РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 596 596**(13) **C2**

(51) МПК
G10L 19/00 (2013.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2012122278/08, 19.10.2010**

(24) Дата начала отсчета срока действия патента:
19.10.2010

Приоритет(ы):
(30) Конвенционный приоритет:
20.10.2009 US 61/253,459

(43) Дата публикации заявки: **27.11.2013** Бюл. № 33

(45) Опубликовано: **10.09.2016** Бюл. № **25**

(56) Список документов, цитированных в отчете о поиске: **US 6449596 B1, 10.09.2002. RU 2185024 C2, 10.07.2002. RU 2197776 C2, 27.01.2003. US 2009/0074052 A1, 19.03.2009. WO 2004/028142 A2, 01.04.2004.**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **17.05.2012**

(86) Заявка PCT:
EP 2010/065726 (19.10.2010)

(87) Публикация заявки PCT:
WO 2011/048099 (28.04.2011)

Адрес для переписки:
410000, г. Саратов, главпочтамт, а/я 62, ООО "ПатентВолгаСервис", Романовой Н.В.

(72) Автор(ы):
ФУШ Гильом (DE),
СУББАРАМАН Вигнеш (DE),
РЕТТЕЛЬБАХ Николаус (DE),
МУЛТРУС Маркус (DE),
ГАЙЕР Марк (DE),
ВАМРБОЛД Патрик (DE),
ГРИЕБЕЛ Кристиан (DE),
ВЕИСС Оливер (DE)

(73) Патентообладатель(и):
Фраунхофер-Гезелльшафт цур Фёрдерунг дер ангевандтен Форшунг Е.Ф. (DE)

RU 2596596 C2

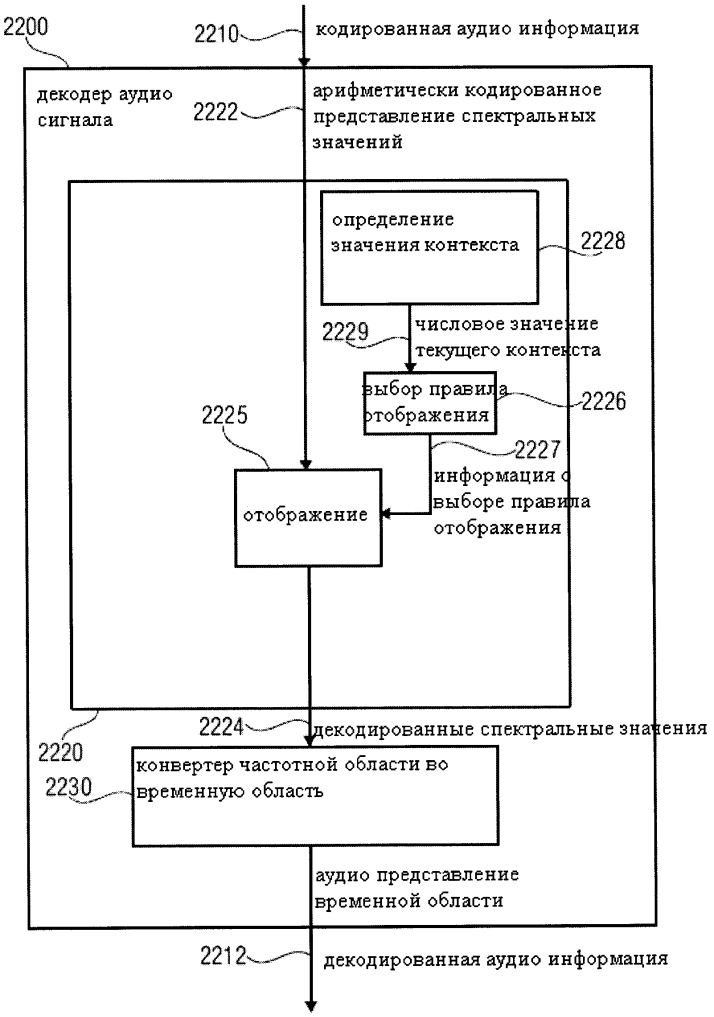
(54) АУДИО КОДЕР, АУДИО ДЕКОДЕР, СПОСОБ КОДИРОВАНИЯ АУДИО ИНФОРМАЦИИ, СПОСОБ ДЕКОДИРОВАНИЯ АУДИО ИНФОРМАЦИИ И КОМПЬЮТЕРНАЯ ПРОГРАММА, ИСПОЛЬЗУЮЩАЯ ЗАВИСИМОЕ ОТ ДИАПАЗОНА АРИФМЕТИЧЕСКОЕ КОДИРУЮЩЕЕ ПРАВИЛО ОТОБРАЖЕНИЯ

(57) Реферат:

Изобретение относится к аудио кодированию. Технический результат заключается в повышении эффективности кодирования при незначительных вычислительных затратах. Аудио декодер включает арифметический декодер для обеспечения множества декодированных спектральных значений на основе арифметически кодированного представления спектральных значений; и конвертер частотной области во временную область; при этом арифметический декодер настроен, чтобы выбрать правило отображения, описывающее отображение значения кода арифметически кодированного отображения на код символа, отображающего одно или более декодированное спектральное значение или как минимум часть одного или более декодированного спектрального значения, в зависимости от состояния контекста; при этом арифметический декодер настроен определять числовое значение текущего контекста, описывающее состояние текущего контекста в зависимости от множества ранее декодированных

RU 2596596 C2

спектральных значений, а также в зависимости от того, где расположено декодируемое спектральное значение - в первой заданном частотной области или во второй заданной частотной области. 6 н. и 11 з.п. ф-лы, 48 ил.



Фиг. 22

RU 2596596 C2

RUSSIAN FEDERATION

(19) **RU** (11) **2 596 596**(13) **C2**

(51) Int. Cl.
G10L 19/00 (2013.01)

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2012122278/08, 19.10.2010**

(24) Effective date for property rights:
19.10.2010

Priority:
(30) Convention priority:
20.10.2009 US 61/253,459

(43) Application published: **27.11.2013** Bull. № **33**

(45) Date of publication: **10.09.2016** Bull. № **25**

(85) Commencement of national phase: **17.05.2012**

(86) PCT application:
EP 2010/065726 (19.10.2010)

(87) PCT publication:
WO 2011/048099 (28.04.2011)

Mail address:
410000, g. Saratov, glavpochtamt, a/ja 62, OOO "PatentVolgaServis", Romanovoj N.V.

(72) Inventor(s):
FUSH Gilom (DE),
SUBBARAMAN Vignesh (DE),
RETTELBAKH Nikolaus (DE),
MULTRUS Markus (DE),
GAJER Mark (DE),
VAMRBOLD Patrik (DE),
GRIEBEL Kristian (DE),
VEISS Oliver (DE)

(73) Proprietor(s):
Fraunkhofer-Gezellshaft tsur Ferderung der angevandten Forshung E.F. (DE)

RU 2596596 C2

(54) AUDIO ENCODER, AUDIO DECODER, METHOD OF ENCODING AUDIO INFORMATION, METHOD OF DECODING AUDIO INFORMATION AND COMPUTER PROGRAM USING RANGE-DEPENDENT ARITHMETIC ENCODING MAPPING RULE

(57) Abstract:

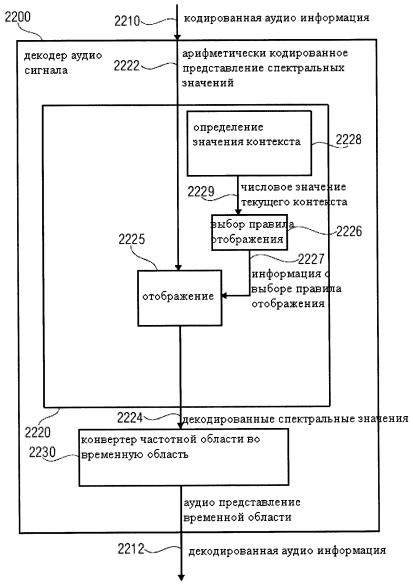
FIELD: acoustics.

SUBSTANCE: invention relates to audio encoding. Audio decoder includes an arithmetic decoder to provide a plurality of decoded spectral values based on an arithmetically encoded representation of spectral values; and frequency-domain converter to time domain; arithmetic decoder is configured to select a mapping rule describing mapping of a code value arithmetically encoded display on a symbol code which displays one or more decoded spectral value or at least part of one or more of decoded spectral values, depending on state of context; arithmetic decoder is configured to determine numeric current context value describing state of current context depending on a plurality of previously decoded spectral values, as well as depending on where there is a spectral value to decode-in first preset frequency domain or in second preset frequency domain.

EFFECT: technical result consists in improvement of efficiency of encoding at low computational costs.

17 cl, 48 dwg

RU 2596596 C2



Фиг. 22

RU 2596596 C2

RU 2596596 C2

Область техники

Воплощения в соответствии с изобретением связаны с аудио декодером для обеспечения декодированной аудио информации на основе кодированной аудио информации, аудио кодером для обеспечения кодированной аудио информации на основе входной аудио информации, способ для получения декодированной аудио информации на основе кодированной аудио информации, способ получения кодированной аудио информации на основе входной аудио информации и компьютерной программой.

Воплощения в соответствии с изобретением связаны с улучшенным спектральным бесшумным кодированием, которое может быть использовано в аудио кодере и декодере, как, например, так называемом единый кодере речи и аудио (USAC).

Предпосылки создания изобретения

Далее будет кратко описана концепция изобретения в целях облегчения понимания настоящего изобретения и его преимуществ. За последние десять лет большие усилия были предприняты для создания возможности для цифрового хранения и распространения аудио содержания с хорошей эффективностью битрейта. Одним из важных достижений на этом пути является определение международного стандарта ISO/IEC 14496-3. Часть 3 данного стандарта связана с кодированием и декодированием аудио содержимого, а подраздел 4 части 3 связан с общим аудио кодированием. ISO/IEC 14496, часть 3, раздел 4 определяет концепцию кодирования и декодирования общего аудио содержания. Кроме того, дальнейшие улучшения были предложены с целью улучшения качества и/или снижения необходимой скорости передачи данных.

Согласно концепции, описанной в указанном стандарте, во временной области звуковой сигнал преобразуется в частотно-временное представление. Преобразование из временной области в частотно-временную область, как правило, осуществляется с помощью блоков преобразования, который обозначаются как "кадры" из образцов временной области. Было установлено, что выгоднее использовать перекрывающиеся кадры, которые перемещаются, например, на половину кадра, так как перекрытие позволяет эффективно избежать (или хотя бы уменьшить) артефакты. Кроме того, было обнаружено, что оконная работа должна быть выполнена для того, чтобы избежать артефактов, происходящих из этой обработки временно ограниченных кадров.

При преобразовании оконной части входного звукового сигнала из временной области в частотно-временную область, уплотнение энергии получается во многих случаях, так что некоторые спектральные значения составляют значительно большую величину, чем множество других спектральных значений. Соответственно, во многих случаях есть сравнительно небольшое число спектральных значений с величиной, которая существенно выше средней величины спектральных значений. Типичным примером преобразования из временной области в частотно-временную область, приводящего к уплотнению энергии, является так называемое модифицированное дискретное косинус преобразование (MDCT).

Спектральные значения часто масштабируются и квантуются в соответствии с психоакустической моделью, так что ошибки квантования сравнительно меньше для психоакустичеки важных спектральных значений и сравнительно больше для психоакустически менее важных спектральных значений. Масштабированные и квантованные спектральные значения кодируются в целях обеспечения эффективного битрейта их представления.

Например, использование так называемого Huffman кодирования квантованных спектральных коэффициентов описано в международном стандарте ISO/IEC 14496-3:

2005 (E), часть 3, раздел 4.

Тем не менее, было установлено, что качество кодирования спектральных значений оказывает значительное влияние на требуемый битрейт. Кроме того, было установлено, что сложность аудио декодирования, которое часто осуществляется в портативных устройствах потребителей, и которое поэтому должно быть дешевыми и потреблять мало энергии, зависит от кодирования, используемого для кодирования спектральных значений.

В связи с этой ситуацией, есть необходимость в концепции кодирования и декодирования аудио содержания, которая предусматривает улучшение компромисса между битрейт эффективностью и эффективностью использования ресурсов.

Сущность изобретения

Примером воплощения изобретения является аудио декодер для получения декодированной аудио информации на основе кодированной аудио информации. Аудио декодер включает в себя арифметический декодер для предоставления множества декодированных спектральных значений на основе арифметически-кодированного представления спектральных значений. Аудио декодер также включает конвертер из частотной области во временную область для обеспечения во временной области аудио представления с помощью декодированных спектральных значений в целях получения декодированной аудио информации. Арифметический декодер предназначен для выбора правила отображения, описывающего отображение значения кода (который может быть извлечен из битового потока, представляющего кодированную аудио информацию) в код символа (который может быть числовым значением, представляющим декодированное спектральное значение или его наиболее значимую битовую плоскость) в зависимости от состояния контекста. Арифметический декодер настроен определять числовое значение текущего контекста, описывающее текущее состояние контекста в зависимости от множества ранее декодированных спектральных значений, а также в зависимости от того, где находится декодируемое спектральное значение - в первой заданной частотной области или во второй заданной частотной области.

Обнаружено, что если учитывать частотную область, в которой находится декодируемое частотное значение, то это позволит значительно улучшить качество вычисления контекста без значительного увеличения объема вычислений, необходимых для вычисления контекста. Кроме этого, принимая во внимание тот факт, что статистические зависимости между ранее декодированными спектральными значениями и расположенным рядом декодируемым спектральным значением изменяются в зависимости от частоты, можно выбрать контекст, который обеспечит высокую эффективность кодирования как для декодирования спектральных значений, соответствующих относительно низким частотам, так и для декодирования спектральных значений, соответствующих относительно высоким частотам. Хорошая адаптация контекста к деталям статистических зависимостей между декодируемым спектральным значением и ранее декодированными спектральными значениями (обычно не находящиеся в непосредственной или косвенной близости от спектрального значения, декодируемого в настоящий момент) позволяет увеличить эффективность кодирования, оставляя при этом вычислительные затраты небольшими. Обнаружено, что учет частотной области возможен при незначительных затратах, т.к. индекс частоты декодируемого спектрального значения, разумеется, определяется в процессе арифметического декодирования. Таким образом, выборочная адаптация контекста может производится при незначительных вычислительных затратах и при этом повышать эффективность кодирования.

В предпочтительном варианте арифметический декодер настроен выборочно модифицировать числовое значение текущего контекста в зависимости от того, где находится декодируемое спектральное значение - в первой заданной частотной области или во второй заданной частотной области. Выборочная модификация числового значения текущего контекста совместно с вычислением (или иным определением) числового значения текущего контекста позволяет комбинировать «обычное» вычисление (или иное определение) числового значения текущего контекста с учетом частотной области, в которой находятся спектральные значения, декодируемые в данный момент.. «Обычное» вычисление числового значения текущего контекста может быть проведено отдельно от адаптации числового значения текущего контекста в зависимости от частотной области, которая обычно упрощает алгоритм и уменьшает вычислительные затраты. Используя данную концепцию, можно легко обновить системы, включающие «обычное» вычисление числового значения текущего контекста.

В предпочтительном варианте арифметический декодер настроен определять числовое значение текущего контекста таким образом, что числовое значение текущего контекста основано на комбинации множества ранее декодированных спектральных значений или на комбинации множества промежуточных значений, производных от множества ранее декодированных спектральных значений, и при этом числовое значение текущего контекста выборочно превышает значение, полученное на основе комбинации множества ранее декодированных спектральных значений или на основе комбинации множества промежуточных значений, производных от множества ранее декодированных спектральных значении, в зависимости от того, где находится декодируемое спектральное значение - в первой заданной частотной области или во второй заданной частотной области. Обнаружено, что выборочное увеличение числового значения текущего контекста в зависимости от частотной области, в которой находится спектральное значение, декодируемое в настоящий момент, позволяет эффективно оценить числовое значение текущего контекста, оставляя при этом вычислительные затраты незначительными.

В предпочтительном варианте арифметический декодер настроен разграничивать по меньшей мере первую частотную область и вторую частотную область для того, чтобы определить числовое значение текущего контекста, при этом первая частотная область включает по меньшей мере 15% спектральных значений, соответствующих данной временной области (например, фрейму или подфрейму) аудио контента, при этом первая частотная область является областью низких частот и включает соответствующее спектральное значение, имеющее самую низкую частоту (в пределах набора спектральных значений, соответствующих данной (текущей) временной области аудио контента. Было обнаружено, что можно достичь хорошей адаптации контекста, если просто рассматривать нижнюю часть спектра (включающую по меньшей мере 15% спектральных значений) как первую частотную область, т.к. статистические зависимости между спектральными значениями не содержат сильных колебаний в области низких частот. В связи с этим количество различных областей может быть небольшим, что поможет избежать использование большого количества различных правил отображения. Однако для некоторых вариантов реализации изобретения может быть достаточно, если первая частотная область включает по меньшей мере одно спектральное значение, по меньшей мере два спектральных значения или по меньшей мере три спектральных значения, хотя предпочтителен выбор более расширенной первой спектральной области.

В предпочтительном варианте арифметический декодер настроен разграничивать

по меньшей мере между первой частотной областью и второй частотной областью для того, чтобы определить числовое значение текущего контекста, при этом вторая частотная область включает по меньшей мере 15% спектральных значений, соответствующих заданной темпоральной области (например, фрейму, или подфрейму) аудио контента, при этом вторая частотная область является областью высоких частот и включает соответствующее спектральное значение, имеющее самую высокую частоту (в пределах набора спектральных значений, соответствующих данной (текущей) временной области аудио контента). Было обнаружено, что хорошей адаптации контекста можно достичь, если просто учитывать верхнюю часть спектра (включающую, по меньшей мере, 15% спектральных значений) как вторую частотную область, так как статистические зависимости между спектральными значениями не содержат сильных колебаний в области высоких частот. В связи с этим количество различных областей может быть небольшим, что в свою очередь поможет избежать использования большого количества различных правил отображения. Однако, для некоторых вариантов реализации изобретения может быть достаточно, если вторая частотная, область включает по меньшей мере одно спектральное значение, по меньшей мере два спектральных значений, или по меньшей мере три спектральных значений, хотя предпочтителен выбор более расширенной второй спектральной области.

В предпочтительном варианте арифметический декодер настроен разграничивать по меньшей мере, первую спектральную область, вторую спектральную область и третью спектральную область для того, чтобы определить числовое значение текущего контекста в зависимости от того, в какой из по меньшей мере трех частотных областей находится декодируемое спектральное значение. В этом случае первая, вторая и третья частотные области включают множества соответствующих спектральных значений. Было обнаружено, что для обычных аудио сигналов предпочтительно разграничивать по меньшей мере три различные частотные области, так как обычно существуют по меньшей мере три частотные области, в которых есть статистические зависимости между спектральными значениями. Рекомендуется (хотя это не обязательно) разграничивать три и более частотных областей даже для узкополосных аудио сигналов (например, для аудио сигналов, имеющих частотный диапазон от 300 Гц до 3 кГц). Для аудио сигналов, имеющих более высокий диапазон частот, предпочтительно (хотя не обязательно) разграничивать три или более расширенные частотные области (каждая их которых имеет более чем одно соответствующее ей спектральное значение).

В предпочтительном варианте по меньшей мере 1/8 спектральных значений (текущей) временной области аудио информации соответствует первой частотной области, по меньшей мере 1/5 спектральных значений (текущей) временной части аудио информации соответствует второй частотной области, и по меньшей мере V* спектральных значений (текущей) временной области аудио информации соответствует третьей частотной области. Рекомендуется использовать достаточно большие частотные области, поскольку достаточно большие частотные области позволяют найти компромисс между эффективным кодированием и сложностью вычисления. Также было обнаружено, что использование очень маленьких частотных областей (например, частотных областей, включающих только одно соответствующее спектральное значение) неэффективно с точки зрения вычисления и может даже ухудшить эффективность кодирования. Кроме этого, необходимо отметить, что выбор достаточно больших частотных областей (например, частотных областей, включающих по меньшей мере два соответствующих спектральных значения) предпочтителен, даже при использовании двух частотных областей.

В предпочтительном варианте арифметический декодер настроен вычислять сумму, включающую, по меньшей мере первое слагаемое и второе слагаемое, для того, чтобы получить числовое значение текущего контекста в результате суммирования. В этом случае первое слагаемое представляет собой комбинацию множества промежуточных значений, которые описывают величины ранее декодированных спектральных значений, и второе слагаемое показывает какой частотной области из множества частотных областей соответствует декодируемое (в настоящий момент) спектральное значение. При таком подходе можно разграничивать вычисление контекста на основе информации о величинах раннее декодируемых спектральных значений и адаптацию контекста в зависимости от области, которой соответствует декодируемое в настоящий момент спектральное значение. Обнаружено, что величины раннее декодированных спектральных значений являются важным индикатором окружения спектрального значения, декодируемого в настоящий момент. Определение статистических зависимостей, которые базируются на оценке величин раннее декодированных спектральных значений, может быть более эффективным, если принимать во внимание частотную область, которой соответствует декодируемое в настоящий момент спектральное значение. С точки зрения вычислений достаточно включать информацию о частотной области в числовое значение текущего контекста в качестве значения суммы, даже такой простой механизм улучшает числовое значение текущего контекста.

В предпочтительном варианте арифметический декодер настроен модифицировать одну или более заданных битовых позиций двоичного представления числового значения текущего контекста в зависимости от того, в какой частотной области из множества различных частотных областей находится декодируемое спектральное значение.

Использование выделенных битовых позиций для информации о частотной области облегчает выбор правила отображения в зависимости от числового значения текущего контекста. Например, при использовании заданной битовой позиции числового значения текущего контекста для описания частотной области, которой соответствует декодируемое спектральное значение, выбор правила отображения может быть упрощен. Например, обычно существует ряд контекстных ситуаций, в которых применяется одно и то же правило отображения. в присутствии определенного окружения (в плане спектральных значений) спектрального значения, декодируемого в настоящий момент независимо от частотной области, которой соответствует декодируемое в настоящий момент спектральное значение. В таких случаях информация относительно частотной области, которой соответствует декодируемое в настоящий момент значение, может остаться не учтенной, чему способствует использование заданной битовой позиции для кодирования информации. Однако в остальных случаях, то есть для различных комбинаций (в плане спектральных значений) окружения спектрального значения, декодируемого в настоящий момент, информация о частотной области, соответствующей декодируемым в настоящий момент спектральным значениям может использоваться при выборе правила отображения.

В предпочтительном варианте арифметический декодер настроен выбирать правило отображения в зависимости от числового значения текущего контекста таким образом, что для множества различных числовых значений текущего контекста в результате выбирается одно и то же правило отображения. Концепцию, согласно которой принимается во внимание частотная область, которой соответствует декодируемое в настоящий момент спектральное значение, можно сочетать с концепцией, согласно которой одно и то же правило отображения соответствует множеству различных числовых значений текущего контекста. Нет необходимости всегда учитывать частоту,

соответствующую декодируемому в настоящий момент спектральному значению, однако, необходимо, по меньшей мере в некоторых случаях, учитывать информацию о частотной области, соответствующую декодируемому в настоящий момент спектральному значению.

В предпочтительном варианте арифметический декодер настроен выполнять двухшаговый выбор правила отображения в зависимости от числового значения текущего контекста. В этом случае арифметический декодер настроен проверять, на первом шаге выбора, идентично ли числовое значение текущего контекста величине значимого состояния, описанного с помощью записи таблицы прямого попадания. Арифметический декодер настроен определять, на втором шаге выбора, который выполняется только в том случае, если числовое значение текущего контекста отличается от величин значимых состояний, описанных с помощью записей таблицей прямого попаданий, в каком из интервалов из интервалов, среди множества интервалов, находится числовое значение текущего контекста. В этом случае арифметический декодер настроен выбирать правило отображения в зависимости от результата первого шага выбора и/или второго шага выбора. Арифметический декодер также настроен выбирать правило выбора в зависимости от того в какой частотной области находится декодируемое спектральное значение - в первой или во второй частотной области. Обнаружено, что комбинация описанной выше концепции вычисления числового значения текущего контекста с двухшаговым выбором правила отображения имеет ряд преимуществ. Например, при использовании данной концепции возможно определение различных конфигураций контекста «прямого попадания», которым соответствует правило отображения на первом шаге выборе, для декодируемых спектральных значений, расположенных в различных частотных областях. Также второй шаг выбора, когда производится выбор правила отображения на основе интервала, подходит для обработки тех ситуаций, (окружения раннее декодированных спектральных значений), в которых не желателен (или, по меньшей мере в нем нет необходимости) учет частотной области, которой соответствует спектральное значение, декодируемое в настоящий момент.

В предпочтительном варианте арифметический декодер настроен выборочно модифицировать одну или более наименее значимых битовых позиций двоичного представления числового значения текущего контекста в зависимости от того в какой частотной области из множества различных частотных областей находится декодируемое спектральное значение. В этом случае арифметический декодер настроен определять в. процессе второго шага выбора в каком интервале из множества интервалов находится двоичное представление числового значения текущего контекста, для того чтобы выбрать отображение таким образом, что для нескольких числовых значений текущего контекста выбирается одно и то же правило отображение независимо от того, в какой частотной области находится декодируемое спектральное значение, а также таким образом, что для нескольких числовых значений текущего контекста правило отображения выбирается в зависимости от того в какой частотной области находится кодируемое спектральное значение. Механизм, согласно которому, частотная область кодируется в наименее значимых битах двоичного представления числового значения текущего контекста, вполне подходит для продуктивной совместной работы с двухшаговым выбором правила отображения.

Другой вариант использования изобретения приводит к созданию аудио кодера для получения кодированной аудио информации на основе входной аудио информации. Аудио кодер включает в себя энергоуплотняющий конвертер из временной области в

частотную для обеспечения в частотной области аудио представления на основе представления входной аудио информации во временной области, так что аудио представление в частотной области включает в себя набор спектральных значений. Арифметический кодер настроен на кодирование спектрального значения, или его предварительно обработанной версии с помощью кодового слова с переменной длиной. Арифметический кодер настроен для отображения спектрального значения, или значения наиболее значимого бита плоскости спектрального значения на значение кода (которое может быть включено в битовый поток, представляющий входную аудио информацию в кодированной форме). Арифметический кодер предназначен для выбора правила отображения, описывающего отображение спектрального значения или наиболее значимого бита плоскости спектрального значения на значение кода в зависимости от состояния контекста. Арифметический кодер предназначен, чтобы определять числовое значение текущего контекста, описывающее текущее состояние контекста в зависимости от множества ранее кодированных спектральных значений, а также в зависимости от того, где находится кодируемое спектральное значение - в первой заданной частотной области или во второй заданной частотной области.

Этот кодер аудио сигнала основан на тех же открытиях, как и декодер аудио сигнала, описанный выше. Было установлено, что механизм адаптации контекста, который показал свою эффективность для декодирования аудио содержания, следует также применять на стороне кодера для того, чтобы обеспечить последовательность системы.

Примером воплощения данного изобретения является создание способа для получения декодированной аудио информации на основе кодированной аудио информации.

Еще одним примером воплощения данного изобретения является создание способа для получения кодированной аудио информации на основе входной аудио информации.

Другой вариант воплощения изобретения содержит компьютерную программу для выполнения одного из указанных способов.

Эти способы и компьютерная программа основываются на тех же открытиях, как и вышеописанные аудио декодер и аудио кодер.

Краткое описание фигур

Использования изобретения будут далее описаны со ссылкой на прилагаемые фигуры, на которых:

Фиг.1 показывает блок-схему аудио кодера, согласно одному из вариантов использования изобретения;

Фиг.2 показывает блок-схему аудио декодера в соответствии с одним из вариантов использования изобретения;

Фиг.3 показывает представление кода псевдо-программы алгоритма "value_decode ()" для декодирования спектрального значения;

Фиг.4 показывает схематическое представление контекста для вычисления контекста;

Фиг.5a показывает представление кода псевдо-программы алгоритма "arith_map_context ()" для отображения контекста;

Фиг.5b и 5 c показывают представление кода псевдо-программы алгоритма "arith_get_context ()" для получения значения состояния контекста;

Фиг.5d показывает представление кода псевдо-программы алгоритма "get_pk(s)" для извлечения значения индекса сводной таблицы частот „pki" из переменной состояния;

Фиг.5e показывает представление кода псевдо-программы алгоритма "arith_get_pk (s)" для извлечения значения индекса сводной таблицы частот „pki" из значения состояния;

Фиг.5f показывает представление кода псевдо-программы алгоритма "get_pk(unsigned

long s)" для извлечения значения индекса сводной таблицы частот „pki" из значения состояния;

Фиг.5g показывает представление кода псевдо-программы алгоритма "arith_decode ()" для арифметического декодирования символа из кодового слова переменной длины;

Фиг.5h показывает представление кода псевдо-программы алгоритма "arith_update_context ()" для обновления контекста;

Фиг.5i показывает легенду определений и переменных;

Фиг.6a показывает синтаксис представления необработанного блока единого кодирования речи и аудио (USAC);

Фиг.6b показывает синтаксис представления единого элемента канала;

Фиг.6c показывает синтаксис представления парного элемента канала;

Фиг.6d показывает синтаксис представления "ics" контрольной информации;

Фиг.6e показывает синтаксис представления потока канала частотной области;

Фиг.6f показывает синтаксис представления арифметически кодированных спектральных данных;

Фиг.6g показывает синтаксис представление для декодирования множества спектральных значений;

Фиг.6h показывает легенду элементов данных и переменных;

Фиг.7 показывает блок-схему аудио кодера, согласно другому варианту осуществления изобретения;

Фиг.8 показывает блок-схему аудио декодера в соответствии с другим вариантом использования изобретения;

Фиг.9 показывает организацию сравнения бесшумного кодирования в соответствии с рабочим проектом 3 проекта стандарта USAC с схемой кодирования в соответствии с настоящим изобретением:

Фиг.10a показывает схематическое представление контекста расчета состояния, так как оно используется в соответствии с рабочим проектом 4 проекта стандарта US AC;

Фиг.10b показывает схематическое представление контекста расчета состояния, так как оно используется воплощениях в соответствии с изобретением;

Фиг.11a показывает обзор таблицы, используемой в схеме арифметического кодирования в соответствии с рабочим проектом 4 проекта стандарта USAC;

Фиг.11b показывает обзор таблицы, используемой в схеме арифметического кодирования в соответствии с изобретением;

Фиг.12a показывает графическое представление запроса памяти только для чтения на схемы бесшумного кодирования в соответствии с настоящим изобретением и в соответствии с рабочим проектом 4 проекта стандарта USAC;

Фиг.12b показывает графическое представление общего запроса данных памяти только для чтения декодера USAC в соответствии с настоящим изобретением и в соответствии с рабочим проектом 4 проекта стандарта USAC;

Фиг.13a показывает таблицу представления средних битрейтов, которые используются кодером единого кодирования речи и аудио, с помощью арифметического кодера в соответствии с рабочим проектом 3 проекта стандарта USAC и арифметическим декодером в соответствии с вариантом осуществления настоящего изобретения;

Фиг.13b показывает таблицу представления контроля резервуара бит для кодера единого кодирования речи и аудио с помощью арифметического кодера в соответствии с рабочим проектом 3 проекта стандарта USAC и арифметического кодера в соответствии с вариантом осуществления настоящего изобретения;

Фиг.14 показывает таблицу представления средних битрейтов USAC кодера в

соответствии с рабочим проектом 3 проекта стандарта USAC и в соответствии с вариантом осуществления настоящего изобретения;

Фиг.15 показывает таблицу представления минимального, максимального и среднего битрейта USAC на основе кадра;

Фиг.16 показывает таблицу представления лучшего и худшего случаев на основе кадра;

Фиг.17(1) и 17(2) показывают таблицу представления содержания таблицы "ari_s_hash [387]";

Фиг.18 показывает таблицу представления содержания таблицы "ari_gs_hash[225]";

Фиг.19 (1) и 19 (2) показывают таблицу представления содержания таблицы "ari_cf_m [64][9]"; и

Фиг.20 (1) и 20 (2) показывают таблицу представления содержания таблицы "ari_s_hash [387]".

Фиг.21 показывает блок-схему аудио кодера в соответствии с_ вариантом использования изобретения; и

Фиг.22 показывает блок-схему аудио декодера в соответствии с вариантом использования изобретения.

Подробное описание вариантов использования изобретения

1. Аудио кодер в соответствии с фиг.7

Фиг.7 показывает блок-схему аудио кодера, согласно одному из вариантов использования изобретения; Аудио декодер 700 настроен на получение входной аудио информации 710 и на представлении на ее основе кодированной аудио информации 712. Аудио кодер включает в себя энергоуплотняющий конвертер из временной области в частотную 720, который предназначен для обеспечения в частотной области аудио представления 722 на основе представления входной аудио информации 710 во временной области, так что аудио представление в частотной области 722 включает в себя набор спектральных значений. Аудио кодер 700 также включает в себя арифметический кодер 730, предназначенный для кодирования спектрального значения (из множества спектральных значений, формирующих в частотной области аудио представление 722), или его предварительно обработанной версии с помощью кодового слова переменной длиной, чтобы получить кодированную аудио информацию 712 (которая может включать, например, множество кодовых слов переменной длины).

Арифметический кодер 730 настроен на отображение спектрального значения или значения наиболее значимого бита плоскости спектрального значения на значение кода (т.е. на кодовое слово переменной длины) в зависимости от состояния контекста. Арифметический кодер 730 предназначен для выбора правила отображения, описывающего отображение спектрального значения или наиболее значимого бита плоскости спектрального значения на значение кода в зависимости от состояния контекста. Арифметический кодер предназначен, чтобы определять текущее состояние контекста в зависимости от множества ранее кодированных смежных спектральных значений. Для этого арифметический кодер настроен на обнаружение группы из множества ранее кодированных смежных спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины, а также для определения текущего состояния контекста в зависимости от результата обнаружения.

