

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2014101346/08, 25.03.2010

(24) Дата начала отсчета срока действия патента:
25.03.2010

Приоритет(ы):

(30) Конвенционный приоритет:
03.04.2009 JP 2009-091245Номер и дата приоритета первоначальной заявки,
из которой данная заявка выделена:
2010148466 03.04.2009

(43) Дата публикации заявки: 27.07.2015 Бюл. № 21

(45) Опубликовано: 20.01.2016 Бюл. № 2

(56) Список документов, цитированных в отчете о
поиске: EP 1628485 A2, 22.02.2006. WO
2008153312 A2, 18.12.2008. US 5937095 A,
10.08.1999. WO 9732437 A1, 04.09.1997. RU
2337503 C1, 27.10.2008.

Адрес для переписки:

109012, Москва, ул. Ильинка, 5/2, ООО
"Союзпатент"

(72) Автор(ы):

СУЗУКИ Терухико (JP),
ХАТТОРИ Синобу (JP)

(73) Патентообладатель(и):

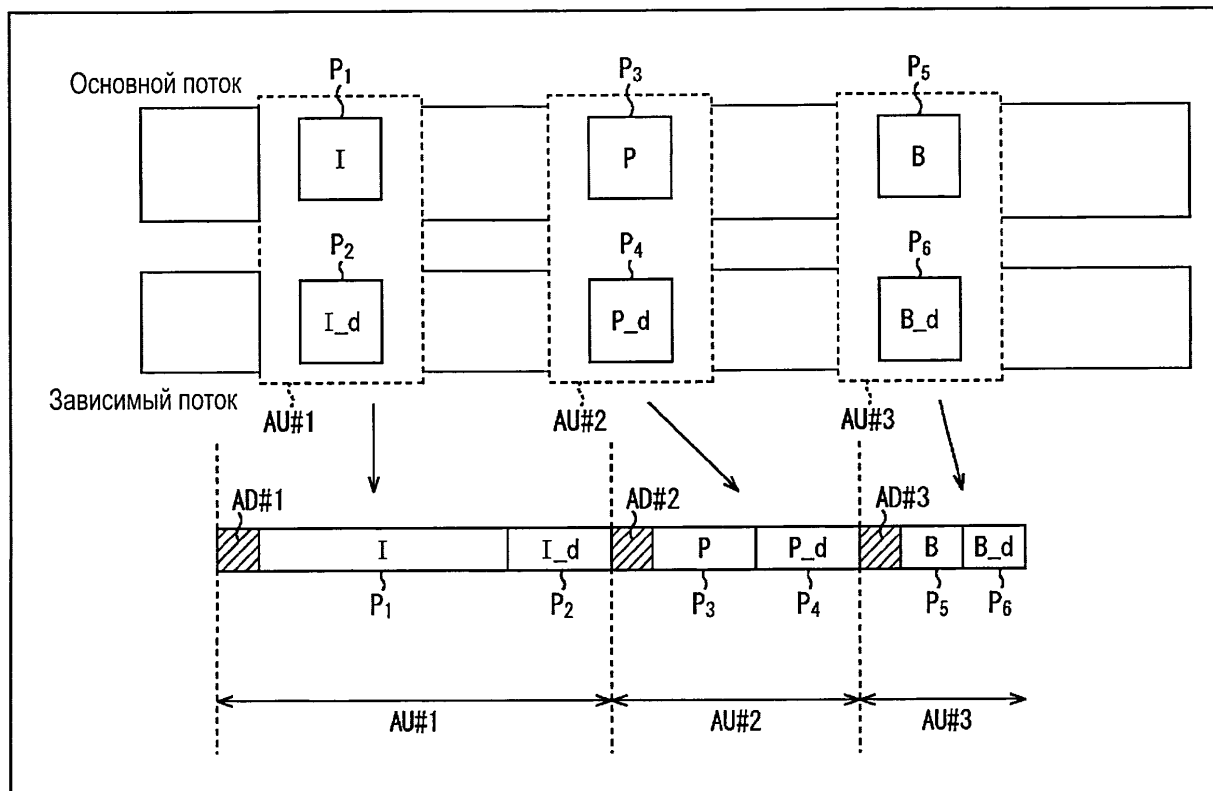
СОНИ КОРПОРЕЙШН (JP)

(54) УСТРОЙСТВО ДЕКОДИРОВАНИЯ СИГНАЛА ИЗОБРАЖЕНИЯ, СПОСОБ ДЕКОДИРОВАНИЯ
СИГНАЛА ИЗОБРАЖЕНИЯ, УСТРОЙСТВО КОДИРОВАНИЯ СИГНАЛА ИЗОБРАЖЕНИЯ, СПОСОБ
КОДИРОВАНИЯ СИГНАЛА ИЗОБРАЖЕНИЯ И ПРОГРАММА

(57) Реферат:

Настоящее изобретение относится к устройству декодирования сигнала изображения, способу декодирования сигнала изображения, устройству кодирования сигнала изображения, способу кодирования сигнала изображения и программе, которые обеспечивают простое детектирование границ изображения зависимого потока в потоке битов. Технический результат - детектирование границ изображения в зависимом потоке по потоку битов с недопущением переполнения или недозаполнения буфера. В потоке битов AD (разграничитель AU) помещают

в начальное положение каждого AU (блок доступа), а DD (зависимый разграничитель) помещают на границе кадра зависимого потока. Начало блока NAL детектируют путем детектирования начального кода и DD, то есть границу кадра (начало) зависимого потока детектируют путем проверки, равно ли 18 значение nal_unit_type, начало которого было детектировано. Настоящее изобретение можно применять к устройству, которое обрабатывает сигнал стереоскопического изображения. 3 н.п. ф-лы, 32 ил.



Фиг. 13



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2014101346/08, 25.03.2010**

(24) Effective date for property rights:
25.03.2010

Priority:

(30) Convention priority:
03.04.2009 JP 2009-091245

Number and date of priority of the initial application,
from which the given application is allocated:
2010148466 03.04.2009

(43) Application published: **27.07.2015** Bull. № 21

(45) Date of publication: **20.01.2016** Bull. № 2

Mail address:

109012, Moskva, ul. Il'inka, 5/2, OOO "Sojuzpatent"

(72) Inventor(s):

**SUZUKI Terukhiko (JP),
KhATTORI Sinobu (JP)**

(73) Proprietor(s):

SONI KORPOREJShN (JP)

(54) **IMAGE SIGNAL DECODING DEVICE, IMAGE SIGNAL DECODING METHOD, IMAGE SIGNAL ENCODING DEVICE, IMAGE SIGNAL ENCODING METHOD AND PROGRAMME**

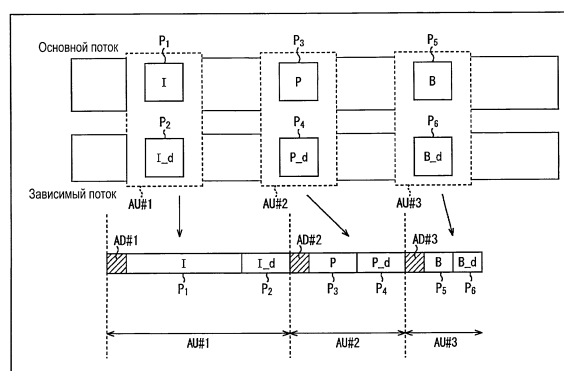
(57) Abstract:

FIELD: physics, computer engineering.

SUBSTANCE: present invention relates to an image signal decoding device, an image signal decoding method, an image signal encoding device, an image signal encoding method and a programme, which provide simple image boundary detection dependent on a stream in a bit stream. In a bit stream, an AD (AU delimiter) is inserted at the start position of each AU (access unit), and a DD (dependent delimiter) is inserted on the frame boundary of the dependent stream. The start of the NAL unit is detected by detecting the initial code and the DD, i.e. the frame boundary (start) of the dependent stream is detected by checking if the value `nal_unit_type`, the start of which was detected, is equal to 18. The present invention can be applied to the device which processes a stereoscopic image signal.

EFFECT: detecting image boundaries in a dependent stream from a bit stream with the prevention of buffer overflow or underflow.

3 cl, 32 dwg



Фиг. 13

Область техники, к которой относится изобретение

Настоящее изобретение относится к устройству декодирования сигнала изображения, способу декодирования сигнала изображения, устройству кодирования сигнала изображения, способу кодирования сигнала изображения и программе, и более конкретно к устройству декодирования сигнала изображения, способу декодирования сигнала изображения, устройству кодирования сигнала изображения, способу кодирования сигнала изображения и программе, которые позволяют легко определять границы изображения зависимого потока в потоке битов.

Уровень техники

В последние годы устройства, соответствующие схемам сжатия на основе ортогонального преобразования, такого как дискретное косинусное преобразование и компенсации движения, такого как MPEG, используя избыточность, присущую информации видеоизображения, с целью реализации высокоэффективной передачи и накопления информации, все в большей степени преобладают при использовании как при распределении информации от станций широковещательной передачи или тому подобного, так и при приеме информации в обычных домах потребителей.

В частности, стандарт MPEG2 (ISO/IEC 13818-2) определен как схема кодирования изображения общего назначения и охватывает как изображения с чересстрочной разверткой, так и изображения с прогрессивной разверткой, а также изображения со стандартной четкостью и изображения с высокой четкостью, которые в настоящее время широко используются в самых разных приложениях, включая в себя профессиональные приложения и потребительские приложения.

При использовании MPEG2 возможно достичь высокой степени сжатия и высокого качества воплощения изображений, например, путем назначения кодовой скорости (скорости передачи битов) 4-8 Мбит/с для изображений с чересстрочной разверткой стандартной четкостью, имеющих 720×480 пикселей. Кроме того, высокая степень сжатия и высокое качество воплощения изображений возможны при назначении скорости кода 18-22 Мбит/с для изображения с прогрессивной разверткой и высокой четкостью, имеющего 1920×1088 пикселей.

MPEG2 в основном был предназначен для кодирования с высокой четкостью, пригодного для широковещательной передачи, но не поддерживает схемы кодирования с более низкой кодовой скоростью, то есть с более высокой степенью сжатия, чем MPEG1.

По мере увеличения популярности мобильных терминалов потребность в таких схемах кодирования в будущем будет увеличиваться. Для удовлетворения этого спроса была стандартизирована схема кодирования MPEG4. Для схем кодирования изображения был принят стандарт ISO/IEC 14496-2 как международный стандарт в декабре 1998 г.

Кроме того, также был принят стандарт, называемый H.264/AVC (MPEG-4 часть 10, ISO/IEC 14496-10 | ITU-T H. 264). Этот стандарт разработан организацией под названием JVT (Объединенная группа по обработке видеоизображений), которая была совместно учреждена ITU-T и ISO/IEC для стандартизации кодирования видеоданных.

Известно, что H.264/AVC требует большего количества вычислений при кодировании и декодировании, чем в обычных схемах кодирования, таких как MPEG2 и MPEG4, но позволяет достигать более высокой эффективности кодирования.

[H.264/AVC]

На фиг. 1 показана блок-схема, иллюстрирующая пример конфигурации устройства кодирования видеoinформации, которое реализует сжатие изображения на основе ортогонального преобразования, такого как дискретное косинусное преобразование

или преобразование Кархунена - Лоэва, и компенсации движения.

1 обозначает модуль A/D преобразования, 2 обозначает буфер изменения порядка вывода на экран, 3 обозначает модуль сумматора, 4 обозначает модуль ортогонального преобразования, 5 обозначает модуль дискретизации, 6 обозначает модуль кодирования без потерь, 7 обозначает буфер накопления, 8 обозначает модуль обратной дискретизации, 9 обозначает модуль обратного ортогонального преобразования, 10 обозначает запоминающее устройство кадра, 11 обозначает модуль прогнозирования/компенсации движения и 12 обозначает модуль управления скоростью передачи данных.

Вводимый сигнал изображения вначале преобразуют в цифровой сигнал с помощью модуля 1 A/D преобразования.

Затем выполняют изменение порядка вывода кадров с помощью буфера 2 изменения порядка вывода на экран в соответствии со структурой GOP (Группа кадров) выводимой информации сжатия изображения.

Для изображения, подвергаемого внутрикадровому кодированию, информацию видеозображения для всех кадров вводят в модуль 4 ортогонального преобразования, в котором выполняют ортогональное преобразование, такое как дискретное косинусное преобразование или преобразование Кархунена - Лоэва.

Коэффициент преобразования, который выводят из модуля 4 ортогонального преобразования, подвергают обработке дискретизации с помощью модуля 5 дискретизации.

Дискретизированный коэффициент преобразования, который выводят из модуля 5 дискретизации, подают в модуль 6 кодирования без потерь, где выполняют кодирование без потерь, такое как кодирование с переменной длиной кодирования или арифметическое кодирование. После этого полученный в результате коэффициент преобразования накапливают в буфере 7 накопления и выводят как информацию сжатия изображения. Работой модуля 5 дискретизации управляют с помощью модуля 12 управления скоростью передачи данных.

Одновременно дискретизированный коэффициент преобразования, который выводят из модуля 5 дискретизации, подают в модуль 8 обратной дискретизации и, кроме того, подвергают обработке обратного ортогонального преобразования, используя модуль 9 обратного ортогонального преобразования, в информацию декодированного видеозображения. Эту информацию накапливают в запоминающем устройстве 10 кадра.

Изображение, подвергаемое межкадровому кодированию, вначале выводят из буфера 2 изменения порядка вывода на экран в модуль 11 прогнозирования/компенсации движения.

Одновременно информацию изображения, на которую будет сделана ссылка, получают из запоминающего устройства 10 кадра и подвергают обработке прогнозирования/компенсации движения. Таким образом, генерируют информацию опорного изображения.

Информацию опорного изображения передают в модуль 3 сумматора и преобразуют здесь в сигнал разности между информацией опорного изображения и информацией изображения.

Модуль 11 компенсации/прогнозирования движения одновременно выводит информацию вектора движения в модуль 6 кодирования без потерь. Информацию вектора движения подвергают обработке кодирования без потерь, такой как кодирование переменной длины или арифметическое кодирование, и вводят в участок заголовка информации сжатия изображения. Другая обработка аналогична выполняемой

для информации сжатия изображения, которую подвергают внутрикадровому кодированию.

На фиг. 2 показана блок-схема, иллюстрирующая конфигурацию примера устройства декодирования информации изображения.

21 обозначает буфер накопления, 22 обозначает модуль кодирования/декодирования без потерь, 23 обозначает модуль обратного квантования, 24 обозначает модуль обратного ортогонального преобразования, 25 обозначает модуль сумматора, 26 обозначает буфер изменения порядка вывода на экран, 27 обозначает модуль ЦА преобразования, 28 обозначает запоминающее устройство кадра и 29 обозначает модуль прогнозирования/компенсации движения.

Вводимую информацию сжатия изображения (поток битов) вначале сохраняют в буфере 21 накопления и после этого передают в модуль 22 кодирования/декодирования без потерь.

В модуле 22 кодирования/декодирования без потерь выполняют обработку, такую как декодирование переменной длины или арифметическое декодирование, в соответствии с определенным форматом информации сжатия изображения.

Одновременно, если кадр представляет собой кадр, подвергнутый межкадровому кодированию, модуль 22 кодирования/декодирования без потерь также декодирует информацию вектора движения, сохраненную на участке заголовка информации сжатия изображения, и выводит эту информацию в модуль 29 прогнозирования/компенсации движения.

Дискретизированный коэффициент преобразования, выводимый из модуля 22 кодирования/декодирования без потерь, подают в модуль 23 обратной дискретизации и здесь выводят как коэффициент преобразования.

Коэффициент преобразования подвергают обратному ортогональному преобразованию, такому как обратное дискретное косинусное преобразование или обратное преобразование Кархунена - Лоэва, с помощью модуля 24 обратного ортогонального преобразования, в соответствии с определенным форматом информации сжатия изображения.

В случае, когда кадр представляет собой кадр, кодированный внутри кадра, информацию изображения, подвергаемую обработке обратного ортогонального преобразования, сохраняют в буфере 26 изменения порядка вывода на экран и выводят после обработки ЦА преобразования.

В случае, когда кадр представляет собой кадр, подвергнутый межкадровому кодированию, опорное изображение генерируют на основе информации вектора движения, подвергнутой процессу декодирования без потерь, и информации изображения, сохраненной в запоминающем устройстве 28 кадра. Опорное изображение и выход модуля 24 обратного ортогонального преобразования комбинируют с помощью модуля 25 сумматора. Другая обработка аналогична выполняемой для кадра, кодированного внутри кадра.

Описанный выше стандарт AVC, разработанный JVT, представляет собой гибридную схему кодирования, сформированную из компенсации движения и дискретного косинусного преобразования, такого как MPEG2 или MPEG4.

Дискретное косинусное преобразование может представлять собой целочисленное преобразование, аппроксимирующее настоящее дискретное косинусное преобразование. Хотя детальные схемы отличаются, например, тем, что способ преобразования при дискретном косинусном преобразовании представляет собой способ, в котором используется целочисленный коэффициент с размером блока 4x4 или с переменным

размером блока при компенсации движения, основная схема аналогична схеме кодирования, воплощенной в конфигурации, показанной на фиг. 1.

В то же время в последние годы по мере развития технологий съемки и отображения стереоскопического изображения были проведены исследования по разработке расширения стандарта H.264/AVC для кодирования сигналов стереоскопического изображения.

В настоящее время разрабатывают стандарты MVC (Кодирование видеоданных, снятых с множества точек съемки), которое обеспечивает кодирование изображений, снятых с множества точек съемки, которые были сняты с использованием множества устройств съемки изображения.

Изображение, для которого предполагается, что оно было снято и отображается с двух точек съемки, называется стереоизображением. Стереоскопический дисплей, предназначенный для обзора невооруженным глазом, выполнен с возможностью поддержки отображения, снятого с множества точек съемки.

