



**ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

**(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ**

(21)(22) Заявка: 2013107605/08, 29.06.2011

(24) Дата начала отсчета срока действия патента:  
29.06.2011

Приоритет(ы):

(30) Конвенционный приоритет:  
21.07.2010 US 61/366,215

(43) Дата публикации заявки: 20.10.2014 Бюл. № 29

(45) Опубликовано: 10.04.2016 Бюл. № 10

(56) Список документов, цитированных в отчете о поиске: Т. WIEGAND "Version 3 of H.264/AVC", опублик. 17.01.2005 на 323 страницах [найдено 16.07.2015], найдено в Интернет по адресу URL:[http://wftp3.itu.int/av-arch/jvt-site/2004\\_07\\_Redmond/JVT-L012d2wcmRelTod1.zip](http://wftp3.itu.int/av-arch/jvt-site/2004_07_Redmond/JVT-L012d2wcmRelTod1.zip). М. HANNUKSELA et al. "Improved Coding of Slice Headers", JVT-C079, опублик. 17.01.2005 на 7 страницах [найдено 17.07.2015], найдено в Интернет по (см. прод.)

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 21.02.2013

(86) Заявка РСТ:  
SE 2011/050876 (29.06.2011)

(87) Публикация заявки РСТ:  
WO 2012/011860 (26.01.2012)

Адрес для переписки:

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

(72) Автор(ы):

**ШЕБЕРГ Рикард (SE),  
ПРИДДЛ Клинтон (AU)**

(73) Патентообладатель(и):

**ТЕЛЕФОНАКТИЕБОЛАГЕТ Л М  
ЭРИКССОН (ПАБЛ) (SE)**

**(54) КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ ИЗОБРАЖЕНИЙ**

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в улучшении идентификации начала секций для первых секций в изображении. Способ кодирования изображения, содержащего  $N \geq 2$  секций, содержит генерирование, для каждой секции из упомянутых  $N$  секций, кодированного представления секции на основе значений

пикселей у пикселей в упомянутой секции; установку флага секции, ассоциированного с первой секцией из  $N$  секций, в первое определенное значение; установку, для каждой секции из  $N-1$  остальных секций из  $N$  секций, флага секции, ассоциированного с упомянутой секцией, во второе определенное значение; генерирование, для каждой секции из  $N-1$  остальных секций,

адреса секции, обеспечивающего возможность идентификации позиции первой единицы кодирования секции в пределах изображения; генерирование кодированного представления изображения для изображения, содержащего упомянутые N кодированных представлений секций, N-1 адресов секций и N флагов секций; и определение длины N-1 адресов секций в терминах числа бит, как одно из  $\text{ceil}(\log_2(P))$  и  $\text{ceil}(\log_2(P-1))$ , причем  $\text{ceil}()$  обозначает функцию наименьшего целого числа, определенную как  $\text{ceil}(x) = \lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем x, а P обозначает общее число возможных адресов секций в упомянутом изображении. 5 н. и 14 з.п. ф-лы, 13 ил.



ФИГ. 1

(56) (продолжение):

адресу URL:[http://wftp3.itu.int/av-arch/jvt-site/2002\\_05\\_Fairfax/JVT-C079.doc](http://wftp3.itu.int/av-arch/jvt-site/2002_05_Fairfax/JVT-C079.doc). US 2010/0021060 A1, 28.01.2010. US 6907068 B2, 14.06.2005. RU 2316909 C2, 10.02.2008.

RU 2580021 C2

RU 2580021 C2



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.  
*H04N 19/174* (2014.01)  
*H04N 19/46* (2014.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2013107605/08, 29.06.2011**

(24) Effective date for property rights:  
**29.06.2011**

Priority:

(30) Convention priority:  
**21.07.2010 US 61/366,215**

(43) Application published: **20.10.2014 Bull. № 29**

(45) Date of publication: **10.04.2016 Bull. № 10**

(85) Commencement of national phase: **21.02.2013**

(86) PCT application:  
**SE 2011/050876 (29.06.2011)**

(87) PCT publication:  
**WO 2012/011860 (26.01.2012)**

Mail address:

**129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,  
OOO "Juridicheskaja firma Gorodisskij i Partnery"**

(72) Inventor(s):

**SHEBERG Rikard (SE),  
PRIDDL Klinton (AU)**

(73) Proprietor(s):

**TELEFONAKTIEBOLAGET L MERIKSSON  
(PABL) (SE)**

(54) **IMAGE ENCODING AND DECODING**

(57) Abstract:

FIELD: physics, computer engineering.

SUBSTANCE: invention relates to computer engineering. A method of encoding an image comprising  $N \geq 2$  sections includes generating, for each section from the said  $N$  sections, an encoded section representation based on pixel values at pixels in the said section; setting a section flag associated with the first section from the  $N$  sections to the first determined value; setting, for each section from the  $N-1$  remaining sections from the  $N$  sections, a section flag associated with the said section to the second determined value; generating, for each section from the  $N-1$  remaining sections, a section address which enables to identify the position of the first section coding unit within the image; generating an encoded image representation for the image comprising the said  $N$  encoded section representation,  $N-1$  section addresses and  $N$  section flags; and determining the length of  $N-1$  section addresses in terms of the number of bits as one of  $\text{ceil}(\log_2(P))$  and  $\text{ceil}(\log_2(P-1))$ , where  $\text{ceil}(\ )$  denotes a

ceiling function, which is defined as  $\text{ceil}(x) = \lceil x \rceil$ , and a ceiling of not less than  $x$  is output, and  $P$  denotes the total number of possible section addresses in the said image.

EFFECT: improved identification of the beginning of sections for the first sections in an image.

19 cl, 13 dwg



Область техники, к которой относится изобретение

Настоящие варианты осуществления изобретения относятся в целом к обработке изображений или видеок кадров, и в частности к кодированию и декодированию таких изображений или видеок кадра.

5 Уровень техники

Стандарт H.264, также называемый MPEG-4 (Motion Picture Expert Group - Экспертная группа по движущимся изображениям) AVC (Advanced Video Coding - Усовершенствованное видеок кодирование), является стандартом кодирования изображений текущего уровня техники. Это гибридный кодек, который основывается на устранении избыточности между кадрами, обозначаемом так называемым интер-кодированием (inter coding - межкадровое кодирование), и в пределах кадров, обозначаемом так называемым интра-кодированием (intra coding - внутрикадровое кодирование). Выходными данными процесса кодирования являются данные VCL (Video Coding Layer - Уровень видеок кодирования), которые далее инкапсулируются в единицы NAL (Network Abstraction Layer - Уровень абстракции сети) перед передачей или сохранением.

Согласно H.264/MPEG-4 AVC, изображение видеопотока состоит из макроблоков с фиксированным размером 16×16 пикселей, и кодирование изображения происходит от макроблока к макроблоку. Каждое изображение видеопотока делится на одну или более секций. Секция является независимо декодируемой частью изображения. Это означает то, что если одна секция изображения потеряна, другие секции изображения все же будут декодируемыми. В дополнение, секции могут использоваться для параллельного кодирования и декодирования, поскольку они являются независимыми от других секций в изображении. В H.264/MPEG-4 AVC должна иметь место граница секций между границами двух смежных, в соответствии с порядком кодирования, макроблоков.

HEVC (High Efficiency Video Coding - Высокоэффективное видеок кодирование) является преемником H.264/MPEG-4 AVC. HEVC направлено на то, чтобы существенно улучшить эффективность кодирования по сравнению с H.264/MPEG-4 AVC, т.е. уменьшить требования к скорости передачи битов, при сохранении качества изображения. HEVC ориентировано на следующее поколение дисплеев HDTV (High Definition Television - Телевидение высокой четкости) и систем захвата контента, которые отличаются скоростями прогрессивно сканируемых кадров и разрешениями отображения от QVGA (Quarter Video Graphics Array - Четвертной видеографический массив) (320×240) до 1080p и Ультра HDTV (Ultra HDTV) (7680×4320), а также улучшенным качеством изображений.

HEVC обеспечивает возможность использования так называемых наибольших единиц кодирования (largest coding units, LCU) которые являются блоками пикселей, имеющими размер, который больше, чем размер макроблоков в H.264/MPEG-4 AVC для обеспечения улучшенной эффективности кодирования. Для того, чтобы обрабатывать как большие однородные области, так и небольшие детализованные области в одном и том же изображении, было предложено иерархическое кодирование для HEVC. Наибольшие единицы кодирования (LCU) в изображении сканируются в заранее определенном порядке, и каждая такая наибольшая единица кодирования (LCU) может быть разделена на меньшие единицы кодирования (coding units, CU), которые в свою очередь могут иерархически разделяться, как квадратное дерево, вплоть до наименьшей единицы кодирования (smallest coding unit, SCU). Изображение может, таким образом, кодироваться как смесь единиц кодирования с различными размерами, ранжированных от LCU до SCU.

В соответствии с H.264/MPEG-4 AVC, изображение видеопотока может быть разделено на одну или более секций в HEVC. Граница секций в HEVC выравнивается с границей двух смежных, в соответствии с заранее определенным порядком, LCU.

5 Оба стандарта H.264/MPEG-4 AVC и HEVC требуют определения и использования адресов для того, чтобы идентифицировать первый макроблок или единицу кодирования секции и, таким образом, начало секции в изображении или видеокадре. Такие адреса, хотя они необходимы в декодере, добавляют служебные данные к данным кодированного изображения. Кроме того, с введением иерархического разделения LCU в HEVC возникают новые проблемы в связи с кодированием и декодированием 10 изображений или видеокадров. Поэтому существует необходимость эффективного кодирования и декодирования, которое может управлять адресами начала секций эффективным и гибким способом.

#### Раскрытие изобретения

15 Общей целью изобретения является обеспечение эффективного управления секциями в изображениях и видеокадрах.

Более конкретной целью изобретения является сигнализация начальных позиций секций эффективным способом.

Эти и другие цели выполняются вариантами осуществления, раскрытыми здесь.

20 Аспект вариантов осуществления определяет способ кодирования изображения, содержащего множество секций. Кодированное представление секции генерируется для каждой секции в изображении на основе значений пикселей для пикселей в упомянутой секции. Соответствующий флаг секции назначается и устанавливается для каждой из секций. Первая секция в изображении имеет флаг секции, установленный в первое определенное значение, тогда как остальные секции имеют соответствующий флаг 25 секции, установленный во второе определенное значение. Адреса секций, обеспечивающие возможность идентификации позиции первой единицы кодирования секции и соответственно начала секции в пределах изображения, генерируются для остальных секций, исключая первую секцию в изображении. Эти адреса секций 30 включаются, вместе с кодированными представлениями секций и флагами секций, в кодированное представление изображения.

Другой аспект вариантов осуществления относится к устройству для кодирования изображения, содержащего множество секций. Генератор представлений упомянутого устройства генерирует соответствующее кодированное представление секции для каждой секции в изображении. Устройство содержит установщик флагов, конфигурированный 35 для установки флага секции, ассоциированного с первой секцией в изображении, в первое значение, тогда как флаг секции (флаги секций) оставшейся секции (остальных секций) устанавливается (устанавливаются) во второе определенное значение. Генератор адреса генерирует соответствующий адрес секции для каждой секции оставшейся секции (остальных секций), чтобы обеспечить возможность идентификации соответствующей 40 позиции первой единицы кодирования и начала секции в пределах изображения. Менеджер представлений генерирует кодированное представление изображения, содержащее кодированные представления секций, адреса секций и флаги секций.

Дополнительный аспект вариантов осуществления определяет способ декодирования кодированного представления изображения, содержащего множество секций. Флаг 45 секции, ассоциированный с секцией в изображении, извлекается из кодированного представления изображения. Значение этого флага секции используется для того, чтобы определить, нужно ли извлекать адрес секции из кодированного представления изображения для того, чтобы идентифицировать начальную позицию секции в пределах

изображения и часть секции, которая принадлежит секции. Если флаг секции имеет первое определенное значение, то настоящая секция является первой секцией в изображении. Значения пикселей для пикселей в упомянутой секции, генерируемые посредством декодирования кодированного представления секции, ассоциированного с секцией и извлеченного из кодированного представления изображения, затем назначаются первой части изображения, начиная с определенного начала секции в пределах изображения, такого как верхний левый угол. Если флаг секции, вместо этого, имеет второе определенное значение, то адрес секции используется, чтобы идентифицировать начало секции в пределах изображения, и значения пикселей, генерируемые посредством декодирования кодированного представления секции, для пикселей, назначаются части изображения, начиная с идентифицированного начала секции.

Еще один аспект вариантов осуществления относится к устройству для декодирования кодированного представления изображения, содержащего множество секций. Устройство извлечения представлений упомянутого устройства извлекает флаг секции, ассоциированный с секцией, которая должна декодироваться из кодированного представления изображения. Устройство извлечения адресов извлекает адрес секции из кодированного представления изображения, если флаг секции имеет второе определенное значение. Если флаг секции вместо этого имеет первое определенное значение, то генератор представлений генерирует декодированное представление значений пикселей для пикселей в упомянутой секции на основе кодированного представления секции, ассоциированного с секцией и извлеченного из представления кодированного изображения. Эти значения пикселей затем назначаются устройством назначения значений первой части изображения, начиная с определенной начальной позиции, которая идентифицируется без необходимости какого-либо сигнализируемого адреса секции, например, верхний левый угол. Если флаг секции вместо этого имеет второе определенное значение, то извлеченный адрес секции используется устройством назначения значений для того, чтобы идентифицировать начало части изображения, которое принадлежит текущей секции, и затем назначить значения пикселей из генератора представлений этой идентифицированной части.

Варианты осуществления обеспечивают эффективное управление секциями в пределах изображений или видеок кадров с точки зрения обеспечения эффективного способа сигнализации и идентификации позиций начала секций в пределах изображения или видеок кадра. Флаги секций вариантов осуществления обеспечивают существенно улучшенную идентификацию начала секций для первых секций в изображении, но без какой-либо необходимости в сигнализации или вычислении адреса секции в декодере.