Как можно видеть, отображение спектрального значения или наиболее значимого бита плоскости спектрального значения на значение кода может осуществляться кодированием спектрального значения 740 с помощью отображения 742. Трекер

состояния 750 может быть сконфигурирован для отслеживания состояния контекста и может включать в себя детектор группы 752 для обнаружения группы из множества ранее кодированных смежных спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины. Трекер состояния 750 также желательно настроить для определения текущего состояния контекста в зависимости от результата этого обнаружения, выполненного детектором группы 752. Таким образом, трекер состояния 750 обеспечивает информацию 754, описывающую текущее состояние контекста. Селектор правила отображения 760 может выбрать правило отображения, например, сводную таблицу частот, описывающую отображение спектрального значения, или наиболее значимого бита плоскости спектрального значения, на значение кода. Соответственно, селектор правила отображения 760 предоставляет информацию правила отображения 742 для спектрального кодирования 740.

Подводя итог вышесказанному, аудио кодер 700 выполняет арифметическое кодирование в частотной области аудио представления, осуществляемого конвертером из временной области в частотную. Арифметическое кодирование зависит от контекста, например, правило отображения (например, сводная таблица частот) выбирается в зависимости от ранее кодированных спектральных значений. Таким образом, спектральные значения, смежные во времени и/или частоте (или, по крайней мере, в заданном окружении) друг с другом и/или с в данный момент кодируемым спектральным значением (т.е. спектральные значения в заданном окружении в данный момент кодируемого спектрального значения) рассматриваются в арифметическом кодировании для регулировки распределения вероятности, оцениваемой арифметическим кодированием. При выборе соответствующего правила отображения, обнаружения проводится с целью выявления, есть ли группа из множества ранее кодированных смежных спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины. Результат этого обнаружения применяется при выборе текущего состояния контекста, т.е. при выборе правила отображения. Определив, существует ли группа из множества спектральных значений, которые являются особенно малыми или особенно большими, можно распознать особенности в частотной области аудио представления, которое может быть частотно-временным представлением. Особые черты, такие как, например, группа из множества особенно малых или особенно больших спектральных значений, показывают, что особое состояние контекста следует использовать, поскольку это особое состояние контекста может дать особенно хорошую эффективность кодирования. Таким образом, выявление группы смежных спектральных значений, которые удовлетворяют заданному условию, что обычно используется в сочетании с альтернативной оценкой контекста, основанной на сочетании множества ранее кодированных спектральных значений, представляет собой механизм, который позволяет эффективно выбирать соответствующий контекст, если входная аудио информация требует некоторых особых состояний (например, содержит большой маскированный диапазон частот).

Соответственно, эффективное кодирование может быть достигнуто при сохранении расчета контекста достаточно простым.

2. Аудио декодер в соответствии с фиг.8

Фиг.8 показывает блок-схему аудио декодера 800. Аудио декодер 800 настроен на получение кодированной аудио информации 810 и на представлении на ее основе декодированной аудио информации 812. Аудио декодер 800 включает в себя арифметический декодер 820, который предназначен для предоставления множества

декодированных спектральных значений 822 на основе арифметически-кодированного представления 821 спектральных значений. Аудио декодер 800 также включает конвертер из частотной области во временную область 830, который предназначен для получения декодированных спектральных значений 822 и предоставления во временной области аудио представления 812, которое может включать декодированную аудио информацию, с помощью декодированных спектральных значений 822, для получения декодированной аудио информации 812.

Арифметический декодер 820 включает в себя определитель спектрального значения 824, настроенный на отображения значения кода арифметически кодированного представления 821 спектральных значений на код символа, представляющий одно или несколько декодированных спектральных значений, или, по крайней мере, часть (например, наиболее значимые биты плоскости) одного или нескольких декодированных спектральных значений. Определитель спектрального значения 824 может быть настроен для выполнения отображения в зависимости от правила отображения, которое может быть описано в информации правила отображения 828a.

Арифметический декодер 820 настроен на выбор правила отображения (например, сводной таблицы частот), описывающего отображение значения кода (описываемого в арифметически кодированном представлении 821 спектральных значений) на код символа (описывающий одно или несколько спектральных значений) в зависимости от состояния контекста (которое может быть описано в информациии состояния контекста 826a). Арифметический декодер 820 настроен, чтобы определить текущее состояние контекста в зависимости от множества ранее декодированных спектральных значений 822. Для этого трекер состояния 826 может быть использован, который получает информацию с описанием ранее декодированных спектральных значений. Арифметический декодер также настроен на обнаружение группы из множества ранее декодированных смежных спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины, а также для определения текущего состояния контекста (описанного, например, в информации состояния контекста 826a) в зависимости от результата обнаружения.

Обнаружение группы из множества ранее декодированных смежных спектральных значений, которые соответствуют заданному условию относительно их величины, может, например, проводиться детектором группы, который является частью трекера состояния 826. Таким образом, получается информация текущего состояния контекста 826a. Выбор правила отображения может выполняться селектором правила отображения 828, который извлекается из информации правила отображения 828a из информации текущего состояния контекста 826a, и который обеспечивает информацию правила отображения 828a для определителя спектрального значения 824.

Что касается функциональных возможностей декодера аудио сигнала 800, следует отметить, что арифметический декодер 820 настроен на выбор правила отображения (например, сводную таблицу частот), которое, в среднем, хорошо адаптировано к спектральному значению для декодирования, так как правило отображения выбирается в зависимости от текущего состояния контекста, что в свою очередь, определяется в зависимости от множества ранее декодированных спектральных значений. Таким образом, статистические зависимости между смежными спектральными значениями для декодирования могут быть использованы. Более того, обнаружив группу из множества ранее декодированных смежных спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины, можно адаптировать правило отображения к особым условиям (или

моделям) ранее декодированных спектральных значений. Например, особое правило отображения может быть выбрано, если группа из множества сравнительно небольших ранее декодированных смежных спектральных значений идентифицирована, или если группа из множества сравнительно больших ранее декодированных смежных спектральных значений идентифицирована. Было обнаружено, что присутствие группы сравнительно больших спектральных значений или группы сравнительно небольших спектральных значений можно рассматривать как существенный признак того, что выделенное правило отображения, специально адаптированное для такого состояния, должно быть использовано. Таким образом, вычислению контекста может способствовать (или ускорять) использование обнаружения такой группы из множества спектральных значений. Кроме того, те характеристики аудио содержания можно рассматривать, которые нельзя рассматривать так же легко без применения вышеупомянутой концепции. Например, обнаружение группы множества спектральных значений, которые соответствуют, по отдельности или вместе взятые, заданному условию относительно их величины,, может быть выполнено на основе различных наборов спектральных значений, по сравнению с набором спектральных значения, используемых для вычисления нормального контекста.

Дальнейшие подробности будут описаны ниже.

3. Аудио кодер в соответствии с фиг.1

Далее будет описан аудио кодер в соответствии с вариантом осуществления настоящего изобретения. Фиг.1 показывает блок-схему такого аудио кодера 100.

Аудио кодер 100 настроен на получение входной аудио информации ПО и на предоставлении на ее основе битового потока 112, который представляет собой кодированную аудио информацию. Аудио декодер 100 может дополнительно включать препроцессор 120, который настроен на получение входной аудио информации ПО и предоставление на ее основе предварительно обработанную входную аудио информацию 110a. на фиг.Аудио кодер 100 также включает в себя энергоуплотняющий трансформер сигнала из временной области в частотную 130, который также обозначается как конвертер сигнала. Ковертер сигнала 130 настроен на получение входной аудио информации ПО, 110a и предоставление на ее основе аудио информации 132 в частотной области, которая предпочтительно имеет вид набора спектральных значений. Например, трансформер сигнала 130 может быть сконфигурирован для получения кадра входной аудио информации ПО, 110a (например, блок образцов временной области) и для предоставления набора спектральных значений, представляющих аудио содержание соответствующего аудио кадра. Кроме того, трансформер сигнала 130 может быть настроен на получение множества последующих, перекрывающихся или неперекрывающихся, аудио кадров входной аудио информации ПО, 110a и предоставления на ее основе аудио представления во временной и частотной области, которое состоит из последовательности последующих наборов спектральных значений, один набор спектральных значений связан с каждым кадром.

Энергоуплотняющий трансформер сигнала из временной области в частотную 130 может включать в себя энергоуплотняющий банк фильтров, который обеспечивает спектральные значения, связанные с различными, перекрывающимися или неперекрывающимися, частотными диапазонами. Например, трансформер сигнала 130 может включать в себя оконный MDCT трансформер 130a, который настроен на оконную работу с входной аудио информацией ПО, 110a (или его кадр) с помощью окна преобразования и выполнения модифицированного дискретного косинус-преобразования оконной входной аудио информации 110, 110a (или оконный кадр).

Таким образом, аудио представление в частотной области 132 может включать в себя набор, например, 1024 спектральных значений в виде MDCT коэффициентов, связанных с кадром входной аудио информации. !

Аудио декодер 100 может дополнительно включать спектральный постпроцессор 140, который настроен на получение аудио представления в частотной области 132 и предоставление на ее основе пост обработанное аудио представление в частотной области 142. Спектральный постпроцессор 140 может, например, быть настроен на выполнение временного ограничения шума и/или долгосрочного прогноза и/или любой другой спектральной пост-обработки, известной в данной области. Аудио кодер дополнительно содержит, по желанию, скейлер / квантователь 150, который настроен на получение в частотной области аудио представления 132 или ее версию пост-обработки 142 и для обеспечения масштабированного и квантованного аудио представления в частотной области 152.

Аудио кодер 100 дополнительно содержит, по желанию, психоакустическую модель процессора 160, который настроен на получение входной аудио информации ПО (или пост-обработанной версии 110a) и для представления на ее основе дополнительной контрольной информации, которая может быть использована для управления энергоуплотняющим трансформером сигнала из временной области в частотную 130 для управления дополнительным спектральным пост-процессором 140 и/или для контроля за дополнительным скейлером/квантователем 150. Например, психоакустическая модель процессора 160 может быть сконфигурирована для анализа входной аудио информации, чтобы определить, какие компоненты входной аудио информации 110, 110a особенно важны для человеческого восприятия аудио содержания и какие компоненты входной аудио информации 110, 110a менее важны для восприятия аудио содержания. Таким образом, психоакустическая модель процессора 160 может обеспечить контрольную информации, которая используется аудио кодером 100 для регулировки масштабирования аудио представления в частотной области 132, 142 скейлером/квантователем 150 и/или разрешением квантования, применяемом скейлером/квантователем 150. Следовательно, важные для восприятия группы масштабных коэффициентов (т.е. группы смежных спектральных значений, которые являются особенно важными для человеческого восприятия аудио содержания) масштабируется с большим коэффициентом масштабирования и квантуются со сравнительно высоким разрешением, в то время как менее важные для восприятия группы масштабных коэффициентов (т.е. группы смежных спектральных значений) масштабируются со сравнительно меньшим коэффициентом масштабирования и квантуются со сравнительно низким разрешением квантования. Таким образом, масштабированные спектральные значения частот более важных для восприятия, как правило, значительно больше, чем спектральные значения частот менее важных для восприятия.;

Аудио кодер также включает в себя арифметический кодер 170, который настроен на получение масштабированой и квантованной версии 152 аудио представления в частотной области 132 (или, наоборот, пост-обработанной версии 142 аудио представления в частотной области 132, или даже само аудио представление в частотной области 132), а также для обеспечения арифметической информации кодового слова 172a на ее основе, например, так что арифметическая информация кодового слова представляет аудио представление в частотной области 152.

Аудио кодер 100 также включает в себя форматтер полезной нагрузки битового потока 190, который настроен на получение арифметической информации кодового слова 172a. Форматтер полезной нагрузки битового потока 190 также обычно настроен

на получение дополнительной информации, как, например, информации коэффициента масштабирования, описывающей какие коэффициенты масштабирования были применены скейлером/квантователем 150. Кроме того, форматтер полезной нагрузки битового потока 190 может быть настроен на получение другой управляющей информации. Форматтер полезной нагрузки битового потока 190 настроен на обеспечение битового потока 112 на основе полученной информации путем сборки битового потока в соответствии с желаемым синтаксисом потока, который будет обсуждаться ниже.

Далее будут описаны подробности, касающиеся арифметического кодера 170. Арифметический кодер 170 настроен на получение множества пост-обработанных и масштабированных и квантованных спектральных значений аудио представления в частотной области 132. Арифметический кодер включает в себя экстрактор наиболее значимых битов плоскости 174, который настроена на извлечение наиболее значимых бит плоскости m спектрального значения. Следует отметить, что наиболее значимый бит плоскости может содержать один или более битов (например, два или три бита), которые являются наиболее значимыми битами спектрального значения. Таким образом, экстрактор наиболее значимых битов плоскости 174 обеспечивает значение наиболее значимого бита плоскости 176 спектрального значения.

Арифметический кодер 170 также включает в себя определитель первого кодового слова 180, который настроен, чтобы определить арифметическое кодовое слово acod_m [pki][m], представляющее значение наиболее значимого бита плоскости значение m. По желанию, определитель кодового слова 180 может также предоставить одно или большее количество управляющих кодовых слов (также обозначенные здесь с "ARITHESCAPE") с указанием, например, как много менее значимых бит плоскости доступны (и, следовательно, с указанием числового веса наиболее значимого бита плоскости). Определитель первого кодового слова 180 может быть сконфигурирован для обеспечения кодового слова, связанного с значением наиболее значимого бита плоскости m с помощью выбранной сводной таблицы частоты, имеющей (или которая ссылается на) индекс сводной таблицы частоты pki.

Для того чтобы определить, какую сводную таблицу частот надо выбрать, арифметический кодер предпочтительно включает в себя трекер состояния 182, который настроен на отслеживание состояния арифметического кодера, например, с помощью наблюдения за тем, какие спектральные значения были кодированы ранее. Трекер состояния 182, следовательно, дает информацию о состоянии 184, например, значение состояния обозначается "s" или "t". Арифметический кодер 170 также включает селектор сводной таблицы частот 186, который настроен на получение информации о состоянии 184 и предоставление информации 188, описывающей выбранную сводную таблицу частот для определителя кодового слова 180. Например, селектор сводной таблицы частот 186 может дать индекс сводной таблицы частот „pki", описывающий какая сводная таблица частот из набора из 64 сводных таблиц частот выбрана для использования определителем кодового слова. Кроме того, селектор сводной таблицы частот 186 может обеспечить всю выбранную сводную таблицу частот для определителя кодового слова. Таким образом, определитель кодового слова 180 может использовать выбранную сводную таблицу частот для предоставления кодового слова acod_m[pki] [m] значения наиболее значимого бита плоскости т, так что фактическое кодовое слово acod_m[pki][m] кодирования значения наиболее значимого бита плоскости m зависит от значения m и индекса сводной таблицы частот pki, и, следовательно, от информации текущего состояния 184. Более подробная информация о процессе кодирования и

формате полученного кодового слова будет описана ниже.

Арифметический кодер 170 также включает в себя экстрактор наименее значимых битов плоскости 189a, который настроен на извлечение одного или более менее значимых бит плоскости из масштабированного и квантованного аудио представления в частотной области 152, если один или несколько спектральных значений для кодирования превышают диапазон кодируемых значений с помощью тольког самых значимых бит плоскости. Менее значимые биты плоскости могут включать один или несколько битов, по желанию. Соответственно, экстрактор наименее значимых битов плоскости 189a предоставляет информацию менее значимых бит плоскости 189b. Арифметический кодер 170 также включает в себя определитель второго кодового слова 189 c, который настроен на получение информации менее значимых бит плоскости 189d и предоставления не ее основе 0, 1 или более кодовых слов "acod_r", представляющих содержание 0, 1 или больше менее значимых бит плоскости. Определитель второго кодового слова 189 c может быть настроен на применение алгоритма арифметического кодирования или любой другой алгоритм кодирования для того, чтобы извлечь кодовые слова менее значимых бит плоскости "acod_r" из информации менее значимых бит плоскости 189b.

Следует отметить, что ряд менее значимых бит плоскости могут варьироваться в зависимости от значения масштабированных и квантованных спектральных значений 152, так что может не быть менее значимых бит плоскости вообще, если масштабированное и квантованное спектральное значение, которое будут кодировано, сравнительно невелико, например, может быть один менее значимый бит плоскости, если текущее масштабированное и квантованное спектральное значение для кодирования имеет средний диапазон и так, что может быть более одного менее значимых бит плоскости, если масштабированное и квантованное спектральное значение для кодирования имеет сравнительно большое значение.

Подводя итог вышесказанному, арифметический кодер 170 настроен на кодирование масштабированных и квантованных спектральных значений, которые описаны в информации 152 с помощью иерархического процесса кодирования. Наиболее значимый бит плоскости (включая, например, один, два или три бита на спектральное значение) кодируется для получения арифметического кодового слова "acod_m[pki][m]" значения наиболее значимого бита плоскости. Один или несколько менее значимых бит плоскости (каждая из менее значимых бит плоскости включает, например, один, два или три бита) кодируются, чтобы получить одно или несколько кодовых слов "acod_r". При кодировании наиболее значимых битов плоскости значение m наиболее значимого бита плоскости отображается в кодовое слово acod_m[pki][m]. Для этого 64 разных сводных таблиц частоты доступны для кодирования значения m в зависимости от состояния арифметического кодера 170, т.е. в зависимости от ранее кодированных спектральных значений. Таким образом, получается кодовое слово "acod_m[pki][m]". Кроме того, одно или несколько кодовых слов "acod_r" предусмотрены и включены в битовый поток, если присутствуют один или несколько менее значимых бит плоскостей.

Описание сброса

Аудио кодер 100 может быть дополнительно настроен на решение о том, можно ли достичь повышения битрейта путем сброса контекста, например, установив индекса состояния на значение по умолчанию. Таким образом, аудио кодер 100 может быть сконфигурирован для обеспечения информации сброса (например, под названием "arith_reset_flag"), указывающей, является ли контекст для арифметического кодирования сброшенным, а также указывающей, следует ли сбросить контекст для арифметического

декодирования в соответствующем декодере.

Подробнее формат битового потока и применяемые сводные таблицы частоты будут рассмотрены ниже.

4. Аудио декодер

Далее будет описан аудио декодер в соответствии с вариантом осуществления настоящего изобретения. Фиг.2 показывает блок-схему такого аудио декодера 200.

Аудио декодер 200 настроен на получение битового потока 210, который представляет кодированную аудио информацию и который может быть одинаковым с битовым потоком 112, предоставляемым кодером 100. Аудио декодер 200 обеспечивает декодированную аудио информацию 212 на основе битового потока 210.

Аудио декодер 200 включает в себя дополнительный де-форматтер полезной нагрузки битового потока 220, который настроен на получение битового потока 210 и извлечение из битового потока 210 кодированного аудио представления в частотной области 222. Например, де-форматтер полезной нагрузки битового потока 220 может быть настроен на извлечение из битового потока 210 арифметически кодированных спектральных данных, таких как, например, арифметическое кодовое слово "acod_m [pki][m]", представляющее значение наиболее значимого бита плоскости m спектрального значения a, а также кодовое слово "acod_r",. представляющее содержание менее значимого бита плоскости спектрального значение a в аудио представлении в частотной области. Таким образом, кодированное аудио представление в частотной области 222 составляет (или включает) арифметически кодированное представление спектральных значений. Де-форматтер полезной нагрузки битового потока 220 дополнительно настроен на извлечение из битового потока дополнительной информации управления, которая не показана на фиг.2. Кроме того, де-форматтер полезной нагрузки битового потока дополнительно настроен на извлечение из битового потока 210 информации сброса состояния 224, которая также обозначается как арифметический флаг сброса или "arith_reset_flag".

Аудио декодер 200 включает в себя арифметический декодер 230, который также обозначается как "спектральный бесшумный декодер". Арифметический декодер 230 настроена на прием кодированного аудио представления в частотной области 220 и, при необходимости, информации о сбросе состояния 224. Арифметический декодер 230 также настроен на предоставление декодированного аудио представления в частотной области 232, которое может включать' в себя декодированное представление спектральных значений. Например, декодированное аудио представление в частотной области 232 может содержать декодированное представление спектральных значений, которые описаны в кодированном аудио представлении в частотной области 220.

Аудио декодер 200 также включает в себя дополнительный обратный квантователь/ ре-скейлер 240, который настроен на получение декодированного аудио представления в частотной области 232 и предоставление на его основе обратно квантованного и ре-масштабированного аудио представления в частотной области 242.

Аудио декодер 200 также дополнительно может включать спектральный пред-процессор 250, который настроен на получение обратно квантованнного и ре-масштабированного аудио представления в частотной области 242 и предоставления на его основе предварительно обработанной версии 252 обратно квантованного и ре-масштабированного аудио представления в частотной области 242. Аудио кодер 200 также включает в себя трансформер сигнала из частотной области в временную 260, который также обозначается как конвертер сигнала. Трансформер сигнала 260 настроена на прием предварительно обработанной версии 252 обратно квантованного

и ре-масштабированного аудио представления в частотной области 242 (или, наоборот, обратно квантованного и ре-масштабированного аудио представления в частотной области 242 или декодированного аудио 'представления в частотной области 232) и предоставления на его основе аудио информации представления 262 во временной области. Трансформер сигнала из частотной области во временную область 260 может, например, включать трансформер для выполнения обратного модифицированного дискретного косинус-преобразования (IMDCT) и соответствующей оконной работы (а также других вспомогательных функций, как, например, перекрытие-и-добавление).

Аудио декодер 200 может дополнительно содержать пост-процессор временной области 270, который настроен на получение представления во временной области 262 аудио информации и для получения декодированной аудио информации 212 с помощью пост-обработки в временной области. Однако, если пост-обработка отсутствует, представление во временной области 262 может быть идентичным декодированной аудио информации 212..

Следует отметить, что обратный квантователь/рескейлер 240, спектральный пред-процессор 250, трансформер сигнала из частотной области во временную область 260 и пост-процессор во временной области 270 могут управляться в зависимости от управляющей информации, которая извлекается из битового потока 210 с помощью де-форматтера полезной нагрузки битового потока 220.

Подводя итог общей функциональности аудио декодера 200, декодированное аудио представление в частотной области 232, например, набор спектральных значений, связанных с аудио кадром кодированной аудио информации, могут быть получены на основе кодированного представления в частотной области 222 с помощью арифметического декодера 230. Следовательно, множество, например, 1024 спектральных значений, которые могут быть MDCT коэффициентами, обратно квантованы, ре-масштабированы и предварительно обработаны. Соответственно, обратно квантованное, ре-масштабированное и спектрально предварительно обработанное множество спектральных значений (например, 1024 MDCT коэффициенты) получается. Впоследствии, представление во временной области аудио кадра извлекается из обратно квантованного, ре-масштабированного и спектрально предварительно обработанного множества значений в частотной области (например, MDCT коэффициенты). Соответственно, получается представление во временной области аудио кадра. Представление во временной области данного аудио кадра может быть объединено с представлениями во временной области предыдущего и/или последующих аудио кадров. Например, перекрытие-и-добавление между представлениями во временной области последующих аудио кадров может быть выполнено для того, чтобы сгладить переходы между представлениями во временной области смежных аудио кадров и с целью получения отмены сглаживания. Для получения дополнительной информации о реконструкции декодированной аудио информации 212 на основе декодированного аудио представления в частотно-временной области 232, делается ссылка, например, на международный стандарт ISO/IEC 14496-3, часть 3, суб-часть 4, где это детально обсуждается. Тем не менее, другие более сложные схемы перекрытия и отмены наложения могут быть использованы.

Далее будут описаны подробности, касающиеся арифметического декодера 230. Арифметический декодер 230 включает в себя определитель наиболее значимого бита плоскости 284, который настроен на получение арифметического кодового слова acod_m [pki][m], описывающего значение m наиболее значимого бита плоскости. Определитель наиболее значимого бита плоскости 284 может быть настроен на использование сводной

таблицы частот из набора, содержащего множество 64 сводных таблиц частот для извлечения значения m наиболее значимого бита плоскости из арифметического кодового слова "acod_m [pki][m]".

Определитель наиболее значимого бита плоскости 284 настроен на извлечение значений 286 наиболее значимого бита плоскости спектральных значений на основе кодового слова acod_m. Арифметический декодер 230 дополнительна включает определитель наименее значимого бита плоскости 288, который настроен на получение одного или нескольких кодовых слов "acod_r", представляющих один или несколько менее значимых бит плоскости спектрального значения. Соответственно, определитель наименее значимого бита плоскости 288 настроен обеспечить декодированные значения 290 одного или нескольких менее значимых бит плоскости. Аудио декодер 200 также включает в себя сумматор бит плоскости 292, который настроен на получение декодированных значений 286 наиболее значимых бит плоскости спектральных значений и декодированных значений 290 одной или нескольких менее значимых бит плоскостей спектральных значений, если такие менее значимые бит плоскости доступные для текущих спектральных значений. Соответственно, сумматор бит плоскости 292 обеспечивает декодированные спектральные значения, которые являются частью декодированного аудио представления в частотной области 232. Естественно, арифметический декодер 230, как правило, настроены на предоставлении множества спектральных значений для того, чтобы' получить полный набор декодированных спектральных значений, связанных с текущим кадром аудио содержания.

Арифметический декодер 230 дополнительно включает селектор сводной таблицы частот 296, который настроен на выбор одной из 64 сводных таблиц частот в зависимости от индекса состояния 298, описывающего состояние арифметического декодера. Арифметический декодер 230 дополнительно включает трекер состояния 299, который настроен для отслеживания состояния арифметического декодера в зависимости от ранее декодированных спектральных значений. Информация о состоянии может необязательно быть сброшена к информации состояния по умолчанию в ответ на информацию сброса состояния 224. Таким образом, селектор сводной таблицы частот 296 настроен для предоставления индекса (например, pki), выбранной сводной таблицы частот или самой выбранной сводной таблицы частот, для применения в декодировании значения m наиболее значимого бита плоскости в зависимости от кодового слова "acod_m".

Подводя итог функциональности аудио декодера 200, аудио декодер 200 настроен на получение битрейт эффективного кодированного аудио представления в частотной области 222 и получение декодированного аудио представления в частотной области на его основе. В арифметическом декодере 230, который используется для получения декодированного аудио представления в частотной области 232 на основе кодированного аудио представления в частотной области 222, вероятность различных комбинаций значений наиболее значимых бит плоскостей смежных спектральных значений используется с помощью арифметического декодера 280, который настроен применять сводную таблицу частот. Другими словами, статистические зависимости между спектральными значениями эксплуатируются путем выбора различных сводных таблиц частоты из набора, включающего 64 различных сводных таблиц частоты в зависимости от индекса состояния 298, который получается при наблюдении за ранее вычисленными декодированными спектральными значениями.

5. Обзор за инструментов спектрального бесшумного кодирования

Далее будут описаны подробности, касающиеся алгоритма кодирования и

декодирования, который выполняется, например, арифметическим кодером 170 и арифметическим декодером 230.

Основное внимание уделяется описанию алгоритма декодирования. Следует отметить, однако, что соответствующий алгоритм кодирования может быть выполнен в соответствии с объяснением алгоритма декодирования, в котором отображения меняются на противоположные.

Следует отметить, что декодирование, которое будет обсуждаться далее, используется для того, чтобы обеспечить так называемое "спектральное бесшумное кодирование" обычно пост-обработанных, масштабированных и квантованных спектральных значений. Спектральное бесшумное кодирование используются в концепции аудио кодирования / декодирования для дальнейшего сокращения избыточности квантованного спектра, которые получают, например, при помощи энергоуплотняющего трансформера из временной области в частотную область.

Схема спектрального бесшумного кодирования, которое используется в вариантах изобретения, основана на арифметическом кодировании в сочетании с динамически адаптированным контекстом. Бесшумное кодирование снабжается (оригинальными или кодированными представлениями) квантованными спектральными значениями и использует контекстно-зависимые сводные таблицы частот, полученные, например, из множества ранее декодированных соседних спектральных значений. Здесь, учитывается соседство как во времени, так и по частоте, как показано на фиг.4. Сводные таблицы частот (о которых будет сказано ниже) затем используются арифметическим кодером для создания двоичного кода переменной длины и арифметическим декодером для извлечения декодированных значений из двоичного кода переменной длины.

Например, арифметический кодер 170 производит двоичный код для данного набора символов, в зависимости от соответствующих вероятностей. Двоичный код образуется путем отображения интервала вероятности, в котором лежит набор символов, на кодовое слово.

Далее будет дан еще один короткий обзор инструментов спектрального бесшумного кодирования. Спектральное бесшумное кодирование используется для дальнейшего сокращения избыточности квантованного спектра. Схема спектрального бесшумного кодирования основывается на арифметическом кодировании в сочетании с динамически адаптированным контекстом. Бесшумное кодирование снабжается квантованными спектральными значениями и использует контекстно-зависимые сводные таблицы частот, полученные, например, из семи ранее декодированных соседних спектральных значений.

Здесь, учитывается соседство как во времени, так и по частоте, как показано на фиг.4. Сводные таблицы частот затем используются арифметическим кодером для генерации двоичного кода переменной длины.

Арифметический кодер производит двоичный код для данного набора символов и их соответствующих вероятностей. Двоичный код образуется путем отображения интервала вероятности, в котором лежит набор символов, на кодовое слово.

6. Процесс декодирования

6.1 Обзор процесса декодирования

Далее будет дан обзор процесса декодирования спектрального значения со ссылкой на фиг.3, которая показывает представление псевдо-программного кода процесса декодирования множества спектральных значений.

Процесс декодирования множества спектральных значений содержит инициализацию 320 контекста. Инициализация 310 контекста включает в себя извлечение текущего

контекста из предыдущего контекста с помощью функции "arith_map_context (Ig)'\ Извлечение текущего контекста из предыдущего контекста может включать в себя сброс контекста. И сброс контекста, и извлечение текущего контекста из предыдущего контекста будут рассмотрены ниже.

Декодирование множества спектральных значений также включает в себя повторение декодирования спектральных значений 312 и обновление контекста 314, которое обновление выполняется функцией "Arith_update_context(a,i,lg)", которая описана ниже. Декодирование спектральных значений 312 и обновление контекста 314 повторяется lg раз, при этом lg указывает число спектральных значений для декодирования (например, для аудио кадра). Декодирование спектральных значений 312 включает в себя расчет значения контекста 312a, декодирование наиболее значимого бита плоскости 312b, и добавление менее значимого бита плоскости 312 c.

Вычисление значения состояния 312a включает в себя вычисление первого значения состояния s при помощи функции "arith_get_context(i, lg, arith_reset_flag, N/2)", которая возвращает первое значение состояния s. Вычисление значения состояния 312a также включает в себя вычисление значения уровня "lev0" и значения уровня "lev", эти значения уровня "lev0", „lev" получаются путем сдвига первого значения состояния s вправо на 24 бит. Вычисление значения состояния 312a также включает в себя вычисление второго значения состояния t в соответствии с формулой, приведенной на фиг.3 на ссылке с номером 312a.

Декодирование наиболее значимого бита плоскости 312b включает в себя итерационное выполнение алгоритма декодирования 312ba, при этом переменная j инициализируется до 0 перед первым выполнением алгоритма 312ba.

Алгоритм 312ba включает в себя вычисление индекса состояния „pki" (который также служит в качестве индекса сводной таблицы частот) в зависимости от второго значения состояния t, а также в зависимости от значений уровня „lev" и lev0, с помощью функции "arith_get_pk()"», которая обсуждается ниже. Алгоритм 312ba также включает в себя выбор сводной таблицы частот в зависимости от индекса состояния pki, где переменная "cum_freq" может быть установлена на начальный адрес одной из 64 сводных таблиц частот в зависимости от индекса pki. Кроме того, переменная "cfl" может быть инициализирована на длину выбранной сводной таблицы частот, которая, например, равна количества символов в алфавите, то есть количеству различных значений, которые могут быть декодированы. Длины всех сводных таблиц частот от "arith_cf_m[pki=0][9]" до "arith_cf_m[pki=63][9]", доступных для декодирования значения наиболее значимого бита плоскости т, составляют 9, так что восемь различных значений наиболее значимых бит плоскости и управляющий символ могут быть декодированы. Впоследствии, значение наиболее значимого бита плоскости m может быть получено путем выполнения функции "arith_decodeO", с учетом выбранной сводной таблицы частоты (описанной переменной "cum_freq" и переменной "cfl"). При извлечении значения наиболее значимого бита плоскости m, биты под названием "acod_m" в битовом потоке 210 могут быть оценены (см., например, фиг.6g).

Алгоритм 312ba также включает в себя проверку того, равно ли значение наиболее значимого бита плоскости m управляющему символу "ARITH_ESCAPE", или нет. Если значение наиболее значимого бита плоскости m не равно арифметическому управляющему символу, алгоритм 312ba прерывается (условие "прерывания"), а остальные инструкции алгоритма 312ba поэтому пропущены. Таким образом, выполнение процесса продолжается установкой спектрального значения a равным значению наиболее значимого бита плоскости m (инструкция "a=m"). В отличие от

этого, если декодированное значение наиболее значимого бита плоскости m совпадает с арифметическим управляющим символом "ARITH_ESCAPE", значение уровня „lev" увеличивается на единицу. Как уже упоминалось, алгоритм 312ba повторяется до тех пор, пока декодированное значение наиболее значимого бита плоскости m отличается от арифметического управляющего, символа.

Как только декодирование наиболее значимого бита плоскости завершено, то есть значение наиболее значимого бита плоскости m, которое отличается от арифметического управляющего символа, декодировано, переменная спектрального значения "a" устанавливается равной значению самого значимого бита плоскости т.Впоследствии, получаются менее значимые биты плоскости, например, как показано на ссылке с номером 312 с на фиг.3. Для каждого менее значимого бита плоскости спектрального значения, одно из двух двоичных значений декодируется. Например, получается значение менее значимого бита плоскости г.Впоследствии, переменная спектрального значения "a" обновляется, сдвигая содержание переменной спектрального значения "a" влево на 1 бит и добавляя значение ранее декодированного менее значимого бита плоскости r как наименее значимого бита. Тем не менее, следует отметить, что концепция для получения значений менее значимых бит плоскостей не имеет особого значения для настоящего изобретения. В некоторых вариантах, декодирование любых менее значимых бит плоскостей может даже быть опущено. Кроме того, различные алгоритмы декодирования могут быть использованы для этой цели.

