(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1993760 B
(45)授权公告日 2011.03.09

(21)申请号 200680000580.8

(22)申请日 2006.02.02

(30)优先权数据
028748/2005 2005.02.04 JP

(85)PCT申请进入国家阶段日
2007.01.31

(86)PCT申请的申请数据
PCT/JP2006/301766 2006.02.02

(87)PCT申请的公布数据
WO2006/082892 JA 2006.08.10

(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 池田航

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 胡建新

(51)Int.Cl.
G11B 27/00(2006.01)
G11B 20/10(2006.01)
G11B 20/12(2006.01)
H04N 5/85(2006.01)
H04N 5/93(2006.01)

(56)对比文件
CN 1460367 A,2003.12.03, 全文.
WO 2004025651 A1,2004.03.25, 全文.
CN 1518741 A,2004.08.04, 全文.
JP 2003249057 A,2003.09.05, 全文.

审查员 卢鹏

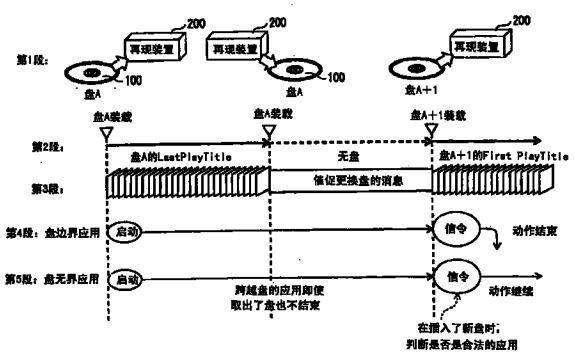
权利要求书 2 页 说明书 30 页 附图 35 页

(54)发明名称

读取装置、记录方法、读取方法

(57)摘要

应用管理器(37)在从第1盘向第2盘交换时执行“应用信令”。此时，继续执行在与第1盘中最后再现的内容相对应的管理表中显示、并且在与第2盘中最初再现的内容相对应的管理表中显示的应用；结束执行虽然在与第1盘中再现的内容相对应的管理表中显示、但在与第2盘中最初再现的内容相对应的管理表中没有显示的应用。



CN 1993760 B

1. 一种读取装置，其特征在于，

具备：

再现单元，读取并再现记录在第 1 盘中的内容；和

应用管理单元，根据与该内容对应的管理表执行应用；

上述应用管理单元：

在从第 1 盘向第 2 盘交换时，继续执行以下应用，该应用在与第 1 盘中最后再现的内容相对应的管理表中显示，并且在与第 2 盘中最初再现的内容相对应的管理表中显示，

结束执行以下应用，该应用虽然在与第 1 盘中最后再现的内容相对应的管理表中显示，但在与第 2 盘中最初再现的内容相对应的管理表中没有显示。

2. 如权利要求 1 所述的读取装置，其特征在于，

上述管理表显示对于应用的状态进行规定的启动属性。

3. 如权利要求 1 所述的读取装置，其特征在于，

在存在虽然在与第 1 盘中最后再现的内容相对应的管理表中显示、但在与第 2 盘中最初再现的内容相对应的管理表中没有显示的应用的情况下，

上述应用管理单元显示存在在取出第 1 盘之后还动作的应用的消息，并且显示对用户询问是否结束动作中的应用的菜单，以用户对该菜单进行了肯定的操作为条件，结束该应用的执行。

4. 如权利要求 3 所述的读取装置，其特征在于，

上述应用管理单元在装填了第 2 盘后显示上述询问的菜单。

5. 如权利要求 1 所述的读取装置，其特征在于，

上述读取装置具备在将第 1 盘取出时对执行中的应用的功能施加限制的功能限制单元；

上述功能限制单元在某个执行中的应用在与第 1 盘中最后再现的内容相对应的管理表中显示、并且在与第 2 盘中最初再现的内容相对应的管理表中显示的情况下，解除对该应用施加的限制。

6. 如权利要求 1 所述的读取装置，其特征在于，

上述内容包括数字流；

在没有装填第 1 盘而装填了第 2 盘的情况下，从开头再现构成上述内容的数字流；

在进行了从第 1 盘向第 2 盘的盘交换的情况下，跳过构成上述内容的数字流中的开头部分进行再现。

7. 如权利要求 1 所述的读取装置，其特征在于，

上述内容包括数字流；

在没有装填第 1 盘而直接装填了第 2 盘的情况下、与在进行了从第 1 盘向第 2 盘的盘交换的情况下，利用不同的再现路径信息将数字流再现。

8. 一种记录方法，其特征在于，

在第 1 盘和第 2 盘中，记录内容以及与该内容对应的管理表；

在与第 1 盘中最后再现的内容相对应的管理表中显示、并且在与第 2 盘中最初再现的内容相对应的管理表中显示的应用，在从第 1 盘向第 2 盘交换时继续执行；

虽然在与第 1 盘中最后再现的内容相对应的管理表中显示、但在与第 2 盘中最初再现

的内容相对应的管理表中没有显示的应用，在从第 1 盘向第 2 盘交换时结束执行。

9. 如权利要求 8 所述的记录方法，其特征在于，

上述管理表显示对于应用的状态进行规定的启动属性。

10. 一种读取方法，其特征在于，

具有 ：

再现步骤，读取并再现记录在第 1 盘中的内容 ；和

应用管理步骤，根据与该内容对应的管理表执行应用 ；

上述应用管理步骤使计算机执行 ：

在从第 1 盘向第 2 盘交换时，继续执行以下应用，该应用在与第 1 盘中最后再现的内容相对应的管理表中显示，并且在与第 2 盘中最初再现的内容相对应的管理表中显示，

结束执行以下应用，该应用虽然在与第 1 盘中最后再现的内容相对应的管理表中显示，但在与第 2 盘中最初再现的内容相对应的管理表中没有显示。

11. 如权利要求 10 所述的读取方法，其特征在于，

上述管理表显示对于应用的状态进行规定的启动属性。

读取装置、记录方法、读取方法

技术领域

[0001] 本发明属于同时执行数字化的电影作品的再现、和应用的执行的再现控制技术的技术领域。

[0002] 背景技术

[0003] 上述再现控制技术在实现将数字化的电影作品和各种应用收容在 1 个包装中销售的媒体组合时发挥很重要的作用。如果该应用是以电影作品的出场人物为角色的游戏程序，并且与数字化的电影作品的一部分同时地执行，则通过电影再现与应用执行的协同增强效应，能够进一步提高电影作品的人气。

[0004] 作为该再现控制技术的现有技术，有在以下的专利文献 1 中记载的技术。

[0005] 专利文献 1 ：日本特许第 2813245 号公报

[0006] 在近年来的电影作品中，如果某个作品较热门，则一般会制作许多其续集。并且，将这些系列的电影作品记录在多个光盘中、作为称作“DVD-BOX”的高价商品进行销售，这种做法在电影商业的领域中已作为 1 种手法被确立。这里，在如上所述在多个光盘中记录有相互关联的多个电影作品的情况下，应怎样控制应用成为问题。

[0007] 例如，如果将应用的执行限制在装填 1 个光盘的期间内，则每次交换光盘时要重复应用的结束、启动，应用的启动延迟变长，对用户操作的响应变差。

[0008] 反之，如果将应用的执行并不限于装填 1 个光盘的期间内，则在没有装填光盘的期间，应用也可以动作。这样，例如在读取装置中恶 意的程序（病毒软件、间谍软件等）入侵的情况下，在没有装填光盘的状态下，即使该恶意的程序动作，也不能区别该程序是恶意的程序、还是构成电影作品的应用。因此，不能结束该恶意的程序的动作，而任由形成被恶意的程序乘虚而入的漏洞。

[0009] 发明内容

[0010] 本发明的目的是提供一种在光盘的交换时能够消除应用的启动延迟、并且能够使与电影作品无关的应用适当地结束的读取装置。

[0011] 为了达到上述目的，有关本发明的读取装置的特征在于，具备 ：再现单元，读取并再现记录在第 1 盘中的内容 ；以及应用管理单元，根据与该内容对应的管理表执行应用 ；上述应用管理单元 ：在从第 1 盘向第 2 盘交换时，继续执行以下应用，该应用在与第 1 盘中最后再现的内容相对应的管理表中显示，并且在与第 2 盘中最初再现的内容相对应的管理表中显示 ；结束执行以下应用，该应用虽然在与第 1 盘中最后再现的内容相对应的管理表中显示，但在与第 2 盘中最初再现的内容相对应的管理表中没有显示。

[0012] 发明效果

[0013] 根据上述结构，由于在从第 1 盘向第 2 盘交换时，继续执行在与第 1 盘中最后再现的内容相对应的管理表中显示、并且在与第 2 盘中最初再现的内容相对应的管理表中显示的应用，所以，不需要在第 1 盘取出后暂时结束、在第 2 盘装填时再次启动的工夫，能够几乎消除用来进行应用的启动的启动延迟。

[0014] 由于结束执行虽然在与第 1 盘中再现的内容相对应的管理表中显示、但没有在与

第 2 盘中最初再现的内容相对应的管理表中显示的应用，所以对于没有显示在第 2 盘的管理表中的应用，在第 2 盘的装填时使动作结束。即使恶意的程序侵入，由于以第 2 盘的装填为机会而结束了该恶意的程序，所以能够实施针对恶意程序的对策。

[0015] 附图说明

[0016] 图 1 是表示有关本发明的读取装置的、关于使用行为的形态的图。

[0017] 图 2 是表示 BD-ROM 的内部结构的图。

[0018] 图 3 是示意地表示被赋予了扩展名 .m2ts 的文件是怎样构成的图。

[0019] 图 4 表示构成 AVClip 的 TS 数据包是经过怎样的过程写入到 BD-ROM 中的。

[0020] 图 5 是表示 BD-ROM 的物理单位、与构成 1 个文件范围的 Source 数据包的对应关系的图。

[0021] 图 6 是表示在 AVClip 中多路复用了怎样的基本流的图。

[0022] 图 7 是表示 Clip 信息的内部结构的图。

[0023] 图 8 是表示对电影的视频流的 EP_map 设定的图。

[0024] 图 9 是表示播放列表信息的数据结构的图。

[0025] 图 10 是表示 AVClip 和播放列表信息的关系的图。

[0026] 图 11 是表示 BD-J Object 的内部结构的图。

[0027] 图 12 是表示由档案文件收容的程序、数据的图。

[0028] 图 13(a) 是表示应用管理表的内部结构的图。

[0029] 图 13(b) 是表示构成应用管理表的信息要素的意思内容的图。

[0030] 图 14 是表示盘的状态迁移的图。

[0031] 图 15(a) 是表示 BD-ROM 整体的时间轴的图。图 15(b) 是表示 BD-ROM 整体的时间轴的结构的图。

[0032] 图 16(a)、图 16(b) 是表示在 BD-ROM 整体的时间轴上、由 BD-J Object 确定的标题再现区间的图。

[0033] 图 17 是表示在图 16(b) 的时间轴上规定的生命周期的典型情况的图。

[0034] 图 18 是表示主体标题、在线购物标题、游戏标题这 3 个标题的图。

[0035] 图 19(a)、图 19(b) 是表示应用管理表、生命周期的一例的图。

[0036] 图 20 是表示启动属性可取的三种形态 (Present、AutoRun、Suspend) 和前面标题中的应用状态的三种形态 (非启动、启动中、Suspend) 可取的组合的图。

[0037] 图 21(a) 是表示播放列表管理表的内部结构的图。

[0038] 图 21(b) 是表示构成播放列表管理表的信息要素的意思内容的图。

[0039] 图 22 是表示播放列表管理表、由播放列表管理表规定的标题的具体例。

[0040] 图 23 是表示当前标题可取的三种形态 (无播放列表管理表、有播放列表管理表且没有指定、有播放列表管理表且是 AutoPlay)、和前面标题中的 PL 的状态 (非再现状态、再现中状态) 可取的 6 种组合的图。

[0041] 图 24 是表示有关本发明的读取装置的内部结构的图。

[0042] 图 25 是将由保存在 ROM24 中的软件、与硬件构成的部分替换为层结构而描绘的图。

[0043] 图 26 是表示 Java(注册商标) 虚拟机 36 的内部结构的图。

[0044] 图 27 是示意表示再现引擎 31 ～模组管理器 33 的处理的图。

[0045] 图 28 是表示基于 BD-Jobject 的 PLMT 的、应用管理器 37 的处理的图。

[0046] 图 29 是表示盘边界应用、盘无界 (disc-unboundary) 应用的动作的图。

[0047] 图 30(a) 是表示与盘 A 中的 LastPlay 标题对应的 AMT、和与盘 A+1 中的 FirstPlay 标题对应的 AMT 的图。

[0048] 图 30(b) 是表示两个 AMT 如图 30(a) 那样规定时的信令 (Signaling) 的图。

[0049] 图 31 是表示应用管理器的处理顺序的流程图。

[0050] 图 32(a) 是催促盘交换的消息的显示例。

[0051] 图 32(b) 是由图 31 的步骤 S8 显示的菜单的一例。

[0052] 图 33 是示意地表示装载了没有预定的盘时的应用管理器 37 的处理的时序图。

[0053] 图 34(a) 表示装载了预定的盘时的处理。

[0054] 图 34(b) 表示装载了与预定的盘不同的盘、并且用户希望进行该盘的再现时的应用管理器 37 的处理。

[0055] 图 34(c) 表示装载了与预定的盘不同的盘、并且用户不希望进行该盘的再现时的应用管理器 37 的处理。

[0056] 图 35 是对比表示将盘 A+1 的 FirstPlay 标题通常再现的情况、和将盘 A+1 的 FirstPlay 标题跳转再现的情况的图。

[0057] 图 36 是表示再现控制引擎 32 所进行的播放列表再现顺序的流程图。

[0058] 标号说明

[0059] 1BD-ROM 驱动器

[0060] 2 读缓存器

[0061] 3 多路分解器

[0062] 4 视频解码器

[0063] 5 视频平面

[0064] 7 音频解码器

[0065] 11 交互图形 (Interactive Graphics) 解码器

[0066] 12 交互图形平面

[0067] 13 演示图形 (Presentation Graphics) 解码器

[0068] 14 演示图形平面

[0069] 15JPEG 解码器

[0070] 16 静止画面 (Still) 平面

[0071] 17 合成部

[0072] 18STC 生成部

[0073] 19ATC 生成部

[0074] 21 指令 ROM

[0075] 22 脚本存储器

[0076] 23PSR 组件

[0077] 24CPU

[0078] 25 通信部

[0079] 26 操作受理部

[0080] 30 虚拟文件系统

[0081] 31 再现引擎

[0082] 32 再现控制引擎

[0083] 33 模组管理器

[0084] 34HDMV 模组

[0085] 35BD-J 平台

[0086] 36Java(注册商标)虚拟机

[0087] 37 应用管理器

[0088] 38 功能限制部

具体实施方式

[0089] (第 1 实施方式)

[0090] 下面对有关本发明的读取装置的实施方式进行说明。首先,说明关于本发明涉及的读取装置的实施行为中的使用行为的形态的图。图 1 是表示有关本发明的读取装置的、关于使用行为的形态的图。在图 1 中,有关本发明的读取装置是读取装置 200。该读取装置 200 在家庭影院系统中使用,该家庭影院系统包括由多个 BD-ROM100 构成的 BD-BOX、遥控器 300、电视机 400、AV 放大器 500、扬声器 600。

[0091] 读取装置 200 是网络对应型的数字家电设备,具有将 BD-ROM100 再现的功能。

[0092] 以上是有关本发明的读取装置的使用行为的形态。

[0093] 接着说明 BD-ROM100 的内部结构。图 2 是表示 BD-ROM 的文件、目录结构的图。在该图中,在 BD-ROM 上的 Root 目录之下,有 BDMV 目录。

[0094] <BD-ROM 的概要 >

[0095] 图 2 是表示 BD-ROM 的内部结构的图。在该图的第 4 段中表示 BD-ROM,在第 3 段中表示 BD-ROM 上的轨道。该图的轨道是将从 BD-ROM 的内周到外周以螺旋状形成的轨道沿横向拉伸而描绘的。该轨道由导入区、卷区、和导出区构成。该图的卷区具有物理层、文件系统层、应用层的层模式。如果使用目录构造表现 BD-ROM 的应用层格式(应用格式),则为图中第 1 段那样。在该第 1 段中,在 BD-ROM 上的 Root 目录之下有 BDMV 目录。

[0096] 在 BDMV 目录中,有被赋予扩展名 bdmv 的文件 (index.bdmv,MoviceObject.bdmv)。并且,在该 BDMV 目录的下面还存在称作 PLAYLIST 目录、CLIPINF 目录、STREAM 目录、BDBJ 目录、BDJA 目录、AUXDATA 目录这 6 个子目录。

[0097] 在 PLAYLIST 目录中,有被赋予了扩展名 mpls 的文件 (00001.mpls)。

[0098] 在 CLIPINF 目录中,有被赋予了扩展名 clpi 的文件 (00001.clpi)。

[0099] 在 STREAM 目录中,有被赋予了扩展名 m2ts 的文件 (00001.m2ts)。

[0100] 在 BDBJ 目录中,存在被赋予了扩展名 bobj 的文件 (00001.bobj)。

[0101] 在 BDJA 目录中,有被赋予了扩展名 jar 的文件 (00001.jar)。

[0102] 根据以上的目录构造,可知在 BD-ROM 上配置有相互不同种类的多个文件。

[0103] <BD-ROM 的结构之一,AVClip>

[0104] 首先,对被赋予了扩展名 .m2ts 的文件进行说明。图 3 是示意地表示被赋予了扩

展名.m2ts的文件是怎样构成的图。被赋予了扩展名.m2ts的文件(00001.m2ts)保存有AVClip。AVClip是MPEG2-传输流形式的数字流。该数据流是通过将使胶片影像、NTSC影像、PAL影像数字化而得到的数字视频、数字音频(上第1段)变换为由PES数据包构成的基本流(上第2段)、再变换为TS数据包(上第3段)、同样将字幕类的显示图形流(Presentation Graphics(PG)流)及对话类的交互图形流(Interactive Graphics(IG)流)(下第1段、下第2段)变换为TS数据包(下第3段)、将它们多路复用而构成的。

[0105] PG流是实现伴随着动画的再现进行的字幕显示的基本流,IG流是实现伴随着动画的再现进程的GUI的基本流。

[0106] 将视频流中的由1个PTS再现的再现单位(图片等)称作“VideoPresentation Unit:视频演示”。将音频流中的由1个PTS再现的再现单位称作“Audio Presentation Unit:音频演示”。

[0107] 这里,构成AVClip的PES数据包由1个以上的“STC_Sequence”构成。所谓的“STC_Sequence”是PES数据包排列,是指在其PTS、DTS参照的(System Time Clock:系统时钟)STC的值中不存在STC不连续点(system time-base discontinuity)。由于没有STC不连续点是STC_Sequence的必要条件,所以从构成1个STC_Sequence的PES数据包串中的、位于STC不连续点的紧后面的PES数据包并包含有PCR(Program Clock Reference:节目参考时钟)的PES数据包开始、到下个STC不连续点的紧前面为止形成1个STC_Sequence。

[0108] 接着,对以上那样构成的AVClip怎样写入到BD-ROM中进行说明。图4表示构成AVClip的TS数据包经过怎样的过程写入到BD-ROM中。在该图的第1段中表示构成AVClip的TS数据包。

[0109] 构成AVClip的188字节的TS数据包如第2段所示,被赋予4 字节的TS_extra_header(图中的阴影部),成为192字节长的Source数据包。该TS_extra_header包括Arrival_Time_Stamp。

[0110] 构成AVClip的Source数据包在第3段的AVClip中构成1个以上的“ATC_Sequence”。所谓的“ATC_Sequence”是Source数据包的排列,是指在其Arrival_Time_Stamp参照的Arrival_Time_Clock中不存在不连续点(no arrival time-base discontinuity)。换言之,将在其Arrival_Time_Stamp参照的Arrival_Time_Clock中存在连续性的Source数据包串称作“ATC_Sequence”。

[0111] 该“ATC_Sequence”成为AVClip,以xxxxx.m2ts的文件名记录到BD-ROM中。

[0112] 该AVClip与通常的计算机文件同样,被分割为多个文件范围(Extents),记录到BD-ROM上的区域中。第4段示意地表示AVClip是怎样记录到BD-ROM中的。在该第4段中,构成文件的各文件范围具有预先设定的Sexetent以上的数据长。

[0113] 下面讨论将AVClip分割为多个范围而记录时的、每一个范围的最小数据长Sexetent。

[0114] 这里,在BD-ROM中光拾波器的跳转所需的时间由

[0115] Tjump = Taccess+Toverhead

[0116] 给出。

[0117] Taccess是根据跳转距离而给出的时间(m秒),

[0118] 如果跳转距离(逻辑块数)为0～5000则为179m秒,

[0119] 如果跳转距离（逻辑块数）为 5001 ～ 10,000 则为 210m 秒，

[0120] 如果跳转距离（逻辑块数）为 10,001 ～ 20,000 则为 270m 秒，

[0121] 如果跳转距离是半行程则为 990m 秒，

[0122] 如果跳转距离是全行程则为 1220m 秒。

[0123] 从 BD-ROM 读取的 TS 数据包在被保存到称作读缓存器的缓存中后被输出给解码器，而在以被称为 Rud 的位速率进行向读缓存器 的输入、设 ECC 块的扇区数为 Secc 的情况下，

[0124] Toverhead 由

[0125] Toverhead ≤ (2×Secc×8)/Rud = 20m 秒

[0126] 的计算给出。

[0127] 从 BD-ROM 读取的 TS 数据包在以 Source 数据包的状态被保存到读缓存器中后，以 TS_Recording_rate 的传送速率被供给到解码器中。

