



(12) 发明专利

(10) 授权公告号 CN 101777372 B

(45) 授权公告日 2012. 09. 19

(21) 申请号 201010129948. 9

G11B 27/34 (2006. 01)

(22) 申请日 2005. 11. 30

(56) 对比文件

(30) 优先权数据

2004-349144 2004. 12. 01 JP

WO 2004074976 A2, 2004. 09. 02,

JP 2002369154 A, 2002. 12. 20,

CN 1463546 A, 2003. 12. 24,

(62) 分案原申请数据

200580015873. 9 2005. 11. 30

审查员 桂煦

(73) 专利权人 松下电器产业株式会社

地址 日本大阪府

(72) 发明人 桥本敏史 大芦雅弘 冈田智之

矢羽田洋

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 徐殿军

(51) Int. Cl.

G11B 27/32 (2006. 01)

G11B 19/02 (2006. 01)

G11B 27/034 (2006. 01)

G11B 27/10 (2006. 01)

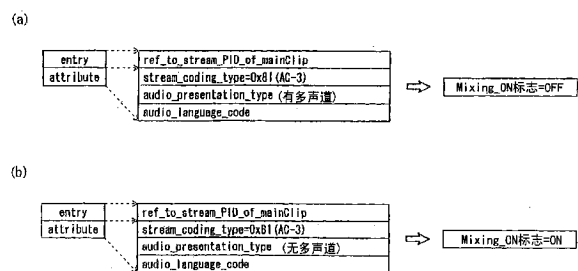
权利要求书 1 页 说明书 39 页 附图 44 页

(54) 发明名称

再现装置和再现方法

(57) 摘要

本发明提供一种再现装置和再现方法。在 BD-ROM 上, 记录着 AVClip 和 Java(注册商标) 应用。在该 AVClip 上, 复用着视频流及音频流。此外, 在 BD-ROM 上, 记录着管理信息和与其对应的 Mixing_On 标志, 上述管理信息是表示应与 Java(注册商标) 应用的执行同时开始的 AVClip 的再现控制的信息, Mixing_On 标志表示是否混合与用户对 Java(注册商标) 应用的操作相应的点击声、和上述再现控制时的 AVClip 的声音输出。



1. 一种再现装置,用于执行应用,并且再现数字流,其特征在于,

包括:

读出单元,从记录介质读出管理信息、以及与该管理信息对应的标志;

平台部,执行上述记录介质上记录的应用;

再现控制引擎部,在执行上述应用的同时执行由上述管理信息所表示的再现控制,从而再现数字流并进行影像输出及声音输出;以及

混合部,在与上述管理信息对应的标志是“开”的情况下,执行与用户对上述执行中的应用的操作相对应的点击声的输出、和声音输出之间的混合,而在与上述管理信息对应的标志是“关”的情况下,不执行该混合,

上述管理信息包含:

再现路径信息,通过表示视频流上的再现开始时刻及再现结束时刻的组,来规定再现路径;和

属性信息,表示上述再现路径的属性,

基于上述再现控制引擎部的再现控制,是根据附加了规定的再现属性的再现路径信息所示的再现路径来再现数字流,规定的再现属性,是表示自动开始用再现路径信息所示的再现路径进行的再现的属性,

上述再现控制引擎部进行的在执行上述应用的同时执行的由上述管理信息所表示的再现控制,是在再现路径信息上附加了表示自动再现的再现属性的情况下进行的。

2. 一种再现方法,用于执行应用,并且再现数字流,其特征在于,

该再现方法包括:

读出步骤,从记录介质读出管理信息、以及与上述管理信息对应的标志;

执行步骤,执行上述记录介质上记录的应用;

再现控制步骤,在执行上述应用的同时执行由上述管理信息所表示的再现控制,从而再现数字流并进行影像输出及声音输出;以及

混合步骤,在与上述管理信息对应的标志是“开”的情况下,执行与用户对上述执行中的应用的操作相对应的点击声的输出、和声音输出之间的混合,而在与上述管理信息对应的标志是“关”的情况下,不执行该混合,

上述管理信息包含:

再现路径信息,通过表示视频流上的再现开始时刻及再现结束时刻的组,来规定再现路径;和

属性信息,表示上述再现路径的属性,

基于上述再现控制步骤的再现控制,是根据附加了规定的再现属性的再现路径信息所示的再现路径来再现数字流,规定的再现属性,是表示自动开始用再现路径信息所示的再现路径进行的再现的属性,

上述再现控制步骤进行的与上述应用的执行同时执行的由上述管理信息所表示的再现控制,是在再现路径信息上附加了表示自动再现的再现属性的情况下进行的。

再现装置和再现方法

[0001] 本申请是申请日为 2005 年 11 月 30 日、申请号为 200580015873.9、名称为“记录媒体、再现装置、程序、再现方法、集成电路”的中国专利申请的分案申请。

技术领域

[0002] 本发明属于点击声（クリック音）再现的技术领域。

背景技术

[0003] 所谓点击声再现，是在用户对电影作品的再现中显示的 GUI 进行了操作的情况下，呼应该操作，在电影作品中的再现声音中混合点击声并输出的技术。

[0004] 例如考虑下述情况：电影作品的再现中显示的 GUI 为接受“开始游戏”这一功能的执行指示，按照该执行指示，将“现在开始游戏”这一语音宣告作为点击声，混合到电影正片的声音中并输出。如果该语音宣告是电影作品中登场的人物的声音，则儿童用户通过该语音宣告感受到电影作品中登场的人物正在对他说话的印象。通过以上的点击声再现的导入，能够将儿童可快乐地游戏的 GUI 导入到电影作品中。

[0005] 这里，在 1 个声音输出上混合别的的声音输出并输出的技术中，有以下的专利文献 1 中记载的技术。

[0006] 专利文献 1：(日本)特开平 7-320411 号公报

[0007] 然而，为了混合点击声，必须将数字化了的音频流暂时展开到 LPCM 状态。这里，在待混合的正片一侧的音频流具有例如 5.1 声道这一多声道属性的情况下，展开了的 LPCM 状态下的数据量很大。因此，在具有例如多声道属性的音频流上混合点击声并进行数字声音输出的情况下，由于频带上的限制，有些种类的数字接口不能进行数字声音输出。近年的电影大多重视音响方面，而用户也往往投入放大器、环绕扬声器等巨额的私人费用，构筑了私人的音响系统，所以如果想要混合点击声，却不能进行电影正片的数字声音输出，则对电影的负面印象不可估量。

[0008] 只要有因导入点击声而有例如“多声道的声音输出该不会中断吧”这一不安的束缚，就经常发生点击声这一功能本身不能用于电影制作中的情况。为了将电影制作工作室从这种束缚中解放出来，将在再现装置中进行重新编码的想法，在所有再现装置中执行即可。即，在再现装置内具备编码器，将具有 5.1 声道这一多声道属性的音频流展开到 LPCM 状态，并混合了点击声后，对其进行重新编码后，进行数字输出即可。具体地说，在经 S/PDIF(Sony/Philips Digital InterConnect Format(索尼/飞利浦数字互连格式)，IS060958-3 标准)这样的数字接口来执行混合后的声音的数字声音输出的情况下，需要进行上述重新编码，并变换为杜比数字或 DTS。但是这种重新编码必须具备编码器，所以无法被采用为 BD-ROM 再现装置的标准模型。

[0009] 就是说，现有技术中，存在着在要实现点击声的声音混合输出时，将对声音输出带来不良影响的技术问题。

发明内容

[0010] 本发明的目的是提供一种再现装置和再现方法,不影响声音输出,就能够实现点击声的混合输出。

[0011] 本发明的再现装置,用于执行应用,并且再现数字流,其特征在于,包括:读出单元,从记录介质上读出管理信息,该管理信息包含:表示上述应用能执行的区间的信息、表示在上述能执行的区间中执行应用的过程中应该开始的数字流的再现控制的信息、以及标志;平台部,在对该应用确定的能执行的区间中执行上述记录介质上记录的应用;再现控制引擎部,在上述能执行的区间中执行应用的过程中,通过执行由上述管理信息所表示的再现控制,同时地再现数字流并进行影像输出及声音输出;以及混合部,在上述管理信息所包含的标志是“开”的情况下,执行使用了与用户对上述执行中的应用的操作相对应的声音数据的点击声的输出、和上述数字流的声音输出之间的混合,而在上述管理信息所包含的标志是“关”的情况下,不执行该混合。

[0012] 本发明的再现方法,用于执行应用,并且再现数字流,对该应用确定有能执行的区间,其特征在于,包括:读出步骤,从记录介质上读出管理信息,该管理信息包含:表示上述应用能执行的区间的信息、表示在上述能执行的区间中执行应用的过程中应该开始的数字流的再现控制的信息、以及标志;执行步骤,在对该应用确定的能执行的区间中执行上述记录介质上记录的应用;再现控制步骤,在上述能执行的区间中执行应用的过程中,通过执行由上述管理信息所表示的再现控制,同时地再现数字流并进行影像输出及声音输出;以及混合步骤,在上述管理信息所包含的标志是“开”的情况下,执行使用了与用户对上述执行中的应用的操作相对应的声音数据的点击声的输出、和上述数字流的声音输出之间的混合,而在上述管理信息所包含的标志是“关”的情况下,不执行该混合。

[0013] 发明效果

[0014] 本发明具有上述结构,所以在不对声音输出带来不良影响的情况下,就能够实现点击声的声音混合输出。例如,能够在执行希望多声道的声音输出的再现控制时使混音无效,而在执行不希望多声道的声音输出的再现控制时使混音有效。

[0015] 由此,例如,电影制作工作室能够进行下述调整:在希望多声道的再现的情况下,禁止点击声;而在希望点击声的再现的情况下,禁止多声道的声音输出。

[0016] 在这种调整的基础上,若将点击声导入到电影制作中,则能够将制作工作室从“多声道的声音输出该不会中断吧”这一不安的束缚中解放出来。由此,能够鼓励在制作电影作品时导入点击声。

附图说明

[0017] 图1是本发明的记录介质的使用行为的图。

[0018] 图2是BD-ROM的内部结构的图。

[0019] 图3是被赋予了扩展名.m2ts的文件如何构成的示意图。

[0020] 图4是构成AVClip的TS包经过什么样的过程而被写入到BD-ROM上的图。

[0021] 图5是构成AVClip的TS包经过什么样的过程而被写入到BD-ROM上的图。

[0022] 图6是Clip信息的内部结构的图。

[0023] 图7是对于电影的视频流的EP_map设定的图。

- [0024] 图 8 是 PlayList 信息的数据结构的图。
- [0025] 图 9 是 AVClip 和 PlayList 之间的关系图。
- [0026] 图 10(a) 是 STN_table 的结构图。
- [0027] (b) ~ (e) 是 entry-attribute(项目属性)的详细情况的图。
- [0028] 图 11 是 PlayList 信息的 PlayListMark 信息的内部结构的图。
- [0029] 图 12 是用 PlayList 信息的 PlayListMark 信息来指定章节位置的图。
- [0030] 图 13 是 BD-J Object 的内部结构的图。
- [0031] 图 14 是存档文件容纳的程序、数据的图。
- [0032] 图 15 是文件 sound.bdmv 的结构图。
- [0033] 图 16(a) 是应用管理表的内部结构的图。
- [0034] (b) 是构成应用管理表的信息元素的意义内容的图。
- [0035] 图 17 是盘内容中的状态转移的图。
- [0036] 图 18(a) 是整个 BD-ROM 的时间轴的图。
- [0037] (b) 是整个 BD-ROM 的时间轴的结构图。
- [0038] 图 19(a) (b) 是在整个 BD-ROM 的时间轴上、由 BD-J Object 确定的标题再现区间的图。
- [0039] 图 20 是图 19(b) 的时间轴上规定的生命周期的典型的图。
- [0040] 图 21 是包含正片标题、在线购物标题、游戏标题这 3 个标题的盘内容的图。
- [0041] 图 22(a) (b) 是应用管理表、生命周期的一例的图。
- [0042] 图 23 是运行属性可取的三态 (Present、AutoRun、Suspend)、紧挨着的前面的标题中的应用状态的三态 (未运行、运行中、Suspend) 可取的组合的图。
- [0043] 图 24(a) 是播放列表管理表的内部结构的图。
- [0044] (b) 是构成播放列表管理表的信息元素的意义内容的图。
- [0045] 图 25 是用播放列表管理表、应用管理表规定的标题的具体例的图。
- [0046] 图 26 是当前标题可取的三态 (无播放列表管理表、有播放列表管理表而且未指定、有播放列表管理表而且 AutoPlay)、和紧挨着的前面的标题中的 PL 的状态 (未再现状态、再现中状态) 可取的 6 种组合的图。
- [0047] 图 27(a) 是播放列表管理表及应用管理表的描述例的图。
- [0048] (b) 是如何根据如图 27(a) 所示描述的应用管理表、播放列表管理表来进行播放列表再现、应用执行的图。
- [0049] 图 28(a) 是播放列表管理表的另一描述例的图。
- [0050] (b) 是基于图 28(a) 的情况的应用执行及进行播放列表再现的图。
- [0051] 图 29 是 SMT 的内部结构的图。
- [0052] 图 30(a) (b) 是 STN_Table 中的音频流的 entry-attribute、和 Mixing_On 标志的设定之间的因果关系的图。
- [0053] 图 31 是在 Mixing_On 标志被设定为 ON 的标题中操作了 GUI 框架上的按钮的情况下再现装置的声音再现的图。
- [0054] 图 32 是在 Mixing_On 标志被设定为 OFF 的标题中操作了 GUI 框架上的按钮的情况下再现装置的声音再现的图。

- [0055] 图 33 是本发明的再现装置的内部结构的图。
- [0056] 图 34 是将由 ROM 24 中保存的软件和硬件组成的部分替换为层结构而绘制的图。
- [0057] 图 35 是 Java(注册商标)虚拟机 36 的内部结构的图。
- [0058] 图 36 是 Presentation Engine 31 ~ 模块管理器 34 进行的处理的示意图。
- [0059] 图 37 是应用管理器 37 根据 BD-J Object 中的 PLMT 进行的处理的图。
- [0060] 图 38 是应用管理器 37 根据 BD-J Object 中的 SMT 进行的处理的图。
- [0061] 图 39 是应用管理器 37 进行的处理过程的流程图。
- [0062] 图 40(a) 是第 2 实施方式的播放列表信息的内部结构的图。
- [0063] (b) 是 PlayItem 信息内所设的、Mixing_On 标志的内容的图。
- [0064] 图 41 是 Playback Control Engine 32 进行的播放列表再现过程的流程图。
- [0065] 图 42 是应用管理器 37 进行的通知处理的处理过程的流程图。
- [0066] 图 43 是第 3 实施方式的应用管理器 37 的处理过程的图。
- [0067] 图 44(a) 示出基于父母锁级别的选择算法的内容。
- [0068] (b) 示出基于 Language for Audio 的选择算法的内容。
- [0069] (c) 示出基于 Player Configuration for Video 的选择算法的内容。
- [0070] 图 45 是使标题无边界应用(title unboundary application)选择播放列表的过程的示意图。

[0071] 标号说明

- [0072] 1 BD-ROM 驱动器
- [0073] 2 读缓冲器
- [0074] 3 解复用器
- [0075] 4 视频解码器
- [0076] 5 视频平面
- [0077] 6 声音处理器
- [0078] 7 声音处理器
- [0079] 8 混音器
- [0080] 9 声音控制器
- [0081] 10 D/A 转换器
- [0082] 11 Interactive Graphics(交互图形)解码器
- [0083] 12 Interactive Graphics 平面
- [0084] 13 Presentation Graphics(演示图形)解码器
- [0085] 14 Presentation Graphics 平面
- [0086] 15 JPEG 解码器
- [0087] 16 Still(静止)平面
- [0088] 17 合成部
- [0089] 18 STC-delta 附加部
- [0090] 19 ATC-delta 附加部
- [0091] 20 局部储存器
- [0092] 21 指令 ROM

- [0093] 22 用户事件处理部
- [0094] 23 PSR 组
- [0095] 24 CPU
- [0096] 25 脚本存储器
- [0097] 26 局部存储器

具体实施方式

[0098] (第1实施方式)

[0099] 以下,说明本发明的记录介质的实施方式。首先,说明本发明的记录介质的实施行为中的使用作为的方式。图1是本发明的记录介质的使用行为的方式的图。在图1中,本发明的记录介质是BD-ROM 100, BD-ROM 100用于向由再现装置200、遥控器300、电视机400、环绕扬声器上连接的放大器500形成的家庭影院系统提供作品。

[0100] 在本系统中,再现装置200再现BD-ROM 100上记录着的各种数据,来进行影像输出及声音输出。该再现装置200的影像输出经HDMI(HighDefinition Multimedia Interface:高清晰度多媒体接口)等模拟接口输出到电视机400。此外,再现装置200的声音输出经S/PDIF或HDMI等模拟接口输出到电视机400或放大器500。

[0101] 以上是本发明的记录介质的使用方式的说明。

[0102] 接着说明本发明的记录介质的生产行为。本发明的记录介质能够用BD-ROM的文件系统上的改进来实现。图2是表示BD-ROM的文件·目录结构的图。在本图中,在BD-ROM上,在Root(根)目录下,有BDMV目录。

[0103] 在BDMV目录中,有被赋予了扩展名bdmv的文件(index.bdmv, MovieObject.bdmv)。而在该BDMV目录的属下,还存在称为PLAYLIST目录、CLIPINF目录、STREAM目录、BDBJ目录、BDJA目录、AUXDATA目录的6个子目录。

[0104] 在PLAYLIST目录中,有被赋予了扩展名mpls的文件(00001.mpls, 00002.mpls, 00003.mpls)。

[0105] 在CLIPINF目录中,有被赋予了扩展名clpi的文件(00001.clpi, 00002.clpi, 00003.clpi)。

[0106] 在STREAM目录中,有被赋予了扩展名m2ts的文件(00001.m2ts, 00002.m2ts, 00003.m2ts)。

[0107] 在BDBJ目录中,存在被赋予了扩展名BOBJ的文件(00001.bobj, 00002.bobj, 00003.bobj)。

[0108] 在BDJA目录中,有被赋予了扩展名jar的文件(00001.jar, 00002.jar, 00003.jar)。

[0109] 在AUXDATA目录中,保存着文件sound.bdmv。

[0110] 通过以上的目录结构可知,种类互不相同的多个文件被配置在BD-ROM上。

[0111] <BD-ROM的结构之1.AVClip>

[0112] 首先,说明被赋予了扩展名.m2ts的文件。图3是被赋予了扩展名.m2ts的文件如何构成的示意图。被赋予了扩展名.m2ts的文件(00001.m2ts, 00002.m2ts, 00003.m2ts……)保存着AVClip。AVClip是MPEG2-TransportStream(MPEG2传输流)形式的数

字流。该数字流是通过将把电影影像、NTSC 影像、PAL 影像数字化而得到的数字视频、把 LPCM 音源、AC-3 音源、DTS 音源数字化而得到的数字音频（上第 1 层）变换为由 PES 包组成的基本流（上第 2 层），进而变换为 TS 包（上第 3 层），同样将字幕类的演示图形流（Presentation Graphics (PG) 流）及对话类的交互图形流（InteractiveGraphics (IG) 流）（下第 1 层）变换为 PES 包串（下第 2 层），进而变换为 TS 包（下第 3 层），并复用它们而构成的。

[0113] 所谓 PG 流，是实现伴随动态图像的再现进行的字幕显示的基本流，而 IG 流则是实现伴随动态图像的再现进行的 GUI 的基本流。

[0114] PES 打包了的数字影像被称为视频流，而 PES 打包了的声音则被称为音频流。这些视频流、音频流是基本流，作为可独立解码的单位，存在“Access Unit（存取单位）”。在视频流中，所谓的 GOP (Group Of Picture : 图片组) 就相当于该“Access Unit”。

[0115] 将视频流中的、用 1 个 PTS 来再现的再现单位（图片等）称为“VideoPresentation Unit（视频演示单位）”。将音频流中的、用 1 个 PTS 来再现的再现单位称为“Audio Presentation Unit（音频演示单位）”。

[0116] 这里，构成 AVClip 的 PES 包构成 1 个以上的“STC_Sequence。所谓“STC_Sequence”，是 PES 包的排列，是指在其 PTS、DTS 参考的 SystemTime Clock（系统时钟）（STC）的值上，不存在 STC 不连续点（system time-basediscontinuity）。没有 STC 不连续点是 STC_Sequence 的必要条件，所以从构成 1 个 STC_Sequence 的 PES 包串中的、位于紧挨着 STC 不连续点后方并包含 PCR (Program Clock Reference : 节目时钟参考) 的 PES 包到紧挨着下一 STC 不连续点之前为 1 个 STC_Sequence。

[0117] 接着，说明如上构成的 AVClip 如何被写入到 BD-ROM 上。图 4 示出构成 AVClip 的 TS 包经过什么样的过程而被写入到 BD-ROM 上。在本图的第 1 层中示出构成 AVClip 的 TS 包。

[0118] 构成 AVClip 的 188 个字节的 TS 包如第 2 层所示被附加了 4 个字节的 TS_extra_header（图中的阴影部分），而成为 192 个字节长的 Source（源）包。该 TS_extra_header 包含 Arrival_Time_Stamp。

[0119] 构成 AVClip 的 Source 包在第 3 层的 AVClip 中构成 1 个以上的“ATC_Sequence”。所谓“ATC_Sequence”，是 Source 包的排列，是指在该 Arrival_Time_Stamp 参考的 Arrival_Time_Clock 上，不存在不连续点（noarrival time-base discontinuity）。换言之，将该 Arrival_Time_Stamp 参考的 Arrival_Time_Clock 上存在连续性的 Source 包串称为“ATC_Sequence”。

[0120] 这种 ATC_Sequence 成为 AVClip，以 xxxxx.m2ts 这一文件名记录到 BD-ROM 上。

[0121] 这种 AVClip 与通常的计算机文件同样，被分割为多个文件盘区（fileextent），记录到 BD-ROM 上的区域上。第 4 层示意性地示出 AVClip 如何被记录到 BD-ROM 上。在该第 4 层中，构成文件的各文件盘区具有预定的 Sextent 以上的数据长度。

[0122] 下面讨论将 AVClip 分割为多个盘区来记录的情况下的、平均每一个盘区的最小数据长度 Sextent。

[0123] 这里，在 BD-ROM 上，光拾取器的跳跃所需的时间由

[0124] $T_{\text{jump}} = T_{\text{access}} + T_{\text{overhead}}$

- [0125] 给出。
- [0126] Taccess 是按照跳跃距离给出的时间 (m 秒)，
- [0127] 如果跳跃距离 (逻辑块数) 是 0 ~ 5000 则为 179m 秒，
- [0128] 如果跳跃距离 (逻辑块数) 是 5001 ~ 10,000 则为 210m 秒，
- [0129] 如果跳跃距离 (逻辑块数) 是 10,001 ~ 20,000 则为 270m 秒，
- [0130] 如果跳跃距离是半程 (half stroke) 则为 990m 秒，
- [0131] 如果跳跃距离是全程 (full stroke) 则为 1220m 秒。
- [0132] 从 BD-ROM 上读出的 TS 包被保存到称为读缓冲器的缓冲器中后，被输出到解码器；而向读缓冲器的输入以 Rud 这一比特率来进行，在设 ECC 块中的扇区数为 Secc 的情况下，
- [0133] Toverhead 通过
- [0134] $Toverhead \leq (2 \times Secc \times 8) / Rud = 20m \text{ 秒}$
- [0135] 这一计算来给出。
- [0136] 从 BD-ROM 上读出的 TS 包以 Source 包的状态被保存到读缓冲器中后，以 TS_Recording_rate 这一传送速率被提供给解码器。
- [0137] 为了使以 TS_Recording_rate 这一传送速率向解码器提供 TS 包不中断，需要在 Tjump 期间持续从读缓冲器向解码器输出 TS 包。这里，从读缓冲器的输出不是以 TS 包、而是以 Source 包的状态来进行的，所以在设 TS 包的与 Source 包的长度比为 192/188 的情况下，需要在 Tjump 期间用 $(192/188 \times TS_Recording_rate)$ 这一传送速率来持续从读缓冲器输出 Source 包。
- [0138] 因此，用于使读缓冲器不下溢的缓冲器存储量为：
- [0139] $Boccupied \geq (Tjump/1000 \times 8) \times ((192/188) \times TS_Recording_rate)$ 。
- [0140] 向读缓冲器的输入速率是 Rud，从读缓冲器的输出速率是 $TS_Recording_rate \times (192/188)$ ，所以向读缓冲器的存储速率通过计算输入速率 - 输出速率来给出，为 $(Rud - TS_Recording_rate \times (192/188))$ 。
- [0141] 将该 Boccupied 存储到读缓冲器中所需的时间 Tx 为
- [0142] $Tx = Boccupied / (Rud - TS_Recording_rate \times (192/188))$ 。
- [0143] 为了从 BD-ROM 上读出，需要在该时间 Tx 内持续以 Rud 来输入 TS 包，所以将 AVClip 分割为多个盘区来记录的情况下的、平均每一个盘区的最小数据长度 Sextent 为
- [0144] $Sextent = Rud \times Tx$
- [0145] $= Rud \times Boccupied / (Rud - TS_Recording_rate \times (192/188))$
- [0146] $\geq Rud \times (Tjump/1000 \times 8) \times ((192/188) \times TS_Recording_rate)$
- [0147] $/(Rud - TS_Recording_rate \times (192/188))$
- [0148] $\geq (Rud \times Tjump/1000 \times 8) \times$
- [0149] $TS_Recording_rate \times 192 / (Rud \times 188 - TS_Recording_rate \times 192)$ 。
- [0150] 因此，
- [0151] $Sextent \geq$
- [0152] $(Tjump \times Rud/1000 \times 8) \times$
- [0153] $TS_Recording_rate \times 192 / (Rud \times 188 - TS_Recording_rate \times 192)$ 。
- [0154] 构成 AVClip 的各文件盘区通过具有这样算出的 Sextent 以上的数据长度，即使构

成 AVClip 的各文件盘区离散地位于 BD-ROM 上,在再现时向解码器提供 TS 包也不会中断,而能连续地读出。

[0155] 图 5 是 BD-ROM 的物理单位和构成 1 个文件盘区的 Source 包之间的对应关系的图。如第 2 层所示,在 BD-ROM 上形成了多个扇区。构成文件盘区的 Source 包如第 1 层所示,每 32 个被分为一组,写入到连续的 3 个扇区中。由 32 个 Source 包组成的组是 6144 个字节 ($= 32 \times 192$),这与 3 个扇区的长度 6144 个字节 ($= 2048 \times 3$) 一致。将 3 个扇区中容纳的 32 个 Source 包称为“Aligned Unit(对准单位)”,在写入到 BD-ROM 中时,以 Aligned Unit 为单位来进行加密。

[0156] 在第 3 层中,扇区以 32 个为单位被附加纠错码,构成 ECC 块。再现装置只要以 Aligned Unit 为单位来访问 BD-ROM,就能够得到 32 个完整的 Source 包。以上是向 BD-ROM 上写入 AVClip 的过程。

[0157] <BD-ROM 的结构之 2. Clip 信息 >

[0158] 接着说明被赋予了扩展名 .clpi 的文件。被赋予了扩展名 .clpi 的文件 (00001.clpi, 00002.clpi, 00003.clpi……) 保存着 Clip 信息。Clip 信息是各个 AVClip 的管理信息。图 6 是 Clip 信息的内部结构的图。如本图的左侧所示,Clip 信息由

[0159] i) 保存着关于 AVClip 的信息的“ClipInfo()”、

[0160] ii) 保存着与 ATC Sequence(ATC 序列)、STC Sequence(STC 序列) 有关的信息的“Sequence Info()”、

[0161] iii) 保存着与 Program Sequence(节目序列) 有关的信息的“ProgramInfo()”、

[0162] iv) “Characteristic Point Info(CPI())”组成。

[0163] Sequence Info 是关于被包含在 AVClip 中的、1 个以上的 STC-Sequence、ATC-Sequence 的信息。设这些信息的意义是为了将 STC、ATC 的不连续点预先通知给再现装置。即,如果存在这种不连续点,则在 AVClip 内有可能出现相同值的 PTS、ATS,在再现时发生问题。为了表示 STC、ATC 连续是从传输流的哪里到哪里,所以设了 Sequence Info。

[0164] 所谓 Program Info,是表示 Program 内容一定的区间(Program Sequence) 的信息。所谓 Program,是共享用于同步再现的时间轴的基本流的集合。设 Program Sequence 信息的意义是为了将 Program 内容的变化点预先通知给再现装置。这里的所谓 Program 内容的变化点,是指视频流的 PID 变化、或者视频流的种类从 SDTV 变换为 HDTV 的点等。

[0165] 接着说明 Characteristic Point Info。图中的引出线 cu2 详细示出了 CPI 的结构。如引出线 cu2 所示,CPI 由 N_c 个 EP_map_for_one_stream_PID(EP_map_for_one_stream_PID(0) ~ EP_map_for_one_stream_PID(-1)) 组成。这些 EP_map_for_one_stream_PID 是属于 AVClip 的各个基本流的 EP_map。EP_map 是与项目时刻 (PTS_EP_start) 相对应地来表示在一个基本流中作为 Access Unit 的起始部分的、Access Unit Delimiter(存取单位定界符) 所存在的项目位置的包号 (SPN_EP_start) 的信息。图中的引出线 cu3 详细示出了 EP_map_for_one_stream_PID 的内部结构。

[0166] 由此可知,EP_map_for_one_stream_PID 由 N_c 个 EP_High(EP_High(0) ~ EP_High(N_c-1))、和 N_f 个 EP_Low(EP_Low(0) ~ EP_Low(N_f-1)) 组成。这里,EP_High 具有表示 Access Unit 的 SPN_EP_start 及 PTS_EP_start 的高位比特的作用,而 EP_Low 具有表示 Access Unit 的 SPN_EP_start 及 PTS_EP_start 的低位比特的作用。

[0167] 图中的引出线 cu4 详细示出了 EP_High 的内部结构。如该引出线所示, EP_High(i) 由作为对 EP_Low 的参照值的“ref_to_EP_Low_id[i]”、表示位于 Access Unit 的起始处的 Non-IDR I 图片、IDR 图片的 PTS 的高位比特的“PTS_EP_High[i]”、以及表示位于 Access Unit 的起始处的 Non-IDR I 图片、IDR 图片的 SPN 的高位的“SPN_EP_High[i]”组成。这里, 所谓 i, 是用于识别任意的 EP_High 的标识符。

[0168] 图中的引出线 cu5 详细示出了 EP_Low 的结构。如引出线 cu5 所示, EP_Low 由表示对应的 Access Unit 是否是 IDR 图片的“is_angle_change_point(EP_Low_id)”、表示对应的 Access Unit 的长度的“I_end_position_offset(EP_Low_id)”、表示对应的 Access Unit (Non-IDR I 图片、IDR 图片) 的 PTS 的低位比特的“PTS_EP_Low(EP_Low_id)”、以及表示对应的 Access Unit (Non-IDR I 图片、IDR 图片) 的 SPN 的低位比特的“SPN_EP_Low(EP_Low_id)”组成。这里, 所谓 EP_Low_id, 是用于识别任意的 EP_Low 的标识符。

[0169] <Clip 信息的说明之 2. EP_map>

[0170] 以下, 通过具体例来说明 EP_map。图 7 是对电影的视频流的 EP_map 设定的图。第 1 层表示按显示顺序配置的多个图片 (MPEG4-AVC 中规定的 IDR 图片、I 图片、B 图片、P 图片), 第 2 层表示该图片的时间轴。第 4 层表示 BD-ROM 上的 TS 包串, 第 3 层表示 EP_map 的设定。

[0171] 假设在第 2 层的时间轴上, 在时刻 t1 ~ t7 上, 存在作为 Access Unit 的 IDR 图片及 I 图片。假设这些 t1 ~ t7 的时间间隔是 1 秒左右, 则电影所用的视频流中的 EP_map 被设定为, 示出 t1 ~ t7 作为项目时刻 (PTS_EP_start), 与其相对应来示出项目位置 (SPN_EP_start)。

[0172] <PlayList 信息>

[0173] 接着, 说明 PlayList 信息。被赋予了扩展名“mpls”的文件 (00001.mpls) 是保存着 PlayList (PL) 信息的文件。

[0174] 图 8 是 PlayList 信息的数据结构的图。在本图中, 如引出线 mp1 所示, PlayList 信息由定义 MainPath 的 MainPath 信息 (MainPath())、和定义章节的 PlayListMark 信息 (PlayListMark()) 组成。

[0175] <PlayList 信息的说明之 1. MainPath 信息>

[0176] 首先说明 MainPath。MainPath 是对作为主影像的视频流或音频流定义的再现路径。

[0177] MainPath 如箭头 mp1 所示, 由多个 PlayItem 信息 #1……#m 来定义。PlayItem 信息定义构成 MainPath 的 1 个以上的逻辑再现区间。PlayItem 信息的结构由引出线 hs1 来详细示出。如该引出线所示, PlayItem 信息由表示再现区间的 IN(入) 点及 Out(出) 点所属的 AVClip 的再现区间信息的文件名的“Clip_Information_file_name”、表示 AVClip 的编码方式的“Clip_codec_identifier”、表示 PlayItem 是否构成多角度的“is_multi_angle”、唯一地表示由该 PlayItem 作为对象的 STC_Sequence 的“ref_to_STC_id[0]”、表示是否无缝地进行该 PlayItem 和其前一个 PlayItem 之间的连接的“connection_condition”、表示再现区间的始点的时间信息“In_time”、表示再现区间的终点的时间信息“Out_time”、表示在该 PlayItem 中应屏蔽的用户操作是哪个的“UO_mask_table”、表示是否允许随机访问该 PlayItem 的中途的“PlayItem_random_access_flag”、表示在该 PlayItem 的再现结束后

是否继续静止显示最后一个图片的“Still_mode”、以及“STN_table”构成。其中,构成再现路径的是表示再现区间的始点的时间信息“In_time”、表示再现区间的终点的时间信息“Out_time”的组,所谓再现路径信息,由该“In_time”及“Out_time”的组构成。

