьшить величину данных, использованных для представления коэффициентов преобразования, обеспечивая дальнейшее сжатие. После того, как видео кодер 20 квантует блок коэффициентов, видео кодер 20 может энтропийно кодировать элементы синтаксиса, указывающие квантованные коэффициенты преобразования. Например, видео кодер 20 может выполнить контекстно-адаптивное двоичное арифметическое кодирование (CABAC) в отношении элементов синтаксиса, указывающих квантованные коэффициенты преобразования. Видео кодер 20 может вывести энтропийно кодированные элементы синтаксиса в потоке битов.

[0064] Видео кодер 20 может вывести поток битов, который включает в себя последовательность битов, которая формирует представление закодированных картинок, и ассоциированные данные. Поток битов может содержать последовательность единиц уровня абстракции сети (NAL). Единица NAL может быть синтаксической структурой, содержащей индикацию типа данных, которые должны следовать, и байты, содержащие эти данные в форме полезных данных необработанной последовательности байтов (RBSP), вставленных по мере необходимости с байтами предотвращения эмуляции. Таким образом, каждая из единиц NAL может включать в себя заголовок единицы NAL и инкапсулировать RBSP. Заголовок единицы NAL может включать в себя элемент синтаксиса, который указывает код типа единицы NAL. Код типа единицы NAL, заданный заголовком единицы NAL единицы NAL, указывает тип единицы NAL. RBSP может быть синтаксической структурой, содержащей целое число байтов, которое инкапсулировано в пределах единицы NAL. В некоторых случаях RBSP включает в себя нулевые биты.

[0065] Различные типы единиц NAL могут инкапсулировать различные типы RBSPs. Например, первый тип единицы NAL может инкапсулировать RBSP для набора параметров картинки (PPS), второй тип единицы NAL может инкапсулировать RBSP для закодированной вырезки, третий тип единицы NAL может инкапсулировать RBSP для SEI, и так далее. Единицы NAL, которые инкапсулируют RBSPs для данных кодирования видео (в противоположность RBSPs для наборов параметров и сообщений SEI) могут упоминаться как уровень кодирования видео (VCL) единицы NAL.

[0066] Видео декодер 30 может принять поток битов, генерируемый видео кодером 20. Кроме того, видео декодер 30 может синтаксически разобрать поток битов, чтобы декодировать элементы синтаксиса из потока битов. Видео декодер 30 может восстановить картинки видео данных на основании, по меньшей мере частично,

элементов синтаксиса, декодированных из потока битов. Процесс для восстановления видео данных может быть в целом обратным процессу, выполненному видео кодером 20. Например, видео декодер 30 может использовать вектора движения единиц PU, чтобы определить предсказывающие блоки для единиц PU текущей CU. Кроме того, видео декодер 30 может обратно квантовать блоки коэффициентов преобразования, ассоциированные с единицами TU текущей CU. Видео декодер 30 может выполнять обратные преобразования в отношении блоков коэффициентов преобразования, чтобы восстановить блоки преобразования, ассоциированные с единицами TU текущей CU. Видео декодер 30 может восстановить блоки кодирования текущей CU посредством суммирования выборок предсказывающих блоков для единиц PU текущей CU с соответствующими выборками преобразованных блоков единиц TU текущей CU. Посредством восстановления блоков кодирования для каждой CU картинки, видео декодер 30 может восстановить картинку.

[0067] При кодировании множественных видов могут быть множественные виды одной и той же сцены с различных точек зрения. Термин «единица доступа» использован для обращения к набору картинок, которые соответствуют одному и тому же моменту времени. Таким образом, видео данные могут быть поняты как последовательность единиц доступа, появляющихся в течение времени. «Компонент вида» может быть закодированным представлением вида в единственной единице доступа. В настоящем раскрытии «вид» может относиться к последовательности компонентов вида, ассоциированных с одним и тем же идентификатором вида.

[0068] Кодирование множественных видео поддерживает предсказание между видами. Предсказание между видами подобно внешнему предсказанию, используемому в H.264/AVC и HEVC, и может использовать те же самые элементы синтаксиса. Однако, когда видео кодировщик выполняет предсказание между видами в отношении текущего видео блока (таком как PU), видео кодер 20 может использовать, в качестве опорной картинки, картинку, которая находится в той же единице доступа что и текущий видео блок, но в другом виде. Напротив, обычное внешнее предсказание использует картинки только в различных единицах доступа в качестве опорных картинок.

[0069] При кодировании множественных видов вид может упоминаться как «базовый вид», если видео декодер (например, видео декодер 30) может декодировать картинки в виде без ссылки на картинки в любом другом виде. При кодировании картинки в одном из небазовых видов кодировщик видео (такой как видео кодер 20 или видео декодер 30) может добавить картинку в список опорных картинок, если картинка находится в другом виде, но в пределах того же самого момента времени (то есть единицы доступа) в качестве картинки, которую в настоящее время кодирует видео кодировщик. Подобно другим опорным картинкам внешнего предсказания, видео кодировщик может вставить опорную картинку предсказания между видами в любой позиции списка опорных картинок.

[0070] Стандарты кодирования видео определяют модели буферизации видео. В H.264/AVC и HEVC модель буферизации упоминается как «гипотетический опорный декодер» или «HRD». В HEVC Working Draft 8 HRD описан в Приложении С.

[0071] HRD описывает, как данные должны быть буферизованы для декодирования и как декодированные данные буферизованы для вывода. Например, HRD описывает работу буфера кодированных картинок («CPB»), буфера декодированных картинок («DPB») и видео декодера. CPB является буфером «первый вошел – первый вышел», содержащим единицы доступа в порядке декодирования, заданном посредством HRD. DPB является буфером, хранящим декодированные картинки для ссылки,

переупорядочения вывода или задержки вывода, заданными посредством HRD.

Поведения CPB и DPB могут быть математически заданы. HRD может непосредственно наложить ограничения на тактирование, размеры буфера и частоты следования битов. Кроме того, HRD может косвенно наложить ограничения на различные характеристики

5 потока битов и статистику.

[0072] В H.264/AVC и HEVC соответствие (согласованность) потока битов и соответствие декодера заданы как части спецификации HRD. Другими словами, модель HRD определяет тесты, чтобы определить, соответствует ли поток битов стандарту и проверяет, чтобы определить, соответствует ли декодер стандарту. Хотя HRD называют

10 некоторым типом декодера, видео кодеры типично используют HRD, чтобы гарантировать соответствие потока битов, в то время как видео декодеры типично не нуждаются в HRD.

[0073] H.264/AVC и HEVC оба задают два типа потока битов или соответствия HRD, а именно, Тип I и Тип II. Поток битов Типа I является потоком единиц NAL, содержащим

15 только единицы NAL VCL и единицу NAL данных заполнения для всех единиц доступа в потоке битов. Поток битов Типа II является потоком единиц NAL, который содержит в дополнение к единицам NAL VCL и единицам NAL данных заполнения для всех единиц доступа в потоке битов, по меньшей мере одно из следующего: дополнительные единицы NAL не-VCL, отличные от единиц NAL данных заполнения; и все элементы синтаксиса

20 leading_zero_8bits, zero_byte, start_coded_prefix_one_3bytes, и trailing_zero_8bits, которые формируют поток байтов из потока единиц NAL.

[0074] Когда устройство выполняет тест соответствия потока битов, который определяет, соответствует ли поток битов стандарту кодирования видео, устройство может выбрать рабочую точку потока битов. Устройство может затем определить

25 набор параметров HRD, применимых к выбранной рабочей точке. Устройство может использовать набор параметров HRD, применимых к выбранной рабочей точке, чтобы конфигурировать поведение HRD. Более подробно, устройство может использовать набор параметров HRD, применимый, чтобы конфигурировать поведения конкретных компонентов HRD, таких как планировщик гипотетического потока (HSS), CPB, процесс

30 декодирования, DPB, и так далее. Затем HSS может ввести закодированные видео данные потока битов в CPB упомянутого HRD согласно конкретному списку.

[0075] Кроме того, устройство может вызвать процесс декодирования, который декодирует закодированные видео данные в CPB. Процесс декодирования может вывести

35 декодированные картинки в DPB. Когда устройство перемещает данные через HRD, устройство может определить, остается ли конкретный набор ограничений удовлетворенным. Например, устройство может определить, имеется ли условие переполнения или потери значимости в CPB или DPB, в то время как HRD декодирует представление рабочей точки выбранной рабочей точки. Устройство может выбрать и обработать каждую рабочую точку потока битов в этом способе. Если никакая

40 рабочая точка потока битов не вызывает нарушение ограничений, устройство может определить, что поток битов соответствует стандарту кодирования видео.

[0076] И H.264/AVC и HEVC определяют два типа соответствия декодера, а именно, соответствие декодера тактированию вывода и соответствие декодера порядку вывода. Декодер, требующий соответствия конкретному профилю, ярусу и уровню, в состоянии

45 успешно декодировать все потоки битов, которые соответствуют требованиям соответствия потока битов стандарта кодирования видео, такого как HEVC. В

настоящем раскрытии «профиль» может относиться к поднабору синтаксиса потока битов. «Ярусы» и «уровни» могут быть определены в пределах каждого профиля.

Уровень яруса может быть заданным набором ограничений, наложенных на значения элементов синтаксиса в потоке битов. Эти ограничения могут быть простыми пределами в отношении значений. Альтернативно, они могут принять форму ограничений на арифметические комбинации значений (например, ширина картинки, умноженная на высоту картинки, умноженная на количество картинок, декодируемых в секунду).
 5 Уровень, заданный для нижнего яруса, является более ограниченным чем уровень, заданный для более высокого яруса.

[0077] Когда устройство выполняет тест соответствия декодера, чтобы определить, соответствует ли подвергаемый тесту декодер (DUT) стандарту кодирования видео, устройство может обеспечить как в HRD так и в DUT поток битов, который
 10 соответствует стандарту кодирования видео. HRD может обработать поток битов способом, описанным выше относительно теста соответствия потока битов. Устройство может определить, что DUT соответствует стандарту кодирования видео, если порядок декодированных картинок, выведенных посредством DUT, соответствует порядку
 15 декодированных картинок, выведенных посредством HRD. Кроме того, устройство может определить, что DUT соответствует стандарту кодирования видео, если тактирование, с которым DUT выводит декодированные картинки, соответствует тактированию, с которым HRD выводит декодированные картинки.

[0078] В дополнение к тестам соответствия потока битов и тестам соответствия декодера, устройства могут использовать параметры HRD в других целях. В качестве
 20 примеров начальные задержки удаления из CPB могут быть использованы для выдачи команды системе установить подходящую начальную сквозную задержку, и времена вывода из DPB могут быть использованы для выведения отметки времени протокола в реальном времени (RTP), когда видео поток битов данных транспортируется по RTP.

[0079] В моделях H.264/AVC и HEVC HRD декодирование или удаление из CPB может быть основанным на единице доступа. То есть, HRD, как предполагается, декодирует
 25 полные единицы доступа в одно время и удаляет полные единицы доступа из CPB. Кроме того, в моделях H.264/AVC и HEVC HRD предполагается, что декодирование картинок является мгновенным. Видео кодер 20 может сигнализировать в сообщениях SEI тактирования картинок времена декодирования, чтобы начать декодировать единиц
 30 доступа. В практических применениях, если соответствующий видео декодер строго следует временам декодирования, сигнализированным, чтобы начать декодировать единицы доступа, самое раннее время, чтобы вывести конкретную декодированную картинку, равно времени декодирования этой конкретной картинке плюс время,
 35 необходимое для декодирования этой конкретной картинке. Однако, в реальном мире время, необходимое для декодирования картинке, не может быть равным нулю.

[0080] Параметры HRD могут управлять различными аспектами HRD. Другими словами, HRD может полагаться на параметры HRD. Параметры HRD могут включать в себя начальную задержку удаления из CPB, размер CPB, частоту следования битов,
 40 начальную задержку вывода из DPB, и размер DPB. Видео кодер 20 может сигнализировать эти параметры HRD в синтаксической структуре `hrd_parameters()`, определенной в наборе параметров видео (VPS) и/или наборе параметров последовательности (SPS). Индивидуальные VPS и/или SPS могут включать в себя множественные синтаксические структуры `hrd_parameters()` для различных наборов
 45 параметров HRD. В некоторых примерах видео кодер 20 может сигнализировать параметры HRD в сообщении SEI периода буферизации или сообщении SEI тактирования картинок.

[0081] Как объяснено выше, рабочая точка потока битов ассоциирована с набором

идентификаторов уровня (то есть, набором значений `nuh_reserved_zero_6bits`) и временным идентификатором. Представление рабочей точки может включать в себя каждую единицу NAL, которая ассоциирована с рабочей точкой. Представление рабочей точки может иметь отличную частоту кадров и/или частоту следования битов, чем первоначальный поток битов. Это имеет место потому, что представление рабочей точки может не включать в себя некоторые картинки и/или некоторые из данных первоначального потока битов. Следовательно, если видео декодер 30 должен был удалить данные из CPB и/или DPB при конкретной скорости передачи, обрабатывая первоначальный поток битов, и если видео декодер 30 должен был удалить данные из CPB и/или DPB с той же частотой следования битов при обработке представления рабочей точки, видео декодер 30 может удалить слишком много или слишком мало данных из CPB и/или DPB. Соответственно, видео кодер 20 может сигнализировать различные наборы параметров HRD для различных рабочих точек. Например, видео кодер 20 может включать, в VPS, множественные синтаксические структуры `hrd_parameters ()`, которые включают в себя параметры HRD для различных рабочих точек.

[0082] В HEVC Working Draft 8 набор параметров HRD необязательно включает в себя набор информации, который характерен для всех подуровней. Другими словами, набор параметров HRD может необязательно включать в себя набор общих элементов синтаксиса, которые применимы к рабочим точкам, которые включают в себя любые временные подуровни. Временный подуровень является временным масштабируемым уровнем временного масштабируемого потока битов, состоящего из единиц NAL VCL с конкретным значением `TemporalId` и ассоциированных единиц NAL не-VCL. В дополнение к набору общей информации наборы параметров HRD могут включать в себя набор элементов синтаксиса, которые являются специфичными для индивидуальных временных подуровней. Например, синтаксическая структура `hrd_parameters ()` может необязательно включать в себя набор информации, которая характерна для всех подуровней и всегда включает в себя специфичную для подуровня информацию. Поскольку набор общей информации является общим для множественных наборов параметров HRD, может быть ненужным сигнализировать набор общей информации во множественных наборах параметров HRD. Вместо этого в HEVC Working Draft 8 общая информация может присутствовать в наборе параметров HRD, когда набор параметров HRD является первым набором параметров HRD в VPS, или общая информация может присутствовать в наборе параметров HRD, когда набор параметров HRD ассоциирован с индексом первой рабочей точки. Например, HEVC, Working Draft 8 поддерживает присутствие общей информации, когда любая синтаксическая структура `hrd_parameters ()` является первой синтаксической структурой `hrd_parameters ()` в VPS или когда синтаксическая структура `hrd_parameters ()` ассоциирована с первой рабочей точкой.

[0083] Таблица 1, ниже, является примерной синтаксической структурой для синтаксической структуры `hrd_parameters ()` в HEVC.

Параметры HRD

ТАБЛИЦА 1

	hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Дескриптор
	if(commonInfPresentFlag) {	
5	timing_info_present_flag	u(1)
	if(timing_info_present_flag) {	
	num_units_in_tick	u(32)
	time_scale	u(32)
	}	
	nal_hrd_parameters_present_flag	u(1)
10	vcl_hrd_parameters_present_flag	u(1)
	if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag){	
	sub_pic_cpb_params_present_flag	u(1)
	if(sub_pic_cpb_params_present_flag) {	
	tick_divisor_minus2	u(8)
15	du_cpb_removal_delay_length_minus1	u(5)
	}	
	bit_rate_scale	u(4)
	cpb_size_scale	u(4)
	initial_cpb_removal_delay_length_minus1	u(5)
	cpb_removal_delay_length_minus1	u(5)
20	dpb_output_delay_length_minus1	u(5)
	}	
	for(i = 0; i <= MaxNumSubLayersMinus1; i++) {	
	fixed_pic_rate_flag[i]	u(1)
25	if(fixed_pic_rate_flag[i])	
	pic_duration_in_tc_minus1[i]	ue(v)
	low_delay_hrd_flag[i]	u(1)
	cpb_cnt_minus1[i]	ue(v)
	if(nal_hrd_parameters_present_flag)	
	sub_layer_hrd_parameters(i)	
30	if(vcl_hrd_parameters_present_flag)	
	sub_layer_hrd_parameters(i)	
	}	
	}	

[0084] В примере из Таблицы 1 выше и других таблиц синтаксиса настоящего раскрытия, элементы синтаксиса с дескриптором типа ue (v) могут быть целыми числами переменной длины без знака, закодированные, используя экспоненциальное кодирование Голомба (Exp-Golomb) $0^{-\text{ГО}}$ порядка с первым левым битом. В примере из Таблицы 1 и следующих таблиц, элементы синтаксиса, имеющие дескрипторы формы u(n), где n - неотрицательное целое число, являются значениями без знака длины n.

[0085] В примерном синтаксисе из Таблицы 1, элементы синтаксиса в блоке «if (commonInfPresentFlag) { ... }» являются общей информацией наборов параметров HRD. Другими словами, общая информация набора параметров HRD может включать в себя элементы синтаксиса timing_info_present_flag, num_units_in_tick, time_scale, nal_hrd_parameters_present_flag, vcl_hrd_parameters_present_flag, sub_pic_cpb_params_present_flag, tick_divisor_minus2, du_cpb_removal_delay_length_minus1, bit_rate_scale, cpb_size_scale, initial_cpb_removal_delay_length_minus1, cpb_removal_delay_length_minus1, и dpb_output_delay_length_minus1.

[0086] Кроме того, в примере из Таблицы 1 элементы синтаксиса fixed_pic_rate_flag

[i], pic_duration_in_tc_minus1 [i], low_delay_hrd_flag [i], и cpb_cnt_minus1 [i] могут быть набором специфичных для подуровня параметров HRD. Другими словами, эти элементы синтаксиса синтаксической структуры hrd_parameters () могут быть применимы только к рабочим точкам, которые включают в себя конкретный подуровень. Таким образом, параметры HRD синтаксической структуры hrd_parameters () могут включать в себя, в дополнение к необязательно включенной общей информации, набор специфичных для подуровня HRD параметров, который является специфичным (присущим) для конкретного подуровня потока битов.

[0087] Элемент синтаксиса fixed_pic_rate_flag [i] может указывать, что, когда HighestTid равен i, временное расстояние между временами вывода HRD любых двух последовательных картинок в порядке вывода ограничено специальным образом. HighestTid может быть переменной, которая идентифицирует наивысший временный подуровень (например, для рабочей точки). Элемент синтаксиса pic_duration_in_tc_minus1 [i] может задавать, когда HighestTid равен i, временное расстояние, в тактах системных тактовых сигналов, между временами вывода HRD любых последовательных картинок в порядке вывода в закодированной видео последовательности. Элемент синтаксиса low_delay_hrd_flag [i] может задавать режим операции HRD, когда HighestTid равен i, как определено в Приложении C в HEVC Working Draft 8. Элемент синтаксиса cpb_cnt_minus1 [i] может задавать количество альтернативных спецификаций CPB в потоке битов закодированной видео последовательности, когда HighestTid равен i.

[0088] Видео кодер 20 может использовать сообщения SEI, чтобы включать в поток битов метаданные, которые не требуются для корректного декодирования значений выборок картинок. Однако, видео декодер 30 или другие устройства могут использовать метаданные, включенные в сообщения SEI, в различных других целях. Например, видео декодер 30 может использовать метаданные в сообщениях SEI для тактирования вывода картинок, отображения картинок, обнаружения потерь, и маскирования ошибок.

[0089] Видео кодер 20 может включать одну или более единиц NAL SEI в единицу доступа. Другими словами, любое количество единиц NAL SEI может быть ассоциировано с единицей доступа. Кроме того, каждая единица NAL SEI может содержать одно или более сообщений SEI. Стандарт HEVC описывает синтаксис и семантику для различных типов сообщений SEI. Однако, стандарт HEVC не описывает обработку сообщений SEI, потому что сообщения SEI не влияют на нормативный процесс декодирования. Одна причина иметь сообщения SEI в стандарте HEVC состоит в том, чтобы разрешить дополнительные данные, интерпретируемые тождественно в различных системах, использующих HEVC. Спецификации и системы, использующие HEVC, могут требовать, чтобы видео кодеры генерировали некоторые сообщения SEI, или могут определять конкретную обработку конкретных типов принятых сообщений SEI. Таблица 2 ниже перечисляет сообщения SEI, определенные в HEVC, и кратко описывает их назначение.

ТАБЛИЦА 2

Краткий обзор сообщений SEI	
Сообщение SEI	Назначение
Период буферизации	Начальные задержки для работы гипотетического опорного декодера (HRD)
Тактирование картинок	Время вывода картинки и время удаления картинки / суб-картинки для работы HRD
Прямоугольник панорамного сканирования	Отображение с отличным соотношением сторон картинки (PAR), чем PAR выведенных картинок
Полезные данные заполнения	Регулируют частоту следования битов для удовлетворения конкретных ограничений

	Пользовательские данные зарегистрированные Пользовательские данные незарегистрированные	Сообщения SEI, которые должны быть заданы внешними объектами
	Точка восстановления	Дополнительная информация для чистого произвольного доступа. Постепенное обновление декодирования.
5	Информация сцены	Информация об изменениях и переходах сцены
	Полнокадровый снимок	Индикация для пометки ассоциированной декодированной картинке как неподвижного изображения видео контента
	Сегмент прогрессивной детализации	Указывает, что некоторые последовательные картинки представляют прогрессивное улучшение качества картинки, вместо движущегося изображения
	Характеристики зерна пленки	Позволяет декодерам синтезировать зерно пленки
	Предпочтения отображения фильтра удаления блочности	Рекомендует, должны ли отображенные картинки подвергаться обработке внутриконтурного фильтра удаления блочности
10	Подсказки пост-фильтра	Обеспечивает предлагаемые коэффициенты пост-фильтра или информацию корреляции для построения пост-фильтра
	Информация отображения тональных сигналов	Переотображение в отличное цветовое пространство, чем то, что используется или предполагается при кодировании
	Компоновка упаковки кадра	Упаковка стереоскопического видео в поток битов HEVC
	Ориентация дисплея	Задаёт переворачивание и/или вращение, которое должно быть применено к выходным картинкам, когда они отображаются
15		
	Индикация поля	Обеспечивает информацию, относящуюся к перемежаемому видео контенту и/или кодированию полей, например, указывает, является ли картинка кадром прогрессивной развертки, полукадром или кадром, содержащим два перемежаемых полукадра
	Хэш декодированной картинке	Контрольная сумма декодированной картинке, которая может использоваться для обнаружения ошибок
20	Тактирование суб-картинок	Время удаления суб-картинок для работы HRD
	Наборы активных параметров	Обеспечивает информацию относительно активных VPS, SPS, и т.д.
	Структура описания картинке	Описывает структуру временного и внешнего предсказания потока битов

[0090] Имеются несколько проблем или недостатков с существующими методами для сигнализации параметров HRD и выбора параметров HRD и других параметров. Например, в HEVC Working Draft 8 только наборы параметров HRD в VPS выбирают для операций HRD. То есть, хотя параметры HRD могут быть предоставлены в наборах SPS, наборы параметров HRD в SPS не выбираются видео декодерами HEVC для операций HRD. Видео декодеры всегда синтаксически разбирают и декодируют VPS потока битов. Следовательно, видео декодеры всегда синтаксически разбирают и декодируют наборы параметров HRD в VPS.

[0091] Это верно независимо от того, включает ли поток битов в себя единицы NAL небазового уровня. Например, только синтаксическая структура `hrd_parameters()`, закодированная в VPS, может быть выбрана для операций HRD, и возможно присутствующая синтаксическая структура `hrd_parameters()` в SPS может никогда быть не выбрана. Это может потребовать синтаксического разбора и обработки VPS, даже при декодировании потока битов, который не содержит `nuh_reserved_zero_6bits`, большее чем 0 (то есть, поток битов содержит только базовый уровень в расширениях множественных видов, 3DV, или SVC в HEVC).

[0092] Таким образом, если поток битов включает в себя единицы NAL небазового уровня, может быть тратой вычислительных ресурсов синтаксически разобрать и обращаться с наборами параметров HRD в SPS. Кроме того, если наборы параметров HRD присутствуют в VPS, наборы параметров HRD в SPS могут быть потраченными впустую битами. Например, если синтаксическая структура `hrd_parameters()` присутствует в SPS, закодированные биты для этой синтаксической структуры могут быть просто тратой битов.

[0093] В соответствии с одним или более способами настоящего раскрытия, видео кодер 20 может генерировать поток битов, который включает в себя SPS, который применим к последовательности картинок. SPS включает в себя набор параметров

HRD. Набор параметров HRD применим к каждой рабочей точке потока битов, которая имеет набор идентификаторов уровня, который соответствует набору целевых идентификаторов уровня. Таким образом, наборы параметров HRD в SPS не расходуются впустую, а вместо этого могут использоваться для операций HRD.

5 Например, операция указывает, какая синтаксическая структура `hrd_parameters()`, закодированная в SPS, может быть ясно задана, например, чтобы быть рабочими точками, для которых только одно значение `nuh_reserved_zero_bbits` (то есть, ID уровня в расширение множественных видов, 3DV или масштабируемого кодирования видео) присутствует в потоке битов.

10 [0094] Например, устройство, такое как видео кодер 20 или видео декодер 30, может выбрать из числа набора параметров HRD в наборе параметров видео и наборе параметров HRD в SPS, набор параметров HRD, применимых к конкретной рабочей точке. В этом примере устройство может выполнять, на основании, по меньшей мере частично, набора параметров HRD, применимых к конкретной рабочей точке, тест

15 соответствия потока битов, который проверяет, соответствует ли поднабор потока битов, ассоциированный с конкретной рабочей точкой, стандарту кодирования видео. Тест соответствия потока битов может верифицировать, что представление рабочей точки соответствует стандарту кодирования видео, такому как HEVC.

[0095] В настоящем раскрытии рабочая точка может быть идентифицирована набором

20 значений `nuh_reserved_zero_bbits`, обозначенным как `OpLayerIdSet`, и значением `TemporalId`, обозначенным как `OpTid`. Ассоциированный поднабор потока битов, выведенный как выходной результат процесса извлечения подпотока битов, как задано в подпункте 10.1 HEVC Working Draft 8, с `OpTid` и `OpLayerIdSet` в качестве входов, являются независимо декодируемыми. Подпункт 10.1 HEVC Working Draft 8 описывает операцию для того,

25 чтобы извлечь подпоток битов (то есть, представление рабочей точки) из потока битов. В частности, подпункт 10.1 HEVC Working Draft 8 обеспечивает, что подпоток битов выводят посредством удаления из потока битов всех единиц NAL с временными идентификаторами (например, `TemporalID`), большими чем `tIdTarget`, или идентификаторами уровня (например, `nuh_reserved_zero_bbits`) не среди значений в

30 `targetDecLayerIdSet`. `tIdTarget` и `targetDecLayerIdSet` являются параметрами процесса извлечения потока битов.