[0128] 为了不使 TS_Recording_rate 的传送速率下的向解码器的 TS 数据包供给中断，需要在 Tjump 的期间持续从读缓存器向解码器输出 TS 数据包。这里，由于来自读缓存器的输出不是以 TS 数据包、而是以 Source 数据包的状态进行的，所以在使 TS 数据包与 Source 数据包的尺寸比为 192/188 的情况下，在 Tjump 的期间，需要通过 (192/188×TS_Recording_rate) 的传送速率持续从读缓存器输出 Source 数据包。

[0129] 因而，读缓存器用来达到不发生下溢的缓存存储量为

[0130] Boccupied ≥ (Tjump/1000×8)×((192/188)×TS_Recording_rate)。

[0131] 由于向读缓存器的输入速率是 Rud、从读缓存器的输出速率是 TS_Recording_rate×(192/188)，所以对读缓存器的存储速率通过输入速率 - 输出速率的计算给出，为 (Rud-TS_Recording_rate×(192/188))。

[0132] 将该 Boccupied 存储到读缓存器中所需的时间 Tx 为

[0133] Tx = Boccupied/(Rud-TS_Recording_rate×(192/188))。

[0134] 由于为了从 BD-ROM 读取，在该时间 Tx 中需要持续以 Rud 输入 TS 数据包，所以将 AVClip 分割为多个范围记录时的、每一个范围的最小数据长 Sexetent 为：

[0135] Sexetent = Rud×Tx

[0136] = Rud×Boccupied/(Rud-TS_Recording_rate×(192/188))

[0137] ≥ Rud×(Tjump/1000×8)×(192/188)×TS_Recording_rate)

[0138] /(Rud-TS_Recording_rate×(192/188))

[0139] ≥ Rud×(Tjump/1000×8)×

[0140] TS_Recording_rate×192/(Rud×188-TS_Recording_rate×192)。

[0141] 由此，

[0142] Sexetent ≥ (Tjump/1000×8)×

[0143] (TS_Recording_rate×192/(Rud×188-TS_Recording_rate×192))。

[0144] 由于构成 AVClip 的各文件范围具有这样计算出的 Sexetent 以上的数据长的范围，即使构成 AVClip 的各文件范围在 BD-ROM 上离散地布位，也能够在再现时不中断向解码器供给 TS 数据包而连续地读取。

[0145] 图 5 是表示 BD-ROM 的物理单位、和构成 1 个文件范围的 Source 数据包的对应关

系的图。如第 2 段所示，在 BD-ROM 上形成有多个扇区。构成文件范围的 Source 数据包如第 1 段所示，每 32 个分组，写入到连续的 3 个扇区中。由 32 个 Source 数据包构成的组是 6144 字节（＝ 32×192），这与 3 个扇区尺寸 6144 字节 (2048×3) 一致。将收容在 3 个扇区中的 32 个 Source 数据包称作“Aligned Unit”，在向 BD-ROM 写入时，以 Aligned Unit 单位进行加密。

[0146] 在第 3 段中，以 32 个单位赋予扇区错误修正码，构成 ECC 块。读取装置只要以 Aligned Unit 单位访问 BD-ROM，就能够得到 32 个完结的 Source 数据包。以上是对 BD-ROM 写入 AVClip 的过程。

[0147] ＜基本流的种类＞

[0148] 图 6 是表示在 AVClip 中多路复用了怎样的基本流的图。

[0149] 如该图所示，在 AVClip 中多路复用了具有 0x1011 的 PID 的高画质视频流、具有从 0x1100 到 0x111F 的 PID 的主要音频流、具有从 0x1200 到 0x121F 的 PID 的 PG 流、具有从 0x1400 到 0x141F 的 PID 的 IG 流。构成这些基本流的数据包被赋予了与其对应的 PID，以该 PID 为线索进行多路分离。

[0150] ＜BD-ROM 的结构之二，Clip 信息＞

[0151] 接着，对被赋予了扩展名 .clpi 的文件进行说明。被赋予了扩展名 .clpi 的文件 (00001.clpi) 保存有 Clip 信息。Clip 信息是关于各个 AVClip 的管理信息。图 7 是表示 Clip 信息的内部结构的图。如该图的左侧所示，Clip 信息包括：

[0152] i) 保存有关于 AVClip 的信息的『ClipInfo()』

[0153] ii) 保存有关于 ATC Sequence、STC Sequence 的信息的『SequenceInfo()』

[0154] iii) 保存有关于 Program Sequence 的信息的『ProgramInfo()』

[0155] iv)『Characteristic Point Info(CPI())』。

[0156] Sequence Info 是包含在 AVClip 中的、关于 1 个以上的 STC-Sequence、ATC-Sequence 的信息。设置这些信息的意义是为了将 STC、ATC 的不连续点预先通知给读取装置。即，如果存在该不连续点，则有可能在 AVClip 内出现相同值的 PTS、ATS，在再现时会发生不良状况。为了表示 STC、ATC 连续到传输流中的哪里为止，而设有 Sequence Info。

[0157] 所谓的 Program Info，是表示 Program 内容为一定的区间 (ProgramSequence) 的信息。所谓的 Program，是共有用来进行同步再现的时间轴的基本流彼此的集合。设置 Program Sequence 信息的意义是为了将 Program 内容的变化点预先通知给读取装置。这里所谓的 Program 内容的变化点，是指视频流的 PID 变化、或视频流的种类从 SDTV 变化为 HDTV 的点等。

[0158] 接着，对 Characteristic Point Info 进行说明。图中的引出线 cu2 详细描述 CPI 的结构。如引出线 cu2 所示，CPI 包括 Ne 个 EP_map_for_one_stream_PID(EP_map_for_one_stream_PID(0) ～ EP_map_for_one_stream_PID(Ne-1))。这些 EP_map_for_one_stream_PID 是关于属于 AVClip 的各个基本流的 EP_map。EP_map 是表示在 1 个基本流上将 Access Unit Delimiter 存在的条目位置的数据包号 (SPN_EP_start) 与条目时刻 (PTS_EP_start) 建立对应而表示的信息。图中的引出线 cu3 详细描述 EP_map_for_one_stream_PID 的内部结构。

[0159] 据此可知，EP_map_for_one_stream_PID 包括 Nc 个 EP_High(EP_High(0) ～ EP_

High(Nc-1))、和 Nf 个 EP_Low_(EP_Low(0) ～ EP_Low(Nf-1))。这里，EP_High 具有表示 Access Unit(Non-IDR I 图片、IDR 图片）的 SPN_EP_start 及 PTS_EP_start 的高位的作用，EP_Low 具有表示 Access Unit(Non-IDR I 图片、IDR 图片）的 SPN_EP_start 及 PTS_EP_start 的低位的作用。

[0160] 图中的引出线 cu4 详细描述 EP_High 的内部结构。如该引出线所示，EP_High(i)包括作为对 EP_Low 的参照值的『ref_to_EP_Low_id[1]』、表示 Access Unit(Non-IDR I 图片、IDR 图片）的 PTS 的高位的『PTS_EP_High[i]』、和表示 Access Unit(Non-IDRI 图片、IDR 图片）的 SPN 的高位的『SPN_EP_High[i]』。这里，i 是用来识别任意 EP_High 的的标识符。

[0161] 图中的引出线详细 cu5 描述 EP_Low 的内部结构。如该引出线 cu5 所示，EP_Low 包括表示所对应的 Access Unit 是否是 IDR 图片的『is_angle_change_point(EP_Low_id)』、表示所对应的 Access Unit 的尺寸的『I_end_position_offset(EP_Low_id)』、表示所对应的 Unit(Non_IDR I 图片、IDR 图片）的 PTS 的低位的『PTS_EP_Low(EP_Low_id)』、和表示所对应的 Access Unit(Non-IDR I 图片、IDR 图片）的 SPN 的低位的『SPN_EP_Low(EP_Low_id)』。这里，所谓的 EP_Low_id 是用来识别任意的 EP_Low 的标识符。

[0162] <Clip 信息的说明之二，EP_map>

[0163] 以下，根据具体例对 EP_map 进行说明。图 8 是表示对电影的视频流的 EP_map 设定的图。第 1 段表示以显示顺序配置的多个图片（由 MPEG4-AVC 规定的 IDR 图片、I 图片、B 图片、P 图片），第 2 段表示该图片的时间轴。第 4 段表示 BD-ROM 上的 TS 数据包串，第 3 段表示 EP_map 的设定。

[0164] 在第 2 段的时间轴上，假设在时刻 t1 ～ t7 存在作为 Access Unit 的 IDR 图片及 I 图片。并且，如果设这些 t1 ～ t7 的时间间隔为 1 秒左右，则将在电影中使用的视频流的 EP_map 设定为，将 t1 ～ t7 表示为条目时刻 (PTS_EP_start)，与其对应而表示条目位置 (SPN_EP_start)。

[0165] < 播放列表信息 >

[0166] 接着，对播放列表信息进行说明。被赋予了扩展名“mpls”的文件 (00001.mpls) 是保存有 PlayList(PL) 信息的文件。

[0167] 图 9 是表示播放列表信息的数据结构的图，在该图中，如引出线 mp1 所示，播放列表信息包括定义 MainPath 的 MainPath 信息 (MainPath())、和定义章节的 PlayListMark 信息 (PlayListMark())。

[0168] <PlayList 信息的说明之一，MainPath 信息 >

[0169] 首先，对 MainPath 信息进行说明。MainPath 是对作为主影像的视频流及音频流定义的再现路径。

[0170] MainPath 如箭头 mp1 所示，由多个 PlayItem 信息 #1……#m 定义。PlayItem 信息定义构成 MainPath 的 1 个以上的逻辑再现区间。通过引出线 hs1 详细描述 PlayItem 信息的结构。如该引出线所示，PlayItem 信息包括表示再现区间的 IN 点及 Out 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』、表示是否无缝地进行该 PlayItem 与其前一个 PlayItem 的连接的『connection_condition』、唯

一地表示将该 PlayItem 作为对象的 STC_Sequence 的『ref_to_STC_id[0]』、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 中要屏蔽的用户操作是哪个的『UO_mask_table』、表示是否许可向该 PlayItem 的中途的随机访问的『PlayItem_random_access_flag』、和表示在该 PlayItem 的再现结束后是否持续进行最后的图片的静止显示的『Still_mode』、和『STN_table』。

[0171] 图 10 是表示 AVClip 与播放列表信息的关系的图。第 1 段表示播放列表信息具有的时间轴。从第 2 段到第 5 段表示由 EP_map 参照的视频流（与图 8 所示相同）。

[0172] 播放列表信息包括 PlayItem 信息 #1、#2 这两个 PlayItem 信息，根据这些 PlayItem 信息 #1、#2 的 in_time、Out_time 定义两个再现区间。如果将这些再现区间排列，则定义与 AVClip 时间轴不同的时间轴。它是在第 1 段中表示的 PlayItem 时间轴。这样，通过 PlayItem 信息的定义，能够定义与 AVClip 不同的时间轴。

[0173] 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，根据以上的 Clip 信息及播放列表信息来定义作为静态的再现单位的播放列表。以上结束了对静态脚本的说明。

[0174] 接着，对“动态脚本”进行说明。所谓的动态脚本，是动态地规定 AVClip 的再现控制的脚本数据。所谓的“动态”，是指根据读取装置中的状态变化及来自用户的键事件来知晓再现控制的内容。在 BD-ROM 中，作为该再现控制的动作环境而假设了两个模式。第 1 个是与 DVD 再现装置的动作环境很相似的动作环境，是基于指令的执行环境。第 2 个是 Java（注册商标）虚拟机的动作环境。这两个动作环境中的第 1 个称作 HDMV 模式。第 2 个称作 BD-J 模式。由于有这两个动作环境，所以动态脚本是假设为其某一个的动作环境而记述的。假设为 HDMV 模式的动态脚本称作 Movie Object。另一方面，假设为 BD-J 模式的动态脚本称作 BD-J Object。

[0175] 首先对 Movie Object 进行说明。

[0176] <Movie Object>

[0177] Movie Object 保存在图 2 所示的 MovieObject.bdmv 的文件中，包括导航指令串。

[0178] 导航指令串包括实现条件分支、读取装置的状态寄存器的设定、状态寄存器的设定值取得等的指令串。以下表示在 Movie Object 中可记述的指令。

[0179] PlayPL 指令

[0180] 格式 :PlayPL（第 1 自变量，第 2 自变量）

[0181] 第 1 自变量是播放列表的号码，可以指定要再现的播放列表。第 2 自变量可以利用包含在该播放列表中的 Playltem、或该播放列表的任意的时刻、章节、标志来指定再现开始位置。

[0182] 将通过 PlayItem 指定了时间轴上的再现开始位置的 PlayPL 函数称作 PlayPlatPlayItem()，

[0183] 将通过章节指定了时间轴上的再现开始位置的 PlayPL 函数称作 PlayPlatChapter()，

[0184] 将通过时刻信息指定了时间轴上的再现开始位置的 PlayPL 函数称作 PlayPlatSpecified Time()

[0185] JMP 指令

[0186] 格式 :JMP 自变量

[0187] JMP 指令是在中途丢弃 (discard) 当前的动态脚本并执行作为自变量的分支目的地动态脚本的分支。在 JMP 指令的形式中,有直接指定分支目的地动态脚本的直接参照的形式、和间接参照分支目的地动态脚本的间接参照的形式。

[0188] Movie Object 中的导航指令的记述由于与 DVD 中的导航指令的记述方式很相似,所以能够有效率地进行将 DVD 上的盘内容移植到 BD-ROM 上的作业。关于 Movie Object,存在在以下的国际公开公报中记载的现有技术。关于详细情况,可以参考该国际公开公报。

[0189] 国际公开公报 WO 2004/074976

[0190] 以上结束了对 Movie Object 的说明。接着对 BD-J Object 进行说明。

[0191] <BD-J Object>

[0192] BD-J Object 是在 Java(注册商标)编程环境中记述的、BD-J 模式的动态脚本,保存在 00001 ～ 00003.bobj 的文件中。

[0193] 图 11 是表示 BD-J Object 的内部结构的图。由应用管理表 (AMT)、和播放列表管理表 (PLMT) 构成。与 Movie Object 的不同点是,在 BD-J Object 中没有直接记述指令。即,在 Movie Object 中,控制顺序是通过导航指令直接记述的。相对于此,在 BD-J Object 中,通过将对 Java(注册商标)应用的指定记载在应用管理表中,来间接地规定控制顺序。通过这样的间接的规定,能够有效率地进行在多个动态脚本中使控制顺序共用化的、控制顺序的共用化。

[0194] 此外,通过记述命令进行播放列表再现的导航指令 (PlayPl 指令),来进行 Movie Object 的播放列表再现,但是可以通过将表示播放列表再现顺序的播放列表管理表组合到 BD-J Object 中,来记述 BD-JObject 的播放列表再现。

[0195] 对该 BD-J 模式中的 Java(注册商标)应用进行说明。这里 BD-J 模式所假设的 Java(注册商标)平台,是完整安装了 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 后的平台。

[0196] 经由 xlet 接口,通过应用管理器控制该 BD-J 模式中的 Java(注册商标)应用。xlet 接口具有“loaded”、“paused”、“active”、“destroyed”这 4 个状态。

[0197] 上述的 Java(注册商标)平台包括用来显示 JFIF(JPEG) 及 PNG、其他图像数据的标准 Java(注册商标)库。因此,Java(注册商标)应用可以在 HDMV 模式下实现与通过 IG 流实现的 GUI 不同的 GUI 架构。Java(注册商标)应用中的 GUI 架构包括由 GEM1.0.2 规定的 HAVi 架构,包括 GEM1.0.2 的远程控制导航单元。

[0198] 由此,Java(注册商标)应用能够实现将基于 HAVi 架构的按钮显示、文本显示、在线显示 (BBS 的内容)的显示与动态图像的显示组合后的画面显示,能够利用远程控制进行对画面显示的操作。

[0199] 对应于该 Java(注册商标)应用的实体的是,保存在图 2 中的 BDMV 目录下属的 BDJA 目录中的 Java(注册商标)档案文件 (00001.jar)。下面对 Java(注册商标)档案文件进行说明。

[0200] <Java(注册商标)档案文件 >

[0201] Java(注册商标)档案文件 (图 2 的 00001.jar) 是通过将 1 个以上的类文件、1

个以上的数据文件等汇总为 1 个而得到的文件，构成要在 BD-J 模式下动作的 Java（注册商标）应用。

[0202] 图 12 是表示由档案文件收容的程序、数据的图。该图中的程序、数据是将配置有框内所示的目录构造的多个文件通过 Java（注册商标）归档器汇总后的程序、数据。框内所示的目录构造包括 Root 目录、Java（注册商标）1、2、3 目录、Image1、2、3 目录，在 Root 目录中配置有 common.pkg，在 Java（注册商标）1、Java（注册商标）2、Java（注册商标）3 目录中配置有类文件 (00001.class ～ 00007.class)，在 Image1、Image2、Image 3 目录中配置有 00001.JPEG ～ 00003.JPEG、00001.PNG ～ 00003.PNG。Java（注册商标）档案文件是由 Java（注册商标）归档器将它们汇总而得到的。该类文件及数据在被从 BD-ROM 读取时被展开，在高速缓存上作为配置在目录中的多个文件进行处理。Java（注册商标）档案文件的文件名中的“zzzzz”这 5 位的数值表示应用的 ID(applicationID)。在该 Java（注册商标）档 案文件被读取到高速缓存中时，通过参照该文件名中的数值，能够将构成任意的 Java（注册商标）应用的程序、数据取出。

[0203] 另外，在本实施方式中，构成应用的程序、数据被汇总到 Java（注册商标）档案文件中，但也可以是 LZH 文件、zip 文件。

[0204] 以上是对 BD-J 模式中的动态脚本的说明。

[0205] <Index.bdmv>

[0206] Index.bdmv 是构成 Title 的、表示 Movie Object 或 BD-J Object 的表。所谓的 Title，是作为 Movie Object 或 BD-J Object、和由它们再现的 PlayList 的再现单位，在 BD-ROM 中作为 1 个内容进行处理。Index.bdmv 在 Title 中定义作为某个 Title 的结构要素的 MovieObject 是哪一个、或者作为某个 Title 的结构要素的 BD-J Object 是哪一个。

[0207] 关于 Index.bdmv，在以下的国际公开公报中记载了详细情况。关于详细情况可以参照该公报。

[0208] 国际公开公报 WO 2004/025651A1 公报

[0209] 下面，对于图 11 所示的各个应用管理表、播放列表管理表进行更详细地说明。

[0210] <应用管理表>

[0211] 对应用管理表 (AMT) 进行说明。所谓的应用管理表 (AMT)，是安装上述 GEM1.0.2for package media targets 中的“应用信令 (application signaling)”的表。所谓的“应用信令”，是在 GEM1.0.2 规定的 MHP(Multimedei Home Platform) 中以“服务”为生命周期进行应用的启动、执行的控制。本实施方式中的应用管理表代替该“服务”而将 BD-ROM 的“标题”作为生命周期，进行应用的启动、执行的控制。

[0212] 图 13(a) 是表示应用管理表的内部结构的图。如该图所示，应用管理表包括『Life_cycle』、『apli_id_ref』、『run_attribute』、和『run_priority』。

[0213] 图 13(b) 表示构成应用管理表的信息要素的意思内容。

[0214] 『Life_cycle』表示应用的“生命周期”。

[0215] 『apli_id_ref』通过记述对应于“应用标识符”的参照值，表示具有左述的生命周期的应用是哪个。应用标识符在 Java（注册商标）档案文件中用作为文件名赋予的 5 位的数值 zzzzz 表现。在『apli_id_ref』中记述有该 5 位的数值。

[0216] 『run_attribute』记述该生命周期中的应用的“启动属性”。在启动属性中，有

AutoRun、Present、Suspend 的种类。

[0217] 『run_priority』记述该生命周期中的应用的“启动优先级”。在 BD-J Object 中，利用这些信息控制应用的动作。

[0218] ＜生命周期＞

[0219] 对在应用管理表中规定的信息中的生命周期进行说明。

[0220] 所谓的生命周期，表示在 BD-ROM 整个时间轴上、在虚拟机的工作存储器上应用能够生存的区间。工作存储器中的“生存”，是指构成该应用的 xlet 程序被读取到 Java（注册商标）虚拟机内的工作存储器中、并能够由 Java（注册商标）虚拟机执行的状态。

[0221] 在 Java（注册商标）虚拟机中，在使应用动作的情况下，重要的是：明确地规定从时间轴的何处开始应用的服务，在时间轴的何处结束应用的服务即所谓的“服务的开始点、结束点”。规定该服务的开始点、结束点的是应用管理表中的生命周期。

[0222] 另一方面，DVD-Video 那样的以读取专用盘供给的盘是以顶部菜单标题为核心的构造。形成从该顶部菜单标题向各个著作物分支而进行再现、然后再次回到顶部菜单标题的独特的状态迁移。图 14 是表示盘的状态迁移的图。该图中的方框是 Title。所谓的 Title，是在盘特有的状态迁移中对应于 1 个“状态”的再现单位，该标题被作为 Java（注册商标）应用的生命周期进行处理。

[0223] 在 Title 中，有在 BD-ROM 的装载时最先再现的『FirstPlayTitle』、构成 Top-Menu 的『Top_menuTitle』、和其他的一般的『Title』。此外，图中的箭头 jh1、2、3、4、5、6、7、8 象征性地表示 Title 间的分支。所谓的该图所示的状态迁移，是指在 BD-ROM 装载时再现『FirstPlayTitle』、发生向『Top_menuTitle』的分支、并成为对顶部菜单的选择等待的情况。

[0224] 如果由用户进行了对顶部菜单的选择操作，则按照选择进行对应的 Title 的再现，再次回到 TopMenu Title，重复以上的处理直到进行了 BD-ROM 的弹出，这是盘特有的状态迁移。