[0178] 图 9 是 AVClip 和 PlayList 之间的关系图。第 1 层表示 PlayList 信息具有的时间轴。第 2 层至第 5 层表示由 EP_map 参照的视频流(与图 7 所示的相同)。

[0179] PlayList 信息包含 PlayItem 信息 #1、#2 这 2 个 PlayItem 信息,用这些 PlayItem 信息 #1、#2 的 In_time、Out_time 定义了 2 个再现区间。如果排列这些再现区间,则定义了与 AVClip 时间轴不同的时间轴。这就是第 1 层所示的 PlayItem 时间轴。这样,通过定义 PlayItem,能够定义与 AVClip 不同的时间轴。

[0180] <STN_table>

[0181] STN_table 是表示用 Play Item(播放项目)的 Clip_Information_file_name 指定的 AVClip 上复用的多个基本流中的可再现者的表。具体地说,STN_table 通过将关于多个基本流的每个的 entry(项目)与 attribute(属性)相对应来构成。

[0182] 图 10(a) 是 STN_table 的内部结构的图。如本图所示,STN_table 包含 STN_table 中的 entry 和 attribute 的多个组(entry-attribute),是表示这些 entry-attribute 的组的个数(number_of_video_stream_entries, number_of_audio_stream_entries, number_of_PG_stream_entries, number_of_IG_stream_entries)的数据结构。

[0183] entry-attribute 的组如图中的括号记号“{}”所示,分别对应于 Play Item 中可再现的视频流、音频流、PG_stream、IG 流。

[0184] 下面说明 entry-attribute 的详细情况。图 10(b) ~ (e) 是 entry-attribute 的详细情况的图。

[0185] 图 10(b) 是与视频流对应的 entry-attribute 的组的图。

[0186] 视频流中的 entry 包含表示在解复用 AVClip 时提取该视频流时所用的 PID 的“ref_to_stream_PID_of_mainClip”。

[0187] 视频流中的 attribute 包含被设定为 0x02 的“stream_coding_type”、和表示视频流的显示速率的“frame_rate”等。

[0188] 图 10(c) 是与音频流对应的 entry-attribute 的组的图。

[0189] 音频流中的 entry 包含表示在解复用 AVClip 时提取该音频流时所用的 PID 的“ref_to_stream_PID_of_mainClip”。

[0190] 音频流中的 attribute 由通过被设定为 0x80(LinearPCM(线性 PCM))、0x81(AC-3)、0x82(DTS) 中的某一个来表示音频流的编码类型的“stream_coding_type”、和表示对应的音频流的声道结构、表示可否进行多声道输出的“audio_presentation_type”、以及表示对应的音频流的语言属性的“audio_language_code”等组成。

[0191] 在多声道中,除了 5.1CH 的环绕声之外,也包含立体声,但是在以下的说明中,假设多声道只意味着 5.1CH 的环绕声来进行说明。

[0192] 图 10(d) 是与 PG 流对应的 entry-attribute 的组的图。

[0193] PG 流中的 entry 包含表示在解复用 AVClip 时提取该 PG 流时所用的 PID 的“ref_to_stream_PID_of_mainClip”。

[0194] PG 流中的 attribute 由通过被设定为 0x90 来表示 PG 流的编解码器(codec)的

“stream_coding_type”、和表示对应的 PG 流的语言属性的“PG_language_code”组成。

[0195] 图 10(e) 是与 IG 流对应的 entry-attribute 的组的图。

[0196] IG 流中的 entry 包含表示在解复用 AVClip 时提取该 IG 流时所用的 PID 的“ref_to_stream_PID_of_mainClip”。

[0197] IG 流中的 attribute 由通过被设定为 0x91 来表示 IG 流的编解码器的“stream_coding_type”、和表示对应的 IG 流的语言属性的“IG_language_code”组成。

[0198] <PlayList 信息的说明之 2.PlayListMark>

[0199] 以上是本实施方式的 PlayItem 信息的说明。接着说明 PlayListMark 信息。

[0200] 图 11 是 PlayList 信息的 PlayListMark 信息的内部结构的图。如本图的图中的引出线 pm0 所示, PlayListMark 信息由多个 PLMark 信息(#1 ~ #n) 组成。PLmark 信息(PLmark()) 是将 PL 时间轴中的任意的区间指定为章节节点的信息。如引出线 pm1 所示, PLmark 信息包含表示作为章节指定的对象的 PlayItem 的“ref_to_PlayItem_Id”、和用时间符号来表示该 PlayItem 中的章节位置的“mark_time_stamp”。

[0201] 图 12 是用 PlayList 信息的 PLMark 信息来指定章节位置的图。本图的第 2 层至第 5 层表示图 10 所示的 EP_map 和 AVClip。

[0202] 本图的第 1 层表示 PLMark 信息和 PL 时间轴。在该第 1 层中, 存在 2 个 PLMark 信息 #1 ~ #2。箭头 kt1、2 表示用 PLMark 信息的 ref_to_PlayItem 进行的指定。由该箭头可知, PLMark 信息的 ref_to_PlayItem_Id 指定了 PlayItem 信息的每一个。此外, Mark_time_stamp 表示 PlayItem 时间轴中应成为 Chapter#1、#2 的时刻。这样, PLMark 信息能够在 PlayItem 时间轴上定义章节节点。

[0203] 能够定义可使 AVClip-SubClip 同步的同步区间, 是 BD-ROM 上的播放列表信息的特征。以上的 Clip 信息及播放列表信息被分类为“静态脚本”。这是因为, 用以上的 Clip 信息及播放列表信息定义了作为静态的再现单位的播放列表。至此结束对静态脚本的说明。

[0204] 接着说明“动态脚本”。所谓动态脚本, 是动态规定 AVClip 的再现控制的脚本数据。所谓“动态”, 是指通过再现装置中的状态变化或来自用户的键事件而知道再现控制的内容。对于 BD-ROM, 设想了 2 个模式作为该再现控制的动作环境。第 1 个是与 DVD 再现装置的动作环境类似的动作环境, 是基于命令的执行环境。第 2 个是 Java(注册商标) 虚拟机的动作环境。这 2 个动作环境中的第 1 个被称为 HDMV 模式。第 2 个被称为 BD-J 模式。由于有这 2 个动作环境, 所以设想其中某一个动作环境来描述动态脚本。设想了 HDMV 模式的动态脚本被称为 Movie Object(电影对象)。而设想了 BD-J 模式的动态脚本被称为 BD-J Object。

[0205] 首先说明 Movie Object。

[0206] <Movie Object>

[0207] Movie Object 被保存在图 2 所示的 MovieObject.bdmv 这一文件中, 包含导航命令串。

[0208] 导航命令串由实现条件转移、再现装置中的状态寄存器的设定、状态寄存器的设定值取得等的命令串组成。在 Movie Object 中可描述的命令如下所示。

[0209] PlayPL 命令

[0210] 格式 :PlayPL(第 1 变量, 第 2 变量)

[0211] 第 1 变量是播放列表的号码,能够指定应再现的播放列表。第 2 变量能够用该播放列表中包含的 PlayItem、或该播放列表中的任意的时刻、Chapter(章节)、Mark(标记)来指定再现开始位置。

[0212] 将用 PlayItem 指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatPlayItem(),

[0213] 将用 Chapter 指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatChapter(),

[0214] 将用时刻信息指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatSpecifiedTime()。

[0215] JMP 命令

[0216] 格式:JMP 变量

[0217] JMP 命令是在途中废弃(discard)当前的动态脚本、执行作为变量的转移目的地动态脚本的转移。在 JMP 命令的形式中,有直接指定转移目的地动态脚本的直接参照的形式、和间接指定转移目标动态脚本的间接参照的形式。

[0218] Movie Object 中的导航命令的描述与 DVD 中的导航命令的描述方式类似,所以能够高效地进行将 DVD 上的盘内容移植到 BD-ROM 上这一作业。对于 Movie Object,存在以下的国际公开公报中记载的现有技术。详细情况请参考该国际公开公报。

[0219] 国际公开公报 WO 2004/074976

[0220] 至此结束对 Movie Object 的说明。接着说明 BD-J Object。

[0221] <BD-J Object>

[0222] BD-J Object 是在 Java(注册商标)编程环境下描述的、BD-J 模式的动态脚本,被保存在 00001 ~ 00003.bobj 这些文件中。

[0223] 图 13 是 BD-J Object.bdmv 的内部结构的图。由应用管理表(AMT)、播放列表管理表(PLMT)、声音管理表(SMT)组成。与 Movie Object 的不同点是在 BD-J Object 中未直接描述命令。即,在 Movie Object 中,控制过程由导航命令直接描述。而在 BD-J Object 中,通过对 Java(注册商标)应用的指定记载到应用管理表中,而间接规定了控制过程。通过这种间接的规定,能够高效地进行在多个动态脚本中共用控制过程这样的控制过程的共用。

[0224] 此外,Movie Object 中的播放列表再现通过描述命令再现播放列表的导航命令(PlayPL 命令)来进行,而 BD-J Object 中的播放列表再现可以通过将表示播放列表再现过程的播放列表管理表包含到 BD-J Object 中来描述。

[0225] 下面说明该 BD-J 模式下的 Java(注册商标)应用。这里,BD-J 模式设想的 Java(注册商标)平台完全安装了 Java(注册商标)2 Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0)、和 Globally Executable MHP specification(GEM 1.0.2) for package media targets。

[0226] 该 BD-J 模式下的 Java(注册商标)应用通过 xlet 接口由 ApplicationManager(应用管理器)来控制。xlet 接口具有“loaded(已加载)”、“paused(已暂停)”、“active(激活)”、“destroyed(已毁灭)”这 4 个状态。

[0227] 上述 Java(注册商标)平台包含用于表示 JFIF(JPEG)或 PNG 等图像数据的标准

Java(注册商标)库。因此,Java(注册商标)应用能够实现与在HDMV模式下用IG流实现的GUI不同的GUI框架。Java(注册商标)应用中的GUI框架包含GEM1.0.2中规定的HAVi框架,包含GEM1.0.2中的遥控导航机构。

[0228] 由此,Java(注册商标)应用能够实现将基于HAVi框架的按钮显示、文本显示、在线显示(BBS的内容)等显示与运动图像的显示组合而成的画面显示,能够用遥控器对该屏幕显示进行操作。

[0229] 相当于该Java(注册商标)应用的实体的是图2中的BDMV目录属下的BDJA目录中保存的Java(注册商标)存档文件(00001.jar,00002.jar)。以下,说明Java(注册商标)存档文件。

[0230] <Java(注册商标)存档文件>

[0231] Java(注册商标)存档文件(图2的00001.jar,00002.jar)是通过将1个以上的类文件、1个以上的数据文件等合并为1个而得到的文件,构成应在BD-J模式下动作的Java(注册商标)应用。

[0232] 图14是存档文件容纳的程序、数据的图。本图中的程序、数据用java(注册商标)归档器(アーカイバ)合并了布置成框内所示的目录结构的多个文件。框内所示的目录结构由Root目录、Java(注册商标)1、2、3目录、Image1、2、3目录组成,在Root目录中布置着common.pkg,在Java(注册商标)1、Java(注册商标)2、Java(注册商标)3目录中布置着类文件(00001.class~00007.class),在Image1、Image2、Image3目录中布置着00001.JPG~00003.JPG、00004.PNG~00006.PNG。java(注册商标)存档文件是通过用java(注册商标)归档器合并它们而得到的。这种类文件及数据在从BD-ROM上读出到高速缓冲存储器时被展开,在高速缓冲存储器上,被看作目录中布置的多个文件而被处理。Java(注册商标)存档文件的文件名中的“zzzzz”这5位的数值表示应用的ID(applicationID)。在本Java(注册商标)存档文件被读出到高速缓冲存储器时,通过引用该文件名中的数值,能够取出构成任意的Java(注册商标)应用的程序、数据。

[0233] 其中,在本实施方式中构成应用的程序、数据被合并为Java(注册商标)存档文件,但是也可以是LZH文件、zip文件。

[0234] 以上是对BD-J模式下的动态脚本的说明。

[0235] <sound.bdmv>

[0236] 接着说明sound.bdmv。sound.bdmv是保存着在对Java(注册商标)应用的GUI框架进行了操作的情况下、应作为点击声来输出的音频数据(将这种音频数据称为声音数据)的文件。

[0237] 图15是文件sound.bdmv的结构图。文件sound.bdmv由Sound Data()和Sound Index()组成。Sound Data()由多个声音数据(sound_data(0),sound_data(1))组成。这些声音数据中的sound_data(0)是对Java(注册商标)应用的GUI框架进行操作时作为第1点击声来输出的音源。sound_data(1)是对Java(注册商标)应用的GUI框架进行操作时作为第2点击声来输出的音源。这些声音数据用称为sound_ID的标识符来指示。

[0238] Sound Index()由声音数(number_of_sound_entries)、对于sound_data(0)的索引、对于sound_data(1)的索引组成。

[0239] 索引由单声道/立体声之别等各声音的属性(sound_attributes)、对应的声音

数据的地址 (sound_data_start_address)、对应的声音数据的连续长度 (sound_data_length) 组成。

[0240] 如图 2 ~ 图 6 所示,电影的剧中使用的音频(台词、BGM、效果音)的音源作为音频流被复用 AVClip 内。这是为了将表示电影剧中的声音的音频流,与视频流的读出同时提供给再现装置。而文件 sound.bdmv 作为与 AVClip 独立的文件被记录在 BD-ROM 上。文件 sound.bdmv 作为与 AVClip 独立的文件被记录,所以在 AVClip 的读出中,如果想要输出声音数据,则发生用于读出文件 sound.bdmv 的光拾取器的跳跃,AVClip 的读出不得不中断。如果发生这种中断,则 AVClip 再现出现中断。

[0241] 为了避免这种 AVClip 的再现中断,需要在未再现 AVClip 的时刻将文件 sound.bdmv 预先预加载到缓冲器中。即,需要在再现 AVClip 之前,预加载文件 sound.bdmv 内的声音数据。以上是对文件 sound.bdmv 的说明。

[0242] <Index.bdmv>

[0243] Index.bdmv 是表示构成标题的 Movie Object 或 BD-J Object 的表。

[0244] 在 Title(标题)中,定义作为某个 Title 的构成要素的 MovieObject 是哪个,或者作为某个 Title 的构成要素的 BD-J Object 是哪个。

[0245] 在以下的国际公开公报中记载了 Index.bdmv 的详细情况。详细情况请参考该公报。

[0246] 国际公开公报 W0 2004/025651 A1 公报

[0247] 以下,更详细地分别说明图 15 所示的应用管理表、播放列表管理表、声音管理表的每一个。

[0248] <应用管理表>

[0249] 下面说明应用管理表 (AMT)。所谓应用管理表 (AMT),是实现了上述 GEM1.0.2 for package media targets 中的“应用信令 (application signaling)”的表。所谓“应用信令”,是指在 GEM1.0.2 规定的 MHP (Multimedia HomePlatform:多媒体家庭平台) 中将“服务”作为生命周期来进行应用的运行、执行的控制。本实施方式中的应用管理表不是将该“服务”,而是将 BD-ROM 上的“标题”作为生命周期,来实现应用的运行、执行的控制。

[0250] 图 16(a) 是应用管理表的内部结构的图。如本图所示,应用管理表由“life_cycle”、“apli_id_ref”、“run_attribute”、以及“run_priority”组成。

[0251] 图 16(b) 是构成应用管理表的信息元素的意义内容的图。

[0252] “life_cycle”表示应用的“生命周期”。

[0253] “apli_id_ref”通过描述对“应用标识符”的参照值,来表示具有下述生命周期的应用是哪个。应用标识符在 Java(注册商标)存档文件中用作为文件名而赋予的 5 位数值 zzzzz 来表现。在“apli_id_ref”中,描述该 5 位数值。

[0254] “run_attribute”描述该生命周期中的应用的“运行属性”。在运行属性中,有 AutoRun、Present、Suspend 等种类。

[0255] “run_priority”描述该生命周期中的应用的“运行优先级”。在 BD-JObject 中,用这些信息来控制应用的举动。

[0256] <生命周期>

[0257] 下面说明应用管理表中规定的信息中的生命周期。

[0258] 所谓生命周期,表示在 BD-ROM 上记录的整个内容的时间轴上、在虚拟机的工作存储器上应用可生存的区间。所谓工作存储器上的“生存”,是指构成该应用的 xlet 程序被读出到 Java(注册商标)虚拟机内的工作存储器上、成为 Java(注册商标)虚拟机可执行的状态。

[0259] 在 Java(注册商标)虚拟机中使应用动作的情况下,明确规定从时间轴的何处开始基于应用的服务、在时间轴的何处结束基于应用的服务这样的“服务的开始点·结束点”很重要。规定该服务的开始点·结束点的就是应用管理表中的生命周期。

[0260] 另一方面,用 DVD-Video 这样的只读盘提供的盘内容采用了以顶级菜单标题为核心的构造。进行下述独特的状态转移:从该顶级菜单标题转移到各个作品来进行再现,其后再次返回到顶级菜单标题。图 17 是盘内容中的状态转移的图。本图中的四方框是 Title。所谓 Title,是在盘内容特有的状态转移中与 1 个“状态”相当的再现单位,该标题被看作 Java(注册商标)应用的生命周期来处理。

[0261] 在 Title 中,有装入 BD-ROM 时首先再现的“FirstPlayTitle”、构成 Top-Menu(顶级菜单)的“Top_menuTitle”、其他一般的“Title”。此外,图中的箭头 jh1、2、3、4、5、6、7、8 象征性地表示 Title 间的转移。本图所示的所谓状态转移,是在装入 BD-ROM 时再现“FirstPlayTitle”,发生向“Top_menuTitle”的转移,等待对顶级菜单的选择。

[0262] 如果用户对顶级菜单进行了选择操作,则根据选择来再现该 Title,再次返回到 TopMenu Title;不断地重复进行上述处理,直至弹出 BD-ROM,这是盘内容特有的状态转移。

[0263] 那么,在进行图 17 那样的状态转移的盘内容中,Title 如何被规定为生命周期呢?假设在装入了 BD-ROM 后,按图 17 中箭头 jh1、2、3、4……所示的参照标号的数值顺序来进行转移,弹出了 BD-ROM。这样,装入 BD-ROM 后,能够将弹出之前的连续时间带看作一根时间轴。将该时间轴设为整个盘的时间轴。图 18(a) 是整个盘的时间轴的图,图 18(b) 示出该时间轴的结构。如图 18(b) 所示,整个盘的时间轴由再现 FirstPlay Title 的区间、再现 TopMenu Title 的区间、再现 title#1 的区间等组成。至于这些 Title 的再现区间如何规定,则由于 Title 由唯一的 BD-J Object 构成,所以可以将某个 MovieObject 或 BD-J Object 有效的期间看作 Title 的再现区间。

[0264] 即,FirstPlay Title、TopMenu Title、其他 Title 都由动态脚本构成,所以可以将构成 Title 的 BD-J Object 中的某个作为当前 BD-J Object 来激活、在再现装置内供解释·执行的期间定义为 Title 的再现区间。图 19(a) 是在整个 BD-ROM 的时间轴上、由标识符 bobj_id 确定的 BD-J Object 所确定的标题再现区间的图。这里,如果由标识符 bobj_id 确定的 BD-J Object 构成 1 个 Title,则可以将由该标识符 bobj_id 确定的 BD-J Object 有效的 BD-ROM 时间轴上的一个区间看作 Title 的再现区间。

[0265] 这里,BD-J Object 被激活的期间的期间终点是到进行 Title 转移为止。即,在进行 Title 转移之前,作为执行的对象的动态脚本被作为当前 BD-J Object 处理,所以将在该 BD-J Object 中发生 JumpTitle 之前的 1 个区间看作 Title 区间。

[0266] 接着说明 Title 区间和 PL 时间轴之间的关系。如上所述,在 MovieObject、BD-J Object 中,可以描述播放列表再现过程作为 1 个处理过程。如果有播放列表再现过程的描述,则上述 PL 时间轴的全部或部分就归属于 Title 区间。假设在图 19(a) 的一例中,在 BD-J Object 中描述了播放列表管理表。在此情况下,如图 19(b) 所示,PL 时间轴归属于与

BD-J Object对应的Title区间。在该PL时间轴上,还可定义多个章节(Chapter#1,#2,#3),所以在BD-ROM上的时间轴上,存在整个BD-ROM-Title-播放列表-章节这些域(domain)。可以用这些域来描述应用的生命周期。其中,再现播放列表与执行应用同时进行,所以有时在再现播放列表的途中发生Title转移。在此情况下,不是整个播放列表时间轴,而是只有播放列表时间轴的一部分归属于1个Title再现区间。即,整个播放列表、还是其一部分归属于1个Title的再现区间,这根据何时发生Title转移来变化。

[0267] 图20是图19(b)的时间轴上规定的生命周期的典型的图。如本图所示,在应用中,有下述3个典型:将Title作为生命周期的“标题边界应用(titleboundary application)”、将Title内的章节作为生命周期的“章节边界应用(chapter boundary application)”、将整个BD-ROM的时间轴作为生命周期的“标题无边界应用(title unboundary application)”。

[0268] 其中,标题边界应用的生命周期可以用该标题的标识符来定义。而章节边界应用的生命周期可以用章节所属的标题的标识符、和该章节的标识符的组来定义。

[0269] 即使平台进行了动作,如果Title或章节这样的生命周期结束,则也可以从应用回收资源。由于能保证资源回收的机会,所以能够稳定平台的动作。

[0270] 将近期的将来可能实施的盘内容选为题材,结合具体例来说明应用管理表中的生命周期描述。这里,作为题材的盘内容包含构成电影正片的正片标题(title#1)、构成在线购物的在线购物标题(title#2)、构成游戏应用的游戏标题(title#3)这3个性质不同的标题。图21是包含正片标题、在线购物标题、游戏标题这3个标题的盘内容的图。在本图中的右侧描述了Index.bdmv,在左侧描述了3个标题。

[0271] 右侧的虚线框表示各应用属于哪个标题这一归属关系。3个标题中的title#1由application#1、application#2、application#3这3个应用组成。title#2包含application#3、application#4这2个应用, title#3包含application#5。在图21的一例中, application#3由title#1、title#2双方来运行。

[0272] 根据图21的虚线所示的归属关系将各应用的生命周期图表化时,则如图22(a)所示。在本图中,横轴是标题再现区间,在纵轴方向上布置了各应用的生命周期。这里, application#1、application#2只归属于title#1,所以它们的生命周期仅限于title#1内。 application#4只归属于title#2,所以它的使用寿命仅限于title#2内。 application#5只归属于title#3,所以它的使用寿命仅限于title#3内。 application#3归属于title#1及title#2,所以它的使用寿命跨title#1-title#2。如果根据该生命周期来描述应用管理表,则title#1、#2、#3的应用管理表如图22(b)所示。如果这样描述应用管理表,则在开始再现title#1时将application#1、application#2、application#3加载到工作存储器中。随后在开始title#2时,从工作存储器中删除application#1、application#2而只剩application#3。与此同样可进行下述控制:在开始再现title#2时将application#4加载到工作存储器中,在开始title#3时,从工作存储器中删除application#3、#4。

[0273] 再者,可进行下述控制:在title#3的再现中将application#5加载到工作存储器中,在title#3的再现结束时,从工作存储器中删除application#5。

[0274] 在有标题间转移的情况下,也只要将在转移源一转移目标中生存着的应用保存到

工作存储器上,将转移源中没有、只存在于转移目标的应用读入到工作存储器中即可,所以将应用读入到工作存储器中的次数为所需最低数。这样,通过减少读入次数,能够实现不使人意识到标题的边界的应用、即无边界(アンバウンダリ)的应用。

[0275] 接着更详细地说明应用的运行属性。在运行属性中,有表示自动运行的“AutoRun”,表示虽然不是自动运行的对象、但是可以放在虚拟机的工作存储器中的“Present”,虽然能放在虚拟机的工作存储器中、但是不能分配 CPU 性能(power)的“Suspend”。

[0276] “AutoRun”是表示与对应的标题的转移同时、将该应用读入到工作存储器中、而且执行的属性。如果有从某个标题到另一个标题的转移,则进行应用管理的管理主体(应用管理器)将在该转移目标标题中生存着的、而且运行属性被设定为 AutoRun 的应用读入到虚拟机的工作存储器中并执行。由此,该应用与标题转移一起自动被运行。

[0277] 运行属性“Present”是继续属性,表示继续转移源 title 中的应用的状态。此外,是表示可以执行对应的应用的属性。在运行属性是“Present”的情况下,被赋予了该运行属性的应用被允许从其他应用调用。如果从运行中的应用进行了调用,则进行应用管理的管理主体(应用管理器)判断是否该应用的 applicationID 在应用管理表中被描述、运行属性是“Present”。如果是“Present”,则将该应用加载到工作存储器中。而在该被调用应用的 applicationID 在应用管理表中未被描述的情况下,不将该应用加载到工作存储器中。应用进行的调用限于被赋予了该“Present”的应用。“Present”是未明确指定运行属性的情况下赋予的缺省的运行属性,所以在某个应用的运行属性是未指定“—”的情况下,表示该应用的运行属性是该 Present。

[0278] 所谓“Suspend”,是指虽然分配了资源,但是应用被置于未分配 CPU 性能的状态。这种 Suspend 在实现例如在游戏标题的执行中经由侧道(サイドパス)这一处理时有意义。

[0279] 图 23 是运行属性可取的三态(Present、AutoRun、Suspend)、紧挨着的前面的标题中的应用状态的三态(未运行、运行中、Suspend)可取的组合的图。在紧前面的状态是“未运行”的情况下,如果运行属性是“AutoRun”,则在转移目标标题中该应用被运行。

[0280] 如果紧前面的状态是“未运行”、运行属性是“Present”、“Suspend”,则在转移目标标题中该应用什么也不做,继续原状态。

[0281] 在紧前面的状态是“运行中”的情况下,如果运行属性是“Present”、“AutoRun”,则在转移目标标题中该应用什么也不做,继续原状态。

[0282] 如果运行属性是“Suspend”,则应用的状态被变为 Suspend。在紧前面的状态是“Suspend”的情况下,如果转移目标标题的运行属性是“Suspend”,则维持 Suspend。如果是“Present”或“AutoRun”,则在转移目标标题中恢复该应用。通过在应用管理表中定义生命周期及运行属性,能够进行使 Java(注册商标)应用沿标题再现区间的进行而动作的同步控制,能够将伴有影像再现和程序执行的各种各样的应用实现。

[0283] 其中,在紧前面的状态是“Suspend”、转移目标标题的运行属性是“Present”的情况下,也可以维持紧前面的状态、即挂起状态。

[0284] 最后,说明各应用的“运行优先级”。

[0285] 该运行优先级取 0 ~ 255 的值,成为应用管理器在存储器资源枯竭时、或 CPU 负担

增高时进行强制性地结束哪个应用、并且从哪个应用那里夺取资源的处理时的判断依据。在此情况下,应用管理器进行结束运行优先级低的应用的动作、使运行优先级高的应用继续动作这一处理。

[0286] 此外,运行优先级也用于对再现中播放列表的请求有竞争的情况下的应用间的仲裁。这里,假设某个应用快进某个播放列表。这里,如果别的应用请求暂停该播放列表,则比较赋予给这些应用的运行优先级。如果命令了快进的应用的运行优先级高,则继续该应用进行的快进。相反,如果命令暂停的应用的运行优先级高,则暂停快进中的播放列表。

[0287] 用以上的生命周期·运行属性·运行优先级,能够在创作时规定将虚拟机上可动作的应用的数目限制在规定数以下。因此,能够保证应用的稳定动作。

[0288] <播放列表管理表>

[0289] 以上是对应用管理表的说明。接着说明播放列表管理表(PLMT)。所谓播放列表管理表,是表示在应用的生命周期中,应与各应用的执行同时进行的再现控制的表。应用的动作不稳定,可能有运行的失败或异常结束。因此,作为运行失败、异常结束的情况下的 Fail Safe(故障保险)机构,在本实施方式中,为应用的每个生命周期,设有播放列表管理表。播放列表管理表是规定在某个应用的生命周期开始时应与其同时进行的再现控制的信息。该所谓再现控制,是基于播放列表信息的 AVClip 再现,通过同时进行基于播放列表信息的再现控制,能同时进行应用执行和播放列表再现。虽然假设为应用的每个生命周期设有播放列表管理表,但是设有播放列表管理表的应用限于标题边界的应用。这是因为,标题无边界应用将全部标题作为生命周期,所以不适合与应用的执行同时来进行播放列表再现。

[0290] 章节边界应用在从 1 个播放列表内的章节起开始执行应用这一前提下规定了生命周期,所以无需规定播放列表再现。由以上事实可知,播放列表管理表被定义为由 1 个以上的 Title 组成的生命周期。

[0291] 图 24(a) 是播放列表管理表的内部结构的图。如本图所示,播放列表管理表由“PL_id_ref”和“Playback_Attribute”组成。

[0292] 图 24(b) 是构成播放列表管理表的信息元素的意义内容的图。

[0293] “PL_id_ref”通过描述对播放列表标识符的“参照值”,来表示在应用的生命周期中可再现的播放列表是哪个。播放列表标识符用文件 YYYYY.MPLS 中作为文件名而赋予的 5 位数值 YYYYY 来表现。通过描述该 YYYYY,“PL_id_ref”表示在对应的 Title 中可再现的播放列表是哪个。

[0294] “Playback_Attribute”是仿照了应用管理表中的运行属性的属性。是规定在标题开始时如何再现“PL_id_ref”中描述的播放列表的再现属性。在对于播放列表的再现属性中,有“AutoPlay”、“Present”等种类。

[0295] 所谓“AutoPlay”,是表示与对应的标题的转移同时来再现该播放列表的属性。如果有从某个标题向别的标题的转移,则进行应用管理的管理主体(应用管理器)开始在该转移目标标题中可再现、而且再现属性被设定为 AutoPlay 的播放列表的再现。由此,运行属性被设定为 AutoPlay 的播放列表与标题转移一起自动被运行。

[0296] 所谓“Present”,与运行属性中的 Present 同样,是继续属性,表示继续转移源 title 中的播放列表的状态。此外,是表示可以再现对应的播放列表的属性。例如假设有连续再现的 2 个 Title,在上一标题的播放列表管理表中,某个播放列表的再现属性被设定

为 AutoPlay, 在当前标题的播放列表管理表中, 该播放列表的再现属性被设定为 Present。这里, 假设播放列表的再现时间是 2 个小时长, 其中在经过了 1 个小时的时刻发生了转移。在此情况下, 在当前标题中, 再现属性被设定为 Present, 所以在当前标题中, 该播放列表从 1 个小时这一已再现区间的后面起被再现。如果这样将再现属性设定为 Present, 则在有 Title 间的转移的情况下, 也能够从其剩余的部分起开始播放列表再现。由此, 在相互转移的一系列 Title 中, 能够容易地实现再现共用的播放列表这一“标题间的播放列表再现的共用化”。此外, 在有多个转移目标标题的情况下, 如果将这多个标题的再现属性都设为 Present, 则不管转移到多个中的哪个, 都能够继续 1 个共用的播放列表再现。

[0297] 其中, Title 的边界也可以不保证无缝再现, 所以在如上所述想要在多个 Title 间再现 1 个播放列表的情况下, 允许在转移前后中断播放列表再现。

[0298] 此外, 在再现属性是“Present”的情况下, 被赋予了该再现属性的播放列表根据来自其他应用的再现请求被再现。如果从运行中的应用发出了再现播放列表的请求, 则进行应用管理的管理主体(应用管理器)判断是否受到请求的播放列表的 PL_id_ref 在播放列表管理表中有描述、再现属性是“AutoPlay”或“Present”中的某一个。如果是“AutoPlay”或“Present”中的某一个, 则再现该播放列表。而在受到请求的播放列表的 PL_id_ref 在播放列表管理表中未描述的情况下, 不再现该播放列表。根据应用的请求来再现播放列表限于被赋予了该“AutoPlay”或“Present”中的某一个的播放列表。“Present”是未明确指定再现属性的情况下赋予的缺省的再现属性, 所以如果某个播放列表的再现属性是未指定“—”, 则表示该播放列表的再现属性是该 Present。

[0299] 图 25 示出用播放列表管理表、应用管理表规定的标题的具体例。图 25 的第 1 层表示 Title 的再现影像, 第 2 层表示 Title 的时间轴。第 3 层表示用 PLMT 规定了再现的播放列表, 第 4 层表示应用执行。在第 4 层中, application#1 随着 Title 的开始被运行, 其后, 在时刻 t1 成为动作状态。另一方面, Playlist#1 随着 Title 的开始被开始再现。Playlist#1 的再现在与 Title 的开始相同的时刻开始, 所以如第 1 层的左侧所示, 在从 Title 的再现刚刚开始之后、到应用成为动作状态的启动延迟中, 播放列表的再现图像 gj1 被全屏显示。通过将播放列表管理表的再现属性设定为“AutoPlay”, 即使 Java(注册商标)应用成为动作状态为止花费了 5 ~ 10 秒这一时间, 在此期间, 也处于“暂且显示了些东西的状态”。用该“暂且显示了些东西的状态”能够补偿开始执行标题时的启动延迟。

[0300] 另一方面, application#1 在时刻 t1 成为动作状态, 所以将播放列表再现图像作为子画面、将应用的执行图像作为母画面而成的合成图像 gj2 在时刻 t1 被显示。应用的执行图像是布置了 Start(开始)按钮、continue(继续)按钮、POWER(电源)指示器的游戏用的 GUI 框架, 通过使 Java(注册商标)执行这种 GUI 框架的绘制处理来显示。

[0301] 能够构成这种将播放列表的再现影像、和 Java(注册商标)应用的 GUI 框架组合而成的再现影像的标题, 是 PLMT 的特征。

[0302] 图 26 是当前标题可取的三态(无播放列表管理表(i)、有播放列表管理表而且 AutoPlay(ii)、有播放列表管理表而且未指定(iii))、和紧挨着的前面的标题中的播放列表的状态(未再现状态、再现中状态)可取的 6 种组合的图。