Хотя следующее описание будет приведено, в основном, для стереоизображения, снятого с двух точек съемки, в качестве примера применение изображений, снятых с множества точек съемки, полученных с трех или более точек съемки, может быть выполнено аналогичным образом.

[MVC]

На фиг. 3 показана схема, иллюстрирующая устройство кодирования изображения, снятого с множества точек съемки.

В устройстве 41 кодирования изображения, снятого с множества точек съемки, кодируют видеосигналы, подаваемые из двух устройств съемки изображения, а именно из устройств 31 и 32 съемки изображения, и выводят потоки битов, генерируемые в результате кодирования. Эти потоки битов, состоящие из данных изображений, снятых с двух точек съемки, могут быть мультиплексированы в один поток, который затем выводят, или могут быть выведены как два или больше потока битов.

На фиг. 4 показана блок-схема, иллюстрирующая пример конфигурации устройства 41 кодирования изображения, снятого с множества точек съемки по фиг. 3.

В устройстве 41 кодирования изображения, снятого с множества точек съемки, изображение, снятое с одной точки съемки, из изображений, снятых с множества точек съемки, кодируют как основной поток, а другие изображения кодируют как зависимые потоки.

В случае стереоскопического изображения одно изображение из изображения L (изображения, снятого с левой точки съемки) и изображения R (изображения, снятого с правой точки съемки), кодируют как основной поток, и другое изображение кодируют как зависимый поток.

Основной поток представляет собой поток битов, аналогичный существующему потоку битов AVC, закодированному с использованием стандарта H.264 AVC/High Profile или тому подобному. Поэтому основной поток становится потоком, который можно декодировать, используя существующий декодер AVC, который поддерживает стандарт H.264 AVC/High Profile.

Изображения, кодируемые как основной поток, вводят в буфер 51 изменения порядка следования и изменяют их порядок следования для того, чтобы сделать пригодными для кодирования, как изображения I, изображения P и изображения B. Изображения с измененным порядком следования выводят в модуль 52 видеокодирования.

Модуль 52 видеокодирования имеет конфигурацию, аналогичную устройству кодирования информации изображения по фиг. 1. В модуле 52 видеокодирования,

например, выполняют кодирование в соответствии с H.2 64 AVC/High Profile, и полученный в результате поток битов выводят в модуль 57 мультиплексирования. Кроме того, локально декодированные изображения сохраняют в запоминающем устройстве 53 кадра и используют как опорное изображение для кодирования

5 следующего изображения или изображения в зависимом потоке.

В то же время изображения, которые должны быть кодированы как зависимый поток, вводят в буфер 54 изменения порядка следования и изменяют их порядок следования для того, чтобы сделать пригодными для кодирования, как изображение I, изображение P и изображение B. Изображения с измененным порядком следования

10 выводят в модуль 55 кодирования зависимого потока.

В модуле 55 кодирования зависимого потока в дополнение к нормальному кодированию AVC выполняют кодирование с использованием в качестве опорного изображения локального декодированного изображения в основном потоке, сохраненном в запоминающем устройстве 53 кадра, и выводят поток битов в модуль

15 57 мультиплексирования. Кроме того, локально декодированное изображение сохраняют в запоминающем устройстве 56 кадра и используют как опорное изображение для кодирования следующего изображения.

В модуле 57 мультиплексирования основной поток и зависимый поток мультиплексируют в один поток битов, который выводят. Основной поток и зависимый

20 поток могут быть выведены как отдельные потоки битов.

На фиг. 5 показана схема, иллюстрирующая пример опорного изображения MVC.

Основной поток кодируют путем выполнения только прогнозирования в направлении времени, аналогично выполняемому в нормальном AVC.

Зависимый поток кодируют путем выполнения в дополнение к прогнозированию в

25 направлении времени в пределах изображения, снятого с той же точки съемки, которое аналогично прогнозированию в нормальном AVC, прогнозирования, используя изображение в основном потоке, которое получено в тот же момент времени, что и опорное изображение. Даже в случае, когда прогнозирование в направлении времени не может быть соответствующим образом выполнено, возможность обращения к

30 изображению, снятому с другой точки съемки, полученному в тот же момент времени, может улучшить эффективность кодирования.

На фиг. 6 показана блок-схема, иллюстрирующая конфигурацию модуля 52 видеокодирования по фиг. 4, который генерирует основной поток, и запоминающее устройство 53 кадра.

Конфигурация, показанная на фиг. 6, аналогична конфигурации устройства кодирования информации изображения по фиг. 1, за исключением того, что к изображению, сохраненному в запоминающем устройстве 53 кадра, обращается модуль

35 55 кодирования зависимого потока.

На фиг. 7 показана блок-схема, иллюстрирующая конфигурацию модуля 55 кодирования зависимого потока по фиг. 4, который генерирует зависимый поток, и

40 запоминающее устройство 56 кадра.

Конфигурация, показанная на фиг. 7, аналогична конфигурации устройства кодирования информации изображения по фиг. 1, за исключением того, что на изображение, сохраненное в запоминающем устройстве 53 кадра, можно ссылаться.

Опорное изображение, считываемое из запоминающего устройства 53 кадра, вводят в модуль 90 прогнозирования/компенсации движения и используют для прогнозирования

45 движения и компенсации движения.

На фиг. 8 показана блок-схема, иллюстрирующая пример конфигурации устройства

101 декодирования изображения, снятого с множества точек съемки.

Основной поток, подаваемый из устройства 41 кодирования изображения, снятого с множества точек съемки, через сеть или носитель записи, вводят в буфер 111 и зависимый поток вводят в буфер 114. В случае, когда подают один

5 мультиплексированный поток, этот поток разделяют на основной поток и зависимый поток, которые вводят в буфер 111 и буфер 114 соответственно.

Основной поток, который задерживают в буфере 111 на заданный период времени, выводят в модуль 112 декодирования видеоданных.

В модуле 112 декодирования видеоданных основной поток декодируют в соответствии с AVC, и полученное в результате декодированное изображение сохраняют в

10 запоминающем устройстве 113 кадра. Декодированное изображение, сохраненное в запоминающем устройстве 113 кадра, используют как опорное изображение для декодирования следующего изображения или изображения в зависимом потоке.

Декодированное изображение, полученное модулем 112 декодирования видеоданных,

15 выводят как видеосигнал в 3D дисплей 102 в заданные моменты времени.

В то же время зависимый поток, который задерживают в буфере 114 на заданный период времени, выводят в модуль 115 декодирования зависимого потока.

В модуле 115 декодирования зависимого потока зависимый поток декодируют и полученное в результате декодированное изображение сохраняют в запоминающем

20 устройстве 116 кадра. Декодированное изображение, сохраненное в запоминающем устройстве 116 кадра, используют как опорное изображение для декодирования следующего изображения.

В модуле 115 декодирования зависимого потока при необходимости изображение, сохраненное в запоминающем устройстве 113 кадра, используют как опорное

25 изображение в соответствии с информацией (такой как флаг) в потоке битов.

Декодированное изображение, полученное модулем 115 декодирования зависимого потока, выводят как видеосигнал в 3D дисплей 102 в заданные моменты времени.

В 3D дисплее 102 стереоизображение отображают в соответствии с видеосигналом, подаваемым из модуля 112 декодирования видеоданных, и видеосигналом, подаваемым

30 из модуля 115 декодирования зависимого потока.

На фиг. 9 показана схема, иллюстрирующая конфигурацию модуля 112 декодирования видеоданных по фиг. 8, который декодирует основной поток, и запоминающее устройство 113 кадра.

Конфигурация, показанная на фиг. 9, аналогична конфигурации устройства

35 декодирования информации изображения по фиг. 2, за исключением того момента, что к изображению, сохраненному в запоминающем устройстве 113 кадра, обращается модуль 115 декодирования зависимого потока.

На фиг. 10 показана блок-схема, иллюстрирующая конфигурацию модуля 115 декодирования зависимого потока по фиг. 8, который декодирует зависимый поток, и

40 запоминающего устройства 116 кадра.

Конфигурация, показанная на фиг. 10, аналогична конфигурации устройства декодирования информации изображения по фиг. 2, за исключением того, что можно обращаться к изображению, сохраненному в запоминающем устройстве 113 кадра. Опорное изображение, считываемое из запоминающего устройства 113 кадра, вводят

45 в модуль 148 прогнозирования/компенсации движения и используют для прогнозирования движения и компенсации движения.

Список ссылок

Патентная литература

PTL 1: Публикация №2007-208917 нерассмотренной заявки на японский патент

Сущность изобретения

Техническая задача изобретения

В AVC и MVC определена виртуальная модель буфера, которая имитирует операцию
5 буфера на стороне устройства декодирования для недопущения переполнения или недозаполнения буфера.

На стороне устройства кодирования кодирование выполняют таким образом, чтобы
предотвратить переполнение или недозаполнение виртуального буфера. Устройство
декодирования может декодировать поток битов, кодированный таким образом, без
10 нарушения работы буфера.

Ниже будет описана виртуальная модель буфера.

На фиг. 11 показана схема, иллюстрирующая пример виртуальной модели буфера.

Входной поток битов вначале подают в буфер 151. Поток битов считывают
непосредственно из буфера 151 блоками, называемыми AU (блоки доступа), и подают
15 модуль 152 декодирования видеоданных.

В модуле 152 декодирования видеоданных данные, считываемые из буфера 151,
мгновенно декодируют и генерируют декодированное изображение. Приведенное выше
предположение делают в виртуальной модели буфера.

Следует отметить, что буфер 151 на фиг. 11 соответствует буферам 111 и 114 в
20 устройстве 101 декодирования изображений, снятых с множества точек съемки по фиг.
8. Кроме того, модуль 152 декодирования видеоданных по фиг. 11 соответствует модулю
112 декодирования видеоданных и модулю 115 декодирования зависимого потока в
устройстве 101 декодирования изображений, снятых с множества точек съемки по фиг.
8.

25 В случае AVC AU хранит данные одного кадра.

В случае MVC один AU хранит данные всех видов (основной поток и зависимый
поток), полученных одновременно. Здесь термин «вид» означает изображение,
полученное с каждой точки съемки.

Поведение буфера 151 в такой виртуальной модели буфера показано на фиг. 12. На
30 фиг. 12 по вертикальной оси представлена степень заполнения буфера, и по
горизонтальной оси представлено время.

Как показано на фиг. 12, поток битов подают в буфер 151 с заданной скоростью
подачи битов. Кроме того, во время считывания каждого AU поток битов, сохраненный
в соответствующем AU, извлекают из буфера 151. Моменты времени t_1 , t_2 , t_3 ...
35 представляют собой моменты времени считывания из буфера 151.

На фиг. 13 показана схема, иллюстрирующая конфигурацию AU потока битов,
кодированного с использованием MVC.

В верхней части фиг. 13 представлены AU №1, №2 и №3.

[0082] AU №1 состоит из кадра P_1 в основном потоке и кадра P_2 в зависимом потоке,
40 которые представляют собой кадры, снятые в один и тот же момент времени (данные
кадров сохранены).

AU №2 состоит из кадра P_3 в основном потоке и кадра P_4 в зависимом потоке, которые
представляют собой кадры, снятые в один момент времени.

45 AU №3 состоит из кадра P_5 в основном потоке и кадра P_6 в зависимом потоке, которые
представляют собой кадры, снятые в один момент времени.

В нижней части на фиг. 13 иллюстрируется конфигурация потока битов.

Поток битов на фиг. 13 представляет собой один поток, составленный в результате

мультиплексирования основного потока и зависимого потока. Каждый AU имеет блок NAL, называемый AD (разграничитель AU), вставленный в его начале. AD представляет собой некоторую уникальную строку данных.

AD представляет начальное положение данных кадра в основном потоке и кадра в зависимом потоке, которые находятся в одной и той же точке времени.

Как показано на фиг. 13, в потоке битов AU №1 состоит из AD №1, кадра P₁ и кадра P₂.

AU №2 состоит из AD №2, кадра P₃ и кадра P₄.

AU №3 состоит из AD №3, кадра P₅ и кадра P₆.

Поиск AD позволяет найти начало AU и легко осуществлять доступ к определенному кадру.

На фиг. 14 представлены схемы, иллюстрирующие пример структуры потоков битов.

В части А на фиг. 14 показана структура в случае, когда основной поток и зависимый поток мультиплексируют в единый поток битов. Структура потока битов в части А на фиг. 14 та же, что и структура потока битов на фиг. 13.

В части В на фиг. 14 иллюстрируется структура в случае, когда основной поток и зависимый поток отдельно включены в два потока битов. Участки, соответствующие участкам, показанным в части А на фиг. 14, обозначены одинаковыми номерами ссылочных позиций.

Как показано в части В на фиг. 14, основной поток выполнен путем размещения AD №1, кадра P₁, AD №2, кадра P₃, AD №3 и кадра P₅ в данном порядке.

В то же время зависимый поток выполнен путем размещения кадра P₂, кадра P₄ и кадра P₆ в данном порядке.

Процесс декодирования для битовых потоков на фиг. 14 будет описан со ссылкой на конфигурацию, показанную на фиг. 11.

Например, в случае, когда поток битов в части А на фиг. 14 подают после детектирования AD №1, модуль 152 декодирования видеоданных на фиг. 11 последовательно считывает кадр P₁ в основном потоке и кадр P₂ в зависимом потоке и по отдельности декодирует эти кадры.

Кроме того, в случае, когда подают битовые потоки, показанные в части В на фиг. 14, для первого AU в основном потоке модуль 152 декодирования видеоданных детектирует AD №1 и считывает и декодирует кадр P₁. Кроме того, для второго AU модуль 152 декодирования видеоданных детектирует AD №2 и считывает и декодирует кадр P₃.

Поскольку зависимый поток не содержит AD, необходимо анализировать синтаксис в потоке битов и определять границы кадров P₂, P₄ и P₆ для считывания каждого кадра. Поэтому обработка становится очень сложной.

Здесь рассматривается случай, в котором AD просто добавляют к началу каждого кадра в зависимом потоке, аналогично основному потоку, таким образом, что начало кадра может быть легко детектировано.

В этом случае, например, если основной поток и зависимый поток мультиплексируют в один поток битов, модуль 152 видеodeкодирования может распознавать AU основного потока и AU зависимого потока как разные AU. Это не позволяет правильно воспроизвести работу виртуальной модели буфера.

Настоящее изобретение было разработано с учетом такой ситуации и предназначено для обеспечения возможности простого детектирования границ изображения в

зависимом потоке по потоку битов.

Решение задачи

Устройство декодирования сигнала изображения в одном аспекте настоящего изобретения представляет собой устройство декодирования сигнала изображения, которое декодирует поток битов, формируемый путем кодирования сигнала стереоскопического изображения, включающего в себя сигналы изображения, полученные с множества точек съемки, включающее в себя средство декодирования для декодирования первой уникальной строки данных, указывающей начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени, причем первая уникальная строка данных кодирована в изображении, полученном с одной точки съемки, для определения начала кодированных данных, полученных в указанный заданный момент времени, для декодирования второй уникальной строки данных, указывающей начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, причем вторая уникальная строка данных отдельно кодирована в начале кодированного потока битов изображения, полученного с другой точки съемки, для определения начала кодированных данных сигнала изображения, полученного с другой точки съемки, и для декодирования потока битов.

Сигналы изображения, полученные с множества точек съемки, могут представлять собой сигналы изображения, полученные с двух точек съемки, и они выполнены в виде одного потока битов.

Сигналы изображения, полученные с множества точек съемки, могут представлять собой сигналы изображения, полученные с двух точек съемки, и они выполнены в виде двух потоков битов.

Средство декодирования может быть выполнено с возможностью декодирования первой уникальной строки данных, указывающей начало кодированных потоков битов изображения, полученных со всех точек съемки в указанный заданный момент времени, причем первая уникальная строка данных кодирована в изображении, полученном с одной из точки съемки, определения начала кодированных данных, полученных в указанный заданный момент времени, декодирования второй уникальной строки данных, указывающей начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, причем вторая уникальная строка данных, кодирована в начале кодированного потока битов изображения, полученного с другой точки съемки, определения начала кодированных данных сигнала изображения, полученного с другой точки съемки, и декодирования потока битов.

Средство декодирования может быть выполнено с возможностью в случае доступа к положению, в котором декодирование может быть начато правильно, декодирования первой уникальной строки данных, указывающей начало кодированных потоков битов изображения, полученных со всех точек съемки в указанный заданный момент времени, причем первая уникальная строка данных кодирована в изображении, полученном с одной точки съемки, определения начала кодированных данных, полученных в указанный заданный момент времени с одной точки съемки, декодирования второй уникальной строки данных, указывающей начало кодированного потока битов изображения, полученного с другой точки съемки, в указанный заданный момент времени, причем вторая уникальная строка данных отдельно кодирована в начале кодированного потока битов изображения, полученного с другой точки съемки, определения начала кодированных данных сигнала изображения, полученного с другой

точки съемки, и декодирования потока битов от определенных начальных положений.

Способ декодирования сигнала изображения в этом аспекте настоящего изобретения представляет собой способ декодирования сигнала изображения для декодирования потока битов, формируемого путем кодирования сигнала стереоскопического изображения, включающего в себя сигналы изображения, полученные с множества точек съемки, включающий в себя следующие этапы: декодируют первую уникальную строку данных, указывающую начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени, причем первая уникальная строка данных кодирована в изображении, полученном с одной точки съемки; определяют начало кодированных данных, полученных в указанный заданный момент времени; декодируют вторую уникальную строку данных, указывающую начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, причем вторая уникальная строка данных отдельно кодирована в начале кодированного потока битов изображения, полученного с другой точки съемки; определяют начало кодированных данных сигнала изображения, полученного с другой точки съемки; и декодируют поток битов.