[0225] 那么，在形成图 14 那样的状态迁移的盘中，Title 是怎样被作为生命周期规定的呢？假设在进行了 BD-ROM 的装载后，按照图 14 中箭头 jh1、2、3、4……所示的参照标号的数值顺序进行分支，将 BD-ROM 弹出。这样，能够将从 BD-ROM 被装载到被弹出的连续时间带看作一条时间轴。将该时间轴作为盘整体的时间轴。图 15(a) 是表示盘整体的时间轴的图，图 15(b) 表示该时间轴的结构。如图 15(b) 所示，盘整体的时间轴包括再现 FirstPlay Title 的区间、再现 TopMenu Title 的区间、再现 title#1 ～ #3 的区间等。如果讨论这些 Title 的再现区间是怎样规定的，则由于 Title 由唯一的 BD-J Object 构成，所以可以将 BD-J Object 为有效的期间考虑为 Title 的再现区间。同样，由于 Title 由 1 个或多个 HDMV Object 构成，所以可以将 HDMVObject 为有效的期间考虑为 Title 的再现区间。

[0226] 即，由于 FirstPlay Title、TopMenu Title 及其他 Title 都由动态脚本构成，所以可以将构成 Title 的 BD-J Object 中的、某个被作为当前 BD-J Object 激活、在读取装置内被供作解读、执行的期间定义为 Title 的再现区间。图 16(a) 是表示在 BD-ROM 整体的时间轴上、根据由标识符 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 为有效的 BD-ROM 时间 轴上的一区间考虑为 Title 的再现区间。

[0227] 这里，BD-J Object 被激活的期间的结束时间是到进行 Title 分支为止。即，在进行 Title 分支之前，将从作为执行的对象的动态脚本被作为 BD-J Object 进行处理开始到在该 BD-J Object 中发生 JumpTitle 为止的一个区间作为 Title 区间进行处理。

[0228] 接着，对 Title 区间与 PL 时间轴的关系进行说明。如上所述，在 Movie Object、BD-J Object 中，可以将播放列表再现顺序作为 1 个处理顺序进行记述。如果有播放列表再现顺序的记述，则上述的 PL 时间轴的全部或一部分归属于 Title 区间。在图 16(a) 的一例中，假设在 BD-J Object 中记述有播放列表管理表。在此情况下，如图 16(b) 所示，PL 时间轴归属于对应于 BD-J Object 的 Title 区间中。由于在该 PL 时间轴上还可以定义多个章节 (Chapter#1、#2、#3)，所以在 BD-ROM 上的时间轴上，存在 BD-ROM 整体 -Title- 播放列表 - 章节这样的域。可以利用这些域来记述应用的生命周期。另外，由于播放列表的再现是与应用执行同时进行的，所以有时在播放列表再现的中途发生 Title 分支。在此情况下，不是播放列表时间轴整体、而是只有播放列表时间轴的一部分归属于 1 个 Title 再现区间内。即，播放列表时间轴的整体、还是其一部分归属于 1 个 Title 再现区间中，这根据 Title 分支何时发生而变化。

[0229] 图 17 是表示在图 16(b) 的时间轴上规定的生命周期的典型情况的图。如该图所示，在应用中，有以 Title 为生命周期的“标题边界应用”、以 Title 内的章节为生命周期的“章节边界应用”、以 BD-ROM 整体的时间轴为生命周期的“标题无界应用”这 3 个典型情况。

[0230] 其中，标题边界应用的生命周期可以利用该标题的标识符定义。此外，章节边界应用的生命周期可以利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。

[0231] 即使平台在动作，只要 Title 或章节的生命周期结束，就能够将 资源从应用回收。由于保证了资源回收的机会，所以能够使平台的动作稳定化。

[0232] 选择在不远的将来可能实施的盘内容为题材，结合具体例对应用管理表中的生命周期记述进行说明。这里作为题材的盘内容包括构成影像主体的主体标题 (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) 这样的类型不同的 3 个标题。图 18 是表示包括主体标题、在线购物标题、游戏标题这 3 个标题的盘内容的图。在该图中的右侧记述有 Index.bdmv，在左侧记述有 3 个标题。

[0233] 右侧的虚线框表示各应用属于哪个标题的归属关系。3 个标题中的 title#1 包括 application#1、application#2、application#3 这 3 个应用。title#2 包括 application#3、application#4 这两个应用，title#3 包括 application#5。在图 21 的一例中，application#3 由 title#1、title#2 两者启动。

[0234] 如果根据图 18 的虚线所示的归属关系将各应用的生命周期图表化，则成为图 19(a) 那样。在该图中，横轴是标题再现区间，在纵轴方向上配置各应用的生命周期。这里，由于 application#1、application#2 仅归属于 title#1，所以它们的生命周期限于 title#1 内。由于 application#4 仅归属于 title#2，所以它们的生命周期限于 title#2 内。由于 application#5 仅归属于 title#3，所以它们的生命周期限于 title#3 内。由于 application#3 归属于 title#1 及 title#2，所以它们的生命周期遍及 title#1-title#2。如果根据该生命周期记述应用管理表，则 title#1、#1、#3 的应用管理表成为图 19(b) 那样。如果这样记述应用管理表，则在 title#1 的再现开始时，将 application#1、application#2、application#3 装载到工作存储器中。并且，在 title#2 的开始时，进行将 application#1、

application#2 从工作存储器中删除而仅留下 application#3 的控制。与此同样，可以 在 title#2 的再现开始时，将 application#4 装载到工作存储器中，在 title#3 的开始时，进行将 application#3、application#4 从工作存储器中删除的控制。

[0235] 进而，在 title#3 的再现中将 application#5 装载到工作存储器中，在 title#3 的再现结束时进行将 application#5 从工作存储器中删除的控制。

[0236] 即使在有标题间分支的情况下，由于只要将在分支源 - 分支目的地中生存的应用保存在工作存储器上、而将在分支源中没有、仅在分支目的地中存在的应用读入到工作存储器中就可以，所以将应用读入到工作存储器中的次数成为必要最低数。这样，通过减少读入次数，能够实现不会意识到标题的边界的应用、即无界应用。

[0237] 接着，对应用的启动属性更详细地进行说明。在启动属性中，有表示自动的启动的『AutoRun』、表示虽然不是自动启动的对象但可以置于虚拟机的工作存储器中的『Present』、虽然置于虚拟机的工作存储器中但不能进行 CPU 能力的分配的『Suspend』。

[0238] 『AutoRun』是表示与所对应的标题的分支同时地将该应用读入到工作存储器中并执行的属性。如果有从某个标题向其他标题的分支，则进行应用管理的管理主体（应用管理器）将在该分支目的地标题中生存、且启动属性设定为 AutoRun 的应用读入到虚拟机的工作存储器中并执行。由此，该应用与标题分支一起自动地启动。

[0239] 启动属性『Present』是继续属性，表示继续分支源 title 中的应用的状态。并且是表示可以执行所对应的应用的属性。在启动属性是『Present』的情况下，被赋予该启动属性的应用被许可从其他应用的调出。如果从启动中的应用有调出，则进行应用管理的管理主体（应用管理器）判断该应用的 applicationID 是否记述在应用管理表中、启动属性是否是『Present』。如果是『Present』，则将该应用装载到工作存储器中。另一方面，在该调出目标应用的 applicationID 没有记述在 应用管理表中的情况下，不将该应用装载到工作存储器中。应用的调出仅限于被赋予了该『Present』的应用。由于『Present』是在没有明示地指定启动属性的情况下赋予的默认的启动属性，所以在某个应用的启动属性是无指定『--』的情况下，意味着该应用的启动属性是该 Present。

[0240] 『Suspend』是指将程序置于虽然分配资源但不分配 CPU 能力的状态。该 Suspend 例如在游戏标题的执行中、在经由侧路径的处理的实现中是有意义的。

[0241] 图 20 是表示启动属性可取的三种形态 (Present、AutoRun、Suspend)、和前面标题中的应用状态的三种形态（非启动、启动中、Suspend) 可取的组合的图。在前面状态为“非启动”的情况下，如果启动属性是“AutoRun”，则在分支目的地标题中启动该应用。

[0242] 如果前面状态为“非启动”、启动属性是“Present”、“Suspend”，则在分支目的地标题中该应用什么也不做而继续原状态。

[0243] 如果前面状态为“启动中”、启动属性是“Present”、“AutoRun”，则在分支目的地标题中该应用什么也不做而继续原状态。

[0244] 如果启动属性是“Suspend”，则使应用的状态成为 Suspend。在前面状态是“Suspend”的情况下，如果分支目的地标题的启动属性是“Suspend”，则维持 Suspend。如果是“Present”或“AutoRun”，则在分支目的地标题中再继续该应用。通过在应用管理表中定义生命周期及启动属性，能够进行随着标题再现区间的进行而使 Java（注册商标）应用动作的同步控制，能够创造伴随着影像再现、程序执行的各种应用。

[0245] 另外，在前面状态是“Suspend”、分支目的地标题的启动属性是“Present”的情况下，也可以维持前面状态即中止状态。

[0246] 最后，说明对应于各应用的“启动优先级”。

[0247] 该启动优先级取 0 ～ 255 的值，在存储器资源枯竭、或 CPU 负荷 变高时，成为应用管理器进行强制地使哪个程序结束、还有从哪个应用夺取资源的处理时的判断材料。在此情况下，应用管理器进行下述处理 :将启动优先级较低的应用的动作结束，使启动优先级较高的应用的动作继续。

[0248] 此外，在对再现中播放列表的请求有竞争时的应用间的协调中也使用启动优先级。假设这里的应用是某个播放列表的快进。这里，如果其他应用进行了对相同的播放列表的暂停请求，则比较赋予给这些应用的启动优先级。并且，如果命令快进的应用的启动优先级较高，则继续进行该应用的快进。反之，如果命令暂停的应用的启动优先级较高，则进行快进中的播放列表的暂停。

[0249] 在创作时能够根据以上的生命周期、启动属性、启动优先级进行规定，以将能够在虚拟机上动作的应用的数量限制在规定数量以下。因此，能够保证应用的稳定动作。

[0250] < 播放列表管理表 >

[0251] 以上是对应用管理表的说明。接着对播放列表管理表 (PLMT) 进行说明。所谓的播放列表管理表，是表示在应用的生命周期中要与各应用执行同时进行的再现控制的表。应用的动作是不稳定的，有可能发生启动的失败或异常结束。所以，作为在有启动失败、异常结束时的故障保险单元，在本实施方式中在每个应用的生命周期中设置播放列表管理表。播放列表管理表是规定在某个应用的生命周期开始时要与其同时进行的再现控制的信息。该再现控制，是基于播放列表信息的 AVclip 再现，通过同时进行播放列表信息的再现控制，同时地进行应用执行和播放列表再现。

[0252] 图 21(a) 是表示播放列表管理表的内部结构的图。如该图所示，播放列表管理表包括『PL_id_ref』、和『Playback_Attribute』。

[0253] 图 21(b) 表示构成播放列表管理表的信息要素的意思内容。

[0254] 『PL_id_ref』通过记述对播放列表标识符的“参照值”，表示在 应用的生命周期中能够再现的播放列表是哪个。播放列表标识符在文件 YYYYY.MPLS 中以作为文件名被赋予的 5 位的数值 YYYYY 表现。通过记述该 YYYYY，『PL_id_ref』表示在所对应的 Title 中能够再现的播放列表是哪个。

[0255] 『Playback_Attribute』是仿照应用管理表中的启动属性的属性。是规定将『PL_id_ref』中记述的播放列表在标题开始时怎样再现的再现属性。在对播放列表的再现属性中，有『AutoPlay』、『Present』的种类。

[0256] 『AutoPlay』是表示与所对应的标题的分支同时地再现该播放列表的属性。如果有从某个标题向其他标题的分支，则进行应用管理的管理主体（应用管理器）开始再现能够在该分支目的地标题中再现、并且再现属性设定为 AutoPlay 的播放列表。由此，启动属性被设定为 AutoPlay 的播放列表随着标题分支而自动地启动。

[0257] 『Present』与启动属性中的 Present 同样，是继续属性，表示继续分支源 title 中的播放列表的状态。并且是表示可以再现所对应的播放列表的属性。例如，假设有连续再现的两个 Title，在前面的标题侧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，在当前标题侧的播放列表管理表中其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为 2 小时长，在其中经过了 1 小时的时刻发生分支。在此情况下，在当前标题中，由于再现属性被设定为 Present，所以在当前标题中，在 1 小时的已再现区间后马上开始再现其播放列表。如果这样将再现属性设定为 Present，则即使在有 Title 间的分支的情况下，也能够从其剩余的部分开始播放列表再现。由此，在相互分支的一系列的 Title 中，能够容易地实现再现共用的播放列表的“标题间的播放列表再现的共用化”。此外，在分支目的地标题有多个的情况下，如果将这些多个标题的再现属性都设为 Present，则不论分支到多个中的哪个，都能够继续 1 个共用的 播放列表再现。

[0258] 另外，Title 的边界也可以不保证无缝再现，所以在如上述那样要在多个 Title 间再现 1 个播放列表的情况下，容许在分支前后中断播放列表再现。

[0259] 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而再现。如果从启动中的应用有播放列表的再现请求，则进行应用管理的管理主体（应用管理器）判断接受到请求的播放列表的 PL_id_ref 是否记述在播放列表管理表中、再现属性是否是『AutoPlay』或『Present』的任一个。如果是『AutoPlay』或『Present』的任一个，则再现该播放列表。另一方面，在接受到请求的播放列表的 PL_id_ref 没有记述在播放列表管理表中的情况下，不再现该播放列表。通过应用的请求进行的播放列表再现仅限于被赋予了该『AutoPlay』或『Present』的任一个的播放列表。『Present』由于是在没有明示地指定再现属性的情况下赋予的默认的再现属性，所以如果某个播放列表的再现属性是无指定『--』，则意味着该播放列表的再现属性是 Present。

[0260] 图 22 表示播放列表管理表、由播放列表管理表规定的标题的具体例。图 22 的第 1 段表示 Title 的再现影像，第 2 段表示 Title 的时间轴。第 3 段表示由 PLMT 规定再现的播放列表，第 4 段表示应用执行。在第 4 段中，application#1 与 Title 的开始一起启动，然后，在时刻 t1 成为动作状态。另一方面，PlayList#1 与 Title 的开始一起开始再现。由于 PlayList#1 的再现在与 Title 的开始相同的时刻开始，所以如第 1 段的左侧所示，在从 Title 的再现开始后到应用成为动作状态为止的启动延迟中，全屏显示播放列表的再现图像 gj1。通过将播放列表管理表的再现属性设定为“AutoPlay”，在 Java（注册商标）应用成为动作状态之前，即使花费 5 ～ 10 秒的时间，在此期间也成为“暂且描绘些什么的状态”。通过该“暂且描绘些什么的状态”，能够 弥补标题执行开始时的启动延迟。

[0261] 另一方面，由于 application#1 在时刻 t1 成为动作状态，所以在时刻 t1 显示以播放列表再现图像为子画面、以应用的执行图像为母画面的合成图像 gj2。应用的执行图像是配置有开始按钮、继续按钮、能量指示器的游戏用 GUI 架构，通过执行 Java（注册商标）应用来进行该 GUI 架构的描绘处理。

[0262] PLMT 的特征在于能够构成这样的形成组合了播放列表的再现图像和 Java（注册商标）应用的 GUI 架构的再现图像的标题。

[0263] 图 23 是表示当前标题可取的三种形态（无播放列表管理表 (i)、有播放列表管理表且 AutoPlay(ii)、有播放列表管理表且无指定

[0264] (iii))、和前面标题中的播放列表的状态（非再现状态、再现中状态）可取的 6 种组合的图。

[0265] 在该图中的 6 种组合中的“前面状态＝非再现状态”、和“当前标题＝有播放列表

管理表、且当前标题的再现属性＝ AutoPlay”的组合中，自动地开始分支目的地标题中的播放列表的再现。

[0266] 此外，在“前面状态＝再现中状态”、和“当前标题＝无播放列表管理表”的组合中，自动地停止分支目的地标题中的播放列表的再现。

[0267] 并且这两个组合以外都继续前面的标题的状态。基于播放列表管理表的播放列表再现的开始由于仅限于在分支源标题中是非再现状态、并且在分支目的地标题中被赋予了 AutoPlay 属性的情况，所以不需要每当发生了标题分支时都开始播放列表再现。即使发生了多个标题间的分支，也能够使开始播放列表再现的次数成为必要最低数。

[0268] 以上是对记录媒体的说明。接着对有关本发明的读取装置进行说明。

[0269] 图 24 是表示有关本发明的读取装置的内部结构的图。有关本发明的读取装置是根据该图所示的内部在工业上生产出来的。有关本发 明的读取装置主要包括系统 LSI、和驱动装置这样的部件，能够通过将这些部件安装到装置的机壳及基板上来工业生产。系统 LSI 是集成了发挥读取装置的功能的各种处理部的集成电路。这样生产的读取装置包括 BD-ROM 驱动器 1、读缓存器 2、多路分解器 3、视频解码器 4、视频平面 5、缓存 6、音频解码器 7、交互图形解码器 11、交互图形平面 12、演示图形解码器 13、演示图形平面 14、JPEG 解码器 15、静止画面平面 16、合成部 17、STC 生成部 18、ATC 生成部 19、本地存储器 20、指令 ROM21、脚本存储器 22、PSR 组件 23、CPU24、通信部 25、操作受理部 26。

[0270] BD-ROM 驱动器 1 进行 BD-ROM 的装载 / 弹出，执行对 BD-ROM 盘的访问。

[0271] 读缓存器 2 是 FIFO 存储器，以先入先出式保存从 BD-ROM 读取的 TS 数据包。

[0272] 多路分解器 3 从读缓存器 2 取出 Source 数据包，将构成该 Source 数据包的 TS 数据包变换为 PES 数据包。并且，将通过变换得到的 PES 数据包中的记载在 STN_Table 中的具有 PID 的 PES 数据包输出给视频解码器 4、音频解码器 7、交互图形解码器 11、演示图形解码器 13 的任一个。

[0273] 视频解码器 4 将从多路分解器 3 输出的多个 PES 数据包解码，得到非压缩形式的图片，写入到视频平面 5 中。

[0274] 视频平面 5 是用来保存非压缩形式的图片的平面。所谓的平面，是用来在读取装置中保存一画面量的像素数据的存储器区域。视频平面 5 的析像度为 1920×1080，保存在该视频平面 5 中的图片数据由以 16 位的 YUV 值表现的像素数据构成。在视频平面 5 中，能够将视频流中的每一帧的再现影像缩放。所谓的缩放，是指使每一帧的再现图像变化为视频平面 5 整体的 1/4（称作 quarter）、1/1（称作全标度）的任一种。由于在 BD-J 模式中按照来自 CPU24 的指示执行该缩 放，所以能够进行使视频流的再现图像局限于画面的角落、或者整面地呈现的画面播演。

[0275] 缓存 6 将从多路分解器 3 输出的 TS 数据包以先入先出式保存，并供给到音频解码器 7。

[0276] 音频解码器 7 进行对主要音频流的解码处理。

[0277] 交互图形 (IG) 解码器 11 将从 BD-ROM 或本地存储器 20 读取的 IG 流解码，将非压缩图形写入到交互图形平面 12 中。

[0278] 交互图形 (IG) 平面 12 在 HDMV 模式中被写入通过 IG 解码器 11 的解码得到的非压缩图形。此外，在 BD-J 模式中被写入通过应用描绘的文字及图形。

[0279] 演示图形 (PG) 解码器 13 将从 BD-ROM 或本地存储器 20 读取的 PG 流解码，将非压缩图形写入到演示图形平面 11 中。通过 PG 解码器 13 的解码，在画面上出现字幕。

[0280] 演示图形 (PG) 平面 14 是具有一画面量的区域的存储器，能够保存一画面量的非压缩图形。

[0281] JPEG 解码器 15 将记录在 BD-ROM 或本地存储器 20 中的 JPEG 数据解码，写入到静止画面平面 16 中。

[0282] 静止画面平面 16 是保存通过将 JPEG 数据展开而得到的非压缩的图形数据的平面。该图形数据作为 Java（注册商标）应用所描绘的 GUI 架构的所谓的“壁纸”使用。

[0283] 合成部 17 得到将交互图形平面 12 的保存内容、演示图形平面 14 的保存内容、视频平面 5 的保存内容、和静止画面平面 16 的保存内容合成后的合成图像。

[0284] STC 生成部 18 生成系统时钟 (STC)。并且，在 STC_Sequence 的切换时刻，通过在此前的 STC_Sepuence 的 STC 值 (STC1) 中加上称作 STC_delta 的偏移值，求出新的 STC_Sepuence 的 STC 值 (STC2)，在设此前的 STC_Sequence 的 STC 值为 STC1，设新的 STC_Sequence 中最后再现的图片的显示开始时刻为 PRS1(1stEND)、设图片的显示期间为 Tpp、设在后续 STC_Sequence 中最初显示的图片的开始时刻为 PTS2(2ndSTART) 的情况下，STC_delta 表现为，

[0285] STC_delta = PTS1(1stEND)+Tpp-PTS2(2ndSTART)。以上这样求出 STC_delta，将加上它后的时钟的计数值输出给各解码器。由此，各解码器能够将对应于两个 STC_Sequence 的流不间断地再现。由此，即使在 1 个 AVClip 中存在两个以上的 STC_Sequence，或者即使要连续再现的两个以上的 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。