[0303] 在本图中的 6 种组合中的“紧挨着的前面的状态=未再现状态”、“当前标题=有播放列表管理表, 而且当前标题的再现属性=AutoPlay”的组合中, 转移目标标题中的播放列

表的再现自动开始。

[0304] 此外,在“紧挨着的前面的状态=再现中状态”和“当前标题=无播放列表管理表”的组合中,转移目标标题中的播放列表的再现自动停止。

[0305] 这2种组合以外全部是继续前一标题的状态。根据播放列表管理表来开始再现播放列表,仅限于在转移源标题中是未再现状态、在转移目标标题中被赋予了 AutoPlay 属性的情况,所以无需每当发生标题的转移就开始再现播放列表。即使发生了许多标题间的转移,也能够使开始再现播放列表的次数为所需最低数。

[0306] 下面参考图 27(a) 来说明播放列表管理表及应用管理表的描述例。这里设想的具体例是2个连续的 Title(title#1、title#2),在其中的 title#1 中,作为 AutoRun 应用,描述了 application#1、application#2。在 title#2 中,作为 AutoRun 应用,描述了 application#2、application#3。另一方面,假设在 title#1 的播放列表管理表中,作为 AutoPlay 播放列表,描述了 PlayList#1;在 title#2 的播放列表管理表中,作为 AutoPlay 播放列表,描述了 PlayList#2。图 27(b) 是表示如何根据如图 27(a) 所示描述的应用管理表、播放列表管理表来进行播放列表再现、应用执行的图。

[0307] 在 title#1 中如上所述设定了应用管理表、播放列表管理表,所以在开始 title#1 时自动运行 application#1、application#2,自动开始再现 PlayList#1。

[0308] 在 title#2 中如上所述设定了应用管理表、播放列表管理表,所以使虽然在 title#1 侧有记载、但是在 title#2 侧没有记载的 application#1 的执行停止。同样,也使在 title#1 侧有记载、但是在 title#2 侧没有记载的 PlayList#1 的再现也停止。

[0309] 在 title#1 侧没有记载、但是在 title#2 侧有记载的 PlayList#2、application#3 自动开始再现及执行。如果有标题转移,则可以以该转移为机会,将应再现的播放列表切换到其他播放列表。通过这样采用应用管理表、播放列表管理表,能够在创作阶段规定以转移为机会来切换播放列表再现这一处理。

[0310] 此外,在图 27 中,向 application#1、application#2、application#3 分别给予了 200、128、200 的运行优先级。通过赋予这些运行优先级,能够进行对 PlayList#1、PlayList#2 的控制请求有竞争的情况下的仲裁。这里假设 application#1 命令 PlayList#1 快进。另一方面,假设 application#2 请求暂停。在此情况下,由于在应用管理表中规定了对于各应用的运行优先级,所以能根据该运行优先级对两个应用进行仲裁。其结果是,能够在创作时规定驳回 application#2 的请求、继续 application#1 的控制这一处理。通过与播放列表管理表一起来利用运行优先级,也能够使再现装置进行在对播放列表的控制有竞争的情况下的仲裁。

[0311] 下面说明播放列表管理表描述的另一具体例。图 28(a) 是播放列表管理表的另一描述例的图。本图中设想的是,在2个连续的标题(title#1、title#2)中,在 title#1 侧的播放列表管理表中,作为 AutoPlay 播放列表,描述了 PlayList#1,作为可再现的播放列表,描述了 PlayList#2;而在 title#1 侧的应用管理表中,作为 AutoPlay 应用,描述了 application#1,作为可执行的应用,描述了 application#2。另一方面,在 title#2 侧的播放列表管理表中,作为可再现的播放列表,描述了 PlayList#2、PlayList#3;在应用管理表中,作为 AutoRun 应用,描述了 application#3。图 28(b) 是根据图 28(a) 的例子来进行应用执行及播放列表再现的图。在 title#1 的应用管理表中,作为 AutoRun 应用,描述了

application#1,所以在开始 title#1时自动运行 application#1。另一方面,在 title#1的应用管理表中,作为可执行的应用,描述了 application#2,所以通过来自 application#1的调用 yd1 来运行 application#2。

[0312] 在 title#2 侧的应用管理表中 application#1、application#2 为非生存,而作为 AutoRun 应用,描述了 application#3。因此,在 title#1-title#2 的边界部,进行停止 application#1、application#2、自动运行 application#3 这一处理。参考播放列表管理表可知, title#1 侧的播放列表管理表描述了 PlayList#1、PlayList#2 可再现,其中 PlayList#1 为 AutoPlay 属性。因此,PlayList#1 在开始 title#1 时自动被再现。

[0313] 在 title#1 侧的播放列表管理表中,描述了除了 PlayList#1 之外,PlayList#2 可再现,所以 application#1 通过停止再现 PlayList#1,而请求再现 PlayList#2,能够执行播放列表交替。

[0314] 在 title#2 侧的播放列表管理表中,作为可再现的播放列表,描述了 PlayList#2、PlayList#3。此外,没有被赋予了 AutoPlay 属性的播放列表。因此,假定在开始 title#1 时自动再现的 PlayList#1 的再现持续到 title#2, PlayList#1 的再现也自动结束。

[0315] 但是,如果在 PlayList#2 再现继续的状态下到达 title#2,则 PlayList#2 再现在 title#2 开始以后也继续。在 title#2 的播放列表管理表中,作为可再现的播放列表,描述了 PlayList#2、PlayList#3。因此,在 title#2 中成为执行中的 application#3 通过停止再现 PlayList#2,取而代之请求再现 PlayList#3,能够交替再现中的播放列表。

[0316] 如果如上所述将播放列表管理表的再现属性设定为“AutoPlay”,则即使 Java(注册商标)应用的运行花费了 5~10 秒这一时间,在进行该运行的期间,也处于“暂且显示了些东西的状态”。在开始执行标题时,即使应用运行花费了时间,画面也处于“暂且显示了些东西的状态”。由此,能够补偿由于运行应用花费时间而造成的启动延迟的延长。

[0317] 通过定义应用管理表及播放列表管理表,能够进行使 Java(注册商标)应用沿标题再现区间的进展来动作的同步控制,能够将伴有影像再现和程序执行的各种各样的应用实现。

[0318] < 声音管理表 >

[0319] 以下说明声音管理表(SMT)。

[0320] 图 29 是 SMT 的内部结构的图。如本图所示,声音管理表包含 Mixing_On 标志。所谓 Mixing_On 标志,是通过被设定为 0 或 1、来表示使混音有效还是无效的标志。所谓“有效”,是指对音频流的再现声音进行了混合后输出;所谓“无效”,是指直接输出音频流的再现声音。

[0321] 在 Mixing_On 标志被设定为 1(ON)的情况下,表示将基于文件 sound.bdmv 的点击声混合到用属于同一 BD-J Object 的 PLMT 再现的播放列表的再现声音中。

[0322] 在 Mixing_On 标志被设定为 0(OFF)的情况下,表示不将基于文件 sound.bdmv 的点击声混合到用属于同一 BD-J Object 的 PLMT 再现的播放列表的再现声音中。

[0323] 可以以 Java(注册商标)应用的生命周期为单位、即以标题为单位来规定是否混合基于文件 sound.bdmv 的点击声。而将该 Mixing_On 标志设定为何种值,与该播放列表具有的 STN_Table 的设定内容相应。

[0324] 图 30(a)(b) 是 STN_Table 中的音频流的 entry-attribute、和 Mixing_On 标

志的设定之间的因果关系的图。在图 30(a) 中,如果 entry-attribute 中的 audio_presentation_type 表示“多声道”,则 Mixing_On 标志被设定为 0(OFF)。在图 30(b) 中,如果 entry-attribute 中的 audio_presentation_type 表示“非多声道”,则 Mixing_On 标志被设定为 1(ON)。通过进行以上那样的设定,在同时执行再现“多声道”的播放列表 (PlayItem) 和应用的情况下,能够使与再现装置中的声音再现的混合无效;而在同时执行再现“非多声道”的播放列表 (PlayItem) 和应用的情况下,能够使与再现装置中的声音再现的混合有效。

[0325] 以下,说明对图 27(a) 中的声音管理表的一例如何进行再现。假设图 27(a) 中的 Title#1、#2 中的 Title#1 的 Mixing_On 标志表示使与声音再现的混合有效,Title#2 中的声音管理表 #2 的 Mixing_On 标志表示使与声音再现的混合无效。

[0326] 再者,假设在再现 Title#1 时,输出了图 25 所示的 GUI 框架、和播放列表的再现影像。用户经遥控器如图 31 所示操作了该 GUI 框架中的按钮。在此情况下,如果绘制了该 GUI 框架的 Java(注册商标) 应用的应用管理表、和属于同一 BD-J Object 的声音管理表被设定为 Mixing_On 标志 = 1,则文件 sound.bdmv 中的声音数据被混合到播放列表的再现声音中。

[0327] 这里,如果播放列表的再现声音是电影剧中的 BGM,文件 sound.bdmv 中的声音数据是“开始游戏了!”这一语音引导,则该电影剧中的 BGM 和语音引导被混合并输出。

[0328] 再者,假设在再现 Title#2 时,输出了与图 25 不同的 GUI 框架和播放列表的再现影像。此外,假设用户经遥控器如图 32 所示操作了该 GUI 框架中的按钮。在此情况下,如果绘制了该 GUI 框架的 Java(注册商标) 应用的应用管理表、和属于同一 BD-J Object 的声音管理表被设定为 Mixing_On 标志 = 0,则文件 sound.bdmv 中的声音数据不被混合到播放列表的再现声音中。这里,如果播放列表的再现声音是电影剧中的多声道的 BGM,则只输出该电影剧中的 BGM。

[0329] <Mixing_On 标志的意义>

[0330] 即使在想要再现的 AVClip 上复用了 5.1Ch 的环绕声的音频流、和单声道的音频流,在 PlayItem 的 STN_Table 中允许再现该单声道的音频流、不允许再现 5.1Ch 的环绕声的音频流的情况下,也能够将与该 PlayItem 对应的 Mixing_On 标志设为 ON,执行与点击声的混合。通过这样做,即使 5.1Ch 的环绕声的音频流、和单声道的音频流构成了同一 AVClip,在用不许再现 5.1Ch 的环绕声的 PlayItem 来再现该 AVClip 这样的情况下,也能够执行再现该 PlayItem 时的点击声的混合。

[0331] 如果考虑以上事实,则在将希望混合点击声的标题、和不希望混合的标题记录到 1 个 BD-ROM 上的情况下,无需将具有 5.1Ch 的环绕声的 AVClip、和具有单声道的 AVClip 分开地记录到 BD-ROM 上。

[0332] 如果只将 1 个复用了 5.1Ch 的环绕声和单声道的 AVClip 记录到 BD-ROM 上,记录允许再现 5.1Ch 的环绕声的音频流的 PlayItem 信息、和不允许再现 5.1Ch 的环绕声的音频流的 PlayItem 信息,则能够用 1 个 BD-ROM 来提供希望混合点击声的标题、和不希望混合的标题。

[0333] 由于不用增多 AVClip 的数目,就能够制作不希望混合点击声而希望充实音响方面的标题、和希望混合点击声并希望充实对话方面的标题这两者,并提供给用户,所以制作

工作室的工时减少,能够提高电影制作的效率。

[0334] <Movie Object、Java(注册商标)应用之间的关系>

[0335] 在HDMV模式下的Movie Object中未包含希望发出点击声的导航命令、在BD-J模式下的Java(注册商标)应用中未包含希望发出点击声的字节码的情况下,即,在应用明确不进行点击声再现的情况下,Mixing_On标志被设定为0(OFF)。相反,在存在希望发出点击声的导航命令或字节码的情况下,Mixing_On标志被设定为1(ON)。通过进行以上那样的设定,在同时执行不再现点击声的应用、和再现音频的PlayItem的情况下,能够使再现装置中的混合无效;而在同时执行再现点击声的应用、和再现音频的PlayItem的情况下,能够使再现装置中的混合有效。

[0336] 以上是对记录介质的说明。接着说明本发明的再现装置。

[0337] 图33是本发明的再现装置的内部结构的图。本发明的再现装置根据本图所示的内部结构,来工业生产。本发明的再现装置主要由系统LSI和驱动器装置这2个部分组成,可以通过将这些部分安装到装置的机箱及电路板上来工业生产。系统LSI是集成了起再现装置的功能的各种各样的处理部的集成电路。这样生产的再现装置由BD-ROM驱动器1、读缓冲器2、解复用器3、视频解码器4、视频平面5、声音处理器6、声音处理器7、混音器8、声音控制器9、D/A转换器10、Interactive Graphics解码器11、Interactive Graphics平面12、Presentation Graphics解码器13、PresentationGraphics平面14、JPEG解码器15、Still平面16、合成部17、STC-delta附加部18、ATC-delta附加部19、局部存储器20、指令ROM 21、用户事件处理部22、PSR组23、CPU 24、脚本存储器25、局部存储器26构成。

[0338] 首先,说明与再现BD-ROM上记录着的AVClip有关的构成要素(BD驱动器1~音频解码器6)。

[0339] BD-ROM驱动器1装入/弹出BD-ROM,执行对BD-ROM的访问。

[0340] 读缓冲器2是FIFO存储器,先入先出式地保存从BD-ROM上读出的TS包。

[0341] 解复用器(De-MUX)3从读缓冲器2中取出Source包,并将构成该Source包的TS包变换为PES包。然后将通过变换得到的PES包中的、具有STN_Table中记载的PID的PES包输出到视频解码器4、音频解码器6、Interactive Graphics解码器11、Presentation Graphics解码器13中的某一个。

[0342] 视频解码器4对从解复用器3输出的多个PES包进行解码而得到非压缩形式的图片并写入到视频平面5上。

[0343] 视频平面5是用于保存非压缩形式的图片的平面。所谓平面,是再现装置中用于保存一个画面的像素数据的存储区域。视频平面5的分辨率是1920×1080,该视频平面5上保存图片数据由用16位的YUV值表现的像素数据构成。在视频平面5上,能够缩放(scaling)视频流中的每一帧的再现影像。所谓缩放,是将每一帧的再现图像变化到整个视频平面5的1/4(称为quarter)、1/1(称为full scale)中的某一个。由于在BD-J模式下根据来自CPU 24的指示来执行这种缩放,所以能够进行将视频流的再现图像安排到画面的角落、或者全屏显示这样的画面显示。

[0344] 声音处理器6包含:解码缓冲器(DB)6a,在PID过滤器3输出构成音频流的PES包时,保存该PES包;和音频解码器6b,对该缓冲器中保存着的PES包进行解码,并输出PCM状态的音频数据。

[0345] 声音处理器 7 包含：预加载存储器 7a, 预加载从 BD-ROM 上读出的文件 sound.bdmv ;和音频解码器 7b, 对预加载的文件 sound.bdmv 中的多个声音数据中的、由 CPU 24 指示的声音数据进行解码, 并输出 PCM 状态的音频数据。向预加载缓冲器 7a 的预加载最好在装入 BD-ROM 时或切换标题时进行。这是因为, 如果想要在 AVClip 的再现中读出文件 sound.bdmv, 则发生用于读出与 AVClip 不同的文件的光拾取器的寻道 (seek)。另一方面, 在装入 BD-ROM 时或切换标题时, AVClip 的再现很少继续, 所以通过在这样的定时读出文件 sound.bdmv, 能够保证 AVClip 再现不中断。

[0346] 混音器 8 混合从声音处理器 6 及声音处理器 7 输出的 PCM 状态的声音数据。在混合时, 混音器 8 为了使采样频率和声道数一致, 进行将从声音处理器 6 输出的声音的属性变换为从声音处理器 7 输出的音频属性这一处理。该混音器 8 进行的混音通过用 CPU 24 解释希望发出点击声的导航命令、或希望发出点击声的字节码来进行。

[0347] 声音控制器 9 切换输出从混音器 8 输出的展开状态的音频数据、及未经声音处理器 6 的压缩状态的音频数据中的哪一个。该声音输出经 S/PDIF 或 HDMI 输出到电视机 400 或放大器 500。在输出压缩状态的音频数据的情况下, 在其输出目的地的电视 400 及放大器 500 中被解码。

[0348] D/A 转换部 10 对从混音器 8 输出的数字的音频数据进行 D/A 转换, 并输出模拟音频。

[0349] I-Graphics 解码器 (IG 解码器) 11 对从 BD-ROM 或局部存储器 20 上读出的 IG 流进行解码, 并将非压缩图形写入到 Interactive Graphics 平面 12 上。

[0350] Interactive Graphics (IG) 平面 12 在 HDMV 模式下被写入由 I-Graphics 解码器 10 解码得到的非压缩图形。而在 BD-J 模式下, 被写入由应用绘制的字符或图形。

[0351] P-Graphics 解码器 13 对从 BD-ROM 或局部存储器 20 上读出的 PG 流进行解码, 并将非压缩图形写入到 Presentation Graphics 平面 14 上。通过 P-Graphics 解码器 13 进行的解码, 字幕出现在画面上。

[0352] Presentation Graphics 平面 14 是具有一个画面的区域的存储器, 能够保存一个画面的非压缩图形。

[0353] JPEG 解码器 15 对 BD-ROM 或局部存储器 20 上记录着的 JPEG 数据进行解码, 并写入到 Still 平面 16 上。

[0354] Still 平面 16 是保存通过展开 JPEG 数据而得到的非压缩的图形数据的平面。该图形数据被用作 Java (注册商标) 应用绘制的 GUI 框架的所谓的“壁纸”。

[0355] 合成部 17 得到合成了 Interactive Graphics 平面 12 的保存内容、Presentation Graphics 平面 14 的保存内容、视频平面 5 的保存内容、以及 Still 平面 16 的保存内容的合成图像。

[0356] STC_delta 附加部 18 生成 System Time Clock (STC)。在切换 STC_Sequence 时, 通过将以前的 STC_Sequence 中的 STC 值 (STC1) 加上称为 STC_delta 的偏移值来求新的 STC_Sequence 的 STC 值 (STC2), 并将以前的 STC_Sequence 中的 STC 值 (STC1)、和新的 STC_Sequence 的 STC 值 (STC2) 变为连续的值。

[0357] 在之前的 STC_Sequence 中最后再现的圖片的显示开始时刻为 PTS1 (1stEnd)、圖片的显示期间为 Tpp、后续 STC_Sequence 中最先显示的圖片的开始时刻为 PTS2 (2ndSTART)

的情况下,STC_delta 可表现为

[0358] $STC_delta = PTS1(1stEND) + Tpp - PTS2(2ndSTART)$ 。

[0359] 如上所述求 STC_delta,将加上了它的时钟的计数值输出到各解码器。由此,各解码器能够不中断地再现与 2 个 STC_Sequence 相当的流。通过以上,即使在 1 个 AVClip 中存在 2 个以上的 STC_Sequence,并且即使应连续再现的 2 个以上的 AVClip 分别具有不同的 STC_Sequence,也能够无缝地执行这些 STC_Sequence 间的解码处理。

[0360] 此外,为了满足缓冲的连续性,满足以下的 1)、2) 即可。

[0361] 1) 满足 $STC2(2ndSTART) > STC2(1stEND)$,

[0362] 这里,STC2(1stEND) 是将 STC1(1stEND) 投影到 STC2 的时间轴上所得的值,用 $STC2(1stEND) = STC1(1stEND) - STC_delta$ 这一计算式来给出。

[0363] 2) 从 TS1 中取出 TS 包、和从 TS2 中取出 TS 包由投影到同一时间轴上的 STC1 和 STC2 来定义,应该不引起缓冲器的下溢或上溢。

[0364] ATC_delta 附加部 19 生成 Arrival Time Clock(到达时钟)(ATC)。在切换 STC_Sequence 时,通过将以前的 ATC_Sequence 中的 ATC 值(ATC1)加上称为 ATC_delta 的偏移值,而将以前的 ATC_Sequence 中的 ATC 值(ATC1)、和新的 ATC_Sequence 的 ATC 值(ATC2)变为连续的值。通过该加法, $ATC2 = ATC1 + ATC_delta$ 。所谓 ATC_delta,是指从以前读出的传输流(TS1)的最后 TS 包的输入时刻 T1、到新读出的传输流(TS2)的最初 TS 包的输入时刻 T2 的偏移值,用“ $ATC_delta \geq N1 / TS_recording_rate$ ”这一计算式来给出。这里,输入时刻 T2 表示将 TS2 的最初 TS 包的输入时刻投影到 TS1 的时间轴上所得的时刻。此外,N1 是 TS1 的最后的视频 PES 包的后续的 TS 包的包数。在 BD-ROM 上,这种 ATC_delta 在 Clip 信息中被描述,所以通过采用它,能够计算 ATC_delta。通过以上的计算,能够将以前的 ATC_Sequence 具有的 ATC 值(ATC1)、和新的 ATC_Sequence 具有的 ATC 值(ATC2)变为连续的值。通过将加上了 ATC_delta 的时钟的计数值输出到解复用器(De-MUX)3,能够实现无缝的缓冲控制。

[0365] 以上是关于 AVClip 再现的构成要素。接着说明关于 BD-J 模式下的动作的构成要素(局部储存器 20 ~ 局部存储器 26)。

[0366] 局部储存器 20 是用于与元数据一起保存从 web 站点下载的内容等从 BD-ROM 以外的记录介质、从通信媒体提供的内容的硬盘。该元数据是用于将下载内容绑定在局部储存器 20 上来管理的信息,通过访问该局部储存器 20,BD-J 模式下的应用能够进行利用了下载内容长度的各种各样的处理。

[0367] 接着,说明再现装置中的实现综合控制的构成要素(指令 ROM 21 ~ 局部存储器 26)。

[0368] 指令 ROM 21 存储着规定再现装置的控制的软件。

[0369] 用户事件处理部 22 按照对遥控器或再现装置的前面板的键操作,将进行该操作的用户事件输出到 CPU 24。

[0370] PSR 组 23 是再现装置中内置的寄存器,由 64 个 Player Status Register(播放器状态寄存器)(PSR)、和 4096 个 General Purpose Register(通用寄存器)(GPR)组成。Player Status Register 的设定值(PSR)中的 PSR4 ~ PSR8 被用来表现当前的再现时刻。

[0371] PSR4 通过被设定为 1 ~ 100 的值,来表示当前的再现时刻所属的标题;通过被设

定为 0,来表示当前的再现时刻是顶级菜单。

[0372] PSR5 通过被设定为 1 ~ 999 的值,来表示当前的再现时刻所属的章节号;通过被设定为 0xFFFF,来表示在再现装置中章节号无效。

[0373] PSR6 通过被设定为 0 ~ 999 的值,来表示当前的再现时刻所属的播放列表(当前 PL)的号码。

[0374] PSR7 通过被设定为 0 ~ 255 的值,来表示当前的再现时刻所属的 PlayItem(当前 Play Item)的号码。

[0375] PSR8 通过被设定为 0 ~ 0xFFFFFFFF 的值,来用 45KHz 的时间精度表示当前的再现时刻(当前 PTM(Presentation Time))。用以上的 PSR4 ~ PSR8,能够在图 18(a) 中的整个 BD-ROM 的时间轴上确定当前的再现时刻在哪里。

[0376] CPU 24 执行指令 ROM 21 中保存着的软件,来执行整个再现装置的控制。该控制的内容按照从用户事件处理部 22 输出的用户事件、及 PSR 组 23 中的各 PSR 的设定值来动态变化。

[0377] 脚本存储器 25 是用于保存当前的 PL 信息或当前的 Clip 信息的存储器。所谓当前 PL 信息,是指 BD-ROM 上记录着的多个播放列表信息中的作为当前处理对象的播放列表信息。所谓当前 Clip 信息,是指 BD-ROM 上记录着的多个 Clip 信息中的作为当前处理对象的 Clip 信息。

[0378] 由于从 BD-ROM 的读出是低速的,所以局部存储器 26 是用于暂时保存 BD-ROM 的记录内容的高速缓冲存储器。由于存在这种局部存储器 26,所以 BD-J 模式下的应用执行的效率提高。

[0379] 以上是本实施方式的再现装置的硬件结构。接着说明本实施方式的再现装置的软件结构。

[0380] 图 34 是将由 ROM 24 中保存的软件和硬件组成的部分置换为层结构而绘制的图。如本图所示,再现装置的层结构由以下的 a)、b)、c) 组成。即,由

[0381] a)BD Player Device(BD 播放器设备)的第 1 层次、

[0382] b)BD Player Model(BD 播放器模型)的第 2 层次、

[0383] c)Application Runtime Environment(应用运行期环境)的第 3 层次组成。

[0384] 这些层次中的图 34 所示的再现装置的硬件结构属于第 1 层次。在本图的第 1 层次“BD Player Device”中,包含与图 34 所示的硬件结构中的视频解码器 4、音频解码器 6、IG 解码器 11、PG 解码器 13 相当的“解码器”、和与视频平面 5、IG 平面 12、PG 平面 14 相当的“平面”、BD-ROM 及其文件系统、局部存储器 20 及其文件系统。

[0385] 第 2 层次“BD Player Model”由以下的 b1)、b2) 的层组成。即,

[0386] b2)Playback Control Engine 32 的层、

[0387] b1)Virtual File System(虚拟文件系统)30 及 Presentation Engine 31 的层组成,向自身的上级的层次提供功能 API。

[0388] 第 3 层次“Application Runtime Environment”由以下的 c1)、c2) 组成。即,由

[0389] c1) 存在模块管理器 34 的层、

[0390] c2) 存在 BD-J 平台 35 的层组成。

[0391] 首先,说明属于第 2 层的 Virtual File System 30 ~ 模块管理器 34。

[0392] Virtual File System 30 是用于与 BD-ROM 上的盘内容一体地处理局部储存器 20 中保存着的下载内容的虚拟的文件系统。这里,局部储存器 20 中保存着的下载内容包含 SubClip、Clip 信息、播放列表信息。该下载内容中的播放列表信息不管是 BD-ROM 及局部储存器 20 中的哪一个中存在的 Clip 信息,都能够指定,在这一点上与 BD-ROM 上的播放列表信息不同。在该指定时,Virtual File System 30 上的播放列表信息无需用完整路径(フルパス)来指定 BD-ROM 或局部储存器 20 上的文件。这是因为,BD-ROM 上的文件系统或局部储存器 20 上的文件系统被作为虚拟的 1 个文件系统(Virtual File System 30)来识别。因此,PlayItem 信息中的 Clip_Information_file_name 通过指定与保存着 Clip 信息的文件的文件主体相当的 5 位数值,能够指定 Virtual File System 30、BD-ROM 上的 AVClip。通过经 Virtual File System 30 读出局部储存器 20 的记录内容,与 BD-ROM 的记录内容动态组合,能够产生各种各样的再现的变化。将局部储存器 20 和 BD-ROM 组合而成的盘内容,被与 BD-ROM 上的盘内容对等看待,所以本申请中的“BD-ROM”也包含由局部储存器 20+BD-ROM 的组合组成的虚拟的记录介质。

[0393] Presentation Engine 31 执行 AV 再现功能。所谓再现装置的 AV 再现功能,是从 DVD 播放器、CD 播放器沿袭的传统的功能组,是开始再现(Play)、停止再现(Stop)、暂停(Pause On)、解除暂停(Pause Off)、解除 Still(静止)功能(still off)、带速度指定的快进(Forward Play(speed))、带速度指定的倒退(Backward Play(speed))、切换声音(Audio Change)、切换副影像(SubtitleChange)、切换角度(Angle Change)等功能。为了实现 AV 再现功能,Presentation Engine 31 控制视频解码器 4、P-Graphics 解码器 13、I-Graphics 解码器 10、音频解码器 6,以便对读出到读缓冲器 2 上的 AVClip 中的、与期望的时刻相当的部分进行解码。作为期望的时刻,通过对 PSR8(当前 PTM)所示的部位进行解码,能够在 AVClip 中再现任意的时刻。

[0394] 再现控制引擎(Playback Control Engine(PCE))32 执行对播放列表的再现控制功能(i)、PSR 组 23 中的状态取得/设定功能(ii)等诸功能。所谓对播放列表的再现控制功能,是指根据当前 PL 信息及 Clip 信息来进行 Presentation Engine 31 进行的 AV 再现功能中的开始再现或停止再现。这些功能(i)~(ii)按照来自 HDMV 模块 33~BD-J 平台 35 的功能调用来执行。

[0395] 模块管理器 34 保持从 BD-ROM 上读出的 Index.bdmv,进行转移控制。该转移控制通过向构成当前标题的动态脚本发出 Terminate(终止)事件、向构成转移目标标题的动态脚本发出 Activate(激活)事件来进行。

[0396] 以上是对 Presentation Engine 31~模块管理器 34 的说明。接着说明 BD-J 平台 35。

[0397] BD-J 平台 35 是所谓的 Java(注册商标)平台,采用了以 Java(注册商标)虚拟机 36 为核心的结构。BD-J 平台 35 除了上述 Java(注册商标)2Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0)、和 GloballyExecutable MHP specification(GEM[1.0.2])for package media targets 以外,还安装了 BD-J Extention(扩展)。BD-J Extention 包含为了将超越了 GEM[1.0.2]的功能提供给 BD-J 平台而专门化了的各种各样的程序包(package)。

[0398] 首先,说明作为 BD-J 平台 35 的核心的 Java(注册商标)虚拟机 36。

[0399] <Java(注册商标)虚拟机 36>

[0400] 图 35 是 Java(注册商标)虚拟机 36 的内部结构的图。如本图所示,Java(注册商标)虚拟机 36 由图 33 所示的 CPU 24、用户类加载器 52、方法区域 (method area)53、工作存储器 54、线程 55a、b...n、Java(注册商标)栈 56a、b...n 构成。

[0401] 用户类加载器 52 从局部存储器 26 等中读出 BDKA 目录的 Java(注册商标)存档文件中的类文件并保存到方法区域 53 中。该用户类加载器 52 进行的类文件读出,通过用应用管理器 37 指示用户类加载器 52 进行指定了文件路径的读出来进行。如果文件路径表示局部存储器 26,则用户类加载器 52 将构成应用的 Java(注册商标)存档文件中的类文件从局部存储器 26 中读出到工作存储器 54 中。如果文件路径表示 Virtual File System 30 上的目录,则用户类加载器 52 将构成应用的 Java(注册商标)存档文件中的类文件从 BD-ROM 或局部存储器 20 中读出到工作存储器 54 中。应用的运行控制通过该用户类加载器 52 进行的类文件读出来实现。在指示读出的类文件不在局部存储器 26 中的情况下,用户类加载器 52 将读出失败通知给应用管理器 37。

[0402] 方法区域 53 保存用户类加载器 52 从局部存储器 26 中读出的类文件。

[0403] 工作存储器 54 是所谓的堆区域 (heap area),保存各种各样的类文件的实例 (instance)。图 34 所示的应用管理器 37 是驻留在该工作存储器 54 中的驻留应用。在工作存储器 54 中,除了保存这些驻留型的实例之外,还保存与读出到方法区域 53 中的类文件对应的实例。该实例是构成应用的 xlet 程序。通过将这种 xlet 程序布置到工作存储器 54 中,应用变为可执行的状态。

[0404] 在图 34 的层模型中,将该工作存储器 54 上的应用管理器 37 描绘在 Java(注册商标)虚拟机 36 上,但是这不过是照顾到易懂性。应用管理器 37 及应用作为实例由线程 55a、b...n 来执行,这是现实性的描述。

[0405] 线程 55a、b...n 是执行工作存储器 54 中保存着的方法的逻辑执行主体,将局部变量、或操作数栈中保存着的变量作为操作数来进行运算,将运算结果保存到局部变量或操作数栈中。图中的箭头 ky1、ky2、kyn 象征性地表示从工作存储器 54 向线程 55a、b...n 提供方法。物理上,执行主体 CPU 只有 1 个,而作为逻辑执行主体的线程在 Java(注册商标)虚拟机 36 内最大可存在 64 个。在该 64 个这一数值内,可以新创建线程,也可以删除已有的线程,线程的动作数在 Java(注册商标)虚拟机 36 的动作中可增减。线程的数目可以适当增加,所以也可以用多线程来并行执行 1 个实例,实现实例的高速化。在本图中,CPU 24 和线程之间的对应关系设为 1 对多的关系,但是在 CPU 有多个的情况下,CPU 和线程之间的对应关系可为多对多的关系。用线程 55a、b...n 来执行方法通过将构成方法的字节码变换为 CPU 24 的本机代码后发出到 CPU 24 来进行。对于该本机代码变换,脱离了本申请的重点,所以省略其说明。

[0406] Java(注册商标)栈 56a、b...n 与线程 55a、b...n 以 1 对 1 的比率存在,在内部具有程序计数器 (图中的 PC) 和 1 个以上的帧。“程序计数器”表示在实例中当前正在执行哪个部分。“帧”是分配给对方法的 1 次调用的栈式的区域,由保存该 1 次调用时的变量的“操作数栈”、和被调用的方法使用的“局部变量栈 (图中的局部变量)”组成。每进行 1 次调用,帧就被堆积到 Java(注册商标)栈 56a、b...n 上,所以在某个方法递归地调用自身的情况下,也堆积 1 个该帧。

[0407] 以上是对 Java(注册商标)虚拟机的说明。

[0408] <应用管理器 37>

[0409] 应用管理器 37 是在 Java(注册商标)虚拟机 36 内的工作存储器上动作的系统软件,每当发生标题转移时,指示 Java(注册商标)虚拟机 36 运行在转移前标题中未执行、但在新的标题中具有 AutoRun 的运行属性的应用。与此同时,结束在转移前标题中执行了、但是不将新的标题作为生命周期的应用。这些运行控制及结束控制在参考了当前 BD-J Object 中的应用管理表后进行。

[0410] 图 36 是应用管理器 37 根据 BD-J Object 中的应用管理表进行的处理的示意图。

[0411] 图 36 中的☆ 1、☆ 2、☆ 3 示意性地表示参考应用管理表(☆ 1)、指示 Java(注册商标)虚拟机 36 运行应用(☆ 2)、指示 Java(注册商标)虚拟机 36 读出 Java(注册商标)存档文件(☆ 3)、加载定义 Java(注册商标)应用的类文件的类(☆ 4、5、6)这一系列过程。Java(注册商标)虚拟机 36 根据该运行指示从局部存储器 26 中将 xlet 程序读出到工作存储器中。