Программа в этом аспекте настоящего изобретения представляет собой программу, обеспечивающую выполнение компьютером процесса декодирования потока битов, получаемого в результате кодирования сигнала стереоскопического изображения, включающего в себя сигналы изображения, полученные с множества точек съемки, при этом программа вызывает выполнение компьютером процесса, содержащего этапы, на которых декодируют первую уникальную строку данных, указывающую начало кодированного потока битов изображения, полученного со всех точек съемки в заданный момент времени, причем первая уникальная строка данных кодирована в изображении, полученном с одной точки съемки; определяют начало кодированных данных, полученных в указанный заданный момент времени; декодируют вторую уникальную строку данных, указывающую начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, причем вторая уникальная строка данных отдельно кодирована в начале кодированного потока битов изображения, полученного с другой точки съемки; определяют начало кодированных данных сигнала изображения, полученного с другой точки съемки; и декодируют поток битов.

Устройство кодирования сигнала изображения в другом аспекте настоящего изобретения представляет собой устройство кодирования сигнала изображения, которое кодирует сигнал стереоскопического изображения, включающий в себя сигналы изображения, полученные с множества точек съемки, включающее в себя средство кодирования для кодирования первой уникальной строки данных в изображении, полученном с одной точки съемки, причем первая уникальная строка данных указывает начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени, и для отдельного кодирования второй уникальной строки данных в начале кодированного потока битов изображения, полученного с другой точки съемки, причем вторая уникальная строка данных указывает начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени.

Средство кодирования может быть выполнено с возможностью мультиплексирования кодированных данных, полученных со всех точек съемки, в один поток битов.

Средство кодирования может быть выполнено с возможностью обеспечения включения кодированных данных, полученных со всех точек съемки, в два потока

битов.

Сигналы изображения, полученные с множества точек съемки, могут представлять собой сигналы изображения, полученные с двух точек съемки.

Способ кодирования сигнала изображения в этом другом аспекте настоящего изобретения представляет собой способ кодирования сигнала изображения для кодирования сигнала стереоскопического изображения, включающего в себя сигналы изображения, полученные с множества точек съемки, включающий в себя следующие этапы: кодируют первую уникальную строку данных в изображении, полученном с одной точки съемки, причем первая уникальная строка данных указывает начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени; и отдельно кодируют вторую уникальную строку данных в начале кодированного потока битов изображения, полученного с другой точки съемки, причем вторая уникальная строка данных указывает начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени.

Программа в этом другом аспекте настоящего изобретения представляет собой программу, вызывающую выполнение компьютером процесса кодирования сигнала стереоскопического изображения, включающего в себя сигналы изображения, полученные с множества точек съемки, причем программа вызывает выполнение компьютером процесса, содержащую следующие этапы: кодируют первую уникальную строку данных в изображении, полученном с одной точки съемки, причем первая уникальная строка данных указывает начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени; и отдельно кодируют вторую уникальную строку данных в начале кодированного потока битов изображения, полученного с другой точки съемки, причем вторая уникальная строка данных указывает начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени.

В одном аспекте настоящего изобретения декодируют первую уникальную строку данных, указывающую начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени, и кодированную в изображении, полученном с одной точки съемки, и определяют начало кодированных данных, полученных в указанный заданный момент времени. Кроме того, декодируют вторую уникальную строку данных, указывающую начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, и отдельно кодированную в начале кодированного потока битов изображения, полученного с другой точки съемки, определяют начало кодированных данных сигнала изображения, полученного с другой точки съемки, и декодируют поток битов.

В другом аспекте настоящего изобретения в изображении, полученном с одной точки съемки, кодируют первую уникальную строку данных, указывающую начало кодированных потоков битов изображения, полученных со всех точек съемки в заданный момент времени, и отдельно кодируют вторую уникальную строку данных, указывающую начало кодированного потока битов изображения, полученного с другой точки съемки в указанный заданный момент времени, в начале кодированного потока битов изображения, полученного с другой точки съемки.

Предпочтительные результаты изобретения

В соответствии с аспектом настоящего изобретения можно легко определять границы кадров в зависимом потоке в потоке битов.

Краткое описание чертежей

На фиг. 1 показана блок-схема, иллюстрирующая пример конфигурации устройства кодирования информации изображения.

На фиг. 2 показана блок-схема, иллюстрирующая пример конфигурации устройства декодирования информации изображения.

5 На фиг. 3 показана схема, иллюстрирующая устройство кодирования изображения, снятого с множества точек съемки.

На фиг. 4 показана блок-схема, иллюстрирующая примерную конфигурацию устройства кодирования изображения, снятого с множества точек съемки по фиг. 3.

На фиг. 5 показана схема, иллюстрирующая пример опорного изображения MVC.

10 На фиг. 6 показана блок-схема, иллюстрирующая конфигурацию модуля кодирования видеоданных по фиг. 4 и запоминающего устройства кадра.

На фиг. 7 показана блок-схема, иллюстрирующая конфигурацию модуля кодирования зависимого потока по фиг. 4 и запоминающего устройства кадра.

15 На фиг. 8 показана блок-схема, иллюстрирующая пример конфигурации устройства декодирования данных для множества точек съемки.

На фиг. 9 показана схема, иллюстрирующая конфигурацию модуля декодирования видеоданных по фиг. 8 и запоминающего устройства кадра.

На фиг. 10 показана блок-схема, иллюстрирующая конфигурацию модуля декодирования зависимого потока по фиг. 8 и запоминающего устройства кадра.

20 На фиг. 11 показана схема, иллюстрирующая пример виртуальной модели буфера.

На фиг. 12 показана схема, иллюстрирующая поведение буфера по фиг. 11.

На фиг. 13 показана схема, иллюстрирующая конфигурацию AU потока битов, кодированных с использованием MVC.

Фиг. 14 включает в себя схемы, иллюстрирующие пример структуры потоков битов.

25 На фиг. 15 показана блок-схема, иллюстрирующая примерную конфигурацию модуля кодирования зависимого потока, который представляет собой часть устройства кодирования изображения, снятого с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения.

Фиг. 16 включает в себя схемы, иллюстрирующие пример структуры потоков битов.

30 На фиг. 17 показана схема, иллюстрирующая пример конфигурации модуля NAL.

На фиг. 18 показана схема, иллюстрирующая синтаксис модуля NAL.

На фиг. 19 показана схема, иллюстрирующая определение nal_unit_type.

На фиг. 20 показана схема, иллюстрирующая определение nal_unit_type.

На фиг. 21 показана схема, иллюстрирующая упрощенный синтаксис модуля NAL.

35 На фиг. 22 показана схема, иллюстрирующая синтаксис, включающий в себя модуль NAL.

На фиг. 23 показана схема, иллюстрирующая упрощение синтаксиса по фиг. 22.

40 На фиг. 24 показана блок-схема, иллюстрирующая пример конфигурации модуля декодирования зависимого потока, который представляет собой часть устройства декодирования изображения, полученного с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения.

На фиг. 25 показана схема, иллюстрирующая пример виртуальной модели буфера.

На фиг. 26 показана схема, иллюстрирующая пример способа доступа к началу AU.

45 На фиг. 27 показана блок-схема последовательности операций, описывающая операцию модуля анализа синтаксиса.

На фиг. 28 показана блок-схема последовательности операций, описывающая процесс обычного анализа.

На фиг. 29 показана схема, иллюстрирующая конфигурацию AU в потоках битов,

полученных путем кодирования видеоданных, полученных с трех точек съемки, используя MVC.

Фиг. 30 включает в себя схемы, иллюстрирующие пример структуры потоков битов.

Фиг. 31 включает в себя схемы, иллюстрирующие пример структуры потоков битов.

5 На фиг. 32 показана блок-схема, иллюстрирующая примерную конфигурацию компьютера.

Подробное описание изобретения

Конфигурация устройства кодирования изображения, снятого с множества точек съемки

10 На фиг. 15 показана блок-схема, иллюстрирующая пример конфигурации модуля кодирования зависимого потока, который представляет собой часть устройства кодирования изображения, снятого с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения.

15 В устройстве кодирования изображения, снятого с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения в основном потоке кодирована информация, указывающая границы AU, и в зависимом потоке кодирована информация, указывающая границы между кадрами в зависимом потоке.

20 Это позволяет устройству декодирования изображения, снятого с множества точек съемки, при декодировании потоков битов стереоскопического изображения определять границы AU или тому подобное в результате простой обработки и выполнять операцию декодирования на основе виртуальной модели буфера.

Устройство кодирования изображения, снятого с множества точек съемки, имеет такую же конфигурацию, что и конфигурация, показанная на фиг. 4. На фиг. 15 иллюстрируется примерная конфигурация модуля 55 кодирования зависимого потока по фиг. 4. Модуль 52 кодирования видеоданных устройства 41 кодирования изображения, снятого с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения имеет такую же конфигурацию, что и конфигурация на фиг. 6. Одинаковые элементы конфигурации обозначены одинаковыми номерами ссылочных позиций.

30 Конфигурация модуля 55 кодирования зависимого потока, показанная на фиг. 15, является той же конфигурацией, что и конфигурация на фиг. 7, за исключением того, что предусмотрен модуль 201 кодирования DD и что выходной сигнал модуля 201 кодирования DD подадут в модуль 86 кодирования без потерь.

35 Модуль 201 кодирования DD вставляет блок NAL, называемый DD (зависимый разграничитель) (блок NAL разграничения представления по виду и зависимости), в начало каждого кадра в зависимом потоке. DD представляет собой уникальную строку данных, такую как AD, но имеет значение, отличающееся от AD.

40 Модуль 86 кодирования без потерь вставляет блок NAL DD, генерируемый модулем 201 кодирования DD, в начало каждого изображения, выполняет кодирование и выводит зависимый поток, в который был вставлен DD.

Фиг. 16 включает в себя схемы, иллюстрирующие пример структуры потоков битов, генерируемых устройством 41 кодирования изображений, снятых с множества точек съемки, включающим в себя модуль 55 кодирования зависимого потока по фиг. 15.

45 В части А на фиг. 16 показан пример структуры в случае, когда основной поток и зависимый поток мультиплексированы в один поток битов.

В части В на фиг. 16 показан пример структуры в случае, когда основной поток и зависимый поток отдельно включены в два потока битов. Участки, соответствующие участкам, представленным в части А на фиг. 16, обозначены теми же номерами

ссылочных позиций.

В случае генерирования потока битов в части А на фиг. 16 после кодирования AD №1 устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует кадр P₁ в основном потоке. Затем устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует DD №1 перед кодированием кадра в зависимом потоке и после этого кодирует кадр P₂ в зависимом потоке.

Когда цель процесса переключают к следующему AU, устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует AD №2 и кодирует кадр P₃ в основном потоке. Затем устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует DD №2 перед кодированием кадра в зависимом потоке и после этого кодирует кадр P₄ в зависимом потоке.

Аналогично устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует DD перед кодированием кадров в зависимом потоке и размещает эти кадры в зависимом потоке непосредственно после DD.

Также в этом примере AD размещены в начале AU. Первый AU состоит из AD №1, кадра P₁, DD №1 и кадра P₂, а второй AU состоит из AD №2, кадра P₃, DD №2 и кадра P₄. Третий AU состоит из AD №3, кадра P₅, DD №3 и кадра P₆.

В случае генерирования потоков битов в части В на фиг. 16 для основного потока после кодирования AD №1 устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует кадр P₁ в основном потоке. Когда цель процесса переключают на следующий AU, устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует AD №2 и кодирует кадр P₃ в основном потоке.

Кроме того, для зависимого потока после кодирования DD №1 устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует кадр P₂ в зависимом потоке. Когда цель процесса переключают на следующий AU, устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует DD №2 и кодирует кадр P₄ в зависимом потоке.

Аналогичным образом устройство 41 кодирования изображения, снятого с множества точек съемки, кодирует DD перед кодированием кадров в зависимом потоке и размещает кадры в зависимом потоке непосредственно после DD.

Как представлено в части В на фиг. 16, основной поток выполнен путем размещения AD №1, кадра P₁, AD №2, кадра P₃, AD №3 и кадра P₅ в указанном порядке.

В то же время зависимый поток выполнен путем размещения DD №1, кадра P₂, DD №2, кадра P₄, DD №3 и кадра P₆ в указанном порядке.

Также в примере в части В на фиг. 16 первый AU состоит из AD №1, кадра P₁, DD №1 и кадра P₂, а второй AU состоит из AD №2, кадра P₃, DD №2 и кадра P₄. Третий AU состоит из AD №3, кадра P₅, DD №3 и кадра P₆.

AD представляет собой начальное положение данных кадра в основном потоке и кадра в зависимом потоке, которые получают в один и тот же момент времени, то есть представляет начальное положение потоков битов кодирования изображения, полученных со всех точек съемки в заданный момент времени. А DD представляет собой положение границы данных кадра в зависимом потоке, то есть представляет начальное положение потока битов кодирования изображения в зависимом потоке.

Синтаксис

Синтаксис DD будет описан ниже.

Все кодированные элементы данных, включая AD и кадры, сохраняют в модуле, называемом модулем NAL, и изменяют их порядок следования в заданном порядке для конфигурирования потока битов. Детали описаны в ISO/IEC 14496-10 | ITU-T H.264.

DD также сохраняют в блоке NAL.

5 На фиг. 17 показана схема, иллюстрирующая пример конфигурации модуля NAL.

Кодированные элементы данных и заголовок обрабатывают побайтно.

NAL_BytesInNALunit представляет собой параметр, указывающий размер блока NAL в байтах, который необходим для идентификации границ блока NAL.

Один способ идентификации границ блока NAL состоит в поиске начальных кодов.

10 В других случаях необходимо передавать размер блока NAL в декодер, используя определенный способ. `rbsp_byte[]` представляет кодированные элементы данных или заголовок, которые содержатся в блоке NAL.

На фиг. 18 показана схема, иллюстрирующая синтаксис блока NAL.

`forbidden_zero_bit` представляет собой 1-битные данные, которые всегда равны 0.

15 `nal_ref_idc` представляет собой 2-битные данные, которые указывают, что на элементы данных в блоке NAL ссылаются другие блоки NAL, если значение `nal_ref_idc` представляет собой другое значение, кроме 0. Если значение `nal_ref_idc` равно 0, другие блоки NAL не ссылаются на эти элементы данных.

`nal_unit_type` представляет собой 5-битные данные, которые указывают содержание 20 блока NAL.

`forbidden_zero_bit`, `nal_ref_idc` и `nal_unit_type` составляют заголовок по фиг. 17.

На фиг. 19 и 20 показаны схемы, иллюстрирующие определение `nal_unit_type`.

Как показано на фиг. 19, `nal_unit_type=9` обозначает, что содержание блока NAL представляет собой AD.

25 Как показано на фиг. 20, `nal_unit_type=18` обозначает, что содержание блока NAL - это DD.

Упрощенный синтаксис блока NAL на фиг. 18 представлен на фиг. 21.

К блоку NAL добавлен начальный код в соответствии с синтаксисом, представленным на фиг. 22, и он составляет поток, называемый потоком байтов. Поток байтов 30 соответствует потоку битов, описанному выше.

`startcode_prefix` (`start_code__prefix_one_3` байта) из 3 байтов (0x000001) по фиг. 22 представляет собой начальный код, который должен быть добавлен к началу блока NAL. `startcode_prefix` представляет собой уникальный код, который не должен появляться в других местах в потоке байтов.

35 Поиск `startcode_prefix` в потоке байтов приводит к детектированию начала блока NAL. Кроме того, проверка `nal_unit_type` в заголовке блока NAL, начало которого было детектировано, позволяет обратиться к требуемому блоку NAL.

Упрощение синтаксиса на фиг. 22 выполнено так, как иллюстрируется на фиг. 23.

40 Таким образом, DD размещены на границах кадров в зависимом потоке, независимо от того, является ли зависимый поток мультимплексированным с основным потоком в одиночный поток битов или включен в другой поток битов.

Начальный код добавляют к началу блока NAL, в котором содержится DD, и значение `nal_unit_type` блока NAL устанавливают равным 18, что обозначает, что содержание представляет собой DD.

45 Устройство декодирования для множества точек съемки может детектировать начало блока NAL путем детектирования начального кода. Кроме того, проверка, что значение `nal_unit_type` блока NAL, начало которого было детектировано, равно 18, позволяет детектировать DD, то есть границу (начало) изображения в зависимом потоке.

Устройство декодирования изображения, снятого с множества точек съемки

На фиг. 24 показана блок-схема, иллюстрирующая пример конфигурации модуля декодирования зависимого потока, который представляет собой часть устройства декодирования изображения, снятого с множества точек съемки, в соответствии с

5 вариантом осуществления настоящего изобретения.

Устройство декодирования изображения, снятого с множества точек съемки в соответствии с вариантом осуществления настоящего изобретения, имеет ту же конфигурацию, что и конфигурация, показанная на фиг. 8. На фиг. 24 представлена

10 примерная конфигурация модуля 115 декодирования зависимого потока по фиг. 8. Модуль 112 декодирования видеоданных устройства 101 декодирования изображения, снятого с множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения имеет ту же конфигурацию, что и конфигурация, показанная на фиг. 9.

Конфигурация модуля 115 декодирования зависимого потока, показанная на фиг. 15 24, является той же конфигурацией, что и конфигурация на фиг. 10, за исключением того, что DD выводят из модуля 142 кодирования/декодирования без потерь.

Модуль 142 кодирования/декодирования без потерь декодирует DD, включенный в зависимый поток, и выводит этот DD. На основе вывода DD начальное положение декодирования или тому подобное во время случайного доступа устанавливают с

20 помощью вышележащего приложения или тому подобного.

