



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2008108075/07, 05.07.2004**

(24) Дата начала отсчета срока действия патента:
05.07.2004

Приоритет(ы):

(30) Конвенционный приоритет:
03.07.2003 US 60/485,207

Номер и дата приоритета первоначальной заявки,
из которой данная заявка выделена:
2006103074 05.07.2004

(43) Дата публикации заявки: **10.09.2009** Бюл. № 25

(45) Опубликовано: **10.03.2013** Бюл. № 7

(56) Список документов, цитированных в отчете о
поиске: **EP 0940986 A2, 08.09.1999. US 2003117529**
A1, 26.06.2003. US 6356707 B1, 12.03.2002. US
2003002858 A1, 02.01.2003. US 6470460 B1,
22.10.2002. EP 0924934 A1, 23.06.1999. WO
02075739 A1, 26.09.2002. EP 1035735 A1,
13.09.2000. US 6580869 B1, 17.06.2003. RU
97106830 A, 20.04.1999. RU 2179377 C2,
10.02.2002.

Адрес для переписки:

129090, Москва, ул. Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. Ю.Д.Кузнецову,
рег.№ 595

(72) Автор(ы):

МАККРОССАН Джозеф (US),
ОКАДА Томоюки (JP),
МОТИНАГА Казухиро (JP)

(73) Патентообладатель(и):

ПАНАСОНИК КОРПОРЕЙШН (JP)

**(54) НОСИТЕЛЬ ЗАПИСИ, УСТРОЙСТВО ВОСПРОИЗВЕДЕНИЯ, СПОСОБ ЗАПИСИ,
ИНТЕГРАЛЬНАЯ СХЕМА, ПРОГРАММА И СПОСОБ ВОСПРОИЗВЕДЕНИЯ**

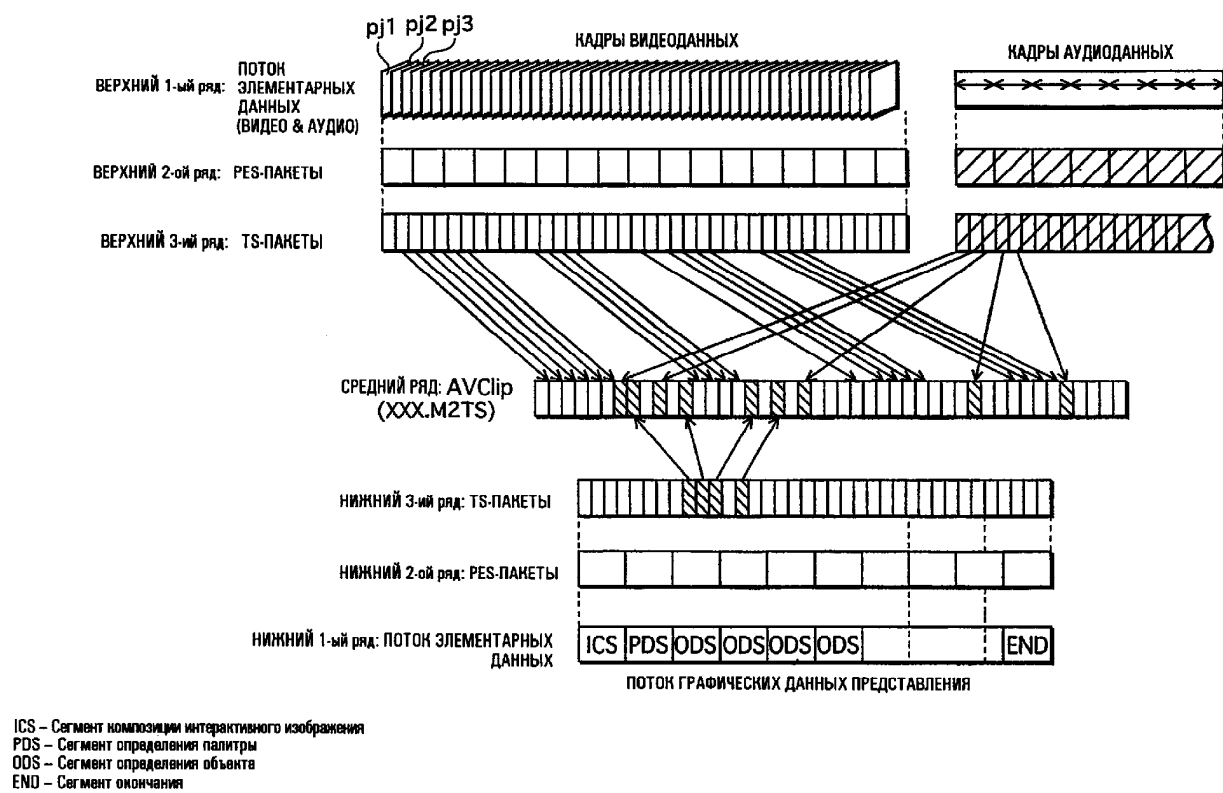
(57) Реферат:

Изобретение относится к носителю записи, такому как BD-ROM (Постоянное запоминающее устройство на диске Blue-ray), в частности к технологии для осуществления отображения субтитров и интерактивного отображения при помощи графических средств. Техническим результатом является создание носителя записи, который реализует высокий уровень разрешения отображения графических данных. Указанный технический результат достигается тем, что поток AVClip, записанный на диск BD-ROM, получают путем

мультиплексирования потока графических данных и потока видеоданных. Поток графических данных представляет собой последовательность PES-пакетов, которая включает в себя 1) PES-пакеты, хранящие графические данные (ODS-сегменты) (Сегменты определения объекта), и 2) PES-пакеты, хранящие управляющую информацию (PCS-сегменты) (Сегменты композиции представления). В каждом ODS-сегменте значения DTS-метки (Временной метки декодирования) и PTS-метки (Временной метки представления) указывают момент времени

начала декодирования для соответствующих графических данных и момент времени конца декодирования для соответствующих графических данных соответственно. В каждом PCS-сегменте значение PTS-метки

указывает момент времени, в который соответствующие декодированные графические данные объединяются с потоком видеоданных. 5 н.п. ф-лы, 80 ил.



ФИГ.41



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 477 574** (13) **C2**

(51) Int. Cl.

H04N 5/92 (2006.01)

G11B 27/10 (2006.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: **2008108075/07, 05.07.2004**

(24) Effective date for property rights:
05.07.2004

Priority:

(30) Convention priority:
03.07.2003 US 60/485,207

Number and date of priority of the initial application,
from which the given application is allocated:
2006103074 05.07.2004

(43) Application published: **10.09.2009 Bull. 25**

(45) Date of publication: **10.03.2013 Bull. 7**

Mail address:

**129090, Moskva, ul. B.Spasskaja, 25, str.3, OOO
"Juridicheskaja firma Gorodisskij i Partnery",
pat.pov. Ju.D.Kuznetsovu, reg.№ 595**

(72) Inventor(s):

**MAKKROSSAN Dzhozef (US),
OKADA Tomojuki (JP),
MOTINAGA Kazukhiro (JP)**

(73) Proprietor(s):

PANASONIK KORPOREhJShN (JP)

(54) **RECORDING MEDIUM, REPRODUCTION APPARATUS, RECORDING METHOD, INTEGRATED CIRCUIT, PROGRAM AND REPRODUCTION METHOD**

(57) Abstract:

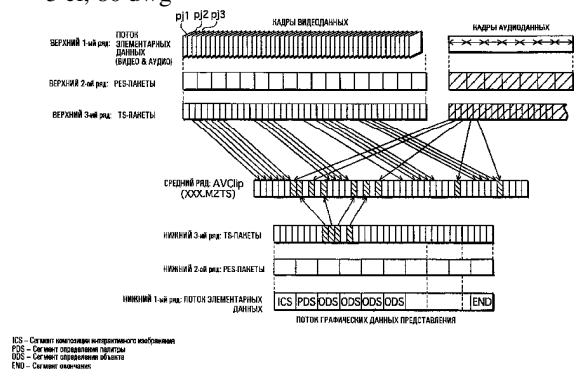
FIELD: information technology.

SUBSTANCE: AVClip recorded in BD-ROM is obtained by multiplexing a graphics stream and a video stream. The graphics stream is a PES packet sequence which includes 1) PES packets storing graphics data (ODS) (Object Definition Segments) and 2) PES packets storing control information (PCS) (Presentation Composition Segments). In each ODS, values of DTS (Decode Time Stamp) and PTS (Presentation Time Stamp) indicate time of decoding start for corresponding graphics data, and time of decoding end for corresponding graphics data, respectively. In each PCS, the value of PTS indicates the time when corresponding decoded graphics data are

combined with the video stream.

EFFECT: design of a recording medium with high resolution of displaying graphics data.

5 cl, 80 dwg



ICS - Сигнал синхронизации интерпретации композиции
PCS - Сигнал представления времени
ODS - Сигнал определения объектов
END - Сигнал окончания

Фиг.41

Область техники, к которой относится изобретение

Настоящее изобретение относится к носителю записи, такому как BD-ROM (Постоянное запоминающее устройство на диске, использующем технологию Blu-ray). В частности, настоящее изобретение относится к технологии для осуществления

Уровень техники

Отображение субтитров при помощи графических средств имеет важное предназначение, заключающееся в передаче значения слов, произносимых персонажами произведения, людям в любом регионе мира. Одной традиционной технологией осуществления отображения субтитров является наложение субтитров по стандарту ETSI EN 300 743 (где ETSI обозначает Европейский институт стандартов в области телекоммуникаций). Наложение субтитров заключается в том, что поток видеоданных подлежит воспроизведению вместе с отображением субтитров, осуществляемым при помощи графических средств. При этом графические данные, соответствующие субтитрам, отображаются в виде потока данных стандарта MPEG2 (Стандарт Экспертной группы по кодированию движущихся изображений). Поток данных представляет собой последовательность PES-пакетов, в которой каждый PES-пакет имеет PTS (временную метку представления). Стандарт ETSI EN 300 743 определяет моменты времени отображения субтитров при наложении субтитров. Этот стандарт устанавливает синхронизацию между кинофильмом и графическими средствами, при которой графические средства отображаются при отображении соответствующих изображений в потоке видеоданных.

В случае, когда наложение субтитров должно быть обеспечено для носителя BD-ROM, существует потребность в дальнейшем повышении уровня разрешения графических средств. Если описать это более конкретно, то желательно повысить уровень разрешения до уровня 1920×1080. Однако реализация столь высокого разрешения несет с собой огромное количество нагрузки по декодированию при воспроизведении.

Стандарт ETSI EN 300 743 определяет средства управления, предназначенные для выполнения декодирования в момент времени, назначенный меткой PTS, и для немедленного отображения его результата. В случае, когда это применяется, в устройстве воспроизведения в моменты времени, непосредственно предшествующие отображению, будет концентрироваться большое количество нагрузки по декодированию. Такая концентрация нагрузки делает необходимым высокий уровень аппаратно-программных возможностей устройства воспроизведения, такой чтобы реализовать отображение графических данных. Если такое условие становится обязательным для устройства воспроизведения, то производственная себестоимость устройств воспроизведения значительно вырастет, что создаст препятствие для широкого использования таких устройств воспроизведения.

Раскрытие изобретения

Задача настоящего изобретения заключается в том, чтобы предложить носитель записи, который реализует высокий уровень разрешения отображения графических данных, равно как позволяет избежать повышения производственной себестоимости.

При этом, если для достижения вышеупомянутой задачи вводится эта концепция буферизации, то технические специалисты, занимающиеся разработкой такого устройства воспроизведения, будут в замешательстве относительно того размера

памяти, который должен быть установлен для того, чтобы гарантировать нормальное функционирование. В то же время технические специалисты, которые производят наложение субтитров, также будут испытывать беспокойство по поводу того, смогут ли наложенные ими субтитры быть гарантированно воспроизведены устройством воспроизведения. Все это объясняется тем, что заполнение памяти при этой буферизации будет изменяться хронологически в ходе воспроизведения по временной шкале воспроизведения. Если хронологическое изменение в заполнении памяти остается неизвестным, то беспокойство этих технических специалистов не будет устранено.

Таким образом, для того чтобы достичь вышеописанных задач, настоящее изобретение предлагает носитель записи, хранящий на себе поток цифровых данных, полученный посредством мультиплексирования потока графических данных и потока видеоданных, при этом поток графических данных представляет собой последовательность пакетов, которая включает в себя пакет данных, хранящий графические данные, и управляющий пакет, хранящий управляющую информацию, при этом

пакет данных имеет временную метку, значение которой указывает время декодирования графических данных,

управляющий пакет включает в себя временную метку представления и временную метку декодирования,

управляющая информация включает в себя информацию о типе, причем когда информация о типе указывает начало управления памятью, значение временной метки представления указывает на время, в которое декодированные графические данные объединяются с потоком видеоданных и отображаются с потоком видеоданных, и значение временной метки декодирования указывает момент времени на временной шкале воспроизведения потока цифровых данных, который соответствует началу управления памятью, и время, в которое управляющая информация считывается в память, и

когда информация о типе представляет собой информацию из числа: информации, указывающей различие с предыдущей управляющей информацией; дубликата предыдущей управляющей информации; и информации, указывающей, что управление памятью продолжается после другого потока цифровых данных, значение временной метки декодирования указывает время, в которое следует считывать управляющую информацию в память устройства воспроизведения, и значение временной метки представления указывает время, в которое следует объединить декодированные графические данные с потоком видеоданных и отобразить декодированные графические данные, объединенные с потоком видеоданных.

Период, в который декодируются графические данные, указывается временной меткой пакета, хранящего графические данные, а отображение графических данных определяется значением временной метки, присвоенной соответствующей управляющей информации. Следовательно, в настоящем изобретении на временной шкале воспроизведения определяется состояние «уже декодировано, но еще не отображено», иначе говоря, состояние, в котором разуплотненные графические данные находятся в буфере.

Определив такой период буферизации, становится возможным избежать концентрации огромного количества нагрузки по декодированию в одном моменте времени. Кроме того, если использование ресурсов аппаратного обеспечения в целях

декодирования одновременно вступает в конфликт с другой обработкой данных, то период буферизации может быть организован таким образом, чтобы переместить период декодирования графических данных и тем самым избежать такого рода конфликт.

5 В соответствии с этой конструкцией начало управления памятью указывается посредством временной метки декодирования из пакета, хранящего управляющую информацию. Следовательно, при обращении к временной метке декодирования становится возможным узнать, в какой момент времени на временной шкале
10 воспроизведения должен быть полностью очищен каждый буфер этой модели декодера. Если эта точка полной очистки рассматривается как начальная точка управления памятью, то легко определить хронологическое изменение в заполнении буфера, который хранит управляющую информацию, буфера, который хранит графические данные перед тем, как они декодируются, и буфера, который хранит
15 графические данные после их декодирования. Изменяя значение этой временной метки декодирования, можно регулировать хронологическое изменение состояния буферов. В соответствии с таким регулированием становится возможным избежать переполнения буфера в устройстве воспроизведения. А следовательно, становится
20 возможным легко воплотить аппаратное/программное обеспечение на стадии разработки устройства воспроизведения.

Кроме того, поскольку становится просто определять и регулировать хронологическое изменение, то соответственно упрощается и проверка того, удовлетворяет ли поток графических данных, полученный посредством авторского
25 процесса, ограничениям модели декодера, которую предполагает стандарт BD-ROM. Следовательно, лицо, ответственное за авторский процесс, может продолжать свою авторскую работу при условии, что обеспечено нормальное функционирование графических компонентов его произведения.

30 Предполагая модель декодера BD-ROM, для осуществления настоящего изобретения необходимым станет еще один составляющий элемент. В этой модели декодера BD-ROM основная часть декодера (то есть процессор) графических данных независима от основной части контроллера (то есть от контроллера), предназначенного для обновления графических данных. Причина, по которой
35 основная часть декодера предусматривается независимой от основной части обновляющего контроллера, заключается в том, чтобы выполнять заблаговременное обновление, такое как поэтапное отображение и удаление графических данных, что, например, полезно для случая, когда графические данные
40 представляют собой субтитры. Когда основная часть обновляющего контроллера независима от основной части контроллера, будет требоваться, чтобы связь процессор-контроллер была более тесной. Это объясняется тем, что после завершения процессором декодирования графических данных контроллер должен выполнять обновление без задержки.

45 То, каким образом контроллер уведомляется о завершении процессором декодирования, зависит от того, каким образом реализованы процессор и контроллер в данном устройстве. Если процессор и контроллер реализованы в виде программ, то уведомление будет выполняться за счет взаимодействия процессов.
50 Если процессор и контроллер реализованы в виде независимых друг от друга компонентов программного обеспечения, то уведомление будет выполняться посредством сигнала прерывания. Длительность временной задержки такого уведомления также зависит от того, каким образом реализовано устройство. Если

реализация делает необходимой большую временную задержку уведомления, то налицо будет ситуация, при которой графические данные не могут быть синхронизированы со скоростью отображения кинофильма.

Для того чтобы избежать возникновения такого рода ситуации, желательно иметь конструкцию, в которой значение временной метки представления получается посредством суммирования некоторого заранее заданного значения со значением временной метки декодирования, где это заранее заданное значение основано на более длительном промежутке времени из числа: промежутка времени, требующегося для очистки экрана, и промежутка времени, требующегося для декодирования графических данных; и промежутка времени, требующегося для записи графических данных на экран.

Временная метка представления пакета, хранящего графические данные, указывает на время окончания декодирования, а временная метка представления пакета, хранящего управляющую информацию, указывает на время, полученное посредством суммирования заранее заданного промежутка времени с временем окончания декодирования. Следовательно, за счет только лишь обращения к временным меткам представления контроллер может выполнять обновление в надлежащие моменты времени, не получая при этом от процессора никакого уведомления о завершении декодирования графических данных. Если выполняется такое обновление, то становится возможным обеспечить синхронизацию обновления со скоростью отображения кинофильма, независимо от способа воплощения устройства воспроизведения.

Поскольку более тесная связь процессор-контроллер реализована независимо от способа воплощения процессора-контроллера в устройстве воспроизведения, то становится возможным поддерживать некоторую степень гибкости при конструировании этого устройства, равно как и способствовать производству устройств с низкими затратами.

Краткое описание чертежей

Фиг.1 - иллюстрация примера использования носителя записи в соответствии с данным изобретением.

Фиг.2 - иллюстрация структуры BD-ROM (Постоянного запоминающего устройства на дисках, использующих технологию Blue-ray).

Фиг.3 - схема, иллюстрирующая в схематическом виде структуру потока AVClip (потока данных аудио-видео фрагментов).

Фиг.4А - иллюстрация структуры потока графических данных представления.

Фиг.4В - иллюстрация PES-пакета, полученного после преобразования функциональных сегментов.

Фиг.5 - иллюстрация логической структуры, составленной из различных видов функциональных сегментов.

Фиг.6 - иллюстрация связи между местом отображения субтитра и сверхкадром.

Фиг.7А - иллюстрация синтаксической структуры, предназначенной для определения графического объекта в Сегменте определения объекта (ODS-сегмента).

Фиг.7В - иллюстрация синтаксической структуры Сегмента определения палитры (PDS-сегмента).

Фиг.8А - иллюстрация синтаксической структуры Сегмента определения окна (WDS-сегмента).

Фиг.8В - иллюстрация синтаксической структуры Сегмента композиции представления (PCS-сегмента).

Фиг.9 - иллюстрация примера описания Отображаемого набора при показе субтитров.

Фиг.10 - иллюстрация примера описания WDS-сегмента и PCS-сегмента в наборе DS1.

Фиг.11 - иллюстрация примера описания PCS-сегмента в наборе DS2.

Фиг.12 - иллюстрация примера описания PCS-сегмента в наборе DS3.

Фиг.13 - проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении эффекта появления/исчезновения изображения объекта.

Фиг.14 - проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении эффекта постепенных выхода их затемнения/ухода в затемнение.

Фиг.15 - проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении прокрутки.

Фиг.16 - проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении эффекта нанесения/стирания изображения объекта.

Фиг.17 - схема, показывающая сравнение двух случаев: окна, имеющего четыре графических объекта, и окна, имеющего два графических объекта.

Фиг.18 - иллюстрация примера алгоритма, предназначенного для расчета продолжительности декодирования.

Фиг.19 - блок-схема алгоритма, показанного на Фиг.18.

Фигуры 20А и В - блок-схемы алгоритма, показанного на Фиг.18.

Фиг.21А - иллюстрация случая, при котором каждое окно имеет Сегмент определения объекта.

Фигуры 21В и С - временные диаграммы, показывающие порядки группирования значений, ссылки на которые имеются на Фиг.18.

Фиг.22А - иллюстрация случая, при котором каждое окно имеет два Сегмента определения объекта.

Фигуры 22В и С - временные диаграммы, показывающие порядки соотношения значений, ссылки на которые имеются на Фиг.18.

Фиг.23А - описание случая, при котором каждое из двух окон включает в себя ODS-сегмент.

Фиг.23В - иллюстрация случая, при котором период (2) декодирования больше, чем сумма периода (1) очистки и периода (31) записи.

Фиг.23С - иллюстрация случая, при котором сумма периода (1) очистки и периода (31) записи больше, чем период (2) декодирования длиннее.

Фиг.24 - иллюстрация хронологического изменения при обновлении, описанного в примере в настоящей спецификации.

Фиг.25А - иллюстрация четырех отображаемых наборов, которые описаны таким образом, чтобы выполнять рассматриваемое выше обновление.

Фиг.25В - временная диаграмма, показывающая настройки DTS и PTS функциональных сегментов, включенных в эти четыре отображаемых набора.

Фиг.26 - иллюстрация внутренней структуры устройства воспроизведения в соответствии с настоящим изобретением.

Фиг.27 - иллюстрация величин скоростей записи Rx, Rc, и Rd, емкости Графической матрицы (8), Буфера (13) закодированных данных, Буфера (15) объектов и Буфера (16) композиции.

Фиг.28 - временная диаграмма, иллюстрирующая конвейерную обработку данных, осуществляемую этим устройством воспроизведения.

Фиг.29 - временная диаграмма для конвейерной обработки данных в случае, при котором декодирование ODS-сегмента заканчивается до завершения очистки Графической матрицы.

Фиг.30 - временная диаграмма, показывающая хронологическое изменение количества аккумулированных данных на Графической матрице.

Фиг.31 - блок-схема алгоритма, иллюстрирующая процесс операции загрузки функционального сегмента.

Фиг.32 - демонстрация примера мультиплексирования.

Фиг.33 - иллюстрация способа, которым осуществляется загрузка набора DS10 в буфер (13) закодированных данных, входящий в состав устройства воспроизведения.

Фиг.34 - иллюстрация случая, при котором выполняется нормальное воспроизведение.

Фиг.35 - иллюстрация загрузки наборов DS1, DS10 и DS20 при нормальном воспроизведении, выполняемом так, как показано на Фиг.34.

Фиг.36 - блок-схема алгоритма, показывающая процесс, выполняемый Графическим контроллером (17).

Фиг.37 - блок-схема алгоритма, показывающая процесс, выполняемый Графическим контроллером (17).

Фиг.38 - блок-схема алгоритма, показывающая процесс, выполняемый Графическим контроллером (17).

Фиг.39 - иллюстрация конвейерного процесса данного устройства воспроизведения, который основан на PTS-метке PDS-сегмента.

Фиг.40 - схема, показывающая значение END-сегмента в конвейерном процессе данного устройства воспроизведения.

Фиг.41 - схема, иллюстрирующая в схематическом виде структуру потока AVClip (потока данных аудио-видео фрагментов) в соответствии со вторым вариантом осуществления изобретения.

Фиг.42А и Фиг.42В - схемы, относящиеся к интерактивному экранному изображению в соответствии со вторым вариантом осуществления изобретения.

Фиг.43 - иллюстрация структуры данных Сегмента композиции интерактивного изображения.

Фиг.44 - демонстрация связи между ODS-сегментом, включенным в набор DS_n, и ICS-сегментом.

Фиг.45 - демонстрация композиции экранного изображения в момент отображения данных «pt1» произвольного кинокадра.

Фиг.46 - демонстрация примера настройки для информации о кнопке в ICS-сегменте.

Фиг.47 - изменение состояний кнопки А - кнопки D.

Фиг.48 - приводимая в качестве одного примера иллюстрация изображений ODS11, 21, 31 и 41.

Фиг.49 - приводимая в качестве одного примера иллюстрация изображений ODS11-19 для кнопки А.

Фиг.50 - иллюстрация групп состояния кнопок и порядок расположения ODS-сегментов в Отображаемом наборе.

Фиг.51 - иллюстрация изменения состояния интерактивного экранного изображения, на котором расположены группы состояния кнопок, показанные на

Фиг.50.

Фиг.52 - иллюстрация порядка расположения ODS-сегментов в отображаемом наборе.

Фиг.53 - демонстрация различий в расположении ODS-сегментов в группе S-ODS-сегментов между случаем, когда default_selected_button_number=0 (номер кнопки, выбираемой по умолчанию = 0), и случаем, когда default_selected_button_number = кнопка B.

Фигуры 54А и 54В - демонстрация значения $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ в случае, при котором N-ODS-сегменты включают в свое число множество ODS-сегментов, составляющих кнопки A-D, и S-ODS-сегменты включают в свое число множество ODS-сегментов, составляющих кнопки A-D.

Фиг.55 - демонстрация временной диаграммы синхронизированного отображения, осуществляемого посредством ICS-сегмента.

Фиг.56 - демонстрация того, каким образом DTS-метка и PTS-метка устанавливаются в случае, при котором первоначальное отображение интерактивного экранного изображения составляет множество ODS-сегментов, и при котором default_selected_button (выбираемая по умолчанию кнопка) является действительной (то есть существующей на самом деле).

Фиг.57 - демонстрация того, каким образом DTS-метка и PTS-метка устанавливаются в случае, при котором первоначальное отображение интерактивного экранного изображения составляет множество ODS-сегментов, и при котором default_selected_button (выбираемая по умолчанию кнопка) является недействительной (то есть не существующей на самом деле).

Фиг.58 - иллюстрация содержимого Буфера (15) объектов в сопоставлении с Графической матрицей.

Фиг.59 - иллюстрация операции, выполняемой Графическим контроллером (17) во время первоначального отображения.

Фиг.60 - иллюстрация операции, выполняемой Графическим контроллером (17) при выполнении обновления интерактивного экранного изображения в соответствии с первым действием пользователя (командой MoveRight («Перемещение вправо»)).

Фиг.61 - иллюстрация операции, выполняемой Графическим контроллером (17) при выполнении обновления интерактивного экранного изображения в соответствии с первым действием пользователя (командой MoveDown («Перемещение вниз»)).

Фиг.62 - иллюстрация операции, выполняемой Графическим контроллером (17) при выполнении обновления интерактивного экранного изображения в соответствии с первым действием пользователя (командой Activated («Активировано»)).

Фиг.63 - временная диаграмма, иллюстрирующая конвейерную обработку данных, выполняемую данным устройством воспроизведения.

Фиг.64 - временная диаграмма, иллюстрирующая конвейерную обработку данных, выполняемую данным устройством воспроизведения в случае, при котором кнопка, выбираемая по умолчанию, изменяется динамически.

Фиг.65 - временная диаграмма, иллюстрирующая хронологические изменения в заполнении Графической матрицы (8), Буфера (15) объектов, Буфера (13) закодированных данных и Буфера (16) композиции.

Фиг.66 - блок-схема алгоритма, показывающая процесс операции загрузки Сегмента.

Фиг.67 - демонстрация примера мультиплексирования.

Фиг.68 - иллюстрация способа, которым осуществляется загрузка набора DS10 в

буфер (13) закодированных данных, входящий в состав устройства воспроизведения.

Фиг.69 - иллюстрация случая, при котором выполняется нормальное воспроизведение.

Фиг.70 - иллюстрация загрузки наборов DS1, DS10 и DS20 при нормальном воспроизведении, выполняемом так, как показано на Фиг.69.

Фиг.71 - блок-схема алгоритма, иллюстрирующая основную программу обработки данных, выполняемой Графическим контроллером (17).

Фиг.72 - блок-схема алгоритма, иллюстрирующая обработку данных для реализации управления синхронизацией, использующего временную метку.

Фиг.73 - блок-схема алгоритма, показывающего процесс операции по записи в Графическую матрицу (8).

Фиг.74 - блок-схема алгоритма, иллюстрирующая обработку автоматической активации для кнопки, выделяемой по умолчанию.

Фиг.75 - блок-схема алгоритма, показывающего процесс отображения анимации.

Фиг.76 - блок-схема алгоритма, иллюстрирующая процесс операции «Операция пользователя».

Фиг.77 - блок-схема алгоритма, иллюстрирующая процесс операции смены текущей кнопки.

Фиг.78 - блок-схема алгоритма, иллюстрирующая процесс операции ввода числового значения.

Фиг.79 - иллюстрация способа изготовления BD-ROM, при котором осуществляется запись PCS-сегмента, рассмотренного в первом варианте осуществления изобретения.

Фиг.80 - иллюстрация способа изготовления BD-ROM, при котором осуществляется запись PCS-сегмента, рассмотренного во втором варианте осуществления изобретения.

Осуществление изобретения

(Первый вариант осуществления изобретения)

Ниже будет описан первый вариант воплощения носителя записи в соответствии с настоящим изобретением.

На Фиг.1 проиллюстрирован пример использования этого носителя записи. На чертеже BD-ROM (100) (Постоянное запоминающее устройство на диске, использующем технологию Blue-ray) представляет собой носитель записи в соответствии с настоящим изобретением. BD-ROM (100) используется для предоставления данных, представляющих собой кинематографическую продукцию, в Системе домашнего театра, состоящей из устройства (200) воспроизведения, телевизионного приемника (300) и пульта (400) дистанционного управления.

Носитель записи, соответствующий настоящему изобретению, изготавливается посредством усовершенствования прикладного уровня BD-ROM. Фиг.2 иллюстрирует структуру BD-ROM.

На этом чертеже диск BD-ROM показан в нижней части чертежа, а дорожка на диске BD-ROM показана над диском BD-ROM. Эта дорожка на самом деле на диске имеет форму спирали, но на чертеже показана в виде линии. Дорожка включает в себя область подвода, область тома и область отвода. Область тома на этом рисунке имеет физический уровень, уровень файловых систем и прикладной уровень. В верхней части рисунка с использованием структуры каталога проиллюстрирован прикладной формат BD-ROM. Как показано на рисунке, диск BD-ROM под корневым каталогом имеет каталог BDMV, а этот каталог BDMV содержит файл,

предназначенный для хранения потока AVClip (потока данных аудио-видео фрагментов) и имеющий расширение M2TS (XXX.M2TS), файл, предназначенный для хранения административной информации для потока AVClip и имеющий расширение CLPI (XXX.CLPI), и файл, предназначенный для определения логического списка воспроизведения (PL) для потока AVClip и имеющий расширение MPLS (YYY.MPLS). Формируя вышеописанный прикладной формат, можно изготовить носитель записи, соответствующий настоящему изобретению. В случае, когда имеется более чем один файл каждого типа, предпочтительно, чтобы в каталоге BDMV были предусмотрены три каталога, озаглавленные STREAM, CLIPINF и PLAYLIST, с тем, чтобы хранить файлы с одним и тем же расширением в одном каталоге. В частности, желательно хранить файлы с расширением M2TS в каталоге STREAM, файлы с расширением CLPI в каталоге CLIPINF, а файлы с расширением MPLS в каталоге PLAYLIST.

Ниже приводится описание потока AVClip (XXX.M2TS) из вышеописанного прикладного формата.

Поток AVClip (XXX.M2TS) представляет собой поток цифровых данных в формате MPEG-TS (где TS обозначает Транспортный поток), полученный посредством мультиплексирования потока видеоданных, по меньшей мере, одного потока аудиоданных и потока графических данных представления. Поток видеоданных представляет кадры фильма, поток аудиоданных представляет звук фильма, а поток графических данных представления представляет субтитры фильма. Фиг.3 представляет собой схему, иллюстрирующую в схематическом виде структуру потока AVClip.

Поток AVClip (XXX.M2TS) имеет следующую структуру. Поток видеоданных, состоящий из множества кадров видеоданных (кинокадр p_{j1}, p_{j2} и p_{j3}), и поток аудиоданных, состоящий из множества кадров аудиоданных (верхний ряд на чертеже), соответственно преобразуются в последовательность PES-пакетов (второй ряд на чертеже), а затем в последовательность TS-пакетов (пакетов транспортного потока) (третий ряд на чертеже). Поток графических данных представления (нижний ряд на чертеже) преобразуется в последовательность PES-пакетов (второй снизу ряд на чертеже), а затем в последовательность TS-пакетов (третий снизу ряд на чертеже). Три последовательности TS-пакетов мультиплексируются, и таким образом составляется поток AVClip (XXX.M2TS).

На чертеже мультиплексируется только один поток графических данных представления. Однако в случае, когда диск BD-ROM совместим со множеством языков, для составления потока AVClip мультиплексируется поток графических данных представления для каждого языка. Поток AVClip, составленный вышеописанным способом, разделяется на более чем один экстенд, подобно обычным компьютерным файлам, и сохраняется в областях на диске BD-ROM.

Далее рассматривается поток графических данных представления. Фиг.4А иллюстрирует структуру потока графических данных представления. Верхний ряд показывает последовательность TS-пакетов, мультиплексируемых в поток AVClip. Второй сверху ряд показывает последовательность PES-пакетов, которая составляет поток графических данных. Последовательность PES-пакетов образуется путем извлечения полезных данных из TS-пакетов, имеющих заданный PID (идентификатор пакета) и соединения этих извлеченных полезных данных.

Третий сверху ряд показывает структуру потока графических данных. Поток графических данных состоит из функциональных сегментов, именуемых Сегментом

композиции представления (PCS-сегментом), Сегментом определения окна (WDS-сегментом), Сегментом определения палитры (PDS-сегментом), Сегментом определения объекта (ODS-сегментом) и Сегментом конца отображаемого набора (END-сегментом). Среди вышеперечисленных функциональных сегментов PCS-сегмент называется сегментом композиции экранного изображения, а WDS-сегмент, PDS-сегмент, ODS-сегмент и END-сегмент называются сегментами определения. PES-пакет и каждый из функциональных сегментов имеют соответствие «один одному» или «один множеству». Иначе говоря, один функциональный сегмент записывается на диск BD-ROM либо после преобразования в один PES-пакет, либо после разделения на фрагменты и преобразования в более чем один PES-пакет.

Фиг.4В иллюстрирует PES-пакет, полученный посредством преобразования функциональных сегментов. Как показано на чертеже, PES-пакет состоит из заголовка пакета и полезных данных, и полезные данные представляют собой основную часть функционального сегмента. Заголовок пакета включает в себя DTS-метку (временную метку декодирования) и PTS-метку (временную метку представления), соответствующие функциональному сегменту. DTS-метка и PTS-метка, включенные в состав заголовка пакета, далее именуется DTS-меткой и PTS-меткой функционального сегмента.

Описанные выше различные виды функциональных сегментов составляют логическую структуру, проиллюстрированную на Фиг.5. Фиг.5 иллюстрирует логическую структуру, составленную из различных видов функциональных сегментов. На чертеже верхний ряд иллюстрирует Сверхкадры, средний ряд иллюстрирует Отображаемые наборы (DS-наборы), а нижний ряд иллюстрирует функциональные сегменты.

Каждый из DS-наборов, показанных в среднем ряду, представляет собой группу функциональных сегментов, которые составляют графические данные для одного экранного изображения, из числа всего множества функциональных сегментов, которые образуют поток графических данных. Пунктирные линии на чертеже показывают тот DS-набор, к которому принадлежат функциональные сегменты, находящиеся в нижнем ряду, и демонстрируют, что один DS-набор образован рядом функциональных сегментов: PCS, WDS, PDS, ODS и END. Устройство воспроизведения способно генерировать графические изображения для одного экранного изображения посредством считывания функциональных сегментов, образующих DS-набор.

Сверхкадры, приведенные в верхнем ряду, показывают периоды времени, и в течение одного Сверхкадра управление памятью осуществляется последовательно во временном отношении вдоль временной шкалы воспроизведения потока AVClip. Один Сверхкадр также представляет группу данных, которая предназначена для одного и того же периода времени. Упомянутая здесь память представляет собой Графическую матрицу, которая хранит графические данные для одного экранного изображения, и Буфер объектов, который хранит разуплотненные графические данные. Последовательность в управлении памятью означает, что полная очистка Графической матрицы или Буфера объектов не происходит во время Сверхкадра, и стирание и визуализация графических данных выполняются только в заранее заданной прямоугольной области на графической матрице («полная очистка» в данном случае означает сброс всего содержимого данных, сохраненных в матрице или буфере). Размер и местоположение этой прямоугольной области являются фиксированными на протяжении одного Сверхкадра. Коль скоро стирание и

визуализация графических данных выполняются только в заранее заданной прямоугольной области на Графической матрице, то гарантируется синхронное воспроизведение кинофильма и графических данных. Иначе говоря, Сверхкадр представляет собой участок на временной шкале воспроизведения, и на этом участке

5 гарантируется синхронное воспроизведение кинофильма и графических данных. При перемещении области, в которой стираются и визуализируются графические данные, в другое место, необходимо определить точку на временной шкале для перемещения этой области, и период после этой точки становится новым Сверхкадром. На

10 границе между двумя Сверхкадрами синхронное воспроизведение не гарантируется.

При просмотре реального фильма один Сверхкадр представляет собой период времени, в который субтитры отображаются в одной и той же прямоугольной области на экране. Фиг.6 иллюстрирует связь между местоположением субтитра и сверхкадрами. В примере, проиллюстрированном на чертеже, места, в которых

15 показаны пять субтитров: «На самом деле,...», «Я скрывала», «свои чувства.», «Я всегда», «любила тебя.», перемещаются в соответствии с изображением в фильме. В частности, субтитры «На самом деле,...», «Я скрывала» и «свои чувства.» видны в нижней части экрана, в то время как субтитры «Я всегда» и «любила тебя.» показаны

20 в верхней части экрана. Местоположение этой прямоугольной области перемещается для того, чтобы с учетом возможности зрительного восприятия фильма субтитры не мешали изображению при просмотре фильма на экране. Период времени, в течение которого субтитры видны в нижней части, представляет собой Сверхкадр 1, а последующий период времени, в течение которого субтитры видны в

25 верхней части, представляет собой Сверхкадр 2. Сверхкадры 1 и 2 каждый имеют отличную от другой область, в которой визуализируются субтитры. Эта область в Сверхкадре 1 представляет собой Окно 1, расположенное в нижней части экрана, и эта область в Сверхкадре 2 представляет собой Окно 2, расположенное в верхней

30 части экрана. Управление памятью является последовательным в каждом из Сверхкадров 1 и 2, и соответственно визуализация субтитров в Окнах 1 и 2 является синхронной с изображениями.

Далее описываются подробности, касающиеся Отображаемого набора (DS-набора).

Пунктирные линии hk11 и hk12, показанные на Фиг.5, указывают, к какому Сверхкадру принадлежит какой функциональный сегмент в среднем ряду. Последовательность DS-наборов: «Начало Сверхкадра», «Точка захвата» и «Нормальный случай» составляют Сверхкадр из верхнего ряда. «Начало

40 Сверхкадра», «Точка захвата», «Нормальный случай» и «Продолжение Сверхкадра» представляют собой типы DS-наборов, при этом порядок, в котором расположены между собой «Точка захвата» и «Нормальный случай» не имеет значения, и любой из них может быть впереди другого.

«Начало Сверхкадра» представляет собой DS-набор, который имеет

45 отображаемый результат «новое изображение», что указывает на начало нового Сверхкадра. По этой причине «Начало Сверхкадра» содержит все функциональные сегменты, необходимые для отображения новой композиции экранного изображения. «Начало Сверхкадра» предусматривается в месте, являющемся

50 целевой точкой для операции пропуска (части потока) в потоке AVClip, такой как раздел в фильме.

«Точка захвата» представляет собой DS-набор, который имеет отображаемый результат «регенерация отображения» и идентичен по содержимому используемому

для визуализации графических данных «Началу Сверхкадра», который является предыдущим DS-набором. «Точка захвата» не предусматривается в начальной точке Сверхкадра, но содержит все функциональные сегменты, необходимые для отображения новой композиции экранного изображения. Следовательно, возможно безотказное отображение графических данных при выполнении операции пропуска до «Точки захвата». Соответственно с помощью «Точки захвата» возможно составление экранного изображения в середине Сверхкадра.

«Точка захвата» предусматривается в месте, которое могло бы быть целевой точкой для операции пропуска. Примером такого места является место, которое могло бы быть указано при выполнении временного поиска. Временной поиск представляет собой операцию, производимую в ответ на ввод пользователем времени и предназначенную для начала воспроизведения с точки воспроизведения, соответствующей времени, указанному пользователем. Это время указывается приблизительно, например, посредством 10-минутных или 10-секундных интервалов, и соответственно точки, в которых начинается воспроизведение, предусматриваются, например, с 10-минутным интервалом или 10-секундным интервалом. Предусматривая «Точку захвата» в точках, в которых может начаться воспроизведение, можно гладко осуществлять воспроизведение после временного поиска.

«Нормальный случай» представляет собой DS-набор, который имеет отображаемый результат «обновление отображения» и содержит только элементы, отличные от предыдущей композиции экранного изображения. В частности, в случае, когда субтитры в отображаемом наборе DS_v являются теми же самыми, что и субтитры в отображаемом наборе, но экранное изображение отображается в случаях DS_v и DS_u различным образом, то DS_v-набор организуется таким образом, чтобы включать в себя только PCS-сегмент, и DS_v-набор делают набором «Нормальный случай». Благодаря этому становится ненужным предусматривать ODS-сегмент с тем же самым содержимым, что и содержимое ODS-сегмента в предыдущем DS-наборе, и может быть уменьшен объем данных на диске BD-ROM. С другой стороны, поскольку DS-набор в ситуации «Нормального случая» содержит только различия, то составить экранное изображение, используя для этого только DS-набор «Нормальный случай» невозможно.

«Продолжение Сверхкадра» указывает, что Сверхкадр продолжается, переходя через границу потоков AVClip. Если Состояние композиции одного отображаемого набора DS_n установлено как «Продолжение Сверхкадра», и если этот DS_n существует в потоке AVClip, отличном от потока, к которому относится отображаемый набор DS_{n-1}, расположенный непосредственно перед DS_n, то DS_n и DS_{n-1} будут принадлежать к одному и тому же Сверхкадру. Следовательно, даже если между этими двумя DS-наборами происходит ветвление потока AVClip, полной очистки графической матрицы/буфера объектов не будет.

Ниже будут подробно описаны Сегменты определения (ODS-сегмент, WDS-сегмент и PDS-сегмент). Сегмент определения объекта (ODS-сегмент) представляет собой функциональный сегмент, который определяет Графический объект. Сначала будет дано объяснение Графического объекта. Существенное преимущество потока AVClip, записанного на диск BD-ROM, заключается в том, что его разрешение столь же высокое, как разрешение класса «hi-vision», а следовательно, разрешение для Графического объекта устанавливается на уровне 1920×1080 пикселей. По причине этого высокого разрешения 1920×1080 пикселей возможно

четкое отображение на экране конкретного типа шрифта для данных субтитров. Что касается цветов субтитров, то выраженная в битах длина значения индекса для каждого пикселя (цветовой контраст по красному Cr, цветовой контраст по синему Cb, яркость Y и прозрачность T), составляет 8 битов, и, таким образом, для субтитров можно выбрать любые 256 цветов из полной цветовой палитры (16777216 цветов). Субтитры, реализованные посредством Графического объекта, визуализируются путем расположения текста на прозрачном фоне.

Синтаксическая структура ODS-сегмента, предназначенная для определения Графического объекта, показана на Фиг.7А. ODS-сегмент состоит из элементов: `segment_type` (тип сегмента), указывающего, что сегмент представляет собой ODS-сегмент; `segment_length` (длина сегмента), указывающего длину данных этого ODS-сегмента; `object_id` (идентификатор объекта), уникальным образом идентифицирующий Графический объект, соответствующий этому ODS-сегменту в Сверхкадре; `object_version_number` (номер версии объекта), указывающий версию этого ODS-сегмента внутри Сверхкадра; `last_in_sequence_flag` (признак последнего ODS-сегмента в последовательности) и `object_data_fragment` (фрагмент данных объекта), который представляет собой последовательность байтов, соответствующих части или всему Графическому объекту.

`Object_id` (идентификатор объекта) предназначен для уникальной идентификации Графического объекта, соответствующего этому ODS-сегменту в Сверхкадре. Сверхкадр потока графических данных содержит более чем один ODS-сегмент, имеющий одинаковый идентификатор. ODS-сегменты, имеющие одинаковый идентификатор, также имеют одинаковую ширину и высоту и получают общую область в Буфере объектов. После того, как один из ODS-сегментов, имеющих одинаковый идентификатор, считан в эту общую область, считанный ODS-сегмент затирается последующим ODS-сегментом, имеющим такой же идентификатор. Посредством затирания ODS-сегмента, считанного в Буфер объектов, последующим ODS-сегментом с таким же идентификатором, по мере того, как продолжается воспроизведение потока видеоданных, графические данные, поставляемые ODS, обновляются соответствующим образом. Ограничение на размер, заключающееся в том, что ширина и высота Графического объекта, имеющего такой же идентификатор, должна быть такой же, применимо только на протяжении одного Сверхкадра, а Графические объекты в различных Сверхкадрах могут иметь различные размеры.

Далее даются разъяснения относительно элементов `last_in_sequence_flag` (признак последнего ODS-сегмента в последовательности) и `object_data_fragment` (фрагмент данных объекта). В некоторых случаях из-за ограничения количества полезных данных, размещаемых в PES-пакете, невозможно хранить разуплотненные графические данные, составляющие субтитр, в одном ODS-сегменте. В таких случаях графические данные разбиваются на ряд последовательных фрагментов, и один фрагмент заносится в `object_data_fragment`. В случае, когда один Графический объект хранится в виде более чем одного фрагмента, каждый фрагмент, за исключением последнего фрагмента, имеет один и тот же размер. Последний фрагмент меньше чем или равен размеру предыдущего фрагмента. ODS-сегменты, содержащие эти фрагменты, появляются в DS-наборе в том же самом порядке следования, при этом конец последовательности указывается ODS-сегментом, имеющим `last_in_sequence_flag`. Хотя вышеописанная синтаксическая структура ODS-сегмента основана на той посылке, что фрагменты пакетируются от

предыдущего PES-пакета, фрагменты могут пакетироваться и, таким образом, что каждый PES-пакет содержит пробельную часть.

Далее описан Сегмент определения палитры (PDS-сегмент). PDS-сегмент используется для определения палитры для цветового преобразования. Фиг.7B показывает синтаксическую структуру PDS-сегмента. PDS-сегмент состоит из элементов: `segment_type` (тип сегмента), указывающего, что сегмент представляет собой PDS-сегмент; `segment_length` (длина сегмента), указывающего длину данных этого PDS-сегмента; `palette_id` (идентификатор палитры), уникальным образом идентифицирующий палитру, содержащуюся в этом PDS-сегменте; `palette_version_number` (номер версии палитры), указывающий версию этого PDS-сегмента внутри Сверхкадра, и `palette_entry_id` (идентификатор точки входа палитры), указывающий идентификационный номер точки входа. `Palette_entry_id` указывает цветовой контраст по красному (`Cr_value`), цветовой контраст по синему (`Cb_value`), яркость (`Y_value`) и прозрачность (`T_value`).

Далее, ниже приводится разъяснение относительно Сегмента определения окна (WDS-сегмента).

WDS-сегмент используется для определения прямоугольной области на Графической матрице. Как было описано выше, управление памятью последовательно только при выполнении стирания и визуализации в пределах некоторой области на Графической матрице. Эта область на Графической матрице определяется WDS-сегментом и называется «окном». Фиг.8A иллюстрирует синтаксическую структуру WDS-сегмента. Как показано на чертеже, WDS-сегмент состоит из элементов: `segment_type` (тип сегмента), указывающего, что сегмент представляет собой WDS-сегмент; `segment_length` (длина сегмента), указывающего длину данных этого WDS-сегмента; `window_id` (идентификатор окна), уникальным образом идентифицирующий Окно на Графической матрице; `window_horizontal_position` (местоположение окна по горизонтали), указывающий адрес по горизонтали верхнего левого пикселя Окна на Графической матрице; `window_vertical_position` (местоположение окна по вертикали), указывающий адрес по вертикали верхнего левого пикселя Окна на Графической матрице; `window_width` (ширина окна), указывающий ширину Окна на Графической матрице; `window_height` (высота окна), указывающий высоту Окна на Графической матрице.

Ниже описаны диапазоны значений, которые могут принимать `window_horizontal_position`, `window_vertical_position`, `window_width` и `window_height`. Система координат для этих значений находится в пределах площади Графической матрицы, и их величина указывается в двух измерениях посредством `window_height` для высоты и `window_width` для ширины.

`Window_horizontal_position` (местоположение окна по горизонтали) определяет адрес по горизонтали верхнего левого пикселя Окна на Графической матрице и находится в пределах диапазона от 0 до $(\text{window_width})-1$.

Также `window_vertical_position` (местоположение окна по вертикали) указывает адрес по вертикали верхнего левого пикселя Окна на Графической матрице и находится в пределах диапазона от 0 до $(\text{window_height})-1$.

`Window_width` (ширина окна) указывает ширину Окна на Графической матрице. Указанная ширина находится в пределах диапазона от 1 до $(\text{video_width})-(\text{window_horizontal_position})$. Кроме того, `window_height` (высота окна) указывает высоту Окна на Графической матрице, и указанная высота находится в пределах диапазона от 1 до $(\text{video_height})-(\text{window_vertical_position})$.

Местоположение и размер Окна на Графической матрице для каждого Сверхкадра определяются значениями `window_horizontal_position`, `window_vertical_position`, `window_width` и `window_height`. Соответственно можно регулировать местоположение и размер Окна на этапе авторского процесса так, чтобы Окно в одном Сверхкадре было видно в положении, при котором оно не мешает изображению при просмотре фильма. Благодаря этому видимость субтитров становится выше. Поскольку WDS-сегмент определяется для каждого Сверхкадра, то можно регулировать местоположение Окна в соответствии с киноизображением даже в том случае, если киноизображение меняется в течение времени. В результате качество фильма поддерживается на таком же высоком уровне, как в случае, когда субтитры вмонтированы в основное «тело» фильма.

Далее рассматривается Сегмент конца отображаемого набора (END-сегмент). END-сегмент предоставляет указание на то, что передача DS-набора завершена. END-сегмент вставляется в поток данных сразу после последнего ODS-сегмента в одном DS-наборе. END-сегмент состоит из элементов: `segment_type` (тип сегмента), указывающего, что сегмент представляет собой END-сегмент, и `segment_length` (длина сегмента), указывающего длину данных этого END-сегмента. END-сегмент не включает в себя никакой другой элемент, который требует дополнительного разъяснения.

Далее, ниже приводится разъяснение в отношении Сегмента композиции представления (PCS-сегмента).

PCS-сегмент представляет собой функциональный сегмент, который используется для составления интерактивного отображения. Фиг.8В иллюстрирует синтаксическую структуру PCS-сегмента. Как показано на этом чертеже, PCS-сегмент состоит из элементов: `segment_type` (тип сегмента), `segment_length` (длина сегмента), `composition_number` (номер композиции), `composition_state` (состояние композиции), `palette_update_flag` (признак обновления палитры), `palette_id` (идентификатор палитры) и `composition_object` (объект композиции) 1-m.

`Composition_number` (номер композиции) идентифицирует Обновление графических данных в DS-наборе, принимая значения из диапазона от 0 до 15. Если между головной частью Сверхкадра и PCS-сегментом имеется Обновление графических данных, то `composition_number` получает приращение всякий раз, когда происходит Обновление графических данных.

`Composition_state` (состояние композиции) указывает на тип DS-набора, в котором содержится PCS-сегмент: «Нормальный случай», «Точка захвата» или «Начало сверхкадра».

`Palette_update_flag` (признак обновления палитры) указывает, что данный PCS-сегмент описывает Обновление отображения только по палитре. Обновление отображения только по палитре указывает, что обновляется только палитра, обновляясь по отношению к непосредственно предшествующей палитре.

`Palette_update_flag` устанавливается в "1" в том случае, если выполняется Обновление отображения только по палитре.

`Palette_id` (идентификатор палитры) идентифицирует палитру, подлежащую использованию при Обновлении отображения только по палитре.

`Composition_object` (объект композиции) 1-m указывает то, каким образом управлять каждым Окном в том DS-наборе, к которому принадлежит PCS-сегмент. При помощи пунктирной линии `wd1` на Фиг.8В подробно показана внутренняя синтаксическая структура `composition_object i` (i-го объекта композиции).

Composition_object i состоит из элементов: object_id (идентификатор объекта), window_id (идентификатор окна), object_cropped_flag (признак «вырезанного» объекта), object_horizontal_position (местоположение объекта по горизонтали), object_vertical_position (местоположение объекта по вертикали) и cropping_rectangle information 1-n (информация о «вырезающем» прямоугольнике 1-n).

Object_id (идентификатор объекта) идентифицирует ODS-сегмент в Окне, соответствующем composition_object i.

Window_id (идентификатор окна) идентифицирует Окно, куда данный Графический объект назначен в PCS-сегменте. В одно окно может быть назначено до двух Графических объектов.

Object_cropped_flag (признак «вырезанного» объекта) используется для переключения между отображением и неотображением «вырезанного» Графического объекта в Буфере объектов. При установке object_cropped_flag в "1" «вырезанный» Графический объект отображается в Буфере объектов, а если этот признак установлен в "0", то Графический объект не отображается.

Object_horizontal_position (местоположение объекта по горизонтали) указывает адрес по горизонтали верхнего левого пикселя Графического объекта на Графической матрице.

Object_vertical_position (местоположение объекта по вертикали) указывает адрес по вертикали верхнего левого пикселя Графического объекта на Графической матрице.

Cropping_rectangle information 1-n (информация о «вырезающем» прямоугольнике 1-n) представляет собой элементы, используемые в случае, когда object_cropped_flag установлен в "1". При помощи пунктирной линии wd2 подробно показана внутренняя синтаксическая структура для cropping_rectangle information 1-n (информации о «вырезающем» прямоугольнике 1-n). Как показано пунктирной линией wd2, cropping_rectangle information i (i-я информация о «вырезающем» прямоугольнике) состоит из четырех полей: object_cropping_horizontal_position (место «вырезания» объекта по горизонтали), object_cropping_vertical_position (место «вырезания» объекта по вертикали), object_cropping_width (ширина «вырезания» объекта) и object_cropping_height (высота «вырезания» объекта).

Object_cropping_horizontal_position (место «вырезания» объекта по горизонтали) указывает адрес по горизонтали верхнего левого угла «вырезающего» прямоугольника, используемого во время перевода Графического объекта в Графическую матрицу. «Вырезающий» прямоугольник представляет собой «вырезающую» рамку, которая используется для указания и «вырезания» части Графического объекта и соответствует «зоне» в терминологии стандарта ETSI EN 300 743 (стандарт Европейского института стандартов в области телекоммуникаций).

Object_cropping_vertical_position (место «вырезания» объекта по вертикали) указывает адрес по вертикали верхнего левого угла «вырезающего» прямоугольника, используемого во время перевода Графического объекта в Графическую матрицу.

Object_cropping_width (ширина «вырезания» объекта) указывает ширину «вырезающего» прямоугольника.

Object_cropping_height (высота «вырезания» объекта) указывает высоту «вырезающего» прямоугольника.

Ниже подробно рассматривается конкретный пример PCS-сегмента. В этом примере, как показано на Фиг.6, последовательно появляются субтитры: «На самом деле,...», «Я скрывала» и «свои чувства.», что осуществляется посредством

трехкратного выполнения записи на Графическую матрицу по мере продолжения воспроизведения кинофильма. Фиг.9 представляет собой пример описания для реализации такого отображения субтитров. Сверхкадр, показанный на чертеже, включает в себя отображаемые наборы: DS1 (Начало Сверхкадра), DS2

5 («Нормальный случай») и DS3 («Нормальный случай»). DS1 содержит WDS-сегмент для указания окна, в котором отображаются субтитры, ODS-сегмент для указания строки «На самом деле,... Я скрывала свои чувства.» и первый PCS-сегмент. DS2 содержит второй PCS-сегмент, а DS3 содержит третий PCS-сегмент.

10 Фиг.10-12 иллюстрируют примеры WDS-сегмента и PDS-сегмент в DS-наборах. Фиг.10 показывает пример PCS-сегмента в DS1.

На Фиг.10 `window_horizontal_position` (местоположение окна по горизонтали) и `window_vertical_position` (местоположение окна по вертикали), входящие в состав WDS-сегмента, указаны точкой LP1, местоположением верхнего левого

15 пикселя Окна на Графической матрице. `Window_width` (ширина окна) и `window_height` (высота окна) указывают соответственно ширину и высоту окна.

На Фиг.10 `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали) указывают точку ST1 начала отсчета «вырезающего» прямоугольника в системе координат, в которой началом координат является верхний левый пиксель Графического объекта. «Вырезающий» прямоугольник представляет собой область, имеющую ширину от ST до `object_cropping_width` (ширины «вырезания» объекта) и

20 высоту от ST до `object_cropping_height` (высоты «вырезания» объекта), (прямоугольник, показанный жирной рамкой). «Вырезанный» Графический объект располагают внутри прямоугольника, показанного пунктирной рамкой `sr1`, с точкой начала отсчета в системе координат с началом координат в точке, задаваемой значениями `object_horizontal_position` и `object_vertical_position` (верхний левый пиксель

25 Графического объекта) в Графической матрице. Благодаря этому субтитр «На самом деле,...» записывается в данное Окно на Графической матрице, и затем объединяется с киноизображением и отображается на экране.

Фиг.11 показывает пример PCS-сегмента в отображаемом наборе DS2. WDS-сегмент, входящий в DS2, не рассматривается, поскольку WDS-сегмент, входящий

35 в DS2, является тем же самым сегментом, что и WDS-сегмент, входящий в DS1. А вот описание информации о «вырезании» в DS2 отлично от описания информации о «вырезании», показанной на Фиг.10.

На Фиг 11 `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали) в информации о «вырезании» указывает верхний левый пиксель субтитра «Я скрывала» из строки «На самом деле,... Я скрывала свои чувства», содержащейся в Буфере объектов. `Object_cropping_width` (ширина «вырезания» объекта) и `object_cropping_height` (высота «вырезания» объекта) указывают ширину и высоту

45 прямоугольника, содержащего субтитр «Я скрывала». Благодаря этому субтитр «Я скрывала» записывается в данное Окно на Графической матрице и затем объединяется с кинофильмом и отображается на экране.

Фиг.12 показывает пример PCS-сегмента в отображаемом наборе DS3. WDS-сегмент, входящий в DS3, не рассматривается, поскольку WDS-сегмент, входящий

50 в DS3, является тем же самым сегментом, что и WDS-сегмент, входящий в DS1. А вот описание информации о «вырезании» в DS3 отлично от описания информации о «вырезании», показанной на Фиг.10.