#### Краткое описание чертежей

Настоящее изобретение, вместе с дополнительными его преимуществами и целями, может быть наиболее полно понято на основе следующего описания, рассматриваемого вместе с сопроводительными чертежами, на которых:

Фиг. 1 является блок-схемой, иллюстрирующей способ кодирования изображения в соответствии с вариантом осуществления;

Фиг. 2 иллюстрирует вариант осуществления изображения, разделенного на множество секций и содержащего множество наибольших единиц кодирования (largest coding units, LCU);

Фиг. 3А и 3В иллюстрируют варианты осуществления совмещения начала секции с границей между единицами кодирования в пределах наибольшей единицы кодирования (LCU);

Фиг. 4 схематично иллюстрирует вариант осуществления порядка кодирования и декодирования для обработки единиц кодирования;

Фиг. 5 является блок-схемой, иллюстрирующей дополнительные этапы способа кодирования на Фиг. 1 в соответствии с вариантом осуществления;

5 Фиг. 6 схематично иллюстрирует вариант осуществления кодированного представления изображения;

Фиг. 7 является блок-схемой иллюстрирующей вариант осуществления декодирования кодированного представления изображения;

10 Фиг. 8 является блок-схемой, иллюстрирующей дополнительные этапы способа на Фиг. 7 в соответствии с вариантом осуществления;

Фиг. 9 является блок-схемой, иллюстрирующей дополнительные этапы способа на Фиг. 7 в соответствии с вариантом осуществления;

Фиг. 10 является схематической блок-схемой устройства для кодирования изображения в соответствии с вариантом осуществления;

15 Фиг. 11 является схематической блок-схемой устройства для декодирования кодированного представления изображения в соответствии с вариантом осуществления; и

Фиг. 12 является схематической блок-схемой медиа-терминала в соответствии с вариантом осуществления.

20 Осуществление изобретения

На всех чертежах, одни и те же ссылочные номера используются для указания одинаковых или соответствующих элементов.

Варианты осуществления в целом относятся к управлению секциями в пределах изображений или видеокадров. Более подробно, варианты осуществления обеспечивают 25 гибкий и битово-эффективный способ сигнализации адресов начала секций. Варианты осуществления применимы к любому кодированию и декодированию изображений или видеокадров, в которых изображение или видеокадр, например видеопоток, может содержать множество, т.е. по меньшей мере две секции, и где начало секций необходимо сигнализировать декодеру. Вариант осуществления может поэтому применяться к 30 известному кодированию изображений или видео, такому как H.264/MPEG-4 AVC, но в частности, является весьма применимым к кодированию и декодированию изображений, которые используют иерархическое разделение блоков пикселей на меньшие блоки пикселей. Варианты осуществления поэтому хорошо подходят для использования в связи с высокоэффективным видеокодированием (High Efficiency Video 35 Coding, HEVC), но не ограничиваются этим.

Характеристикой вариантов осуществления является дифференциация между первой секцией в изображении или видеокадре и остальными секциями. В предшествующем уровне техники, адрес секции определяется во время кодирования для каждой секции в изображении, и включается в кодированное представление изображения для 40 использования декодером, чтобы идентифицировать начало секции в пределах области изображения или видеокадра. В зависимости от общего размера изображения, эти адреса секций могут быть весьма длинными, и таким образом, добавлять существенное количество служебной информации в кодированные данные изображения. Например, H.264/MPEG-4 AVC изображение или видеокадр может состоять из 1280×960 пикселей. 45 Если начало секции выравнивается с границами макроблока, и начало секции выражается как X и Y координаты относительные фиксированного начала отсчета, т.е. обычно верхний левый угол изображения, то для адреса секции необходимо будет  $\log_2(\text{ceil}(1280/16))+\log_2(\text{ceil}(960/16))=7+6=13$  бит в этом простом примере. Функция Ceil ( ) обозначает

функцию наименьшего целого числа, определяемую как  $\lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем  $x$ . Это может быть не очень большим количеством данных, но с учетом современных кадровой и битовой скоростей для видео декодирования и рендеринга, несколько сотен секций обычно используются для доступа каждую секунду, поэтому общий суммарный размер данных адресов секций, которые необходимо генерировать в кодере и направлять к декодеру является весьма существенным для видеопотока. В дополнение, адреса секций нужно обнаружить, извлечь и обработать в декодере перед тем, как фактические данные пикселей секции могут быть декодированы и назначены правильной части изображения.

Упомянутые выше проблемы с обработкой адресов секций могут стать еще более заметными, когда используется иерархическое кодирование и декодирование, такое как HEVC, если начало секций может выравниваться с единицами кодирования более низкого уровня, чем наибольшие единицы кодирования. Фиг. 2 схематично иллюстрирует эту концепцию.

Изображение 1, например, видеокادر в видеопотоке, может быть разделено на множество наибольших кодовых единиц (largest coding units, LCU) 10, также называемыми наибольшими блоками дерева кодирования (largest coding tree blocks, LCTB) или наибольшими блоками кодирования (largest coding blocks, LCB) в известном уровне техники. LCU 10 является наибольшим возможным блоком пикселей, который может обрабатываться в течение кодирования и декодирования, и может обрабатываться, например, в соответствии с хорошо известными режимами интер- или интра-кодирования/декодирования. LCU 10 может в свою очередь быть иерархически разделена на множество меньших, в терминах числа пикселей, блоков пикселей, обычно называемых единицами кодирования (coding units, CU) 20A, 20B, блоками дерева кодирования (coding tree blocks, CTB) или блоками кодирования (coding blocks, CB). Эти единицы кодирования (CU) 20A, 20B могут, в свою очередь, делиться далее на еще более мелкие блоки 30A, 30B пикселей в иерархическом порядке, вплоть до минимально возможного блока пикселей, называемого наименьшей единицей кодирования (smallest coding unit, SCU), наименьшим блоком дерева кодирования (smallest coding tree block, SCTB) или наименьшим блоком кодирования (smallest coding block, SCB).

В явной противоположности предшествующему уровню техники, который ограничивает расположение начала секции 2, 3 в изображении 1, которое должно выравниваться с границей между двумя смежными LCU 10A, 10B, некоторые варианты осуществления изобретения обеспечивают намного более гибкое расположение начала секций, потенциально обеспечивая возможность выравнивания начала секции 4, 5 с границей между любыми смежными CU 10A, 10B, 20A, 20B, 30A, 30B в изображении, начиная с уровня SCU и вплоть до уровня LCU. Поэтому в данном случае возможно, что начало секции будет располагаться внутри LCU, когда последняя CU предыдущей секции и первая CU текущей секции находятся в одной и той же LCU. Фиг. 2 схематично иллюстрирует это. Граница между первой секцией 2 и второй секцией 3 совпадает с границей двух смежных, в соответствии с порядком обработки, LCU 10A, 10B. Граница между второй секцией 3 и третьей секцией 4 вместо этого выровнена с границей двух единиц 20A, 20B кодирования, каждая из которых имеет одну четвертую в размере по сравнению с LCU 10. Соответственно, граница между третьей секцией 4 и четвертой секцией 5 на Фиг. 2 выровнена с границей двух CU 30A, 30B, каждая из которых составляет 1/16 в размере по сравнению с LCU 10.

Однако иерархическое деление и начала секций, выровненные с небольшими CU, возможно даже SCU, могут привести к еще большим адресам секций по сравнению с

MPEG-4/AVC. Например, к ранее рассмотренному примеру, изображение 1280×960 пикселей может иметь 19200 потенциальных позиций начала секций, если начала секций могут выравниваться с границей SCU 8×8 пикселей. Если адреса секций также представлены в форме координат X и Y, то для них должно требоваться 8+7=15 бит.

5 В соответствии с вариантами осуществления, термин "секция" используется для обозначения независимо кодируемой и декодируемой части изображения или видеокadra. Изображение может, таким образом, быть составлено из одной секции или множества, т.е. по меньшей мере двух секций.

10 Фиг. 1 является блок-схемой, иллюстрирующей способ кодирования изображения или видеокadra, содержащего N секций. Параметр N является положительным целым числом, равным или большим, чем два. Упомянутый способ начинается на этапе S1, на котором генерируется закодированное представление секции для каждой секции в изображении. Это закодированное представление секции генерируется на основе значений пикселей у пикселей в упомянутой секции в соответствии с хорошо известными схемами кодирования, такими как интер- или интра-кодирование.

15 Следующие этапы S2 и S3 устанавливают так называемые флаги секций для секций в изображении. Более подробно, этап S2 устанавливает флаг секции, ассоциированный с первой секцией из N секций в изображении, в первое определенное значение, такое как  $1_{bin}$ , или логическая единица, или какой-нибудь другой определенный символ, чтобы  
20 указать, что настоящая секция является первой секцией изображения и, таким образом, имеет свое начало секции в определенной позиции в изображении, обычно в верхнем левом углу изображения. Этап S3 соответственно устанавливает флаг секции, ассоциированный с каждой секцией из остальных N-1 секций изображения, во второе определенное значение, такое как  $0_{bin}$ , или логический ноль, или какой-нибудь другой  
25 определенный символ. Это означает, что флаги секций могут использоваться как дополнительная информация для того, чтобы проводить дифференциацию между первой секцией в изображении и остальными секциями.

30 Этим остальным секциям необходимо иметь ассоциированные адреса секций, чтобы обеспечить декодеру возможность идентифицировать, где, в пределах области изображения, находятся начала секций. В этом нет необходимости для первой секции, которая вместо этого имеет определенную начальную позицию в пределах изображения, и предпочтительно начинается в первом пикселе изображения в верхнем левом углу. Альтернативно, изображение может быть разделено на области на более высоком  
35 уровне, чем секции. В таком случае, определенная начальная позиция может быть началом отсчета такой области в изображении.

Этап S4 поэтому генерирует адрес секции для каждой секции из остальных N-1 секций, исключая, по этой причине, первую секцию, для которой не требуется адрес секции в соответствии с вариантами осуществления. Адрес секции, генерируемый на этапе S4  
40 для секции, обеспечивает возможность идентификации позиции первой единицы кодирования секции и соответственно начала секции в пределах изображения. Первая единица кодирования тогда составляет первую единицу кодирования секции, а предшествующая, в соответствии с определенным порядком обработки, единица кодирования тогда является последней единицей кодирования предыдущей секции в изображении. Первая единица кодирования может быть любым блоком пикселей, а в  
45 MPEG-4/AVC первая единица кодирования является первым макроблоком секции. Соответственно, в HEVC первая единица кодирования может быть LCU, но с иерархическим разделением она, преимущественно, может быть любой единицей кодирования, от размера LCU до размера SCU, если не наложены какие-либо

ограничения на то, где позиции начала секций могут быть найдены в изображении, что обсуждается далее.

Порядок обработки, в котором единицы кодирования изображения обрабатываются, т.е. кодируются, а затем последовательно декодируются, может быть любым известным порядком обработки. Примером такого порядка обработки является порядок растрового сканирования, или любой другой порядок кодирования/декодирования, такой как порядок Мортон (Morton) или Z-порядок, который рассматривается далее.

N кодированных представлений секций, N-1 адресов секций и N флагов секций затем используются для генерирования кодированного представления изображения на этапе S5. Кодированное представление изображения обычно имеет форму последовательности или потока бит, хотя другие символные алфавиты, кроме двоичного алфавита, могут использоваться и попадают в объем вариантов осуществления, такие как шестнадцатеричный или десятичный алфавит. Фиг. 6 является схематической иллюстрацией варианта осуществления такого кодированного представления 50 изображения. Как правило, кодированное представление 50 изображения содержит две главных части для каждой секции, т.е. заголовок 54 секции и кодированные данные 56. Заголовок 54 секции обычно содержит флаг секции, который устанавливается для секции на этапе S2 или S3, например, в форме кодового слова `first_slice_in_pic_flag`. Заголовок 54 секции каждой из остальных секций, кроме первой секции, в изображении предпочтительно также содержит адрес секции, генерируемый на этапе S4, например, в форме кодового слова `first_cu_in_slice` или кодового слова `slice_address`. В некоторых вариантах осуществления, в заголовок 54 секции может включаться дополнительная информация, включая, например, тип кодирования секции.

Кодированные данные 56 переносят кодированные данные изображения пикселей в упомянутой секции, т.е. кодированные представления секций, генерируемые на этапе S1.

Кодированное представление 50 может опционально также содержать, или же быть ассоциированным с набором параметров изображения (picture parameter set, PPS) и/или набором параметров последовательности (sequence parameter set, SPS) 52. PPS/SPS 52 может быть частью кодированного представления 50 изображения. В таком случае, каждое кодированное представление 50 изображения видеопотока может иметь соответствующее поле PPS и/или поле SPS 52. В альтернативном подходе, не всем таким кодированным представлениям 50 изображения видеопотока необходимо переносить поле PPS и/или поле SPS 52. Например, первое кодированное представление 50 изображения видеопотока может включать в себя поле PPS и/или поле SPS 52, а далее такие поля включаются в состав только кодированное представление другого изображения видеопотока, если какие-нибудь из параметров в поле PPS и/или поле SPS 52 обновляются или изменяются. Дополнительным вариантом является сигнализирование поля PPS и/или поля SPS 52 вне полосы по отношению к представлению 50 кодированного изображения. В таком случае, поле PPS и/или поле SPS 52 может передаваться отдельно от представления 50 кодированного изображения, но таким способом, чтобы декодер был в состоянии идентифицировать, к какому видеопотоку или кодированному представлению изображения принадлежит данное поле PPS и/или поле SPS 52. Это может достигаться посредством включения идентификатора сеанса, потока и/или изображения как в кодированное представление 50 изображения, так и в поле PPS и/или поле SPS 52.

Как было упомянуто выше, секции являются независимо кодируемыми и декодируемыми единицами изображения. Это означает, что генерирование

кодированных представлений секций на этапе S1, установка флагов секций на этапов S2, S3 и генерирование адресов этапа S4 может выполняться последовательно, или же, по меньшей мере частично, параллельно для различных секций в изображении.

Параллельное кодирование секций, обычно, уменьшает общее время кодирования изображения. Способ, содержащий этапы с S1 по S5, обычно, затем повторяется для любых остальных изображений или видеокадров, например, для видеопотока. В дополнение, этап S2 или этапы S3/S4 могут выполняться после, перед или по меньшей мере частично параллельно с этапом S1.

В следующем описании, варианты осуществления настоящего изобретения будут рассматриваться в связи с HEVC, как примером стандарта видео кодирования и декодирования, к которому могут применяться варианты осуществления. Однако это следует рассматривать лишь как иллюстративный пример стандарта кодирования/декодирования изображения или видео, который может использоваться с вариантами осуществления изобретения, причем варианты осуществления им не ограничиваются.

В соответствии с HEVC, изображение или видеокадр содержит множество LCU, имеющих выбранный размер в терминах числа пикселей. Это означает, что каждая LCU изображения предпочтительно имеет одно и то же число пикселей. LCU могут быть прямоугольными, но предпочтительно являются квадратными, т.е. содержат  $M \times M$  пикселей, где  $M$  является определенным положительным целым числом, равным или, предпочтительно, большим, чем два; и предпочтительно  $M=2^m$ , где  $m$  является положительным целым числом. В не ограничивающих примерах подходящих значений,  $M$  равно 64 или 128. Каждая LCU изображения может потенциально быть иерархически разделена на множество меньших CU, имеющих соответствующие размеры, которые являются меньшими, чем выбранный размер LCU.