[0286] ATC 生成部 19 生成到达时钟 (ATC :Arrival Time Clock)。并且，在 STC_Sequence 的切换时刻，通过在此前 ATC_Sequence 的 ATC 值 (ATC1) 中加上称作 ATC_dalta 的偏移值，形成将此前的 ATC_Sequebce 的 ATC 值 (ATC1) 与新的 ATC_Sequence 的 ATC 值 (ATC2) 连续的值。通过该加法，成为 ATC2 = ATC1+ATC_delta。所谓的 ATC_delta，是指从此前读取的传输流 (TS1) 的最后的 TS 数据包的输入时刻 T1 到新读取的传输流 (TS2) 的最初的 TS 数据包的输入时刻 T2 的偏移值，通过“ATC_delta ≥ N1/TS_recording_rate”的计算式给出。这里，输入时刻 T2 是指将 TS2 的最初的数据 TS 数据包的输入时刻投影到 TS1 的时间轴上的时刻。此外，N1 是后续于 TS1 的最后的视频 PES 数据包的 TS 数据包的包数。在 BD-ROM 中，该 ATC_delta 由于记述在 Clip 信息中，所以通过使用它能够计算 ATC_delta。通过以上的计算，能够使此前的 ATC_Sequence 所具有的 ATC 值 (ATC1)、和新的 ATC_Sequence 所具有的 ATC 值 (ATC2) 成为连续的值。通过将加上 ATC_delta 后的时钟的计数值输出给多路分解器 (De-MUX)3，能够实现无缝的缓存控制。

[0287] 此外，为了满足缓存的连续性，只要满足以下的 1)、2) 就可以。

[0288] 1) 满足 STC2(2ndSTART) > STC2(1stEND)

[0289] 这里，STC2(1stEND) 是将 STC1(1stEND) 投影到 STC2 的时间轴上的值，通过 STC2(1stEND) = STC1(1stEND)-STC_delta 的计算式给出。

[0290] 2) 通过投影在相同的时间轴上的 STC1 和 STC2 定义 TS 数据包从 TS1 的取出、和 TS 数据包从 TS2 的取出，不会带来缓存的下溢或溢出。

[0291] 本地存储器 20 是用来将从 web 站点下载的内容等、从 BD-ROM 以外的记录媒体、通信媒体供给的内容与元数据一起保存的硬盘。该元数据是用来将下载内容结合到本地存储器 20 中来进行管理的信息，通过访问该本地存储器 20，BD-J 模式的应用能够进行利用下载内容长度的各种处理。

[0292] 指令 ROM21 存储有规定读取装置的控制的软件。

[0293] 脚本存储器 22 是用来保存当前的 PL 信息及当前的 Clip 信息的存储器，所谓的当前 PL 信息，是指记录在 BD-ROM 中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。所谓的当前 Clip 信息，是指记录在 BD-ROM 中的多个 Clip 信息中的、作为当前处理对象的 Clip 信息。

[0294] PSR 组件 23 是内置在读取装置中的寄存器，包括 64 个播放器状态 / 设置寄存器 (PSR)、和 4096 个通用寄存器 (GPR)。播放器状态 / 设置寄存器的设定值 (PSR) 中的 PSR4 ～ PSR8 用于表现当前的再现时刻。

[0295] PSR4 通过设定为 1 ～ 100 的值，表示当前的再现时刻所属的标题，通过设定为 0，表示当前的再现时刻是顶部菜单。

[0296] PSR5 通过设定为 1 ～ 999 的值，表示当前的再现时刻所属的章节号码，通过设定为 0xFFFF，表示在读取装置中章节号码无效。

[0297] PSR6 通过设定为 0 ～ 999 的值，表示当前的再现时刻所属的播放列表 (当前 PL) 的号码。

[0298] PSR7 通过设定为 0 ～ 255 的值，表示当前的再现时刻所属的 PlayItem(当前 PlayItem) 的号码。

[0299] PSR8 通过设定为 0 ～ 0xFFFFFFFF 的值，利用 45KHz 的时间精度表示当前的再现时刻 (当前 PTM(Presentation Time)。通过以上的 PSR4 ～ PSR8，能够在图 16(a) 的 BD-ROM 整体的时间轴上确定当前的再现时刻是哪里。

[0300] CPU24 执行保存在指令 ROM21 中的软件，执行读取装置整体的控制。该控制的内容根据从操作受理部 26 输出的表示用户事件的信息、以及 PSR 组件 23 中的各 PSR 的设定值而动态地变化。

[0301] 通信部 25 实现读取装置的通信功能，如果在 BD-J 模式中被从 Java(注册商标) 应用给予了 URL 指定，则建立与对应于该 URL 的 web 站点的 TCP 连接、FTP 连接等。通过该连接的建立，使 Java(注册商标) 应用进行从 web 站点的下载。

[0302] 操作受理部 26 从用户受理对遥控器进行的操作，将表示该操作的即用户事件的信息通知给 CPU24。

[0303] 以上是有关本实施方式的读取装置的硬件。接着对有关本实施方式的读取装置的软件结构进行说明。

[0304] 图 25 是将由保存在指令 ROM21 中的软件、与硬件构成的部分替换为层结构而描绘的图。如该图所示，读取装置的层结构包括以下的 a)、b)、c) 构成。即，包括 :

[0305] a)BD 播放器驱动器的第 1 层、

[0306] b)BD 播放器模式的第 2 层、

[0307] c) 应用运行期环境的第 3 层构成。

[0308] 这些层中的读取装置的硬件结构属于第 1 层。在该图的第 1 层“BD 播放器驱动

器”中，包括硬件结构中的对应于视频解码器 4、音频解码器 7、IG 解码器 11、PG 解码器 13 的“解码器”、对应于视频平面 5、IG 平面 12、PG 平面 14 的“平面”、BD-ROM 及其文件系 统、本地存储器 20 及其文件系统。

[0309] 第 2 层“BD 播放器模式”包括以下的 b1)、b2) 层构成。即，包括 ：

[0310] b1) 再现控制引擎 32 的层、

[0311] b2) 虚拟文件系统 30 及再现引擎 31 的层，

[0312] 对比自身高级的层提供功能 API。

[0313] 第 3 层“应用运行期环境”包括以下的 c1)、c2) 的层构成。即，包括 ：

[0314] c1) 模组管理器 33 存在的层、

[0315] c2)BD-J 平台 35 存在的层。

[0316] 首先，对属于第 2 层及第 3 层的虚拟文件系统 30 ～ HDMV 模组 34 进行说明。

[0317] 虚拟文件系统 30 是用来将保存在本地存储器 20 中的下载内容与 BD-ROM 中的盘内容一体地处理的虚拟的文件系统。这里，保存在本地存储器 20 中的下载内容包括 SubClip 信息、Clip 信息、播放列表信息。该下载内容中的播放列表信息无论是存在于 BD-ROM 及本地存储器 20 的哪个中的 Clip 信息，在能够指定的点上都与 BD-ROM 上的播放列表信息不同。在该指定时，虚拟文件系统 30 上的播放列表信息不需要通过全路径指定 BD-ROM 或本地存储器 20 中的文件。这是因为 BD-ROM 上的文件系统及本地存储器 20 上的文件系统能够被识别为虚拟的 1 个文件系统 (虚拟文件系统 30)。因此，PlayItem 信息对虚拟文件系统 30 上的 AVClip、BD-ROM 上的 AVClip 的哪个都能够指定再现空间。通过经由虚拟文件系统 30 读取本地存储器 20 的记录内容、与 BD-ROM 的记录内容动态地组合，能够产生各种再现的变化。由于组合本地存储器 20、BD-ROM 而成的盘内容与 BD-ROM 中的盘内容等同地处理，所以本申请中的“BD-ROM”也包括由本地存储器 20+BD-ROM 的组合构成的虚拟的记录媒体。

[0318] 再现引擎 31 执行 AV 再现功能。所谓的读取装置的 AV 再现功能，是从 DVD 播放器、CD 播放器沿袭的传统的功能组，是再现开始 (Play)、再现停止 (Stop)、暂时停止 (Pause On)、暂时停止的解除 (Pause Off)、静止画面功能的解除 (still off)、带速度指定的快进 (Forward Play(speed))、带速度指定的后退 (Backward Play(speed))、声音切换 (Audio Change)、副影像切换 (Subtitle Change)、角度切换 (Angle Change) 的功能。为了实现 AV 再现功能，再现引擎 31 控制视频解码器 4、PG 解码器 13、IG 解码器 11、音频解码器 7，以进行读取到读缓存器 2 上的 AVClip 中的、对应于期望时刻的部分的解码。通过设置所谓期望的时刻而由 PSR8(当前 PTM) 表示的部位的解码，在 AVClip 中能够再现任意的时刻。

[0319] 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 组件 23 中的状态取得 / 设定功能 (ii) 的各功能。所谓的对播放列表的再现控制功能，是指使再现引擎 31 进行的 AV 再现功能中的再现开始及再现停止按照当前 PL 信息及 Clip 信息进行的功能。根据来自 HDMV 模组 34 ～ BD-J 平台 35 的功能调用，来执行这些功能 (i) ～ (ii)。

[0320] 模组管理器 33 保持从 BD-ROM 读取的 index.bdmv，进行分支控制。该分支控制是通过对构成当前标题的动态脚本发出结束事件、对构成分支目的地脚本的动态脚本发出激活事件来进行的。

[0321] HDMV 模组 34 是 HTMV 模式的执行主体，将 MovieObject 读取到存储器中，解读记述在该 MovieObject 中的导航指令，根据解读结果执行对再现控制引擎 32 的功能调用。

[0322] 以上是对再现引擎 31 ～ HDMV 模组 34 的说明。接着对 BD-J 平台 35 进行说明。

[0323] BD-J 平台 35 是所谓的 Java（注册商标）平台，是以 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 Globally Executable MHP Specification(GEM[1.0.2])for package mediatargets 以外，还安装有 BD-J Extention。BD-J Extention 包括为了将超过 GEM[1.0.2] 的功能赋予给 BD-J 平台而特殊化的各种数据包。

[0324] 下面对 BD-J 平台 35 的内部结构进行说明。首先，对作为 BD-J 平台 35 的核心的 Java（注册商标）虚拟机 36 进行说明。

[0325] <Java（注册商标）虚拟机 36>

[0326] 图 26 是表示 Java（注册商标）虚拟机 36 的内部结构的图。如该图所示，Java（注册商标）虚拟机 36 包括图 24 所示的 CPU24、用户类加载器 52、方法区域 53、工作存储器 54、线程 55a、55b、……55n、Java（注册商标）堆栈 56a、56b、……56n。

[0327] 用户类加载器 52 将 BDJA 目录的 Java（注册商标）档案文件中的类文件从脚本存储器 22 等读取并保存在方法区域 53 中。通过应用管理器 37 对用户类加载器 52 指示指定了文件路径的读取，来进行该用户类加载器 52 的类读取。如果文件路径表示脚本存储器 22，则用户类加载器 52 将构成应用的 Java（注册商标）档案文件中的类文件从脚本存储器 22 读取到工作存储器 54 中。如果文件路径表示虚拟文件系统 30 上的目录，则用户类加载器 52 将构成应用的 Java（注册商标）档案文件中的类文件从 BD-ROM 或本地存储器 20 读取到工作存储器 54 中。通过该用户类加载器 52 的类文件读取，来实现应用的启动控制。在脚本存储器 22 中没有被指示了读取的类文件的情况下，用户类加载器 52 将读取失败通知给应用管理器 37。

[0328] 方法区域 53 保存有由用户类加载器 52 从脚本存储器 22 读取的类文件。

[0329] 工作存储器 54 是所谓的堆区域，保存有各种类文件的实例。图 25 所示的应用管理器 37 是常驻在该工作存储器 54 中的常驻应用。在工作存储器 54 中，除了这些常驻型的实例以外，还保存有对应于 读取到方法区域 53 中的类文件的实例。该实例是构成应用的 xlet 程序。通过将该 xlet 程序配置在工作存储器 54 中，应用成为可执行的状态。

[0330] 在图 25 的层模式中，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，但是这只不过是为了实现容易理解的考虑。现实的记述是应用管理器 37 及应用作为实例由线程 55a、55b、……55n 执行。

[0331] 线程 55a、55b、……55n 是执行保存在工作存储器 54 中的方法的逻辑的执行主体，将本地变量及保存在操作数堆栈中的自变量作为操作数进行运算，将运算结果保存在本地变量或操作数堆栈中。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、55b、……55n 的方法供给。相对于物理的执行主体是 CPU 唯一，作为逻辑执行主体的线程可以在 Java（注册商标）虚拟机 36 内存在最多 64 个。在该 64 个的数值内，既可以新制作线程，也可以将现有的线程删除，线程的动作数可以在 Java（注册商标）虚拟机 36 的动作中增减。由于线程的数量可以适当增加，所以也可以通过由多个线程进行 1 个实例的并列执行来实现实例的高速化。在该图中，CPU24 与线程的对应关系为 1 对多的关系，但在有多

个 CPU 的情况下，CPU 与线程的对应关系可以为多对多的关系。通过线程 55a、55b、……55n 进行的方法执行是在将构成方法的字节码变换为 CPU24 的内部码后发送给 CPU24 来进行的。对于该内部码变换，由于脱离了本发明的着眼点，所以省略说明。

[0332] Java（注册商标）堆栈 56a、56b、……56n 与线程 55a、55b、……55n 以 1 对 1 的比例存在，在内部中具有程序计数器（图中的 PC）、和 1 个以上的帧。“程序计数器”表示在实例中当前在执行哪个部分。“帧”是对对应于方法的 1 次调用分配的堆栈式的区域，包括保存该 1 次调用时的自变量的“操作数堆栈”、和被调用的方法所使用的“本 地变量堆栈（图中的本地变量）”。由于帧每当进行了 1 次调用就堆积在 Java（注册商标）堆栈 56a、56b、……56n 上，所以在某个方法递归调用自身的情况下，该帧也堆积 1 个。

[0333] 以上是对 Java（注册商标）虚拟机的说明。

[0334] ＜应用管理器 37＞

[0335] 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，在发生了标题间的分支的情况下，利用对应于前面的标题的 AMT 和对应于当前标题的 AMT，来执行信令。该信令是下述控制，即，使虽然记载在对应于前面的标题的 AMT 中、但没有记载在对应于当前标题的 AMT 中的应用的动作结束，使没有记载在对应于前面的标题的 AMT 中、而记载在对应于当前标题的 AMT 中的应用的动作开始。

[0336] 图 27 是表示基于 BD-Jobject 的应用管理表的、应用管理器 37 的处理的图。

[0337] 图 27 中的☆ 1、☆ 2、☆ 3 示意地表示应用管理表参照（☆ 1）、对 Java（注册商标）虚拟机 36 的应用启动指示（☆ 2）、Java（注册商标）虚拟机 36 的对 Java（注册商标）档案文件的读取指示（☆ 3）、定义 Java（注册商标）应用的类文件的类装载（☆ 4、5、6）这一系列的过程。通过该启动指示，Java（注册商标）虚拟机 36 将 xlet 程序从脚本存储器 22 读取到工作存储器中。

[0338] 图 28 是表示基于 BD-Jobject 的 PLMT 的、应用管理器 37 的处理的图。▽1表示 BD-Jobject 的 PLMT 的参照，▽2表示对再现引擎 31 的播放列表信息的读取指示。

[0339] 图 28 中的◎ 1、2、3、4 示意地表示经由虚拟文件系统 30 的播放列表信息读取（◎ 1）、构成播放列表信息的 PlayItem 信息的解读（◎ 2）、经由虚拟文件系统 30 的 Clip 信息读取（◎ 3）、Clip 信息的解读。如果经过以上的过程解读 Clip 信息、播放列表信息，则将构成 AVClip 的 TS 数据包经由虚拟文件系统 30 交接给再现引擎 31。如果这样将 TS 数据包依次交给再现引擎 31，则再现引擎 31 将构成 AVClip 的 TS 数据包输出给解码器，显示在平面上。图中的☆ 1、2、3、4 示意地表示构成 AVClip 的 TS 数据包的读取（☆ 1、2）、从虚拟文件系统 30 向再现引擎 31 的 TS 数据包交接（☆ 3）、向解码器的 TS 数据包投入（☆ 4）、从解码器向各种平面的解码结果输出（☆ 5）。

[0340] 以上是对应用管理器 37 的说明。

[0341] ＜功能限制部 38＞

[0342] 功能限制部 38 是对应于 J2ME PBP1.0 的 JSSE optional package 的结构要素，对要赋予给应用的功能施加限制或解除其限制。J2MEPBP1.0 的 JSSE optional package 是在 BD-J 平台的安装中必须的数据包，JSSE 实现 Java（注册商标）2 安全模式。所谓的 Java（注册商标）2 安全模式，是认证 Sigend 应用、并对认证后的应用许可超越核心功能的功能。所谓的超越核心功能的功能有以下这些。

[0343] • 本地寄存器的读写

[0344] • 网络连接的利用

[0345] • BD-ROM 的访问

[0346] • BD-ROM 中的其他标题的选择

[0347] • 其他 BD-J 平台的执行控制

[0348] 为了获得这些功能的许可，必须利用许可请求文件。上述的功能的许可能够从该许可请求文件获得。

[0349] 此外，JSSE 是用来建立安全的连接的数据包。包括“Java（注册商标）安全连接数据包”。

[0350] 通过使用该数据包，BD-J 平台能够进行与因特网的服务器的连接。物理的连接是 Ethernet（注册商标）、电话的哪种都可以。连接的条件是支持 TCP/IP，并且能够使用 HTTP 协议。BD-J 平台在利用网络连接前必须被认证，并且必须得到用来进行网络连接的适当的许 可。

[0351] 功能限制部 38 的特征是，根据 Java（注册商标）2 安全模式，对应用能够执行的功能施加限制、或将该限制解除。

[0352] 以上，结束了对 BD-J 平台 35 的内部结构的说明。

[0353] ＜跨越盘的应用信令＞

[0354] 以上说明了从一个 BD-ROM 中的某个标题向其他标题切换再现时的应用信令。下面，对从某个 BD-ROM 中的标题向其他 BD-ROM 中的标题切换再现时的应用信令进行说明。

[0355] 作为从某个 BD-ROM（盘 A）中的标题向其他 BD-ROM（盘 A+1）中的标题切换再现的例子，可以举出长篇的电影作品或系列的电影作品作为 BD-BOX 收录在多个 BD-ROM 中的情况。由于电影作品是长篇的情况、或者由于电影作品是系列作品，所以可以假设电影作品不能收录到一个盘中的情况。在此情况下，由于将 1 个电影作品作为多个标题收录在多个 BD-ROM 中，所以需要盘的交换作业。在该交换作业中，必须根据与在盘 A 的最后再现的标题（LastPlay 标题）相对应的 AMT、和与在盘 A+1 的最初再现的标题（FirstPlay 标题）相对应的 AMT 执行信令。

[0356] 该信令由于在新盘的装载时进行应用的启动、结束的控制，所以根据应用，也有在盘的交换前后动作的应用。这样将在盘的交换前后动作的应用称作“盘无界应用”。反之，将在盘的交换后结束的动作的应用称作“盘边界应用”。

[0357] 优选地将进行在前剧结束时显示催促用户交换盘的消息、将进行一旦插入了收录有后剧的盘则能够马上开始再现那样的处理的应用，规定为盘无界应用。

[0358] 这里，在新盘的装载时进行信令是根据以下的理由。

[0359] 通常，在盘被取出的情况下结束所有的应用对于用户来说容易理解。DVD-Video 中的再现控制是这样规定的。规定该再现控制的应 用是对应于通常盘的应用，如果盘被取出则不再有对应的盘，最好结束应用。但是，虽然是特殊的例子，但在长篇的电影作品及系列的电影作品作为 BD-BOX 被收录在多个 BD-ROM 中的情况下，如果应用进行处理以便催促盘的切换，则是非常方便的。根据这样的情况，导入了盘无界应用。

[0360] 图 29 是表示盘边界应用、盘无界应用的动作的图。第 1 段表示盘 A 的装载、盘 A 中的标题的再现、盘 A 的弹出、盘 A+1 的装载、盘 A+1 中的标题的再现这一系列的流程。

[0361] 第 2 段表示再现盘 A 的 LastPlay 标题的期间、没有盘的期间、再现盘 A+1 的 FirstPlay 标题的期间。

[0362] 第 3 段表示 LastPlay 标题的再现内容、催促交换盘的消息、FirstPlay 标题的再现内容。第 4 段、第 5 段表示盘边界应用、盘无界应用的生命周期。根据该第 4 段，盘边界应用在盘 A 的装填中开始动作。该盘边界应用的生命周期在装填有盘 A 的期间、以及没有盘的期间也持续。如果装载了盘 A+1、开始 FirstPlay 标题的再现，则执行比对与 LastPlay 标题对应的 AMT、和与 FirstPlay 标题对应的 AMT 的信令，通过该控制，盘边界应用的动作结束。

[0363] 另一方面，根据图 29 的第 5 段，盘无界应用在盘 A 的装填中开始动作。该盘无界应用的生命周期在装填有盘 A 的期间、以及没有盘的期间也持续。如果装载了盘 A+1、开始 FirstPlay 标题的再现，则执行比对与 LastPlay 标题对应的 AMT、和与 FirstPlay 标题对应的 AMT 的应用的启动——结束控制。通过该控制，盘无界应用继续其动作。

[0364] 图 30(a) 是表示与盘 A 中的 LastPlay 标题对应的 AMT、和与标题 A+1 的 FirstPlay 标题对应的 AMT 的图，图 30(b) 是表示两个 AMT 如图 30(a) 那样规定时的信令的图。

[0365] 在图 30(a) 中可知，在与盘 A 的 LastPlay 标题对应的 AMT 中 记载有应用 #1、应用 #2 的 ID，在与盘 A+1 的 FirstPlay 标题对应的 AMT 中记载有应用 #2 的 ID。

[0366] 图 30(b) 的第 1 段表示盘 A 的装载、盘 A 中的标题的再现、盘 A 的弹出、盘 A+1 的装载、盘 A+1 中的标题的再现这一系列的流程。