[0096] В другой примерной проблеме или недостатке существующих методов для сигнализации параметров HRD, устройство, такие как видео кодер, видео декодер или

35 другой тип устройства, может выполнить тест соответствия потока битов в отношении представления рабочей точки для рабочей точки. Как упомянуто выше, набор целевых идентификаторов уровня и временный идентификатор могут быть использованы для идентификации рабочей точки. Набор целевых идентификаторов уровня может быть обозначен как «`TargetDecLayerIdSet`». Временный идентификатор может быть обозначен как «`TargetDecHighestTid`». Проблемой является то, что HEVC Working Draft 8 не задает

40 как `TargetDecLayerIdSet` или `TargetDecHighestTid` установлены при выполнении теста соответствия потока битов. Например, когда вызывается процесс декодирования для теста соответствия потока битов, семантика элементов синтаксиса явно не заданы как значения `TargetDecLayerIdSet`, и `TargetDecHighestTid` должным образом не установлены.

[0097] Один или более способов настоящего раскрытия указывают, как

45 `TargetDecLayerIdSet` и `TargetDecHighestTid` устанавливаются при выполнении теста соответствия потока битов. Например, общий процесс декодирования для потока битов (или представление рабочей точки) модифицируют таким образом, что, если поток битов (или представление рабочей точки) декодируют в тесте соответствия потока

битов, TargetDecLayerIdSet устанавливаются так, как задано в подпункте C.1 стандарта HEVC. Аналогично, общий процесс декодирования для потока битов (или представление рабочей точки) может быть модифицирован таким образом, что, если поток битов (или представление рабочей точки) декодируют в тесте соответствия потока битов,

5 TargetDecHighestTid устанавливаются так, как задано в подпункте C.1 HEVC Working Draft 8. Другими словами, устройство может определить целевой набор идентификаторов уровня конкретной рабочей точки, который содержит каждый идентификатор уровня, присутствующий в поднаборе потока битов, и этот набор идентификаторов уровня конкретной рабочей точки является поднабором идентификаторов уровня, присутствующих в потоке битов. Кроме того, устройство может определить целевой временный идентификатор конкретной рабочей точки, который равен наибольшему временному идентификатору, присутствующему в поднаборе потока битов, и целевой временный идентификатор конкретной рабочей точки меньше чем или равен

15 [0098] В подпункте C.1 HEVC Working Draft 8 TargetDecLayerIdSet устанавливаются в targetOpLayerIdSet. targetOpLayerIdSet содержит набор значений для nuh_reserved_zero_bbits, присутствующего в представлении рабочей точки для рабочей точки, подвергаемой тесту. targetOpLayerIdSet является поднабором значений для nuh_reserved_zero_bbits, присутствующего в потоке битов, подвергаемом тесту.

20 [0099] Кроме того, переменная TargetDecHighestTid идентифицирует наивысший временный подуровень, который должен быть декодирован. Временным подуровнем является временный масштабируемый уровень временного масштабируемого потока битов, состоящего из единиц NAL VCL с конкретным значением TemporalId и ассоциированными единицами NAL не-VCL. В подпункте C.1 стандарта HEVC TargetDecHighestTid устанавливаются в targetOpTid. targetOpTid равно наибольшему temporal_id, присутствующего в представлении рабочей точки для рабочей точки, подвергаемом тесту, и меньше чем или равен наибольшему temporal_id, присутствующему в потоке битов, подвергаемом тесту. Таким образом, когда процесс декодирования вызывается для теста соответствия потока битов, значения TargetDecLayerIdSet и TargetDecHighestTid устанавливаются в набор значений nuh_reserved_zero_bbits и самое

30 большое значение TemporalId, присутствующее в подпотоке битов, соответствующем рабочей точке, подвергаемой тесту для конкретного теста соответствия потока битов. [0100] Таким образом, устройство (такое как видео кодер 20, видео декодер 30, дополнительное устройство 21, или другое устройство), в соответствии с одним или

35 более способами настоящего раскрытия, может выполнить процесс декодирования в качестве части выполнения теста соответствия потока битов. Выполнение процесса декодирования может содержать выполнение процесса извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки для рабочей точки, определенной целевым набором идентификаторов уровня, и целевой наивысший временной

40 идентификатор. Целевой набор идентификаторов уровня (то есть, TargetDecLayerIdSet) содержит значения элементов синтаксиса идентификатора уровня (например, элементы синтаксиса nuh_reserved_zero_bbits), присутствующих в представлении рабочей точки. Целевой набор идентификаторов уровня является поднабором значений элементов синтаксиса идентификатора уровня потока битов. Целевой наивысший временный

45 идентификатор (то есть, TargetDecHighestTid) равен наибольшему временному идентификатору, присутствующему в представлении рабочей точки. Целевой наивысший временный идентификатор меньше чем или равен наибольшему временному идентификатору, присутствующему в потоке битов. Выполнение процесса декодирования

также содержит декодирование единиц NAL представления рабочей точки.

[0101] Процесс декодирования не всегда выполняется в качестве части выполнения теста соответствия потока битов. Вместо этого процесс декодирования может быть общим процессом для декодирования потока битов. Когда процесс декодирования не выполняется как часть теста соответствия потока битов, внешний источник может определить TargetDecLayerIdSet и TargetDecHighestTid для рабочей точки. Внешний источник может быть любым источником информации вне потока битов. Например, устройство CDN может программно определить и задать TargetDecLayerIdSet и TargetDecHighestTid на основании конфигурации конкретного видео декодера.

Устройство, выполняющее процесс декодирования может использовать внешне указанные TargetDecLayerIdSet и TargetDecHighestTid, чтобы извлечь представление рабочей точки из потока битов. Устройство, выполняющее процесс декодирования может затем декодировать единицы NAL извлеченного представления рабочей точки.

[0102] Таким образом, когда процесс декодирования не выполняется как часть теста соответствия потока битов, устройство, выполняющее процесс декодирования, может принять из внешнего источника целевой набор идентификаторов уровня и целевой наивысший временный идентификатор. Целевой набор идентификаторов уровня содержит значения элементов синтаксиса идентификатора уровня, присутствующих в представлении рабочей точки. Целевой наивысший временный идентификатор равен наибольшему временному идентификатору, присутствующему во втором представлении рабочей точки. Кроме того, устройство, выполняющее процесс декодирования, может выполнить процесс извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки. Устройство, выполняющее процесс декодирования, может затем декодировать единицы NAL представления рабочей точки.

[0103] В других случаях внешний источник не задает TargetDecLayerIdSet или TargetDecHighestTid. В таких случаях процесс декодирования может быть выполнен в отношении целого потока битов. Например, устройство может выполнить процесс извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки. В этом примере 0 является единственным значением элементов синтаксиса идентификатора уровня (то есть, nuh_reserved_zero_bbits), присутствующим в представлении рабочей точки. Кроме того, в этом примере наибольший временный идентификатор, присутствующий в потоке битов, равен наибольшему временному идентификатору, присутствующему в представлении рабочей точки. В этом примере устройство, выполняющее процесс декодирования, может декодировать единицы NAL представления рабочей точки.

[0104] Как указано выше, SPS может включать в себя массив элементов синтаксиса, обозначенных как sps_max_dec_pic_buffering [i], где i ранжируется от 0 до максимального количества временных уровней в потоке битов. sps_max_dec_pic_buffering [i] указывает максимальный требуемый размер DPB, когда наивысший временный идентификатор (HighestTid) равен i. sps_max_dec_pic_buffering [i] указывает требуемый размер в терминах единиц буферов хранения картинок. Кроме того, SPS может включать в себя массив элементов синтаксиса, обозначенных sps_max_num_reorder_pics [i], где i ранжируется от 0 до максимального количества временных уровней в потоке битов.

sps_max_num_reorder_pics [i] указывает максимальное разрешенное количество картинок, предшествующих любой картинке в порядке декодирования и следующих за этой картинкой в порядке вывода, когда наивысший временный идентификатор (HighestTid) равен i. Кроме того, набор параметров HRD может включать в себя массив элементов синтаксиса, обозначенный crb_cnt_minus1 [i], где i ранжируется от 0 до максимального

количества временных уровней в потоке битов. `sps_cnt_minus1 [i]` задает количество альтернативных спецификаций CPB в потоке битов закодированной видео последовательности, когда наивысший временный идентификатор (`HighestTid`) равен `i`.

5 [0105] Поскольку HEVC, Working Draft 8 не задает, что обозначается наивысшим временным идентификатором (`HighestTid`), HEVC Working Draft 8, `sps_max_dec_pic_buffering [i]`, `sps_max_num_reorder_pics [i]`, и `sps_cnt_minus1 [i]` должным образом не выбирают в операциях HRD, операциях соответствия потока битов, и ограничениях уровня. Другими словами, параметры `sps_max_num_reorder_pics [i]`,
10 `sps_max_dec_pic_buffering [i]`, и `sps_cnt_minus1 [i]` в операциях HRD, требующих соответствия потока битов и ограничения уровня, должным образом не выбираются.

[0106] В соответствии с одним или более способами настоящего раскрытия, `sps_max_dec_pic_buffering [i]` определен таким образом, что `sps_max_dec_pic_buffering [i]` указывает максимальный требуемый размер DPB, когда `TargetDecHighestTid` равен `i`. `TargetDecHighestTid` определен способом, описанным выше. Это может быть в
15 контрасте HEVC Working Draft 8, где `HighestTid` не определен. Значение `sps_max_dec_pic_buffering [i]` должно быть в диапазоне от 0 до `MaxDpbSize` (как определено в подпункте 4 HEVC Working Draft 8), включительно. Когда `i` будет больше чем 0, `sps_max_dec_pic_buffering [i]` должно быть равным или больше чем
20 `sps_max_dec_pic_buffering [i - 1]`. Значение `sps_max_dec_pic_buffering [i]` должно быть меньше чем или равно `vps_max_dec_pic_buffering [i]` для каждого значения `i`.

[0107] Аналогично, в соответствии с одним или более способами настоящего раскрытия, `sps_max_num_reorder_pics [i]` определен таким образом, что `sps_max_num_reorder_pics [i]` указывает максимальное разрешенное количество картинок,
25 предшествующих любой картинке в порядке декодирования и следующих за этой картинкой в порядке вывода, когда `TargetDecHighestTid` равен `i`. `TargetDecHighestTid` определен способом, описанным выше. Значение `sps_max_num_reorder_pics [i]` должно быть в диапазоне от 0 до `sps_max_dec_pic_buffering [i]`, включительно. Когда `i` больше чем 0, `sps_max_num_reorder_pics [i]` должно быть равным или больше чем
30 `sps_max_num_reorder_pics [i - 1]`. Значение `sps_max_num_reorder_pics [i]` должно быть меньше чем или равно `vps_max_num_reorder_pics [i]` для каждого значения `i`.

[0108] Кроме того, в соответствии с одним или более способами настоящего раскрытия, `sps_cnt_minus1 [i]` может задавать количество альтернативных спецификаций CPB в потоке битов закодированной видео последовательности, когда
35 `TargetDecHighestTid` равен `i`, где `i` ранжируется от 0 до максимального количества временных уровней в потоке битов. `TargetDecHighestTid` определен способом, описанным выше. Значение `sps_cnt_minus1 [i]` находится в диапазоне от 0 до 31, включительно. Когда `low_delay_hrd_flag [i]` равен 1, `sps_cnt_minus1 [i]` равно 0. Когда `sps_cnt_minus1 [i]` не присутствует, `sps_cnt_minus1 [i]` логически выводится, чтобы быть равным 0.

40 [0109] Таким образом, в соответствии с одним или более способами настоящего раскрытия, устройство может определить, на основании наивысшего временного идентификатора, конкретный элемент синтаксиса из числа массива элементов синтаксиса. Наивысший временный идентификатор определен таким образом, что наивысший временный идентификатор всегда идентифицирует наивысший временный уровень, который должен быть декодирован. Таким образом, `sps_max_num_reorder_pics [i]`,
45 `sps_max_dec_pic_buffering [i]`, и `sps_cnt_minus1 [i]` в операциях HRD, требующих соответствия потока битов и ограничения уровня, последовательно выбирают с `i`, равным явно заданному значению `TargetDecHighestTid`.

[0110] Таким образом, устройство (такой видео кодер 20, видео декодер 30, дополнительное устройство 21, или другое устройство) может выполнить операцию HRD, чтобы определить соответствие потока битов стандарту кодирования видео или определить соответствие видео декодера стандарту кодирования видео. В качестве части выполнения операции HRD, устройство может определить наивысший временный идентификатор поднабора потока битов, ассоциированный с выбранной рабочей точкой потока битов. Кроме того, устройство может определить, на основании наивысшего временного идентификатора, конкретный элемент синтаксиса из числа массива элементов синтаксиса (например, `sps_max_num_reorder_pics [i]`, `sps_max_dec_pic_buffering [i]`, или `crb_cnt_minus1 [i]`). Кроме того, устройство может использовать конкретный элемент синтаксиса в операции HRD.

[0111] Кроме того, в HEVC Working Draft 8, каждая из синтаксических структур `hrd_parameters ()` в VPS может быть ассоциирована с синтаксической структурой `operation_point_layer_ids ()` на основании того, какая синтаксическая структура `hrd_parameters ()` выбрана для использования в операциях HRD. Соответствующий каждой выбранной синтаксической структуре `hrd_parameters ()`, набор сообщений SEI периода буферизации и сообщений SEI тактирования картинок, также может быть необходим в операциях HRD. Однако, не имеется способа ассоциировать сообщение SEI периода буферизации или сообщение SEI тактирования картинок с синтаксической структурой `hrd_parameters ()`, для которой ассоциированная синтаксическая структура `operation_point_layer_ids ()` включает в себя множественные значения `nuh_reserved_zero_6bits` (то есть, ID множественных уровней в расширении множественных видов, 3DV или масштабируемого кодирования видео HEVC).

[0112] Решение этой проблемы может состоять в том, чтобы применить сообщение SEI масштабируемого вложения кодирования множественных видов, как задано в Приложении Н в H.264/AVC или подобном. Однако, сообщение SEI масштабируемого вложения кодирования множественных видов или подобные сообщения SEI, могут иметь следующие недостатки. Во-первых, так как единицы NAL SEI в H.264/AVC имеют только однобайтовый заголовок единицы NAL, не может иметь место способ использовать информацию, переносимую в `nuh_reserved_zero_6bits` и `temporal_id_plus1` в заголовке единицы NAL HEVC единицы NAL SEI для ассоциации сообщения SEI периода буферизации или тактирования картинок к рабочим точкам. Во-вторых, каждое вложенное сообщение SEI может быть ассоциировано только с одной рабочей точкой.

[0113] Один или более способов настоящего раскрытия могут обеспечить механизм, чтобы ясно задать рабочие точки, к которым сообщение SEI периода буферизации, сообщение SEI тактирования картинок или сообщение SEI тактирования суб-картинок применяется, с помощью синтаксической структуры `applicable_operation_points ()`, которую можно передать в сообщении SEI периода буферизации, сообщении SEI тактирования картинок или в сообщении SEI тактирования суб-картинок. Этот механизм может разрешить использование информации, переносимую в элементах синтаксиса `nuh_reserved_zero_6bits` и `temporal_id_plus1`, в заголовке единицы NAL единиц NAL SEI, и может разрешить совместное использование информации, переданной в одном и том же сообщении SEI периода буферизации, тактировании картинок или тактировании суб-картинок с помощью множественных рабочих точек.

[0114] Фиг. 2 является блок-схемой, иллюстрирующей примерный кодер видео 20, который может реализовать способы настоящего раскрытия. Фиг. 2 предоставлена в целях объяснения и не должна быть рассмотрена как ограничивающая способы, которые широко иллюстрируются и описаны в настоящем раскрытии. В целях объяснения

настоящее раскрытие описывает видео кодер 20 в контексте кодирования HEVC. Однако, способы настоящего раскрытия могут быть применимыми к другим стандартам или способам кодирования.

[0115] В примере согласно Фиг. 2, видео кодер 20 включает в себя модуль 100 обработки предсказания, модуль 102 генерирования остатка, модуль 104 обработки преобразования, модуль 106 квантования, модуль 108 обратного квантования, модуль 110 обработки обратного преобразования, модуль 112 реконструкции, модуль 114 фильтров, буфер 116 декодированных картинок, и модуль 118 энтропийного кодирования. Модуль 100 обработки предсказания включает в себя модуль 120 обработки внешнего предсказания и модуль 126 обработки внутреннего предсказания. Модуль 120 обработки внешнего предсказания включает в себя модуль 122 оценки движения и модуль 124 компенсации движения. В других примерах видео кодер 20 может включать в себя больше, меньше, или другие функциональные компоненты.

[0116] Видео кодер 20 может принять видео данные. Видео кодер 20 может закодировать каждую STU в вырезке картинки видео данных. Каждая из единиц STU может быть ассоциирована с одинакового размера блоками дерева кодирования (СТВ) яркости и соответствующими СТВ упомянутой картинки. Как часть кодирования STU, модуль 100 обработки предсказания может выполнить разделение квадродерева, чтобы разделить блоки СТВ в STU в прогрессивно меньшие блоки. Меньший блок может быть блоками кодирования единиц CU. Например, модуль 100 обработки предсказания может разделить СТВ, ассоциированный с STU, на четыре одинакового размера суб-блока, разделить один или более суб-блоков на четыре одинакового размера суб-суб-блока, и так далее.

[0117] Видео кодер 20 может закодировать единицы CU в STU, чтобы генерировать закодированные представления единиц CU (то есть, закодированные единицы CU). Как часть кодирования CU, модуль 100 обработки предсказания может разделить блоки кодирования, ассоциированные с CU среди одной или более единиц PU упомянутой CU. Таким образом, каждая PU может быть ассоциирована с блоком предсказания яркости и соответствующими блоками предсказания насыщенности цвета. Видео кодер 20 и видео декодер 30 могут поддерживать единицы PU, имеющие различные размеры. Размер CU может относиться к размеру блока кодирования яркости упомянутой CU, и размер PU может относиться к размеру блока предсказания яркости упомянутой PU. Предполагая, что размер конкретной CU равен $2N \times 2N$, видео кодер 20 и видео декодер 30 могут поддерживать размеры PU, равные $2N \times 2N$ или $N \times N$ для внутреннего предсказания, и симметричные размеры PU, равные $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, или подобные для внешнего предсказания. Видео кодер 20 и видео декодер 30 могут также поддерживать асимметричное разделение для размеров PU, равных $2N \times nU$, $2N \times nD$, $nL \times 2N$, и $nR \times 2N$ для внешнего предсказания.

[0118] Модуль 120 обработки внешнего предсказания может генерировать предсказывающие данные для PU, выполняя внешнее предсказание в отношении каждой PU в CU. Предсказывающие данные для PU могут включать в себя предсказывающие блоки PU и информацию движения для PU. Модуль 120 обработки внешнего предсказания может выполнить различные операции для PU в CU в зависимости от того, находится ли PU в I вырезке, P вырезке, или B вырезке. В I вырезке все единицы PU являются внутренне предсказанными. Следовательно, если PU находится в I вырезке, модуль 120 обработки внешнего предсказания не выполняет внешнее предсказание в отношении этой PU. Таким образом, для блоков, закодированных в I-режиме, предсказывающий блок формируют, используя пространственное предсказание из

ранее закодированных соседних блоков в пределах того же самого кадра.

[0119] Если PU находится в Р вырезке, модуль 122 оценки движения может искать опорные картинки в списке опорных картинок (например, «RefPicList0») для опорной области для PU. Опорная область для PU может быть областью в пределах опорной картинки, которая содержит блоки выборок, которые наиболее близко соответствуют блокам предсказания упомянутой PU. Модуль 122 оценки движения может генерировать опорный индекс, который указывает позицию в RefPicList0 опорной картинки, содержащей опорную область для PU. Кроме того, модуль 122 оценки движения может генерировать вектор движения, который указывает пространственное смещение между блоком предсказания PU и опорным местоположением, ассоциированным с опорной областью. Например, вектор движения может быть двумерным вектором, который обеспечивает смещение от координат в текущей картинке к координатам в опорной картинке. Модуль 122 оценки движения может вывести опорный индекс и вектор движения в качестве информации движения упомянутой PU. Модуль 124 компенсации движения может генерировать предсказывающие блоки PU на основании фактических или интерполированных выборок в опорном местоположении, обозначенном вектором движения этой PU.

[0120] Если PU находится в В вырезке, модуль 122 оценки движения может выполнить однонаправленное предсказание или би-предсказание для PU. Чтобы выполнить однонаправленное предсказание для PU, модуль 122 оценки движения может искать опорные картинки RefPicList0 или второго списка опорных картинок («RefPicList1») для опорной области для PU. Модуль 122 оценки движения может вывести, в качестве информации движения PU, опорный индекс, который указывает позицию в RefPicList0 или RefPicList1 опорной картинки, которая содержит опорную область, вектор движения, который указывает пространственное смещение между блоком выборок PU и опорным местоположением, ассоциированным с опорной областью, и один или более индикаторов направления предсказания, которые указывают, находится ли опорная картинка в RefPicList0 или RefPicList1. Модуль 124 компенсации движения может генерировать предсказывающие блоки PU на основании, по меньшей мере частично, фактических или интерполированных выборок в опорной области, обозначенной вектором движения этой PU.

[0121] Чтобы выполнить двунаправленное внешнее предсказание для PU, модуль 122 оценки движения может искать опорные картинки в RefPicList0 для опорной области для PU и может также искать опорные картинки в RefPicList1 для другой опорной области для этой PU. Модуль 122 оценки движения может генерировать опорные индексы, которые указывают позиции в RefPicList0 и RefPicList1 опорных картинок, которые содержат опорные области. Кроме того, модуль 122 оценки движения может генерировать вектора движения, которые указывают пространственные смещения между опорным местоположением, ассоциированным с опорными областями, и блоком выборок упомянутой PU. Информация движения PU может включать в себя опорные индексы и вектора движения PU. Модуль 124 компенсации движения может генерировать предсказывающие блоки PU на основании, по меньшей мере частично, фактических или интерполированных выборок в опорной области, обозначенной вектором движения этой PU.

[0122] Модуль 126 обработки внутреннего предсказания может генерировать предсказывающие данные для PU посредством выполнения внутреннего предсказания в отношении PU. Предсказывающие данные для PU могут включать в себя предсказывающие блоки для PU и различные элементы синтаксиса. Модуль 126

обработки внутреннего предсказания может выполнить внутреннее предсказание в отношении единиц PU в I вырезках, P вырезках, и B вырезках.

[0123] Чтобы выполнить внутреннее предсказание в отношении PU, модуль 126 обработки внутреннего предсказания может использовать множественные режимы внутреннего предсказания, чтобы генерировать множественные наборы предсказывающих данных для PU. Чтобы использовать режим внутреннего предсказания, чтобы генерировать набор предсказывающих данных для PU, модуль 126 обработки внутреннего предсказания может расширять выборки из блоков выборок соседних единиц PU «через» блоки выборок PU в направлении, ассоциированном с режимом внутреннего предсказания. Соседние единицы PU могут быть выше, выше и справа, выше и слева, или слева от PU, принимая порядок кодирования слева направо, сверху вниз для единиц PU, единиц CU и единиц STU. Модуль 126 обработки внутреннего предсказания может использовать различные количества режимов внутреннего предсказания, например, 33 направленных режима внутреннего предсказания. В некоторых примерах количество режимов внутреннего предсказания может зависеть от размера области, ассоциированной с PU.

[0124] Модуль 100 обработки предсказания может выбрать предсказывающие данные для единиц PU в CU из числа предсказывающих данных, генерируемых модулем 120 обработки внешнего предсказания для единиц PU или предсказывающих данных, генерируемых модулем 126 обработки внутреннего предсказания для единиц PU. В некоторых примерах модуль 100 обработки предсказания выбирает предсказывающие данные для единиц PU в CU на основании метрик «скорость передачи/искажение» наборов предсказывающих данных. Предсказывающие блоки выбранных предсказывающих данных могут быть упомянуты здесь как выбранные предсказывающие блоки.

[0125] Модуль 102 генерирования остатка может генерировать, на основании блока кодирования яркости, Cb и Cr упомянутой CU и выбранных предсказывающих блоков яркости, Cb и Cr единиц PU в CU, остаточные блоки яркости, Cb и Cr упомянутой CU. Например, модуль 102 генерирования остатка может генерировать остаточные блоки упомянутой CU таким образом, что каждая выборка в остаточных блоках имеет значение, равное разности между выборкой в блоке кодирования упомянутой CU, и соответствующей выборкой в соответствующем выбранном предсказывающем блоке PU упомянутой CU.

[0126] Модуль 104 обработки преобразования может выполнять разделение квадродерева, чтобы разделить остаточные блоки, ассоциированные с CU, в блоки преобразования, ассоциированные с единицами TU в CU. Таким образом, TU может быть ассоциирована с блоком преобразования яркости, и двумя блоками преобразования насыщенности цвета. Размеры и позиции блоков преобразования яркости и насыщенности цвета единицы TU в CU могут быть или могут не быть основаны на размерах и позициях блоков предсказания единиц PU в CU. Структура квадродерева, известная как «остаточное квадродерево» (RQT), может включать в себя узлы, ассоциированные с каждой из областей. Единицы TU в CU могут соответствовать листовым узлам RQT.

[0127] Модуль 104 обработки преобразования может генерировать блоки коэффициентов преобразования для каждой TU в CU посредством применения одного или более преобразований к блокам преобразования упомянутой TU. Модуль 104 обработки преобразования может применять различные преобразования к блоку преобразования, ассоциированному с TU. Например, модуль 104 обработки

преобразования может применить дискретное косинусное преобразование (DCT), направленное преобразование, или концептуально подобное преобразование к блоку преобразования. В некоторых примерах модуль 104 обработки преобразования не применяет преобразование к блоку преобразования. В таких примерах блок

5 преобразования можно рассматривать как блок коэффициентов преобразования.

[0128] Модуль 106 квантования может квантовать коэффициенты преобразования в блоке коэффициентов. Процесс квантования может уменьшить битовую глубину, ассоциированную с некоторыми или всеми коэффициентами преобразования. Например, l -битовый коэффициент преобразования может быть округлен в меньшую сторону до

10 m -битового коэффициента преобразования во время квантования, где l больше чем m . Модуль 106 квантования может квантовать блок коэффициентов, ассоциированный с TU в CU на основании значения параметра квантования (QP), ассоциированного с CU. Видео кодер 20 может регулировать степень квантования, применяемую к блокам

коэффициентов, ассоциированным с CU, посредством регулировки значения QP,

15 ассоциированного с CU. Квантование может ввести потерю информации, таким образом, квантованные коэффициенты преобразования могут иметь более низкую точность, чем первоначальные.

[0129] Модуль 108 обратного квантования и модуль 110 обработки обратного преобразования может применить обратное квантование и обратные преобразования

20 к блоку коэффициентов, соответственно, чтобы восстановить остаточный блок из блока коэффициентов. Модуль 112 реконструкции может добавить восстановленный остаточный блок к соответствующим выборкам из одного или более предсказывающих блоков, генерируемых модулем 100 обработки предсказания, чтобы сформировать

восстановленный блок преобразования, ассоциированный с TU. Посредством

25 восстановления блоков преобразования для каждой TU в CU таким образом, видео кодер 20 может восстановить блоки кодирования упомянутой CU.

[0130] Модуль 114 фильтров может выполнять одну или более операций удаления блочности, чтобы уменьшить артефакты блочности в блоках кодирования,

ассоциированных с CU. Буфер 116 декодированных картинок может сохранить

30 восстановленные блоки кодирования после того, как модуль 114 фильтров выполняет одну или более операций удаления блочности в отношении восстановленных блоков кодирования. Модуль 120 внешнего предсказания может использовать опорную

картинку, которая содержит восстановленные блоки кодирования, чтобы выполнить внешнее предсказание в отношении единиц PU других картинок. Кроме того, модуль

35 126 обработки внутреннего предсказания может использовать восстановленные блоки кодирования в буфере 116 декодированных картинок, чтобы выполнить внутреннее предсказание в отношении других единиц PU в той же картинке как упомянутая CU.