Как правило, иерархическое разделение LCU включает в себя разделение LCU по образу квадрадерева. Как хорошо известно в данной области техники, квадрадерево представляет собой древовидную структуру данных, в которой каждый внутренний узел имеет ровно четыре потомка. Таким образом, иерархическое разделение LCU предполагает разделение двумерного пространства изображения, занимаемого LCU, посредством рекурсивного разделения, на четыре квадранта или области. В предпочтительном варианте осуществления, рекурсивное разделение включает в себя разделение на четыре одинаковых по размеру CU. В соответствии с вариантами осуществления, если единица кодирования, т.е. либо LCU, либо меньшая CU, разделяется, то так называемый флаг разделения единицы кодирования, ассоциированный с единицей кодирования, устанавливается в определенное значение, предпочтительно, равное  $1_{bin}$  или логической единице или любому другому определенному символу, указывающему, что единица кодирования иерархически делится на множество, предпочтительно, на четыре, меньших CU. Соответственно, если подходящая для разделения единица кодирования, т.е. единица кодирования, которая больше, чем SCU, не делится, то флаг разделения единицы кодирования, ассоциированный с единицей кодирования, предпочтительно, вместо этого устанавливается в  $0_{bin}$  или логический ноль или любой другой определенный символ. Термин "подходящая для разделения" единица кодирования относится здесь к единице кодирования, которая может быть иерархически разделена на множество, предпочтительно на четыре меньших единиц кодирования. Как правило, любая единица кодирования кроме SCU является подходящей для разделения единицей кодирования. Хотя единица кодирования может разделяться на меньшие единицы кодирования, она не должна делиться, например, если такое

разделение не улучшает качество кодирования изображения.

Иерархическое деление в вариантах осуществления предпочтительно обрабатывает LCU за LCU в определенном порядке обработки, таком как порядок растрового сканирования. Порядок растрового сканирования, как правило, содержит проход слева направо и сверху вниз. Альтернативно, может быть использован другой порядок кодирования/декодирования, например, порядок Мортонa или Z-порядок. Фиг. 4 иллюстрирует принципы Порядка Мортонa. Если LCU делится на, предпочтительно, четыре равных по размеру, CU, то эти CU могут далее обрабатываться в порядке обработки для того, чтобы выбрать, следует ли их иерархически разделять на, предпочтительно, четыре равных по размеру и еще более меньших, CU. Этот порядок обработки может быть тем же порядком, что и порядок, в котором обрабатываются LCU в изображении. В альтернативном подходе, LCU обрабатываются в порядке растрового сканирования, а CU обрабатываются в порядке кодирования/декодирования, например, в порядке Мортонa. Представленные выше порядки обработки являются только примерами порядков, которые могут использоваться, и варианты осуществления не ограничиваются ими.

Таким образом, для каждой единицы кодирования определяется, разделять ли единицу кодирования далее на меньшие единицы кодирования, если единица кодирования не является SCU, которая не может далее иерархически разделяться. Каждый раз, когда единица кодирования делится, флаг разделения единицы кодирования, ассоциированный с единицей кодирования, предпочтительно устанавливается в единицу, а если определяется, что единица кодирования далее не делится на меньшие единицы кодирования, то ее ассоциированный флаг разделения единицы кодирования предпочтительно устанавливается в ноль. SCU обычно нет необходимости иметь какой-либо ассоциированный флаг разделения единицы кодирования, поскольку она не может делиться далее.

Решение, разделять ли единицу кодирования, основывается на процессе кодирования. Например, область изображения, которая представляет довольно однородный фон, более эффективно представляется с использованием CU большого размера, например, LCU, по сравнению с использованием разделения области изображение на меньшие единицы кодирования. Однако, области изображения с мелкими деталями или с большим количеством таких деталей не могут, как правило, быть правильно представлены, если используются большие единицы кодирования. В таком случае, более эффективно и предпочтительно с точки зрения качества кодирования использовать множество меньших CU для этой области изображения. Выбор того, делить ли далее CU, может, таким образом, выполняться в соответствии с технологиями, описанными в уровне техники и предпочтительно основываемыми на качестве и эффективности кодирования.

Флаги разделения единицы кодирования, генерируемые во время кодирования секций на этапе S1 по Фиг. 1 обычно включаются в часть 56 кодированных данных кодированного представления 50 изображения, как иллюстрируется на Фиг. 6.

Фиг. 5 является блок-схемой, иллюстрирующей дополнительные этапы способа кодирования Фиг. 1. Упомянутый способ начинается на опциональном этапе S10, где для изображения определяется иерархическая степень разбиения. Иерархическая степень разбиения определяет иерархический уровень для выравниваний границ секций в пределах изображения. Этот иерархический уровень определяет и ограничивает размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в изображении. Иерархический уровень и определенная иерархическая степень разбиения, таким образом, определяют максимальное число

позиций начала для секций, которые являются потенциально доступными в изображении, и в которых может размещаться начало секции. Это означает, что иерархическая степень разбиения определяет число адресуемых CU в пределах изображения, где начало секции может выравниваться между границей такой адресуемой CU и предыдущей, в соответствии с определенным порядком обработки, CU в изображении.

Например, LCU, имеющая выбранный размер в  $64 \times 64$  пикселей, может иметь степень разбиения секций, определяющую уровень степени разбиения от 0 до 3, с размером SCU равным  $8 \times 8$  пикселей. В таком случае, уровень 0 степени разбиения указывает, что начала секций могут выравниваться только с границами между LCU. В случае изображения в  $1280 \times 960$  пикселей это предполагает  $20 \times 15 = 300$  возможных позиций начал секций. Если уровень степени разбиения вместо этого равен 1, то наименьшая возможная единица кодирования, на которой может выравниваться начало секции, составляет вместо этого  $32 \times 32$  пикселей, с общим числом в  $40 \times 30 = 1200$  возможных позиций начал секций. Соответственно, уровень 2 степени разбиения означает, что имеется  $80 \times 60 = 4800$  возможных позиций начал секций, поскольку начало секций может выравниваться с CU размера  $16 \times 16$  пикселей или больше. Наконец, уровень 3 степени разбиения указывает, что начала секций могут выравниваться с границами SCU, предоставляя общее число  $160 \times 120 = 19200$  возможных позиций начал секций.

Причина, почему хотелось бы иметь возможность выбирать между этими случаями от 220 до 14400 возможных позиций начал секций, в настоящем примере состоит в том, что чем больше позиций для начал секций, которые доступны в изображении, тем более длинные адреса секций необходимы, из-за чего увеличиваются служебные данные в кодированных данных изображения. Таким образом, если нет никаких конкретных требований на целевые размеры секций, например, по размещению данных секции в одном пакете IP данных, то может быть выгодным ограничить число позиций начал секций в изображении, чтобы таким образом уменьшить количество данных для адресов, которые необходимо генерировать и передавать декодеру.

Следующий этап S11 определяет длину адреса секции для N-1 секций в изображении, где длина задается в терминах числа бит или других символов в адресе секции. Длина адреса секции зависит от числа потенциальных позиций начал секций и числа адресуемых единиц кодирования в пределах изображения. В случае MPEG-4/AVC это число позиций начал секций зависит от размера изображения, поскольку начала секций могут выравниваться только с границами макроблоков. Это означает, что с учетом общего размера изображения, число возможных позиций начал секций может вычисляться для фиксированного размера макроблока. Длина адреса секции может затем вычисляться исходя из этого числа, например, как  $\log_2(P)$  или  $\log_2(P-1)$ , где P представляет собой число возможных позиций начал секций и соответственно общее число возможных адресов секций в изображении. Размер изображения обычно включается в поле заголовка, ассоциированное с кодированными представлениями изображения, или может находиться в ранее упомянутом поле PPS или поле SPS 52 в кодированном представлении 50 изображения, или ассоциированном с кодированным представлением 50 изображения, смотрите Фиг. 6.

В HEVC длина адреса секции предпочтительно определяется на этапе S11 на основе иерархической степени разбиения, которая определяется на этапе S10 для изображения. Иерархическая степень разбиения может затем использоваться для определения размера адресуемых единиц кодирования и соответственно наименьших возможных по размеру единиц кодирования, на которых может выравниваться начало секции. Информация о степени разбиения предпочтительно используется вместе с информацией о размере

изображения или об общем числе LCU в изображении, для того, чтобы определить длину адреса секции на этапе S11.

В варианте осуществления, размер LCU в изображении может быть заранее определен и, таким образом, известен кодеру и декодеру. Например, размер LCU может быть равен 128×128 пикселей или 64×64 пикселей. Таким образом, не требуется определения или сигнализации размера LCU. Соответственно, размер SCU в изображении может быть заранее определен. Примерами таких фиксированных и заранее определенных размеров SCU, которые могут использоваться, являются размеры в 16×16 пикселей или 8×8 пикселей.

В альтернативных вариантах осуществления, процесс кодирования может дополнительно определить размер LCU и/или размер SCU для использования для текущего изображения (текущих изображений). Может быть выгодным, таким образом, адаптировать размеры LCU и/или SCU к конкретным характеристикам настоящего изображения. Например, для некоторых изображений, которые представляют собой, в основном, вид равномерного однородного фона, LCU и SCU больших размеров могут быть предпочтительными, и могут приводить к более эффективному кодированию, по сравнению с изображениями с большим числом мелких деталей, где могут быть предпочтительными LCU и SCU меньших размеров.

В варианте осуществления, размер LCU и/или размер SCU, поэтому, определяются во время кодирования, например, на основе значений пикселей в изображении. Уведомление об определенном размере LCU и/или уведомление об определенном размере SCU затем ассоциируется с кодированным представлением изображения. Ассоциация уведомления (уведомлений) и кодированного представления изображения может выполняться в соответствии с различными вариантами осуществления. Например, уведомления могут включаться в состав кодированного представления изображения. Альтернативой является включение уведомлений в состав PPS или SPS.

Размер SCU может затем быть определен на основе параметра  $\log_2\_min\_coding\_block\_size\_minus3$ , предпочтительно, посредством вычисления параметра  $Log2MinCUSize$ , как  $Log2MinCUSize = \log_2\_min\_coding\_block\_size\_minus3 + 3$ . Этот параметр  $Log2MinCUSize$  затем используется в качестве представления размера SCU, и выдает размер SCU как  $MinCUSize = (1 \ll Log2MinCUSize)$ , где знак « $\ll$ » обозначает оператор левого сдвига. В зависимости от значения параметра  $Log2MinCUSize$  и соответственно параметра  $\log_2\_min\_coding\_block\_size\_minus3$  размер SCU может быть равен 8 или 16, в качестве иллюстративных примеров.

Размер LCU предпочтительно определяется по отношению к размеру SCU, посредством определения параметра  $\log_2\_diff\_max\_min\_coding\_block\_size$ . Более подробно, параметр  $Log2MaxCUSize$  вычисляется как  $Log2MaxCUSize = \log_2\_min\_coding\_block\_size\_minus3 + 3 + \log_2\_diff\_max\_min\_coding\_block\_size$ . Этот параметр  $Log2MaxCUSize$  затем используется, в качестве представления размера LCU, и выдает размер LCU как  $MaxCUSize = (1 \ll Log2MaxCUSize)$ . В зависимости от значения параметра  $Log2MaxCUSize$  и соответственно параметров  $\log_2\_min\_coding\_block\_size\_minus3$  и  $\log_2\_diff\_max\_min\_coding\_block\_size$ , размер LCU может быть равен 64 или 128, в качестве иллюстративных примеров.

Уведомлениями размера SCU и размера LCU могут таким образом быть параметры  $\log_2\_min\_coding\_block\_size\_minus3$  и  $\log_2\_diff\_max\_min\_coding\_block\_size$ .

В альтернативном варианте осуществления, размер LCU не определяется по отношению к размеру SCU. Это означает, что нет необходимости в параметре SCU для определения размера LCU.

Адрес секции, генерируемый для каждой секции, кроме первой секции, в изображении, может определять позицию начала секции и первой CU секции, в виде простого числа. Различные возможные позиции начал секций и адресуемые единицы кодирования, затем, нумеруются от нуля и выше. Например, изображение в 1280×960 пикселей имеет 4800 уникальных позиций начал секций, если иерархическая степень разбиения определяет, что размер наименьшей возможной единицы кодирования, на которой может выравниваться начало секции в изображении, составляет 16×16 пикселей. Эти позиции могут быть пронумерованы от 0 до 4799, таким образом, требуя 13-битных адресов секций.

Альтернативой является обработка координат X и Y отдельно. В приведенном выше примере, координата X находится в пределах от 0 до 79, а координата Y находится в пределах 0 до 59, таким образом, требуется 7 плюс 6 бит для адресов секций.

В качестве дополнительной альтернативы, можно определить адрес секции так, что координаты LCU и координаты суб-LCU могут быть получены из него. В таком случае, координаты позиции LCU в пределах изображения определяются для секции. В этом случае, начало секции и первая CU секции размещены в изображении внутри этой LCU. Координаты, тогда, являются относительными к глобальному началу отсчета или начальной точке, обычно, верхнему левому углу изображения. Координаты LCU могут, тогда, быть координатами LCU по отношению к этому глобальному началу отсчета, например, в терминах числа LCU или в терминах координат X и Y, как упомянуто выше. Также определяются координаты позиции первой CU и соответственно начала секции в пределах LCU. Эти координаты, тогда, являются относительными к локальному началу отсчета или начальной точке, обычно, верхнему левому углу LCU. Эти суб-LCU координаты могут также быть представлены, как число, или в терминах X и Y координат.

Адрес секции затем генерируется на основе LCU координат и суб-LCU координат и иерархической степени разбиения. Иерархическая степень разбиения используется при определении суб-LCU координат путем ограничения и определения возможных позиций начала для секции и первой CU секции в пределах LCU.

В варианте осуществления, первое, т.е. LCU, представление генерируется на основе определенных координат LCU, а второе, т.е. суб-LCU, представление генерируется на основе суб-LCU координат. Адрес секции может содержать эти два представления. Альтернативно, адрес секции генерируется таким способом, что LCU координаты и суб-LCU координаты могут определяться или вычисляться исходя из адреса секции.

Извлечение LCU и суб-LCU координат может быть выполнено в соответствии с приведенным ниже, не ограничивающим, но иллюстративным примером.