[0367] 第 2 段表示再现盘 A 的 LastPlay 标题的期间、没有盘的期间、再现盘 A+1 的 FirstPlay 标题的期间。

[0368] 第 3 段表示 LastPlay 标题的再现内容、催促交换盘的消息、FirstPlay 标题的再现内容。第 4 段、第 5 段表示对应用 #1、应用 #2 的控制。

[0369] 在第 4 段中，应用 #1 虽然记载在盘 A 的 LastPlay 标题的 AMT 中，但没有记载在盘 A+1 的 FirstPlay 标题的 AMT 中，所以应用管理器 37 使应用 #1 结束。

[0370] 在第 5 段中，应用 #2 记载在盘 A 的 LastPlay 标题的 AMT 中、并记载在盘 A+1 的 FirstPlay 标题的 AMT 中，所以应用管理器 37 不进行结束处理而使该应用 #2 原样继续动作。

[0371] 假设在盘 A 中记录有具有恶意的应用，并启动了该应用。在尽管执行了恶意的程序但还是弹出盘 A 并装载了盘 A+1 的情况下，有可能发生由该恶意的程序拷贝盘 A+1 的不测的情况。为了禁止这样的恶意的程序的动作，应用管理器 37 在盘不存在的期间中，对功能限制部 38 指示以使继续动作的应用从 Signed 应用变化为 Unsigned 应用。并且，在盘 A+1 装载时的信令中，在判断继续应用的动作时，使继续动作的应用从 Unsigned 应用变化为 Signed 应用。通过该变化，即使在盘 A 中存在恶意的程序，由于该恶意的程序的功能被限制，也不用担心上述那样的盘 A+1 的内容被拷贝。

[0372] 此外，在与盘 A+1 的 FirstPlay 标题对应的 AMT 中没有注册有当前执行的应用 ID、或者插入了错误的盘的情况下，应用管理器 37 使当前动作中的应用结束。通过该结构，不用担心恶意的程序与盘一起动作。

[0373] 图 31 是表示应用管理器 37 的处理顺序的流程图。

[0374] 该流程图首先执行步骤 S1 ～步骤 S2 的循环处理。步骤 S1 是是否发生了标题跳

转的判断，如果有标题跳转，则在步骤 S3 中进行标题的切换。

[0375] 步骤 S2 是是否进行了盘弹出的判断，如果进行了盘弹出，则对功能限制部 38 指示以使动作中的 Signed 应用变更为 Unsigned 应用（步骤 S4），在显示催促交换盘的消息后（步骤 S5），成为新盘的装载等待状态（步骤 S6）。如果进行了装载，则判断被装载的盘是否是预定的盘（步骤 S7）。如果不是预定的盘，则转移到步骤 S8。

[0376] 这里，是否装载了预定的盘的判断是以以下的顺序进行的。在构成 BD-BOX 的各 BD-ROM 中，记录有“接着要装载的盘的标识符”。应用管理器 37 在盘 A 的装载时读取记录在盘 A 中的“接着要装载的盘的标识符”。并且，在盘 A 的弹出后，如果装载了新的 BD-ROM，则读取该盘的标识符，判断从该新 BD-ROM 读取的标识符是否与记录在盘 A 中的“接着要装载的盘的标识符”一致。如果一致，则新装载的 BD-ROM 是构成 BD-BOX 的多个 BD-ROM 中的、构成续编的 BD-ROM，做出判断结果“装载了预定的盘”。

[0377] 如果不一致，则新装载的 BD-ROM 不是构成 BD-BOX 的续编的 BD-ROM，做出判断结果“装载了预定外的盘”。通过这样的判断顺序，能够区别是正确地装载了盘 A+1、还是装载了完全不同的盘。

[0378] 步骤 S8 显示规定的菜单，确认是否可以结束跨越盘的应用。图 32(a) 是催促交换盘的消息的显示例，图 32(b) 是在步骤 S8 中显示的菜单的一例。图 32(a) 的消息包括盘 A 的再现结束的消息、和将盘 A 弹出并提示盘 A+1 的装载的消息。图 32(b) 的菜单包括“启动了不能由装填的盘执行的应用。结束应用吗？”的消息、和“继续 按钮”、“重放盘按钮”。

[0379] 显示该菜单的理由如下。

[0380] 在用户错将盘 A+1 以外的其他盘插入的情况下，由于在该盘中的 FirstPlay 标题的 AMT 中没有记述有当前动作中的应用的 ID，所以与其他盘的再现开始同时地结束当前动作中的应用。

[0381] 原本期待盘无界应用即使换了盘也动作。尽管本来是想要持续启动的应用，但是因用户错误地放入了盘而结束，这是不方便的。因此，在尽管启动了盘无界应用、但在新插入的盘的 FirstPlay 标题中的 AMT 中没有显示有盘无界应用的情况下，假设用户放入了错误的盘的可能性较高，显示上述的菜单，确认是否要结束当前执行中的盘无界应用。

[0382] 步骤 S9 判断是否确定了菜单的继续按钮、重放盘按钮的某一个。在用户想要进行与盘 A 无关系的其他的盘的再现而确定了继续按钮的情况下，结束动作中的 Unsigned 应用的动作（步骤 S10），将新盘的 FirstPlay 标题作为当前标题（步骤 S11），从开头开始当前标题的 AutoPlay 播放列表的再现（步骤 S12）。

[0383] 在弄错了插入的盘而确定了菜单中的『重放盘』按钮的情况下，将盘弹出（步骤 S20），转移到步骤 S6。此时，由于没有进行应用的结束，所以应用的动作继续。

[0384] 这样，在装载了与想要的盘不同的盘的情况下，通过使用户选择重放盘、还是继续应用的动作，使盘无界应用不会与没有关联的盘同时执行，能够防止用户的混乱。

[0385] 图 33 是示意地表示装载了没有预定的盘时的应用管理器 37 的处理的时序图。

[0386] 该图中的第 1 段表示盘 A 的弹出、没有盘的期间、没有预定的盘（盘 B）的装载这一系列的流程。

[0387] 第 2 段表示在第 1 段的流程中显示的消息、菜单。在没有盘的期 间中，显示“催促交换盘的消息”，但如果装载了没有预定的盘，则显示菜单，该菜单是图 32(b) 所示的菜单。

[0388] 第 3 段表示对第 2 段中的菜单确定了继续按钮时的处理。在此情况下，结束盘无界应用的动作，开始新装载的盘的 FirstPlay 标题的再现。

[0389] 第 4 段表示对第 2 段中的菜单确定了重放盘按钮时的处理。在此情况下，不结束盘无界应用的动作，而等待装载其他的盘。图 34 是将该图 33 所示的应用管理器 37 的处理分情况表示的图。

[0390] 图 34(a) 表示装载了预定的盘时的处理，图 34(b) 表示装载了与预定的盘不同的盘、并且用户希望进行该盘的再现时的应用管理器 37 的处理。在该图 34(b) 中，在装载了与预定的盘不同的盘的情况下，显示图 32(b) 所示的菜单，通过确定该菜单中的继续按钮，开始记录在该盘中的 FirstPlay 标题的再现。

[0391] 图 34(c) 表示装载了与预定的盘不同的盘、并且用户不希望进行该盘的再现时的应用管理器 37 的处理。在该图 35(c) 中，在装载了与预定的盘不同的盘的情况下，显示图 32(b) 所示的菜单，通过确定该菜单中的重放盘按钮，将该盘弹出。

[0392] 再开始图 31 中的各步骤的处理的说明。

[0393] 如果装载的盘不是预定的盘，则步骤 S7 为“是”，转移到步骤 S13。

[0394] 步骤 S13 将新盘的 FirstPlay 标题作为当前标题 (步骤 S13)，判断是否存在虽然记载在前标题的 AMT 中、但没有记载在当前标题的 AMT 中的动作中的 Unsigned 应用 x(步骤 S14)。如果存在这样的 Unsigned 应用，则结束该 Unsigned 应用 x 的动作 (步骤 S15)。

[0395] 步骤 S16 判断是否存在记载在前标题的 AMT 中、且记载在当前标题的 AMT 中的动作中的 Unsigned 应用 y。如果存在这样的 Unsigned 应用 y(步骤 S16 中“是”)，则对功能限制部 38 指示以将 该 Unsigned 应用 y 变更为 Signed 应用 (步骤 S17)，继续应用的动作 (步骤 S18)。

[0396] 然后，在步骤 S19 中，跳过当前标题的 AutoPlay 播放列表中的、对应于警告文字等的开头部分，开始 AutoPlay 播放列表的再现。其理由如下。

[0397] 在记录在 DVD-Video 等光盘中的电影作品的再现中，惯例是 :如果装载盘，则显示禁止拷贝等的警告文字，然后播放其他盘的预告等，转移到菜单。在菜单中选择主体的再现才开始主体的再现，但是，在跨越多个盘的内容的情况下，每次替换盘都显示这样的警告文字及菜单等，内容的连贯感会很差。但是，关于这些警告文字及菜单，在 BD-BOX 中，一般是在插入每个盘时，不论从哪个盘再现都一定再现。

[0398] 所以，在本实施方式中，在继续盘无界应用的动作的情况下，在装载了预定的盘 A+1 的情况下，执行“从盘 A 向盘 A+1 的交换时特有的再现控制”，将对应于警告文字等的开头部分跳过。

[0399] 图 35 是对比表示将盘 A+1 的 FirstPlay 标题通常再现的情况、和将盘 A+1 的 FirstPlay 标题跳转再现时的图。该图的第 1 段表示盘 A 的装载、盘 A 中的标题的再现、盘 A 的弹出、盘 A+1 的装载、盘 A+1 中的标题的再现这一系列的流程。第 2 段表示将盘 A+1 的 FirstPlay 标题通常再现的情况，第 3 段表示将盘 A+1 的 FirstPlay 标题的开头部分跳过的情况。比较第 2 段、第 3 段可知，在第 2 段中，在显示了警告文字、其他盘的预告后显示主体，在第 3 段中，不显示它们而直接开始主体。这样，在视听了盘 A 后继续视听盘 A+1 那样的情况下，由于盘 A+1 的 FirstPlay 标题中的警告文字等不视听也可以，所以能够实现步调较好的连续再现。当然，由于在不再现盘 A 而从盘 A+1 开始再现的情况下再现上述警告文

字等，所以能够遵守电影再现时的习惯。

[0400] 以上是对应用管理器 37 的说明。

[0401] 下面，参照图 36 的流程图，说明再现控制引擎 32 的具体的控制顺序。

[0402] 图 36 是表示再现控制引擎 32 的播放列表再现顺序的流程图。该再现顺序包括对再现引擎 31 的控制（步骤 S106）、和对 BD-ROM 驱动器 1 或本地存储器 20 的控制（步骤 S108）。在该流程图中，设作为处理对象的 PlayItem 为 PlayItem#x。该流程图进行当前 PL 信息（.mpls）的读入（步骤 S101），然后执行步骤 S102 ～步骤 S110 的处理。这里，步骤 S102 ～步骤 S110 构成循环处理，该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 的处理直到步骤 S109 成为“是”。在该循环处理中，将作为处理对象的 PlayItem 称作 PlayItem#x(PI#x)。该 PlayItem#x 在步骤 S102 中被初始化。上述循环处理的结束条件是该 PlayItem#x 成为当前播放列表的最后的 PlayItem（步骤 S109），如果不是最后的 PlayItem，则将当前播放列表中的下个 PlayItem 设定为 PlayItem#x（步骤 S110）。

[0403] 在循环处理中重复执行的步骤 S103 ～步骤 S110 中，将由 PlayItem#x 的的 Clip_information_file_name 指定的 Clip 信息读入到脚本存储器 22 中（步骤 S103），利用当前 Clip 信息的 EPmap 信息，将 PlayItem#x 的 In_time 变换为 I 图片地址 u（步骤 S104），利用当前 Clip 信息的 EP_map 信息，将 PlayItem#x 的 Out_time 变换为 I 图片地址 v（步骤 S105）。这里，为了进行对应于 PlayItem#x 的 Out_time 的图片数据的解码，仅通过位于地址 v 的 I 图片是不够的，还需要后续于 PlayItem#x 的 Out_time 的图片数据。这是因为，对应于 PlayItem#x 的 Out_time 的图片数据可能参照未来方向的图片数据。

[0404] 如上所述，在 EP_map 中，将 Access Unit、即对应于 GOP 的开头的 I 图片的地址与 I 图片的再现时刻建立对应地表示。因此，可以通过确定 EP_map 中所示的 I 图片地址中的、I 图片地址 v 的下个地址、求出其前一个的 I 图片的地址（设为地址 w），来决定对应于 PlayItem#x 的 Out_time 的图片数据所归属的 GOP 的末端地址（步骤 S107）。如果利用这样计算的地址 w，对 BD-ROM 驱动器 1 或本地存储器 20 命令进行从 I 图片地址 u 到地址 w 的 TS 数据包的读取（步骤 S108），则将对应于 PlayItem#x 的 Out_time 的图片数据所参照的图片数据都读入到解码器内。

[0405] 然后，判断 PlayItem#x 是否是当前播放列表的最后的 PI（步骤 S109）。

[0406] 如果 PlayItem#x 不是当前播放列表的最后的 PI，则将当前播放列表中的下个 PlayItem 设定为 PlayItem#x（步骤 S110），回到步骤 S103。通过重复以上的步骤 S103 ～步骤 S110，将构成播放列表的 PI 依次再现。

[0407] 在该流程图中，在步骤 S102 中，在当前的播放列表信息中，将从应用管理器 37 指定的 PlayItem 设定为 PlayItem#x。这里，在应用管理器 37 不是指示对应于警告文字等的 PlayItem、而是指示了对应于主体的开头的 PlayItem 的情况下，跳过该警告文字等，而开始播放列表的再现。

[0408] 以上，根据本实施方式，由于能够使应用跨越多个盘而连续地动作，所以在应用保持有由用户选择的声音的语言属性、及字幕的语言属性等的情况下，即使交换了盘，也不需要再次设定该声音的语言属性、字幕的语言属性等。由于能够不再次设定这些属性而再现交换后的盘，所以有能够将系列或长篇的电影作品适当地再现的效果。

[0409] （第 2 实施方式）

[0410] 本实施方式关于第 1 实施方式所示的 Java(注册商标)应用的详细情况。具体而言,该 Java(注册商标)应用以构成 BD-BOX 的多个 BD-ROM 中的第 2 片以后的 BD-ROM(第 1 实施方式中所述的盘 A+1)的 FirstPlay 标题作为生命周期。该 Java(注册商标)应用在再现、启动 FirstPlay 标题时,对应用管理器 37 询问“是从盘 A 向盘 A +1 交换的情况?”、“还是没有装载盘 A 而直接装载盘 A+1 的情况?”。并且,根据该询问的结果来改变再现控制。

[0411] 所谓的“直接装载盘 A+1 的情况的再现控制”,是从记述在 BD-JObject 的播放列表记录表中的、构成 PlayList 信息的多个 PlayItem 信息中的开头开始再现。

[0412] 所谓的“从盘 A 向盘 A+1 交换时特有的再现控制”,是从记述在 BD-J Object 的播放列表记录表中的、构成具有 AutoPlay 属性的 PlayList 信息的多个 PlayItem 信息中的中途开始再现(实现所谓的第 1 实施方式所示的跳转再现)。

[0413] 此外,在从盘 A 向盘 A+1 交换的情况下,以 FirstPlay 标题为生命周期的 Java(注册商标)应用也可以将与具有 AutoPlay 属性的 PlayList 信息不同的 PlayList 信息再现。如果该不同的 PlayList 信息定义了省略警告文字的再现的再现路径,则可以不使用户视听第 1 实施方式所述那样的警告文字,而能够适当地执行多个盘的连续再现。

[0414] 以上,根据本实施方式,通过使根据“是从盘 A 向盘 A+1 交换的情况?”、“还是没有装载盘 A 而直接装载盘 A+1 的情况?”而进行不同的处理的 Java(注册商标)应用,作为以 FirstPlay 标题为生命周期的 Java(注册商标)应用,记录在 BD-BOX 中的多个 BD-ROM 中的 1 个中,能够步调良好地进行 BD-BOX 中的多个 BD-ROM 的连续再现。

[0415] (备注)

[0416] 以上,在本申请的提出申请时,对申请人能够知道的最优选的实施方式进行了说明,但对于以下所示的技术课题,能够加以进一步的改良或变更实施。想要提起注意的是,按照上述实施方式所示那样实施、还是实施它们的改良、变更都是任意的,是按照实施的人的意愿进行的。

[0417] (应用 ID 的管理)

[0418] 在应用管理器 37 执行上述那样的跨越盘的信令的情况下,需要按每个创作站点(authoring site)管理应用 ID 以使其不会重复等、使盘 A+1 的应用 ID 不会与恶意程序的应用 ID 偶然重复。

[0419] (用来连续再现的补充的处理)

[0420] 为了进行两个盘的连续再现,优选为,除了启动应用本身以外,还保存属性等,由在盘 A+1 插入时启动的应用参照。

[0421] (应用结束时的确认方法)

[0422] 应用管理器 37 在装载了没有预定的光盘的情况下通过菜单确认应用的结束,但也可以不是这样,而是询问是否可以开始装载的盘的再现等、通过其他的方法来判断是继续应用的动作、还是结束。

[0423] (控制顺序的实现)

[0424] 由于在各实施方式中引用流程图说明的控制顺序、功能性的结构要素的控制顺序是利用硬件资源具体地实现的,所以可以说是利用了自然法则的技术思想的创作,满足作为“程序的发明”的成立条件。

[0425] • 有关本发明的程序的生产方式

[0426] 有关本发明的程序是计算机能够执行的执行形式的程序（目标程序），由使计算机执行的 1 个以上的程序代码构成各实施方式所示的流程图的各步骤、功能性结构要素的各个顺序。这里，程序代码如处理器的原始代码、Java（注册商标）字节码那样，有各种种类。此外，在程序代码所进行的各步骤的实现中，有各种方式。在能够利用外部函数实现各步骤的情况下，调用该外部函数的调用语句成为程序代码。此外，也有实现 1 个步骤的程序代码属于各个目标程序的情况。在限制了指令种类的 RISC 处理器中，也有通过组合算术运算指令及逻辑运算指令、分支指令等、来实现流程图的各步骤的情况。

[0427] 有关本发明的程序可以如以上那样制作。首先，软件开发者利用编程语言，记述实现各流程图及功能性结构要素的软件程序。在该记 述时，软件开发者按照编程语言的语法，利用类构造体及变量、数组变量、外部函数的调用，记述具体实现各流程图及功能性结构要素的源程序。

[0428] 将所记述的源程序作为文件交给编译器。编译器翻译这些源程序，从而生成目标程序。

[0429] 编译器的翻译包括语法解析、最优化、资源分配、代码生成的过程构成。在语法解析中，进行源程序的语句解析、语法解析及意思解析，将源程序变换为中间程序。在最优化中，对中间程序进行基本块化、控制流解析、数据流解析的作业。在资源分配中，为了实现向作为对象的处理器的指令组的匹配，将中间程序中的变量分配给作为对象的处理器的处理器所具有的寄存器或存储器。在代码生成中，将中间程序中的各中间指令变换为程序代码，得到目标程序。

[0430] 如果生成了目标程序，则编程者启动链接程序。链接程序将这些目标程序、及关联的库程序分配到存储器空间中，将它们结合为一个，生成装载模组。这样生成的装载模组是以计算机的读取为前提的，使计算机执行各流程图所示的处理顺序及功能性结构要素的处理顺序。经过以上的处理，能够制作有关本发明的程序。

[0431] • 有关本发明的程序的使用方式

[0432] 有关本发明的程序可以如以下这样使用。

[0433] (i) 作为嵌入程序的使用

[0434] 在将有关本发明的程序作为嵌入程序使用的情况下，将对应于程序的装载模组与基本输入输出程序 (BIOS) 及各种中间件（操作系统）一起写入到指令 ROM 中。通过将这样的指令 ROM 嵌入到控制部中而使 CPU 执行，能够将有关本发明的程序作为读取装置 200 的控制程序使用。

[0435] (ii) 作为应用的使用

[0436] 在读取装置 200 是硬盘内装模式的情况下，基本输入输出程序 (BIOS) 嵌入在指令 ROM 中，各种中间件（操作系统）被安装在硬盘中。此外，在读取装置 200 中设有用来从硬盘启动系统的启动 ROM。

[0437] 在此情况下，仅将装载模组经由可移植型记录媒体或网络供给到读取装置 200 中，作为 1 个应用装载到硬盘中。这样，读取装置 200 进行启动 ROM 的启动引导，在启动操作系统后，作为 1 个应用，使 CPU 执行该程序，来使用有关本发明的程序。

[0438] 在硬盘模式的读取装置 200 中，由于可以将本发明的程序作为 1 个应用使用，所以能够将有关本发明的程序以单体转让、出借、或经由网络供给。

[0439] （指令 ROM21、CPU24）

[0440] 指令 ROM21、CPU24 可以作为 1 个系统 LSI 实现。

[0441] 所谓的系统 LSI，是在高密度基板上安装裸片、并封装而成的。通过将多个裸片安装在高密度基板上并封装，使多个裸片具有宛如 1 个 LSI 那样的外形构造者，也包含在系统 LSI 中（将这样的系统 LSI 称作多芯片模组）。

[0442] 这里，如果着眼于封装的种类，则在系统 LSI 中有 QFP（方形扁平封装）、PGA（引脚网格阵列）的种类。QFP 是在封装的四个侧面上安装有引脚的系统 LSI。PGA 是在整个底面上安装有多个引脚的系统 LSI。

[0443] 这些引脚起到作为与其他电路的接口的作用。由于在系统 LSI 的引脚上存在这样的接口的作用，所以通过将其他电路连接在系统 LSI 的这些引脚上，系统 LSI 起到作为读取装置 200 的核心的作用。