[0131] Модуль 118 энтропийного кодирования может принять данные от других функциональных компонентов видео кодера 20. Например, модуль 118 энтропийного

40 кодирования может принять блоки коэффициентов от модуля 106 квантования и может принять элементы синтаксиса от модуля 100 обработки предсказания. Модуль 118 энтропийного кодирования может выполнить одну или более операций энтропийного кодирования в отношении данных, чтобы генерировать энтропийно кодированные

данные. Например, модуль 118 энтропийного кодирования может выполнить операцию контекстно-адаптивного кодирования с переменной длиной кода (CAVLC), операцию

45 CABAC, операцию кодирования с длиной «от переменной к переменной» (V2V), основанную на синтаксисе операцию контекстно-адаптивного двоичного арифметического кодирования (SBAC), операцию энтропийного кодирования с

разделением интервала вероятности (PIPE), операцию экспоненциального кодирования по Голомбу, или другой тип операции энтропийного кодирования в отношении данных. Видео кодер 20 может вывести поток битов, который включает в себя энтропийно кодированные данные, генерируемые блоком 118 энтропийного кодирования. Например,

поток битов может включать в себя данные, которые представляют RQT для CU. [0132] Как указано в другом месте в настоящем раскрытии, видео кодер 20 может сигнализировать VPS в потоке битов. В HEVC Working Draft 8 конкретные элементы синтаксиса VPS (то есть, `vps_max_dec_pic_buffering [i]`, `vps_max_num_reorder_pics [i]`, и `vps_max_latency_increase [i]`) определены со ссылкой на значение `HighestTid`, которое не определено. В соответствии с одним или более способами настоящего раскрытия, эти элементы синтаксиса VPS могут быть определены со ссылкой на значение `TargetDecHighestTid`, которое определено так, как `TargetDecHighestTid`, которое описано в другом месте в настоящем раскрытии. Таблица 3 ниже иллюстрирует синтаксис VPS в соответствии с одним или более способами настоящего раскрытия.

VPS

ТАБЛИЦА 3

	video_parameter_set_rbsp() {	Дескриптор
	<code>video_parameter_set_id</code>	u(4)
	<code>vps_temporal_id_nesting_flag</code>	u(1)
	<code>vps_reserved_zero_2bits</code>	u(2)
20	<code>vps_reserved_zero_6bits</code>	u(6)
	<code>vps_max_sub_layers_minus1</code>	u(3)
	<code>profile_tier_level(l, vps_max_sub_layers_minus1)</code>	
	<code>vps_reserved_zero_12bits</code>	u(12)
	<code>for(i = 0; i <= vps_max_sub_layers_minus1; i++) {</code>	
25	<code> vps_max_dec_pic_buffering[i]</code>	ue(v)
	<code> vps_max_num_reorder_pics[i]</code>	ue(v)
	<code> vps_max_latency_increase[i]</code>	ue(v)
	<code>}</code>	
	<code>vps_num_hrd_parameters</code>	ue(v)
	<code>for(i = 0; i < vps_num_hrd_parameters; i++) {</code>	
30	<code> operation_point_layer_ids(i)</code>	
	<code> hrd_parameters(i == 0, vps_max_sub_layers_minus1)</code>	
	<code>}</code>	
	<code>vps_extension_flag</code>	u(1)
	<code>if(vps_extension_flag)</code>	
	<code> while(more_rbsp_data())</code>	
35	<code> vps_extension_data_flag</code>	u(1)
	<code> }</code>	
	<code> rbsp_trailing_bits()</code>	
	<code>}</code>	

[0133] Выделенные части Таблицы 3 и других таблиц синтаксиса или описаний семантики в течение настоящего раскрытия могут указывать отличия от HEVC Working Draft 8. В соответствии с одним или более способами настоящего раскрытия, семантика для следующих элементов синтаксиса VPS может быть изменена следующим образом. Семантика для других элементов синтаксиса VPS может остаться такой же как в HEVC Working Draft 8.

`vps_max_dec_pic_buffering [i]` задает требуемый размер буфера декодированных картинок в единицах буферов хранения картинок, когда `TargetDecHighestTid` равен `i`. Значение `vps_max_dec_pic_buffering [i]` должно быть в диапазоне от 0 до `MaxDpbSize` (как определено в подпункте 4), включительно. Когда `i` больше чем 0,

vps_max_dec_pic_buffering [i] должно быть равным или больше чем vps_max_dec_pic_buffering [i-1].

vps_max_num_reorder_pics [i] указывает максимальное разрешенное количество картинок, предшествующих любой картинке в порядке декодирования и следующих за этой картинкой в порядке вывода, когда TargetDecHighestTid равен i. Значение vps_max_num_reorder_pics [i] должно быть в диапазоне от 0 до vps_max_dec_pic_buffering [i], включительно. Когда i больше чем 0, vps_max_num_reorder_pics [i] должно быть равным или большим, чем vps_max_num_reorder_pics [i-1].

vps_max_latency_increase [i] не равно 0 используется для вычисления значения MaxLatencyPictures [i] как задано посредством установки MaxLatencyPictures [i] равным vps_max_num_reorder_pics [i]+vps_max_latency_increase [i]. Когда vps_max_latency_increase [i] не равно 0, значение MaxLatencyPictures [i] задает максимальное количество картинок, которые могут предшествовать любой картинке в закодированной видео последовательности в порядке вывода и следуют за этой картинкой в порядке декодирования, когда TargetDecHighestTid равен i. Когда vps_max_latency_increase [i] равно 0, соответствующий предел не выражен. Значение vps_max_latency_increase [i] должно быть в диапазоне от 0 до $2^{32} - 2$, включительно.

[0134] Как показано выше, семантика vps_max_dec_buffering [i], vps_max_num_reorder_pics [i], и vps_max_latency_increase [i] может быть определена в отношении TargetDecHighestTid. Напротив, HEVC Working Draft 8 определяет vps_max_dec_pic_buffering [i], vps_max_num_reorder_pics [i], и vps_max_latency_increase [i] со ссылками HighestTid, где HighestTid не определен.

[0135] Как показано в примерном синтаксисе из Таблицы 3, VPS включает в себя пары синтаксических структур operation_point_layer_ids () и синтаксических структур hrd_parameters (). Синтаксические структуры hrd_parameters () включают в себя элементы синтаксиса, которые задают наборы параметров HRD. Синтаксическая структура operation_point_layer_ids () включает в себя элементы синтаксиса, которые идентифицируют набор рабочих точек. Набор параметров HRD, заданные в синтаксической структуре hrd_parameters (), может быть применимым к рабочим точкам, идентифицированным элементами синтаксиса в соответствующей синтаксической структуре operation_point_layer_ids (). Таблица 4 ниже обеспечивает примерный синтаксис для синтаксической структуры operation_point_layer_ids ().

ID уровней рабочие точки

ТАБЛИЦА 4

	Дескриптор
operation_point_layer_ids(opIdx) {	
op_num_layer_id_values_minus1[opIdx]	ue(v)
for(i = 0; i <= op_num_layer_id_values_minus1[opIdx]; i++)	
op_layer_id[opIdx][i]	u(6)
}	

[0136] Секция 7.4.4 в HEVC Working Draft 8 описывает семантику синтаксической структуры op_point. В соответствии с одним или более способами настоящего раскрытия, секция 7.4.4 HEVC Working Draft 8 может быть изменена следующим образом, чтобы обеспечить семантику для синтаксической структуры operation_point_layer_ids () из Таблицы 4.

Синтаксическая структура operation_point_layer_ids (opIdx) задает набор значений nuh_reserved_zero_6bits, включенных в OpLayerIdSet рабочих точек, к которым синтаксическая структура opIdx-th hrd_parameters () применяется в наборе параметров видео.

op_num_layer_id_values_minus1 [opIdx] плюс 1 задает количество значений nuh_reserved_zero_bbits, включенных в OpLayerIdSet рабочих точек, к которым синтаксическая структура opIdx-th hrd_parameters () применяется в наборе параметров видео. op_num_layer_id_values_minus1 [opIdx] должно быть меньше чем или равным 63.
 5 В потоках битов, соответствующих этой Спецификации, op_num_layer_id_values_minus1 [opIdx] должно быть равным 0. Хотя требуется чтобы значение op_num_layer_id_values_minus1 [opIdx] было равным 0 в этой версии этой Спецификации, декодеры должны позволить другим значениям появляться в синтаксисе op_num_layer_id_values_minus1 [opIdx].

10 op_layer_id [opIdx] [i] задает i-е значение nuh_reserved_zero_bbits, включенного в OpLayerIdSet рабочих точек, к которым синтаксическая структура opIdx-th hrd_parameters () применяется в наборе параметров видео. Никакое значение в op_layer_id [opIdx] [i] не должно быть равным op_layer_id [opIdx] [j], когда i не равно j и оба i и j не находятся в диапазоне от 0 до op_num_layer_id_values_minus1, включительно. op_layer_id [0] [0]
 15 логически выводится, чтобы быть равным 0.

[0137] Как указано выше, элемент синтаксиса op_num_layer_id_values_minus1 [opIdx] плюс 1 задает количество значений nuh_reserved_zero_bbits, включенных в OpLayerIdSet рабочих точек, к которым синтаксическая структура opIdx-th hrd_parameters () применяется в наборе параметров видео. Напротив, HEVC Working Draft 8 обеспечивает
 20 элемент синтаксиса op_num_layer_id_values_minus1 [opIdx], плюс 1, задает количество значений nuh_reserved_zero_bbits, включенных в рабочую точку, идентифицированную посредством opIdx. Аналогично, в примере из Таблицы 4 элемент синтаксиса op_layer_id [opIdx] [i] задает i-е значение nuh_reserved_zero_bbits, включенное в OpLayerIdSet рабочих точек, к которым синтаксическая структура opIdx-th hrd_parameters () применяется в
 25 наборе параметров видео. Напротив, HEVC Working Draft 8 обеспечивает, что элемент синтаксиса op_layer_id [opIdx] [i] задает i-е значение nuh_reserved_zero_bbits, включенного в рабочую точку, идентифицированную посредством opIdx.

[0138] Секция 7.4.2.2 в HEVC Working Draft 8 описывает семантику для SPS. В соответствии с одним или более способами настоящего раскрытия следующие изменения
 30 могут быть сделаны в секции 7.4.2.2 HEVC Working Draft 8. Семантика для других элементов синтаксиса SPS может быть той же как в HEVC Working Draft 8:

sps_max_dec_pic_buffering [i] задает максимальный требуемый размер буфера декодированных картинок в единицах буферов хранения картинок, когда TargetDecHighestTid равен i. Значение sps_max_dec_pic_buffering [i] должно быть в
 35 диапазоне от 0 до MaxDpbSize (как задано в подпункте 4), включительно. Когда i больше чем 0, sps_max_dec_pic_buffering [i] должно быть равным или больше чем sps_max_dec_pic_buffering [i – 1]. Значение sps_max_dec_pic_buffering [i] должно быть меньше чем или равен vps_max_dec_pic_buffering [i] для каждого значения i.

sps_max_num_reorder_pics [i] указывает максимальное разрешенное количество картинок, предшествующих любой картинке в порядке декодирования и следующих
 40 за этой картинкой в порядке вывода, когда TargetDecHighestTid равен i. Значение sps_max_num_reorder_pics [i] должно быть в диапазоне от 0 до sps_max_dec_pic_buffering [i], включительно. Когда i больше чем 0, sps_max_num_reorder_pics [i] должно быть равным или больше чем sps_max_num_reorder_pics [i–1]. Значение
 45 sps_max_num_reorder_pics [i] должно быть меньше чем или равно vps_max_num_reorder_pics [i] для каждого значения i.

sps_max_latency_increase [i] не равное 0 используется для вычисления значения MaxLatencyPictures [i], как задано посредством установки MaxLatencyPictures [i] равным

$sps_max_num_reorder_pics [i] + sps_max_latency_increase [i]$. Когда $sps_max_latency_increase [i]$ не равно 0, значение $MaxLatencyPictures [i]$ задает максимальное количество картинок, которые могут предшествовать любой картинке в закодированной видео последовательности в порядке вывода и следуют за этой картинкой в порядке декодирования, когда $TargetDecHighestTid$ равен i . Когда $sps_max_latency_increase [i]$ равно 0, никакой соответствующий предел не выражается. Значение

$sps_max_latency_increase [i]$ должно быть в диапазоне от 0 до $2^{32} - 2$, включительно. Значение $sps_max_latency_increase [i]$ должно быть меньше чем или равно $vps_max_latency_increase [i]$ для каждого значения i .

[0139] Как показано выше, семантики $sps_max_dec_pic_buffering [i]$, $sps_max_num_reorder_pics [i]$, и $sps_max_latency_increase [i]$ определены в терминах $TargetDecHighestTid$. $TargetDecHighestTid$ определен так, как описано в другом месте в настоящем раскрытии. Напротив, HEVC Working Draft 8 определяет семантику $sps_max_dec_pic_buffering [i]$, $sps_max_num_reorder_pics [i]$, и $sps_max_latency_increase [i]$ со ссылкой на $HighestTid$, который не определен.

[0140] Секция 7.4.5.1 в HEVC Working Draft 8 описывает общую семантику заголовка вырезки. В соответствии с одним или более способами настоящего раскрытия, следующие изменения могут быть сделаны к секции 7.4.5.1 в HEVC Working Draft 8. Другие части секции 7.4.5.1 HEVC Working Draft 8 могут остаться такими же.

$no_output_of_prior_pics_flag$ задает, как ранее декодированные картинки в буфере декодированных картинок рассматриваются после декодирования картинки IDR или BLA. См. Приложение C. Когда текущая картинка является картинкой CRA, или текущая картинка является картинкой IDR или BLA, которая является первой картинкой в потоке битов, значение $no_output_of_prior_pics_flag$ не имеет эффекта на процесс декодирования. Когда текущая картинка является картинкой IDR или BLA, которая не является картинкой в потоке битов, и значение $pic_width_in_luma_samples$ или $pic_height_in_luma_samples$, или $sps_max_dec_pic_buffering [TargetDecHighestTid]$ выведенное из активного набора параметров последовательности, отличается от значения $pic_width_in_luma_samples$ или $pic_height_in_luma_samples$, или $sps_max_dec_pic_buffering [TargetDecHighestTid]$, выведенного из набора параметров последовательности, активного для предыдущей картинки, $no_output_of_prior_pics_flag$ равный 1 может (но не должен) быть логически выведенным декодером независимо от фактического значения $no_output_of_prior_pics_flag$.

[0141] Как показано выше, семантика $no_output_of_prior_pics_flag$ определена со ссылками на $sps_max_dec_pic_buffering [TargetDecHighestTid]$. $TargetDecHighestTid$ определен так, как описано в другом месте в настоящем раскрытии. Напротив, HEVC Working Draft 8 определяет семантику $no_output_of_prior_pics_flags$ со ссылками на $sps_max_dec_pic_buffering [HighestTid]$, где $HighestTid$ не определен.

[0142] Секция 8.1 HEVC Working Draft 8 описывает общий процесс декодирования. В соответствии с одним или более способами настоящего раскрытия, общий процесс декодирования HEVC Working Draft 8 может быть изменен следующим образом.

Входными данными этого процесса являются поток битов, и выходными данными - список декодированных картинок.

Набор $TargetDecLayerIdSet$, который задает набор значений для $nuh_reserved_zero_6bits$ единиц NAL VCL, которые должны быть декодированы, задан следующим образом:

- Если некоторое внешнее средство, не заданное в этой Спецификации, доступно, чтобы установить $TargetDecLayerIdSet$, $TargetDecLayerIdSet$ устанавливаются этим внешним средством.

- Иначе, если процесс декодирования вызывается в тесте соответствия потока битов, как задано в подпункте C.1, TargetDecLayerIdSet устанавливаются как определено в подпункте C.1.

- Иначе, TargetDecLayerIdSet содержит только одно значение для
5 nuh_reserved_zero_6bits, которое равно 0.

Переменная TargetDecHighestTid, которая идентифицирует наивысший временный подуровень, который должен быть декодирован, задана следующим образом:

- Если некоторое внешнее средство, не определенное в этой Спецификации, доступно, чтобы установить TargetDecHighestTid, TargetDecHighestTid устанавливаются этим внешним
10 средством.

- Иначе, если процесс декодирования вызывается в тесте соответствия потока битов, как задано в подпункте C.1, TargetDecHighestTid устанавливаются как определено в подпункте C.1.

- Иначе, TargetDecHighestTid устанавливаются в sps_max_sub_layers_minus1.

15 Процесс извлечения подпотока битов, как задано в подпункте 10.1, применяют с TargetDecHighestTid и TargetDecLayerIdSet в качестве входных данных, и выходные данные назначают на поток битов, называемый BitstreamToDecode.

Нижеследующее относится к каждой закодированной картинке (называемой текущей картинкой, которая обозначена переменной CurrPic) в BitstreamToDecode.

20 В зависимости от значения chroma_format_idc количество массивов выборок текущей картинки является следующим.

- Если chroma_format_idc равен 0, текущая картинка состоит из 1 массива выборок S_L .

- Иначе (chroma_format_idc не равно 0), текущая картинка состоит из 3 массивов
25 выборок S_L , S_{Cb} , S_{Cr} .

Процесс декодирования для текущей картинки принимает элементы синтаксиса и заглавные переменные из пункта 7 в качестве входных данных. При интерпретации семантика каждого элемента синтаксиса в каждой единице NAL и «потока битов» или его части (например, закодированной видео последовательности) вовлечены, причем
30 поток битов или его часть означает BitstreamToDecode или его часть.

Процесс декодирования задан таким образом, что все декодеры должны произвести численно идентичные результаты. Любой процесс декодирования, который приводит к идентичным результатам процесса, описанного здесь, соответствует требованиям
35 процесса декодирования этой Спецификации.

Когда текущая картинка является картинкой CRA, применяется следующее:

- Если некоторое внешнее средство, не определенное в этой Спецификации, доступно, чтобы установить переменную HandleCraAsBlaFlag в некоторое значение, HandleCraAsBlaFlag устанавливаются в значение, предоставленное внешним средством.

- Иначе, значение HandleCraAsBlaFlag установлено в 0.
40

Когда текущая картинка является картинкой CRA, и HandleCraAsBlaFlag равен 1, применяется следующее во время процессов синтаксического разбора и декодирования для каждой закодированной единицы NAL вырезки:

- Значение nal_unit_type установлено в BLA_W_LP.

- Значение no_output_of_prior_pics_flag установлено в 1.
45

ЗАМЕЧАНИЕ 1 - реализации Декодера могут выбрать устанавливать значение no_output_of_prior_pics_flag в 0, когда установка не затрагивает декодирование текущей картинки и следующих картинок в порядке декодирования, например, когда всегда имеется буфер хранения картинок, доступный когда необходимо.

Каждая картинка, упомянутая в этом пункте, является полной закодированной картинкой.

В зависимости от значения `separate_colour_plane_flag` процесс декодирования структурирован следующим образом.

- 5 - Если `separate_colour_plane_flag` равен 0, процесс декодирования вызывается однократно с текущей картинкой, являющейся выходными данными.
- Иначе (`separate_colour_plane_flag` равно 1), процесс декодирования вызывается три раза. Входными данными к процессу декодирования являются все единицы NAL закодированной картинки с идентичным значением `colour_plane_id`. Процесс
- 10 декодирования единиц NAL с конкретным значением `colour_plane_id` задается, как будто только закодированная видео последовательность с монохромным форматом цвета с этим конкретным значением `colour_plane_id` присутствует в потоке битов. Выходные данные каждого из трех процессов декодирования назначаются на 3 массива выборок текущей картинки с единицами NAL с `colour_plane_id` равным 0, назначаемыми на S_L ,
- 15 единицами NAL с `colour_plane_id` равным 1 назначаемыми на S_{Cb} , и единицами NAL с `colour_plane_id` равным 2 назначаемыми на S_{Cr} .

ЗАМЕЧАНИЕ 1 - переменная `ChromaArrayType` выводится как 0, когда `separate_colour_plane_flag` равен 1, и `chroma_format_idc` равен 3. В процессе декодирования

20 значение этой переменной оценивают, приводя к операциям, идентичным таковой из монохромных картинок с `chroma_format_idc`, равным 0.

Процесс декодирования оперирует следующим образом для текущей картинки `CurrPic`:

1. Декодирование единиц NAL задано в подпункте 8.2.
2. Процессы в подпункте 8.3 задают процессы декодирования, используя элементы
- 25 синтаксиса в уровне вырезки и выше:
 - Переменные и функции, касающиеся счета по порядку картинок, вызываются в подпункте 8.3.1 (который только необходим, чтобы быть вызванным для первой вырезки картинки).
 - Процесс декодирования для набора опорных картинок в подпункте 8.3.2 вызывается,
 - 30 в котором опорные картинки могут быть отмечены как «неиспользуемая для ссылки» или «используемая для долгосрочной ссылки» (который должен быть вызван только для первой вырезки картинки).
 - Когда текущая картинка является картинкой BLA или является картинкой CRA, которая является первой картинкой в потоке битов, процесс декодирования для того,
 - 35 чтобы генерировать недоступные опорные картинки, заданные в подпункте 8.3.3, вызывается (который должен быть вызван только для первой вырезки картинки).
 - `PicOutputFlag` устанавливаются следующим образом:
 - Если текущая картинка является картинкой TFD, и предыдущая картинка RAP в порядке декодирования является картинкой BLA или является картинкой CRA, которая
 - 40 является первой закодированной картинкой в потоке битов, `PicOutputFlag` устанавливаются равным 0.
 - Иначе, `PicOutputFlag` устанавливаются равным `pic_output_flag`.
 - В начале процесса декодирования для каждой P или B вырезки, процесс декодирования для построения списков опорных картинок, заданных в подпункте 8.3.4, вызывается для выведения списка опорных картинок 0 (`RefPicList0`), и при декодировании
 - 45 B вырезки, список опорных картинок 1 (`RefPicList1`).
 - После того, как все вырезки текущей картинки были декодированы, декодированная картинка отмечена как «используется для краткосрочной ссылки».

3. Процессы в подпунктах 8.4, 8.5, 8.6, и 8.7 задают процессы декодирования, используя элементы синтаксиса в уровне кодирования блока дерева и выше.

[0143] Как указано в другом месте в настоящем раскрытии, в HEVC Working Draft 8, когда процесс декодирования вызывается для теста соответствия потока битов, семантики элементов синтаксиса ясно не определены в качестве значений TargetDecLayerIdSet, и TargetDecHighestTid должным образом не установлено. Модификации, показанные выше, общего процесса декодирования могут исправить эту проблему. Как показано выше, когда общий процесс декодирования вызывается для теста соответствия потока битов, значения TargetDecLayerIdSet и TargetDecHighestTid установлены как задано в подпункте C.1. Как описано ниже, модифицированная версия подпункта C.1 может установить TargetDecLayerIdSet в набор значений nuh_reserved_zero_6bits, присутствующих в подпотоке битов, соответствующем рабочей точке, подвергаемой тесту. Модифицированная версия подпункта C.1 может установить TargetDecHighestTid в наибольшее значение TemporalId, присутствующее в подпотоке битов, соответствующем рабочей точке, подвергаемой тесту.

[0144] Таким образом, устройство, такое как видео декодер 30, может выполнить процесс декодирования в качестве части выполнения теста соответствия потока битов. Выполнение процесса декодирования может содержать выполнение процесса извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки для рабочей точки, определенной целевым набором идентификаторов уровня, и целевой наивысший временной идентификатор. Целевой набор идентификаторов уровня может содержать значения элементов синтаксиса идентификатора уровня, присутствующих в представлении рабочей точки, причем целевой набор идентификаторов уровня является поднабором значений элементов синтаксиса идентификатора уровня потока битов. Целевой наивысший временной идентификатор может быть равным наибольшему временному идентификатору, присутствующему в представлении рабочей точки, причем целевой наивысший временной идентификатор является меньшим чем или равным наибольшему временному идентификатору, присутствующему в потоке битов. Кроме того, устройство может декодировать единицы NAL представления рабочей точки.

[0145] Как указано в модификациях к секции 8.1 выше, процесс декодирования не обязательно выполняется как часть теста соответствия потока битов. В некоторых случаях, когда процесс декодирования не выполняется как часть теста соответствия потока битов, устройство может выполнить процесс извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки для рабочей точки. В этом случае 0 может быть единственным значением элементов синтаксиса идентификатора уровня (например, nuh_reserved_zero_6bits), присутствующих в представлении рабочей точки, и наибольший временной идентификатор, присутствующий в потоке битов, равен наибольшему временному идентификатору, присутствующему в представлении рабочей точки для рабочей точки. Устройство может декодировать единицы NAL представления рабочей точки второй рабочей точки.

[0146] Альтернативно, устройство может принять, из внешнего источника, целевой набор идентификаторов уровня и целевой наивысший временной идентификатор. Целевой набор идентификаторов уровня может содержать значения элементов синтаксиса идентификатора уровня, присутствующих в представлении рабочей точки для рабочей точки, которая определена целевым набором идентификаторов уровня, и целевой наивысший временной идентификатором. Целевой наивысший временной идентификатор может быть равным наибольшему временному идентификатору, присутствующему в представлении рабочей точки для рабочей точки. Кроме того,

устройство может выполнить процесс извлечения потока битов, чтобы извлечь из потока битов представления рабочей точки для рабочей точки. Кроме того, устройство может декодировать единицы NAL представления рабочей точки для рабочей точки.

[0147] Кроме того, в соответствии с одним или более способами настоящего раскрытия, процесс извлечения подпотока битов, описанный в подпункте 10.1 HEVC, Working Draft 8, может быть изменен следующим образом.

Имеется требование соответствия потока битов, чтобы любой подпоток битов, который включен в выходные данные процесса, заданного в этом подпункте, с `tIdTarget`, равным любому значению в диапазоне от 0 до 6, включительно, и с `layerIdSetTarget`, содержащим только значение 0, должен соответствовать этой Спецификации.

ЗАМЕЧАНИЕ - соответствующий поток битов содержит одну или более закодированных единиц NAL вырезки с `nuh_reserved_zero_6bits`, равным 0 и `TemporalId`, равным 0.

Входными данными к этому процессу являются переменная `tIdTarget` и набор `layerIdSetTarget`.

Выходными данными этого процесса является подпоток битов.

Подпоток битов выводят посредством удаления из потока битов всех единиц NAL с `TemporalId`, большим чем `tIdTarget` или `nuh_reserved_zero_6bits` не среди значений в `layerIdSetTarget`.

[0148] В подпункте 10.1 HEVC Working Draft 8 используется имя переменной `targetDecLayerIdSet`, где `layerIdSetTarget` используется выше. Изменения, показанные выше, в подпункте 10.1 HEVC Working Draft, чтобы использовать `layerIdSetTarget`, могут служить для пояснения, что может быть различие между набором идентификаторов уровня, используемых в процессе извлечения подпотока битов, и `targetDecLayerIdSet`, который, как описано в другом месте в настоящем раскрытии, имеет конкретное определение.

[0149] Кроме того, в соответствии с одним или более способами настоящего раскрытия, общие спецификации ярусов и уровней секции 4.1 HEVC Working Draft 8 могут быть изменены следующим образом. В настоящем раскрытии «профиль» может относиться к поднабору синтаксиса потока битов. «Ярусы» и «уровни» могут быть заданы в пределах каждого профиля. Уровень яруса может быть заданным набором ограничений, наложенных на значения элементов синтаксиса в потоке битов. Эти ограничения могут быть простыми пределами в отношении значений.