6.2 Порядок декодирования в соответствии с фиг.4

Далее будет описан порядок декодирования спектральных значений.

Спектральные коэффициенты бесшумно кодируются и передаются (например, в битовом потоке), начиная с самого низкочастотного коэффициента и переходя к самому высокочастотному коэффициенту.

Коэффициенты из перспективного звукового кодирования (AAC) (например, полученные с помощью модифицированного дискретного косинус преобразования, как описано в ISO/IEC 14496, часть 3, подчасть 4) хранятся в массиве "x_ac_quant[g][win][sfb][bin]", а порядок передачи кодового слова бесшумного кодирования (т.е. acod_m, acod_r) такой, что, когда они декодируются в порядке поступления и хранятся в массиве, "bin" (индекс частоты) является наиболее быстро увеличивающимся индексом и "g" является наиболее медленно увеличивающимся индексом.

Спектральные коэффициенты, связанные с более низкой частотой, кодируются перед спектральными коэффициентами, связанными с более высокой частотой.

Коэффициенты из преобразования кодированного возбуждения (TCX) хранятся непосредственно в массиве x_tcx_invquant[win][bin], а порядок передачи кодовых слов бесшумного кодирования такой, что, когда они декодируются в порядке поступления и хранятся в массиве, "bin" является наиболее быстро увеличивающимся индексом и "win" является наиболее медленно увеличивающимся индексом. Другими словами, если спектральные значения описывают преобразование кодированного возбуждения фильтра линейного предсказания кодера речи, спектральные значения а связаны со смежными и увеличивающимися частотами преобразование кодированного возбуждения.

Спектральные коэффициенты, связанные с более низкой частотой, кодируются перед спектральными коэффициентами, связанными с более высокой частотой.

Примечательно, что аудио декодер 200 может быть настроен на применение декодированного аудио представления в частотной области 232, которое обеспечивается арифметическим декодером 230, как для "прямой" генерации представления аудио сигнала во временной области с помощью преобразования сигнала из частотной области

во временную область, так и для "косвенного" предоставления представления аудио сигнала, используя как декодер из частотной области во временную область, так и фильтр линейного предсказания, возбуждаемый выходом трансформера сигнала из частотной области во временную область.

Другими словами, арифметический декодер 200, функциональность которого обсуждается здесь в деталях, хорошо подходит для декодирования спектральных значений представления во временной и частотной области аудио содержания, кодированного в частотной области, и для обеспечения представления во временной и частотной области сигнала стимула для фильтра линейного предсказания, адаптированного для декодирования речевого сигнала, кодированного в области линейного предсказания. Таким образом, арифметический декодер хорошо подходит для использования в аудио декодере, способном работать как с аудио содержанием, кодированном в частотной области, так и с аудио содержанием, кодированном в линейно предсказанной частотной области (режим преобразования кодированного возбуждения области линейного предсказания).

6.3. Инициализация контекста в соответствии с фиг.5a и 5b

Далее будет описана инициализация контекста (также обозначается как "отображение контекста"), которая выполняется в шаге 310.

Инициализация контекста включает сопоставление между прошлым контекстом и текущим контекстом в соответствии с алгоритмом "arith_map_ context()", который показан на фиг.5a. Как видно, текущий контекст хранится в глобальной переменной q [2] [ncontext], которая принимает форму массива, имеющего первое измерение из двух и второе измерение из n_context. Прошлый контекст хранится в переменной qs[n_context], которая принимает форму таблицы, имеющей измерение из n_context. Переменная "previous_lg" описывает количество спектральных значений прошлого контекста.

Переменная "lg" описывает количество спектральных коэффициентов для декодирования в кадре. Переменная "previous_lg" описывает предыдущее количество спектральных линий предыдущего кадра.

Отображение контекста может быть выполнено в соответствии с алгоритмом "arith_map_context()". Следует отметить, что функция "arith_map_context()" устанавливает записи q[0][i] текущего массива контекста q в значения qs[i] предыдущего массива контекста qs, если количество спектральных значений, связанных с текущим (например, кодированном в частотной области) аудио кадром, совпадает с количеством спектральных значений, связанных с предыдущим аудио кадром для i=0 до i=lg-1.

Однако, более сложное отображение выполняется, если количество спектральных значений, связанных с текущим аудио кадром, отличается от количества спектральных значений, связанных с предыдущим аудио кадром. Однако подробности, касающиеся отображения в данном случае, не особенно важны для ключевой идеи настоящего изобретения, так что за более подробной информацией делается ссылка на псевдо программный код на фиг.5a.

6.4 Вычисление значения состояния в соответствии с фиг.5b и 5c

Далее вычисление значение состояния 312a будет описано более подробно.

Следует отметить, что первое значение состояния s (как показано на фиг.3) может быть получено в качестве возвращаемого значения функции "arith_get_context(i, lg, arith_reset_flag, N/2)", представление псевдо программного кода, которое показано на фиг.5b и 5c.

Что касается вычисления значения состояния, делается также ссылка на фиг.4, которая показывает контекст, используемый для оценки состояния. Фиг.4 показывает двумерное

представление спектральных значений как по времени, так и по частоте. Абсцисса 410 описывает время, а ордината 412 описывает частоту. Как видно на фиг.4, спектральное значение 420 для декодирования, связано с индексом времени t0 и индексом частоты i. Как видно, для индекса времени Ю, кортежи, имеющие индексы частоты i-1, i-2 и i-3, уже декодированы в то время, когда спектральное значение 420 с индексом частоты i должно быть декодировано. Как видно из фиг.. 4, спектральное значение 430, имеющее индекс времени t0 и индекс частоты i-1, уже декодировано до того, как спектральное значение 420 декодировано, а спектральное значение 430 рассматривается для контекста, который используется для декодирования спектрального значения 420. Таким же образом, спектральное значение 434, имеющее индекс времени t0 и индекс частоты i-2, уже декодировано, до того как спектральное значение 420 декодируется, и спектральное значение 434 рассматривается для контекста, который используется для декодирования спектрального значения 420. Таким же образом, спектральное значение 440, имеющее индекс времени Ми индекс частоты i-2, спектральное значение 444, имеющее индекс времени t-1 и индекс частоты i-1, спектральное значение 448, имеющее индекс времени t-1 и индекс частоты i, спектральное значение 452, имеющее индекс времени t-1 и индекс частоты i+1, и спектральное значение 456, имеющее индекс времени t-1 и индекс частоты i+2 уже декодированы, до того как спектральное значение 420 декодируется, и рассматриваются для определения контекста, который используется для декодирования спектрального значения 420. Спектральные значения (коэффициенты), уже декодированные в то время, когда спектральное значение 420 декодируется и рассматривается для контекста, показаны в заштрихованных квадратах. В отличие от этого, некоторые другие спектральные значения, уже декодированные (в то время, когда спектральное значение 420 декодируется), которые представлены квадратами с пунктирными линиями, а также другие спектральные значения, которые до. сих пор не декодированы (в то время, когда спектральное значение 420 декодируется) и которые показаны кружками с пунктирными линиями, которые не используются для определения контекста для декодирования спектрального значения 420.

Тем не менее, следует отметить, что некоторые из этих спектральных значений, которые не используются для "обычного" (или "нормального") вычисления контекста для декодирования спектрального значения 420, могут, тем не менее, быть оцененными для выявления множества ранее декодированных смежных спектральных значений, которые выполняют, по отдельности или вместе взятые, заданное условие относительно их величины.

Обратимся к фиг.5b и 5 c, которые показывают функциональность функции "arith_get_context()" в виде псевдо программного кода, больше подробностей относительно расчета первого значения контекста "s", который осуществляется с помощью функции "arith_get_contextO", будут описаны.

Следует отметить, что функция "arith_get_context()" получает, в качестве входных переменных индекс i спектрального значения для декодирования. Индекс i, как правило, является индексом частоты. Входная переменная lg описывает (общее) количество ожидаемых квантованных коэффициентов (для текущего аудио кадра). Переменная N описывает количество линий преобразования. Флаг "arith_reset_flag" указывает должен ли контекст быть сброшен. Функция "arith_get_context" предоставляет, в качестве выходного значения, переменную "t", которая представляет собой сцепленный индекс состояния s и предсказанный уровень бита плоскости lev0.

Функция "arith_get_contextO" использует целочисленные переменные a0, c0, c1, c2, c3, c4, c5, c6, lev0, и «область».

Функция "arith_get_context()" содержит в качестве основных функциональных блоков, обработку первого арифметического сброса 510, обнаружение 512 группы из множества ранее декодированных смежных нулевых спектральных значений, установку первой переменной 514, установку второй переменной 516, адаптацию уровня 518, установку значения области 520, адаптацию уровня 522, ограничение уровня 524, обработку арифметического сброса 526, установку третьей переменной 528, установку четвертой переменной 530, установку пятой переменной 532, адаптацию уровня 534, и селективное вычисление возвращаемого значения 536.

При обработке первого арифметического сброса 510 проверяется, установлен ли флаг арифметического сброса "arith_reset_flag", когда индекс спектрального значения для декодирования равен нулю. В этом случае нулевое значение контекста возвращается, а функция прерывается.

При обнаружении 512 группы из. множества ранее декодированных нулевых спектральных значений, которое производится, только если флаг арифметического сброса неактивен, а индекс i спектрального значение для декодирования отличается от нуля, переменная с именем "flag" устанавливается в 1, как показано на ссылке с номером 512a, и область спектрального значения, которое оценивается, определяется как показано на ссылке с номером 512b. Впоследствии область спектральных значений, которая определяется, как показано на ссылке с номером 512b, оценивается, как показано на ссылке с номером 512c. Если установлено, что имеется достаточная область ранее декодированных нулевых спектральных значений, значение контекста 1 возвращается, как показано на ссылке с номером 512d. Например, верхняя граница индекса частоты "Hm_max" устанавливается в положение i+6, если индекс i спектрального значения для декодирования не близок к максимальному индексу частоты lg-1, и в этом случае специальная установка верхней границы индекса частоты производится, как показано на ссылке с номером 512b. Кроме того, нижняя граница индекса частоты "lim_min" устанавливается в положение -5, если индекс i спектрального значения для декодирования не близок к нулю (i+Um_min<0), и в этом случае специальное вычисление нижней границы индекса частоты lim_min производится, как показано на ссылке с номером 512b. При оценке области спектральных значений, определенных в шаге 512b, оценка сначала исполнена для отрицательных индексов частоты к между нижней границей индекса частоты lim_min и нулем. Для индексов частоты к между lim_min и нулем проверяется, равен ли хотя бы один из значений контекста q[0][k].c и q[1][k].c нулю. Если, однако, оба значения контекста q[0][k].c и q[1][k].c отличны от нуля для любых индексов частоты к между lim_min и нулем, можно сделать вывод, что нет достаточной группы нулевых спектральных значений, и оценка 512 с прерывается. Далее значения контекста q[0][k].c для индексов частоты между нулем и lim_max оцениваются. Если обнаруживается, что любые из значений контекста q[0][k].c для любых индексов частоты между нулем и lim_max отличаются от нуля, можно сделать вывод, что нет достаточной группы ранее декодированных нулевых спектральных значений, и оценка 512 с прерывается. Однако, если будет установлено, что для каждого индекса частоты к между lim_min и нулем, то есть по крайней мере одно значение контекста q[0][k].c или q[1][k].c, которое равно нулю, и если есть нулевое значение контекста q[0][k].c для каждого индекса частоты к между нулем и lim_max, можно сделать вывод, что есть достаточная группа ранее декодированных нулевых спектральных значений. Таким образом, значение контекста 1 возвращается в этом случае, чтобы указать на это условие, без каких-либо дополнительных расчетов. Другими словами, расчеты 514, 516, 518, 520, 522, 524, 526, 528, 530, 532, 534, 536 пропускаются,

если достаточная группа множества значений контекста q[0][k].c, q[1][k].c, имеющих нулевое значение, выявлена. Другими словами, возвращаемое значение контекста, которое описывает состояние контекста (s), определяется независимо от ранее декодированных спектральных значений в ответ на обнаружение, что заданное условие выполнено.

В противном случае, т.е. если нет достаточной группы значений контекста [q][0][k].c, [q][1][k].c, которые равны нулю, по крайней мере некоторые из вычислений 514, 516, 518, 520, 522,524 526, 528, 530, 532, 534, 536, выполняются.

При установке первой переменной 514, которая избирательно выполняется, если (и только если) индекс i спектрального значения для декодирования меньше 1, то переменная a0 инициализируется для принятия значения контекста q[1][i-1], а переменная c0 инициализируется для принятия абсолютного значения переменной a0. Переменная "lev0" инициализируется для принятия значения нуля. Впоследствии переменные "lev0" и c0 увеличиваются, если переменная a0 содержит сравнительно большое абсолютное значение, т.е. меньше, чем -4, или больше или равно 4. Увеличение переменных "lev0" и c0 выполняется итеративно, пока значение переменной a0 приводится в диапазон между -4 и 3 путем операции сдвига направо (шаг 514b).

Впоследствии переменные c0 и "lev0" ограничиваются максимальными значениями 7 и 3 соответственно (шаг 514 c).

Если индекс i спектрального значения для декодирования равен 1, а флаг арифметического сброса ("arith_reset_flag") является активным, значение контекста возвращается, которое рассчитывается лишь на основе переменных cO и levO (шаг 514d). Таким образом, только одно ранее декодированное спектральное значение, имеющее один и тот же индекс времени как спектральное значение для декодирования, и имеющее индекс частоты, который меньше, на 1, чем индекс частоты i спектрального значения для декодирования, рассматривается для вычисления контекста (шаг 514d). В противном случае, т.е. если нет функциональности арифметического сброса, переменная c4 инициализируется (шаг 514e).

В заключение, установка первой переменной 514, переменных c0 и "lev0" инициализируются в зависимости от ранее декодированных спектральных значений, декодированных за тот же кадр, как и спектральное значение для текущего декодирования и для предыдущей спектральной ячейки Переменная c4 инициализируется в зависимости от ранее декодированного спектрального значения, декодированного из предыдущего аудио кадра (имеющего индекс времени t-1) и имеющего частоту, которая ниже (например, на одну ячейку частоты), чем частота, связанная с спектральным значением для текущего декодирования.

Установка второй переменной 516, которая избирательно выполняется, если (и только если) индекс частоты спектрального значения для текущего декодирования больше 1, включает в себя инициализацию переменных c1 и c6 и обновление переменной lev0. Переменная c1 обновляется в зависимости от значения контекста q[1][i-2].c, связанного с ранее декодированным спектральным значением текущего аудио кадра, частота которого меньше (например, на две ячейки частоты), чем частота спектрального значения для текущего декодирования. Кроме того, переменная c6 инициализируется в зависимости от значения контекста q[0][i-2].c, которое описывает ранее декодированное спектральное значение предыдущего кадра (имеющего индекс времени t-1), связанная частота которого меньше (например, две ячейки частоты), чем частота, связанная со спектральным значение для текущего декодирования. Кроме того, переменная уровня "lev0" устанавливается на значение уровня q[1][i-2].1, связанное с

ранее декодированным спектральным значением текущего кадра, связанная частота которого меньше (например, на две ячейки частоты), чем частота, связанная со спектральным значением для текущего декодирования, если q[1][i-2].1 больше, чем lev0.

Адаптация уровня 518 и установка значения области 520 выборочно выполняются, если (и только если) индекс i спектрального значения для декодирования больше, чем 2. При адаптации уровня 518, переменная уровня "lev0" увеличивается на значение q[1][i-3].1, если значение уровня q[1][i-3].1, связанное с ранее декодированным спектральным значением текущего кадра, связанная частота которого меньше (например, на три ячейки частоты), чем частота, связанная со спектральным значением для текущего декодирования, больше, чем значение уровня lev0.

При установке значения области 520 переменная «область» устанавливается в зависимости от оценки, в которой спектральной области, из множества спектральных областей, спектральное значения для текущего декодирования получается. Например, если установлено, что спектральное значение для текущего декодирования связано с ячейкой частоты (имеющей индекс ячейки частоты i), которая есть в первой (самой нижней) четверти ячеек частоты ($0 \le i < N/4$), переменная области «область» равна нулю. В противном случае, если спектральное значение для текущего декодирования связано с ячейкой частоты, которая во второй четверти ячеек частоты, связанное с текущим кадром ($N/4 \le i < N/2$), переменная области устанавливается в значение 1. В противном случае, если спектральное значение для текущего декодирования связано с ячейкой частоты, которая во второй (верхней) половиной ячеек частоты ($N/2 \le i < N$), переменная области устанавливается в значение 2. Таким образом, переменная области устанавливается в зависимости от оценки, с какой частотной областью спектральное значение для текущего декодирования связано. Можно выделить две или более частотных областей.

Дополнительная адаптация уровня 522 выполняется, если (и только если) спектральное значение для текущего декодирования включает в себя спектральный индекс, который больше, чем 3. В этом случае переменная уровня "lev0" увеличивается (устанавливается на значение q[1][i-4].l), если значение уровня q[i][i-4].l, связанное с ранее декодированным спектральным значением текущего кадра, который связан с частотой, которая меньше, например, на четыре ячейки частоты, чем частота, связанная со спектральным значением для текущего декодирования, больше, чем текущий уровень „lev0" (шаг 522). Переменная уровня "lev0" ограничивается максимальным значением 3 (шаг 524).

Если условие арифметического сброса обнаруживается и индекс i спектрального значения для текущего декодирования больше, чем 1, значение состояния возвращается в зависимости от переменных c0, c1, lev0, a, также в зависимости от переменной области "область" (шаг 526). Таким образом, ранее декодированные спектральные значения любого предыдущего кадра остаются без внимания, если условие арифметического сброса дается.

В установке третьей переменной 528 переменная c2 устанавливается в значение контекста q[0][i].c, которое связано с ранее декодированным спектральным значением предыдущего аудио кадра (имеющего индекс времени t-1), которое ранее декодированное спектральное значение связано с той же частотой как и спектральное значение для текущего декодирования.

В установке четвертой переменной 530 переменная c3 устанавливается в значение контекста q[0][i+1].c, которое связано с ранее декодированным спектральным значением предыдущего аудио кадра, имеющего индекс частоты i+1, если спектральное значение

для текущего декодирования не связано с самым большим возможным индексом частоты lg-1.

В установке пятой переменной 532 переменная c5 устанавливается в значение контекста q[0][i+2].c, которое связано с ранее декодированным спектральным значением предыдущего аудио кадра, имеющего индекс частоты i+2, если индекс частоты i спектрального значения для текущего декодирования не слишком близко к максимальному значению индекса частоты (т.е. имеет значение индекса частоты lg-2 или lg-D.

Дополнительная адаптация переменной уровня "lev0" выполняется, если индекс частоты i равен нулю (т.е. если спектральное значение для текущего декодирования является самым нижним спектральным значением). В этом случае переменная уровня "lev0" увеличивается от нуля до 1, если переменная c2 или c3 имеет значение 3, что указывает на то, что ранее декодированное спектральное значение предыдущего аудио кадра, который связан с той же частотой или даже с более высокой частотой по сравнению с частотой, связанной со спектральным значением для текущего кодирования, имеет сравнительно большое значение.

В выборочном вычислении возвращаемого значения 536 возвращаемое значение вычисляется в зависимости от того, имеет ли индекс i спектрального значения для текущего декодирования значение нуль, 1 или большее значение. Возвращаемое значение вычисляется в зависимости от переменных c2, c3, c5 и lev0, как указано в ссылке с номером 536a, если индекс i принимает значения нуль. Возвращаемое значение вычисляется в зависимости от переменных c0, c2, c3, c4, c5, и "lev0", как показано на ссылке с номером 536b, если индекс i принимает значение 1. Возвращаемое значение вычисляется в зависимости от переменных c0, c2, c3, c4, c5, c6, "область" и "lev0", если индекс i принимает значение, которое отличается от нуля или 1 (ссылка с номером 536 c).

Подводя итог сказанному выше, вычисление значения контекста "arith_get_context ()" включает в себя обнаружение 512 группы множества ранее декодированных нулевых спектральных значений (или, по крайней мере, достаточно малых спектральных значений). Если обнаружена достаточная группа ранее декодированных нулевых спектральных значений, наличие специального контекста указывается путем установки возвращаемого значения в 1. В противном случае, производится вычисление значения контекста. В целом можно сказать, что при вычислении значения контекста значение индекса i оценивается для того, чтобы решить, сколько ранее декодированных спектральных значений должно быть оценено. Например, количество оцененных ранее декодированных спектральных значений уменьшается, если индекс частоты i спектрального значения для текущего декодирования близок к нижней границе (например, нулю), или близок к верхней границе (например, lg-1). Кроме того, даже если индекс частоты i спектрального значения для текущего декодирования достаточно далек от минимального значения, разные спектральные области выделяются установкой значения области 520. Соответственно, различные статистические свойства различных спектральных областей (например, во-первых, низкочастотная спектральная область, во-вторых, среднечастотная спектральная область, и, в-третьих, высокочастотная спектральная область) принимаются во внимание. Значение контекста, которое рассчитывается в качестве возвращаемого значения, зависит от переменной «область», такой, что возвращаемое значение контекста зависит от того, находится ли спектральное значение для текущего декодирования в первой заданной частотной области или во второй заданной частотной области (или в любой другой заданной частотной области).

6.5 Выбор правила отображения

Далее будет описан выбор правила отображения, например, сводной таблицы частот, которая описывает отображение значения кода на код символа. Выбор правила отображения производится в зависимости от состояния контекста, который описывается значением состояния s или t.

6.5.1 Выбор правила отображения с помощью алгоритма в соответствии с Фиг.5d

Далее описывается выбор правила отображения с помощью функции "get_pk" в соответствии с фиг.5d. Следует отметить,, что функция "get_pk" может быть выполнена, чтобы получить значение " pki " в суб-алгоритме 312ba алгоритма на фиг.3. Таким образом, функция "get_pk" может заменить функцию "arith_get_pk" в алгоритме на фиг.3.

Следует также отметить, что функция "get_pk" в соответствии с фиг.5d может оценить таблицу "ari_s_hash [387]" в соответствии с фиг.17 (1) и 17 (2) и таблицу "ari_gs_hash" [225] в соответствии с фиг.18.

Функция "get_pk" получает, в качестве входной переменной, значение состояния s, которое может быть получено путем сочетания переменной "t" в соответствии с фиг.3 и переменных "lev", "lev0" в соответствии с фиг.3. Функция "get_pk" также в качестве возвращаемого значения может вернуть значение переменной " pki ", которая характеризует правило отображения или сводную таблицу частот. Функция "get_pk" настроена отобразить значение состояния s на значение индекса правила отображения "pki".

Функция "get_pk" включает в себя первую оценочную таблицу 540 и вторую оценочную таблицу 544. Первая оценочная таблица 540 включает в себя инициализацию переменной 541, в которой инициализируются переменные i_min, i_max, и i, как показано на ссылке с номером 541. Первая оценочная таблица 540 также включает в себя итеративный поиск в таблице 542, в ходе которого определяется, есть ли запись в таблице "ari_s_hash", которая соответствует, значению состояния s. Если такое совпадение выявляется в ходе поиска итерационной таблицы 542, функция get_pk прерывается, при этом возвращаемое значение функции определяется записью таблицы "ari_s_hash", которая соответствует значению состояния s, что будет описано более подробно далее. Если, однако, в ходе итерационного поиска таблицы 542 не выявляется идеальное соответствие значения состояния s и записи таблицы "ari_s_hash", выполняется проверка граничной записи 543.

Обратимся теперь к деталям первой оценочной таблицы 540, видно, что интервал поиска определяется переменными i_min и i_max. Итеративный поиск таблицы 542 повторяется до тех пор, пока интервал, определенный переменными i_min и i_max, достаточно велик, что может быть истинным, если условие i_max-i_min>1 выполняется. Впоследствии устанавливается переменная i, по крайней мере приблизительно, для обозначения середины интервала (i=i_min+(i_max-i_min)/2). Далее устанавливается переменная j на значение, которое определяется массивом "ari_s_hash" в положении массива, обозначенном переменной i (ссылка с номером 542). Здесь следует отметить, что каждая запись в таблице "ari_s_hash" описывает как значение состояния, которое связано с записью таблицы, так и значение индекса правила отображения, которое связано с записью таблицы. Значение состояния, которое связано с записью таблицы, описывается более значимыми битами (8-31 биты) записи таблицы, в то время как значения индекса правила отображения характеризуются нижними битами (например, биты 0-7) записи указанной таблицы. Нижняя граница i_min или верхняя граница i_max адаптированы в зависимости от того, если значение состояния s меньше, чем значение-состояния, описыаемое наиболее значимыми 24 битами записи "ari_s_hash[i]" таблицы

"ari_s_hash", которая ссылается на переменную i. Например, если значение состояния s меньше, чем значение состояния, описанное более значимыми 24 битами записи "ari_s_hash [i]", верхняя граница i_max интервала таблицы устанавливается в значение i. Соответственно, интервал таблицы для следующей итерации итеративного поиска таблицы 542 ограничен нижней половиной интервала таблицы (от i_min в i_max), используемой для текущей итерации итеративного поиска.таблицы 542. Если, напротив, значение состояния s больше значений состояния, описываемого более значимыми 24 битами записи таблицы "ari_s_hash [i]", то нижняя граница i_min интервала таблицы для следующей итерации итеративного поиска таблицы 542 устанавливается в значение i, так что верхняя половина текущего интервала таблицы (между i_min и i_max) используется в качестве интервала таблицы для следующего итеративного поиска таблицы. Однако, если будет установлено, что значение состояния s идентично значению состояния, описанному наиболее значимыми 24 битами записи таблицы' "arijs_hash [i] ", значение индекса правила отображения, описываемое менее значимыми 8 битами записи таблицы "ari_s_hash [i]", возвращается функцией "get_pk", а функция отменяется.

Итеративный поиск таблицы 542 повторяется, пока интервал таблицы, определяемый переменными i_min и i_max, становится достаточно малым.

Проверка граничной записи 543 (дополнительно) выполняется в дополнение к итеративному поиску таблицы 542. Если индексная переменная i равна индексной переменной i_max после завершения итеративного поиска таблицы 542, окончательная проверка производится, является ли значение состояния s равным значению состояния, описываемому более значимыми 24 битами записи таблицы "ari_s_hash [i_min] ", и значение индекса правила отображения, описываемое менее значимыми 8 битами записи "ari_s_hash [imin]" возвращается, в этом случае, как результат функции "get_pk C другой стороны, если индексная переменная i отличается от индексной переменной i_max, то выполняется проверка, является ли значение состояния s равным значению состояния, описываемому более значимыми 24 битами записи таблицы "ari_s_hash [i_max]", и значение индекса правила отображения, описываемое менее значимыми 8 битами записи указанной таблицы "ari_s_hash [i_max]", возвращается в виде возвращаемого значения функции" get_pk "в данном случае.

Тем не менее, следует отметить, что проверка граничной записи 543 может рассматриваться как дополнительная в полном объеме.

После первой оценочной таблицы 540 выполняется вторая оценочная таблица 544, если не произошло "прямое попадание" во время первой оценочной таблицы 540, в которой значение состояния s совпадает с одним из значений состояния, описываемых записями таблицы "ari_s_hash" (или, точнее, более значимыми 24 битами).

Вторая оценочная таблица 544 включает в себя инициализацию переменной 545, в которой индексные переменные i_min, i и i_max инициализируются, как показано на ссылке с номером 545. Вторая оценочная таблица 544 также включает в себя итеративный поиск таблицы 546, в ходе которого в таблице "ari_gs_hash" ищется запись, которая представляет собой значение состояния, идентичное значению состояния s. Наконец, вторая таблица поиска 544 включает в себя определение возвращаемого значения 547.

Итеративный поиск таблицы 546 повторяется до тех пор, пока интервал таблицы, определяемый индексными переменными i_min и i_max, является достаточно высоким (например, до тех пор, пока i_max - i_min>1). В процессе итерации итеративного поиска таблицы 546 переменная i устанавливается к центру интервала таблицы, определяемому i_min и i_max (шаг 546a). Впоследствии запись j таблицы "ari_gs_hash" выполняется в

части таблицы, определяемой по индексной переменной i (546b). Другими словами, запись таблицы "ari_gs_hash [i]" является записью таблице в центре текущего интервала таблицы, определяемого индексами таблицы i_min и i_max. Далее определяется интервал таблицы для следующей итерации итеративного поиска таблицы 546. Для этой цели значение индекса ijmax, описывающее верхнюю границу интервала таблицы, устанавливается в значение i, если значение состояния s меньше, чем значение состояния, описываемое более значимыми.24 битами записи таблицы "j=ari_gs_hash [i] "(546 c). Другими словами, нижняя половина текущего интервала таблицы выбирается в качестве нового интервала таблицы для следующей итерации итеративного поиска таблицы 546 (шаг 546 c). В противном случае, если значение состояния s больше, чем значение состояния, описываемое более значимыми 24 битами записи таблицы "j=ari_gs_hash значение индекса i_min устанавливается в значение i. Таким образом, верхняя половина текущего интервала таблицы выбирается в качестве нового интервала таблицы для следующей итерации итеративного поиска таблицы 546 (шаг 546d). Однако, если будет установлено, что значение состояния s совпадает с значением состояния, описываемым более значимыми 24 битами записи таблицы "j=ari_gs_hash [i]", индексная переменная i_max устанавливается в значение i+1 или в значение 224 (если i+1 больше, чем 224), и итеративный поиск таблицы 546 отменяется. Однако, если значение состояния s отличается от значения состояния, описываемого более значимыми 24 битами таблицы "j=ari_gs_hash [i]", итеративный поиск таблицы 546 повторяется с вновь установленным интервалом таблицы, определяемым обновленными значениями индекса i_min и i_max, пока интервал таблицы не будет слишком мал (i_max - i_min<1). Таким образом, длительность интервала интервала таблицы (определяемого i_min и i_max) итеративно уменьшается, пока "прямое попадание" не будет обнаружено (s=(j»8)), или интервал достигнет минимально допустимую длительность (i_max - i_min<1). Наконец, после прекращения итеративного поиска таблицы 546 определяется запись таблицы "j= ari_gs_hash [i_max]" и значение индекса правила отображения, описываемое менее значимыми 8 битами записи указанной таблицы "j=ari_gs_hash [i_max]", возвращается в качестве возвращаемого значения функции "get_pk ". Таким образом, значение индекса правила отображения определяется в зависимости от верхней границы i_max интервала таблицы (определяемого i_min и i_max) после завершения или отмены итеративного поиска таблицы 546.

Описанные выше оценочные таблицы 540, 544, которые обе используют итеративный поиск таблиц 542, 546, позволяют проверить таблицы "ari_s_hash" и "ari_gs_hash" на наличие данного значимого состояния с очень высокой вычислительной эффективностью. В частности, количество операций доступа к таблице может оставаться умеренно небольшим даже в худшем случае. Было установлено, что числовая упорядоченность таблицы "ari_s_hash" и "ari_gs_hash" позволяет ускорить поиск соответствующего хэш-значения. Кроме того, размер таблицы может оставаться небольшим, так как включение управляющих символов в таблицах "ari_s_hash" и "ari_gs_hash" не требуется. Таким образом, устанавливается эффективный механизм контекстного хэширования, хотя существует большое количество различных состояний: На первом этапе (первая оценочная таблица 540) ведется поиск прямого попадания (s= (j>>8)).

На втором этапе (вторая оценочная таблица 544) диапазоны значения состояния s можно отобразить на значения индекса правила отображения. Таким образом, может выполняться хорошо отрегулированная обработка особенно значимых состояний, для которых существует соответствующая запись в таблице "ari_s_hash", и менее значимых

состояний, для которых существует поэтапная обработка. Таким образом, функция "get_pk" представляет собой эффективное выполнение выбора правила отображения.

За более подробной информацией сделана ссылка на псевдо программный код на фиг.5d, который показывает функциональность функции "get_pk" в представлении в соответствии с известным языком программирования C.

6.5.2 Выбор правила отображения с помощью алгоритма в соответствии с Фиг.5e

Далее будет описан другой алгоритм для выбора правила отображения, показанный на фиг.5e. Следует отметить, что алгоритм "arith_get_pk" на фиг.5e получает, в качестве входной переменной, значение состояния s, описывающее состояния контекста. Функция "arith_get_j)k» предоставляет, в качестве выходного значения, или возвращаемого значения, индекс "pki" вероятностной модели, которая может быть индексом для выбора правила отображения (например, сводная таблица частот).

Следует отметить, что функция "arith_get_pk" на фиг.5e может заменить функциональность функции "arith_get_pk" функции "value_decode"На фиг.3.

Следует также отметить, что функция "arith_get_pk" может, например, оценить таблицу ari_s_hash в соответствии с фиг.20 и таблицу ari_gs_hash в соответствии с на фиг.18.

Функция "arith_get_pk" на фиг.5e состоит из первой оценочной таблицы 550 и второй оценочной таблицы 560. В первой оценочной таблице 550 проводится линейное сканирование с помощью таблицы ari_s_hash, чтобы получить запись j=ari_s_hash [i] указанной таблицы. Если значение состояния, описываемое более значимыми 24 битами записи таблицы j=ari_s_hash[i] таблицы ari_s_hash, равно значению состояния s, значение индекса правила отображения „pki", описываемое менее значимыми 8 битами указанной выявленной таблицы, запись j=ari_s_hash[i] возвращается, и функция "arith_get_pk" отменяется. Соответственно, все 387 записи в таблице ari_s_hash оцениваются в возрастающем порядке, пока не идентифицируется "прямое попадание" (значение состояния s, равное значению состояния, описанному более значимыми 24 битами записи таблицы]).