[0444] 封装在系统 LSI 中的裸片包括“前置部”、“后置部”、“数字处理部”。“前终端部”是将模拟信号数字化的部分，“后终端部”是将数字处理的结果所得到的数据模拟化而输出的部分。

[0445] 在上述实施方式中作为内部结构图表示的各结构要素安装在该 数字处理部内。

[0446] 首先，如“作为嵌入程序的使用”中所述，在指令 ROM 中，写入对应于程序的装载模组、基本输入输出程序 (BIOS)、各种中间件（操作系统）。在本实施方式中，特别创造的是对应于该程序的装载模组部分，所以通过将保存有对应于程序的装载模组的指令 ROM 作为裸片封装，能够生产有关本发明的系统 LSI。

[0447] 对于具体的安装，优选为能够使用 SoC 安装或 SIP 安装。所谓的 SoC(System on chip：基于片上系统）安装，是在一个芯片上烧制多个电路的技术。所谓的 SIP(System in Package：系统级封装）安装，是将多个芯片通过树脂等做成 1 个封装的技术。经过以上的过程，有关本发明的系统LSI能够基于各实施方式所示的读取装置200的内部结构图制作。

[0448] 另外，如上述那样生成的集成电路有时根据集成度的差异也被称作 IC、LSI、超级 LSI、超大规模 LSI。

[0449] 进而，也可以将各记录读取装置的结构要素的一部分或全部作为 1 个芯片构成。集成电路化并不限于上述的 SoC 安装、SIP 安装，也可以通过专用电路或通用过程来实现。可以考虑利用在 LSI 制造后可编程的 FPGA(Field Programmable Gate Array：可现场编程门阵列）、以及可重构 LSI 内部的电路单元的连接及设定的可重构处理器。进而，如果因为半导体技术的进步或派生技术而出现代替 LSI 的集成电路化的技术，则当然也可以利用该技术进行功能块的集成电路化。例如作为可能性，有生物技术的应用。

[0450] 工业实用性

[0451] 有关本发明的读取装置在上述实施方式中公开了内部结构，并显然可以根据该内部结构而批量生产，所以在实质上能够进行工业应用。因此，有关本发明的读取装置具有工业上的可利用性。

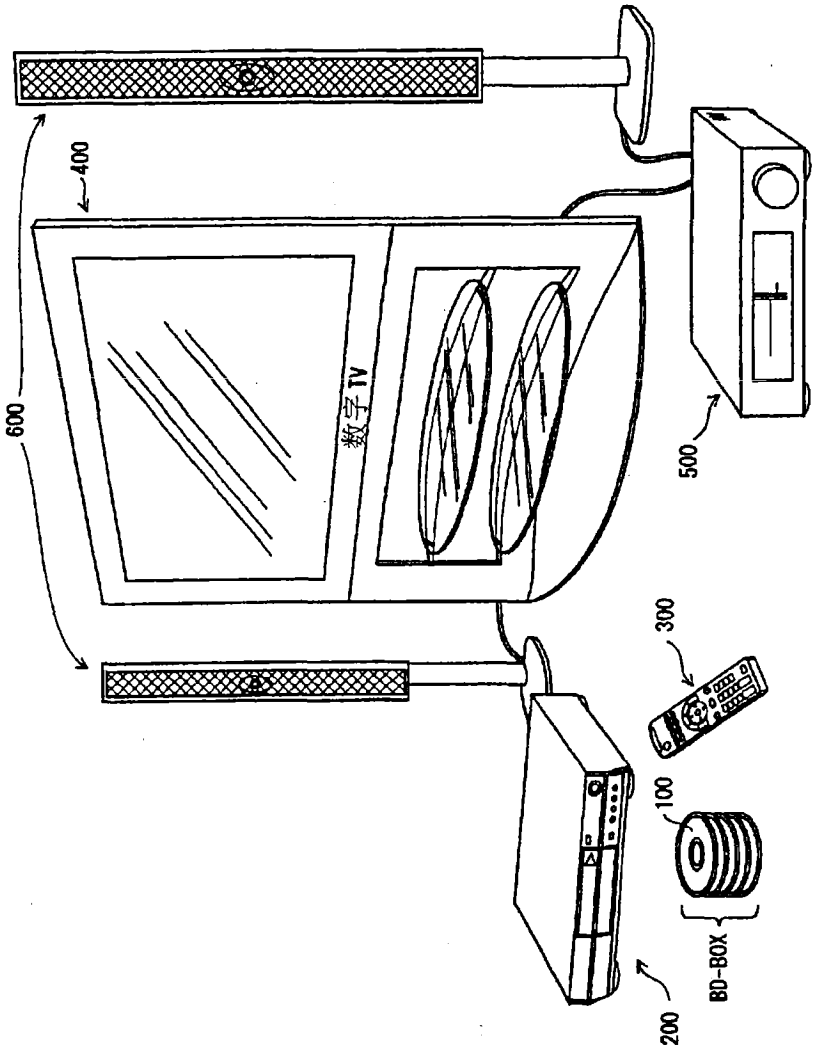


图 1

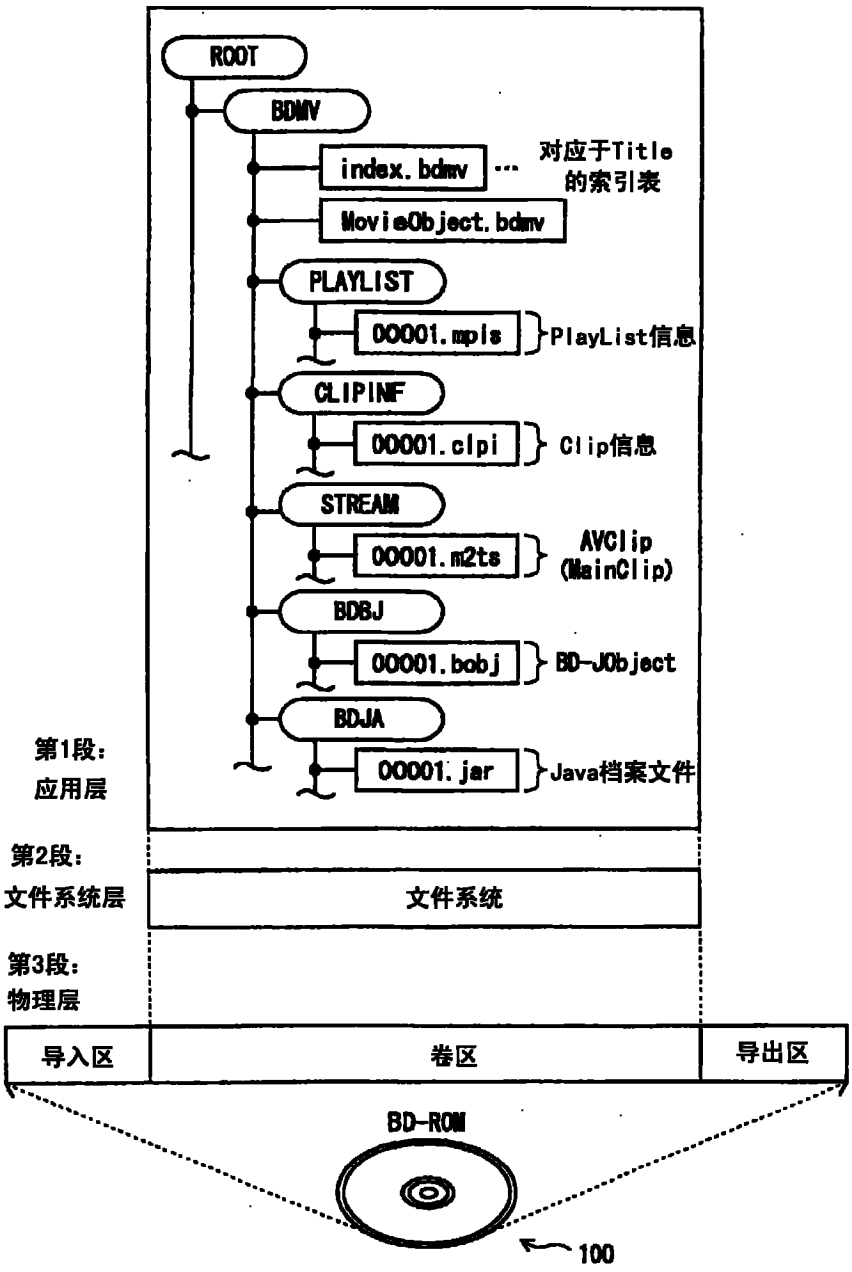

ROOT
BDMV
index.bdmv
对应于Title的索引表
MovieObject.bdmv
PLAYLIST
00001.mpls
PlayList信息
CLIPINF
00001.clpi
Clip信息
STREAM
00001.m2ts
AVClip (MainClip)
BDBJ
00001.bobj
BD-JObject
BDJA
00001.jar
Java档案文件
第1段：应用层
第2段：文件系统层
文件系统
第3段：物理层
导入区
卷区
导出区
BD-ROM
100

图 2

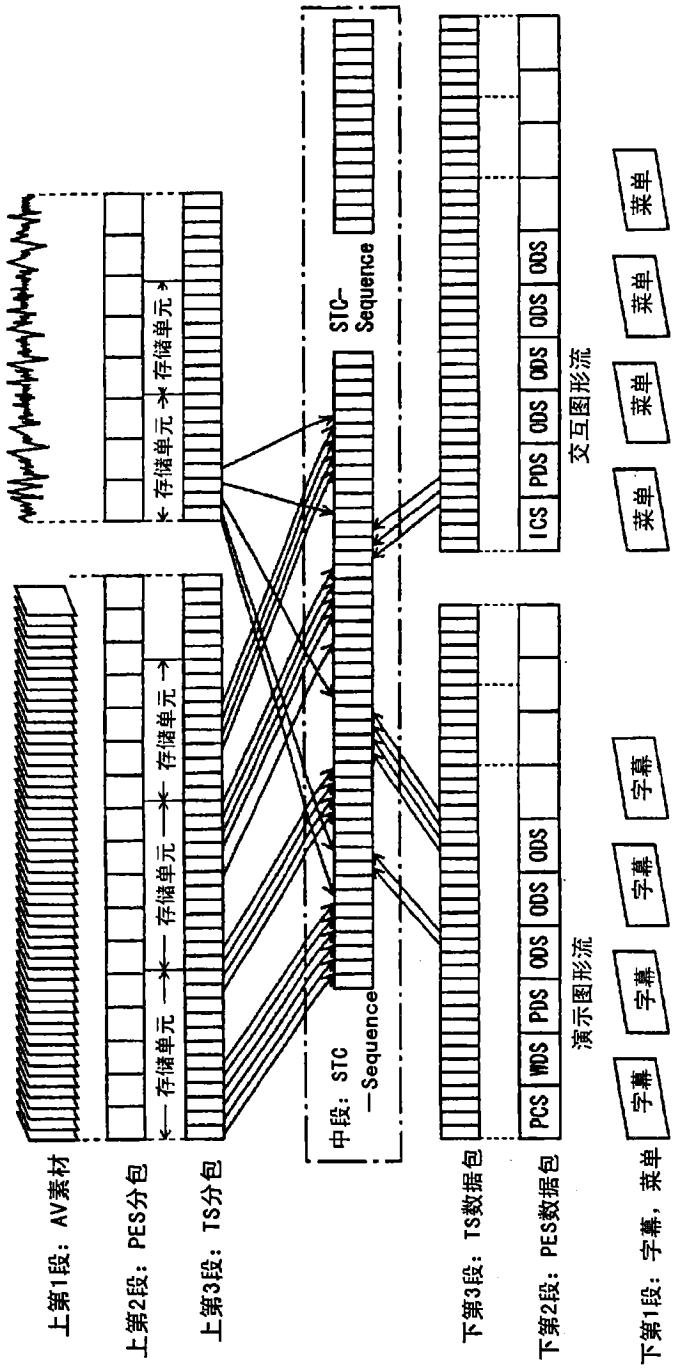

上第1段：AV素材
上第2段：PES分包
上第3段：TS分包
存储单元
存储单元
存储单元
存储单元
中段：STC
—Sequence
STC-
Sequence
下第3段：TS数据包
下第2段：PES数据包
PCS
WDS
PDS
ODS
ODS
ODS
演示图形流
ICS
PDS
ODS
ODS
ODS
ODS
交互图形流
下第1段：字幕，菜单
字幕
字幕
字幕
字幕
菜单
菜单
菜单
菜单

图3

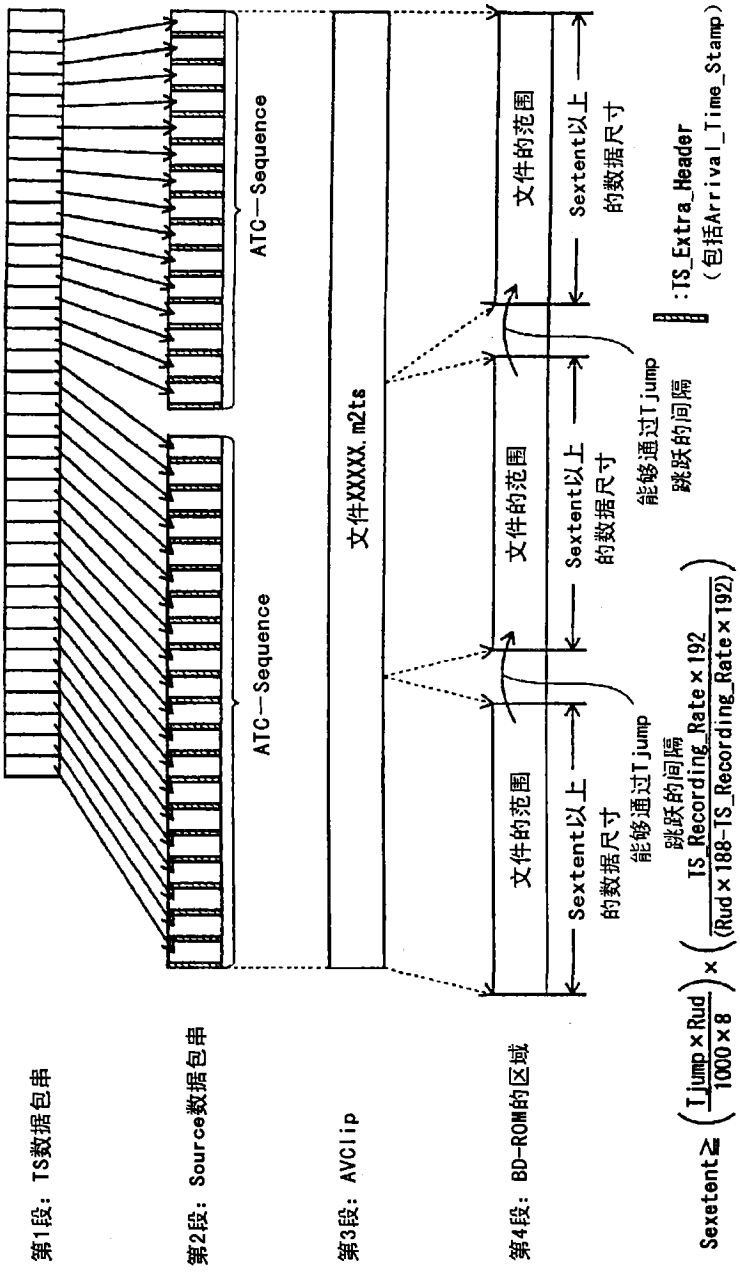

第1段：TS数据包串
第2段：Source数据包串
第3段：AVClip
第4段：BD-ROM的区域
ATC—Sequence
ATC—Sequence
文件XXXXX.m2ts
文件的范围
文件的范围
文件的范围
Sextent以上
的数据尺寸
Sextent以上
的数据尺寸
Sextent以上
的数据尺寸
能够通过Tjump
跳跃的间隔
能够通过Tjump
跳跃的间隔
:TS_Extra_Header
（包括Arrival_Time_Stamp）
$Sextent \geq \left(\frac{Tjump \times Rud}{1000 \times 8}\right) \times \left(\frac{TS\_Recording\_Rate \times 192}{(Rud \times 188 - TS\_Recording\_Rate \times 192)}\right)$

图4

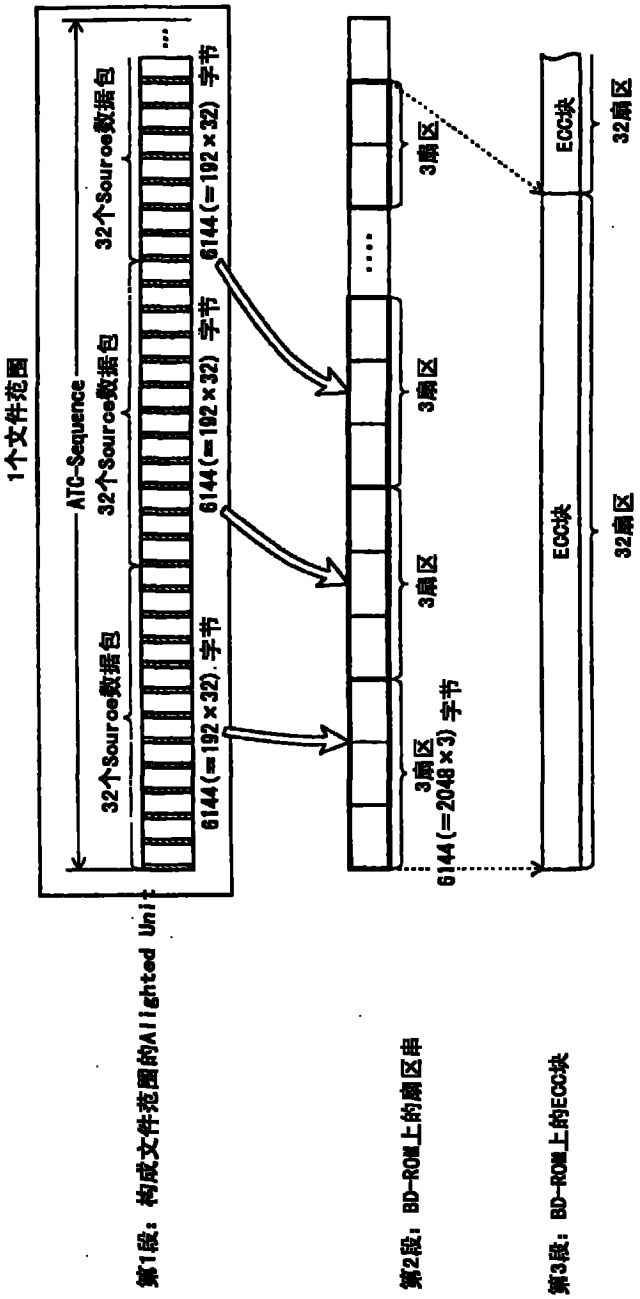

1个文件范围
ATC-Sequence
32个Source数据包
32个Source数据包
32个Source数据包
6144(=192×32) 字节
6144(=192×32) 字节
6144(=192×32) 字节
第1段：构成文件范围的Aligned Unit
3扇区
3扇区
3扇区
3扇区
6144(=2048×3) 字节
第2段：BD-ROM上的扇区串
ECC块
ECC块
32扇区
32扇区
第3段：BD-ROM上的ECC块

图5

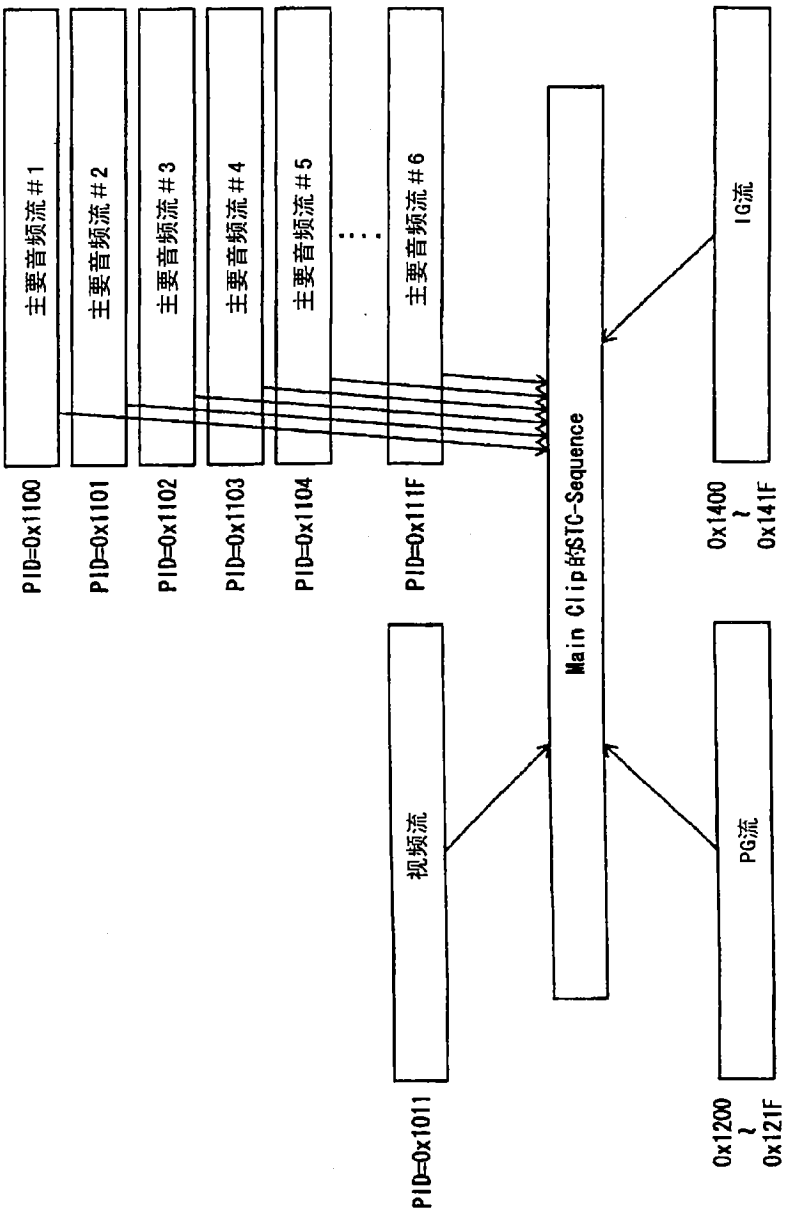

主要音频流#1
主要音频流#2
主要音频流#3
主要音频流#4
主要音频流#5
主要音频流#6
PID=0x1100
PID=0x1101
PID=0x1102
PID=0x1103
PID=0x1104
PID=0x111F
视频流
PID=0x1011
Main Clip的STC-Sequence
IG流
0x1400
~
0x141F
PG流
0x1200
~
0x121F