Модель виртуального буфера MVC будет описана ниже.

На фиг. 25 показана схема, иллюстрирующая пример модели виртуального буфера в случае, когда поток битов MVC состоит из двух потоков, то есть основного потока и зависимого потока.

25 Подаваемый на вход основной поток сохраняют в буфере 151A, а зависимый поток сохраняют в буфере 151B. Битовые потоки считывают мгновенно из буферов 151A и 151B по блокам AU и декодируют мгновенно с помощью устройства 152 декодирования видеоданных.

Следует отметить, что буфер 151A на фиг. 25 соответствует буферу 111 устройства 30 101 декодирования изображения, снятого с множества точек съемки, на фиг. 8, а буфер 151B на фиг. 25 соответствует буферу 114 устройства 101 декодирования изображения, снятого с множества точек съемки, на фиг. 8. Кроме того, модуль 152 декодирования видеоданных на фиг. 25 соответствует модулю 112 декодирования видеоданных и модулю 115 декодирования зависимого потока устройства 101 декодирования

35 изображения, снятого с множества точек съемки, по фиг. 8.

Например, в случае, когда основной поток в части В на фиг. 16 сохраняют в буфере 151A, а зависимый поток сохраняют в буфере 151B в определенный момент времени, кадр P_1 считывают из буфера 151A в соответствии с AD №1, а кадр P_2 считывают из буфера 151B в соответствии с DD №1. Считанные кадры декодируют с помощью модуля

40 152 видеodeкодирования.

Кроме того, в следующий момент времени кадр P_3 считывают из буфера 151A в соответствии с AD №2, а кадр P_4 считывают из буфера 151B в соответствии с DD №2.

Виртуальная модель буфера в случае, когда поток битов MVC состоит из одного потока, имеющего основной поток и зависимый поток, представляет собой то же, что

45 показано на фиг. 11.

Например, в случае, когда поток битов в части А на фиг. 16 сохраняют в буфере 151 на фиг. 11, в определенный момент времени кадры P_1 и P_2 считывают из буфера 151 в

соответствии с AD №1.

Кроме того, в следующий момент времени кадры P₃ и P₄ считывают из буфера 151 в соответствии с AD №2.

5 Далее, со ссылкой на фиг. 26 будет описан пример способа доступа к началу заданного AU во время случайного доступа в момент возникновения ошибки или тому подобного.

Входной поток битов вначале подают в буфер 211 и временно сохраняют. После этого поток битов подают в модуль 212 детектирования начального кода и в декодер 214.

10 Модуль 212 детектирования начального кода детектирует начальный код, имеющий уникальную строку данных. После детектирования начального кода модуль 212 детектирования начального кода выводит последовательные данные в модуль 213 анализа синтаксиса.

15 Модуль 213 анализа синтаксиса анализирует данные, подаваемые из модуля 212 детектирования начального кода. Поскольку данные следуют после начального кода, данные, подаваемые из модуля 212 детектирования начального кода, представляют собой данные блока NAL.

После определения начального положения AU в результате анализа заголовка или тому подобного блока NAL модуль 213 анализа синтаксиса выводит сигнал, 20 указывающий положение начала AU, в декодер 214 и в буфер 211.

В случае, когда присутствует AD (в случае, когда подают данные блока NAL), модуль 213 анализа синтаксиса проверяет, удовлетворяется или нет `nal_unit_type=9`. В случае, когда `nal_unit_type=9` удовлетворяется, как описано выше, содержание блока NAL представляет собой AD. Таким образом, детектируют начало AU.

25 Аналогично, в случае, когда присутствует AD, модуль 213 анализа синтаксиса проверяет, удовлетворяется ли `nal_unit_type=18`. В случае, когда `nal_unit_type=18` удовлетворяется, как описано выше, содержание блока NAL представляет собой DD. Таким образом, детектируют начало изображения в зависимом потоке.

30 Декодер 214 и буфер 211 последовательно выполняют декодирование изображений на основе блоков AU нормальным образом в соответствии с сигналом из модуля 213 анализа синтаксиса.

Операция детектирования начала изображения в зависимом потоке будет описана со ссылкой на блок-схему последовательности операций, показанную на фиг. 27.

35 Здесь, как показано в части В на фиг. 16, предполагается, что основной поток и зависимый поток составляют разные потоки битов, и что зависимый поток имеет кодированный DD.

На этапе S1 модуль 212 детектирования начального кода выполняет поиск начального кода.

40 После детектирования начального кода на этапе S2 модуль 213 анализа синтаксиса проверяет, удовлетворяется или нет `nal_unit_type=18`.

В случае, когда на этапе S2 определяют, что условие `nal_unit_type=18` удовлетворяется, на этапе S3, модуль 213 анализа синтаксиса детектирует, что блок NAL, имеющий детектированный начальный код, представляет собой блок, в котором хранится DD и который находится в начале изображения зависимого потока.

45 Операция в случае, когда, как показано в части В на фиг. 14, поток состоит из двух потоков битов, и когда, как в обычном MVC, зависимый поток не имеет DD, показана на фиг. 28.

Как показано на фиг. 28, в случае, когда зависимый поток не имеет DD, заголовки

сегмента также декодируют и выполняют множество обычных определений. В конечном итоге на этапе S23 становится возможным детектировать начало AU (начало кадра).

На фиг. 28 значения, используемые при соответствующих определениях, вплоть до этапа S23, представляют собой части информации, записанные в заголовках среза, сохраненные в RBSP (фиг. 17) блока NAL. В отличие от `nal_unit_type=18`, описанного выше, представленные выше части информации не записаны в заголовках модулей NAL, и поэтому, выполняется сложная обработка анализа.

Поток битов для трех точек съемки

Потоки битов для трех точек съемки будут описаны ниже.

На фиг. 29 показана схема, иллюстрирующая конфигурацию AU, полученных в результате кодирования видеоизображений для трех точек съемки, используя MVC.

Как описано выше, в случае MVC, в одном AU содержатся данные всех точек съемки, полученные в один момент времени. То же относится также и к случаю, когда количество точек съемки равно трем.

На фиг. 29 AU №11 состоит из кадра P_{11} в основном потоке, кадра P_{12} в зависимом потоке 1 и кадра P_{13} в зависимом потоке 2.

AU №12 состоит из кадра P_{14} в основном потоке, кадра P_{15} в зависимом потоке 1 и кадра P_{16} в зависимом потоке 2.

AU №13 состоит из кадра P_{17} в основном потоке, кадра P_{18} в зависимом потоке 1 и кадра P_{19} в зависимом потоке 2.

На фиг. 30 приведены схемы, иллюстрирующие пример структуры потоков битов.

Потоки битов, показанные на фиг.30, не включают в себя DD.

В части А на фиг. 30 иллюстрируется пример структуры в случае, когда основной поток, зависимый поток 1 и зависимый поток 2 мультиплексированы в один поток битов.

В части В на фиг. 30 иллюстрируется пример структуры в случае, когда основной поток, зависимый поток 1 и зависимый поток 2 включены, в общей сложности, в три разных потока битов. Участки, соответствующие участкам, показанным в части А на фиг. 30, обозначены теми же номерами ссылочных позиций.

Например, в случае, когда поток битов в части А на фиг.30 подают после детектирования AD №11, модуль 152 декодирования видеоданных последовательно считывает кадр P_{11} в основном потоке, кадр P_{12} в зависимом потоке 1 и кадр P_{13} в зависимом потоке 2 и по отдельности декодирует эти изображения.

Кроме того, в случае, когда подают потоки битов в части В на фиг. 30, для первого AU в основном потоке модуль 152 видеodeкодирования детектирует AD №11 и считывает и декодирует кадр P_{11} . Кроме того, модуль 152 видеodeкодирования детектирует для второго AU, AD №12 и считывает и декодирует кадр P_{14} .

Зависимые потоки 1 и 2 не имеют DD. Поэтому для считывания отдельных изображений существует потребность в анализе синтаксиса в потоках битов в соответствии с процессом, показанным на фиг. 28, и детектировании границ кадров P_{12} , P_{13} , P_{15} , P_{16} , P_{18} и P_{19} .

На фиг. 31 включены схемы, иллюстрирующие пример структуры потоков битов, генерируемых устройством 101 кодирования для множества точек съемки, в соответствии с вариантом осуществления настоящего изобретения.

В части А на фиг. 31 иллюстрируется пример структуры в случае, когда основной поток, зависимый поток 1 и зависимый поток 2 мультиплексированы в один поток

битов.

В части В на фиг. 31 иллюстрируется пример структуры в случае, когда основной поток, зависимый поток 1 и зависимый поток 2 включены, в общей сложности, в три разных потока битов.

В случае, когда подают поток битов части А на фиг. 31 после детектирования AD №11, модуль 152 декодирования видеоданных последовательно считывает кадр P₁₁ в основной поток, кадр P₁₂ в зависимый поток 1 и кадр P₁₃ в зависимый поток 2 и по отдельности декодирует кадры.

В случае, когда потоки битов части В на фиг. 31 подают для первого AU в основном потоке, после детектирования AD №11 модуль 152 декодирования видеоданных декодирует кадр P₁₁. Для второго AU после детектирования AD №12 модуль 152 декодирования видеоданных декодирует кадр P₁₄.

В дополнение к зависимому потоку 1 после детектирования DD №11 модуль 152 декодирования видеоданных декодирует кадр P₁₂. Кроме того, после детектирования DD №12 модуль 152 декодирования видеоданных декодирует кадр P₁₅.

Кроме того, для зависимого потока 2 после детектирования DD №21 модуль 152 декодирования видеоданных декодирует кадр P₁₃. Кроме того, после детектирования DD №22 модуль 152 декодирования видеоданных декодирует кадр P₁₆.

Таким образом, в устройстве 101 декодирования для множества точек съемки выполняют операцию таким образом, чтобы требуемый AU был детектирован и декодирован в соответствии с AD, который представляет собой часть информации, иллюстрирующей границу AU, и DD, который представляет собой часть информации, иллюстрирующей границу между изображениями в зависимом потоке, и так, что отображается стереоскопическое изображение.

Поскольку нет необходимости в анализе заголовка сегмента в случае детектирования начального положения определенного изображения в зависимом потоке, становится осуществимым высокоскоростной случайный доступ.

Последовательность обработки, описанная выше, может быть выполнена с использованием аппаратных средств или программных средств. В случае, когда последовательность считывают, выполняют с помощью программных средств программу, составляющую программное средство, устанавливают с носителя записи программы в компьютер, в котором встроено специализированное аппаратное средство, персональный компьютер общего назначения или тому подобное.

На фиг. 32 показана блок-схема, иллюстрирующая пример конфигурации аппаратных средств компьютера, который выполняет последовательность обработки, описанную выше в соответствии с программой.

ЦПУ (центральное процессорное устройство) 301, ПЗУ (постоянное запоминающее устройство) 302 и ОЗУ (оперативное запоминающее устройство) 303 соединены друг с другом через шину 304.

Интерфейс 305 ввода/вывода дополнительно соединен с шиной 304. Модуль 306 ввода, имеющий клавиатуру, мышь и т.п., и модуль 307 вывода, имеющий дисплей, громкоговорители и т.п., соединены с интерфейсом 305 ввода/вывода. Кроме того, модуль 308 накопителя, имеющий жесткий диск, энергонезависимое запоминающее устройство и т.п., модуль 309 передачи данных, имеющий сетевой интерфейс и т.п., и привод 310, в котором осуществляют привод съемного носителя 311 информации, соединены с шиной 304.

В компьютере, имеющем описанную выше конфигурацию, например, ЦПУ 301 загружает программу, сохраненную в модуле 308 накопителя в ОЗУ 303 через интерфейс 305 ввода/вывода и шину 304, и выполняет эту программу. Таким образом, выполняют последовательность обработок, описанную выше.

5 Программа, выполняемая ЦПУ 301, записана, например, на съемном носителе 311 информации или ее предоставляют через кабельную или беспроводную среду передачи данных, такую как локальная вычислительная сеть, Интернет или цифровая широкополосная передача, и устанавливают в модуле 308 накопителя.

10 Следует отметить, что программа, выполняемая компьютером, может представлять собой программу, в соответствии с которой процессы выполняют во временной последовательности, в порядке, описанном здесь, или может представлять собой программу, в соответствии с которой процессы выполняют параллельно или в требуемые моменты времени, например по вызову.

15 Варианты выполнения настоящего изобретения не должны быть ограничены описанным выше вариантом осуществления, и различные изменения могут быть выполнены без отхода от сущности настоящего изобретения.

Список номеров ссылочных позиций

41 устройство кодирования изображения, снятого с множества точек съемки, 51 буфер изменения порядка следования, 52 модуль видеокодирования, 53 запоминающее устройство кадра, 54 буфер изменения порядка следования, 55 модуль кодирования 20 зависимого потока, 56 запоминающее устройство кадра, 57 модуль мультиплексирования, 101 устройство декодирования изображения, снятого с множества точек съемки, 102 3D дисплей, 111 буфер, 112 модуль декодирования видеоданных, 113 запоминающее устройство кадра, 114 буфер, 115 модуль декодирования зависимого 25 потока, 116 запоминающее устройство кадра, 201 модуль кодирования DD

Формула изобретения

1. Способ декодирования для декодирования потока битов, включающего в себя основной поток, сгенерированный путем кодирования первого сигнала изображения, 30 полученного с первой точки съемки, и зависимый поток, сгенерированный путем кодирования второго сигнала изображения, полученного со второй точки съемки, при этом способ содержит этапы, на которых

35 обнаруживают на основе разграничителя блока доступа, указывающего границу между блоками доступа, границу в потоке битов, образованную путем вставки разграничителя блока доступа в виде блока NAL, включающего в себя параметр nal_unit_type, причем параметр nal_unit_type указывает содержание блока NAL, а блок доступа содержит кадр основного потока и кадр зависимого потока;

40 обнаруживают на основе зависимого разграничителя, указывающего границу между кадрами в зависимом потоке, границу кадра в зависимом потоке, образованную путем вставки зависимого разграничителя в начало кадра зависимого потока, причем зависимый разграничитель представляет собой блок NAL, включающий в себя параметр nal_unit_type, значение которого отличается от значения параметра nal_unit_type, включенного в блок NAL, являющийся разграничителем блока доступа; и

45 декодируют основной поток на основе обнаруженной границы в потоке битов и декодируют зависимый поток на основе обнаруженной границы кадра в зависимом потоке.

2. Устройство декодирования для декодирования потока битов, включающего в себя основной поток, сгенерированный путем кодирования первого сигнала изображения,

полученного с одной точки съемки, и зависимый поток, сгенерированный путем кодирования второго сигнала изображения, полученного со второй точки съемки, при этом устройство декодирования содержит:

5 средство обнаружения на основе разграничителя блока доступа, указывающего границу между блоками доступа, границы в потоке битов, образованной путем вставки разграничителя блока доступа в виде блока NAL, включающего в себя параметр nal_unit_type, причем параметр nal_unit_type указывает содержание блока NAL, а блок доступа содержит кадр основного потока и кадр зависимого потока;

10 средство обнаружения на основе зависимого разграничителя, указывающего границу между кадрами в зависимом потоке, границы кадра в зависимом потоке, образованной путем вставки зависимого разграничителя в начало кадра зависимого потока, причем зависимый разграничитель представляет собой блок NAL, включающий в себя параметр nal_unit_type, значение которого отличается от значения параметра nal_unit_type, включенного в блок NAL, являющийся разграничителем блока доступа; и

15 средство декодирования основного потока на основе обнаруженной границы в потоке битов и декодирования зависимого потока на основе обнаруженной границы кадра в зависимом потоке.