[0412] 图 37 是应用管理器 37 根据 BD-J Object 中的 PLMT 进行的处理的图。▽1 表示参考 BD-J Object 中的 PLMT,▽2 表示指示 Presentation Engine 31 读出播放列表信息。

[0413] 图 37 中的◎ 1、2、3、4 示意性地表示经由 Virtual File System 30 来读出播放列表信息(◎ 1)、解释构成播放列表信息的 PlayItem 信息(◎ 2)、经由 Virtual File System 30 来读出 Clip 信息(◎ 3)、解释 Clip 信息(◎ 4)。如果经过以上的过程解释出 Clip 信息、播放列表信息,则将构成 AVClip 的 TS 包通过 Virtual File System 30 转交给 Presentation Engine 31。如果这样依次将 TS 包交给 Presentation Engine 31,则 Presentation Engine 31 将构成 AVClip 的 TS 包输出到解码器,显示到平面上。图中的☆ 1、2、3、4 示意性地表示读出构成 AVClip 的 TS 包(☆ 1、2)、从 Virtual File System 30 向 Presentation Engine 31 转交 TS 包(☆ 3)、将 TS 包投入到解码器中(☆ 4)、从解码器向各种平面输出解码结果(☆ 5)。

[0414] 图 38 是应用管理器 37 根据 BD-J Object 中的声音管理表进行的处理的图。图 38 中的◎ 0、1、2 示意性地表示应用管理器 37 参考 BD-J Object 内的声音管理表(◎ 0)、指示使每个应用生命周期的混音有效、无效(◎ 1)、指示输出与操作了的按钮对应的声音数据或指示对其进行解码(◎ 2)。

[0415] 以下,说明用软件来实现应用管理器 37。图 39 是应用管理器 37 进行的处理过程的流程图。本图中的处理过程具有由步骤 S1-步骤 S2-步骤 S3-步骤 S4 组成的主循环。步骤 S1 判断是否进行了 TitleJump,如果是,则切换 Title(步骤 S7)。

[0416] 步骤 S8 判断在与当前标题对应的 BD-J Object 中是否存在 PLMT。如果不存在,则在前面的标题中,停止再现 PLMT 中记载着的播放列表(步骤 S9)。

[0417] 如果存在,则开始再现在前面标题中未记载在 PLMT 中、但是在当前标题中记载在 PLMT 中、被赋予了 AutoPlay 属性的播放列表(步骤 S10)。

[0418] 在步骤 S11 中,判断在与当前 Title 对应的 BD-J Object 中是否存在声音管理表。步骤 S12 是在存在声音管理表的情况下执行的判断步骤,判断该声音管理表中的 Mixing_On 标志是否是 1。如果 Mixing_On 标志是 1,则指示声音控制器 9 使混音有效(步骤 S13);如果 Mixing_On 标志是 0,则指示声音控制器 9 使混音无效(步骤 S14)。

[0419] 步骤 S15 判断在与当前 Title 对应的 BD-J Object 中是否存在应用管理表,如果存

在,则运行未将前面的 Title 作为生命周期、但是将当前 Title 作为生命周期、具有 AutoRun 属性的 Java(注册商标)应用(步骤 S16)。如果不存在,则停止将前面的 Title 作为生命周期、但是未将当前 Title 作为生命周期的应用(步骤 S17)。

[0420] 其后,判断 Java(注册商标)应用的运行是否成功(步骤 S18),如果运行成功(步骤 S18 中“是”),则实现将 AutoPlayPL 的再现图像变换为四分之一(1/4)(步骤 S19)这一过程。

[0421] 而如果该步骤 S18 是“否”,则执行由步骤 S23、S24、S16、S18 组成的循环处理。本循环处理中的控制变量是重新运行计数器。重新运行计数器是规定应用的重新运行次数的计数器。重新运行计数器在运行本流程图时被复位,在步骤 S23 中,判断是否是 0。在不是 0 的情况下,在步骤 S24 中重新运行计数器被递减。通过由以上的步骤 S23、S24、S16、S18 ~ 步骤 S19 组成的循环处理,只要重新运行计数器不是 0,就重复运行 AutoRun 应用。通过这种重复,保证了应用的运行。

[0422] 步骤 S2 判断是否是未执行主应用的状态,如果是,则进行步骤 S5 的判断。步骤 S5 判断应用是否正常结束。如果异常结束,则执行步骤 S21、步骤 S22 的处理。如果正常结束,则不执行步骤 S21 ~ 步骤 S22,而返回到由步骤 S1 ~ 步骤 S4 组成的主循环。

[0423] 步骤 S21 判断是否是 AutoPlayPL 的再现中,如果是再现中,则指示 Playback Control Engine 32 将 AutoPlayPL 的再现图像变为全屏(步骤 S22)。其后,转移到步骤 S23。通过转移到步骤 S23,即使在异常结束时也执行由步骤 S14 ~ 步骤 S17 组成的循环处理。由此重复应用的重新运行,直至重新运行计数器的次数变为 0。

[0424] 步骤 S4 判断在 BD 驱动器 1 中是否存在 BD-ROM,如果不存在 BD-ROM,则对所有应用发出结束指示(步骤 S6)。

[0425] 如上所述,根据本实施方式,能够在执行希望输出多声道的声音的再现控制时使混音无效,而在执行不希望输出多声道的声音的再现控制时使混音有效。

[0426] 由此,电影制作工作室能够进行下述调整:在希望多声道的再现的情况下,禁止点击声;而在希望点击声的再现的情况下,禁止多声道的声音输出。

[0427] 通过导入点击声,能够将制作工作室从“多声道的声音输出该不会中断吧”这一不安的束缚中解放出来,鼓励在制作电影作品时导入点击声。

[0428] 其中,在第 1 实施方式中,说明了在多声道是 5.1CH 的环绕声、音频流具有该多声道的属性的情况下,将 Mixing_On 标志设为 OFF,但是在多声道表示 2CH 的立体声、音频流具有立体声属性的情况下,也可以将 Mixing_On 标志设为 OFF。

[0429] 展开立体声后的数据量并不太大,在展开为 LPCM 状态后,即使进行混音,有时也能够进行数字输出。这样,在展开立体声后的数据量可望不太大、可确认能够进行数字输出的情况下,也可以将多声道设定为 ON。

[0430] (第 2 实施方式)

[0431] 在第 1 实施方式中,为自动再现了播放列表的应用的每个生命周期设有 Mixing_On 标志,但是从另一个角度来看,播放列表和应用生命周期是等价的,可以考虑为构成播放列表的每个再现区间进行基于 Mixing_On 标志的控制。

[0432] 鉴于这些,在本实施方式中,提出将 Mixing_On 标志设在播放列表信息中的技术。

[0433] 图 40(a) 是第 2 实施方式的播放列表信息的内部结构的图。本图与图 8(a) 所示的

播放列表信息的内部结构的不同点是,在 PlayItem 信息内设有 Mixing_On 标志。图 40(b) 是 PlayItem 信息内所设的 Mixing_On 标志的内容的图。

[0434] 在 Mixing_On 标志被设定为 1 的情况下,表示将基于文件 sound.bdmv 的点击声混合到 PlayItem 的再现声音中。

[0435] 在 Mixing_On 标志被设定为 0 的情况下,表示不将基于文件 sound.bdmv 的点击声混合到 PlayItem 的再现声音中。

[0436] 可以以播放列表为单位来规定是否混合基于文件 sound.bdmv 的点击声。将该 Mixing_On 标志设定为何种值,与该 PlayItem 信息具有的 STN_Table 的设定内容对应。

[0437] 以上是第 2 实施方式的记录介质的改进。接着说明第 2 实施方式的再现装置的改进。第 2 实施方式的再现装置的改进点存在于 Playback Control Engine 32 中。

[0438] 以下参考图 41 的流程图,来说明 Playback Control Engine 32 进行的具体的控制过程。

[0439] 图 41 是 Playback Control Engine 32 进行的播放列表再现过程的流程图。该再现过程主要包含对 Presentation Engine 31 的控制(步骤 S106)、和对 BD-ROM 驱动器 1 或局部储存器 20 的控制(步骤 S108)。在本流程图中,将作为处理对象的 PlayItem 设为 PlayItem#x。本流程图,读入当前 PL 信息(.mpls)(步骤 S101),其后,执行步骤 S102 ~ 步骤 S110 的处理。这里,步骤 S102 ~ 步骤 S110 构成下述循环处理:对构成当前 PL 信息的各个 PI 信息,重复步骤 S103 ~ 步骤 S110 的处理,直至步骤 S109 为“是”。在该循环处理中,将作为处理对象的 PlayItem 称为 PlayItem#x(PI#x)。该 PlayItem#x 通过被设定为当前播放列表的第一个 PlayItem,而被初始化(步骤 S102)。结束上述循环处理的必要条件是,该 PlayItem#x 为当前播放列表的最后一个 PlayItem(步骤 S109),如果不是最后一个 PlayItem,则当前播放列表中的下一个 PlayItem 被设定为 PlayItem#x(步骤 S110)。

[0440] 在循环处理中重复执行的步骤 S103 ~ 步骤 S110 将用 PlayItem#x 的 Clip_information_file_name 指定的 Clip 信息读入到脚本存储器 25 中(步骤 S103),用当前 Clip 信息的 EPmap 将 PlayItem#x 的 In_time 变换为 I 图片地址 u(步骤 S104),用当前 Clip 信息的 EP_map 将 PlayItem#x 的 Out_time 变换为 I 图片地址 v(步骤 S105),来求通过这些变换得到的地址 v 的下一个 I 图片,并将该地址的上一个地址设定为地址 w(步骤 S107),命令 BD-ROM 驱动器 1 或局部储存器 20 用这样算出的地址 w 来读出从 I 图片地址 u 到地址 w 的 TS 包(步骤 S108)。

[0441] 另一方面,命令 Presentation Engine 31 进行当前 PLMark 的 mark_time_stamp 至 PlayItem#x 的 Out_time 的输出(步骤 S106)。通过以上的步骤 S105 ~ 步骤 S108,在 AVClip 中,再现了用 PlayItem#x 指示的部分。

[0442] 其后,判断 PlayItem#x 是否是当前播放列表的最后一个 PI(步骤 S109)。

[0443] 如果 PlayItem#x 不是当前播放列表的最后一个 PI,则将当前播放列表中的下一个 PlayItem 设定为 PlayItem#x(步骤 S110),返回到步骤 S103。通过重复以上的步骤 S103 ~ 步骤 S110,依次再现了构成播放列表的 PI。

[0444] 在本流程图中,步骤 S115 ~ 步骤 S117 是每当步骤 S103 ~ 步骤 S110 转一圈时执行的步骤。步骤 S115 判断 PlayItem#x 中的 Mixing_On 标志是否是 1。如果 Mixing_On 标志是 1,则指示声音控制器 9 使混音有效(步骤 S116);如果 Mixing_On 标志是 0,则指示声

音控制器 9 使混音无效（步骤 S117）。

[0445] 如上所述,根据本实施方式,不管在 HDMV 模式下,还是在 BD-J 模式下,都在利用的播放列表信息中设 Mixing_On 标志,所以即使在 HDMV 模式下用 IG 流来实现 GUI 的情况下,也能够规定点击声的混音的有效、无效。

[0446] 此外,不限于点击声,在实现与从 WWW 站点下载的其他音频流混音的情况下,也能够将该混音的有效 / 无效规定在 Mixing_On 标志中。

[0447] （第 3 实施方式）

[0448] 在第 1 实施方式中,Java(注册商标)虚拟机中的播放列表再现能够用 BD-J Object 内的播放列表管理表来规定。这里成问题的是播放列表管理表。即,是否可以再现播放列表,在每个 BD-J Object 的播放列表管理表中有描述,所以会在某个 Title 中能够再现、而在别的 Title 中不能再现。此外,即使能够再现播放列表,但是从著作权保护的观念来看,有时想禁止从某种应用再现。为了实现这种播放列表再现的限制,在第 3 实施方式中,在 Java(注册商标)平台 35 中设有权限控制器。

[0449] 在某个应用请求再现播放列表的情况下,权限控制器与该应用进行相互认证,判断发出请求的应用是否有再现播放列表的权限。如果有,则请求 Playback Control Engine 32 进行该再现;如果没有,则将表示不允许的响应事件输出到发出请求的应用。根据该权限控制器的是否允许判断,即使与别的发行公司的发行有关的应用请求了与某个发行公司的发行有关的播放列表,也可以不允许这种请求。因此,能够避免没有正当权限的应用擅自引用播放列表。应允许的播放列表和应用的组合、不应允许的播放列表和应用的组合在另外记录在 BD-ROM 中的 Permission(权限)文件中有规定,权限控制器的判断基于此。这种文件的详细情况不是本申请的重点,所以省略其说明。

[0450] 在第 3 实施方式中,应用管理器 37 按照来自应用请求来通知在当前的再现时刻可再现的播放列表。图 42 是该应用管理器 37 进行的通知处理的处理过程的流程图。本流程图中,在应用的运行中,监视应用是否发出了请求通知可再现的播放列表的请求(GetPL)(步骤 S45)。如果发出了,则判断在构成当前的再现时刻所属的 Title 的 BD-J Object 中是否存在播放列表管理表(步骤 S46)。如果有播放列表的描述,则将播放列表管理表中描述的播放列表作为可再现的播放列表通知给发出请求的应用(步骤 S47)。

[0451] 如果没有播放列表的描述,则将不能再现播放列表的意思通知给发出请求的应用(步骤 S48)。以上是第 3 实施方式的应用管理器 37 的处理过程。

[0452] 接着说明在被请求再现播放列表的情况下的应用管理器 37 的处理。第 3 实施方式的应用管理器 37 根据图 43 的流程图来进行处理。

[0453] 在图 43 中,应用管理器 37 判断是否存在请求了再现播放列表的应用(步骤 S51)。如果某个应用请求再现播放列表,则使权限控制器认证发出请求的应用是否有再现播放列表的权利(步骤 S52)。如果有再现的权利,则指示 Playback Control Engine 32 开始再现(步骤 S53),等待来自 PlaybackControl Engine 32 的访问响应(步骤 S54)。

[0454] 如果有这种再现请求,则 Playback Control Engine 32 检查播放列表信息的正当性。在这种正当性检查中,有检查在播放列表信息、Clip 信息、AVClip 所在的 BD-ROM 及局部储存器 20 中是否构成了正当的播放列表,或用播放列表信息中的 clip_Information_file_name 指定的 Clip 信息及 AVClip 是否现存于 BD-ROM 及局部储存器 20 中。在正确的文

件未由 clip_Information_file_name 参照的情况下,或者在由 BD-ROM 及局部储存器 20 构成的虚拟封装有矛盾、不能构成正确的播放列表的情况下,PlaybackControl Engine 32 返回表示 false(假)的响应。此外,在具有比发出请求的应用更高的运行优先级的应用正在再现该播放列表、在实现播放列表再现的资源中发生了竞争的情况下,Playback Control Engine 32 也返回表示 false 的响应。

[0455] 经过以上的过程,如果有访问响应,则将表示播放列表再现成功的事件输出到发出请求的应用(步骤 S55)。

[0456] 如果没有访问响应,则将表示播放列表再现失败的事件输出到发出请求的应用(步骤 S56)。另一方面,在步骤 S52 中如果发出请求的应用没有再现的权利,则将表示不能再现播放列表的事件输出到发出请求的应用(步骤 S57)。

[0457] 如上所述,根据本实施方式,即使可否再现播放列表因各 Title 而异,即使有具有再现播放列表的权限的应用和没有再现播放列表的权限的应用等各种应用,也能够按照来自应用的请求来执行适当的播放列表再现。因此,能够表现组合了应用执行和播放列表再现的多彩的内容。

[0458] (第 4 实施方式)

[0459] 在第 1 实施方式中,在开始 Title 时,向想开始再现的播放列表赋予表示 AutoPlay 的再现属性,并命令再现装置再现 AutoPlayPL。而在本实施方式中,涉及下述改进:将无边界应用记录到 BD-ROM 上,在开始 Title 时,使无边界应用选择应自动开始再现的 Title。

[0460] 无边界应用是 Playback Control Engine 32 那样的、处于与再现装置中的驻留应用对等的地位的应用,按照来自 Playback Control Engine 32 的请求来执行从播放列表管理表中描述的多个播放列表信息中选择与再现装置侧的 PSR 设定值一致者、并进行通知的处理。

[0461] 在使无边界应用选择播放列表的情况下,在需要这种选择的 Title 时,将播放列表管理表中的再现属性都设定为未指定。这是因为,以“都未指定”为触发,使标题无边界应用请求 Playback Control Engine 32 选择播放列表。

[0462] 该无边界应用进行的选择基于创作时规定的选择算法。图 44(a) ~ (c) 是将无边界应用中包含的选择算法的内容做成表形式的图。该表相对应地示出了 PSR 的值可取的值的范围、和在 PSR 值为这些值时应再现的播放列表。其中,图 44(a) 示出基于父母锁级别的选择算法的内容。这里,父母锁级别在再现装置中由 PSR(14) 来表示。具体地说,在 PSR(14) 中,设定了表示用户的年龄的整数,再现装置将其解释为父母锁级别。在图 44(a) 中,PSR(14) 可取的值被分为 14 岁不到、14 岁以上 18 岁不到、18 岁以上这 3 个范围。应再现的播放列表与这些范围中的每一个相对应。如果无边界应用根据这种选择算法来进行选择,则如果 PSR 的设定值为 14 岁不到就选择 PlayList#1,如果为 14 以上 18 岁不到就选择 PlayList#2,如果为 18 岁以上就选择 PlayList#3。

[0463] 图 44(b) 示出基于 Language for Audio 的选择算法的内容。这里,Language for Audio 在再现装置中由 PSR(16) 来表示。具体地说,在 PSR(16) 中,设定了整数,再现装置将其解释为声音再现用的语言设定。在图 44(b) 中,PSR(16) 可取的值被分为表示英语的值、表示日语的值、其他值这 3 个范围。应再现的播放列表与这些范围中的每一个相对应。如果无边界应用根据这种选择算法来进行选择,则如果 PSR(16) 的设定值表示英语

就选择 Playlist#1, 如果表示日语就选择 Playlist#2, 如果为英语、日语以外的值就选择 Playlist#3。

[0464] 图 44(c) 示出基于 Player Configuration for Video 的选择算法的内容。这里, Player Configuration for Video 在再现装置中由 PSR(14) 来表示。具体地说, 在 PSR(14) 中, 设定了整数值, 再现装置将其解释为影像再现用的环境设定。在图 44(c) 中, PSR(14) 可取的值被分为分辨率 525×600TVsystem(电视制式)LetterBox(信箱)、分辨率 525×600TVsystem、1920×1080TVsystem 这 3 个范围。应再现的播放列表与这些范围中的每一个相对应。如果无边界应用根据这种选择算法来进行选择, 则如果 PSR(14) 的设定值表示分辨率 525×600TVsystem LetterBox 就选择 Playlist#1, 如果表示分辨率 525×600 就选择 Playlist#2, 如果表示 1920×1080TVsystem 就选择 Playlist#3。图 44(a) ~ (c) 所示的选择算法可以通过用计算机描述语言描述这些图所示的条件转移来创建。

[0465] 以上是本实施方式的记录介质的改进。接着说明本实施方式的再现装置的改进。本实施方式中的改进点主要在于应用管理器 37、Playback Control Engine 32。

[0466] 应用管理器 37 在发生了 Title 的转移、参考了播放列表管理表时, 判断在该播放列表管理表中是否存在 AutoPlayPL。如果没有 AutoPlayPL, 则将播放列表管理表转交给 Playback Control Engine 32, 并请求 Playback Control Engine 32 自动再现该播放列表管理表中记载着的播放列表中的某个。

[0467] Playback Control Engine 32 如果接受播放列表管理表的转交, 则请求无边界应用选择播放列表。如果按照该请求从无边界应用通知了可再现的播放列表的列表, 则判断该 PL 列表中记载着的播放列表中的、存在于从 PlayItem 转交的播放列表管理表中的播放列表。如果在由无边界应用选择出的播放列表中有播放列表管理表中记载着的, 则自动开始再现它。

[0468] 图 45 是使标题无边界应用进行 PL 选择的过程的示意图。本图的左侧示出再现装置中的软件的层结构, 本图的右侧示出 BD-ROM 的记录内容。图中的 © 1、2、3、4 示意性地描绘了发现了没有 AutoPlay 的播放列表管理表的情况下的来自应用管理器 37 的通知 (© 1)、Playback Control Engine 32 查询可再现的播放列表 (© 2)、标题无边界应用取得 PSR 设定值 (© 3)、从标题无边界应用向 Playback Control Engine 32 通知可再现的播放列表 (© 4)。

[0469] 其中, 在图 45 中, 标题无边界应用在 BD-ROM 上有描述, 但是这不过是照顾到要将标题无边界应用描述得易懂。由于标题无边界应用是 Java(注册商标)应用, 所以在 Java(注册商标)虚拟机 36 内的工作存储器 54 中作为实例由线程 55 来执行, 这是现实性的描述。

[0470] 如上所述, 根据本实施方式, 使在 Title 的边界上生存着的应用进行上述那样的判断, 所以再现装置侧的 Playback Control Engine 32 能够在开始 Title 时的较早的阶段知道 BD-ROM 上的多个播放列表中的、与再现装置侧的状态设定相应的是哪个。即使未决定再现属性 = AutoPlay 的应用, 也能够开始 Title 时决定应开始再现的播放列表, 所以在 BD-J 模式下也能够实现语言字幕(ラングエーヅクレジツト)或父母锁(パレンタルロック)这样的再现控制。

[0471] 其中, 本实施方式中的选择算法是将 PSR 可取的值与播放列表相对应, 但是也可

以预先规定在再现装置中的 PSR 的设定值是设想外的值的情况下,使再现装置再现的播放列表。

[0472] (第 5 实施方式)

[0473] 本实施方式涉及在切换标题时预先使混音无效的改进。在切换标题时,会发生文件 sound.bdmv 的读出,所以在这种切换时,作为缺省,预先使混音无效。通过这样做,能够防止由于文件 sound.bdmv 不存在于预加载缓冲器 7b 内而造成的点击声再现的中断。

[0474] (第 6 实施方式)

[0475] 本实施方式是 xlet 程序通过利用 JMF (Java (注册商标)MediaFrameWork :Java 媒体框架)方式的接口,生成对于 mpls 文件的实例 (JMF 播放器实例),来命令再现装置再现播放列表的情况下的实施方式。

[0476] 所谓 mpls 文件,是规定播放列表信息的实体方面的文件,所以如第 2 实施方式所示,在该播放列表信息中设有 Mixing_On 标志。在将 Mixing_On 标志设在播放列表中的情况下,Java (注册商标)应用通过请求 PlaybackControl Engine 32 再现播放列表信息,能够按照指定的播放列表信息内的 Mixing_On 标志,使混音有效或无效。

[0477] (第 7 实施方式)

[0478] 在第 1 实施方式中,使应用管理器 37 具有用于指示声音控制器 9 使混音有效或无效等的功能,而本实施方式是使 Java (注册商标)应用具有这种功能的实施方式。

[0479] 因此,在本实施方式中,使 Java (注册商标)平台 35 具有用于从 Java (注册商标)应用接受这种功能的调用的 API (Application Interface)。这样,在 Java (注册商标)应用执行了请求使混音有效或无效的功能调用的情况下,指示声音控制器 9 使混音有效或无效等。

[0480] 其中,也可以使模块管理器 34 具有这种指示有效、无效的功能。

[0481] (第 8 实施方式)

[0482] 在再现装置中存在可以以 PCM 来传输环绕声的输出形态、该输出形态由用户选择了的情况下,无需对混合声音进行重新编码,所以有时可以原样地输出。在这种输出形态中,有 HDMI 的输出形态、模拟的输出形态。

[0483] 为了使再现装置判断在再现装置中是否存在可以以 PCM 来传输环绕声的输出形态、并且该输出形态是否已被选择,在本实施方式中,在 PSR 组 23 中描述了:用户希望何种声音输出形态,或者根据再现装置的 I/F 的连接形态能够进行何种声音输出。

[0484] 以下,说明本实施方式的 PSR 的内容。在本实施方式中,对 PSR 组 23 中的第 21 个 PSR (PSR21)、第 22 个 PSR (PSR22) 加以以下的改进。

[0485] PSR21 示出表示能否混音 (例如混合 2ch 的添加声音)的“Audio MixingCapability (混音能力)”、表示可混合的声道数 (或可混合的各个声道名)的“Audio Mixing Channel Number (混音声道数)”、表示当前用户选择了的音频输出形态的最大的输出 ch 数的“Audio Output Channel Number (音频输出声道数)”。

[0486] 例如,在选择了 S/PDIF 作为声音输出形态的情况下,环绕声由于频带上的问题,如果不压缩编码,就不能输出。因此,需要在混合后进行重新编码;而在选择了与高比特率对应的数字 I/F—HDMI 等的情况下,以非压缩 (LPCM) 的形式也能够原样地传送环绕声。因此,在 S/PDIF 连接时,Audio Output Channel Number 的值因在混合后是否进行编码而异。

[0487] 在 PSR22 中,作为再现装置的一个属性,有表示是不进行视频输出的再现装置的 Audio Only Player(只播放音频的播放器)、表示是符合规定的规范(profile)的再现装置的 Profile1 Player(规范 1 播放器)、Profile2 Player(规范 2 播放器)。例如,支持 Profile1、却不支持 Profile2 的再现装置将 Profile1Player 设定为有效,将 Profile2 Player 设定为无效。这些值可以从日语 L 参考,所以能够进行与再现装置的规范相应的动态选择再现路径选择,或选择有无菜单等。特别是在 Audio Only Player 有效的情况下,能够创建标题来再现播放列表,以使用日语 L 来提供与 CD 再现装置同等的用户操作性,在此情况下,用户能够与音乐 CD 毫无两样地操作 BD-ROM。当然,虽是同一盘,但是对 Audio Only Player 无效的再现装置,可以像以前那样假定影像输出来显示基于 GUI 的菜单等,进行交互式再现。这种设想在采用了 BD 的汽车音响系统等中很有效。

[0488] 由于添加了以上那样的 PSR,所以本实施方式的 Playback ControlEngine 32 在与 PlayItem 信息或播放列表信息对应的 Mixing_On 标志表示“混音=有效”的情况下,根据 PSR21 来判断能否混音。在 PSR21 为能够混音的情况下,从 PSR21 中取得可混合的 ch 数,将具有该 ch 数以下的 ch 数的音频流或声音数据选择性地输出到混音器 8、声音控制器 9,使其进行混音。

[0489] 如上所述,根据本实施方式,在 PSR 中描述了用户希望何种声音输出形态、或者根据再现装置的 I/F 的连接形态能够进行何种声音输出,所以能够更详细地规定可否混音输出。

[0490] (备考)

[0491] 以上的说明并未示出本发明的所有实施行为。通过实施了下述 (A) (B) (C) (D)……的变更的实施行为的方式,也能够实施本发明。本申请的权利要求的各发明是以上记载的多个实施方式及它们的变形方式的扩展了的记载、以及一般化了的记载。扩展以及一般化的程度基于本发明的技术领域的申请当时的技术水平的特性。

[0492] (A) 各实施方式的 BD-ROM 可以通过依次执行以下的工序来制作。

[0493] 首先,规划决定按何种计划来再现 BD-ROM(规划工序),进行动态图像收录、音频收录等素材创建(素材创建工序),根据规划工序中创建的计划,来创建卷结构信息(脚本创建工序)。

[0494] 所谓卷结构信息,是用抽象的描述来表示光盘的应用层的格式的信息。

[0495] 其后,通过对视频素材、音频素材、字幕素材、菜单素材分别进行编码,来得到基本流(素材编码工序)。其后,复用多个基本流(复用工序)。

[0496] 如果这样进行了复用,则进行使复用了的流及卷结构信息适合 BD-ROM 的应用层格式的业,得到应记录到 BD-ROM 的卷区域上的数据的总体表示(一般称为卷数据)(格式化工序)。

[0497] 这里,本发明的记录介质的应用层格式是用编程语言描述类结构的实例,通过根据 BD-ROM 标准、BD-J 标准中规定的语法,描述类结构的实例,由此能够创建 BD-J object、Clip 信息、PlayList 信息等。在此情况下,表形式的数据可以用编程语言的 for 语句来定义,其他只在特定的条件下需要的数据可以用 if 语句来定义。

[0498] 这种适应处理之后,如果得到了卷数据,则试着再现卷数据来确认脚本创建工序的结果是否正确(仿真工序)。在该仿真工序中,最好进行 BD-ROM 播放器模型的缓冲状态

的模拟。

[0499] 最后进行模压工序。

[0500] 在该模压工序中,将卷图像(ポリユーモイメーヅ)变换为物理数据串,并用该物理数据串来刻制母盘,创建光盘母盘。进而根据模压装置制成的母盘来制造 BD-ROM。该制造主要由基板成形、反射膜成膜、保护膜涂布、粘合、标签的印刷等诸工序组成。

[0501] 经过以上的工序,能够制作各实施方式所示的记录介质(BD-ROM)。

[0502] (B) 各实施方式所示的流程图、或功能性构成要素进行的信息处理是用硬件资源来具体实现的,所以可称为利用了自然法则的技术思想,满足作为“程序的发明”的成立必要条件。

[0503] • 本发明的程序的生产方式

[0504] 本发明的程序可以如下来制作。首先,软件开发者用编程语言来描述实现各流程图、或功能性构成要素的源程序。在该描述时,软件开发者根据编程语言的语法,用类结构或变量、数组变量、外部函数的调用,来描述实现各流程图或功能性构成要素的源程序。

[0505] 具体地说,流程图中的循环处理用上述语法中规定的 for 语句等来描述。判断处理用上述语法中规定的 if 语句、switch 语句等来描述。对解码器的再现控制、或驱动器的访问控制等对硬件的控制通过调用硬件的制造商提供的外部函数来描述。

[0506] 描述的源程序作为文件被提供给编译器。编译器编译这些源程序来生成目标程序。

[0507] 编译器进行的编译由语法分析、优化、资源分配、代码生成等过程组成。在语法分析中,进行源程序的词法分析、语法分析及语义分析,将源程序变换为中间程序。在优化中,对中间程序进行基本块化、控制流分析、数据流分析这些作业。在资源分配中,为了适合目标处理器的指令集,将中间程序中的变量分配给目标处理器具有的寄存器或存储器。在代码生成中,将中间程序内的各中间指令变换为程序代码,得到目标程序。

[0508] 这里生成的目标程序由使计算机执行各实施方式所示的流程图的各步骤、或功能性构成要素的各个过程的 1 个以上的程序代码构成。这里,程序代码有许多种类,像处理器的本机代码、JAVA(注册商标)字节码等。用程序代码来实现各步骤有各种各样的形态。在能够利用外部函数来实现各步骤的情况下,调用该外部函数的调用语句就成为程序代码。此外,也有时实现 1 个步骤的程序代码归属于不同的目标程序。在限制指令种类的 RISC 处理器中,也可以通过组合算术运算指令或逻辑运算指令、转移指令等,来实现流程图的各步骤。

[0509] 生成了目标程序后,程序员对它们运行连接程序。连接程序将这些目标程序、或关联的库程序分配给存储空间,将它们连接为 1 个,生成加载模块。这样生成的加载模块是将计算机进行的读取作为前提的,使计算机执行各流程图所示的处理过程或功能性构成要素的处理过程。经过以上的处理,能够制作本发明的程序。

[0510] (C) 本发明的程序可以如下使用。

[0511] (i) 用作嵌入程序

[0512] 在将本发明的程序用作嵌入程序的情况下,将与程序相当的加载模块与基本输入输出程序(BIOS)或各种各样的中间件(操作系统)一起写入到指令 ROM 中。通过将这种指令 ROM 包含到控制部中,使 CPU 来执行,能够将本发明的程序用作再现装置的控制程序。

[0513] (ii) 用作应用

[0514] 在再现装置是内置硬盘模型的情况下,基本输入输出程序(BIOS)被包含在指令ROM中,各种各样的中间件(操作系统)被预装在硬盘上。此外,用于从硬盘起动系统的引导ROM被设在再现装置中。

[0515] 在此情况下,只将加载模块通过移动型的记录介质或网络,提供给再现装置,作为1个应用安装到硬盘中。这样,再现装置进行基于引导ROM的引导(bootstrapping)、并起动操作系统后,使CPU执行该应用作为1个应用,使用本发明的程序。

[0516] 在内置硬盘模型的再现装置中,可将本发明的程序用作1个应用,所以可以以单体来转让、出租、或者通过网络来提供本发明的程序。

[0517] (D) 本发明的系统LSI的生产·使用行为

[0518] 所谓系统LSI,是指将裸片安装、封装到高密度衬底上而成的集成电路。通过将多个裸片安装、封装到高密度衬底上,使多个裸片好像具有1个LSI那样的外形构造,这也被包含在系统LSI中(这种系统LSI被称为多芯片模块。)

[0519] 这里,如果着眼于封装的种类,则在系统LSI中,有QFP(方型扁平式封装)、PGA(插针网格阵列)这些种类。QFP是在封装的四个侧面上安装了管脚的系统LSI。PGA是在整个底面上安装了许多管脚的系统LSI。

[0520] 这些管脚担负与驱动器装置的输入输出接口、与遥控器的输入接口、与电视的接口等IEEE1394接口或与PCI总线的接口的作用。系统LSI中的管脚有这种接口的作用,所以通过在系统LSI的这些管脚上连接驱动器等或再现装置的各种电路,使系统LSI起再现装置的核心作用。

[0521] 所谓系统LSI中封装的裸片,是实现各实施方式中作为内部结构图示出的各构成要素的功能的指令ROM或CPU、解码器LSI等。

[0522] 如前面“用作嵌入程序”中所述,在指令ROM中,写入了与程序相当的加载模块、或基本输入输出程序(BIOS)、各种各样的中间件(操作系统)。在本实施方式中,特别创作的是与该程序相当的加载模块的部分,所以通过将保存着与程序相当的加载模块的指令ROM作为裸片来封装,能够生产本发明的系统LSI。