На Фиг.12 `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали) в информации о «вырезании» указывает верхний левый пиксель субтитра «свои чувства» из строки «На самом деле,... Я скрывала свои чувства»,
 5 содержащейся в Буфере объектов. `Object_cropping_width` (ширина «вырезания» объекта) и `object_cropping_height` (высота «вырезания» объекта) указывают ширину и высоту прямоугольника, содержащего субтитр «свои чувства». Благодаря этому субтитр «свои чувства» записывается в данное Окно на Графической матрице, и
 10 затем объединяется с кинофильмом и отображается на экране.

Посредством описания отображаемых наборов DS1, DS2 и DS3 таким образом, как это описано выше, можно достигнуть эффекта отображения этих субтитров на экране. Также можно достигнуть и других видов эффектов, ниже описываются протоколы описаний для реализации других эффектов.

15 Во-первых, рассмотрим протокол описания для эффекта появления/исчезновения изображения объекта. Фиг.13 показывает проиллюстрированный вдоль временной шкалы пример описания DS-набора при осуществлении эффекта появления/исчезновения изображения объекта.

20 На этом чертеже значения x и y в Window (x,y,u,v) соответственно указывают значения `window_vertical_position` (местоположения окна по вертикали) и `window_horizontal_position` (местоположения окна по горизонтали), а значения u и v соответственно указывают значения `window_width` (ширины окна) и `window_height` (высоты окна). Также на этом чертеже значения a и b в `Cropping_Rectangle` (a,b,c,d)
 25 соответственно указывают значения `object_cropping_vertical_position` (места «вырезания» объекта по вертикали) и `object_cropping_horizontal_position` (места «вырезания» объекта по горизонтали), а значения c и d указывают значения `object_cropping_width` (ширины «вырезания» объекта) и `object_cropping_height` (высоты «вырезания» объекта) соответственно. Отображаемые наборы DS11, DS12 и DS13 находятся в точках $t11$, $t12$ и $t13$ на временной шкале воспроизведения,
 30 показанной на чертеже.

DS11 в точке $t11$ включает в себя: PCS#0, в котором `composition_state` (состояние композиции) представляет собой «Начало свержкадра» («Epoch Start»),
 35 и `object_cropped_flag` (признак «вырезанного» объекта) равен "0" (`no_cropping_rectangle_visible` (не видим никакой «вырезающий» прямоугольник)); WDS#0 с оператором, задающим Окно с размерами ширина 700 × высота 500 в точке (100,100) в Графической матрице; PDS#0; ODS#0 с указанием субтитра «Участники постановки:»; и END (Конец).
 40

DS12 в точке $t12$ включает в себя PCS#1, в котором `composition_state` (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 600×400 от точки (0,0) в Буфере объектов («вырезающий» прямоугольник: `cropping_rectangle#0` (0,0,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)).
 45

DS13 в точке $t13$ включает в себя PCS#2, в котором `composition_state` (состояние композиции) представляет собой «Нормальный случай» («Normal Case»), и в котором `object_cropped_flag` (признак «вырезанного» объекта) устанавливается в "0", так чтобы стереть вырезанный Графический объект (`no_cropping_rectangle_visible`).
 50

С вышеописанными отображаемыми наборами субтитр «Участники постановки:» не отображаются в момент $t11$, появляются в момент $t12$ и затем снова становятся

неотображаемыми в момент t_{13} , и реализуется эффект появления/исчезновения изображения объекта.

Во-вторых, рассмотрим протокол описания для эффекта выхода из затемнения/ухода в затемнение. Фиг.14 показывает проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении эффекта выхода из затемнения/ухода в затемнение. Отображаемые наборы DS21, DS22, DS23 и DS24 находятся в точках t_{21} , t_{22} , t_{23} и t_{24} на временной шкале воспроизведения, показанной на чертеже.

DS21 в точке t_{21} включает в себя: PCS#0, в котором composition_state (состояние композиции) представляет собой «Начало свехкадра» («Epoch Start») и который задает операцию «вырезания» Графического объекта размером 600×400 от точки (0,0) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (0,0,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)); WDS#0 с оператором, задающим Окно с размерами ширина 700 × высота 500 в точке (100,100) в Графической матрице; PDS#0; ODS#0 с указанием субтитра «Конец фильма»; и END (Конец).

DS22 в точке t_{22} включает в себя: PCS#1, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case»), и PDS#1. PDS#1 указывает тот же самый уровень Cг и Cb, что и PDS#0, но яркость, указанная в PDS#1 выше, чем яркость в PDS#0.

DS23 в точке t_{23} включает в себя: PCS#2, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case»); PDS#2 и END. PDS#2 указывает тот же самый уровень Cг и Cb, что и PDS#1, но яркость, указанная в PDS#2 ниже, чем яркость в PDS#1.

DS24 в точке t_{24} включает в себя: PCS#2, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и object_cropped_flag (признак «вырезанного» объекта) равен 0 (no_cropping_rectangle_visible), и END.

Каждый DS-набор указывает PDS-сегмент, отличный от предыдущего DS-набора, и соответственно яркость Графического объекта, визуализируемого посредством более чем одного PCS-сегмента в одном свехкадре, постепенно становится высокой или низкой. Благодаря этому можно реализовать эффект выхода из затемнения/ухода в затемнение.

Далее рассмотрим протокол описания для прокрутки. Фиг.15 показывает проиллюстрированный вдоль временной шкалы пример описания Отображаемого набора при осуществлении прокрутки. Отображаемые наборы DS31, DS32, DS33 и DS34 находятся в точках t_{31} , t_{32} , t_{33} и t_{34} на временной шкале воспроизведения, показанной на чертеже.

DS31 в точке t_{31} включает в себя: PCS#0, в котором composition_state (состояние композиции) представляет собой «Начало свехкадра» («Epoch Start»), и object_cropped_flag (признак «вырезанного» объекта) равен "0" (no_cropping_rectangle_visible); WDS#0 с оператором, задающим Окно с размерами ширина 700 × высота 500 в точке (100,100) в Графической матрице; PDS#0; ODS#0 с указанием субтитра «Участники постановки: Компания»; и END (Конец).

DS32 в точке t_{32} включает в себя PCS#1, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 600×400 от

точки (0,0) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (0,0,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)). Область размером 600×400 от точки (0,0) в Буфере объектов включает в себя часть

5 «Участники постановки:» из субтитра «Участники постановки: Компания», показанного в две строки, и, таким образом, часть «Участники постановки:» появляется на Графической матрице.

DS33 в точке t33 включает в себя PCS#2, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который

10 задает операцию «вырезания» Графического объекта размером 600×400 от точки (0,100) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (0,100,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)). Область

15 размером 600×400 от точки (0,100) в Буфере объектов включает в себя часть «Участники постановки:» и часть «Компания» из субтитра «Участники постановки: Компания», показанного в две строки, и, таким образом, часть «Участники постановки:» и часть «Компания» появляются расположенными в две строки на

20 Графической матрице.

DS34 в точке t34 включает в себя PCS#3, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 600×400 от

25 точки (0,200) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (0,200,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)). Область размером 600×400 от точки (0,200) в Буфере объектов включает в себя часть «Компания» из субтитра «Участники постановки: Компания», показанного в две

30 строки, и, таким образом, часть «Компания» появляется на Графической матрице. Благодаря вышеприведенному описанию PCS-сегмента можно осуществлять направленную вниз прокрутку субтитров, расположенных в две строки.

Наконец, рассмотрим протокол описания для эффекта нанесения/стирания изображения объекта. Фиг.16 показывает проиллюстрированный вдоль временной

35 шкалы пример описания Отображаемого набора при осуществлении эффекта нанесения/стирания изображения объекта. Отображаемые наборы DS21, DS22, DS23 и DS24 находятся в точках t21, t22, t23 и t24 на временной шкале воспроизведения, показанной на чертеже.

DS51 в точке t51 включает в себя: PCS#0, в котором composition_state (состояние композиции) представляет собой «Начало свехкадра» («Epoch Start»),

40 и object_cropped_flag (признак «вырезанного» объекта) равен "0" (no_cropping_rectangle_visible); WDS#0 с оператором, задающим Окно с размерами ширина 700 × высота 500 в точке (100,100) в Графической матрице; PDS#0; ODS#0 с

45 указанием субтитра «Конец фильма»; и END (Конец).

DS52 в точке t52 включает в себя PCS#1, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 600×400 от

50 точки (0,0) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (0,0,600,400)) и располагает вырезанный Графический объект в точке с координатами (0,0) в Графической матрице (в окне Window#0(0,0)). Область размером 600×400 от точки (0,0) в Буфере объектов включает в себя субтитр «Конец

фильма», и, таким образом, субтитр «Конец фильма» появляется на Графической матрице.

DS53 в точке t53 включает в себя PCS#2, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 400×400 от точки (200,0) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (200,0,400,400)) и располагает вырезанный Графический объект в точке с координатами (200,0) в Графической матрице (в окне Window#0(200,0)). Благодаря этому область, заданная координатами (200,0) и (400,400) в Окне, становится отображаемой областью, а область, заданная координатами (0,0) и (199,400), становится неотображаемой областью.

DS54 в точке t54 включает в себя PCS#3, в котором composition_state (состояние композиции) представляет собой «Нормальный случай» («Normal Case») и который задает операцию «вырезания» Графического объекта размером 200×400 от точки (400,0) в Буфере объектов («вырезающий» прямоугольник: cropping_rectangle#0 (400,0,200,400)), и располагает вырезанный Графический объект в точке с координатами (400,0) в Графической матрице (в окне Window#0(400,0)). Благодаря этому область, заданная координатами (0,0) и (399,400), становится неотображаемой областью.

Благодаря этому, по мере того, как неотображаемая область становится больше, отображаемая область становится меньше, и, таким образом, реализуется эффект нанесения/стирания изображения объекта.

Как было сказано выше, используя соответствующие сценарии, можно реализовывать разнообразные эффекты, такие как появление/исчезновение изображения объекта, выход из затемнения/уход в затемнение, нанесение/стирание изображения объекта и прокрутка, и, следовательно, различные варианты монтажа при визуализации субтитров.

Ограничения при реализации вышеописанных эффектов заключаются в следующем. Для реализации эффекта прокрутки становятся необходимы операции по очистке и повторному вычерчиванию содержимого Окна. Если обратиться к примеру, показанному на Фиг.15, то необходимо выполнить «очистку окна» с тем, чтобы в момент времени t32 стереть Графический объект «Участники постановки:» с Графической матрицы, и затем выполнить «повторное вычерчивание окна» с тем, чтобы в интервал времени между t32 и t33 записать в Графическую матрицу нижнюю часть надписи «Участники постановки:» и верхнюю часть надписи «Компания». При условии, что этот интервал времени одинаков с интервалом времени между видеокадрами, важным аспектом становится необходимая для прокрутки скорость передачи данных между Буфером объектов и Графической матрицей.

Далее рассмотрим ограничение на предмет того, сколь велико может быть Окно. Пусть Rc представляет собой скорость передачи данных между Буфером объектов и Графической матрицей. Наихудший сценарий при этом заключается в том, чтобы выполнять обе из операций «очистки Окна» и «повторного вычерчивания Окна» со скоростью Rc. В этом случае каждую из операций «очистки Окна» и «повторного вычерчивания Окна» требуется выполнять со скоростью, равной половине Rc (Rc/2).

Для того чтобы синхронизировать операции «очистки Окна» и «повторного вычерчивания Окна» с видеокадром, должно выполняться приводимое ниже уравнение:

Размер Окна × Частота кадров $\square$ Rc/2

Если Частота кадров равна 29,97, то R_c может быть выражено приводимым ниже уравнением.

$$R_c = \text{Размер Окон} \times 2 \times 29,97$$

При визуализации субтитров Размер Окон занимает, по меньшей мере, от 25% до 33% Графической матрицы. Общее количество пикселей в Графической матрице составляет 1920×1080. Принимая, что выраженная в битах длина индекса, приходящаяся на пиксель, составляет 8 битов, получаем, что общая емкость Графической матрицы составляет 2 мегабайта ($\square 1920 \times 1080 \times 8$).

Принимая, что Размер окна равен 1/4 от общей емкости Графической матрицы, получаем, что Размер окна составит 500 килобайтов (= 2 мегабайта /4). Подставляя это значение в приведенное выше уравнение, вычисляем значение R_c , равное 256 мегабитов в секунду (= 500 килобайтов $\times 2 \times 29,97$). Если частота для «Очистки Окон» и «Повторного вычерчивания Окон» может быть равной половине или четверти частоты кадров, то можно удвоить или увеличить в четыре раза размер Окон даже при том же самом значении R_c .

Удерживая Размер Окон в диапазоне от 25% до 33% Графической матрицы и осуществляя отображение субтитров со скоростью передачи данных, равной 256 мегабитов в секунду, можно поддерживать синхронность отображения между графическими данными и кинофильмом независимо от того, какой отображаемый эффект должен быть реализован.

Далее рассматриваются местоположение, размер и площадь Окон. Как было сказано выше, местоположение и площадь Окон не меняется на протяжении одного Сверхкадра. Местоположение и размер Окон устанавливаются таким образом, чтобы быть одинаковыми на протяжении одного Сверхкадра, по той причине, что в случае, если эти местоположение и размер изменяются, то необходимо изменить целевой адрес записи на Графической матрице, а изменение этого адреса вызывает накладные издержки, которые снижают скорость передачи данных из Буфера объектов в Графическую матрицу.

Количество Графических объектов, приходящихся на Окно, имеет ограничение. Ограничение на это количество вводится для того, чтобы снизить накладные издержки при передаче декодированного Графического объекта. Накладные издержки в данном случае возникают при установке адреса края Графического объекта, и чем больше количество краев, тем больше возникает накладных издержек.

Фиг.17 показывает сравнение двух примеров: примера, в котором Окно имеет четыре Графических объекта, и другого примера, в котором Окно имеет два графических объекта. Количество краев в примере с четырьмя Графическими объектами в два раза больше количества краев в примере с двумя Графическими объектами.

Без ограничения на количество Графических объектов, становится неизвестно, сколько накладных издержек могло бы возникнуть при передаче графических данных, и таким образом нагрузка по передаче резко увеличивается и уменьшается. С другой стороны, в случае, когда максимальное количество Графических объектов в Окне равно двум, скорость передачи данных может быть установлена с учетом до четырех накладных издержек. Соответственно проще установить значение минимальной скорости передачи данных.

Далее приводится объяснение того, как DS-набор, имеющий PCS-сегмент и ODS-сегмент, привязывается к временной шкале потока AVClip. Сверхкадр представляет собой период времени, в который управление памятью последовательно вдоль

временной шкалы воспроизведения. Поскольку Сверхкадр состоит из более чем одного DS-набора, то важным вопросом является то, как осуществлять привязку DS-набора к временной шкале воспроизведения потока AVClip. Временная шкала воспроизведения потока AVClip представляет собой временную шкалу, предназначенную для указания моменты времени для декодирования и воспроизведения каждой порции данных кинофильма, которые составляют поток видеоданных, мультиплексированный в поток AVClip. Моменты времени декодирования и воспроизведения задаются на временной шкале воспроизведения с точностью 90 кГц. DTS-метка (временная метка декодирования) и PTS-метка (временная метка представления), прикрепленные к PCS-сегменту и ODS-сегменту в DS-наборе, указывают на временной шкале воспроизведения моменты времени для синхронного управления. Привязка Отображаемого набора к временной шкале воспроизведения означает выполнение синхронного управления с использованием DTS-метки и PTS-метки, прикрепленных к PCS-сегменту и ODS-сегменту.

В первую очередь, ниже будет рассмотрено то, каким образом синхронное управление выполняется с использованием DTS-метки и PTS-метки, прикрепленных к ODS-сегменту.

DTS-метка с точностью до 90 кГц указывает момент времени, когда начинается декодирование ODS-сегмента, а PTS-метка указывает момент времени, когда декодирование заканчивается.

Декодирование ODS-сегмента не заканчивается сразу, а имеет некоторую протяженность во времени. В ответ на запрос о четком указании начальной точки и конечной точки, определяющих продолжительность декодирования, DTS-метка и PTS-метка из PCS-сегмента указывают соответственно моменты времени, в которые начинается и заканчивается декодирование.

Значение PTS-метки указывает конечный срок, и, следовательно, необходимо, чтобы декодирование ODS-сегмента было завершено к моменту времени, указанному PTS-меткой и разуплотненные данные Графического объекта были записаны в Буфер объектов, имеющийся на устройстве воспроизведения.

Время начала декодирования любого сегмента ODS_j в наборе DS_n указывается посредством обозначения DTS(DS_n[ODS_j]) с точностью 90 кГц. Прибавление к DTS(DS_n[ODS_j]) максимальной величины продолжительности декодирования дает момент времени, в который заканчивается декодирование ODS_j.

Если размер ODS_j обозначить как «SIZE(DS_n[ODS_j])», а скорость декодирования ODS-сегмента как Rd, то максимальное время, необходимое для декодирования, указанное в секундах, выражается как «SIZE(DS_n[ODS_j])/Rd». Символ «//» указывает оператор деления с округлением после десятичной разряда.

Преобразуя этот максимальный период времени в число, выраженное с точностью 90 кГц, и прибавляя к DTS-метке сегмента ODS_j, вычисляем указываемый PTS-меткой момент времени, когда заканчивается декодирование, (выраженный с точностью 90 кГц).

PTS-метка сегмента ODS_j в наборе DS_n выражается следующим уравнением:

$$\text{PTS}(\text{DS}_n[\text{ODS}_j]) = \text{DTS}(\text{DS}_n[\text{ODS}_j]) + 90000 \times (\text{SIZE}(\text{DS}_n[\text{ODS}_j]) / \text{Rd})$$

Кроме того, необходимо, чтобы соотношение между двумя следующими друг за другом ODS-сегментами: ODS_j и ODS_j+1, удовлетворяло следующему выражению:

$$\text{PTS}(\text{DS}_n[\text{ODS}_j]) \leq \text{DTS}(\text{DS}_n[\text{ODS}_{j+1}])$$

Далее рассматривается установка значений DTS-метки и PTS-метки PCS-сегмента.

Необходимо, чтобы PCS-сегмент был загружен в Буфер объектов, имеющийся на устройстве воспроизведения, перед моментом времени начала декодирования DTS(DSn[ODS1]) первого ODS-сегмента (ODS1) в наборе DSn и перед моментом времени PTS(DSn[PDS1]), когда первый PDS-сегмент (PDS1) становится действующим. Соответственно необходимо, чтобы DTS-метка была установлена таким образом, чтобы удовлетворять следующим выражениям:

$$DTS(DSn[PCS]) \leq DTS(DSn[ODS1])$$

$$DTS(DSn[PCS]) \leq PTS(DSn[PDS1])$$

Кроме того, PTS-метка PCS-сегмента в наборе DSn выражается следующим выражением:

$$PTS(DSn[PCS]) \geq DTS(DSn[PCS]) + \text{decodeduration}(DSn)$$

«Decodeduration(DSn)» (продолжительность декодирования набора DSn) указывает продолжительность времени для декодирования всех Графических объектов, используемых для обновления PCS-сегмента. Продолжительность декодирования не является фиксированной величиной, но не изменяется в зависимости от состояния устройства воспроизведения и устройства или программного обеспечения, установленных на устройство воспроизведения. В случае, когда Объект, используемый для составления экранного изображения DSn.PCSn, обозначается как DSn.PCSn.OBJ[j], decodeduration(DSn) определяется временем (i), необходимым для очистки Окна, значениями продолжительности (ii) декодирования по декодированию DSn.PCSn.OBJ и временем (iii), необходимым для записи DSn.PCSn.OBJ. В случае, когда Rd и Rc заданы, значение decode_duration(DSn) всегда одинаково. Следовательно, PTS-метка вычисляется посредством вычисления значений этих продолжительностей на этапе авторского процесса.

Вычисление decode_duration (продолжительности декодирования) выполняется на основе программы, показанной на Фиг.19. Фигуры 19, 20A и 20B представляют собой блок-схемы, показывающие в схематическом виде алгоритмы этой программы. Ниже со ссылкой на эти чертежи приводится разъяснение относительно вычисления decode_duration. На блок-схеме алгоритма, показанной на Фиг.19, сначала осуществляется вызов функции PLANEINITIALIZE («инициализация матрицы») (этап S1 на Фиг.19). Функция PLANEINITIALIZE используется для вызова функции для вычисления периода времени, необходимого для инициализации Графической матрицы для визуализации DS-набора. На этапе S1 на Фиг.19 эта функция осуществляет вызов этой функции с аргументами DSn, DSn.PCS.OBJ[0] и decode_duration.

Далее со ссылкой на Фиг.20A рассматривается функция PLANEINITIALIZE. На этом чертеже initialize_duration (продолжительность инициализации) представляет собой переменную, указывающую возвращаемое значение функции PLANEINITIALIZE.

Этап S2 на Фиг.20 представляет собой оператор условного перехода, предназначенный для переключений операций в зависимости от того, указывает или нет page_state (состояние страницы) в PCS-сегменте в наборе DSn «Начало Сверхкадра» («Epoch_Start»). Если page_state указывает «Начало Сверхкадра» (DSn.PCS.page_state==epoch_start, этап S2=Yes (Да) на Фиг.18), то переменной initialize_duration присваивается значение периода времени, необходимого для очистки Графической матрицы (этап S3).

В случае, когда скорость Rc передачи данных между Буфером объектов и

Графической матрицей составляет 256000000, как было сказано выше, а общий размер Графической матрицы устанавливается равным $\text{video_width} \times \text{video_height}$ (ширина видео * высота видео), период времени, необходимый для очистки, составляет $\langle \text{video_width} \times \text{video_height} / 256000000 \rangle$. Будучи умноженным на 90000 Гц с тем, чтобы быть выраженным с точностью указания времени PTS-метки, период времени, необходимый для очистки Графической матрицы, составляет $\langle 90000 \times \text{video_width} \times \text{video_height} / 256000000 \rangle$. Этот период времени прибавляется к `initialize_duration`.

Если `page_state` не указывает Начало Сверхкадра (этап S2=No (Нет)), то к переменной `initialize_duration` прибавляется для всех Окон значение периода времени, необходимого для очистки окна `Window[i]`, определенного WDS-сегментом (этап S4). В случае, когда скорость R_c передачи данных между Буфером объектов и Графической матрицей составляет 256000000, как было сказано выше, а суммарный размер `Window[i]`, принадлежащих WDS-сегменту, составляет $\sum \text{SIZE}(\text{WDS.WIN}[i])$, период времени, необходимый для очистки, составляет $\langle \sum \text{SIZE}(\text{WDS.WIN}[i]) / 256000000 \rangle$. Будучи умноженным на 90000 Гц с тем, чтобы быть выраженным с точностью указания времени PTS-метки, период времени, необходимый для очистки Окон, принадлежащих WDS-сегменту, составляет $\langle 90000 \times \sum \text{SIZE}(\text{WDS.WIN}[i]) / 256000000 \rangle$. Этот период времени прибавляется к `initialize_duration`, а `initialize_duration` возвращается в качестве результата. Описанное выше представляет собой функцию `PLANEINITIALIZE`.

Этап 5 на Фиг.19 предназначен для переключения операций в зависимости от того, равняется ли количество Графических объектов в DSn -наборе 2 или 1 ($\text{if}(\text{DSn.PCS.num_of_object}==2, \text{if}(\text{DSn.PCS.num_of_object}==1$ на Фиг.18), и если это количество равно 1 (этап S5), то к `decode_duration` прибавляется время ожидания для декодирования Графического объекта (этап S6). Вычисление времени ожидания выполняется посредством вызова функции `WAIT` (`decode_duration += WAIT(DSn, DS.PCS.OBJ[0], decode_duration)` на Фиг.18). Вызов функции осуществляется с использованием аргументов, установленных в DSn , $\text{DSn.PCS.OBJ}[0]$, `decode_duration`, а возвращаемое значение представляет собой `wait_duration`.

Фиг.20В представляет собой блок-схему алгоритма, показывающего действие функции `WAIT`.

На этой блок-схеме алгоритма значение `decode_duration` из оператора вызова присваивается `current_duration` (текущей продолжительности). `Object_definition_ready_time` (время готовности определения объекта) представляет собой переменную, которой присваивается значение PTS-метки Графического объекта из DS -набора.

`Current_time` представляет собой переменную, которой присваивается значение суммы `current_duration` и DTS-метки PCS-сегмента в DSn . В случае, когда `object_definition_ready_time` больше, чем `current_time` (ответ «Да» на этапе S7, $\text{if}(\text{current_time} < \text{object_definition_ready_time})$), `wait_duration` (время ожидания), как возвращаемое значение, устанавливается равным разности между `object_definition_ready_time` и `current_time` (этап S8, `wait_duration += object_definition_ready_time - current_time`). `Decode_duration` (продолжительность декодирования) устанавливается равной периоду времени, составляющему сумму возвращаемого значения функции `WAIT` и периода времени, необходимого для повторного вычерчивания этого Окона, $(90000 \times (\text{SIZE}(\text{DSn.WDS.WIN}[0])) / 256000000)$.

Приведенное выше разъяснение рассматривает случай, в котором количество

Графических объектов равно одному. На этапе S5 на Фиг.5 определяется, равно ли количество Графических объектов двум. Если количество Графических объектов в DS_n больше чем два (if(DS_n.PCS.num_of_object==2) на Фиг.18), то осуществляется вызов функции WAIT с использованием OBJ[0] в PCS-сегменте в качестве аргумента, и к decode_duration прибавляется ее возвращаемое значение (этап S10).

На последующем этапе S11 производится определение того, является ли то Окно, к которому принадлежит OBJ[0] набора DS_n, тем же самым Окном, что и Окно, к которому принадлежит Графический объект OBJ[1] (if(DS_n.OBJ[0].window_id==DS_n.PCS.OBJ[1].window_id). Если Окно является тем же самым, то осуществляется вызов функции WAIT с использованием OBJ[1] в качестве аргумента, и к decode_duration прибавляется ее возвращаемое значение (этап S12), и к decode_duration прибавляется время, необходимое для повторного вычерчивания того Окна, к которому принадлежит объект OBJ[0] ($90000 \times (\text{SIZE}(\text{DS}_n.\text{WDS.OBJ}[0].\text{window_id}) // 256000000)$) (этап S13).

Если определено, что Окна являются разными (этап S11, ответ «разные»), то к decode_duration прибавляется время, необходимое для повторного вычерчивания того Окна, к которому принадлежит объект OBJ[0] ($90000 \times (\text{SIZE}(\text{DS}_n.\text{WDS.OBJ}[0].\text{window_id}) // 256000000)$) (этап S15), осуществляется вызов функции WAIT с использованием OBJ[1] в качестве аргумента, и к decode_duration прибавляется возвращаемое значение wait_duration (этап S16), и к decode_duration прибавляется время, необходимое для повторного вычерчивания того Окна, к которому принадлежит объект OBJ[1] ($90000 \times (\text{SIZE}(\text{DS}_n.\text{WDS.OBJ}[1].\text{window_id}) // 256000000)$) (этап S17).

Decode_duration вычисляется в соответствии с вышеописанным алгоритмом. Ниже рассматривается конкретный способ, которым устанавливается PTS-метка OCS.

Фиг.21А иллюстрирует случай, в котором один ODS-сегмент включен в состав одного Окна. Фиг.21В и 21С представляют собой временные диаграммы, показывающие значения в порядке времени, которые упоминаются на Фиг.18. Нижняя строка «Декодирование ODS-сегмента» и средняя строка «Доступ к Графической матрице» в каждой диаграмме указывает две операции, которые при воспроизведении выполняются одновременно. Вышеназванный алгоритм описан в предположении, что эти две операции выполняются параллельно.

Доступ к Графической матрице включает в себя период (1) очистки и период (3) записи. Период (1) очистки указывает либо период времени, необходимый для очистки всей Графической матрицы ($90000 \times (\text{размер Графической матрицы} // 256000000)$), либо период времени, необходимый для очистки всех Окон на Графической матрице ($\sum (90000 \times (\text{размер Окна } [i] // 256000000))$).

Период (3) записи указывает период времени, необходимый для визуализации всего Окна ($90000 \times (\text{размер Окна } [i] // 256000000)$).

Кроме того, период (2) декодирования указывает период времени между DTS-меткой и PTS-меткой ODS-сегмента.

Продолжительности периода (1) очистки, периода (2) декодирования и периода (3) записи могут варьироваться в зависимости от области, подлежащей очистке, размера ODS-сегмента, подлежащего декодированию, и размера Графического объекта, подлежащего записи в Графическую матрицу. Для удобства начальная точка периода (2) декодирования на чертеже является той же самой точкой, что и начальная точка периода (1) очистки.

Фиг.21В иллюстрирует случай, при котором период (2) декодирования является

длинным, и `decode_duration` равняется сумме периода (2) декодирования и периода (3) записи.

Фиг.21С иллюстрирует случай, при котором период (1) очистки является длинным, и `decode_duration` равняется сумме периода (1) очистки и периода (3) записи.

Фигуры с 22А по 22С иллюстрируют случай, при котором в состав одного Окна включены два ODS-сегмента. Период (2) декодирования на обеих Фигурах 22В и 22С указывает суммарный период времени, необходимый для декодирования двух Графических объектов. Аналогичным образом, период (3) записи указывает суммарный период времени, необходимый для декодирования двух Графических объектов в Графическую матрицу.

Даже, несмотря на то, что количество ODS-сегментов равно двум, можно вычислять `decode_duration` таким же образом, как и в случае, показанном на Фиг.21. Когда период (3) декодирования, служащий для декодирования двух ODS-сегментов, является длинным, `decode_duration` равняется сумме периода (2) декодирования и периода (3) записи, что и показано на Фиг.22В.

Когда период (1) очистки является длинным, `decode_duration` равняется сумме периода (1) очистки и периода (3) записи.

Фиг.23А описывает случай, при котором каждое из двух Окон включает в себя один ODS-сегмент. Как и в предыдущих случаях, когда период (1) очистки длиннее, чем период (3) декодирования, служащий для декодирования двух ODS-сегментов, `decode_duration` равняется сумме периода (1) очистки и периода (3) записи. Однако, когда период (1) очистки короче, чем период (3) декодирования, можно осуществлять запись в первое Окно еще до окончания периода (2) декодирования. Соответственно `decode_duration` не равняется ни сумме периода (1) очистки и периода (3) записи, ни сумме периода (2) декодирования и периода (3) записи.

В ситуации, когда период времени, необходимый для декодирования первого ODS-сегмента, является периодом (31) записи, и период времени, необходимый для декодирования второго ODS-сегмента, является периодом (32) записи. Фиг.23В иллюстрирует случай, при котором период (2) декодирования длиннее, чем сумма периода (1) очистки и периода (31) записи. В этом случае `decode_duration` равняется сумме периода (2) декодирования и периода (32) записи.

Фиг.23С иллюстрирует случай, при котором сумма периода (1) очистки и периода (31) записи длиннее, чем период (2) декодирования. В этом случае `decode_duration` равняется сумме периода (1) очистки, периода (31) записи и периода (32) записи.

Размер Графической матрицы известен заранее для модели устройства воспроизведения. Также размер Окна и размер и количество ODS-сегментов также известны на этапе авторского процесса. Соответственно можно определить, какой комбинации периодов времени равняется `decode_duration`: периода (1) очистки и периода (3) записи; периода (2) декодирования и периода (3) записи; периода (2) декодирования и периода (32) записи; или периода (1) очистки, периода (3) записи и периода (32) записи.

Устанавливая PTS-метку ODS-сегмента на основе вычисления `decode_duration`, описанного выше, можно осуществлять синхронное отображение графических данных с данными кинофильма с высокой точностью. Такое синхронное отображение, осуществляемое с высокой точностью, становится возможным благодаря определению Окна и ограничения области повторного вычерчивания этим Окном. Таким образом, введение концепции Окна в среду авторского процесса

имеет большое значение.

Далее приведено разъяснение, касающееся настроек DTS-метки и PTS-метки WDS-сегмента в DS_n. DTS-метка WDS-сегмента может быть установлена таким образом, чтобы удовлетворять приведенному ниже выражению.

$$DTS(DS_n[WDS]) \geq DTS(DS_n[PCS])$$

С другой стороны OTS WDS-сегмента в наборе DS_n указывает конечный срок для начала записи в Графическую матрицу. Поскольку достаточно осуществить запись в Окно на Графической матрице, то время для начала записи в Графическую матрицу определяется путем вычитания продолжительности времени, указанной PTS-меткой PCS-сегмента из периода времени, необходимого для записи WDS-сегмента. В случае, когда общий размер WDS-сегмента равен $\sum SIZE(WDS.WIN[i])$, время, необходимое для очистки и повторного вычерчивания, составляет « $\sum SIZE(WDS.WIN[i]) // 256000000$ ». Выраженное с точностью указания времени, равной 90 кГц, это время составляет « $90000 \times \sum SIZE(WDS.WIN[i]) // 256000000$ ».

Соответственно можно вычислить PTS-метку WDS-сегмента из следующего уравнения.

$$PTS(DS_n[WDS]) =$$

$$PTS(DS_n[PCS]) - 90000 \times \sum SIZE(WDS.WIN[i]) // 256000000$$

PTS-метка, указанная в WDS-сегменте, представляет собой конечный срок, и можно начинать запись в Графическую матрицу раньше, чем указывает PTS-метка. Иначе говоря, как показано на Фиг.23, как только произведено декодирование ODS-сегмента, подлежащего визуализации в одном из Окон, в этот момент можно начинать запись Графического объекта, полученного посредством этого декодирования.

Как было описано выше, можно осуществить привязку Окна к любому моменту времени на временной шкале воспроизведения потока AVClip, используя для этого DTS-метку и PTS-метку, добавленные в WDS-сегмент.

Ниже со ссылкой на конкретный пример, проиллюстрированный на Фиг.24-25, приводятся разъяснения, касающиеся примера настроек DTS-меток и PTS-меток на основе данных настроек. Этот пример касается случая, при котором субтитры отображаются посредством четырехкратного осуществления записи в Графическую матрицу, и для отображения каждого из двух субтитров «что такое blu-ray.» и «blu-ray - повсюду.» производится обновление. Фиг.24 иллюстрирует хронологическое изменение при обновлении в этом примере. До момента времени t1 отображается «что», а после t1 и до t2 отображается «что такое», и затем в момент t3 отображается «что такое blu-ray.». После того, как появилось все предложение первого субтитра, в момент времени t4 отображается второй субтитр «blu-ray - повсюду.»

Фиг.25А иллюстрирует четыре отображаемых набора, которые описаны таким образом, чтобы выполнять рассмотренное выше обновление. Набор DS1 включает в себя сегмент PCS1.2 для управления обновлением в момент времени t1, сегмент PDS1 для цветовой окраски, сегмент ODS1, соответствующий субтитру «что такое blu-ray.», и сегмент END как код окончания DS1.

Набор DS2 включает в себя сегмент PCS1.2 для управления обновлением в момент времени t2 и сегмент END. Набор DS3 включает в себя сегмент PCS1.3 для управления обновлением в момент времени t3 и сегмент END. Набор DS 4 включает в себя сегмент PCS2 для управления обновлением в момент времени t2, сегмент PDS2 для преобразования цвета, сегмент ODS2, соответствующий субтитру «blu-ray - повсюду.» и сегмент END.

Обратившись к временной диаграмме, показанной на Фиг.25B, рассмотрим настройки DTS-меток и PTS-меток для каждого функционального в четырех отображаемых наборах.

Временная шкала воспроизведения на этой временной диаграмме является такой же, как временная шкала, показанная на Фиг.24. На временной диаграмме, показанной на Фиг.25A, метки PTS(PCS1.1), PTS(PCS1.2), PTS(PCS1.3) и PTS(PCS2) установлены соответственно на момент t1 отображения для отображения «что», на момент t1 отображения для отображения «что», на момент t2 отображения для отображения «что такое», на момент t3 отображения для отображения «что такое blu-ray.», и на момент t4 отображения для отображения «blu-ray - повсюду.». Каждая PTS-метка устанавливается так, как сказано выше, потому что необходимо, чтобы управляющие действия, такие как «вырезание», описанные в каждом PCS-сегменте, были выполнены в момент отображения каждого субтитра.

Метки PTS(ODS1) и PTS(ODS2) устанавливаются таким образом, чтобы указывать моменты времени, вычисленные путем вычитания decode_duration из моментов времени, указанных метками PTS(PCS1.1) и PTS(PCS2) соответственно, поскольку требуется устанавливать PTS(PCS) таким образом, чтобы удовлетворять формуле, приведенной ниже.

$$PTS(DSn[PCS]) \geq DTS(DSn[PCS]) + \text{decodeduration}(DSn)$$

На Фиг.25B PTS(ODS2) устанавливается таким образом, чтобы указывать момент t5, наступающий ранее момента t4, а PTS(ODS1) устанавливается таким образом, чтобы указывать момент t0, наступающий ранее момента t1.

Метки DTS(ODS1) и DTS(ODS2) устанавливаются таким образом, чтобы указывать моменты времени, вычисленные путем вычитания decode_duration из моментов времени, указанных метками PTS(ODS1) и PTS(ODS2) соответственно, поскольку требуется устанавливать DTS(ODS) таким образом, чтобы удовлетворять уравнению, приведенному ниже.

$$PTS(DS[ODSj]) = DTS(DSn[ODSj]) + 90000 \times (\text{SIZE}(DSn[ODSj]) / R_d)$$

На Фиг.25B PTS(ODS2) устанавливается таким образом, чтобы указывать момент t5, наступающий ранее момента t0, а PTS(ODS1) устанавливается таким образом, чтобы указывать момент, наступающий ранее момента t0. Здесь соблюдается соотношение, имеющее вид: $DTS(ODS2) = PTS(ODS1)$.

Благодаря установке PTS-метки ODS-сегмента непосредственно после PTS-метки предыдущего ODS-сегмента, который должен быть отображен раньше, устройство воспроизведения выполняет операцию, при которой ODS-сегмент считывается в память таким образом, чтобы затереть собой предыдущий ODS-сегмент, и, таким образом, возможно выполнения процесса воспроизведения посредством малого размера памяти. Благодаря реализации такого процесса воспроизведения возможности выбора размера памяти для устройства воспроизведения становятся шире.

DTS-метка сегмента PCS1.1 устанавливается таким образом, чтобы $DTS(PCS1.1) = DTS(ODS1)$, поскольку значение DTS сегмента PCS1.1 может быть любым моментом времени до момента, указанного меткой DTS(ODS1).

PTS-метка сегмента ODS1, DTS-метка сегмента ODS2 и PTS-метка сегментов PCS1.2, PCS1.3 и PCS2 устанавливаются на момент t0, так чтобы удовлетворять соотношению, выраженному приведенным ниже уравнением.

$$PTS(ODS1) = DTS(ODS2) = PTS(PCS1.2) = PTS(PCS1.3) = PTS(PCS2)$$

Это объясняется тем, что значения DTS-меток сегментов PCS1.2 и PCS1.3 могут

представлять собой любые моменты времени до момента времени, указанного меткой PTS(PCS1.3), и DTS-метка сегмента PCS2 может представлять собой любой момент до момента, указанного меткой DTS(PCS2).

Как было сказано выше, считывая более чем один PCS-сегмент в одно и то же время, можно выполнять обновление последующих PCS-сегментов, как только завершено обновление предыдущего PCS-сегмента.

Достаточно, чтобы DTS-метка и PTS-метка PCS-сегмента и DTS-метка и PTS-метка ODS-сегмента удовлетворяли соотношениям, заданным вышеприведенными формулами. Соответственно становится возможным, чтобы были установлены следующие значения: $DTS(ODS2)=PTS(ODS1)$ или $PTS(ODS1)=DTS(ODS2)=PTS(PCS1.2)=PTS(PCS1.3)=DTS(PCS2)$.

Благодаря таким настройкам временных меток можно отрегулировать продолжительность периода, в который возрастает нагрузка по декодированию или требуется больше буферов. Такое регулирование расширяет возможности управления во время воспроизведения, и выгодно для тех, кто осуществляет авторский процесс или производит устройства воспроизведения.

Рассмотренные выше структуры данных для Отображаемых наборов (PCS-сегмент, WDS-сегмент, PDS-сегмент, ODS-сегмент) представляют собой экземпляр структуры класса, описанной на языке программирования. Производители, осуществляющие авторский процесс, могут получить эти структуры данных на диске BD-ROM, описывая структуру класса в соответствии с синтаксической структурой, предусмотренной в Формате предзаписи диска Blu-ray.

Далее, ниже рассматривается практический пример устройства воспроизведения в соответствии с настоящим изобретением. Фиг.26 иллюстрирует внутреннюю структуру устройства воспроизведения в соответствии с настоящим изобретением. Устройство воспроизведения в соответствии с настоящим изобретением производится промышленным способом на основе внутренней структуры, показанной на этом чертеже. Устройство воспроизведения в соответствии с настоящим изобретением в основном образовано из трех частей: системной большой интегральной схемы, дисководом и микрокомпьютерной системы, и можно осуществлять промышленное производство данного устройства воспроизведения, монтируя эти три части на корпусе или плате этого устройства. Системная большая интегральная схема представляет собой интегральную схему, в которой интегрированы разнообразные блоки обработки данных, предназначенные для выполнения функций данного устройства воспроизведения. Устройство воспроизведения, произведенное вышеописанным образом, содержит дисковод (1) для BD-диска (диска по технологии Blu-ray), Буфер (2) считывания, Фильтр (3) PID (идентификаторов пакетов), Транспортные буферы (4a-4c), периферийную схему (4d), Декодер (5) видеоданных, Видеоматрицу (6), Декодер (7) аудиоданных, Графическую матрицу (8), блок (9) CLUT (блок управления цветом, яркостью и прозрачностью), блок (10) суммирования, Декодер (12) графических данных, Буфер (13) закодированных данных, периферийную схему (13a), Процессор (14) потоковых графических данных, Буфер (15) объектов, Буфер (16) композиции и Графический контроллер (17).

Дисковод (1) для BD-диска осуществляет загрузку/считывание/выдачу диска BD-ROM и доступ к диску BD-ROM.

Буфер (2) считывания представляет собой память обратного магазинного типа, предназначенную для хранения TS-пакетов (пакетов транспортного потока),

считанных с диска BD-ROM, в обратном магазинном порядке.

Фильтр (3) PID (идентификаторов пакетов) осуществляет фильтрацию более чем одного TS-пакета, выводимого из Буфера (2) считывания. Фильтрация, осуществляемая Фильтром (3) PID, заключается в том, чтобы записывать в Транспортные буферы (4а-4с) только те TS-пакеты, которые имеют требуемый PID (идентификатор пакета). Для фильтрации, осуществляемой Фильтром (3) PID, буферизация не требуется, и соответственно TS-пакеты, вводимые в Фильтр (3) PID, записываются в Транспортные буферы (4а-4с) без задержки.

Транспортные буферы (4а-4с) предназначены для хранения TS-пакетов, выводимых из Фильтра (3) PID, в обратном магазинном порядке. Скорость, с которой TS-пакеты выводятся из Транспортных буферов (4а-4с), представляет собой скорость Rx. Периферийная схема (4d) представляет собой «монтажную» логику для преобразования TS-пакетов, считанных из Транспортных буферов (4а-4с), в функциональные сегменты. Функциональные сегменты, полученные посредством этого преобразования, хранятся в Буфере (13) закодированных данных.

Декодер (5) видеоданных осуществляет декодирование этих, более чем один, TS-пакетов, выводимых из Фильтра (3) PID, в разуплотненный кинокадр и записывает его в Видеоматрицу (6)

Видеоматрица (6) представляет собой память, типа матрицы, предназначенную для киноизображения.

Декодер (7) аудиоданных осуществляет декодирование TS-пакетов, выводимых из Фильтра (3) PID, и выводит разуплотненные аудиоданные.

Графическая матрица (8) представляет собой память, типа матрицы, имеющую область для одного экранного изображения и способную хранить разуплотненные графические данные для одного экранного изображения. Блок (9) CLUT преобразует задаваемый индексом цвет из разуплотненных графических данных, хранящихся в Графической матрице (8), основываясь при этом на значениях Y, Cг и Cб, указанных PDS-сегментом.

Блок (10) суммирования умножает разуплотненные графические данные, для которых Блоком (9) CLUT было выполнено преобразование цвета, на значение Т (прозрачности), указанной PDS-сегментом, по пикселям суммирует с ними выделенные в отдельную составляющую данные кинокадра, хранящиеся в Видеоматрице, после чего получает и выводит составное изображение.

Декодер (12) графических данных осуществляет декодирование потока графических данных с целью получения разуплотненных графических данных, и осуществляет запись разуплотненных графических данных в Графическую матрицу (8). Благодаря декодированию потока графических данных на экране появляются субтитры и меню. Декодер (12) графических данных включает в себя Буфер (13) закодированных данных, периферийную схему (13а), Процессор (14) потоковых графических данных, Буфер (15) объектов, Буфер (16) композиции и Графический контроллер (17).

Буфер (13) закодированных данных представляет собой буфер, в котором функциональный сегмент хранится наряду с DTS-меткой и PTS-меткой. Функциональный сегмент получают посредством удаления заголовка TS-пакета и заголовка PES-пакета из каждого TS-пакета в транспортном потоке, хранящемся в Транспортных буферах (4а-4с), и посредством расположения полезных данных этих пакетов последовательно. PTS-метки и DTS-метки из удаленных заголовка TS-пакета и заголовка PES-пакета сохраняются после установления соответствия между PES-

пакетами.

Периферийная схема (13а) представляет собой «монтажную» логику, которая реализует передачу данных между Буфером (13) закодированных данных и Процессором (14) потоковых графических данных, и передачу данных между Буфером (13) закодированных данных и Буфером (16) композиции. В ходе операции передачи данных, когда текущее время является временем, указанным DTS-меткой ODS-сегмента, ODS-сегмент передается из Буфера (13) закодированных данных в Процессор (14) потоковых графических данных. Когда же текущее время является временем, указанным DTS-меткой PCS-сегмента и PDS-сегмента, PCS-сегмент и PDS-сегмент передаются в Буфер (16) композиции.

Процессор (14) потоковых графических данных осуществляет декодирование ODS-сегмента и осуществляет запись разуплотненных графических данных с цветом, задаваемым индексом, полученных при декодировании, в качестве Графического объекта в Буфер (15) объектов. Декодирование, осуществляемое Процессором (14) потоковых графических данных, начинается в момент времени, установленный DTS-меткой, соответствующей этому ODS-сегменту, и заканчивается в момент окончания декодирования, указанного PTS-меткой, соответствующей этому ODS-сегменту. Скорость R_d декодирования графического объекта представляет собой скорость вывода данных Процессором (14) потоковых графических данных.

Буфер (15) объектов представляет собой буфер, который соответствует «буферу пикселей» по терминологии стандарта ETSI EN 300 743 (стандарт Европейского института стандартов в области телекоммуникаций), и в котором размещается Графический объект, полученный посредством декодирования, выполняемого Процессором (14) потоковых графических данных. Буфер (15) объектов должен быть сделан в два или четыре раза большим, чем Графическая матрица (8), поскольку в случае, когда выполняется эффект прокрутки, Буфер (15) объектов должен хранить Графический объект, который в два или четыре раза больше Графической матрицы.

Буфер (16) композиции представляет собой память, в которой размещаются PCS-сегмент и PDS-сегмент.

Графический контроллер (17) осуществляет декодирование PCS-сегмента, размещенного в Буфере (16) композиции, и осуществляет управление, основанное на этом PCS-сегменте. Момент времени для осуществления этого управления основан на PTS-метке, прикрепленной к этому PCS-сегменту.

Далее рассматриваются рекомендуемые значения скорости передачи данных и размера буфера, служащие для определения структуры Фильтра (3) PID, Транспортных буферов (4а-4с), Графической матрицы (8), блока (9) CLUT, Буфера (13) закодированных данных и Графического контроллера (17). Фиг.27 иллюстрирует величины скоростей записи R_x , R_c и R_d , емкости Графической матрицы (8), Буфера (13) закодированных данных, Буфера (15) объектов и Буфера (16) композиции.

Скорость R_c передачи данных между Буфером (15) объектов и Графической матрицей (8) является самой высокой скоростью передачи данных в устройстве воспроизведения согласно настоящему варианту осуществления изобретения и имеет вычисленное значение 256 мегабит в секунду ($= 500$ килобайтов $\times 29,97 \times 2$), полученное на основе размера окна и частоты кадров.

В отличие от R_c при скорости R_d передачи данных (скорость декодирования пикселей) между Процессором (14) потоковых графических данных и Буфером (15)

объектов не требуется обновлять каждый цикл видеокадра, и для Rd достаточным является значение, равное 1/2 или 1/4 от значения Rc. Соответственно Rd равно либо 128 мегабитам в секунду, либо 64 мегабитам в секунду.

5 Скорость Rx вывода данных из Транспортного буфера, действующая на участке между Транспортным буфером (4а-4с) и Буфером (13) закодированных данных, представляет собой скорость передачи ODS-сегментов в уплотненном состоянии. Соответственно для скорости Rx вывода данных из Транспортного буфера достаточным является значение, равное скорости Rd передачи данных, умноженной
10 на коэффициент уплотнения. При условии, что коэффициент уплотнения ODS-сегмента составляет 25%, достаточным является значение 16 мегабит в секунду (= 64 мегабит в секунду × 25%).

Скорости передачи данных и емкости буферов, показанные на чертеже, являются минимальным стандартом, и можно устанавливать более высокие скорости и
15 большие емкости.

В устройстве воспроизведения, имеющем вышеописанную структуру, каждые элементы выполняют операцию декодирования в конвейерной структуре.

Фиг.28 представляет собой временную диаграмму, иллюстрирующую
20 конвейерную обработку данных, осуществляемую этим устройством воспроизведения. 5-й ряд на чертеже представляет собой Отображаемый набор на диске BD-ROM, 4-й ряд показывает периоды считывания из PCS-сегмента, WDS-сегмента, PDS-сегмента и ODS-сегмента в Буфер (13) закодированных данных. 3-й ряд показывает периоды декодирования каждого ODS-сегмента Процессором (14)
25 потоковых графических данных. 1-й ряд показывает операции, которые выполняет Графический контроллер (17).

DTS-метки (метки времени начала декодирования), прикрепленные к сегментам ODS1 и ODS2, указывают показанные на чертеже моменты времени t31
30 и t32 соответственно. Поскольку время начала декодирования устанавливается DTS-меткой, то требуется считать каждый ODS-сегмент в Буфер (13) закодированных данных. Соответственно считывание ODS1 в Буфер (13) закодированных данных завершается до наступления периода dp1 декодирования, в ходе которого ODS1 декодируется. Также считывание ODS2 в Буфер (13) закодированных данных
35 завершается до наступления периода dp2 декодирования, в ходе которого ODS2 декодируется.

С другой стороны, PTS-метки (метки времени окончания декодирования), прикрепленные к сегментам ODS1 и ODS2, указывают показанные на чертеже
40 моменты времени t32 и t33 соответственно. Декодирование ODS1 Процессором (14) потоковых графических данных завершается к моменту времени t32, а декодирование ODS2 завершается к моменту времени, указанному t33. Как было сказано выше, Процессор (14) потоковых графических данных считывает ODS-сегмент в Буфер (13) закодированных данных к моменту времени, указанному DTS-меткой ODS-сегмента, и декодирует ODS-сегмент, считанный в Буфер (13)
45 закодированных данных, к моменту времени, указанному PTS-меткой ODS-сегмента, и записывает декодированный сегмент в Буфер (15) объектов.

Период cd1, показанный в 1-ом ряду на чертеже, указывает период, необходимый
50 Графическому контроллеру (17) для очистки Графической матрицы. Также период td1 указывает период, необходимый для записи Графического объекта, полученного в Буфере объектов в Графическую матрицу (8). PTS-метка WDS-сегмента указывает конечный срок для начала записи, а PTS-метка PCS-сегмента

указывает окончание записи и момент времени для отображения. В момент времени, указанный PTS-меткой PCS-сегмента, разуплотненные графические данные для составления интерактивного экранного изображения получены в Графической матрице (8).

После того, как блок (9) CLUT выполняет преобразование цвета для разуплотненных графических данных, и блок (10) суммирования выполняет композицию выделенных в отдельную составляющую графических данных и выделенного в отдельную составляющую кинокадра, хранящегося в

Видеоматрице (6), получают составное изображение.

В декодере (12) графических данных процессор (14) потоковых графических данных непрерывно выполняет декодирование, в то время как Графический контроллер (17) выполняет очистку Графической матрицы (8). Благодаря вышеописанной конвейерной обработке данных имеется возможность выполнять скорое отображение графических данных.

На Фиг.28 рассматривается случай, при котором очистка Графической матрицы заканчивается раньше завершения декодирования ODS-сегмента. Фиг.29 иллюстрирует временную диаграмму для конвейерной обработки данных в случае, при котором декодирование ODS-сегмента заканчивается до завершения очистки Графической матрицы. В этом случае невозможно осуществить запись в Графическую матрицу в момент завершения декодирования ODS-сегмента. Когда очистка Графической матрицы завершена, становится возможным записать графические данные, полученные в результате декодирования, в Графическую матрицу.

Далее рассматривается хронологическое изменение в заполнении буферов. Фиг.30 представляет собой временную диаграмму, показывающую хронологические изменения в следующих компонентах, показанных на Фиг.26: Буфере (16) композиции, Буфере (15) объектов, Буфере (13) закодированных данных и Графической матрицы (8). Ряды с первого по четвертый иллюстрируют хронологические изменения в заполнении Графической матрицы (8), Буфера (15) объектов, Буфера (13) закодированных данных и Буфера (16) композиции соответственно. При этом хронологическое изменение описано с использованием линейного графика, на котором поперечная ось представляет временную шкалу, а продольная ось представляет заполнение.

Четвертый ряд на Фиг.30 показывает хронологическое изменение в заполнении для Буфера (16) композиции. Как показывает четвертый ряд, хронологическое изменение для Буфера (16) композиции включает в себя участок «vfo» простого возрастания, которое объясняется сохранением PCS-сегмента, выводимого из Буфера (13) закодированных данных.

Третий ряд показывает хронологическое изменение в заполнении для Буфера (13) закодированных данных. Как показывает третий ряд, хронологическое изменение для Буфера (13) закодированных данных включает в себя два участка «vf1» и «vf2» простого возрастания и два участка «vg1» и «vg2» простого убывания. Углы наклона участков «vf1» и «vf2» простого возрастания зависят от скорости Rx вывода данных из Транспортных буферов (4a, b, c) в Буфер (13) закодированных данных, а углы наклона участков «vg1» и «vg2» простого убывания представляют декодирование, выполненное процессором (14) потоковых графических данных, которое выполняется мгновенно. Иначе говоря, декодирование для ODS-сегмента выполняется мгновенно, и процессор (14) потоковых графических данных сохраняет

разуплотненные графические данные, полученные посредством декодирования. Скорость записи канала передачи от процессора (14) потоковых графических данных до буфера (15) объектов составляет 128 мегабит в секунду. Следовательно, заполнение буфера (15) объектов возрастает в соответствии с этой скоростью записи.

Второй ряд представляет хронологическое изменение в заполнении для буфера (15) объектов. Как показывает второй ряд, хронологическое изменение для буфера (15) объектов включает в себя участки *vh1* и *vh2* простого возрастания, которые объясняются сохранением ODS-сегментов, выводимых из процессора (14) потоковых графических данных. Углы наклона участков *vh1* и *vh2* простого возрастания зависят от скорости *Rc* передачи данных из процессора (14) потоковых графических данных в буфер (15) объектов. Периоды времени, во время которых участки простого убывания в третьем ряду и участки простого возрастания во втором ряду имеют место быть, соответствуют «периоду декодирования». Начало такого периода декодирования указывается DTS-меткой ODS-сегмента, а конец этого периода декодирования указывается PTS-меткой ODS-сегмента. Если разуплотненные графические данные хранятся в буфере (15) объектов до момента времени, показанного DTS-меткой ODS-сегмента, то это означает завершение декодирования, направленного на ODS-сегмент. При условии, что разуплотненные графические данные хранятся в Буфере (15) объектов до момента времени, показанного PTS-меткой ODS-сегмента, участки простого возрастания и участки простого убывания во время этого периода декодирования могут принимать любую форму.

Первый ряд представляет хронологическое изменение в заполнении для Графической матрицы (8). Как показывает первый ряд, хронологическое изменение для Графической матрицы (8) включает в себя участок *vf3* простого возрастания, который объясняется сохранением уже декодированного ODS-сегмента, выводимого из Буфера (15) объектов. Угол наклона участка *vf3* простого возрастания зависит от скорости *Rd* передачи данных из Буфера (15) объектов в Графическую матрицу (8). Конец участка простого возрастания показан PTS-меткой ODS-сегмента.

Графики, такие как те, что показаны на Фиг.27, проиллюстрированы с использованием: DTS-метки и PTS-метки, присвоенных ODS-сегменту, DTS-метки и PTS-метки, присвоенных ICS-сегменту, емкостей и скоростей передачи данных каждого буфера, проиллюстрированного на Фиг.27. Кроме того, строя графики, такие как те, что на этой диаграмме, пользователи на стадии авторского процесса могут узнать, каким образом меняется состояние каждого буфера.

Поскольку изменение состояния каждого буфера может быть отрегулировано посредством корректировки DTS-метки и PTS-метки, становится возможным избегать возникновения в устройстве воспроизведения такой нагрузки по декодированию, которая превысила бы технические характеристики декодера, и избегать возникающего при воспроизведении переполнения буфера. В соответствии с этим на стадии разработки устройства воспроизведения реализация его аппаратного/программного обеспечения станет проще.

Далее, ниже рассматривается то, каким образом реализованы блок (20) управления и Декодер (12) графических данных. Блок (20) управления реализован в результате написания программы, выполняющей операцию, показанную на Фиг.30, и наличия центрального процессора общего назначения, предусматриваемого для исполнения этой программы. Операция, выполняемая блоком (20) управления, рассматривается со ссылкой на Фиг.30.

Фиг.31 представляет собой блок-схему алгоритма, иллюстрирующую процесс операции загрузки функционального сегмента. На этой блок-схеме алгоритма SegmentK представляет собой переменную, указывающую каждый из сегментов (PCS-сегмент, WDS-сегмент, PDS-сегмент и ODS-сегмент), который считан при воспроизведении потока AVClip. Признак игнорирования представляет собой признак, определяющий, игнорируется ли этот SegmentK или загружается. Блок-схема алгоритма имеет структуру цикла, в которой сначала задается начальное значение признака игнорирования, равное 0, а затем для каждого SegmentK (этап S25 и этап S26) повторяются этапы S21-S24 и этапы S27-S31.

Этап S21 предназначен для определения того, является ли SegmentK PCS-сегментом, и если SegmentK является PCS-сегментом, то выполняются операции определения на этапе S27 и S28.

Этап S22 предназначен для определения того, равен ли 0 признак игнорирования. Если признак игнорирования равен 0, то операция переходит на этап S23, а если признак игнорирования равен 1, то операция переходит на этап S24. Если признак игнорирования равен 0 (ответ «Да» на этапе S22), то на этапе S23 SegmentK загружается в Буфер (13) закодированных данных.

Если признак игнорирования равен 1 (ответ «Нет» на этапе S22), то на этапе S24 SegmentK игнорируется. При этом все остальные функциональные сегменты, которые принадлежат этому DS-набору, игнорируются, поскольку на этапе S22 ответ «Нет» (этап S24).