3. Устройство декодирования для декодирования потока битов, включающего в себя основной поток, сгенерированный путем кодирования первого сигнала изображения, полученного с одной точки съемки, и зависимый поток, сгенерированный путем кодирования второго сигнала изображения, полученного со второй точки съемки, при этом устройство декодирования содержит:

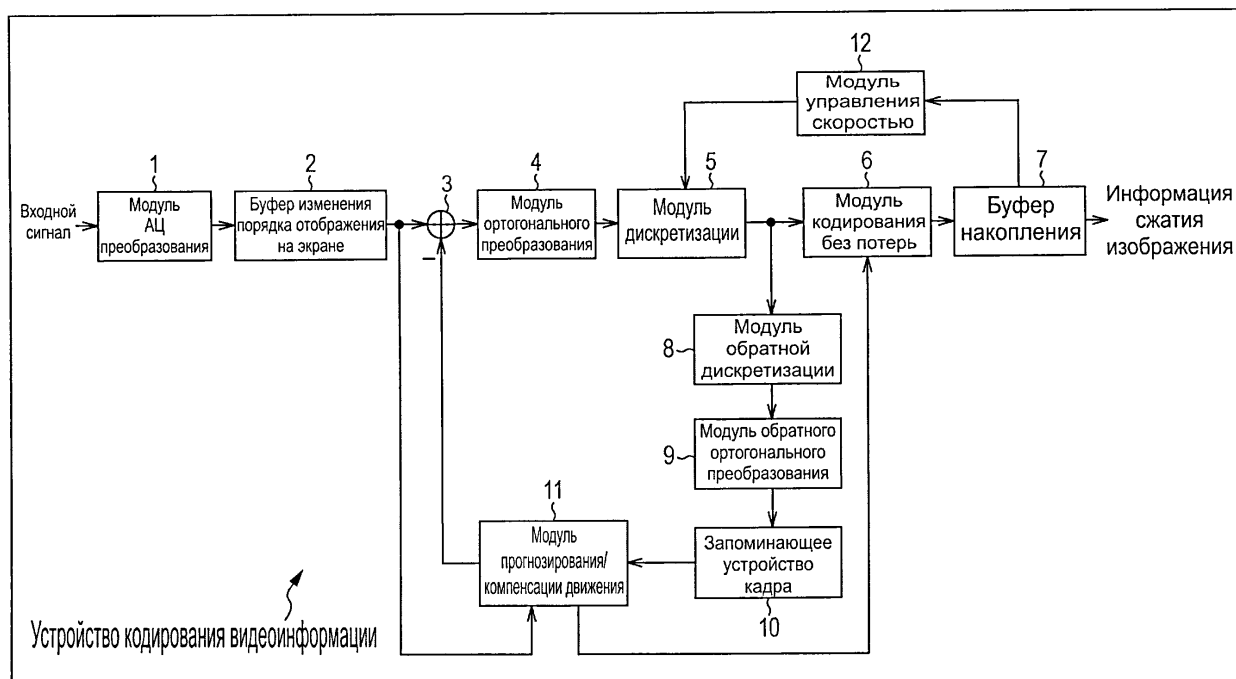
20 блок обработки, выполненный с возможностью обнаружения на основе разграничителя блока доступа, указывающего границу между блоками доступа, границы в потоке битов, образованной путем вставки разграничителя блока доступа в виде блока NAL, включающего в себя параметр nal_unit_type, причем параметр nal_unit_type указывает содержание блока NAL, а блок доступа содержит кадр основного потока и кадр зависимого потока;

30 обнаружения на основе зависимого разграничителя, указывающего границу между кадрами в зависимом потоке, границы кадра в зависимом потоке, образованной путем вставки зависимого разграничителя в начало кадра зависимого потока, причем зависимый разграничитель представляет собой блок NAL, включающий в себя параметр nal_unit_type, значение которого отличается от значения параметра nal_unit_type, включенного в блок NAL, являющийся разграничителем блока доступа; и

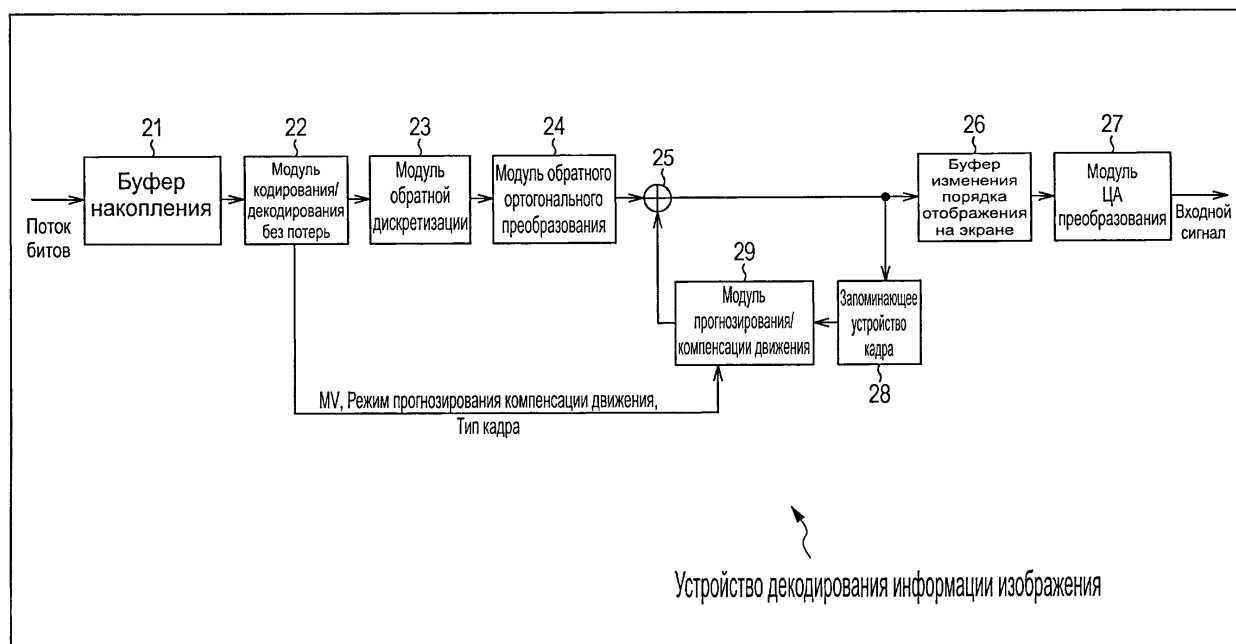
35 декодирования основного потока на основе обнаруженной границы в потоке битов и декодирования зависимого потока на основе обнаруженной границы кадра в зависимом потоке.