[0523] 作为具体的实现,最好是SoC实现或SiP实现。所谓SoC(System onchip,系统芯片)封装,是在1个芯片上刻制多个电路的技术。而所谓SiP(System in Package,系统级封装)封装,是将多个芯片用树脂等做成1个封装的技术。经过以上的过程,本发明的系统LSI能够根据各实施方式所示的再现装置的内部结构图来制作。

[0524] 其中,如上所述生成的集成电路由于集成度的差异,有时也称为IC、LSI、Super LSI(超LSI)、Ultra LSI(甚LSI)。

[0525] 再者,也可以用1个芯片来构成各再现装置的构成要素的一部分或全部。集成电路化并不限于上述SoC实现、SiP实现,也可以用专用电路或通用工艺来实现。可以利用在制造LSI后可编程的FPGA(FieldProgrammable Gate Array;现场可编程门阵列)、或可重构LSI内部的电路单元的连接或设定的可重构处理器。再者,如果由于半导体技术的进步或派生的技术而出现了代替LSI的集成电路化的技术,则当然也可以用该技术来进行功能块的集成电路化。例如,有可能采用生物技术等。

[0526] (E) 在所有实施方式中,用BD-ROM实施了本发明的光盘,但是也可以的任何记录

介质。例如,也可以是 DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R、CD-RW 等光盘、PD、MO 等磁光盘。

[0527] (F) 各实施方式中的视频流是 BD-ROM 标准的 AVClip,但是也可以是 DVD-Video 标准、DVD-Video Recording 标准的 VOB (VideoObject)。VOB 是通过复用视频流、音频流而得到的符合 ISO/IEC13818-1 标准的程序流。此外, AVClip 中的视频流也可以是 MPEG4 或 WMV 方式的。再者,音频流也可以是 Linear-PCM 方式、Dolby (杜比)-AC3 方式、MP3 方式、MPEG-AAC 方式、dts 方式。

[0528] (G) 在各实施方式中,根据 MPEG4-AVC (也称为 H. 264 或 JVT) 进行了说明,但是也可以是 MPEG2 视频流;此外,在其他形式 (VC-1 等) 的图像的情况下,如果是能够单独解码的图像,则也可以容易地应用。

[0529] 产业上的可利用性

[0530] 本发明的记录介质及再现装置在上述实施方式中公开了内部结构,可以根据该内部结构来量产,所以在实质上可以在工业上利用。由此,本发明的记录介质及再现装置具有产业上的可利用性。