Иерархическая степень разбиения, определяемая на этапе S10, определяется посредством кодового слова `slice_granularity`. `slice_granularity` является обычно 2-битным значением в диапазоне от  $00_{bin}=0$  до максимально  $11_{bin}=3$ . Это обеспечивает четыре различных иерархических уровня. Если необходимо только два таких иерархических уровня, то `slice_granularity` может, вместо этого, задаваться 1-битным значением. Соответственно, для более чем четырех иерархических уровней, необходим 3-битный или более длинный параметр `slice_granularity`. Альтернативно, является возможным кодирование с переменной длиной для сигнализации иерархической степени разбиения.

Кодовое слово `slice_granularity`, опционально, определяется так, чтобы не быть больше, чем минимальное из двух других кодовых слов, определенных во время кодирования изображения или видеопотока: `Log2MaxCUSize-4` и `log2_diff_max_min_coding_block_size`. Кодовое слово `slice_granularity`, затем, во время декодирования,

используется для вычисления параметра SliceGranularity, в виде  $\text{SliceGranularity} = (\text{slice\_granularity} \ll 1)$ .

Адрес секции, генерируемый во время кодирования, является кодовым словом slice\_address. Это кодовое слово определяет адрес секции с разрешением согласно степени разбиения секции, с которым секция начинается. Длина адреса секции, т.е. slice\_address, как упомянуто выше, определяется на основе иерархической степени разбиения. В конкретном варианте осуществления, длина slice\_address в терминах числа бит равна  $\text{ceil}(\log_2(\text{NumLCUsInPicture}) + \text{SliceGranularity})$ .

Параметр NumLCUsInPicture определяет общее число LCU в изображении, и определяется на основе размера изображения и на основе размера LCU, который является либо фиксированным или определяется, как это упоминалось выше. NumLCUsInPicture может затем включаться в состав кодированного представления изображения или ассоциироваться с ним, например, включаться в поле PPS или поле SPS. Альтернативно, декодер способен самостоятельно вычислять параметр NumLCUsInPicture на основе размера LCU ( $\log_2\_min\_coding\_block\_size\_minus3$  и  $\log_2\_diff\_max\_min\_coding\_block\_size$ ) и общего размера изображения, который сигнализируется декодеру в кодированном представлении изображения, или ассоциируется с ним.

LCU часть адреса секции, в соответствии с порядком обработки, таким как порядок растрового сканирования, затем определяется, как  $\text{LCUAddress} = (\text{slice\_address} \gg \text{SliceGranularity})$ , где знак  $\gg$  обозначает оператор правого сдвига. Суб-LCU часть адреса секции, в соответствии с порядком обработки, таким как порядок Мортонa, вычисляется как  $\text{GranularityAddress} = \text{slice\_address} - (\text{LCUAddress} \ll \text{SliceGranularity})$ .

Адрес секции затем определяется на основе LCUAddress и GranularityAddress, как  $\text{SliceAddress} = (\text{LCUAddress} \ll (\log_2\_diff\_max\_min\_coding\_block\_size \ll 1)) + (\text{GranularityAddress} \ll ((\log_2\_diff\_max\_min\_coding\_block\_size \ll 1) - \text{SliceGranularity}))$ .

Адрес секции, генерируемый для остальных секций, исключая первую секцию, в изображении, может быть адресом фиксированной длины, причем длина адреса является фиксированной для изображения и зависит от размера наименьшей возможной единицы кодирования, на которой может выравниваться начало секции в этом изображении, и общего размера изображения. В качестве альтернативы, можно использовать представление переменной длины. Примером кода переменной длины, который может использоваться, является универсальное кодирование с переменной длиной (universal variable length coding, UVLC), описанное в Lee и Кuo, Моделирование сложности в H.264/AVC CAVLC/UVLC энтропийных декодерах, Международный симпозиум IEEE по схемам и системам, 2008 год, страницы 1616-1619 (Complexity Modeling of H.264/AVC CAVLC/UVLC Entropy Decoders, IEEE International Symposium on Circuits and Systems (ISCAS2008), 2008, pp. 1616-1619). Кратко, UVLC использует код EG (Exp-Golomb). Код EG для целого числа C без знака равен  $[P \text{ zeros}][1][\text{info}]$ , где  $P = \text{floor}(\log_2(C+1))$  и  $\text{info} = C + 1 - 2^P$ .

Адрес секции не только определяет позицию первой CU и соответственно начало секции, но дополнительно определяет размер наибольшей возможной CU, которая может занимать позицию в изображении, определяемую посредством адреса секции. Это означает, что данный размер зависит от позиции, определяемой адресом секции. Хотя адрес секции дает размер наибольшей возможной CU, которая может занимать позицию, размер первой CU не должен быть равным размеру этой наибольшей возможной CU, которая может занимать позицию. Фиг. 3А и 3В иллюстрируют эту

концепцию. На фигурах ссылочные номера 2, 3 обозначают две различные секции в изображении, а жирная линия определяет границу между двумя секциями 2, 3. Граница секции находится в этих примерах в пределах области изображения, занимаемой LCU 10. Ссылочный номер 20 обозначает CU, полученную для степени разбиения 1, когда LCU 10 иерархически делится на четыре CU 20. С детализацией 2 эта CU 20 может быть иерархически разделена на четыре меньшие CU 30. На Фиг. 3А и 3В, изображающих случай со степенью разделения 3, CU 30 может делиться на четыре SCU 40.

На Фиг. 3А первая CU секции 3, указана под номером 30, тогда как на Фиг. 3В она указана под номером 40В. Ссылочные номера 40 (Фиг. 3А) и 40А (Фиг. 3В) обозначают предыдущую CU для LCU 10 в соответствии с определенным порядком обработки, который в этом примере является порядком Мортонa. На обеих Фиг. 3А и 3В начало секции и позиция первой CU 30, 40В являются одинаковыми, хотя размеры первой CU 30, 40В отличаются в этих двух примерах. Адрес секции является, однако, одинаковым в обоих этих случаях, и размер наибольшей возможной CU 30, которая может занимать соответствующую позицию, является одинаковым. Между этими двумя случаями могут быть проведены различия (дифференциация) посредством дополнения адреса секции дополнительной информацией в терминах, так называемого, флага разделения единиц кодирования.

Например, предположим, что размер LCU составляет 128×128 пикселей, а соответствующий размер SCU составляет 16×16 пикселей. Далее предположим, что LCU 10 по Фиг. 3А и 3В состоят из двух секций 2, 3, тогда кодированное представление может быть определено следующим образом:

```

Slice_header_syntax( )//секция 2 на Фиг. 3А - включает в себя информацию адреса
split_coding_unit_flag=1//разделение 128×128 LCU на четыре 64×64 CU
split_coding_unit_flag=0//первая 64×64 CU не делится далее
code of first (код первой) 64×64 CU//кодированное представление значений пикселей
первой 64×64 CU
split_coding_unit_flag=1//разделение второй 64×64 CU на четыре 32×32 CU
split_coding_unit_flag=0//первая 32×32 CU не делится далее
code of first (код первой) 32×32 CU//кодированное представление значений пикселей
первой 32×32 CU
split_coding_unit_flag=1//разделение второй 32×32 CU на четыре SCU, нет
необходимости далее во флагах разделения единиц кодирования
code of first (код первой) SCU//кодированное представление значений пикселей первой
SCU
code of second (код второй) SCU//кодированное представление значений пикселей
второй SCU
code of third (код третьей) SCU//кодированное представление значений пикселей
третьей SCU
code of fourth (код четвертой) SCU//кодированное представление значений пикселей
четвертой SCU
split_coding_unit_flag=1//разделение третьей 32×32 CU на четыре SCU, нет
необходимости далее во флаге разделения единиц кодирования
code of first (код первой) SCU//кодированное представление значений пикселей первой
SCU
code of second (код второй) SCU//кодированное представление значений пикселей
второй SCU
code of third (код третьей) SCU//кодированное представление значений пикселей

```

третьей SCU

code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

5 split\_coding\_unit\_flag=1// разделение четвертой 32×32 CU на четыре SCU, нет необходимости далее во флаге разделения единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

10 code of third (код третьей) SCU//кодированное представление значений пикселей для пикселей третьей SCU

code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

split\_coding\_unit\_flag=1//разделение третьей 64×64 CU на четыре 32×32 CU

15 split\_coding\_unit\_flag=1//разделение первой 32×32 CU на четыре SCU, нет необходимости далее во флаге разделения единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

20 code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU//кодированное представление значений пикселей третьей SCU

code of fourth (код четвертой) SCU//кодированное представление значений пикселей предшествующей CU 40 на Фиг. 3А

25 Slice\_header\_syntax( )//секция 3 на Фиг. 3А - включает в себя информацию адреса split\_coding\_unit\_flag=0//флаг разделения единицы кодирования первой CU

устанавливается в ноль, поскольку размер первой CU равен размеру наибольшей возможной CU, которая может занимать позицию в изображении, определенную

30 посредством генерируемой информации адреса. Наибольшей возможной CU в этом адресе является единица 32×32, и нет необходимости во флагах разделения для разделения ниже, чем 32×32. Вместо этого, размер 32×32 извлекается из адреса и степени разбиения.

code of first (код первой) CU//кодированное представление значений пикселей CU 30 на Фиг. 3А

35 split\_coding\_unit\_flag=1//разделение 32×32 CU на четыре SCU, нет необходимости далее во флаге разделения единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

40 code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU//кодированное представление значений пикселей третьей SCU

code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

45 split\_coding\_unit\_flag=1//разделение 32×32 CU на четыре SCU, нет необходимости далее во флаге разделения единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU// кодированное представление значений пикселей третьей SCU

5 code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

split\_coding\_unit\_flag=0//нет далее деления для 64×64 CU

code of 64×64 (код единицы 64×64) CU//кодированное представление значений пикселей первой 64×64 CU

10 В варианте осуществления, иллюстрированном на Фиг. 3B, код для первой секции 2 будет тем же, как указано выше, тогда как для второй секции 3 код будет, вместо этого, следующим:

Slice\_header\_syntax( )//секция 3 на Фиг. 3B - включает в себя информацию адреса

split\_coding\_unit\_flag=1//флаг деления единицы кодирования первой CU

15 устанавливается в единицу, поскольку размер первой CU меньше, чем размер наибольшей возможной CU, которая может занимать позицию в изображении, определенную посредством генерируемой информации адреса, нет необходимости далее во флаге деления единиц кодирования.

code of first (код первой) CU//кодированное представление значений пикселей первой CU 40B на Фиг. 3B

code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU//кодированное представление значений пикселей третьей SCU

25 code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

split\_coding\_unit\_flag=1//разделение 32×32 CU на четыре SCU, нет необходимости далее во флаге деления единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU//кодированное представление значений пикселей третьей SCU

35 code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

split\_coding\_unit\_flag=1//разделение 32×32 CU на четыре SCU, нет необходимости далее во флаге деления единиц кодирования

code of first (код первой) SCU//кодированное представление значений пикселей первой SCU

code of second (код второй) SCU//кодированное представление значений пикселей второй SCU

code of third (код третьей) SCU//кодированное представление значений пикселей третьей SCU

45 code of fourth (код четвертой) SCU//кодированное представление значений пикселей четвертой SCU

split\_coding\_unit\_flag=0// нет далее деления для 64×64 CU

code of 64×64 (код единицы 64×64) CU//кодированное представление значений пикселей

первой 64×64 CU

Фиг. 7 является блок-схемой, иллюстрирующей способ декодирования кодированного представления изображения, содержащего множество секций. Упомянутый способ начинается на этапе S20, где флаг секции, ассоциированный с секцией, извлекается из представления кодированного изображения, предпочтительно, из заголовка секции, назначенного текущей секции в кодированном представлении изображения. Следующий этап S21 генерирует декодированное представление значений пикселей для пикселей в упомянутой секции на основе кодированного представления секции, ассоциированного с секцией и извлеченного из кодированного представления изображения, обычно, из части, соответствующей кодированным данным. Декодированные представления значений пикселей генерируются в соответствии с хорошо известными технологиями декодирования, такими как декодирование в интер- или интра-режиме.

Значение пиксела, как использовано здесь, обозначает любое значение свойства пиксела, назначаемое пикселу. В обычных осуществлениях для HEVC, значение пиксела является кодом цвета. Различные форматы цвета известны в данной области техники и могут использоваться в соответствии с вариантами осуществления. Например, код цвета может содержать обе компоненты яркость и цветность, обычно одно значение яркости и две компоненты цветности. Значение пиксела может поэтому быть значением яркости пиксела, значением цветности пиксела или более того оба значения яркость и цветность. Другой пример общего формата цветов является так называемым форматом RGB, что обозначает Red-Green-Blue (красный-зеленый-голубой цвета). Код цвета содержит тогда как значение красного, так и зеленого голубого цветов. Значение пиксела может тогда быть значением RGB, т.е. значением красного, значением зеленого или значением голубого цветов. Также варианты формата RGB, такие как RGBA, известны и могут использоваться в соответствии с вариантами осуществления.

Фактически, варианты осуществления не обязательно должны ограничиваться использованием кодов цветов в качестве значений пикселей, но также могут применяться к другим известным значениям пикселей, включающих в себя значения серой шкалы, нормальные значения, т.е. значения координат X, Y, Z и т.д.

Флаг секции, извлеченный на этапе S20, затем используется на этапе S22 для того, чтобы определить, является ли текущая секция первой секцией в изображении, которая, таким образом, не имеет какого-либо ассоциированного адреса секции, или является ли текущая секция не первой секцией, для которой, поэтому, требуется адрес секции.

Если на этапе S22 определяется, что текущая секция действительно является первой секцией, например, когда флаг секции имеет значение единицы, упомянутый способ переходит на этап S23. Этап S23 просто назначает значения пикселей, генерируемые на этапе S21, первой части изображения, начиная с определенного начала секции в изображении, обычно верхнего левого угла изображения. Значения пикселей обычно назначаются пикселям в определенном порядке обработки, таком как ранее упомянутые порядок Мортон или порядок растрового сканирования. В обычном варианте осуществления, применяемом к HEVC, единицы кодирования, меньше чем LCU, обрабатываются в порядке Мортон, тогда как LCU изображения обрабатываются в порядке растрового сканирования. Это предполагает, что декодирование начинается с первой LCU секции и затем, если эта LCU делится на меньшие CU, эти меньшие CU декодируются в порядке Мортон. После декодирования LCU процесс продолжается со следующей LCU в соответствии с порядком растрового сканирования, и любые меньшие CU в этой следующей LCU декодируются в порядке Мортон.