Как было сказано выше, то, игнорируется ли SegmentK или загружается, определяется признаком игнорирования. Этапы S27-S31, S34 и S35 представляют собой этапы для установки признака игнорирования.

На этапе S27 производится определение того, является ли segment_type (тип сегмента) этого SegmentK «Точкой захвата». Если SegmentK является «Точкой захвата», то операция переходит на этап S28, а если SegmentK является либо «Началом Сверхкадра» или «Нормальным случаем», то операция переходит на этап S31.

На этапе S28 производится определение того, имеется ли предыдущий DS-набор в каком-либо из буферов в Декодере (12) графических данных (буфере (13) закодированных данных, процессоре (14) потоковых графических данных, буфере (15) объектов и буфере (16) композиции). Определение на этапе S28 производится в том случае, когда определение на этапе S27 дает ответ «Да». Случай, при котором предыдущий DS-набор в Декодере (12) графических данных не имеется, показывает случай, при котором выполняется операция пропуска. В этом случае отображение начинается с того DS-набора, который является «Точкой захвата», и, следовательно, операция переходит на этап S30 (ответ «Нет» на этапе S28). На этапе S30 признак игнорирования устанавливается в 0, и операция переходит на этап S22.

Случай, при котором предыдущий DS-набор в Декодере (12) графических данных имеется, показывает случай, при котором выполняется нормальное воспроизведение. В этом случае операция переходит на этап S29 (ответ «Да» на этапе S28). На этапе S29 признак игнорирования устанавливается в 1, и операция переходит на этап S22.

На этапе S31 производится определение того, является ли composition_state (состояние композиции) PCS-сегмента «Нормальным случаем». Если PCS-сегмент является «Нормальным случаем», то операция переходит на этап S34, а если PCS-

сегмент является «Началом Сверхкадра», то на этапе S30 признак игнорирования устанавливается в 0.

На этапе S34, как и на этапе S28, производится определение того, имеется ли предыдущий DS-набор в каком-либо из буферов в Декодере (12) графических данных. Если предыдущий DS-набор имеется, то признак игнорирования устанавливается в 0 (этап S30). Если предыдущий DS-набор не имеется, то невозможно получить достаточного количества функциональных сегментов для составления интерактивного экранного изображения и признак игнорирования устанавливается в 1 (этап S35).

Благодаря установке признака игнорирования, производимой вышеописанным образом, функциональные сегменты, которые составляют «Нормальный случай», игнорируются в том случае, когда в Декодере (12) графических данных не имеется предыдущего DS-набора.

Взяв в качестве примера случай, при котором DS-набор мультиплексирован так, как показано на Фиг.31, рассмотрим то, каким образом осуществляется считывание DS-набора. В примере, приведенном на Фиг.31, три DS-набора мультиплексированы с кинофильмом. Composition_state (состояние композиции) набора DS1 представляет собой «Начало Сверхкадра», composition_state набора DS10 представляет собой «Точку захвата», а composition_state набора DS20 представляет собой «Нормальный случай».

При условии, что в потоке AVClip, в котором мультиплексированы эти три DS-набора и кинофильм, выполняется операция пропуска до данных кинокадра pt10, что показано стрелкой am1, набор DS10 является ближайшим к целевой точке операции пропуска, и, следовательно, DS10 и является DS-набором, описанным на блок-схеме алгоритма на Фиг.30. Хотя на этапе S27 определено, что composition_state является «Точкой захвата», признак игнорирования устанавливается в 0 (поскольку в Буфере (13) закодированных данных не имеется предыдущего DS-набора), и набор DS10 загружается в Буфер (13) закодированных данных, входящий в состав устройства воспроизведения, по стрелке md1 на Фиг.32. С другой стороны, в случае, при котором целевая точка операции пропуска находится после набора DS10 (стрелка am2 на Фиг.31), набор DS20 должен быть проигнорирован, поскольку DS20 представляет собой Отображаемый набор «Нормального случая» и поскольку в Буфере (13) закодированных данных не имеется предыдущего DS-набора (стрелка md2 на Фиг.32).

Фиг.33 иллюстрирует загрузку наборов DS1, DS10 и DS20 при нормальном воспроизведении. Набор DS1, у которого composition_state (состояние композиции) PCS-сегмента представляет собой «Начало Сверхкадра», загружается в Буфер (13) закодированных данных в полном составе (этап S23). Однако, поскольку признак игнорирования в наборе DS10, у которого composition_state (состояние композиции) PCS-сегмента представляет собой «Точку захвата», установлен в 1 (этап S29), функциональные сегменты, которые составляют набор DS10, игнорируются и не загружаются в Буфер (13) закодированных данных (стрелка rd2 на Фиг.34 и этап S24). Далее, набор DS20 загружается в Буфер (13) закодированных данных, поскольку composition_state (состояние композиции) PCS-сегмента набора DS20 представляет собой «Нормальный случай» (стрелка rd3 на Фиг.34 и этап S34).

Далее рассматриваются операции Графического контроллера (17). Фиг.35-37 иллюстрируют блок-схему алгоритма, показывающую операции, выполняемые Графическим контроллером (17).

Этапы S41-S44 представляют собой этапы основной программы этой блок-схемы и являются этапами ожидания наступления любого из событий, предусмотренных на этапах S41-S44.

Этап S41 предназначен для определения того, совпадает ли текущее время воспроизведения с временем, указанным DTS-меткой PCS-сегмента, и если определено, что «Да», то выполняется операция на этапах S45-S53.

Этап S45 предназначен для определения того, установлен ли элемент `composition_state` (состояние композиции) из OCS-сегмента в значение `epoch_start` (Начало Сверхкадра), и если определено, что `composition_state` установлен в значение `epoch_start`, то на этапе S46 осуществляется очистка всей Графической матрицы (8). Если же определено, что `composition_state` отлично от `epoch_start`, то очищается Окно, указанное значениями `window_horizontal_position` (местоположение окна по горизонтали), `window_vertical_position` (местоположение окна по вертикали), `window_width` (ширина окна) и `window_height` (высота окна) из состава WDS-сегмента.

Этап S48 представляет собой этап, выполняемый после очистки, выполненной на этапе S46 или на этапе S47, и предназначен для определения того, прошло ли время, указанное PTS-меткой какого-либо сегмента ODSx. Декодирование какого-либо сегмента ODSx могло бы быть уже завершено к моменту времени, когда заканчивается очистка, поскольку очистка всей Графической матрицы (8) требует времени. Следовательно, на этапах S48 определяется, завершено ли уже к моменту времени, когда заканчивается очистка, декодирование какого-либо сегмента ODSx. Если определено, что «Нет», то операция возвращается к основной программе. Если время, указанное PTS-меткой какого-либо сегмента ODSx уже прошло, то выполняется операция на этапах S49-S51. На этапе S49 производится определение того, равен ли `object_cropped_flag` (признак «вырезанного» объекта) 0, и если признак показывает 0, то Графический объект устанавливается в «неотображаемое состояние» (этап S50).

Если на этапе S49 признак не равен 0, то в Окно на Графической матрице (8) в место, указанное значениями `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали), записывается объект, «вырезанный» на основе значений `object_cropping_horizontal_position`, `object_cropping_vertical_position`, `cropping_width` (ширина «вырезания») и `cropping_height` (высота «вырезания») (этап S51). Посредством вышеописанной операции один или более Графических Объектов визуализируются в этом Окне.

На этапе S52 производится определение того, прошло ли время, соответствующее PTS-метке другого сегмента ODSy. При записи ODSx в Графическую матрицу (8), если декодирование ODSy уже завершено, то ODSy становится ODSx (этап S53), и операция переходит на этап S49. При этом операция, описанная на этапах S49-S51, также выполняется и для другого ODS-сегмента.

Далее, ниже со ссылкой на Фиг.36 рассматриваются этап S42 и этапы S54-59.

На этапе S42 производится определение того, совпадает ли текущий момент времени воспроизведения с PTS-меткой WDS-сегмента. Если определено, что текущий момент времени воспроизведения совпадает с PTS-меткой WDS-сегмента, то на этапе S54 производится определение того, равно или нет количество Окон одному. Если определено, что количество Окон равно двум, то операция возвращается к основной программе. Если определено, что количество Окон равно

одному, то выполняется циклическая обработка данных на этапах S55-S59. В ходе этой циклической обработки данных операции на этапах S55-S59 выполняются для каждого из двух Графических объектов, отображаемых в Окне. На этапе S57 производится определение того, показывает ли `object_cropped_flag` (признак «вырезанного» объекта) 0. Если он показывает 0, то графические данные не отображаются (этап S58).

Если он не показывает 0, то в Окно на Графической матрице (8) в место, указанное значениями `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали), записывается объект, «вырезанный» на основе значений `object_cropping_horizontal_position`, `object_cropping_vertical_position`, `cropping_width` (ширина «вырезания») и `cropping_height` (высота «вырезания») (этап S59). Посредством повторения вышеописанных операций более чем один Графических Объектов визуализируются в этом Окне.

На этапе S44 производится определение того, совпадает ли текущий момент времени воспроизведения с PTS-меткой PDS-сегмента. Если определено, что текущий момент времени воспроизведения совпадает с PTS-меткой PDS-сегмента, то на этапе S60 производится определение того, равен ли единице `pallet_update_flag` (признак обновления палитры). Если определено, что этот признак равен единице, то PDS-сегмент, указанный посредством `Palette_id` (идентификатора палитры), устанавливается в блоке CLUT (этап S61). Если же определено, что этот признак равен 0, то этап S61 пропускается.

После этого блок CLUT выполняет преобразование цвета графического объекта на Графической матрице (8), подлежащего объединению с киноизображением (этап S62).

Далее, ниже со ссылкой на Фиг.37 рассматриваются этап S43 и этапы S64-S66.

На этапе S43 производится определение того, совпадает ли текущий момент времени воспроизведения с PTS-меткой ODS-сегмента. Если определено, что текущий момент времени воспроизведения совпадает с PTS-меткой ODS-сегмента, то на этапе S63 производится определение того, равно или нет количество Окон двум. Если определено, что количество Окон равно одному, то операция возвращается к основной программе. Если определено, что количество Окон равно двум, то выполняются операции на этапах S64-S66. На этапе S64 производится определение того, показывает ли `object_cropped_flag` (признак «вырезанного» объекта) 0. Если он показывает 0, то графические данные не отображаются (этап S65).

Если он не показывает 0, то в Окно на Графической матрице (8) в место, указанное значениями `object_cropping_horizontal_position` (место «вырезания» объекта по горизонтали) и `object_cropping_vertical_position` (место «вырезания» объекта по вертикали), записывается «вырезанный» объект на основе значений `object_cropping_horizontal_position`, `object_cropping_vertical_position`, `cropping_width` (ширина «вырезания») и `cropping_height` (высота «вырезания») (этап S66). Посредством повторения вышеописанных операций в каждом окне визуализируется Графический Объект.

Вышеприведенные разъяснения касаются DTS-метки и PTS-метки PCS-сегмента, и DTS-метки и PTS-метки ODS-сегмента, которые принадлежат набору DSn. DTS-метка и PTS-метка PDS-сегмента, и DTS-метка и PTS-метка END-сегмента не рассмотрены. Сначала рассмотрим DTS-метку и PTS-метку PD, который принадлежит набору DSn.

Что касается PDS-сегмента, который принадлежит набору DS_n , то достаточно, чтобы PDS-сегмент имелся в блоке (9) CLUT на тот момент, когда PCS-сегмент загружается в Буфер (16) композиции ($DTS(DS_n[PCS])$) после момента начала декодирования первого ODS-сегмента ($DTS(DS_n[ODS1])$). Соответственно требуется

$$DTS(DS_n[PCS]) \leq PTS(DS_n[PDS1])$$

$$PTS(DS_n[PDS_j]) \leq PTS(DS_n[PDS_{j+1}]) \leq PTS(DS_n[PDS_{\text{последний}}])$$

$$PTS(DS_n[PDS_{\text{последний}}]) \leq DTS(DS_n[ODS1])$$

Отметим, что для того, чтобы удовлетворить требования стандарта MPEG2, обращение к DTS-метке PDS-сегмента во время воспроизведения не производится, DTS-метка ODS-сегмента устанавливается в то же самое значение, что и PTS-метка PDS-сегмента.

Ниже приводится разъяснение, касающееся роли DTS-метки и PTS-метки в конвейерной обработке данных в данном устройстве воспроизведения, в случае, когда DTS-метка и PTS-метка установлены таким образом, чтобы удовлетворять вышеприведенным соотношениям. Фиг.38 иллюстрирует конвейер устройства воспроизведения, который основан на PTS-метке PDS-сегмента. Фиг.38 основана на Фиг.26. Первый ряд на Фиг.38 показывает установку ODS-сегмента в блок (9) CLUT. Под первым рядом располагаются ряды, одинаковые с рядами с первого по пятый на Фиг.26. Установка сегментов PDS1-последний в блок (9) CLUT выполняется одновременно с началом декодирования ODS1 (стрелки up2, up3). (Установка сегментов PDS1-PDSпоследний в блок (9) CLUT выполняется после передачи PCS-сегмента и WDS-сегмента и до декодирования сегмента ODS1, и соответственно установка сегментов PDS1-PDSпоследний в блок (9) CLUT выполняется до момента времени, указанного DTS-меткой сегмента ODS1, что и показано стрелками up2, up3).

Как было сказано выше, установка PDS-сегмента выполняется перед декодированием ODS-сегмента.

Далее, рассматривается установка PTS-метки сегмента END (Конец Отображаемого набора) в наборе DS_n . END-сегмент, принадлежащий набору DS_n , указывает конец набора DS_n , и соответственно необходимо, чтобы PTS-метка END-сегмента указывала момент времени окончания декодирования сегмента ODS2. Момент времени окончания декодирования указывается PTS-меткой ($PTS(DS_n[ODS_{\text{последний}}])$) сегмента ODS2($ODS_{\text{последний}}$), и, следовательно, требуется, чтобы PTS-метка END-сегмента была установлена в значение, которое удовлетворяет приведенному ниже уравнению.

$$PTS(DS_n[END]) = (PTS(DS_n[ODS_{\text{последний}}]))$$

Что касается соотношения между набором DS_n и PCS-сегмента, принадлежащего набору DS_{n+1} , то PCS-сегмент в наборе DS_n загружается в Буфер (16) композиции до наступления момента времени загрузки первого ODS-сегмента ($ODS1$), и, следовательно, PTS-метка END-сегмента должна быть после момента времени загрузки PCS-сегмента, входящего в набор DS_n , и до момента времени загрузки PCS-сегмента, принадлежащего набору DS_{n+1} . Соответственно требуется, чтобы PTS-метка END-сегмента удовлетворяла приведенному ниже соотношению.

$$DTS(DS_n[PCS]) \leq PTS(DS_n[END]) \leq DTS(DS_{n+1}[PCS])$$

С другой стороны, момент времени загрузки первого ODS-сегмента ($ODS1$) предшествует моменту времени загрузки последнего PDS-сегмента ($PDS_{\text{последний}}$), и, следовательно, PTS-метка END-сегмента ($PTS(DS_n[END])$) должна быть после

момента времени загрузки PDS-сегмента, принадлежащего набору DS_n (PTS(DS_n[PDSпоследний])). Соответственно требуется, чтобы PTS-метка END-сегмента удовлетворяла приведенному ниже соотношению.

$$PTS(DS_n[PDS_{\text{последний}}]) \leq PTS(DS_n[END])$$

Ниже приводится разъяснение, касающееся значения PTS-метки END-сегмента в конвейерной обработке данных в данном устройстве воспроизведения. Фиг.39 представляет собой схему, показывающую значение END-сегмента в конвейерном процессе данного устройства воспроизведения. Фиг.39 основана на Фиг.28, и каждый ряд на Фиг.39, по существу, такой же, как и на Фиг.28, за исключением того, что первый ряд на Фиг.39 показывает содержимое Буфера (16) композиции. Кроме того, на Фиг.39 проиллюстрированы 2 Отображаемых набора: DS_n и DS_{n+1}. Сегмент ODSпоследний в наборе DS_n представляет собой последний сегмент ODS_n из числа сегментов A-ODS, и соответственно момент времени, указываемый PTS-меткой END-сегмента, предшествует DTS-метке PCS-сегмента в наборе DS_{n+1}.

При помощи PTS-метки END-сегмента можно обнаружить, когда во время воспроизведения завершается загрузка ODS-сегмента, входящего в набор DS_n.

Отметим, что хотя во время воспроизведения обращение к DTS-метке END-сегмента не производится, для того, чтобы удовлетворить требования стандарта MPEG2, DTS-метка END-сегмента устанавливается в то же самое значение, что и PTS-метка END-сегмента.

Как было сказано выше, в качестве Окна для отображения графических данных в соответствии с данным вариантом осуществления изобретения указывается часть Графической матрицы, и, следовательно, устройство воспроизведения не должно осуществлять визуализацию графических данных для всей Матрицы. Устройство воспроизведения может осуществлять визуализацию графических данных только для Окна заранее заданного размера, например, составляющего от 25% до 33% Графической матрицы. Поскольку визуализация графических данных, отличных от графических данных в Окне, не требуется, то нагрузка на программное обеспечение в устройстве воспроизведения снижается.

Даже в самом худшем случае, при котором обновление графических данных выполняется, например, на 1/4 Графической матрицы, можно посредством данного устройства воспроизведения отображать Графические данные синхронно с кинофильмом, выполняя запись в Графическую матрицу на заранее заданной скорости передачи данных, такой как 256 мегабит в секунду и устанавливая размеры Окна таким образом, чтобы обеспечить синхронное отображение с кинофильмом.

(Второй вариант осуществления изобретения)

Первый вариант осуществления изобретения, рассмотренный выше, предусматривается для графических данных, предназначенных для отображения субтитров. В противоположность этому второй вариант осуществления изобретения касается графических данных, предназначенных для интерактивного отображения.

Среди вариантов реализации носителя записи в соответствии с настоящим изобретением один пример использования этого носителя записи заключается в следующем. Как и в первом варианте осуществления изобретения, носитель записи по второму варианту осуществления изобретения также может быть изготовлен посредством усовершенствования прикладного уровня BD-ROM (Постоянного запоминающего устройства на диске, использующем технологию Blue-ray). Фиг.41 представляет собой схему, иллюстрирующую в схематическом виде структуру потока AVClip (потока данных аудио-видео фрагментов) по второму варианту

осуществления изобретения.

Поток AVClip (показанный в средней части чертежа) имеет следующую структуру. Поток видеоданных, состоящий из множества кадров видеоданных (кинокадр $rp1$, $rp2$ и $rp3$), и поток аудиоданных, состоящий из множества кадров аудиоданных (верхний ряд на чертеже), соответственно преобразуются в последовательность PES-пакетов (второй ряд на чертеже), а затем в последовательность TS-пакетов (пакетов транспортного потока) (третий ряд на чертеже). Поток интерактивных графических данных (нижний ряд на чертеже) преобразуется в последовательность PES-пакетов (второй снизу ряд на чертеже), а затем в последовательность TS-пакетов (третий снизу ряд на чертеже). Три последовательности TS-пакетов мультиплексируются, и таким образом составляется поток AVClip.

Далее рассматривается поток интерактивных графических данных. Поток интерактивных графических данных вместо PCS-сегмента имеет Сегмент композиции интерактивного изображения (ICS-сегмент) и не имеет WDS-сегмента. Поток интерактивных графических данных аналогичен потоку графических данных представления в том, что он имеет функциональные сегменты, именуемые Сегментом определения палитры (PDS-сегментом), Сегментом определения объекта (ODS-сегментом) и Сегментом конца отображаемого набора (END-сегментом).

Размещение частей графического пользовательского интерфейса на экране создает интерактивное экранное изображение, определяемое этими функциональными сегментами. Фиг.42А представляет собой схему, иллюстрирующую такое интерактивное экранное изображение, реализованное посредством потока интерактивных графических данных. Это интерактивное экранное изображение включает в себя четыре части графического пользовательского интерфейса, именуемые кнопками А - кнопками D.

Интерактивность, реализуемая посредством потока интерактивных графических данных, означает изменение состояния этих частей графического пользовательского интерфейса (то есть кнопок) в соответствии с операциями пользователя. Состояние частей графического пользовательского интерфейса (кнопок) включает в себя «нормальное состояние $bt1$ », «выделенное состояние $bt2$ » и «активное состояние $bt3$ », которые показаны на Фиг.42А. Нормальное состояние представляет собой состояние, в котором обеспечивается простое отображение. В противоположность этому выделенное состояние представляет собой состояние, в котором обеспечивается фокус (сосредоточение внимания) в соответствии с операцией пользователя, но подтверждение еще не принято. Состояние кнопки может быть изменено нажатием на клавиши пульта (400) дистанционного управления, который показан в первом варианте осуществления изобретения.

Фиг.42В представляет собой схему, иллюстрирующую клавиши пульта (400) дистанционного управления, через который принимается операция пользователя, направленная на интерактивное экранное изображение. Как показано на этом чертеже, на пульте (400) дистанционного управления предусмотрены клавиша MoveUp («Перемещение вверх»), клавиша MoveDown («Перемещение вниз»), клавиша MoveRight («Перемещение вправо») и клавиша MoveLeft («Перемещение влево»).

Клавиша «Перемещения вверх» предназначена для того, чтобы в случае, когда кнопка на интерактивном экранном изображении находится в выделенном состоянии, устанавливать в выделенное состояние кнопку, расположенную над этой выделенной кнопкой. Клавиша «Перемещения вверх» предназначена для того,

чтобы устанавливать в выделенное состояние кнопку, расположенную под этой выделенной кнопкой. Клавиша «Перемещения вправо» предназначена для того, чтобы устанавливать в выделенное состояние кнопку, расположенную справа от этой выделенной кнопки, а клавиша «Перемещения влево» предназначена для того, чтобы устанавливать в выделенное состояние кнопку, расположенную слева от этой выделенной кнопки.

Клавиша Activated («Активировано») предназначена для установки выделенной кнопки в активное состояние (то есть для ее активации). Числовые клавиши «0»-«9» предназначены для установки кнопки, которой присвоен соответствующий номер, в выделенное состояние. Клавиша «+10» предназначена для приема операции, заключающейся в прибавлении 10 к уже введенным числовым значениям. Здесь следует отметить, что клавиша «0» и клавиша «+10» обе служат для приема ввода числового значения не меньше, чем из 10 разрядов. Следовательно, для пульта (400) дистанционного управления достаточно любой одной из этих клавиш.

Каждое состояние (то есть, нормальное состояние, выделенное состояние и активное состояние) образовано из множества графических данных в разуплотненном состоянии. Каждая совокупность разуплотненных графических данных, которая предназначена для представления каждого состояния кнопок, именуется «Графическим объектом». Причина, по которой одно состояние кнопки представлено множеством разуплотненных графических данных, заключается в том, чтобы учесть выполнение для каждого состояния каждой кнопки анимационного отображения.

Далее рассматриваются усовершенствования в данном варианте осуществления изобретения, направленные на сегменты определения (ODS-сегмент, PDS-сегмент). ODS-сегмент и PDS-сегмент имеют ту же самую структуру данных, что и сегменты по первому варианту осуществления изобретения. Единственное различие в том, что касается ODS-сегмента, заключается в идентификаторе «object_ID» (идентификаторе объекта). ODS-сегмент во втором варианте осуществления изобретения создает анимацию, используя для этого множество Графических объектов, определенных множеством ODS-сегментов. При создании анимации object_ID (идентификатор объекта) добавляется к последовательности ODS-сегментов, где object_ID представляет собой порядковый номер.

Далее рассматривается ICS-сегмент. Сегмент композиции интерактивного изображения (ICS-сегмент) представляет собой функциональный сегмент, составляющий интерактивное экранное изображение. Сегмент композиции интерактивного изображения имеет структуру данных, показанную на Фиг.43. Как показано на этом чертеже, ICS-сегмент состоит из элементов: segment_type (тип сегмента); segment_length (длина сегмента); composition_number (номер композиции), composition_state (состояние композиции); command_update_flag (признак обновления команды); composition_time_out_pts (PTS-метка истечения времени композиции); selection_time_out_pts (PTS-метка истечения времени выделенного состояния); UO_mask_table (таблица маски операций пользователя); animation_frame_rate_code (код частоты кадра анимации); default_selected_button_number (номер кнопки выделяемой по умолчанию); default_activated_button_number (номер кнопки активируемой по умолчанию); и наборы информации о кнопках (button_info (1) (2) (3) ...).

«composition_number» (номер композиции) представляет числовые значения от 0 до 15, которые указывают выполнение обновления.

«composition_state» (состояние композиции) показывает, что DS-набор

(Отображаемый набор), начинающийся с данного ICS-сегмента, представляет собой «Нормальный случай», «Точку захвата» или «Начало Сверхкадра».

«command_update_flag» (признак обновления команды) показывает, изменилась ли команда кнопки в данном ICS-сегменте по сравнению с предыдущим ICS-сегментом. Например, если DS-набор, к которому принадлежит некоторый ICS-сегмент, представляет собой «Точку захвата», то этот ICS-сегмент, в принципе, будет иметь то же самое содержимое, что и ICS-сегмент, который непосредственно предшествует этому ICS-сегменту. Однако, если «command_update_flag» установлен во включенное состояние, то, как ICS-сегмент, может быть установлена команда кнопки, отличная от непосредственно предшествующего ICS-сегмента. Этот признак устанавливается в действующее состояние, когда требуется, чтобы при наложении графического объекта команда была изменена.

«composition_time_out_pts» (PTS-метка истечения времени композиции) описывает время окончания интерактивного экранного изображения. Во время этого окончания отображение интерактивного экранного изображения более не действительно и потому не осуществляется. Желательно, чтобы время «composition_time_out_pts» было описано с точностью указания времени на временной шкале воспроизведения для данных кинофильма.

«selection_time_out_pts» (PTS-метка истечения времени выделенного состояния) описывает время окончания действующего периода выделения кнопки. Во время selection_time_out_pts активируется кнопка, указанная номером default_activated_button_number (номером кнопки, активируемой по умолчанию). Период selection_time_out_pts равен или короче, чем период composition_time_out_pts. Selection_time_out_pts описывается с точностью указания времени видеокадра.

«UO_mask_table» (таблица маски операций пользователя) представляет разрешение/запрещение операций пользователя для Отображаемого набора, соответствующего данному ICS-сегменту. Если поле маски установлено как запрет, то операции пользователя, направленные на устройство воспроизведения, будут недействительны.

«animation_frame_rate_code» (код частоты кадра анимации) описывает частоту кадра, подлежащую применению к кнопке анимационного типа. Частоту кадра анимации получают делением частоты видеокадра на значение этого поля. Если значение в этом поле равно «00», то отображается только ODS-сегмент, указываемый идентификатором start_object_id_xxx (идентификатором начального объекта), но не анимация, причем этот ODS-сегмент находится среди ODS-сегментов, определяющих Графические объекты для кнопок.

«default_selected_button_number» (номер кнопки, выделяемой по умолчанию) указывает номер кнопки, которая должна быть установлена в выделенное состояние по умолчанию при начале отображения интерактивного экранного отображения. Если значение этого поля равно 0, то в активное состояние будет автоматически установлена кнопка, имеющая номер кнопки, хранящийся в регистре устройства воспроизведения. Если значение этого поля не равно 0, то это означает, что данное поле указывает действительное (то есть, имеющееся на самом деле) значение кнопки.

«default_activated_button_number» (номер кнопки, активируемой по умолчанию) представляет кнопку, подлежащую автоматической установке в активное состояние в случае, когда пользователь до наступления момента времени, определенного значением selection_time_out_pts, не установил никакую кнопку в активное состояние.

Если default_activated_button_number равен «FF», то в момент времени, определенный значением selection_time_out_pts, будет автоматически выбрана кнопка, находящаяся на текущий момент в выделенном состоянии. Если этот default_activated_button_number равен «00», то автоматический выбор не производится. Если же этот номер
 5 отличен от «00» и «FF», то это поле будет интерпретироваться как указывающее действительный (то есть существующий на самом деле) номер кнопки.

Информация о кнопке (button_info) определяет каждую кнопку, сформированную на интерактивном экранном изображении. Указательная линия на этом чертеже
 10 фокусирует внимание на внутренней структуре информации (i) о кнопке, которая относится к i-ой кнопке, управляемой ICS-сегментом. Ниже рассматриваются элементы этой информации, составляющие информацию (i) о кнопке.

Номер «button_number» представляет собой значение, уникальным образом идентифицирующее кнопку (i) в ICS-сегменте.

15 «numerically_selectable_flag» (признак возможности выбора числового значения) указывает, разрешен ли выбор числового значения для кнопки (i).

«auto_action_flag» (признак автоматического действия) указывает, устанавливать ли кнопку (i) автоматически. Если этот признак «auto_action_flag» установлен во включенное состояние (то есть значение бита равно 1), то кнопка (i) будет
 20 установлена в активное состояние вместо того, чтобы быть установленной в выделенное состояние. И наоборот, если auto_action_flag установлен в выключенное состояние (то есть значение бита равно 0), то кнопка (i) будет установлена в простое выделенное состояние, даже если эта кнопка уже была выделена.

25 «object_horizontal_position» (местоположение объекта по горизонтали) и «object_vertical_position» (местоположение объекта по вертикали) соответственно указывают местоположение по горизонтали и местоположение по вертикали верхнего левого пикселя кнопки (i) на интерактивном экранном отображении.

30 «upper_button_number» (номер верхней кнопки) указывает номер кнопки, устанавливаемой в выделенное состояние вместо кнопки (i), в случае, когда клавиша MoveUp («Перемещение вверх») была нажата в момент времени, когда кнопка (i) находится в выделенном состоянии. Если в этом поле был установлен номер, соответствующий кнопке (i), то нажатие, направленное на клавишу MoveUp
 35 («Перемещение вверх»), будет игнорироваться.

«lower_button_number» (номер нижней кнопки), «left_button_number» (номер левой кнопки) и «right_button_number» (номер правой кнопки) указывают номер кнопки, которая будет в выделенном состоянии вместо кнопки (i), в случае, когда
 40 клавиша MoveDown («Перемещение вниз»), клавиша MoveLeft («Перемещение влево») и клавиша MoveRight («Перемещение вправо») были соответственно нажаты в момент времени, когда кнопка (i) находится в выделенном состоянии. Если в этом поле был установлен номер, соответствующий кнопке (i), то нажатие, направленное на эти клавиши, будет игнорироваться.

45 «start_object_id_normal» (идентификатор начального объекта в нормальном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки (i) в нормальном состоянии осуществляется посредством анимации, в этом поле «start_object_id_normal» указывается первый номер из числа порядковых номеров, присвоенных множеству ODS-сегментов, составляющих эту анимацию.
 50

«end_object_id_normal» (идентификатор конечного объекта в нормальном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки (i) в нормальном состоянии осуществляется посредством анимации, в этом

поле «end_object_id_normal» указывается последний номер из числа порядковых номеров (то есть, object_id), присвоенных множеству ODS-сегментов, составляющих эту анимацию. Если идентификатор, указанный в end_object_id_normal, является тем же самым, что и идентификатор, указанный в start_object_id_normal, то изображением для кнопки (i) будет неподвижное изображение, соответствующее этому идентификатору в Графическом объекте.

«repeated_normal_flag» (признак повторения в нормальном состоянии) указывает, продолжать ли многократно анимационное отображение кнопки (i), пребывающей в нормальном состоянии.

«start_object_id_selected» (идентификатор начального объекта в выделенном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки (i) в выделенном состоянии осуществляется в виде анимации, в этом поле «start_object_id_selected» указывается первый номер из числа порядковых номеров, присвоенных множеству ODS-сегментов, составляющих эту анимацию.

«end_object_id_selected» (идентификатор конечного объекта в выделенном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки в выделенном состоянии осуществляется посредством анимации, в этом поле «end_object_id_selected» указывается последний номер из числа порядковых номеров, присвоенных множеству ODS-сегментов, составляющих эту анимацию. Если идентификатор, указанный в end_object_id_selected, является тем же самым, что и идентификатор, указанный в start_object_id_selected, то изображением для кнопки (i) будет неподвижное изображение, соответствующее этому идентификатору в Графическом объекте.

«repeat_selected_flag» (признак повторения в выделенном состоянии) указывает, продолжать ли многократно анимационное отображение для кнопки (i), пребывающей в выделенном состоянии. Если start_object_id_selected и end_object_id_selected имеют одно и то же значение, то это поле будет установлено в 00.

«start_object_id_activated» (идентификатор начального объекта в активированном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки (i) в активном состоянии осуществляется посредством анимации, в этом поле «start_object_id_activated» указывается первый номер из числа порядковых номеров, присвоенных множеству ODS-сегментов, составляющих эту анимацию.

«end_object_id_activated» (идентификатор конечного объекта в активированном состоянии) представляет собой такое поле, что в случае, когда визуализация кнопки в активном состоянии осуществляется посредством анимации, в этом поле «end_object_id_activated» указывается последний номер из числа порядковых номеров (то есть object_id), присвоенных множеству ODS-сегментов, составляющих эту анимацию.

Далее рассматривается команда кнопки.

Команда кнопки (button_command) представляет собой команду, подлежащую исполнению при установке кнопки (i) в активное состояние.

Ниже рассматривается пример интерактивного управления посредством ICS-сегмента. Этот пример предполагает ODS-сегмент и ICS-сегмент, такими как те, что проиллюстрированы на Фиг.44. Фиг.44 показывает связи между ODS-сегментом, включенным в набор DSn, и ICS-сегментом. Предполагается, что набор DSn содержит сегменты ODS11-19, 21-29, 31-39 и 41-49. Среди этих ODS-сегментов сегменты ODS11-19 соответственно иллюстрируют состояния кнопки А;

сегменты ODS21 - 29 иллюстрируют состояния кнопки В; ODS31-39 - состояния кнопки С; и ODS41-49 - состояния кнопки D (Просьба обратить внимание на фигурную скобку «}»). Затем, предполагается, что средства управления состояниями для этих кнопок: кнопки А - кнопки D описаны в элементах `button_info` (1), (2), (3), (4) соответственно.

Момент времени осуществления управления посредством ICS-сегмента совпадает с моментом времени отображения данных (`pt1`) произвольного кинокадра из кинофильма, показанного на Фиг.45, и это означает, что интерактивное экранное изображение `tm1`, составленное из кнопки А - кнопки D, будет отображаться будучи вставленным (`gs1`) в эти данные (`pt1`) кинокадра (`gs2`). Поскольку интерактивное экранное изображение образовано из множества кнопок в соответствии с содержимым кинофильма, то с использованием кнопок и при помощи ICS-сегмента становится возможным визуализировать весьма реалистичные изображения.

Фиг.46 иллюстрирует описательный пример ICS-сегмента в случае, при котором состояние кнопки А - кнопки D меняется так, как показано на Фиг.47. Стрелки `hh1` и `hh2`, показанные на Фиг.47, представляют в символическом виде изменение, порождаемое `neighbor_info()` (информацией о соседних кнопках) из состава `button_info(1)` (информации (1) о кнопке). Информация `neighbor_info()` из состава `button_info(1)` имеет в своем составе элемент `lower_button_number` (номер нижней кнопки), куда занесена кнопка С. Следовательно, если в то время, пока кнопка А находится в выделенном состоянии, производится операция пользователя, заключающаяся в нажатии клавиши `MoveDown` («Перемещение вниз») (Фиг.47, `up1`), то в выделенном состоянии будет кнопка С (Фиг.47, `sj1`). Поскольку значение `right_button_number` (номер правой кнопки) в `neighbor_info()` из состава `button_info(1)` установлено как кнопка В, то если в то время, пока кнопка А находится в выделенном состоянии, производится операция пользователя, заключающаяся в нажатии клавиши `MoveRight` («Перемещение вправо») (Фиг.47, `up2`), то в выделенном состоянии будет кнопка В (Фиг.47, `sj2`).

Стрелка `hh3` на Фиг.47 показывает операцию управления для изменения состояния информации `button_info(3)`, при помощи информации `neighbor_info()`. Поскольку значение `upper_button_number` (номер верхней кнопки) в `neighbor_info()` из состава `button_info(3)` установлено как кнопка А, то если, в то время, пока кнопка С находится в выделенном состоянии, производится операция пользователя, заключающаяся в нажатии клавиши `MoveUp` («Перемещение вверх»), а кнопка А возвратится в выделенное состояние.

Далее рассматриваются изображения для кнопки А - кнопки D. При этом предполагается, что сегменты ODS11, 21, 31 и 41 имеют изображения, проиллюстрированные на Фиг.49 и т.п. Поскольку в ICS-сегменте информация `normal_state_info()` (информация о нормальном состоянии) из состава информации `button_info(1)` имеет идентификаторы `start_object_id_normal` (идентификатор начального объекта в нормальном состоянии) и `end_object_id_normal` (идентификатор конечного объекта в нормальном состоянии), которые указывают сегменты ODS11-13, то нормальное состояние для кнопки А представлено как анимация ODS11-13. Кроме того, поскольку информация `selected_state_info()` (информация о выделенном состоянии) из состава информации `button_info(1)` имеет идентификаторы `start_object_id_selected` (идентификатор начального объекта в выделенном состоянии) и `end_object_id_selected` (идентификатор конечного объекта в выделенном состоянии), которые указывают сегменты ODS14-16, то выделенное

состояние для кнопки А представлено как сегменты ODS14-16. В результате этого при установке кнопки А пользователем в выделенное состояние портрет, представляющий собой изображение на кнопке А, изменится с того, что относится к сегментам ODS11-13 на тот, что относится к сегментам ODS14-16. При этом, если
 5 выполнена настройка, при которой «repeat_normal_flag» (признак повторения в нормальном состоянии) и «repeat_selected_flag» (признак повторения в выделенном состоянии) установлены в 1 в информации normal_state_info() и selected_state_info() соответственно, то анимация ODS11-13 и анимация ODS14-16 будет отображаться
 10 многократно и непрерывно, что показано знаками «->(A)», «(A)->», «->(B)» и «(B)->» на чертеже.

Если кнопке А - кнопке D присвоено множество ODS-сегментов, которые способны визуализировать анимацию, и соответствующие средства управления описаны в ICS-сегменте, то управление состоянием кнопки может быть реализовано
 15 более изощренно и быстро (например, изменяя при изменении операции пользователя выражение лица персонажа на изображении).

Далее рассматривается порядок расположения ODS-сегмента в Отображаемом наборе. Как было упомянуто выше, ODS-сегмент, принадлежащий к
 20 Отображаемому набору, указывается ICS-сегментом для представления одного состояния кнопки. Порядок расположения ODS-сегмента в Отображаемом наборе определяется в соответствии с указанием на то, какое состояние кнопки он должен представлять.

Если описать это более конкретно, то ODS-сегменты в Отображаемом наборе сгруппированы в соответствии с тем, какое состояние они представляют: (1) как представляющие нормальное состояние, (2) как представляющие выделенное состояние, (3) как представляющие активное состояние и т.п. Каждая из таких групп, которая представляет одно состояние кнопки, именуется «группа состояния
 25 кнопок». Затем эти группы состояний кнопок располагаются в порядке, таком как «нормальное состояние -> выделенное состояние -> активное состояние». Определение этого порядка расположения ODS-сегмента соответственно определит и порядок расположения ODS-сегмента в Отображаемом наборе.

Фиг.50 иллюстрирует порядок расположения ODS-сегментов, принадлежащих к
 35 Отображаемому набору. Во втором ряду этой схемы показаны три группы состояний кнопок этого Отображаемого набора. Этот чертеж показывает три набора ODS-сегментов: набор ODS-сегментов, который визуализирует нормальное состояние (ODS-сегменты для Нормального состояния); набор ODS-сегментов, который визуализирует выделенное состояние (ODS-сегменты для Выделенного состояния); и набор ODS-сегментов, который визуализирует активное состояние (ODS-сегменты для Активного состояния). Эти группы состояний кнопок
 40 расположены в порядке «нормальное состояние -> выделенное состояние -> активное состояние». Цель этого порядка расположения заключается в том, чтобы считывать те компоненты, которые составляют первоначальное отображение интерактивного экранного изображения ранее считывания других компонентов, которые составляют экранное отображение после обновления.

Первый ряд на Фиг.50 показывает Графические объекты «An, Bn, Cn, Dn, As, Bs, Cs, Ds, Aa, Ba, Ca, Da». Индекс «n», присвоенный An, Bn, Cn, Dn, представляет
 50 нормальное состояние соответствующей кнопки. Аналогичным образом, индекс «s», присвоенный As, Bs, Cs, Ds, представляет выделенное состояние соответствующей кнопки, а индекс «a», присвоенный Aa, Ba, Ca, Da, представляет активное состояние

соответствующей кнопки. Второй ряд на Фиг.50 показывает группы состояния кнопок, к которым принадлежат Графические объекты из первого ряда. Следует отметить, что на этом рисунке каждому набору сегментов ODS1-ODSn присвоен одинаковый номер, такой как 1 и n. Однако эти наборы отличаются друг от друга, и принадлежат к N-ODS-сегментам, S-ODS-сегментам и A-ODS-сегментам соответственно. Это также относится и к каждому аналогичному чертежу, следующему ниже.

Фиг.51 иллюстрирует изменения состояния интерактивного экранного изображения, на котором расположены группы состояния кнопок, показанные на Фиг.50. Интерактивное экранное изображение на этом чертеже имеет множество состояний, которыми являются: «первоначальное отображение», «обновленное отображение в соответствии с первым действием пользователя» и «обновленное отображение в соответствии со вторым действием пользователя». Стрелки на этом чертеже представляют действия пользователя, которые запускают соответствующую смену состояния. В соответствии с этим чертежом четыре кнопки A, B, C и D имеют «нормальное состояние», «выделенное состояние», «активное состояние» соответственно. Можно понять, что при выполнении первоначального отображения необходимы Графические объекты, предназначенные для визуализации трех нормальных состояний, и Графический объект, предназначенный для визуализации одного выделенного состояния.

Даже в том случае, когда кнопка, выделяемая по умолчанию, не определена, и то, какая кнопка подлежит установке в выделенное состояние, изменяется динамическим образом, первоначальное отображение будет реализовано, как только завершается декодирование для Графических объектов, представляющих нормальное состояние и выделенное состояние для каждой кнопки. Учитывая это, данный вариант осуществления изобретения располагает группы состояния кнопок, каждая из которых соответствует одному из состояний, отличному от других, в порядке «нормальное состояние -> выделенное состояние -> активное состояние», что показано во втором ряду Фиг.50. Такое расположение реализует первоначальное отображение даже в случае, когда считывание и декодирование ODS-сегментов, составляющих активное состояние, не завершены, и помогает сократить период, начинающийся с начала считывания Отображаемого набора и заканчивающегося завершением первоначального отображения.

Далее, ниже рассматривается то, в каком порядке расположены ODS-сегменты, проиллюстрированные на Фиг.48 и 49. Фиг.52 иллюстрирует порядок расположения ODS-сегментов в отображаемом наборе. На этом чертеже ODS-сегменты для Нормального состояния составляют сегменты ODS11-13, ODS21-23, ODS31-33 и ODS41-43. ODS-сегменты для Выделенного состояния составляют сегменты ODS14-16, ODS24-26, ODS34-36 и ODS44-46, и ODS-сегменты для Активированного состояния составляют сегменты ODS17-19, ODS27-29, ODS37-39 и ODS47-49. Сегменты ODS11-13 предназначены для визуализации изменения в выражении лица персонажа, показанного на Фиг.49. То же самое относится к сегментам ODS21-23, ODS31-33 и ODS41-43. Следовательно, располагая эти ODS-сегменты в верхней группе состояния кнопок, становится возможным организовать подготовку первоначального отображения даже в середине считывания Отображаемого набора. В соответствии с этим интерактивное экранное изображение, содержащее анимацию, может быть осуществлено без задержки.

Далее рассматривается порядок расположения ODS-сегментов, на которые

осуществляются множественные ссылки со стороны множества кнопок. При этом наличие множественных ссылок означает, что object_id (идентификатор объекта) ODS-сегмента указан в двух или более информациях normal_state_info() (информация о нормальном состоянии), selected_state_info() (информация о выделенном состоянии) 5 activated_state_info() (информация об активированном состоянии). Благодаря использованию такого способа осуществления множественных ссылок выделенное состояние кнопки может быть визуализировано с использованием Графического объекта, предназначенного для визуализации нормального состояния другой 10 кнопки. Это позволяет осуществлять совместное использование изображения Графического объекта. Такое совместное использование помогает сократить количество ODS-сегментов. В этом случае проблема, касающаяся ODS-сегмента, используемого при осуществлении множественных ссылок, состоит в том, к какой группе состояния кнопок принадлежит этот ODS-сегмент.

15 Если описать это более конкретно, то в случае, когда нормальное состояние одной кнопки и выделенное состояние другой кнопки визуализируются посредством одного ODS-сегмента, необходимо принять решение на предмет того, принадлежит ли этот ODS-сегмент к группе состояния кнопок, соответствующей нормальному состоянию, или к группе состояния кнопок, соответствующей выделенному 20 состоянию.

В этом случае ODS-сегмент размещается только один раз в той группе состояния кнопок, которая соответствует состоянию, появляющемуся раньше других.

25 Если некоторый ODS-сегмент используется при осуществлении множественных ссылок, как в нормальном состоянии, так и в выделенном состоянии, то этот ODS-сегмент будет расположен в группе состояния кнопок, соответствующей нормальному состоянию (N-ODS-сегменты), а не в группе состояния кнопок, соответствующей выделенному состоянию (S-ODS-сегменты). Кроме того, если 30 другой ODS-сегмент используется при осуществлении множественных ссылок, как в выделенном состоянии, так и в активном состоянии, то этот ODS-сегмент будет расположен в группе состояния кнопок, соответствующей выделенному состоянию (S-ODS-сегменты), а не в группе состояния кнопок, соответствующей активному состоянию (A-ODS-сегменты). В общем, ODS-сегменты при таком 35 способе осуществления множественных ссылок будут расположены только один раз в той группе состояния кнопок, которая соответствует состоянию, появляющемуся раньше других.

40 Далее рассматривается порядок расположения ODS-сегментов в группе S-ODS-сегментов. В S-ODS-сегментах то, какой ODS-сегмент следует первым, зависит от того, определена ли кнопка, выделяемая по умолчанию, статически, или она определяется динамически. Кнопка, выделяемая по умолчанию, определяемая статически, имеет действительное (то есть существующее на самом деле) значение, которое установлено (за исключением значения 00) в 45 элементе default_selected_button_number (номер кнопки, выделяемой по умолчанию) в ICS-сегменте, и это значение определяет кнопку. В случае, когда default_selected_button_number указывает действительное значение, и при условии, что в N-ODS-сегментах нет ODS-сегмента, представляющего эту кнопку, 50 выделяемую по умолчанию, ODS-сегмент, представляющий эту кнопку, выделяемую по умолчанию, будет расположен первым.

Если default_selected_button_number указывает значение 00, то кнопка, устанавливаемая в выделенное состояние по умолчанию, будет изменяться

динамически в соответствии с состоянием, сложившимся со стороны устройства воспроизведения.

Случай, при котором `default_selected_button_number` должен быть установлен на указание значения 0, представляет собой, например, ситуацию, при которой
 5 поток AVClip, в котором был мультиплексирован Отображаемый набор, служит точкой соединения для множества дорожек воспроизведения. Если, например, предыдущие множественные дорожки воспроизведения представляют собой соответственно первый, второй и третий разделы, и Отображаемый набор, являясь
 10 точкой соединения, предназначен для отображения кнопок, соответствующих первому, второму и третьему разделам, было бы неадекватным определять в качестве умолчания кнопку, которая должна быть в выделенном состоянии, в элементе `default_selected_button_number`.

В таком случае идеальным было бы изменять кнопку, которая должна быть в
 15 выделенном состоянии, в соответствии с тем, какая дорожка из предыдущих множественных дорожек воспроизведения воспроизводилась на текущий момент, до достижения этого Отображаемого набора (например, делая такой кнопкой кнопку второго раздела при достижении этого набора из первого раздела, кнопку третьего
 20 раздела при достижении этого набора из второго раздела и кнопку четвертого раздела при достижении этого набора из третьего раздела). В случае, когда кнопка, которая должна быть в выделенном состоянии, будет меняться, `default_selected_button_number` (номер кнопки, выделяемой по умолчанию) будет задан как недействительный (не существующий на самом деле), то есть ему присваивается
 25 значение 0. Поскольку кнопка, которая должна быть в выделенном состоянии, будет меняться, то предусматривать расположение определенного ODS-сегмента в начале группы состояния кнопок не требуется.

Фиг.53 показывает различие в расположении ODS-сегментов в группе S-ODS-сегментов между случаем, когда `default_selected_button_number`=0 (номер кнопки, выделяемой по умолчанию = 0), и случаем, когда он « = кнопка В ». На этом чертеже пунктирная линия ss1 показывает расположение ODS-сегментов в группе S-ODS-сегментов в случае, при котором `default_selected_button_number` (номер кнопки
 30 выделяемой по умолчанию) указывает кнопку В; а пунктирная линия ss2 показывает расположение ODS-сегментов в группе S-ODS-сегментов в случае, при котором `default_selected_button_number` указывает значение = 0. Как показывает запись на этом чертеже, в случае, когда `default_selected_button_number` указывает кнопку В, первыми среди S-ODS-сегментов расположены сегменты ODSB, показывающие выделенное состояние кнопки В, а ODS-сегменты других кнопок
 40 расположены после них. В противоположность этому, в случае, когда `default_selected_button_number` указывает значение 0, первыми расположены сегменты ODSA, показывающие выделенное состояние кнопки А. Таким образом, то, является ли `default_selected_button_number` (номер кнопки, выделяемой по
 45 умолчанию) действительным (существующем на самом деле) или нет, заставляет порядок расположения S-ODS-сегментов сильно меняться.

Далее, ниже рассматривается то, каким образом Отображаемый набор, имеющий эти ICS-сегмент и ODS-сегменты, расположен на временной шкале для
 50 потока AVClip. DTS-метка и PTS-метка ODS-сегмента могут быть установлены на основе выражений, показанных в первом варианте осуществления изобретения. В противоположность этому DTS-метка и PTS-метка ICS-сегмента будут отличаться от тех, что показаны в первом варианте осуществления изобретения. Ниже

рассматриваются значения DTS-метки и PTS-метки ICS-сегмента.

В случае нахождения непосредственно после начала Сверхкадра PTS-метка в ICS-сегменте будет установлена как значение, равное или большее, чем значение, полученное в результате суммирования (1) значения PTS-метки ODS-сегмента, у которого время декодирования является последним среди ODS-сегментов, составляющих первоначальное отображение набора DS_n, (2) времени, необходимого для того, чтобы очистить Графическую матрицу, и (3) времени записи для записи Графического объекта, полученного посредством декодирования ODS-сегмента в Графическую матрицу. С другой стороны, в случае нахождения в точке захвата она будет установлена как значение, равное или большее, чем значение, полученное путем сложения (3) времени записи матрицы с (1) значением PTS-метки ODS-сегмента.

В случае, когда в ICS-сегменте указан default_selected_button_number (номер кнопки, выделяемой по умолчанию), первоначальное отображение может быть осуществлено за время, требуемое для (i) декодирования ODS-сегментов для визуализации нормального состояния всех кнопок и (ii) декодирования ODS-сегментов для визуализации выделенного состояния кнопки, выделяемой по умолчанию. ODS-сегменты для визуализации выделенного состояния множественных кнопок при первоначальном отображении именуются S-ODS-сегменты, а ODS-сегмент, у которого время декодирования наступает первым среди ODS-сегментов (в данном случае ODS-сегмент для визуализации кнопки, выделяемой по умолчанию), именуется первым из S-ODS-сегментов. Значение PTS-метки этого первого из S-ODS-сегментов устанавливается как значение PTS-метки ODS-сегмента, у которого время декодирования наступает последним, и используется как опорное значение PTS-метки в ICS-сегменте.

В случае, когда default_selected_button_number (номер кнопки, выделяемой по умолчанию) не указан в ICS-сегменте, в выделенном состоянии может быть любая кнопка. Следовательно, подготовка к первоначальному отображению не будет полной без подготовки к визуализации нормального состояния и выделенного состояния для всех кнопок. Среди S-ODS-сегментов, предназначенных для визуализации выделенного состояния множественных кнопок при первоначальном отображении, тот сегмент, у которого время декодирования наступает последним, именуется последним из S-ODS-сегментов. Значение PTS-метки для этого последнего из S-ODS-сегментов устанавливается как значение PTS-метки ODS-сегмента, у которого время декодирования наступает последним, и используется как опорное значение PTS-метки в ICS-сегменте.

Если время окончания для декодирования первого из S-ODS-сегментов принять как PTS(DS_n[первый из S-ODS-сегментов]), то PTS(DS_n[ICS]) будет представлять собой значение, полученное в результате прибавления к значению PTS(DS_n[первый из S-ODS-сегментов]) следующих значений: (2) времени, необходимого для очистки Графической матрицы, и (3) времени записи для записи Графического объекта, полученного в результате декодирования ODS-сегмента в Графическую матрицу.

Предположим при этом, что в пределах Графической матрицы ширина и высота прямоугольной области для визуализации кинокадра определяются соответственно как «video-width» («ширина видеоизображения») и «video-height» («высота видеоизображения»), и что скорость записи в Графическую матрицу составляет 128 мегабит в секунду. Тогда время, необходимое для очистки Графической матрицы, выражается как «8*video_width*video_height//128000000». При выражении этого значения с точностью указания времени, равной 90 кГц, время (2) очистки

Графической матрицы составит $90000 \cdot (8 \cdot \text{video_width} \cdot \text{video_height} // 128000000)$.

Кроме того, предположим при этом, что суммарный размер Графических объектов, указанных всей информацией о кнопках, включенной в состав ICS-сегмента, представляет собой $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$, и что скорость записи в Графическую матрицу составляет 128 мегабит в секунду, тогда время, необходимое для записи в Графическую матрицу, выражается как $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) // 128000000$. Если выразить это значение с точностью указания времени, равной 90 кГц, то время (2) очистки Графической матрицы составит $90000 \cdot (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) // 128000000)$.

При этом $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ представляет собой суммарный размер Графического объекта, отображаемого первым, из числа Графических объектов, представляющих все кнопки. Это выражение $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ будет давать отличное значение в случае, при котором кнопка, выделяемая по умолчанию, была уже определена, от значения, в случае, при котором кнопка, выделяемая по умолчанию, меняется динамически. В ситуации, когда кнопка, выделяемая по умолчанию, была статически определена, значение $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ будет представлять собой сумму 1) ODS-сегмента, отображаемый первым среди множества ODS-сегментов, предназначенных для представления выделенного состояния кнопки, выделяемой по умолчанию, и 2) ODS-сегмента, отображаемый первым среди множества ODS-сегментов, предназначенных для представления нормального состояния кнопок за исключением кнопки, выделяемой по умолчанию.

В противоположность этому в ситуации, когда кнопка, выделяемая по умолчанию, меняется динамически, следует принять тот случай, при котором время записи является самым длинным, поскольку трудно представить, какая кнопка будет кнопкой, выделяемой по умолчанию. В этом случае предполагается, что графический объект, отображаемый первым, представляет собой Графический объект, имеющий самый большой размер ($\text{Max}(\text{ODSn1}, \text{ODSs1})$), из числа 1) Графических объектов, представляющих первую страницу в нормальном состоянии произвольной кнопки x (ODSn1), и 2) Графических объектов, представляющих первую страницу в выделенном состоянии этой кнопки x (ODSs1).

Результат суммирования этого значения ($\text{Max}(\text{ODSn1}, \text{ODSs1})$) по каждой кнопке составит $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$.

Фиг.54А, 54В иллюстрируют то, какие значения принимает выражение $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ в случае, при котором группа N-ODS-сегментов включает в себя множество ODS-сегментов, составляющих кнопки A-D, и при котором группа S-ODS-сегментов включает в себя множество ODS-сегментов, составляющих кнопки A-D. При этом в случае, когда `default_selected_button_number` (номер кнопки, выделяемой по умолчанию) указывает действительное (то есть существующее на самом деле) значение, величина $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ будет равна суммарному размеру для четырех ODS-сегментов, показанных рамкой, выполненной жирной линией. «As1» представляет собой ODS-сегмент, отображаемый первым среди множества ODS-сегментов, представляющих выделенное состояние кнопки A. «Bn1», «Cn1» и «Dn1» представляют соответствующие ODS-сегменты, отображаемыми первыми среди множества ODS-сегментов, представляющих нормальные состояния кнопки B-кнопки D. Если обозначить эти размены, как `size()`, то $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ составит:

`size(As1)+size(Bn1)+size(Cn1)+size(Dn1)`.

С другой стороны, в случае, когда `default_selected_button_number` является «=0»,

$\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ составит:

Наибольший ODS-сегмент из числа сегментов $\text{An1}, \text{As1}$ + Наибольший ODS-сегмент из числа сегментов $\text{Bn1}, \text{Bs1}$ + Наибольший ODS-сегмент из числа сегментов $\text{Cn1}, \text{Cs1}$ + Наибольший ODS-сегмент из числа сегментов $\text{Dn1}, \text{Ds1}$.

5 Следовательно, $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ выражается следующим образом:

$\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$

$= \max(\text{size}(\text{Cn1}), \text{size}(\text{Cs1})) + \max(\text{size}(\text{Dn1}), \text{size}(\text{Ds1}))$

10 Используя вышеприведенное выражение, $\text{PTS}(\text{DSn}[\text{ICS}])$ непосредственно после начальной точки Начала Сверхкадра выражается следующим образом:

$\text{PTS}(\text{DSn}[\text{ICS}]) \geq \text{PTS}(\text{DSn}[\text{первый из S-ODS-сегментов}])$

$+ 90000 * (8 * \text{video_width} * \text{video_height} / 128000000)$

$+ 90000 * (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) / 128000000)$

15 На Фиг.55 показан пример реализации синхронного отображения, осуществляемой за счет установки PTS-метки и DTS-метки в соответствии с вышеизложенным. Этот чертеж предполагает случай, при котором кнопка отображается в момент времени отображения данных (py1) какого-либо кинокадра в кинофильме. В таком случае значение PTS-метки в ICS-сегменте должно быть
20 установлено таким образом, чтобы совпадать с моментом времени отображения соответствующих данных кинокадра.

Кроме того, значение PTS-метки в ODS-сегменте должно быть установлено на момент времени (1), показанный на этом чертеже, поскольку к этому моменту времени, получаемому вычитанием из значения PTS-метки в ICS-сегменте периода
25 «cd1» очистки экрана и периода «td1» передачи Графического объекта, декодирование ODS-сегмента, у которого время декодирования наступает последним среди ODS-сегментов, составляющих первоначальное отображение набора DSn , должно быть завершено. Помимо этого, поскольку декодирование ODS-сегмента занимает период dd1 , то DTS-метка ODS-сегмента должна быть
30 установлена ранее этой PTS-метки на значение периода dd1 .

На Фиг.55 имеется только один ODS-сегмент, подлежащий объединению с кинофильмом, что представляет собой упрощенный пример. Для того чтобы реализовать первоначальное отображение интерактивного экранного изображения,
35 объединяемого с кинофильмом, для множества ODS-сегментов, PTS-метка и DTS-метка в ICS-сегменте и PTS-метка, DTS-метка в ODS-сегменте должны быть установлены так, как показано на Фиг.56.