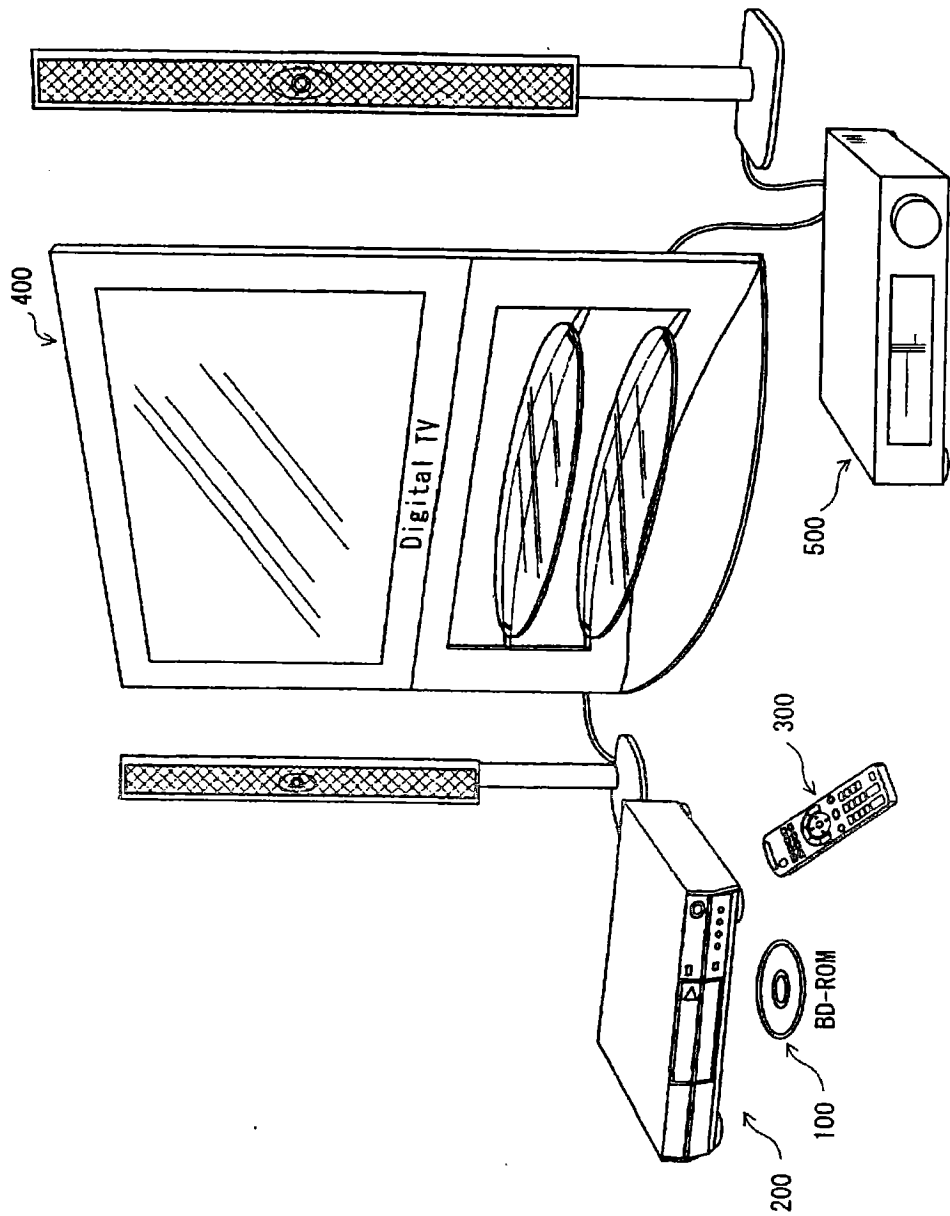


图 1

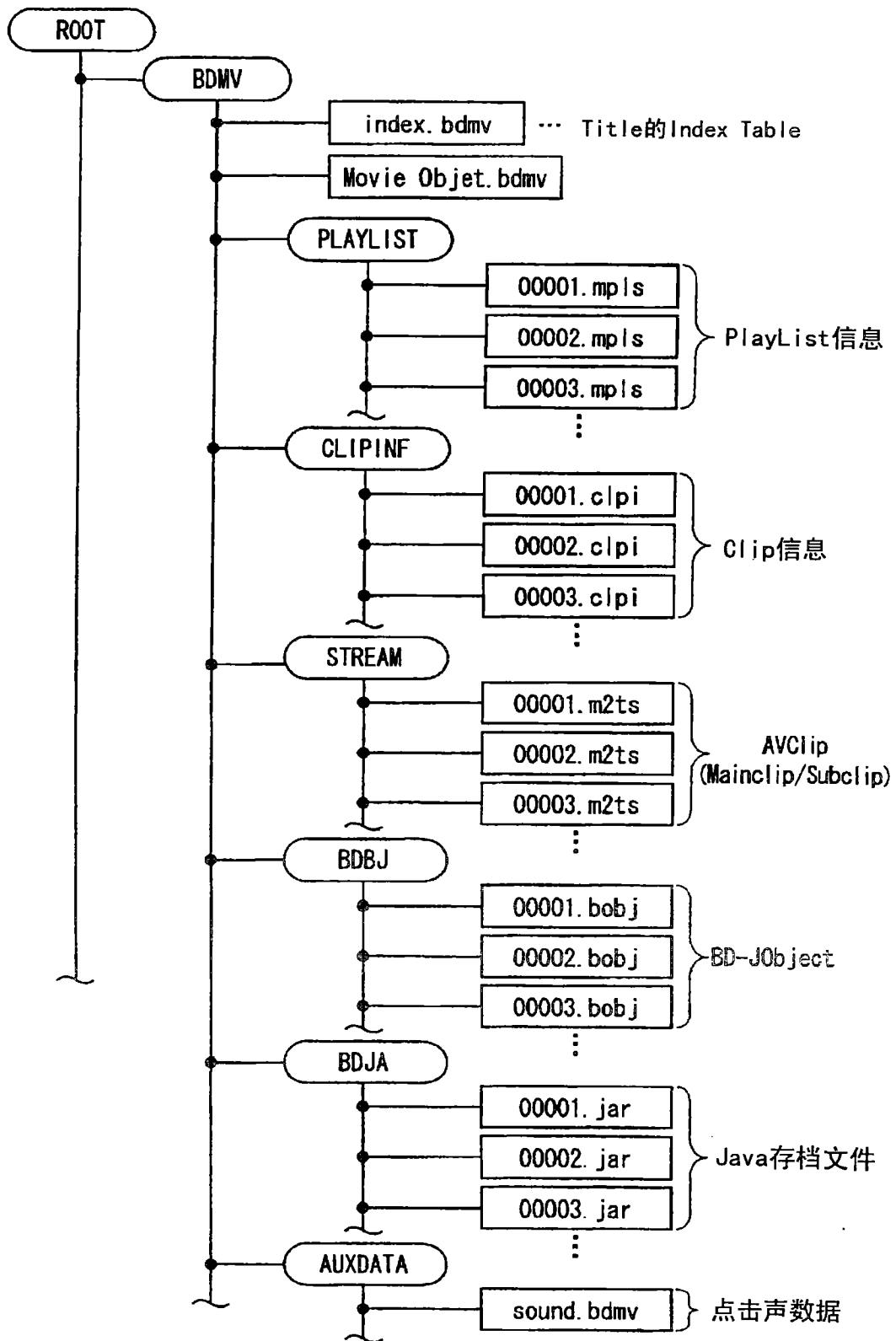


图 2

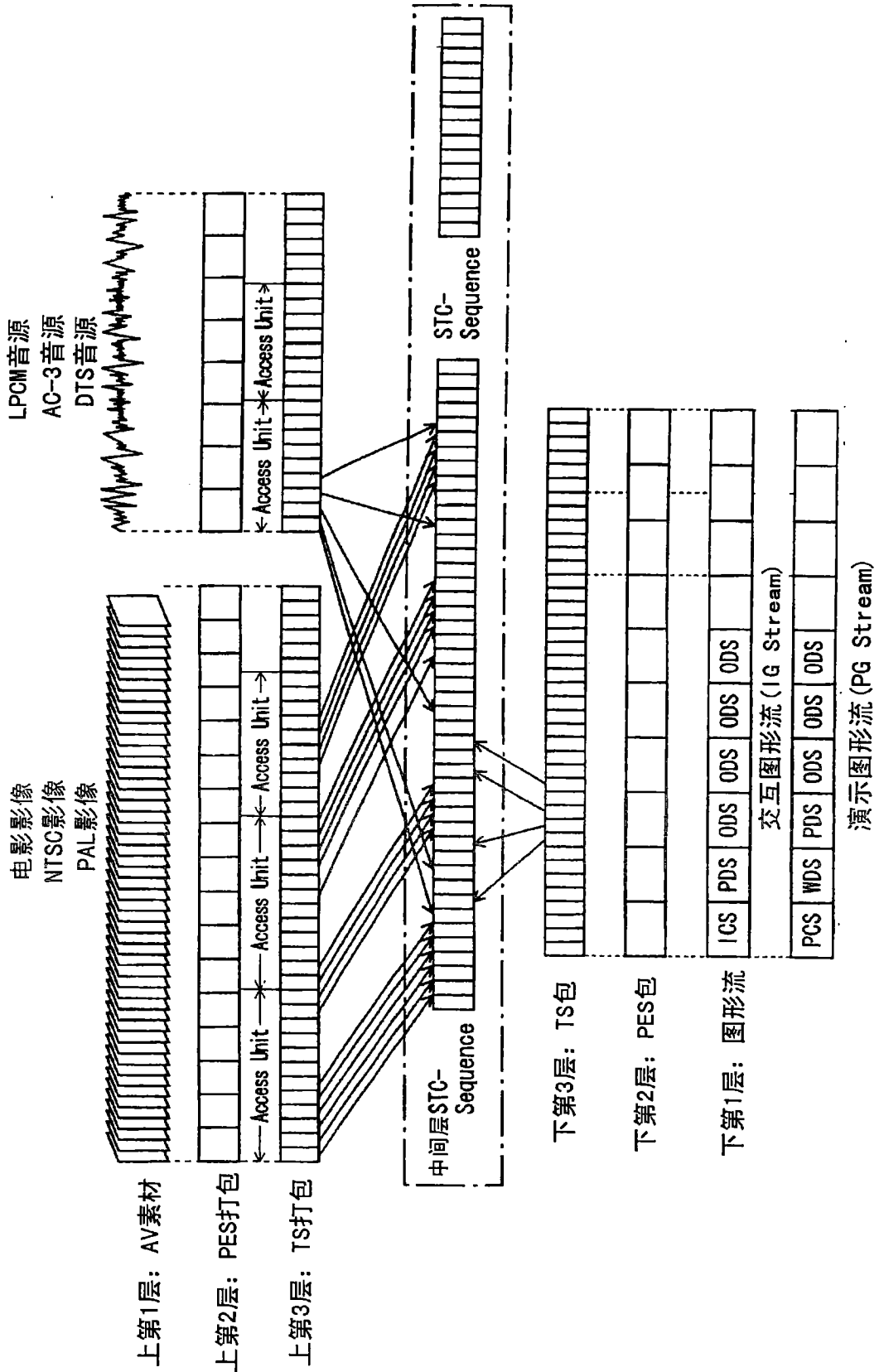


图 3

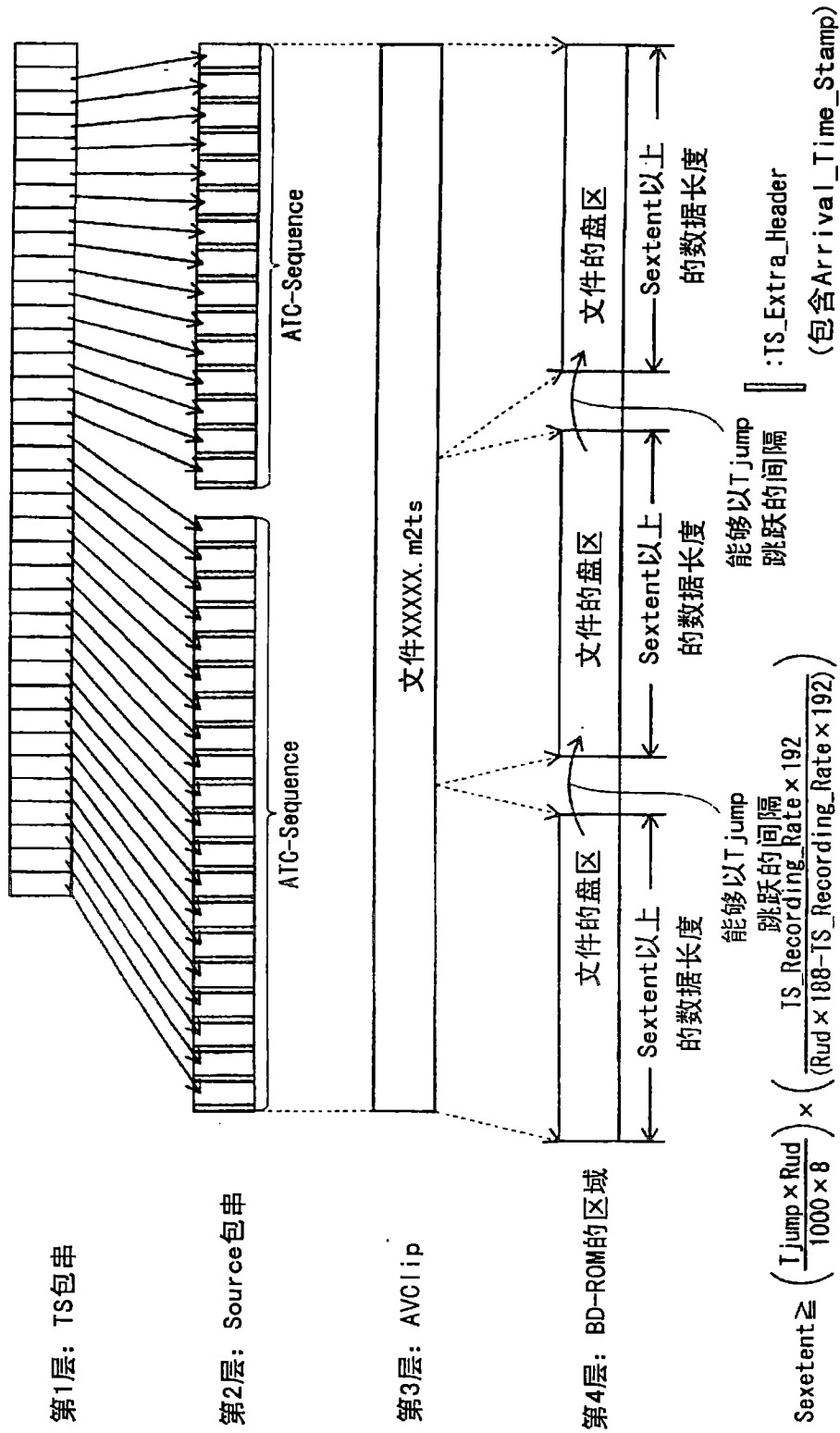


图 4

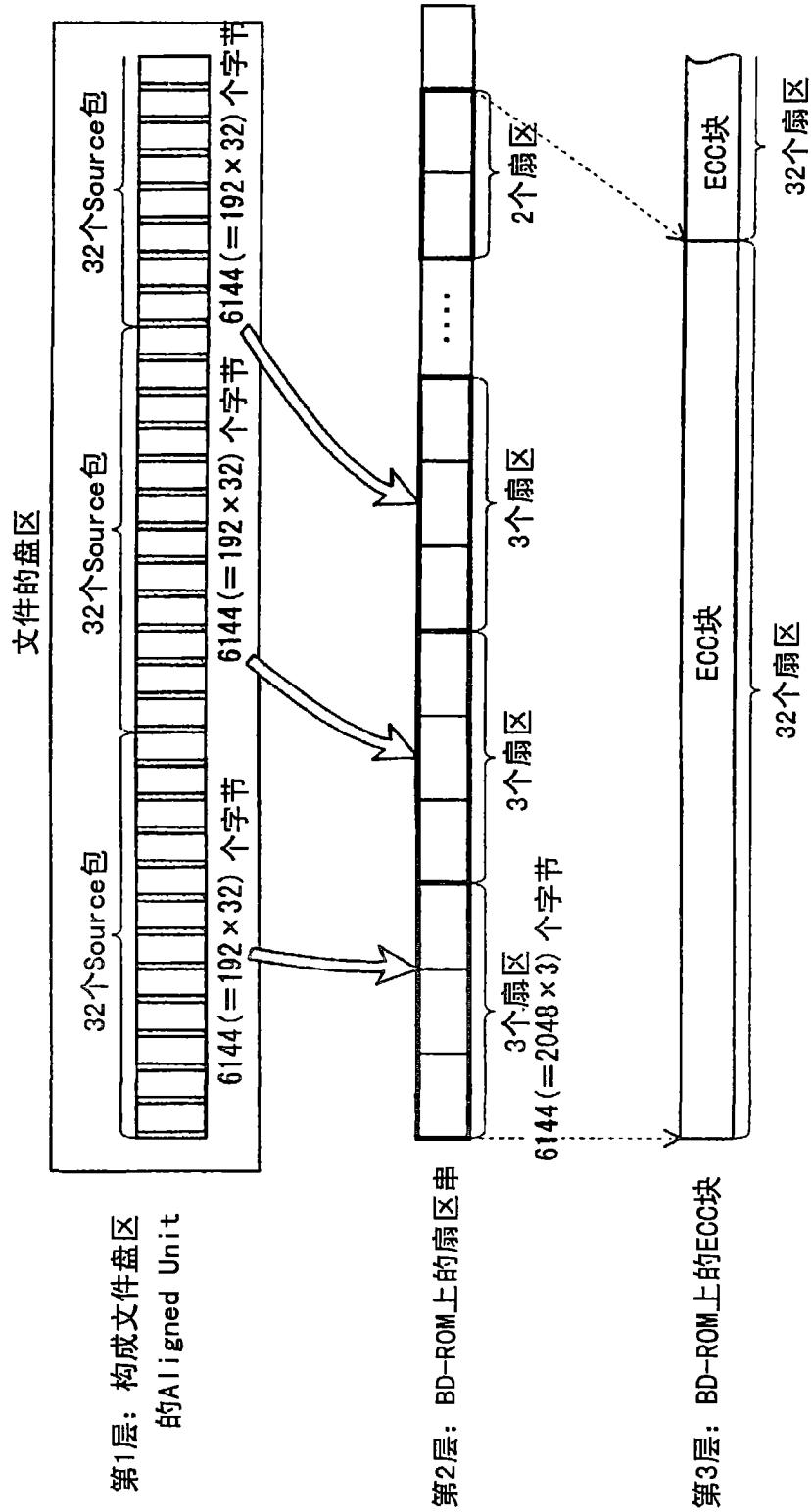


图 5

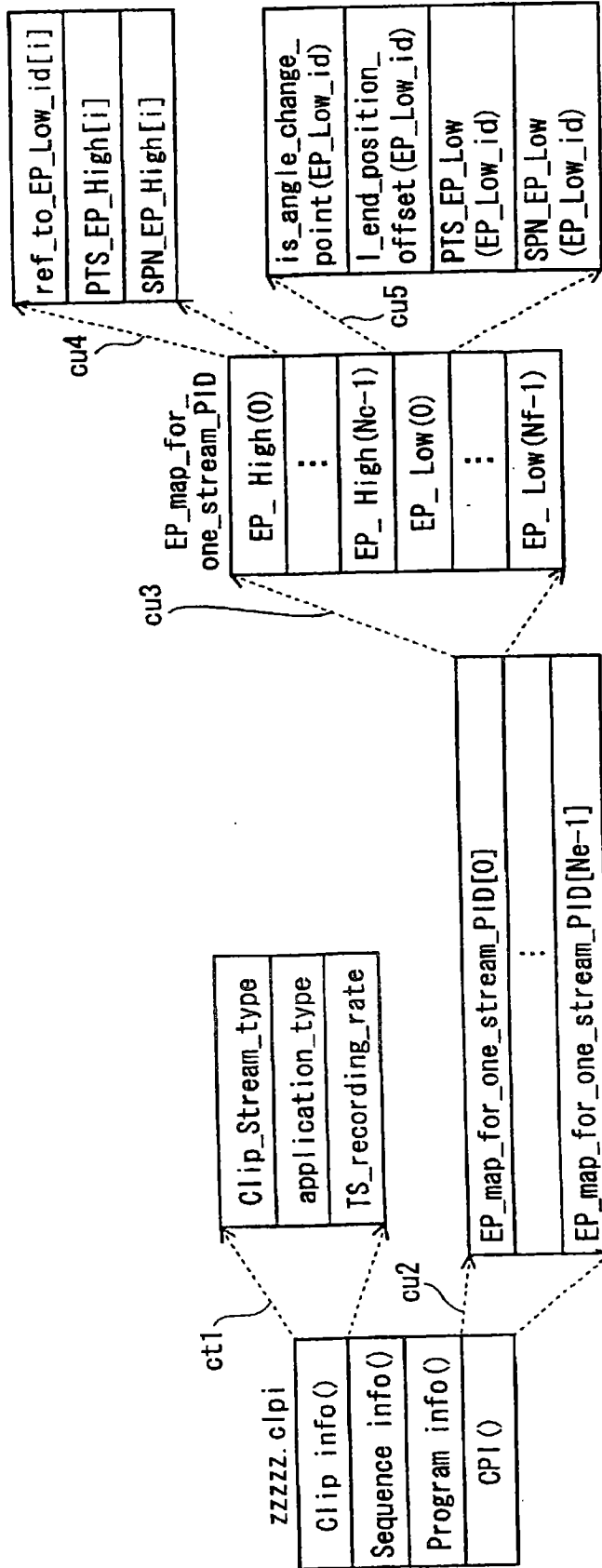


图 6

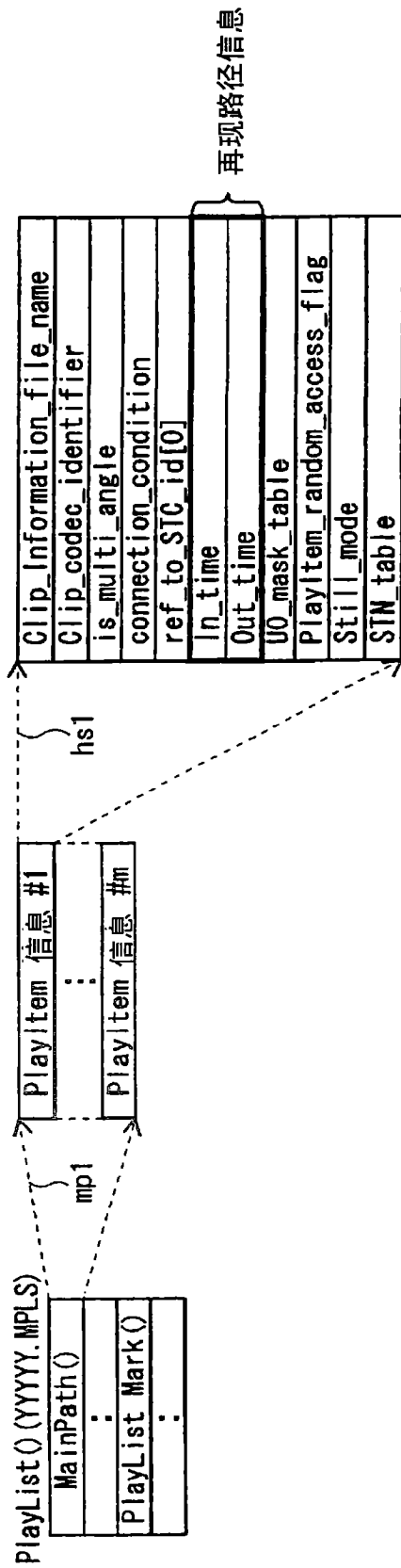


图 8

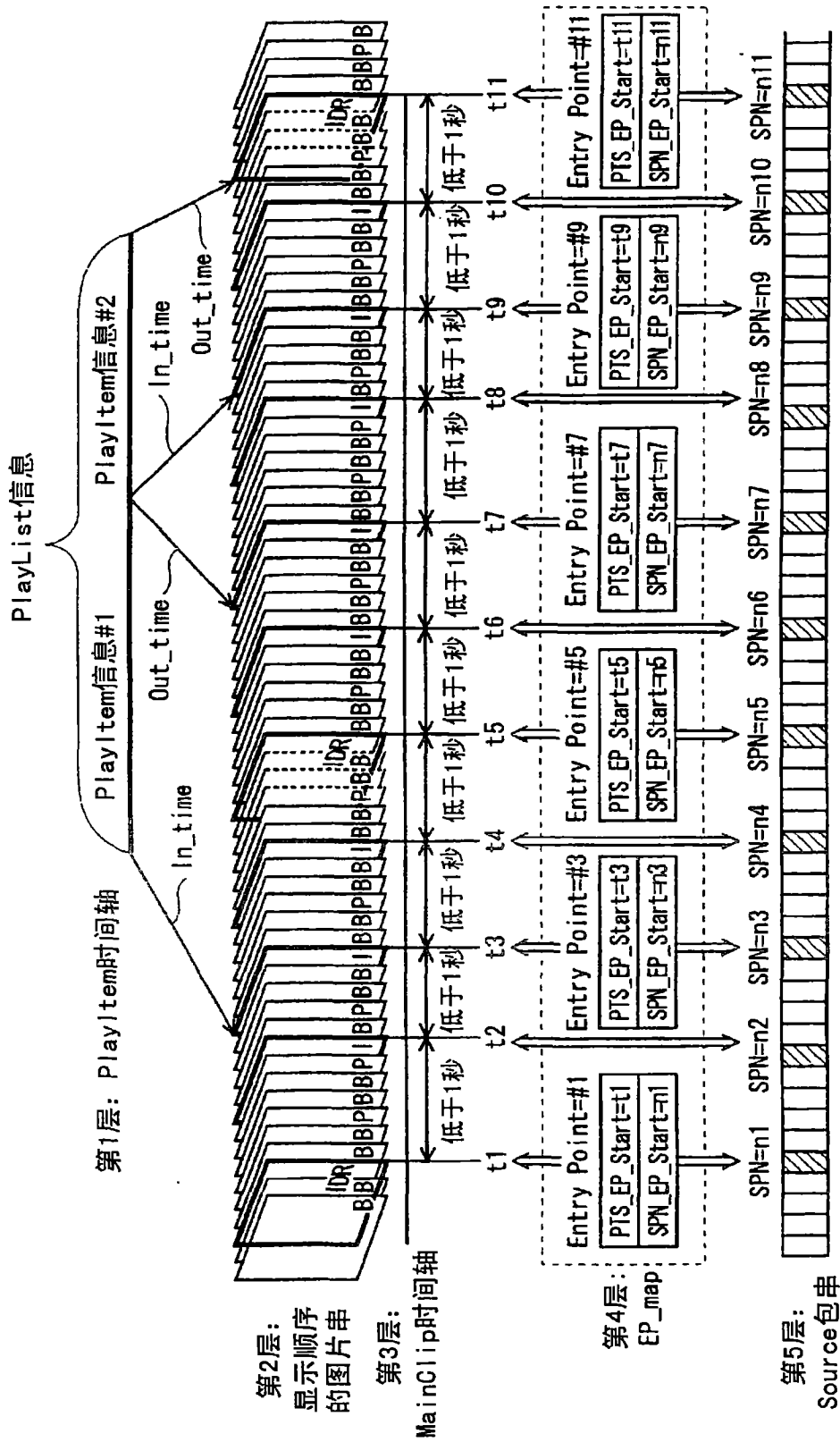


图 9

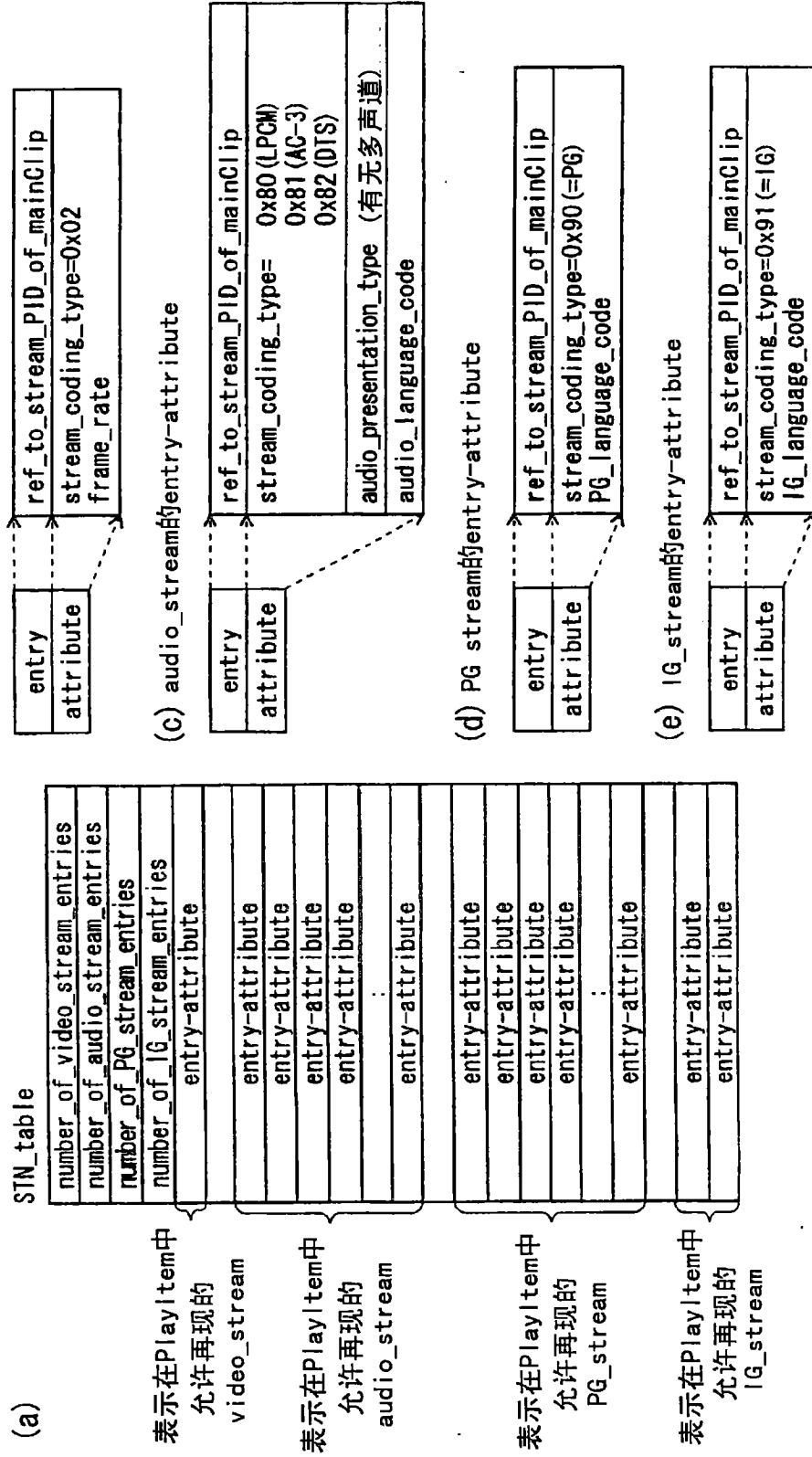


图 10

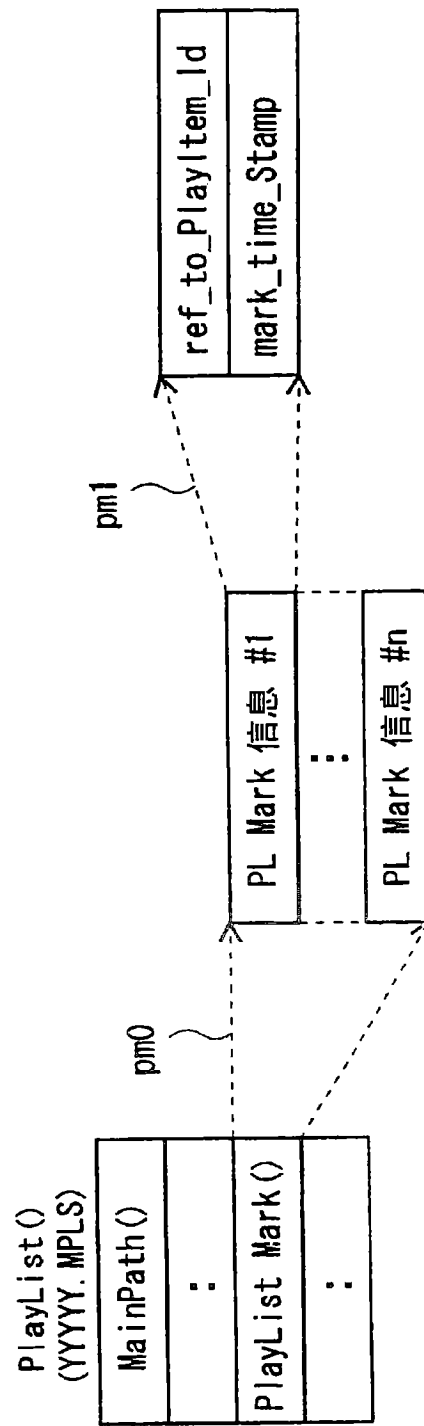


图 11

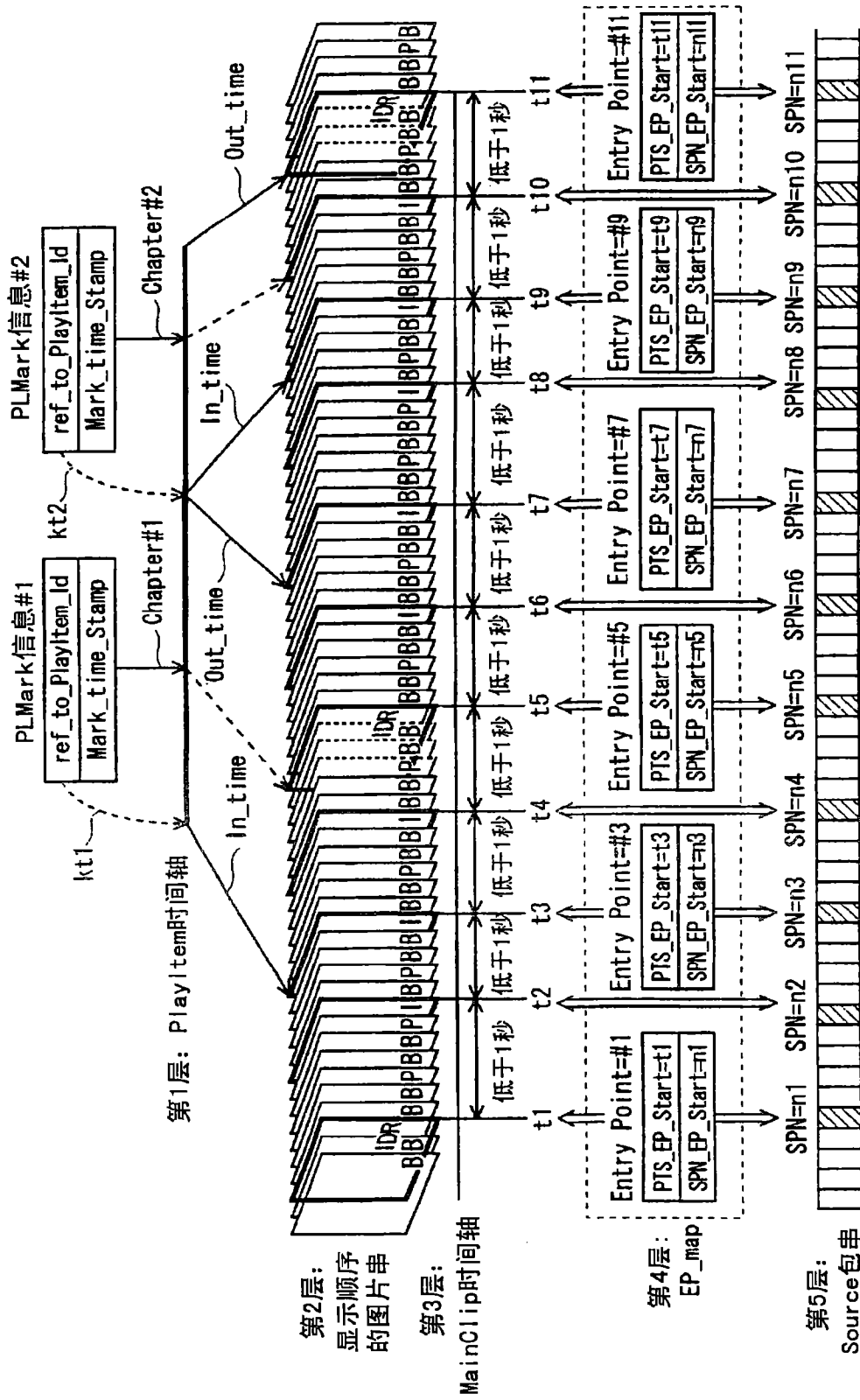


图 12

将Java应用和流相关联的信息 ...BD-JObject
XXXXX. BOBJ

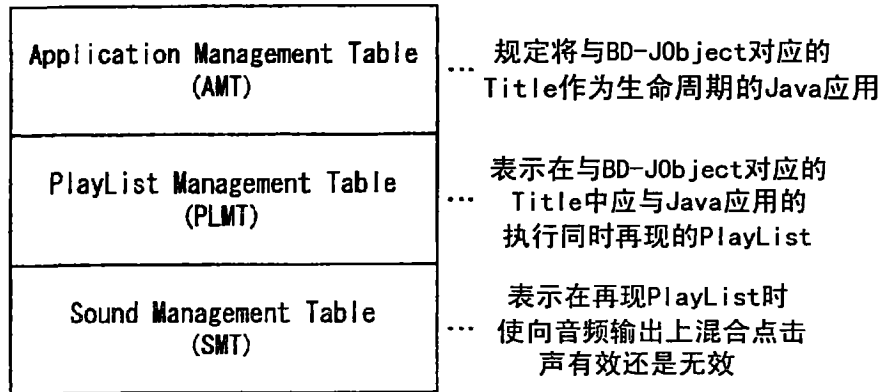


图 13

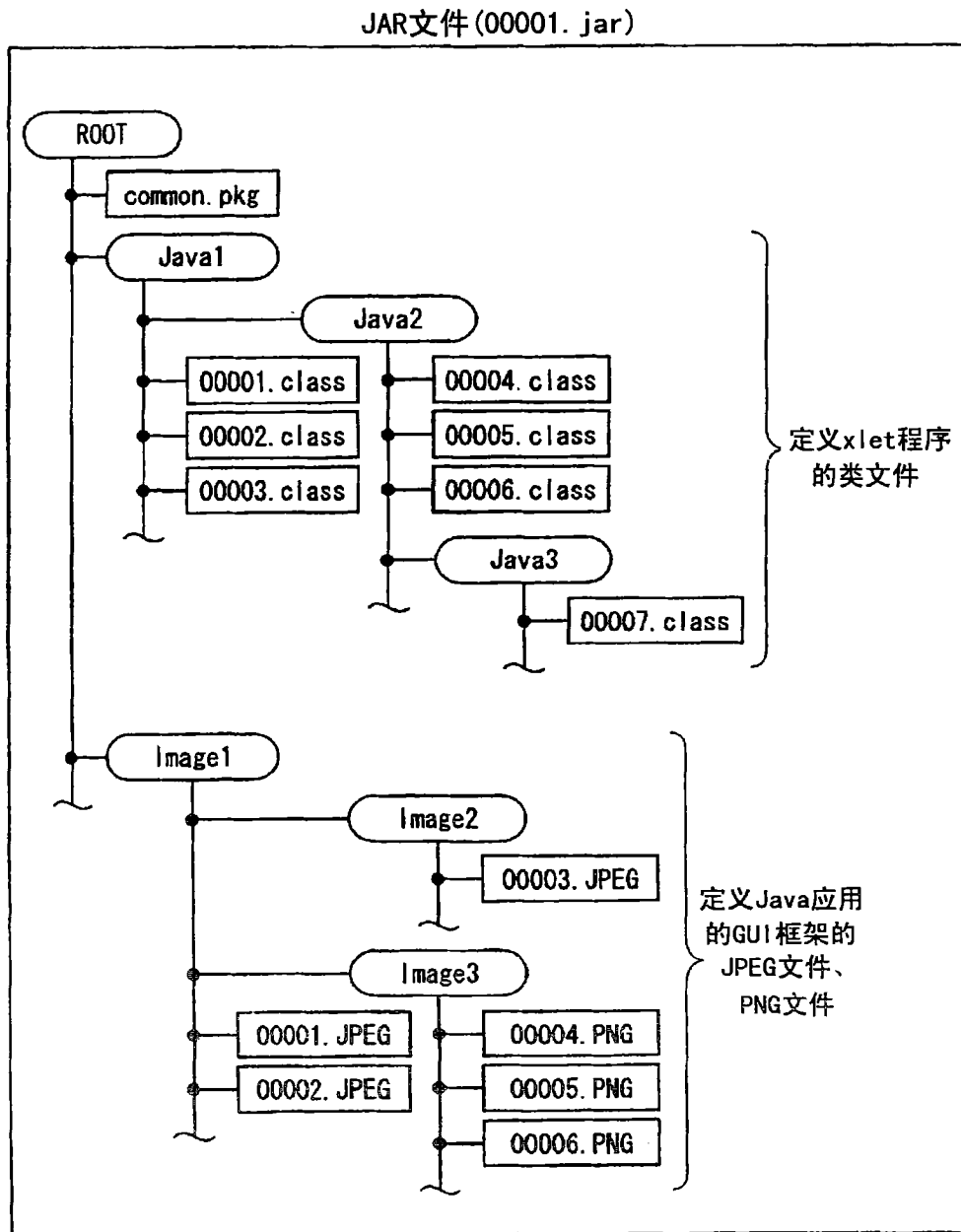


图 14

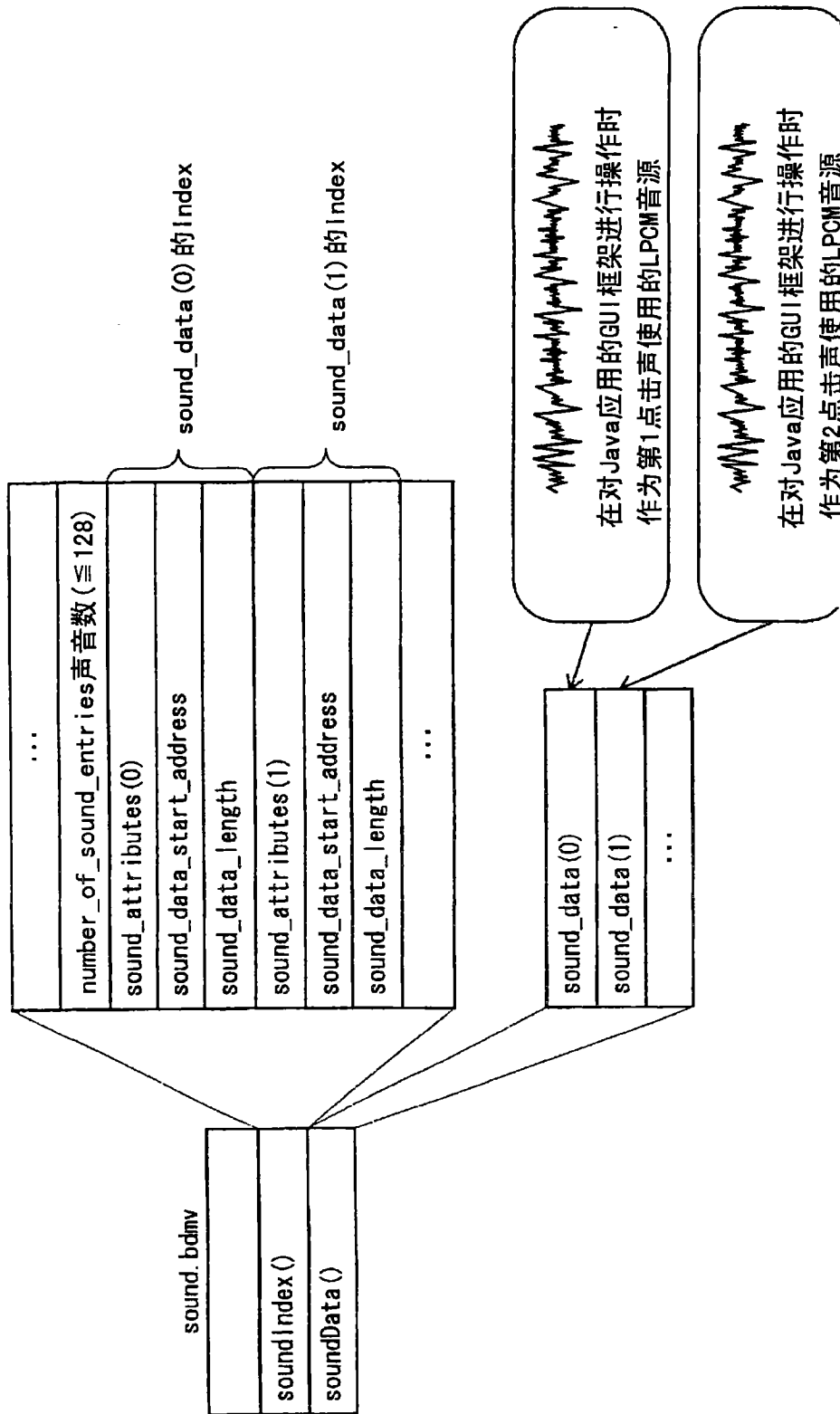


图 15

(a) 应用管理表 (AMT)

life_cycle
apli_id_ref
run_attribute
run_priority

(b) 应用管理表 (AMT)

life_cycle	apli_id_ref	run_attribute	run_priority
用title、PL、Chapter表现的应用的“生命周期”	向JAR的文件名赋予的5位整数ZZZZZ——“应用ID”的参照值	<ul style="list-style-type: none"> • Auto Run • Present (未指定) • Suspend 中的某一个——“运行属性”	取0~255的值的“运行优先级”

图 16

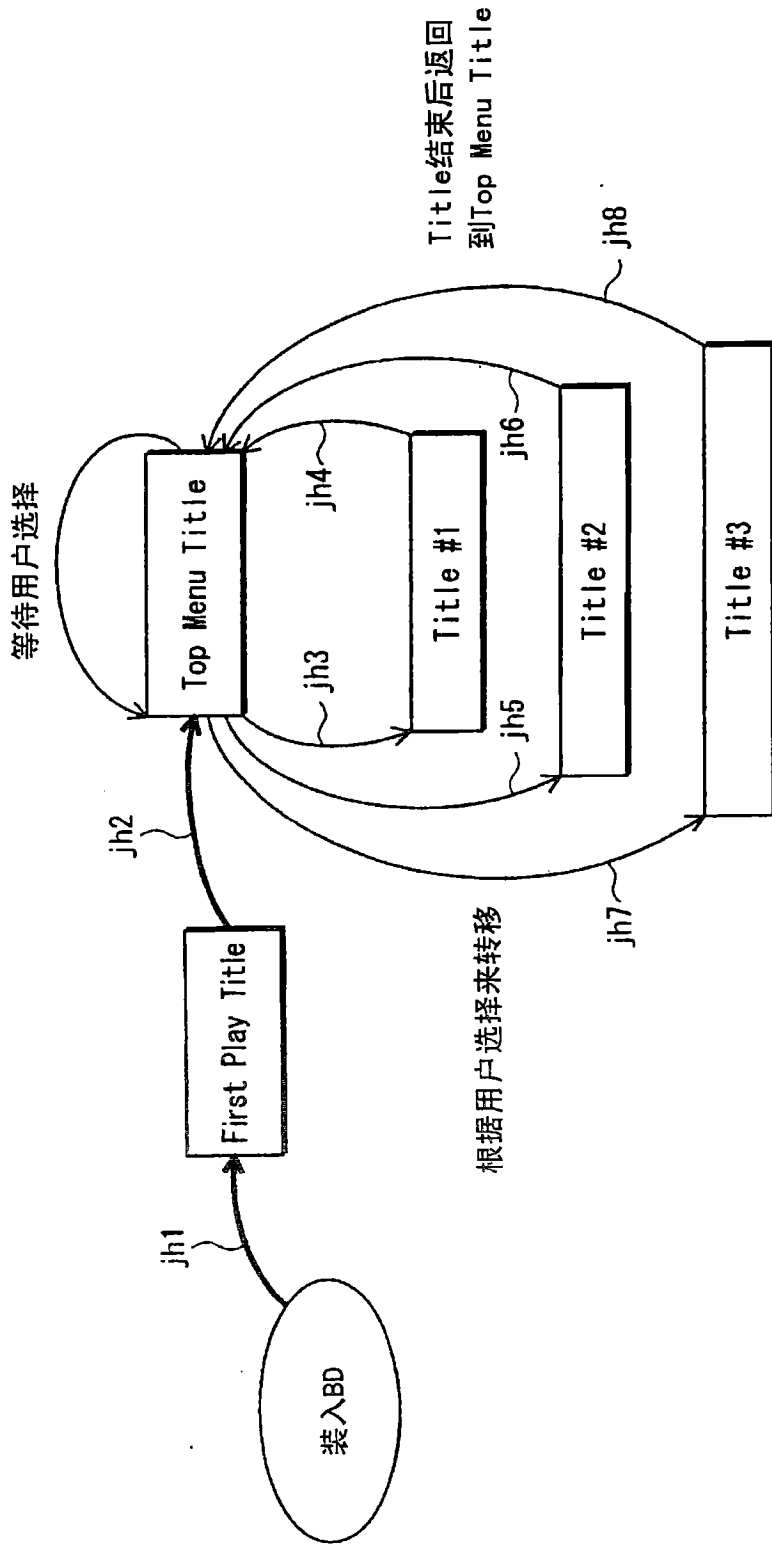


图 17

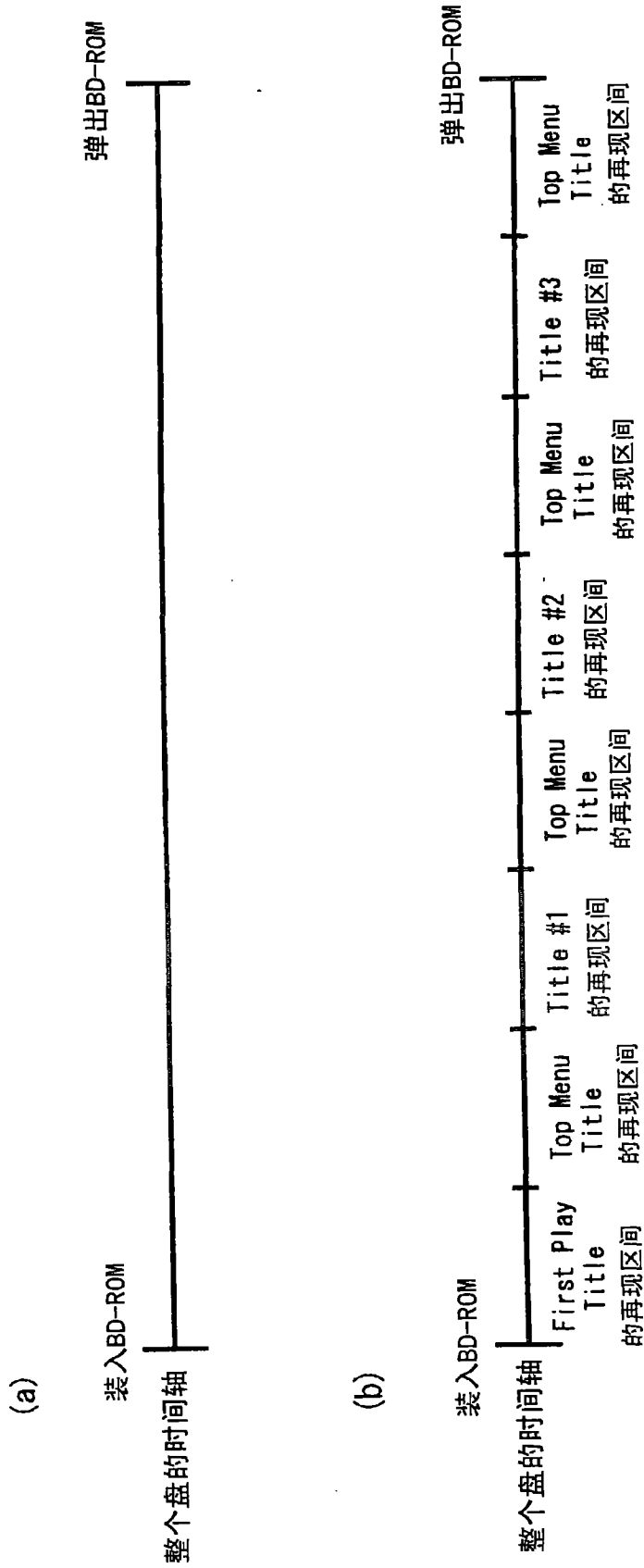
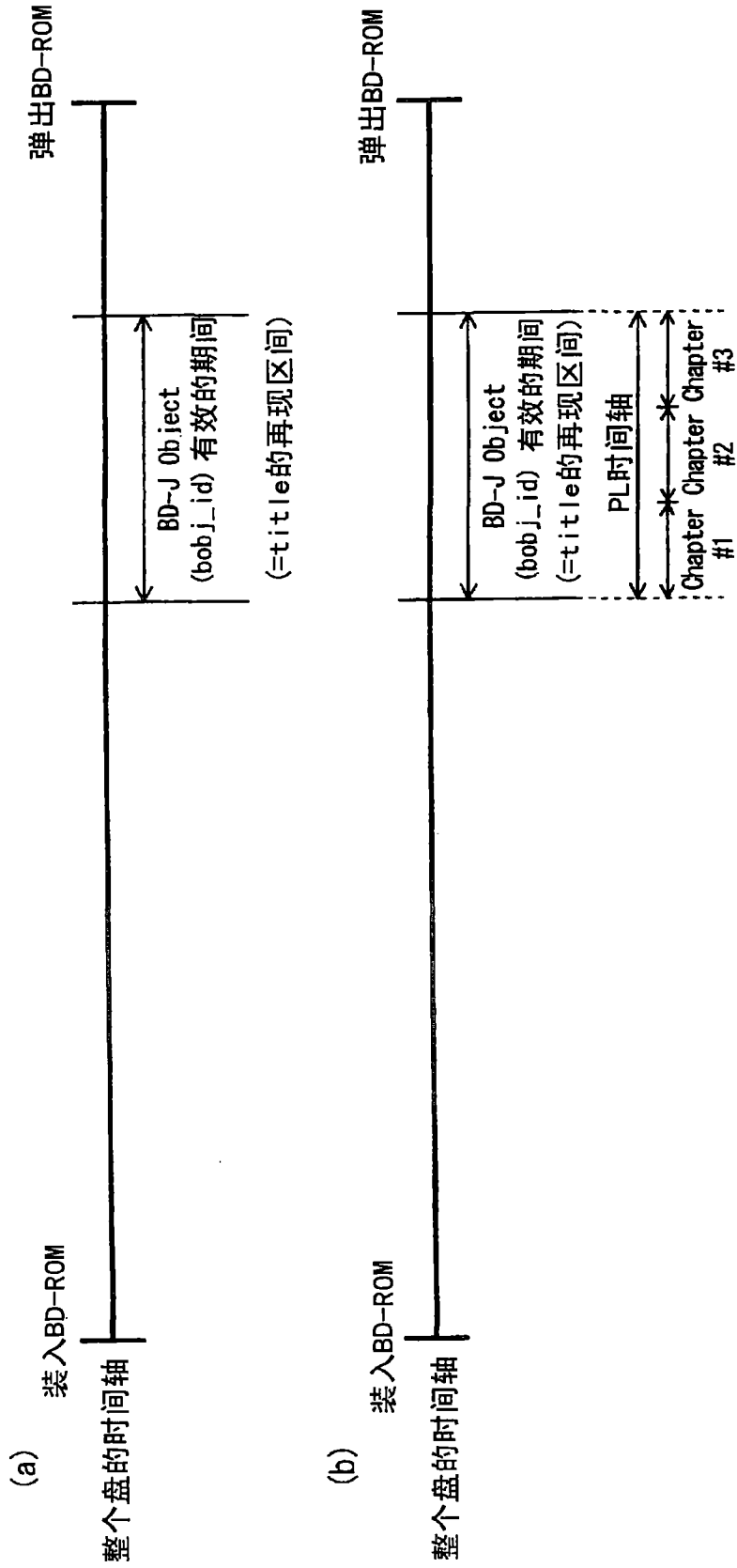


图 18



※在BD-J Object包含PL再现过程的情况下，PL时间轴归属于title再现区间

图 19

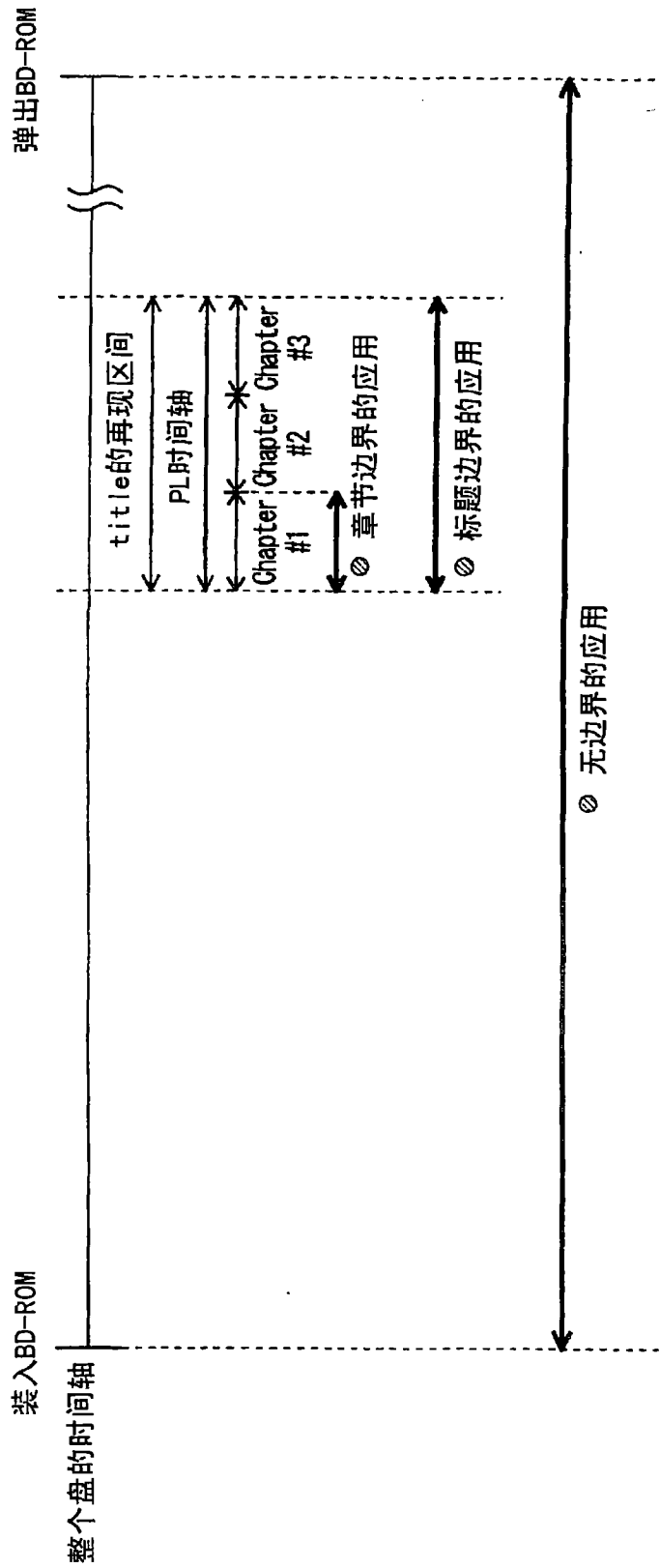


图 20

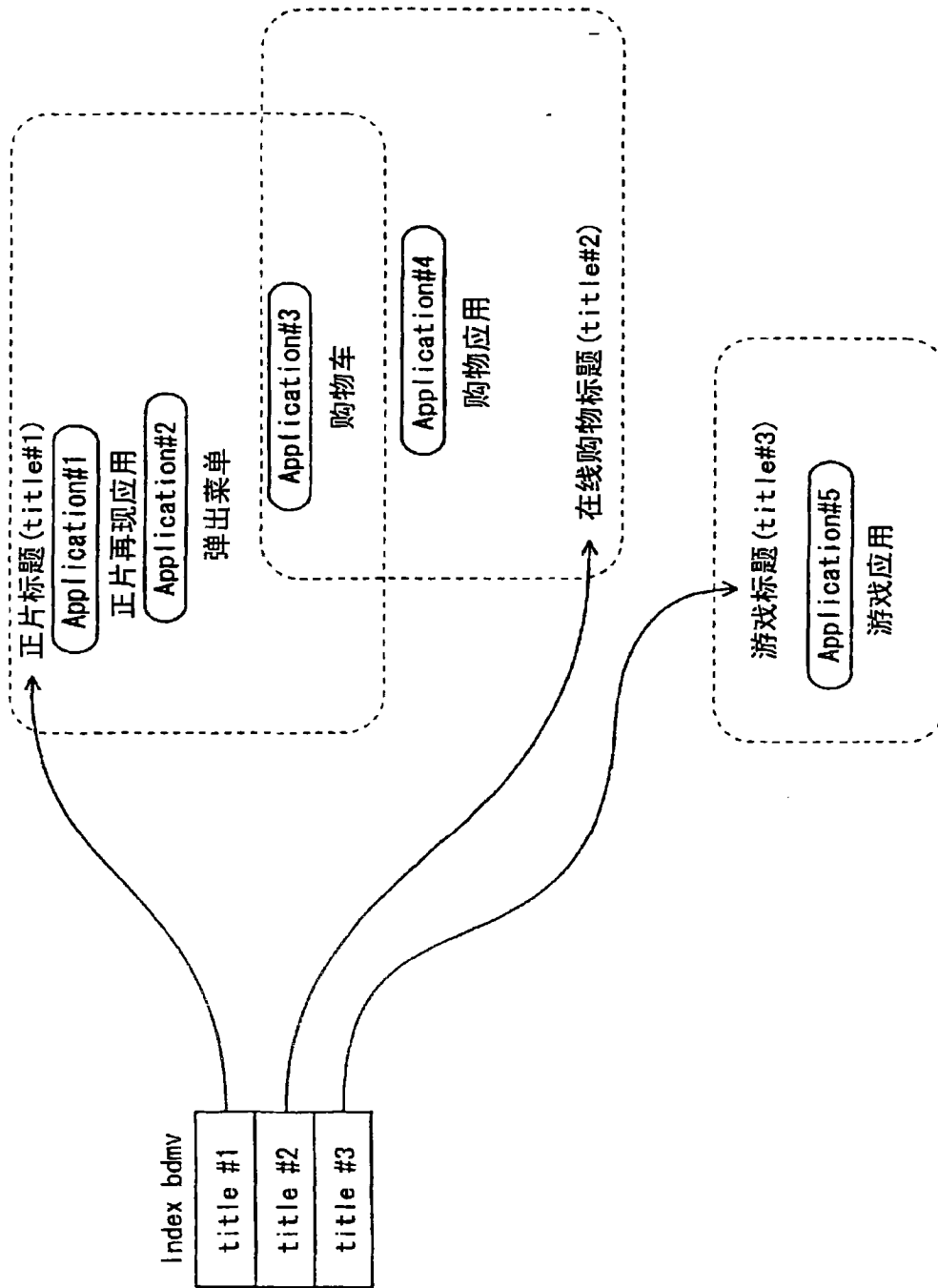
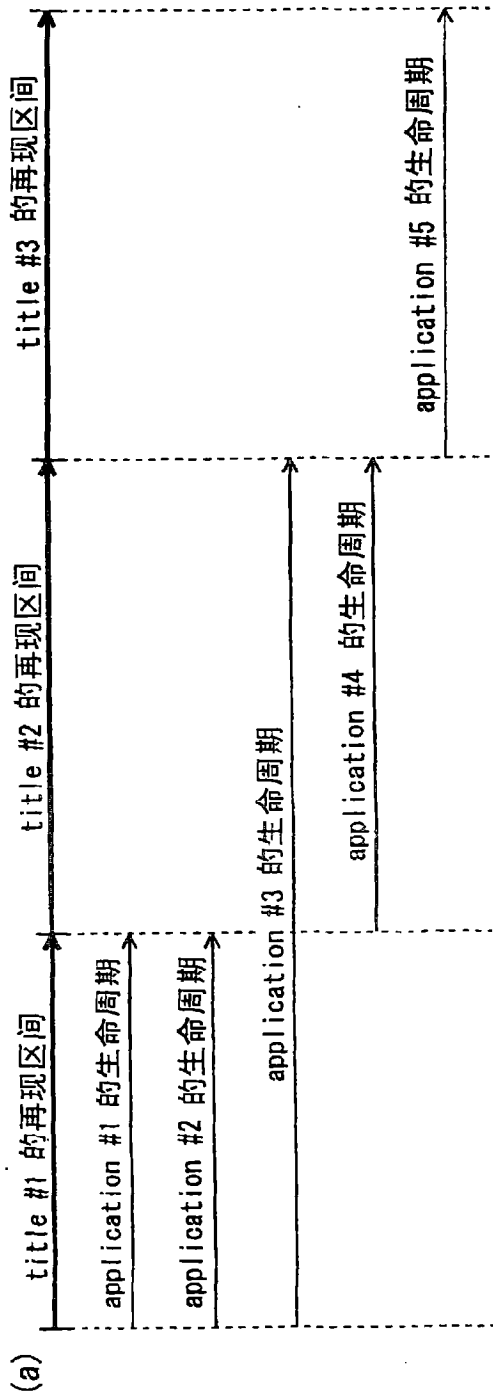