В конкретном варианте осуществления, этап S22 в действительности выполняется

перед этапом S21 для того, чтобы определить, что настоящая секция действительно является первой секцией в изображении. Затем кодированные данные представления кодированного изображения, принадлежащего текущей секции, декодируются и назначаются пикселям единицы кодирования в каждой единице кодирования. Это  
 5 означает, что этапы S21 и S23 затем формируют цикл, который проходит через различные CU секции и декодирует каждую CU, одну за другой, и назначает значение пикселя для пикселей CU в каждой CU, в соответствии с упомянутым выше порядком обработки.

Если на этапе S22 вместо этого определяется, что настоящая секция является не  
 10 первой секцией изображения, на основе значения ассоциированного флага секции, например, имеющего значение нуля, то упомянутый способ переходит на этап S24. Этап S24 извлекает адрес секции для секции из представления кодированного изображения обычно из заголовка секции. Адрес секции используется для того, чтобы идентифицировать начало секции в пределах изображения и таким образом, части  
 15 изображения, которая принадлежит секции. Следующий этап S25 затем назначает значения пикселей пикселям в идентифицированной части изображения, чтобы, таким образом, генерировать декодированную секцию.

Подобно этапам S21 и S23 выше, этапы S22 и S24 могут выполняться перед этапами S21 и S25, чтобы, таким образом, сначала заключить, что настоящая секция является  
 20 не первой секцией, а затем идентифицировать и считать адрес секции из кодированного представления изображения. После этого начало секции идентифицируется на основе адреса секции, и декодирование кодированных данных для секции может быть начато. Декодирование может переходить от одной CU к другой CU и затем назначать декодированные значения пикселей для пикселей в текущей CU перед переходом к  
 25 следующей CU, в соответствии с порядком обработки.

В альтернативном подходе этап S22 выполняется перед этапом S21. Таким образом, этап S22 исследует, установлен ли флаг секции или нет, и затем продолжается посредством генерирования декодированного представления значений пикселей и назначает, как указано на этапе S23, или сначала извлекает и использует информацию  
 30 адреса на этапе S24, для того, чтобы идентифицировать, какой части изображения назначать значения пикселей на этапе S25.

После того как все кодированные данные секции декодированы и назначены части пикселей, идентифицированной для секции на этапе S23 или S25, упомянутый способ завершается или продолжается далее в отношении следующей секции настоящего  
 35 изображения или другой секции другого изображения в видеопотоке. В таком случае, упомянутый способ Фиг. 7 повторяется для такой следующей или другой секции.

Однако, в предпочтительных вариантах осуществления, которые уменьшают общее время декодирования изображения, множество секций могут декодироваться параллельно. В таком случае, упомянутый способ Фиг. 7 выполняется для каждой из  
 40 этих секций, и решение этапа S22 проводится для каждой из секций на основе соответствующего флага секции, которая должна декодироваться.

Фиг. 8 является блок-схемой, иллюстрирующей дополнительные этапы способа на Фиг. 7. Упомянутый способ продолжается с этапа S22 Фиг. 7, на котором заключается, что настоящая секция является не первой секцией в изображении, на основе ее  
 45 ассоциированного флага секции. Следующий этап S30 извлекает информацию об иерархической степени разбиения для кодированного представления изображения. Как обсуждалось выше, информация о степени разбиения может включаться в состав кодированного представления изображения и затем извлекается из него на этапе S30.

Альтернативно, информация о степени разбиения может быть включена в ранее принятое кодированное представление изображения, относящееся к тому же видеопотоку. В таком случае, информация о степени разбиения извлекается из него и хранится для более позднего использования при декодировании следующих кодированных представлений изображения. Информация о степени разбиения может также передаваться отдельно от любого кодированного представления изображения, например, как отдельное поле PPS или поле SPS. Идентификаторы сеанса, изображения или потока могут тогда использоваться, чтобы идентифицировать соответствующую информацию степени разбиения для настоящего кодированного представления изображения.

Следующий опциональный этап S31 извлекает информацию о числе LCU в настоящем изображении. Эта информация может просто идентифицировать число таких LCU или может использоваться декодером для вычисления числа LCU. Например, кодовое слово NumLCUsInPicture может быть извлечено из кодированного представления изображения или из информации глобальных заголовков, например, поля PPS или SPS.

Альтернативно, NumLCUsInPicture вычисляется на основе информации общего размера изображения, извлеченного из кодированного представления изображения или из глобального заголовка, и информации размера LCU, например, ранее упомянутых кодовых слов  $\log_2\_min\_coding\_block\_size\_minus3$  и  $\log_2\_diff\_max\_min\_coding\_block\_size$ .

Следующий этап S32 определяет длину адреса секции текущей секции на основе информации об иерархической степени разбиения и, предпочтительно, на основе числа LCU в изображении. В конкретном варианте осуществления, длина адреса секции определяется как  $\text{ceil}(\log_2(\text{NumLCUsInPicture}) + \text{SliceGranularity})$ . Таким образом, в конкретном варианте осуществления длина адреса секции определяется на основе информации об иерархической степени разбиения и на основе информации о числе LCU в текущем изображении. Параметр SliceGranularity предпочтительно извлекается прямо из информации о степени разбиения slice\_granularity, как  $\text{SliceGranularity} = (\text{slice\_granularity} \ll 1)$ .

Упомянутый способ затем переходит к этапу S24 Фиг. 7, где адрес секции текущей секции извлекается из представления кодированного изображения на основе информации длины адреса секции, определенного на этапе S32. Таким образом, эта длина используется для того, чтобы идентифицировать, какие биты или символы кодированного представления изображения определяют адрес секции, посредством определения длины адреса секции, который предпочтительно имеет фиксированную начальную точку в заголовке секции, но в котором конечная точка зависит от длины адреса секции.

В том случае, когда настоящие варианты осуществления применяются к H.264/MPEG-4 AVC, не имеется доступной информации о степени разбиения, и этап S30 может таким образом быть пропущен. Этап S31 извлекает информацию или обеспечивает определение числа макроблоков в изображении, и эта информация используется на этапе S32 для того, чтобы определить длину адреса секции.

Фиг. 9 является блок-схемой, иллюстрирующей конкретный вариант осуществления идентификации позиции первой единицы кодирования секции и соответственно начала секции в пределах изображения. Упомянутый способ продолжается с этапа S24 Фиг. 7. Следующий этап S40 определяет адрес LCU, представляющий позицию LCU в пределах изображения, в котором присутствует первая CU и начало секции. Этап S40 использует адрес секции для того, чтобы определить адрес LCU. Например, параметр LCUAddress может быть определен как  $\text{slice\_address} \gg \text{SliceGranularity}$ , где slice\_address

представляет собой адрес секции. Следующий этап S41, соответственно, определяет адрес суб-LCU, представляющий позицию первой CU в пределах LCU, идентифицированной на этапе S40. Этот адрес суб-LCU также определяется на основе адреса секции. Например, параметр GranularityAddress определяется как slice\_address  
 5 (LCUAddress«SliceGranularity).

Части LCU и суб-LCU могут затем использоваться для вычисления конечного адреса секции, как  $(LCUAddress \ll \log_2\_diff\_max\_min\_coding\_block\_size \ll 1) + (GranularityAddress \ll ((\log_2\_diff\_max\_min\_coding\_block\_size \ll 1) - SliceGranularity))$ , который используется на этапе S42 для того, чтобы идентифицировать часть изображения, которая принадлежит  
 10 настоящей секции. Таким образом, эта часть начинается с начала секции и первой CU, идентифицированной на основе адреса секции, и продолжается в соответствии с порядком обработки в изображении, пока все кодированные данные секции не будут декодированы и назначены CU в изображении.

В альтернативных вариантах осуществления, адрес секции, извлеченный из  
 15 кодированного представления изображения, используется напрямую, чтобы идентифицировать начало секции и первую единицу кодирования. Адрес секции может тогда соответствовать числу позиций начал секций или адресуемых CU, в которых начинается секция. Адрес секции может тогда быть индексом в списке всех возможных адресов в порядке кодирования/декодирования. Альтернативно, из адреса секции  
 20 получают координаты X и Y и используются для определения начала секции. Дополнительный вариант состоит в получении или вычислении из адреса секции LCU координат и суб-LCU координат, как описывалось ранее.

Адрес секции не только определяет позицию первой CU секции и начало секции, но также предпочтительно определяет размер первой CU. Таким образом, размер первой  
 25 CU определяется, по меньшей мере частично, на основе адреса секции. Более подробно, адрес секции предписывает наибольший возможный размер в терминах числа пикселей, который может иметь первая CU. Это означает, что первая CU может иметь размер, равный этому наибольшему возможному размеру, или размер меньше, чем наибольший возможный размер. В последнем случае флаг разделения единицы кодирования далее  
 30 используется в дополнение к адресу секции для того, чтобы определить правильный размер первой CU, как обсуждается далее.

Например, первая CU может быть ассоциирована с флагом разделения единицы кодирования, включаемым в кодированное представление изображения, обычно в  
 35 часть, соответствующую кодированным данным. Значение флага разделения единицы кодирования затем используется вместе с адресом секции для того, чтобы определить правильный размер первой CU. Таким образом, если флаг разделения единицы кодирования устанавливается в определенное значение, предпочтительно равное  
 40 единице, то размер первой CU меньше, чем размер наибольшей возможной CU, которая может занимать позицию в пределах изображения, определяемую на основе адреса секции, см. Фиг. 3В. Однако если флаг разделения единицы кодирования устанавливается в другое определенное значение, предпочтительно равное нулю, то размер первой CU равен размеру наибольшей возможной CU, которая может занимать позицию в изображении, определенную адресом секции, см. Фиг. 3А.

Возможно, что первая CU ассоциируется с несколькими флагами разделения единицы кодирования. Например, если размер наибольшей возможной CU равен 32×32 пикселя, тогда как размер первой CU равен 8×8 пикселей, с размером LCU и размером SCU в 64×64 пикселей и 8×8 пикселей, код будет следующим:

split\_coding\_unit\_flag=1//32×32 CU делится на 16×16 CU

split\_coding\_unit\_flag=1//первая 16×16 CU делится на SCU, нет далее необходимости во флаге разделения единицы кодирования, поскольку теперь достигнут целевой размер первой CU, и это также размер SCU, подразумевающий, что теперь возможно дальнейшее разделение.

- 5 В некоторых вариантах осуществления, размер первой CU может определяться исключительно на основе адреса секции, без использования какого-нибудь флага разделения единицы кодирования в качестве дополнительной информации. Это является возможным, когда размер наибольшей возможной CU, которая может занимать позицию в пределах изображения, определенную на основе адреса секции, равен размеру SCU.
- 10 В таком случае, не является возможным делить эту наибольшую возможную CU далее, поскольку она в действительности является SCU.

Фиг. 10 является блок-схемой кодера или устройства 100 для кодирования изображения, содержащего множество секций. Устройство 100 содержит генератор 110 представлений, конфигурированный для генерирования соответствующего

15 кодированного представления секции для каждой секции в изображении на основе значений пикселей у пикселей в упомянутой секции. Генератор 110 представлений выполняет кодирование этих пикселей в соответствии с известными схемами кодирования, такими как интер- или интра-кодирование. Установщик 120 флагов упомянутого устройства 100 конфигурируется для установки флага секции, ассоциированного с секцией. Если настоящая секция является первой секцией в изображении, то установщик 120 флагов устанавливает флаг секции в первое

20 определенное значение, такое как единица, тогда как для оставшейся секции (остальных секций) в изображении соответствующий флаг секции устанавливается во второе определенное значение, такое как ноль.

25 Генератор 130 адресов генерирует соответствующий адрес секции для каждой секции, кроме первой секции, в изображении, т.е. для каждой секции с флагом секции, установленном в ноль. Адрес секции, генерируемый посредством генератора 130 адресов, обеспечивает возможность идентификации позиции первой CU секции в пределах изображения и соответственно начальной позиции секции в пределах изображения.

30 Устройство 100 также содержит менеджер 140 представлений, конфигурированный, чтобы включать соответствующие кодированные представления секций от генератора 110 представлений, флаги секций из установщика 120 флагов и адрес секции (адреса секций) из генератора 130 адресов секций в кодированное представление изображения. В конкретном варианте осуществления, флаг секции обеспечивается в кодированном

35 представлении перед адресом секции (адресами секций). В таком случае возможен синтаксический разбор, поскольку флаг секции решает, имеется ли поле адреса секции в кодированном представлении секции, или нет.

В варианте осуществления, генератор 130 адреса генерирует адрес секции на основе иерархической степени разбиения, определенной для изображения посредством

40 устройства 100. В таком случае, опциональное устройство 150 определения длины может быть реализовано в устройстве 100 для использования иерархической степени разбиения, чтобы определять длину адреса секции и соответственно число бит, которые должен содержать адрес секции. Устройство 150 определения длины дополнительно, предпочтительно, использует информацию об общем числе LCU в изображении при

45 определении длины адреса секции, причем это общее число LCU может вычисляться, как раскрыто выше. В другом варианте осуществления, устройство 150 определения длины отсутствует, и генератор 130 адреса сам определяет длину адреса секции.

В случае H.264/MPEG-4 AVC, устройство 150 определения длины предпочтительно

определяет длину адреса секции на основе числа макроблоков в изображении, которое может вычисляться на основе информации об общем размере изображения.

Генератор 130 адреса использует затем эту информацию о длине, когда генерирует адрес секции. В конкретном варианте осуществления, генератор 130 адреса определяет координаты позиции LCU в пределах изображения и координаты позиции суб-LCU в пределах LCU, как раскрыто выше. Адрес секции может тогда содержать представления для представлений этих позиций LCU и суб-LCU или может определяться исходя из них.

Устройство 100 преимущественно используется для того, чтобы кодировать множество секций параллельно, чтобы уменьшить общее время кодирования изображения или видеопотока.