Фиг.56 показывает то, каким образом DTS-метка и PTS-метка устанавливаются в
40 случае, при котором первоначальное отображение интерактивного экранного изображения составляется множеством ODS-сегментов, и при котором $\text{default_selected_button}$ (выбираемая по умолчанию кнопка) является статически определенной. Если декодирование для первого из S-ODS-сегментов, у которого декодирование выполняется последним среди ODS-сегментов,
45 необходимых для реализации первоначального отображения, закончится во время периода dd1 , показанного на этом чертеже, то метка $\text{PTS}(\text{DSn}[\text{первый из S-ODS-сегментов}])$ этого первого из S-ODS-сегментов должна быть установлена таким образом, чтобы указывать время периода dd1 .

50 Кроме того, перед первоначальным отображением должны быть выполнены очистка экрана и передача уже декодированных Графических объектов. Следовательно, метка $\text{PTS}(\text{DSn}[\text{ICS}])$ ICS-сегмента должна быть установлена таким образом, чтобы быть после времени, полученного прибавлением к значению этой

метки PTS(DSn[первый из S-ODS-сегментов]) периода, требующегося для очистки экрана, $(90000 * (8 * \text{video_width} * \text{video_height} // 128000000))$ и периода передачи данных декодированного Графического объекта $(90000 * (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])) // 128000000))$.

Фиг.57 показывает то, каким образом DTS-метка и PTS-метка устанавливаются в случае, при котором первоначальное отображение интерактивного экранного изображения составляется множеством ODS-сегментов, и при котором default_selected_button (выбираемая по умолчанию кнопка) не определена.

Если декодирование для последнего из S-ODS-сегментов, у которого декодирование выполняется последним среди ODS-сегментов, необходимых для реализации первоначального отображения, закончится во время периода dd2, показанного на этом чертеже, то метка PTS(DSn[последний из S-ODS-сегментов]) этого последнего из S-ODS-сегментов должна быть установлена таким образом, чтобы указывать время периода dd2.

Кроме того, перед первоначальным отображением должны быть выполнены очистка экрана и передача уже декодированных Графических объектов. Следовательно, метка PTS(DSn[ICS]) ICS-сегмента должна быть установлена таким образом, чтобы быть после времени, полученного прибавлением к значению этой метки PTS(DSn[последний из S-ODS-сегментов]) периода, требующегося для очистки экрана, $(90000 * (8 * \text{video_width} * \text{video_height} // 128000000))$ и периода передачи данных декодированного Графического объекта $(90000 * (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])) // 128000000))$.

Здесь следует отметить, что управление синхронизацией посредством PTS-метки в ICS-сегменте, описанное выше, включает в себя не только управление отображением кнопки в некоторый момент времени на временной шкале воспроизведения, но также включает в себя управление, обеспечивающее отображение Всплывающего меню во время некоторого периода на временной шкале воспроизведения. Всплывающее меню представляет собой меню, отображаемое в виде всплывающего изображения посредством нажатия, направленного на клавишу «меню», предусмотренную на пульте (400) дистанционного управления. Управление синхронизацией посредством PTS-метки в ICS-сегменте также включает в себя обеспечение отображения этого Всплывающего меню в момент времени отображения данных некоторого кинокадра в потоке AVClip. ODS-сегменты, составляющие это Всплывающее меню, декодируются в первую очередь и затем записываются в Графическую матрицу, точно так же, как ODS-сегменты, составляющие кнопку. До тех пор, пока запись в Графическую матрицу не завершена, невозможно ответить на вызов меню, осуществляемый пользователем. Ввиду вышесказанного время, в которое отображение Всплывающего меню становится возможным при синхронном отображении Всплывающего меню, записывается в PTS-метку в ICS-сегменте.

После вышеприведенного разъяснения, касающегося носителя записи по настоящему изобретению, ниже рассматривается устройство воспроизведения в соответствии с настоящим изобретением. Внутренняя структура устройства воспроизведения в соответствии со вторым вариантом осуществления изобретения по существу одинакова с внутренней структурой устройства воспроизведения в соответствии с первым вариантом осуществления изобретения, за исключением некоторых усовершенствований, вносимых в Буфер (15) объектов и в Графический контроллер (17). По этой причине ниже подробно рассматриваются эти

усовершенствования, вносимые в Буфер (15) объектов и в Графический контроллер (17).

Графические объекты, получаемые посредством декодирования, выполняемого Процессором (14) потоковых графических данных, и предназначенные для
 5 составления интерактивного экранного изображения, размещаются в Буфере (15) объектов, соответствующем второму варианту осуществления изобретения. Фиг.58 иллюстрирует содержимое Буфера (15) объектов в сопоставлении с Графической матрицей. Содержимое Буфера (15) объектов предполагает случай, при
 10 котором ODS-сегменты, показанные на Фиг.48 и Фиг.49, записаны в Буфер (15) объектов. Примеры, показанные на Фиг.48 и Фиг.49, реализуют анимацию для четырех кнопок, осуществляемую посредством 36 ODS-сегментов (ODS11-ODS49), причем ODS-сегменты, представляющие все кадры этой анимации, хранятся в этом Буфере (15) объектов, и место отображения каждого из ODS-сегментов, хранящихся
 15 в этом Буфере (15) объектов, определено на Графической матрице (8). Это место отображения определено значениями `Button_horizontal_position` (местоположение кнопки по горизонтали) и `Button_vertical_position` (местоположение кнопки по вертикали) из состава информации о соответствующей кнопке. Анимация
 20 реализуется посредством осуществления записи множественных ODS-сегментов, хранящихся в Буфере (15) объектов, в соответствующее место отображения на Графической матрице (8) по одному кадру одновременно.

Графический контроллер (17) по второму варианту осуществления изобретения интерпретирует блок, в котором расположен ICS-сегмент, - Буфер (16) композиции, и
 25 осуществляет управление на основе этого ICS-сегмента. Момент времени осуществления этого управления основан на значении PTS-метки, присвоенной этому ICS-сегменту. Важной задачей этого Графического контроллера (17) является операция записи во время первоначального отображения интерактивного экранного
 30 изображения и во время его обновления. Ниже со ссылкой на Фиг.59 приводится описание операции записи во время первоначального отображения интерактивного экранного изображения и во время его обновления. Фиг.59 иллюстрирует операцию, выполняемую Графическим контроллером во время первоначального отображения. Как показывает эта схема, Графический контроллер (17) выполняет управление
 35 таким образом, что ODS-сегменты, принадлежащие к S-ODS-сегментам кнопки А, записываются в место отображения, определенное значениями `Button_horizontal_position` (местоположение кнопки по горизонтали) и `Button_vertical_position` (местоположение кнопки по вертикали) из состава
 40 информации о кнопке по кнопке А; и аналогичным образом ODS-сегменты, принадлежащие к N-ODS-сегментам кнопок В, С, D, записываются в соответствующие места отображения, определенные соответствующими значениями `Button_horizontal_position` и `Button_vertical_position` из состава информации
 45 о кнопке по кнопкам В, С и D. Отметим здесь, что стрелки w1, w2, w3 и w4 в символическом виде показывают вышеупомянутое осуществление записи. Благодаря выполнению этой записи будет выполнено первоначальное отображение, показанное на Фиг.51. При этом следует особо отметить, что для реализации первоначального отображения интерактивного экранного изображения необходимы
 50 не все ODS-сегменты, и при условии, что Буфер (15) объектов содержит ODS-сегменты, принадлежащие к S-ODS-сегментам кнопки, выделяемой по умолчанию, и ODS-сегменты, принадлежащие к N-ODS-сегментам других кнопок, этого достаточно для завершения первоначального отображения интерактивного

экранного изображения. Следовательно, когда ODS-сегменты, принадлежащие к S-ODS-сегментам кнопки, выделяемой по умолчанию, и ODS-сегменты, принадлежащие к N-ODS-сегментам других кнопок, декодированы, можно сказать, что Графический контроллер (17) готов начать выполнение записи для

Фиг.60 иллюстрирует операцию, выполняемую Графическим контроллером (17) при выполнении обновления интерактивного экранного изображения в соответствии с первым действием пользователя (командой MoveRight («Перемещение вправо»)).

Как показывает этот чертеж, Графический контроллер (17) выполняет управление таким образом, что ODS-сегменты, принадлежащие к S-ODS-сегментам кнопки В, записываются в место отображения, определенное значениями Button_horizontal_position (местоположение кнопки по горизонтали) и Button_vertical_position (местоположение кнопки по вертикали) из состава информации о кнопке по кнопке В; и аналогичным образом ODS-сегменты, принадлежащие к N-ODS-сегментам кнопки А, записываются в место отображения, определенное значениями Button_horizontal_position и Button_vertical_position из состава информации о кнопке по кнопке А. Отметим здесь, что стрелки w5, w6, w7 и w8 в символическом виде показывают вышеупомянутое осуществление записи. Благодаря выполнению этой записи будет реализована смена состояний, показанная на Фиг.51. Кнопки С и D пребывают в нормальном состоянии так же, как во время первоначального отображения, но запись в Графическую матрицу непрерывно выполняется и для них, так чтобы продолжать анимацию.

Аналогично тому, что сказано выше, Фиг.61 и 62 иллюстрируют операции, выполняемые Графическим контроллером (17) при обновлении интерактивного экранного изображения, в случаях, когда первым действием пользователя являются команды MoveDown («Перемещение вниз») и Activated («Активировано»). Во время обновления интерактивного экранного изображения S-ODS-сегменты и A-ODS-сегменты кнопок, отличных от кнопки, выделяемой по умолчанию, и таким образом, желательно, чтобы в Буфере (15) объектов были сохранены все ODS-сегменты.

В устройстве воспроизведения, сконструированном так, как описано выше, каждый компонент выполняет операции декодирования способом конвейерной обработки данных так же, как в первом варианте осуществления изобретения.

Фиг.63 представляет собой временную диаграмму, иллюстрирующую конвейерную обработку данных, выполняемую данным устройством воспроизведения. Четвертый ряд показывает Отображаемый набор с диска BD-ROM, а третий ряд показывает периоды считывания ICS-сегмента, PDS-сегмента, ODS-сегментов в Буфер (13) закодированных данных. Второй ряд показывает периоды декодирования ODS-сегментов, где декодирование выполняется Процессором (14) потоковых графических данных. Первый ряд показывает периоды операций Графического контроллера (17). Моменты времени начала декодирования для ODS-сегментов показаны метками DTS11, DTS12 и DTS13 соответственно. Сохранение в Буфере (13) закодированных данных первого ODS-сегмента (N-ODS-сегменты[ODS1])

из числа ODS-сегментов, принадлежащих к N-ODS-сегментам, будет завершено к моменту DTS11. Сохранение в Буфере (13) закодированных данных последнего ODS-сегмента (N-ODS-сегменты[ODSn]) из числа ODS-сегментов, принадлежащих к N-ODS-сегментам, будет завершено к моменту DTS12. По существу, каждый ODS-сегмент будет считан в Буфер (13) закодированных данных к моменту времени, показанному его собственной DTS-меткой.

С другой стороны, момент времени окончания декодирования каждого ODS-сегмента показан метками PTS11, PTS12 и PTS13, приведенными на чертеже. Декодирование сегмента (N-ODS-сегменты[ODS1]), выполняемое Процессором (14) потоковых графических данных, будет завершено к моменту PTS11; а декодирование (N-ODS-сегменты[ODSn]) будет завершено к моменту PTS12. По существу, к моменту времени, показанному DTS-меткой каждого ODS-сегмента, этот ODS-сегмент был считан в Буфер (13) закодированных данных, и каждый ODS-сегмент, считанный в Буфер (13) закодированных данных, будет декодирован и записан в Буфер (15) объектов к моменту времени, показанному соответствующей PTS-меткой. Процессор (14) потоковых графических данных выполняет эту последовательность операций способом конвейерной обработки данных.

В случае, когда кнопка, выделяемая по умолчанию, определяется статически, все графические объекты, необходимые для первоначального отображения интерактивного экранного изображения, будут готовы в Буфере (15) объектов тогда, когда декодирование завершено для 1) группы состояния кнопок, соответствующей нормальному состоянию, и 2) первому ODS-сегменту группы состояния кнопок, соответствующей выделенному состоянию. На этом чертеже в момент времени, показанный меткой PTS13, все графические объекты, необходимые для первоначального отображения интерактивного экранного изображения, готовы.

На этом чертеже период cd1 в первом ряду представляет собой период, необходимый для очистки Графической матрицы (8). Кроме того, период td1 представляет собой период, необходимый для записи в Графическую матрицу (8) Графических объектов, составляющих первую страницу интерактивного экранного изображения, которые находятся среди Графических объектов, полученных в Буфер (15) объектов. Точное место сохранения Графических объектов на Графической матрице (8) представляет собой место, показанное значениями button_horizontal_position (местоположение кнопки по горизонтали) и button_vertical_position (местоположение кнопки по вертикали). Иначе говоря, cd1 (период очистки экрана), td1 (период записи декодированных Графических объектов) прибавляются к PTS13 ODS-сегмента, и в полученный период на Графической матрице (8) будут получены разуплотненные графические данные, составляющие интерактивное экранное изображение. Затем посредством 1) использования блока (9) CLUT для преобразования цвета разуплотненных графических данных и 2) использования блока (10) суммирования для объединения с этими данными разуплотненных данных кинокадра, хранящихся в видеоматрице (6), будет получено составное изображение.

В противоположность случаю, при котором первоначальное отображение выполняется после декодирования всех ODS-сегментов, включенных в состав Отображаемого набора, становится возможным в вышеизложенном случае выполнять первоначальное отображение независимо от того, было ли завершено декодирование группы состояния кнопок, соответствующей выделенному состоянию, или от того, было ли завершено декодирование группы состояния кнопок, соответствующей активному состоянию. Следовательно, первоначальное отображение будет в этом случае выполнено раньше на период hu1, показанный на чертеже.

Следует отметить, что на этом чертеже каждому набору сегментов ODS1-ODSn присвоен одинаковый номер, такой как 1 и n. Однако эти наборы отличаются друг

от друга и принадлежат к N-ODS-сегментам, S-ODS-сегментам и A-ODS-сегментам соответственно. Это также относится и к каждому аналогичному чертежу, следующему ниже.

В Декодере (12) графических данных даже в то время, пока Графический контроллер (17) продолжает исполнять очистку Графической матрицы (8), Процессор (14) потоковых графических данных продолжает выполнять декодирование (период декодирования ODS_n, период декодирования ODS₁ и период n декодирования ODS_n во втором ряду). Следовательно, становится возможным закончить декодирование ODS-сегментов, отличных от тех, которые обрабатывает Графический контроллер (17), раньше, чем традиционным способом, поскольку эти отличные ODS-сегменты будут декодироваться одновременно с декодированием ODS-сегментов, которые обрабатывает Графический контроллер (17). Поскольку становится возможным рано осуществлять подготовку к обновлению интерактивного экранного изображения посредством завершения декодирования этих отличных ODS-сегментов, то обновление интерактивного экранного изображения, которое будет использовать эти отличные ODS-сегменты, будет соответственно закончено раньше, чем традиционным способом. Вышеупомянутая конвейерная обработка данных позволяет выполнять как первоначальное отображение интерактивного экранного изображения, так и его обновление, без задержки.

Фиг.63 предполагает случай, при котором кнопка, выделяемая по умолчанию, была определена статически. В противоположность этому Фиг.64 представляет собой временную диаграмму, иллюстрирующую конвейерную обработку данных, выполняемую данным устройством воспроизведения в случае, при котором кнопка, выбираемая по умолчанию, изменяется динамически. В случае, когда кнопка, выбираемая по умолчанию, изменяется динамически, Графические объекты, необходимые для первоначального отображения, будут готовы тогда, когда все ODS-сегменты, принадлежащие к группе состояния кнопок, были декодированы, и эти Графические объекты получены в Графической матрице. В противоположность случаю, при котором первоначальное отображение выполняется после декодирования всех ODS-сегментов, включенных в состав группы состояния кнопок, соответствующей активному состоянию, вышеизложенный случай делает возможным выполнение первоначального отображения независимо от того, было ли завершено декодирование группы состояния кнопок, соответствующей активному состоянию. Следовательно, первоначальное отображение будет в этом случае выполнено раньше на период hy_2 , показанный на чертеже.

Фиг.65 представляет собой временную диаграмму, иллюстрирующую хронологические изменения в заполнении Графической матрицы (8), Буфера (15) объектов, Буфера (13) закодированных данных и Буфера (16) композиции. Обозначения заполнения, использованные на этом чертеже, согласуются с теми, что использованы на Фиг.30. Поскольку во втором варианте осуществления изобретения должны быть декодированы ODS-сегменты, составляющие N-ODS-сегменты, S-ODS-сегменты, A-ODS-сегменты, то количество участков простого возрастания и участков простого убывания больше, чем на Фиг.30. За исключением этого различия Фиг.65 является такой же, как Фиг.30. Также как в первом варианте осуществления изобретения, графики для примера, показанного на Фиг.65, проиллюстрированы с использованием: DTS-метки и PTS-метки, присвоенных ODS-сегменту; DTS-метки и PTS-метки, присвоенных ICS-сегменту, емкостей и скоростей передачи данных

каждого буфера, проиллюстрированного на Фиг.27. Кроме того, строя такие графики, пользователи на стадии авторского процесса могут узнать, каким образом 5 меняется состояние каждого буфера. Поскольку изменение состояния каждого буфера может быть отрегулировано посредством корректировки DTS-метки и PTS-метки, становится возможным также и в этом варианте осуществления изобретения избегать возникновения такой нагрузки по декодированию, которая превысила бы 10 технические характеристики декодера, находящегося на стороне устройства воспроизведения, и избегать возникающего при воспроизведении переполнения буфера. В соответствии с этим на стадии разработки устройства воспроизведения реализация его аппаратного/программного обеспечения станет проще.

Далее рассматриваются усовершенствования программного обеспечения, необходимые для реализации устройства воспроизведения по второму варианту осуществления изобретения.

15 Фиг.66 представляет собой блок-схему алгоритма, показывающую процесс операции загрузки функционального сегмента. Этот чертеж составлен на основе блок-схемы алгоритма, показанной на Фиг.31. Различие заключается в том, что после этапа S29 на Фиг.66 добавлены этапы S36 и S67.

20 Этап S36 предназначен для определения того, равен ли 1 `command_update_flag` (признак обновления команды). Если он равен 1 (этап S36: ответ «Да»), то в Буфер (13) закодированных данных загружается только команда кнопки из состава информации о кнопке, а другие элементы игнорируются (этап S37). Если он равен 0, то управление передается на этап S22, и тем самым игнорируется ICS-сегмент, 25 представляющий Точку захвата (этап S24).

Далее, предположив случай, при котором мультиплексирование выполняется так, как показано на Фиг.67, рассмотрим ниже то, каким образом осуществляется считывание DS-набора. В примере, приведенном на Фиг.67, три DS-набора 30 мультиплексированы с кинофильмом. Среди этих трех DS-наборов первый набор DS1 имеет в качестве `Composition_state` (состояния композиции) значение `Epoch_Start` (Начало Сверхкадра), включает в себя команду кнопки, именуемую `LinkPL (PL#5)`, и его `Command_update_flag` (признак обновления команды) установлен в 0.

35 Набор DS10 представляет собой «Дубликат» набора DS1 и имеет в качестве `Composition_state` (состояния композиции) значение `Acquisition Point` (Точка захвата), включает в себя команду кнопки, именуемую `LinkPL (PL#5)`, и его `Command_update_flag` (признак обновления команды) установлен в 0.

40 Набор DS20 представляет собой «Наследника» набора DS1 и имеет в качестве `Composition_state` (состояния композиции) значение `Acquisition Point` (Точка захвата). Отличие от набора DS1 заключается в команде кнопки (`LinkPL (PL#10)`), и, таким образом, для представления этого факта его `Command_update_flag` (признак обновления команды) установлен в 1.

Предположим здесь, что эти три потока и кинофильм мультиплексированы в 45 поток `AVClip`, и выполняется операция пропуска до данных кинокадра `pt10`, показанная стрелкой `ms1`. В этом случае набор DS10, который является ближайшим к целевой точке операции пропуска, составляет цель на Фиг.66. На этапе S27 `composition_state` (состояния композиции) будет определено как Точка захвата, но в 50 Декодере (12) графических данных не имеется предыдущего DS-набора. Следовательно, признак игнорирования устанавливается в 0, и этот набор DS10 загружается в Буфер (13) закодированных данных, входящий в состав устройства воспроизведения (`hs1` на Фиг.68). С другой стороны, в случае, когда целевая точка

операции пропуска попадает за то место, в котором существует набор DS10 (стрелка ms2), в Буфер (13) закодированных данных должен быть считан набор DS20.

Фиг.70 показывает загрузку наборов DS1, DS10 и DS20 при нормальном воспроизведении, выполняемом так, как показано на Фиг.69. Среди этих трех DS-наборов в Буфер (13) закодированных данных в полном составе загружается набор DS1, у которого composition_state (состояние композиции) ICS-сегмента представляет собой «Начало Сверхкадра» (этап S23). Однако набор DS10, у которого composition_state (состояние композиции) ICS-сегмента представляет собой «Точку захвата», имеет признак игнорирования, равный 1 (этап S29). Следовательно, функциональные сегменты, составляющие набор DS10, не будут загружены в Буфер (13) закодированных данных, а вместо этого будут проигнорированы (этап S24). Кроме того, что касается набора DS20, то истинно то, что его Composition_state (состояние композиции) ICS-сегмента «Точка захвата», но его Command_update_flag (признак обновления команды) установлен в 1. Следовательно, этап S36 дает ответ «Да», и таким образом, загружается только его команда кнопки, и только команда кнопки из ICS-сегмента DS-набора, расположенного в Буфере (13) закодированных данных, заменяется на команду кнопки из набора DS20 (этап S37). Однако признак игнорирования по-прежнему представляет 1, и, следовательно, другие элементы, отличные от этой команды кнопки, не будут загружены, а вместо этого будут проигнорированы.

При достижении набора DS20 отображаемое содержимое остается тем же самым, однако команда кнопки изменилась с LinkPL(#5) DS-набора на команду LinkPL(#19). Такая замена команды кнопки позволяет управлять содержимым команды кнопки. Далее рассматривается обработка данных, выполняемая Графическим контроллером. Фиг.71 представляет собой блок-схему алгоритма, иллюстрирующую основную программу обработки данных, выполняемую Графическим контроллером (17). На этой блок-схеме многократно выполняются следующие три операции: операция синхронизации по временной метке (этап S35), операция отображения анимации (этап S36) и операция «Операция Пользователя» (UO) (этап S37).

Далее рассматриваются обработка данных, выполняемая Графическим контроллером (17). Обработка данных, выполняемая Графическим контроллером (17), значительно изменилась: с той, что показана на Фиг.36-38, на ту, что показана на Фиг.71-78. Фиг.71 представляет собой блок-схему алгоритма, иллюстрирующую основную программу обработки данных, выполняемую Графическим контроллером (17). Фиг.72 представляет собой блок-схему алгоритма, иллюстрирующую обработку данных для реализации управления синхронизацией, использующего временную метку. На этой блок-схеме производится определение того, соблюдается ли какое-либо из условий, показанных на этапах S41, S43-S47. Если какое-либо из этих условий соблюдается, то выполняется соответствующая операция, и управление возвращается к основной программе. Описанная же обработка данных представляет собой подпрограмму.

Этап S41 предназначен для определения того, является ли текущий момент времени воспроизведения моментом времени из числа: момента времени, представленного PTS-меткой первого из S-ODS-сегментов, и момента времени, представленного PTS-меткой последнего из S-ODS-сегментов. Если определено, что текущий момент времени воспроизведения является одним из названных моментов времени, то вычисляется его период α . Период α получается посредством

суммирования (2) периода, необходимого для очистки Графической матрицы, и (1) периода, необходимого для осуществления записи Графического объекта, полученного посредством декодирования ODS-сегмента, в Графическую матрицу.

На этапе S42 Графический контроллер (17) обращается к Composition_state (состоянию композиции) в ICS-сегменте, и а) если Composition_state представляет собой «Начало Сверхкадра», то он устанавливает значение α , равное «период (2) очистки матрицы + период (3) записи в матрицу»; б) если же Composition_state представляет собой «Точку захвата», то он устанавливает значение α , равное периоду (3) записи в матрицу. Вычисление периода (3) записи в матрицу выполняется следующим образом: если default_selected_button_number (номер кнопки, выделяемой по умолчанию) представляет собой действительное (то есть существующее на самом деле) значение, то используется способ вычисления, показанный на Фиг.54А; а если default_selected_button_number равен 0, то используется способ вычисления, показанный на Фиг.54В. Когда α вычислено, управление возвращается к циклической обработке данных.

Этап S43 предназначен для определения того, является ли текущий момент времени воспроизведения моментом времени, представленным значением PTS- α в ICS-сегменте. Если определение имеет своим результатом утвердительный ответ, то выполняется операция записи в Графическую матрицу (8) (этап S51), и управление возвращается к основной программе.

Этап S45 предназначен для определения того, равен ли текущий момент времени воспроизведения PTS-метке в ICS-сегменте. Если определение имеет своим результатом утвердительный ответ, то отдается команда на осуществление вывода содержимого Графической матрицы. Получателем этого содержимого является блок (9) CLUT. Блок (9) CLUT выполняет преобразование цвета для этого содержимого. Затем интерактивное экранное изображение объединяется с содержимым видеоматрицы (9). В результате этого выполняется первоначальное отображение (этап S52). Затем переменная «animation (p)» («анимация (p)») (где $p = 1, 2, 3, \dots, n$) устанавливается в 0 (этап S53), и управление возвращается в основную программу. При этом переменная animation (p) представляет собой глобальную переменную, указывающую то, какой номер кадра в последовательности кадров отображается в текущий момент, и используемую при исполнении анимационного отображения кнопки (b) (глобальная переменная - это переменная, которая действительна повсюду во множестве блок-схем алгоритмов). Следовательно, на этапе S53 кнопка (p) всех кнопок будет установлена в 0.

Этап S46 и этап S47 предназначены для определения того, достиг ли текущий момент времени воспроизведения значений информации о времени, описанной в ICS-сегменте.

Этап S46 предназначен для определения того, является ли текущий момент времени воспроизведения моментом времени, представленным значением selection_Timeout_pts (PTS-метка истечения времени выделенного состояния), и если определение имеет своим результатом утвердительный ответ, то выполняется операция по активации кнопки, представленной значением default_selected_button_number (номер кнопки, выделяемой по умолчанию), и управление возвращается в основную программу (этап S54).

Этап S47 предназначен для определения того, является ли текущий момент времени воспроизведения моментом времени, представленным значением composition_time_out_pts (PTS-метка истечения времени композиции), и

если определение имеет своим результатом утвердительный ответ, то производится очистка экрана, после чего управление возвращается в основную программу (этап S55). В ходе описанной выше операции синхронизации каждая из операций на этапе S51 и этапе S51 и на этапе S54 выполняется как подпрограмма. Далее со ссылкой на Фиг.73 рассматривается подпрограмма, исполняемая на этапе S51.

Фиг.73 представляет собой блок-схему алгоритма, иллюстрирующую операцию записи первоначального отображения меню в Графическую матрицу (8). Этап S64 предназначен для определения того, установлен ли элемент Composition_state (состояние композиции) в ICS-сегменте в значение «Начало Сверхкадра», и если это определение дает утвердительный ответ, то на этапе S65 производится очистка Графической матрицы, и выполняются операции этапов S66-S73. Период, необходимый для очистки Графической матрицы (8), представляет собой период cd1, показанный на Фиг.56 и Фиг.57. Если определение на этапе S64 дает отрицательный ответ, то этап S65 пропускается, и выполняются операции этапов S66-S73.

Этапы S66-S73 составляют циклическую обработку данных, которая должна быть повторена для каждой порции информации о кнопках из ICS-сегмента (этап S66, S67). Информация о кнопке, которая должна подвергнуться этой циклической обработке, именуется информация (p) о кнопке.

Этап S67 предназначен для определения того, указывает ли default_selected_button_number (номер кнопки, выделяемой по умолчанию) действительное (существующее на самом деле) или недействительное значение. Этап S68 предназначен для определения того, является ли button_info(p) информацией о кнопке, соответствующей кнопке, выделяемой по умолчанию, указываемой значением default_selected_button_number.

Если определение на этапе S68 дает отрицательный ответ, то в Буфере (15) объектов находят Графический объект, соответствующий значению start_object_id_normal (идентификатора начального объекта в нормальном состоянии), указанному в normal_state_info (информации о нормальном состоянии) из состава button_info(p), и идентифицируют этот объект как Графический объект(p) (этап S69).

Если определение на этапе S68 дает утвердительный ответ, то в Буфере (15) объектов находят Графический объект, соответствующий значению start_object_id_selected (идентификатора начального объекта в выделенном состоянии), указанному в selected_state_info (информации о выделенном состоянии) из состава button_info(p), и идентифицируют этот объект как Графический объект(p) (этап S70), после чего кнопка (p) делается текущей кнопкой (этап S71). Текущая кнопка представляет собой кнопку, которая была установлена в выделенное состояние на интерактивном экранном изображении, отображаемом в текущий момент времени. Устройство воспроизведения сохраняет идентификатор этой текущей кнопки как PSR(10).

После того, как в результате этапа S69 и этапа S70 был идентифицирован Графический объект (p), Графический объект (p) записывается в то место на Графической матрице (8), которое указано значениями button_horizontal_position (местоположение кнопки по горизонтали) и button_vertical_position (местоположение кнопки по вертикали) из состава button_info(p) (этап S72). В результате повторения вышеописанных операций для каждой порции информации о кнопках первый Графический объект, который находится среди множества графических объектов, каждый из которых показывает состояние соответствующей кнопки, должен быть

записан в Графическую матрицу (8). Период времени, требующийся для выполнения этой операции, которая направлена на Графический объект, необходимый для, по меньшей мере, первоначального отображения из Буфера (15) объектов, показан как период $td1$ на Фиг.56 и Фиг.57.

В случае, когда значение `default_selected_button_number` (номер кнопки, выделяемой по умолчанию) представляет собой «=0», что означает, что `default_selected_button_number` изменяется динамически, ответом на этапе S67 будет «Нет», и производится определение того, соответствует ли `button_info(p)` текущей кнопке. Если определение на этапе S67 имеет своим результатом утвердительный ответ, то управление переходит на этап S70; а если это определение имеет своим результатом отрицательный ответ, то управление передается на этап S69.

Далее со ссылкой на Фиг.74 рассматривается обработка данных в подпрограмме на этапе S54.

Фиг.74 представляет собой блок-схему алгоритма, иллюстрирующую обработку автоматической активации для кнопки, выделяемой по умолчанию. Сначала производится определение того, является ли значение `default_selected_button_number` (номер кнопки, выделяемой по умолчанию) равным 0 или FF (этап S75). Если определение на этапе S75 имеет своим результатом «00», то никакой обработки данных не производится, и управление возвращается в основную программу; а если определение на этапе S75 имеет своим результатом «FF», то текущая кнопка i переводится в активное состояние (этап S77), переменная `animation(i)` устанавливается в 0, и управление возвращается в основную программу (этап S78).

Если же определение на этапе S75 не имеет своим результатом ни «00», ни «FF», то в качестве текущей кнопки устанавливается кнопка, указанная как `default_activated_button_number` (номер кнопки, активируемой по умолчанию) (этап S76), текущая кнопка i переводится в активное состояние (этап S77), переменная `animation(i)`, соответствующая текущей кнопке i , устанавливается в 0, и управление возвращается в основную программу (этап S78).

Вышеописанная обработка данных делает возможным перевод кнопки в выделенном состоянии в активное состояние после заранее заданного времени.

Далее рассматривается анимация посредством меню. Фиг.75 представляет собой блок-схему алгоритма, показывающего обработку данных при отображении анимации.

При этом первоначальное отображение реализуется посредством осуществления записи графического объекта в Графическую матрицу, где Графический объект указан 1) значением `start_object_id_normal` (идентификатора начального объекта в нормальном состоянии) из `normal_state_info` (информации о нормальном состоянии) и 2) значением `start_object_id_selected` (идентификатора начального объекта в выделенном состоянии) из `selected_state_info` (информации о выделенном состоянии) для каждой `button_info` (информации о кнопке). При этом «анимация» представляет собой обработку данных, имеющую целью обновление Графической матрицы данными произвольного кадра для каждой кнопки (то есть данными Графического объекта q -го кадра), всякий раз, когда завершается цикл циклической обработки данных, проводимой на этапе S35 - этапе S37. Это обновление выполняется посредством возвращения в основную программу, посредством осуществления записи Графических объектов, указанных в `normal_state_info` и `selected_state_info` из `button_info`, одного за другим в Графическую матрицу (8). При этом переменная q

используется для идентификации каждого Графического объекта, указанного в `normal_state_info` и `selected_state_info` из `button_info`, для каждой порции информации о кнопках.

Обработка данных, осуществляемая для реализации этого отображения анимации, подробно описывается со ссылкой на Фиг.75. В целях упрощения описания в этой блок-схеме алгоритма предполагается случай, при котором признаки `repeat_normal_flag` (признак повторения в нормальном состоянии) и `repeat_selected_flag` (признак повторения в выделенном состоянии) из ICS-сегмента установлены таким образом, чтобы указывать «повторение необходимо».

Этап S80 предназначен для определения того, закончено ли первоначальное отображение. Если определение на этапе S80 имеет своим результатом отрицательный ответ, то осуществляется возврат управления без выполнения какой бы то ни было обработки данных; если же определение на этапе S80 имеет своим результатом утвердительный ответ, то выполняются этап S81 - этап S93. Этап S81 - этап S93 составляют циклическую обработку данных, заключающуюся в повторении этапа S81 - этапа S93 для каждой информации `button_info` в ICS-сегменте (этап S81, этап S82).

Этап S83 предназначен для присвоения значения переменной `animation(p)`, соответствующей информации `button_info(p)`, переменной `q`. Благодаря выполнению этого этапа переменная `q` будет указывать текущий номер кадров, который соответствует информации `button_info(p)`.

Этап S84 предназначен для определения того, соответствует ли информация `button_info(p)` кнопке, находящейся в текущий момент времени в выделенном состоянии (далее именуемой «текущей кнопкой»).

Если определено, что `button_info(p)` отлична от текущей кнопки, то производится определение на этапе S86.

Этап S86 предназначен для определения того, находится ли текущая кнопка в активном состоянии, и если дан утвердительный ответ, то идентификатору `ID(q)` присваивается значение идентификатора, полученного путем сложения переменной `q` с идентификатором `start_object_id_actioned` (идентификатором начального объекта в активированном состоянии), содержащимся в информации `button_info(p).actioned_state_info`. После этого исполняется одна команда кнопки из числа тех команд, что включены в состав `button_info(p)` (этап S88).

Если определено, что текущая кнопка не находится в активном состоянии, то идентификатору `ID(q)` присваивается значение идентификатора, полученного путем сложения переменной `q` с идентификатором `start_object_id_selected` (идентификатором начального объекта в выделенном состоянии), содержащимся в информации `button_info(p).selected_state_info` (этап S89).

После того, как в результате вышеописанных операций определено значение `ID(q)`, Графический объект (`p`), имеющий идентификатор `ID(q)` и находящийся в Буфере (15) объектов, записывается в место на Графической матрице (8), указанное значениями `button_horizontal_position` (местоположение кнопки по горизонтали) и `button_vertical_position` (местоположение кнопки по вертикали) из состава информации `button_info(p)` (этап S90).

Посредством вышеописанной циклической обработки из числа множественных Графических объектов, которые составляют выделенное состояние (или активное состояние) текущей кнопки и нормальное состояние других кнопок, в Графическую матрицу записывается тот Графический объект, который соответствует `q`-й странице.

Этап S91 предназначен для определения того, достигла ли сумма `start_object_id_normal + q` значения `end_object_id_normal` (идентификатора конечного объекта в нормальном состоянии). Если определение на этапе S91 имеет своим результатом отрицательный ответ, то переменной «animation(p)»
 5 присваивается значение, полученное путем приращения переменной `q` на 1 (этап S92). Если определение на этап S91 имеет своим результатом утвердительный ответ, то производится инициализация переменной «animation(p)» с обращением ее значения в 0 (этап S93). Вышеописанные операции повторяются для
 10 всех `button_info(p)` в ICS-сегменте (этап S81, этап S82). Когда все `button_info` подвергнутся вышеописанным операциям, управление возвратится в основную программу.

Во время вышеописанных этапов S80 - этапа S93 всякий раз, когда основная программа (этап S35 - этап S37) выполняется один раз, изображение каждой кнопки
 15 на интерактивном экранном изображении будет обновлено на новый Графический объект. Это означает, что когда вышеупомянутая основная программа (этап S35 - этап S37) выполняется несколько раз, реализуется так называемая анимация. При анимации Графический контроллер (17) регулирует время таким образом, чтобы
 20 интервал времени отображения для одного кадра Графического объекта составлял значение, указанное кодом `animation_frame_rate_code` (кодом частоты кадра анимации).

Здесь следует отметить, что на этапе S88 команды кнопки, включенные в состав информации `button_info(p)`, исполняются по одной. Однако также возможно
 25 исполнять команды кнопки все вместе, после того, как была отображена последовательность Графических объектов, соответствующих активному состоянию. Далее со ссылкой на Фиг.76 рассматривается процесс операции «Операция пользователя», который выполняется на этапе S37 основной программы.

Фиг.76 представляет собой блок-схему алгоритма, иллюстрирующую процесс
 30 операции «Операция пользователя». На этой блок-схеме производится определение того, соблюдаются ли какое-либо из условий, показанных на этапе S100 - этапе S103. Если какое-либо из этих условий соблюдается, то выполняется соответствующая обработка, и затем управление переходит назад в основную программу. Этап S100
 35 предназначен для определения того, установлено ли в «1» значение `UOMaskTable`, и если это определение дает утвердительный ответ, то управление возвращается в основную программу без исполнения какой бы то ни было обработки данных.

Этап S101 предназначен для определения того, была ли нажата клавиша `MoveUp` («Перемещение вверх»)/`Down` («Вниз»)/`Left` («Влево»)/`Right` («Вправо»). Если это
 40 определение дает утвердительный ответ, то текущая кнопка меняется (этап S104), и затем производится определение того, установлен ли признак `auto_action_flag` (признак автоматического действия) в 01 (этап S108). Если определение на этапе S108 дает отрицательный ответ, то управление возвращается в основную программу.
 45 Если же определение на этапе S108 дает утвердительный ответ, то управление переводится на этап S105.

Этап S102 предназначен для определения того, нажата ли клавиша `Activated` («Активировано»). Если это определение дает утвердительный ответ, то текущая
 50 кнопка переводится в активное состояние (этап S105). После этого переменная «animation(i)» устанавливается в «0» (этап S106).

Этап S103 предназначен для определения того, осуществляется ли в данном случае ввод числового значения. Если это определение дает утвердительный ответ,

то выполняется соответствующая операция числового ввода (этап S107), и управление возвращается в основную программу. Среди процессов, показанных на Фиг.76, каждый этап из числа этапа S104 и этапа S107 представляет собой подпрограмму. Процессы этих подпрограмм показаны на Фиг.77 и Фиг.78. Ниже

Фиг.77 представляет собой блок-схему алгоритма, показывающую процесс операции смены текущей кнопки. Сначала среди номеров: `upper_button_number` (номер верхней кнопки), `lower_button_number` (номер нижней кнопки), `Left_button_number` (номер левой кнопки) и `right_button_number` (номер правой кнопки), которые принадлежат информации `neighbor_info` текущей кнопки, определяется номер, соответствующий нажатой клавише (этап S110).

Затем значение текущей кнопки присваивается «кнопке i», а значение кнопки, которая должна быть новой текущей кнопкой, присваивается «кнопке j» (этап S111). Этап S112 предназначен для определения того, соответствует ли кнопка j, значение которой присвоено на этапе S111, кнопке i. Если они соответствуют друг другу, то управление возвратится в основную программу без выполнения какой бы то ни было обработки данных. Если же они не соответствуют друг другу, то значение кнопки j присваивается текущей кнопке (этап S113), переменная «`animation(i)`» и переменная «`animation(j)`» устанавливаются в 0, и управление возвращается в основную программу (этап S114).

Фиг.78 представляет собой блок-схему алгоритма, показывающую процесс операции ввода числового значения. Здесь производится определение того, имеется ли информация `Button_info.j`, которая имеет `button_number` (номер кнопки), соответствующий введенному числовому значению (этап S121). Затем производится определение того, установлен ли признак `numerically_selectable_flag` (признак возможности выбора числового значения), относящийся к `Button_info.j`, в 1 (этап S122). Если ответами на этапе S121 и этапе S122 являются «Да», то текущая кнопка переводится в нормальное состояние, и значение кнопки j присваивается текущей кнопке (этап S123), и переменная «`animation(i)`» и переменная «`animation(j)`» устанавливаются в 0 (этап S124). После этих операций производится определение того, установлен ли признак `auto_action_flag` (признак автоматического действия), относящийся к `Button_info.j`, в 1 (этап S125). Если это определение дает отрицательный ответ, то управление возвращается в основную программу.

Если же это определение дает утвердительный ответ, то на этапе S126 текущая кнопка переводится в активное состояние, и управление возвращается в основную программу.

Если ответ на любом из этапов S121-S122 является «Нет», то управление возвратится в основную программу.

Графический контроллер (17) выполняет вышеописанные процессы для выполнения синхронного отображения. При этом следует иметь в виду, что при выполнении отображения интерактивного экранного изображения, запущенного операцией пользователя с использованием отображения Всплывающего меню или ему подобного, Процессор (14) потоковых графических данных и Графический контроллер (17) выполняют нижеследующие операции, которые являются теми же самыми операциями, что и выполняемые для выполнения синхронного отображения. При выполнении нижеследующих операций Графический объект получают в Графической матрице (8). После того, как Графический объект получен в соответствии с тем, что сказано выше, следует ожидание, продолжающееся до тех

пор, пока текущий момент времени воспроизведения не пройдет указанную PTS-метку, присвоенную ICS-сегменту. Затем, если после упомянутого времени контроллер (18) Операций пользователя принимает Операцию пользователя, указывающую вызов меню, то она будет выведена в блок (9) CLUT и отдаст блоку (9) CLUT команду представить Графический Объект, хранящийся в Графической матрице (8). Если такое осуществление вывода выполняется в синхронизации с Операцией пользователя, то будет реализовано отображение Всплывающего меню в соответствии с нажатием клавиши вызова меню.

Приведенное выше разъяснение описывает установку PTS-метки в ICS-сегменте и DTS-метки в ODS-сегменте и PTS-метки, которые принадлежат набору DS_n. Однако в нем не упоминаются DTS-метка в ICS-сегменте; DTS-метка и PTS-метка в PDS-сегменте; DTS-метка и PTS-метка в END-сегменте. В виду этого ниже рассматриваются временные метки, относящиеся к этим элементам. Поскольку во втором варианте осуществления изобретения WDS-сегмент не существует, то ICS-сегмент должен быть загружен в Буфер (16) композиции до наступления 1) момента времени начала декодирования первого PDS-сегмента (PDS₁) набора DS_n (то есть DTS(DS_n[ODS₁])), и 2) момента времени, в который первый PDS-сегмент (PDS₁) в наборе DS_n становится доступным (то есть PTS(DS_n[PDS₁])). Иначе говоря, следует установить значение, которое удовлетворяет следующему выражению:

$$DTS(DS_n[ICS]) \leq DTS(DS_n[ODS_1])$$

$$DTS(DS_n[ICS]) \leq PTS(DS_n[PDS_1])$$

Далее рассматривается установка DTS-метки и PTS-метки для каждого PDS-сегмента, принадлежащего набору DS_n.

Момент времени, когда каждый PDS-сегмент, принадлежащий набору DS_n, становится действующим в блоке (9) CLUT, находится в промежутке от 1) момента времени, в который ICS-сегмент загружается в Буфер (16) композиции, до 2) момента времени начала декодирования для первого ODS-сегмента (DTS(DS_n[ODS₁])). Ввиду этого значение PTS-метки каждого PDS-сегмента, принадлежащего набору DS_n, (то есть сегментов PDS₁-PDS_{последний}), должно быть установлено в значение, которое удовлетворяет следующим соотношениям.

$$DTS(DS_n[ICS]) \leq PTS(DS_n[PDS_1])$$

$$PTS(DS_n[PDS_j]) \leq PTS(DS_n[PDS_{j+1}]) \leq PTS(DS_n[PDS_{последний}])$$

$$PTS(DS_n[PDS_{последний}]) \leq DTS(DS_n[ODS_1])$$

Далее, рассматривается установка PTS-метки сегмента END («Конец Отображаемого набора») в наборе DS_n. END-сегмент, принадлежащий набору DS_n, показывает окончание набора DS_n. И, следовательно, это должен быть момент времени окончания декодирования последнего ODS-сегмента (ODS_{последнего}) из набора DS_n. Этот момент времени окончания декодирования указывается PTS-меткой (PTS(DS_n[ODS_{последний}])), и, таким образом, PTS-метка END-сегмента должна быть установлена в значение, указанное приводимым ниже выражением:

$$PTS(DS_n[END]) = (PTS(DS_n[ODS_{последний}]))$$

Что касается соотношения с ICS-сегментами, принадлежащими наборам DS_n и DS_{n+1}, то ICS-сегмент в наборе DS_n загружается в Буфер (16) композиции до наступления момента времени загрузки первого ODS-сегмента (то есть ODS₁).

Следовательно, PTS-метка END-сегмента должна быть 1) после момента времени загрузки ICS-сегмента, принадлежащего набору DS_n, и до 2) до момента времени загрузки ICS-сегмента, принадлежащего набору DS_{n+1} (то есть DTS(DS_{n+1}[ICS])). Соответственно PTS-метка в END-сегменте должна удовлетворять нижеследующему

соотношению:

$$DTS(DS_n[ICS]) \leq PTS(DS_n[END]) \leq DTS(DS_{n+1}[ICS])$$

С другой стороны, момент времени загрузки первого ODS-сегмента (то есть ODS1) PTS-метка END-сегмента (то есть PTS(DS_n[END])) должна быть после момента времени загрузки PDS-сегмента, принадлежащего набору DS_n. Соответственно, PTS-метка в END-сегменте должна удовлетворять нижеследующему соотношению:

$$PTS(DS_n[PDS_{\text{последний}}]) \leq PTS(DS_n[END]).$$

Поскольку ICS-сегмент, PDS-сегмент, ODS-сегмент, в которых

устанавливаются DTS-метка и PTS-метка, включаются в состав потока AVClip заранее, то становится удобно описывать интерактивное управление, имеющее своей целью заставлять устройство воспроизведения исполнять некоторую операцию в тот момент времени, когда на экране появляется кадр с некоторым киноизображением. Иначе говоря, вышеописанные средства удобны для описания интерактивного управления, синхронизированного с высокой точностью с содержимым кинофильма. Кроме того, ICS-сегмент, PDS-сегмент и ODS-сегмент мультиплексированы в сам поток AVClip. Следовательно, в случае, при котором секции, в отношении которых пользователь хотел бы выполнять управление воспроизведением, имеются в большом количестве, таком как несколько сотен, нет необходимости хранить в памяти все ICS-сегменты, PDS-сегменты и ODS-сегменты, которые соответствуют всем этим разделам. Поскольку ICS-сегменты, PDS-сегменты и ODS-сегменты подлежат считыванию с диска BD-ROM, то достаточно организовать работу с ними следующим способом. Этот способ заключается в том, что ICS-сегменты, PDS-сегменты и ODS-сегменты, которые соответствуют той секции кинофильма, которая подлежит воспроизведению в данный момент, остаются резидентными в памяти. После того, как воспроизведение для этой секции кинофильма закончено, соответствующие ICS-сегменты, PDS-сегменты и ODS-сегменты удаляются из памяти, и вместо них в памяти сохраняются ICS-сегменты, PDS-сегменты и ODS-сегменты, соответствующие последующей секции кинофильма. Поскольку ICS-сегменты, PDS-сегменты и ODS-сегменты подлежат мультиплексированию в поток AVClip, то даже в том случае, если количество ICS-сегментов, PDS-сегментов и ODS-сегментов достигает нескольких сотен, заполнение памяти может быть ограничено до минимального необходимого уровня.

Как было сказано выше, данный вариант осуществления изобретения предусматривает 360 страниц ODS-сегментов для того, чтобы реализовывать анимацию. Следовательно, в случае, когда относящийся к кнопкам материал сгруппирован по трем состояниям, ODS-сегменты будут сгруппированы по 120 страниц (то есть в три группы состояния кнопок). Группы состояния кнопок расположены таким образом, что группа, соответствующая состоянию, которое появляется раньше, размещается ближе к началу, чем группа, соответствующая состоянию, которое появляется позже. Благодаря этому при воспроизведении группа состояния кнопок, соответствующая состоянию, которое появляется раньше, загружается в устройство воспроизведения соответственно раньше, чем группа, соответствующая состоянию, которое появляется позже. В соответствии с этим, даже если декодирование всех 360 страниц ODS-сегментов и не завершено, но если завершено хотя бы приблизительно 1/2-1/3 общего количества ODS-сегментов, то, по меньшей мере, первоначальное отображение готово для выполнения. Поскольку операция первоначального отображения может быть начата при завершении приблизительно 1/2-1/3 общего количества ODS-сегментов, то даже в том случае,

если имеются многочисленные ODS-сегменты, которые необходимо считать и декодировать, первоначальное отображение не будет задержано. Следовательно, интерактивное экранное изображение исполняется плавно даже в том случае, если экранное изображение содержит для развлечения пользователей анимацию.

(Третий вариант осуществления изобретения)

Данный вариант осуществления изобретения относится к способу изготовления диска BD-ROM. Фиг.79 иллюстрирует способ изготовления PCS-сегмента, рассмотренного в первом варианте осуществления изобретения.

Способ изготовления диска BD-ROM включает в себя: этап S201 производства материала, в ходе которого, например, осуществляют съемку киноизображения и запись соответствующих аудиоданных; авторский этап S202, на котором генерируют прикладной формат; и этап S203 печати, на котором завершают изготовление диска BD-ROM, выполняя его печать/ламинирование.

Входящий в число этих этапов авторский этап, направленный на создание диска BD-ROM, включает в себя следующие этап S204 - этап S210.

На этапе S204 описывается управляющая информация, информация определения окна, информация определения палитры и графические данные. На этапе S205 управляющая информация, информация определения окна, информация определения палитры и графические данные соответственно преобразуются в функциональный сегмент. На этапе S206 в PCS-сегменте устанавливается PTS-метка в соответствии с тем, когда появляется кинокадр, подлежащий синхронному отображению. На этапе S207 в соответствии с этим значением PTS[PCS] устанавливаются метки DTS [ODS] и PTS[ODS]. На этапе S208 в соответствии со значением метки DTS[ODS] устанавливаются метки DTS[PCS], PTS[PDS], DTS[WDS] и PTS[WDS], а на этапе S209 в виде графика изображается хронологическое изменение в заполнении каждого буфера модели проигрывающего устройства. На этапе S210 производится определение того, удовлетворяет ли это хронологическое изменение, изображенное в виде графика, ограничению, накладываемому моделью проигрывающего устройства. Если определение на этапе S210 имеет своим результатом утвердительный ответ, то на этапе S212 создается поток графических данных и посредством мультиплексирования потока графических данных с потоком видеоданных и потоком аудиоданных, которые были созданы отдельно от потока графических данных, получают поток AVClip. Затем поток AVClip приводят в соответствие с форматом BD-ROM и тем самым завершают прикладной формат.

Вышеприведенное разъяснение сделано для способа изготовления носителя записи в соответствии с первым вариантом осуществления изобретения. Способ изготовления носителя записи в соответствии со вторым вариантом осуществления изобретения, показан на Фиг.80. На Фиг.80 этап S304 - этап S308 заменяют этап S204-S208, показанные на Фиг.79.

Ниже рассматриваются этап S304 - этап S308. На этапе S304 описывается управляющая информация, информация определения палитры и графические данные. На этапе S305 управляющая информация, информация определения палитры и графические данные соответственно преобразуются в функциональный сегмент. На этапе S306 в ICS-сегменте устанавливается PTS-метка в соответствии с тем, когда появляется кинокадр, подлежащий синхронному отображению. Затем на этапе S307 в соответствии с этим значением PTS[ICS] устанавливаются метки DTS[ODS] и PTS [ODS]. На этапе S308 в соответствии со значением метки DTS[ODS] устанавливаются метки DTS[ICS] и PTS[PDS].

(Замечание)

Нет необходимости напоминать, что вышеприведенное разъяснение не показывает все варианты осуществления и формы использования настоящего изобретения. Настоящее изобретение также реализуется в вариантах осуществления, в которые внесены нижеследующие изменения: (A), (B), (C), (D), ... и т.д. Следует иметь в виду, что пункты формулы изобретения по настоящему изобретению представляют собой расширенно толкуемые или обобщенные описания либо вышеописанных вариантов осуществления изобретения, либо измененных вариантов осуществления, основанных на нижеследующих изменениях. Степень этого расширенного толкования и обобщения отражает состояние данной области техники на момент подачи этой заявки.

(A) Во всех этих вариантах осуществления изобретения предполагается, что носитель записи в соответствии с настоящим изобретением представляет собой диск BD-ROM. Однако признаки носителя записи по настоящему изобретению относятся к потоку графических данных, хранящемуся на носителе записи, и эти признаки не зависят от физической природы диска BD-ROM. Иначе говоря, при реализации настоящего изобретения может быть использован любой носитель записи, который функционально пригоден для записи потока графических данных. Примеры включают в себя: оптический диск, такой как DVD-ROM (постоянное запоминающее устройство на универсальном цифровом диске), DVD-RAM (оперативное запоминающее устройство на универсальном цифровом диске), DVD-RW (универсальный цифровой диск формата (-) с возможностью многократной перезаписи), DVD-R (универсальный цифровой диск формата (-) для однократной записи), DVD+RW (универсальный цифровой диск формата (+) с возможностью многократной перезаписи), DVD+R (универсальный цифровой диск формата (+) для однократной записи), CD-R (компакт-диск формата (-) для однократной записи), CD-RW (компакт-диск формата (-) с возможностью многократной перезаписи); и оптический магнитный диск, такой как PD и MD. Примеры также включают в себя карту полупроводниковой памяти, такую как компактную карту флэш-памяти, интеллектуальная карта, карта memory stick, карта мультимедиа и карта PCM-CIA (карта по стандарту Международной ассоциации карт памяти персональных компьютеров). Более того, эти примеры включают в себя: диск для магнитной записи, такой как гибкий диск, SuperDisk, Click!; съемный накопитель на жестких магнитных дисках, такой как ORB, Jaz, SparQ, SyJet, EZFley и накопитель на микродисках. Эти примеры также включают в себя жесткий диск, встроенный в устройство.

(B) Во всех этих вариантах осуществления изобретения устройство воспроизведения декодирует поток AVClip, хранящийся на диске BD-ROM, до вывода его в телевизионный приемник. Однако возможна также конструкция, в которой устройство воспроизведения представляет собой просто дисковод для диска BD-ROM, а другие компоненты включены в состав телевизионного приемника. В этом случае устройство воспроизведения и телевизионный приемник могут быть соединены друг с другом в соответствии со стандартом IEEE 1394 (Стандарт Института инженеров по электротехнике и радиоэлектронике) и образовывать домашнюю сеть. Устройство воспроизведения по этим вариантам осуществления изобретения предназначено для использования его с подсоединенным к нему телевизионным приемником. Однако это устройство воспроизведения может быть выполнено как единое целое с устройством воспроизведения. Кроме того, в

устройстве воспроизведения по каждому из этих вариантов осуществления изобретения изобретением может признаваться только системная большая интегральная схема (интегральная схема), которая составляет сущность этой обработки данных. Устройство воспроизведения и интегральная схема оба
 5 представляют собой изобретения, описанные в настоящей спецификации, и, таким образом, акт изготовления устройства воспроизведения, имеющего любые из описанных форм и способов, основанных на внутренней структуре устройства воспроизведения по первому варианту осуществления изобретения, также
 10 представляет собой вариант осуществления настоящего изобретения. Любой акт передачи, как требующий оплаты, так и не требующий (продажа - в случае, если требуется оплата, и подарок - в случае, если оплата не требуется), аренда и импорт составляют вариант осуществления настоящего изобретения. Кроме того, любой акт по выполнению этих передачи и аренды посредством экспонирования в магазинах,
 15 предложения через каталоги и распространение рекламных буклетов также составляет вариант осуществления настоящего изобретения.

(C) Обработка данных, показанная на каждой блок-схеме алгоритма, конкретно реализована с использованием средств аппаратного обеспечения. Следовательно,
 20 любые программы, процессы которых показаны на блок-схемах алгоритмов, могут соответственно составлять независимое изобретение. Все варианты осуществления изобретения, относящиеся к программам, предполагают, что программа имеет форму, встроенную в соответствующее устройство воспроизведения. Однако сама программа, показанная в первом варианте осуществления изобретения, может
 25 составлять вариант осуществления изобретения, независимый от соответствующего устройства воспроизведения. Вариант осуществления изобретения, связанный с самой программой, включает в себя: (1) акт изготовления программы; (2) акт передачи программы, как требующий оплаты, так и нет; (3) акт аренды, (4) акт
 30 импорта; (5) акт предоставления публике через сеть интерактивных электронных коммуникаций; и (6) предложение передачи широкому пользователю посредством экспонирования в магазинах предложения через каталоги и распространение рекламных буклетов и тому подобное.

(D) Если концепция времени, присутствующая на каждом этапе, исполняемом
 35 хронологически на каждой блок-схеме алгоритма, признается обязательным признаком для определения настоящего изобретения, то каждый процесс на блок-схеме алгоритма толкуется как раскрывающий модель применения способа воспроизведения. Если процессы на вышерассмотренных блок-схемах алгоритмов
 40 исполняются посредством исполнения в хронологическом порядке каждого их этапов, таким образом, чтобы быть эффективными для достижения задачи настоящего изобретения и полезными в ее достижении, то это будет соответствовать варианту осуществления способу записи по настоящему изобретению.

(E) При записи на диск BD-ROM желательно, чтобы каждому TS-пакету (пакету
 45 транспортного потока), составляющему поток AVClip, был присвоен дополнительный заголовок. Этот дополнительный заголовок именуется «TP_extra_header» («Дополнительный заголовок транспортного пакета»), включает в себя «Arrival_Time_Stamp» («Временная метка поступления») и
 50 «copy_permission_indicator» («указатель разрешения на копирование») и имеет длину 4 байта. TS-пакеты, которым присвоены заголовки TP_extra_header, (именуемые TS-пакетом с присвоенным EX) делятся на группы, каждая из которых включает в себя 32 TS-пакета, и записываются в три сектора. Общий размер групп, каждая из

которых образована из 32 TS-пакетов с присвоенным EX, составляет 6144 байтов (= 32*192), что равно суммарному размеру трех секторов (6144 байтов = (2048*2)). 32 TS-пакета с присвоенным EX, хранящихся в одном секторе, именуются «Выровненный блок».