图 21



(b) title #1的应用管理表

生命周期	应用ID的参照值	运行属性	运行优先级
title #1	application #1		
title #1	application #2		
title #1	application #3		

title #2的应用管理表

生命周期	应用ID的参照值	运行属性	运行优先级
title #2	application #3		
title #2	application #4		

title #3的应用管理表

生命周期	应用ID的参照值	运行属性	运行优先级
title #3	application #5		

图 22

紧前标题中的 应用的状态	未运行	运行属性	
		Present	Suspend
		什么也不做, 继续原状态	什么也不做, 继续原状态
	运行中	AutoRun	挂起
		什么也不做, 继续原状态	什么也不做, 继续原状态
	Suspend	恢复	什么也不做, 继续原状态

图 23

(a) 播放列表管理表 (PLMT)

PL_id_ref
Playback_Attribute

(b) 播放列表管理表 (PLMT)

PL_id_ref	Playback_Attribute
MPLS文件的文件名中的5位数值——“Play List ID”	<ul style="list-style-type: none"> • Auto Play • Present (未指定) 中的某一个——“再现属性”

图 24

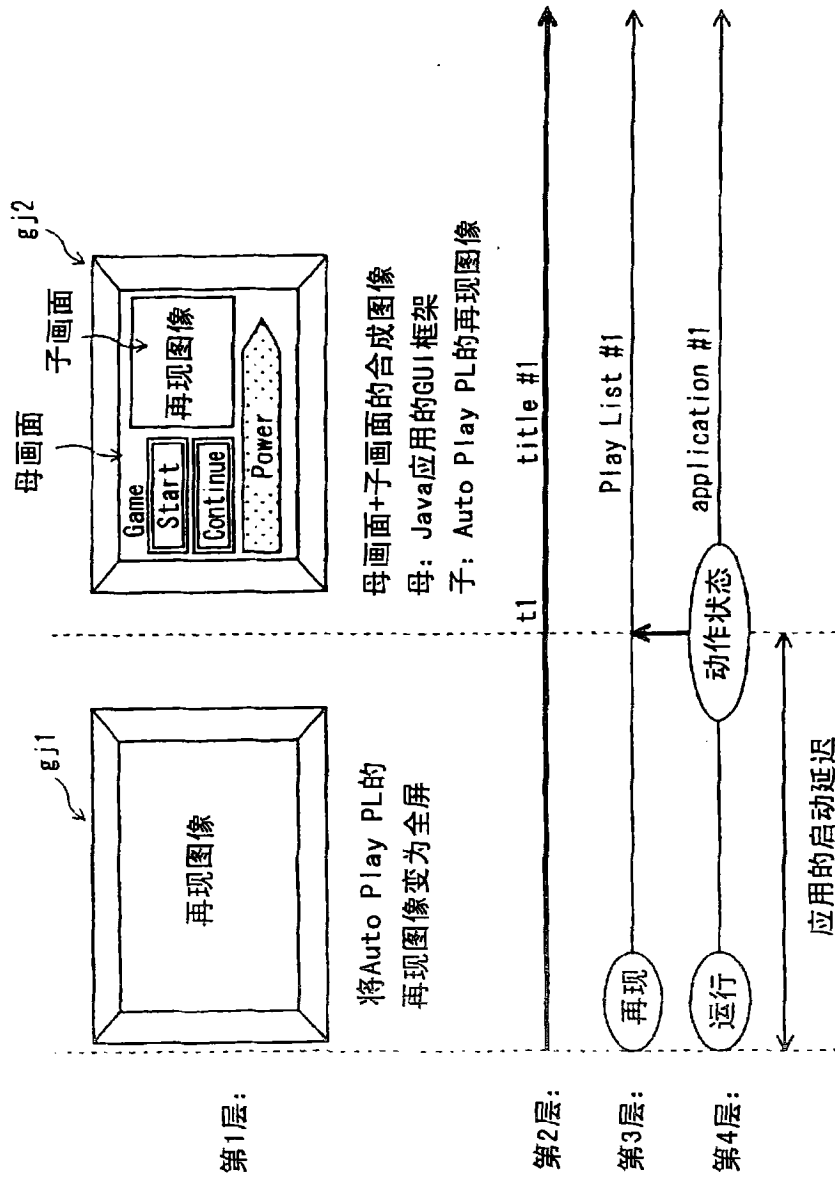


图 25

	在转移目标title中有PLMT	
	再现属性: Auto Play	再现属性: Present
在转移目标title中 没有PLMT	继续原状态	继续原状态
转移源title处于再现中状态	开始自动再现	继续原状态
转移源title处于未再现状态		

图 26

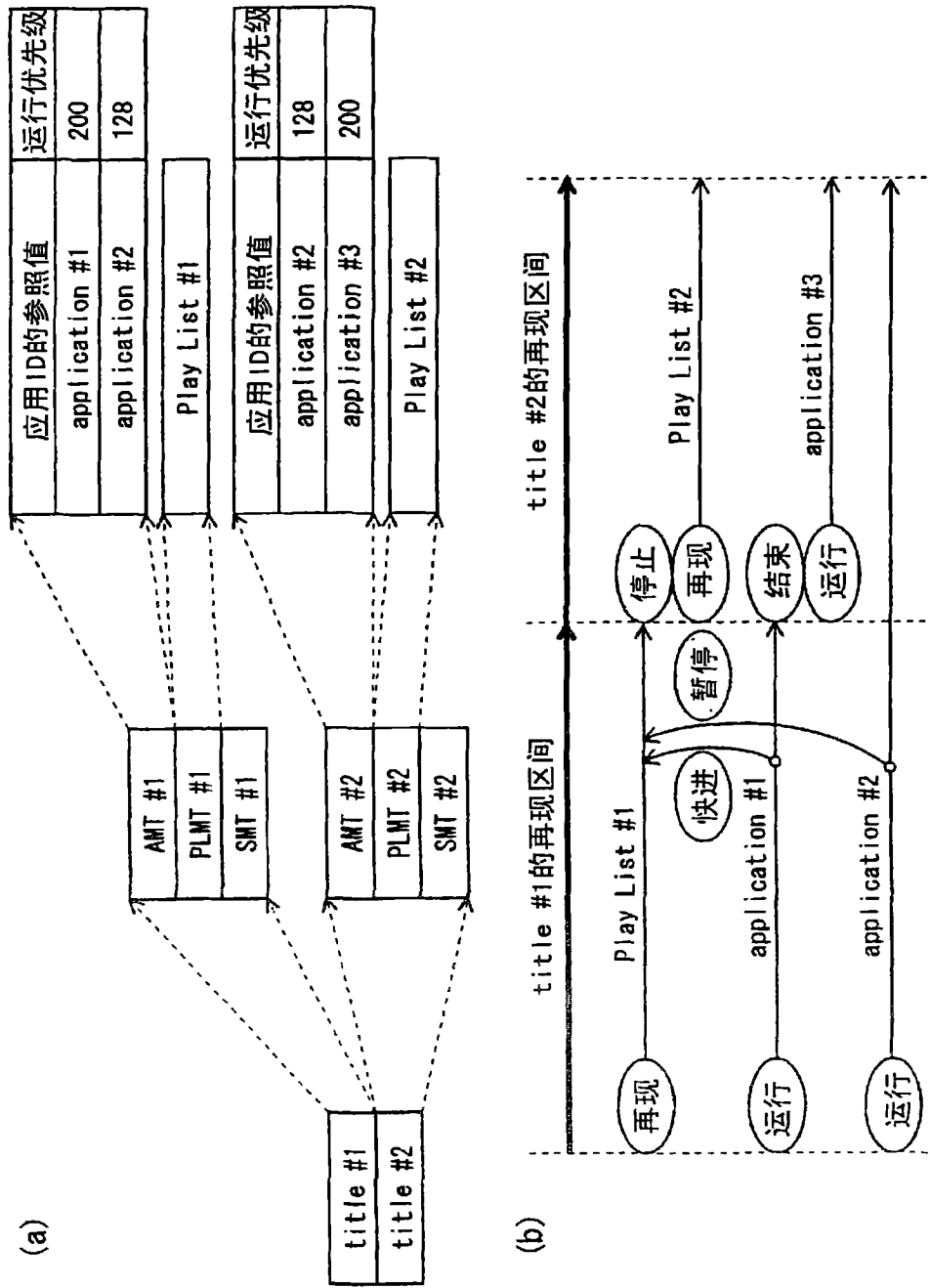


图 27

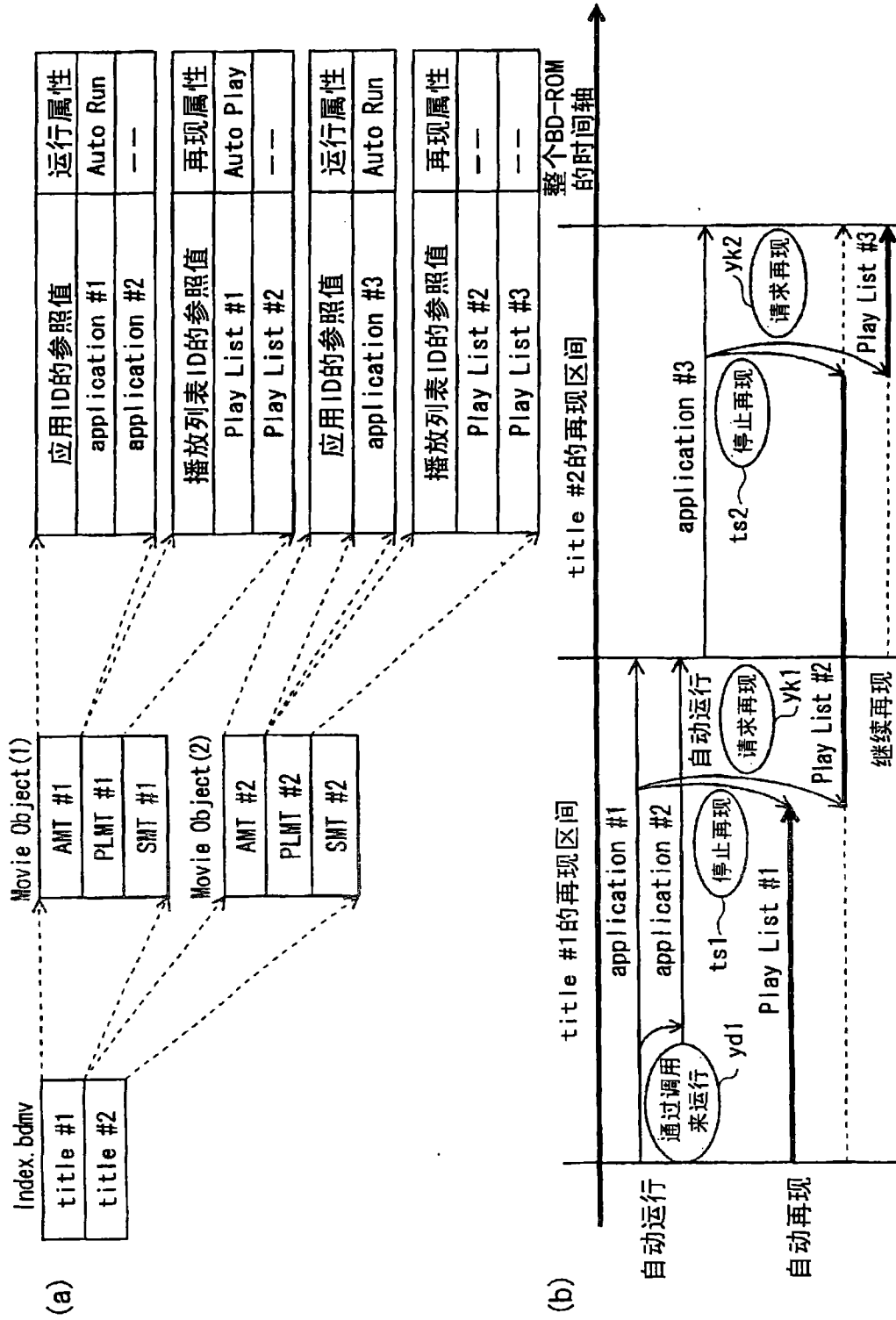


图 28

SMT (声音管理表)

Mixing_ON标志 (使混音有效)	“1” 表示混合用属于同一BD-JOb ject的PLMT再现的 PL的再现声音、和基于sound. bdmv的点击声 “0” 表示不混合用属于同一BD-JOb ject的PLMT再现的 PL的再现声音、和基于sound. bdmv的点击声	缺省值 0
------------------------	---	----------