[0150] Альтернативно, ограничения могут принять форму ограничений на арифметические комбинации значений (например, ширина картинки, умноженная на высоту картинки, умноженную на количество картинок, декодируемых в секунду). Уровень, заданный для более низкого яруса, является более ограниченным, чем уровень, заданный для более высокого яруса. В соответствии с примером настоящего раскрытия, секция «общие спецификации уровня» (то есть секция 4.1) HEVC Working Draft 8 заново названа «Общие спецификации яруса и уровня», и текст изменен следующим образом. Таблица A-1 может остаться такой же как в HEVC Working Draft 8.

В целях сравнения возможностей яруса, ярус с `general_tier_flag` равным 0, должен рассматриваться как более низкий ярус, чем ярус с `general_tier_flag` равным 1.

В целях сравнения возможностей уровня, для конкретного яруса, более низкий уровень имеет более низкое значение `general_level_idc`.

Следующее задано для того, чтобы выразить ограничения в этом приложении.

- Пусть единица доступа n является n -ной единицей доступа в порядке декодирования, с первой единицей доступа, являющейся единицей доступа 0 (то есть 0-ой единицей

доступа).

- Пусть картинка n является закодированной картинкой или соответствующей декодированной картинкой единицы доступа n .

- Пусть переменная fR установлена в $1 \div 300$.

5 Потоки битов, соответствующие профилю на заданном уровне, должны подчиняться следующим ограничениям для каждого теста соответствия потока битов, как определено в Приложении С:

а) Номинальное время удаления единицы доступа n ($c > 0$) из CPB как задано в подпункте С.2.2 удовлетворяет ограничению, что $t_{r,n}(n) - t_r(n-1)$ равно или больше
10 чем $\text{Max}(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$ для значения PicSizeInSamplesY картинки $n-1$, где MaxLumaSR - значение, заданное в Таблице А-1, которая применяется к картинке $n-1$.

б) Разность между последовательными временами вывода картинок из DPB, как определено в подпункте С.3.2, удовлетворяет ограничению, что $\Delta t_{o,dpb}(n) \geq \text{Max}$
15 $(\text{PicSizeInSamplesY} \div \text{MaxLumaSR}, fR)$ для значения PicSizeInSamplesY картинки n , где MaxLumaSR - значение, определенное в Таблице А-1 для картинки n , при условии, что картинка n является картинкой, которая выдается и не является последней картинкой потока битов, который выводится.

с) $\text{PicSizeInSamplesY} \leq \text{MaxLumaPS}$, где MaxLumaPS задан в Таблице А-1.

20 d) $\text{pic_width_in_luma_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$

e) $\text{pic_height_in_luma_samples} \leq \text{Sqrt}(\text{MaxLumaPS} * 8)$

f) $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}] \leq \text{MaxDpbSize}$, где MaxDpbSize выводятся как задано ниже:

25 if ($\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \gg 2)$)

$\text{MaxDpbSize} = \text{Min}(4 * \text{MaxDpbPicBuf}, 16)$

else if ($\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \gg 1)$)

$\text{MaxDpbSize} = \text{Min}(2 * \text{MaxDpbPicBuf}, 16)$

else if ($\text{PicSizeInSamplesY} \leq (\text{MaxLumaPS} \ll 1) / 3$)

30 $\text{MaxDpbSize} = \text{Min}((3 * \text{MaxDpbPicBuf}) \gg 1, 16)$

else if ($\text{PicSizeInSamplesY} \leq ((3 * \text{MaxLumaPS}) \gg 2)$)

$\text{MaxDpbSize} = \text{Min}((4 * \text{MaxDpbPicBuf}) / 3, 16)$

Else

$\text{MaxDpbSize} = \text{MaxDpbPicBuf}$

35 где MaxLumaPS задано в Таблице А-1, и MaxDpbPicBuf равно 6.

Таблица А-1 задает пределы для каждого уровня каждого яруса. Использование колонки параметров MinCR Таблицы А-1 задано в подпункте 4.2.

Ярус и уровень, которым соответствует поток битов, должны быть указаны элементами синтаксиса general_tier_flag и general_level_idc следующим образом.

40 - general_tier_flag , равный 0, указывает соответствие ярусу Main, и general_tier_flag равный 1 указывает соответствие ярусу High, согласно спецификациям ограничения ярусов в Таблице А-1. general_tier_flag должен быть равным 0 для уровней ниже уровня 4 (в соответствии с записями в Таблице А-1, помеченными посредством «-»). Пределы уровней, отличных от MaxBR и MaxCPB в Таблице А-1, являются общими и для яруса Main и для яруса High.

45 - general_level_idc должен быть установлен равным значению 30, умноженному на номер уровня, заданный в Таблице А-1.

[0151] Как указано в пункте (f) выше, потоки битов, соответствующие профилю на заданном уровне, подчиняются ограничению что $\text{sps_max_dec_pic_buffering}$

[TargetDecHighestTid]<= MaxDpbSize. TargetDecHighestTid может быть определен способом, описанным в другом месте в настоящем раскрытии. Напротив, HEVC Working Draft 8 указывает для пункта (f), что потоки битов, соответствующие профилю на заданном уровне, подчиняются ограничению что `sps_max_dec_pic_buffering`
 5 [sps_max_temporal_layers_minus1]<= MaxDpbSize. Как указано в другом месте в настоящем раскрытии, параметры `sps_max_dec_pic_buffering [i]` могут не быть должным образом выбраны в ограничениях уровня. Замена `sps_max_temporal_layers_minus1` с помощью TargetDecHighestTid в качестве индекса `i` `sps_max_dec_pic_buffering [i]`, в соответствии с одним или более способами настоящего раскрытия, может гарантировать, что
 10 ограничения уровня последовательно выбраны с `i`, равным явно заданному значению TargetDecHighestTid.

[0152] Таким образом, процесс декодирования HRD может декодировать, из SPS, массив элементов синтаксиса (например, `sps_max_dec_pic_buffering []`), где каждый из элементов синтаксиса в массиве указывает максимальный требуемый размер DPB
 15 упомянутого HRD. Кроме того, когда устройство выполняет операцию HRD, устройство может определить, на основании целевого наивысшего временного идентификатора (например, TargetDecHighestTid), конкретный элемент синтаксиса в массиве (например, `sps_max_dec_pic_buffering [TargetDecHighestTid]`). Кроме того, устройство может определить, что поток битов не находится в соответствии со стандартом кодирования
 20 видео, когда значение конкретного элемента синтаксиса больше чем максимальный размер DPB (например, MaxDpbSize).

[0153] Кроме того, в соответствии с одним или более примерными способами настоящего раскрытия, секция 4.2 HEVC Working Draft 8 может быть изменена следующим образом. Секция 4.2 HEVC Working Draft 8 описывает специфичные для
 25 профиля пределы уровня для профиля Main. Таблица A-2 может остаться такой же как в HEVC Working Draft 8.

Потоки битов, соответствующие профилю Main в указанном ярусе и уровне, должны подчиняться следующим ограничениям для тестов соответствия потока битов, как задано в Приложении C:

30 а) Количество вырезов (с `dependent_slice_flag` равным или 0 или 1) в картинке меньше чем или равно MaxSlicesPerPicture, где MaxSlicesPerPicture задано в Таблице A-1.

б) Для параметров HRD VCL,

$\text{BitRate}[\text{SchedSelIdx}] \leq \text{spbBrVclFactor} * \text{MaxBR}$ и

35 $\text{CpbSize}[\text{SchedSelIdx}] \leq \text{spbBrVclFactor} * \text{MaxCPB}$ для по меньшей мере одного значения SchedSelIdx, где `spbBrVclFactor` задано в Таблице A-2 и `BitRate[SchedSelIdx]` и `CpbSize[SchedSelIdx]`, даны следующим образом.

- Если `vcl_hrd_parameters_present_flag` равен 1, `BitRate[SchedSelIdx]` и `CpbSize[SchedSelIdx]` заданы Уравнениями E-45 и E-46, соответственно, используя элементы синтаксиса, которые выбраны как определено в подпункте C.1.

40 - Иначе (`vcl_hrd_parameters_present_flag` равно 0), `BitRate[SchedSelIdx]` и `CpbSize[SchedSelIdx]` логически выводятся как задано в подпункте E.2.3 для VCL HRD параметры.

MaxBR и MaxCPB заданы в Таблице A-1 в единицах `spbBrVclFactor` бит/сек и `spbBrVclFactor` битов, соответственно. Поток битов должен удовлетворять этим условиям
 45 для по меньшей мере одного значения SchedSelIdx в диапазоне от 0 до `spb_cnt_minus1 [TargetDecHighestTid]`, включительно.

с) Для параметров HRD NAL,

$\text{BitRate}[\text{SchedSelIdx}] \leq \text{spbBrNalFactor} * \text{MaxBR}$ и

$CpbSize [SchedSelIdx] \leq cpbBrNalFactor * MaxCPB$ для по меньшей мере одного значения $SchedSelIdx$, где $cpbBrNalFactor$ задано в Таблице A-2 и $BitRate [SchedSelIdx]$ и $CpbSize [SchedSelIdx]$, заданы следующим образом.

- Если $nal_hrd_parameters_present_flag$ равен 1, $BitRate[SchedSelIdx]$ и $CpbSize [SchedSelIdx]$ заданы Уравнениями E-45 и E-46, соответственно, используя элементы синтаксиса, которые выбраны как задано в подпункте C.1.

- Иначе ($nal_hrd_parameters_present_flag$ равно 0), $BitRate[SchedSelIdx]$ и $CpbSize [SchedSelIdx]$ логически выводятся, как задано в подпункте E.2.3 для параметров HRD NAL.

$MaxBR$ и $MaxCPB$ заданы в Таблице A-1 в единицах $cpbBrNalFactor$ бит/сек и $cpbBrNalFactor$ битов, соответственно. Поток битов должен удовлетворять этим условиям для по меньшей мере одного значения $SchedSelIdx$ в диапазоне от 0 до $cpb_cnt_minus1 [TargetDecHighestTid]$, включительно.

d) Сумма переменных $NumBytesInNALunit$ для единицы доступа 0 меньше чем или равна $1.5 * (Max(PicSizeInSamplesY, fR * MaxLumaSR) + MaxLumaSR * (t_r(n) - t_{r,n}(0)) \div MinCR$ для значения $PicSizeInSamplesY$ картинки 0, где $MaxLumaPR$ и $MinCR$ - значения, заданные в Таблице A-1, которые применяются к картинке 0.

e) Сумма переменных $NumBytesInNALunit$ для единицы доступа n с $n > 0$ меньше чем или равна $1.5 * MaxLumaSR * (t_r(n) - t_r(n-1)) \div MinCR$, где $MaxLumaSR$ и $MinCR$ - значения, заданные в Таблице A-1, которые применяются к картинке n .

f) Для уровня 5 и более высоких уровней, переменная $StbSizeY$ должна быть равной 32 или 64.

g) Значение $NumPocTotalCurr$ должно быть меньше чем или равным 8.

h) Значение $num_tile_columns_minus1$ должно быть меньше, чем $MaxTileCols$ и $num_tile_rows_minus1$ должно быть меньше чем $MaxTileRows$, где $MaxTileCols$ и $MaxTileRows$ таковы, как задано в Таблице A-1.

[0154] Как указано в другом месте в настоящем раскрытии, параметры $cpb_cnt_minus1 [i]$ могут не быть должным образом выбраны в ограничениях уровня. HEVC Working Draft 8 задает, что «поток битов должен удовлетворять этим условиям для по меньшей мере одного значения $SchedSelIdx$ в диапазоне от 0 до cpb_cnt_minus1 , включительно...». Задание $TargetDecHighestTid$ в качестве индекса i $cpb_cnt_minus1 [i]$, в соответствии с одним или более способами настоящего раскрытия, может гарантировать, что ограничения уровня последовательно выбраны с i равным явно заданному значению $TargetDecHighestTid$.

[0155] Кроме того, в соответствии с одним или более способами настоящего раскрытия, общий подпункт C.1 в Приложении C HEVC Working Draft 8 может быть модифицирован. Фигуры C-1 и C-2 подпункта C.1 HEVC Working Draft 8 могут остаться такими же как в HEVC WD8. Текст подпункта C.1 HEVC Working Draft 8 может быть изменен следующим образом.

Это приложение задает гипотетический опорный декодер (HRD) и его использование, чтобы проверить соответствие декодера и поток битов.

Два типа потоков битов являются подлежащими проверке соответствия HRD для данной Спецификации. Первый тип потока битов, называемый Типом I потоков битов, является потоком единиц NAL, содержащим только единицы NAL VCL и единицы NAL с nal_unit_type , равным FD_NUT (единицы NAL данных заполнителя) для всех единиц доступа в потоке битов. Второй тип потока битов, названный Типом II потоков битов, содержит в дополнение к единицам NAL VCL и единицам NAL данных заполнителя для всех единиц доступа в потоке битов, по меньшей мере одно из следующего:

- дополнительные единицы NAL не-VCL, отличные от единиц NAL данных заполнителя,

- все элементы синтаксиса `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes`, и `trailing_zero_8bits`, которые формируют поток байтов из потока единиц NAL (как определено в Приложении В).

Фигура С-1 показывает типы точек соответствия потока битов, проверенные посредством HRD.

Элементы синтаксиса единиц NAL не-VCL (или их значения по умолчанию для некоторых из элементов синтаксиса), требуемые для HRD, заданы в семантических подпунктах пункта 7, Приложения D и E.

Используются два типа параметров HRD (HRD параметры NAL и HRD параметры VCL). Параметры HRD сигнализируют через синтаксическую структуру набора параметров видео или через информацию удобства и простоты использования видео, как задано в подпунктах E.1 и E.2, который является частью синтаксической структуры набора параметров последовательности.

Множественные тесты могут быть необходимы для того, чтобы проверить соответствие потока битов. Для каждого теста следующие этапы применяются в перечисленном порядке:

1. Выбирают рабочую точку, подвергаемой тесту, обозначенную как `TargetOp`.

`TargetOp` идентифицируется посредством `OpLayerIdSet` равным `targetOpLayerIdSet`, и `OpTid` равным `targetOpTid`. `targetOpLayerIdSet` содержит набор значений для `nuh_reserved_zero_6bits`, присутствующего в поднаборе потока битов, ассоциированном с `TargetOp` и должен быть поднабором значений для `nuh_reserved_zero_6bits`, присутствующего в потоке битов, подвергаемого тесту. `targetOpTid` равен наибольшему `TemporalId`, присутствующему в поднаборе потока битов, ассоциированном с `TargetOp`, и должен быть меньше чем или равен `greatestTemporalId`, присутствующему в потоке битов, подвергаемом тесту.

2. `TargetDecLayerIdSet` устанавливаются в `targetOpLayerIdSet`, `TargetDecHighestTid` устанавливаются в `targetOpTid`, и `BitstreamToDecode` устанавливаются в выходные данные процесса извлечения подпотока битов, как задано в подпункте 10.1 с `TargetDecHighestTid` и `TargetDecLayerIdSet` в качестве входных данных.

3. Синтаксическая структура `hrd_parameters ()` и синтаксическая структура `sub_layer_hrd_parameters ()`, применимые к `TargetOp`, выбираются. Если `TargetDecLayerIdSet` содержит только значение 0, выбирают синтаксическую структуру `hrd_parameters ()` в активном наборе параметров последовательности. Иначе выбирают синтаксическую структуру `hrd_parameters ()`, которая находится в активном наборе параметров последовательности (или предоставлена через внешнее средство) и для которой набор значений, заданных посредством `op_layer_id [opIdx] [i]`, для i в диапазоне от 0 до `op_num_layer_id_values_minus1 [opIdx]`, включительно, идентичен `TargetDecLayerIdSet`.

В пределах выбранной синтаксической структуры `hrd_parameters ()`, если `BitstreamToDecode` является потоком битов Типа I, выбирают синтаксическую структуру `sub_layer_hrd_parameters (TargetDecHighestTid)`, которая непосредственно следует за условием, «если (`vcl_hrd_parameters_present_flag`)» (в этом случае переменную `NalHrdModeFlag` устанавливают равной 0), иначе (`BitstreamToDecode` является потоком битов Типа II) выбирают синтаксическую структуру `sub_layer_hrd_parameters (TargetDecHighestTid)`, которая или непосредственно следует за условием, «если (`vcl_hrd_parameters_present_flag`)» (в этом случае переменную `NalHrdModeFlag` устанавливают равной 0) или за условием «если (`nal_hrd_parameters_present_flag`)» (в

этом случае переменную `NalHrdModeFlag` устанавливают равной 1), и все единицы NAL не-VCL за исключением единиц NAL данных заполнителя отбрасывают из `BitstreamToDecode` в первом случае, и результат назначают на `BitstreamToDecode`.

4. Единица доступа, ассоциированная с сообщением SEI периода буферизации, применимое к `TargetOp`, выбирают как точку инициализации HRD и на которую ссылаются как единицу доступа 0.

5. Сообщения SEI, включающие информацию тактирования, выбирают. Выбирают сообщение SEI периода буферизации, которое закодировано в единице доступа 0 и применяется к `TargetOp`, как указано синтаксической структурой `applicable_operation_points` (). Для каждой единицы доступа в `BitstreamToDecode`, начиная с единицы доступа 0, выбирают сообщение SEI тактирования картинок, которое ассоциировано с этой единицей доступа и применяется к `TargetOp`, как указано синтаксической структурой `applicable_operation_points` (), и когда `SubPicCpbFlag` равен 1 и `sub_pic_cpb_params_in_pic_timing_sei_flag` равен 0, выбирают сообщения SEI тактирования суб-картинок, которые ассоциированы с единицами декодирования в единице доступа и применяются к `TargetOp`, как указано синтаксическими структурами `applicable_operation_points` ().

6. Значение `SchedSelIdx` выбирают. Выбранное `SchedSelIdx` должно быть в диапазоне от 0 до `cpb_cnt_minus1 [TargetDecHighestTid]`, включительно, где `cpb_cnt_minus1 [TargetDecHighestTid]` находят в синтаксической структуре `sub_layer_hrd_parameters (TargetDecHighestTid)`, как выбрано выше.

7. Начальное смещение задержки и задержки удаления из CPB выбирают, и единицы доступа TFD, ассоциированные с единицей доступа 0, могут быть отброшены из `BitstreamToDecode`. Если закодированная картинка в единице доступа 0 имеет `nal_unit_type`, равный `CRA_NUT` или `BLA_W_LP`, и `rap_cpb_params_present_flag` в выбранном сообщении SEI периода буферизации равен 1, или начальную задержку удаления из CPB и смещение задержки по умолчанию, представленные посредством `initial_cpb_removal_delay [SchedSelIdx]` и `initial_cpb_removal_delay_offset [SchedSelIdx]` соответствующие `NalHrdModeFlag` (в этом случае переменную `DefaultInitCpbParamsFlag` устанавливают равной 1) или альтернативные начальное смещение задержки и задержку удаления из CPB, представленные посредством `initial_alt_cpb_removal_delay [SchedSelIdx]` и `initial_alt_cpb_removal_delay_offset [SchedSelIdx]`, соответствующие `NalHrdModeFlag` (в этом случае переменную `DefaultInitCpbParamsFlag` устанавливают равной 0) выбирают, и единицы доступа TFD, ассоциированные с единицей доступа 0, отбрасывают из `BitstreamToDecode` в последнем случае, и результат назначают на `BitstreamToDecode`.
Иначе, начальная задержка удаления из CPB и смещение задержки выбирают по умолчанию (в этом случае переменную `DefaultInitCpbParamsFlag` устанавливают равной 1).

Количество выполненных тестов соответствия потока битов равно $N1 * N2 * N3 * (N4 * 2 + N5)$, где значения $N1$, $N2$, $N3$, $N4$ и $N5$ заданы следующим образом.

- $N1$ - количество рабочих точек, содержащихся в потоке битов, подвергаемом тесту.
- Если `BitstreamToDecode` является потоком битов Типа I, $N2$ равно 1, иначе (`BitstreamToDecode` является потоком битов Типа II), $N2$ равно 2.
- $N3$ равно `cpb_cnt_minus1 [TargetDecHighestTid] + 1`.
- $N4$ - количество единиц доступа, ассоциированных с сообщениями SEI периода буферизации, применимыми к `TargetOp` в `BitstreamToDecode`, где закодированная картинка в каждой из этих единиц доступа имеет `nal_unit_type`, равный `CRA_NUT` или `BLA_W_LP`, и ассоциированное сообщение SEI периода буферизации, применимое к `TargetOp`, имеет `rap_cpb_params_present_flag` равный 1.

- N5 - количество единиц доступа, ассоциированных с сообщениями SEI периода буферизации, применимыми к TargetOp в BitstreamToDecode, где закодированная картинка в каждой из этих единиц доступа имеет nal_unit_type, не равный одному из CRA_NUT и BLA_W_LP, или ассоциированное сообщение SEI периода буферизации, применимое к TargetOp, имеет rap_cpb_params_present_flag равный 0.

Когда BitstreamToDecode является потоком битов Типа II, если синтаксическая структура sub_layer_hrd_parameters (TargetDecHighestTid), которая непосредственно следует за условием, «если (vcl_hrd_parameters_present_flag)» выбирается, то тест проводится в точке соответствия Типа I, показанной на фигуре C-1, и только единицы NAL VCL и данных заполнителя подсчитывают для частоты следования битов ввода и хранения CPB; иначе синтаксическая структура (sub_layer_hrd_parameters (TargetDecHighestTid), которая непосредственно следует за условием, «если (nal_hrd_parameters_present_flag)» выбирается, тесты проводятся в точке соответствия Типа II, показанной на фигуре C-1, и все единицы NAL (поток единиц NAL Типа II) или все байты (потока байтов) подсчитывают для частоты следования битов ввода и хранения CPB.

ЗАМЕЧАНИЕ 3 - HRD параметры NAL, установленные значением SchedSelIdx для точек соответствия Типа II, показанных на фигуре C-1, являются достаточными, чтобы также установить HRD соответствие VCL для точек соответствия Типа I, показанных на фигуре C-1 для тех же значений InitCpbRemovalDelay [SchedSelIdx], BitRate[SchedSelIdx], и CpbSize [SchedSelIdx] для случая VBR (cbr_flag [SchedSelIdx] равен 0). Это имеет место потому, что поток данных в точке соответствия Типа I является поднабором потока данных в точке соответствия Типа II и потому что для случая VBR, для CPB разрешено стать пустым и оставаться пустым до времени, когда следующая картинка, которая запланирована, начнет прибывать. Например, при декодировании закодированной видео последовательности, соответствующей одному или более профилям, заданным в Приложении A, используя процесс декодирования, заданный в пунктах 2-9, когда HRD параметры NAL предоставлены для точек соответствия Типа II, которые не только находятся в пределах набора границ для HRD параметров NAL для соответствия профиля в пункте c) подпункта 4.2, но также и находятся в пределах набора границ для HRD параметров VCL для соответствия профиля в пункте b) подпункта 4.2, соответствия HRD VCL для точек соответствия Типа I также обеспечивается, что попадают в пределы границ пункта b) подпункта 4.2.

Все наборы параметров видео, наборы параметров последовательности и наборы параметров картинки, упомянутые в единицах NAL VCL, и соответствующие сообщения SEI периода буферизации и тактирования картинок должны быть переданы к HRD своевременно или в потоке битов (посредством единиц NAL не-VCL) или другим средством, не заданным в этой Спецификации.

В Приложениях C, D, и E, также удовлетворяется спецификация для «присутствия» единиц NAL не-VCL, когда эти единицы NAL (или только некоторые из них) передают к декодерам (или к HRD) другим средством, не заданным этой Спецификацией. С целью подсчета битов только подсчитывают соответствующие биты, которые фактически присутствуют в потоке битов.

ЗАМЕЧАНИЕ 1 - Как пример, синхронизация единицы NAL не-VCL, переданной средством, отличным от присутствия в потоке битов, с единицами NAL, которые присутствуют в потоке битов, может быть достигнута посредством индикации двух точек в потоке битов, между которыми единица NAL не-VCL может присутствовать в потоке битов, если кодер решил передать ее в потоке битов.

Когда содержимое единицы NAL не-VCL передано для применения некоторым средством, отличным от присутствия в пределах потока битов, представление содержимого единицы NAL не-VCL не обязано использовать тот же самый синтаксис как задано в этой Спецификации.

5 ЗАМЕЧАНИЕ 2 - Когда информация HRD содержится в пределах потока битов, возможно верифицировать соответствие потока битов требованиям этого подпункта, на основании исключительно информации, содержащейся в потоке битов. Когда информация HRD не присутствует в потоке битов, как это имеет место для всех «автономных» потоков битов Типа I, соответствие может только быть верифицировано, 10 когда данные HRD подаются некоторым другим средством, не заданным в этой Спецификации.

HRD содержит буфер кодированных картинок (CPB), процесс мгновенного декодирования, буфер декодированных картинок (DPB), и выходные данные, обрезанные, как показано на фигуре C-2.

15 Для каждого теста соответствия потока битов размер CPB (количество битов) равно CpbSize [SchedSelIdx] как задано Уравнением E-46, где SchedSelIdx и параметры HRD выбраны как задано выше в этом подпункте. Размер DPB (количество буферов хранения картинок) равно sps_max_dec_pic_buffering [TargetDecHighestTid].

Переменная SubPicCpbPreferredFlag или задается внешним средством, или, если не 20 задается внешним средством, устанавливается в 0.

Переменная SubPicCpbFlag выводится следующим образом:

SubPicCpbFlag=SubPicCpbPreferredFlag && sub_pic_cpb_params_present_flag (c-1)

Если SubPicCpbFlag равен 0, CPB работает на уровне единицы доступа, и каждая единица декодирования является единицей доступа. Иначе CPB работает на уровне суб- 25 картинок, и каждая единица декодирования является поднабором единицы доступа.

HRD работает следующим образом. Данные, ассоциированные с единицами декодирования, которые поступают в CPB согласно указанному плану прибытия, доставляются посредством HSS. Данные, ассоциированные с каждой единицей декодирования, удаляются и декодируются мгновенно процессом мгновенного 30 декодирования во время удаления из CPB единицы декодирования. Каждая декодированная картинка помещается в DPB. Декодированная картинка удаляется из DPB, как задано в подпункте C.3.1 или подпункте C.5.2.

Операция CPB для каждого теста соответствия потока битов задана в подпункте C.2. Мгновенная операция декодера задана в пунктах 2-9. Операция DPB для каждого 35 теста соответствия потока битов задана в подпункте C.3. Обрезание выходных данных для каждого теста соответствия потока битов задано в подпункте C.3.2 и подпункте C.5.2.

Информация HSS и HRD относительно количества перечисленных планов (списков) доставки и их ассоциированных частот следования битов и размеров буферов заданы 40 в подпунктах E.1.1, E.1.2, E.2.1, и E.2.2. HRD инициализируется, как задано сообщением SEI периода буферизации, как задано в подпунктах D.1.1 и D.2.1. Тактирование удаления единиц декодирования из CPB и тактирование вывода декодированных картинок из DPB заданы в сообщении SEI тактирования картинок, как задано в подпунктах D.1.2 и D.2.1. Вся информация тактирования, касающаяся конкретной единицы декодирования, 45 должна прибыть до времени удаления из CPB единицы декодирования.

Требования для соответствия потока битов заданы в подпункте C.4, и HRD используется для проверки соответствия декодеров, как задано в подпункте C.5.

ЗАМЕЧАНИЕ 3 - В то время как соответствие гарантируется согласно

предположению, что все скорости передачи картинок и системные тактовые сигналы, использованные для генерирования потока битов, точно совпадают со значениями, сигнализированными в потоке битов, в реальной системе каждое из них может изменяться от сигнализированного или заданного значения.

5 Вся арифметика в этом приложении сделана с реальными значениями, так чтобы ошибки округления не могли распространяться. Например, количество битов в CPB непосредственно до или после удаления единицы декодирования является не обязательно целым числом.