图6

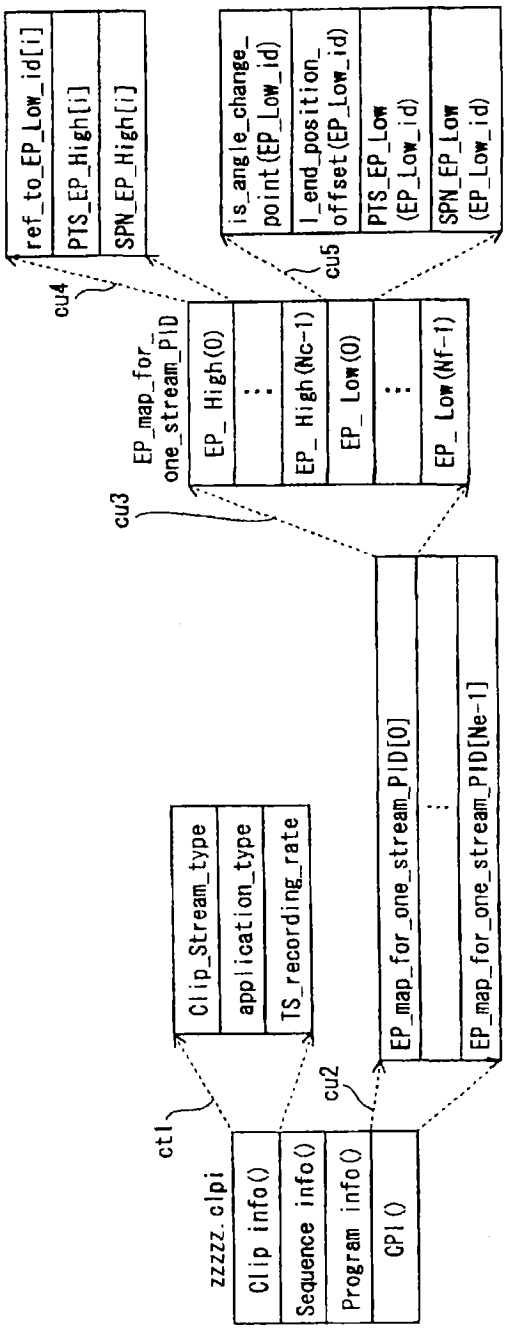

ref_to_EP_Low_id[i]
PTS_EP_High[i]
SPN_EP_High[i]
is_angle_change_point(EP_Low_id)
I_end_position_offset(EP_Low_id)
PTS_EP_Low (EP_Low_id)
SPN_EP_Low (EP_Low_id)
cu4
cu5
EP_map_for_one_stream_PID
EP_High(0)
EP_High(Nc-1)
EP_Low(0)
EP_Low(Nf-1)
cu3
Clip_Stream_type
application_type
TS_recording_rate
EP_map_for_one_stream_PID[0]
EP_map_for_one_stream_PID[Ne-1]
cu2
ctl
zzzzz.clpi
Clip info()
Sequence info()
Program info()
CPI()

图7

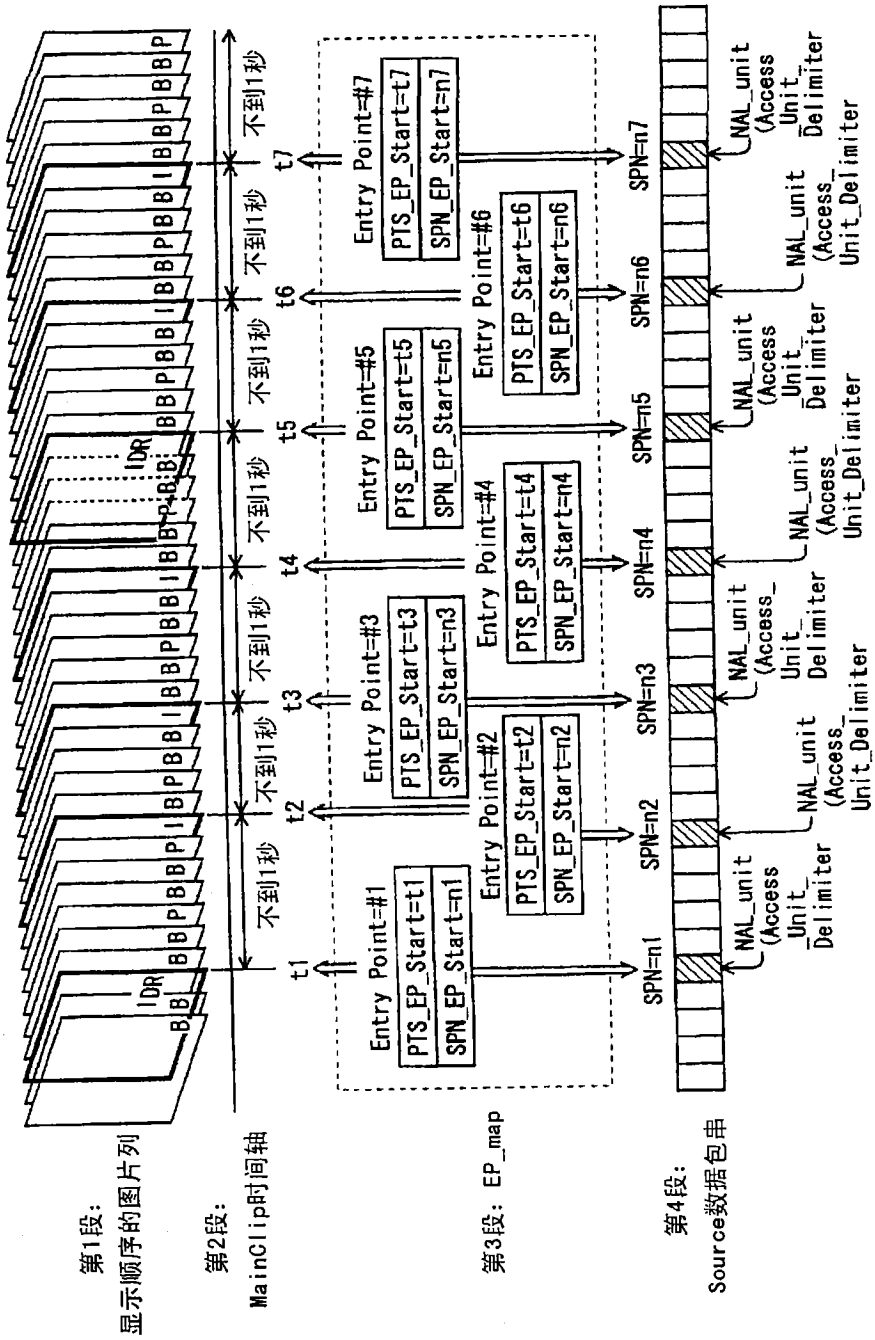

第1段:
显示顺序的图片列
第2段:
MainClip时间轴
第3段: EP_map
第4段:
Source数据包串
IDR
B B P B B P B B P I B B P B B I B B P B B I B B P B B P B B P
不到1秒
t1
t2
t3
t4
t5
t6
t7
Entry Point=#1
PTS_EP_Start=t1
SPN_EP_Start=n1
Entry Point=#2
PTS_EP_Start=t2
SPN_EP_Start=n2
Entry Point=#3
PTS_EP_Start=t3
SPN_EP_Start=n3
Entry Point=#4
PTS_EP_Start=t4
SPN_EP_Start=n4
Entry Point=#5
PTS_EP_Start=t5
SPN_EP_Start=n5
Entry Point=#6
PTS_EP_Start=t6
SPN_EP_Start=n6
Entry Point=#7
PTS_EP_Start=t7
SPN_EP_Start=n7
SPN=n1
SPN=n2
SPN=n3
SPN=n4
SPN=n5
SPN=n6
SPN=n7
NAL_unit (Access_Unit_Delimiter
NAL_unit (Access_Unit_Delimiter

图8

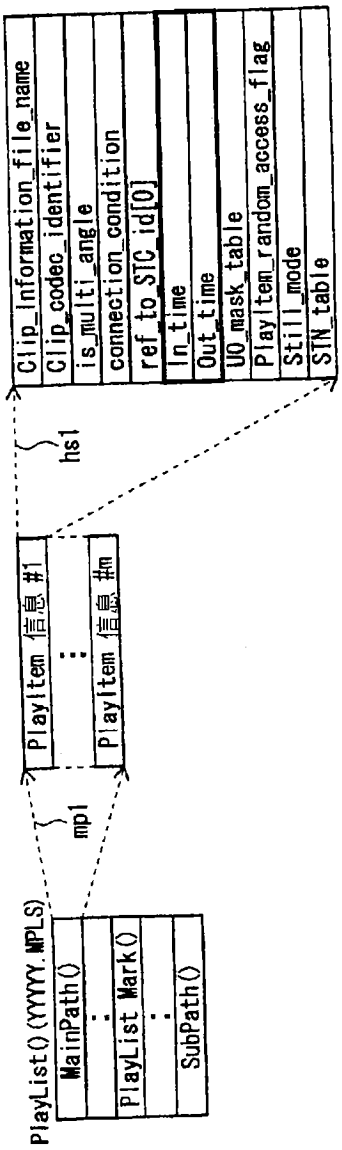

PlayList() (YYYY. MPLS)
MainPath()
PlayList Mark()
SubPath()
mpl
PlayItem 信息 #1
PlayItem 信息 #m
hs1
Clip_Information_file_name
Clip_codec_identifier
is_multi_angle
connection_condition
ref_to_STC_id[0]
In_time
Out_time
UO_mask_table
PlayItem_random_access_flag
Still_mode
STN_table

图9

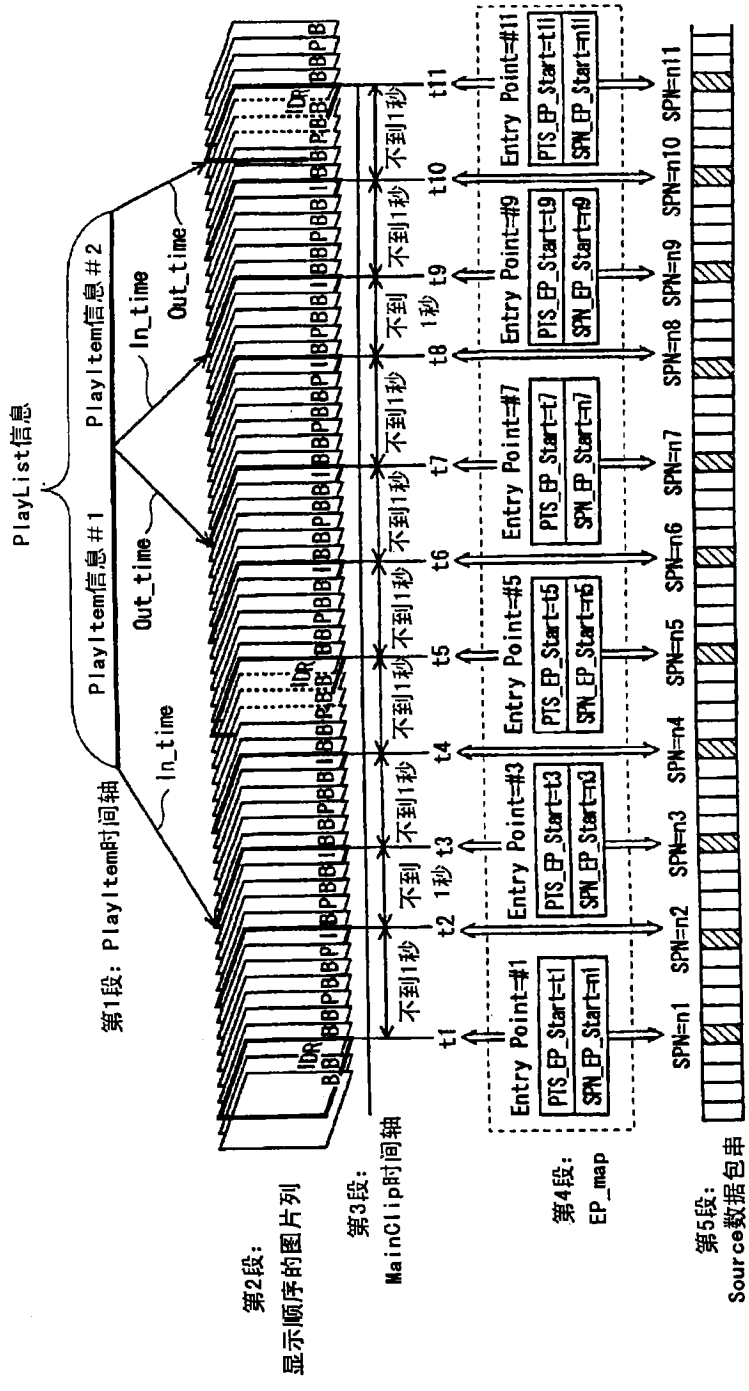

PlayList信息
第1段：PlayItem时间轴
PlayItem信息#1
PlayItem信息#2
In_time
Out_time
In_time
Out_time
第2段：
显示顺序的图片列
第3段：
MainClip时间轴
不到1秒
t1
t2
t3
t4
t5
t6
t7
t8
t9
t10
t11
第4段：
EP_map
Entry Point=#1
PTS_EP_Start=t1
SPN_EP_Start=n1
Entry Point=#3
PTS_EP_Start=t3
SPN_EP_Start=n3
Entry Point=#5
PTS_EP_Start=t5
SPN_EP_Start=n6
Entry Point=#7
PTS_EP_Start=t7
SPN_EP_Start=n7
Entry Point=#9
PTS_EP_Start=t9
SPN_EP_Start=n9
Entry Point=#11
PTS_EP_Start=t11
SPN_EP_Start=n11
第5段：
Source数据包串
SPN=n1
SPN=n2
SPN=n3
SPN=n4
SPN=n5
SPN=n6
SPN=n7
SPN=n8
SPN=n9
SPN=n10
SPN=n11

图10

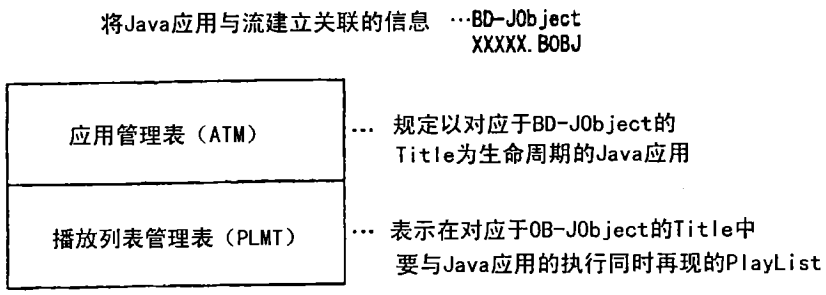

将Java应用与流建立关联的信息 …BD-JObject
XXXXX.BOBJ
应用管理表（ATM）
播放列表管理表（PLMT）
… 规定以对应于BD-JObject的Title为生命周期的Java应用
… 表示在对应于OB-JObject的Title中要与Java应用的执行同时再现的PlayList

图 11

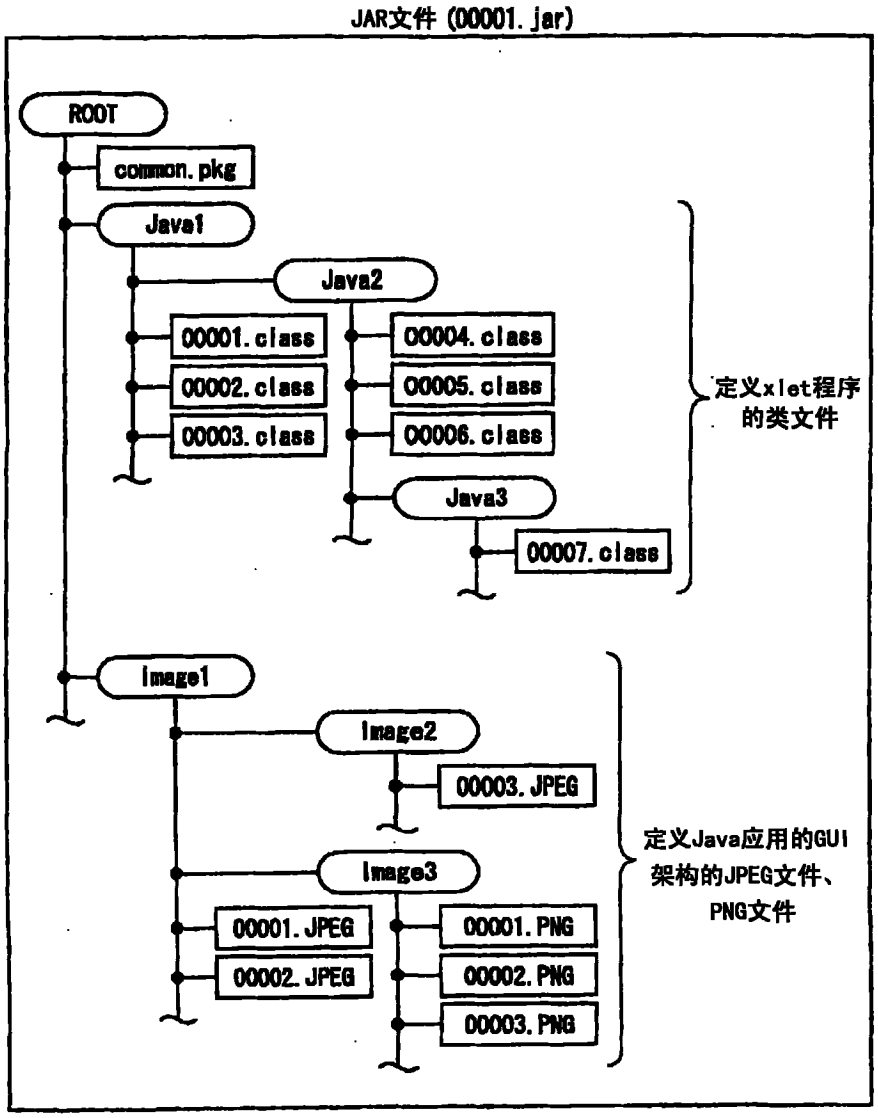

JAR文件 (00001.jar)
ROOT
common.pkg
Java1
Java2
00001.class
00002.class
00003.class
00004.class
00005.class
00006.class
Java3
00007.class
定义xlet程序的类文件
Image1
Image2
00003.JPEG
Image3
00001.JPEG
00002.JPEG
00001.PNG
00002.PNG
00003.PNG
定义Java应用的GUI架构的JPEG文件、PNG文件

图 12

(a) 应用管理表（ATM）

life_cycle
apli_id_ref
run_attribute
run_priority

(b) 应用管理表（ATM）

life_cycle	apli_id_ref	run_attribate	run_priority
利用title，PL，Chapter表现的应用的"生命周期"	赋予给JAR的文件名的5位的整数ZZZZZ即"应用ID"的参照值	· Auto Run · Present （无指定） · Suspend 的某一个的"启动属性"	取0～255的值的"启动优先级"

图13

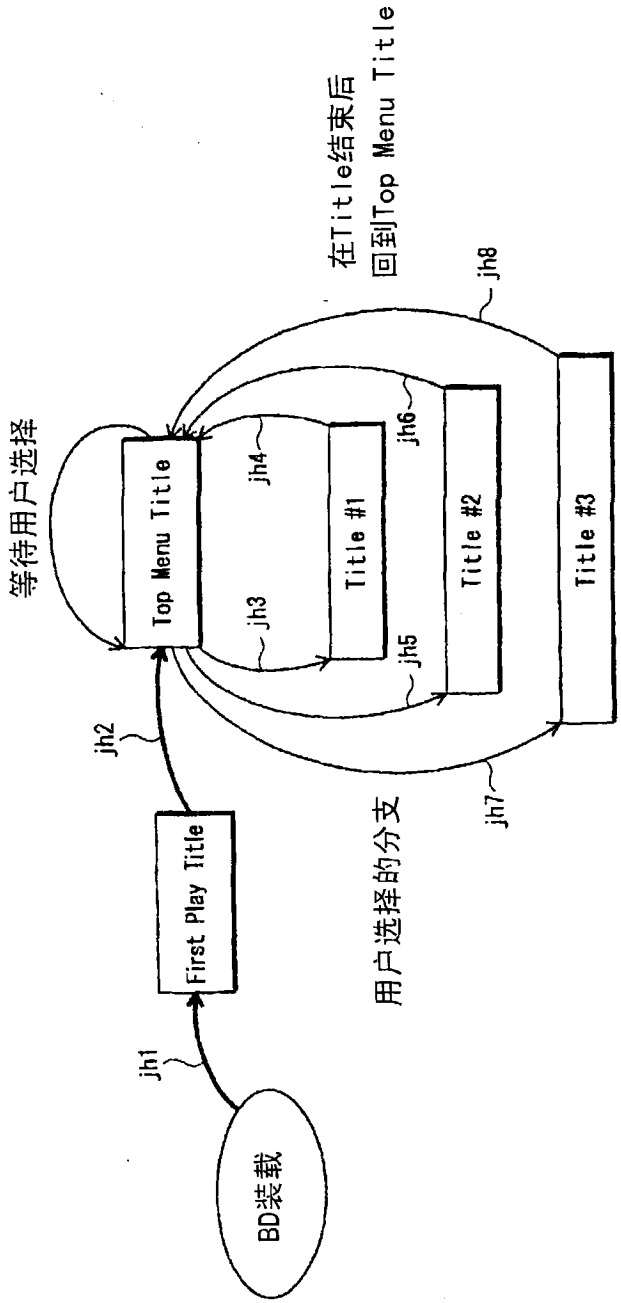

BD装载
jh1
First Play Title
jh2
Top Menu Title
等待用户选择
jh3
jh4
Title #1
jh5
Title #2
jh6
Title #3
jh7
jh8
用户选择的分支
在Title结束后
回到Top Menu Title