图 29

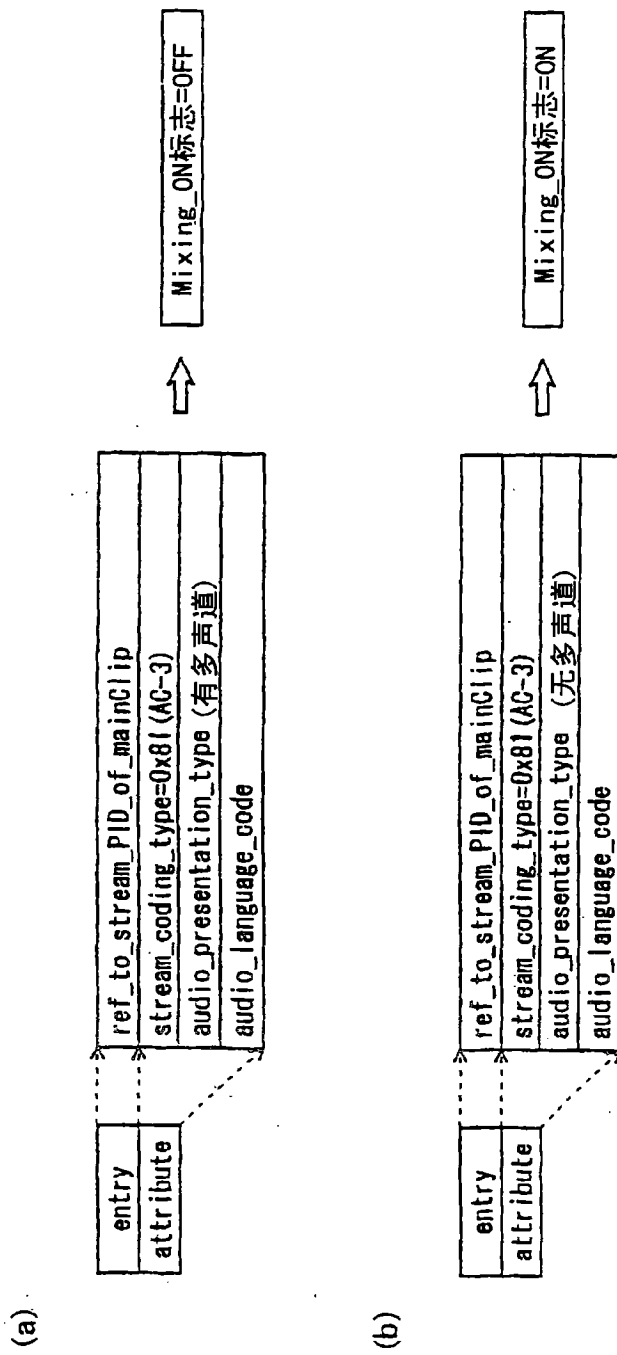


图 30

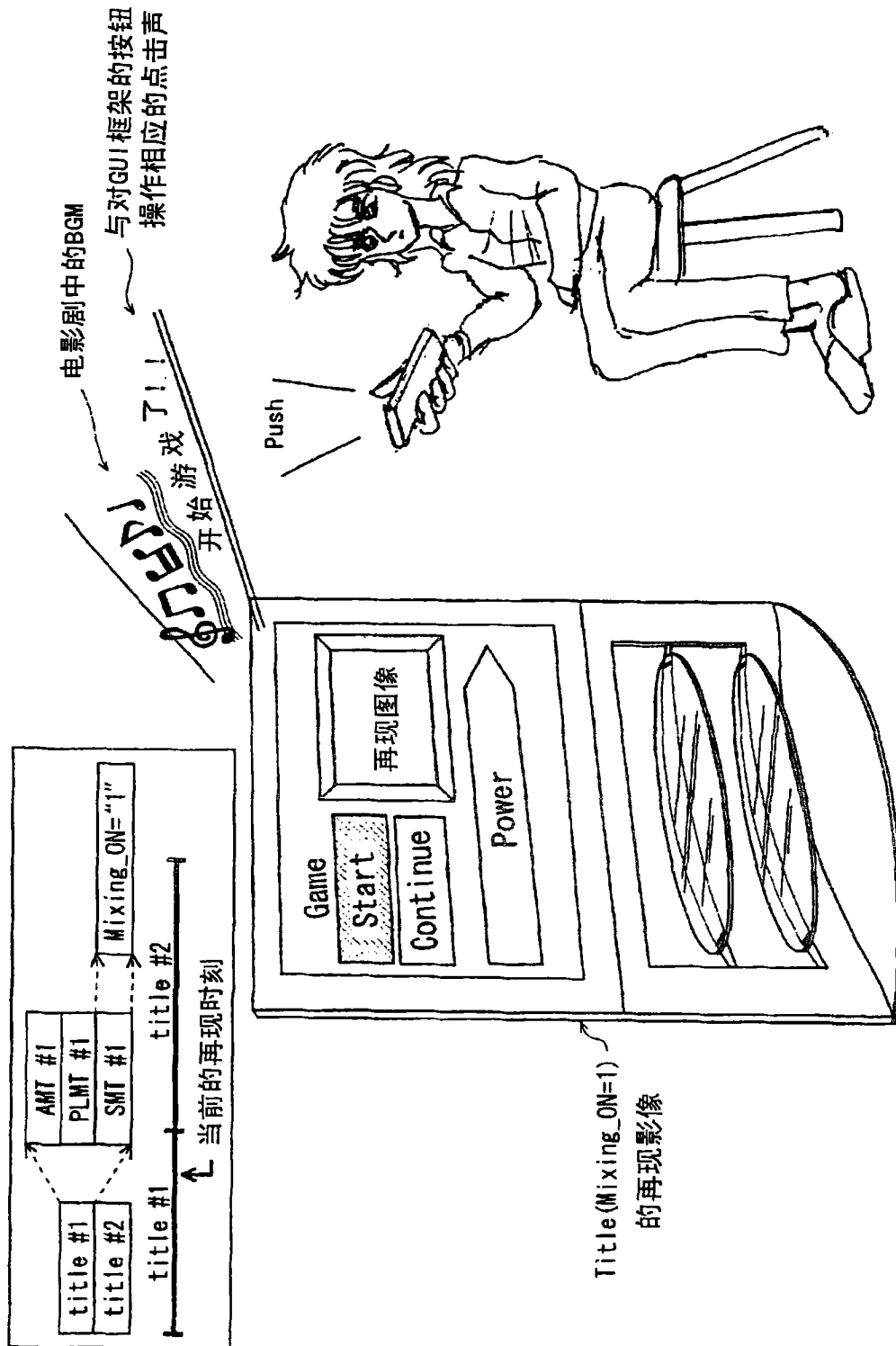


图 31

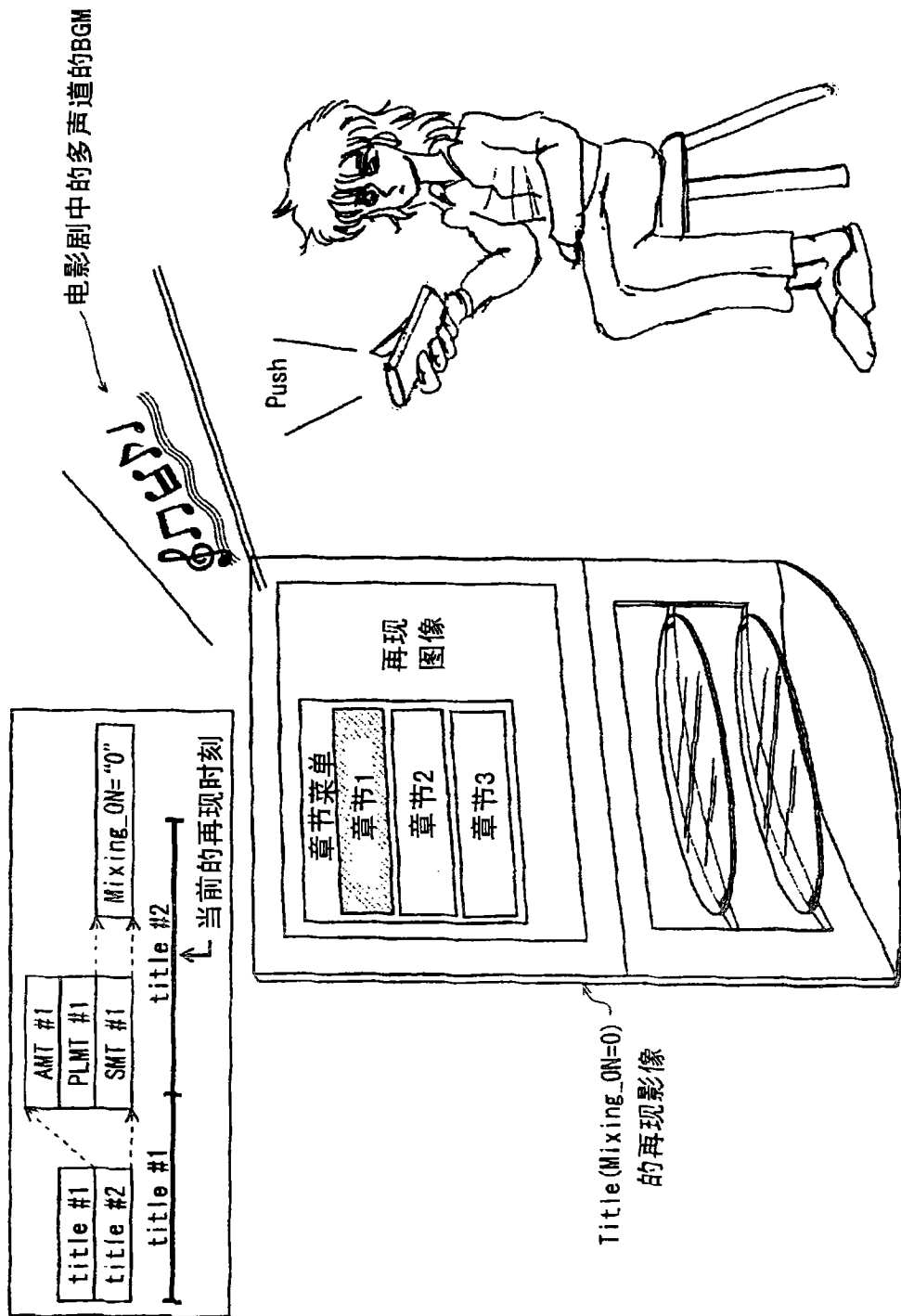


图 32

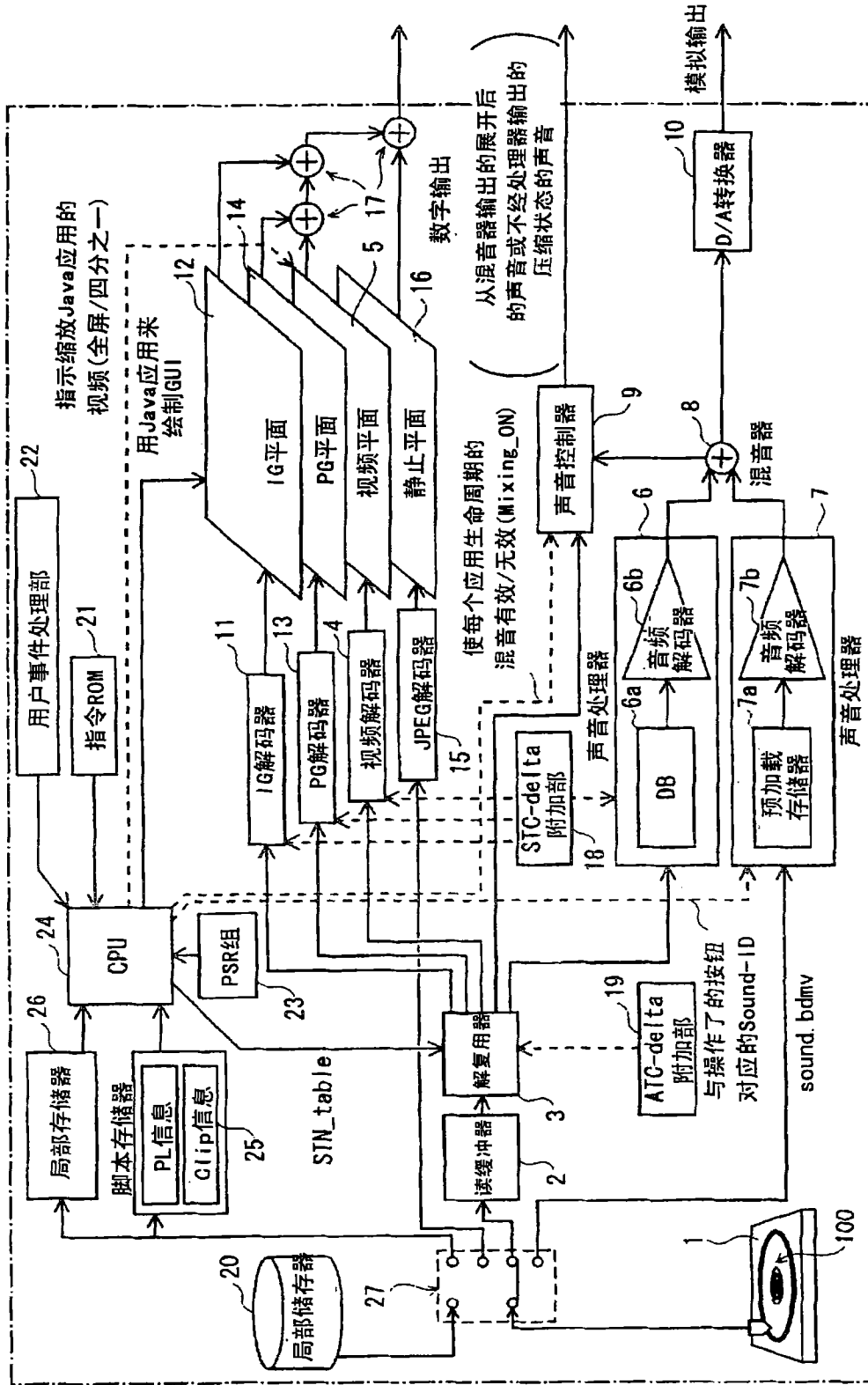


图 33

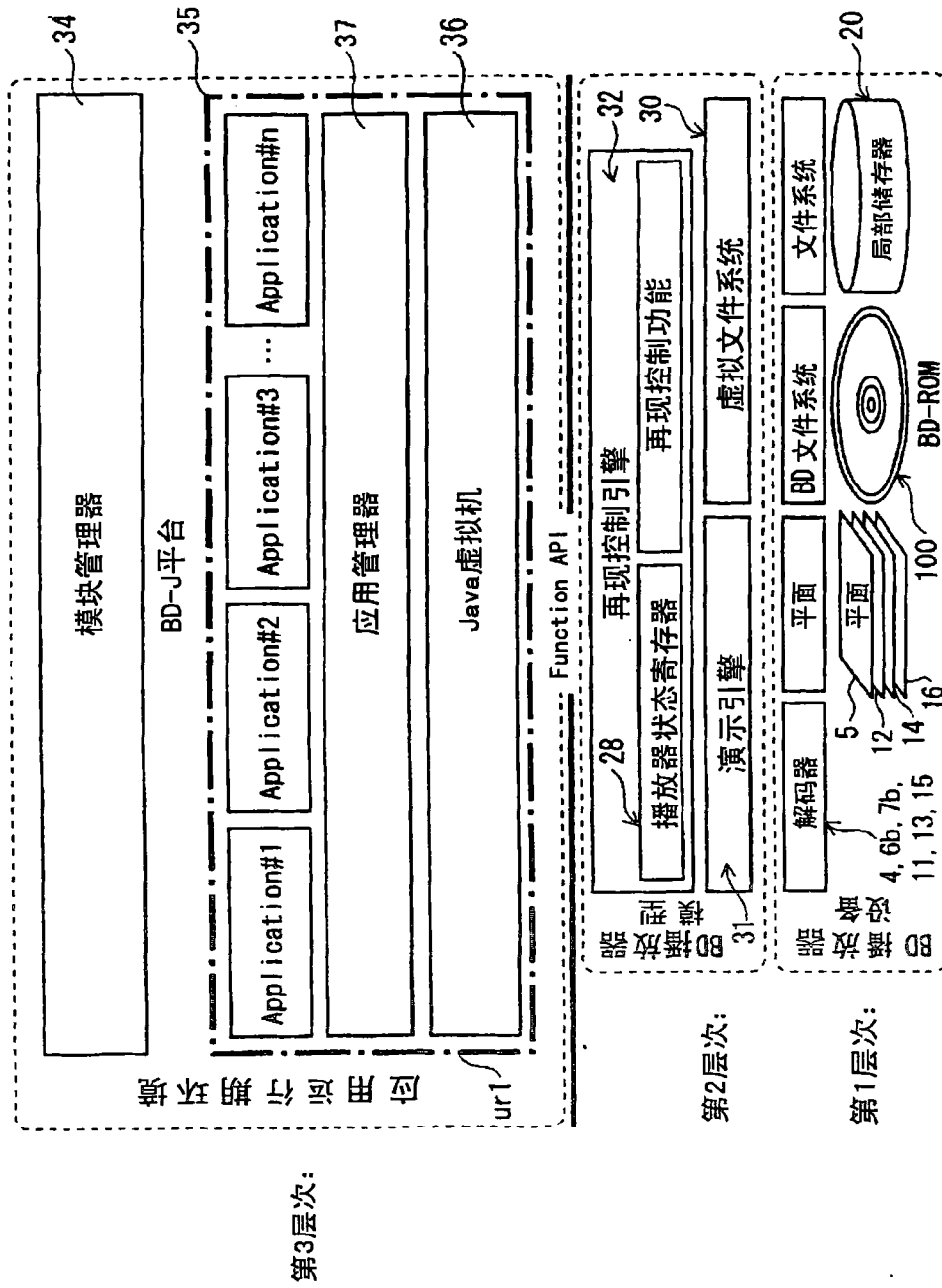


图 34

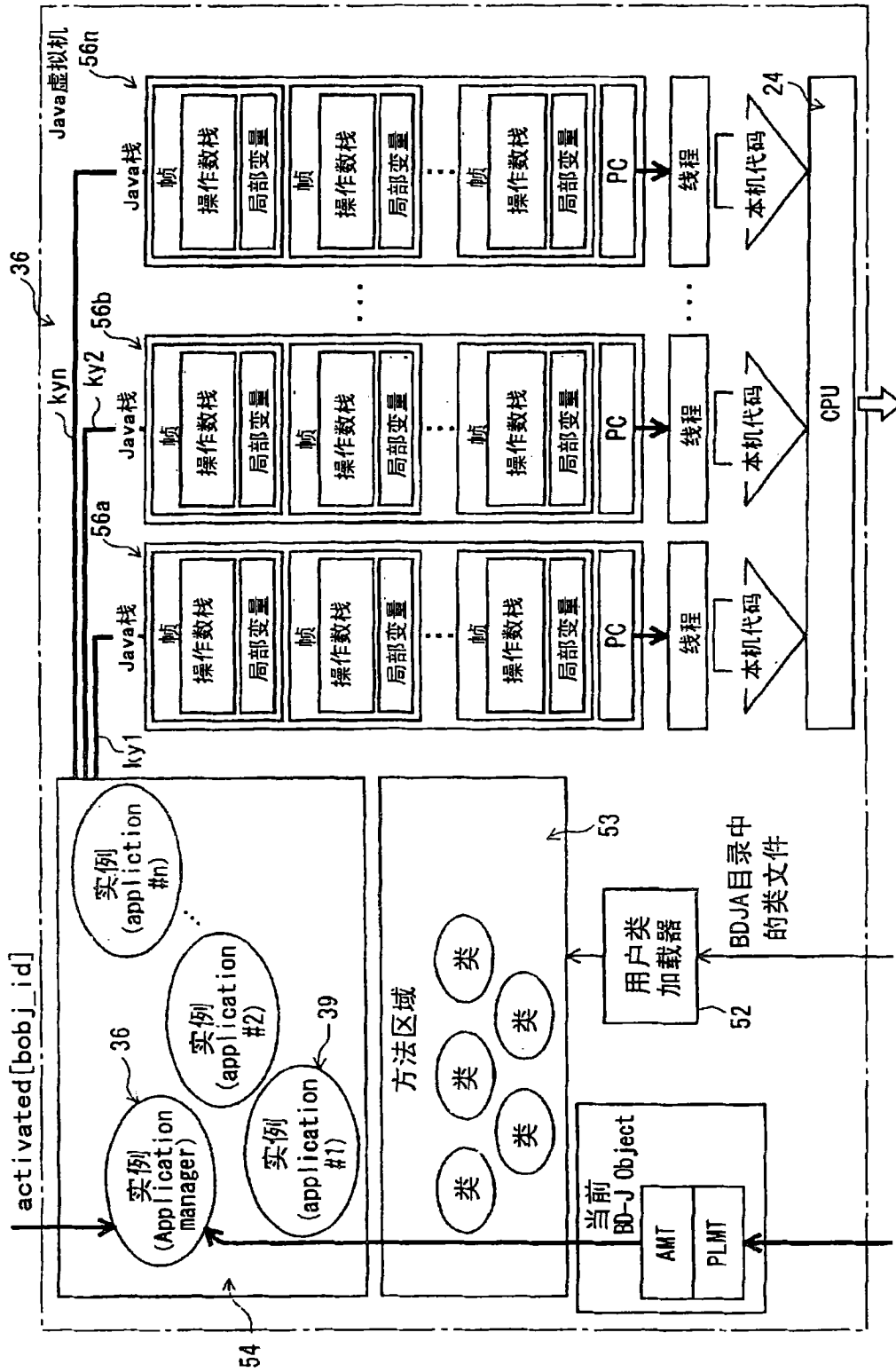


图 35

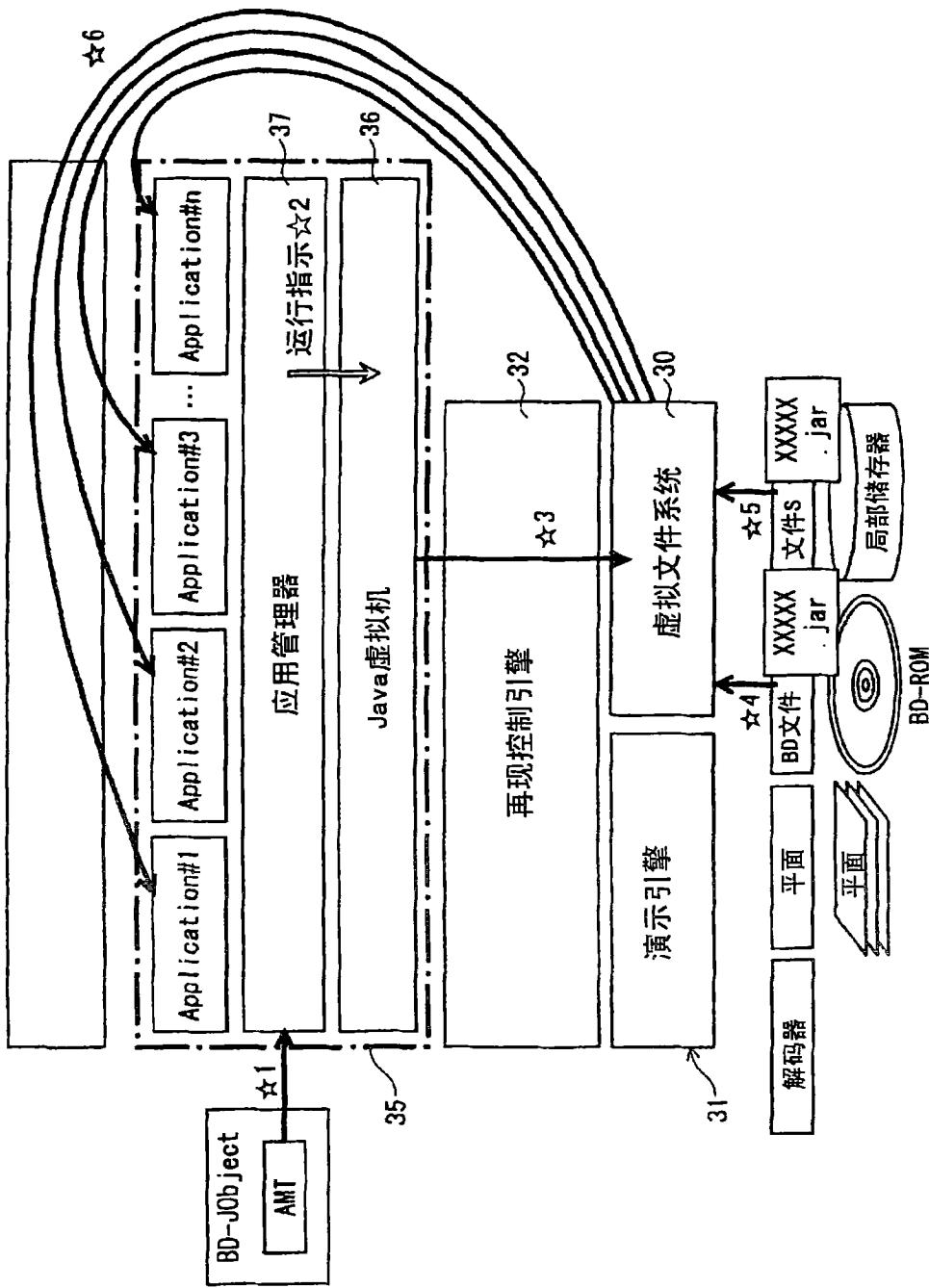


图 36

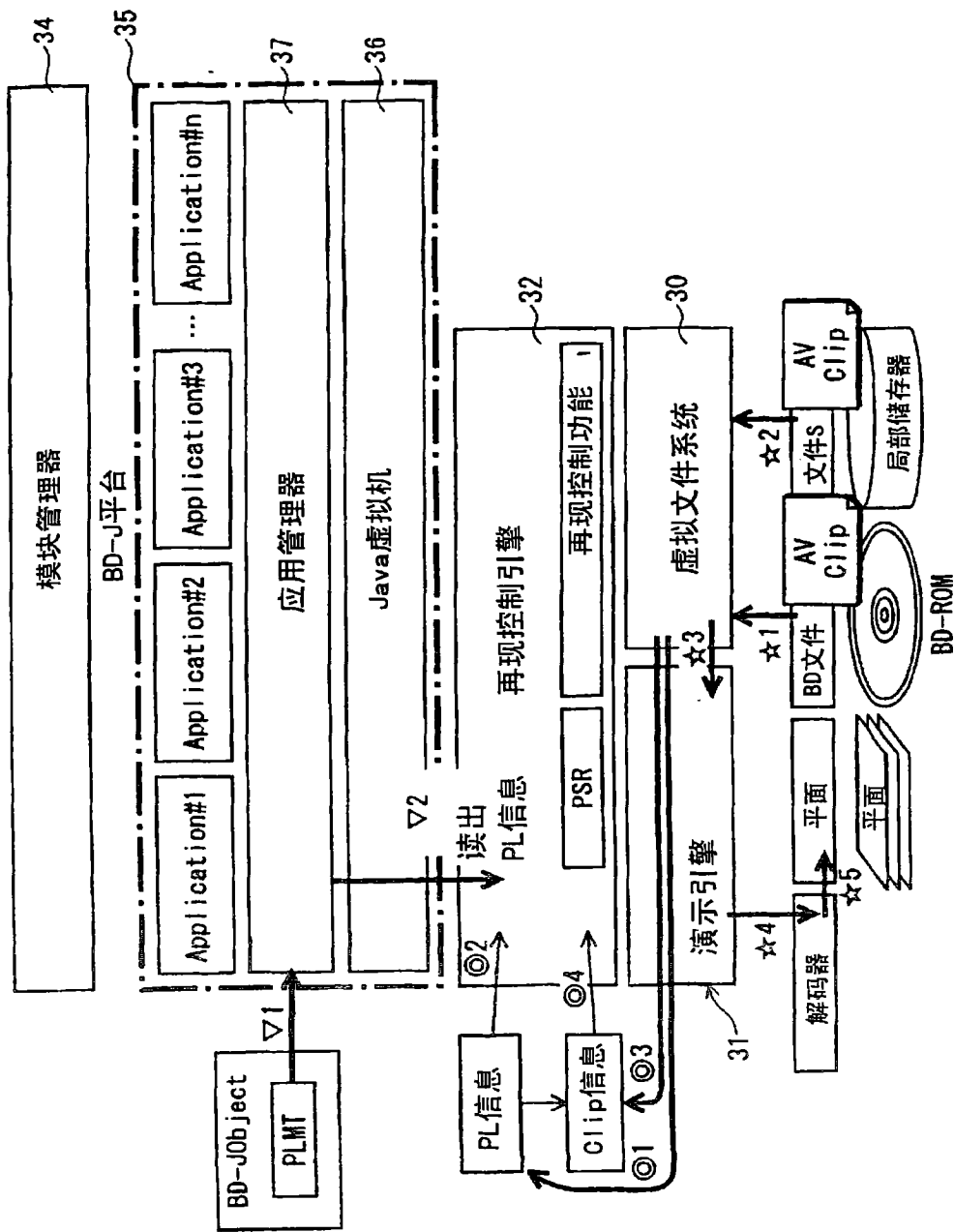


图 37

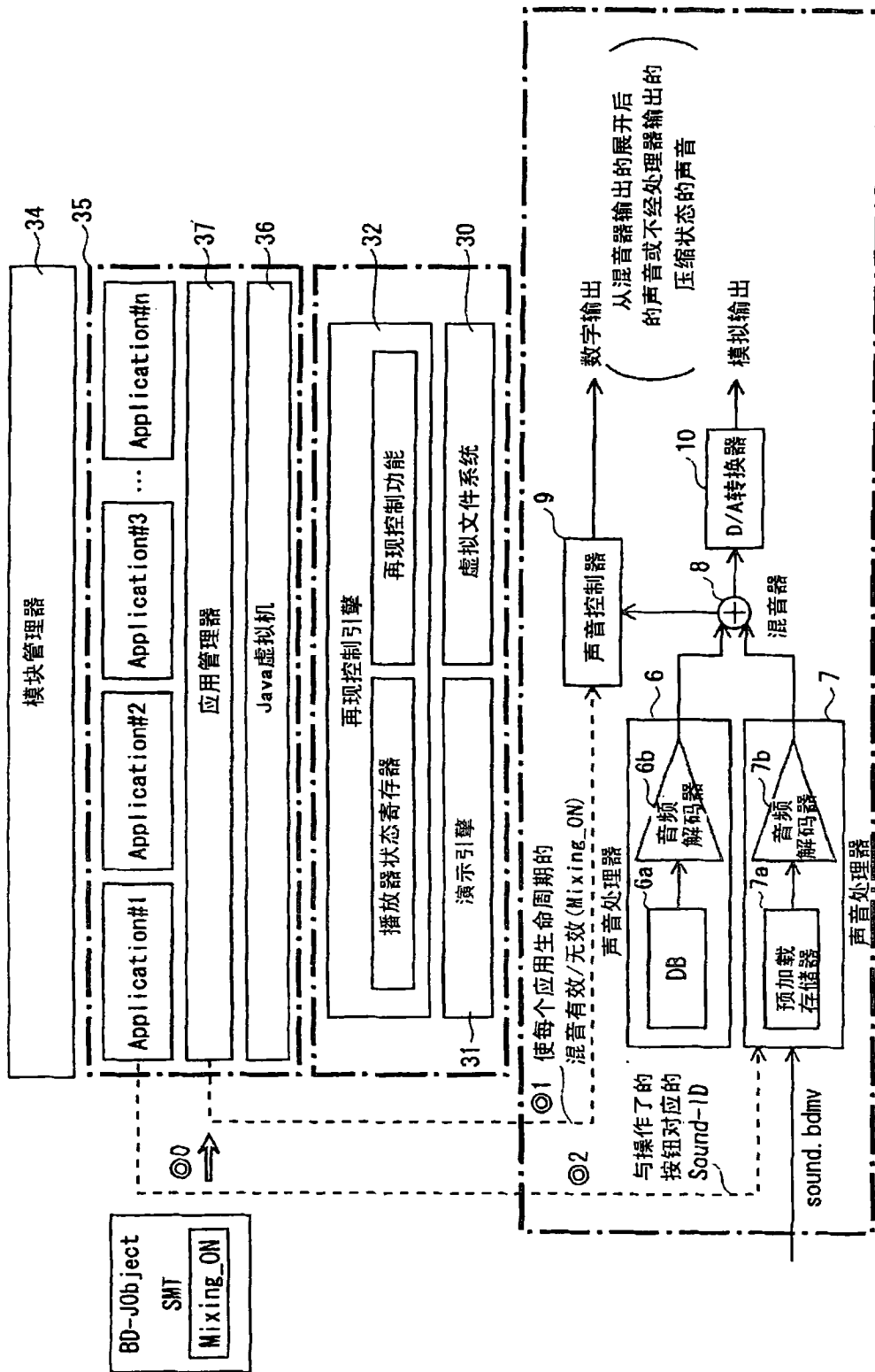


图 38

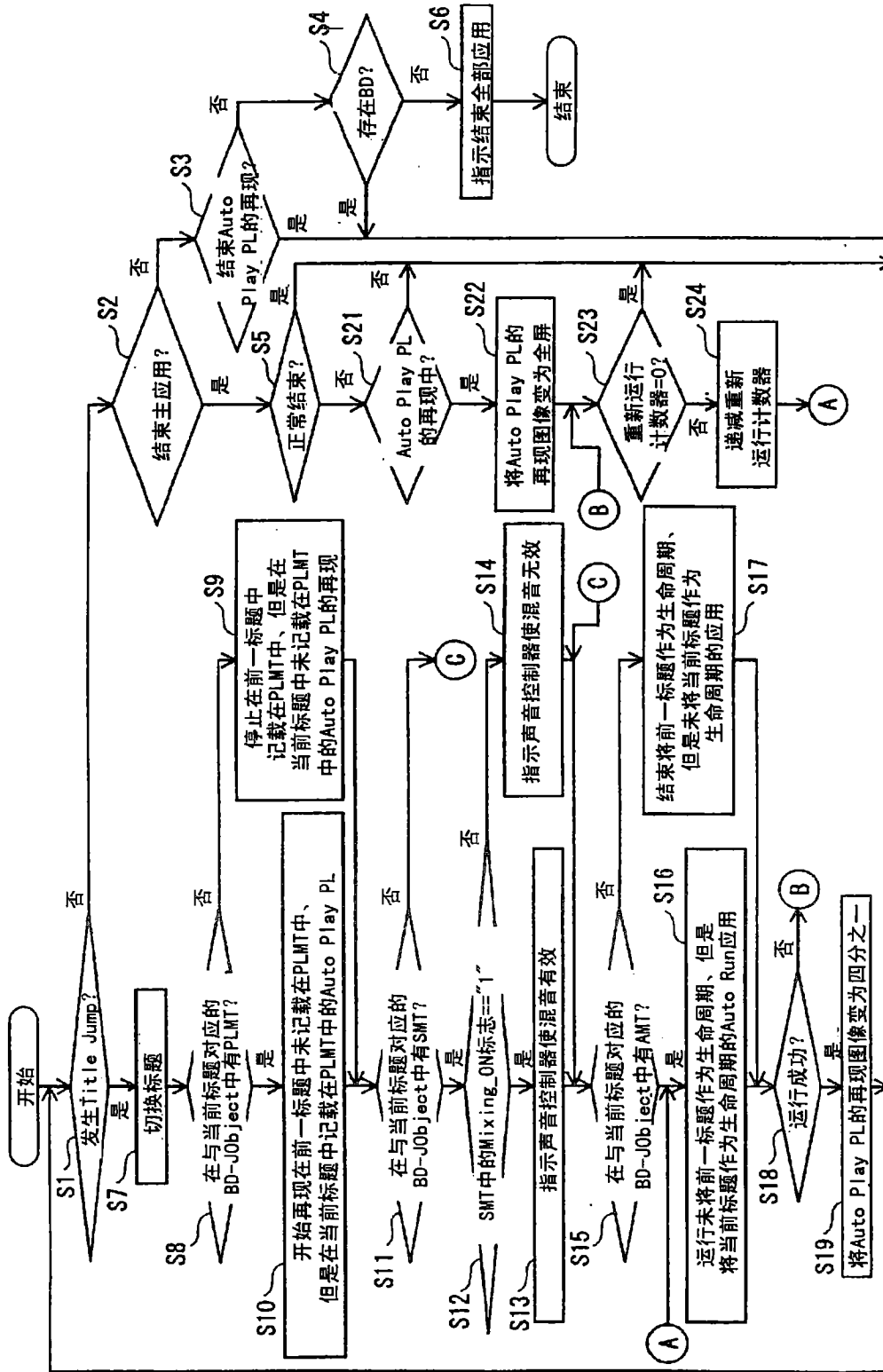


图 39

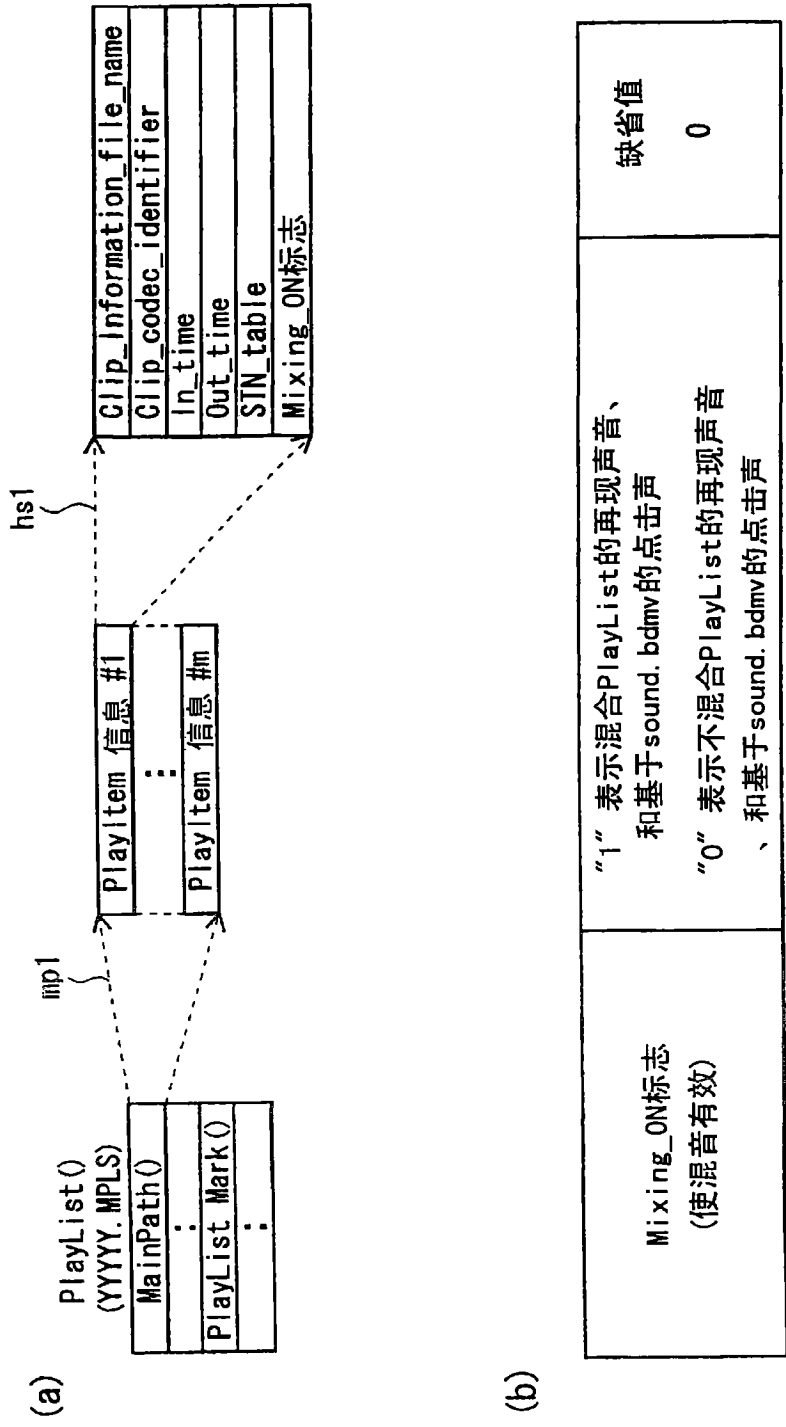


图 40

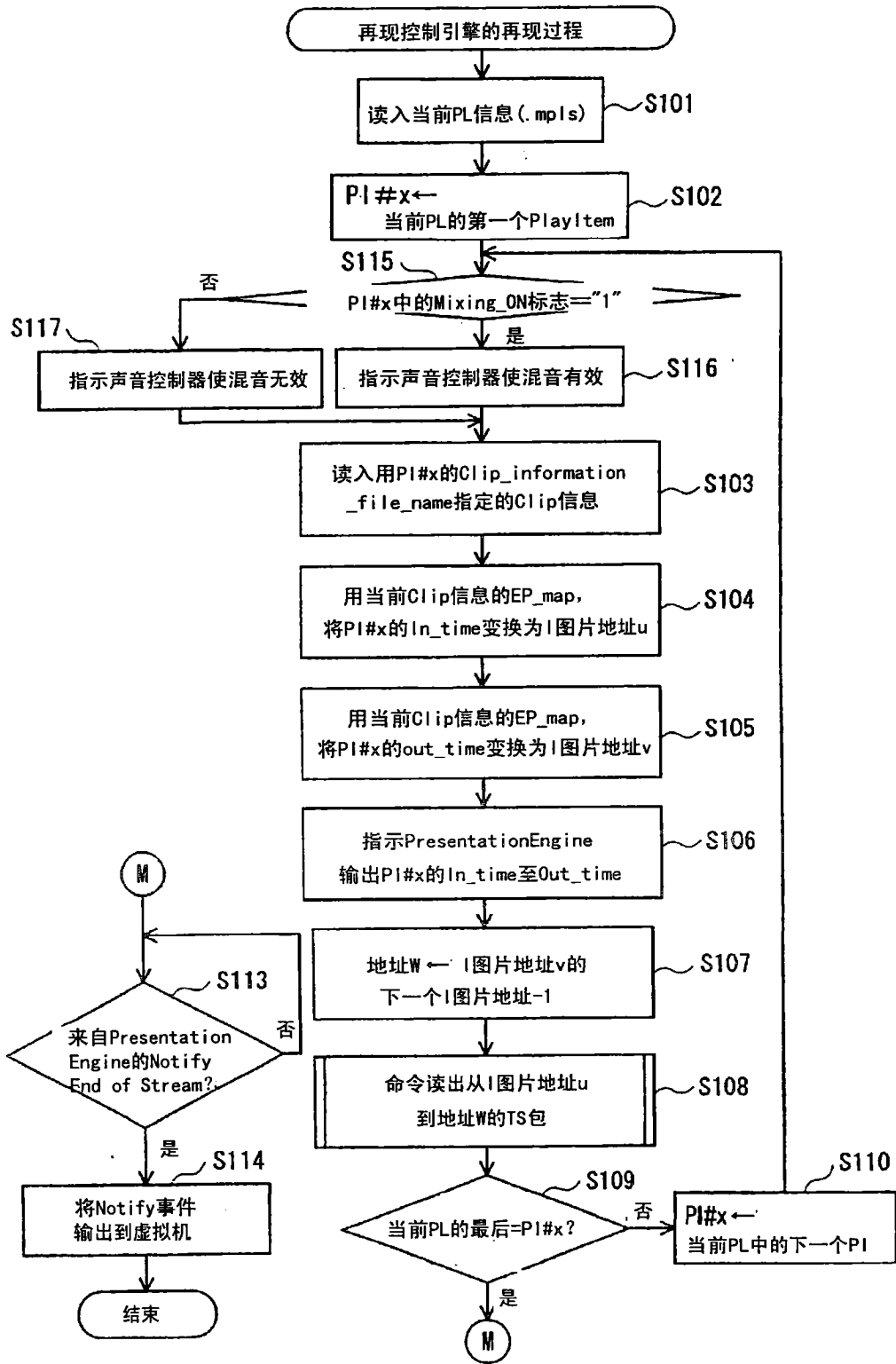


图 41

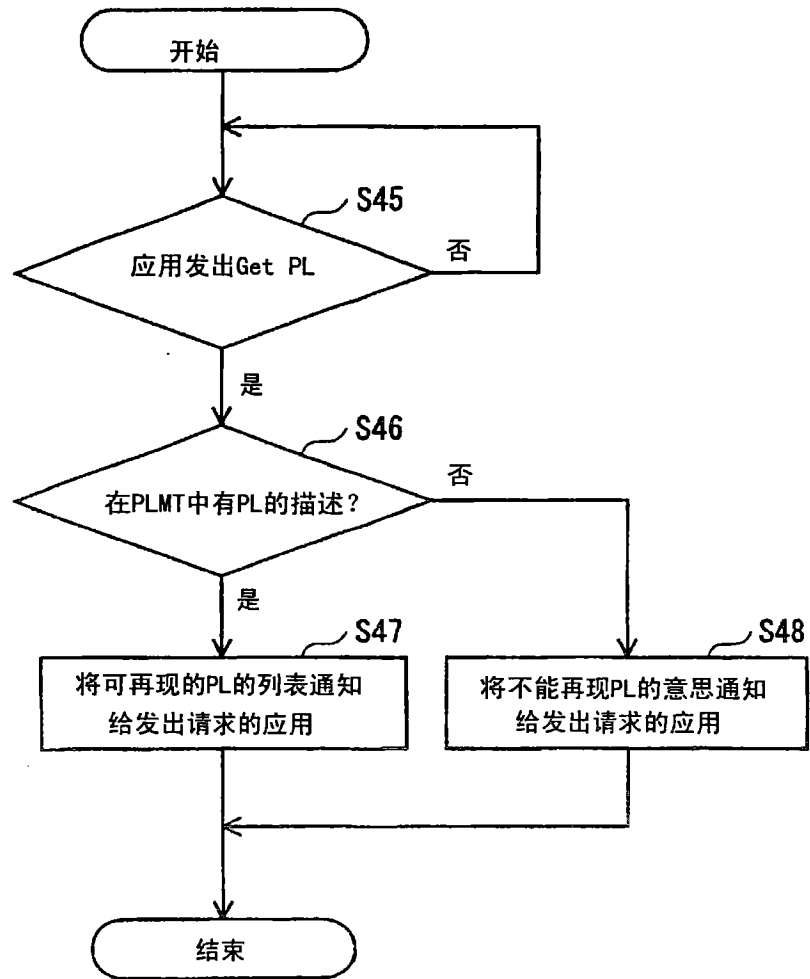


图 42

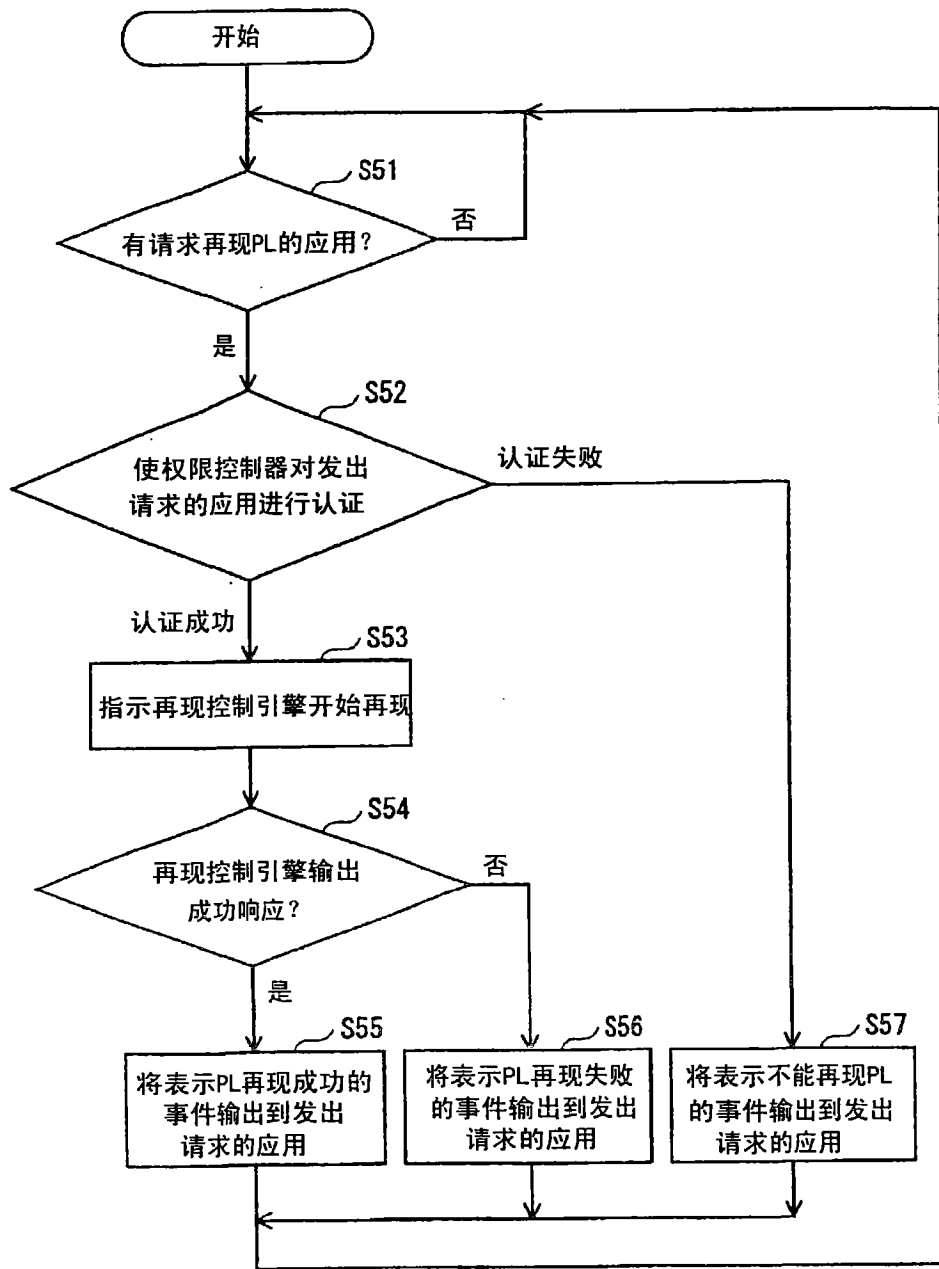


图 43

(a)

基于Parental Level的选择算法

PSR range	Play List_id_ref
$PSR(13) < 14$	Play List #1
$14 \leq PSR(13) < 18$	Play List #2
$18 \leq PSR(13)$	Play List #3

(b)

基于Language Code for Audio的选择算法

PSR range	Play List_id_ref
$PSR(16) = \text{English}$	Play List #1
$PSR(16) = \text{Japanese}$	Play List #2
$PSR(16) = \text{others}$	Play List #3

(c)

基于Player Configuration for Video的选择算法

PSR range	Play List_id_ref
$PSR(14) = 525 \times 60 \text{ TV system } 4:3 \text{ Letter Box}$	Play List #1
$PSR(14) = 525 \times 60 \text{ TV system } 16:9$	Play List #2
$PSR(14) = 1920 \times 1080 \text{ TV system}$	Play List #4

图 44

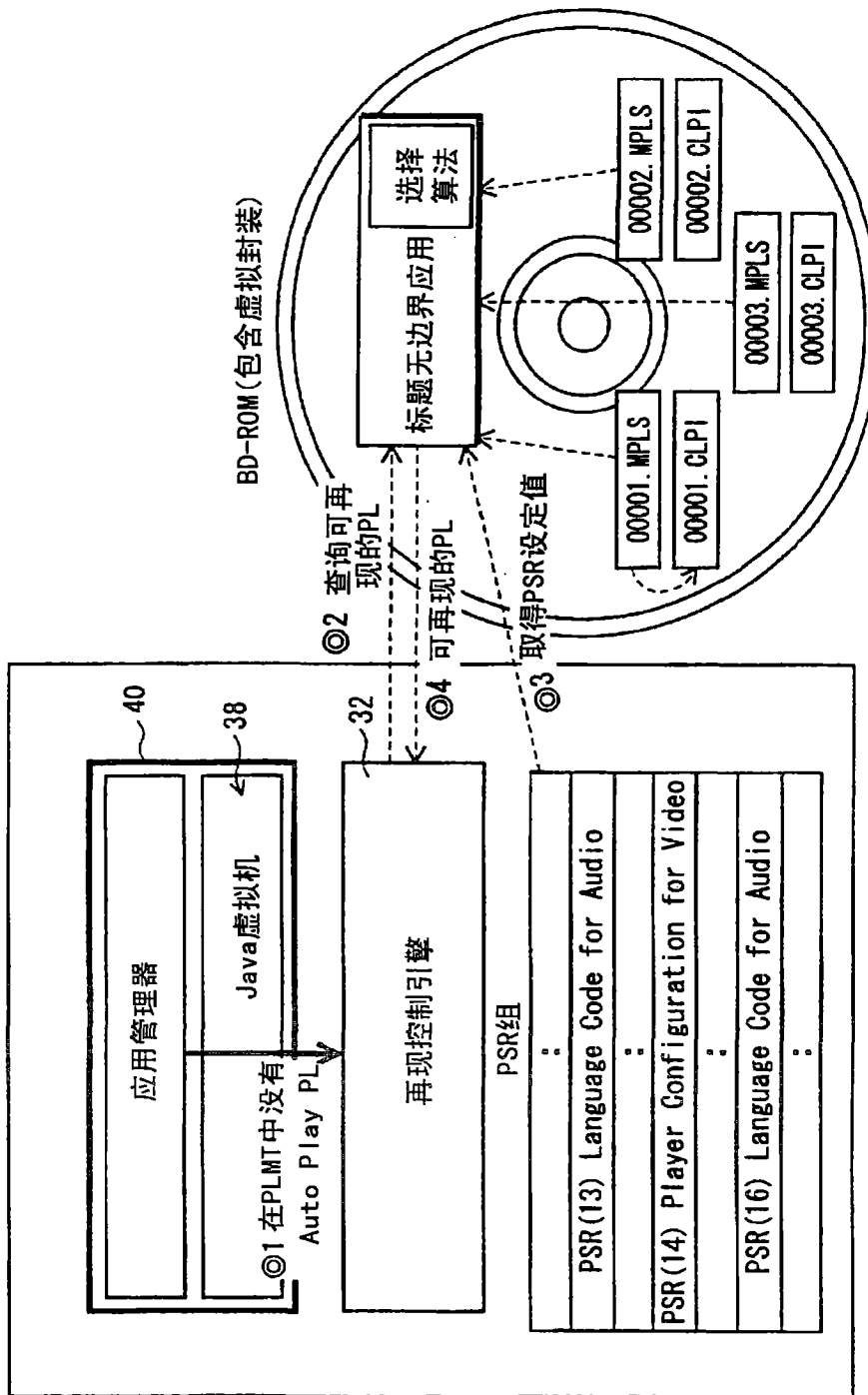


图 45