40

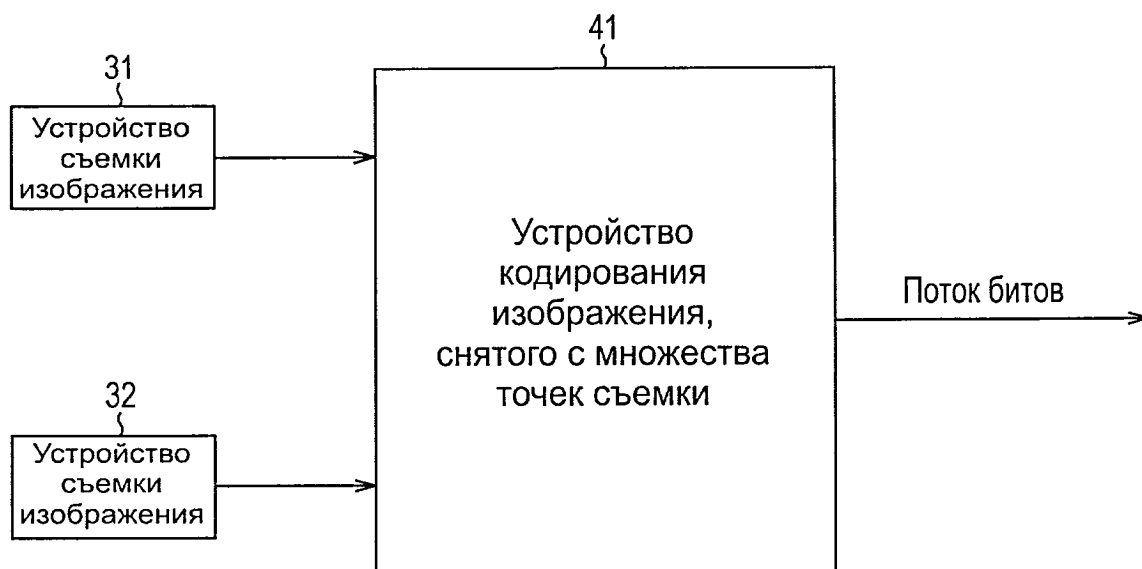
45



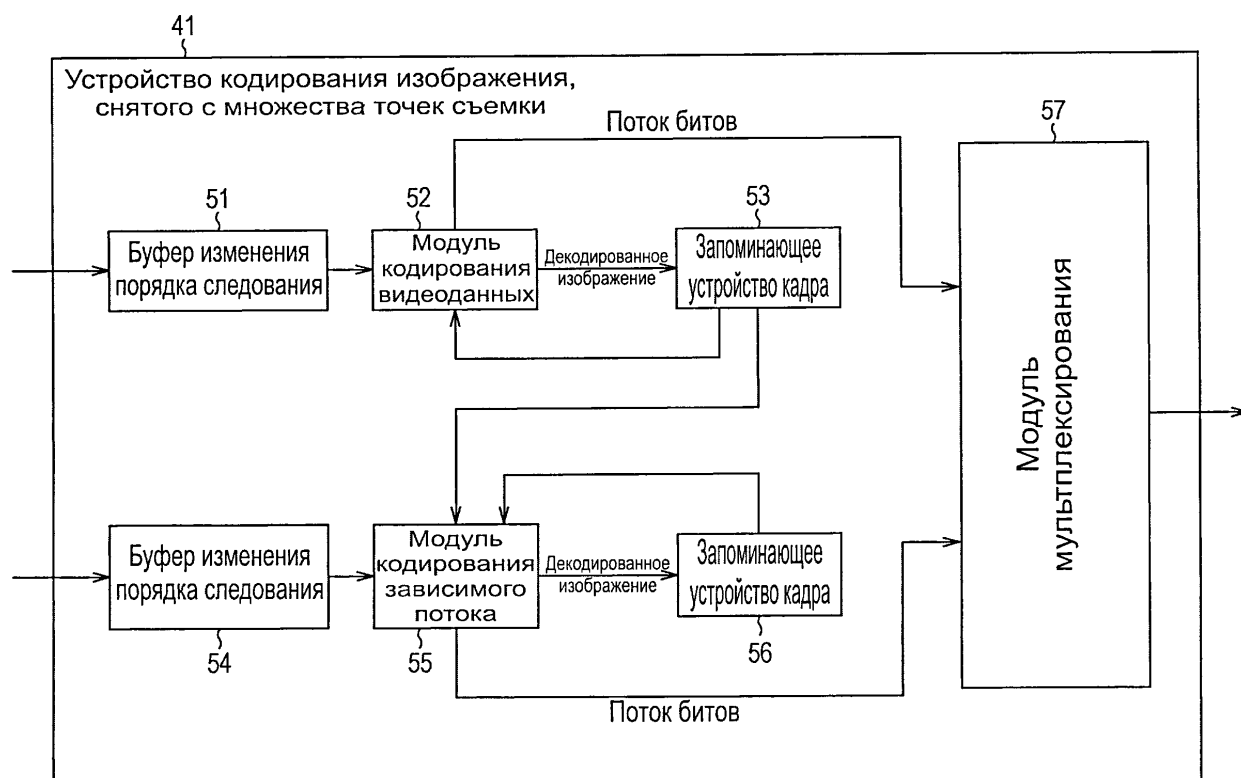
Фиг. 1



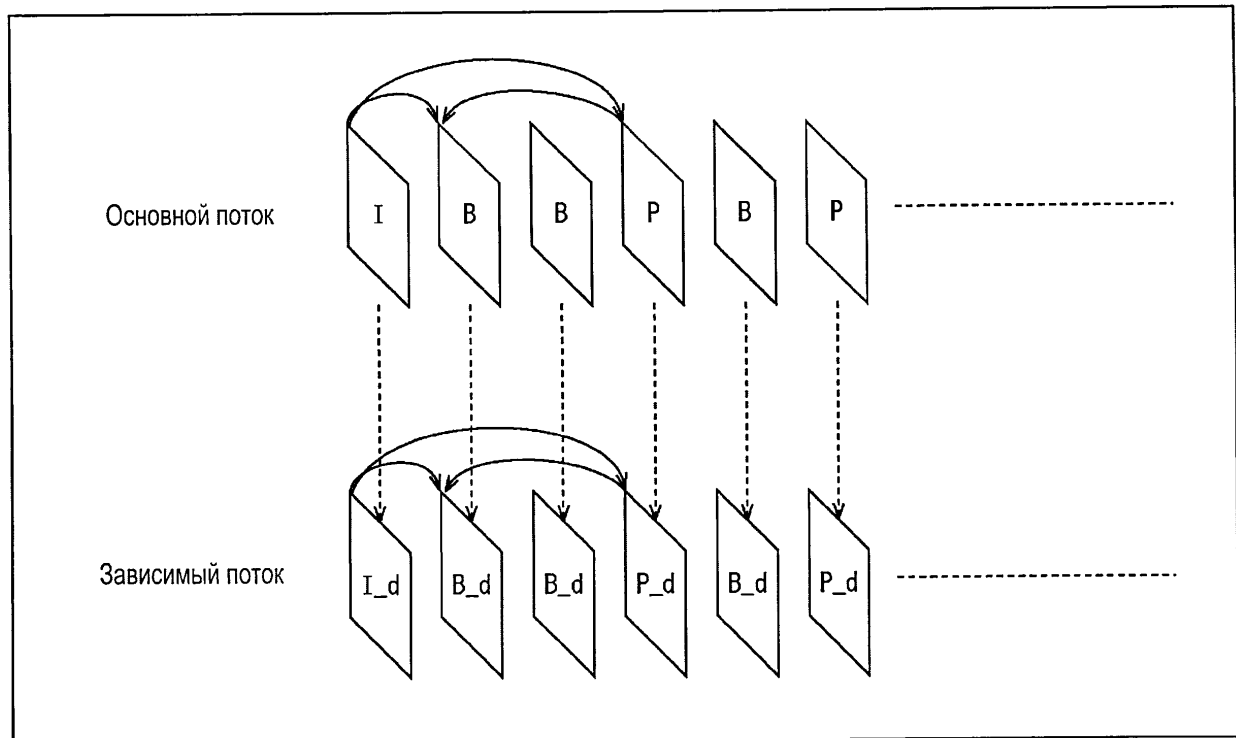
Фиг. 2



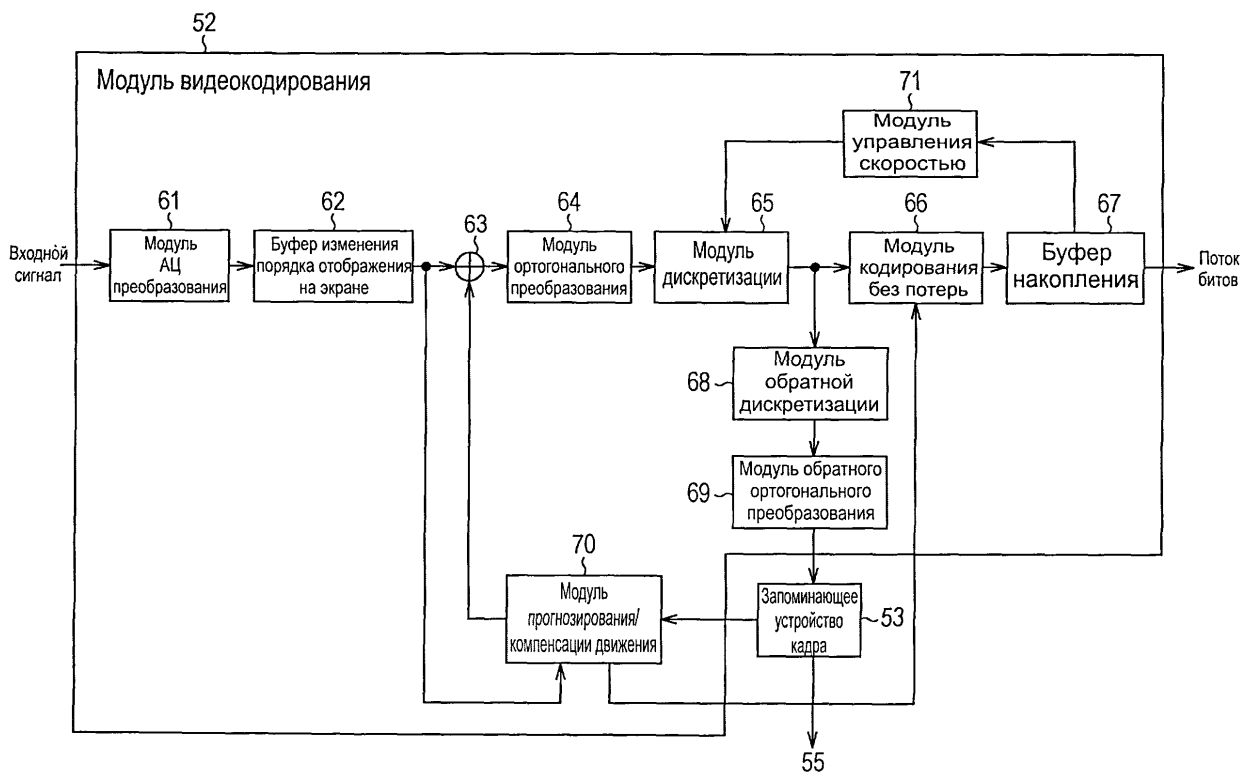
Фиг. 3



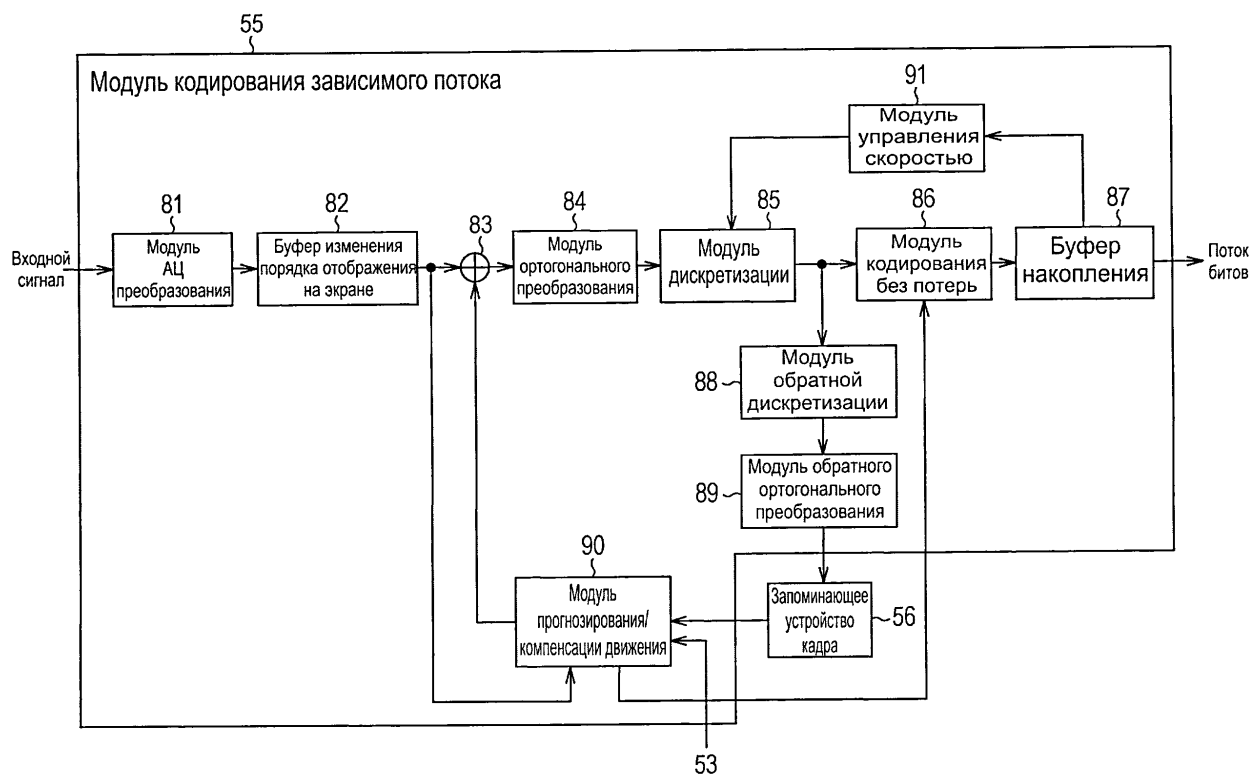
Фиг. 4



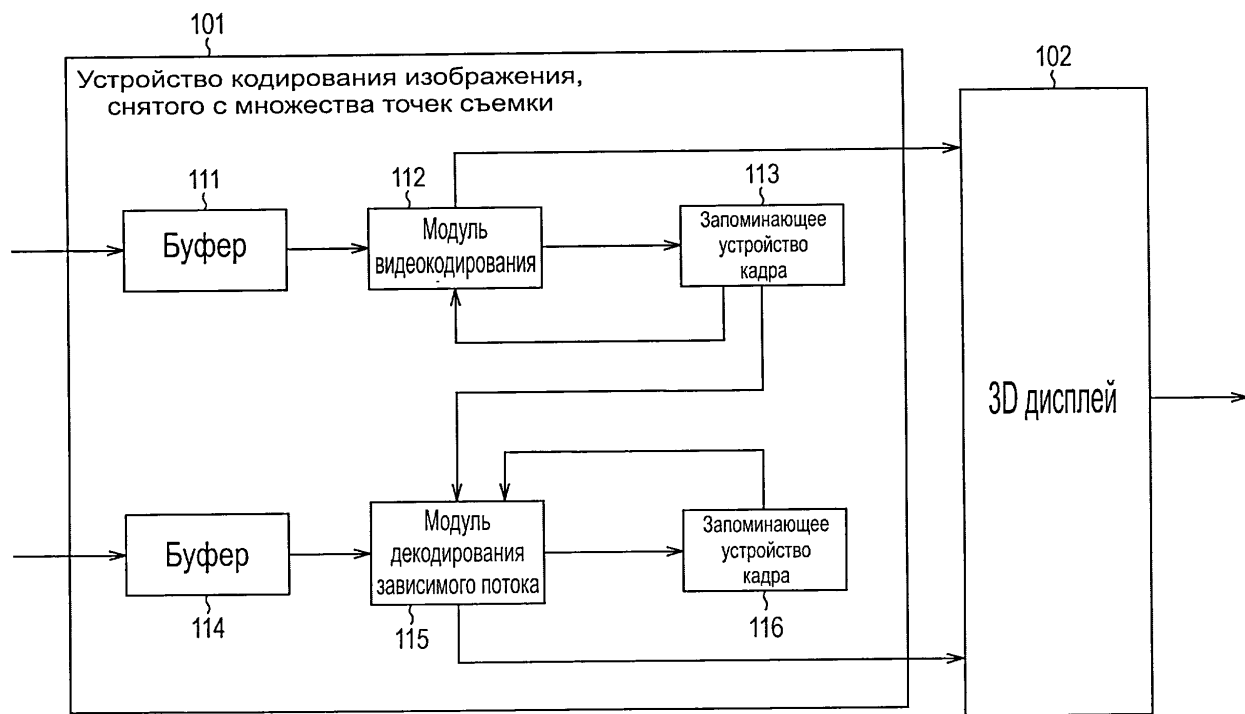
Фиг. 5



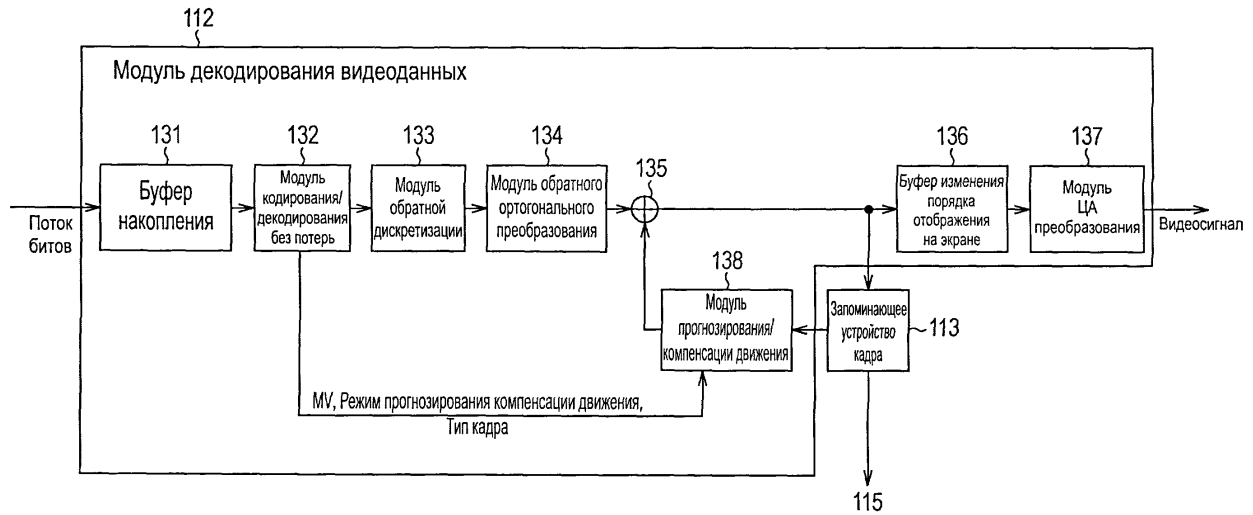
Фиг. 6



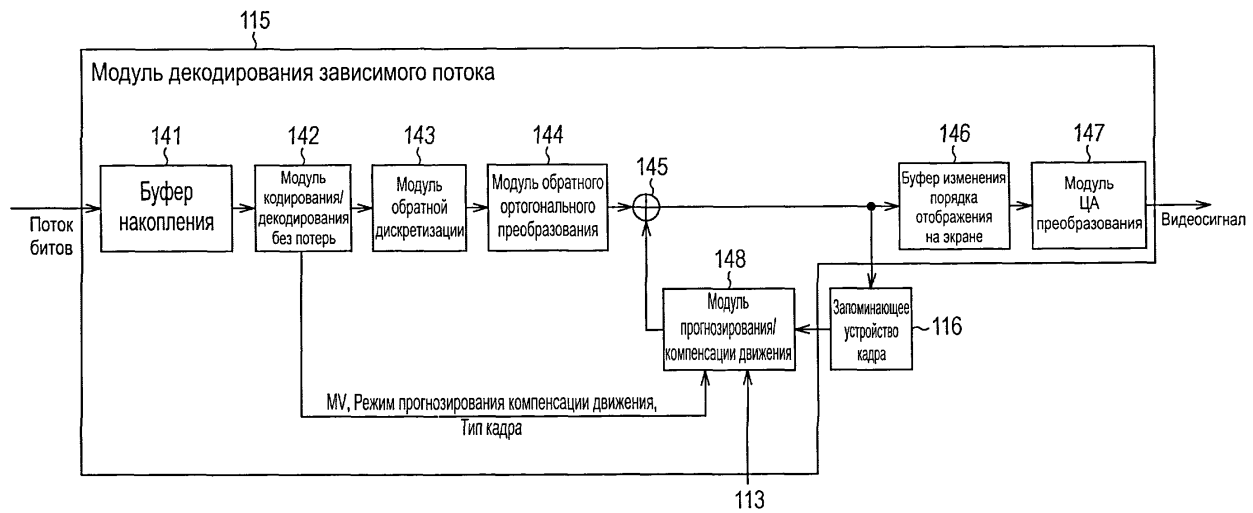
Фиг. 7



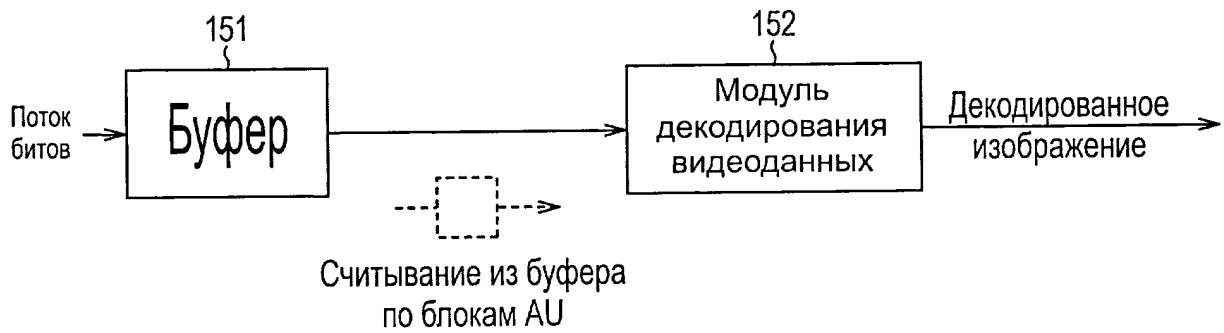
Фиг. 8



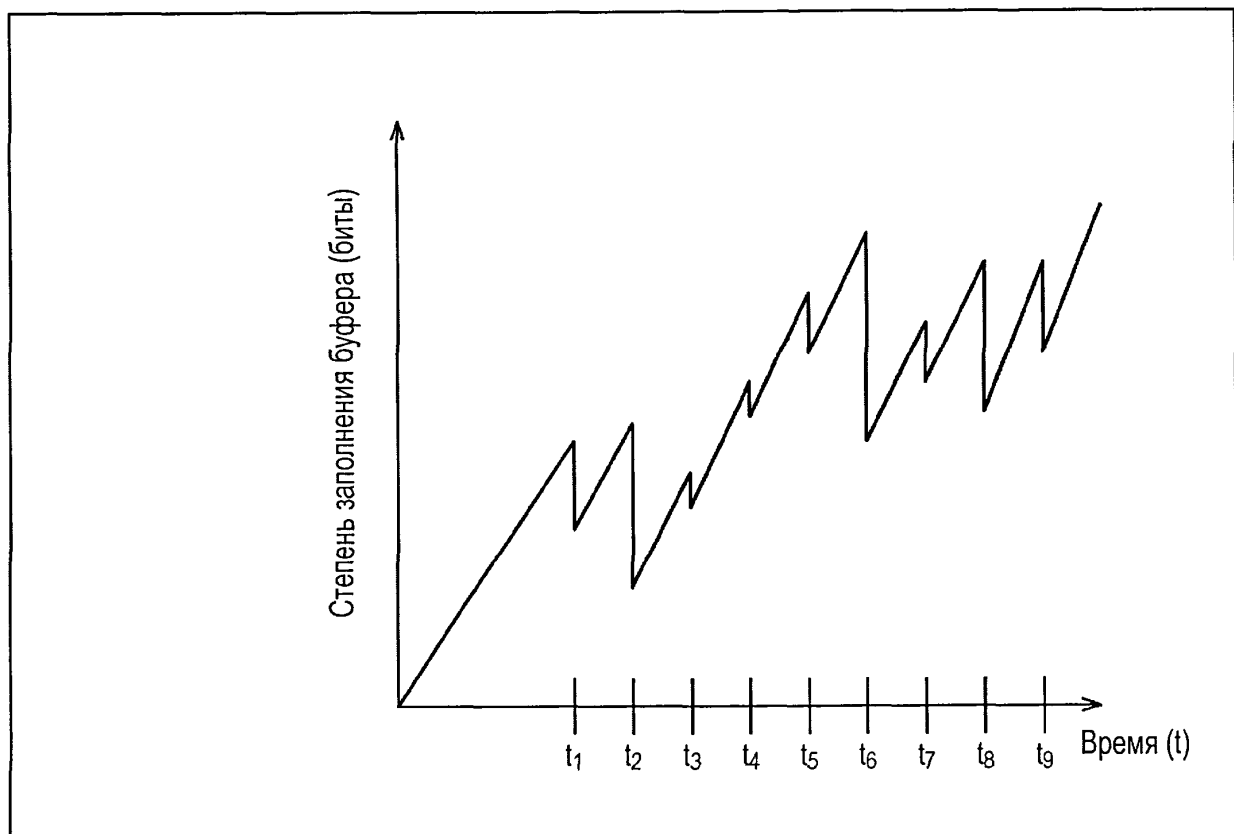
Фиг. 9



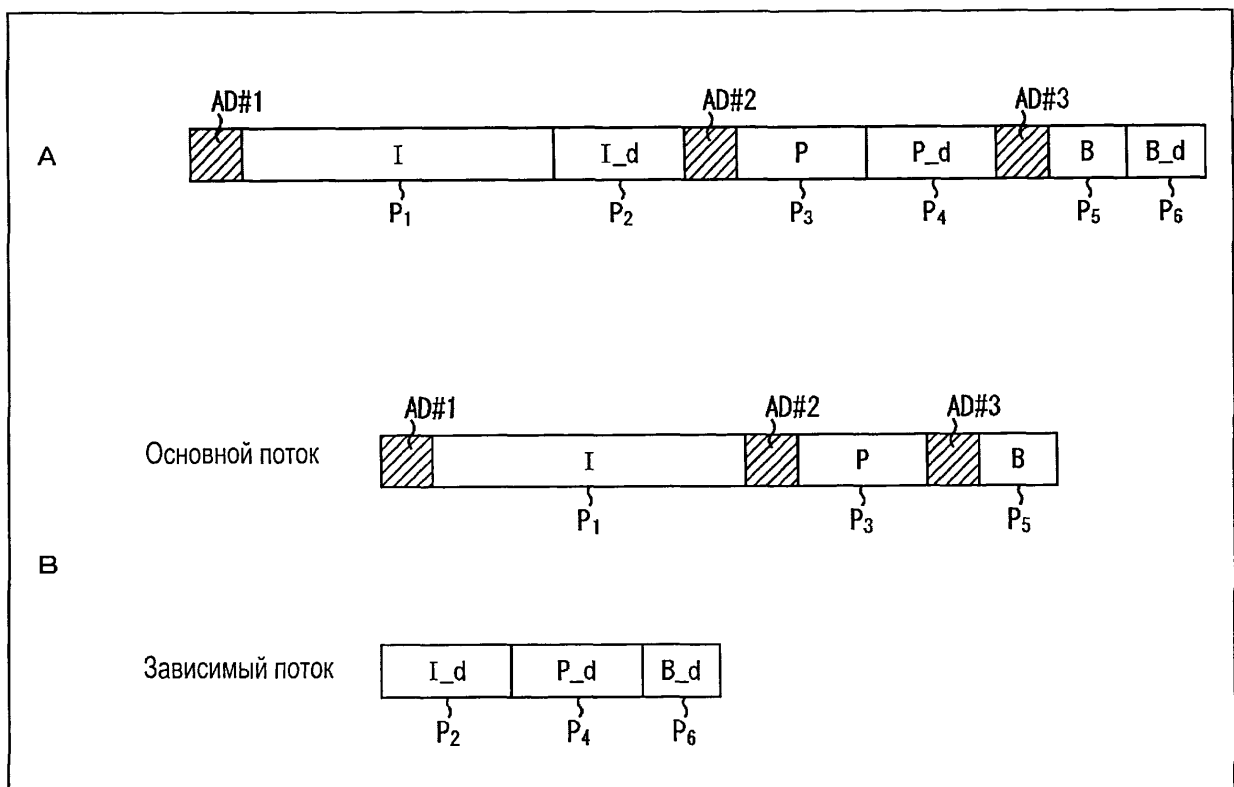
Фиг. 10



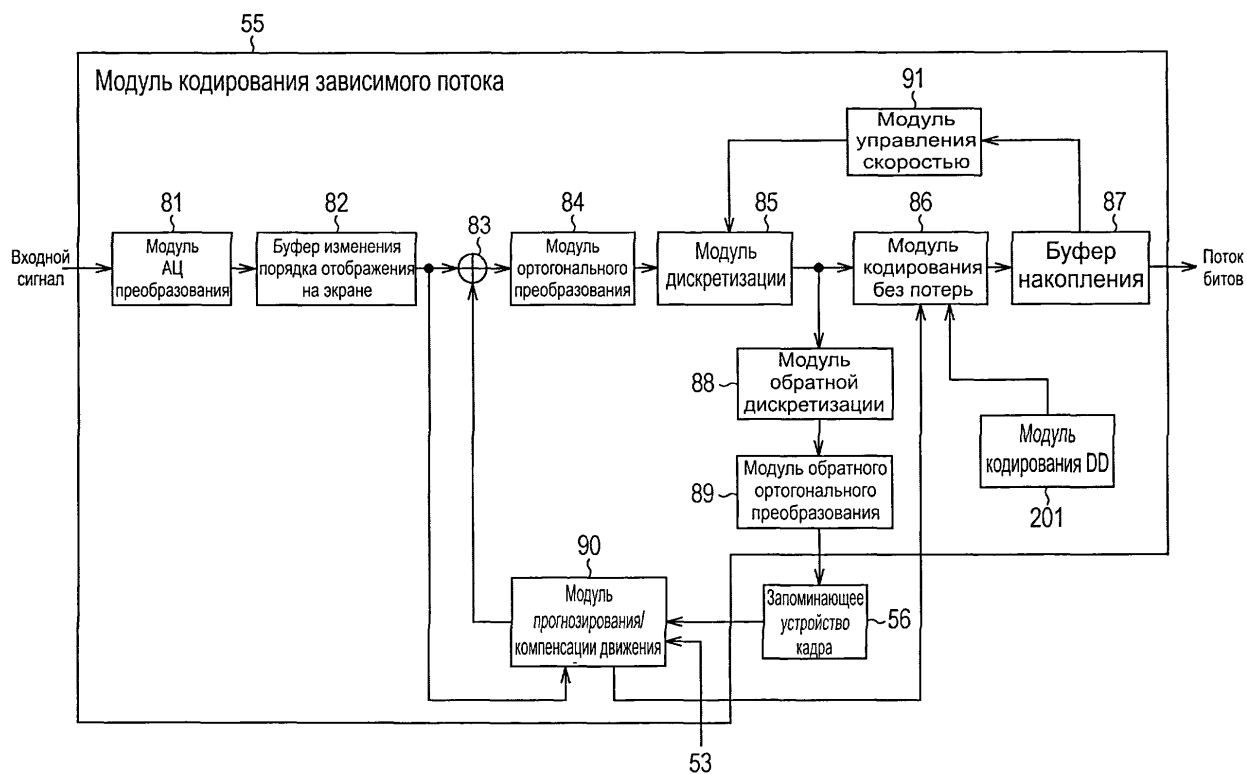
Фиг. 11



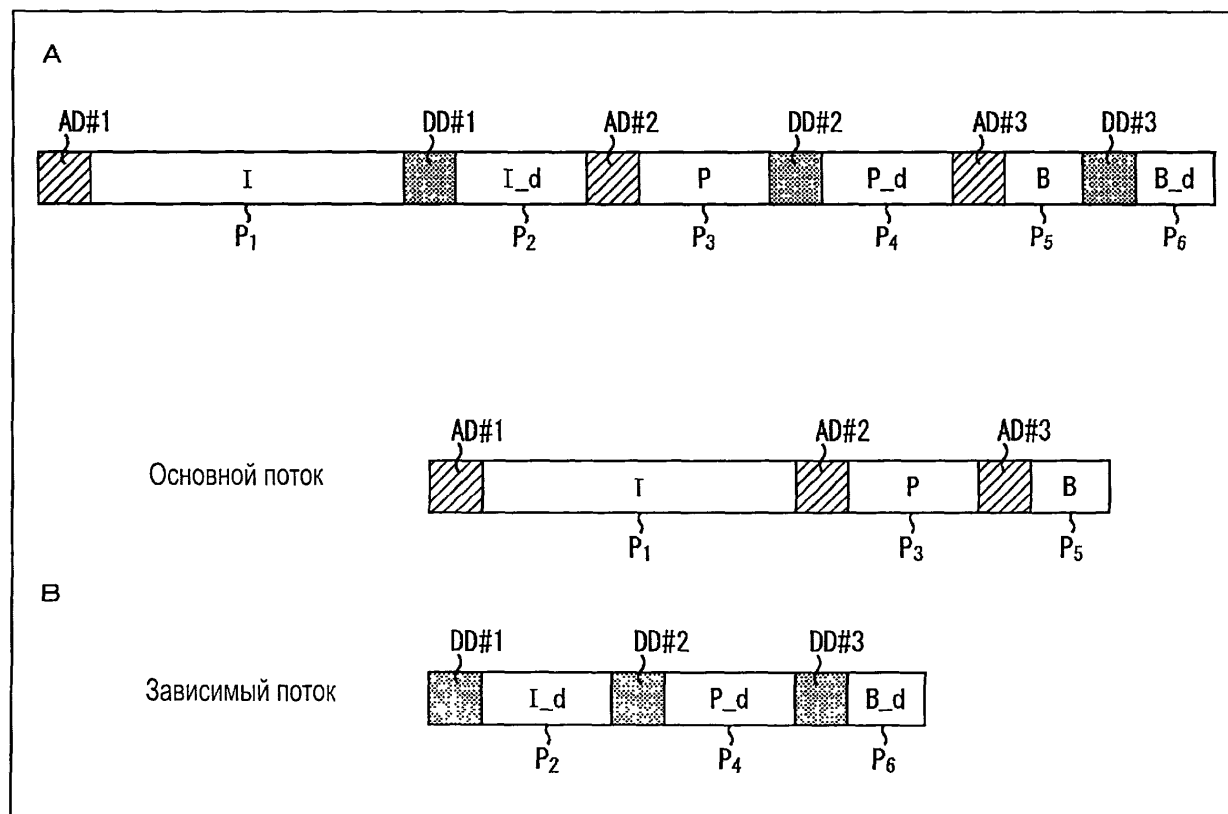
Фиг. 12



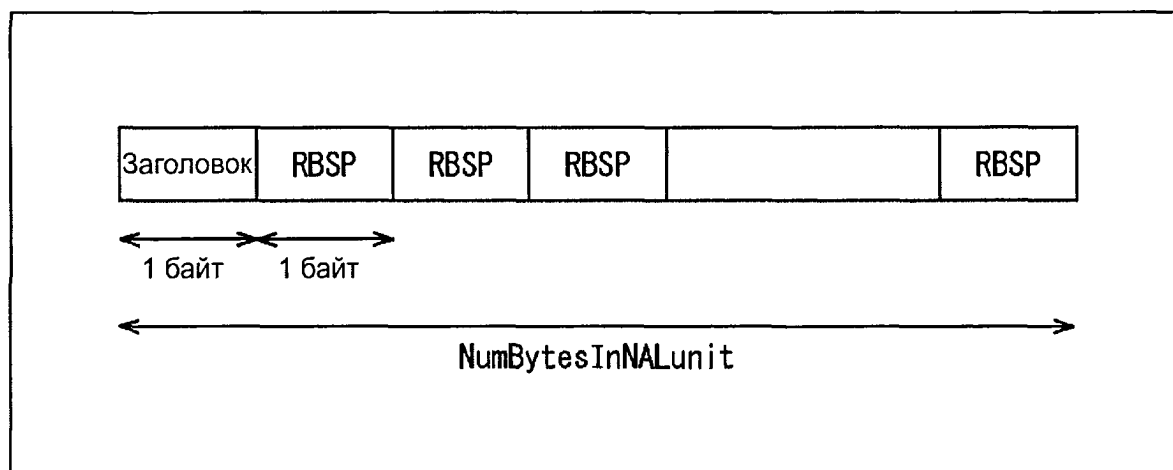
Фиг. 14



Фиг. 15



Фиг. 16



Фиг. 17

nal_unit(NumBytesInNALunit) {	C	Дескриптор
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRbsp = 0		
nalUnitHeaderBytes = 1		
if(nal_unit_type == 14 nal_unit_type == 20) {		
svc_extension_flag	All	u(1)
if(svc_extension_flag)		
nal_unit_header_svc_extension() /* указано в приложении G */	All	
else		
nal_unit_header_mvc_extension() /* указано в приложении H */	All	
nalUnitHeaderBytes += 3		
}		
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {		
if(i+2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte[NumBytesInRbsp++]	All	b(8)
rbsp_byte[NumBytesInRbsp++] /* равно 0x03 */	All	b(8)
i += 2		
emulation_prevention_three_byte	All	f(8)
} else		
rbsp_byte[NumBytesInRbsp++]	All	b(8)
}		
}		

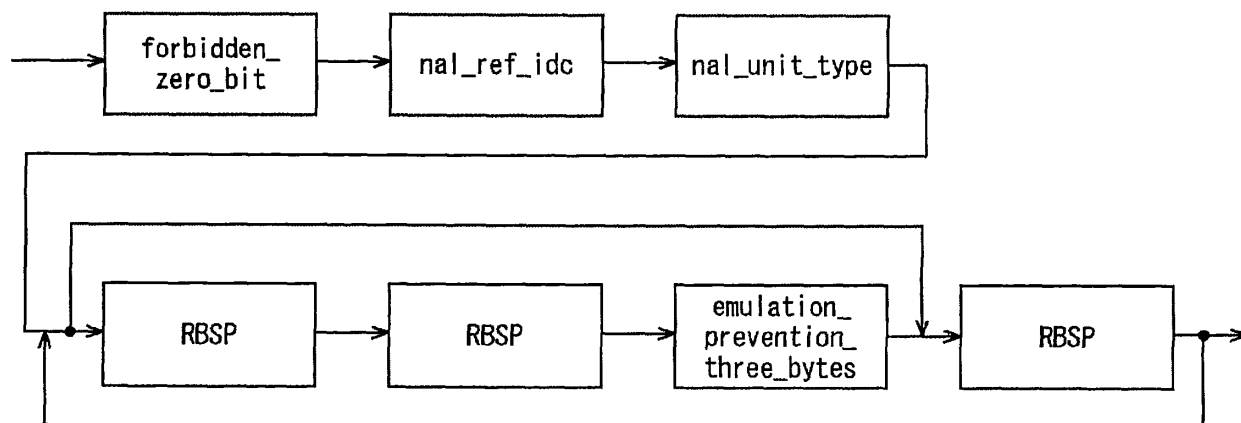
Фиг. 18

nal_ unit_ type	Содержание блока NAL и структура синтаксиса RBSP	c	Приложение A класс типа блока NAL	Приложение G и приложение H класс типа блока NAL
0	Не установлено		He VCL	He VCL
1	Кодированный сегмент не IDR изображения slice_layer_without_partitioning_rbsp()	2, 3, 4	VCL	VCL
2	Кодированные данные сегмента раздела A slice_data_partition_a_layer_rbsp()	2	VCL	He применимо
3	Кодированные данные сегмента раздела B slice_data_partition_b_layer_rbsp()	3	VCL	He применимо