Устройство 100 может осуществляться, по меньшей мере частично, в программном обеспечении. В таком варианте осуществления устройство 100 осуществляется в виде компьютерного программного продукта, сохраненного в памяти и загружаемого и запускаемого на компьютере общего назначения или специально адаптированном компьютере, процессоре или микропроцессоре, таком как центральный процессор (CPU). Программное обеспечение включает в себя элементы компьютерного программного кода или части кода программного обеспечения, выполняющие операции по меньшей мере генератора 110 представлений, установщика 120 флагов, генератора 130 адреса, менеджера 140 представлений и опционального устройства 150 определения длины. Программа может храниться целиком или частично на или в одном или более подходящих энергозависимых машиночитаемых средствах хранения данных или носителях, таких как RAM, или одном или более энергонезависимых машиночитаемых средствах хранения данных или носителях, таких как магнитные диски, CD-ROM диски, DVD диски, жесткие диски, ROM или флэш-память. Средства хранения данных могут быть локальными средствами хранения данных или обеспечиваемыми удаленно, например, в сервере данных. Программное обеспечение может, таким образом, загружаться в оперативную память компьютера или эквивалентной системы обработки для исполнения его процессором. Компьютер/процессор не должен быть выделенным, чтобы только выполнять описываемые выше функции, но может также выполнять другие задачи программного обеспечения. Не ограничивающий пример программного кода, используемого для определения устройства 100, включает в себя код SIMD (single instruction multiple data - один поток команд и много потоков данных).

Альтернативно, устройство 100 может быть реализовано в аппаратных средствах. Имеются многочисленные варианты элементов схем, которые могут использоваться и объединяться для достижения функций блоков упомянутого устройства 100. Такие варианты охватываются вариантами осуществления изобретения. Конкретными примерами аппаратной реализации упомянутого устройства 100 является осуществление в аппаратном обеспечении цифрового сигнального процессора (digital signal processor, DSP) и технологии интегральных схем, включающей в себя как электронные схемы общего назначения, так и прикладные схемы.

Фиг. 11 является блок-схемой варианта осуществления декодера или устройства 200 для декодирования кодированного представления изображения, содержащего множество секций. Устройство 200 содержит устройство 210 извлечения представлений, конфигурированное для извлечения флага секции, ассоциированного с секцией, предназначенной для декодирования из кодированного представления изображения обычно из заголовка секции в кодированном представлении изображения. Генератор 220 представлений обеспечивается в устройстве 200 для генерирования декодированного представления значений пикселей для пикселей в упомянутой секции на основе

кодированного представления изображения. Генератор 220 представлений генерирует значения пикселей в соответствии с известными технологиями, такими как схемы декодирования в интра- или интер-режиме.

5 Устройство 230 извлечения адресов действует, если флаг секции, извлеченный для текущей секции устройством 210 извлечения представлений, имеет второе определенное значение, например, равное нулю, указывающее, что упомянутая секция является не первой секцией в изображении. Устройство 230 извлечения адресов в этом случае считывает и извлекает адрес секции, ассоциированный с секцией, из кодированного представления изображения, например, из заголовка секции в кодированном представлении изображения. Устройство 240 назначения значений затем назначает значения пикселей, получаемые от генератора 220 представлений, по мере того как секция декодируется, пикселям в части секции, идентифицированной на основе адреса секции, извлеченного посредством устройства 230 извлечения адресов.

15 Если флаг секции, извлеченный устройством извлечения представлений для текущей секции, имеет первое определенное значение, например, равное единице, устройство 240 назначения значений может напрямую идентифицировать часть секции, которой должны назначаться значения пикселей от генератора 220 представлений. Это, как правило, первая секция изображения в порядке кодирования/декодирования, например, верхняя левая часть. Таким образом, в таком случае нет необходимости в адресе секции, чтобы идентифицировать эту первую часть изображения.

В случае HEVC реализации и если информация об иерархической степени разбиения назначается для кодированных данных изображения, устройство 250 извлечения информации о степени разбиения может обеспечиваться в устройстве 200, чтобы извлекать информацию об иерархической степени разбиения, применимой к настоящей секции, которая должна декодироваться. Устройство 250 извлечения информации о степени разбиения может извлекать информацию о степени разбиения из кодированного представления изображения или из поля глобального заголовка, такого как поле PPS или поле SPS, ассоциированного с кодированным представлением изображения. Информация о степени разбиения, извлеченная посредством устройства 250 извлечения информации о степени разбиения, используется опциональным устройством 260 определения длины, чтобы определить длину адреса секции и, таким образом, определить число бит, которые устройство 230 извлечения адресов должно считать, чтобы получить адрес секции. Альтернативно, это устройство 260 определения длины может отсутствовать, а устройство 230 извлечения адресов может самостоятельно определять длину адреса на основе информации о степени разбиения.

Опциональное устройство 270 извлечения информации об единицах кодирования выгодно осуществлять в устройстве 200, чтобы извлекать информацию об общем числе LCU в изображении из кодированного представления изображения, например, из поля глобального заголовка, поля PPS или поля SPS. Эта информация может быть ранее упомянутой информацией  $\log_2\_min\_coding\_block\_size\_minus3$  и  $\log_2\_diff\_max\_min\_coding\_block\_size$ , которая позволяет устройству 270 извлечения информации о единицах кодирования вычислять число LCU в изображении, выдавая информацию об общем размере изображения, которая предпочтительно является также доступной из кодированного представления изображения или из поля глобального заголовка, поля PPS или поля SPS.

45 Устройство 260 определения длины затем преимущественно определяет длину адреса секции на основе информации о степени разбиения из устройства 250 извлечения информации о степени разбиения и общего числа LCU, определенного устройством 270

извлечения информации о единицах кодирования.

В варианте осуществления, устройство 230 извлечения адресов конфигурируется, чтобы определять адрес первого представления или адрес LCU координат позиции LCU в пределах изображения на основе адреса секции, если текущей секцией является не первая секция в изображении. Устройство 230 извлечения адресов предпочтительно также определяет адрес второго представления или адрес суб-LCU координат позиции первой единицы кодирования секции и соответственно начало секции в пределах LCU. Адрес LCU и адрес суб-LCU затем используются устройством 230 извлечения адресов, чтобы идентифицировать часть изображения, которая принадлежит текущей секции, на основе адреса LCU и адреса суб-LCU, как раскрывается здесь.

Например, устройство 230 извлечения адресов может определять параметр `LCUAddress=slice_address»SliceGranularity` на основе адреса секции (`slice_address`) и на основе информации об иерархической степени разбиения (`SliceGranularity`). Адрес суб-LCU предпочтительно определяется как `GranularityAddress=slice_address - (LCUAddress«SliceGranularity)` на основе адреса секции (`slice_address`), информации об иерархической степени разбиения (`SliceGranularity`) и адреса LCU.

Генератор 220 представлений предпочтительно определяет размер первой CU в упомянутой секции в терминах числа пикселей на основе, по меньшей мере частично, адреса секции. Адрес секции затем определяет размер наибольшей возможной CU, которая может занимать позицию, определенную адресом секции в изображении. В варианте осуществления, размер первой CU определяется посредством генератора 220 представлений на основе исключительно адреса секции. Это является возможным, когда размер первой CU является равным размеру SCU, а дальнейшее деление CU не является возможным. В других вариантах осуществления, генератор 220 представлений дополнительно использует один или более флагов разделения единиц кодирования, включаемых в кодированное представление изображения вместе с адресом секции, чтобы определить размер первой CU. Если отдельный флаг разделения единицы кодирования равен нулю или какому-нибудь другому определенному значению, то размер первой CU является равным размеру наибольшей CU, которая может занимать позицию в пределах изображения, определенную адресом секции. Если флаг разделения единицы кодирования вместо этого равен единице или какому-нибудь другому определенному значению, то размер первой CU меньше, чем предпочтительно одна четверть размера наибольшей возможной CU, которая может занимать позицию в пределах изображения, определенную посредством адреса секции.

Например, если размер наибольшей возможной CU на начальной координате равен 32×32 пикселей (с размером LCU в 64×64 пикселей и размером SCU в 8×8 пикселей), то флаг разделения единицы кодирования (флаги разделения единиц кодирования) будет следующим:

```
split_coding_unit flag=0
для 32×32 пикселей размера первой CU
split_coding_unit_flag=1
split_coding_unit_flag=0
для 16×16 пикселей размера первой CU
split_coding_unit_flag=1
split_coding_unit_flag=1
для 8×8 пикселей размера первой CU
```

Устройство 200 может осуществляться, по меньшей мере частично, в программном обеспечении. В таком варианте осуществления устройство 200 осуществляется в виде

компьютерного программного продукта, сохраненного в памяти и загружаемого и запускаемого на компьютере общего назначения или специально адаптированном компьютере, процессоре или микропроцессоре, таком как центральный процессор (CPU). Программное обеспечение включает в себя элементы компьютерного программного кода или части кода программного обеспечения, выполняющие операции по меньшей мере устройства 210 извлечения представлений, генератора 220 представлений, устройства 230 извлечения адресов, устройства 240 назначения значений, опционального устройства 250 извлечения информации о степени разбиения, опционального устройства 260 определения длины и опционального устройства 270 извлечения информации о единицах кодирования. Программа может храниться целиком или частично на или в одном или более подходящих энергозависимых машиночитаемых средствах хранения данных или носителях, например, RAM или одном или более энергонезависимых машиночитаемых средствах хранения данных или носителях, таких как магнитные диски, диски CD-ROM, DVD диски, жесткие диски, в ROM или флэш-памяти. Средства хранения данных могут быть локальными средствами хранения данных или обеспечиваемыми удаленно, такими как в сервере данных. Программное обеспечение может, таким образом, загружаться в оперативную память компьютера или эквивалентную систему обработки для исполнения его процессором. Компьютер/процессор не должен быть выделенным, чтобы только выполнять описываемые выше функции, но может также выполнять другие задачи программного обеспечения. Не ограничивающий пример программного кода, используемого для определения устройства 200, включает в себя код SIMD.

Альтернативно, устройство 200 может осуществляться в аппаратных средствах. Имеются многочисленные варианты элементов схемы, которые могут использоваться и объединяться для достижения функций блоков упомянутого устройства 200. Такие варианты охватываются вариантами осуществления изобретения. Конкретными примерами аппаратного осуществления упомянутого устройства 200 является осуществление на аппаратном обеспечении цифрового сигнального процессора (DSP) и технологии интегральных схем, включающей в себя как электронные схемы общего назначения, так и прикладные схемы.

Фиг. 12 является блок-схемой медиа-терминала 300, содержащего устройство 200 для декодирования кодированного представления изображения. Медиа-терминал 300 может быть любым устройством, имеющим функции декодирования медиа, которое работает с кодированным битовым потоком, таким как видеопоток кодированных видеокладов, чтобы, таким образом, декодировать видеоклады и сделать видео данные доступными. Не ограничивающие примеры таких устройств включают в себя мобильные телефоны и другие портативные медиа-плееры, компьютеры, декодеры, игровые консоли и т.д. Медиа-терминал 300 содержит память 320 конфигурированную для хранения кодированного представления изображения, такого как кодированные видеоклады. Кодированное представление может генерироваться самим медиа-терминалом 300. В таком случае, медиа-терминал 300 предпочтительно содержит медиа-машину или устройство записи вместе с соединенным с ним кодером, таким как устройство для кодирования изображений по Фиг. 10. Альтернативно, кодированные представления генерируются каким-либо другим устройством и передаются беспроводным образом или передаются посредством проводов на медиа-терминал 300. Медиа-терминал 300 далее содержит приемопередатчик 310 (передатчик и приемник) или входной и выходной порт для выполнения передачи данных.

Кодированное представление доставляется из памяти 320 в устройство 200 для

декодирования, такое как устройство, иллюстрированное на Фиг. 11. Устройство 200 декодирует кодированное представление в декодированное изображение или в декодированные видеокadres. Декодированные данные предоставляются медиа-плееру 330, который конфигурирован, чтобы осуществлять рендеринг декодированных данных изображения или видеокadres в данные, которые являются отображаемыми на дисплее или экране 340 медиа-терминала 300 или соединенного с медиа-терминалом 300.

На Фиг. 12 медиа-терминал 300 иллюстрируется как содержащий как устройство 200 для декодирования, так и медиа-плеер 330. Однако это должно рассматриваться только как иллюстративный, но не ограничивающий пример реализации варианта осуществления изобретения для медиа-терминала 300. Также возможны распределенные осуществления, в которых устройство 200 и медиа-плеер 330 обеспечиваются в двух физически отдельных устройствах, которые попадают в пределы раскрытого здесь медиа-терминала 300. Дисплей 340 может также обеспечиваться как отдельное устройство, соединенное с медиа-терминалом 300, где происходит фактическая обработка данных.

Варианты осуществления, описанные выше, следует понимать как множество иллюстративных примеров настоящего изобретения. Специалистам в данной области техники будет понятно, что различные модификации, сочетания и изменения могут быть сделаны в вариантах осуществления без отклонения от объема настоящего изобретения. В частности, различные частные решения в различных вариантах осуществления могут сочетаться в других конфигурациях, где это технически возможно. Однако объем настоящего изобретения определяется прилагаемой формулой изобретения.

#### Формула изобретения

1. Способ кодирования изображения, содержащего  $N \geq 2$  секций, причем упомянутый способ содержит:

генерирование, для каждой секции из упомянутых  $N$  секций, кодированного представления секции на основе значений пикселей у пикселей в упомянутой секции;

установку флага секции, ассоциированного с первой секцией из упомянутых  $N$  секций, в первое определенное значение;

установку, для каждой секции из  $N-1$  остальных секций из упомянутых  $N$  секций, флага секции, ассоциированного с упомянутой секцией, во второе определенное значение;

генерирование, для каждой секции из упомянутых  $N-1$  остальных секций, адреса секции, обеспечивающего возможность идентификации позиции первой единицы кодирования упомянутой секции в пределах упомянутого изображения;

генерирование кодированного представления изображения для упомянутого изображения, содержащего упомянутые  $N$  кодированных представлений секций, упомянутые  $N-1$  адресов секций и упомянутые  $N$  флагов секций; и

определение длины упомянутых  $N-1$  адресов секций в терминах числа бит, как одно из  $\text{ceil}(\log_2(P))$  и  $\text{ceil}(\log_2(P-1))$ , причем  $\text{ceil}()$  обозначает функцию наименьшего целого числа, определенную как  $\text{ceil}(x) = \lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем  $x$ , а  $P$  обозначает общее число возможных адресов секций в упомянутом изображении.

2. Способ по п. 1, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый

выбранный размер, причем упомянутый способ дополнительно содержит определение длины упомянутых  $N-1$  адресов секций в терминах числа бит на основе иерархической степени разбиения, выбранной для упомянутого изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, где упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении.

3. Способ по п. 2, в котором определение упомянутой длины упомянутых  $N-1$  адресов секций содержит определение упомянутой длины упомянутых  $N-1$  адресов секций на основе упомянутой иерархической степени разбиения и на основе общего числа наибольших единиц кодирования в упомянутом изображении.