Если прямое попадание не идентифицируется в первой оценочной таблице 550, выполняется вторая оценочная таблица 560. В ходе второй оценочной таблицы выполняется линейное сканирование с индексами записи i, увеличивающееся линейно от 0 до максимального значения 224. Во второй оценочной таблице запись "ari_gs_hash [i]" таблицы "ari_gs_hash" для таблицы i прочитывается, и запись таблицы "j=ari_gs_hash [i]" оценивается таким образом, что определяется является ли значение состояния, определяемое более значимыми 24 битами записи таблицы j, большим, чем значение состояния s. В этом случае значение индекса правила отображения, описанное менее значимыми 8 битами записи указанной таблицы j, возвращается в качестве возвращаемого значения функции "arith_get_pk", а выполнение функции "arith_get_pk" отменяется. Если, однако, значение состояния s не меньше значения состояния, описанного более значимым числом 24 бит текущей записи таблицы j=ari_gs_hash[i], сканирование записей таблицы ari_gs_hash продолжается, увеличивая индекс таблицы i. Если, однако, значение состояния s больше или равно любому из значений состояния, описанных записями таблицы ari_gs_hash, значение индекса правила отображения „pki", определенное менее значимыми 8 битами последней записи таблицы ari_gs_hash, возвращается в качестве возвращаемого значения функции "arith_get_pk".

Итак, функция "arith_get_pk" соответственно фиг.5e выполняет двушаговое хэширование. На первом этапе выполняется поиск прямого попадания, при этом определяется равно ли значение состояния s значению состояния, определенному любыми записями первой таблицы "ari_s_hash". Если прямое попадание

идентифицируется в первой оценочной таблице 550, возвращаемое значение получается из первой таблицы "ari_s_hash", и функция "arith_get_pk" отменяется. Однако, если прямое попадание не идентифицировано в первой оценочной таблице 550, выполняется вторая оценочная таблица 560. Во второй оценочной таблице выполняется оценка диапазона. Последующие записи второй таблицы "ari_gs_hash" определяют диапазоны. Если будет установлено, что значение состояния s лежит в пределах такого диапазона (о чем свидетельствует тот факт, что значение состояния, описанное более значимыми 24 битами текущей записи таблицы "j=ari_gs_hash[i]", больше значения состояния s, значение индекса правила отображения "pki", описанное менее значимыми 8 битами записи таблицы j=ari_gs_hash[i] возвращается.

6.5.3 Выбор правила отображения с помощью алгоритма в соответствии с Onr.5f

Функция "get_pk" на фиг.5f в основном эквивалентна функции "arith_get__pk" на фиг.5e. Поэтому сделана ссылка на вышеизложенное пояснение. Для более детальной информации сделана ссылка на псевдо программное представление на фиг.5f.

Следует отметить, что функция "get_pk" на фиг.5f может заменить функцию "arith_get_pk", вызванную в функции "value_decode" на фиг.3.

6.6. Функция "arith decode 0"на фиг.5g

Далее будет подробно объяснена функциональность функции "arith_decode 0" в соответствии с фиг.5g. Следует отметить, что функция "arith_decode 0" использует вспомогательную функцию "arith_first_symbol (void)", которая возвращает TRUE, если это первый символ последовательности и FALSE, если не первый. Функция "arith_decode ()" также использует вспомогательную функцию "arith_get_next_bit (void)", которая получает и предоставляет следующий бит битового потока.

Кроме того, функция "arith_decode 0" использует глобальные переменные "low", "high" и "value". Кроме того, функция "arith_decode ()" получает в качестве входной переменной, переменную "cum_freq []", которая указывает на первую запись или элемент (имеющий индекс элемента или индекс записи 0) выбранной сводной таблицы частоты. Кроме того, функция "arith_decode ()" использует входную переменную "cfl", которая указывает на длину выбранной сводной таблицы частот, обозначенной переменной "cum_freq [] ".

Функция "arith_decode 0" включает в себя в качестве первого этапа инициализацию переменной 570a, которая выполняется, если вспомогательная функция "arith_first_symbol 0"показывает, что первый символ последовательности символов декодируется. Инициализация значения 550a инициализирует переменную "value" в зависимости от множества, например, 20 бит, которые получаются из битового потока, используя вспомогательную функцию "arith_get_next_bit", так, что переменная "value" имеет значение, представленное указанным числом бит. Кроме того, переменная "low" инициализируется, чтобы принять значение 0, а переменная "high" инициализируется, чтобы принять значение 1048575.

На втором этапе 570b переменная " range" устанавливается в значение, которое больше на 1, чем разница между значениями переменных "high" и "low". Переменная "cum" устанавливается в значение, которое представляет собой относительное положение значения переменной "value" между значением переменной "low" и значением переменной "high". Таким образом, например, переменная "cum" принимает значение от 0 до 2^{16} в зависимости от значения переменной "value".

Указатель p инициализируется в значение, которое меньше на 1, чем начальный адрес выбранной сводной таблицы частот.

Алгоритм "arith_decode ()" также включает в себя итеративный поиск сводной таблицы

частот 570c. Итеративный поиск сводной таблицы частот повторяется, пока переменная cfl меньше или равна 1. В итеративном поиске сводной таблицы частот 570c указатель переменной q устанавливается в значение, равное сумме текущего значения указателя переменной p и половине значения переменной "cfl". Если значение записи *q выбранной сводной таблицы частот, запись которой адресована указателем переменной q, больше, чем значение переменной "cum", указатель переменной p устанавливается в значение указателя переменной q, и переменная "cfl" увеличивается. Наконец, переменная "cfl" смещается вправо на один бит, тем самым фактически разделяя значение переменной "cfl" на 2 и пренебрегая частью модуля.

Таким образом, итеративный поиск сводной таблицы частот 570 с фактически сравнивает значение переменной "cum" и множество записей выбранной сводной таблицы частот, чтобы определить интервал выбранной сводной таблицы частот, который ограничен записями сводной таблицы частот, так что значение cum находится в пределах выявленного интервала. Соответственно, записи выбранной сводной таблицы частот определяют интервалы, в которых соответствующие значения символа связанны с каждым из интервалов выбранной сводной таблицы частот. Кроме того, ширины интервалов между двумя смежными значениями сводной таблицы частот определяет вероятности символов, связанных с указанными интервалами, так что выбранная сводная таблица частот в целом определяет вероятность распределения разных символов (или значений символов). Подробнее о доступных сводных таблицах частот будет рассказано ниже, см. на фиг.19.

Возвращаясь к фиг.5g, значение символа получено из значения переменной указателя p, в котором значение символа извлекается как показано на фиг.570d. Таким образом, разница между значением переменной указателя p и начальным адресом "cum_freq" оценивается для того, чтобы получить значение символа, которое представлено переменной "symbol".

Алгоритм "arith_decode" также включает в себя адаптацию 570e переменных "high" и "low". Если значение символа представлено переменной "symbol" отличается от 0, переменная "high" обновляется, как показано на фиг.570e. Кроме того, значение переменной "low" обновляется, как показано на ссылке с номером 570e. Переменная "high" устанавливается в значение, которое определяется значением переменной "low", переменная "range" и запись с индексом "symbol -1" выбранной сводной таблицы частот. Переменная "low" увеличивается, причем величина роста определяется переменной "range" и записью выбранной сводной таблицы частот с индексом "symbol". Соответственно, разница между значениями переменных "low" и "high" регулируется в зависимости от числовой разницы между двумя смежными записями выбранной сводной таблицы частот.

Соответственно, если значение символа, имеющее низкую вероятность, обнаружено, интервал между значениями переменных "low" и "high" сводится к малой ширине. Напротив, если обнаруженное значение символа содержит сравнительно большую вероятность, ширина интервала между значениями переменных "low" и "high" устанавливается в сравнительно большое значение. Опять ширина интервала между значениями переменных "low" и "high"зависит от обнаруженного символа и соответствующих записей сводной таблицы частот.

Алгоритм "arith_decode 0" также включает перенормировка интервала 570f, в котором интервал, определенный на шаге 570e, итеративно изменяется и масштабируется, пока условие "break" не будет достигнуто. В перенормировке интервала 570f выполняется выборочная операция 570fa сдвига вниз. Если переменная "high" меньше, чем 524286,

ничего не делается, а перенормировка интервала продолжает операцию увеличения размера интервала 570fb. Однако, если переменная "high" не меньше 524286, а переменная "low" больше или равна 524 286, переменные "values", "low" и "high" сокращаются на 524 286, так что интервал, определенный переменными "low" and "high", смещается вниз, и так, что значение переменной "value" также сдвигается вниз. Однако, если будет установлено, что значение переменной "high" не меньше 524286, а также, что переменная "low" не превышает или равна 524286, а также, что переменная "low" больше или равна 262143 и, что переменная "high" меньше, чем 786429, переменные "value", "low" и "high" сокращаются 262143, таким образом, смещая вниз интервал между значениями переменных "high" и "low", а также значение переменной "value". Если, однако, ни одно из указанных выше условий не выполняется, перенормировка интервала отменяется.

Однако, если любое из вышеуказанных условий, которые оцениваются в шаге 570fa, выполняется, операция увеличения интервала 570fb выполняется. В операции увеличения интервала 570fb значение переменной "low" удваивается. Кроме того, значение переменной "high" удваивается, и результат удвоения увеличивается на 1. Кроме того, удваивается значение переменной "уа1ие"(сдвигается влево на один бит), и бит битового потока, который получен вспомогательной функцией "arith_get_next_bit", используется как наименее значимый бит. Соответственно, размер интервала между значениями переменных "low" и "high" приблизительно удваивается, и точность переменной "value" увеличивается за счет нового бита битового потока. Как уже упоминалось выше, шаги 570fa и 570fb повторяются, пока не выполнится условие "break", то есть, пока интервал между значениями переменных "1ow"h "high" достаточно велик.

Что касается функциональности алгоритма "arith_decode 0"» следует отметить, что интервал между значениями переменных "low" и "high" сокращается на шаге 570e в зависимости от двух смежных записей сводной таблицы частот, на которую ссылается переменная "cum_freq". Если интервал между двумя смежными значениями выбранной сводной таблицы частот маленький, то есть, если смежные значения сравнительно близки друг к другу, интервал между значениями переменных "low" и "high", которые получается в шаге 570e, будет сравнительно небольшой. С другой стороны, если две смежные записи сводной таблицы частот расположены дальше, интервал между значениями переменных "low" и "high", который получается в шаге 570e, будет сравнительно большим.

Следовательно, если интервал между значениями переменных "low" и "high", который получается в шаге 570e, сравнительно невелик, будет выполнено большое количество шагов перенормировки интервала, чтобы перемасштабировать интервал к достаточному размеру (так, что ни одно из условий 570fa оценки условий не выполняется). Таким образом, сравнительно большое количество бит битового потока будет использовано для того, чтобы повысить точность переменной "value". Если, напротив, размер интервала, полученного в шаге 570e, является сравнительно большим, потребуется только меньшее количество повторений шагов перенормировки интервала 570fa и 570fb, чтобы перенормировать интервал между значениями переменных "low" и "high" до «достаточного» размера. Соответственно, будет использоваться лишь сравнительно небольшое количество бит битового потока, чтобы увеличить точность переменной "value" и подготовить декодирование следующего символа.

Подводя итог вышесказанному, если символ декодирован, который содержит сравнительно высокую вероятность, и с которым связан большой интервал записей выбранной сводной таблицы частот, лишь сравнительно небольшое количество бит будет считано из битового потока, с тем чтобы обеспечить декодирование последующих

символов. С другой стороны, если символ декодирован, который содержит сравнительно небольшую вероятность, и с которым связан малый интервал записей выбранной сводной таблицы частот, из битового потока будет взято сравнительно большое количество бит, чтобы подготовить декодирование следующего символа.

Таким образом, записи сводных таблиц частот отражают вероятности разных символов, а также отражает количество бит, необходимых для декодирования последовательности символов. Изменяя сводную таблицу частот в зависимости от контекста, т.е. в зависимости от ранее декодированных символов (или спектральных значений), например, путем выбора разных сводных таблиц частот в зависимости от контекста, могут быть использованы стохастические зависимости между разными символами, что обеспечит особенно битрейт-эффективное кодирование последующих (или смежных) символам.

Подводя итог вышесказанному, функция "arith_decode ()" которая была описана, ссылаясь на фиг.5g, вызывается сводной таблицей частот стол "arith_cf_m[pki][]", соответственно индексу "pki", возвращаемому функцией " arith_get_pk () ", чтобы определить значение наиболее' значимого бита плоскости m (которое может быть установлено в значение символа, представляемого возвращаемой переменной "symbol").

6.7 Механизм перехода

Хотя декодированное значение m наиболее значимого бита плоскости (которое возвращается как значение символа функцией "arith_decode ()" является символом перехода "ARITH_ESCAPE", дополнительное значение m наиболее значимого бита плоскости декодируется, и переменная "lev" увеличивается на 1. Таким образом, получается информация о числовой значимости значения m наиболее значимого бита плоскости, а также о количестве менее значимых бит плоскости для декодирования.

Если символ перехода "APJTH_ESCAPE" декодируется, переменная уровня "lev" увеличивается на 1. Соответственно, значение состояния, которое заложено в функцию "arith_get_pk", также изменяется, так что' значение, представленное самыми высшими битами (биты 24 и выше), увеличивается для следующих итераций алгоритма 312ba.

6.8 Обновление контекста в соответствии с фиг.5h

После того как спектральное значение полностью декодировано (т.е. добавлены все наименее значимые биты плоскости), обновляются контекстные таблицы q и qs, вызывая функцию "arith_update_context(a,i,lg))". Далее будет подробно описана функция "arith_update_context(a,i,lg)", ссылаясь на фиг.5h, которая показывает псевдо программный код представления указанной функции.

Функция "arith_update_context 0" получает в качестве входных переменных декодированный квантованный спектральный коэффициент a, индекс i спектрального значения для декодирования (или декодированное спектральное значение), и количество lg спектральных значений (или коэффициентов), связанных с текущим аудио кадром.

В шаге 580 текущее декодированное квантованное спектральное значение (или коэффициент) a копируется в контекстную таблицу или контекстный массив q. Таким образом, запись q[1][i] контекстной таблицы q установлена в a. Кроме того, переменная "a0" установлена в значение "a".

В шаге 582 определяется значение уровня q[1][i].l контекстной таблицы q. По умолчанию, значение уровня q[1][i].l контекстной таблицы q равно нулю. Однако, если абсолютное значение текущего кодированного спектрального значения больше 4, значение уровня q[1][i].l увеличивается. С каждым увеличением переменная "a" смещается вправо на один бит. Увеличение значения уровня q[1][i].l повторяется, пока абсолютное значение переменной a0 меньше или равно 4.

В шаге 584 устанавливается 2-битное контекстное значение q[1][i].c контекстной таблицы q. 2-битное контекстное значение q[1][i].c устанавливается в значение 0, если текущее декодированное спектральное значение равно нулю. В противном случае, если абсолютное значение декодированного спектрального значения a меньше или равно 1, 2-битное контекстное значение q[1][i].c устанавливается в значение 1. Или, если абсолютное значение текущего декодированного спектрального значения a меньше или равно 3, 2-битное контекстное значение q[1][i].c устанавливается в значение 2. Или, если, например, абсолютное значение текущего декодированного спектрального значения a больше 3, то 2-битное контекстное значение q[1][i].c устанавливается в значение 3. Таким образом, 2-битное контекстное значение q[1][i].c получается именно с помощью крупно-модульного квантования текущего декодированного спектрального коэффициента a.

В следующем шаге 586, который производится, только если индекс i текущего декодированного спектрального значения равен количеству lg коэффициентов (спектральных значений) в кадре, то есть, если последнее спектральное значение кадра уже декодировано), и основной режим является основным режимом линейно предсказанной области, (которая указывается в "core_mode=l"), записи q[1][j].c копируются в контекстную таблицу qs[k]. Копирование выполняется как показано на ссылке с номером 586, так, что количество lg спектральных значений в текущем кадре учитывается для копирования записей q[1][j].c в контекстную таблицу qs[k]. Кроме того, переменная "previous_lg" принимает значение 1024.

Или, однако, записи q[1][j]-c контекстной таблицы q копируются в контекстную таблицу qs[j], если индекс i текущего декодированного спектрального коэффициента достигает значения lg, и основной режим является основным режимом с чвстотной областью (как указано в "core_mode=0").

В этом случае переменная "previous_lg" устанавливается на минимум между значением 1024 и количеством lg спектральных значений в кадре.

6.9 Обобщение процесса декодирования

Далее кратко описывается процесс декодирования. За более подробной информацией обратитесь к вышеизложенному описанию, а также к фиг.3, 4 и 5a до 5i.

Квантованные спектральные коэффициенты a бесшумно кодируются и передаются, начиная с самого низкого частотного коэффициента и увеличиваясь до самого высокого частотного коэффициента.

Коэффициенты перспективного звукового кодирования (AAC) хранятся в массиве "x_ac_quant[g][win][sfb][bin]", и порядок передачи кодовых слов бесшумного кодирования таков, что, когда они декодируются в порядок получения и хранения в массиве, ячейка является самым наиболее быстро увеличивающимся индексом, a g самым медленно увеличивающимся индексом. Индекс ячейки означает ячейки частоты. Индекс "sfb" обозначает полосы коэффициента масштабирования. Индекс "win" обозначает окна. Индекс "g" обозначает аудио кадр.

Коэффициенты преобразования кодированного возбуждения хранятся непосредственно в массиве "x_tcx_invquant[win][bin]", и порядок передачи кодовых слов бесшумного кодирования таков, что, когда они декодируются в порядке получения и хранения в массиве, "bin" является самым наиболее быстро увеличивающимся индексом, и "win" является самым медленно увеличивающимся индексом.

Во-первых, отображение осуществляется между сохраненным прошлым контекстом в контекстной таблице или массиве "qs" и контекстом текущего кадра q (хранится в контекстной таблице или массиве q). Прошлый контекст "qs" хранится в 2 битах на

линию частоты (или на ячейку частоты).

Отображение между сохраненным прошлым контекстом в контекстной таблице "qs" и контекстом текущего кадра в контекстной таблице "q" выполняется с помощью функции "arith_map_context()", представление псевдо программного кода которой показано на фиг.5a.

Бесшумный декодер выводит подписанные квантованные спектральные коэффициенты "a".

Сначала состояние контекста рассчитывается на основе ранее декодированных спектральных коэффициентов, окружающих квантованные спектральные коэффициенты для декодирования. Состояние контекста s соответствует 24 первым битам значения, возвращаемого функцией "arith_get_context()". Биты после 24 го бита возвращаемого значения соответствуют прогнозируемому уровню битовой плоскости lev0. Переменная „lev" установлена в исходное значение lev0. Представление псевдо программного кода функции "arith_get_context" показано на фиг.5b и 5 c

Если состояние s и предсказанный уровень "levO" известны, наиболее значимая 2-битная плоскость m декодируется с помощью функции "arith_decode()", подкрепленной соответствующей сводной таблицей частот, соответствующей вероятностной модели, соответствующей контекстному состоянию.

Соответствие осуществляется функцией "arith_get_pk ()".

Представление псевдо программного кода функции "arith_get_pk ()" показано на фиг.5e.

Псевдо программный код другой функции "get_pk", которая может заменить функцию "arith_get_pk ()", показан на фиг.5f. Псевдо программный код другой функции "get_pk", которая может заменить функцию "arith_get_pk ()", показан на фиг.5d.

Значение m декодируется с помощью функции "arith_decode 0"» вызванной сводной таблицей частот, "arith_cf_m[pki][], где „pki" соответствует индексу, возвращаемому функцией "arith_get_pk0" (или же функцией "get_pk ()").

Арифметический кодер является целочисленным осуществлением с помощью способа генерации тэга с масштабированием (см., например, K. Sayood "Introduction to Data Compression" third edition, 2006, Elsevier Inc.) Псевдо-C код, изображенный на фиг.5g, описывает используемый алгоритм.

Когда декодированное значение m является символом перехода, "ARITH_ESCAPE", другое значение m декодируется, и переменная „lev" увеличивается на 1. Если значение m не является символом перехода, "ARITH_ESCAPE", оставшиеся битовые плоскости затем декодируются от самого значимого до наименее значимого уровня, вызывая „lev" раз функцию "arith_decode 0" с сводной таблицей частот "arith_cf_r []". Указанная сводная таблица частот ("arith_cf_r [] может, например, описывать равномерное распределение вероятностей..

Декодированные биты плоскости r обеспечивают уточнение ранее декодированного значения m следующим способом:

```
a = m;
for (i=0; i<lev;i++) {
        r = arith_decode (arith_cf_r,2);
        a = (a<<1) | (r&1);
}
```

Если спектральный квантованный коэффициент a полностью декодирован,

контекстная таблица q, или сохраненный контекст qs обновляется функцией "arith_update_context()" для декодирования следующих квантованных спектральных коэффициентов.

Представление псевдо программного кода функции "arith__update_context 0" показано на фиг.5h.

Кроме того, условные обозначения определений показаны на фиг.5i.

7. Таблицы отображения

В одном из вариантов осуществления изобретения особенно эффективные таблицы "ari_s_hash" и "ari_gs_hash" и "ari_cf_m" используются для выполнения функции "get_pk", которая описывалась со ссылкой на фиг.5d, или для выполнения функции "arith_get_pk", которая описывалась со ссылкой на фиг.5e, или для выполнения функции "get_pk", которая описывалась со ссылкой на фиг.5f и для выполнения функции "arith_decode", которая описывалась со ссылкой на фиг.5g.

7.1. Таблица "ari s hash [3871" в соответствии с фиг.17

Содержание особо эффективного применения таблицы "ari_s_hash", которая используется функцией "get_pk", описанной со ссылкой на фиг.5d, показано в таблице на фиг.17. Следует отметить, что таблица на фиг.17 содержит 387 записей таблицы "ari_s_hash [387]". Следует также отметить, что табличное представление на фиг.17 показывает элементы в порядке индексов элементов, так, что первое значение "0x00000200" соответствует записи таблицы "ari_s_hash [0]", имеющей индекс элемента (или табличный индекс) 0, так, что последнее значение "0x03D0713D" соответствует записи таблицы "ari_s_hash [386]", имеющей индекс элемента или табличный индекс 386. Далее следует отметить, что "Ox" означает, что записи таблицы в таблице "ari_s_hash" представлены в шестнадцатеричном формате. Кроме того, записи таблицы в таблице "ari_s_hash" на фиг.17 расположены по числовому порядку, чтобы обеспечить выполнение первой оценочной таблицей 540 функции "get_pk".

Следует также отметить, что наиболее значимые 24 бит записи таблицы в таблице "ari_s_hash" представляет значения состояния, а наименее значимые 8 бит представляют собой значения индекса правила отображения pki.

Таким образом, записи таблицы "ari_s_hash" описывают отображение "прямого попадания" значения состояния в значении индекса правила отображения "pki".

7.2 _Таблица "ari_gs_hash" в соответствии с Фиг.18

Содержание особо эффективного выполнения таблицы "ari_gs_hash" показано в таблице на фиг.18. Следует отметить, что эта таблица из таблицы 18 содержит записи таблицы "ari_gs_hash". На указанные записи ссылается индекс одномерной записи целочисленного типа (также именуемая " индекс элемента" или "индекс массива" или "табличный индекс"), которая, например, обозначается "i". Следует отметить, что таблица "ari_gs_hash", которая включает в себя всего 225 записей, хорошо подходит для использования второй оценочной таблицей 544 функции "get_pk", описанной на фиг.5d.

Следует отметить, что записи таблицы "ari_gs_hash" перечислены в порядке возрастания индекса табличного индекса i таблицы для значений табличного индекса i от нуля до 224. Термин "Ox" означает, что записи в таблице приведены в шестнадцатеричном формате. Соответственно, первая запись таблицы "0X00000401" соответствует записи таблицы Vari_gs_hash [0]" с табличным индексом 0, и последняя запись таблицы "0Xffffff3f" соответствует записи таблицы "ari_gs_hash [224]" с табличным индексом 224.

Следует также отметить, что записи таблицы упорядочены в численно восходящем

порядке, так, что записи таблицы хорошо подходят для второй оценочной таблицы 544 функции "get_pk". Наиболее значимые 24 бит записей таблицы в таблице "ari_gs_hash" описывают границы между диапазонами значений состояния, и 8 наименее значимых бит записей описывают значения индекса правила отображения "pki", связанный с диапазонами значений состояния, определенных наиболее значимыми 24 бит.

7.3 Таблица "ari_cf_m" в соответствии с Фиг.19

Фиг.19 показывает набор 64 сводных таблиц частот "ari_cf_m[pki][9]", одна из которых выбрана аудио кодером 100, 700, или аудио декодером 200, 800, например, для выполнения функции "arith_decode", то есть для декодирования значения наиболее значимой битовой плоскости. Выбранная одна из 64 сводных таблиц частот, показанная на фиг.19, выбирает функцию таблицы "cum_freq []" для выполнения функции "arith_decode ()".

Как видно из фиг.19, каждая строка представляет собой сводную таблицу частот с 9 записями. Например, первая строка 1910 представляет 9 записей сводной таблицы частот для "pki=0". Вторая строка 1912 представляет 9 записей сводной таблицы частот для "pki=l". Наконец, 64-я строка 1964 представляет 9 записей сводной таблицы частот для "pki=63". Таким образом, фиг.19 фактически представляет 64 разных сводных таблиц частоты для "pki=0" до "pki=63", где каждая из 64 сводных таблиц частот представлена одной строкой, и где каждая из указанных сводных таблиц частот включает 9 записей.

В строке (например, строке 1910, или строке1912, или строке 1964), самое левое значение описывает первую запись сводной таблицы частот, и самое правое значение описывает последнюю запись сводной таблицы частот.

Таким образом, каждая строка 1910, 1912, 1964 представления таблицы на фиг.19, представляет записи сводной таблицы частот для использования функцией "arith_decode" как на фиг.5g. Входная переменная "cum_freq []" функции "arith_decode" описывает, какая из 64 сводных таблиц частот (представлены отдельными строками из 9 записей) таблицы "ari_cf_m" должна быть использована для декодирования текущих спектральных коэффициентов.

7.4 Таблица "ari s hash" в соответствии с Фиг.20

Фиг.20 показывает альтернативную возможность для таблицы "ari_s_hash", которая может быть использована в сочетании с альтернативной функцией "arith_get_pk 0" или "get_pk 0" в соответствии с фиг.5e или 5f.

Таблица "ari_s_hash" в соответствии с фиг.20 содержит 386 записей, которые приведены на фиг.20 в порядке возрастания табличного индекса. Таким образом, первое значение таблицы "Ox0090D52E" соответствует записи таблицы "ari_s_hash [0]" с табличным индексом 0, а последняя запись таблицы "0x03D0513C" соответствует записи таблицы "ari_s_hash [386]" с табличным индексом 386.

"0x" означает, что записи таблицы представлены в шестнадцатеричном формате. Наиболее значимые 24 бит записей таблицы "ari_s_hash" описывают значимые состояния, и наименее значимые 8 бит записей таблицы "ari_s_hash" описывают значения индекса правила отображения.

Соответственно, записи таблицы "ari_s_hash" описывают отображение значимых состояний на значения индекса правила отображения "pki".

8. Оценка функционирования и преимущества

Вариант осуществления в соответствии с изобретением использует обновленные функции (или алгоритмы) и обновленный набор таблиц, как отмечалось выше, чтобы получить улучшенный выбор оптимального соотношения между сложностью вычислений, требованиями к памяти, а также эффективностью кодирования.

В общих чертах, вариант осуществления в соответствии с изобретением создает улучшенное спектральное бесшумное кодирование.

Настоящее описание характеризует вариант осуществления для CE на улучшения спектрального бесшумного кодирования спектральных коэффициентов. Предложенная схема основана на "оригинальной" схеме арифметического кодирования на основе контекста, как описано в рабочем проекте 4 проекта стандарта USAC, но существенно снижает требования к памяти (RAM, ROM), в то же время сохраняя бесшумное кодирование. Перекодирование без потери информации WD3 (т.е. выход аудио кодера, обеспечивающего битовый поток в соответствии с рабочим проектом 3 проекта стандарта USAC) является доказанным. При этом описанная схема в общем изменяема, позволяет дальнейший выбор оптимального соотношения между требованием к памяти и выполнению кодирования. Вариант осуществления в соответствии с изобретением стремится заменить спектральную бесшумную схему кодирования как использовано в рабочем проекте 4 проекта стандарта USAC.

Описанная схема арифметического кодирования основана на схеме как в эталонной модели 0 (RM0) или рабочем проекте 4 (WD4) проекта стандарта USAC. Спектральные коэффициенты, расположенные, ранее по частоте или по времени, представляют собой модель контекста. Этот контекст используется для выбора сводных таблиц частот для арифметического кодера (кодера или декодера). По сравнению с вариантом осуществления в соответствий с WD4 контекстное моделирование еще более усовершенствовано и таблицы, содержащие вероятности символа, были усовершенствованы. Число различных вероятностных моделей увеличилось с 32 до 64.

Варианты осуществления в соответствии с изобретением уменьшают размеры таблицы (запрос на данные ROM) до 900 слов длиной 32 бит или 3600 байт. Напротив, вариант осуществления в соответствии с WD4 проекта стандарта USAC требует 16894,5 слов или 76 578 байт. Статический запрос RAM снижается, в некоторых вариантах осуществления в соответствии с изобретением, с 666 слов (2664 байт) до 72 (288 байт) на основной канал кодера. В то же время, он полностью сохраняет выполнение кодирования и может даже достичь прироста приблизительно от 1,04% до 1,39%, по сравнению с общей скоростью передачи данных по всем 9 рабочим точкам. Все битовые потоки рабочего проекта 3 (WD3) могут быть перекодированы без потерь, не влияя на ограничения резервуара бит.

Предложенная схема в соответствии с вариантом осуществления изобретения изменяема: возможен гибкий выбор оптимального соотношения между требованиями памяти и выполнением кодирования. Увеличивая размеры таблиц для кодирования, прирост может быть в дальнейшем увеличен.

Далее будет приведено краткое обсуждение концепции кодирования в соответствии с WD4 проекта стандарта USAC для облегчения понимания преимуществ концепции, описанной в этом документе. В USAC WD4, схема контекстно-зависимого арифметического кодирования используется для бесшумного кодирования квантованных спектральных коэффициентов. В качестве контекста используются декодированные спектральные коэффициенты, которые были раньше по частоте и времени. В соответствии с WD4 максимальное число 16ти спектральных коэффициентов используются в качестве контекста, 12 из которых были раньше по времени. Спектральные коэффициенты, используемые для контекста и для декодирования, сгруппированы в 4-кортежи (т.е. четыре спектральных коэффициента, соседних по частоте, см. фиг.10а). Контекст сокращается и отображается на сводной таблице частот, которая затем используется для декодирования следующего 4-кортежа спектральных

коэффициентов.

Для полной схемы бесшумного схемы кодирования WD4 требуется запрос памяти (ROM) из 16894,5 слов (67578 байт). Кроме того, 666 слов (2664 байт) статической ROM на основной канал кодера требуются для хранения состояний для следующего кадра.

Табличное представление на фиг.11a описывает таблицы, используемые в схеме арифметического кодирования USAC WD4.

Общий запрос памяти полного декодера USAC WD4 оценивается в 37000 слов (148000 байт) для данных ROM без программного кода, и от 10000 до 17000 слов для статической RAM. Ясно видно, что таблицы бесшумного кодера потребляют около 45% общего запроса данных ROM. Самая крупная отдельная таблица уже потребляет 4096 слов (16384 байт).

Было установлено, что и размер сочетания всех таблиц и отдельные крупные таблицы превышают типичные размеры кэша, которые содержатся в фиксированных точечных чипах для малобюджетных портативных устройств, которые находятся в обычном диапазоне 8-32 кбайт (например, ARM9E, TIC64xx и т.д.). Это означает, что набор таблиц, вероятно, не может сохраняться в быстрых данных RAM, что позволяет быстрый произвольный доступ к данным. Это приводит к тому, что весь процесс декодирования замедляется.

Далее будет кратко описана новая предлагаемая схема.

Для преодоления проблем, упомянутых выше, предлагается заменить схему WD4 проекта стандарта USAC на улучшенную схему бесшумного кодирования. Являясь схемой контекстно-зависимого арифметического кодирования, она основана на схеме WD4 проекта стандарта USAC, но имеет модифицированную схему вывода сводных таблиц частот из контекста. Далее, вывод контекста и кодирование символа осуществляется на детализации одного спектрального коэффициента (в отличие от 4-кортежей, как в WD4 проекта стандарта USAC). В общей сложности, 7 спектральных коэффициентов используются для контекста (по крайней мере в некоторых случаях). Сокращая отображение, выбирается одна из всех 64 вероятностных моделей или сводных таблиц частот (в WD4:32).

Фиг.10b показывает графическое представление контекста для расчета состояния, используемого в предлагаемой схеме (при этом контекст, используемый для обнаружения нулевой области, не показан на фиг.10b).

Далее предлагается краткое описание, касающееся сокращения требования памяти, что может быть достигнуто с помощью предлагаемой схемы кодирования. Предлагаемая новая схема показывает общий запрос ROM 900 слов (3600 байт) (см. таблицу на фиг.11b, которая описывает таблицы, используемые в предлагаемой схеме кодирования).

По сравнению с запросом ROM схемы бесшумного кодирования в WD4 проекта стандарта USAC, запрос ROM сокращается на 15994,5 слов (64 978 байт) (см. также фиг.12a, которая показывает графическое представление запроса ROM предложенной схемы бесшумного кодирования, и бесшумной схемы кодирования в WD4 проекта стандарта USAC). Это сокращает общий запрос ROM полного декодера USAC от примерно 37000 слов до примерно 21000 слов, или более чем на 43% (см. фиг.12b, которая дает графическое представление общего декодера USAC запроса на данные ROM в соответствии с WD4 проекта стандарта USAC, а также в соответствии с настоящим предложением).

Далее, количество информации, необходимой для вывода контекста в следующем кадре (статическая RAM), также уменьшается. В соответствии с WD4, полный набор коэффициентов (максимально 1152) с разрешением 16 бит дополнительно к индексу

группы индексов на 4-кортеж разрешения 10 бит, необходимых для хранения, которое насчитывает 666 слов (2664 байт) на основной канал кодера (полный декодер USAC WD4: приблизительно от 10000 до 17000 слов).

Новая схема, которая используется в способах осуществления в соответствии с изобретением, сокращает постоянную информацию к всего 2 бит на спектральный коэффициент, который насчитывает до 72 слов (288 байт) в общем на основной канал кодера. Запрос на статическую память может быть сокращен на 594 слова(2376 байт).