5 При использовании в домашней сети, соединенной в соответствии со стандартом IEEE1394, устройство воспроизведения выполняет передачу Выровненного блока, осуществляя нижеследующую обработку данных при передаче. Эта обработка заключается в том, что устройство-отправитель
10 удаляет TP_extra_header из каждого из 32 TS-пакетов с присвоенным EX, входящих в состав Выровненного блока, кодирует основное тело TS-пакетов и осуществляет их вывод. При осуществлении вывода TS-пакетов во многих местах между TS-пакетами вставляются изохронные пакеты. Точные места их вставки основаны на времени, показываемом меткой Arrival_Time_Stamp из заголовка TP_extra_header. В ответ на
15 осуществление вывода этих пакетов устройство воспроизведения выводит дескриптор DTCP_Descriptor. Дескриптор DTCP_Descriptor показывает настройку заголовка TP_extra_header в отношении разрешения/запрещения копирования. Следовательно, если дескриптор DTCP_Descriptor описан таким образом, чтобы
20 показывать «запрещение копирования», то TS-пакеты не будут записываться на другом устройстве во время их использования в домашней сети, соединенной в соответствии со стандартом IEEE1394.

(F) Поток цифровых данных в каждом из этих вариантов осуществления изобретения представляет собой поток AVClip в стандарте BD-ROM. Однако в
25 качестве альтернативы он может представлять собой VOB (Видеообъект) в стандарте DVD-Video (видеостандарте для цифровых универсальных дисков) или в стандарте DVD-Video Recording (стандарте видеозаписи для цифровых универсальных дисков). VOB представляет собой программный поток, соответствующий стандарту ISO_IEC13818-1 (являющемуся стандартом
30 Международной электротехнической комиссии Международной организации по стандартам), и получается посредством мультиплексирования потока видеоданных и потока аудиоданных. Поток видеоданных в потоке AVClip может в качестве альтернативы быть в формате MPEG4 или формате WMV. Кроме того, поток
35 аудиоданных может в качестве альтернативы быть в формате Dolby=AC3, формате MP3, формате MPEG4-AAC, формате dts.

(G) Кинематографическая продукция в этих вариантах осуществления изобретения может быть получена посредством кодирования аналоговых сигналов изображения, или потоковых данных, состоящих из транспортных потоков, транслируемых
40 посредством цифрового вещания. Кроме того, содержимое может быть получено посредством кодирования аналоговых/цифровых сигналов киноизображения, записанных на видеопленку, или могут представлять собой цифровую продукцию, распространя с сервера распространения.

45 (H) Графические объекты, показанные в первом и втором вариантах осуществления изобретения, представляют собой растровые данные, которые были закодированы способом группового кодирования. Причина, по которой в качестве способа уплотнения/кодирования Графических объектов используется способ
50 группового кодирования, заключается в том, что способ группового кодирования наилучшим образом подходит для уплотнения/разуплотнения данных в субтитрах. Субтитры характеризуются тем, что значение непрерывной длины одного пикселя в горизонтальном направлении является сравнительно большим. Следовательно, при

использовании уплотнения данных способом группового кодирования достигается высокая степень уплотнения. Кроме того, нагрузка, связанная с разуплотнением данных, невелика и, таким образом, приемлема для создания программного обеспечения для декодирующей обработки данных. В настоящем изобретении способ уплотнения/разуплотнения данных, используемый для субтитров, используется и для Графических объектов, так что один структурный компонент устройства, предназначенный для выполнения декодирования, используется совместно субтитрами и Графическими объектами. Однако использование способа группового кодирования не является обязательным признаком настоящего изобретения, и Графические объекты могут в качестве альтернативы представлять собой данные в формате PNG (Портативном сетевом графическом формате). Кроме того, вместо растровых данных могут быть векторные данные, или «прозрачные» видеоданные.

(I) Отображаемые эффекты, задаваемые PCS-сегментом, могут быть применены к графическим данным субтитров, выбранным в соответствии с языковой настройкой, имеющейся на стороне устройства. Благодаря этому отображаемый эффект, обычно реализуемый символами, представленными на современных дисках DVD (универсальных цифровых дисках) в самом основном «теле» кинофильма, может быть реализован с использованием графических данных субтитров, отображаемых в соответствии с языковой настройкой, имеющейся на стороне устройства. Это имеет большую практическую ценность.

(J) Отображаемые эффекты, задаваемые PCS-сегментом, могут быть применены к графическим данным субтитров, выбранным на стороне устройства в соответствии с настройкой отображения. Если описать это более конкретно, то на диск BD-ROM записаны графические данные для разнообразных режимов отображения, таких как широкоэкранное изображение, панорамированная развертка и рамка с текстом, и устройство воспроизведения выбирает один из них в соответствии с настройкой телевизионного приемника, с которым соединено это устройство, и осуществляет отображение выбранного типа графических данных. В данном случае отображаемый эффект применен к графическим данным субтитров, отображаемых вышеописанным способом. Следовательно, графика субтитров будет выглядеть лучше. Благодаря этому отображаемый эффект, обычно реализуемый символами, представленными на современных дисках DVD в самом основном «теле» кинофильма, может быть реализован с использованием отображений графических данных субтитров, соответствующих настройке отображения, имеющейся на стороне устройства. Это имеет большую практическую ценность.

(K) В первом варианте осуществления изобретения скорость записи R_c в Графическую матрицу определена таким образом, что размер окна составляет 25% полного размера, так что очистка Графической матрицы и ее повторная визуализация возможны в пределах одного видеокadra. Однако в качестве альтернативы, если принять, что время обратного хода по кадру составляет 25% от 1/29,93, то R_c составит 1 гигабит в секунду. Благодаря установке такого значения R_c отображение графических данных будет улучшено. Это имеет большую практическую ценность.

В дополнение к записи во время обратного хода по кадру может одновременно выполняться запись, синхронизированная с разверткой при записи. Благодаря этому даже в случае, если скорость записи составляет $R_c=256$ мегабит в секунду, отображение будет улучшено.

(L) В каждом из этих вариантов осуществления изобретения устройство воспроизведения оснащено Графической матрицей. Однако вместо Графической матрицы на устройстве воспроизведения может быть установлен буфер строки, который может хранить разуплотненные данные пикселей для одной строки.

Поскольку преобразование в сигнал изображения может выполняться для каждого горизонтального ряда (то есть строки), то если такой буфер строки предусмотрен, устройство воспроизведения способно выполнять преобразование в сигнал изображения.

(M) Субтитры, являющиеся графическими данными, описаны выше как последовательности символов, предназначенные для представления слов, произносимых в кинематографической продукции. Однако эти субтитры могут содержать сочетание рисунка, символа и цвета, образующих товарный знак. Кроме того, субтитры могут содержать разного рода государственную символику, официальные знаки, принятые в государствах для целей надзора и выдачи разрешений, знаки международных организаций, знаки, представляющие места происхождения конкретных товаров, и тому подобное.

(N) Первый вариант осуществления изобретения предполагает, что субтитры отображаются в верхней/нижней частях экрана, и поэтому окна определены в верхней/нижней частях Графической матрицы соответственно. Однако также возможно определять окна в правой/левой частях Графической матрицы. Это полезно при отображении субтитров на японском языке, расположенных в продольном направлении.

(O) Поток AVClip в каждом из этих вариантов осуществления изобретения составляет кинематографическую продукцию. Однако поток AVClip может быть предназначен для реализации «караоке» (аккомпанемент на заранее записанной пленке). В этом случае в ходе исполнения песни PCS-сегмент может, например, реализовывать эффект отображения, который изменяет цвет субтитров.

(P) В случае, где множественные дорожки воспроизведения соединяются одна с другой, и кнопка, выделяемая по умолчанию, меняется в зависимости от того, какая дорожка воспроизведения подлежит выбору, предпочтительным является следующая схема. Она заключается в том, что управление воспроизведением в «динамическом» сценарии описывается таким образом, что в момент, когда проходит каждая дорожка воспроизведения, собственное значение для этой дорожки воспроизведения устанавливается в регистре устройства воспроизведения, и в том, что процессы воспроизведения описаны таким образом, что кнопки будут устанавливаться в выделенное состояние в соответствии со значениями, установленными в этом регистре. По этой схеме кнопка, подлежащая установке в выделенное состояние, может быть изменена в соответствии с тем, какая дорожка воспроизведения должна проходиться.

Промышленная применимость

Носитель записи и устройство воспроизведения по настоящему изобретению реализуют отображение субтитров, имеющее отображаемый эффект, и интерактивное отображение, содержащее анимацию, и, таким образом, обеспечивают рынок кинематографической продукции с высокой добавленной стоимостью, что помогает насыщать рынок кинофильмов и рынок потребительских товаров. Соответственно носитель записи и устройство воспроизведения по настоящему изобретению являются весьма полезными в кинематографической промышленности и в промышленности потребительских товаров.

Формула изобретения

1. Устройство воспроизведения, содержащее блок получения, выполненный с
возможностью получения с носителя записи потока графических данных,
5 включающего в себя пакет данных и управляющий пакет, причем пакет данных
включает в себя графические данные и первую временную метку декодирования и
первую временную метку представления, при этом первая временная метка
декодирования указывает начальное время декодирования графических данных, а
10 первая временная метка представления указывает конечное время декодирования
графических данных, причем управляющий пакет включает в себя вторую
временную метку декодирования и вторую временную метку представления, при
этом вторая временная метка декодирования указывает начальное время очистки
графической матрицы, а вторая временная метка представления указывает время
15 отображения графических данных, которое является временем в момент или после
конечного времени декодирования графических данных; процессор, выполненный с
возможностью (i) декодировать графические данные во время первой временной
метки декодирования и (ii) заканчивать декодирование графических данных ко
20 времени первой временной метки представления, и контроллер, выполненный с
возможностью (i) записи декодированных графических данных в графическую
матрицу ко времени второго времени представления, причем графическая матрица
представляет собой область, в которой визуализируются декодированные
графические данные, и (ii) очистки графической матрицы со времени второй
25 временной метки декодирования.

2. Способ воспроизведения, содержащий этапы, на которых получают с носителя
записи поток графических данных, включающий в себя пакет данных и управляющий
пакет, причем пакет данных включает в себя графические данные и первую
30 временную метку декодирования и первую временную метку представления, при
этом первая временная метка декодирования указывает начальное время
декодирования графических данных, а первая временная метка представления
указывает конечное время декодирования графических данных, причем
управляющий пакет включает в себя вторую временную метку декодирования и
35 вторую временную метку представления, при этом вторая временная метка
декодирования указывает начальное время очистки графической матрицы, а вторая
временная метка представления указывает время отображения графических данных,
которое является временем в момент или после конечного времени декодирования
40 графических данных; начинают декодирование графических данных во время первой
временной метки декодирования; заканчивают декодирование графических данных
ко времени первой временной метки представления, и записывают декодированные
графические данные в графическую матрицу ко времени второго времени
представления, причем графическая матрица представляет собой область, в которой
45 визуализируются декодированные графические данные, и очищают графическую
матрицу со времени второй временной метки декодирования.

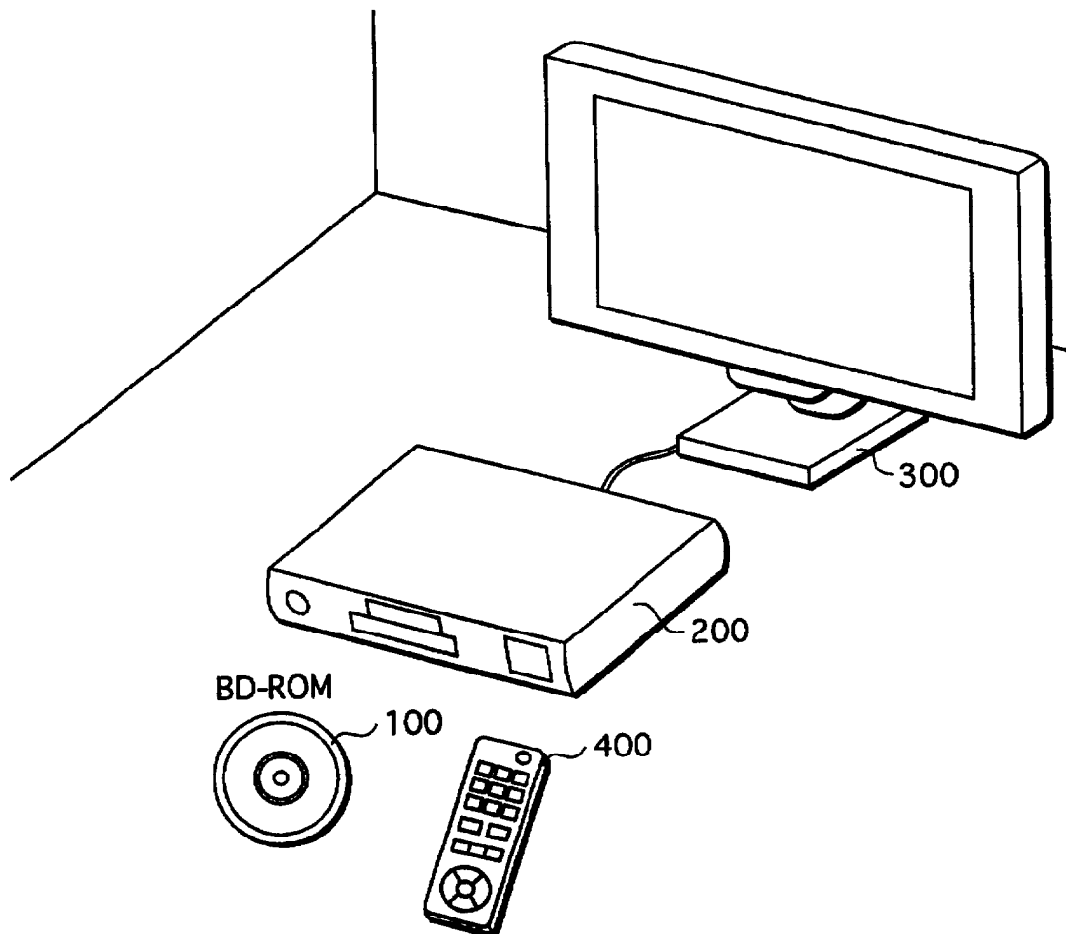
3. Носитель записи, содержащий записанный на нем поток графических данных,
причем поток графических данных включает в себя пакет данных и управляющий
50 пакет, пакет данных включает в себя графические данные, и первую временную
метку декодирования, и первую временную метку представления, первая временная
метка декодирования указывает начальное время декодирования графических
данных, и первая временная метка представления указывает конечное время

декодирования графических данных, при этом управляющий пакет включает в себя вторую временную метку декодирования и вторую временную метку представления, причем вторая временная метка декодирования указывает начальное время очистки графической матрицы, а вторая временная метка представления указывает время отображения графических данных, которое является временем в момент или после конечного времени декодирования графических данных, и при этом декодированные графические данные должны быть визуализированы в графической матрице ко времени второй временной метки представления, причем графическая матрица представляет собой область, в которой визуализируются декодированные графические данные.

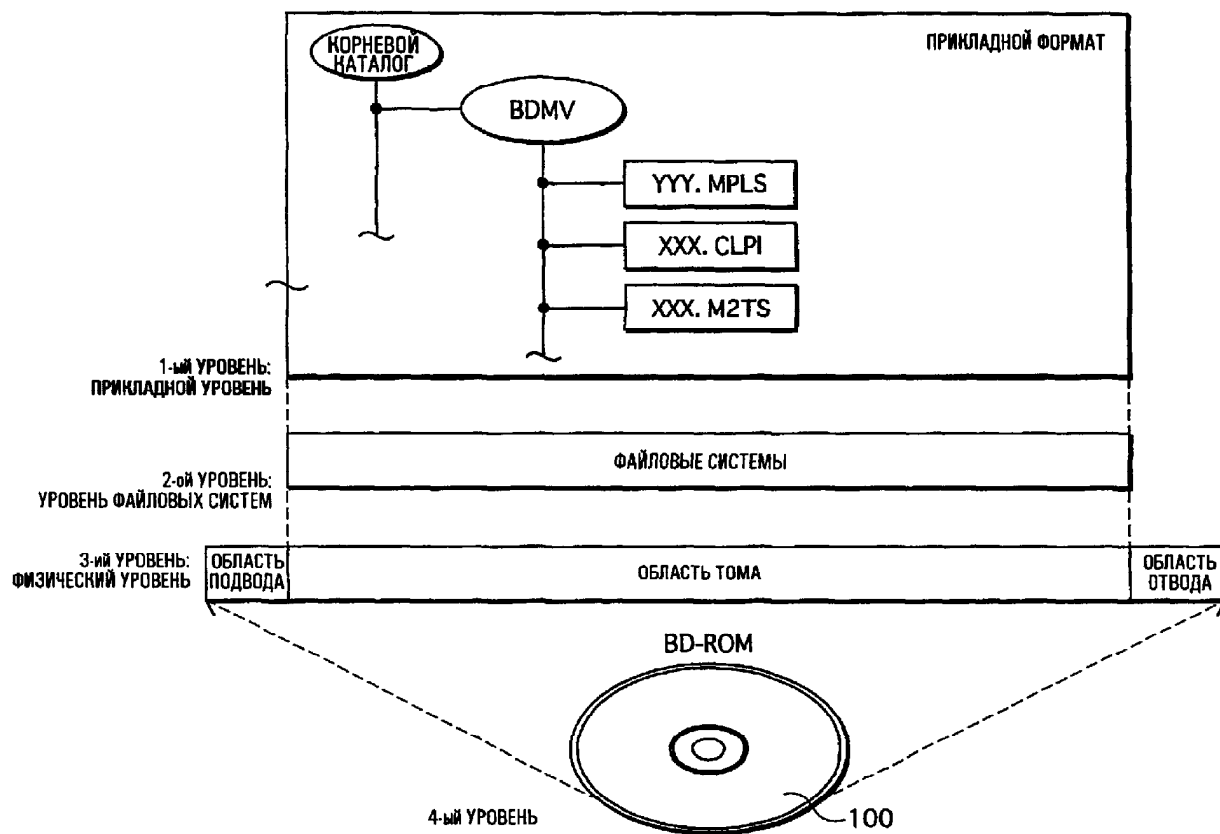
4. Устройство записи для записи на носителе записи потока графических данных, причем поток графических данных включает в себя пакет данных и управляющий пакет, пакет данных включает в себя графические данные, первую временную метку декодирования и первую временную метку представления, первая временная метка декодирования указывает начальное время декодирования графических данных, и первая временная метка представления указывает конечное время декодирования графических данных,

при этом управляющий пакет включает в себя вторую временную метку декодирования и вторую временную метку представления, причем вторая временная метка декодирования указывает начальное время очистки графической матрицы, а вторая временная метка представления указывает время отображения графических данных, которое является временем в момент или после конечного времени декодирования графических данных, и при этом декодированные графические данные должны быть визуализированы в графической матрице ко времени второй временной метки представления, причем графическая матрица представляет собой область, в которой визуализируются декодированные графические данные.

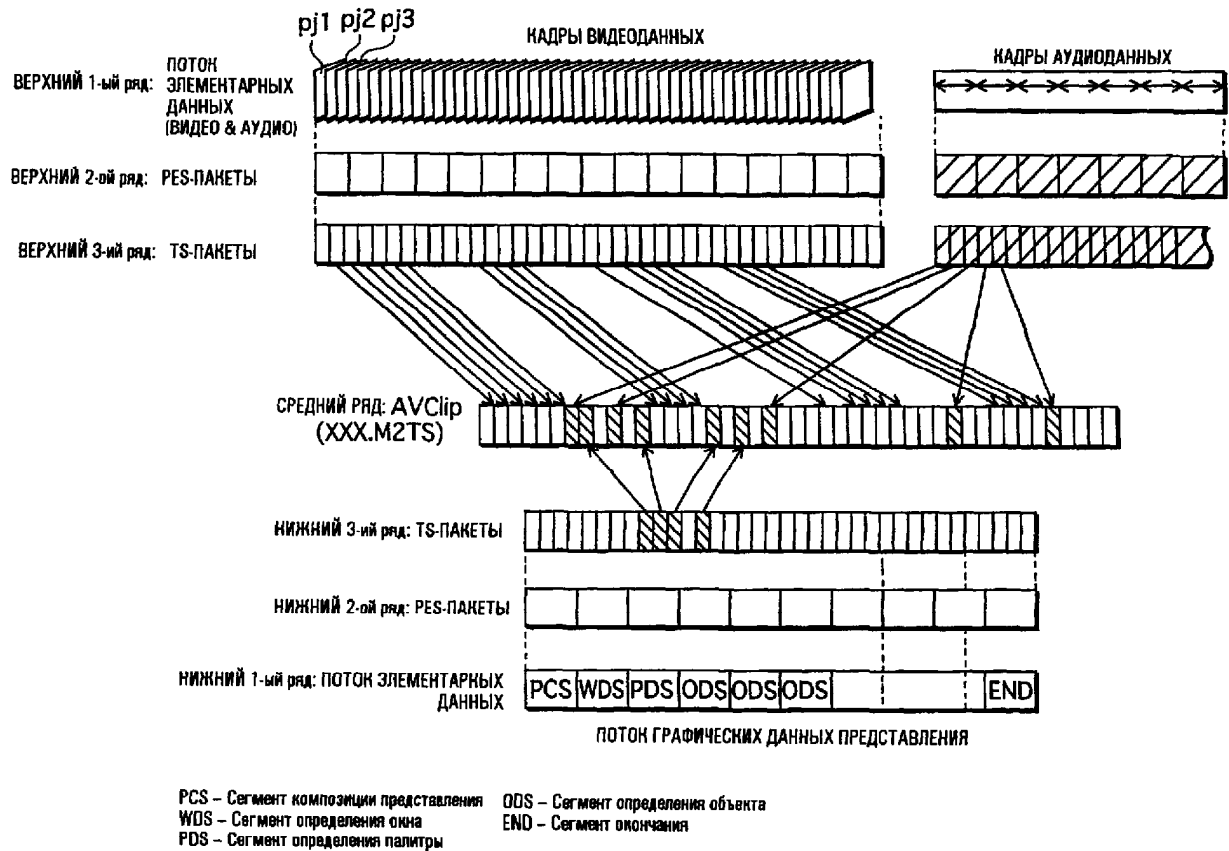
5. Способ записи для записи на носителе записи потока графических данных, причем поток графических данных включает в себя пакет данных и управляющий пакет, пакет данных включает в себя графические данные, первую временную метку декодирования и первую временную метку представления, первая временная метка декодирования указывает начальное время декодирования графических данных, и первая временная метка представления указывает конечное время декодирования графических данных, при этом управляющий пакет включает в себя вторую временную метку декодирования и вторую временную метку представления, причем вторая временная метка декодирования указывает начальное время очистки графической матрицы, а вторая временная метка представления указывает время отображения графических данных, которое является временем в момент или после конечного времени декодирования графических данных, и при этом декодированные графические данные должны быть визуализированы в графической матрице ко времени второй временной метки представления, причем графическая матрица представляет собой область, в которой визуализируются декодированные графические данные.



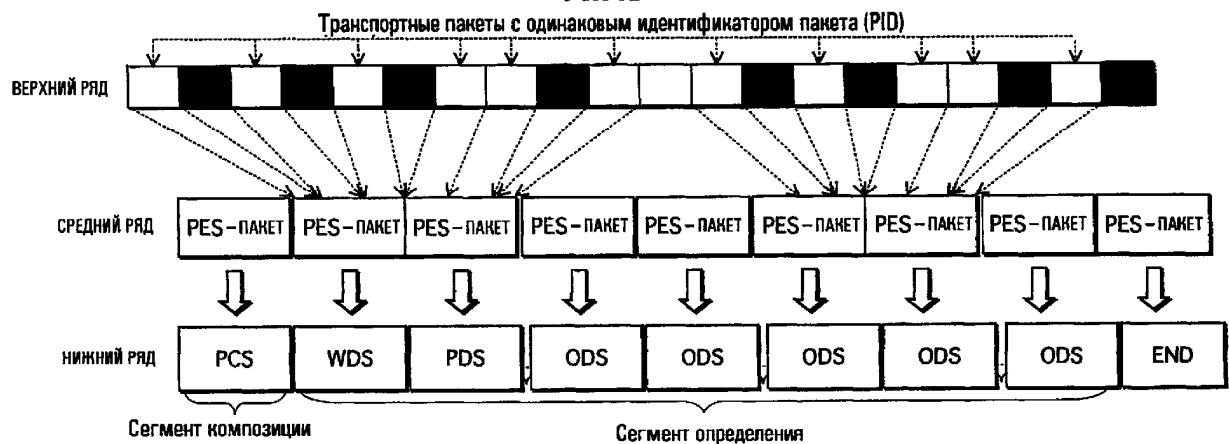
ФИГ.1



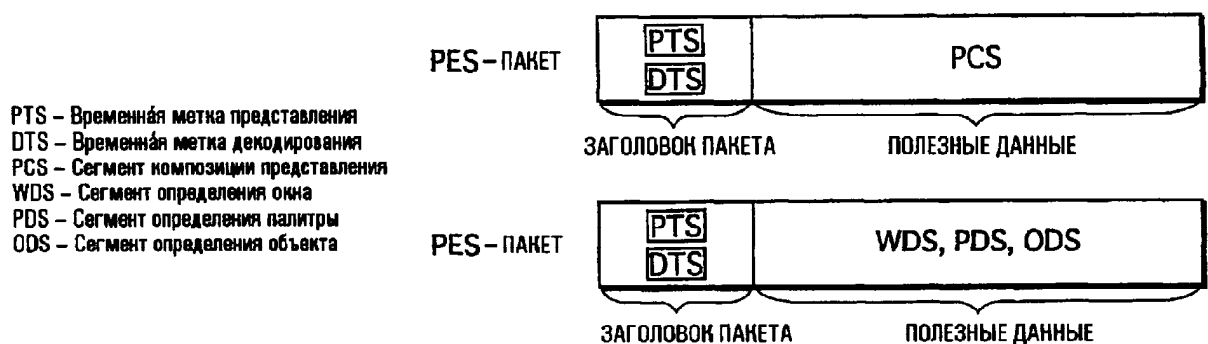
ФИГ.2



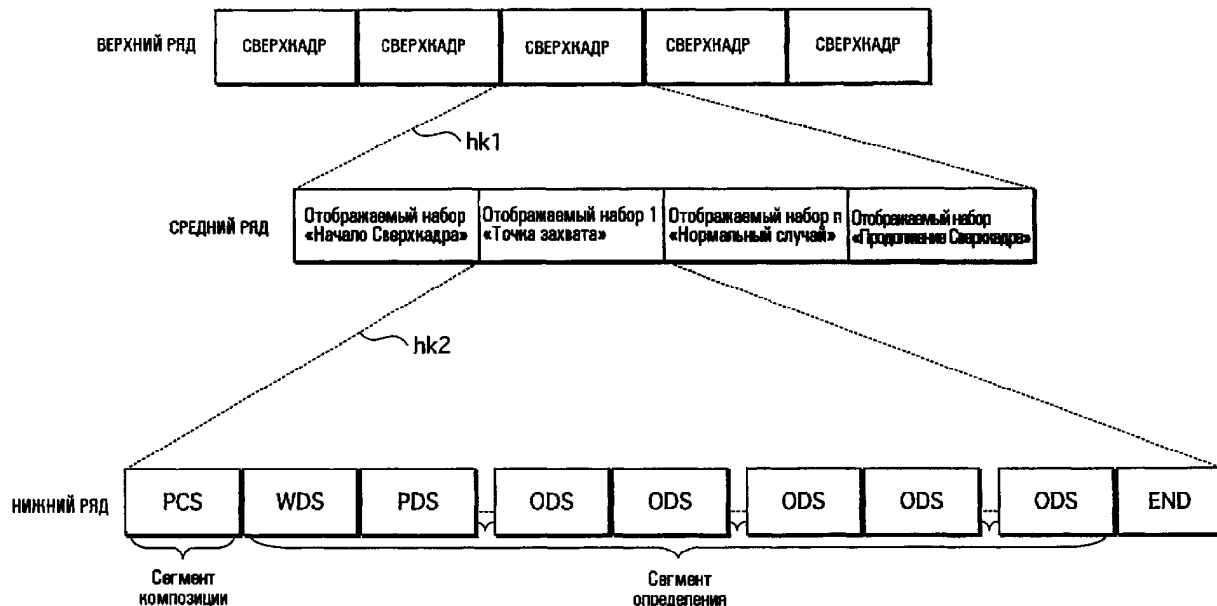
ФИГ.3



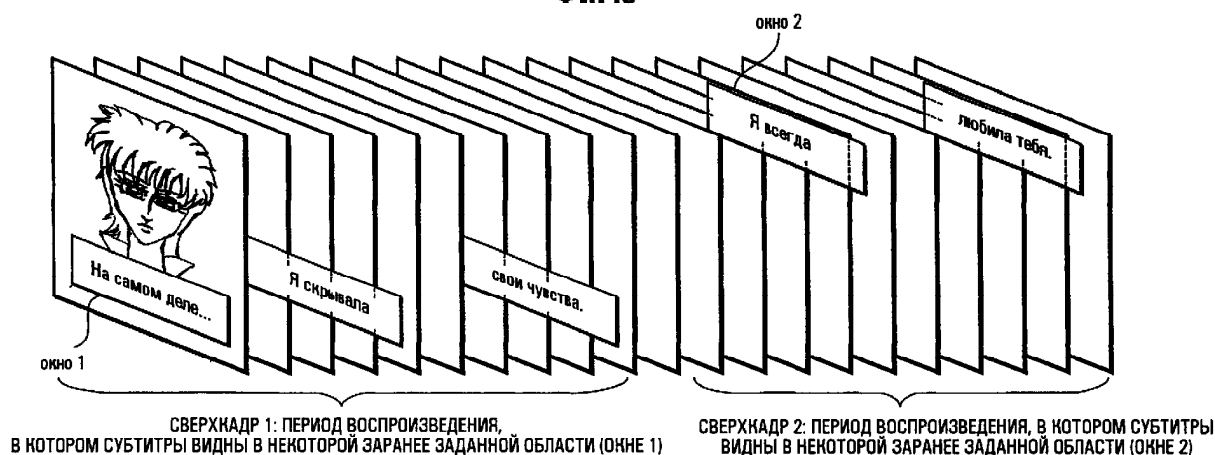
ФИГ.4А



ФИГ.4В



ФИГ.5



ФИГ.6

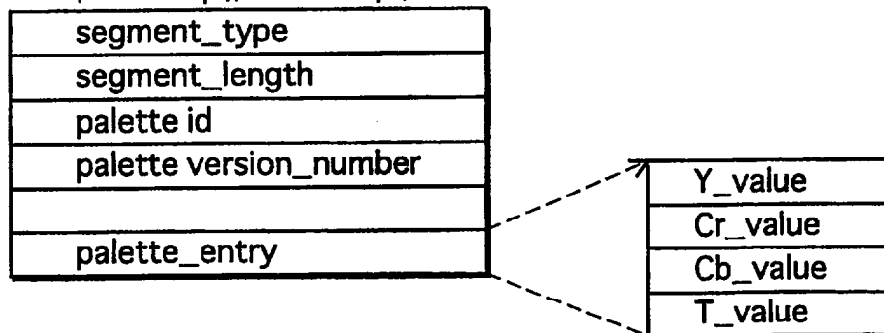
object_definition_segment
(Сегмент определения объекта)

segment_type	УПЛОТНЕННЫЙ ГРАФИЧЕСКИЙ ОБЪЕКТ
segment_length	
object_id	
object_version_number	
last in sequence flag	
object_data_fragment	

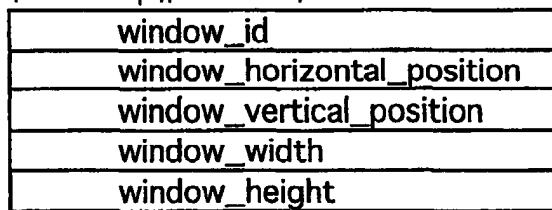
ФИГ.7А

palette_definition_segment

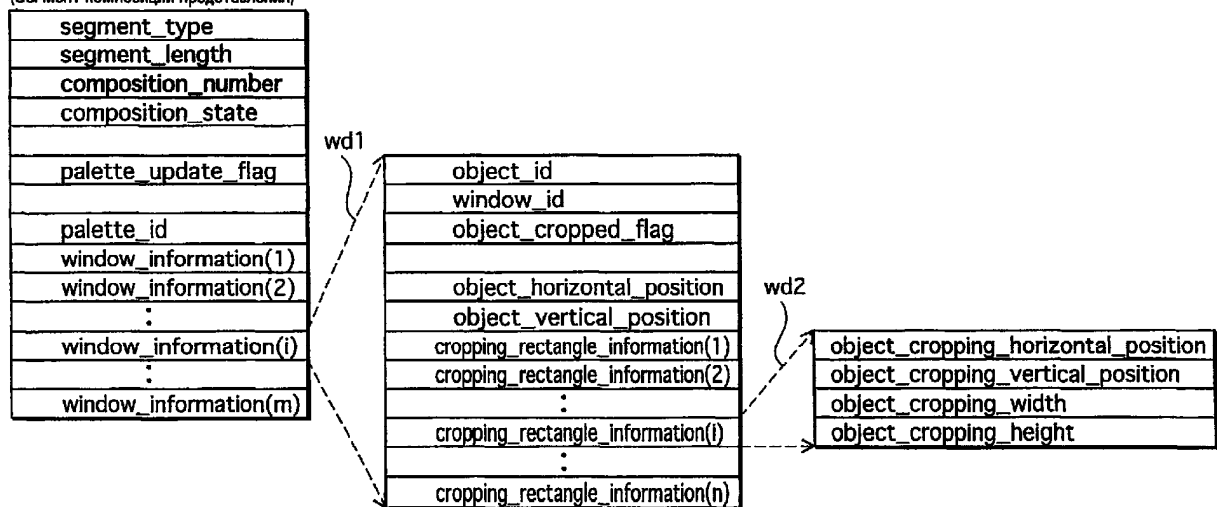
(Сегмент определения палитры)

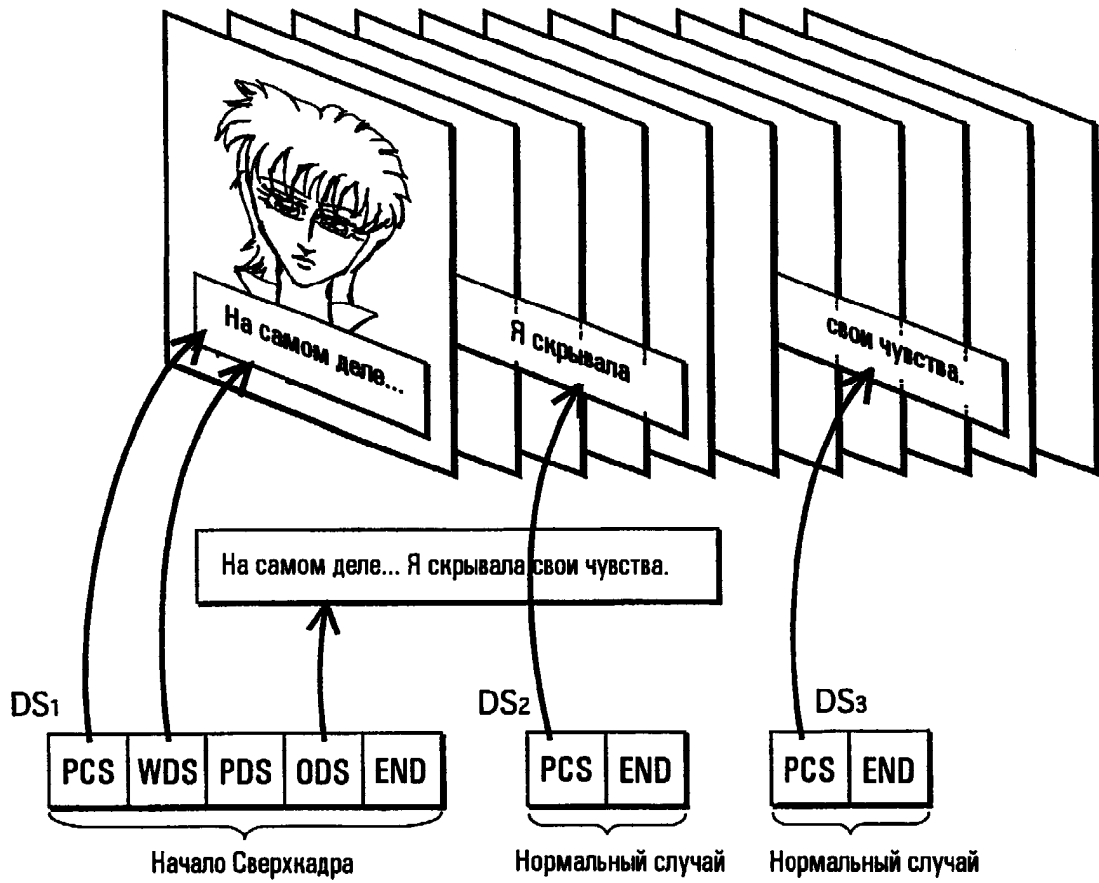
**ФИГ.7В****window_definition_segment**

(Сегмент определения окна)

**ФИГ.8А****presentation_composition_segment**

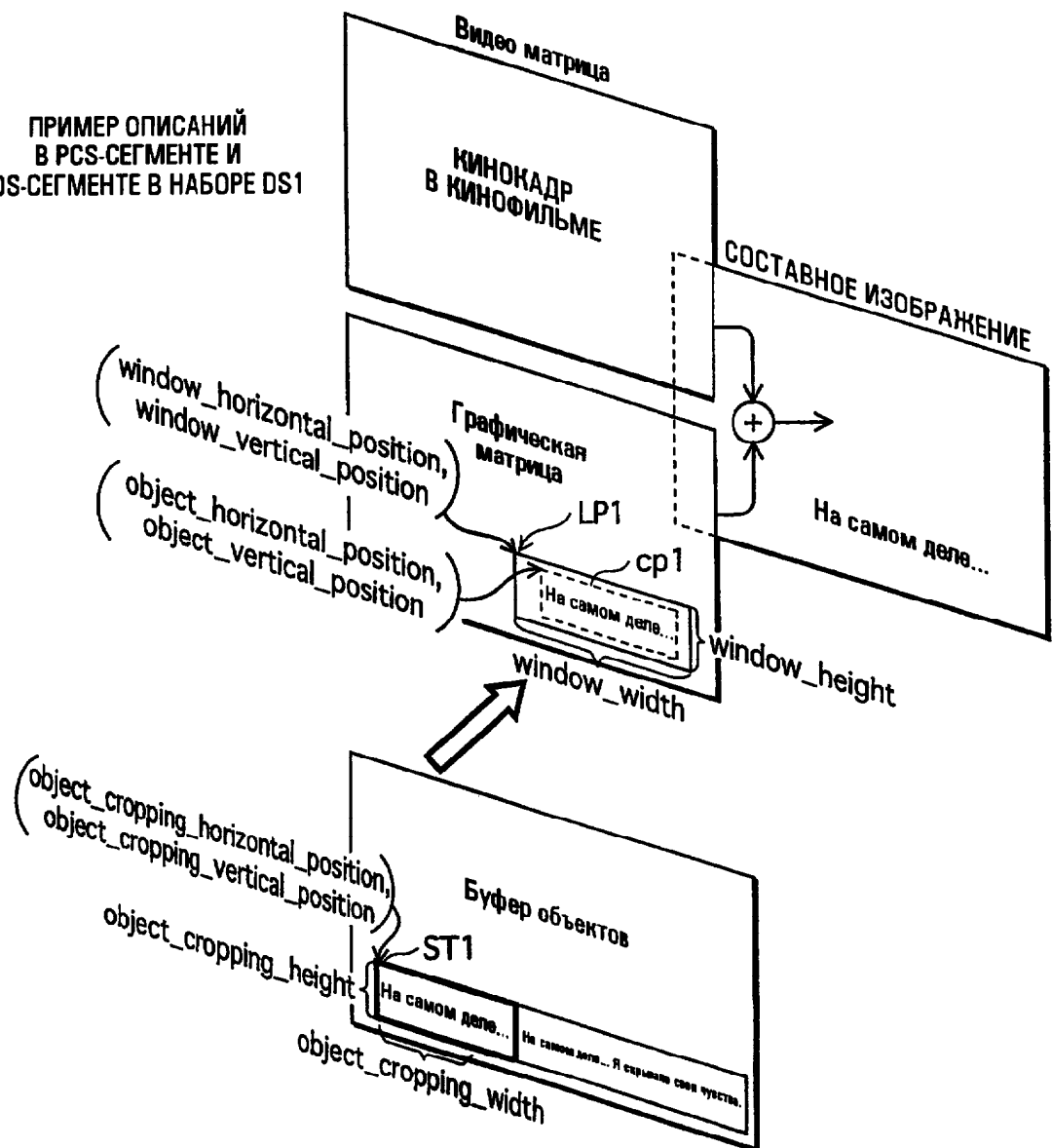
(Сегмент композиции представления)

**ФИГ.8В**

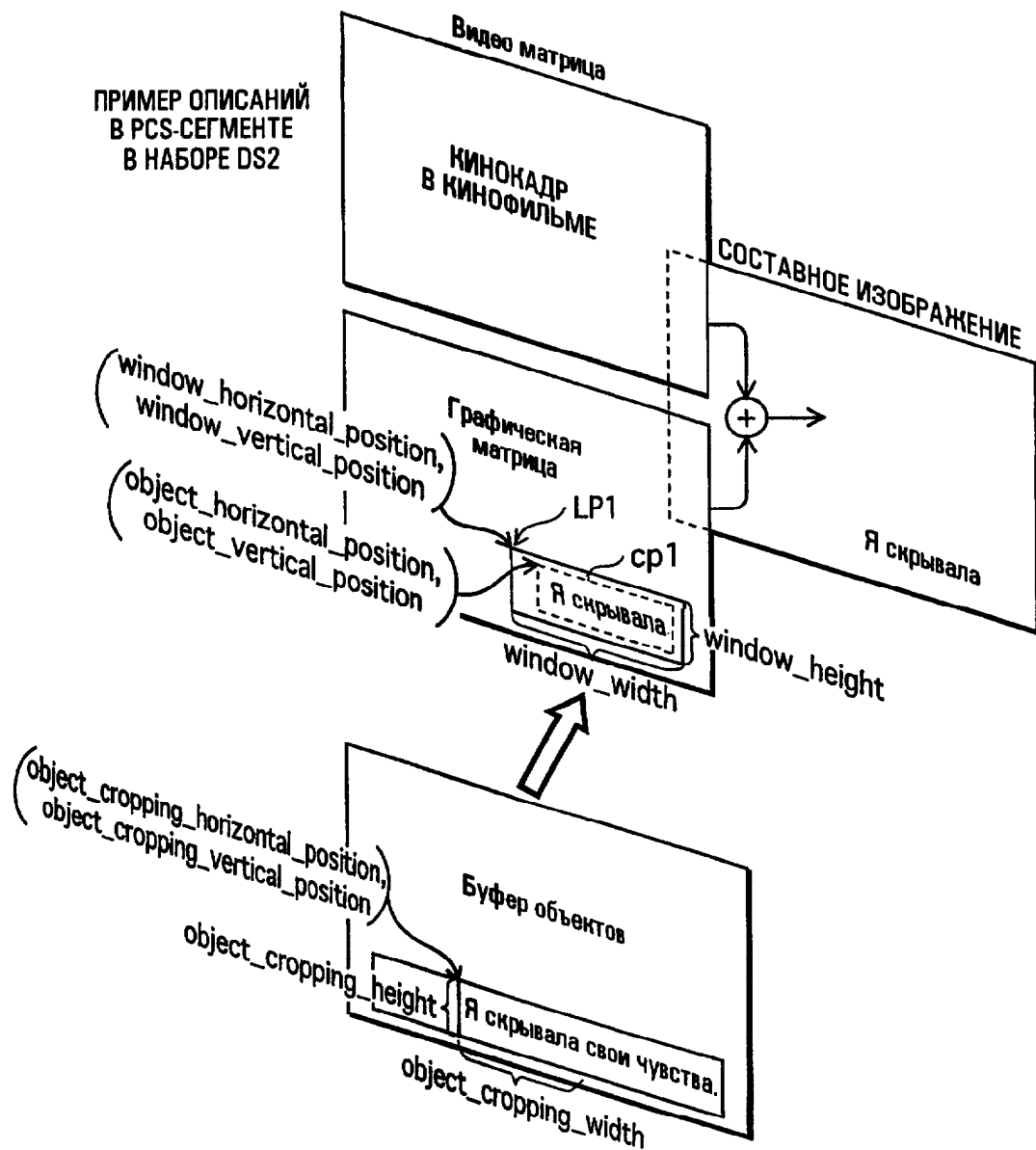


ФИГ.9

ПРИМЕР ОПИСАНИЙ
В PCS-СЕКМЕНТЕ И
WDS-СЕКМЕНТЕ В НАБОРЕ DS1

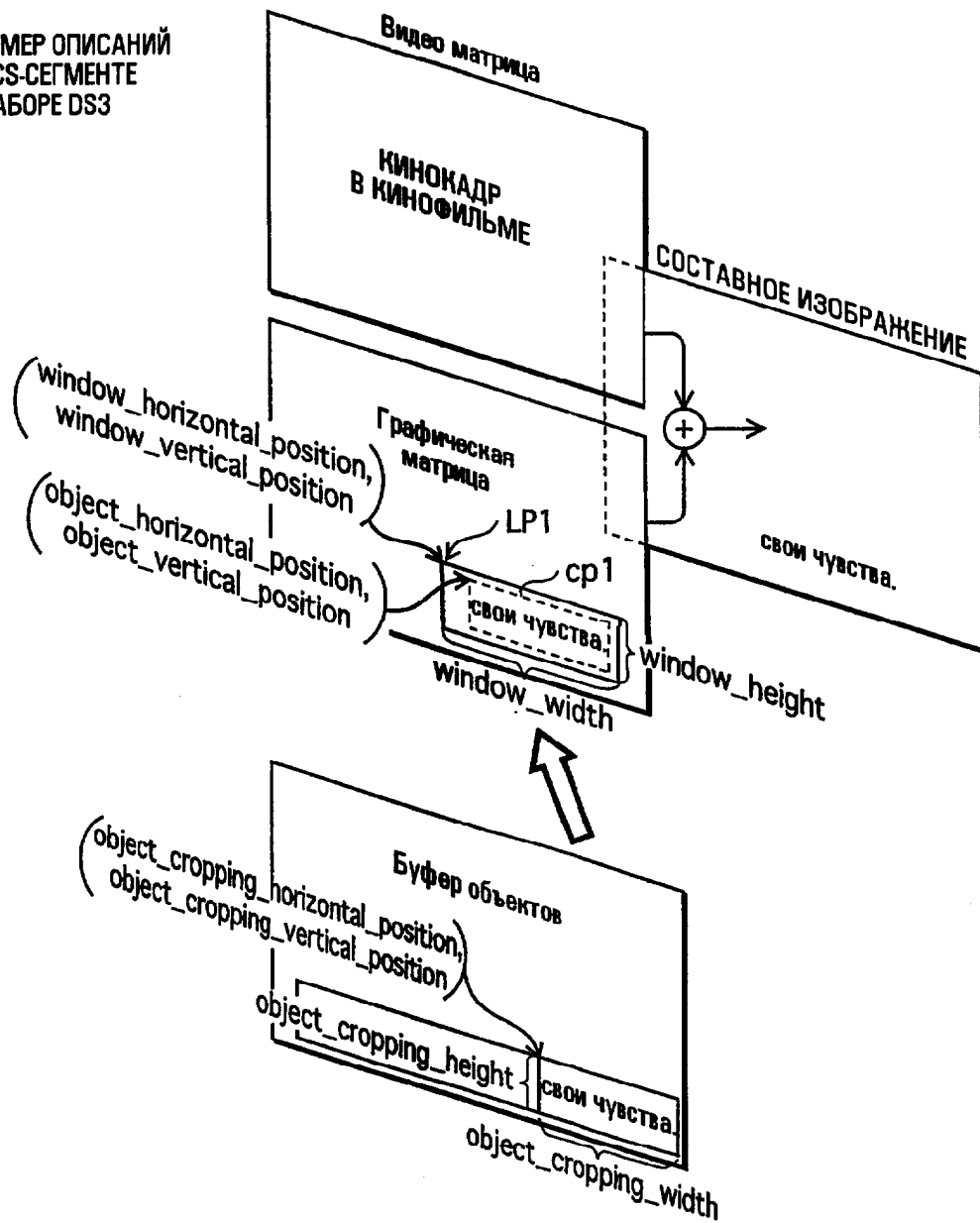


ФИГ.10

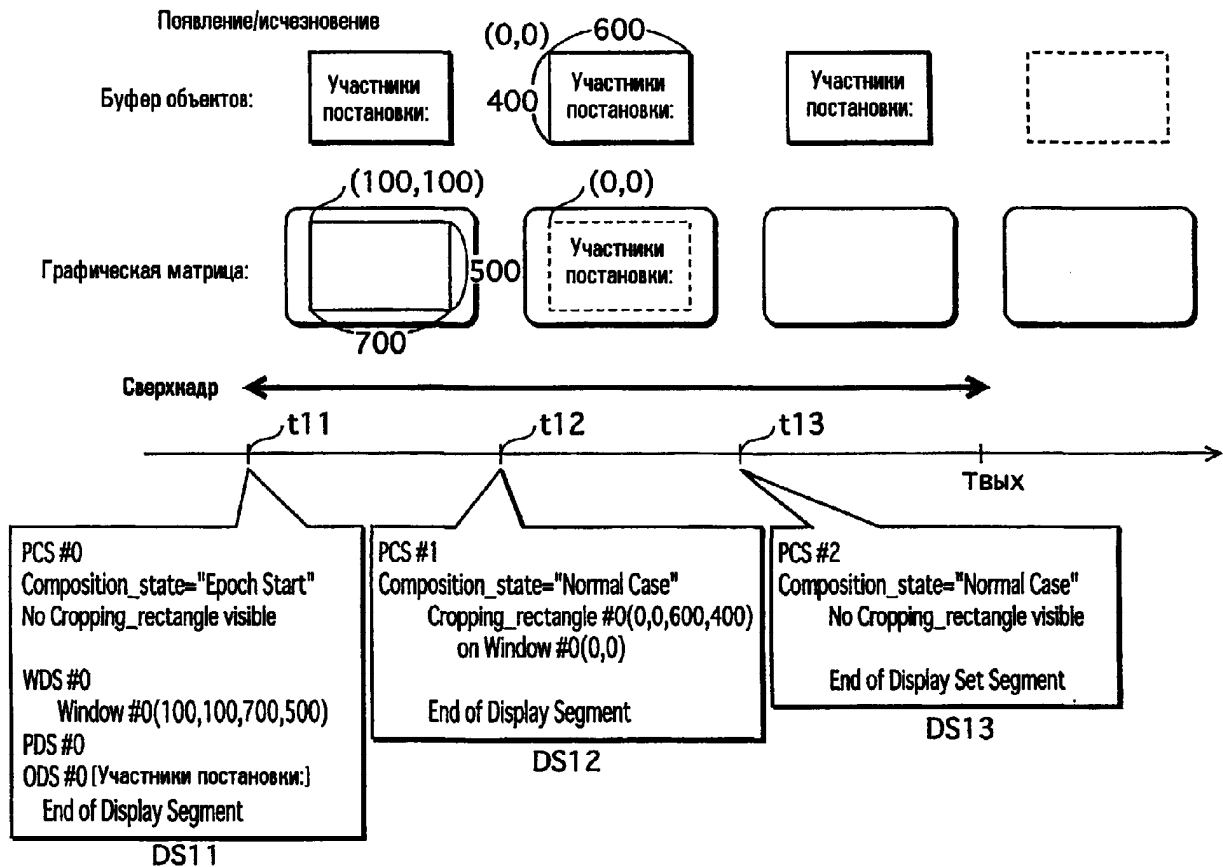


ФИГ.11

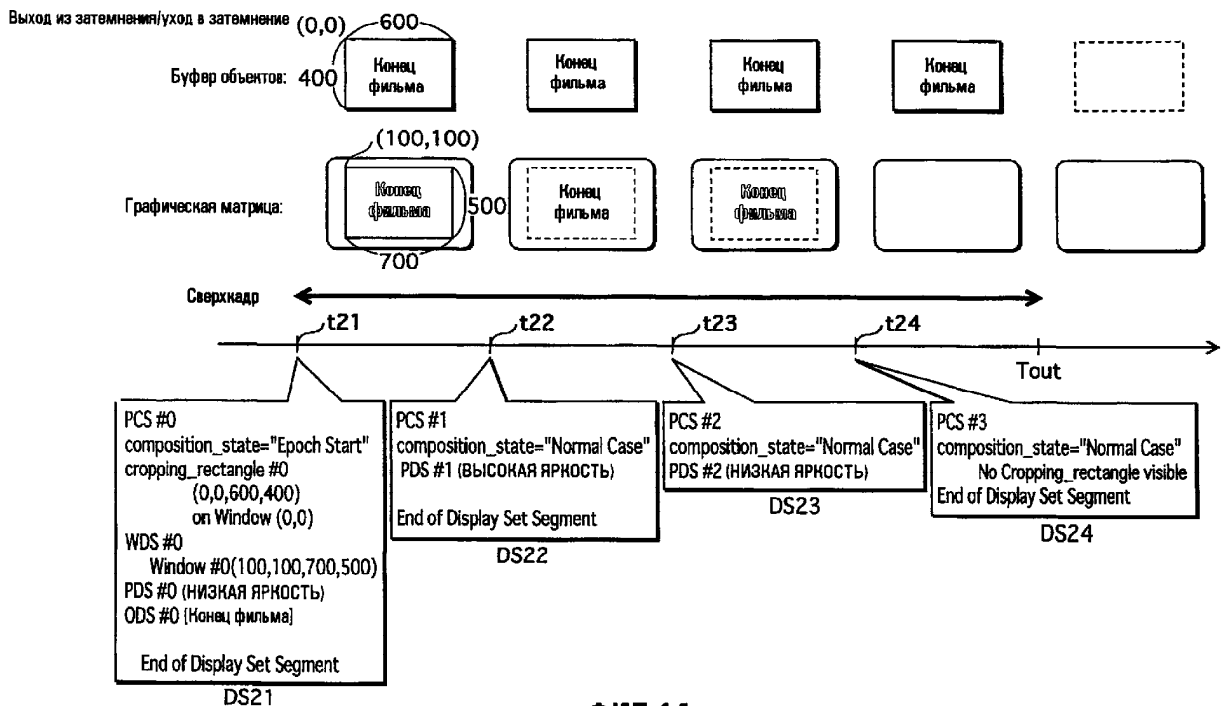
ПРИМЕР ОПИСАНИЙ
В PCS-СЕКМЕНТЕ
В НАБОРЕ DS3



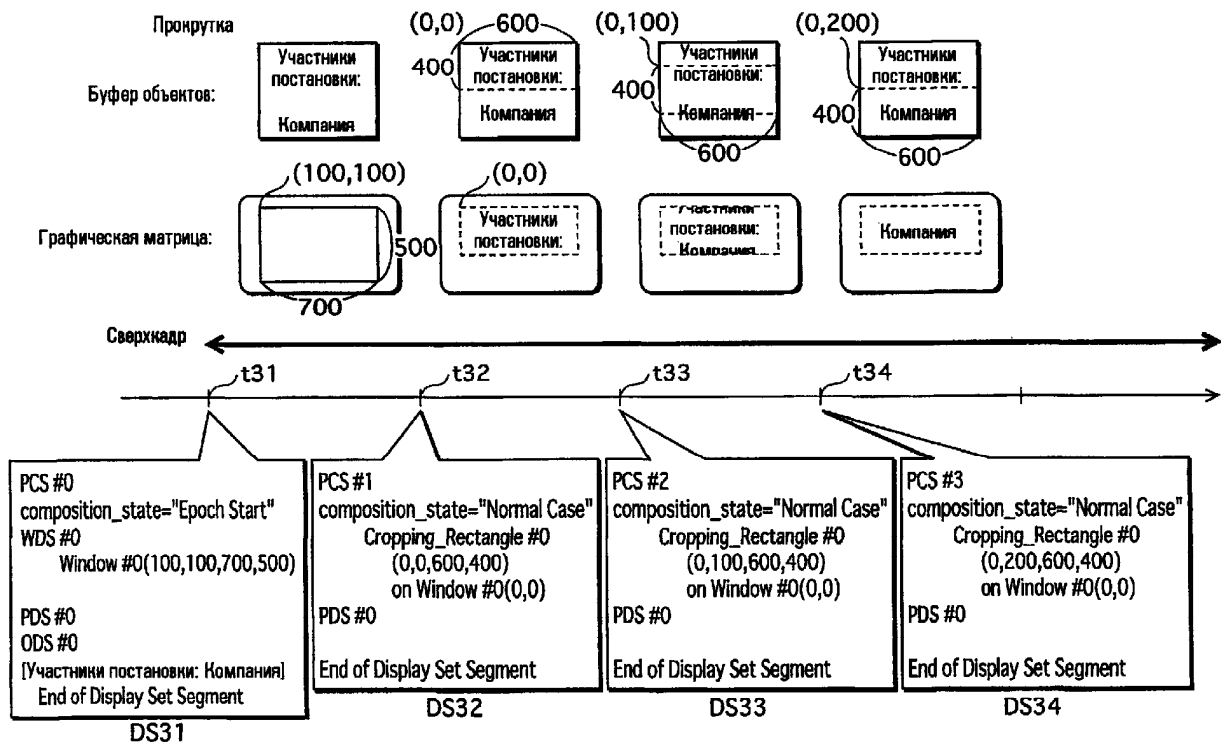
ФИГ.12



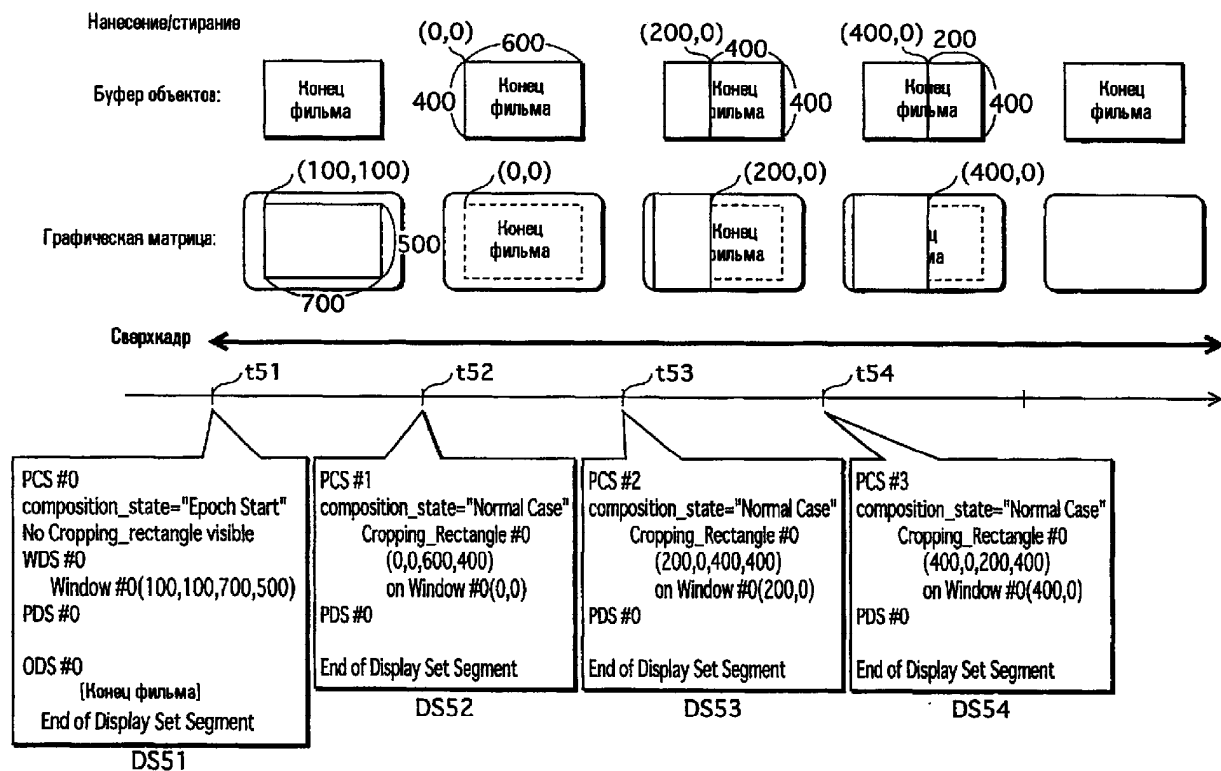
ФИГ.13



ФИГ.14

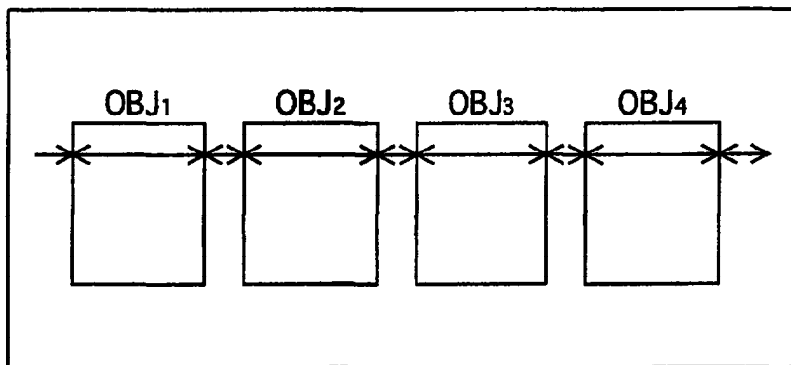
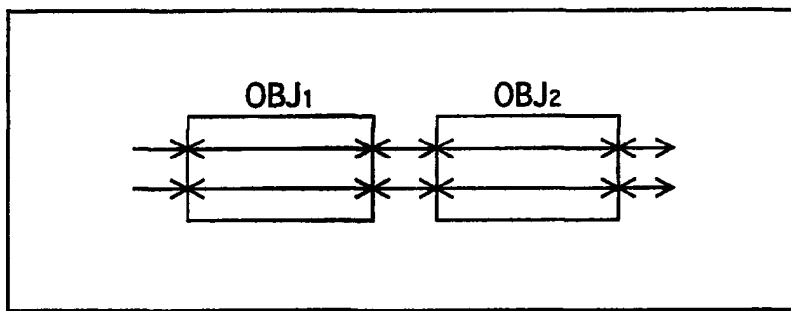