4. Способ декодирования кодированного представления изображения для изображения, содержащего  $N \geq 2$  секций, причем упомянутый способ содержит:

извлечение, из упомянутого кодированного представления изображения, флага секции, ассоциированного с секцией из упомянутых  $N$  секций;

генерирование декодированного представления значений пикселей для пикселей в упомянутой секции на основе кодированного представления секции, ассоциированного с упомянутой секцией и извлеченного из упомянутого кодированного представления изображения;

назначение упомянутых значений пикселей пикселям в первой части упомянутого изображения, если флаг секции, ассоциированный с упомянутой секцией и извлеченный из упомянутого кодированного представления изображения, имеет первое определенное значение;

извлечение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет второе определенное значение, адреса секции из упомянутого кодированного представления изображения; и

назначение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутых значений пикселей пикселям в части упомянутого изображения, идентифицированной на основе упомянутого адреса секции; и

определение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, длины упомянутого адреса секции в терминах числа бит, как одно из  $\text{ceil}(\log_2(P))$  и  $\text{ceil}(\log_2(P-1))$ , причем  $\text{ceil}()$  обозначает

функцию наименьшего целого числа, определенную как  $\text{ceil}(x) = \lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем  $x$ , а  $P$  обозначает общее число возможных адресов секций в упомянутом изображении.

5. Способ по п. 4, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый выбранный размер, причем упомянутый способ дополнительно содержит:

извлечение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, причем упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в

упомянутом изображении; и

определение длины упомянутого адреса секции в терминах числа бит на основе упомянутой информации об упомянутой иерархической степени разбиения.

5 6. Способ по п. 5, дополнительно содержащий извлечение информации об общем числе наибольших единиц кодирования в упомянутом изображении из упомянутого кодированного представления изображения, причем определение упомянутой длины упомянутого адреса секции содержит определение упомянутой длины упомянутого адреса секции на основе упомянутой информации об упомянутой иерархической степени разбиения и на основе упомянутого общего числа наибольших единиц кодирования.

10 7. Способ по п. 4, дополнительно содержащий:

определение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, первого представления координат позиции наибольшей единицы кодирования в пределах упомянутого изображения на основе упомянутого адреса секции;

15 определение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, второго представления координат позиции первой единицы кодирования упомянутой секции в пределах упомянутой наибольшей единицы кодирования; и

20 идентифицирование упомянутой части упомянутого изображения (1) на основе упомянутого первого представления и упомянутого второго представления.

8. Способ по п. 7, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый  
25 выбранный размер, причем упомянутый способ дополнительно содержит:

извлечение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень  
30 для выравнивания границ секций в пределах упомянутого изображения, причем упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении; и

определение длины упомянутого адреса секции в терминах числа бит на основе  
35 упомянутой информации об упомянутой иерархической степени разбиения, причем определение упомянутого первого представления содержит определение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутого первого представления на основе упомянутого адреса секции и упомянутой информации об упомянутой иерархической  
40 степени разбиения.

9. Способ по п. 7, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый  
45 выбранный размер, причем упомянутый способ дополнительно содержит:

извлечение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения,

причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, причем упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении; и

определение длины упомянутого адреса секции в терминах числа бит на основе упомянутой информации об упомянутой иерархической степени разбиения, причем определение упомянутого второго представления содержит определение, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутого второго представления на основе упомянутого адреса секции, упомянутой информации об упомянутой иерархической степени разбиения и упомянутого первого представления.

10. Устройство для кодирования изображения, содержащего  $N \geq 2$  секций, причем упомянутое устройство содержит:

генератор представлений, конфигурированный для генерирования, для каждой секции из упомянутых  $N$  секций, кодированного представления секции на основе значений пикселей у пикселей в упомянутой секции;

установщик флагов, конфигурированный для установки флага секции, ассоциированного с первой секцией из упомянутых  $N$  секций, в первое определенное значение и установки, для каждой секции из  $N-1$  остальных секций из упомянутых  $N$  секций, флага секции, ассоциированного с упомянутой секцией, во второе определенное значение;

генератор адресов, конфигурированный для генерирования, для каждой секции из упомянутых  $N-1$  остальных секций, адреса секции, который идентифицирует позицию первой единицы кодирования упомянутой секции (3-5) в пределах упомянутого изображения (1); и

менеджер представлений, конфигурированный для генерирования кодированного представления изображения для упомянутого изображения, содержащего упомянутые  $N$  кодированных представлений секций, упомянутые  $N-1$  адресов секций и упомянутые  $N$  флагов секций; и

определитель длины, конфигурированный для определения длины упомянутых  $N-1$  адресов секций в терминах числа бит, как одно из  $\text{ceil}(\log_2(P))$  и  $\text{ceil}(\log_2(P-1))$ , причем  $\text{ceil}()$  обозначает функцию наименьшего целого числа, определенную как  $\text{ceil}(x) = \lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем  $x$ , а  $P$  обозначает общее число возможных адресов секций в упомянутом изображении.

11. Устройство по п. 10, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый выбранный размер, причем упомянутое устройство дополнительно содержит определитель длины, конфигурированный для определения длины упомянутых  $N-1$  адресов секций в терминах числа бит на основе иерархической степени разбиения, выбранной для упомянутого изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, где упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении.

12. Устройство по п. 11, в котором упомянутый определитель длины конфигурирован

для определения упомянутой длины упомянутых N-1 адресов секций на основе упомянутой иерархической степени разбиения и на основе общего числа наибольших единиц (10) кодирования в упомянутом изображении.

13. Устройство для декодирования кодированного представления изображения для изображения, содержащего  $N \geq 2$  секций, причем упомянутое устройство содержит:

устройство извлечения представлений, конфигурированное для извлечения, из упомянутого кодированного представления изображения, флага секции, ассоциированного с секцией из упомянутых N секций;

генератор представлений, конфигурированный для генерирования декодированного представления значений пикселей для пикселей в упомянутой секции на основе кодированного представления секции, ассоциированного с упомянутой секцией и извлеченного из упомянутого кодированного представления изображения;

устройство извлечения адресов, конфигурированное для извлечения, если флаг секции, ассоциированный с упомянутой секцией и извлеченный из упомянутого кодированного представления изображения, имеет второе определенное значение, адреса секции, ассоциированного с упомянутой секцией, из упомянутого кодированного представления изображения; и

устройство назначения значений, конфигурированное для назначения упомянутых значений пикселей пикселям в первой части упомянутого изображения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет первое определенное значение, и назначения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутых значений пикселей пикселям в части упомянутого изображения, идентифицированной на основе упомянутого адреса секции; и

определитель длины, конфигурированный для определения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, длины упомянутых N-1 адресов секций в терминах числа бит, как одно из  $\text{ceil}(\log_2(P))$  и  $\text{ceil}(\log_2(P-1))$ , причем  $\text{ceil}()$  обозначает функцию наименьшего целого числа,

определенную как  $\text{ceil}(x) = \lceil x \rceil$ , и выводит наименьшее целое число не меньше, чем x, а P обозначает общее число возможных адресов секций в упомянутом изображении.

14. Устройство по п. 13, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый выбранный размер, причем упомянутое устройство дополнительно содержит:

устройство извлечения информации о степени разбиения, конфигурированное для извлечения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, причем упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении; и

определитель длины, конфигурированный для определения длины упомянутого адреса секции в терминах числа бит на основе упомянутой информации об упомянутой иерархической степени разбиения.

15. Устройство по п. 14, дополнительно содержащее устройство извлечения

информации о единицах кодирования, конфигурированное для извлечения информации об общем числе наибольших единиц кодирования в упомянутом изображении из упомянутого кодированного представления изображения, причем упомянутый определитель длины конфигурирован для определения упомянутой длины упомянутого адреса секции на основе упомянутой информации об упомянутой иерархической степени разбиения и на основе упомянутого общего числа наибольших единиц кодирования.

16. Устройство по п. 13, в котором упомянутое устройство извлечения адресов конфигурировано для i) определения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, первого представления координат позиции наибольшей единицы кодирования в пределах упомянутого изображения на основе упомянутого адреса секции, ii) определения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, второго представления координат позиции первой единицы кодирования упомянутой секции в пределах упомянутой наибольшей единицы кодирования, и iii) идентифицирования упомянутой части упомянутого изображения на основе упомянутого первого представления и упомянутого второго представления.

17. Устройство по п. 16, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый выбранный размер, причем упомянутое устройство дополнительно содержит:

устройство извлечения информации о степени разбиения, конфигурированное для извлечения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, причем упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении; и

определитель длины, конфигурированный для определения длины упомянутого адреса секции в терминах числа бит на основе упомянутой информации об упомянутой иерархической степени разбиения, причем упомянутое устройство извлечения адресов конфигурировано для определения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутого первого представления на основе упомянутого адреса секции и упомянутой информации об упомянутой иерархической степени разбиения.

18. Устройство по п. 16, в котором упомянутое изображение содержит множество наибольших единиц кодирования, которые имеют выбранный размер в терминах числа пикселей и могут быть иерархически разделены на множество единиц кодирования, имеющих соответствующие размеры, которые являются меньшими, чем упомянутый выбранный размер, причем упомянутое устройство дополнительно содержит:

устройство извлечения информации о степени разбиения, конфигурированное для извлечения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, информации об иерархической степени разбиения, ассоциированной с упомянутым кодированным представлением изображения, причем упомянутая иерархическая степень разбиения определяет иерархический уровень для выравнивания границ секций в пределах упомянутого изображения, причем

упомянутый иерархический уровень определяет размер наименьшей возможной адресуемой единицы кодирования, на которой может выравниваться начало секции в упомянутом изображении; и

5 определитель длины, конфигурированный для определения длины упомянутого адреса секции в терминах числа бит на основе упомянутой информации об упомянутой иерархической степени разбиения, причем упомянутое устройство извлечения адресов конфигурировано для определения, если упомянутый флаг секции, ассоциированный с упомянутой секцией, имеет упомянутое второе определенное значение, упомянутого  
10 второго представления на основе упомянутого адреса секции, упомянутой информации об упомянутой иерархической степени разбиения и упомянутого первого представления.

19. Медиа-терминал, содержащий:

память (320), хранящую кодированное представление изображения, содержащего множество секций и состоящего из множества наибольших единиц кодирования, имеющих выбранный размер в терминах числа пикселей; и

15 устройство для декодирования упомянутого кодированного представления по п. 13.