10 Переменная t_c выводится следующим образом и названа тактом системных тактовых сигналов:

$$t_c = \text{num_units_in_tick} \div \text{time_scale} (c-1)$$

Переменная t_{c_sub} выводится следующим образом и названа тактом системных тактовых сигналов суб-картинок:

15 $t_{c_sub} = t_c \div (\text{tick_divisor_minus2} + 2) (c-2)$

Следующее задано для того, чтобы выразить ограничения в этом приложении:

- Пусть единица доступа n является n -й единицей доступа в порядке декодирования с первой единицей доступа, являющейся единицей доступа 0 (то есть 0-ой единицей доступа).

20 - Пусть картинке n является закодированной картинкой или декодированной картинкой единицы доступа n .

- Пусть единица декодирования m является m -й единицей декодирования в порядке декодирования с первой единицей декодирования, являющейся единицей декодирования 0.

25 [0156] Модификации к секции C.1 HEVC Working Draft 8 выше могут пояснить тесты соответствия потока битов. Как указано выше, когда процесс декодирования вызывается для теста соответствия потока битов в HEVC Working Draft 8, семантика элементов синтаксиса явно не задана в качестве значений TargetDecLayerIdSet, и TargetDecHighestTid должным образом не установлены. Модификации к секции C.1 поясняют определения

30 TargetDecLayerIdSet и TargetDecHighestTid.

[0157] Как показано в вышеупомянутых модификациях к секции C.1 HEVC Working Draft 8, устройство может выполнить операцию HRD (такую как тест соответствия потока битов), который выбирает рабочую точку, определяет целевой набор идентификаторов уровня (TargetDecLayerIdSet) рабочей точки и наивысший временной

35 идентификатор (TargetDecHighestTid). Кроме того, при операции HRD устройство может выбрать набор параметров HRD, применимых к рабочей точке, и использовать выбранный набор параметров HRD, чтобы конфигурировать HRD, который выполняет процесс декодирования. Набор параметров HRD, применимых к конкретной рабочей точке, может включать в себя параметры, которые задают начальную задержку удаления

40 из CPB, размер CPB, частоту следования битов, начальную задержку вывода из DPB, размер DPB, и так далее. Операция HRD может включать в себя выполнение процесса декодирования.

[0158] В некоторых примерах устройство может выбрать, из числа одного или более наборов параметров HRD (например, синтаксической структуры hrd_parameters ()) в

45 VPS и наборе параметров HRD в SPS, набор параметров HRD, применимых к рабочей точке. В некоторых примерах устройство может определить, что набор параметров HRD в SPS применим к конкретной рабочей точке, когда набор идентификаторов уровня рабочие точки содержит набор всех идентификаторов уровня, присутствующих

в закодированной видео последовательности, ассоциированной с SPS. Кроме того, в некоторых примерах устройство может выбрать набор параметров HRD в SPS в ответ на определение, что целевой набор идентификаторов уровня (например, TargetDecLayerIdSet) рабочей точки содержит только значение 0. В некоторых примерах
 5 устройство может выбрать набор параметра HRD в VPS в ответ на определение, что набор идентификаторов уровня (например, op_layer_id [] []), идентичен целевому набору идентификаторов уровня (например, TargetDecLayerIdSet) рабочей точки.

[0159] Кроме того, как показано в вышеупомянутых модификациях к секции C.1 HEVC Working Draft 8 и других частях настоящего раскрытия, устройство может
 10 декодировать, из SPS, массив элементов синтаксиса (sps_max_dec_pic_buffering []), который каждый указывает максимальный требуемый размер DPB упомянутого HRD. Устройство может определить, на основании целевого наивысшего временного идентификатора, конкретный элемент синтаксиса в массиве (то есть, sps_max_dec_pic_buffering [TargetDecHighestTid]). Как указано выше, количество буферов
 15 хранения картинок в DPB указаны конкретным элементом синтаксиса (то есть, размер DPB (количество буферов хранения картинок) является sps_max_dec_pic_buffering [TargetDecHighestTid]).

[0160] Кроме того, процесс декодирования может декодировать структуру синтаксиса параметров HRD (hrd_parameters ()), которая включает в себя выбранный набор
 20 параметров HRD. Выбранный набор параметров HRD включает в себя массив элементов синтаксиса (cpb_cnt_minus1 []), который каждый указывает количество альтернативных спецификаций CPB в потоке битов. Модификации к секции C.1 HEVC, Working Draft 8 поясняют, что, когда устройство выполняет операцию HRD, устройство может выбрать, на основании целевого наивысшего временного идентификатора (TargetDecHighestTid),
 25 конкретный элемент синтаксиса в массиве (cpb_cnt_minus1 [TargetDecHighestTid]) и может выбрать индекс выбора планировщика (SchedSelIdx) в диапазоне от 0 до значения конкретного элемента синтаксиса. Устройство может определить, на основании, по меньшей мере частично, индекса выбора планировщика, начальную задержку удаления из CPB упомянутого HRD.

[0161] Секция C.2.1 HEVC Working Draft 8 относится к удалению картинок из DPB для соответствия потока битов. В соответствии с одним или более примерными способами настоящего раскрытия, секция C.2.1 HEVC Working Draft 8 может быть изменена следующим образом:

**СПЕЦИФИКАЦИИ В ЭТОМ ПОДПУНКТЕ ПРИМЕНЯЮТСЯ НЕЗАВИСИМО
 35 К КАЖДОМУ НАБОРУ ПАРАМЕТРОВ DPB, ВЫБРАННЫХ, КАК ЗАДАНО В ПОДПУНКТЕ C.1.**

Удаление картинок из DPB прежде декодирования текущей картинки (но после синтаксического разбора заголовка вырезки первой вырезки текущей картинки) осуществляется мгновенно во время удаления из CPB первой единицы декодирования
 40 единицы доступа n (содержащий текущую картинку) и продолжается следующим образом.

Процесс декодирования для набора опорных картинок, как задано в подпункте 8.3.2, вызывается.

Если текущая картинка является картинкой IDR или BLA, применяется следующее:

1. Когда картинка IDR или BLA не является первой декодированной картинкой и значение pic_width_in_luma_samples или pic_height_in_luma_samples, или
 45 sps_max_dec_pic_buffering [TargetDecHighestTid] выведенное из активного набора параметров последовательности, отличается от значения pic_width_in_luma_samples или

pic_height_in_luma_samples, или sps_max_dec_pic_buffering [TargetDecHighestTid], выведенного из набора параметров последовательности, который был активным для предыдущей картинки, соответственно, no_output_of_prior_pics_flag логически выводится, чтобы быть равным 1 посредством HRD, независимо от фактического значения

5 no_output_of_prior_pics_flag.

ЗАМЕЧАНИЕ - Реализации декодера должны попытаться обрабатывать картинки или изменения размера DPB более изящно, чем HRD в отношении изменений в pic_width_in_luma_samples, pic_height_in_luma_samples, или sps_max_dec_pic_buffering [TargetDecHighestTid].

10 2. Когда no_output_of_prior_pics_flag равен 1 или логически выводится, чтобы быть равным 1, все буфера хранения картинок в DPB опустошаются без вывода картинок, которые они содержат, и заполненность DPB устанавливается в 0.

Все картинки k в DPB, для которой оба из следующих условий верны, удаляются из DPB:

- 15
- картинка k отмечена как «неиспользованная для ссылки»,
 - картинка k имеет PicOutputFlag равный 0, или ее время вывода из DPB меньше чем или равно времени удаления из CPB первой единицы декодирования (обозначенный как единица декодирования m) текущей картинки n; то есть $t_{o, dpb}(k) \leq t_r(m)$

Когда картинка удалена из DPB, заполненность DPB уменьшается на единицу.

20 [0162] Как указано в другом месте в настоящем раскрытии, параметры sps_max_dec_pic_buffering [i] могут не быть должным образом выбраны в операциях HRD. HEVC Working Draft 8 просто указывает sps_max_dec_pic_buffering [i] вместо sps_max_dec_pic_buffering [TargetDecHighestTid], как показано выше. HEVC Working Draft 8 не указывает семантику индекса i в секции C.2.1. Задание TargetDecHighestTid в качестве

25 индекса i в sps_max_dec_pic_buffering [i], в соответствии с одним или более способами настоящего раскрытия, может гарантировать, что i, равное явно указанному значению TargetDecHighestTid, используется в sps_max_dec_pic_buffering [i] при выполнении операции HRD удаления картинок из DPB.

30 [0163] Как показано в вышеупомянутых модификациях к секции C.2.1 HEVC Working Draft 8, устройство может декодировать из SPS, активного для текущей картинки, первый массив элементов синтаксиса (sps_max_dec_pic_buffering []), где каждый указывает максимальный требуемый размер DPB упомянутого HRD. Кроме того, устройство может декодировать из SPS, активного для предыдущей картинки, второй массив

35 элементов синтаксиса (sps_max_dec_pic_buffering []), где каждый указывает максимальный требуемый размер DPB упомянутого HRD. Устройство может определить, на основании целевого наивысшего временного идентификатора (TargetDecHighestTid), первый элемент синтаксиса в первом массиве (sps_max_dec_pic_buffering [TargetDecHighestTid]). Кроме того, устройство может определить, на основании целевого наивысшего временного

40 идентификатора, второй элемент синтаксиса во втором массиве (sps_max_dec_pic_buffering [TargetDecHighestTid]). Когда текущая картинка является картинкой мгновенного обновления декодирования (IDR) или картинкой доступа с разорванной ссылкой (BLA), и значение первого элемента синтаксиса отлично от значения второго элемента синтаксиса, устройство может логически вывести значение

45 третьего элемента синтаксиса (no_output_of_prior_pics_flag) независимо от значения, указанного третьим элементом синтаксиса. Третий элемент синтаксиса может определить, как ранее декодированные картинки в DPB обрабатываются после декодирования картинки IDR или картинки BLA.

[0164] Картинка IDR может быть картинкой точки произвольного доступа (RAP),

для которой каждый сегмент вырезки имеет `nal_unit_type`, равный `IDR_W_LP` или `IDR_N_LP`. Картинка `IDR` содержит только I вырезки, и может быть первой картинкой в потоке битов в порядке декодирования, или может появиться позже в потоке битов. Картинка `IDR`, имеющая `nal_unit_type`, равный `IDR_N_LP`, не имеет ассоциированных начальных картинок, присутствующих в потоке битов. Начальная картинка является картинкой, которая предшествует ассоциированной картинке `RAP` в порядке вывода. Картинка `IDR`, имеющая `nal_unit_type`, равный `IDR_W_LP`, не имеет ассоциированных тегированных для отбрасывания картинок, присутствующих (`TFD`) в потоке битов, но может иметь ассоциированные картинки `DLP` в потоке битов.

[0165] Картинка `BLA` является картинкой `RAP`, для которой каждый сегмент вырезки имеет `nal_unit_type`, равный `BLA_W_TFD`, `BLA_W_DLP` или `BLA_N_LP`. Картинка `BLA`, имеющая `nal_unit_type`, равный `BLA_W_TFD`, может иметь ассоциированные картинки `TFD`, присутствующие в потоке битов. Картинка `BLA`, имеющая `nal_unit_type`, равный `BLA_N_LP`, не имеет ассоциированных начальных картинок, присутствующих в потоке битов. Картинка `BLA`, имеющая `nal_unit_type`, равный `BLA_W_DLP`, не имеет ассоциированных картинок `TFD`, присутствующего в потоке битов, но может иметь ассоциированные картинки `DLP` в потоке битов.

[0166] Секция С.3 HEVC Working Draft 8 описывает операции соответствия потока битов. В соответствии с одним или более примерными способами настоящего раскрытия, секция С.3 HEVC Working Draft 8 может быть модифицирована следующим образом:

Поток битов закодированных данных, соответствующих этой Спецификации, должен удовлетворять всем требованиям, заданным в этом подпункте.

Поток битов должен быть построен согласно синтаксису, семантике, и ограничениям, определенным в этой Спецификации вне приложения.

Первая закодированная картинка в потоке битов должна быть картинкой `RAP`, то есть картинкой `IDR`, картинкой `CRA`, или картинкой `BLA`.

Для каждой текущей картинки, которая декодирована, пусть переменные `maxPicOrderCnt` и `minPicOrderCnt` установлены равными максимуму и минимуму, соответственно, значений `PicOrderCntVal` следующих картинок:

- текущая картинка.
- предыдущая картинка в порядке декодирования, которая имеет `TemporalId` равный 0.
- краткосрочные опорные картинки в наборе опорных картинок текущей картинки.
- все картинки `n`, которые имеют `PicOutputFlag` равный 1 и $t_r(n) < t_r(\text{currPic})$ и $t_{o, dpb}(n) \geq t_r(\text{currPic})$, где `currPic` - текущая картинка.

Все следующие условия должны быть выполнены для каждого из тестов соответствия потока битов:

1. Для каждой единицы доступа `n`, с $n > 0$, ассоциированной с сообщением `SEI` периода буферизации, с $\Delta t_{g, 90}(n)$, заданным посредством

$$\Delta t_{g, 90}(n) = 90000 * (t_{r, n}(n) - t_{af}(n-1)) \quad (C-18)$$

значение `InitCpbRemovalDelay [SchedSelIdx]` должно быть ограничено следующим образом.

- Если `cbr_flag [SchedSelIdx]` равен 0,

$$\text{InitCpbRemovalDelay [SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g, 90}(n)) \quad (C-19)$$

- Иначе (`cbr_flag [SchedSelIdx]` равен 1),

$$\text{Floor}(\Delta t_{g, 90}(n)) \leq \text{InitCpbRemovalDelay [SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g, 90}(n)) \quad (C-20)$$

ЗАМЕЧАНИЕ 4 - точное количество битов в `CPB` во время удаления каждой картинки

может зависеть от того, какое сообщение SEI периода буферизации выбрано, чтобы инициализировать HRD. Кодеры должны принять во внимание это, чтобы гарантировать, что все указанные ограничения должны соблюдаться, независимо от того, какое сообщение SEI периода буферизации выбрано, чтобы инициализировать HRD, поскольку HRD может быть инициализировано в любом из сообщений SEI периода буферизации.

2. Переполнение CPB задано как условие, в котором общее количество битов в CPB больше чем размер CPB. CPB никогда не должен переполняться.

3. Потеря значимости CPB задано как условие, в котором номинальное время удаления из CPB единицы декодирования $t_{r,n}(m)$ является меньшим, чем окончное время прибытия в CPB единицы декодирования $t_{af}(m)$ для по меньшей мере одного значения m . Когда `low_delay_hrd_flag` будет равен 0, CPB никогда не должен иметь потерю значимости.

4. Когда `low_delay_hrd_flag` равен 1, потеря значимости CPB может произойти в единице декодирования m . В этом случае окончное время прибытия в CPB единицы доступа n , содержащий единицу декодирования m $t_{af}(n)$, должно быть больше чем номинальное время удаления из CPB единицы доступа n , содержащей единицу декодирования m $t_{r,n}(n)$.

5. Номинальные времена удаления картинок из CPB (начиная со второй картинке в порядке декодирования), должны удовлетворять ограничениям на $t_{r,n}(n)$ и $t_r(n)$, выраженным в подпунктах 4.1-4.2.

6. Для каждой текущей картинке, которая декодируется, после вызова процесса для удаления картинок из DPB, как задано в подпункте C.3.1, количество декодированных картинок в DPB, включая все картинке n , которые отмечены как «используется для ссылки» или которые имеют `PicOutputFlag`, равный 1 и $t_{o,dpb}(n) \geq t_r(\text{currPic})$, где `currPic` - текущая картинке, будет меньше чем или равно $\text{Max}(0, \text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}] - 1)$.

7. Все опорные картинке должны присутствовать в DPB, когда необходимо для предсказания. Каждая картинке, которая имеет `OutputFlag`, равный 1, должна присутствовать в DPB во время его вывода из DPB, если только она не удалена из DPB перед временем ее вывода одним из процессов, заданных в подпункте C.3.

8. Для каждой текущей картинке, которая декодируется, значение $\text{maxPicOrderCnt} - \text{minPicOrderCnt}$ должно быть меньше чем $\text{MaxPicOrderCntLsb}/2$.

9. Значение $\Delta_{t_{o,dpb}}(n)$, как задано Уравнением C-17, которое является разностью между временем вывода картинке и таковым первой картинке, следующей за ней в порядке вывода и имеющей `PicOutputFlag`, равный 1, должно удовлетворять ограничению, выраженному в подпункте 4.1 для профиля, яруса и уровня, заданного в потоке битов, используя процесс декодирования, заданный в пунктах 2-9.

[0167] Как указано в другом месте в настоящем раскрытии, параметры `sps_max_dec_pic_buffering [i]` могут не быть должным образом выбраны в операциях соответствия потока битов. В пункте 6 секции C.3, HEVC Working Draft 8 указывает, что «количество декодированных картинок в DPB ... должно быть меньше чем или равно $\text{Min}(0, \text{sps_max_dec_pic_buffering}[\text{TemporalId}] - 1)$ », где `TemporalId` не определен. Задавая `TargetDecHighestTid` как индекс i для `sps_max_dec_pic_buffering [i]`, в соответствии с одним или более способами настоящего раскрытия, можно гарантировать, что i , равное явно заданному значению `TargetDecHighestTid`, используется в `sps_max_dec_pic_buffering [i]`, при выполнении операции соответствия потока битов.

[0168] Когда устройство выполняет процесс декодирования в качестве части операции

HRD, устройство может декодировать, из SPS, массив элементов синтаксиса (sps_max_dec_pic_buffering []), который каждый указывает максимальный требуемый размер DPB упомянутого HRD. Кроме того, в качестве части выполнения операции HRD, устройство может определять, на основании целевого наивысшего временного идентификатора (TargetDecHighestTid), конкретный элемент синтаксиса в массиве. Кроме того, как показано в вышеупомянутых модификациях к секции C.3 HEVC Working Draft 8, устройство может определять, на основании, по меньшей мере частично, того, меньше ли или равны количество декодированных картинок в DPB максимуму из 0 и значения конкретного элемента синтаксиса минус 1, соответствует ли поток битов стандарту кодирования видео.

[0169] Секция C.4 HEVC Working Draft 8 описывает соответствие декодера. В соответствии с одним или более примерными способами настоящего раскрытия, секция C.4 HEVC Working Draft 8 может быть изменена следующим образом:

Декодер, соответствующий этой Спецификации, должен удовлетворять всем требованиям, заданным в этом подпункте.

Декодер, требующий соответствия конкретному профилю, ярусу и уровню, должен быть в состоянии успешно декодировать все потоки битов, которые соответствуют требованиям соответствия потока битов, определенным в подпункте C.4, способом, заданном в Приложении A, при условии, что все видео наборы параметров, наборы параметров последовательности, и наборы параметров картинки, упомянутые в единицах NAL VCL, и соответствующих сообщениях SEI тактирования картинок и периода буферизации, переданные к декодеру, своевременно или в потоке битов (посредством единиц NAL не-VCL), или внешним средством, не заданным в этой Спецификации.

Когда поток битов содержит элементы синтаксиса, которые имеют значения, которые заданы как зарезервированные, и задано, что декодеры должны игнорировать значения элементов синтаксиса или единицы NAL, содержащие элементы синтаксиса, имеющие зарезервированные значения, и поток битов иначе соответствует этой Спецификации, соответствующий декодер должен декодировать поток битов тем же образом, как он декодировал бы соответствующий поток битов и игнорировал бы значения элементов синтаксиса или единицы NAL, содержащие элементы синтаксиса, имеющие зарезервированные значения как задано.

Имеются два типа соответствия, которое может требоваться декодером: соответствие тактирования вывода и соответствие порядка вывода.

Чтобы проверить соответствие декодера, тестовые потоки битов, соответствующие требуемому профилю, ярусу и уровню, как задано в соответствии с подпунктом C.4, доставляются планировщиком гипотетического потока (HSS) как к HRD, так и к декодеру, подвергаемому тесту (DUT). Все картинки, выведенные посредством HRD, должны также быть выведены посредством DUT и, для каждой картинки, выведенной посредством HRD, значения всех выборок, которые выведены посредством DUT для соответствующей картинки, должны быть равны значениям выборок, выведенным посредством HRD.

Для соответствия декодера тактированию вывода, HSS работает как описано выше, с планами (списками) доставки, выбранными только из поднабора значений SchedSelIdx, для которого частота следования битов и размер CPB ограничены, как задано в Приложении A для заданного профиля, яруса и уровня, или с «интерполированными» планами доставки, как задано ниже, для которых частота следования битов и размер CPB ограничены, как задано в Приложении A. Один и тот же план доставки используется и для HRD и для DUT.

Когда параметры HRD и сообщения SEI периода буферизации присутствуют с cpb_cnt_minus1 [TargetDecHighestTid] большим, чем 0, декодер должен быть способен к декодированию потока битов, который доставляется от HSS, работающего с использованием «интерполированного» списка доставки, заданного как имеющего пиковую частоту следования битов r , размер CPB $c(r)$, и начальную задержку удаления из CPB ($f(r) \div r$) следующим образом:

$$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) \div (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]), \quad (\text{C-22})$$

$$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha) * \text{CpbSize}[\text{SchedSelIdx} - 1], \quad (\text{C-23})$$

$$f(r) = \alpha * \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] + (1 - \alpha)$$

$$* \text{InitCpbRemovalDelay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1] \quad (\text{C-24})$$

для любого $\text{SchedSelIdx} > 0$ и r таким образом, что $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$ таким образом, что r и $c(r)$ находятся в пределах, как задано в Приложении А для максимальной частоты следования битов и размере буфера для заданного профиля, яруса и уровня.

ЗАМЕЧАНИЕ 1 - $\text{InitCpbRemovalDelay}[\text{SchedSelIdx}]$ может отличаться от одного периода буферизации к другому и должно перевычисляться.

Для соответствия декодера тактированию вывода, HRD как описано выше, используется и тактирование (относительно времени доставки первого бита) вывода картинок является одним и тем же и для HRD и для DUT вплоть до фиксированной задержки.

Для соответствия декодера порядку вывода применяется следующее.

- HSS доставляет поток битов BitstreamToDecode к DUT «по требованию» от DUT, означая, что HSS доставляет биты (в порядке декодирования), только когда DUT требует больше битов для возобновления его обработки.

ЗАМЕЧАНИЕ 2 - Это означает, что для этого теста буфер кодированных картинок для DUT может быть столь же малым, как размер наибольшей единицы декодирования.

- Модифицированный HRD, как описано ниже, используется, и HSS доставляет поток битов к HRD в соответствии с одним из списков (планов), заданных в потоке битов BitstreamToDecode таким образом, что частота следования битов и размер CPB ограничены, как задано в Приложении А. Порядок картинок вывода должен быть одним и тем же и для HRD и для DUT.

- Для соответствия декодера порядку вывода размер CPB равен $\text{CpbSize}[\text{SchedSelIdx}]$ как задано Уравнением E-46, где SchedSelIdx и параметры HRD выбирают, как задано выше в подпункте C.1. Размер DPB равен $\text{sps_max_dec_pic_buffering}[\text{TargetDecHighestTid}]$.

Время удаления из CPB для HRD равно времени прибытия окончательному биту, и декодирование является немедленным. Операция DPB этого HRD как описана в подпунктах C.5.1-C.5.3.

[0170] Как указано в другом месте в настоящем раскрытии, параметры $cpb_cnt_minus1[i]$ и $\text{sps_max_dec_pic_buffering}[i]$ могут не быть должным образом выбраны в требованиях соответствия декодера. Например, секция C.4 HEVC Working Draft 8 не задает индекс для cpb_cnt_minus1 . Задавая $\text{TargetDecHighestTid}$ как индекс i для $cpb_cnt_minus1[i]$ и $\text{sps_max_dec_pic_buffering}[i]$, в соответствии с одним или более способами настоящего раскрытия, можно гарантировать, что операции соответствия декодера последовательно выполняются с i , равным явно заданному значению $\text{TargetDecHighestTid}$.

[0171] Кроме того, секция C.4.2 HEVC Working Draft 8 описывает удаление картинок из DPB для соответствия декодера. В соответствии с одним или более примерными способами настоящего раскрытия, название секции C.4.2 может быть изменено от

«удаление картинок из DPB» на «вывод и удаление картинок из DPB». Текст секции C.4.2 HEVC Working Draft 8 может быть изменен следующим образом:

Вывод и удаление картинок из DPB прежде, чем декодировать текущую картинку (но после синтаксического разбора заголовка вырезки первой вырезки текущей картинке) происходят мгновенно, когда первая единица декодирования единицы доступа, содержащей текущую картинку, удаляется из CPB и происходит следующим образом.

Процесс декодирования для набора опорных картинок, как задано в подпункте 8.3.2, вызывается.

- Если текущая картинка является картинкой IDR или BLA, применяется следующее.

1. Когда картинка IDR или BLA не является первой декодированной картинкой и значение `pic_width_in_luma_samples` или `pic_height_in_luma_samples`, или `sps_max_dec_pic_buffering [TargetDecHighestTid]`, выведенное из активного набора параметров последовательности, отличается от значения `pic_width_in_luma_samples` или `pic_height_in_luma_samples`, или `sps_max_dec_pic_buffering [TargetDecHighestTid]`, выведенного из набора параметров последовательности, который был активным для предыдущей картинки, соответственно, `no_output_of_prior_pics_flag` логически выводится, чтобы быть равным 1 посредством HRD, независимо от фактического значения `no_output_of_prior_pics_flag`.

ЗАМЕЧАНИЕ - реализации декодера должны попытаться обрабатывать картинку, или размер DPB изменяется более корректно, чем HRD в отношении изменений в `pic_width_in_luma_samples`, `pic_height_in_luma_samples` или `sps_max_dec_pic_buffering [TargetDecHighestTid]`.

2. Когда `no_output_of_prior_pics_flag` равен 1 или логически выводится, чтобы быть равным 1, все буферы хранения картинок в DPB опустошаются без вывода картинок, которые они содержат.

3. Когда `no_output_of_prior_pics_flag` не равен 1 и логически не выводится, чтобы быть равным 1, буферы хранения картинок, содержащие картинку, которая маркирована как «не является необходимой для вывода» и «неиспользуемая для ссылки» опустошаются (без вывода), и все непустые буфера хранения картинок в DPB опустошаются посредством повторного вызова процесса «выталкивания» (bumping), заданного в подпункте C.5.2.1.

- Иначе (текущая картинка не является картинкой IDR или BLA), буфера хранения картинок, содержащие картинки, которые отмечены как «не является необходимой для вывода» и «неиспользуемая для ссылки», опустошаются (без вывода). Когда одно или более следующих условий верны, процесс «выталкивания», заданный в подпункте C.5.2.1, повторно вызывается, пока не будет существовать пустой буфер хранения картинок, чтобы сохранить текущую декодированную картинку.

1. Количество картинок в DPB, которые отмечены как «необходимы для вывода», больше чем `sps_max_num_reorder_pics [TargetDecHighestTid]`.

2. Количество картинок в DPB равно `sps_max_dec_pic_buffering [TargetDecHighestTid]`.
ПРОЦЕСС «ВЫТАЛКИВАНИЯ»

Процесс «выталкивания» вызывается в следующих случаях.

- Текущая картинка является картинкой IDR или BLA, и `no_output_of_prior_pics_flag` не равен 1 и логически не выводится, чтобы быть равным 1, как задано в подпункте C.5.2.

- Текущая картинка не является картинкой IDR или BLA, и количество картинок в DPB, которые отмечены как «необходимые для вывода», больше чем `sps_max_num_reorder_pics [TargetDecHighestTid]`, как задано в подпункте C.5.2.