ФИГ.15



ФИГ.16

БУФЕР ОБЪЕКТОВ



Х:КРАЙ

ФИГ.17

$PTS(DS_n[PCS]) \geq DTS(DS_n[PCS]) + DECODEDURATION(DS_n)$

Где:

- $DECODEDURATION(DS_n)$ вычисляется следующим образом:

```

decode_duration = 0 ;
decode_duration += PLANEINITIALIZATIONTIME( DS_n ) ;
if( DS_n.PCS.num_of_objects == 2 )
{
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
    if( DS_n.PCS.OBJ[0].window_id == DS_n.PCS.OBJ[1].window_id )
    {
        decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
    }
    else
    {
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
        decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[1], decode_duration ) ;
        decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[1].window_id )//256*106 ) ;
    }
}
else if( DS_n.PCS.num_of_objects == 1 )
{
    decode_duration += WAIT( DS_n, DS_n.PCS.OBJ[0], decode_duration ) ;
    decode_duration += 90000*( SIZE( DS_n.PCS.OBJ[0].window_id )//256*106 ) ;
}
return decode_duration ;

```

- $PLANEINITIALIZATIONTIME(DS_n)$ вычисляется следующим образом:

```

initialize_duration=0 ;
if( DS_n.PCS.composition_state == EPOCH_START )
{
    initialize_duration = 90000*( 8*video_width*video_height//256*106 ) ;
}
else
{
    for( i=0 ; i < WDS.num_windows ; i++ )
    {
        if( EMPTY( DS_n.WDS.WIN[i], DS_n ) )
            initialize_duration += 90000*( SIZE( DS_n.WDS.WIN[i] )//256*106 ) ;
    }
}
return initialize_duration ;

```

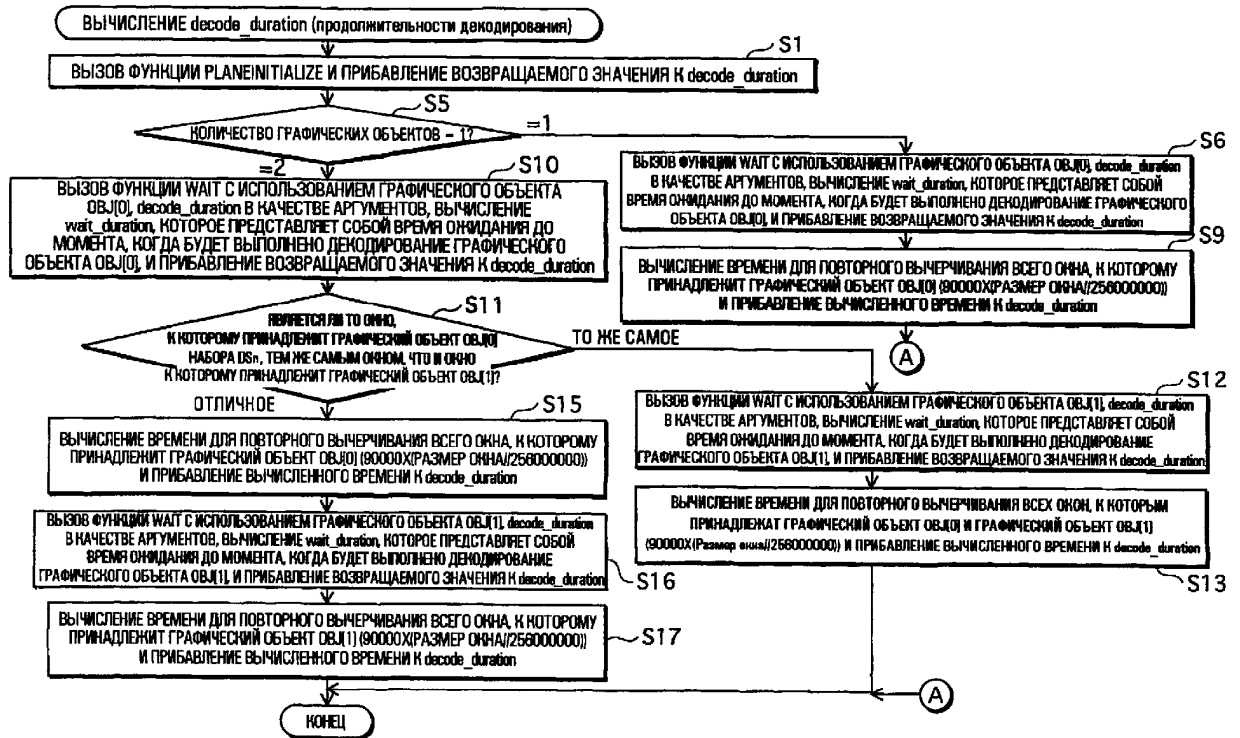
- $WAIT(DS_n, OBJ, current_duration)$ вычисляется следующим образом:

```

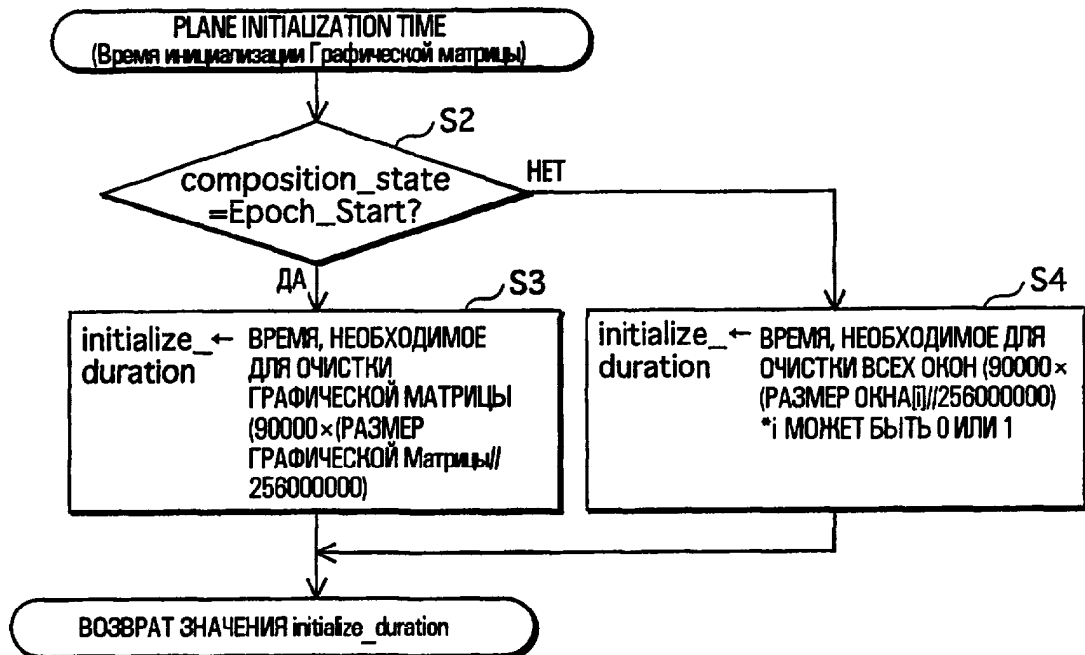
wait_duration = 0 ;
if( EXISTS( OBJ.object_id, DS_n ) )
{
    object_definition_ready_time = PTS( GET( OBJ.object_id, DS_n ) ) ;
    current_time = DTS( DS_n.PCS )+current_duration ;
    if( current_time < object_definition_ready_time )
        wait_duration += object_definition_ready_time - current_time ;
}
return wait_duration ;

```

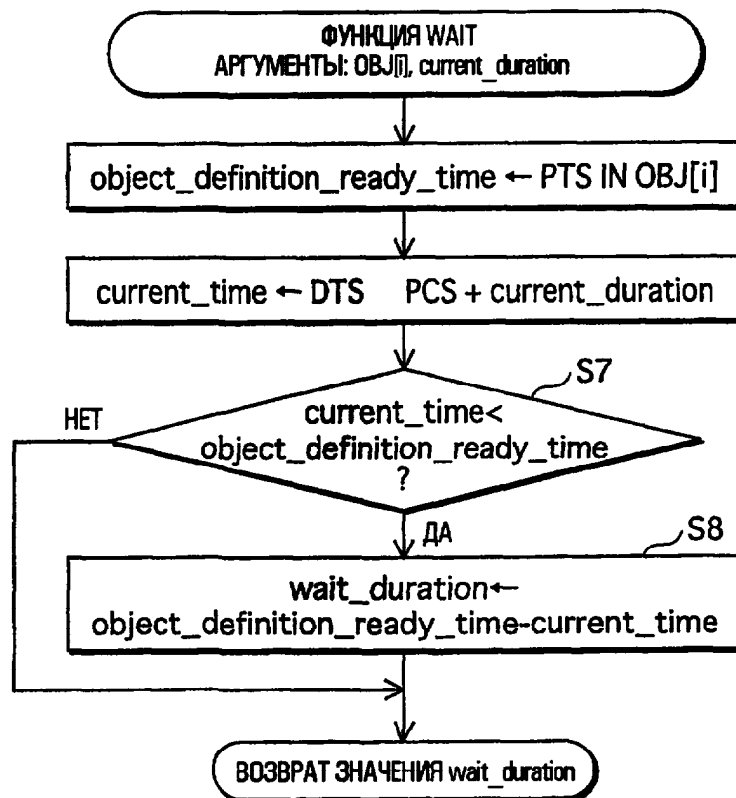
ФИГ.18



ФИГ.19

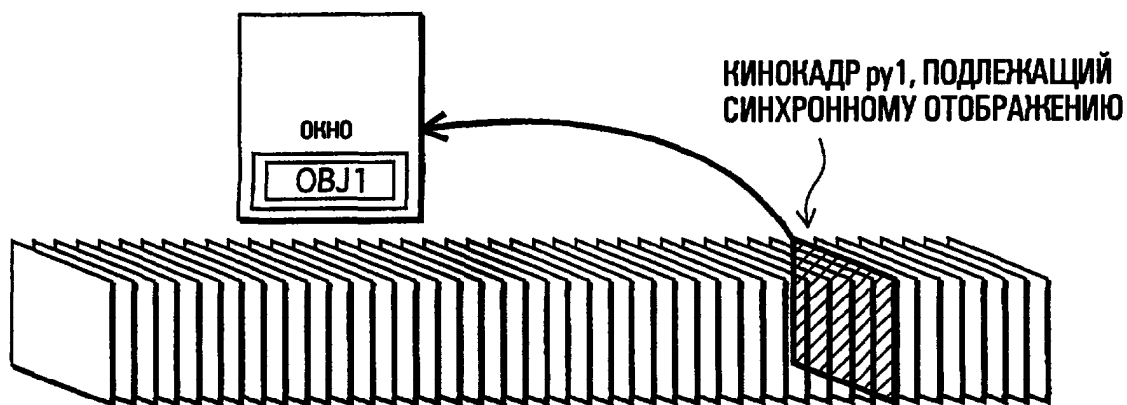


ФИГ.20А

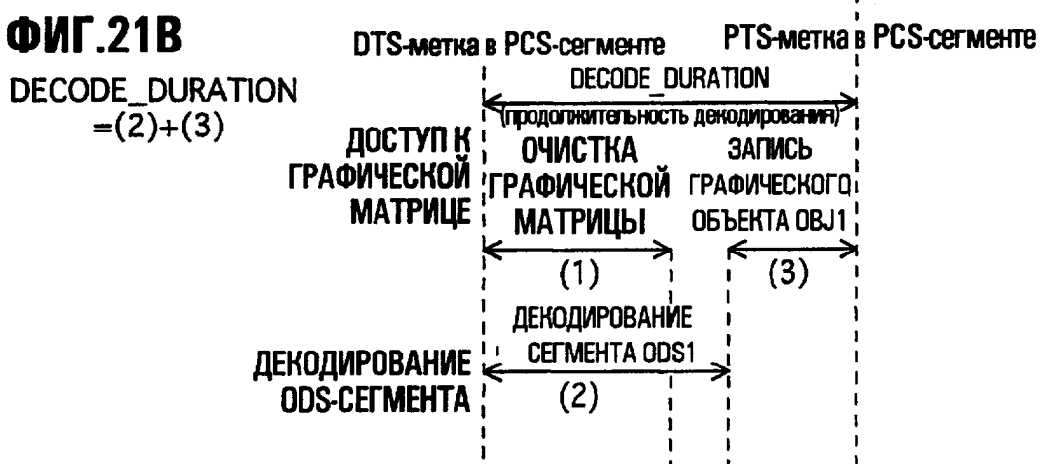


ФИГ.20В

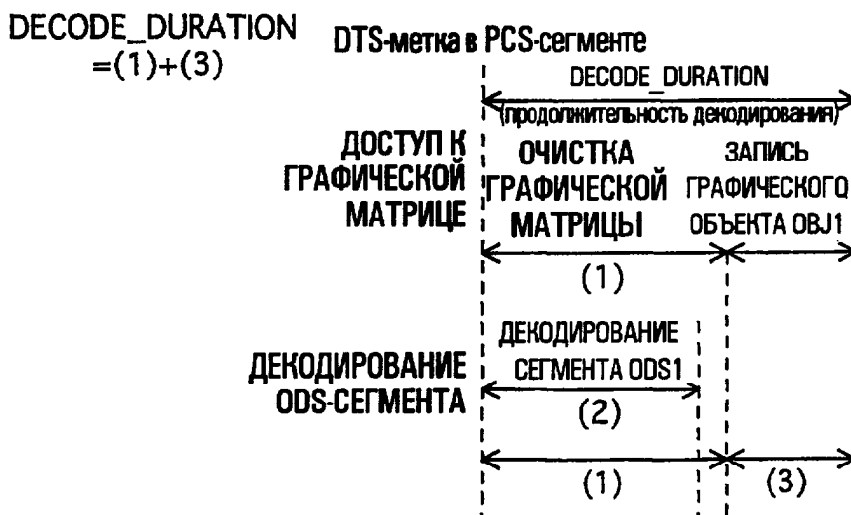
ФИГ.21А



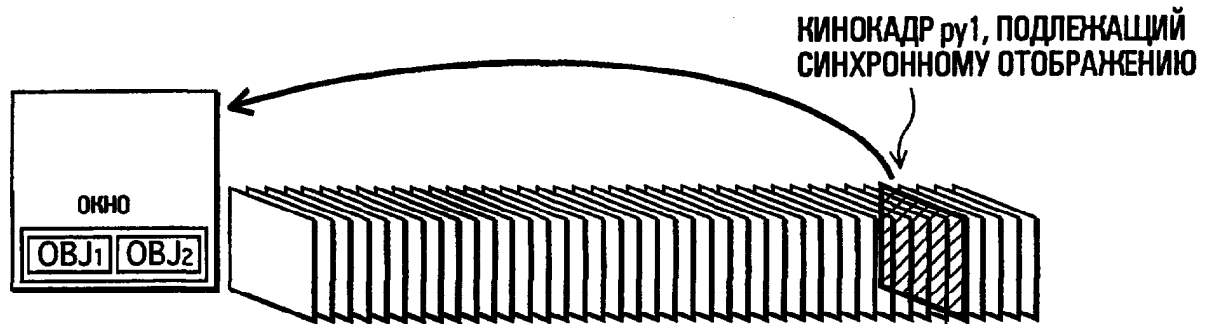
ФИГ.21В



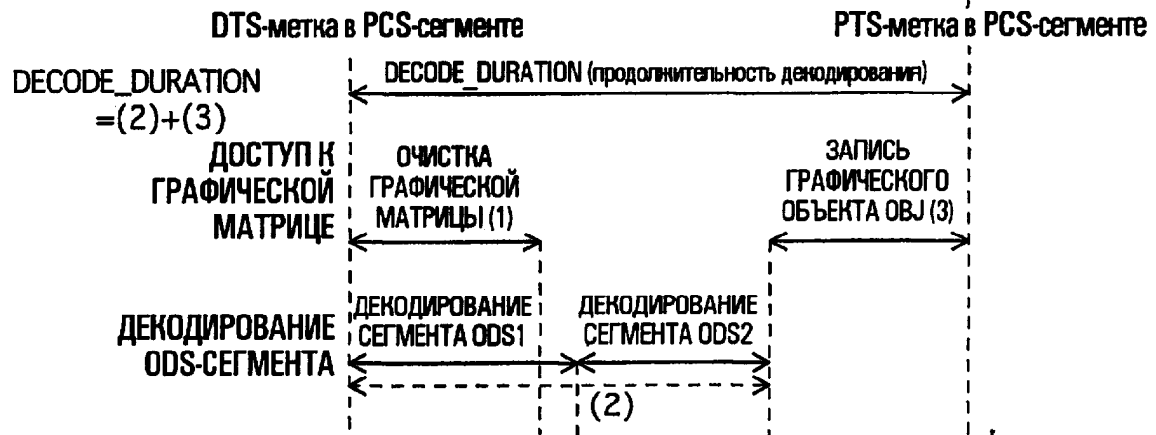
ФИГ.21С



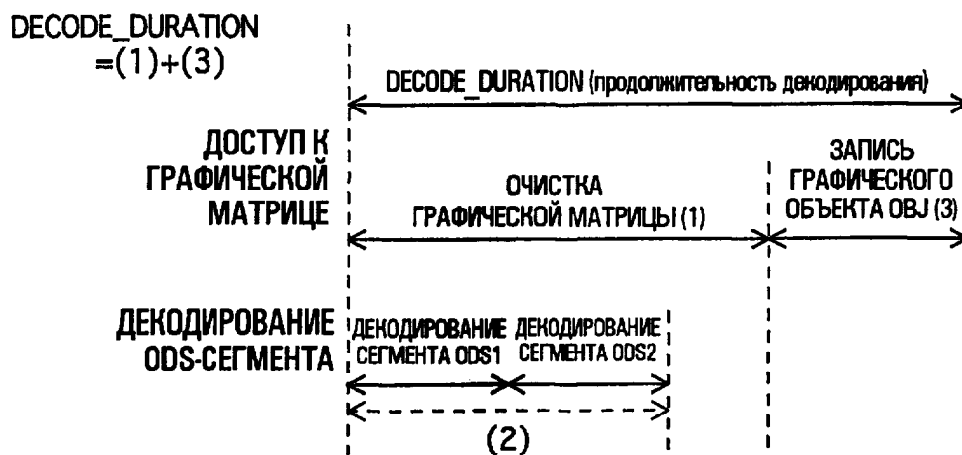
ФИГ.22А



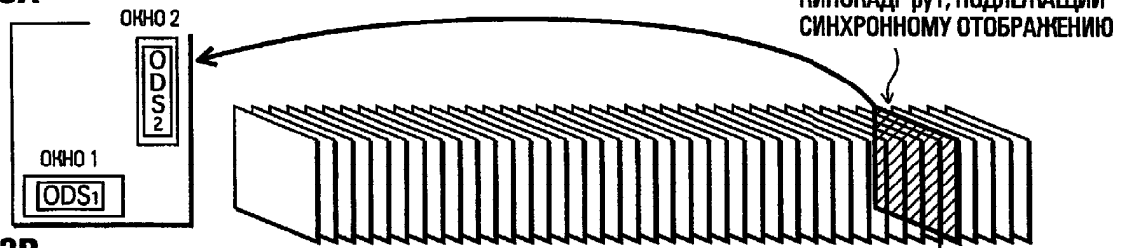
ФИГ.22В



ФИГ.22С

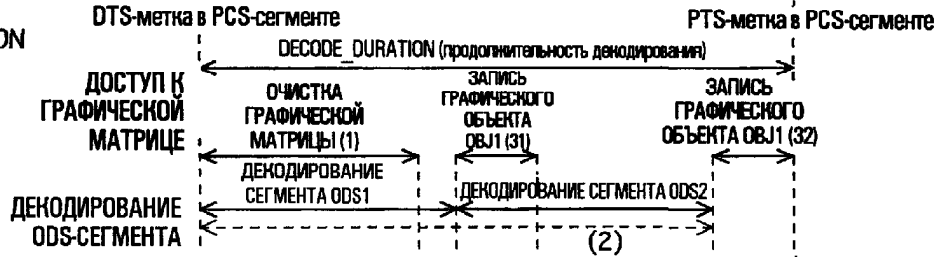


ФИГ.23А



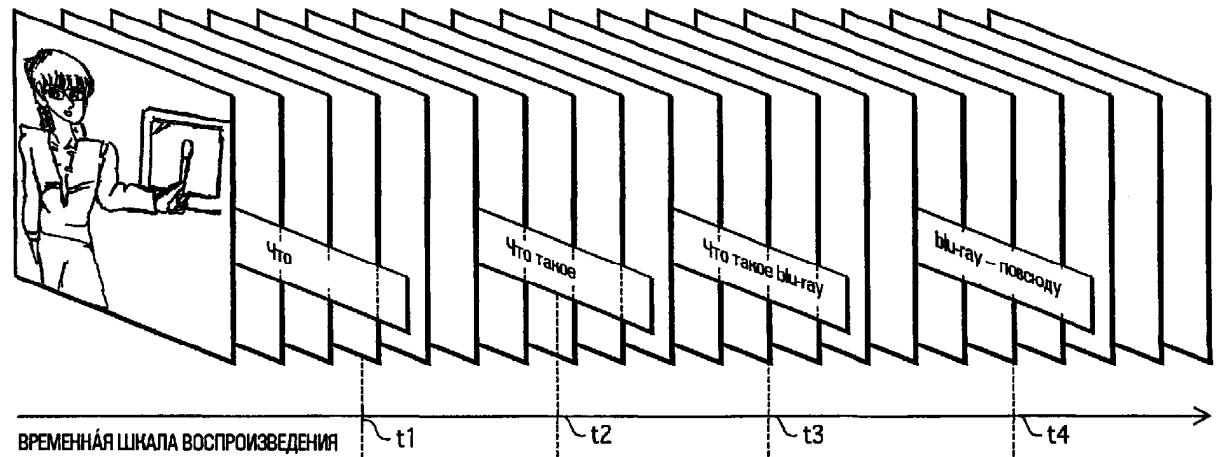
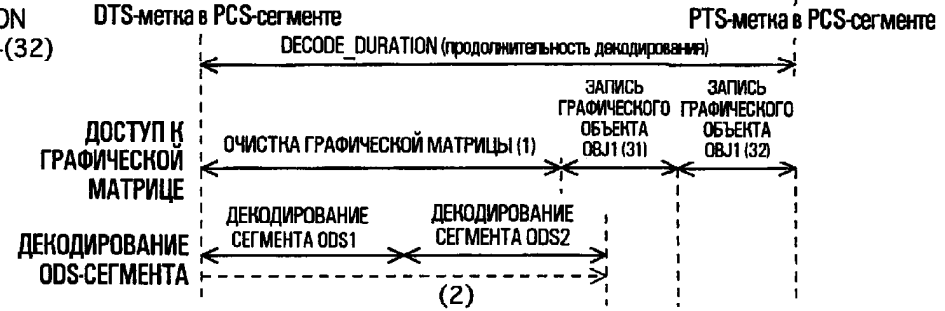
ФИГ.23В

DECODE_DURATION
=(2)+(32)



ФИГ.23С

DECODE_DURATION
=(1)+(31)+(32)



ФИГ.24

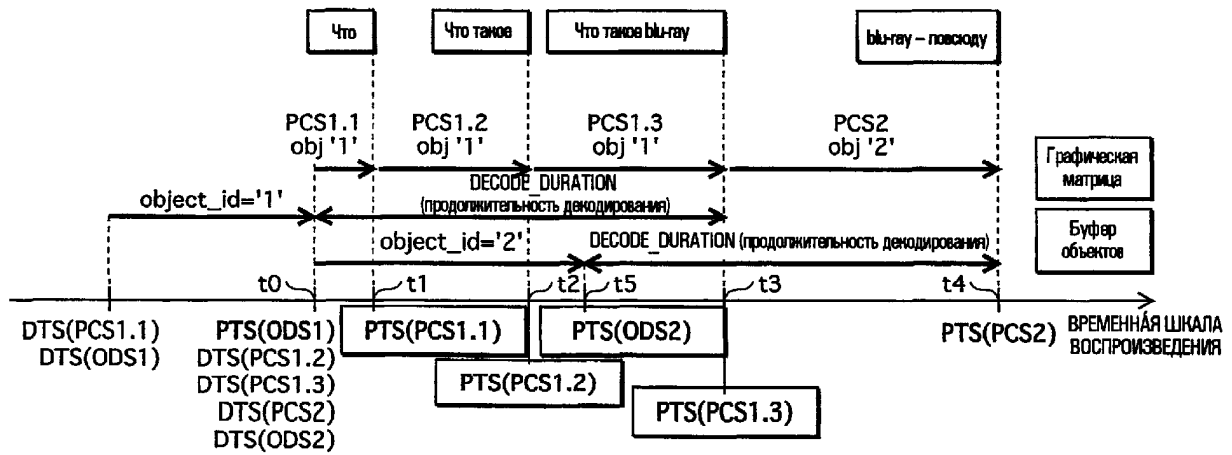
(DS1) PCS1.1-PDS1-ODS1-END
ODS1 «Что такое blu-ray»

(DS2) PCS1.2-END

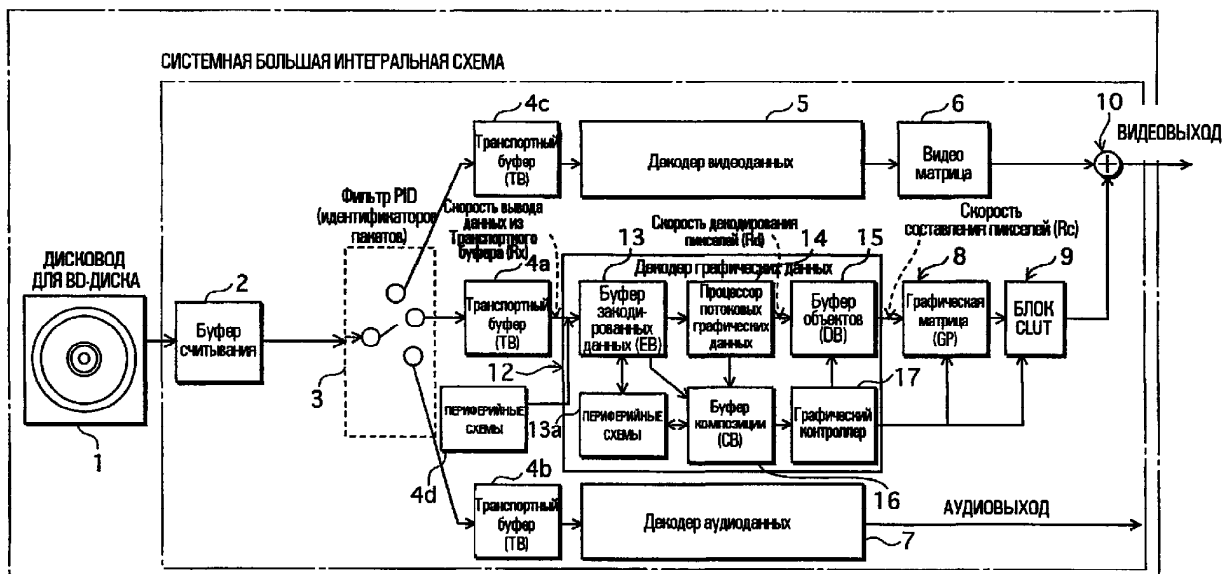
(DS3) PCS1.3-END

(DS4) PCS2-ODS2-END
ODS2 «blu-ray – повсюду»

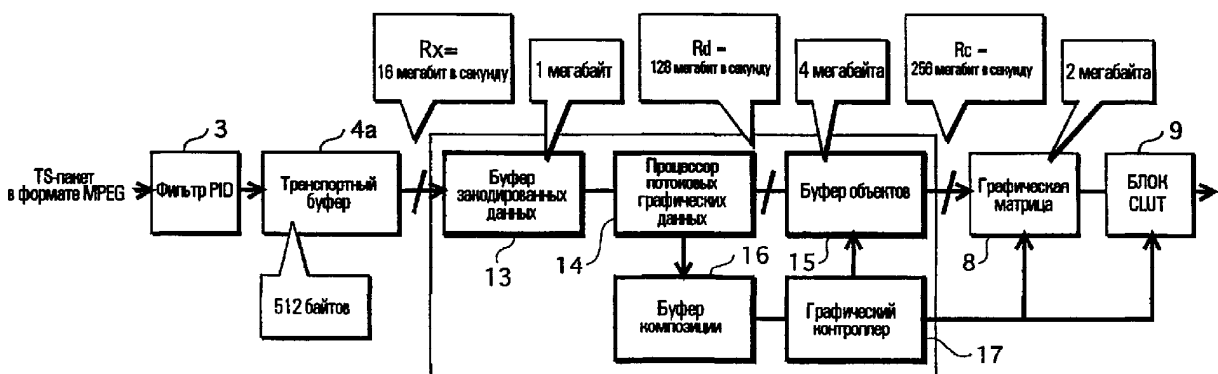
ФИГ.25А



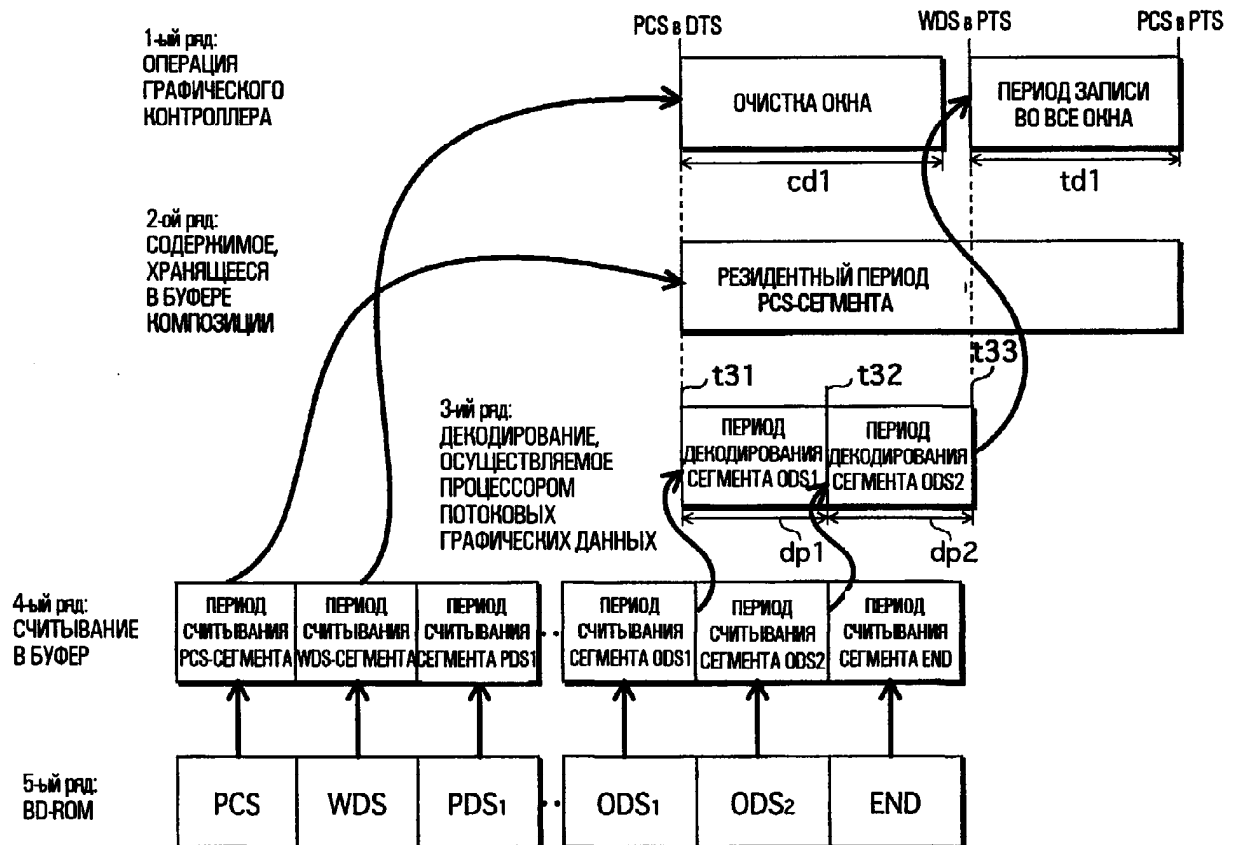
ФИГ.25В



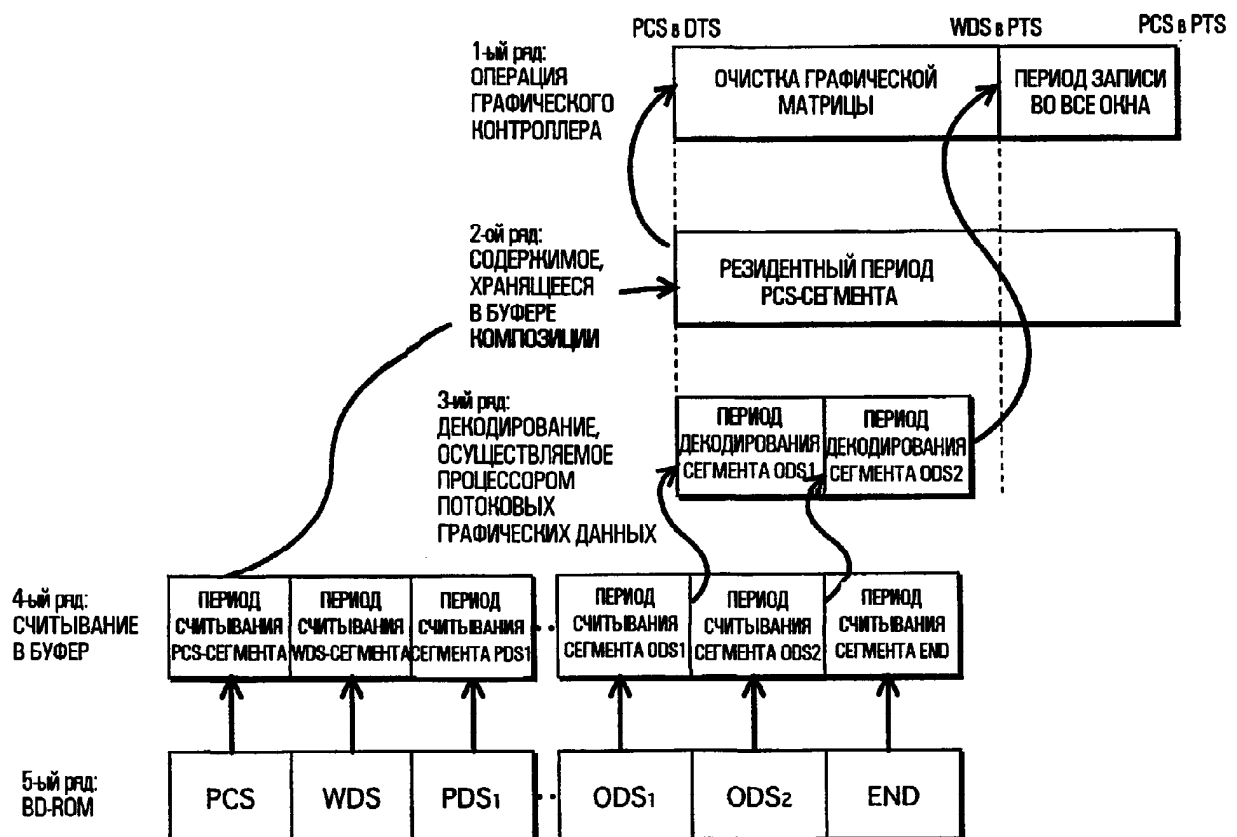
ФИГ.26



ФИГ.27



ФИГ.28



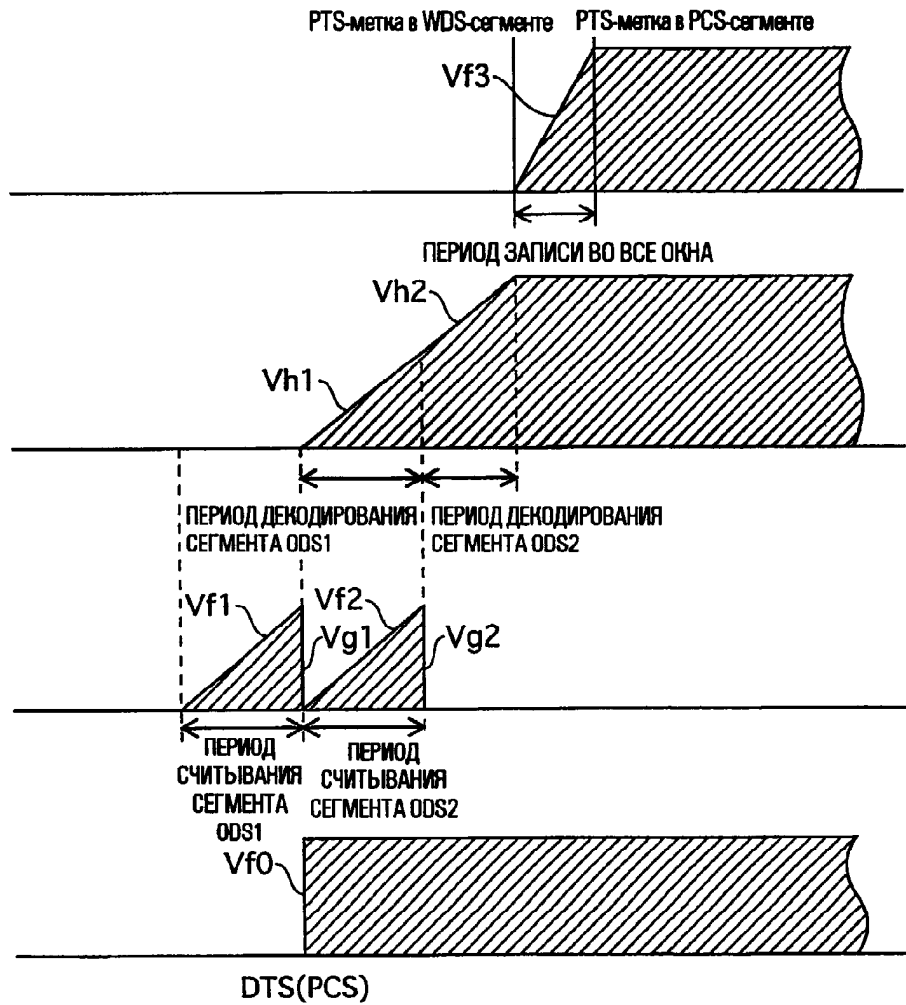
ФИГ.29

1-й ряд:
СОСТОЯНИЕ БУФЕРА ДЛЯ
Графической матрицы

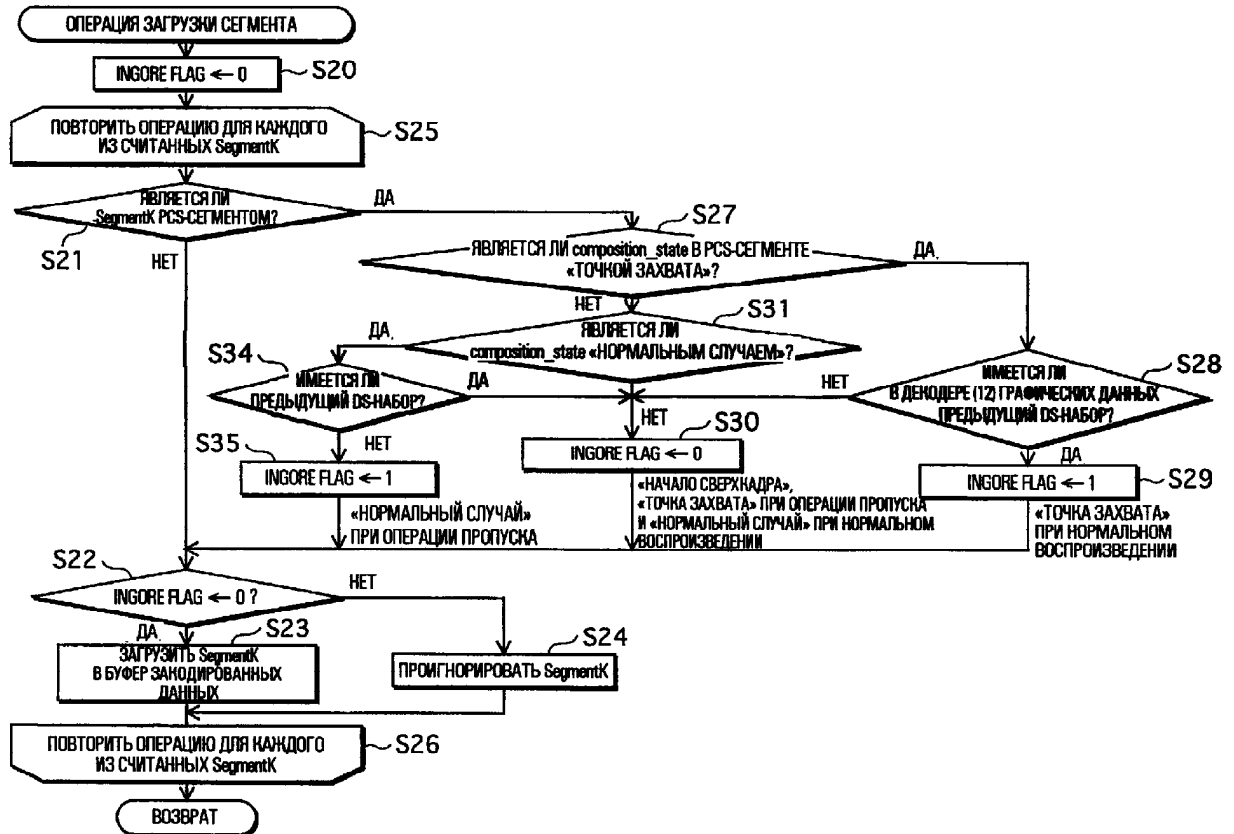
2-ой ряд:
СОСТОЯНИЕ БУФЕРА ДЛЯ
Буфера объектов

3-ий ряд:
СОСТОЯНИЕ БУФЕРА ДЛЯ
Буфера закодированных данных

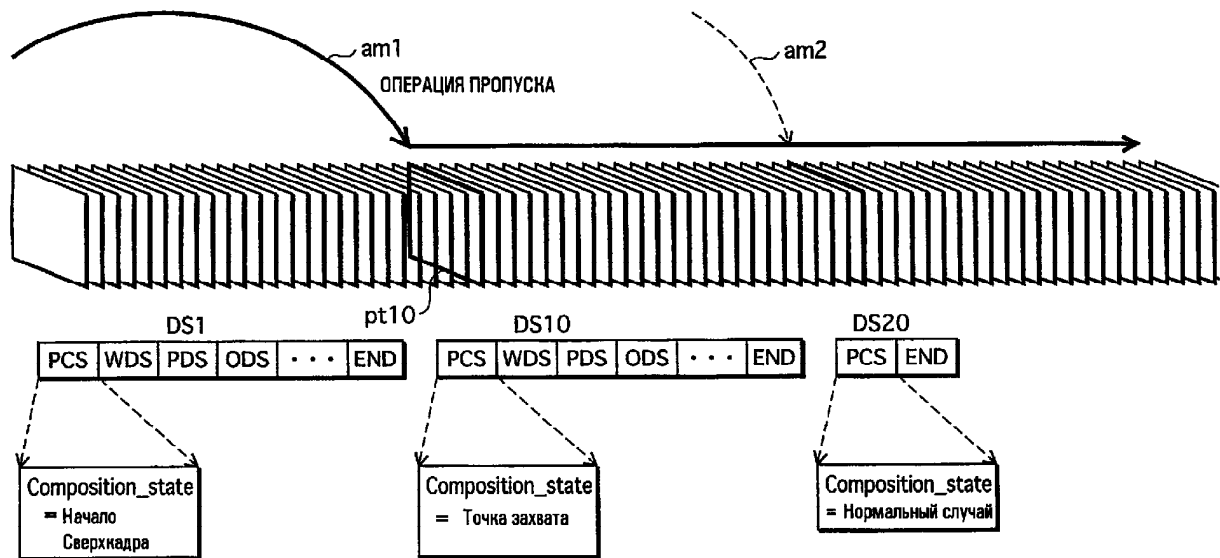
4-ый ряд:
СОСТОЯНИЕ БУФЕРА ДЛЯ
Буфера композиции