20

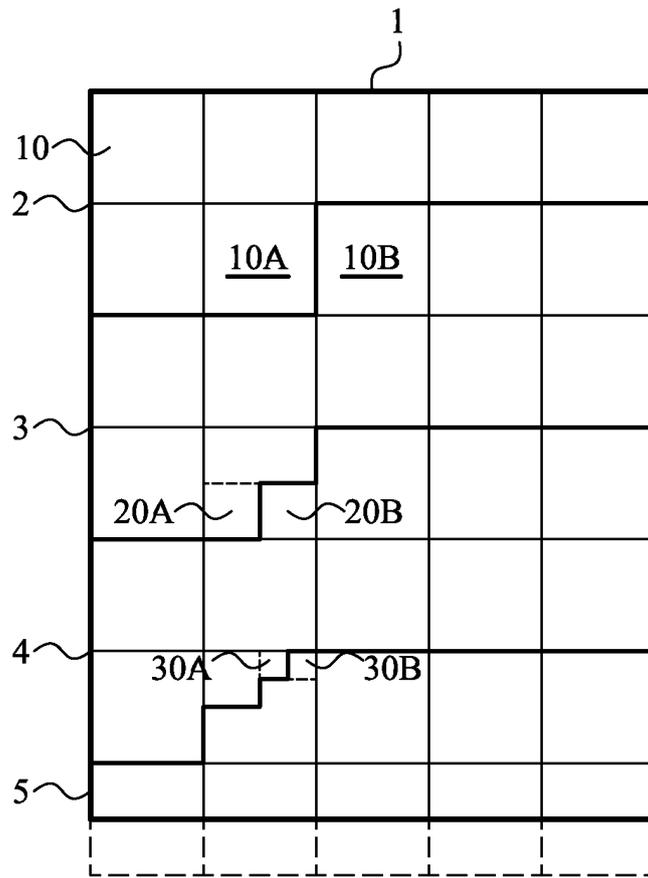
25

30

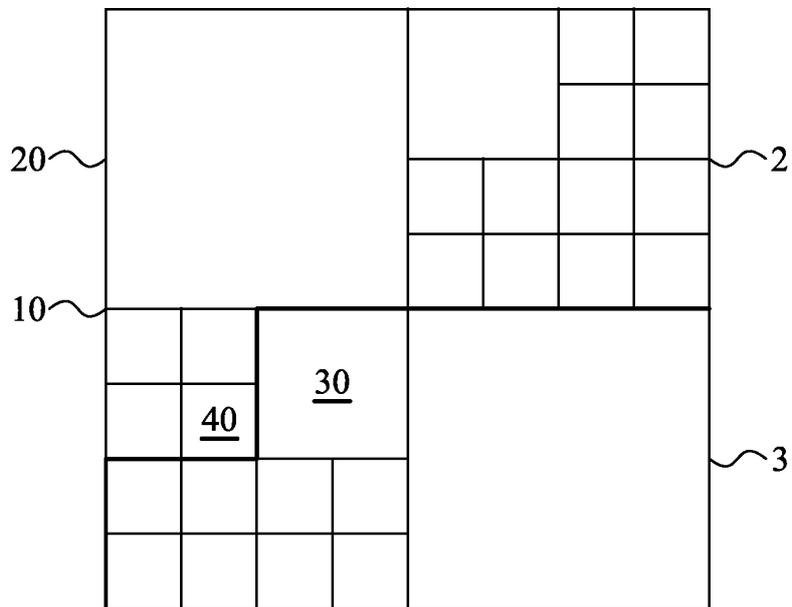
35

40

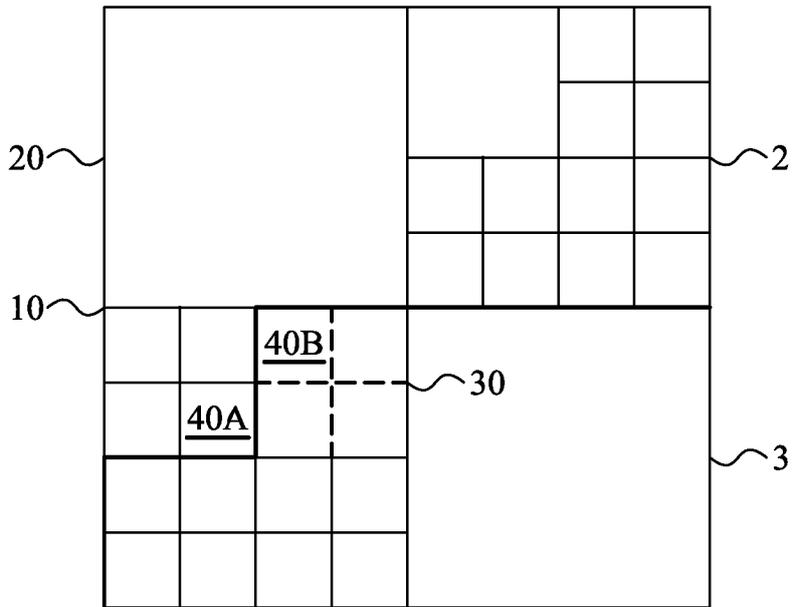
45



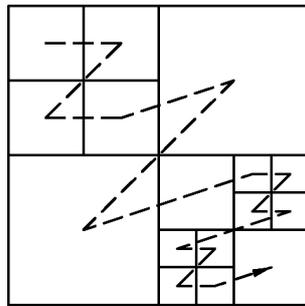
ФИГ.2



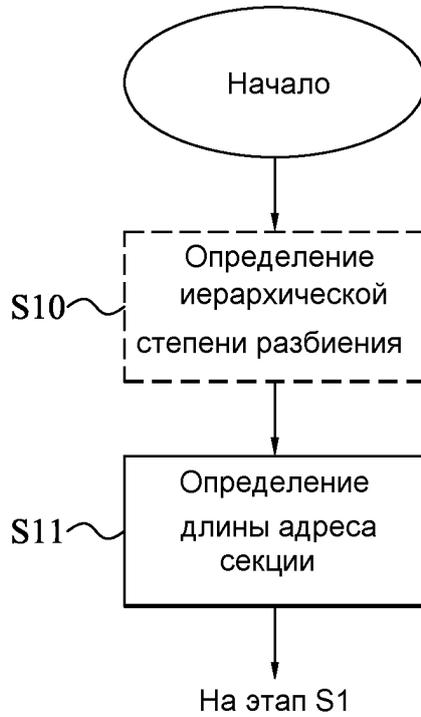
ФИГ.3А



ФИГ.3В



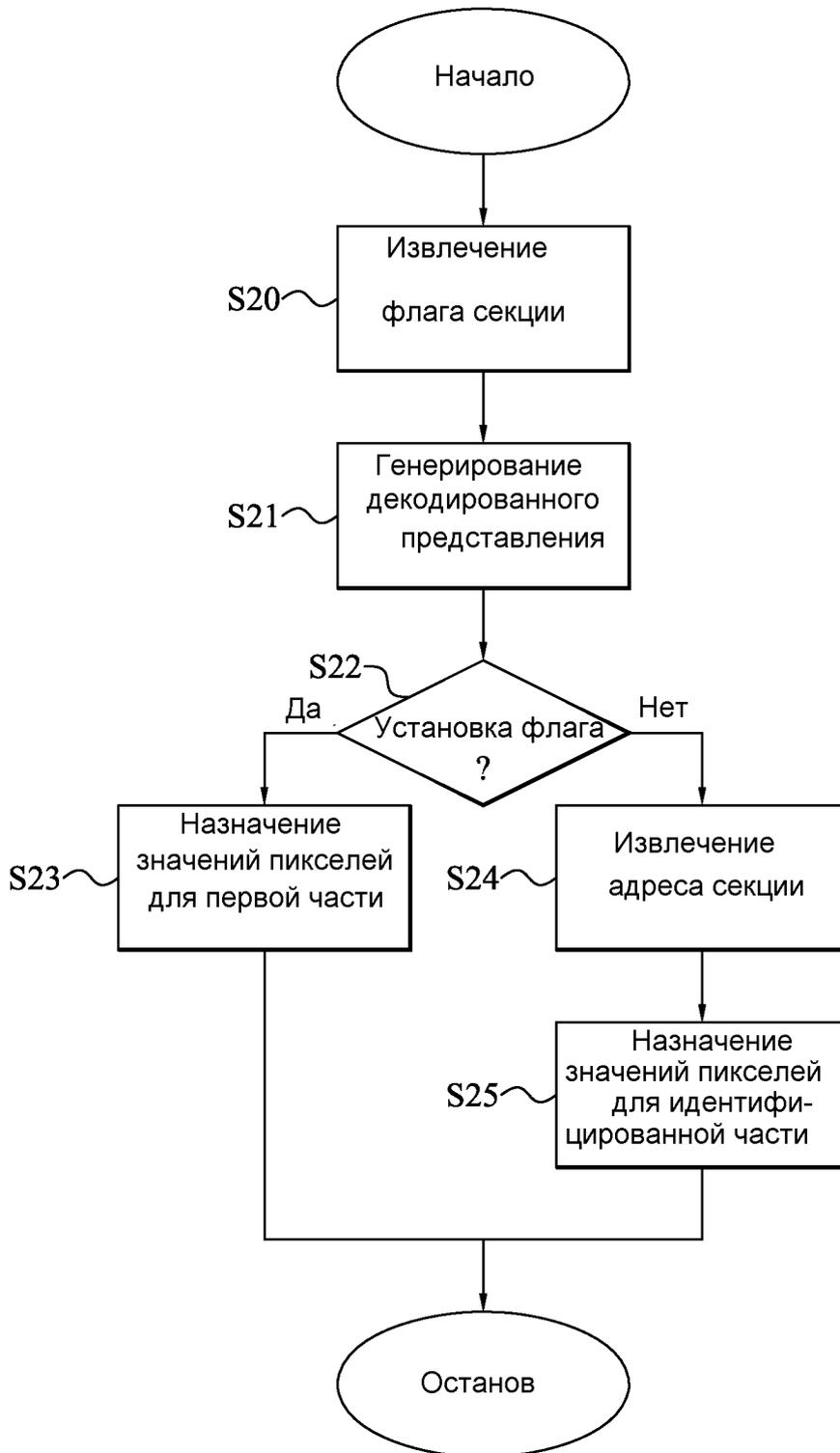
ФИГ.4



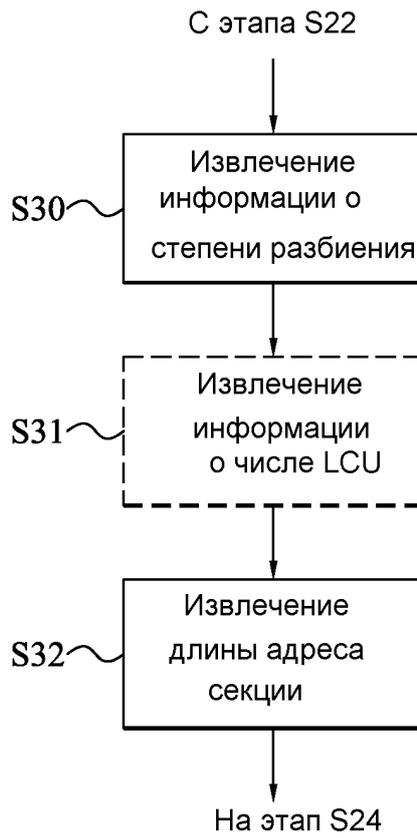
ФИГ.5



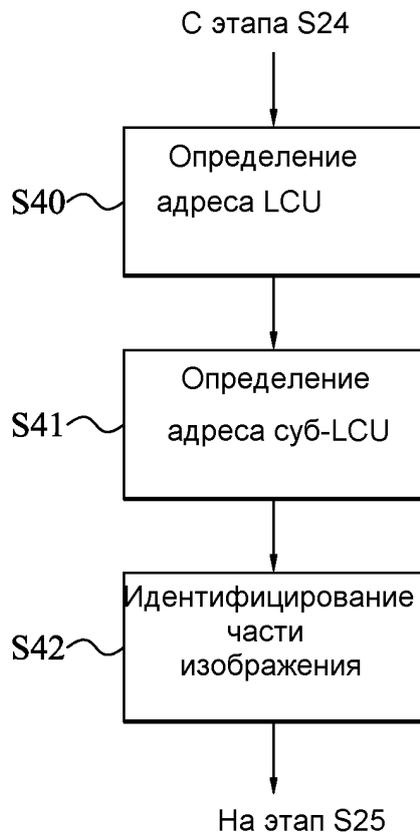
ФИГ. 6



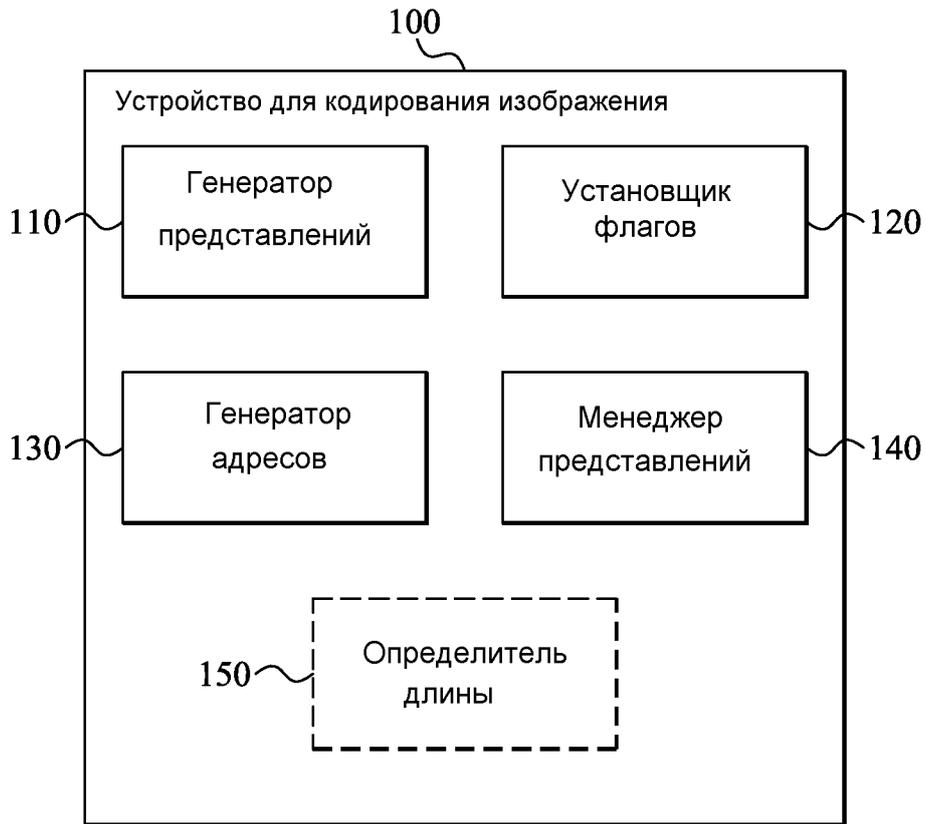
ФИГ. 7



**ФИГ.8**



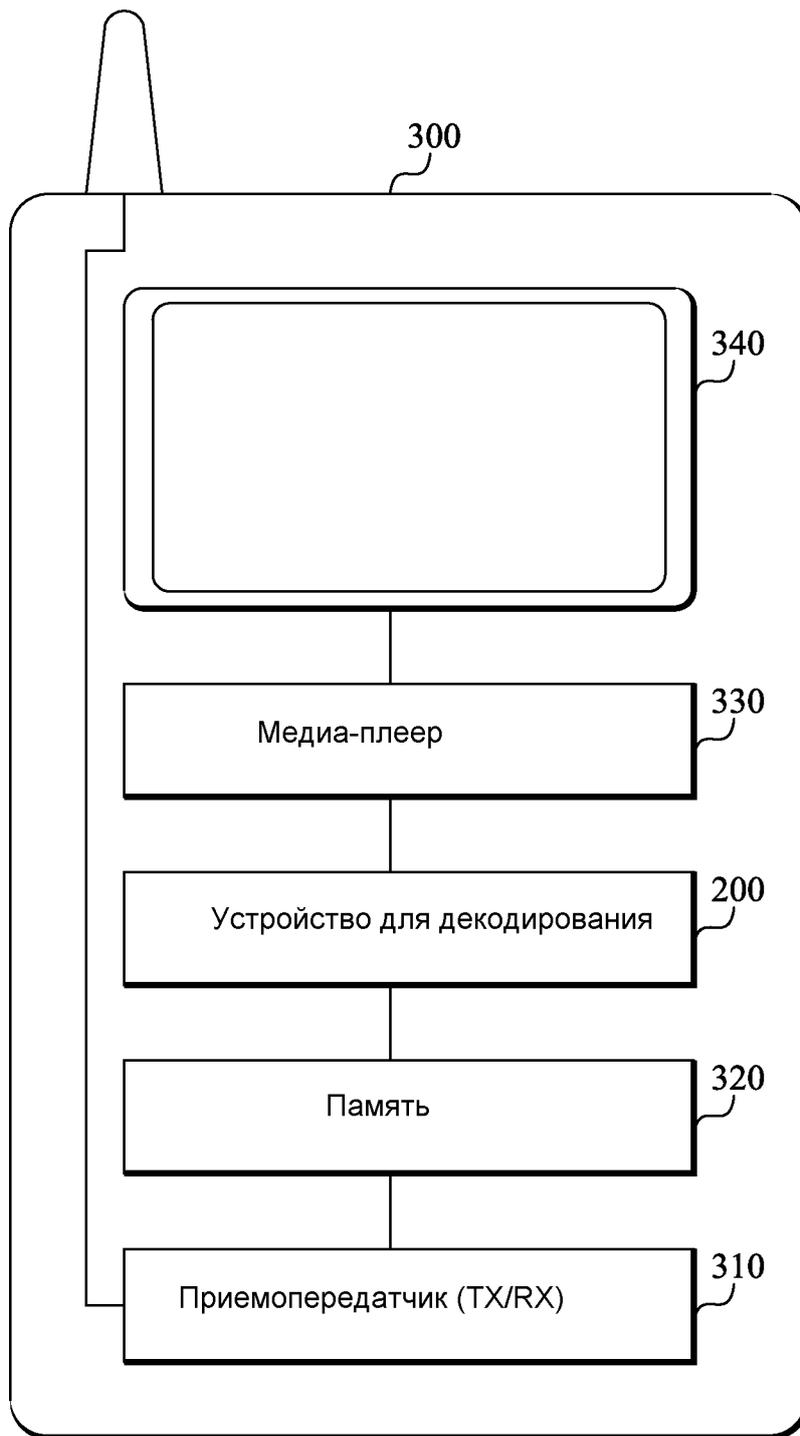
**ФИГ.9**



ФИГ.10



ФИГ.11



ФИГ.12