Далее описываются некоторые подробности, касающиеся возможного повышения эффективности кодирования. Эффективность кодирования способа осуществления в соответствии с новым предложением сравнивалась с битовыми потоками эталонного качества в соответствии с WD3 проекта стандарта USAC. Сравнение проводилось с помощью транскодера, на основе эталонного программного декодера. Для дополнительной информации относительно сравнения бесшумного кодирования в соответствии с WD3 проекта стандарта USAC и предлагаемой схемы кодирования есть ссылка на фиг.9, которая показывает схематичное представление тестирования.

Хотя запрос памяти резко снижается в вариантах в соответствии с изобретением, при сравнении с вариантами в соответствии с WD3 или WD4 проекта стандарта USAC эффективность кодирования не только сохраняется, но и немного увеличивается. Эффективность кодирования в среднем увеличилась на 1,04% до 1,39%. Для получения дополнительной информации есть ссылка на таблицу на фиг.13a, которая показывает табличное представление среднего битрейта, произведенного кодером USAC с помощью рабочего проекта арифметического кодера и аудио кодера (например, аудио кодер USAC) в соответствии с вариантом осуществления изобретения.

Измеряя уровень заполнения резервуара бит, было показано, что предлагаемое бесшумное кодирование может без потерь перекодировать битовый поток WD3 для каждой рабочей точки. Для получения дополнительной информации есть ссылка на таблицу на фиг.13b, которая показывает табличное представление контроля резервуара бит для аудио кодера в соответствии с USAC WD3 и аудио кодера в соответствии с вариантом осуществления настоящего изобретения.

Подробная информация о среднем битрейте на режим обработки, минимальном, максимальном и среднем битрейтах на базе кадра и рабочих характеристиках в самых благоприятных/неблагоприятных условиях на базе кадра можно найти в таблицах на фиг.14, 15 и 16, где таблица на фиг.14 показывает табличное представление средних битрейтов для аудио кодера в соответствии с USAC WD3 и для аудио кодера в соответствии с вариантом осуществления настоящего изобретения, где таблица на фиг.15 показывает табличное представление минимального, максимального и среднего битрейтов аудио кодера USAC на базе кадра, где таблица на фиг.16 показывает табличное представление лучших и худших рабочих характеристик на базе кадра.

Кроме того, следует отметить, что варианты в соответствии с настоящим изобретением обеспечивает хорошую масштабируемость. Адаптируя размер таблицы, выбор оптимального соотношения между требования к памяти, вычислительной сложностью и эффективностью кодирования могут быть скорректированы в соответствии с требованиями.

9. Синтаксис битового потока

9.1. Полезная нагрузка спектрального бесшумного кодера

Далее описываются некоторые подробности, касающиеся полезной нагрузки спектрального бесшумного кодера. В некоторых вариантах есть множество различных режимов кодирования, таких как, например, так называемый режим линейного

прогнозирования области, "режим кодирования" и режим кодирования "частотной области". В режиме кодирования линейного прогнозирования области ограничение шума производится на основе анализа с линейным предсказанием аудио, и шумоподобный сигнал кодируется в частотной области. В режиме частотной области ограничение шума осуществляется на основе психоакустического анализа и шумоподобная версия аудио-контента кодируется в частотной области.

Спектральные коэффициенты кодированного сигнала "линейного предсказания области" и кодированного сигнала "частотной области" скалярно квантуются, а затем бесшумно кодируются адаптивным контекстно-зависимым арифметическим кодированием. Квантованные коэффициенты передаются от самых низких частот к самым высоким частотам. Каждый отдельный квантованный коэффициент делится на наиболее значимые 2 бит плоскости m и остальные менее значимые плоскости бит r.. Значение m кодируется в соответствии с окрестностями коэффициента Остальные менее значимые биты плоскости г энтропийно-кодируются без учета контекста. Значения гит образуют символы арифметического кодера.

Подробная процедура арифметического декодирования описана здесь.

9.2. Элементы синтаксиса

Далее описывается синтаксис битового потока битового потока, несущего арифметически кодированную спектральную информацию, со ссылкой на фиг.6a-6h.

Фиг.6a показывает синтаксическое представление так называемого блока необработанных данных USAC ("usac_raw_data_block ()")-

Блок необработанных данных USAC состоит из одного или более одноканальных элементов ("single_channel_element ()") и/или одного или более двухканальных элементов ("channel_pair_element ()").

Теперь перейдем к фиг.6b, где описывается синтаксис одноканального элемента. Одноканальный элемент состоит из потока канала линейного предсказания области ("lpd_channel_stream ()") или потока канала частотной области ("fd_channel_stream ()") в зависимости от основного способа.

Фиг.6 с показывает синтаксическое представление двухканального элемента. Двухканальный элемент включает в себя информацию об основном режиме ("core_modeO", "core_modei"). Кроме того, двухканальный элемент может включать в себя конфигурационные данные "ics_info ()". Кроме того, в зависимости от информации об основном режиме двухканальный элемент состоит из потока канала линейного предсказания области или потока канала частотной области, связанного с первым из каналов, и двухканальный элемент также включает в себя поток канала линейного предсказания области или поток канала частотной области, связанный со вторым из каналов.

Конфигурационные данные "ics_info 0", синтаксическое представление которых показано на фиг.6d, содержит множество различных элементов конфигурационных данных, которые не представляют особого интераса для настоящего изобретения.

Поток канала частотной области ("fd_channel_stream 0")» синтаксическое представление которого показано на фиг.6e, включает в себя получение информации ("global_gain") и конфигурационные данные ("ics_info 0")- Кроме того, поток канала частотной области содержит данные коэффициента масштабирования ("scale_factor_data ()"), которые описывают коэффициенты масштабироваеия, используемые для масштабирования спектральных значений разных полос коэффициентов масштабирования, и который применяется, например, масштабирующим устройством 150 и рескейлером 240. Поток канала частотной области также включает в себя

арифметически кодированные спектральные данные ("ac_spectral_data ()") который представляет арифметически кодированные спектральные значения.

Арифметически кодированные спектральные данные ("ac_spectral_data ()") синтаксическое представление которых • показано на фиг.6f, включают в себя дополнительный флаг арифметического сброса ("arith_reset_fiag"), который используется для выборочного сброса контекста, как описано выше. Кроме того, арифметически кодированные спектральные данные включают в себя множество блоков арифметических данных ("arith_data"), которые несут арифметически кодированные спектральные значения. Структура блоков арифметически кодированных данных зависит от числа частотных полос (представленных переменной "num_bands"), а также от состояния флага арифметического сброса, о чем будет рассказано далее.

Структура арифметически кодированных блоков данных будет описана со ссылкой на фиг.6g, которая показывает синтаксическое представление указанных блоков арифметически кодированных данных. Представление данных в арифметически кодированных блоках данных зависит от числа lg спектральных значений для кодирования, статуса флага арифметического сброса, а также от контекста, то есть ранее кодированных спектральных значений.

Контекст для кодирования текущего набора спектральных значений определяется в соответствии с алгоритмом определения контекста, показанным со ссылкой на номер 660. Подробности относительно алгоритма определения контекста были рассмотрены выше (фиг.5а). Блок арифметически кодированных данных включает в себя наборы lg кодовых слов, каждый набор кодовых слов представляет спектральное значение. Набор кодовых слов включает в себя арифметическое кодовое слово "acodjn [pki][m]", представляющее собой значение наиболее значимого бит плоскости m спектрального значения с помощью от 1 до 20 бит. Кроме того, набор кодовых слов включает в себя одно или больше кодовых слов "acod_r [г]", если спектральное значение требует больше битовых плоскостей, чем более значимая битовая плоскость для правильного представления. Кодовое слово "acod_r [г]» представляет собой менее значимую битовую плоскость, используя от 1 до 20 бит.

Однако, если требуется одна или больше менее значимых битовых плоскостей (в дополнение к более значимым битовым плоскостям) для правильного представления спектрального значения, то это сигнализируется с помощью одного или более арифметических кодовых слов перехода ("ARITH_ESCAPE"). Таким образом, в целом можно сказать, что для спектрального значения определяется, как много требуется битовых плоскостей (наиболее значимая бит плоскость и, возможно, одна или более дополнительных менее значимых бит плоскостей). Если требуется одна или больше менее значимых бит плоскостей, то это сигнализируется одним или более арифметическими кодовыми словами перехода "acod_m [pki][ARITH_ESCAPE]", которые кодируются в соответствии с текущей выбранной сводной таблицей частот, индекс сводной таблицы частот которой задается переменной pki. Кроме того, контекст адаптирован, как можно увидеть на ссылках 664, 662, если одно или более арифметических кодовых слов перехода включены в битовый поток. Следуя за одним или несколькими арифметическими кодовыми словами перехода, арифметическое кодовое слово "acod_m [pki][m]" включается в битовый поток, как показано на ссылке 663, где pki определяет текущий действующий индекс -вероятностной модели (учитывая адаптацию контекста, вызванную включением арифметических кодовых слов перехода), и где m обозначает значение наиболее значимой битовой плоскости спектрального значения для кодирования или декодирования.

Как уже говорилось выше, наличие любой менее значимой битовой плоскости приводит к наличию одного или более кодовых слов "acod_r [r]", каждое из которых представляет один бит наименее значимой битовой плоскости. Одно или более кодовых слов "acod_r[r]" кодируется в соответствии с соответствующей сводной таблицей частот, которая является постоянной или независимой от контекста.

Кроме того, следует отметить, что контекст обновляется после кодирования каждого спектрального значения, как показано на ссылке 668, так что контекст, как правило, различен для кодирования двух последующих спектральных значений.

Фиг.6h показывает условные обозначения определений и вспомогательных элементов, определяющих синтаксис арифметически кодированного блока данных.

Подводя итог вышесказанному, был описан формат битового потока, который может быть обеспечен аудио кодером 100, и который может быть оценен аудио декодером 200. Битовый поток арифметически кодированных спектральных значений кодируется так, что он подходит алгоритму декодирования, описанному выше.

Кроме того, в целом следует отметить, кодирование является обратной операцией декодирования, так, что в целом можно предположить, что декодер выполняет поиск в таблице, используя рассмотренные выше таблицы, что примерно обратно поиску в таблице, выполняемому декодером. Вообще, можно сказать, что специалист в данной области, который знает алгоритм декодирования и/или синтаксис желаемого битового потока, с легкостью сможет разрабатывать арифметический кодер, который обеспечивает данные, определенные в синтаксисе битового потока, и требуемые арифметическим декодером.

10. Воплощение изобретения согласно фиг.21 и фиг.22

Далее будут описаны упрощенные варианты настоящего изобретения.

На фиг.21 показана блок-схема аудио кодера 2100 согласно варианту реализации изобретения. Аудио кодер 2100 настроен получать входную аудио информацию 2110 и обеспечивать на ее основе кодированную аудио информацию 2112. Аудио кодер 2100 включает энергосберегающий конвертер из временной области в частотную область 2120, который настроен получать представление временной области 2120 входной аудио информации 2110 и обеспечивать на ее основе аудио представление частотной области 2124 так, что аудио представление частотной области включает набор спектральных значений (например, спектральных значений a). Кодер аудио сигнала 2100 также включает арифметический кодер 2130, который настроен кодировать спектральные значения 2124 или их раннее обработанную версию, используя кодовое слово переменной длины. Арифметический кодер 2130 настроен отображать спектральное значение или значение наиболее значимой битовой плоскости спектрального значения на значение кода (например, значение кода, представляющее кодовое слово переменной длины).

Арифметический кодер 2130 включает выбор правила отображения 2132 и определение значения контекста 2136. Арифметический кодер настроен выбирать правило отображения, описывающее отображение спектрального значения 2124, или наиболее значимую битовую плоскость спектрального значения 2124, на значение кода (которое может представлять кодовое слово переменной длины) в зависимости от числового значения текущего контекста, описывающее состояние контекста. Арифметический декодер настроен определять числовое значение текущего контекста 2134, которое используется для выбора правила отображения 2132 в зависимости от множества раннее кодированных спектральных значений, а также в зависимости от того, где находится кодируемое спектральное значение - в первой заданной частотной

области или во второй заданной частотной области. Соответственно отображение 2131 адаптируется к определенным характеристикам различных частотных областей.

На фиг.22 показана блок-схема декодера аудио сигнала 2200 согласно другому варианту изобретения. Декодер' аудио сигнала 2200 настроен получать кодированную аудио информацию 2210 и обеспечивать на ее основе декодированную аудио информацию 2212. Декодер аудио сигнала 2200 включает арифметический декодер 2220, который настроен получать арифметически кодированное представление 2222 спектральных значений и обеспечивать на их основе множество декодированных спектральных значений 2224 (например, декодированных спектральных значений a). Декодер аудио сигнала 2200 также включает конвертер частотной области во временную область 2230, который настроен получать декодированные спектральные значения 2224 и обеспечивать аудио представление временной области, используя декодированные спектральные значения для того, чтобы получить декодированную аудио информацию 2212.

Арифметический декодер 2220 включает отображение 2225, которое используется для того, чтобы отображать значение кода (например, значение кода, извлеченное из битового потока, представляющего кодированную аудио информацию), на код символа (код символа, который может описывать, например, декодированное спектральное значение или наиболее значимую битовую плоскость декодированного спектрального значения). Арифметический декодер далее включает выбор правила отображения 2226, которое обеспечивает информацию о выборе правила отображения 2227, которая станет отображением 2225. Арифметический декодер 2220 также включает определение значения контекста 2228, которое обеспечивает числовое значение текущего контекста 2229 для выбора правила отображения 2226. Арифметический декодер 2220 настроен выбирать правило отображения, описывающее отображение значение кода (например, значение кода, извлеченное из битового потока, представляющего кодированную аудио информацию) на код символа (например, числовое значение, представляющее декодированное спектральное значение или числовое значение, представляющее наиболее значимую битовую плоскость декодированного спектрального значения) в зависимости от состояния контекста. Арифметический декодер настроен определять числовое значение текущего контекста, описывающее состояние текущего контекста в зависимости от множества раннее декодированных спектральных значений, а также в зависимости от того, где находится декодируемое спектральное значение - в первой заданной частотной области или во второй заданной частотной области.

Соответственно при отображении 2225 учитываются различные характеристики различных частотных областей, что обычно повышает эффективность кодирования без значительного увеличения вычислительных затрат.

11. Альтернативные варианты использования

Хотя некоторые аспекты уже были описаны в контексте аппарата, ясно, что эти аспекты также представляют собой описание соответствующего способа, где блок или устройство соответствуют шагу способа или черте шага способа. Аналогично, аспекты, изложенные в контексте шага способа, также представляют собой описание соответствующего блока или элемента или черты соответствующего аппарата. Некоторые или все шаги способов могут быть выполнены (с помощью) аппаратного обеспечения, как, например, микропроцессор, программируемый компьютер или электронная схема. В некоторых вариантах один или несколько из самых важных шагов способа могут быть выполнены таким аппаратным обеспечением.

Изобретенный кодированный аудио сигнал может быть сохранен на цифровом

носителе или может быть передан с помощью передающего средства, такого как беспроводное средство передачи или проводное средство передачи, например Интернет.

В зависимости от требований определенных реализаций, воплощения изобретения могут быть реализованы в виде аппаратного обеспечения или программного обеспечения. Воплощение может быть осуществлено с помощью цифрового носителя, например дискеты, DVD, Blue-Ray, CD, ROM, PROM, EPROM, EEPROM или флэш-памяти, имеющего сохраненные на нем электронно читаемые контролирующие сигналы, которые сотрудничают (или способны работать вместе) с программируемой компьютерной системой так, что соответствующий способ выполняется. Таким образом, цифровой носитель может быть машиночитаемым.

Некоторые воплощения в соответствии с изобретением содержат носитель данных, имеющий электронно читаемые контролирующие сигналы, которые способны сотрудничать с программируемой компьютерной системой так, что выполняется одним из способов, описанных в данном документе.

Как правило, варианты осуществления настоящего изобретения могут быть реализованы в виде программного продукта с программным кодом, который задействован для осуществления одного из способов, когда компьютерный программный продукт работает на компьютере. Программный код, например, может быть сохранен на машиночитаемом носителе.

Другие варианты включают компьютерную программу для выполнения одного из способов, описанных в данном документе, хранящуюся на машиночитаемом носителе.

Иными словами, воплощением изобретенного способа, следовательно, является компьютерная программа, имеющая программный код для выполнения одного из способов, описанных в данном документе, когда компьютерная программа работает на компьютере.

Еще одним вариантом использования изобретенных способов, таким образом, является носитель информации (или цифровой носитель, или машиночитаемый носитель), включающий записанную на нем компьютерную программу для выполнения одного из способов, описанных в данном документе.

Еще одним вариантом использования изобретенного способа является, таким образом, поток данных или последовательность сигналов, представляющих компьютерную программу для выполнения одного из способов, описанных в данном документе. Поток данных или последовательность сигналов, например, может быть настроена для передачи через соединение передачи данных, например, через Интернет.

Еще один вариант использования включает в себя средства обработки, например, компьютер или программируемое логическое устройство, настроенное или адаптированное для выполнения одного из способов, описанных в данном документе.

Еще один вариант использования включает компьютер, с установленной на нем компьютерной программой для выполнения одного из способов, описанных в данном документе.

В некоторых вариантах использования программируемое логическое устройство (например, поле-программируемая вентильная матрица) может быть использовано для выполнения некоторых или всех функциональных возможностей способов, описанных в данном документе. В некоторых вариантах поле-программируемая вентильная матрица может сотрудничать с микропроцессором для выполнения одного из способов, описанных в данном документе. Как правило, способы предпочтительно выполнять с помощью аппаратных средств.

Описанные выше варианты осуществления изобретения являются только

иллюстрацией принципов данного изобретения. Подразумевается, что модификации и вариации механизмов и деталей, описанных в данном документе, будут очевидны для других специалистов в данной области. Таким образом, данный документ ограничивается только областью предстоящих патентных притязаний, а не конкретными деталями, представленными в виде описания и объяснения использования изобретения в настоящем документе.

В то время как все вышеописанное было показано и описано со ссылкой на конкретные варианты осуществления, для специалистов в данной области будет понятно, что различные изменения в форме и деталях могут быть сделаны без отступления от сущности и объема изобретения. Следует понимать, что различные изменения могут быть сделаны в процессе адаптации к различным вариантам, не отходя от более широкой концепции, описанной здесь и подтвержденной патентными притязаниями далее.

12. Заключение

В заключении можно отметить, что варианты в соответствии с изобретением создают улучшенную схему спектрального бесшумного кодирования. Варианты в соответствии с новым предложением делают возможным значительное сокращение запроса памяти с 16894,5 слов до 900 слов (ROM) и с 666 до 72 слов (статической RAM на основной канал кодера). Это создает возможность для сокращения запроса на данные ROM всей системы примерно на 43% в одном варианте. Одновременно выполнение кодирования не только полностью сохраняется, но в среднем даже увеличивается. Перекодирование без потерь WD3 (или битового потока в соответствии с WD3 проекта стандарта USAC) признано возможным. Таким образом, вариант в соответствии с изобретением получается благодаря внедрению бесшумного декодирования, описанного здесь, в предстоящий рабочий проект стандарта USAC.

Итак, в варианте изобретения предлагаемое новое бесшумное кодирование может вызвать изменения в рабочем проекте MPEG USAC относительно синтаксиса элемента битового потока "arith_data 0" (как показано на фиг.6g), относительно полезной нагрузки спектрального бесшумного кодера (как описано выше и показано на фиг.5h), относительно спектрального бесшумного кодирования как описано выше, относительно контекста для расчета состояния (как показано на фиг.4), относительно определений (как показано на фиг.5i), относительно процесса декодирования (как описано выше, ссылаясь на фиг.5a, 5b, 5 c, 5e, 5g, 5h), и относительно таблиц (как показано на фиг.17, 18, 20), и относительно функции "get_pk" (как показано на фиг.5d). Кроме того, тем не менее, таблица "ari_s_hash" в соответствии с фиг.20 может быть использована вместо таблицы "ari_s_hash" на фиг.17, а функция "get_pk" на фиг.5f может быть использована вместо функции "get_pk" в соответствии с фиг.5d..

Формула изобретения

1. Аудио декодер (200, 800, 2200) для обеспечения декодированной аудио информации (212, 812, 2212) на основе кодированной аудио информации (210, 810, 2210), аудио декодер, включающий:

арифметический декодер (230; 820, 2220) для обеспечения множества декодированных спектральных значений 232, 822, 2224 а) на основе арифметически кодированного представления (222; 821, 2222) спектральных значений; и

конвертер частотной области во временную область (260, 830, 2230) для обеспечения аудио представления временной области, используя декодированные спектральные значения (232, 822, 2224), а) для получения декодированной аудио информации;

при этом арифметический декодер настроен, чтобы выбрать правило отображения,

описывающее отображение значения кода арифметически кодированного отображения на код символа, отображающего одно или более декодированное спектральное значение или как минимум часть одного или более декодированного спектрального значения, в зависимости от состояния контекста;

при этом арифметический декодер настроен определять числовое значение текущего контекста (s), описывающее состояние текущего контекста в зависимости от множества ранее декодированных спектральных значений (a), а также в зависимости от того, где расположено декодируемое спектральное значение (a) - в первой заданной частотной области или во второй заданной частотной области.

2. Аудио декодер по п. 1, где арифметический декодер настроен, чтобы выборочно модифицировать числовое значение текущего контекста (s) в зависимости от того, где расположено декодируемое спектральное значение (a) - в первой заданной частотной области или во второй заданной частотной области.

3. Аудио декодер по п. 1, где арифметический декодер настроен определять числовое значение текущего контекста (s) таким образом, что числовое значение текущего контекста (s) основывается на комбинации множества ранее декодированных спектральных значений или на комбинации множества промежуточных значений (c0, c1, c2, c3, c4, c5, c6), производных от множества ранее декодированных спектральных значений (a), и числовое значение текущего контекста (s) выборочно превышает значение, полученное на основе множества ранее декодированных спектральных значений или на комбинации множества промежуточных значений (c0, c1, c2, c3, c4, c5, c6), производных от множества ранее декодированных спектральных значений в зависимости от того, где расположено декодируемое спектральное значение (a) - в первой заданной частотной области или во второй заданной частотной области.

4. Аудио декодер по п. 1, где арифметический декодер настроен, чтобы разграничивать по меньшей мере первую частотную область и вторую частотную область для того, чтобы определить числовое значение текущего контекста (s),

при этом первая частотная область включает по меньшей мере 15% спектральных значений, соответствующих заданной временной области аудио контента, при этом первая частотная область является диапазоном низких частот и включает спектральное значение, соответствующее самой низкой частоте.

5. Аудио декодер по п. 1, где арифметический декодер настроен, чтобы разграничивать по меньшей мере первую частотную область и вторую частотную область для того, чтобы определить числовое значение текущего контекста (s),

при этом вторая частотная область включает по меньшей мере 15% спектральных значений, соответствующих заданной временной области аудио контента, при этом вторая частотная область является диапазоном высоких частот и включает спектральное значение, соответствующее самой высокой частоте.

6. Аудио декодер по п. 1, где арифметический декодер настроен, чтобы разграничивать по меньшей мере первую частотную область, вторую частотную область и третью частотную область для того, чтобы определить числовое значение текущего контекста (s) в зависимости от того, в какой из по меньшей мере трех частотных областей находится декодируемое спектральное значение; и

при этом каждая из трех частотных областей включает множество соответствующих спектральных значений.

7. Аудио декодер по п. 6, где по меньшей мере одна восьмая часть спектральных значений заданной временной области аудио информации соответствует первой частотной области, по меньшей мере одна пятая часть спектральных значений заданной

временной области аудио информации соответствует второй частотной области, по меньшей мере одна четвертая часть спектральных значений заданной временной области аудио информации соответствует третьей частотной области.

8. Аудио декодер по п. 1, где арифметический декодер настроен вычислять сумму, включающую по меньшей мере первое слагаемое и второе слагаемое, для получения в результате суммирования числового значения текущего контекста (s),

при этом первое слагаемое получается на основе комбинации множества промежуточных значений (c0, c1, c2, c3, c4, c5, c6), описывающих величины ранее декодированных спектральных значений (a), и

при этом второе слагаемое (область) представляет, какой частотной области, из множества частотных областей, соответствует декодируемое спектральное значение.

9. Аудио декодер по п. 1, где арифметический декодер настроен модифицировать одну или более заданных битовых позиций двоичного представления числового значения текущего контекста (s) в зависимости от того, в какой частотной области из множества различных частотных областей находится декодируемое спектральное значение.

10. Аудио декодер по п. 1, где арифметический декодер настроен выбирать правило отображения в зависимости от числового значения текущего контекста (s) так, что для множества различных числовых значений текущего контекста (s) в результате выбирается одно правило отображения.

11. Аудио декодер по п. 1, где арифметический декодер настроен выполнять двухшаговый выбор правила отображения в зависимости от числового значения текущего контекста;

при этом арифметический декодер настроен контролировать, на первом шаге выбора, идентично ли числовое значение текущего контекста (s) или производное от него значение величине значимого состояния (j>> 8), описанного с помощью записи таблицы прямого попадания (ari_s_hash); и

при этом арифметический декодер настроен определять, на втором шаге выбора, который выполняется только в том случае, если числовое значение текущего контекста (s) или производное от него значение отличается от величин значимого состояния, описанных с помощью записей таблицы прямого попадания, в каком из интервалов, среди множества интервалов, находится числовое значение текущего контекста (s); и

при этом арифметический декодер настроен выбирать правило отображения в зависимости от результата первого шага выбора или второго шага выбора; и

при этом арифметический декодер настроен выбирать правило отображения в процессе первого шага выбора или второго шага выбора в зависимости от того, в какой частотной области находится декодируемое спектральное значение - в первой или второй частотной области.

12. Аудио декодер по п. 11, где арифметический декодер настроен выборочно модифицировать одну или более наименее значимых битовых частей двоичного представления числового значения текущего контекста (s) в зависимости от того, в какой частотной области из множества различных частотных областей находится декодируемое спектральное значение;

при этом арифметический декодер настроен определять, в процессе второго шага выбора, в каком интервале из множества интервалов находится двоичное представление числового значения текущего контекста (s),

выбирать отображение таким образом, что для нескольких числовых значений текущего контекста в результате выбирается одно правило отображения независимо от того, в какой частотной области находится декодируемое спектральное значение,

и

для нескольких числовых значений текущего контекста правило отображения выбирается в зависимости от того, в какой частотной области находится декодируемое спектральное значение.

13. Кодер аудио сигнала (100; 700; 2100) для обеспечения кодированной аудио информации (112; 712; 2112) на основе входной аудио информации (110; 710; 2110), аудио кодер, включающий:

энергосберегающий конвертер из временной области в частотную область (130, 720, 2120) для обеспечения аудио представления частотной области (132; 722; 2124) на основе представления временной области (110; 710; 2122) входной аудио информации так, что аудио представление частотной области включает в себя набор спектральных значений (a);

арифметический кодер (170; 730; 2130) настроен кодировать спектральные значения (a) или их обработанные ранее версии, используя кодовое слово переменной длины,

при этом арифметический кодер настроен отображать спектральное значение (a) или значение (m) наиболее значимой битовой плоскости спектрального значения (a) на значение кода, отображающего кодовое слово переменной длины,

при этом арифметический кодер настроен выбирать правило отображения, описывающее отображение спектрального значения (a), или наиболее значимой битовой плоскости (m) спектрального значения (a), на значение кода в зависимости от состояния контекста (s),

при этом арифметический кодер настроен определять числовое значение текущего контекста (s), описывающее состояние текущего контекста, в зависимости от множества ранее кодированных спектральных значений, а также в зависимости от того, в первой или второй заданной частотной области находится кодируемое спектральное значение.

14. Способ предоставления декодированной аудио информации на основе кодированной аудио информации, включающий:

предоставление множества декодированных спектральных значений на основе арифметически кодированного представления спектральных значений; и

выполнение преобразования из частотной области во временную область, чтобы обеспечить аудио представление временной области, используя декодированные спектральные значения для получения декодированной аудио информации;

при этом правило отображения, описывающее отображение значения кода арифметически кодированного отображения на код символа, отображающего одно или более декодированное спектральное значение или как минимум часть одного или более декодированного спектрального значения, выбирается в зависимости от состояния контекста; и

при этом числовое значение текущего контекста, описывающее состояние текущего контекста, определяется в зависимости от множества ранее декодированных спектральных значений, а также в зависимости от того, в первой или второй заданной частотной области находится декодируемое спектральное значение.

15. Способ предоставления кодированной аудио информации на основе входной аудио информации, включающий:

выполнение энергосберегающего преобразования из временной области в частотную область для получения аудио представления частотной области на основе представления во временной области входной аудио информации так, что аудио представление частотной области включает в себя набор спектральных значений, и

кодирование спектрального значения, или его обработанной ранее версии, используя

кодовое слово переменной длины,

при этом спектральное значение или значение наиболее значимой битовой плоскости спектрального значения отображается на значении кода, отображающего кодовое слово переменной длины;

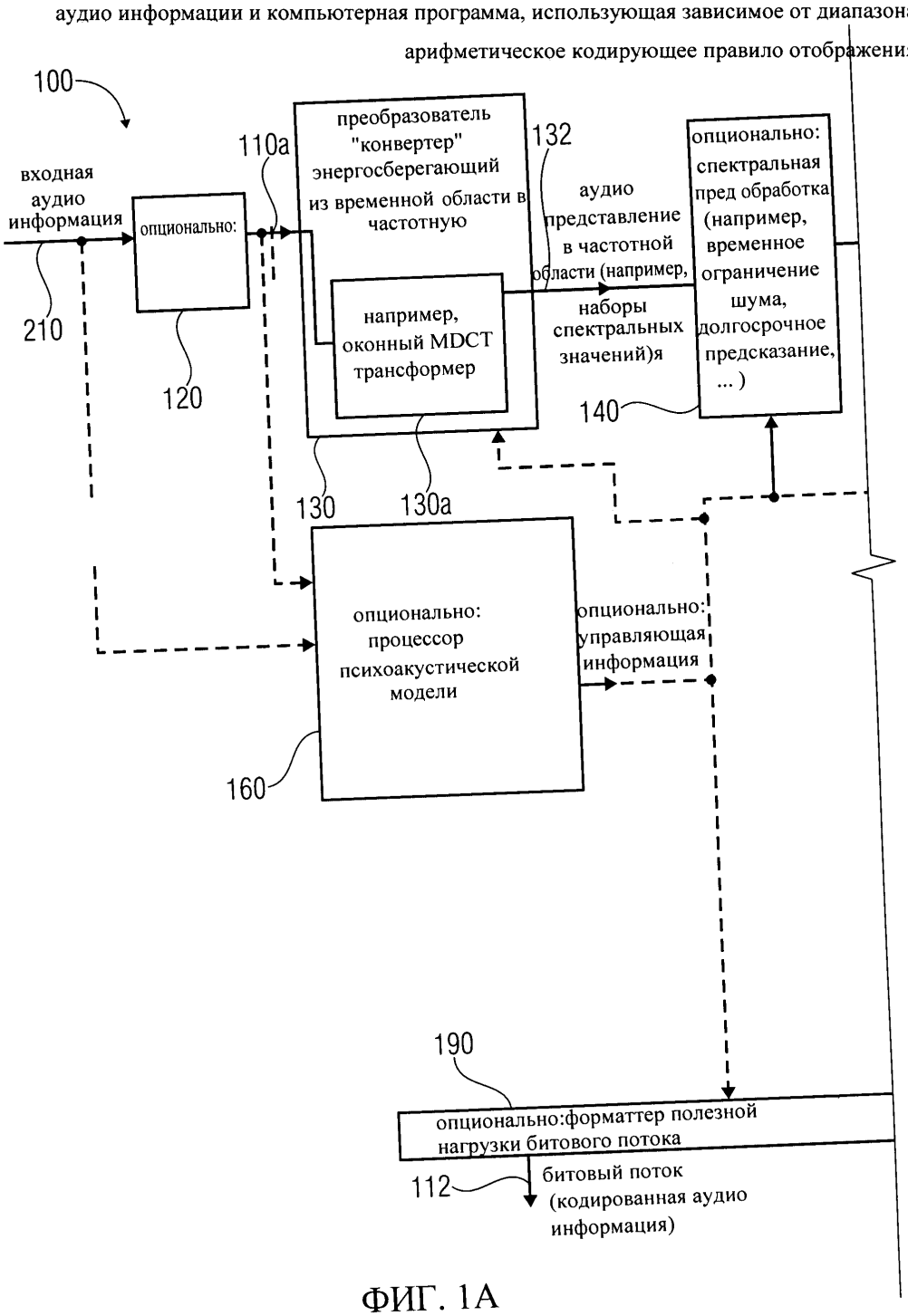
при этом правило отображения, описывающее отображение спектрального значения или наиболее значимой битовой плоскости спектрального значения, на значение кода выбирается в зависимости от состояния контекста;

при этом числовое значение текущего контекста, описывающее состояние текущего контекста, определяется в зависимости от множества ранее кодированных спектральных значений, а также в зависимости от того, в первой или второй заданной частотной области находится кодируемое спектральное значение.

16. Машиночитаемый носитель информации с записанной на него компьютерной программой для осуществления способа по п. 14, когда программа запускается на компьютере.