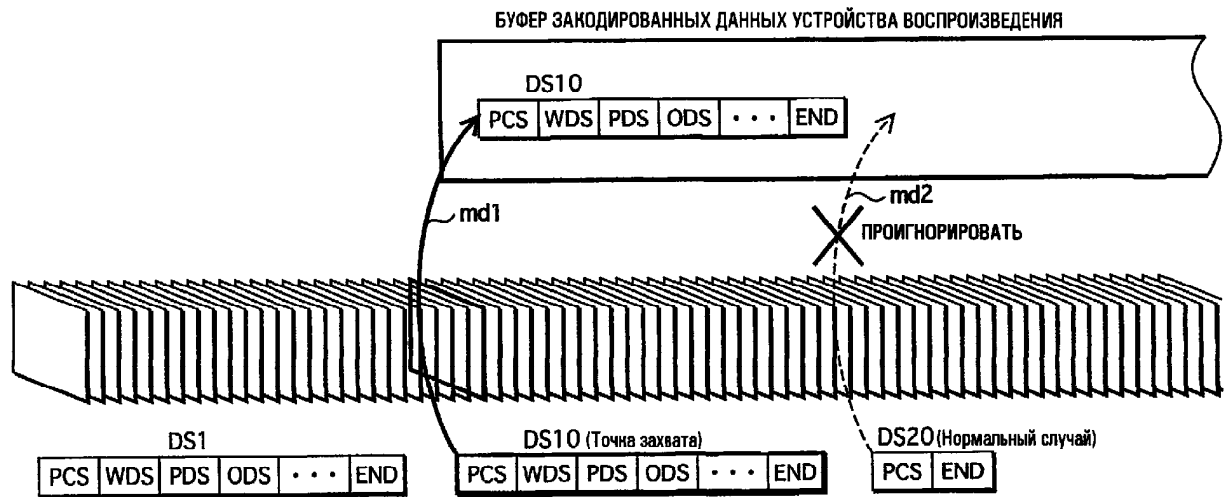
ФИГ.30



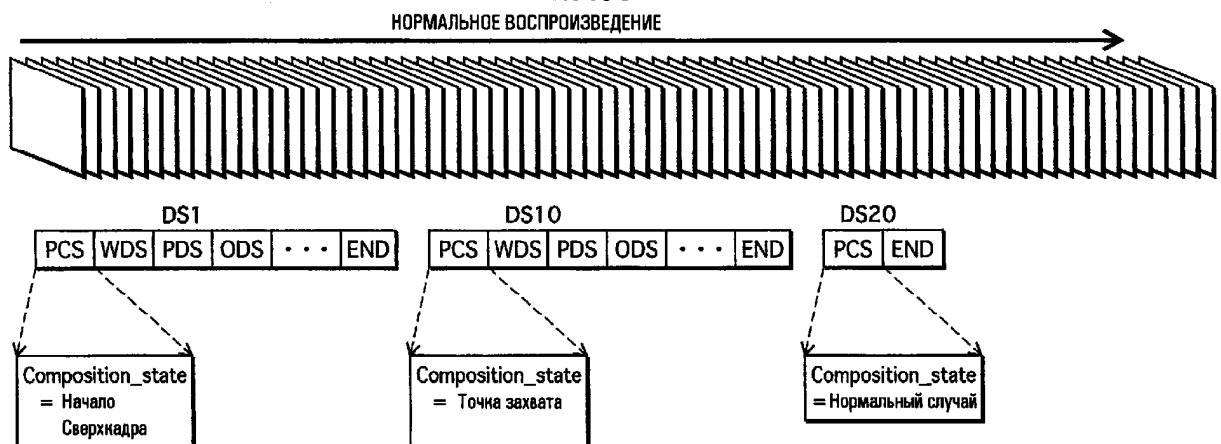
ФИГ.31



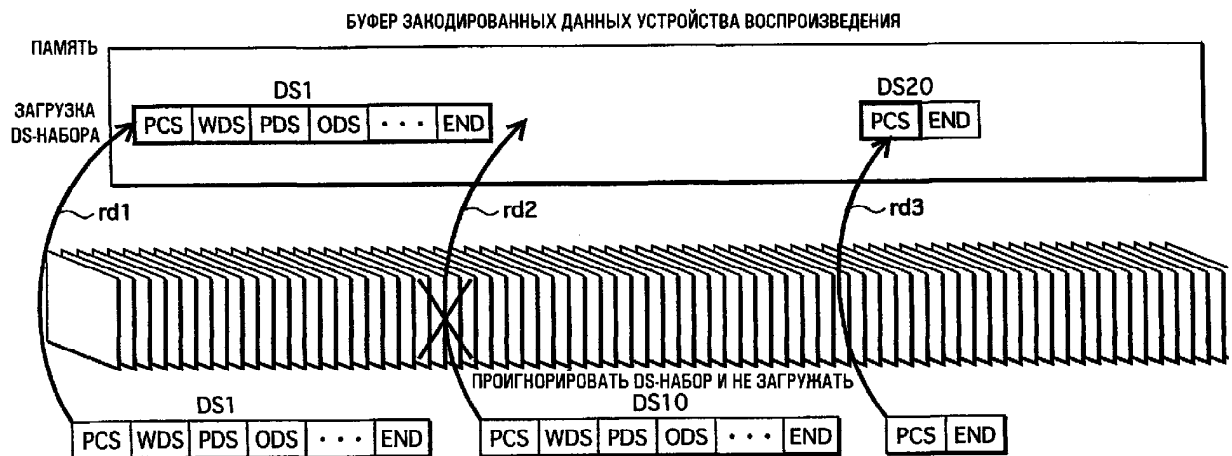
ФИГ.32



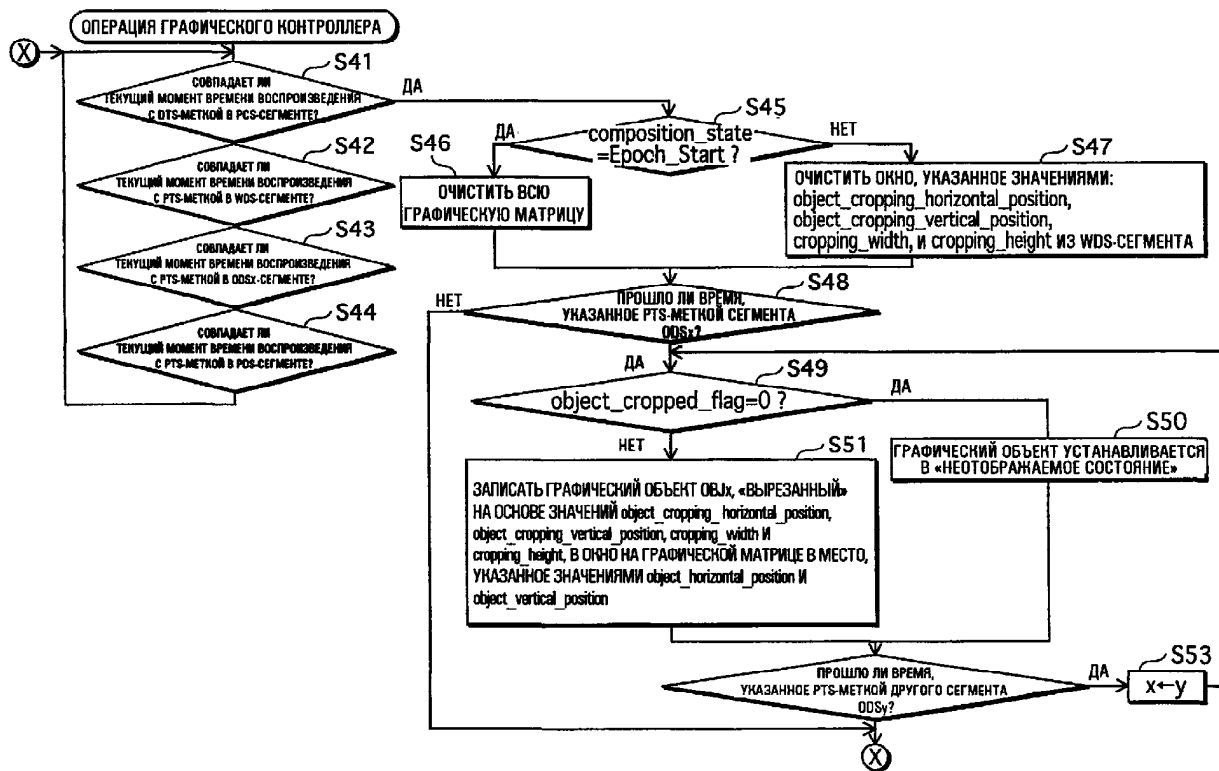
ФИГ.33



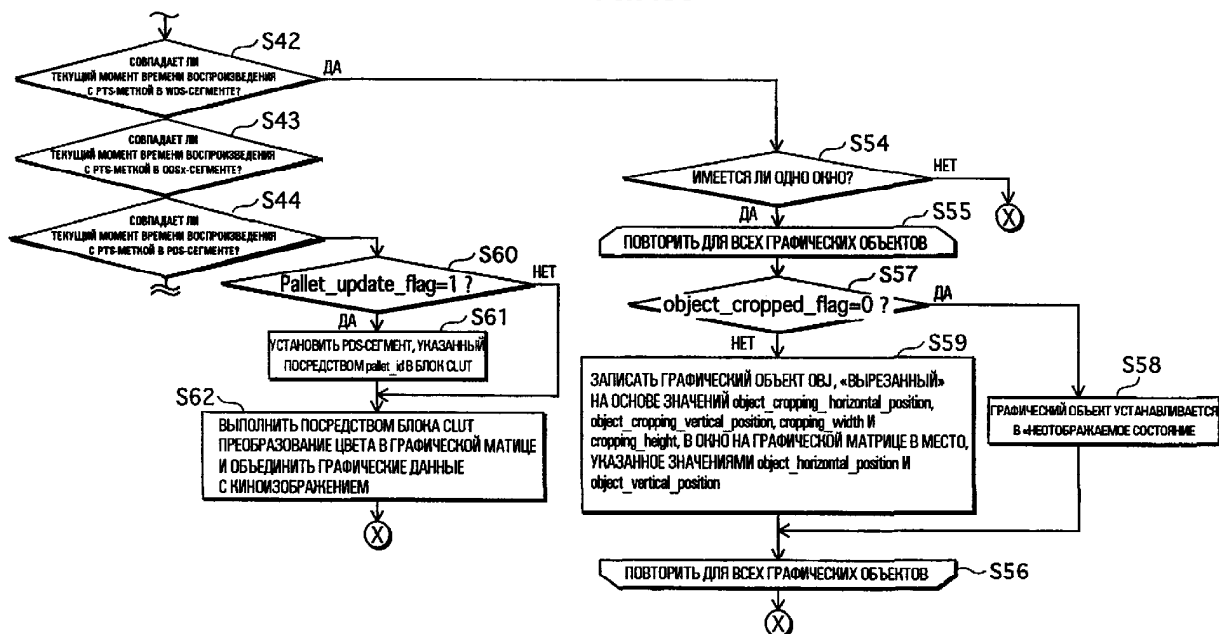
ФИГ.34

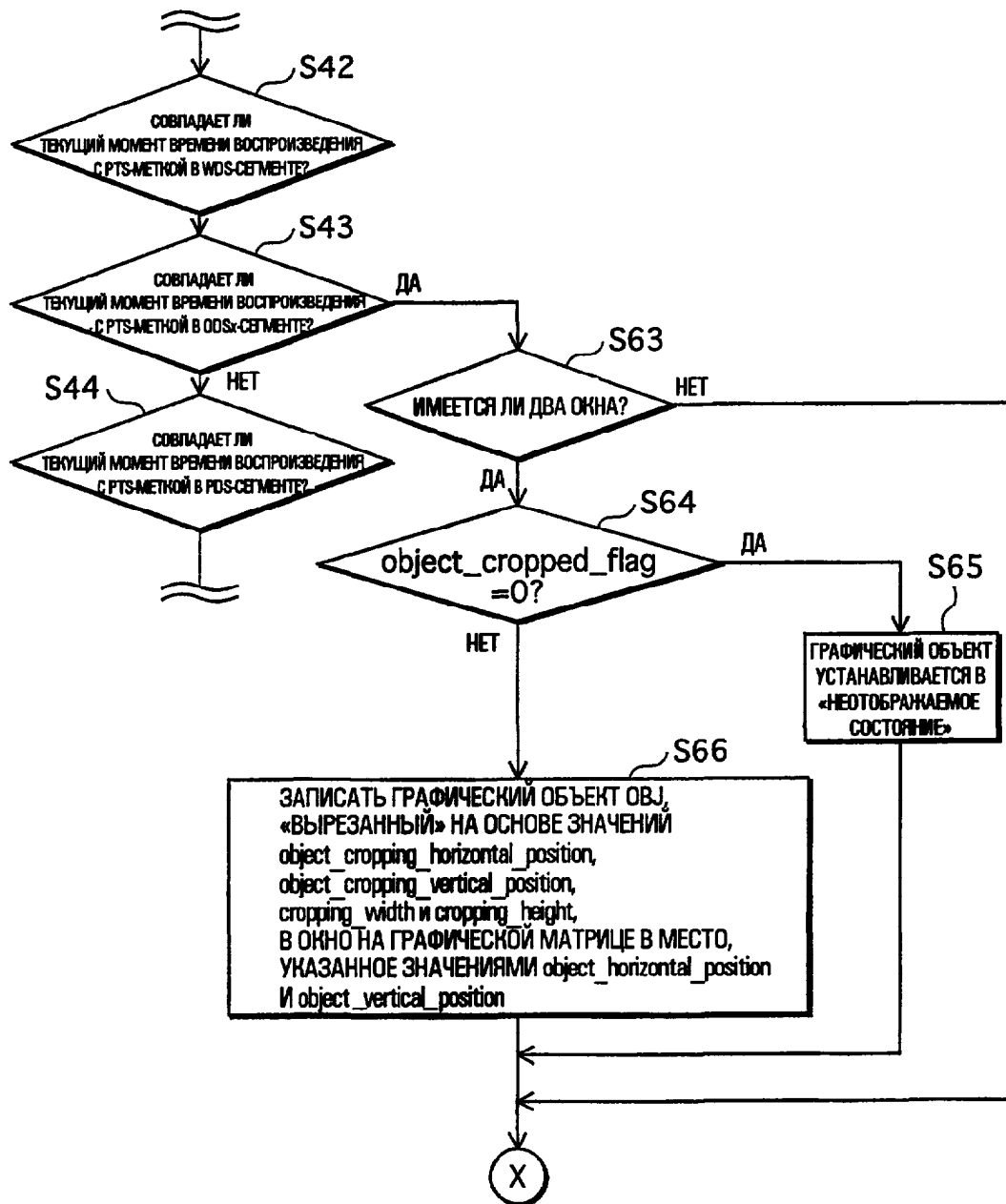


ФИГ.35

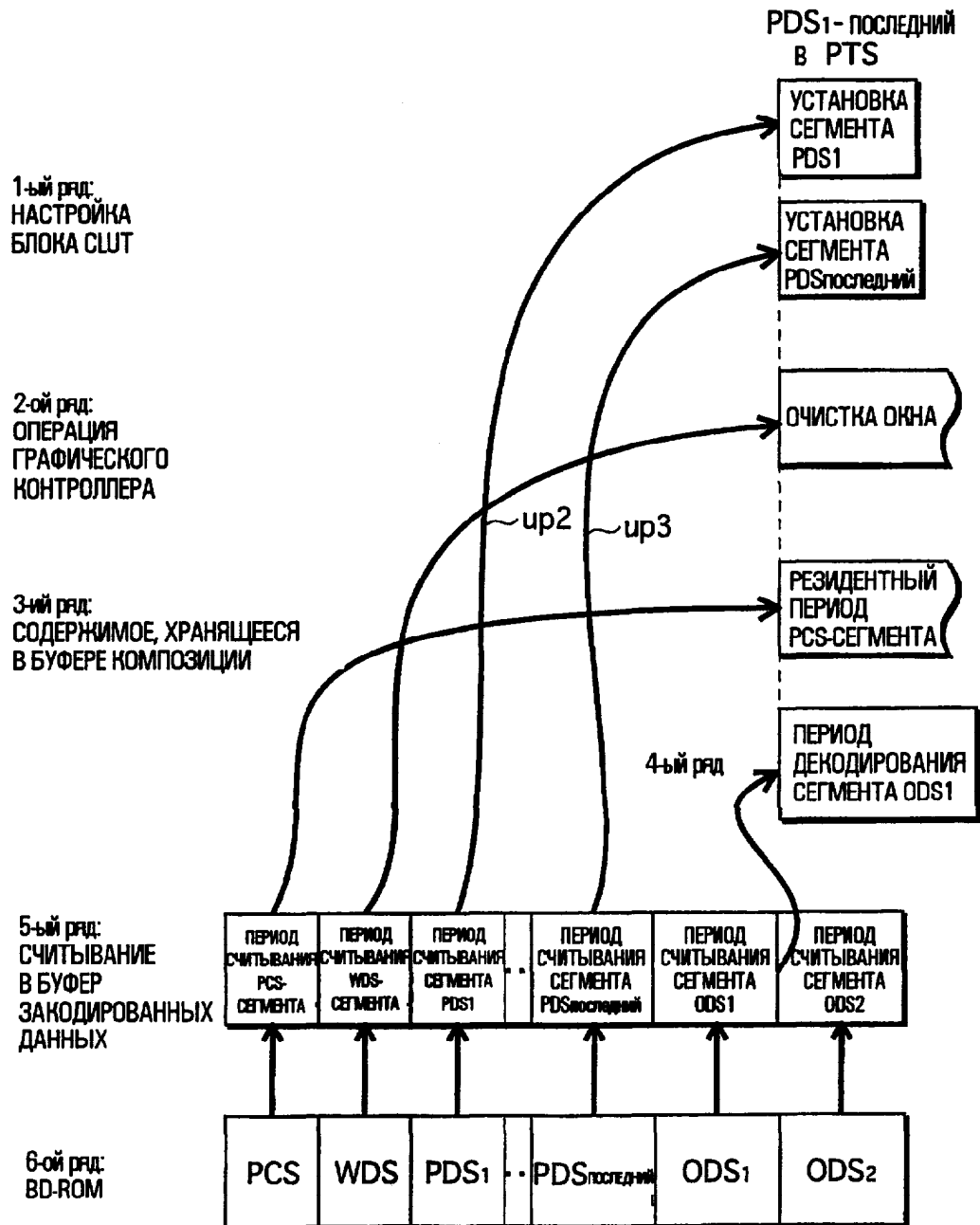


ФИГ.36

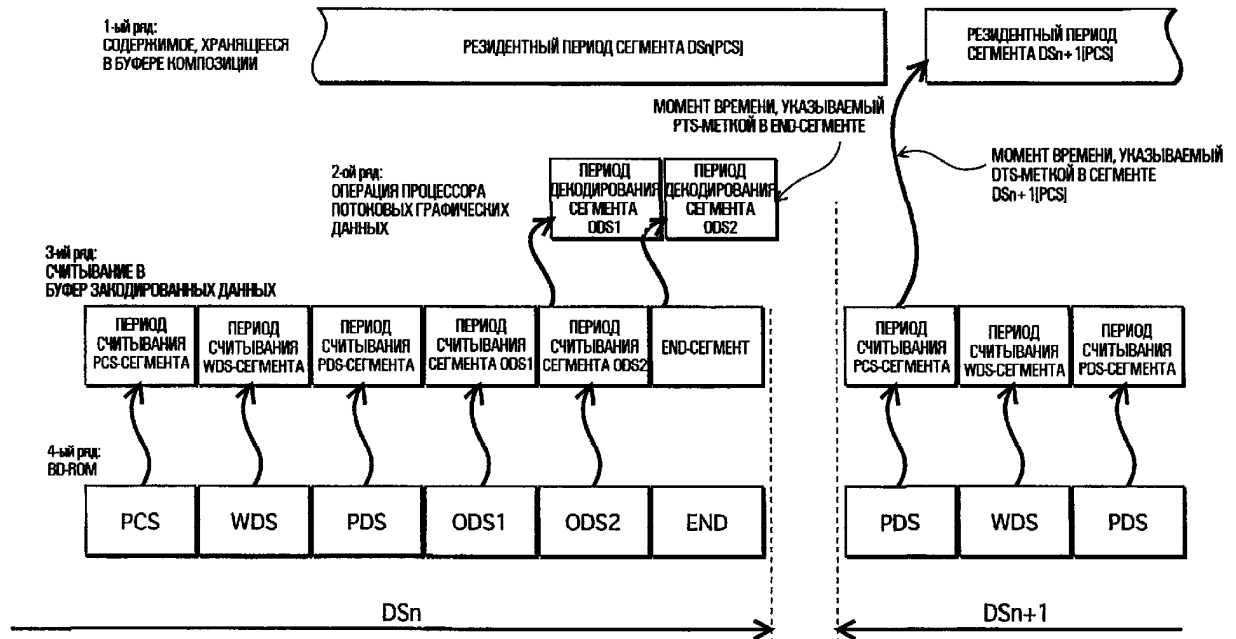




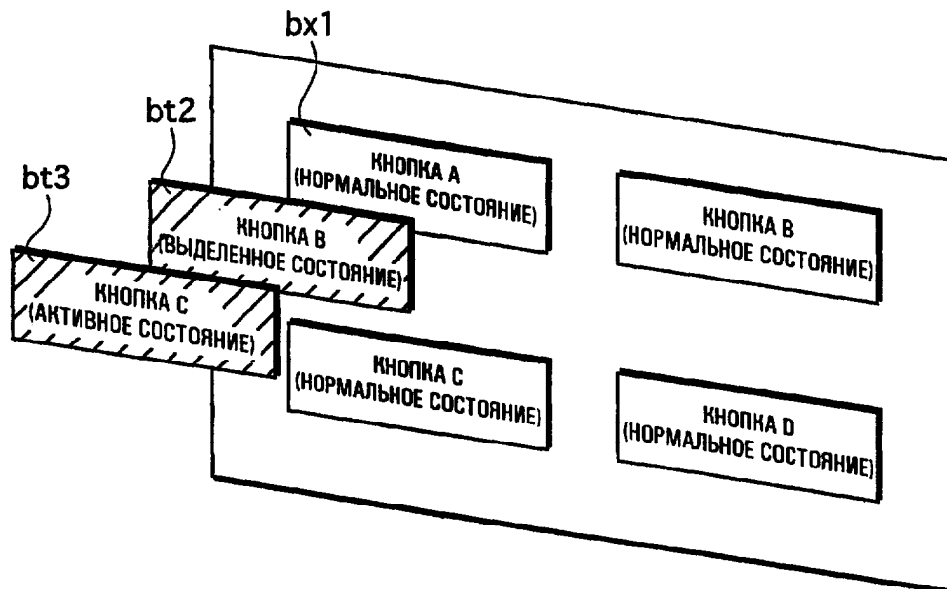
ФИГ.38



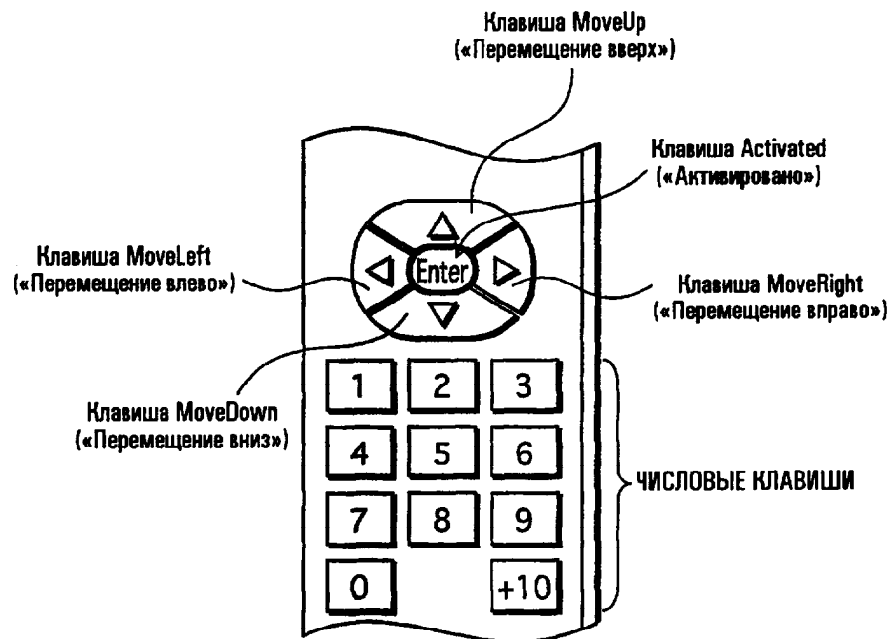
ФИГ.39



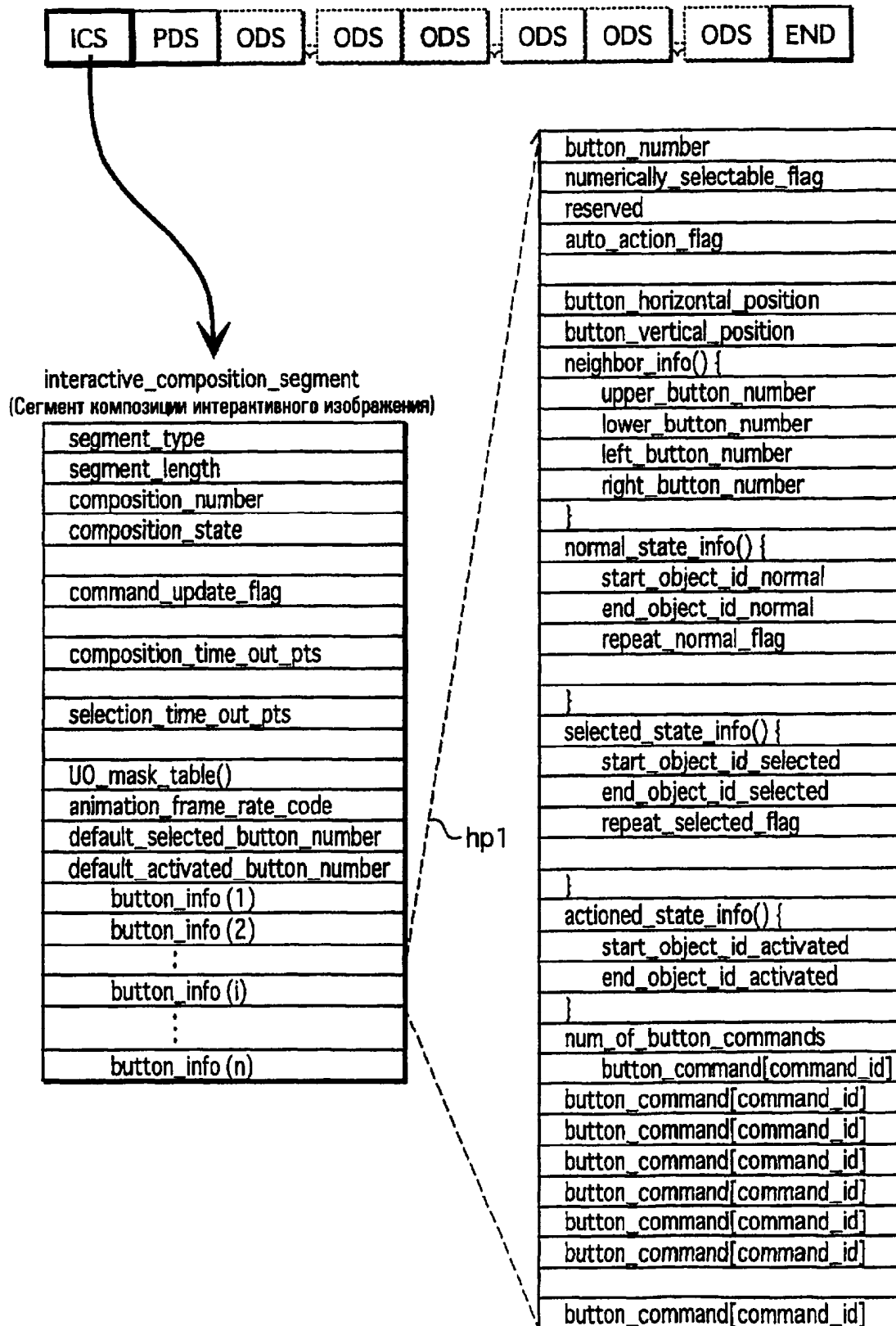
ФИГ.40



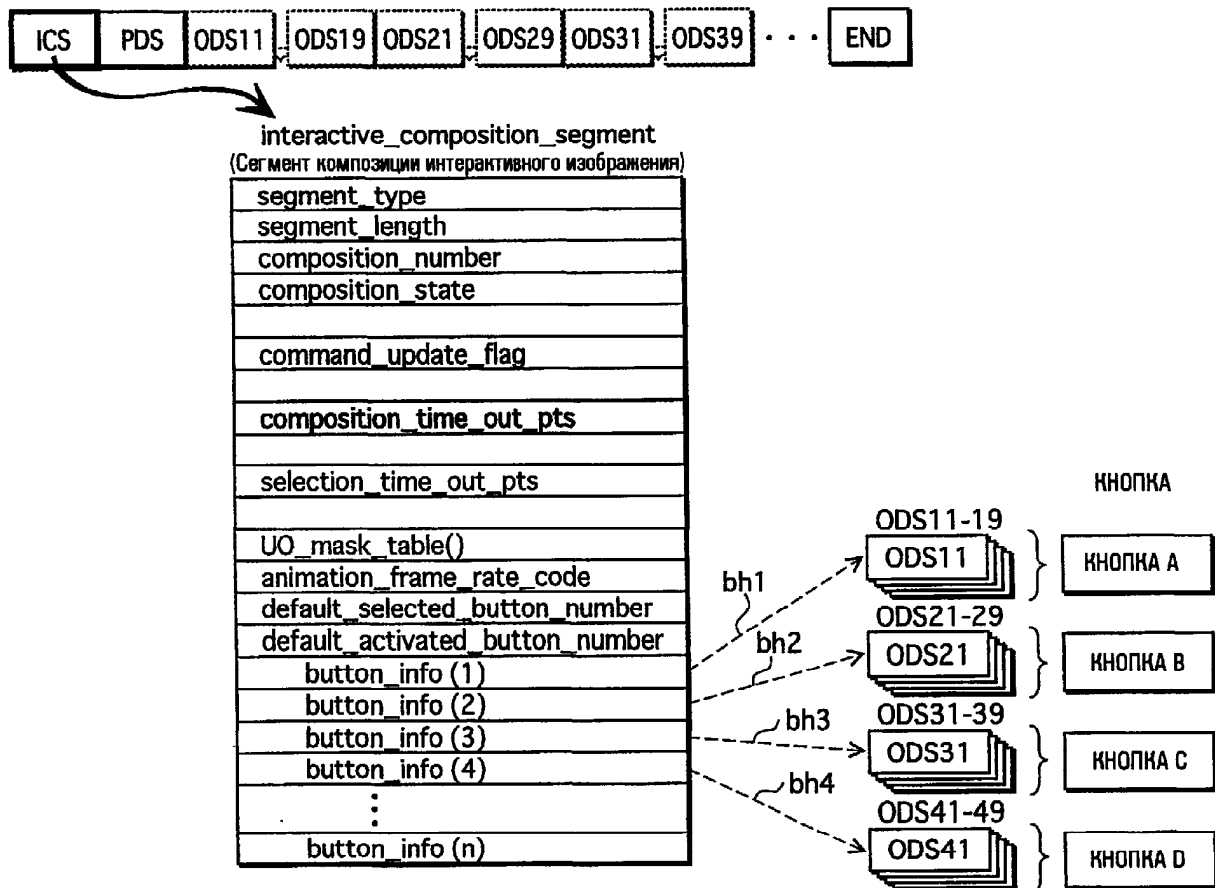
ФИГ.42А



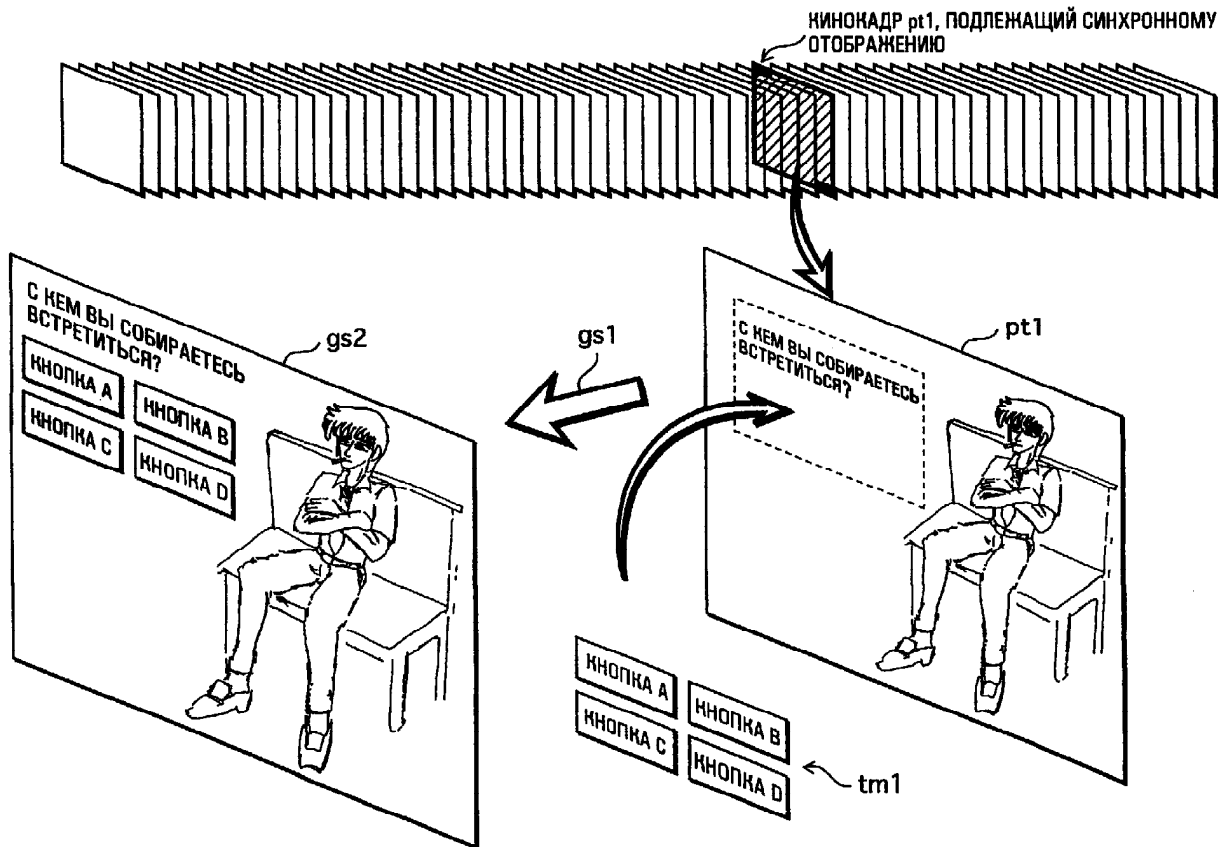
ФИГ.42В



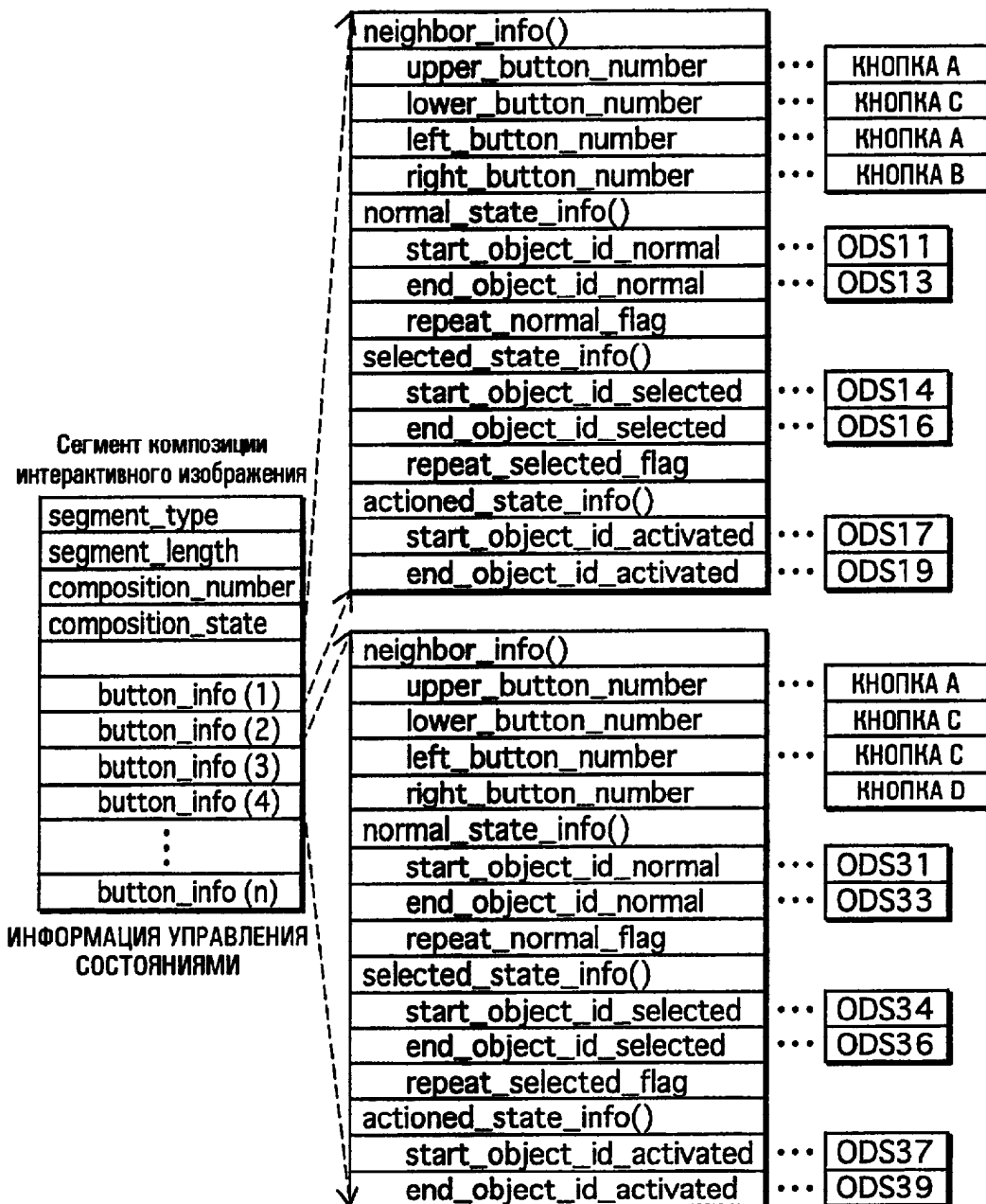
ФИГ.43



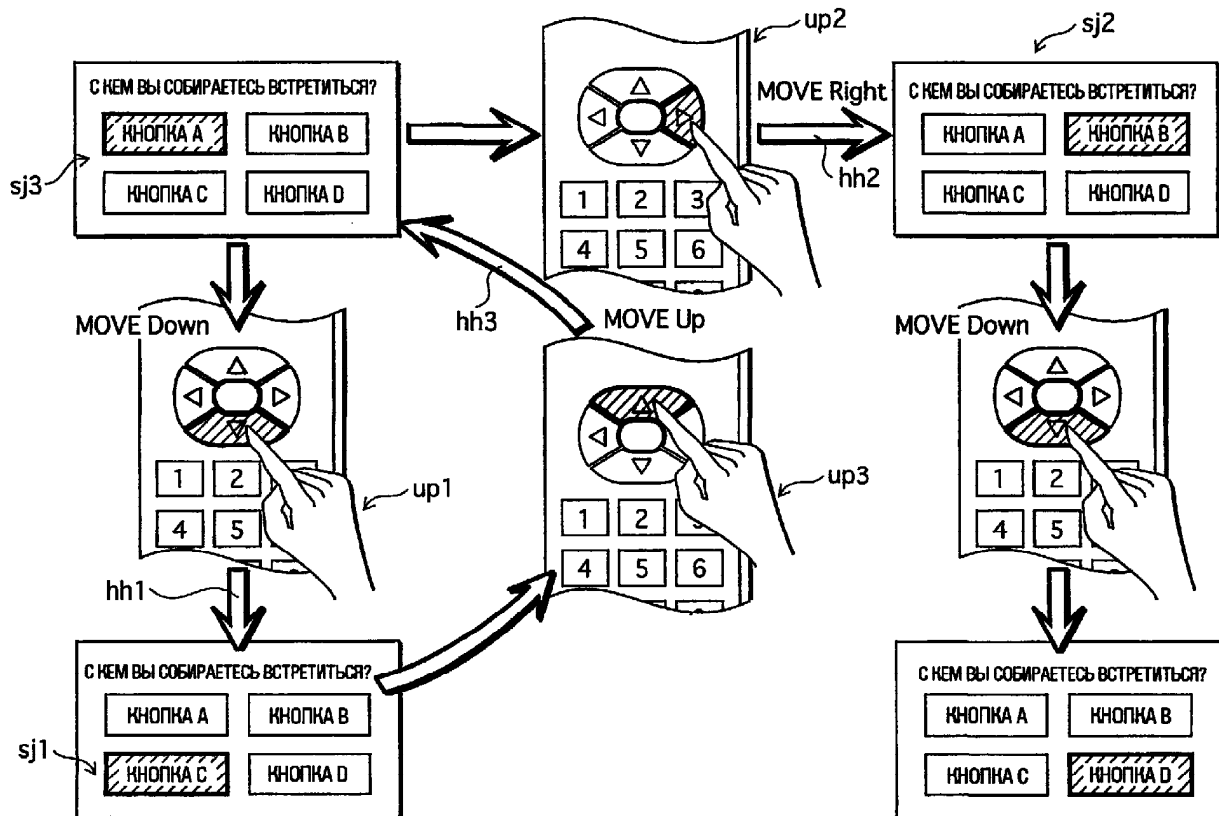
ФИГ.44



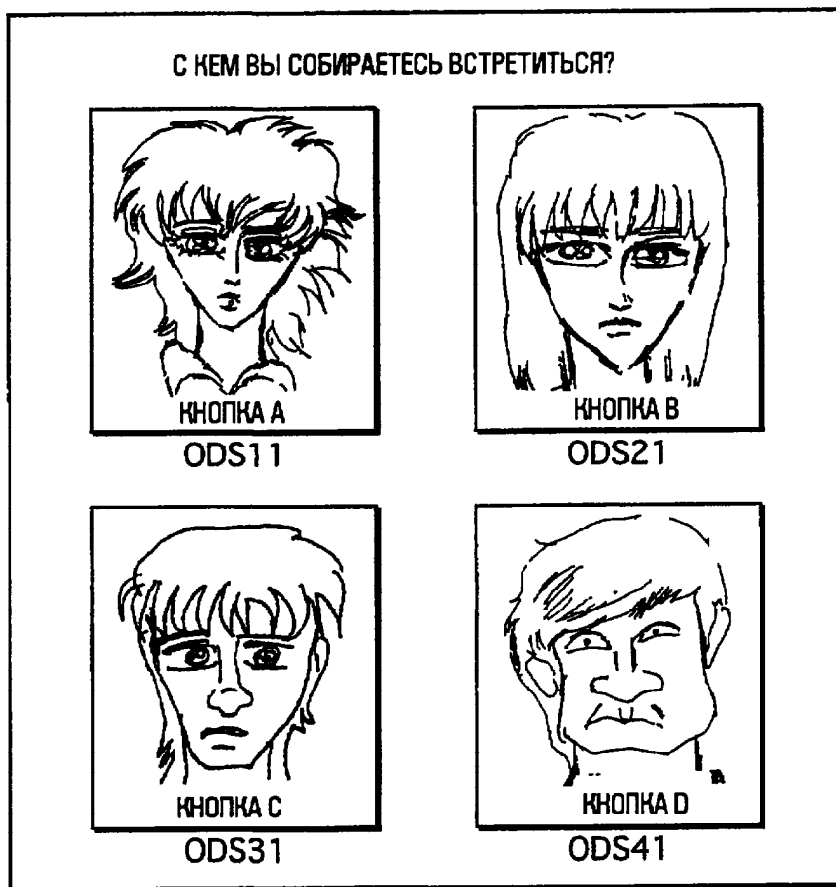
ФИГ.45



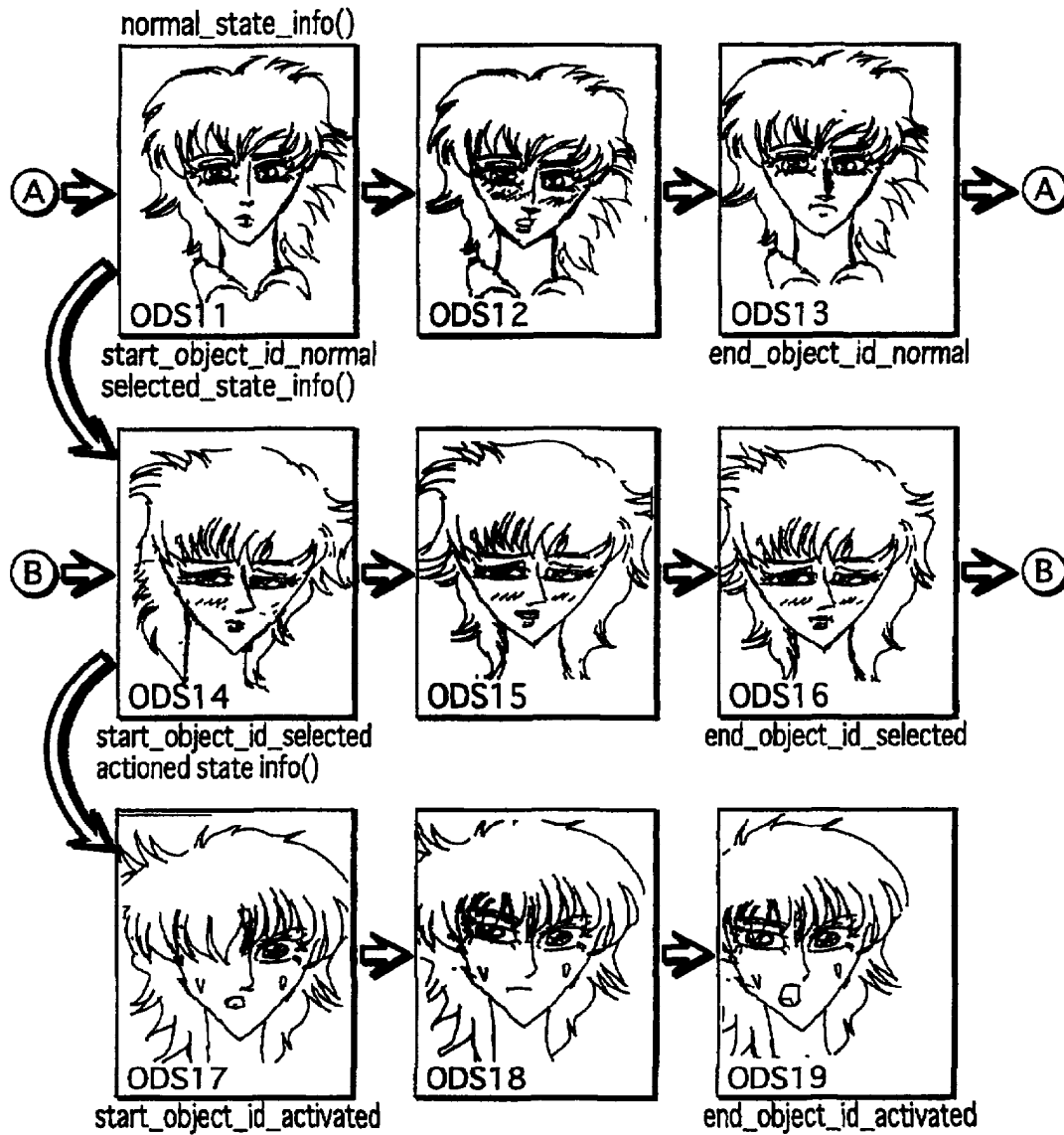
ФИГ.46



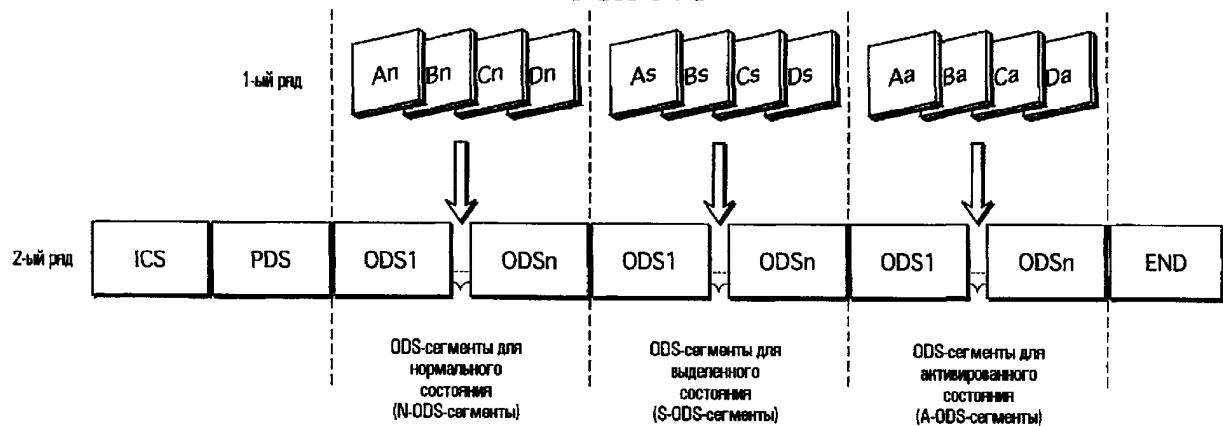
ФИГ.47



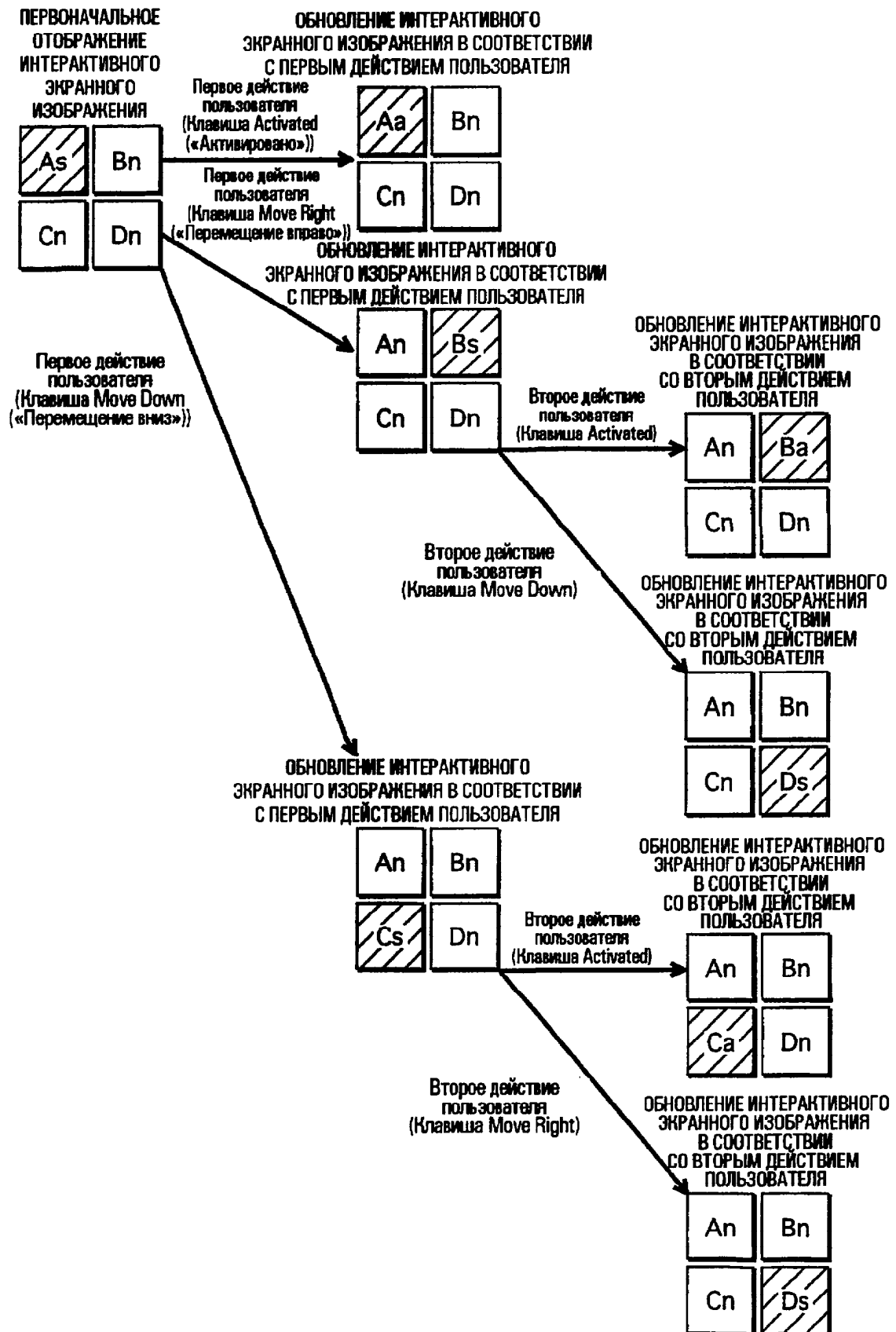
ФИГ.48



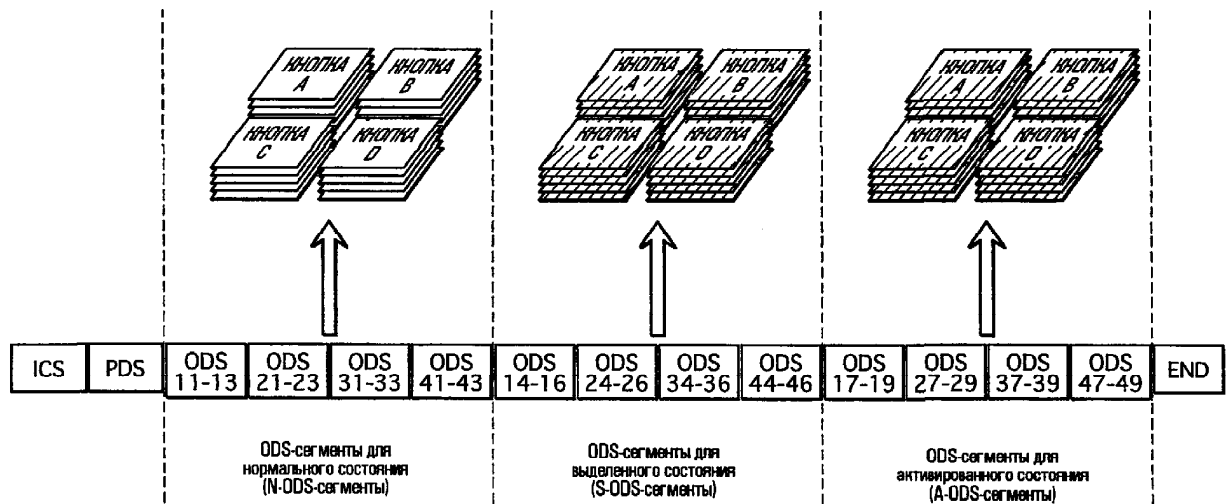
ФИГ.49



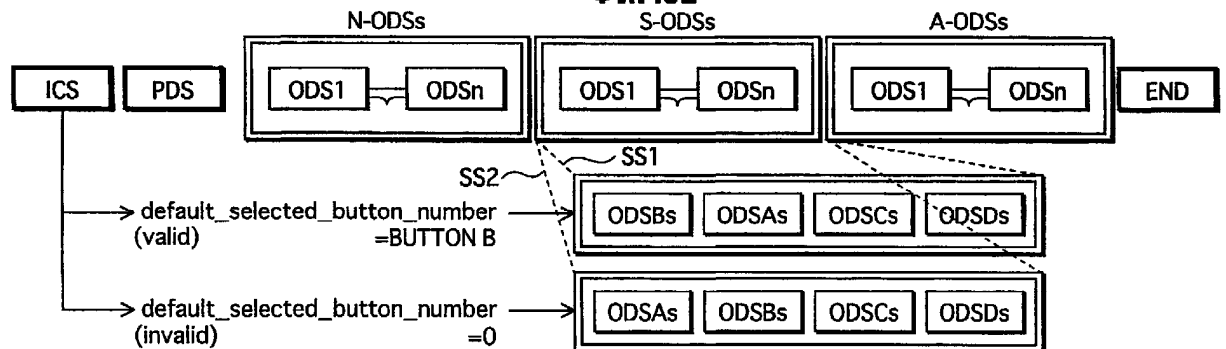
ФИГ.50



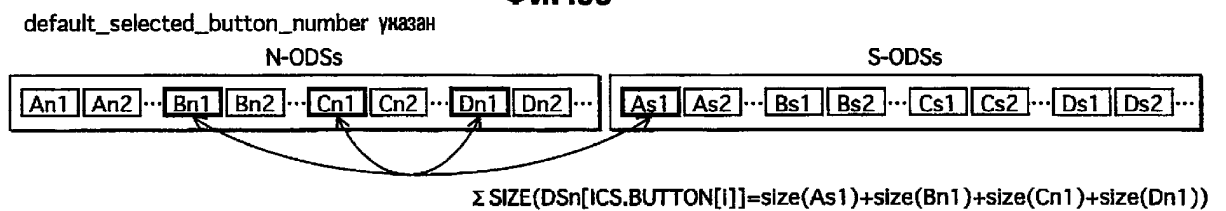
ФИГ.51



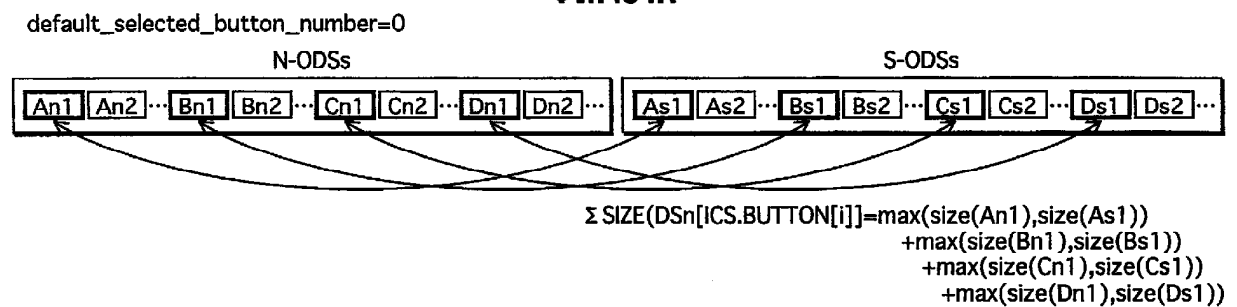
ФИГ.52



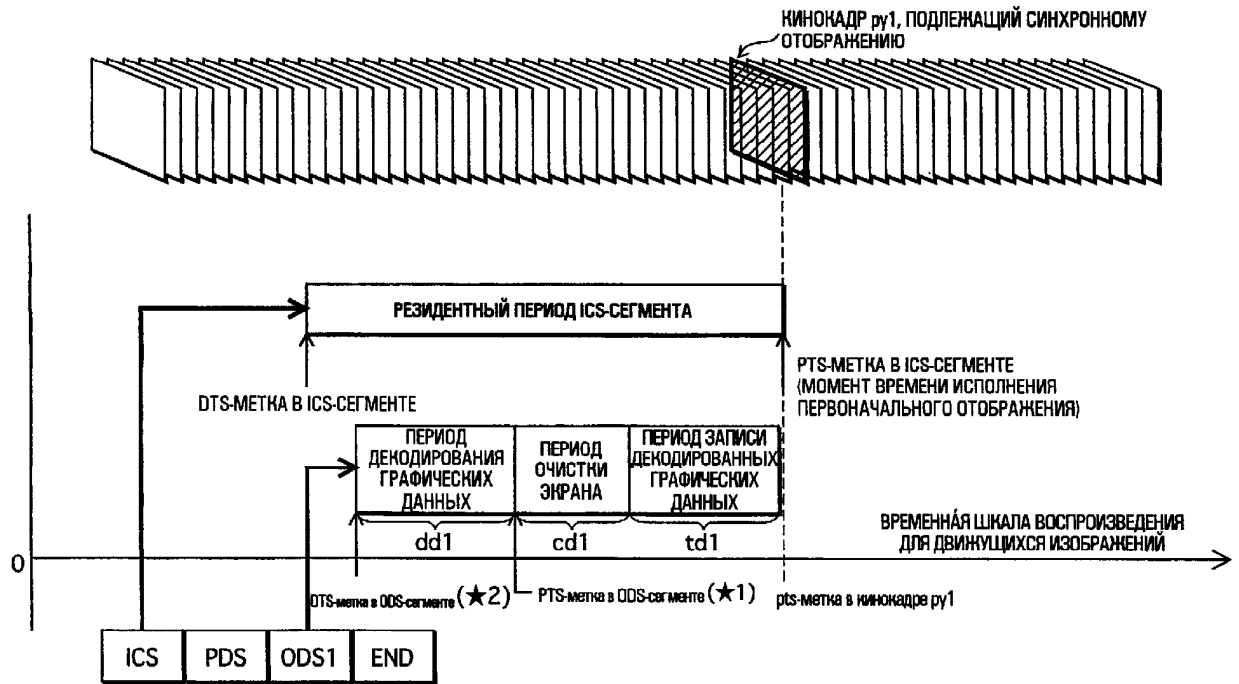
ФИГ.53



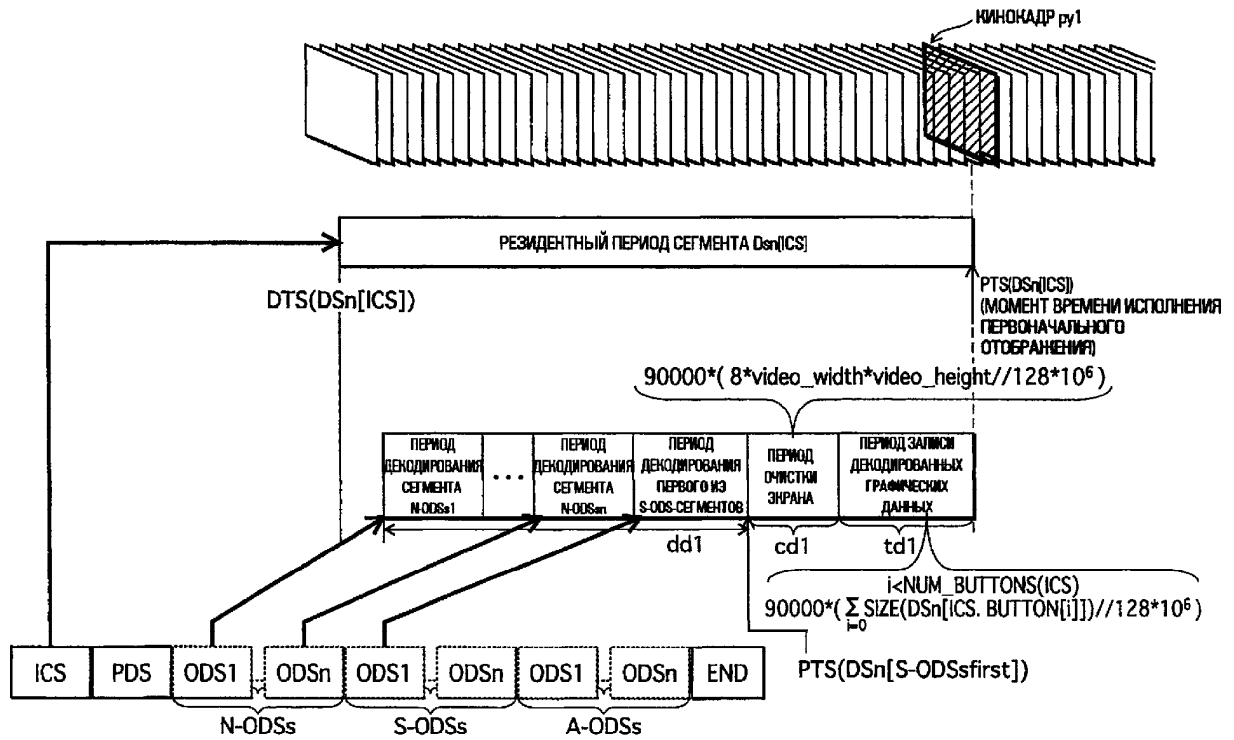
ФИГ.54A



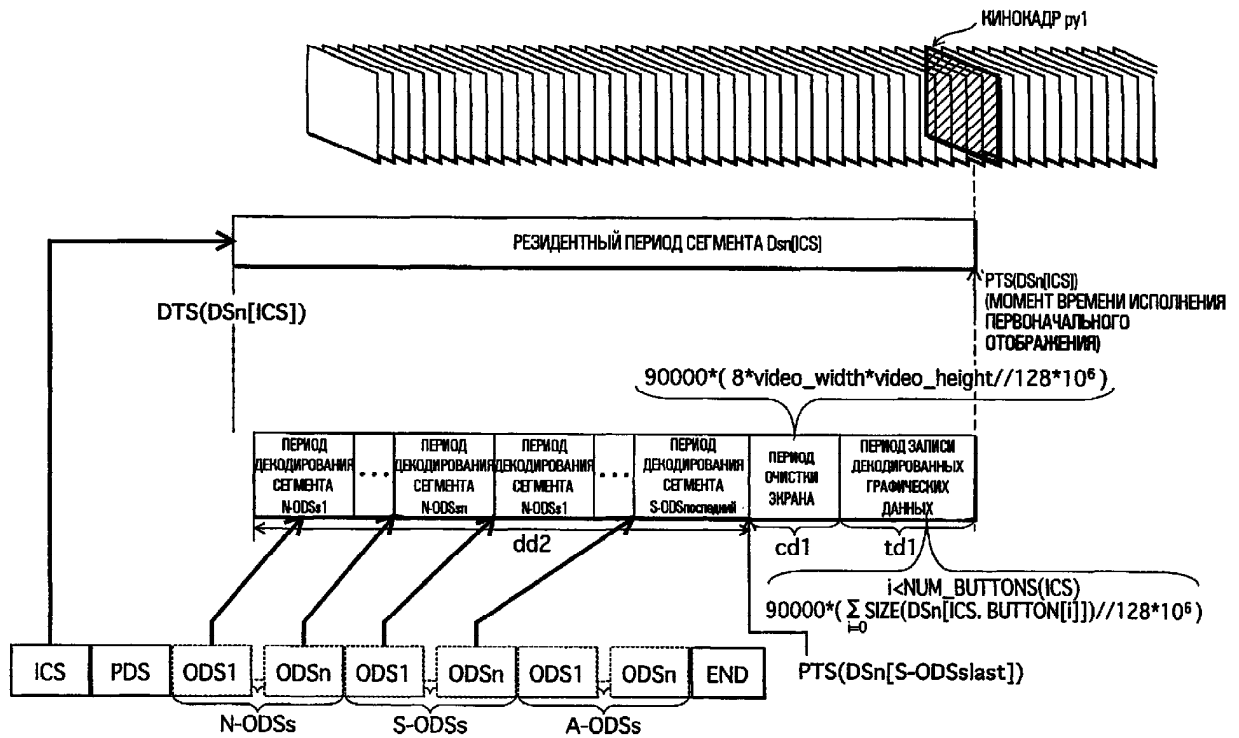
ФИГ.54B



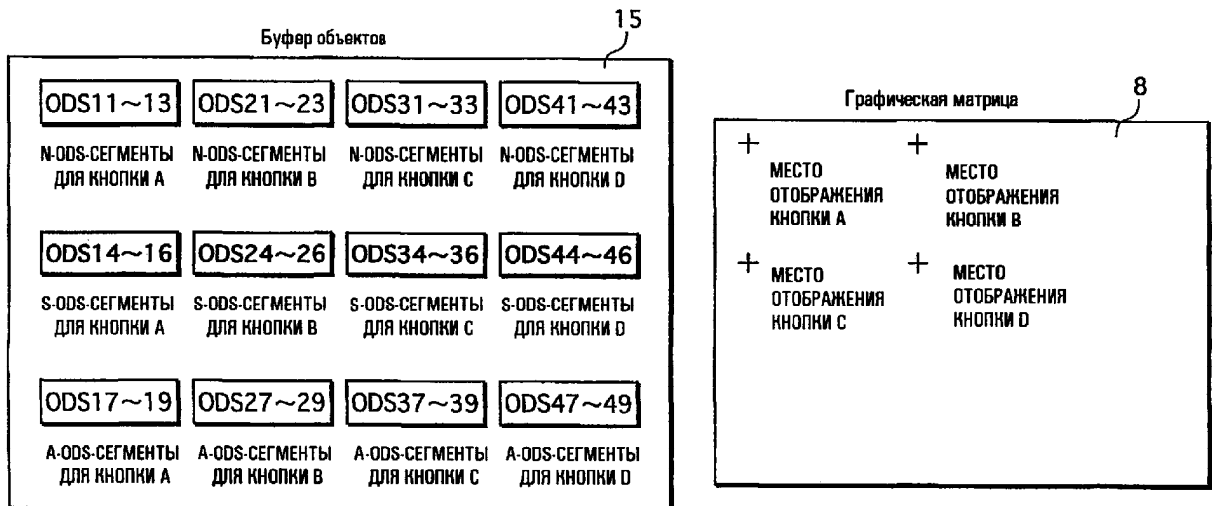
ФИГ.55



ФИГ.56



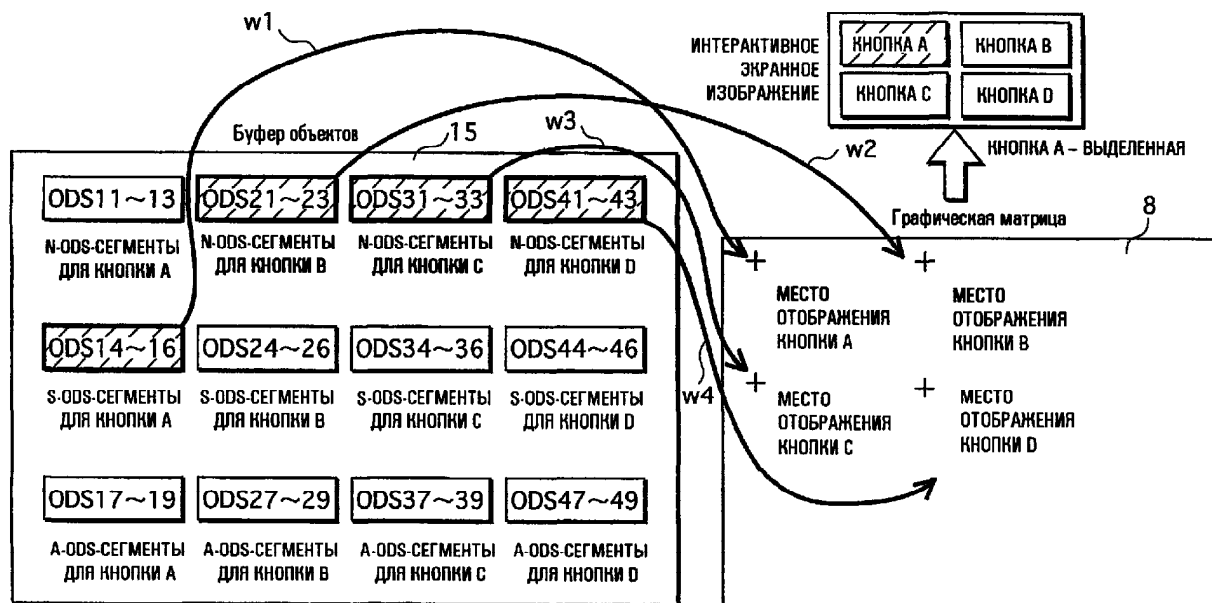
ФИГ.57



МЕСТО ОТОБРАЖЕНИЯ КНОПКИ – МЕСТО ОТОБРАЖЕНИЯ КНОПКИ,
ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЯМИ button_horizontal_position,
button_vertical_position ИЗ СОСТАВА ИНФОРМАЦИИ О КНОПКЕ