- Текущая картинка не является картинкой IDR или BLA, и количество картинок в DPB равно `sps_max_dec_pic_buffering [TargetDecHighestTid]`, как задано в подпункте C.5.2.

Процесс «выталкивания» состоит из следующих упорядоченных этапов:

- 5 1. Картинка, которая является первой для вывода, выбирается как имеющая наименьшее значение `PicOrderCntVal` из всех картинок в DPB, отмеченных как «необходима для вывода».
2. Картинка обрезается, используя прямоугольник обрезания, заданный в активном наборе параметров последовательности для картинки, обрезанная картинка выводится, 10 и картинка помечается как «не является необходимой для вывода».
3. Если буфер хранения картинок, который включал в себя картинку, которая была обрезана и выведена, содержит картинку, отмеченную как «неиспользуемая для ссылки», буфер хранения картинок опустошается.

[0172] Как указано в другом месте в настоящем раскрытии, параметры 15 `sps_max_dec_pic_buffering [i]` и `sps_max_num_reorder_pics [i]` могут не быть должным образом выбраны в операциях HRD, таких как удаление картинок из DPB. Задавая `TargetDecHighestTid` как индекс `i` для `sps_max_dec_pic_buffering [i]` и `sps_max_num_reorder_pics [i]`, в соответствии с одним или более способами настоящего раскрытия, можно гарантировать, что `i`, равное явно заданному значению 20 `TargetDecHighestTid`, используется в `sps_max_dec_pic_buffering [i]` и `sps_max_num_reorder_pics [i]`, при выполнении операции HRD удаления картинок из DPB.

[0173] Когда устройство выполняет процесс декодирования во время операции HRD, устройство может декодировать, из SPS, массив элементов синтаксиса 25 (`sps_max_dec_pic_buffering []`), который каждый указывает максимальный требуемый размер DPB упомянутого HRD. Кроме того, когда устройство выполняет операцию HRD, устройство может определить, на основании целевого наивысшего временного идентификатора, конкретный элемент синтаксиса в массиве (`sps_max_dec_pic_buffering [TargetDecHighestTid]`). Кроме того, устройство может выполнить процесс выталкивания, 30 который опустошает один или более буферов хранения картинок DPB, когда текущая картинка не является картинкой IDR или картинкой BLA, и количество картинок в DPB, отмеченных как необходимых для вывода, больше чем значение конкретного элемента синтаксиса.

[0174] Аналогично, когда устройство выполняет процесс декодирования во время операции HRD, устройство может декодировать, из SPS, массив элементов синтаксиса 35 (`sps_max_dec_pic_buffering []`), который каждый указывает максимальный требуемый размер DPB упомянутого HRD. Кроме того, когда устройство выполняет операцию HRD, устройство может определить, на основании целевого наивысшего временного идентификатора, конкретный элемент синтаксиса в массиве (`sps_max_dec_pic_buffering [TargetDecHighestTid]`). Кроме того, устройство может выполнить процесс выталкивания, 40 который опустошает один или более буферов хранения картинок DPB, когда текущая картинка не является картинкой IDR или картинкой BLA, и количество картинок в DPB равно указанному конкретным элементом синтаксиса.

[0175] Кроме того, в соответствии с одним или более способами настоящего раскрытия, синтаксическая структура `applicable_operation_points ()` и ассоциированная семантика могут быть добавлены к HEVC Working Draft 8. Таблица 5 ниже показывает 45 примерный синтаксис синтаксической структуры `applicable_operation_points ()`.

Применимые рабочие точки

ТАБЛИЦА 5

	<code>applicable_operation_points() {</code>	
	<code> num_applicable_ops_minus1</code>	<code>ue(v)</code>
5	<code> if(num_applicable_ops_minus1 > 0)</code>	
	<code> default_op_applicable_flag</code>	<code>u(1)</code>
	<code> numOpsSignalled = default_op_applicable_flag ?</code> <code> num_applicable_ops_minus1 : num_applicable_ops_minus1 + 1</code>	
	<code> for(i = 0; i < numOpsSignalled; i++) {</code>	
	<code> operation_point_layer_ids(i)</code>	
10	<code> op_temporal_id[i]</code>	<code>u(3)</code>
	<code> }</code>	
	<code> }</code>	

[0176] Синтаксическая структура `applicable_operation_point()`, показанная в Таблице 5, задает рабочие точки, к которым применяется сообщение SEI, ассоциированное с этой синтаксической структурой. Сообщение SEI, ассоциированное с синтаксической структурой `applicable_operation_point()`, (также называемое ассоциированное сообщение SEI), является сообщением SEI, которое содержит синтаксическую структуру `applicable_operation_point()`. Сообщение SEI, ассоциированное с синтаксической структурой `applicable_operation_point()`, может быть сообщением SEI периода буферизации, сообщением SEI тактирования картинок или сообщением SEI тактирования суб-картинок.

[0177] Рабочая точка по умолчанию может быть задана как рабочая точка, идентифицированная посредством `OpLayerIdSet`, содержащим значения от 0 до `nuh_reserved_zero_6bits`, включительно, где `nuh_reserved_zero_6bits` закодирован в заголовке единицы NAL единицы NAL SEI, содержащей ассоциированное сообщение SEI, и `OpTid` равно значению `TemporalId` единицы NAL SEI, содержащей ассоциированное сообщение SEI. Альтернативно, рабочая точка по умолчанию может быть задана как рабочая точка, идентифицированная посредством `OpLayerIdSet`, содержащим только `nuh_reserved_zero_6bits` в заголовке единицы NAL единицы NAL SEI, содержащей ассоциированное сообщение SEI, и `OpTid` равно значению `TemporalId` единицы NAL SEI, содержащей ассоциированное сообщение SEI. Альтернативно, рабочая точка по умолчанию может быть задана как рабочая точка, идентифицированная посредством `OpLayerIdSet`, содержащим только значение 0, и `OpTid` равно значению `TemporalId` единицы NAL SEI, содержащей ассоциированное сообщение SEI.

[0178] Если `default_op_applicable_flag` равен 1, рабочие точки, к которым применяется ассоциированное сообщение SEI, являются рабочими точками по умолчанию и рабочими точками `num_applicable_ops_minus1`, идентифицированными посредством `OpLayerIdSet`, как задано посредством `operation_point_layer_ids(i)` и `OpTid` равным `op_temporal_id[i]`, с `i` в диапазоне от 0 до `num_applicable_ops_minus1`, включительно. Иначе (`default_op_applicable_flag` равно 0), рабочие точки, к которым применяется ассоциированное сообщение SEI, могут быть рабочими точками `num_applicable_ops_minus1+1`, идентифицированными посредством `OpLayerIdSet`, как задано посредством `operation_point_layer_ids(i)` и `OpTid`, равным `op_temporal_id[i]`, с `i` в диапазоне от 0 до `num_applicable_ops_minus1+1`, включительно.

[0179] Кроме того, в примерном синтаксисе из Таблицы 5 элемент синтаксиса `num_applicable_ops_minus1`, плюс 1, задает количество рабочих точек, к которым применяется ассоциированное сообщение SEI. Значение `num_applicable_ops_minus1`

может быть в диапазоне от 0 до 63, включительно. В примере из Таблицы 5 элемент синтаксиса `default_or_applicable_flag`, равный 1, задает, что ассоциированное сообщение SEI применяется к рабочей точке по умолчанию. Элемент синтаксиса `default_or_applicable_flag`, равный 0, задает, что ассоциированное сообщение SEI не применяется к рабочей точке по умолчанию. Элемент синтаксиса `op_temporal_id [i]` задает *i*-е значение `OpTid`, явно сигнализированное в синтаксической структуре `applicable_operation_point ()`. Значение `op_temporal_id [i]` может быть в диапазоне от 0 до 6, включительно.

[0180] Как указано выше, HEVC Working Draft 8 не обеспечивает способа

ассоциировать сообщение SEI периода буферизации или сообщение SEI тактирования картинок к синтаксической структуре `hrd_parameters ()`, для которой ассоциированная синтаксическая структура `operation_point_layer_ids ()` включает в себя множественные значения `nuh_reserved_zero_bbits` (то есть, множественные ID уровня в расширении множественных видов, 3DV или масштабируемого кодирования видео в HEVC).

Включение синтаксической структуры `applicable_operation_point ()` может по меньшей мере частично решить эту проблему. Синтаксическая структура `applicable_operation_point ()` может ясно задать рабочие точки, к которым применяется сообщение SEI периода буферизации, сообщение SEI тактирования картинок или сообщение SEI тактирования суб-картинок. Это может позволить использовать информацию, которую переносят в элементах синтаксиса `nuh_reserved_zero_bbits` и `temporal_id_plus1` в заголовке единицы NAL единиц NAL SEI, и может обеспечить совместное использование информации, переданной в одном и том же сообщении SEI периода буферизации, тактирования картинок или тактирования суб-картинок для того, чтобы обработать видео данные, ассоциированные со множественными рабочими точками.

[0181] Секция D.1.1 HEVC Working Draft 8 описывает синтаксис сообщений SEI периода буферизации. В соответствии с одним или более примерными способами настоящего раскрытия, синтаксис сообщения SEI периода буферизации может быть изменен как показано в Таблице 6 ниже. Изменения к синтаксису сообщения SEI периода буферизации могут позволить сообщению SEI периода буферизации включать в себя синтаксическую структуру `applicable_operation_points ()`.

35

40

45

	Buffering_period(payloadSize) {	Дескриптор
	seq_parameter_set_id	ue(v)
5	<i>applicable_operation_points()</i>	
	if(!sub_pic_cpb_params_present_flag)	
	rap_cpb_params_present_flag	u(1)
	if(NalHrdBpPresentFlag) {	
10	for(SchedSelIdx = 0; SchedSelIdx < CpbCnt; SchedSelIdx++)	
	{	
	initial_cpb_removal_delay [SchedSelIdx]	u(v)
	initial_cpb_removal_delay_offset [SchedSelIdx]	u(v)
	if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
15	initial_alt_cpb_removal_delay [SchedSelIdx]	u(v)
	initial_alt_cpb_removal_delay_offset [SchedSelIdx]	u(v)
	}	
	}	
20	}	
	if(VclHrdBpPresentFlag) {	
	for(SchedSelIdx = 0; SchedSelIdx < CpbCnt; SchedSelIdx++)	
	{	
	initial_cpb_removal_delay [SchedSelIdx]	u(v)
25	initial_cpb_removal_delay_offset [SchedSelIdx]	u(v)
	if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
	initial_alt_cpb_removal_delay [SchedSelIdx]	u(v)
	initial_alt_cpb_removal_delay_offset [SchedSelIdx]	u(v)
30	}	
	}	
	}	
	}	

[0182] Секция D.2.1 в HEVC Working Draft 8 описывает семантику элементов синтаксиса сообщения SEI периода буферизации. В соответствии с одним или более способами настоящего раскрытия, семантика синтаксической структуры `buffering_period` (`payloadSize`) может быть изменена следующим образом. Семантика для элементов синтаксиса, не упомянутых, является такой же как в HEVC Working Draft 8.

Сообщение SEI периода буферизации предоставляет информацию начальной задержки удаления из CPB и начальное смещение задержки удаления из CPB.

Следующее применяется для синтаксиса и семантики сообщения SEI периода буферизации:

- Элементы синтаксиса `initial_cpb_removal_delay_length_minus1` и `sub_pic_cpb_params_present_flag`, и переменные `NalHrdBpPresentFlag`, `VclHrdBpPresentFlag`, `CpbSize` [`SchedSelIdx`], `BitRate` [`SchedSelIdx`], и `CpbCnt` находят в или выводят из элементов синтаксиса, найденных в синтаксической структуре `hrd_parameters()` и синтаксической структуре `sub_layer_hrd_parameters()`, применимой к любой из рабочих точек, к которым применяется сообщение SEI периода буферизации.

- Любые две рабочие точки, к которым применяется сообщение SEI периода буферизации, имеющие различные значения OpTid, tIdA, и tIdB указывает, что значения `cpb_cnt_minus1 [tIdA]` и `cpb_cnt_minus1 [tIdB]`, закодированные в синтаксической структуре `(ax) hrd_parameters ()`, применимой(ым) к этим двум рабочим точкам, являются

5 идентичными.

- Любые две рабочие точки, к которым применяется сообщение SEI периода буферизации, имеющие различные значения `OpLayerIdSet`, `layerIdSetA` и `layerIdSetB` указывает, что значения `nal_hrd_parameters_present_flag` и `vcl_hrd_parameters_present_flag`, соответственно, для этих двух синтаксических структур `hrd_parameters ()`, применимых

10 к этим двум рабочим точкам, являются идентичными.

- Поток битов (или часть его) ссылается на поднабор потока битов (или часть его), ассоциированной с любой из рабочих точек, к которой относится сообщение SEI период буферизации.

Если `NalHrdBpPresentFlag` или `VclHrdBpPresentFlag` равны 1, сообщение SEI периода буферизации, применимое к указанным рабочим точкам, может присутствовать в любой

15 единице доступа в закодированной видео последовательности, и сообщение SEI периода буферизации, применимое к указанным рабочим точкам, должно присутствовать в каждой единице доступа RAP, и в каждой единице доступа, ассоциированной с сообщением SEI точки восстановления. Иначе (`NalHrdBpPresentFlag` и `VclHrdBpPresentFlag`

20 оба равны 0), никакая единица доступа в закодированной видео последовательности не должна иметь сообщение SEI периодабуферизации, применимое к заданным рабочим точкам.

ЗАМЕЧАНИЕ - Для некоторых приложений частое присутствие сообщения SEI периода буферизации может быть желательным.

25 Когда единица NAL SEI, которая содержит сообщение SEI периода буферизации и имеет `nuh_reserved_zero_bbits`, равное 0, присутствует, единица NAL SEI должна предшествовать, в порядке декодирования, первой единице NAL VCL в единице доступа.

Период буферизации задан как набор единиц доступа между двумя экземплярами сообщения SEI периода буферизации, *последовательных* в порядке декодирования.

30 Переменная `CpbCnt` выводится, чтобы быть равной `cpb_cnt_minus1 [tId]+1`, где `cpb_cnt_minus1 [tId]` закодировано в синтаксической структуре `hrd_parameters ()`, которая применима к любой из рабочих точек, к которым применяется сообщение SEI период буферизации и которое имеет `OpTid`, равное `tId`.

`seq_parameter_set_id` относится к активному набору параметров последовательности.

35 Значение `seq_parameter_set_id` должно быть равным значению `seq_parameter_set_id` в наборе параметров картинки, на который ссылается закодированная картинка, ассоциированная с сообщением SEI периода буферизации. Значение `seq_parameter_set_id` должно быть в диапазоне от 0 до 31, включительно.

`rap_cpb_params_present_flag`, равный 1, определяет присутствие элементов синтаксиса

40 `initial_alt_cpb_removal_delay [SchedSelIdx]` и `initial_alt_cpb_removal_delay_offset [SchedSelIdx]`. Если не присутствует, значение `alt_cpb_params_present_flag` логически выводится, чтобы быть равным 0. Когда ассоциированная картинка не является ни картинкой CRA, ни картинкой BLA, значение `alt_cpb_params_present_flag` должно быть равным 0.

45 `initial_cpb_removal_delay [SchedSelIdx]` и `initial_alt_cpb_removal_delay [SchedSelIdx]` задают заданные по умолчанию и альтернативные начальные задержки удаления из CPB, соответственно, для `SchedSelIdx`-th CPB. Элементы синтаксиса имеют длину в битах, заданную посредством `initial_cpb_removal_delay_length_minus1 + 1`, и находятся

в единицах тактовых сигналов 90 кГц. Значения элементов синтаксиса не должны быть равными 0 и должны быть меньше чем или равными $90000 * (CpbSize [SchedSelIdx] \div BitRate[SchedSelIdx])$, эквивалентными по времени размера CPB в единицах тактовых сигналов 90 кГц.

5 `initial_cpb_removal_delay_offset [SchedSelIdx]` и `initial_alt_cpb_removal_delay_offset [SchedSelIdx]` задают по заданные умолчанию и альтернативные начальные смещения удаления из CPB, соответственно, для SchedSelIdx-th CPB. Элементы синтаксиса имеют длину в битах, заданную посредством `initial_cpb_removal_delay_length_minus1 + 1`, и находятся в единицах тактовых сигналов 90 кГц. Эти элементы синтаксиса не
10 используются декодерами и могут быть необходимы только для планировщика доставки (HSS), определенного в Приложении С.

[0183] Сообщение SEI периода буферизации может включать в себя параметры HRD (например, `initial_cpb_removal_delay [SchedSelIdx]`, `initial_cpb_removal_delay_offset [SchedSelIdx]`, `initial_alt_cpb_removal_delay [SchedSelIdx]` и `initial_alt_cpb_removal_delay_offset [SchedSelIdx]`). Как указано выше, HEVC Working Draft 8 не обеспечивает способа ассоциировать сообщение SEI периода буферизации с синтаксической структурой `hrd_parameters ()` в VPS, для которой ассоциированная синтаксическая структура `operation_point_layer_ids ()` включает в себя множественные значения `nuh_reserved_zero_6bits` (то есть множественные ID уровней в расширении кодирования видео множественных видов, 3DV или масштабируемого кодирования в HEVC).
20 Следовательно, в соответствии с одним или более способами настоящего раскрытия элемент синтаксиса `applicable_operation_points ()` в сообщении SEI периода буферизации задает рабочие точки, к которым применяется сообщение SEI периода буферизации.

[0184] Секция D.1.2 в HEVC Working Draft 8 указывает синтаксис сообщения SEI тактирования картинок. В соответствии с одним или более способами настоящего раскрытия, синтаксис сообщения SEI тактирования картинок может быть изменен, как показано в Таблице 7 ниже. Изменения в синтаксисе сообщения SEI тактирования картинок могут позволить сообщению SEI тактирования картинок включать в себя синтаксическую структуру `applicable_operation_points ()`.

30 Сообщение SEI тактирования картинок ТАБЛИЦА 7

	Сообщение SEI тактирования картинок	Дескриптор
	<code>pic_timing(payloadSize) {</code>	
	<code> applicable_operation_points()</code>	
	<code> au_cpb_removal_delay_minus1</code>	<code>u(v)</code>
	<code> pic_dpb_output_delay</code>	<code>u(v)</code>
35	<code> if(sub_pic_cpb_params_present_flag) {</code>	
	<code> num_decoding_units_minus1</code>	<code>ue(v)</code>
	<code> du_common_cpb_removal_delay_flag</code>	<code>u(1)</code>
	<code> if(du_common_cpb_removal_delay_flag)</code>	
	<code> du_common_cpb_removal_delay_minus1</code>	<code>u(v)</code>
40	<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>	
	<code> num_nalus_in_du_minus1[i]</code>	<code>ue(v)</code>
	<code> if(!du_common_cpb_removal_delay_flag)</code>	
	<code> du_cpb_removal_delay_minus1[i]</code>	<code>u(v)</code>
	<code> }</code>	
	<code> }</code>	
45	<code>}</code>	

[0185] Кроме того, семантика сообщения SEI тактирования картинок, может быть изменена следующим образом. Семантика для тех элементов синтаксиса синтаксической структуры `pic_timing (payloadSize)`, которые не упомянуты ниже, может быть той же

как таковая в HEVC Working Draft 8.

Сообщение SEI тактирования картинок предоставляет информацию задержки удаления из CPB, и задержку вывода из DPB единицы доступа, ассоциированные с сообщением SEI.

5 Следующее применяется для синтаксиса и семантики сообщения SEI тактирования картинок:

- Элементы синтаксиса `sub_pic_cpb_params_present_flag`, `cpb_removal_delay_length_minus1`, `dpb_output_delay_length_minus1`, и `du_cpb_removal_delay_length_minus1`, и переменную `CpbDpbDelaysPresentFlag` находят в или выводят из элементов синтаксиса, найденных в синтаксической структуре `hrd_parameters()` и синтаксической структуре `sub_layer_hrd_parameters()`, применимой к любой из рабочих точек, к которым применяется сообщение SEI тактирования картинок.

- Поток битов (или часть его) ссылается на поднабор потока битов (или часть его), ассоциированный с любой из рабочих точек, к которым применяется сообщение SEI тактирования картинок.

15 ЗАМЕЧАНИЕ 1 - синтаксис сообщения SEI тактирования картинок зависит от содержимого синтаксической структуры `hrd_parameters()`, применимой к рабочим точкам, к которым применяется сообщение SEI тактирования картинок. Эти синтаксические структуры `hrd_parameters()` находятся в наборе параметров видео и/или наборе параметров последовательности, которые являются активными для закодированной картинке, ассоциированной с сообщением SEI тактирования картинок. Когда сообщение SEI тактирования картинок ассоциировано с единицей доступа CRA, которая является первой единицей доступа в потоке битов, единицей доступа IDR, или единицей доступа BLA, если только ему не предшествует сообщение SEI периода буферизации в пределах одной и той же единицы доступа, активация набора параметров видео и набора параметров последовательности (и, для картинок IDR или BLA, которые не являются первой картинкой в потоке битов, определение, что закодированная картинка является картинкой IDR или картинкой BLA) не происходит до тех пор пока не будет выполнено декодирование первой закодированной единицы NAL вырезки закодированной картинке. Так как закодированная единица NAL вырезки закодированной картинке следует за сообщением SEI тактирования картинок в порядке единиц NAL, могут иметь место случаи, в которых для декодера необходимо сохранить RBSP, содержащий сообщение SEI тактирования картинок, до тех пор пока не будет выполнено определение активного набора параметров видео и/или активного набора параметров последовательности, и затем выполнить синтаксический разбор сообщения SEI тактирования картинок.

Присутствие сообщения SEI тактирования картинок в потоке битов задано следующим образом.

- Если `CpbDpbDelaysPresentFlag` будет равен 1, одно сообщение SEI тактирования картинок, применимое к заданным рабочим точкам, должно присутствовать в каждой единице доступа закодированной видео последовательности.

- Иначе (`CpbDpbDelaysPresentFlag` равен 0), никаких сообщений SEI тактирования картинок, применимых к заданным рабочим точкам, не должно присутствовать ни в какой единице доступа закодированной видео последовательности.

45 Когда единица NAL SEI, которая содержит сообщение SEI тактирования картинок и имеет `nuh_reserved_zero_6bits` равный 0, присутствует, единица NAL SEI должна предшествовать, в порядке декодирования, первой единице NAL VCL в единице доступа.

`au_cpb_removal_delay_minus1` плюс 1 задает, когда HRD работает на уровне единицы

доступа, сколько тактов системных тактовых сигналов ждать после удаления из CPB единицы доступа, ассоциированной с новым сообщением SEI периода буферизации в предыдущей единице доступа, прежде чем удалить из буфера данных единицы доступа, ассоциированной с сообщением SEI тактирования картинок. Это значение также
 5 используется для вычисления самого раннего времени прибытия данных единицы доступа в CPB для HSS. Элемент синтаксиса является кодом фиксированной длины, длина которого в битах задана посредством `cpb_removal_delay_length_minus1+1`.

ЗАМЕЧАНИЕ 2 - значение `cpb_removal_delay_length_minus1`, которое определяет длину (в битах) элемента синтаксиса `au_cpb_removal_delay_minus1`, является значением
 10 `cpb_removal_delay_length_minus1`, закодированным в наборе параметров видео или наборе параметров последовательности, который является активным для закодированной картинки, ассоциированной с сообщением SEI тактирования картинок, хотя `au_cpb_removal_delay_minus1` плюс 1 задает количество тактов системных тактовых сигналов относительно времени удаления предыдущей единицы доступа, содержащей
 15 сообщение SEI периода буферизации, которое может быть единицей доступа другой закодированной видео последовательности.

`pic_dpb_output_delay` используется для вычисления времени вывода из DPB картинки. Оно задает, сколько тактов системных тактовых сигналов ждать после удаления
 20 последней единицы декодирования в единице доступа из CPB прежде, чем декодированная картинка будет выведена из DPB.

ЗАМЕЧАНИЕ 3 Картинка является не удаляемой из DPB в ее время вывода, когда она все еще отмечена как «используется для краткосрочной ссылки», или «используемая для долгосрочной ссылки».

ЗАМЕЧАНИЕ 4 - Только один `pic_dpb_output_delay` задан для декодированной
 25 картинки.

Длина элемента синтаксиса `pic_dpb_output_delay` задана в битах `dpb_output_delay_length_minus1 + 1`. Когда `sps_max_dec_pic_buffering [minTid]` равен 1, где `minTid` - минимум значений `OpTid` всех рабочих точек, к которым сообщение SEI тактирования картинок применяется, `pic_dpb_output_delay` должен быть равным 0.

30 Время вывода, выведенное из `pic_dpb_output_delay` любой картинки, которая выдается из декодера, соответствующего тактированию вывода, должно предшествовать времени вывода, выведенному из `pic_dpb_output_delay` всех картинок в любой последующей закодированной видео последовательности в порядке декодирования.

Порядок вывода картинок, установленный значениями этого элемента синтаксиса,
 35 должен быть тем же порядком, как установлен значениями `PicOrderCntVal`.

Для картинок, которые не выдаются процессом «*выталкивания*», потому что они предшествуют в порядке декодирования картинке IDR или BLA с `no_output_of_prior_pics_flag` равным 1 или логически выведенным, чтобы быть равными 1, времена вывода, выведенные из `pic_dpb_output_delay`, должны увеличиваться с
 40 значением увеличения `PicOrderCntVal` относительно всех картинок в пределах одной и той же закодированной видео последовательности.

`du_common_cpb_removal_delay_flag`, равный 1, задает, что элемент синтаксиса `du_common_cpb_removal_delay_minus1` присутствует. `du_common_cpb_removal_delay_flag`, равный 0, задает, что элемент синтаксиса `du_common_cpb_removal_delay_minus1` не
 45 присутствует.

`du_common_cpb_removal_delay_minus1` плюс 1 задает, сколько тактов системных тактовых сигналов суб-картинок (см. подпункт E.2.1) ждать, прежде удаления из CPB каждой единицы декодирования в единице доступа, ассоциированной с сообщением

SEI тактирования картинок, после удаления из CPB предыдущей единицы декодирования в порядке декодирования. Это значение также используется для вычисления самого раннего времени прибытия данных единицы декодирования в CPB для HSS, как задано в Приложении С. Элемент синтаксиса является кодом фиксированной длины, длина которого в битах задана посредством `du_cpb_removal_delay_length_minus1+1`.

[0186] Как указано выше, HEVC Working Draft 8 не обеспечивает способа ассоциировать сообщение SEI тактирования картинок с синтаксической структурой `hrd_parameters ()` в VPS, для которой ассоциированная синтаксическая структура `operation_point_layer_ids ()` включает в себя множественные значения для `nuh_reserved_zero_6bits` (то есть множественные ID уровня в расширении множественных видов, 3DV или масштабируемом кодировании видео HEVC). Следовательно, в соответствии с одним или более способами настоящего раскрытия элемент синтаксиса `applicable_operation_points ()` в сообщении SEI тактирования картинок задает рабочие точки, к которым применяется сообщение SEI периода буферизации.

[0187] Кроме того, в соответствии с одним или более способами настоящего раскрытия, синтаксис сообщения SEI тактирования суб-картинок может быть изменен, как показано в Таблице 8 ниже. Изменения к синтаксису сообщения SEI тактирования суб-картинок могут позволять сообщениям SEI тактирования суб-картинок включать в себя синтаксическую структуру `applicable_operation_points ()`. В HEVC Working Draft 8 сообщение SEI тактирования суб-картинок не включает в себя синтаксическую структуру `applicable_operation_points ()`.