4	Кодированные данные сегмента раздела C slice_data_partition_c_layer_rbsp()	4	VCL	He применимо
5	Кодированный сегмент IDR изображения slice_layer_without_partitioning_rbsp()	2, 3	VCL	VCL
6	Вспомогательная информация расширения (SEI) sei_rbsp()	5	He VCL	He VCL
7	Набор параметров последовательности seq_parameter_set_rbsp()	0	He VCL	He VCL
8	Набор параметров изображения pic_parameter_set_rbsp()	1	He VCL	He VCL
9	Разграничитель блока доступа access_unit_delimiter_rbsp()	6	He VCL	He VCL
10	Конец последовательности end_of_seq_rbsp()	7	He VCL	He VCL
11	Конец потока end_of_stream_rbsp()	8	He VCL	He VCL
12	Данные заполнителя filler_data_rbsp()	9	He VCL	He VCL

Фиг. 19

nal_unit_type	Содержание блока NAL и структура синтаксиса RBSP	c	Приложение A класс типа блока NAL	Приложение G и приложение H класс типа блока NAL
13	Расширение набора параметров последовательности seq_parameter_set_extentsion_rbsp()	10	He VCL	He VCL
14	Блок NAL префикса prefix_nal_unit_rbsp()	2	He VCL	Зависит от суффикса
15	Набор параметров последовательности поднабора subset_seq_parameter_set_rbsp()	0	He VCL	He VCL
16. . 17	Зарезервировано		He VCL	He VCL
18	Зависимый разграничитель dependent_delimiter_rbsp()		He VCL	He VCL
19	Кодированный сегмент вспомогательного кодированного изображения без разделений slice_layer_without_partitioning_rbsp()	2, 3, 4	He VCL	He VCL
20	Расширение кодированного сегмента slice_layer_extention_rbsp()	2, 3, 4	He VCL	VCL
21. . 23	Зарезервировано		He VCL	He VCL
24. . 31	Не установлен		He VCL	He VCL

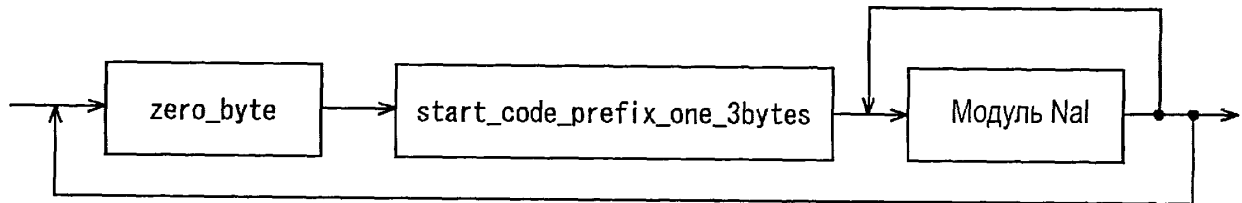
Фиг. 20



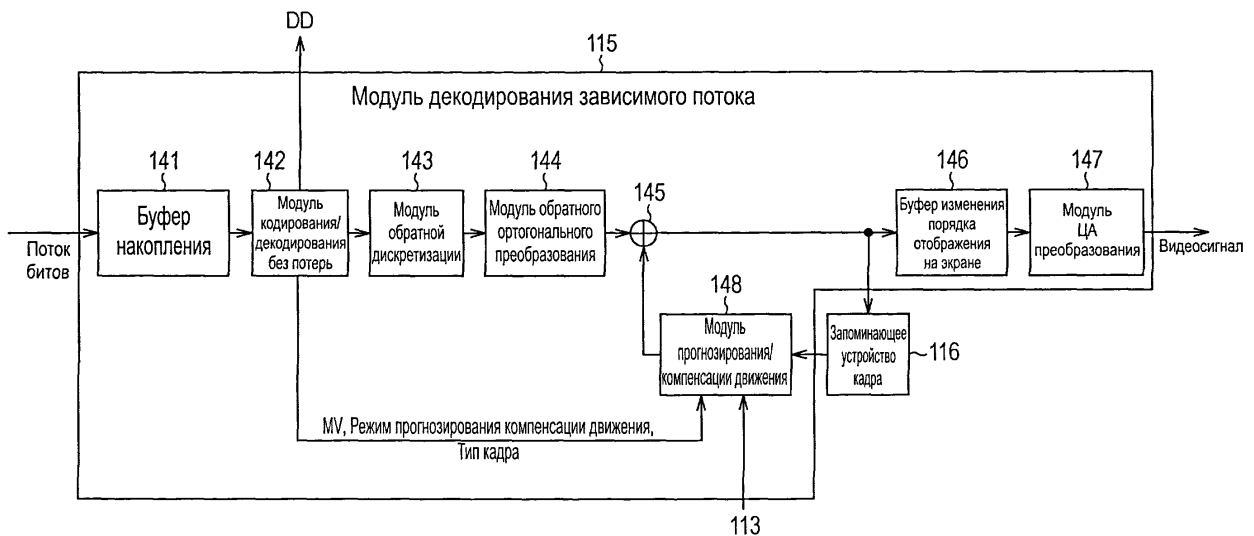
Фиг. 21

byte_stream_nal_unit(NumBytesInNALunit){	C	Дескриптор
while(next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
leading_zero_8bits /* равно 0x00 */		f (8)
if(next_bits(24) != 0x000001)		
zero_byte /* равно 0x00 */		f (8)
start_code_prefix_one_3bytes /* равно 0x000001 */		f (24)
nal_unit(NumBytesInNALunit)		
while(more_data_in_byte_stream() && next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
trailing_zero_8bits /* равно 0x00 */		f (8)
}		