ФИГ.58

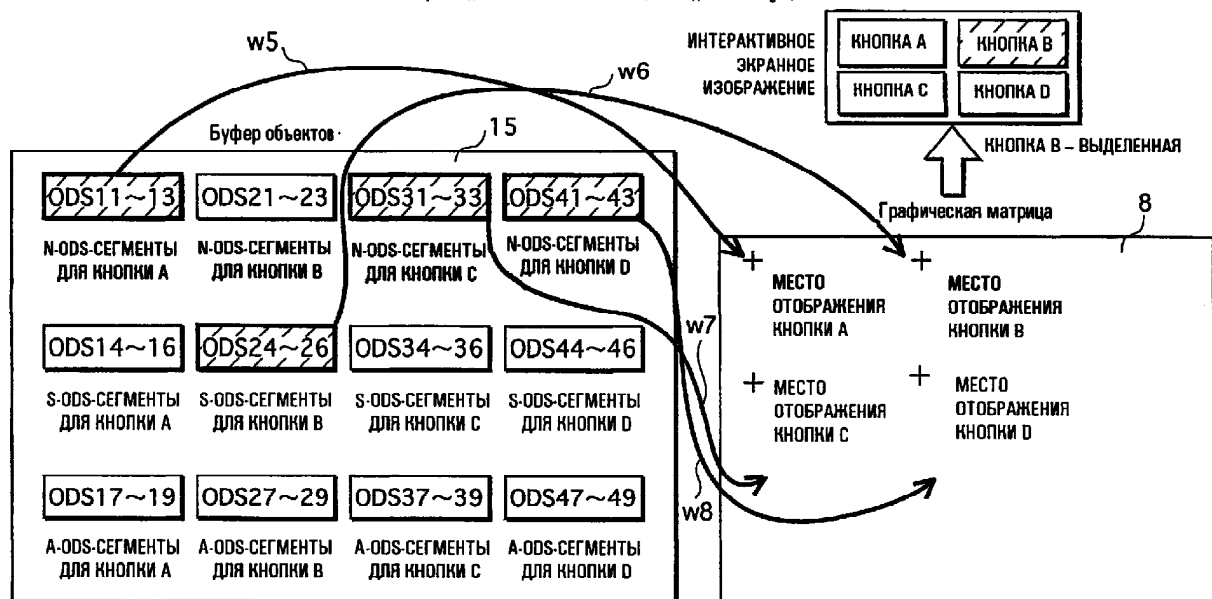
ОПЕРАЦИЯ ЗАПИСИ, ВЫПОЛНЯЕМАЯ ГРАФИЧЕСКИМ КОНТРОЛЛЕРОМ ПРИ ПЕРВОНАЧАЛЬНОМ ОТОБРАЖЕНИИ



МЕСТО ОТОБРАЖЕНИЯ КНОПКИ - МЕСТО ОТОБРАЖЕНИЯ КНОПКИ, ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЯМИ `button_horizontal_position`, `button_vertical_position` ИЗ СОСТАВА ИНФОРМАЦИИ О КНОПКЕ

ФИГ.59

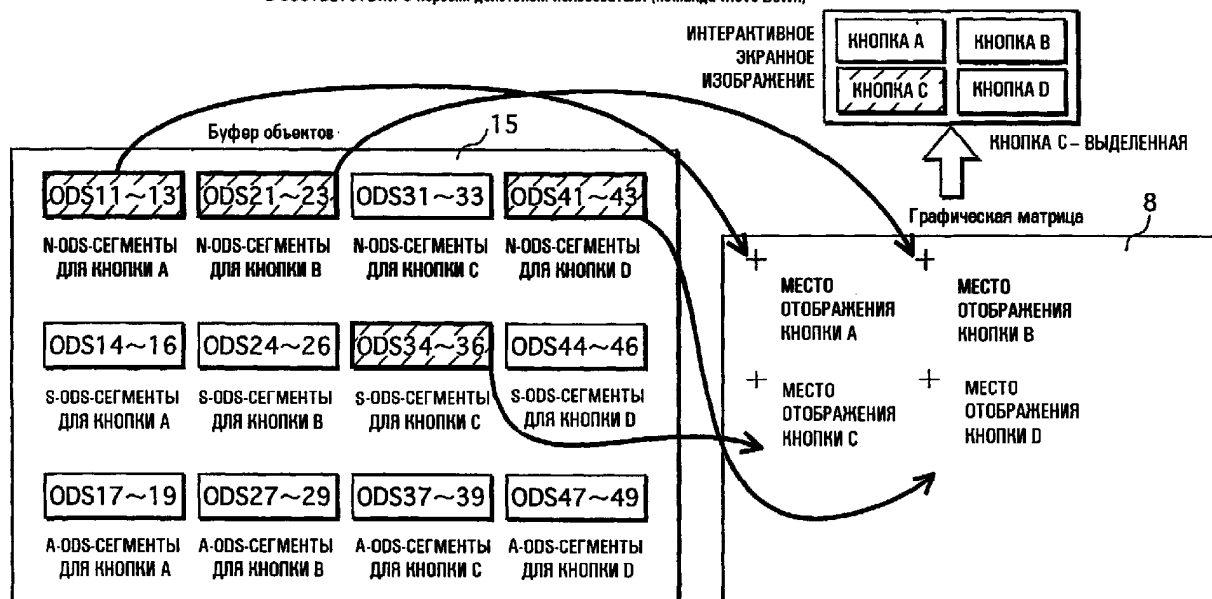
ОПЕРАЦИЯ ЗАПИСИ, ВЫПОЛНЯЕМАЯ ГРАФИЧЕСКИМ КОНТРОЛЛЕРОМ ПРИ ОБНОВЛЕНИИ ИНТЕРАКТИВНОГО ЭКРАННОГО ИЗОБРАЖЕНИЯ В СООТВЕТСТВИИ С ПЕРВЫМ ДЕЙСТВИЕМ ПОЛЬЗОВАТЕЛЯ (команда `Move Right`)



МЕСТО ОТОБРАЖЕНИЯ КНОПКИ - МЕСТО ОТОБРАЖЕНИЯ КНОПКИ, ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЯМИ `button_horizontal_position`, `button_vertical_position` ИЗ СОСТАВА ИНФОРМАЦИИ О КНОПКЕ

ФИГ.60

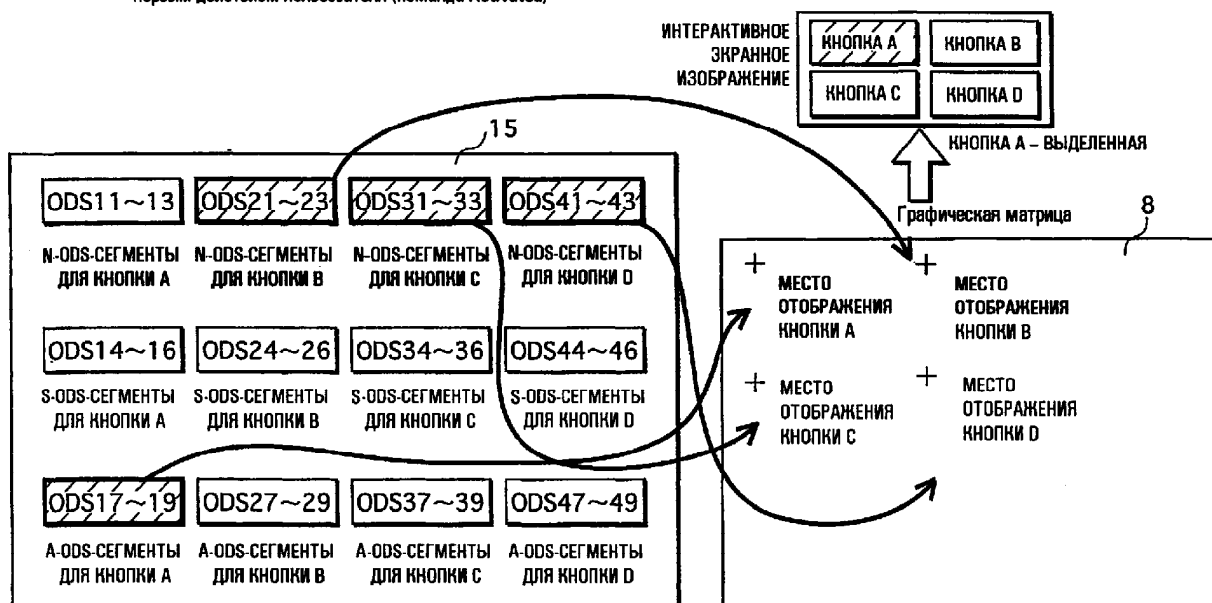
ОПЕРАЦИЯ ЗАПИСИ, ВЫПОЛНЯЕМАЯ ГРАФИЧЕСКИМ КОНТРОЛЛЕРОМ
ПРИ ОБНОВЛЕНИИ ИНТЕРАКТИВНОГО ЭКРАННОГО ИЗОБРАЖЕНИЯ
В СООТВЕТСТВИИ С ПЕРВЫМ ДЕЙСТВИЕМ ПОЛЬЗОВАТЕЛЯ (команда Move Down)



МЕСТО ОТОБРАЖЕНИЯ КНОПКИ – МЕСТО ОТОБРАЖЕНИЯ КНОПКИ,
ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЯМИ button_horizontal_position,
button_vertical_position ИЗ СОСТАВА ИНФОРМАЦИИ О КНОПКЕ

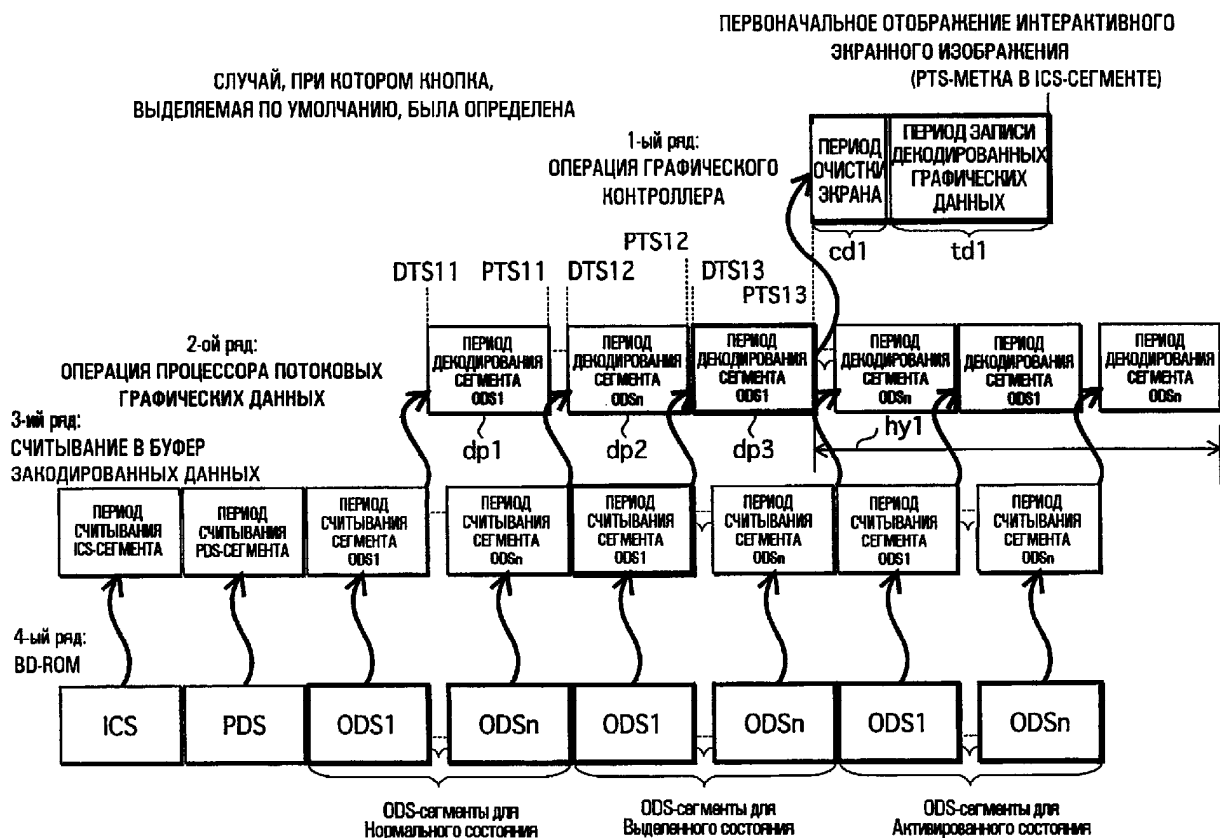
ФИГ.61

ОПЕРАЦИЯ ЗАПИСИ, ВЫПОЛНЯЕМАЯ ГРАФИЧЕСКИМ КОНТРОЛЛЕРОМ ПРИ
ОБНОВЛЕНИИ ИНТЕРАКТИВНОГО ЭКРАННОГО ИЗОБРАЖЕНИЯ В СООТВЕТСТВИИ С
ПЕРВЫМ ДЕЙСТВИЕМ ПОЛЬЗОВАТЕЛЯ (команда Activated)

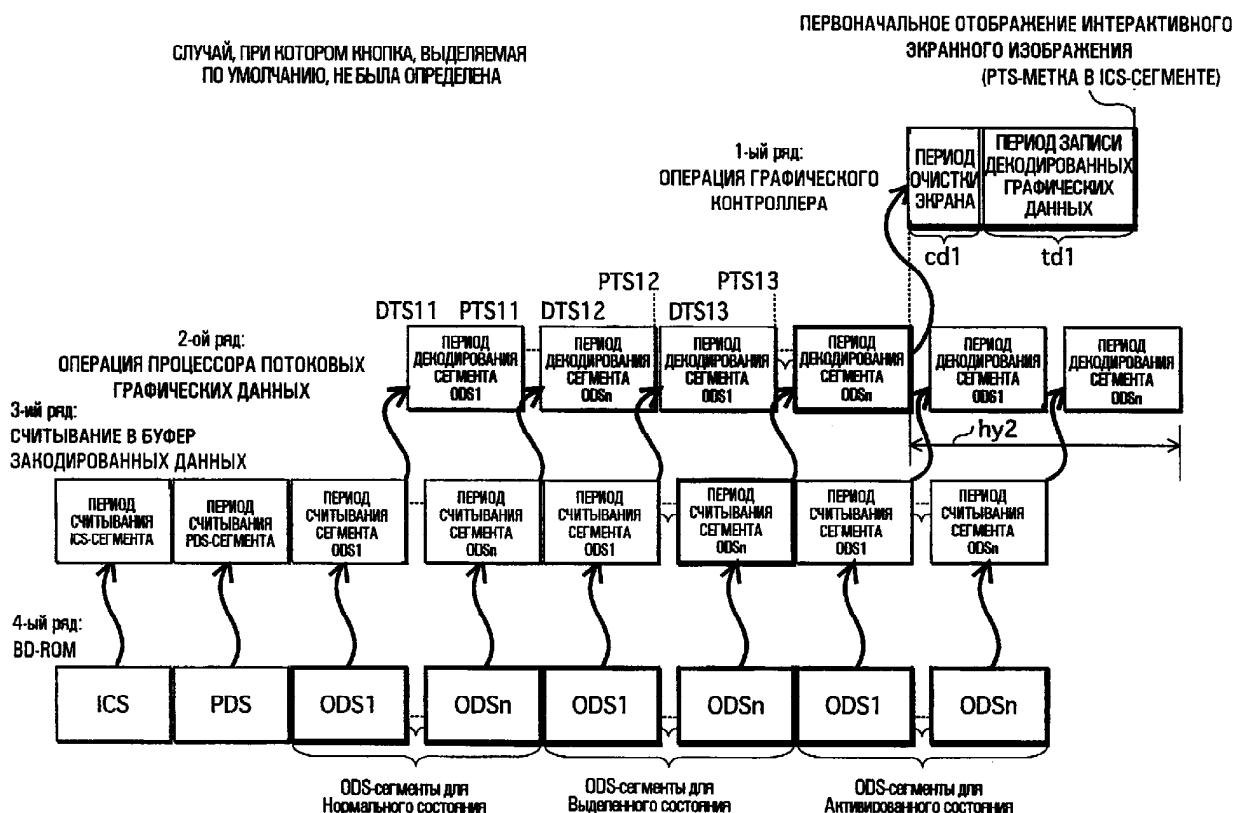


МЕСТО ОТОБРАЖЕНИЯ КНОПКИ – МЕСТО ОТОБРАЖЕНИЯ КНОПКИ,
ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЯМИ button_horizontal_position,
button_vertical_position ИЗ СОСТАВА ИНФОРМАЦИИ О КНОПКЕ

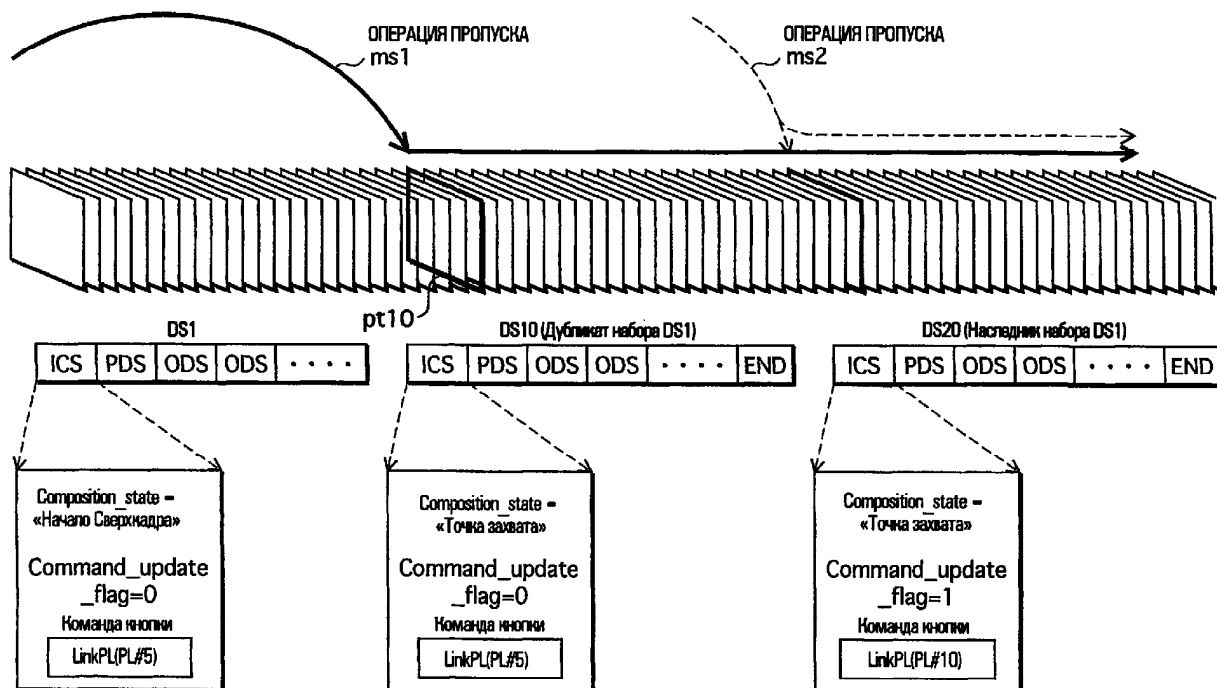
ФИГ.62



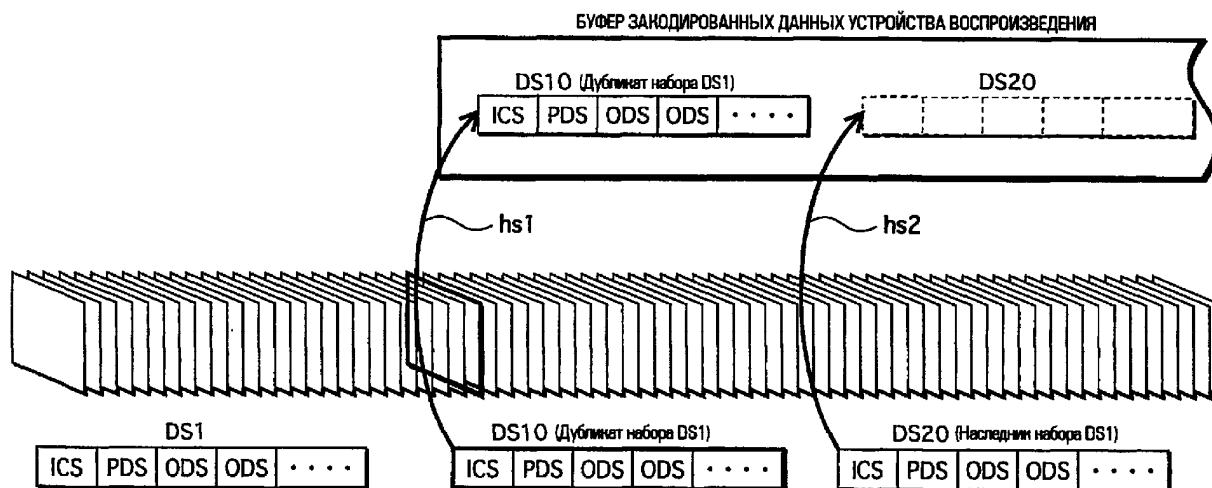
ФИГ.63



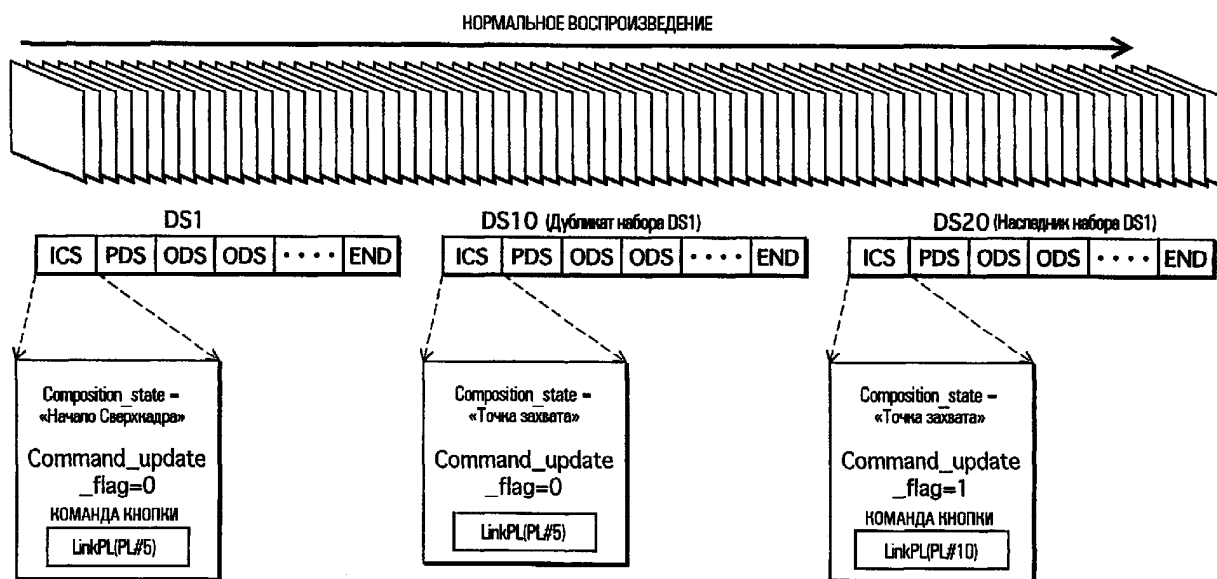
ФИГ.64



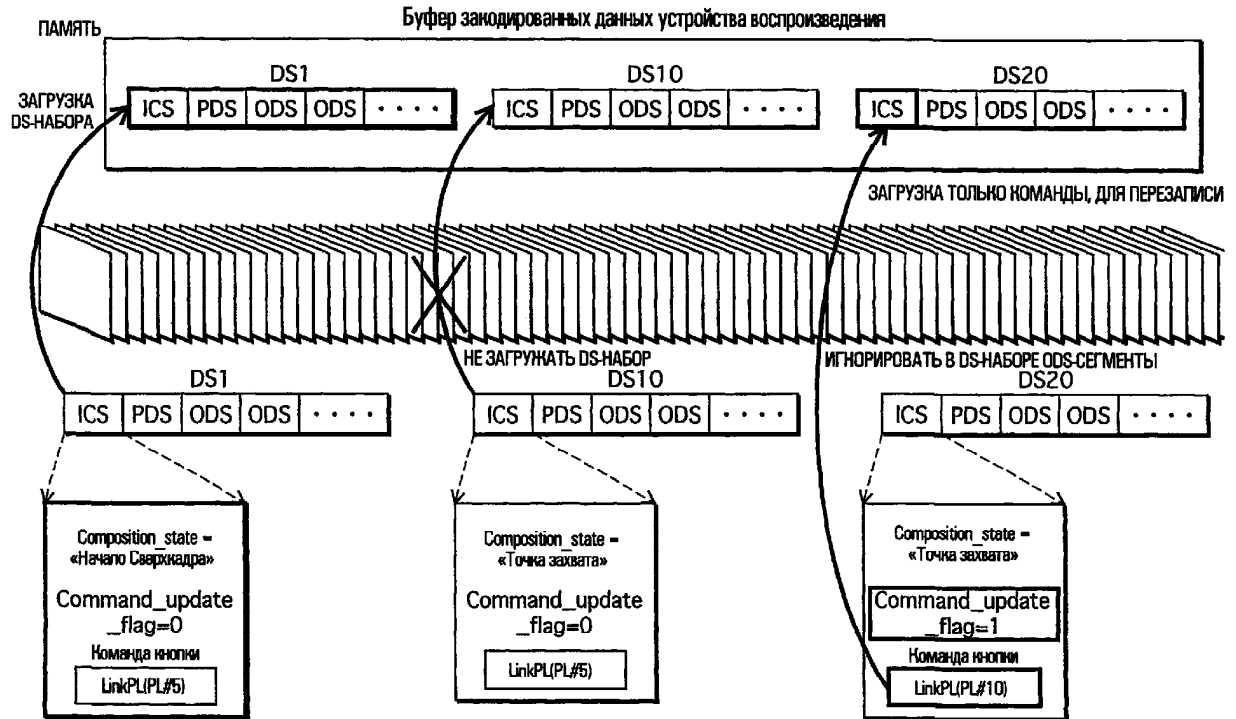
ФИГ.67



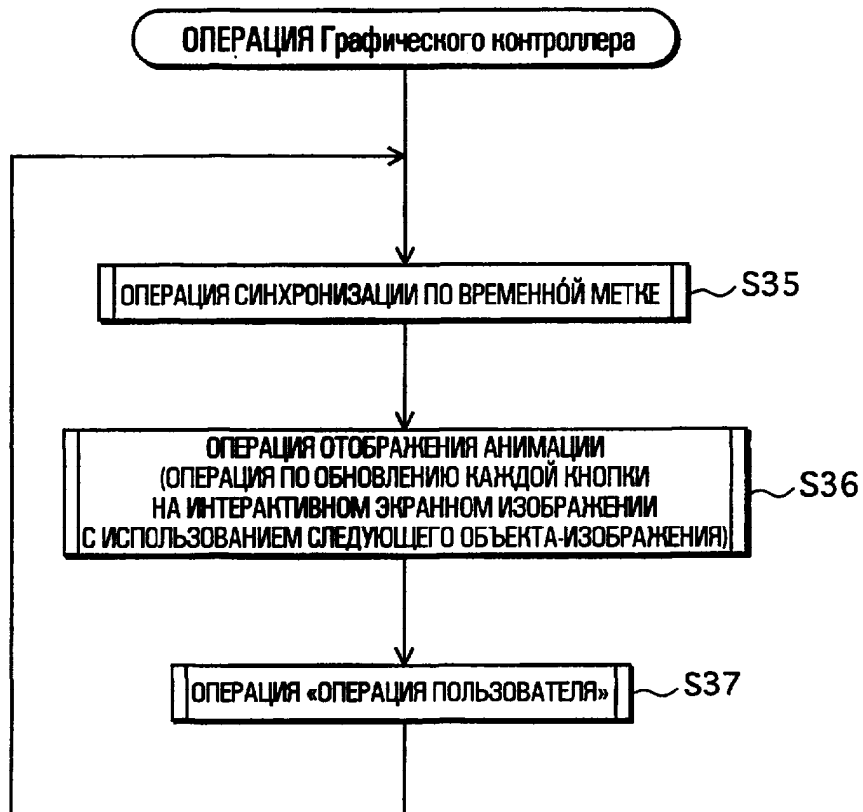
ФИГ.68



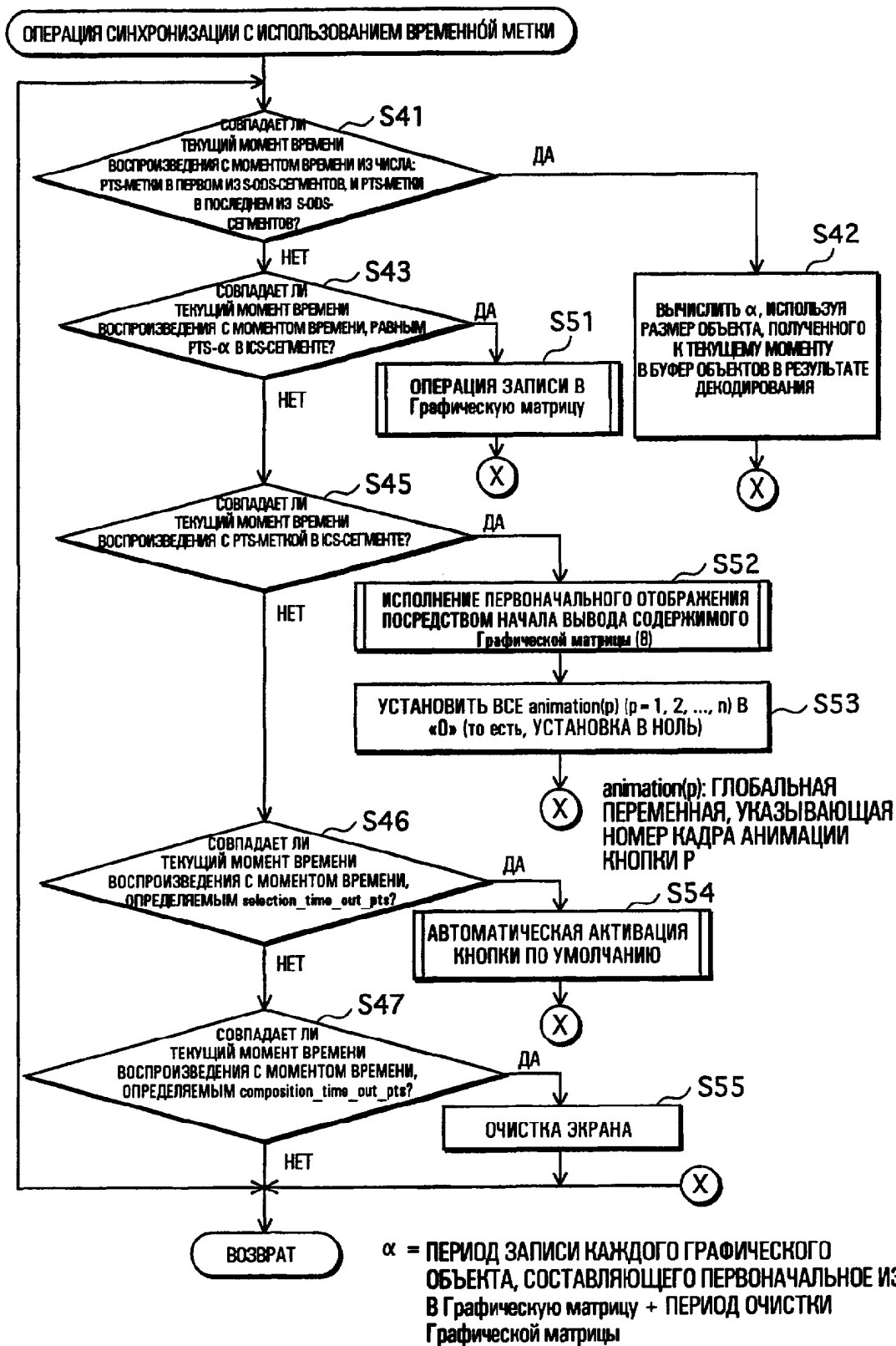
ФИГ.69



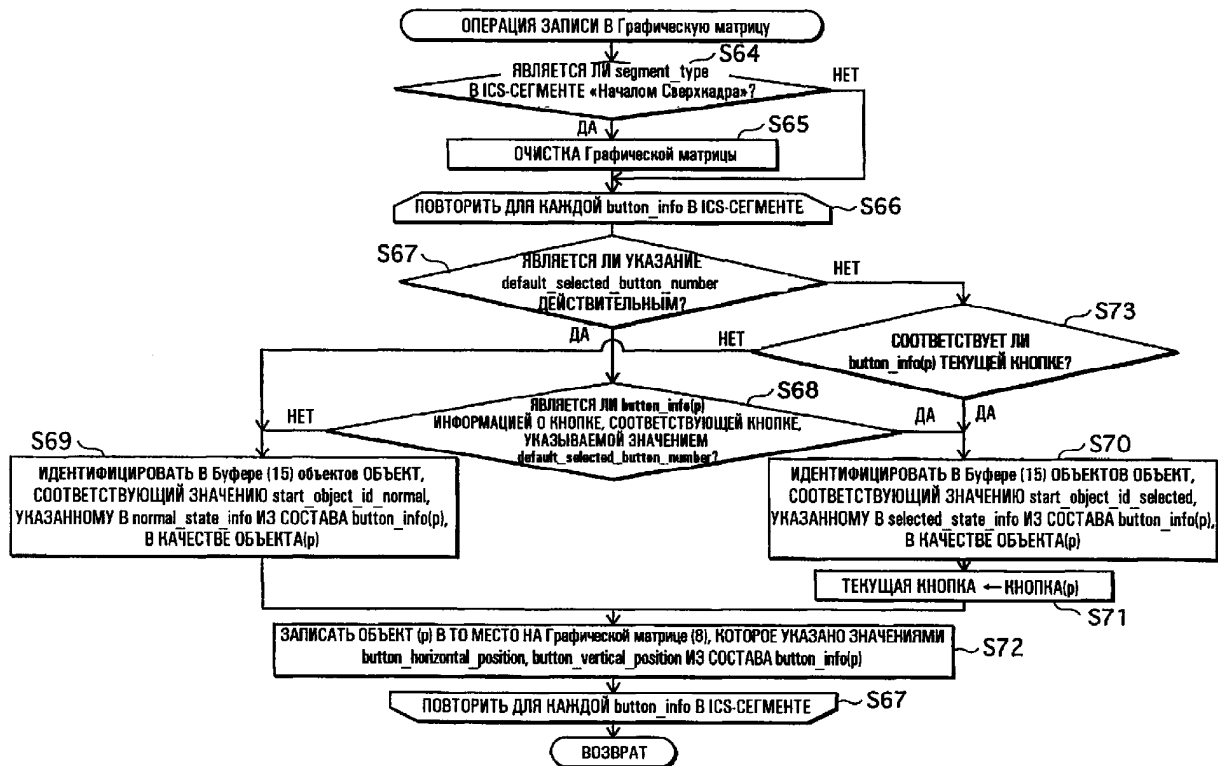
ФИГ.70



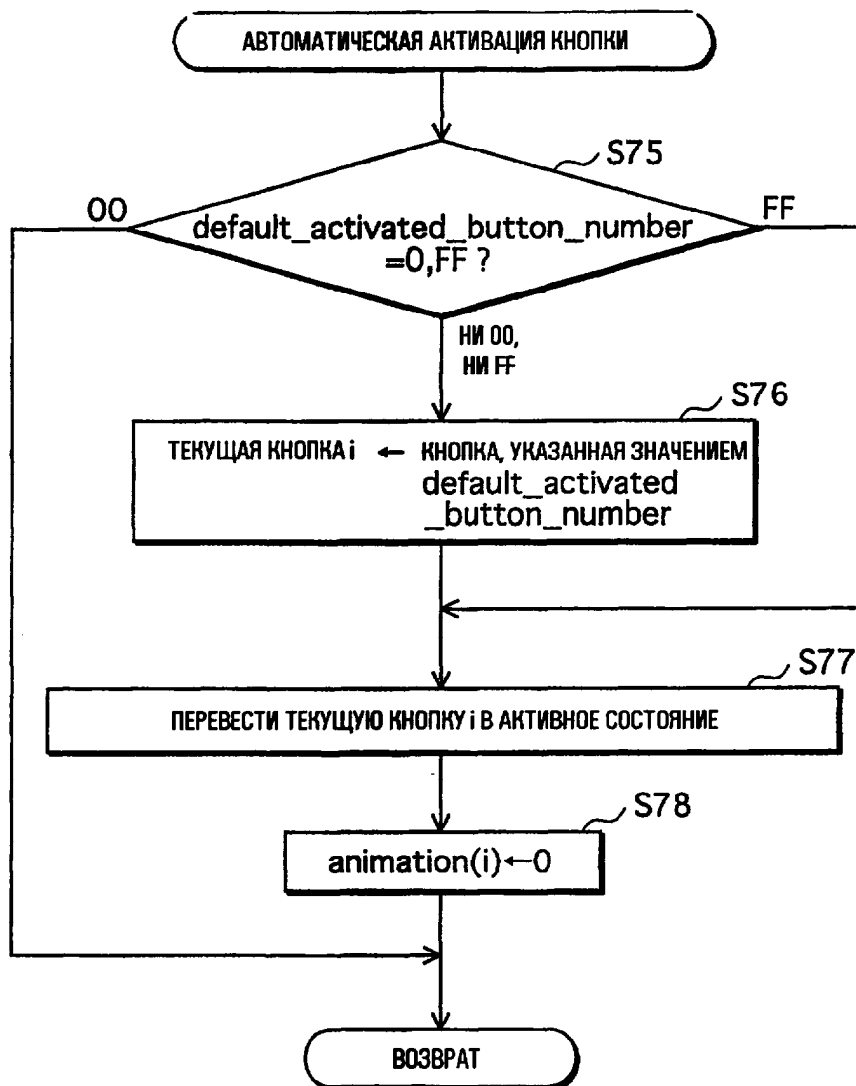
ФИГ.71



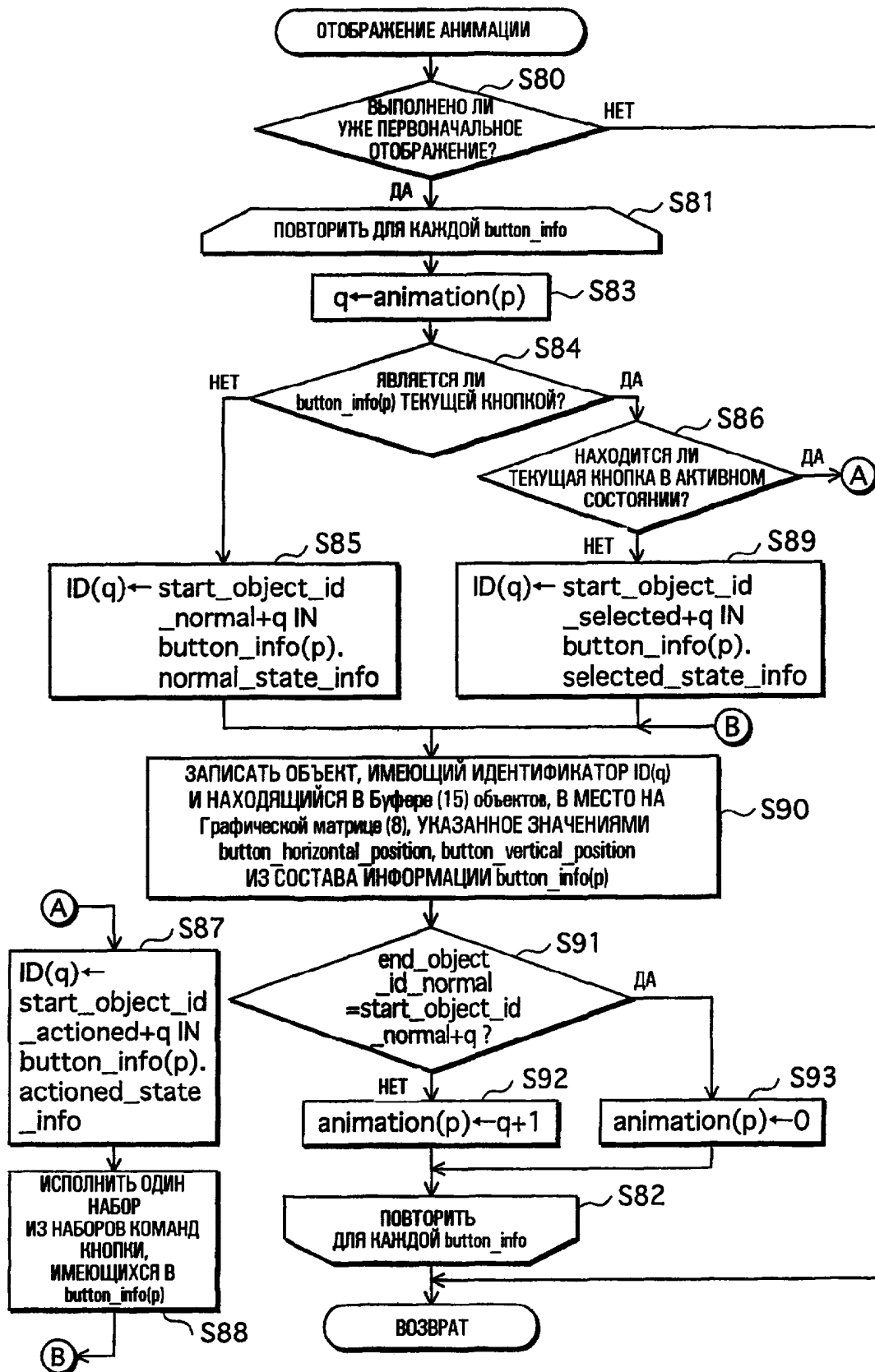
ФИГ.72



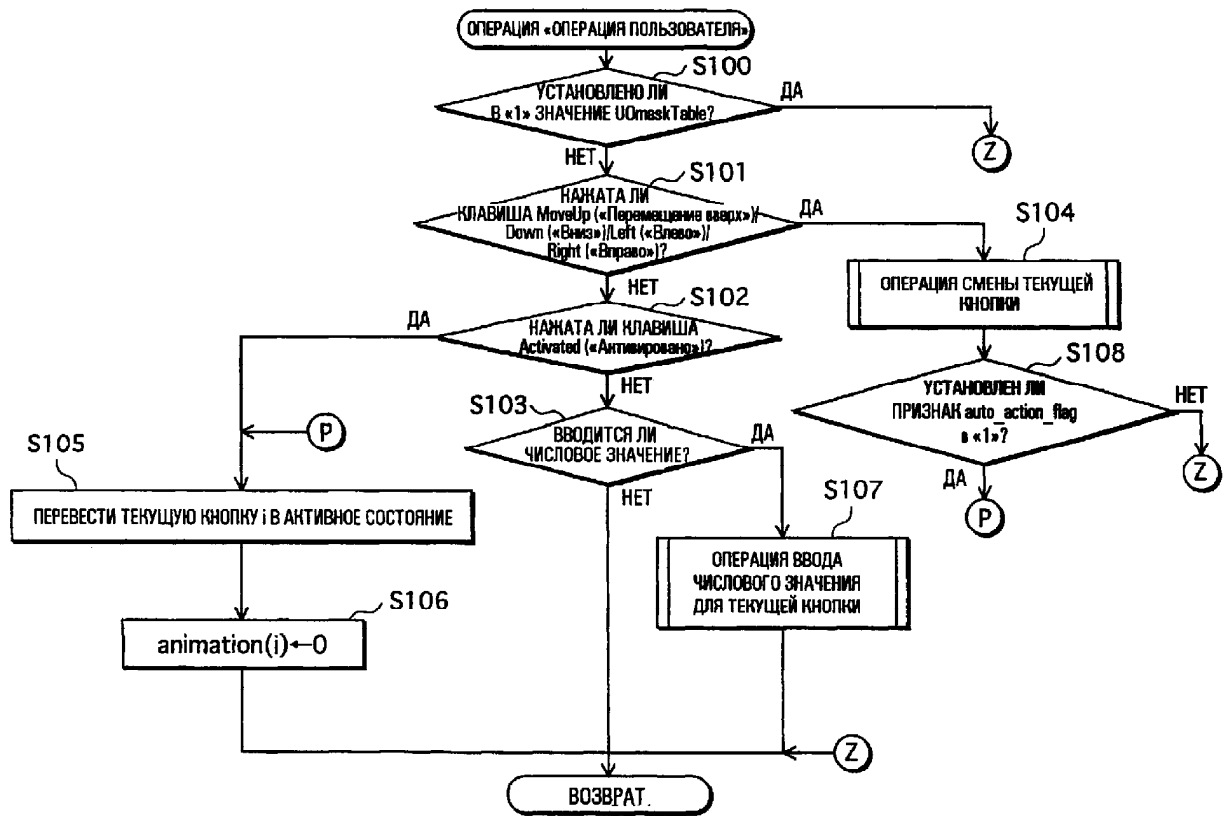
ФИГ.73



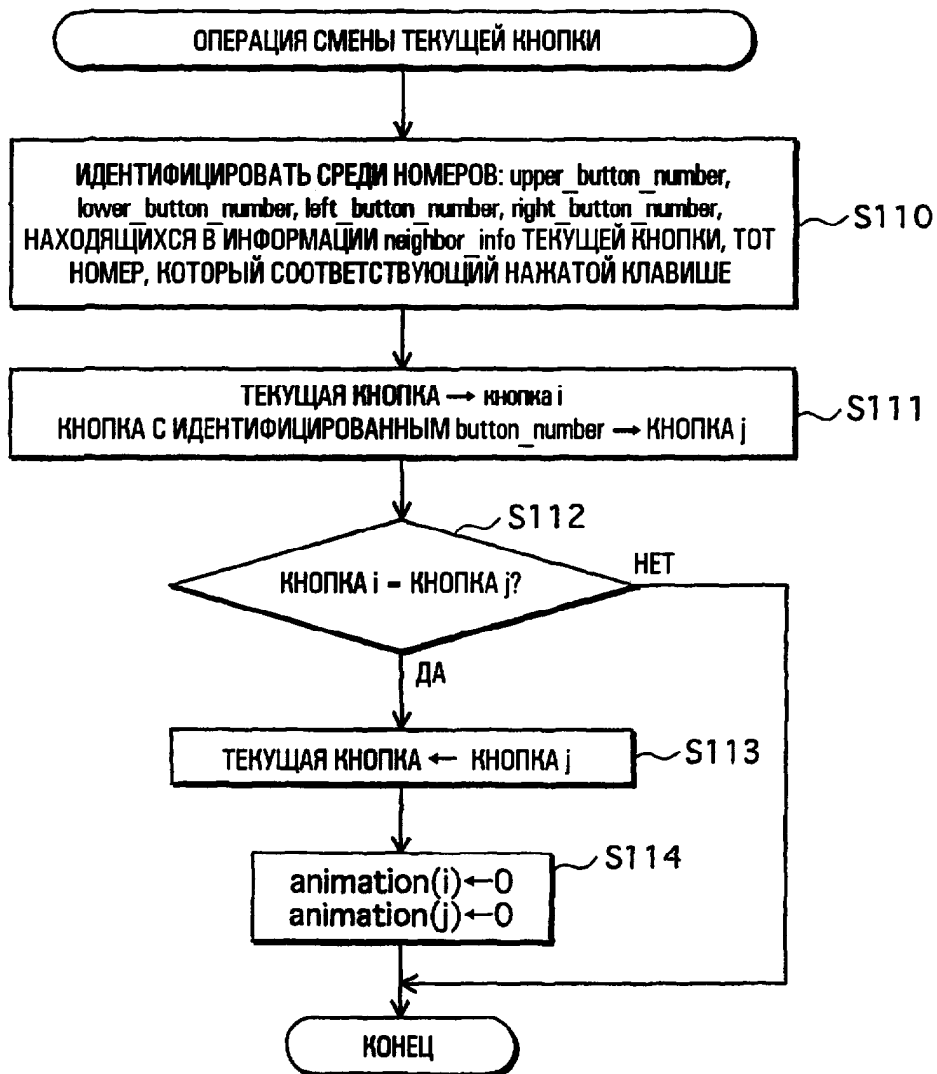
ФИГ.74



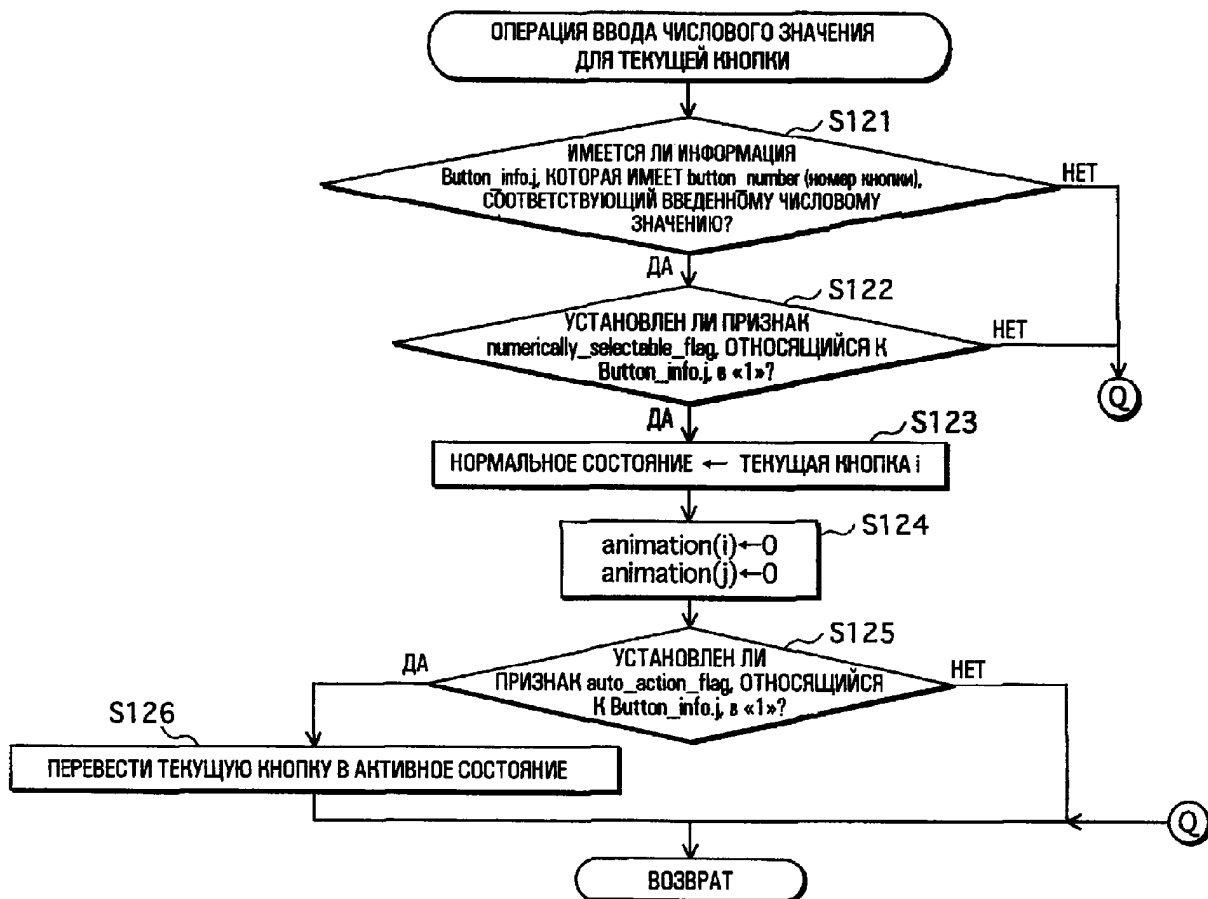
ФИГ.75



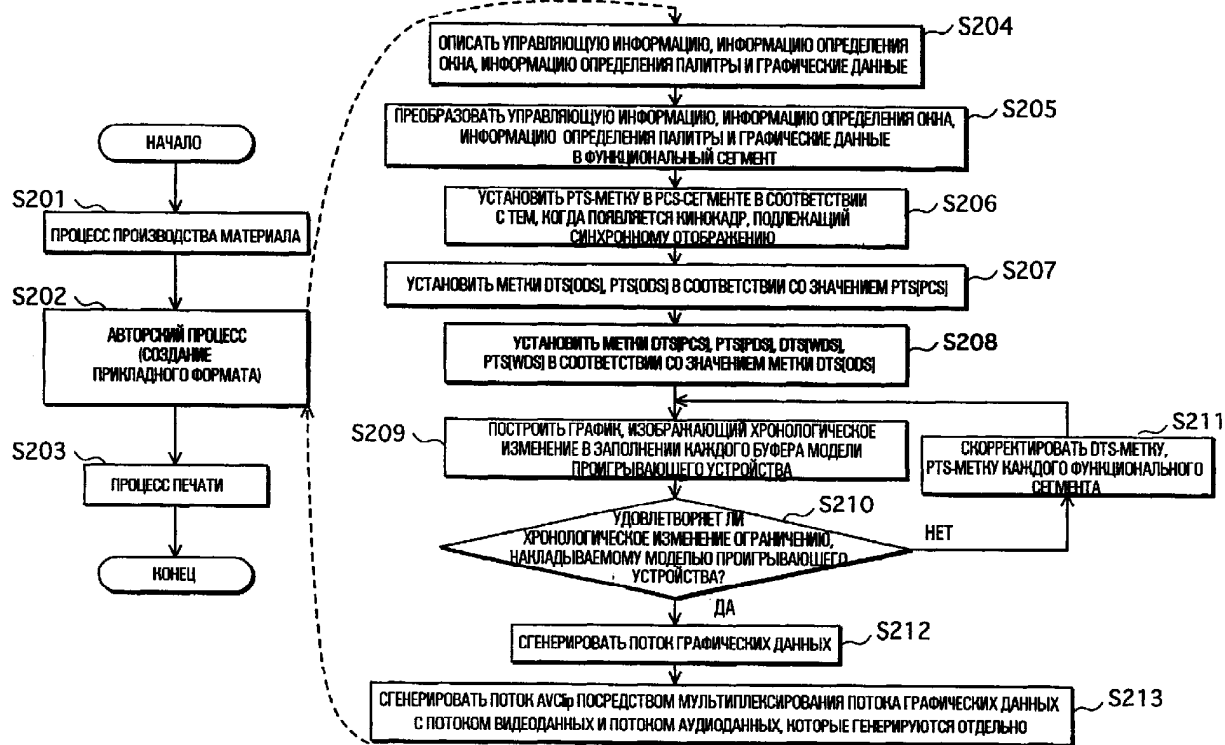
ФИГ.76



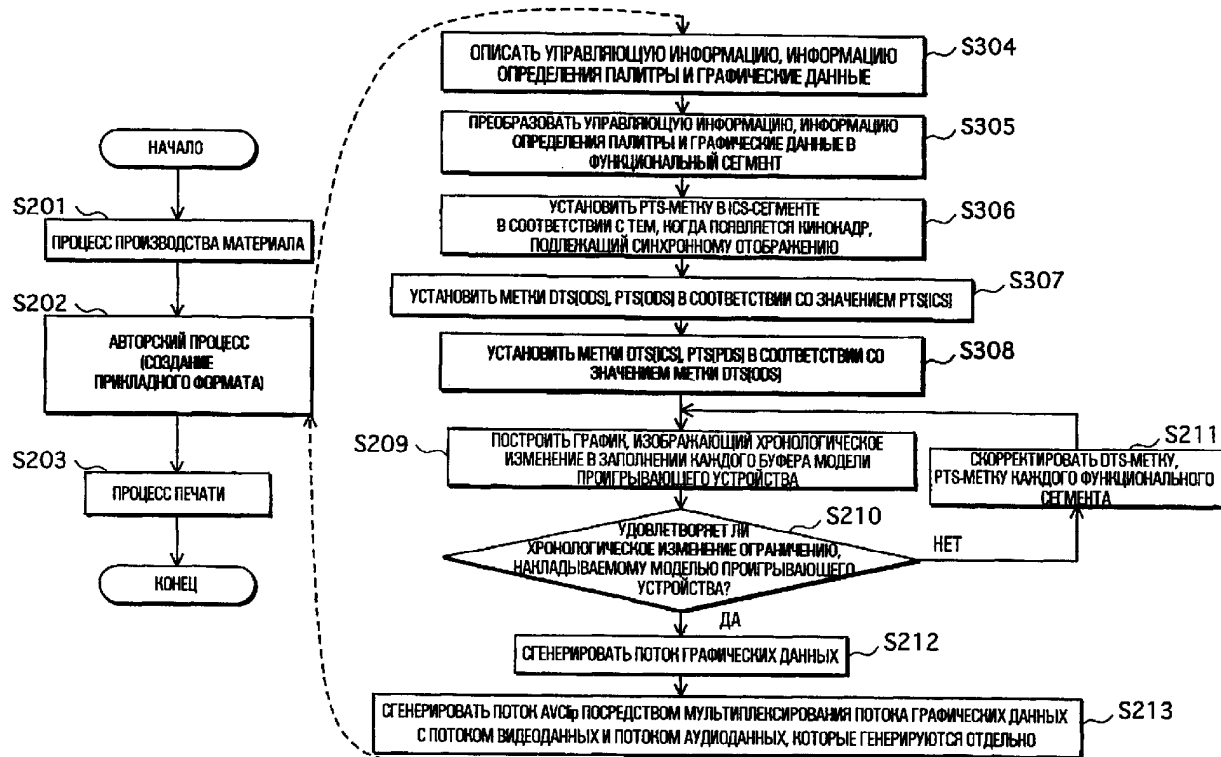
ФИГ.77



ФИГ.78



ФИГ.79



ФИГ.80