Сообщение SEI тактирования суб-картинок ТАБЛИЦА 8

	<code>sub_pic_timing(payloadSize) {</code>	Дескриптор
	<code> applicable_operation_points()</code>	
25	<code> du_spt_cpb_removal_delay_minus1</code>	u(v)
	<code>}</code>	

[0188] Секция D.2.2.2 HEVC Working Draft 8 описывает семантику сообщения SEI тактирования суб-картинок. В соответствии с одним или более способами настоящего раскрытия, секция D.2.2.2 HEVC Working Draft 8 может быть модифицирована следующим образом:

Сообщение SEI тактирования суб-картинок предоставляет информацию задержки удаления из CPB для единицы декодирования, ассоциированной с сообщением SEI.

Следующее применяется для синтаксиса и семантику сообщения SEI тактирования суб-картинок:

- Элементы синтаксиса `sub_pic_cpb_params_present_flag` и `cpb_removal_delay_length_minus1`, и переменная `CpbDpbDelaysPresentFlag` находятся или выводят из элементов синтаксиса, найденных в синтаксической структуре `hrd_parameters ()` и синтаксической структуре `sub_layer_hrd_parameters ()`, применимой к любой из рабочих точек, к которым применяется сообщение SEI тактирования суб-картинок.

- Поток битов (или часть его) относится к поднабору потока битов (или часть его), ассоциированному с любой из рабочих точек, к которым применяется сообщение SEI тактирования суб-картинок.

Присутствие сообщения SEI тактирования суб-картинок в потоке битов, задано следующим образом.

- Если `CpbDpbDelaysPresentFlag` равен 1, и `sub_pic_cpb_params_present_flag` равен 1, одно сообщение SEI тактирования суб-картинок, применимое к указанным рабочим точкам, может присутствовать в каждой единице декодирования в закодированной видео последовательности.

- Иначе (CpbDpbDelaysPresentFlag равен 0 или sub_pic_cpb_params_present_flag, равно 0), никакие сообщения SEI тактирования суб-картинок, применимые к заданным рабочим точкам, не должны присутствовать в закодированной видео последовательности.

5 Единица декодирования, ассоциированная с сообщением (SEI) тактирования суб-картинок, состоит, в порядке декодирования, из единицы NAL SEI, содержащей сообщение SEI тактирования суб-картинок, сопровождаемое одной или более единицами NAL, которые не содержат сообщение SEI тактирования суб-картинок, включая все последующие единицы NAL в единице доступа вплоть до, но включая, любой последующей единицы NAL SEI, содержащей сообщение SEI тактирования суб-картинок.
10 В каждой единице декодирования должна быть по меньшей мере одна единица NAL VCL. Все единицы NAL не-VCL, ассоциированные с единицей NAL VCL, должны быть включены в одну и ту же единицу декодирования.

du_spt_cpb_removal_delay_minus1 plus1 задает, сколько тактов системных тактовых сигналов суб-картинок ожидать после удаления из CPB последней единицы
15 декодирования в единице доступа, ассоциированной с самым недавним сообщением SEI периода буферизации в предыдущей единице доступа, прежде удаления из CPB единицы декодирования, ассоциированной с сообщением (SEI) тактирования суб-картинок. Это значение также используется для вычисления самого раннего времени прибытия данных единицы декодирования в CPB для HSS, как задано в Приложении
20 С. Элемент синтаксиса представлен кодом фиксированной длины, длина которого в битах задана посредством cpb_removal_delay_length_minus1+1.

ЗАМЕЧАНИЕ - значение cpb_removal_delay_length_minus1, которое определяет длину (в битах) элемента синтаксиса du_spt_cpb_removal_delay_minus1, является значением cpb_removal_delay_length_minus1, закодированным в наборе параметров видео или наборе
25 параметров последовательности, который является активным для единицы доступа, содержащей единицу декодирования, ассоциированную с сообщением (SEI) тактирования суб-картинок, хотя du_spt_cpb_removal_delay_minus1 плюс 1 задает количество такты системных тактовых сигналов суб-картинок относительно времени удаления *последней* единицы декодирования в предыдущей единице доступа, содержащей сообщение SEI
30 периода буферизации, которое может быть единицей доступа другой закодированной видео последовательности.

[0189] Секция E.2.2 HEVC Working Draft 8 описывает семантику параметра HRD. В соответствии с одним или более способами настоящего раскрытия, секция E.2.2 HEVC Working Draft 8 может быть изменена следующим образом. Семантика для тех элементов
35 синтаксиса параметров HRD, что не упомянуты ниже, может быть той же как в HEVC Working Draft 8.

Синтаксическая структура hrd_parameters () обеспечивает параметры HRD, используемые в операциях HRD. Когда синтаксическая структура hrd_parameters () включена в набор параметров видео, набор из количества значений
40 nuh_reserved_zero_bits, включенных в OpLayerIdSet рабочих точек, к которым применяется синтаксическая структура, или задано соответствующей синтаксической структурой operation_point_layer_ids () в наборе параметров видео или неявно выводится, как задано в подпункте 7.4.4. Когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимыми рабочими точками являются
45 все рабочие точки с OpLayerIdSet, содержащим только значение 0. Альтернативно, когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимыми рабочими точками являются все рабочие точки с OpLayerIdSet идентичные с TargetDecLayerIdSet.

Имеется требование соответствия потока битов, что для всей синтаксической структуры `hrd_parameters ()` в закодированной видео последовательности (или в наборе параметров видео или в наборе параметров последовательности), не должно быть больше одной из них, которая применяется к одной и той же рабочей точке.

5 Альтернативно, требуется, что не должно быть больше одной синтаксической структуры `hrd_parameters ()` в наборе параметров видео, которая применяется к одной и той же рабочей точке. Альтернативно, требуется, что набор параметров видео не должен включать в себя синтаксическую структуру `hrd_parameters ()`, которая применяется к рабочим точкам с `OpLayerIdSet`, содержащим только значение 0.

10 `du_cpb_removal_delay_length_minus1` плюс 1 задает длину, в битах, `du_cpb_removal_delay_minus1 [i]` и `du_common_cpb_removal_delay_minus1` элементы синтаксиса сообщения SEI тактирования картинок.

`cpb_removal_delay_length_minus1` плюс 1 задает длину, в битах, элементов синтаксиса `au_cpb_removal_delay_minus1` в сообщении SEI тактирования картинок и элемента синтаксиса `du_spt_cpb_removal_delay_minus1` в сообщении SEI тактирования суб-картинок. 15 Когда элемент синтаксиса `cpb_removal_delay_length_minus1` не присутствует, он логически выводится, чтобы быть равным 23.

`dpb_output_delay_length_minus1` плюс 1 задает длину, в битах, элемента синтаксиса `pic_dpb_output_delay` в сообщении SEI тактирования картинок. Когда элемент синтаксиса 20 `dpb_output_delay_length_minus1` не присутствует, он логически выводится, чтобы быть равным 23.

`fixed_pic_rate_flag [i]` равный 1, указывает, что, когда *TargetDecHighestTid* равен *i*, временное расстояние между временами вывода HRD любых двух последовательных картинок в порядке вывода ограничено следующим образом. `fixed_pic_rate_flag [i]` 25 равно 0, указывает, что никакие такие ограничения не применяются к временному расстоянию между временами вывода HRD любых двух последовательных картинок в порядке вывода.

Когда `fixed_pic_rate_flag [i]` не присутствует, он логически выводится, чтобы быть равным 0.

30 Когда *TargetDecHighestTid* равен *i* и `fixed_pic_rate_flag [i]` равен 1 для закодированной видео последовательности, содержащей картинку *n*, значение, вычисленное для $\Delta t_{o,dpb}(n)$, как задано в Уравнении C-17, должно быть равным $t_c * (pic_duration_in_tcs_minus1 [i]+1)$, в котором t_c как задано в Уравнении C-1 (использующем значение t_c для закодированной видео последовательности, содержащей картинку *n*), когда одно или 35 более следующих условий верны для следующей картинке *nn*, которая задана для использования в Уравнении C-17:

- картинка *nn* находится в той же закодированной видео последовательности как картинка *n*.

40 - картинка *nn* находится в другой закодированной видео последовательности, и `fixed_pic_rate_flag [i]` равен 1 в закодированной видео последовательности, содержащей картинку *nn*, значение $num_units_in_tick \div time_scale$ является одним и тем же для обеих закодированных видео последовательностей, и значение `pic_duration_in_tc_minus1 [i]` является одним и тем же для обеих закодированных видео последовательностей.

45 `pic_duration_in_tc_minus1 [i]` плюс 1 задает, когда *TargetDecHighestTid* равен *i*, временное расстояние в тактах системных тактовых сигналов между временами вывода HRD любых двух последовательных картинок в порядке вывода в закодированной видео последовательности. Значение `pic_duration_in_tc_minus1 [i]` должно быть в

диапазоне от 0 до 2047, включительно.

`low_delay_hrd_flag [i]` задает рабочий режим HRD, когда `TargetDecHighestTid` равен `i`, как задано в Приложении С. Когда `fixed_pic_rate_flag [i]` равен 1, `low_delay_hrd_flag [i]` должен быть равен 0.

5 ЗАМЕЧАНИЕ 3 - Когда `low_delay_hrd_flag [i]` равен 1, «большие картинки», которые нарушают номинальные времена удаления из СРВ из-за количества битов, используемых единицей доступа, разрешены. Ожидается, но не требуется, что такие «большие картинки» имеют место только иногда.

`cpb_cnt_minus1 [i]` плюс 1 задает количество альтернативных спецификаций СРВ в
10 потоке битов закодированной видео последовательности, когда `TargetDecHighestTid` равен `i`. Значение `cpb_cnt_minus1 [i]` должно быть в диапазоне от 0 до 31, включительно. Когда `low_delay_hrd_flag [i]` равен 1, `cpb_cnt_minus1 [i]` должно быть равным 0. Когда `cpb_cnt_minus1 [i]` не присутствует, оно логически выводится, чтобы быть равным 0.

[0190] Как описано в другом месте в настоящем раскрытии, в HEVC Working Draft
15 8, только синтаксические структуры `hrd_parameters ()` в VPS могут быть выбраны для операций HRD, в то время как синтаксические структуры `hrd_parameters ()` в SPS никогда не выбираются. Изменения, показанные выше, в семантике синтаксической структуры `hrd_parameters ()` поясняют, что, когда синтаксическая структура `hrd_parameters ()` включена в SPS, рабочие точки, к которым применима синтаксическая структура
20 `hrd_parameters ()`, могут быть все рабочими точками с `OpLayerIdSet`, идентичным с `TargetDecLayerIdSet`. Как указано выше в модифицированном общем процессе декодирования, если внешнее средство доступно, чтобы установить `TargetDecLayerIdSet`, `TargetDecLayerIdSet` может быть задан внешним средством. Иначе, если процесс декодирования вызывается в тесте соответствия потока битов, `TargetDecLayerIdSet`
25 может быть набором идентификаторов уровня рабочей точки, подвергаемой тесту. Иначе, `TargetDecLayerIdSet` может содержать только один идентификатор уровня (то есть, только одно значение `nuh_reserved_zero_bbits`), который равен 0. В одном примере внешним средством может быть API, который является частью реализации терминала, и который обеспечивает функцию, чтобы установить значение `TargetDecLayerIdSet`. В
30 этом примере реализация терминала может содержать реализацию декодера и некоторые функции, которые не являются частями реализации декодера.

[0191] Таким образом, устройство (такое как видео кодер 20, видео декодер 30, дополнительное устройство 21, или другое устройство) может выбрать, из числа набора параметров HRD в наборе параметров видео и наборе параметров HRD в SPS, набор
35 параметров HRD, применимых к конкретной рабочей точке. Кроме того, устройство может выполнять, на основании, по меньшей мере частично, набора параметров HRD, применимых к конкретной рабочей точке, тест соответствия потока битов, который проверяет, соответствует ли поднабор потока битов, ассоциированный с конкретной рабочей точкой, стандарту кодирования видео.

[0192] Как указано выше, секция E.2.2 HEVC Working Draft 8 может быть
40 модифицирована, чтобы указывать, что, когда синтаксическая структура `hrd_parameters ()` включена в набор параметров последовательности, применимыми рабочими точками являются все рабочие точки с `OpLayerIdSet`, идентичным с `TargetDecLayerIdSet`. Кроме того, как описано выше, `TargetDecLayerIdSet` устанавливаются в `targetOpLayerIdSet`, который
45 содержит набор значений `nuh_reserved_zero_bbits`, присутствующих в поднаборе потока битов, ассоциированном с `TargetOp`. `TargetOp` является рабочей точкой, подвергаемой тесту в операции HRD. Кроме того, операции HRD (например, тест соответствия потока битов и тест соответствия декодера) могут вызвать общий процесс декодирования.

[0193] Как объяснено выше, секция 8.1 HEVC Working Draft 8 может быть модифицирована, чтобы обеспечить, что процесс извлечения подпотока битов, как задано в подпункте 10.1, применяется с TargetDecHighestTid и TargetDecLayerIdSet в качестве входных данных, и вывод назначается на поток битов, называемый BitstreamToDecode. Следовательно, только значения nuh_reserved_zero_bbits, присутствующие в BitstreamToDecode, являются значениями nuh_reserved_zero_bbits в TestDecLayerIdSet (то есть, набором значений nuh_reserved_zero_bbits, присутствующих в поднаборе потока битов, ассоциированном с TargetOp). Секция 8.1 далее объясняет, что при интерпретации семантики каждого элемента синтаксиса в каждой единице NAL и «этот поток битов» или его часть (например, закодированная видео последовательность) вовлекается, поток битов или его часть означают BitstreamToDecode или его часть.

[0194] Следовательно, при интерпретации секции, описывающей семантику параметров HRD (например, секции E.2.2 HEVC Working Draft 8), термин «закодированная видео последовательность», означает часть BitstreamToDecode. TargetDecLayerIdSet эквивалентен набору всех значений nuh_reserved_zero_bbits, присутствующих в BitstreamToDecode. Из этого следует, что фраза в секции, описывающей семантику параметров HRD, «когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимые рабочие точки является всеми рабочими точками с OpLayerIdSet, идентичным с TargetDecLayerIdSet», эквивалентна «когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимыми рабочими точками являются все рабочие точки с OpLayerIdSet, идентичным набору значений nuh_reserved_zero_bbits, присутствующих в BitstreamToDecode».

[0195] Поскольку «закодированная видео последовательность» является частью BitstreamToDecode, набор nuh_reserved_zero_bbits, присутствующий в закодированной видео последовательности, является поднабором набора nuh_reserved_zero_bbits, присутствующим в BitstreamToDecode. Следовательно, фраза «когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимые рабочие точки является всеми рабочими точками с OpLayerIdSet, идентичной набору значений nuh_reserved_zero_bbits, присутствующему в BitstreamToDecode», обязательно влечет за собой «когда синтаксическая структура hrd_parameters () включена в набор параметров последовательности, применимыми рабочими точками являются все рабочие точки с OpLayerIdSet, содержащим все значения nuh_reserved_zero_bbits, присутствующие в закодированной видео последовательности». Другими словами, если набор nuh_reserved_zero_bbits рабочей точки идентичен набору nuh_reserved_zero_bbits, присутствующему в BitstreamToDecode, то набор nuh_reserved_zero_bbits рабочей точки обязательно содержит все значения nuh_reserved_zero_bbits, присутствующие в закодированной видео последовательности BitstreamToDecode. В этой фразе «закодированная видео последовательность» может относиться к закодированной видео последовательности, ассоциированной с конкретным SPS.

[0196] При выполнении операции HRD устройство может определить из синтаксических структур hrd_parameters (), указанных в VPS, и синтаксической структуры hrd_parameters (), указанной в SPS, синтаксическую структуру hrd_parameters (), применимую к TargetOp. Конкретная синтаксическая структура hrd_parameters () в VPS является применимой к TargetOp, если набор id уровня для TargetOp соответствует набору идентификаторов уровня, заданных в VPS для конкретной синтаксической структуры hrd_parameters (). Синтаксическая структура hrd_parameters () в SPS может

быть применимой к TargetOp, если набор id уровня для TargetOp (то есть, TargetDecHighestTid) (то есть, набор nuh_reserved_zero_6bits, присутствующий в BitstreamToDecode) содержит все nuh_reserved_zero_6bits, присутствующие в закодированной видео последовательности SPS (который является поднабором набора nuh_reserved_zero_6bits в BitstreamToDecode). Поскольку набор nuh_reserved_zero_6bits в TargetOp может необходимо содержать все значения nuh_reserved_zero_6bits, присутствующие в закодированной видео последовательности, ассоциированной с SPS, синтаксическая структура hrd_parameters () в SPS может всегда быть применимой к TargetOp. Однако, не все SPS имеют синтаксические структуры hrd_parameters (). Если SPS действительно имеет синтаксическую структуру hrd_parameters () и набор nuh_reserved_zero_6bits, присутствующий в BitstreamToDecode, содержит все nuh_reserved_zero_6bits, присутствующие в закодированной видео последовательности SPS, то синтаксическая структура hrd_parameters () SPS должна использоваться. Поскольку не все SPS имеют синтаксическую структуру hrd_parameters (), VPS может все еще быть выбран.

[0197] Кроме того, как показано выше в модификациях к секции E.2.2 HEVC Working Draft 8, когда устройство выполняет тест соответствия потока битов, видео декодер может определить, что поток битов не соответствует стандарту кодирования видео, когда для всех наборов параметров HRD в закодированной видео последовательности больше чем один набор параметров HRD применяется к одной и той же рабочей точке. Кроме того, когда устройство выполняет тест соответствия потока битов, видео декодер может определить, что поток битов не соответствует стандарту кодирования видео, когда больше чем один набор параметров HRD в VPS применяется к одной и той же рабочей точке. Кроме того, когда устройство выполняет тест декодирования потока битов, устройство может определить, что поток битов не соответствует стандарту кодирования видео, когда VPS включает в себя набор параметров HRD, который применяется к рабочим точкам, имеющим наборы id уровня, содержащие только значение 0.

[0198] Фиг. 3 является блок-схемой, иллюстрирующей примерный декодер видео 30, который сконфигурирован, чтобы реализовать способы настоящего раскрытия. Фиг. 3 предоставлена в целях объяснения и не ограничивается способами, как широко иллюстрировано и описано в настоящем раскрытии. В целях объяснения настоящее раскрытие описывает видео декодер 30 в контексте кодирования HEVC. Однако, способы настоящего раскрытия могут быть применимыми к другим стандартам или способам кодировки.

[0199] В примере согласно Фиг. 3, видео декодер 30 включает в себя модуль 150 энтропийного декодирования, модуль 152 обработки предсказания, модуль 154 обратного квантования, модуль 156 обработки обратного преобразования, модуль 158 реконструкции, модуль 160 фильтров, и буфер 162 декодированных картинок. Модуль 152 обработки предсказания включает в себя модуль 164 компенсации движения и модуль 166 обработки внутреннего предсказания. В других примерах видео декодер 30 может включать в себя больше, меньше, или другие функциональные компоненты.

[0200] Буфер кодированных картинок (CPB) 151 может принимать и хранить закодированные видео данные (например, единицы NAL) потока битов. Модуль 150 энтропийного декодирования может принять единицы NAL из CPB 151 и синтаксически разобрать единицы NAL, чтобы декодировать элементы синтаксиса. Модуль 150 энтропийного декодирования может энтропийно декодировать энтропийно кодированные элементы синтаксиса в единицах NAL. Модуль 152 обработки

предсказания, модуль 154 обратного квантования, модуль 156 обработки обратного преобразования, модуль 158 реконструкции и модуль 160 фильтров могут генерировать декодированные видео данные на основании элементов синтаксиса, извлеченных из потока битов.

5 [0201] Единицы NAL потока битов могут включать в себя закодированные единицы NAL вырезки. Как часть декодирования потока битов, модуль 150 энтропийного декодирования может извлекать и энтропийно декодировать элементы синтаксиса из закодированных единиц NAL вырезки. Каждая из закодированных вырезок может
10 включать в себя заголовок вырезки и данные вырезки. Заголовок вырезки может содержать элементы синтаксиса, имеющие отношение к вырезке. Элементы синтаксиса в заголовке вырезки могут включать в себя элемент синтаксиса, который идентифицирует PPS, ассоциированный с картинкой, которая содержит вырезку.

[0202] В дополнение к декодированию элементов синтаксиса из потока битов видео декодер 30 может выполнить операцию по реконструкции в отношении неразделенной
15 CU. Чтобы выполнить операцию по реконструкции в отношении неразделенной CU, видео декодер 30 может выполнить операцию по реконструкции в отношении каждой TU в CU. Выполняя операцию по реконструкции для каждой TU в CU, видео декодер 30 может восстановить остаточные блоки CU.

[0203] В качестве части выполнения операции по реконструкции в отношении TU в
20 CU, модуль 154 обратного квантования может инверсно квантовать, то есть, деквантовать, блоки коэффициентов, ассоциированные с TU. Модуль 154 обратного квантования может использовать значение QR, ассоциированное с CU упомянутой TU, чтобы определить степень квантования и, аналогично, степень обратного квантования для модуля 154 обратного квантования для применения. Таким образом, степень сжатия,
25 то есть, отношение количества битов, использованных для представления первоначальной последовательности и сжатых, может управляться, регулируя значение QR, используемого при квантовании коэффициентов преобразования. Степень сжатия может также зависеть от способа используемого энтропийного кодирования.

[0204] После того, как модуль 154 обратного квантования обратно квантует блок
30 коэффициентов, модуль 156 обработки обратного преобразования может применить одно или более обратных преобразований к блоку коэффициентов, чтобы генерировать остаточный блок, ассоциированный с TU. Например, модуль 156 обработки обратного преобразования может применить обратное DCT, обратное целочисленное преобразование, обратное преобразование Karhunen-Loeve (KLT), обратное
35 вращательное преобразование, обратное направленное преобразование, или другое обратное преобразование к блоку коэффициентов.

[0205] Если PU закодирован, используя внутреннее предсказание, модуль 166
обработки внутреннего предсказания может выполнить внутреннее предсказание, чтобы генерировать предсказывающие блоки для PU. Модуль 166 обработки
40 внутреннего предсказания может использовать режим внутреннего предсказания, чтобы генерировать предсказывающие блоки яркости, Cb и Cr для PU на основании блоков предсказания пространственно соседних единиц PU. Модуль 166 обработки внутреннего предсказания может определить режим внутреннего предсказания для PU на основании одного или более элементов синтаксиса, декодированных из потока битов.

45 [0206] Модуль 152 обработки предсказания может построить первый список опорных картинок (RefPicList0) и второй список опорных картинок (RefPicList1), на основании элементов синтаксиса, извлеченных из потока битов. Кроме того, если PU закодирована, используя внешнее предсказание, модуль 150 энтропийного декодирования может

извлечь информацию движения для PU. Модуль 164 компенсации движения может определить, на основании информации движения PU, одну или более опорных областей для PU. Модуль 164 компенсации движения может генерировать, на основании блоков выборок в одном или более опорных блоках для PU, предсказывающие блоки яркости, Сb и Сr для этой PU.

[0207] Модуль 158 реконструкции может использовать блоки преобразования яркости, Сb и Сr, ассоциированные с единицами TU в CU и блоки предсказания яркости, Сb и Сr единиц PU в CU, то есть, или данные внутреннего предсказания или данные внешнего предсказания, как применимые, чтобы восстановить блоки кодирования яркости, Сb и Сr упомянутой CU. Например, модуль 158 реконструкции может добавить выборки блоков преобразования яркости, Сb и Сr к соответствующим выборкам блока предсказания яркости, Сb и Сr, чтобы восстановить блоки кодирования яркости, Сb и Сr упомянутой CU.

[0208] Модуль 160 фильтров может выполнять операцию удаления блочности, чтобы уменьшить артефакты блочности, ассоциированные с блоками кодирования яркости, Сb и Сr упомянутой CU. Видео декодер 30 может сохранить блоки кодирования яркости, Сb и Сr упомянутой CU в буфере 162 декодированных картинок. Буфер 162 декодированных картинок может обеспечить опорные картинки для последующей компенсации движения, внутреннего предсказания, и представления на устройстве отображения, таком как устройство 32 отображения на Фиг. 1. Например, видео декодер 30 может выполнять, на основании блоков яркости, Сb и Сr в буфере 162 декодированных картинок операции внутреннего предсказания или внешнего предсказания в отношении единиц PU других единиц CU. Таким образом, видео декодер 30 может декодировать из потока битов уровни коэффициентов преобразования значимого блока коэффициентов яркости, обратно квантовать уровни коэффициентов преобразования, применять преобразование к уровням коэффициента преобразования, чтобы генерировать преобразованный блок, генерировать, на основании, по меньшей мере частично, преобразованного блока, блок кодирования, и вывести блок кодирования для отображения.

[0209] Фиг. 4 является последовательностью операций, иллюстрирующей примерную операцию 200 устройства, в соответствии с одним или более способами настоящего раскрытия. Операция 200 может быть выполнена видео кодером 20, видео декодером 30, дополнительным устройством 21 или другим устройством. Как иллюстрировано в примере согласно Фиг. 4, устройство может выбрать, из набора параметров гипотетического HRD (например, синтаксической структуры `hrd_parameters`) в VPS и набора параметров HRD в SPS, набор параметров HRD, применимых к конкретной рабочей точке потока битов (202). Кроме того, устройство может выполнять, на основании, по меньшей мере частично, набора параметров HRD, применимых к конкретной рабочей точке, операцию HRD в отношении поднабора потока битов, ассоциированного с конкретной рабочей точкой (204). Например, устройство может выполнить тест соответствия потока битов или тест соответствия декодера.

[0210] Фиг. 5 является последовательностью операций, иллюстрирующей примерную операцию 250 устройства в соответствии с одним или более способами настоящего раскрытия. Операция 200 может быть выполнена видео кодером 20, видео декодером 30, дополнительным устройством 21 или другим устройством. Как иллюстрировано в примере согласно Фиг. 5, устройство может выполнить тест соответствия потока битов, который определяет, соответствует ли поток битов стандарту кодирования видео (252). Устройство может выполнить процесс декодирования в качестве части выполнения

теста соответствия потока битов (254).

[0211] Как иллюстрировано в примере согласно Фиг. 5, при выполнении процесса декодирования устройство может выполнить процесс извлечения потока битов, чтобы извлечь из потока битов представление рабочей точки рабочей точки, определенной целевым набором идентификаторов уровня, и целевой наивысший временной идентификатор (256). Целевой набор идентификаторов уровня может содержать значения элементов синтаксиса идентификатора уровня, присутствующих в представлении рабочей точки. Целевой набор идентификаторов уровня может быть поднабором значений элементов синтаксиса идентификатора уровня потока битов. Целевой наивысший временной идентификатор может быть равным наибольшему временному идентификатору, присутствующему в представлении рабочей точки, причем целевой наивысший временной идентификатор является меньшим чем или равным наибольшему временному идентификатору, присутствующему в потоке битов. Кроме того, устройство может декодировать единицы NAL представления рабочей точки (258).

[0212] Фиг. 6 является последовательностью операций, иллюстрирующей пример операции HRD 300 устройства в соответствии с одним или более способами настоящего раскрытия. Операция HRD 300 может быть выполнена видео кодером 20, видео декодером 30, дополнительным устройством 21 или другим устройством. Другие устройства могут включать в себя модуль проверки потока битов на соответствие, который берет поток битов как введен, и выводит индикацию того, является ли поток битов ввода соответствующим потоком битов или нет. В некоторых примерах операция HRD 300 может определить соответствие потока битов стандарту кодирования видео. В других примерах операция HRD 300 может определить соответствие декодера стандарту кодирования видео. В качестве части выполнения операции HRD 300, устройство может определить наивысший временной идентификатор поднабора потока битов, ассоциированного с выбранной рабочей точкой потока битов (302). Кроме того, устройство может определить, на основании наивысшего временного идентификатора, конкретный элемент синтаксиса из числа массива элементов синтаксиса (например, `sps_max_num_reorder_pics [i]`, `sps_max_dec_pic_buffering [i]`, и `spb_cnt_minus1 [i]`) (304). Устройство может использовать конкретный элемент синтаксиса в операции HRD (306).