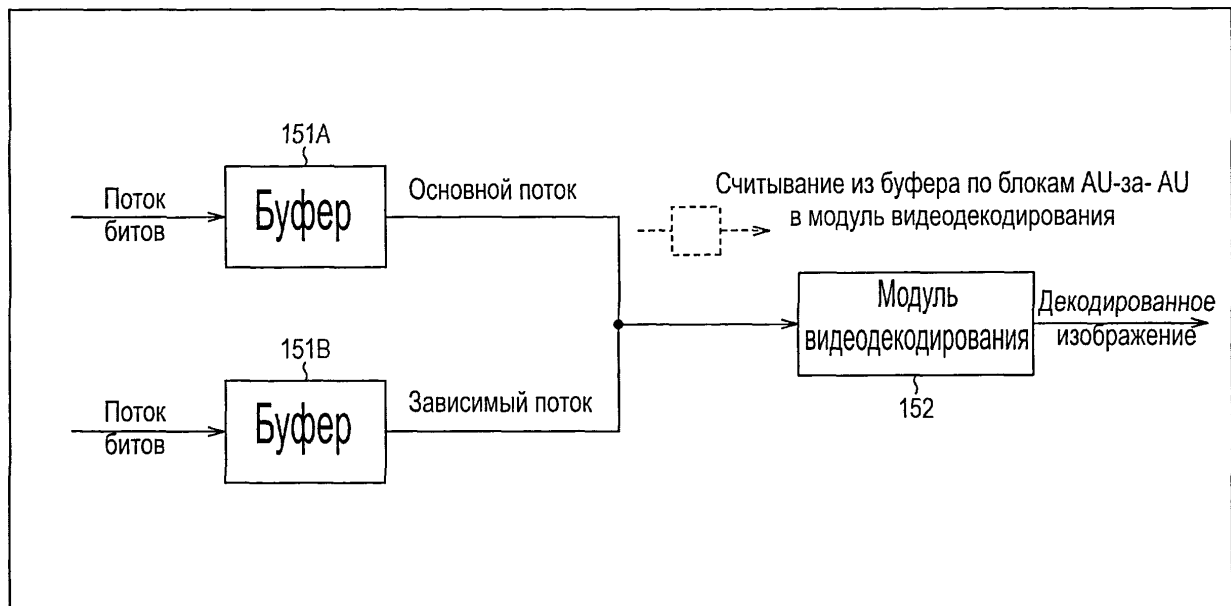
Фиг. 22



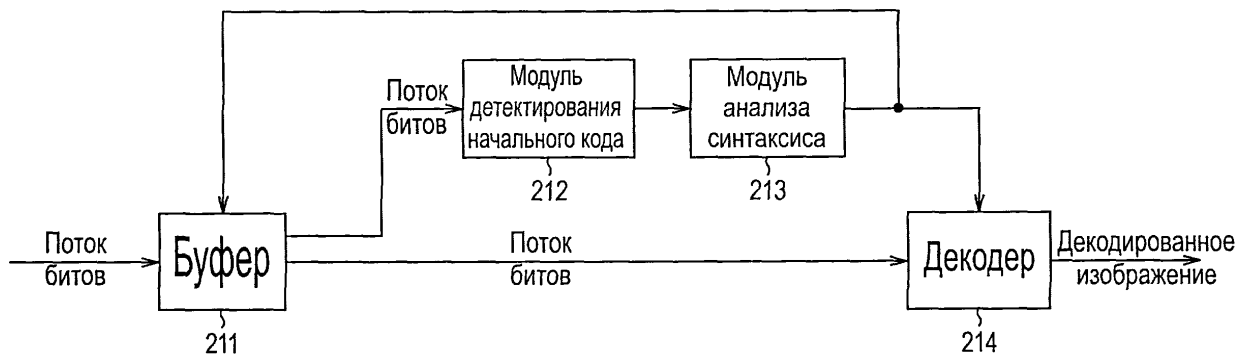
Фиг. 23



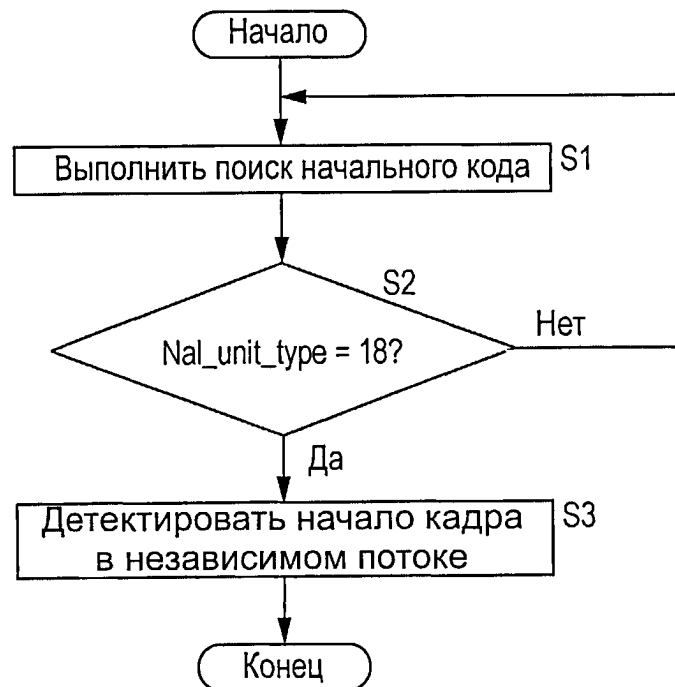
Фиг. 24



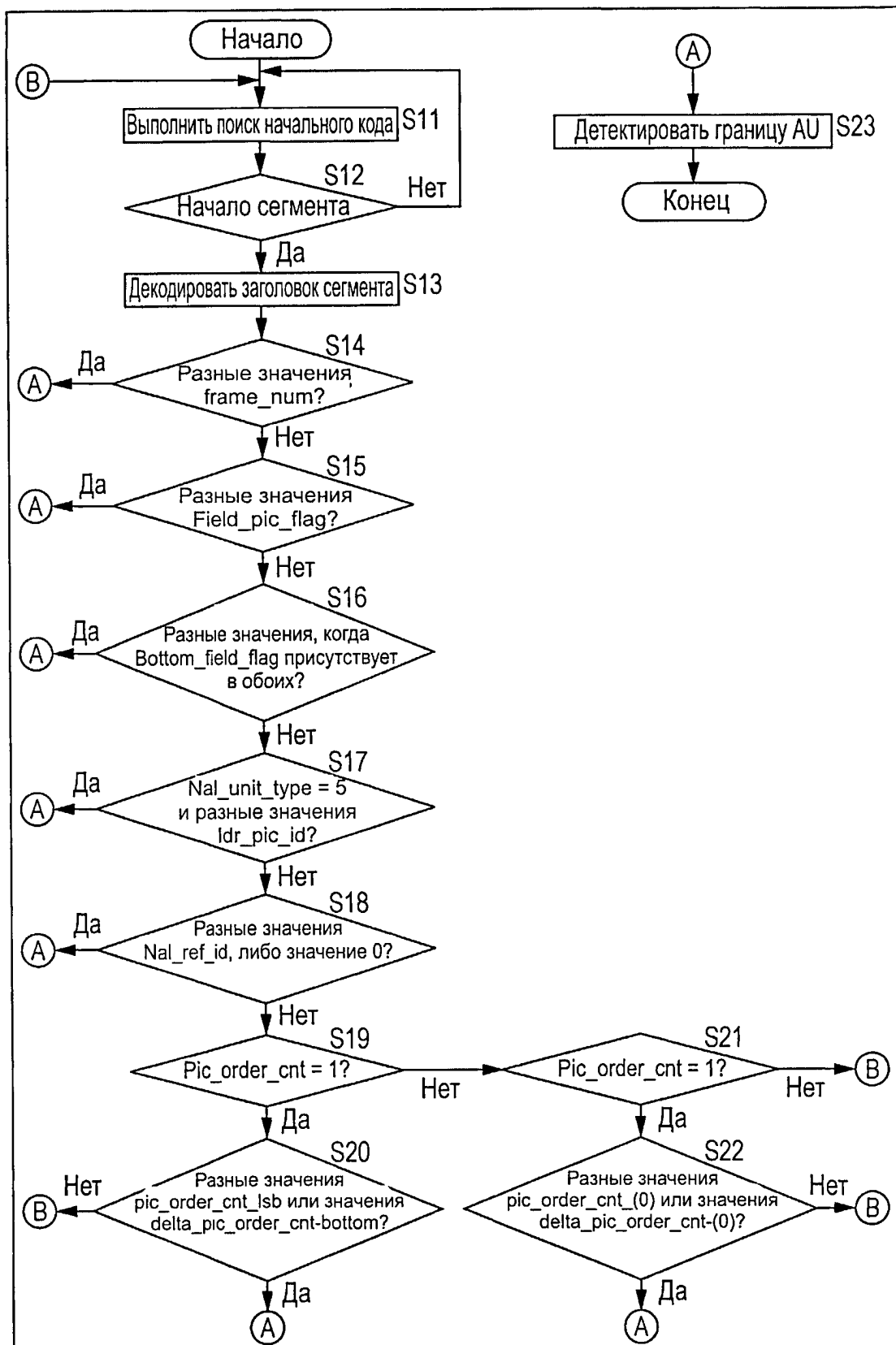
Фиг. 25



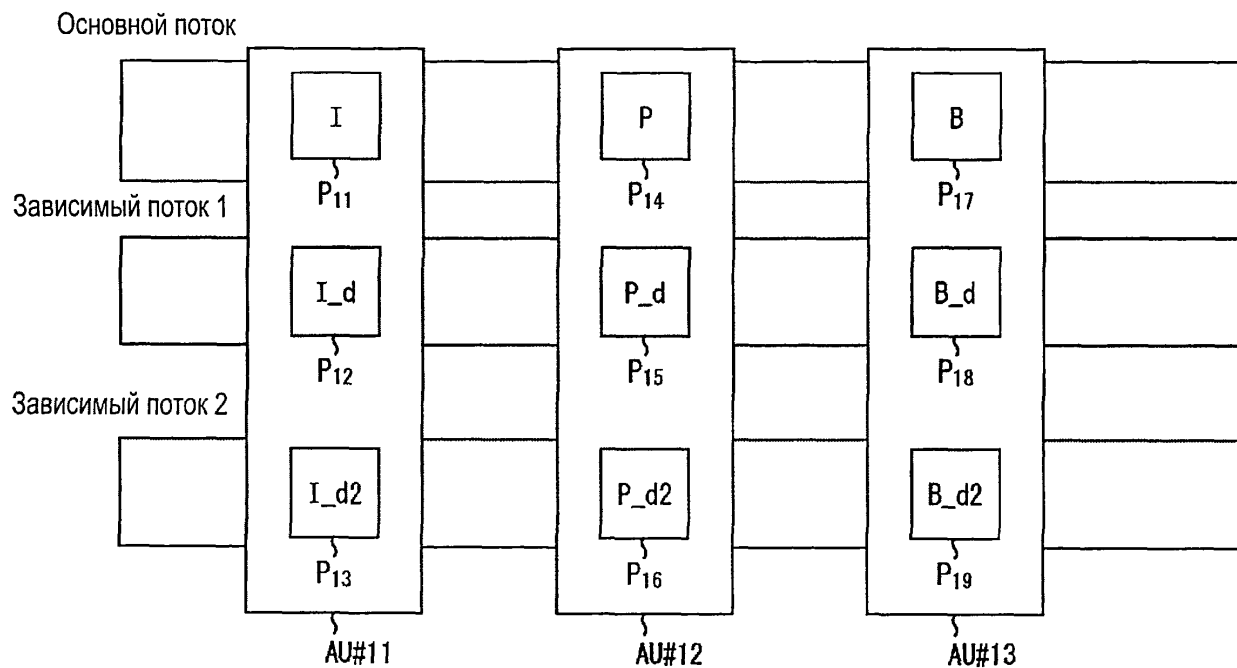
Фиг. 26



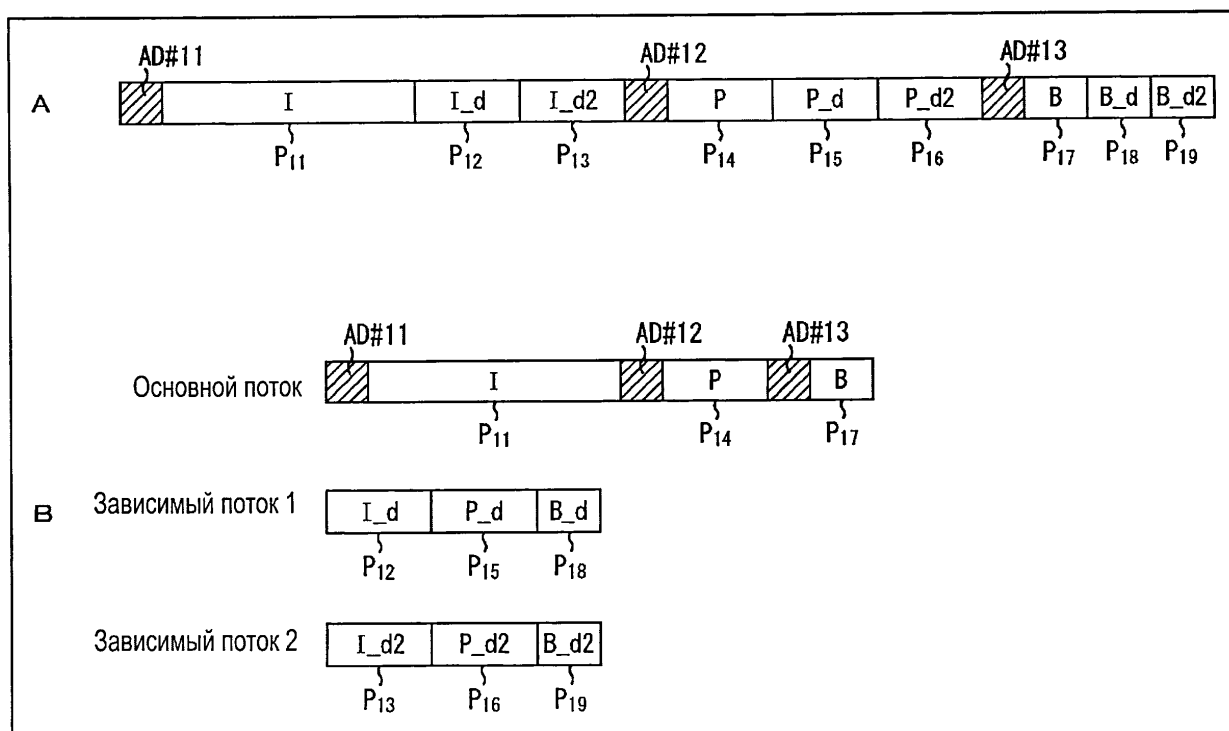
Фиг. 27



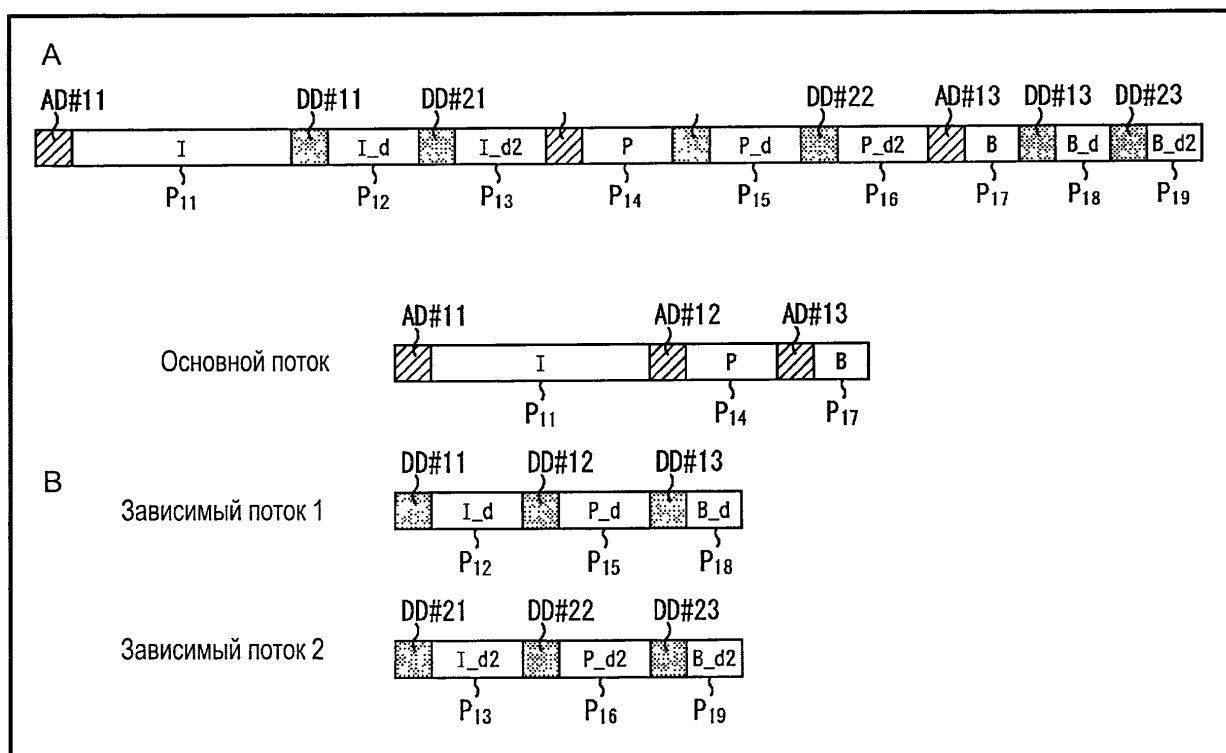
Фиг. 28



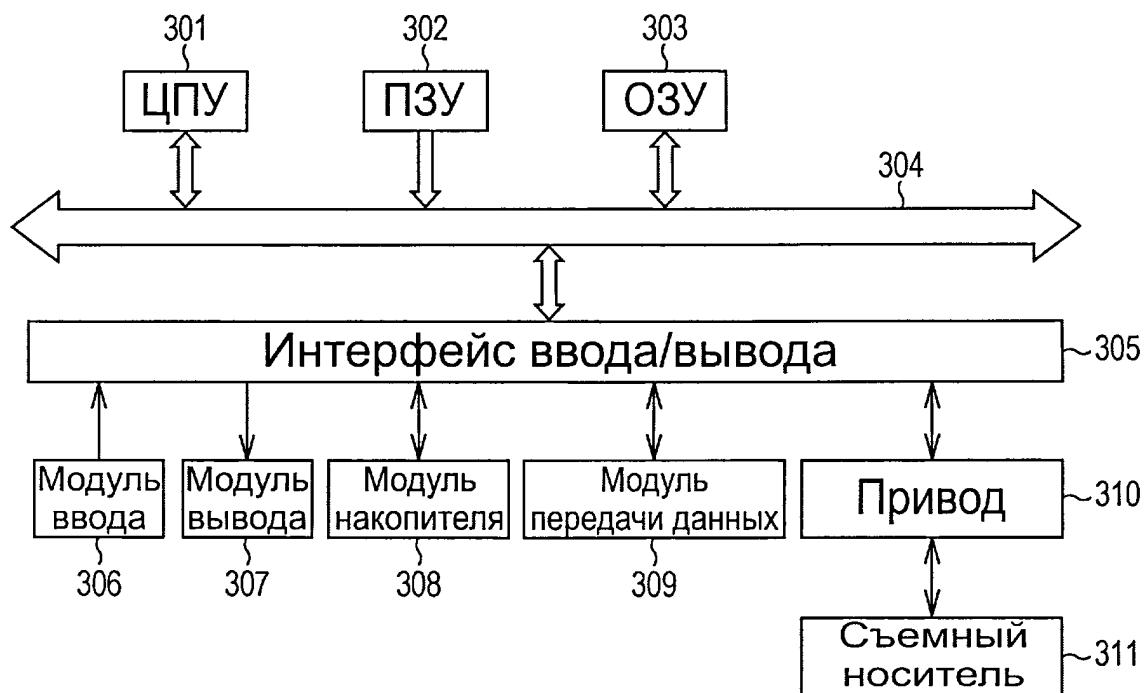
Фиг. 29



Фиг. 30



Фиг. 31



Фиг. 32