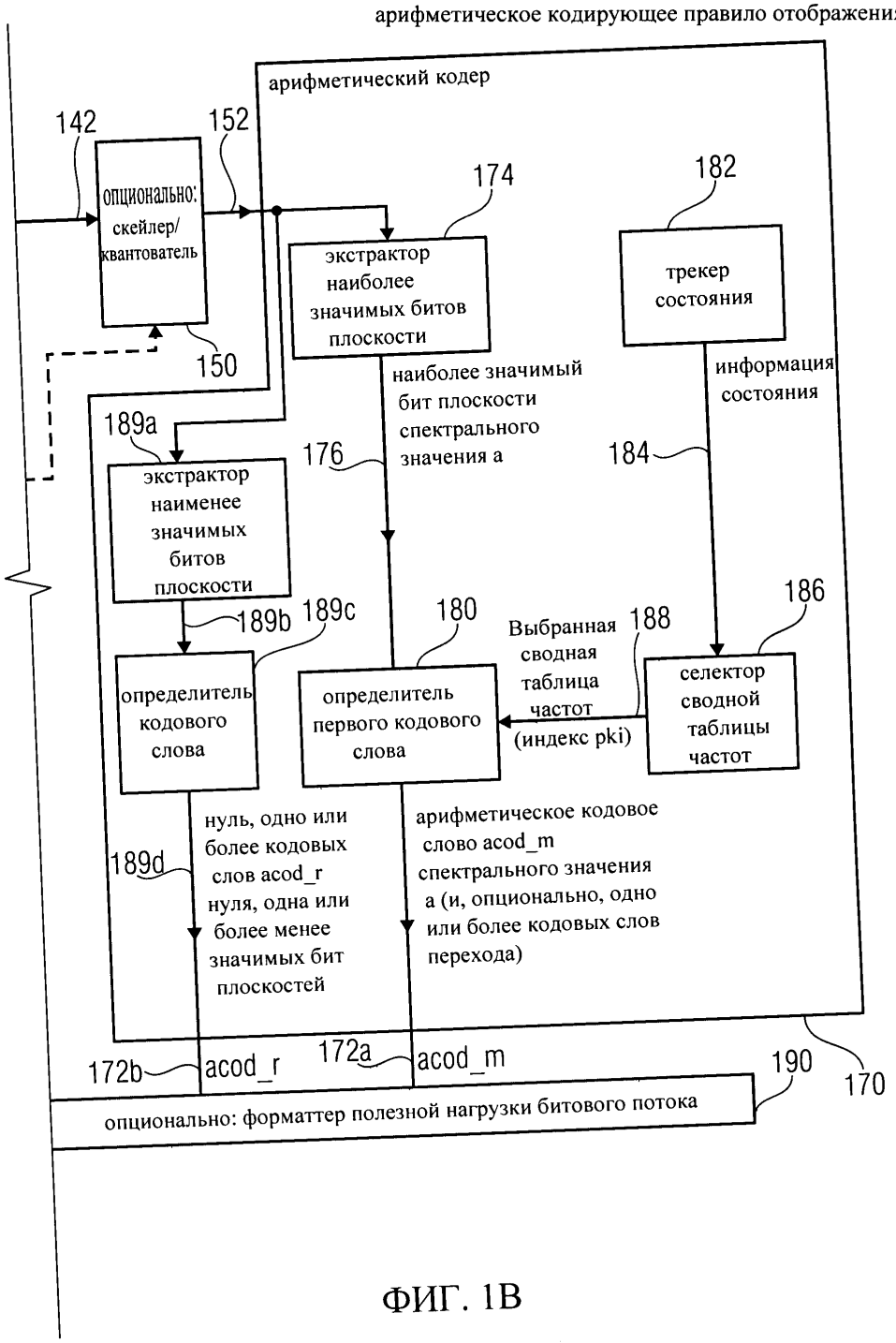
17. Машиночитаемый носитель информации с записанной на него компьютерной программой для осуществления способа по п. 15, когда программа запускается на компьютере.

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



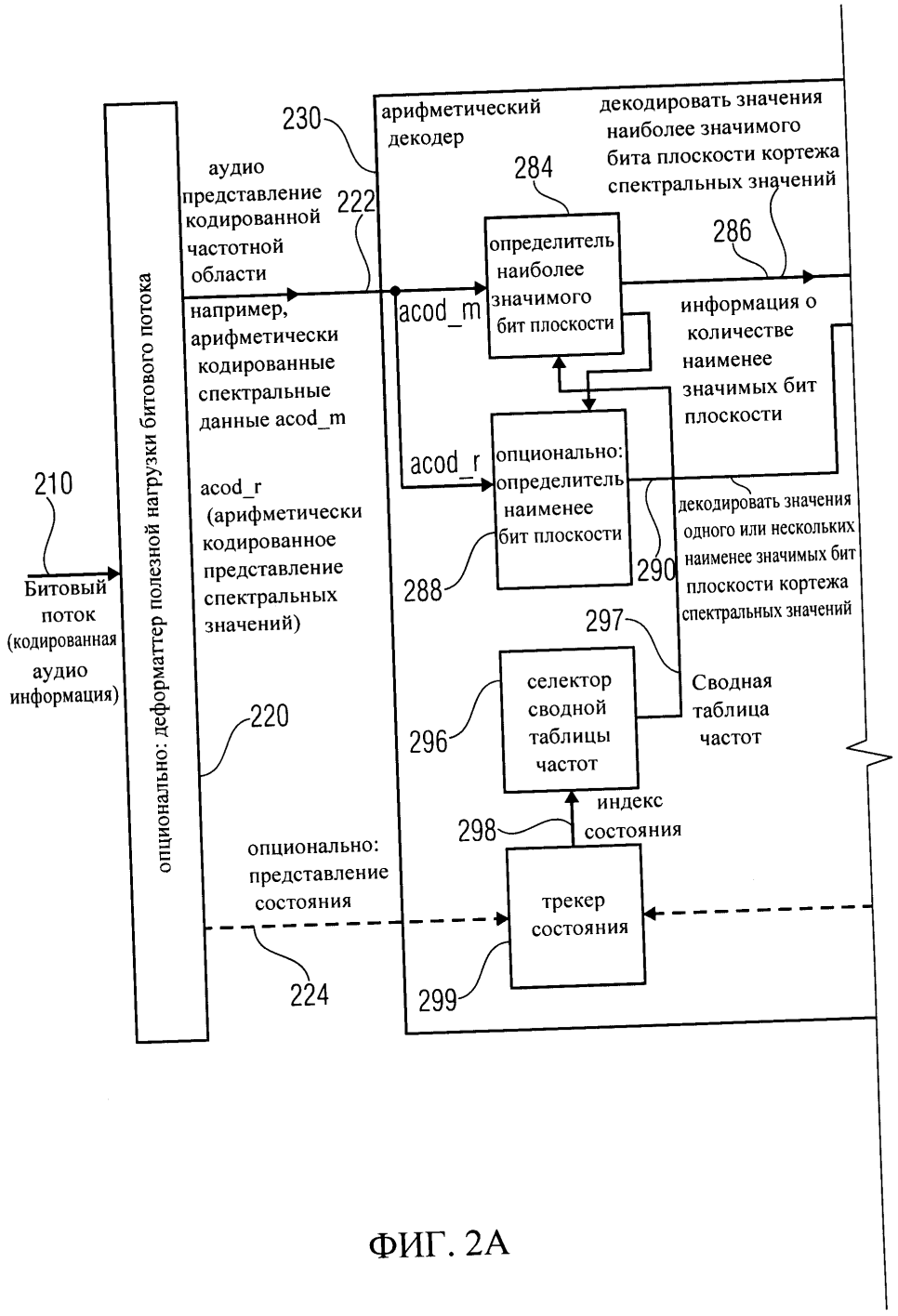
ФИГ. 1А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



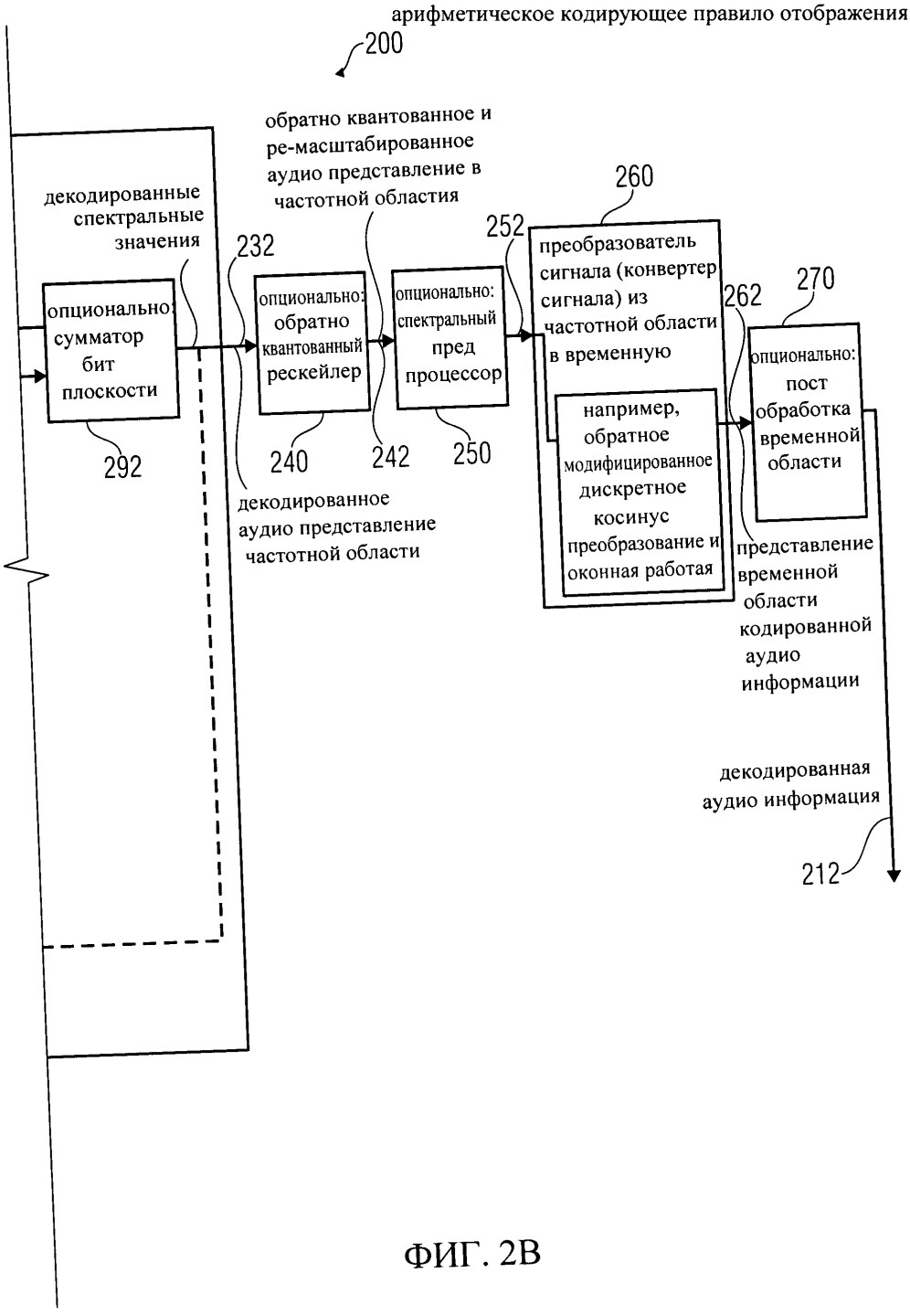
ФИГ. 1B

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



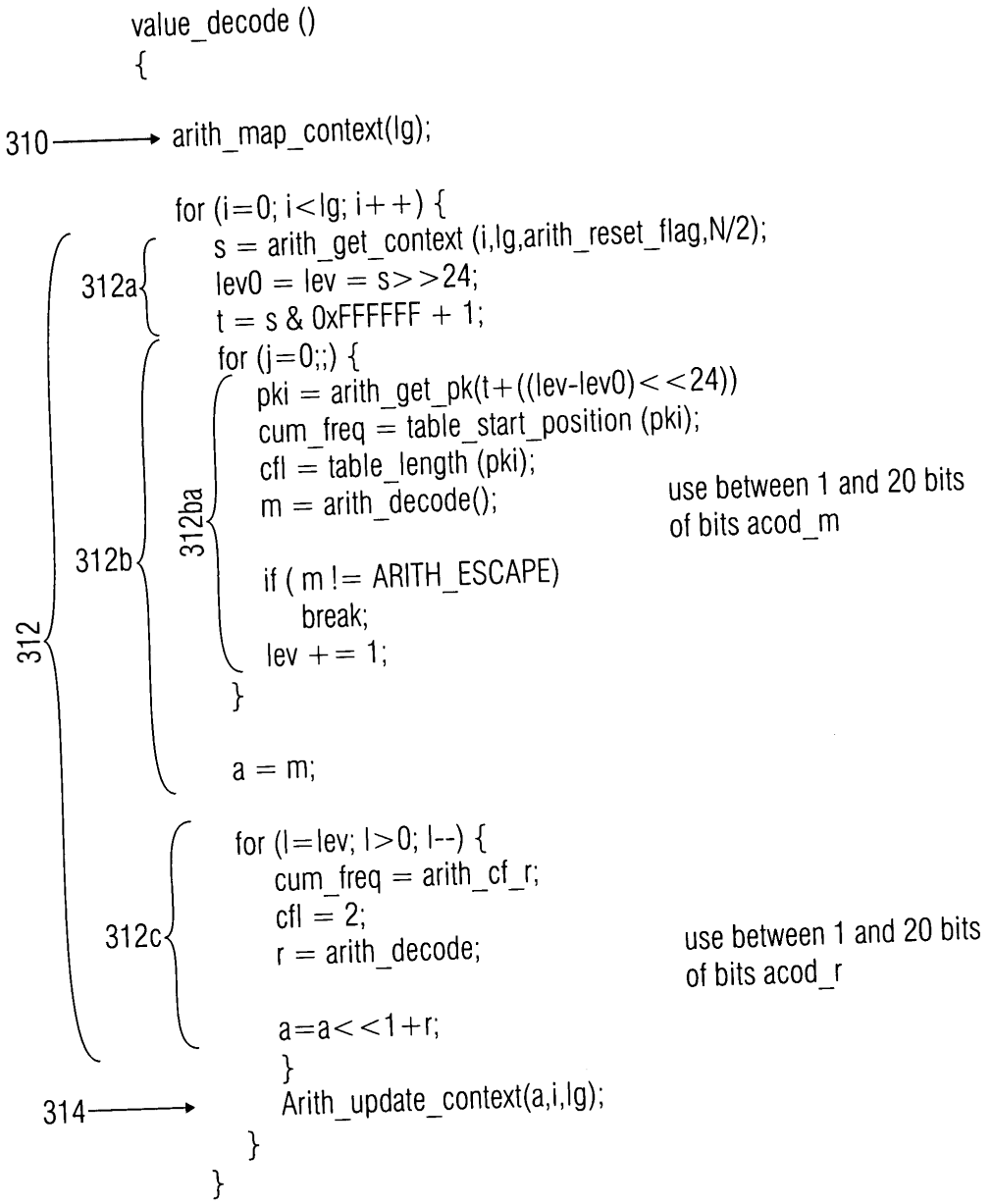
ФИГ. 2А

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



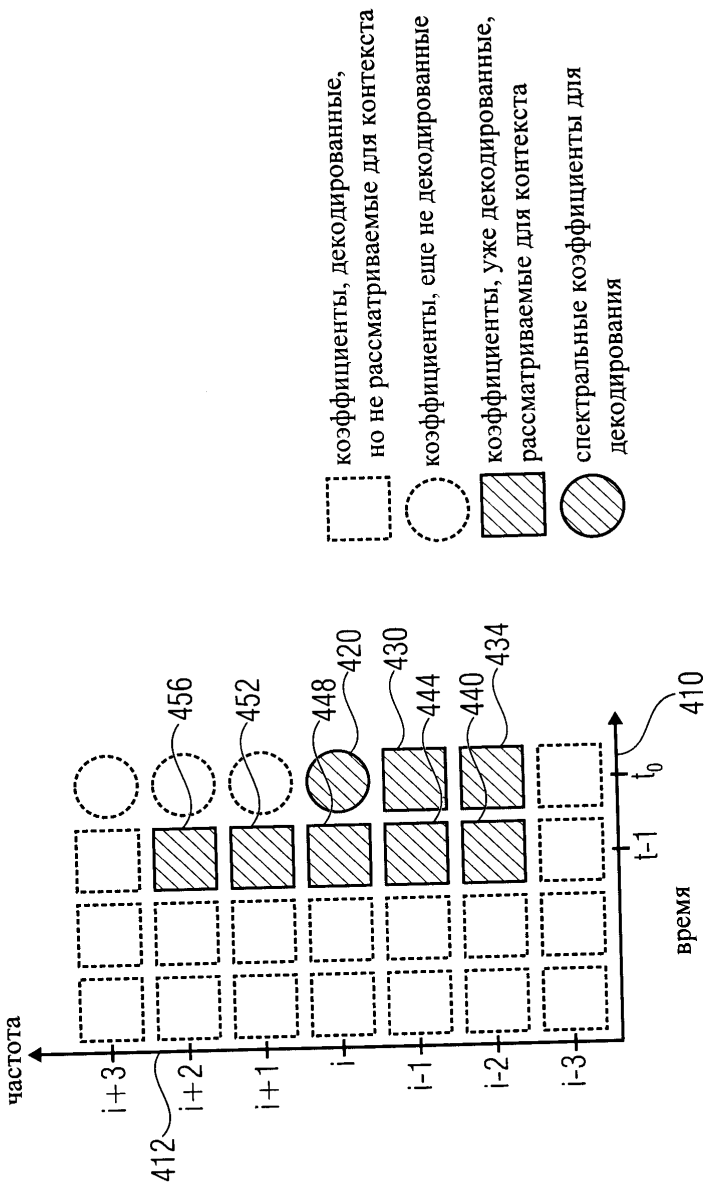
ФИГ. 2В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 3

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 4

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
/*Input variables*/
lg /*number of sepctral coefficients to decode in the frame*/
previous_lg /Previous number of spectral lines of the previous frame*/

arith_map_context()
{
      v=w=0

      ratio= ((float)previous_lg)/((float)lg);
      for(j=0; j<lg; j++){
            k = (int) ((float)) ((j)*ratio);
            q[0][v++].c = qs[w+k];
      }

      previous_lg=lg;
}
```

ФИГ. 5A

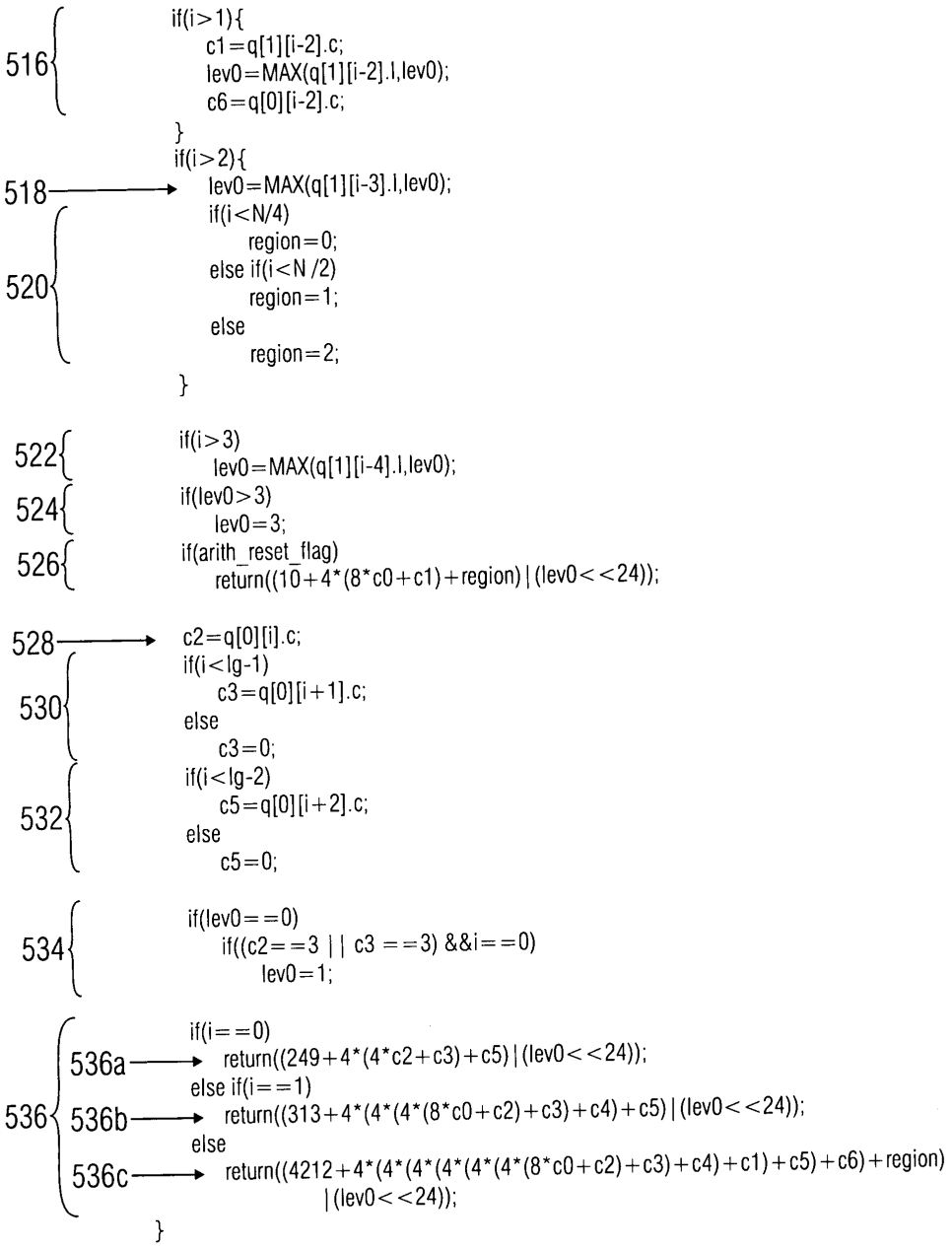
Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
/*Input variables*/
i /*Index of the spectral value to decode in the vector*/
lg /*Number of expected quantized coefficients*/
N /*Number of lines of the transformation*/
ari_reset_flag /*flag indicating whether the context should be reset*/
/*Output value*/
t /*Concatenated state index s and predicted bit-plane level lev0*/

arith_get_context()
{
    int a0,c0,c1,c2,c3,c4,c5,c6,lev0,region;

    if(arith_reset_flag && i==0)
        return(0);
    if((!arith_reset_flag) && (i!=0)){
        int k;
        int lim_min,lim_max;
        int flag=1;
        lim_max = i+6;
        if((i+lim_max)>lg-1)
            lim_max=lg-1-i;
        lim_min = -5;
        if((i+lim_min)<0)
            lim_min=-i;
        for(k=lim_min;k<0;k++)
            if(q[0][k].c!=0 && q[1][k].c!=0)
                flag=0; break;
        for(;k<=lim_max;k++)
            if(q[0][k].c!=0)
                flag=0; break;
        if(flag)
            return(1);
    }
    if(i>0){
        a0=q[1][i-1];
        c0=ABS(a0);
        lev0=0;
        while((a0<-4)||(a0>=4)){
            a0>>=1;
            lev0++;
            c0=4+lev0;
        }
        if(c0>7)
            c0=7;
        if(lev0>3)
            lev0=3;

        if(arith_reset_flag && i==1)
            return((2+c0)|(lev0<<24));
        c4=q[0][i-1].c;
    }
```

510, 512, 512a, 512b, 512c, 512d, 514, 514a, 514b, 514c, 514d, 514e

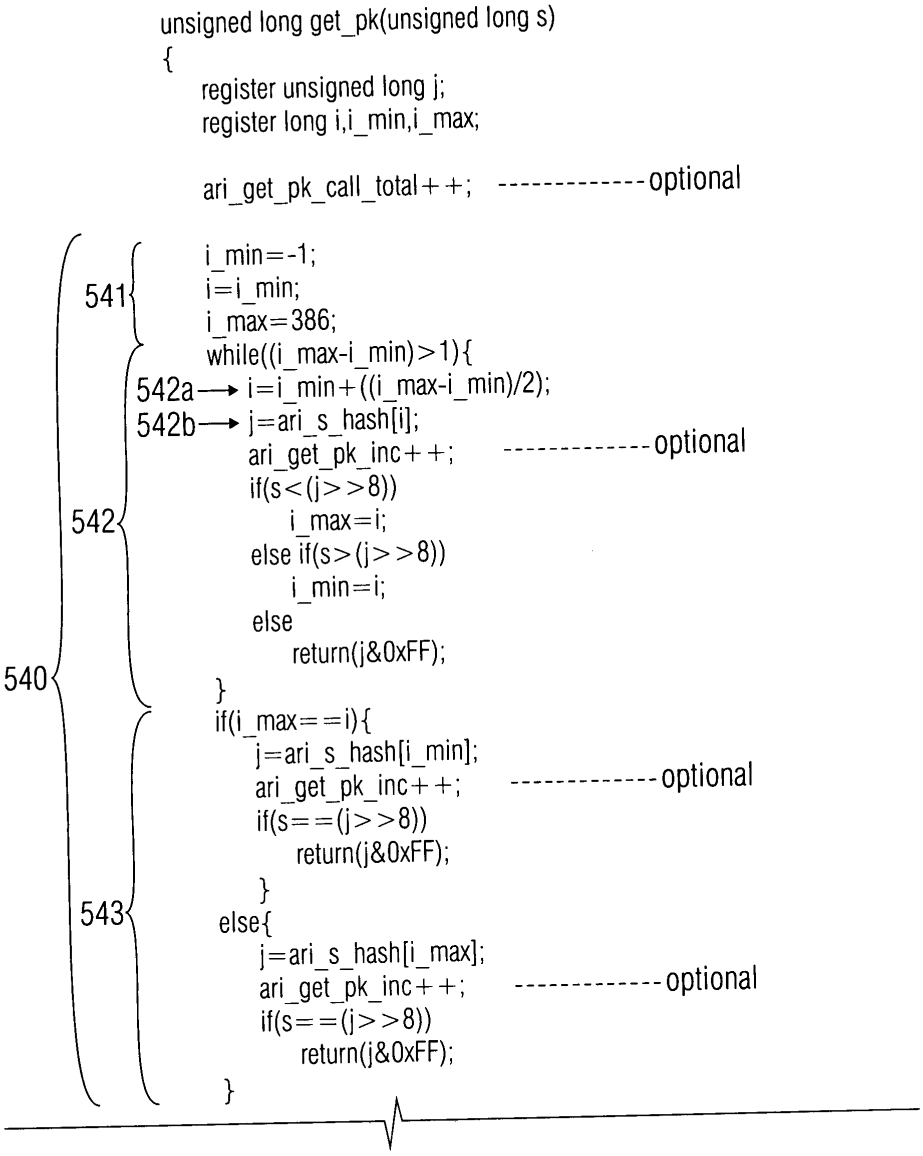
ФИГ. 5B

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



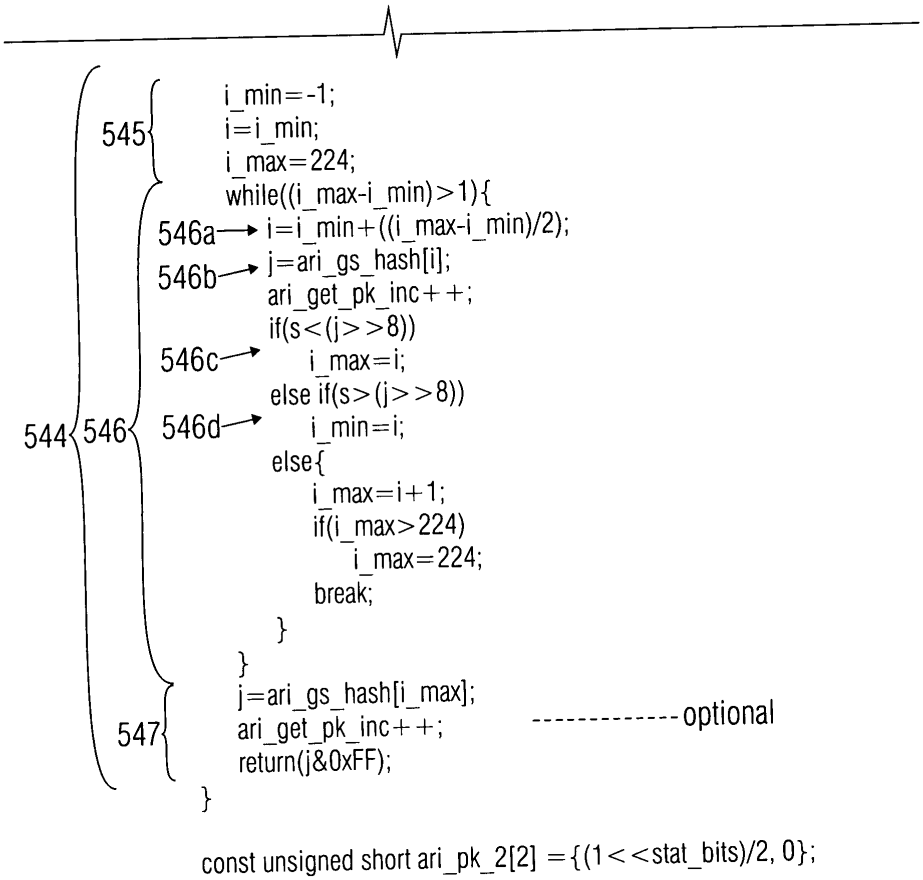
ФИГ. 5C

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 5D1

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 5D2

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
/*Input variable*/
s  /*State of the context*/
/*Output value*/
pki /*Index of the probability model */

arith_get_pk(s)
{
    register unsigned long i,j;

    for (i=0;i<387;i++)
    {
        j=ari_s_hash[i];
        if ( (j>>8)==s ) return j&255;
    }
    for (i=0;i<225;i++)
    {
        j=ari_gs_hash[i];
        if ( s<(j>>8) ) return j&255;
    }
    return j&255;
}
```

550

560

ФИГ. 5E

```
unsigned long get_pk(unsigned long s)
{
    register unsigned longlong j;
    register unsigned long  i;

    for (i=0;i<387;i++)
    {
        j=ari_s_hash[i];
        if ( (j>>8)==s )
            return j&0xFF;
    }
    for(i=0;i<225;i++){
        j=ari_gs_hash[i];
        if ( s<(j>>8) ) return j&0xFF;
    }
    return(j&0xFF);
}
```

ФИГ. 5F

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
/*helper funtions*/
bool arith_first_symbol(void);
    /* Return TRUE if it is the first symbol of the sequence, FALSE otherwise*/
Ushort arith_get_next_bit(void);
    /* Get the next bit of the bitstream*/

/* global variables */
low
high
value

/* Input variables */
cum_freq[];  /* cumulative frequencies table*/
cfl;         /* length of cum_freq[] */

arith_decode()
{
    if(arith_first_symbol())
    {
        value = 0;
        for (i=1; i<=20; i++)
        {
        value = (val<<1) | arith_get_next_bit();
        }
        low=0;
        high=1048575;
    }

    range = high-low+1;
    cum =((((int64) (value-low+1))<<16)-((int64) 1))/((int64) range);
    p = cum_freq-1;

    do
    {
        q=p+(cfl>>1);
        if ( *q > cum ) {p=q; cfl++; }
        cfl>>=1;
    }
        while ( cfl>1 );
```

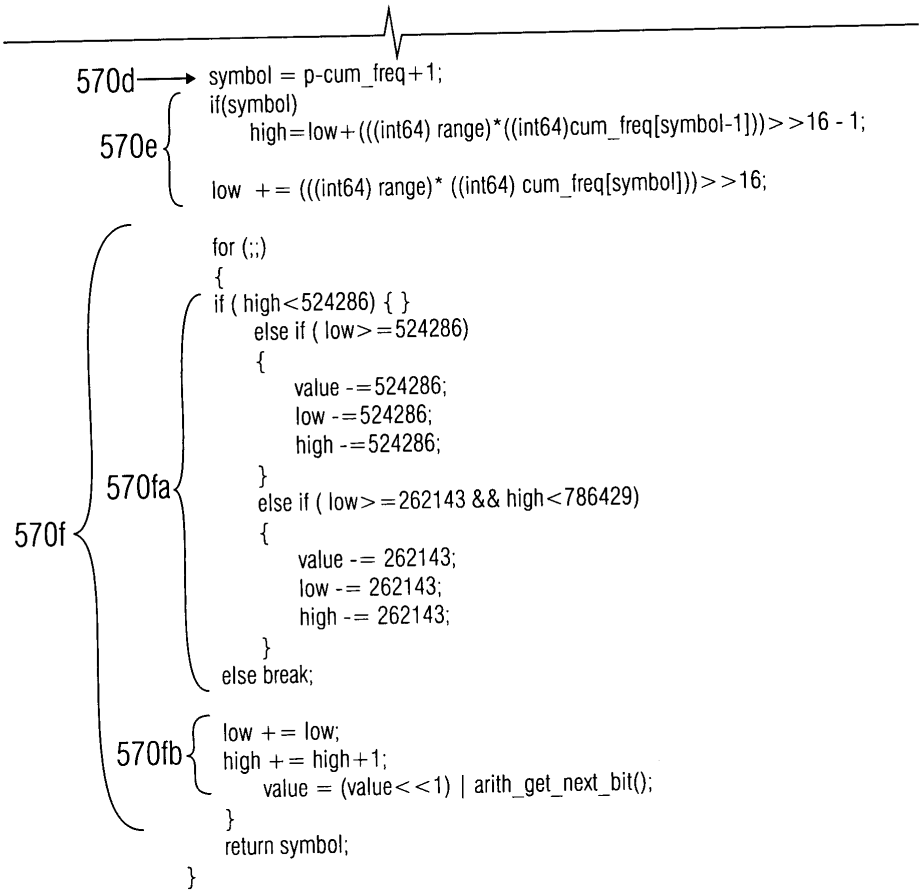
570a

570b

570c

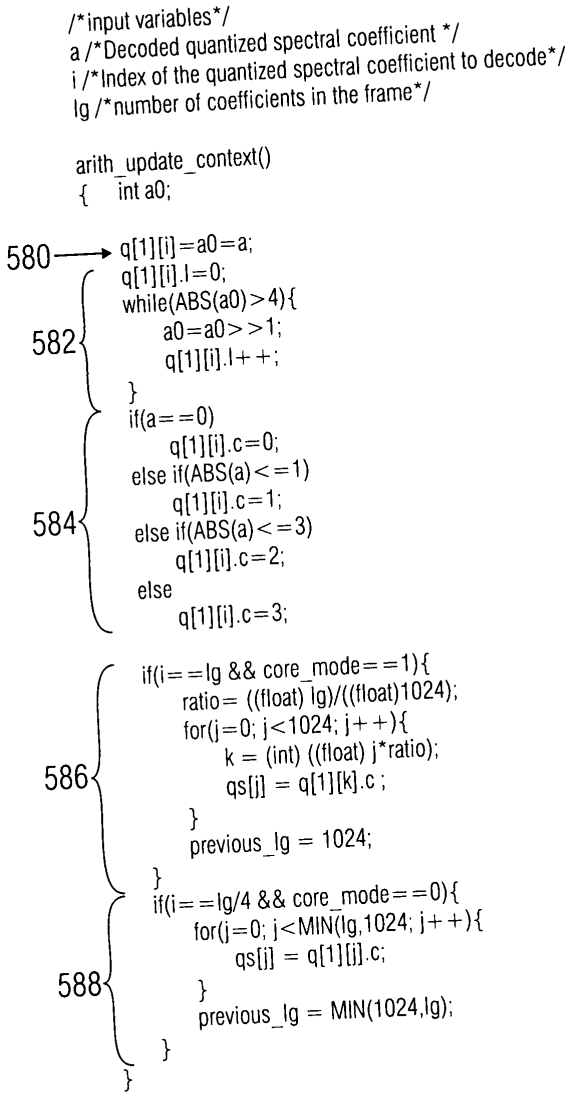
ФИГ. 5G1

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 5G2

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 5H

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

Определения

a	квантованный коэффициент для декодирования
m	Наиболее значимая 2-битная плоскость квантованного спектрального коэффициента для декодирования
r	Наиболее значимая 2-битная плоскость квантованного спектрального коэффициента для декодирования
lev	уровень оставшихся бит плоскостей Он соответствует количеству бит плоскостей менее значимых, чем наиболее значимые 2 битные плоскости.
lev0	предсказанный уровень бит плоскости
arith_s_hash[]	хэш таблица, отображающая состояния контекста для индекса pki сводной таблицы частот
arith_gs_hash[]	хэш таблица, отображающая группу состояний контекста для индекса pki сводной таблицы частот
arith_cf_m[pki][9]	модели сводных частот для наиболее значимой 2-битной плоскости m и символа ARITH_ESCAPE
arith_cf_r []	сводные частоты для наименее значимых бит плоскостей символа r
previous_lg	количество переданных спектральных коэффициентов ранее декодированных арифметическим декодером
N	длина окна Для AAC выводится из window_sequence (см. разделе 6.8.3.1) и для TCX N=2.lg.
q[2][]	текущий контекст для спектрального коэффициента для декодирования
qs[]	прошлый контекст, сохраненный для следующего кадра
arith_reset_flag	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст

ФИГ. 5I

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
usac_raw_data_block ()
{
    single_channel_element (); and/or
    channel_pair_element ();
}
```

ФИГ. 6A

Syntax of single_channel_element()

Syntax	No. of bits	Mnemonic
single_channel_element() { **core_mode**	1	uimsbf

if (core_mode == 1) {
lpd_channel_stream();
}
else {
fd_channel_stream();
}
}		

ФИГ. 6B

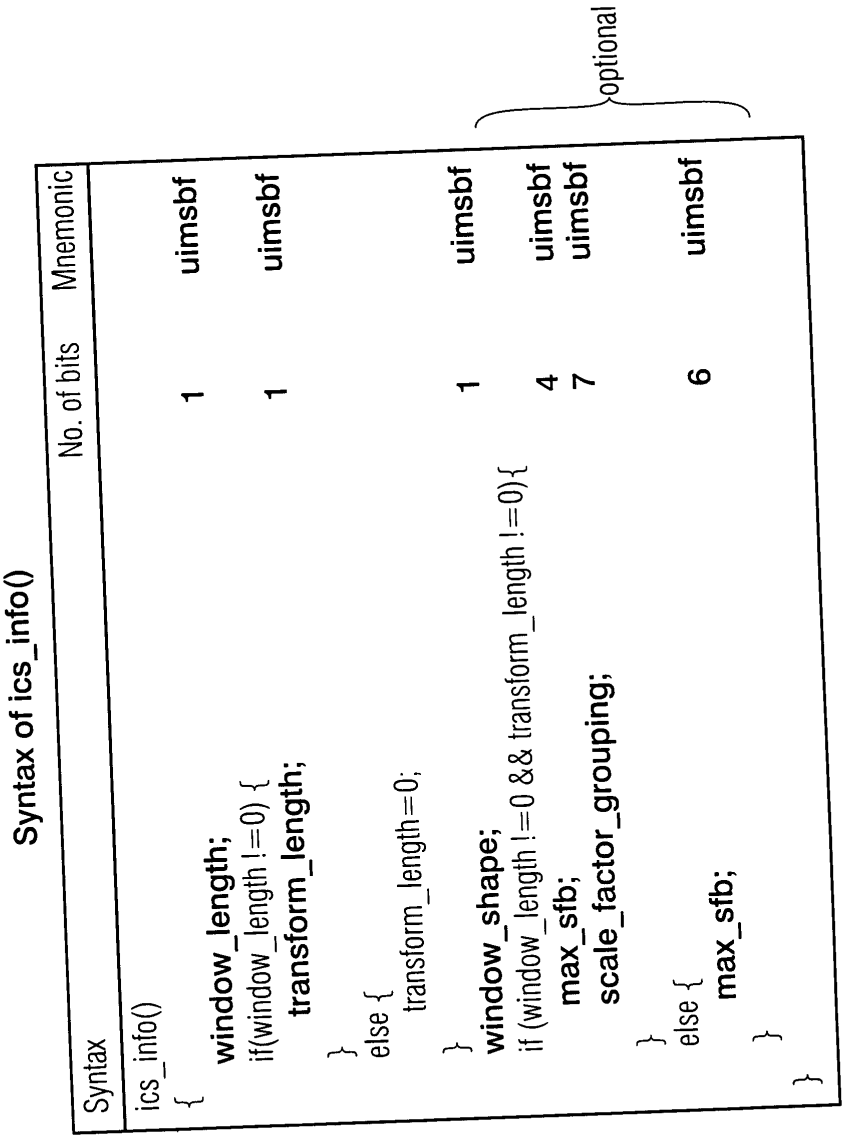
Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

Syntax of channel_pair_element()

Syntax	No. of bits	Mnemonic
channel_pair_element()		
{		
core_mode0	**1**	**uimsbf**
core_mode1	**1**	**uimsbf**
ics_info();	optional: common ics_info for two channels	
if (core_mode0 == 1) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
if (core_mode1 == 1) {		
lpd_channel_stream();		
}		
else {		
fd_channel_stream();		
}		
}		

ФИГ. 6C

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



Syntax of ics_info()

Syntax	No. of bits	Mnemonic
ics_info()		
{		
window_length;	1	**uimsbf**
if(window_length !=0) {		
transform_length;	1	**uimsbf**
}		
else {		
transform_length=0;		
}		
window_shape;	1	**uimsbf**
if (window_length !=0 && transform_length !=0){		
max_sfb;	4	**uimsbf**
scale_factor_grouping;	7	**uimsbf**
}		
else {		
max_sfb;	6	**uimsbf**
}		
}		

optional

ФИГ. 6D

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

Syntax of fd_channel_stream()

Syntax	No. of bits	Mnemonic
fd_channel_stream() {		
global_gain;	8	uimsbf
ics_info(); (unless included in channel pair element)		
scale_factor_data ();		
ac_spectral_data ();		
}		

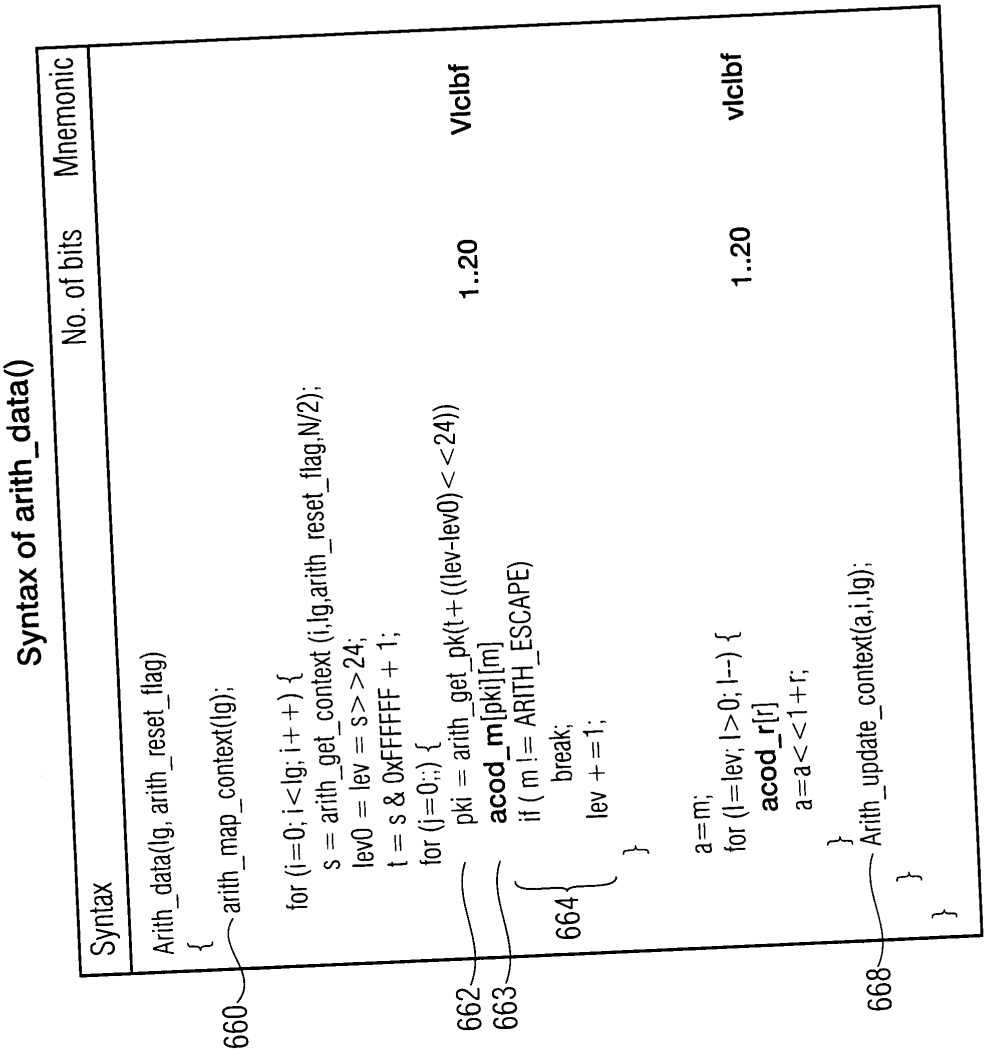
ФИГ. 6E

Syntax of ac_spectral_data()

Syntax	No. of bits	Mnemonic
ac_spectral_data() {		
arith_reset_flag	1	uimsbf
for (win=0; win<num_windows; win++){		
arith_data(num_bands, arith_reset_flag)		
}		
}		

ФИГ. 6F

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



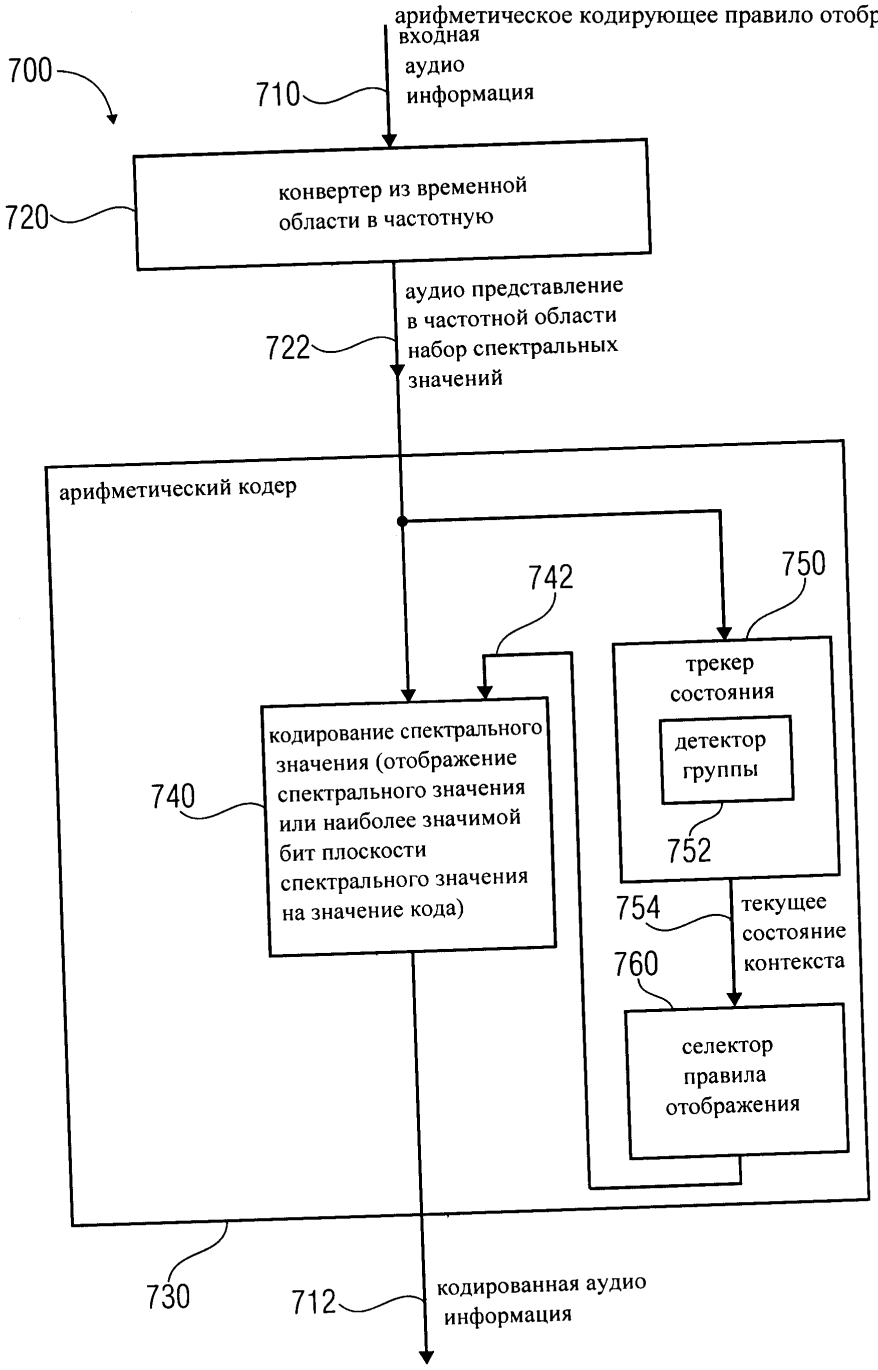
ФИГ. 6G

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона

Определения	арифметическое кодирующее правило отображения
arith_data()	элемент данных для декодирования данных спектрального бесшумного кодера
arith_reset_flag	флаг, который указывает, должен ли быть сброшен спектральный бесшумный контекст
acod _cf_m[pki][a]	Арифметическое кодовое слово, необходимое для арифметического декодирования наиболее значимой 2-битной плоскости a квантованного спектрального коэффициента
arith_cf_r[]	Арифметическое кодовое слово, необходимое для арифметического декодирования остаточных бит плоскостей квантованного спектрального коэффициента
Help elements	
a	спектральный квантованный коэффициент для декодирования
m	Наиболее значимая 2-битная плоскость квантованного спектрального коэффициента для декодирования
r	Наиболее значимая 2-битная плоскость квантованного спектрального коэффициента для декодирования
N	длина окна Для AAC выводится из window_sequence (см. разделе 6.8.3.1) и для TCX N=2.lg.
lg	количество квантованных коэффициентов для декодирования
i	индекс квантованных коэффициентов для декодирования в кадре
pki	индекс сводной таблицы частот, используемый арифметическим декодером для декодирования a.
arith_get_pk ()	Функция, которая возвращает индекс pki сводной таблицы частот, необходимый для декодирования кодового слова acod _ng[pki][a].
t	состояние контекста
arith_get_context ()	Функция, которая возвращает состояние контекста.
lev0	предсказанный уровень бит плоскости
s	состояние контекста, объединенное с предсказанным уровнем бит плоскости lev0.
lev	Уровень бит плоскостей для декодирования за пределами наиболее значимой 2-битной плоскости.
ARITH_ESCAPE	Символ перехода, который указывает дополнительные бит плоскости для декодирования за пределами предсказанного уровня бит плоскости lev0.

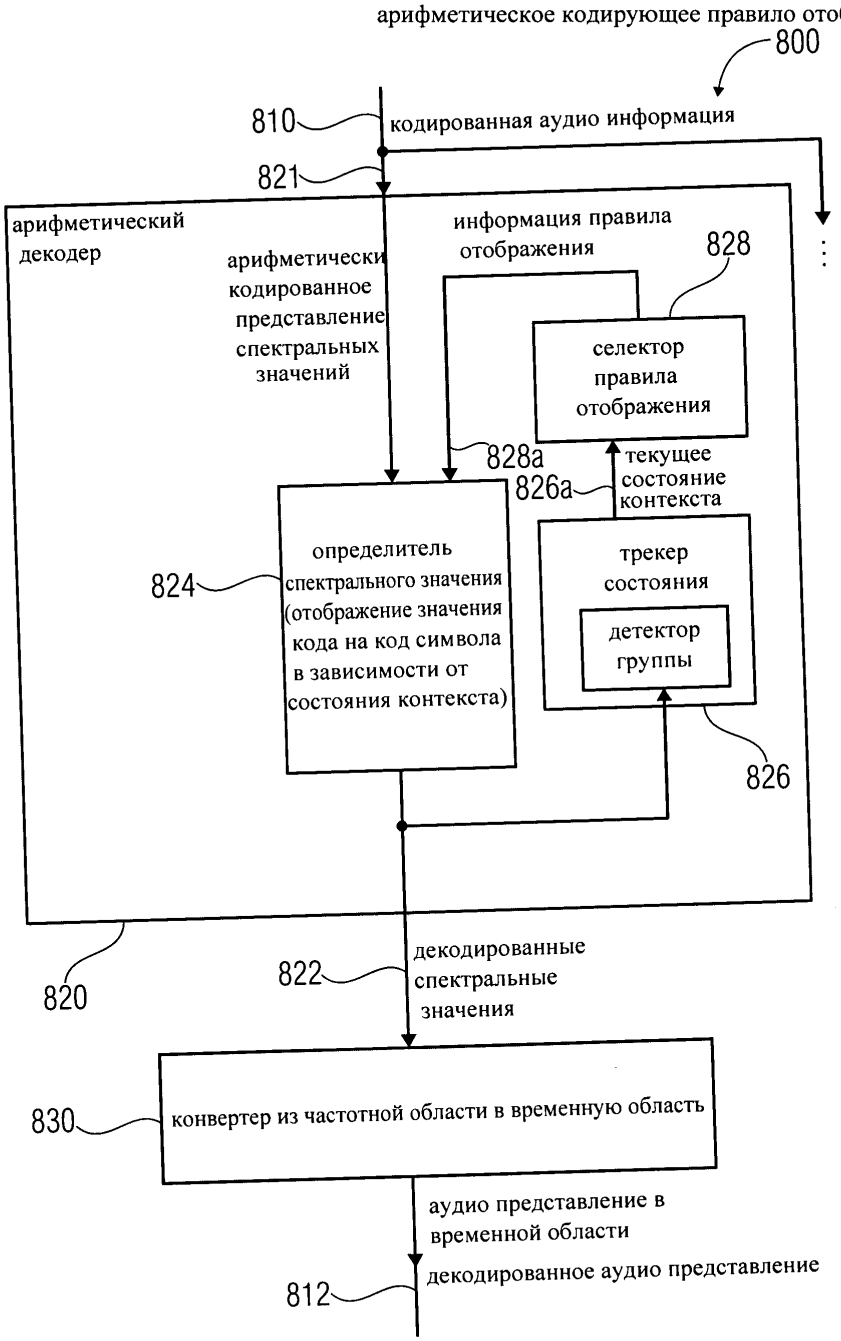
ФИГ. 6H

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



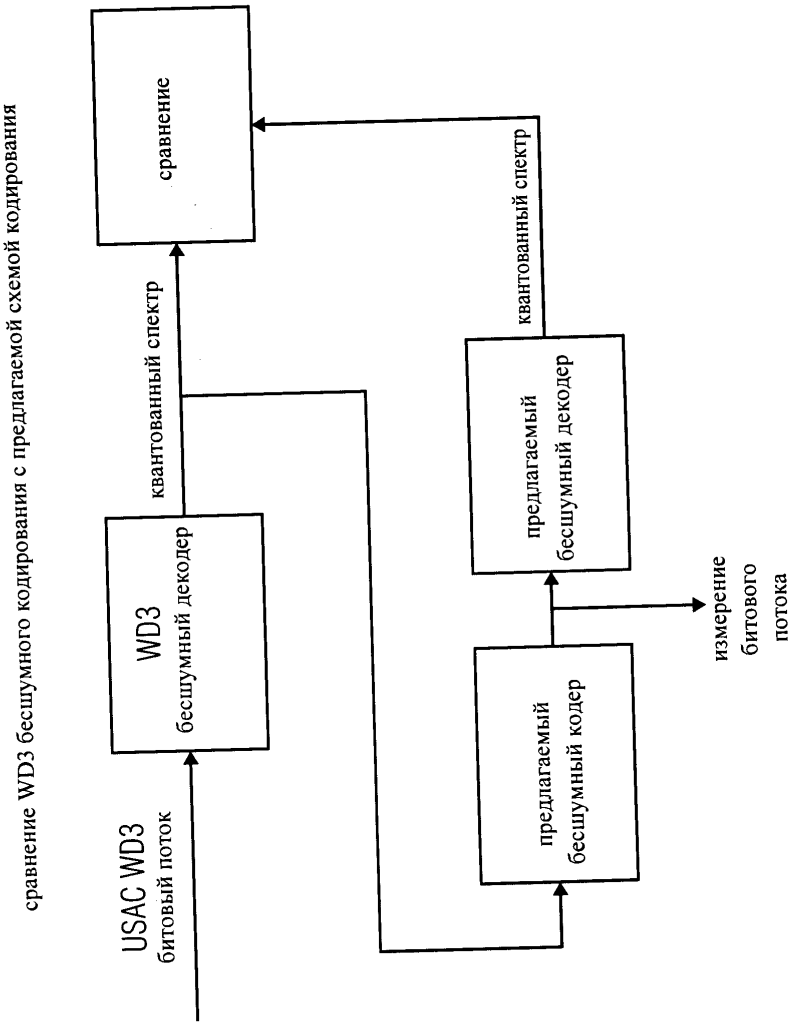
ФИГ. 7

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



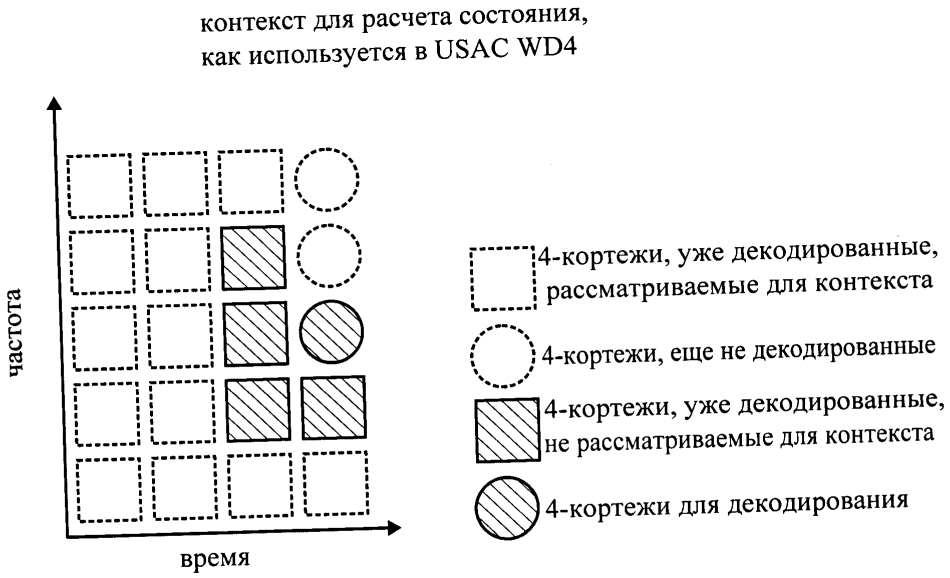
ФИГ. 8

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



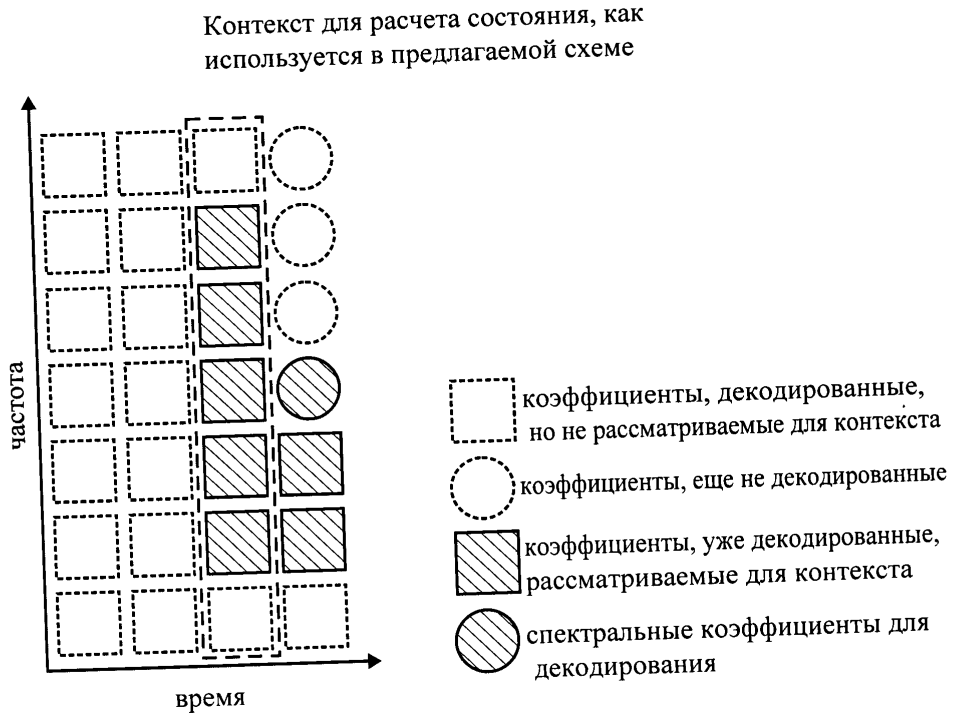
ФИГ. 9

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 10A

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 10В

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

имя таблицы	описание	данных	память (слов в 32 битах)
arith_cf_ng_hash[128]	хэш таблица, отображающая контекст на индекс модели вероятности	word	128
arith_cf_ng[32][545]	сводные частоты групп для каждого режима распределения вероятности, l	1/2 word	8720
egroups[8][8][8][8]	индекс группы для 4 кортежа	1/2 word	2048
dgvectors[4*4096]	индекс группы отображения и индекс элемента для 4 кортежа	1/4 word	4096
dgroups[544]	индекс группы отображения для кардинального числа группы и смещение в dg векторах	word	544
arith_cf_ne[2701]	сводные частоты символа индекса элемента	1/2 word	1350.5
arith_cf_r[16]	сводные частоты наименее значимых бит плоскостей	1/2 word	8
общий			**16894.5**

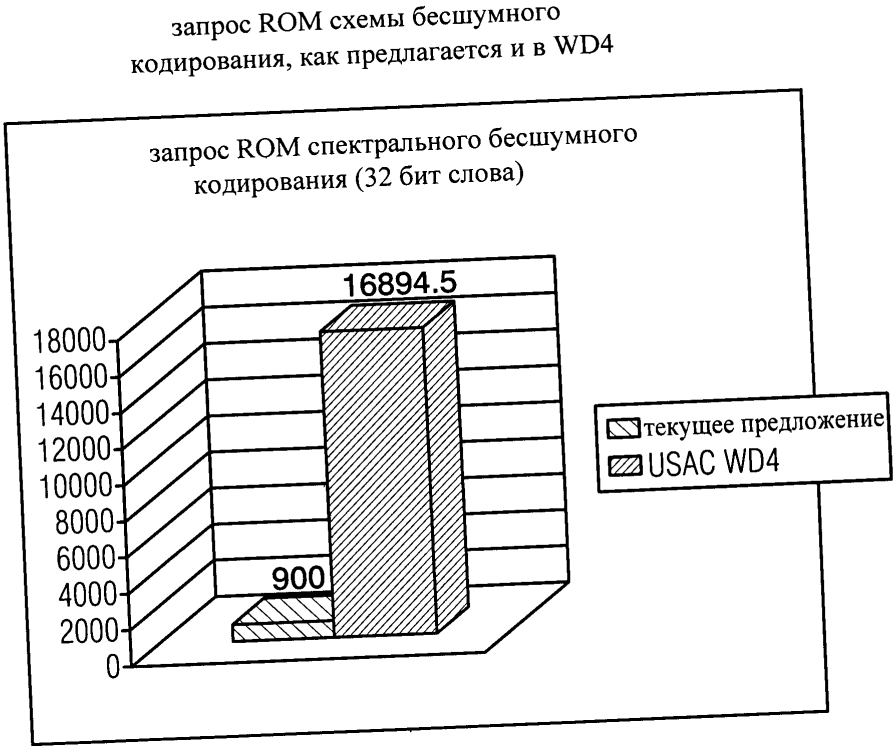
ФИГ. 11A

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

имя таблицы	описание	данных	память (слов в 32 битах)
arith_s_hash[387]	хэш таблица, отображающая состояния контекста для сводной таблицы частот	word	387
arith_gs_hash[225]	хэш таблица, отображающая группу состояний контекста для сводной таблицы частот	word	225
arith_cf_m[64][9]	модели сводных частот для наиболее значимой 2-битной плоскости m и символа ARITH_ESCAPEя	1/2 word	288
общий			900

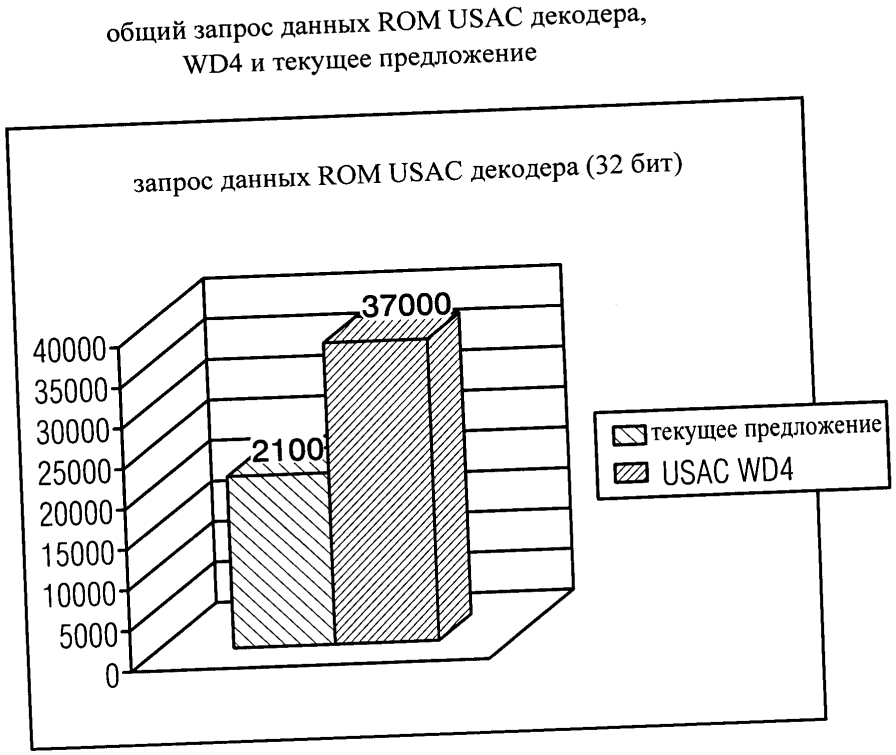
ФИГ. 11B

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 12A

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения



ФИГ. 12B

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

средние битрейты, производимые USAC кодером, использующим WD арифметический кодер и новое предложение

рабочий режим	WD (kbit/s)	новое предложение (kbit/s)	разница после перекодирования (kbit/s)	разница после перекодирования (% общего битрейта)
Test 1, 64kbps stereo	64.00	63.34	-0.66	-1.04
Test 2, 32kbps stereo	32.00	31.66	-0.34	-1.05
Test 3, 24kbps stereo	24.00	23.73	-0.27	-1.11
Test 4, 20kbps stereo	20.00	19.78	-0.22	-1.11
Test 5, 16kbps stereo	16.00	15.82	-0.18	-1.10
Test 6, 24kbps mono	24.00	23.68	-0.32	-1.32
Test 7, 20kbps mono	20.00	19.72	-0.28	-1.39
Test 8, 16kbps mono	16.00	15.79	-0.21	-1.31
Test 9, 12kbps mono	12.00	11.86	-0.14	-1.19

ФИГ. 13A

контроль бит резервуара для USAC WD3 и новое предложение

рабочий режим	контроль бит резервуара					
	новое предложение			WD		
	min	max	avg	min	max	avg
Test 1, 64kbps stereo	3653	9557	8137	2314	9557	7018
Test 2, 32kbps stereo	1808	4505	4196	581	4505	3530
Test 3, 24kbps stereo	1538	4704	4408	957	4704	3871
Test 4, 20kbps stereo	2367	4864	4600	712	4864	3854
Test 5, 16kbps stereo	2712	5006	4804	724	5006	4234
Test 6, 24kbps mono	2185	4704	4457	1002	4704	3927
Test 7, 20kbps mono	2599	4864	4630	1192	4864	3935
Test 8, 16kbps mono	2820	5006	4876	1434	5006	4450
Test 9, 12kbps mono	3529	5184	5081	2256	5184	4787

ФИГ. 13B

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

средние битрейты для USAC WD3 и новое предложение

рабочий режим	средний битрейт в kbit/s					
	новое предложение			WD		
	FD режим	wLPT режим	общий	FD режим	wLPT режим	общий
Test 1, 64kbps stereo	53.73	---	53.73	54.40	---	54.40
Test 2, 32kbps stereo	25.31	26.34	25.60	25.80	26.61	26.02
Test 3, 24kbps stereo	18.27	19.17	18.50	18.66	19.40	18.85
Test 4, 20kbps stereo	15.50	15.93	15.61	15.83	16.12	15.90
Test 5, 16kbps stereo	12.45	12.60	12.52	12.80	12.73	12.77
Test 6, 24kbps mono	19.94	19.51	19.73	20.41	19.42	20.15
Test 7, 20kbps mono	16.15	15.91	16.08	16.56	16.12	16.45
Test 8, 16kbps mono	13.02	12.59	12.81	13.45	12.73	13.09
Test 9, 12kbps mono	9.35	9.66	9.51	9.68	9.71	9.70

ФИГ. 14

минимальный, максимальный и средний битрейты USAC на базе кадра

рабочий режим	минимальный битрейт в kbit/s	максимальный битрейт в kbit/s	средний битрейт в kbit/s
Test 1, 64kbps stereo	15.26	101.79	63.34
Test 2, 32kbps stereo	13.13	48.61	31.66
Test 3, 24kbps stereo	11.69	36.58	23.73
Test 4, 20kbps stereo	3.09	30.94	19.78
Test 5, 16kbps stereo	4.02	26.47	15.82
Test 6, 24kbps mono	1.47	37.35	23.68
Test 7, 20kbps mono	1.38	31.13	19.72
Test 8, 16kbps mono	11.40	24.64	15.79
Test 9, 12kbps mono	8.72	18.91	11.86

ФИГ. 15

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

лучший и худший варианты на базе кадра

рабочий режим	лучший вариант		худший вариант	
	(bit/s)	(%)	(bit/s)	(%)
Test 1, 64kbps stereo	-30.87	-33.06	6.14	9.07
Test 2, 32kbps stereo	-10.33	-28.63	2.17	6.77
Test 3, 24kbps stereo	-11.86	-30.75	1.85	7.71
Test 4, 20kbps stereo	-7.45	-30.27	1.67	8.36
Test 5, 16kbps stereo	-5.43	-27.89	1.50	9.42
Test 6, 24kbps mono	-17.06	-45.83	1.25	4.36
Test 7, 20kbps mono	-15.86	-41.46	0.88	3.38
Test 8, 16kbps mono	-4.75	-24.85	1.11	7.31
Test 9, 12kbps mono	-3.95	-26.33	0.82	6.99

ФИГ. 16

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
/*
Entropy:
fu mem.: 1.2792 bit (100.00 %)
no mem.  : 1.6289 bit (127.34 %)
split:   : 1.2971 bit (101.40 %)
*/

/*  1224 States, Entropy increase: 0.000384 */

/*Final Entropy : 1.297556 */

/*Total states = 612;*/
/*Signicant states = 387;*/
/*Pseudo states = 225;*/
/*Proba models = 64;*/
unsigned long long ari_get_pk_inc=0;
unsigned long long ari_get_pk_call_total=0;

static unsigned long ari_s_hash[387] = {
0x00000200,0x00000B01,0x00000C02,0x00000D03,0x00000F25,0x0000101C,0x0000110B,
0x00001327,
0x0000142F,0x00002B25,0x00002C22,0x00002D14,0x00002F2D,0x0000302B,0x0000312B,
0x00003330,
0x00003432,0x00003532,0x00004C32,0x00005031,0x00005131,0x0000FA02,0x0000FB01,
0x0000FC1C,
0x0000FE1C,0x0000FF1C,0x0001001E,0x00010A2E,0x00010B25,0x00010E25,0x00010F25,
0x00013938,
0x00013A04,0x00013B02,0x00013C01,0x0010393D,0x00107504,0x00107605,0x00107706,
0x0010790D,
0x00107A07,0x00107B08,0x0010850D,0x00108609,0x0010870A,0x00108B0E,0x0010B50B,
0x0010B60C,
0x0010B70D,0x0010B90B,0x0010BA1D,0x0010BB16,0x0010C615,0x0010C70C,0x0010F521,
0x0010F628,
0x0010F728,0x00110528,0x00117516,0x0011760E,0x0011770F,0x00117A12,0x00117B07,
0x0011870E,
0x0011B514,0x0011B615,0x0011B70C,0x0011B914,0x0011BA15,0x0011BB1D,0x0011C619,
0x0011C715,
0x00147516,0x00147610,0x00147711,0x0014791D,0x00147A0C,0x00147B0E,0x0014851B,
0x00148616,
0x00148707,0x0014890B,0x00148A1D,0x00148B16,0x0014950B,0x0014961D,0x0014B615,
0x0014B71D,
0x0014BA14,0x0014BB15,0x0014C614,0x0014C715,0x0015052D,0x0015750B,0x0015760C,
0x00157710,
0x0015790B,0x00157A1D,0x00157B16,0x0015850B,0x0015861D,0x0015870C,0x00158914,
0x00158A15,
0x00158B1D,0x0015B619,0x0015B714,0x0020751D,0x00207612,0x00207713,0x0020791D,
0x00207A0C,
0x00207B0E,0x0020850B,0x0020860C,0x00208710,0x00208A1D,0x00208B0C,0x0020B615,
0x0020B71D,
0x0021750B,0x0021760C,0x0021770E,0x0021790B,0x00217A1D,0x00217B12,0x00218514,
0x00218615,
0x0021870C,0x0021891C,0x00218A15,0x00218B1D,0x0021B619,0x0021B715,0x0021BA22,
0x0021BB19,
0x00247514,0x0024761D,0x0024770E,0x00247914,0x00247A15,0x00247B0C,0x00248514,
0x0024861D,
0x00248716,0x0024891C,0x00248A15,0x00248B1D,0x0024B619,0x0024B715,0x0025751C,
0x0025760B,
```