[0213] В одном или более примерах, описанные функции могут быть реализованы в аппаратном обеспечении, программном обеспечении, программно-аппаратных средствах, или любой их комбинации. Если реализованы в программном обеспечении, функции могут быть сохранены на или переданы, как одна или более инструкций или код, считываемом компьютером носителе и выполнены основанным на аппаратном обеспечении блоком обработки. Считываемый компьютером носитель может включать в себя считываемые компьютером запоминающие носители, которые соответствуют материальному носителю, такому как запоминающие носители данных, или коммуникационные носители, включая любой носитель, который облегчает передачу компьютерной программы от одного места к другому, например, согласно протоколу связи. В этом способе считываемый компьютером носитель вообще может соответствовать (1) материальным считываемым компьютером запоминающим носителям, которые являются невременными, или (2) коммуникационному носителю, такому как сигнал или несущая. Запоминающие носители данных могут быть любым доступным носителем, к которому может 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.

[0214] Посредством примера, и не ограничения, такие считываемые компьютером запоминающие носители могут содержать RAM, ROM, EEPROM, CD-ROM или другой накопитель на оптических дисках, накопитель на магнитных дисках, или другие магнитные 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 веб-сайта, сервера, или другого удаленного источника, используя коаксиальный кабель, волоконно-оптический кабель, витую пару, цифровую абонентскую линию (DSL), или беспроводные технологии, такие как инфракрасное, радио-, и микроволновое излучение, то эти коаксиальный кабель, волоконно-оптический кабель, витая пара, DSL, или беспроводные технологии, такие как инфракрасное, радио-, 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 переходные носители, но вместо этого направлены на непереходные, материальные запоминающие носители. Диск и диск, как используется в настоящем описании, включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), дискету и диск Blu-ray, где диски (disks) обычно воспроизводят данные магнитно, в то время как диски (dicks) воспроизводят данные оптически с помощью лазеров. Комбинации вышеупомянутого должны также быть включены в рамки считываемого компьютером носителя.

[0215] Инструкции могут быть выполнены одним или более процессорами, такими как один или более цифровые сигнальные процессоры (DSPs), микропроцессоры общего назначения, специализированные интегральные схемы (ASICs), программируемые пользователем логические матрицы (FPGAs), или другая эквивалентная интегрированная или дискретная логическая схема. Соответственно, термин «процессор», как используется в настоящем описании, 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 в настоящем описании, могут быть предоставлены в пределах специализированного аппаратного обеспечения и/или программных модулей, конфигурируемых для кодирования и декодирования, или встроенных в объединенный кодек. Кроме того, способы могут быть полностью реализованы в одной или более схемах или логические элементах.

[0216] 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 потребовать реализации различными блоками аппаратного обеспечения. Вместо этого, как описано выше, различные блоки могут быть объединены в блоке аппаратного обеспечения кодека или предоставлены объединением взаимодействующих блоков аппаратного обеспечения, включая один или более процессоров, как описано выше, в соединении с подходящим программным обеспечением и/или программно-аппаратными средствами.

[0217] Были описаны различные примеры. Эти и другие примеры попадают в объем нижеследующей формулы изобретения.

(57) Формула изобретения

1. Способ обработки видеоданных, причем способ содержит:

прием набора параметров видео (VPS) и набора параметров последовательности (SPS), причем VPS содержит множество синтаксических структур параметров гипотетического эталонного декодера (HRD), содержащих соответствующие наборы параметров HRD, причем VPS дополнительно включает в себя, для каждой соответствующей синтаксической структуры параметров HRD в VPS, элементы синтаксиса, указывающие соответствующий набор идентификаторов уровня для соответствующей синтаксической структуры параметров HRD, и SPS содержит синтаксическую структуру параметров HRD, содержащую набор параметров HRD;

выбор набора параметров HRD, применимых к некоторой конкретной рабочей точке, при этом набор параметров HRD в наборе синтаксической структуры параметров HRD в SPS применим к упомянутой конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки идентичен набору из всех идентификаторов уровня, присутствующих в потоке битов, который подвергается тестированию с помощью теста соответствия потока битов, и набор параметров HRD в некоторой конкретной синтаксической структуре параметров HRD из упомянутого множества синтаксических структур параметров HRD в VPS применим к упомянутой конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки задан элементами синтаксиса, указывающими идентификаторы уровня для упомянутой конкретной синтаксической структуры параметров HRD в VPS; и

выполнение операции HRD с использованием выбранного набора параметров HRD.

2. Способ по п. 1, в котором набор параметров HRD, применимых к конкретной рабочей точке, включает в себя по меньшей мере параметры, указывающие начальную задержку удаления из буфера кодирования изображений (CPB), размер CPB и частоту следования битов.

3. Способ по п. 1, дополнительно содержащий:

определение целевого временного идентификатора конкретной рабочей точки, который равен наибольшему временному идентификатору в потоке битов.

4. Способ по п. 1, в котором выполнение операции HRD содержит выполнение теста соответствия потока битов, который определяет, соответствует ли поток битов стандарту кодирования видео.

5. Способ по п. 4, в котором выполнение теста соответствия потока битов содержит определение, что поток битов не соответствует стандарту кодирования видео, когда для всех наборов параметров HRD в кодированной видеопоследовательности больше чем один набор параметров HRD применяется к одной и той же рабочей точке.

6. Способ по п. 4, в котором выполнение теста соответствия потока битов содержит определение, что поток битов не соответствует стандарту кодирования видео, когда больше чем один набор параметров HRD в VPS применяется к одной и той же рабочей точке.

7. Способ по п. 4, в котором выполнение теста соответствия потока битов содержит определение, что поток битов не соответствует стандарту кодирования видео, когда VPS включает в себя набор параметров HRD, который применяется к рабочим точкам, имеющим наборы идентификаторов уровня, содержащие только значение 0.

8. Устройство обработки видеоданных, содержащее:

запоминающий носитель, хранящий закодированные видеоданные; и

один или более процессоров, сконфигурированных, чтобы:

принимать набор параметров видео (VPS) и набор параметров последовательности (SPS), причем VPS содержит множество синтаксических структур параметров гипотетического эталонного декодера (HRD), содержащих соответствующие наборы параметров HRD, причем VPS дополнительно включает в себя, для каждой соответствующей синтаксической структуры параметров HRD в VPS, элементы синтаксиса, указывающие соответствующий набор идентификаторов уровня для соответствующей синтаксической структуры параметров HRD, и SPS содержит синтаксическую структуру параметров HRD, содержащую набор параметров HRD;

выбирать набор параметров HRD, применимых к некоторой конкретной рабочей точке, при этом один или более процессоров сконфигурированы, чтобы определять, что набор параметров HRD в синтаксической структуре параметров HRD в SPS применим к упомянутой конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки идентичен набору из всех идентификаторов уровня, присутствующих в потоке битов, который подвергается тесту соответствия потока битов, и набор параметров HRD в некоторой конкретной синтаксической структуре параметров HRD из упомянутого множества синтаксических структур параметров HRD в VPS применим к упомянутой конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки задан элементами синтаксиса, указывающими идентификаторы уровня для упомянутой конкретной синтаксической структуры параметров HRD в VPS; и

выполнять операцию HRD с использованием выбранного набора параметров HRD.

9. Устройство по п. 8, в котором набор параметров HRD, применимых к конкретной рабочей точке, включает в себя по меньшей мере параметры, указывающие начальную задержку удаления из буфера кодирования изображений (CPB), размер CPB и частоту следования битов.

10. Устройство по п. 8, в котором один или более процессоров сконфигурированы, чтобы:

определять целевой временной идентификатор конкретной рабочей точки, который равен наибольшему временному идентификатору, присутствующему в потоке битов.

11. Устройство по п. 8, в котором один или более процессоров сконфигурированы, чтобы выполнить тест соответствия потока битов, который определяет, соответствует ли поток битов стандарту кодирования видео.

12. Устройство по п. 11, в котором один или более процессоров сконфигурированы, чтобы определить, что поток битов не соответствует стандарту кодирования видео, когда для всех наборов параметров HRD в кодированной видеопоследовательности больше чем один набор параметров HRD применяется к одной и той же рабочей точке.

13. Устройство по п. 11, в котором один или более процессоров сконфигурированы, чтобы определить, что поток битов не соответствует стандарту кодирования видео, когда больше чем один набор параметров HRD в VPS применяется к одной и той же рабочей точке.

14. Устройство по п. 11, в котором один или более процессоров сконфигурированы, чтобы определить, что поток битов не соответствует стандарту кодирования видео, когда VPS включает в себя набор параметров HRD, который применяется к рабочим точкам, имеющим наборы идентификаторов уровня, содержащие только значение 0.

15. Устройство обработки видеоданных, содержащее:

средство для приема набора параметров видео (VPS) и набора параметров последовательности (SPS), причем VPS содержит множество синтаксических структур

параметров гипотетического эталонного декодера (HRD), содержащих соответствующие наборы параметров HRD, причем VPS дополнительно включает в себя, для каждой соответствующей синтаксической структуры параметров HRD в VPS, элементы синтаксиса, указывающие соответствующий набор идентификаторов уровня для

5 соответствующей синтаксической структуры параметров HRD, и SPS содержит синтаксическую структуру параметров HRD, содержащую набор параметров HRD;

средство для выбора набора параметров HRD, применимых к некоторой конкретной рабочей точке, при этом набор параметров HRD в наборе синтаксической структуры параметров HRD в SPS применим к упомянутой конкретной рабочей точке на основе

10 того, что набор идентификаторов уровня этой конкретной рабочей точки идентичен набору из всех идентификаторов уровня, присутствующих в потоке битов, который подвергается тестированию с помощью теста соответствия потока битов, и набор параметров HRD в некоторой конкретной синтаксической структуре параметров HRD из упомянутого множества синтаксических структур параметров HRD в VPS применим

15 к упомянутой конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки задан элементами синтаксиса, указывающими идентификаторы уровня для упомянутой конкретной синтаксической структуры параметров HRD в VPS; и

средство для выполнения операции HRD с использованием выбранного набора

20 параметров HRD.

16. Устройство по п. 15, дополнительно содержащее:

средство для определения целевого временного идентификатора конкретной рабочей точки, который равен наибольшему временному идентификатору, присутствующему в потоке битов.

17. Считываемый компьютером запоминающий носитель данных, имеющий инструкции для обработки видеоданных, сохраненные на нем, которые, когда выполняются одним или более процессорами устройства, конфигурируют это устройство:

принимать набор параметров видео (VPS) и набор параметров последовательности (SPS), причем VPS содержит множество синтаксических структур параметров гипотетического эталонного декодера (HRD), содержащих соответствующие наборы параметров HRD, причем VPS дополнительно включает в себя, для каждой соответствующей синтаксической структуры параметров HRD в VPS, элементы синтаксиса, указывающие соответствующий набор идентификаторов уровня для

30 соответствующей синтаксической структуры параметров HRD, и SPS содержит синтаксическую структуру параметров HRD, содержащую набор параметров HRD;

выбирать набор параметров HRD, применимых к некоторой конкретной рабочей точке, при этом набор параметров HRD в наборе синтаксической структуры параметров HRD в SPS применим к упомянутой конкретной рабочей точке на основе того, что

40 набор идентификаторов уровня этой конкретной рабочей точки идентичен набору всех идентификаторов уровня, присутствующих в потоке битов, который подвергается тестированию с помощью теста соответствия потока битов, и набор параметров HRD в некоторой конкретной синтаксической структуре параметров HRD из упомянутого множества синтаксических структур параметров HRD в VPS применим к упомянутой

45 конкретной рабочей точке на основе того, что набор идентификаторов уровня этой конкретной рабочей точки задан элементами синтаксиса, указывающими идентификаторы уровня для упомянутой конкретной синтаксической структуры параметров HRD в VPS; и

выполнять операцию HRD с использованием выбранного набора параметров HRD.

18. Считываемый компьютером запоминающий носитель данных по п. 17, в котором инструкции конфигурируют устройство:

5 определять целевой временной идентификатор конкретной рабочей точки, который равен наибольшему временному идентификатору, присутствующему в потоке битов.

10

15

20

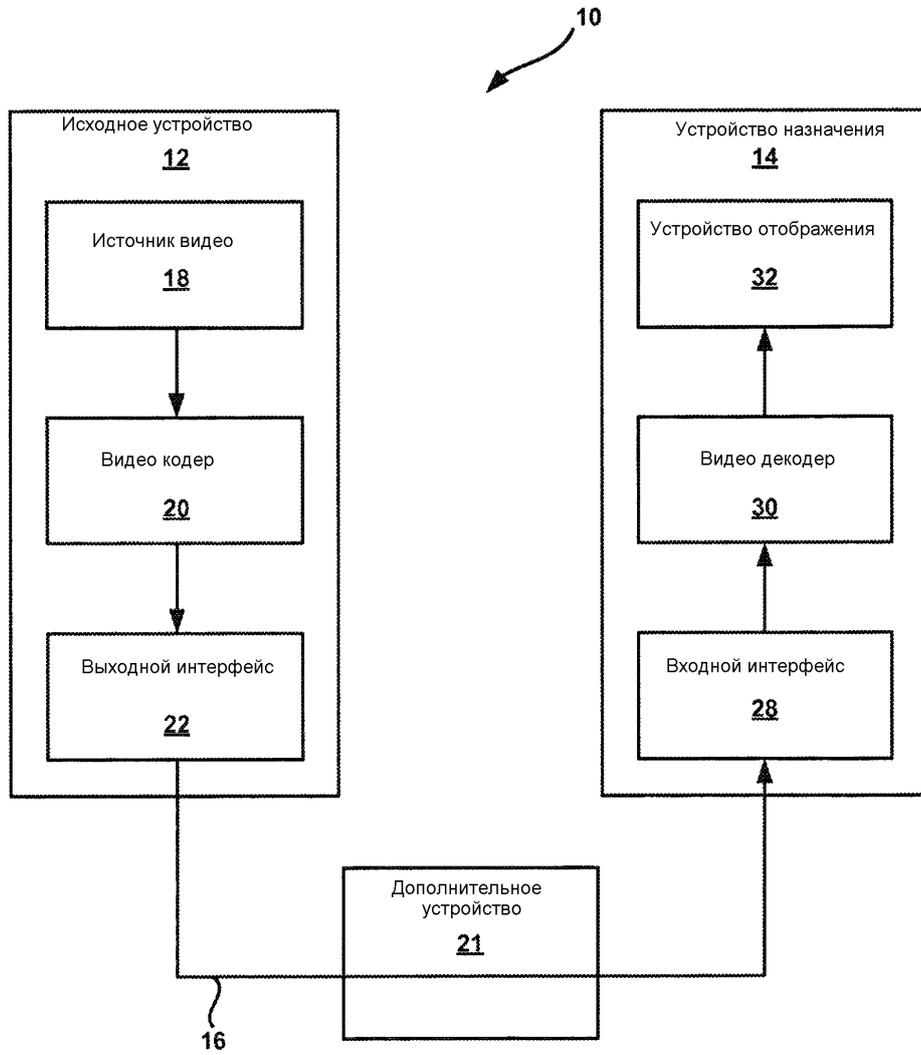
25

30

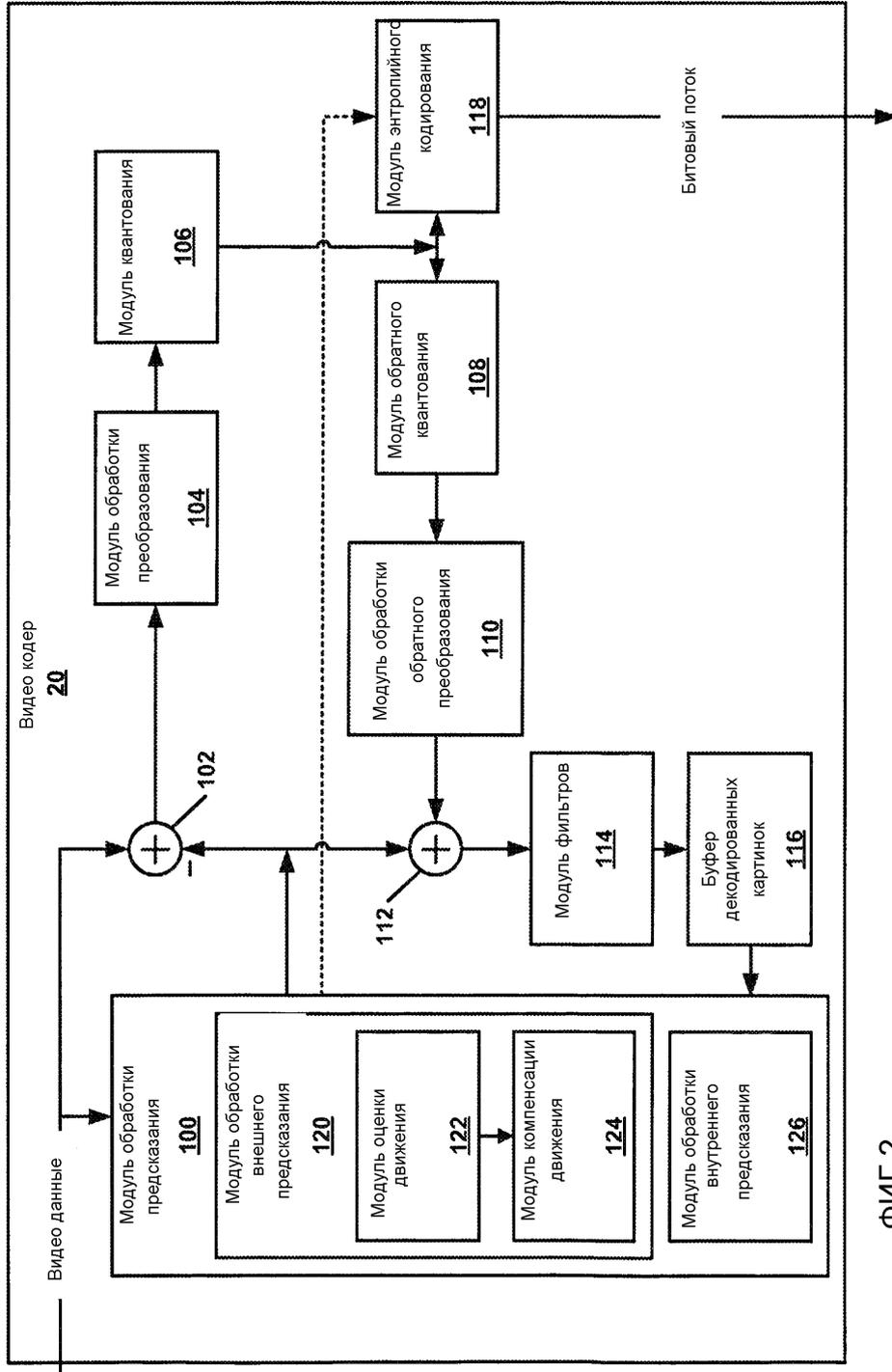
35

40

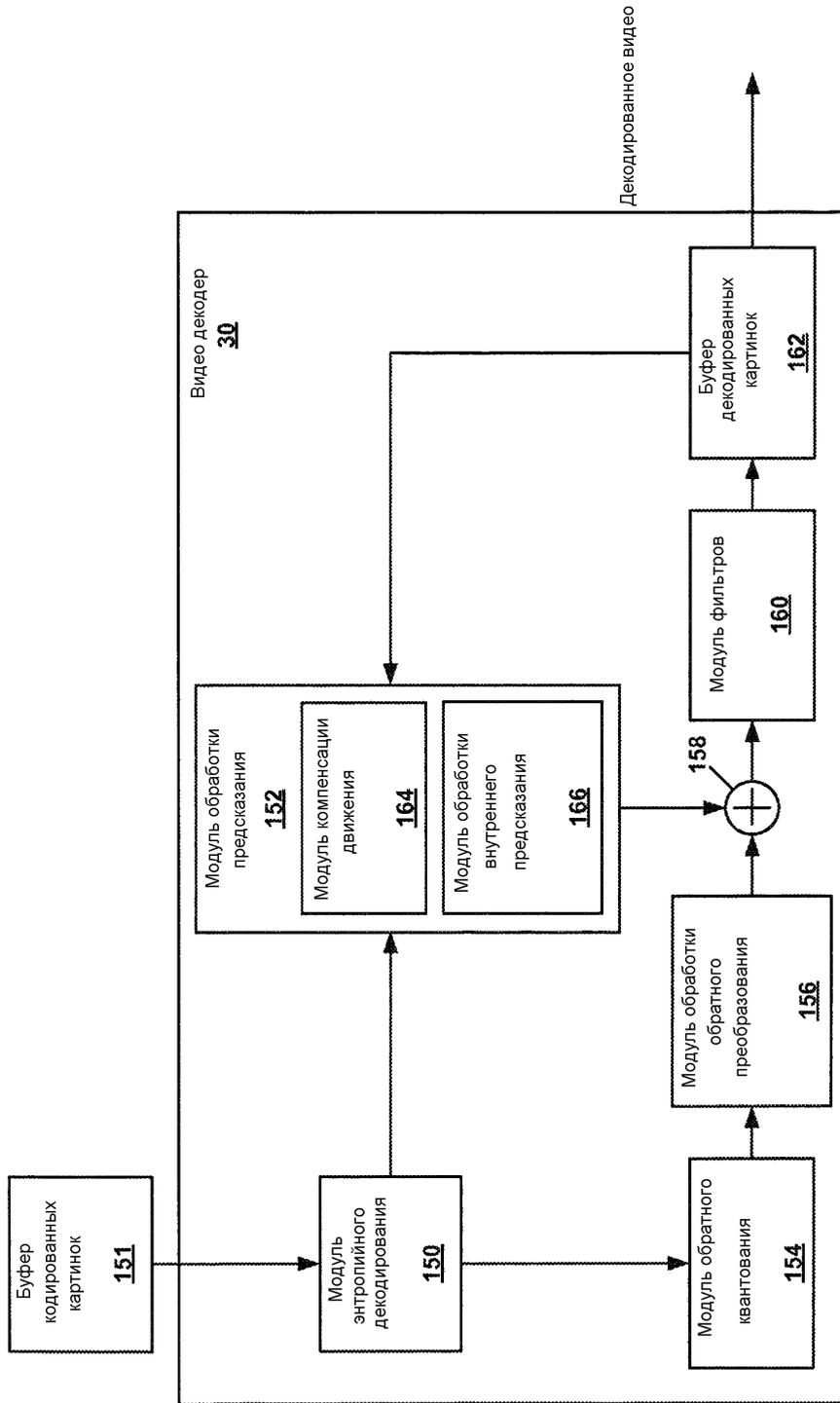
45



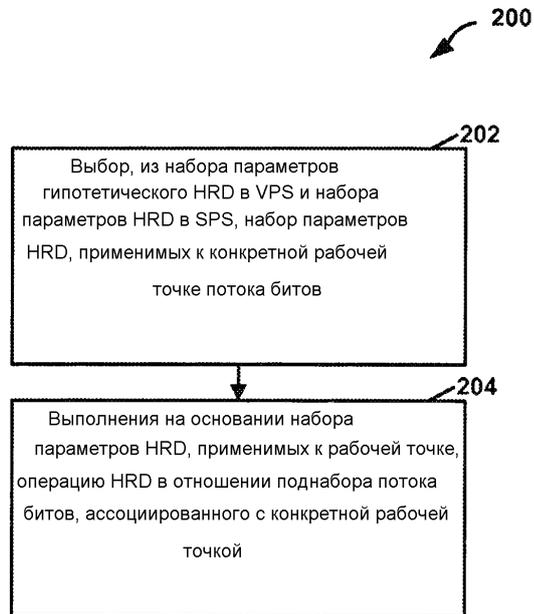
ФИГ.1



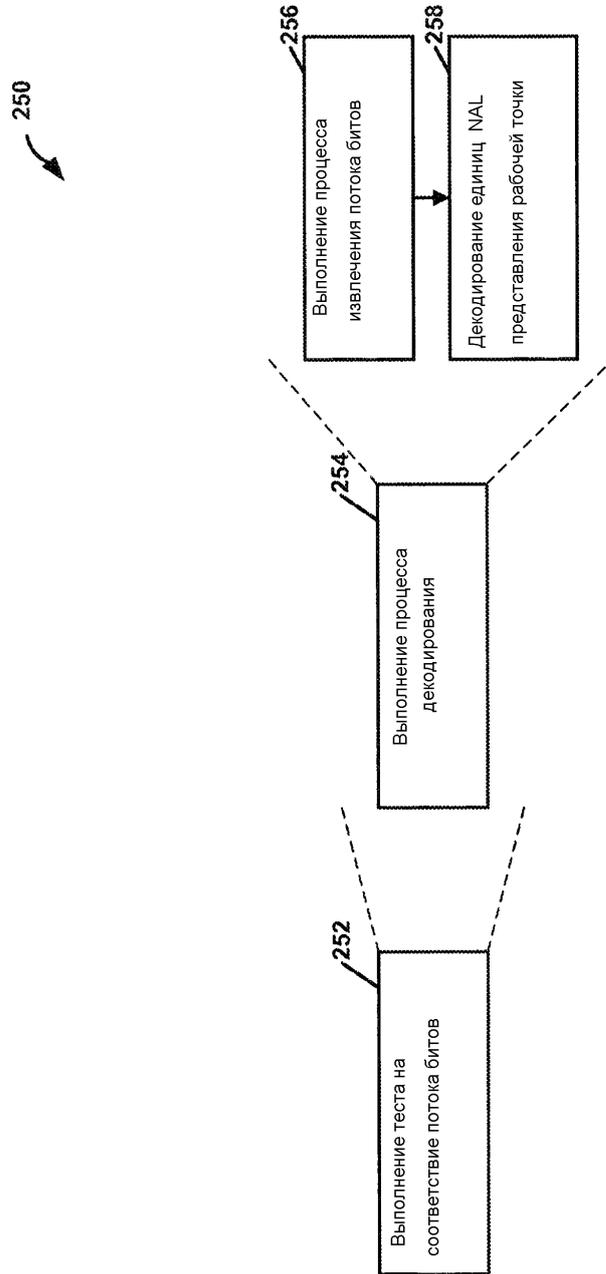
ФИГ. 2



ФИГ.3

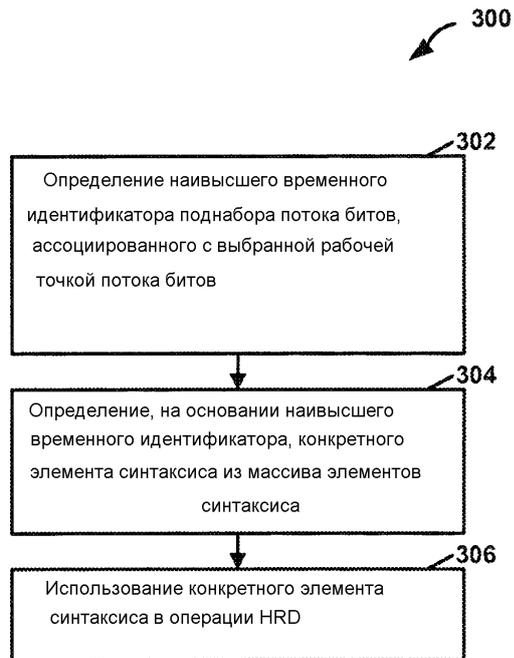


ФИГ.4



ФИГ.5

6/6



ФИГ.6