ФИГ. 17 (1)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
0x0025771B,0x0025791C,0x00257A0B,0x00257B1B,0x0025851C,0x0025860B,0x00258E1B,
0x0025890B,
0x00258A1D,0x00258B1B,0x0025B618,0x0025B722,0x0025BA1F,0x0025BB18,0x0025C61F,
0x0025C718,
0x0025CA1F,0x0025CB1F,0x003A9D1C,0x003A9E0B,0x003A9F0B,0x004FB125,0x004FB21C,
0x004FB31C,
0x00907514,0x00907615,0x00907716,0x00907919,0x00907A15,0x00907B1D,0x00907D1C,
0x00907E1C,
0x00907F14,0x00908522,0x00908614,0x0090871D,0x00908918,0x00908A19,0x00908B14,
0x00908D24,
0x00909523,0x0090961F,0x0090992B,0x0090B517,0x0090B618,0x0090B719,0x0090B91F,
0x0090BA22,
0x0090BB19,0x0090BD1C,0x0090BE1C,0x0090BF1C,0x0090C52B,0x0090C61F,0x0090C718,
0x0090C917,
0x0090CA1F,0x0090CB1F,0x0090CD23,0x0090D52E,0x0090D62C,0x0090D92C,0x0090F52D,
0x0090F62D,
0x0090F72F,0x0090F925,0x0090FA2E,0x0090FB2D,0x0090FD1E,0x0090FE1E,0x0091052F,
0x0091062F,
0x00910928,0x00910D25,0x00917519,0x00917615,0x0091771D,0x00917922,0x00917A14,
0x00917B15,
0x00918619,0x00918714,0x00918A18,0x00918B18,0x0091B618,0x0091B722,0x0091BA18,
0x0091BB18,
0x00947518,0x00947614,0x0094771D,0x0094791F,0x00947A19,0x00947B14,0x00948520,
0x00948619,
0x00948714,0x00948A18,0x00948B18,0x0094B61F,0x0094B718,0x0094BA17,0x0094BB1F,
0x0094C617,
0x0094C717,0x00957520,0x00957622,0x00957714,0x00957924,0x00957A18,0x00957B18,
0x00958524,
0x00958618,0x00958718,0x0095892B,0x00958A17,0x00958B1F,0x0095B52B,0x0095B617,
0x0095B71F,
0x0095B92B,0x0095BA17,0x0095BB17,0x0095C52C,0x0095C617,0x0095C717,0x0095C92C,
0x0095CA2B,
0x0095CB2C,0x00A0751F,0x00A07614,0x00A07715,0x00A0791F,0x00A07A19,0x00A07B14,
0x00A0851F,
0x00A08622,0x00A08714,0x00A08917,0x00A08A18,0x00A08B18,0x00A0B52B,0x00A0B61F,
0x00A0B718,
0x00A0BA17,0x00A0BB1F,0x00A0C617,0x00A0C717,0x00A17524,0x00A17622,0x00A17714,
0x00A17924,
0x00A17A18,0x00A17B18,0x00A1861F,0x00A18718,0x00A18A17,0x00A18B17,0x00A1B617,
0x00A1B71F,
0x00A1BA17,0x00A1BB17,0x00A47524,0x00A47618,0x00A47714,0x00A4792B,0x00A47A18,
0x00A47B18,
0x00A4852B,0x00A4861F,0x00A48718,0x00A4892B,0x00A48A17,0x00A48B17,0x00A4B52C,
0x00A4B617,
0x00A4B717,0x00A4B92C,0x00A4BA17,0x00A4BB17,0x00A4C52E,0x00A4C62B,0x00A4C72B,
0x00A4C92E,
0x00A4CA2C,0x00A4CB2C,0x00A5752C,0x00A57617,0x00A57718,0x00A5792B,0x00A57A17,
0x00A57B1F,
0x00A5852C,0x00A58617,0x00A5871F,0x00A5892C,0x00A58A17,0x00A58B17,0x00A5B52E,
0x00A5B62B,
0x00A5B717,0x00A5B92E,0x00A5BA2C,0x00A5BB2C,0x00A5C52E,0x00A5C62C,0x00A5C72C,
0x00A5C92E,
0x00A5CA2C,0x00A5CB2C,0x00BA9D2D,0x00BA9E2E,0x00BA9F2E,0x00CDD938,0x00CE2D38,
0x00CEE938,
0x00CF2D38,0x00D02D38,0x00D0313A,0x00D0713A,0x0110762F,0x0110B632,0x0110B731,
0x03D0113B,
0x03D0213C,0x03D02D3D,0x03D0313D,0x03D0413C,0x03D04D3D,0x03D0513C,0x03D05D3D,
0x03D06139,
0x03D0693D,0x03D06D3D,0x03D0713D
};
```

ФИГ. 17 (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
static unsigned long ari_gs_hash[225]={
0x00000401,0x0001491A,0x0001590B,0x00017621,0x0001891C,0x0009492A,0x000DFA38,
0x000F6D3D,
0x0010003B,0x0010B21B,0x0011321C,0x00116B29,0x0011B31D,0x00126B1E,0x00136623,
0x00146729,
0x00146F3B,0x0015321F,0x00156E27,0x00163320,0x00182725,0x00186727,0x00196323,
0x001C4721,
0x001E3F30,0x001E433B,0x00203F2A,0x0020463B,0x0020F322,0x00216A2E,0x00226723,
0x00245625,
0x00256724,0x00286625,0x002D3726,0x002D573A,0x00316627,0x00326628,0x00344729,
0x00366628,
0x003D4329,0x00416A2A,0x0042533A,0x00916A2A,0x00926B2B,0x0093E72E,0x00956B2C,
0x009D362D,
0x009D3B39,0x009E4330,0x00A2672E,0x00AD372F,0x01145630,0x01146B27,0x011C8231,
0x01226732,
0x012CC333,0x01413B34,0x019CA335,0x019CB338,0x01ACB736,0x01AD823D,0x01C37F37,
0x02156738,
0x0218AB3B,0x021C9B35,0x021E0738,0x021FB73D,0x0220E335,0x02216B3C,0x02217234,
0x0222B33C,
0x02239B3B,0x0223B23A,0x0224673B,0x0238A739,0x0240B23D,0x024CBF38,0x024CC23D,
0x024D8738,
0x0297AF3A,0x02986727,0x0298A33B,0x0298A738,0x029CAF3B,0x029CC33A,0x02A0AB35,
0x02A3E736,
0x02AC773B,0x02B0B335,0x02B3A73B,0x02C0D73C,0x02C1E735,0x03108E3D,0x03109737,
0x0311D639,
0x03147F3C,0x0314B236,0x0317A639,0x0317D629,0x0317DB33,0x03187627,0x0318AF3B,
0x0318F61A,
0x0319D739,0x031C953B,0x031D633C,0x031FCF39,0x0320873B,0x0320963A,0x03222639,
0x0323833C,
0x03239A27,0x0323EA2F,0x03242631,0x03242B3B,0x03249727,0x0325AB39,0x0327A73C,
0x0327C728,
0x03287727,0x03287E3A,0x03288737,0x032BAA39,0x032C7527,0x032D2337,0x032E9B39,
0x032EA23B,
0x032EBF3C,0x032F7E39,0x0330C63C,0x0332B23B,0x0332F230,0x03339F3B,0x0333EE27,
0x03348F30,
0x0336AB3C,0x0338A73B,0x033A7639,0x033A7F1A,0x033C793B,0x033C9A34,0x033CA33B,
0x033CA738,
0x033D0A3C,0x033DB339,0x033DFF3C,0x033E9739,0x0340CB3C,0x0344573B,0x0344AA3C,
0x0348263B,
0x034C7B3C,0x034CBB3A,0x034CD33C,0x0390B73D,0x0390E937,0x0393653D,0x0394B73B,
0x0394E33D,
0x0394FA38,0x03950A3C,0x0396CF3D,0x03971A36,0x0398673C,0x0398E13B,0x03994E39,
0x039C733B,
0x039D191A,0x039D4536,0x039E053C,0x039E6E3D,0x039E9D34,0x039F8D39,0x03A0C93B,
0x03A67939,
0x03A69D29,0x03A6D637,0x03A85A3C,0x03AE5B3B,0x03AEDB3D,0x03AF2E3C,0x03B0A13B,
0x03B2B139,
0x03B3123B,0x03B36339,0x03B3AD3C,0x03B42E33,0x03B4733B,0x03B4F53C,0x03B51F36,
0x03B59139,
0x03B5CB3C,0x03B61737,0x03B93A3C,0x03B98F39,0x03B9F53C,0x03BA063B,0x03BA2A3C,
0x03BB2739,
0x03BD3B3B,0x03BDC939,0x03BDF534,0x03BF9A39,0x03C1653B,0x03C19E2A,0x03C20527,
0x03C3633B,
0x03C3823C,0x03C3A527,0x03C45A3B,0x03C4993C,0x03C5B23B,0x03C5D527,0x03C9563B,
0x03C9A93C,
0x03CA063B,0x03CB0E3C,0x03CCB53B,0x03CD1E3C,0x03CED23D,0x03CEDF3C,0x03CFFA39,
0x40BC673E,
0xFFFFFF3F
};
```

ФИГ. 18

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
unsigned short ari_cf_m[64][9] = {
{65535,65534,65532,65215,  321,    4,    2,    1,    0},   <- pki=0
{65490,65339,64638,58133, 7463,  973,  270,  125,    0},   <- pki=1
{65530,65509,65319,60216, 5308,  222,   30,    9,    0},
{65534,65528,65470,62535, 3012,   67,    8,    2,    0},
{65533,65524,65435,62110, 3434,  104,   14,    5,    0},
{65535,65533,65499,62363, 3173,   37,    3,    1,    0},
{65535,65534,65522,63164, 2371,   14,    2,    1,    0},
{65535,65530,65448,59939, 5612,   88,    7,    2,    0},
{65535,65533,65500,61498, 4044,   38,    3,    1,    0},
{65535,65530,65444,59855, 5667,   92,    6,    1,    0},
{65535,65532,65495,61386, 4140,   39,    3,    1,    0},
{65522,65458,64905,55424,10056,  634,   88,   28,    0},
{65532,65511,65238,57072, 8457,  297,   27,    6,    0},
{65534,65522,65364,59096, 6461,  171,   15,    3,    0},
{65535,65530,65426,59204, 6342,  109,    8,    2,    0},
{65535,65533,65492,61008, 4512,   43,    3,    1,    0},
{65535,65529,65417,58998, 6519,  118,    6,    1,    0},
{65535,65533,65490,60856, 4679,   46,    4,    1,    0},
{65535,65528,65384,58400, 7127,  149,    9,    1,    0},
{65535,65532,65483,60544, 4984,   56,    4,    1,    0},
{65517,65413,64537,53269,12264, 1002,  138,   38,    0},
{65531,65503,65125,55553, 9985,  420,   37,    7,    0},
{65534,65518,65303,57889, 7650,  235,   20,    3,    0},
{65490,65288,63679,49500,15949, 1903,  301,   94,    0},
{65522,65428,64429,51580,13957, 1113,  114,   22,    0},
{65526,65447,64600,52808,12743,  937,   93,   17,    0},
{63814,60228,53108,40709,26294,15412, 8961, 5729,    0},
{65526,65486,65133,57227, 8244,  400,   58,   20,    0},
{65500,65346,64297,52845,12477, 1283,  230,   70,    0},
{65528,65486,65077,56652, 8871,  465,   56,   16,    0},
{65464,65186,63581,50731,14351, 1992,  396,  128,    0},
{65489,65278,63861,51225,14185, 1726,  302,   96,    0},
{65485,65249,63632,50425,14933, 1943,  332,   96,    0},
{65292,64495,61270,47805,17600, 4502, 1337,  542,    0},
{65519,65421,64478,52517,12971, 1068,  129,   33,    0},
{65470,65181,63344,49862,15299, 2233,  418,  132,    0},
{65472,65197,63407,49933,15445, 2176,  396,  123,    0},
```

1910 → {65535,65534,65532,65215, 321, 4, 2, 1, 0},

1912 → {65490,65339,64638,58133, 7463, 973, 270, 125, 0},

ФИГ. 19 (1)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
{65376,64781,62057,48496,16676, 3614,  923,  340,    0},
{65259,64356,60836,47316,18158, 4979, 1517,  623,    0},
{64883,63190,58260,45006,21034, 8378, 3559, 1909,    0},
{65261,64180,60126,46710,18694, 5578, 1582,  531,    0},
{64933,63355,58991,46299,19470, 7245, 2989, 1449,    0},
{63999,61383,56309,44712,24964,14237, 9489, 7028,    0},
{65451,65091,62953,48747,16324, 2626,  522,  168,    0},
{65400,64870,62109,47037,18198, 3526,  794,  278,    0},
{65200,64074,59673,44322,20692, 6133, 1836,  739,    0},
{65376,64798,61822,46437,18673, 3881,  932,  368,    0},
{65151,63887,59083,43617,21491, 6768, 2081,  841,    0},
{64592,62314,56211,42184,24450,11142, 5265, 3075,    0},
{64908,62840,56205,41474,23652, 9844, 3388, 1379,    0},
{65021,63308,57341,42286,22972, 8709, 2895, 1232,    0},
{64790,62474,55461,40843,24327,10719, 3921, 1677,    0},
{64053,60476,52429,39583,26962,15208, 7592, 4166,    0},
{63317,58934,51305,40469,29263,19682,12661, 8553,    0},
{63871,59872,52031,39473,26093,15132, 7866, 4080,    0},
{63226,58553,50425,39191,28586,18779,11388, 7035,    0},
{62219,57006,49569,40492,32376,24784,18716,14447,    0},
{62905,58273,50651,39619,28123,18379,11633, 7478,    0},
{63420,59073,51922,41516,29863,20328,13529, 9237,    0},
{63582,59263,51165,37880,24026,13893, 7771, 4535,    0},
{63223,58418,49833,37279,25503,15421, 9122, 5802,    0},
{62322,56878,48746,39095,30723,22195,15849,11887,    0},
{61826,47222,47123,47015,46913,46806,13713, 6895,    0},
{60678,44085,44084,44083,44082,44081,16715, 9222,    0}
};
```

1964 → ← pki=63

ФИГ. 19 (2)

Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
static unsigned long ari_s_hash[387]={
0x0090D52E,0x0090CB1F,0x00A4CB2C,0x00003330,0x00107A07,0x00907A15,0x00207A0C,
0x00147A0C,
0x00247A15,0x00A07A19,0x00947A19,0x00A47A18,0x0010B70D,0x0090B719,0x0020B71D,
0x0014B71D,
0x00A0B718,0x0094B718,0x0024B715,0x00A4B717,0x0110B731,0x0000FE1C,0x0090FE1E,
0x00013B02,
0x00A5C92E,0x0095C92C,0x003A9E0B,0x00000B01,0x00BA9E2E,0x0090992B,0x0011B514,
0x00A5B52E,
0x0095B52B,0x0090D62C,0x0010850D,0x0014851B,0x00248514,0x0020850B,0x00908522,
0x00948520,
0x00A4852B,0x00A0851F,0x00003432,0x00107B08,0x00207B0E,0x00907B1D,0x00147B0E,
0x00247B0C,
0x00A07B14,0x00947B14,0x00A47B18,0x00910928,0x03D0713D,0x00D0713A,0x0000FF1C,
0x0090F52D,
0x0010F521,0x00013C01,0x03D05D3D,0x00A5CA2C,0x0025CA1F,0x0095CA2B,0x003A9F0B,
0x00000C02,
0x0021790B,0x0025791C,0x00917922,0x0015790B,0x00A17924,0x00A5792B,0x00957924,
0x00BA9F2E,
0x00CF2D38,0x00000200,0x0011B615,0x0091B618,0x0021B619,0x0015B619,0x00A1B617,
0x0095B617,
0x0025B618,0x00A5B62B,0x004FB125,0x00108609,0x00148616,0x0020860C,0x00908614,
0x0024861D,
0x00948619,0x00A08622,0x00A4861F,0x0090CD23,0x00003532,0x00010A2E,0x00002B25,
0x0010B90B,
0x0090B91F,0x00A4B92C,0x0001001E,0x03D0213C,0x0090F62D,0x0010F628,0x00A5CB2C,
0x0025CB1F,
0x0095CB2C,0x00000D03,0x00117A12,0x00217A1D,0x00917A14,0x00257A0B,0x00157A1D,
0x00A17A18,
0x00A57A17,0x00957A18,0x0011B70C,0x0091B722,0x0021B715,0x0015B714,0x00A1B71F,
0x0025B722,
0x0095B71F,0x00A5B717,0x004FB21C,0x0010870A,0x00148707,0x00208710,0x0090871D,
0x00248716,
0x00948714,0x00A08714,0x00A48718,0x00907D1C,0x00010B25,0x00002C22,0x0010BA1D,
0x0090BA22,
0x0014BA14,0x00A0BA17,0x0094BA17,0x00A4BA17,0x03D0693D,0x0090F72F,0x0010F728,
0x0025851C,
0x0015850B,0x00218514,0x00A5852C,0x00958524,0x00117B07,0x00217B12,0x00257B1B,
0x00917B15,
0x00157B16,0x00A17B18,0x00A57B1F,0x00957B18,0x0090D92C,0x004FB31C,0x03D0413C,
0x00907E1C,
0x0090C52B,0x00A4C52E,0x00002D14,0x00D02D38,0x03D02D3D,0x0010BB16,0x0090BB19,
0x0014BB15,
0x00A0BB1F,0x0094BB1F,0x00A4BB17,0x0025860B,0x0015861D,0x00218615,0x00918619,
0x00A58617,
0x00958618,0x00A1861F,0x00000F25,0x00CEE938,0x00004C32,0x0011B914,0x00A5B92E,
0x0095B92B,
0x0014890B,0x0024891C,0x00908918,0x00A4892B,0x00A08917,0x00907F14,0x0010C615,
0x0090C61F,
0x0014C614,0x0094C617,0x00A4C62B,0x00A0C617,0x00910D25,0x00107504,0x00907514,
0x00147516,
0x0020751D,0x00247514,0x00A0751F,0x00947518,0x00A47524,0x0090F925,0x0011870E,
0x0025871B,
0x0015870C,0x0021870C,0x00918714,0x00A5871F,0x00A18718,0x00958718,0x03D06139,
0x0000101C,
0x03D04D3D,0x0011BA15,0x0091BA18,0x0021BA22,0x00A1BA17,0x0025BA1F,0x00A5BA2C,
0x0095BA17,
0x00148A1D,0x00208A1D,0x00248A15,0x00908A19,0x00948A18,0x00A08A18,0x00A48A17,
0x0010393D,
```

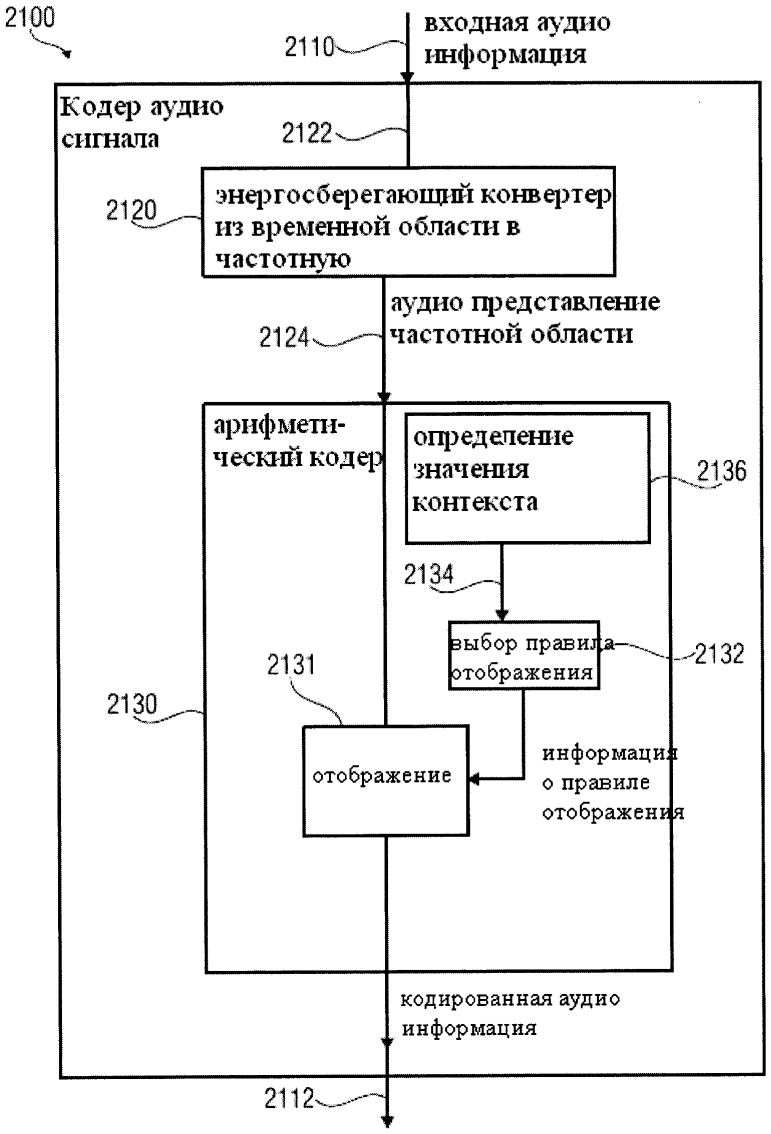
ФИГ. 20 (1)

40

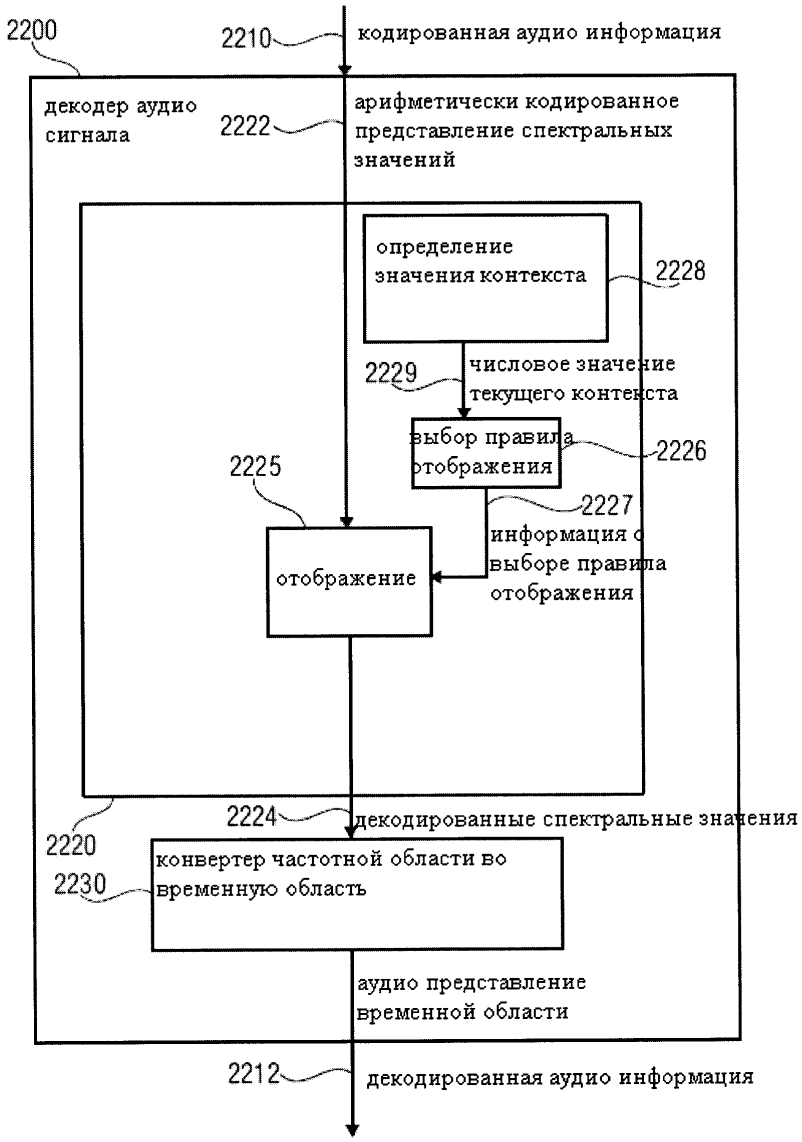
Аудио кодер, аудио декодер, способ кодирования аудио информации, способ декодирования аудио информации и компьютерная программа, использующая зависимое от диапазона арифметическое кодирующее правило отображения

```
0x0010C70C,0x0090C718,0x0014C715,0x0094C717,0x00A0C717,0x00A4C72B,0x00010E25,
0x00002F2D,
0x00107605,0x00907615,0x00147610,0x00207612,0x0024761D,0x00A07614,0x00947614,
0x00A47618,
0x0110762F,0x0090BD1C,0x00CDD938,0x0000FA02,0x0090FA2E,0x0000110B,0x03D0113B,
0x00A5C52E,
0x0095C52C,0x0011BB1D,0x0091BB18,0x0021BB19,0x0014950B,0x00A1BB17,0x0025BB18,
0x00A5BB2C,
0x0095BB17,0x00909523,0x00108B0E,0x00148B16,0x00208B0C,0x00248B1D,0x00908B14,
0x00948B18,
0x00A08B18,0x00A48B17,0x00010F25,0x0000302B,0x00107706,0x00907716,0x00147711,
0x00207713,
0x0024770E,0x00A07715,0x0094771D,0x00A47714,0x0091052F,0x00110528,0x0090BE1C,
0x0015052D,
0x03D06D3D,0x0000FB01,0x0090FB2D,0x0025890B,0x00A5892C,0x00158914,0x0021891C,
0x0095892B,
0x0011C619,0x0025C61F,0x00A5C62C,0x0095C617,0x00117516,0x0021750B,0x0015750B,
0x00917519,
0x0025751C,0x00A17524,0x00957520,0x00A5752C,0x0014961D,0x0090961F,0x0090C917,
0x00A4C92E,
0x0000312B,0x00D0313A,0x03D0313D,0x0090BF1C,0x0091062F,0x0010B50B,0x0090B517,
0x00A0B52B,
0x00A4B52C,0x0000FC1C,0x00258A1D,0x00A58A17,0x00158A15,0x00218A15,0x00918A18,
0x00A18A17,
0x00958A17,0x00013938,0x00001327,0x0011C715,0x0025C718,0x00A5C72C,0x0095C717,
0x0011760E,
0x0021760C,0x00917615,0x0015760C,0x0025760B,0x00A17622,0x00957622,0x00A57617,
0x00005031,
0x00908D24,0x0090CA1F,0x00A4CA2C,0x0010790D,0x00907919,0x0020791D,0x0014791D,
0x00247914,
0x00A0791F,0x0094791F,0x00A4792B,0x00CE2D38,0x0010B60C,0x0090B618,0x0014B615,
0x0020B615,
0x00A0B61F,0x0094B61F,0x0024B619,0x00A4B617,0x0110B632,0x00258B1B,0x00158B1D,
0x00218B1D,
0x00A58B17,0x00918B18,0x00A18B17,0x00958B1F,0x0090FD1E,0x00013A04,0x0000142F,
0x003A9D1C,
0x00BA9D2D,0x0011770F,0x0021770E,0x00157710,0x0091771D,0x0025771B,0x00A17714,
0x00957714,
0x00A57718,0x00005131,0x03D0513C
};
```

ФИГ. 20 (2)


2100
2110
входная аудио информация
Кодер аудио сигнала
2122
2120
энергосберегающий конвертер из временной области в частотную
аудио представление частотной области
2124
арифметический кодер
определение значения контекста
2136
2134
выбор правила отображения
2132
2131
2130
отображение
информация о правиле отображения
кодированная аудио информация
2112

Фиг. 21


2200
2210
кодированная аудио информация
декодер аудио сигнала
2222
арифметически кодированное представление спектральных значений
определение значения контекста
2228
2229
числовое значение текущего контекста
выбор правила отображения
2226
2225
2227
информация о выборе правила отображения
отображение
2224
декодированные спектральные значения
2220
2230
конвертер частотной области во временную область
аудио представление временной области
2212
декодированная аудио информация

Фиг. 22