图14

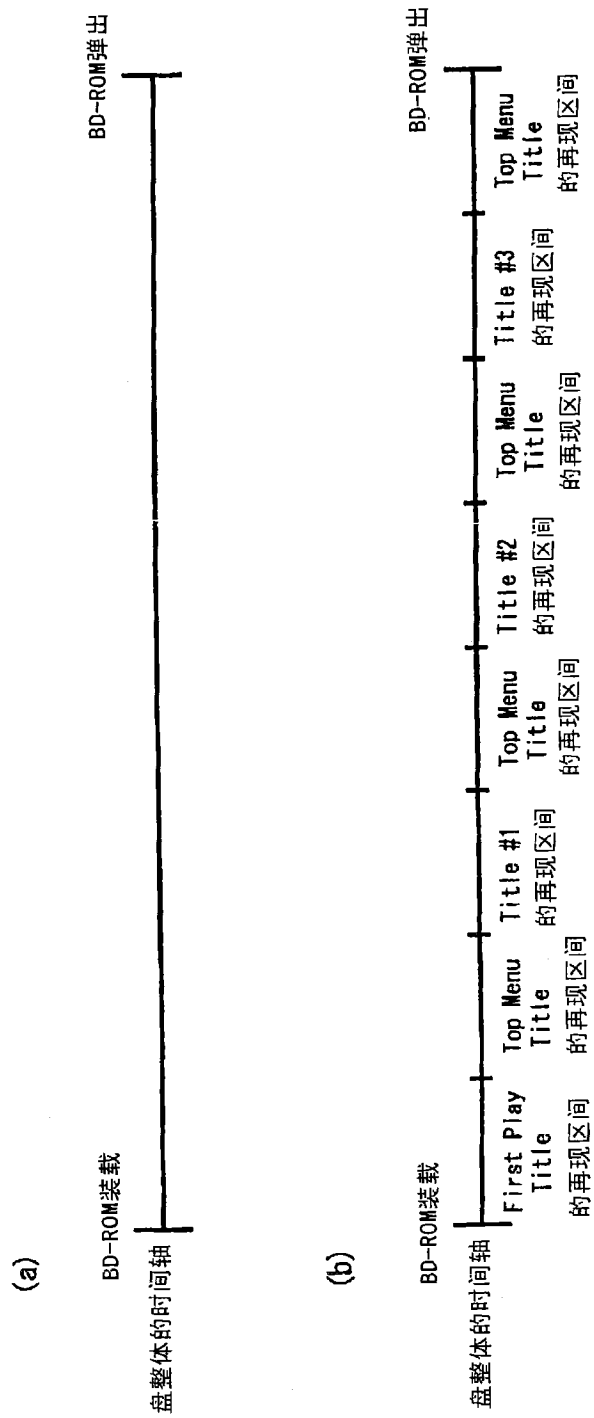

(a)
BD-ROM装载
BD-ROM弹出
盘整体的时间轴
(b)
BD-ROM装载
BD-ROM弹出
盘整体的时间轴
First Play Title 的再现区间
Top Menu Title 的再现区间
Title #1 的再现区间
Top Menu Title 的再现区间
Title #2 的再现区间
Top Menu Title 的再现区间
Title #3 的再现区间
Top Menu Title 的再现区间

图15

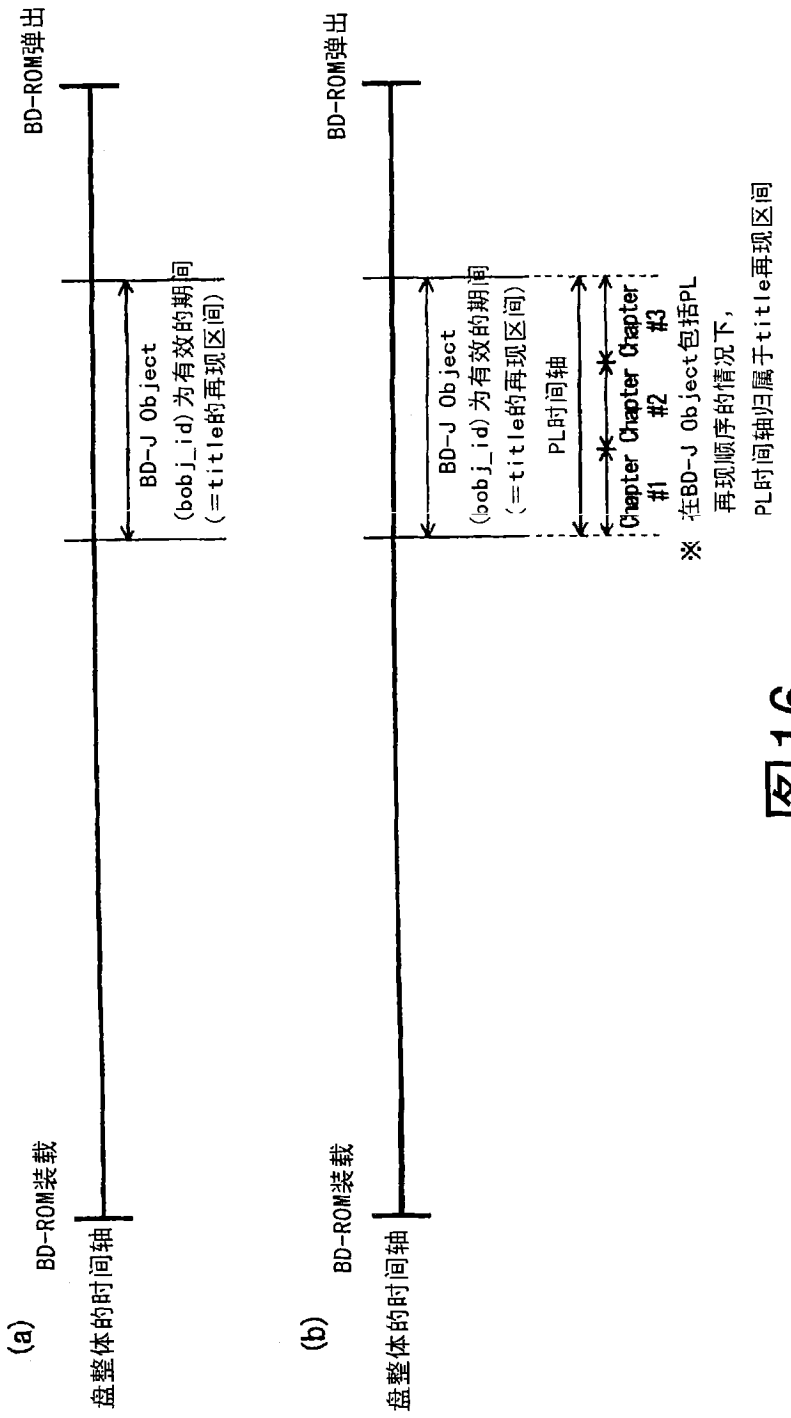

(a)
盘整体的时间轴
BD-ROM装载
BD-ROM弹出
BD-J Object
(bobj_id)为有效的期间
(=title的再现区间)
(b)
盘整体的时间轴
BD-ROM装载
BD-ROM弹出
BD-J Object
(bobj_id)为有效的期间
(=title的再现区间)
PL时间轴
Chapter #1
Chapter #2
Chapter #3
※在BD-J Object包括PL
再现顺序的情况下,
PL时间轴归属于title再现区间

图16

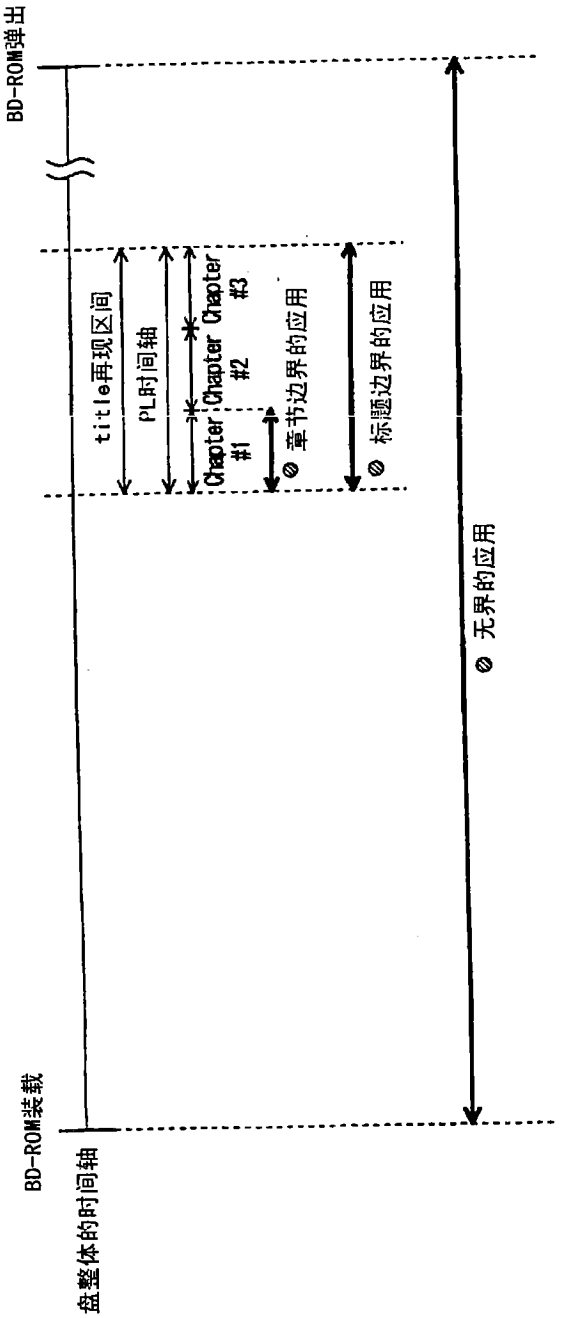

BD-ROM弹出
title再现区间
PL时间轴
Chapter #1
Chapter #2
Chapter #3
章节边界的应用
标题边界的应用
无界的应用
BD-ROM装载
盘整体的时间轴

图17

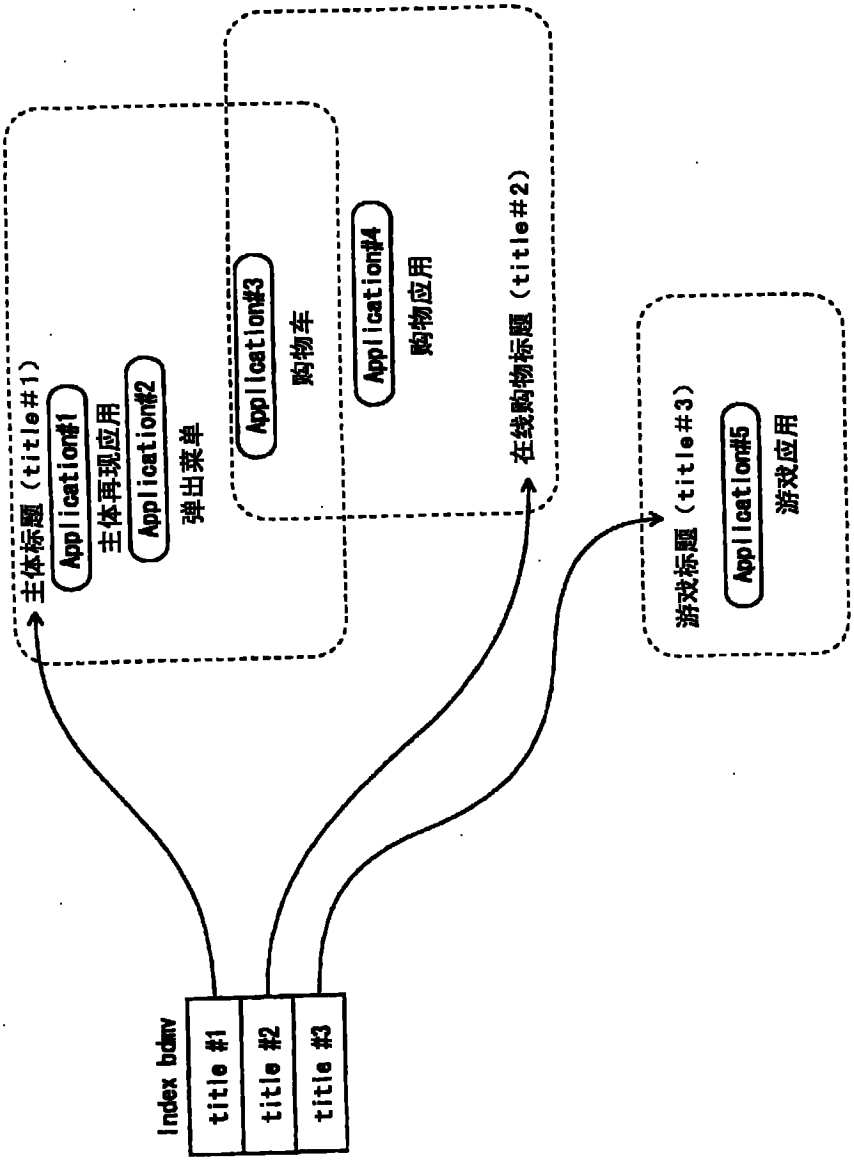

主体标题（title#1）
Application#1
主体再现应用
Application#2
弹出菜单
Application#3
购物车
Application#4
购物应用
在线购物标题（title#2）
游戏标题（title#3）
Application#5
游戏应用
Index bdmv
title #1
title #2
title #3

图18

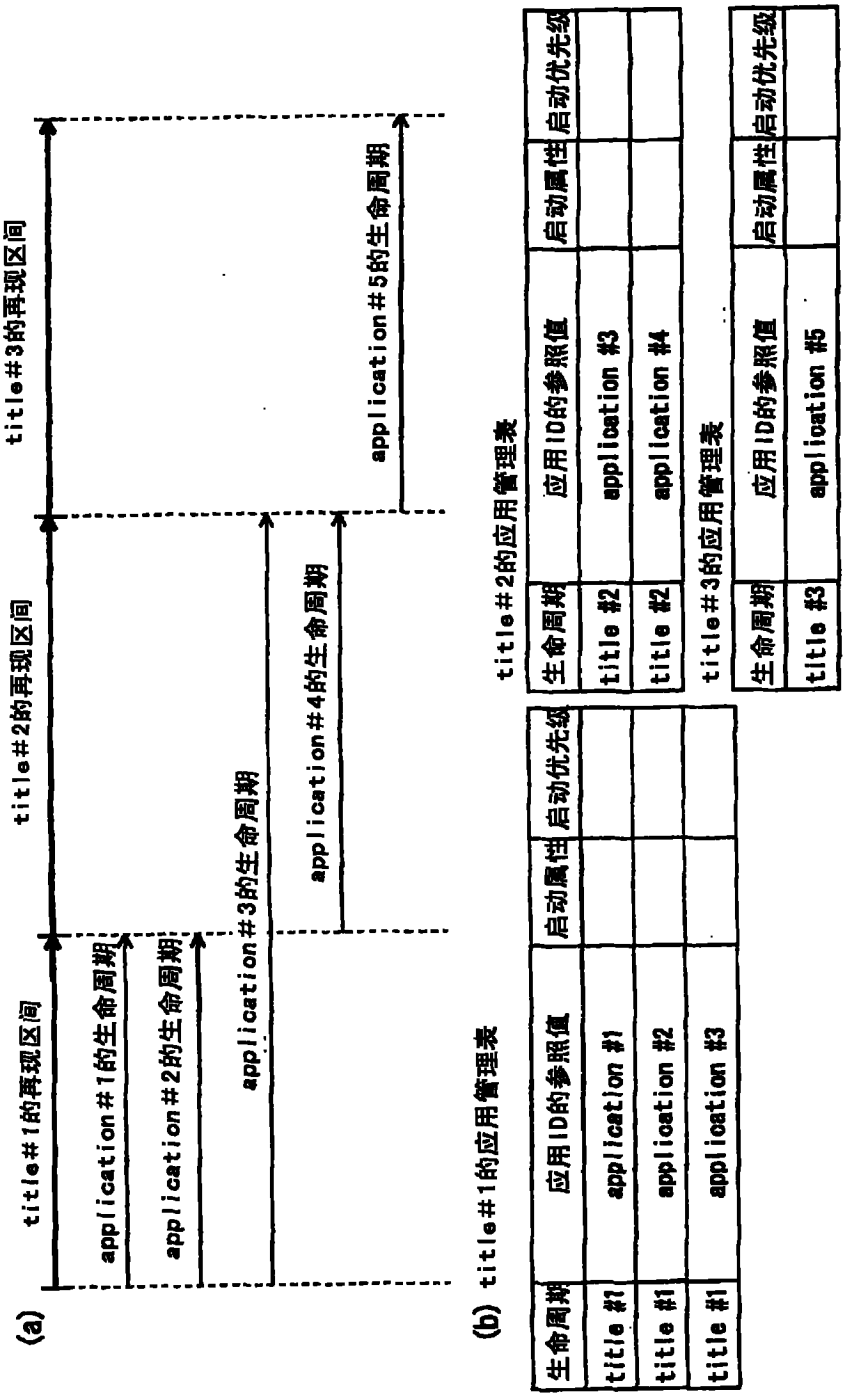

(a)
title#1的再现区间
title#2的再现区间
title#3的再现区间
application#1的生命周期
application#2的生命周期
application#3的生命周期
application#4的生命周期
application#5的生命周期
(b) title#1的应用管理表
生命周期
应用ID的参照值
启动属性
启动优先级
title #1
application #1
title #1
application #2
title #1
application #3
title#2的应用管理表
生命周期
应用ID的参照值
启动属性
启动优先级
title #2
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
启动属性
启动优先级
title #3
application #5

图19

启动属性带来的应用状态的变化

前面标题中的应用的状态	启动属性		
	Present	AutoRun	Suspend
非启动	什么都不做而状态继续	什么都不做而状态继续	什么都不做而状态继续
启动中	什么都不做而状态继续	什么都不做而状态继续	中止
Suspend	再继续	再继续	什么都不做而状态继续

图20

(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位的数值即"Play List ID"	· Auto Play · Present 无指定）的哪一个的"再现属性"

图 21

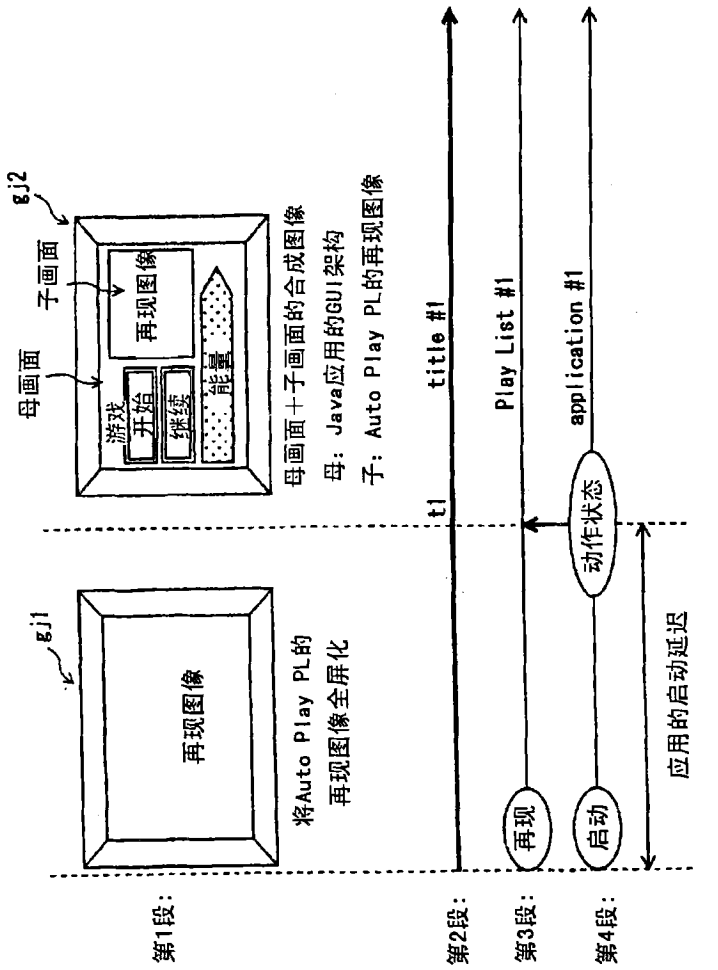

gj2
子画面
母画面
再现图像
游戏
开始
继续
能量
母画面＋子画面的合成图像
母：Java应用的GUI架构
子：Auto Play PL的再现图像
gj1
再现图像
将Auto Play PL的
再现图像全屏化
t1
title #1
Play List #1
application #1
动作状态
再现
启动
应用的启动延迟
第1段：
第2段：
第3段：
第4段：

图22

	在分支目的地title中没有PLMT	在分支目的地title中有PLMT	
		再现属性：Auto Play	再现属性：Present
分支源title为再现中状态	再现停止	状态继续	状态继续
分支源title为非再现状态	状态继续	自动再现开始	状态继续

图23

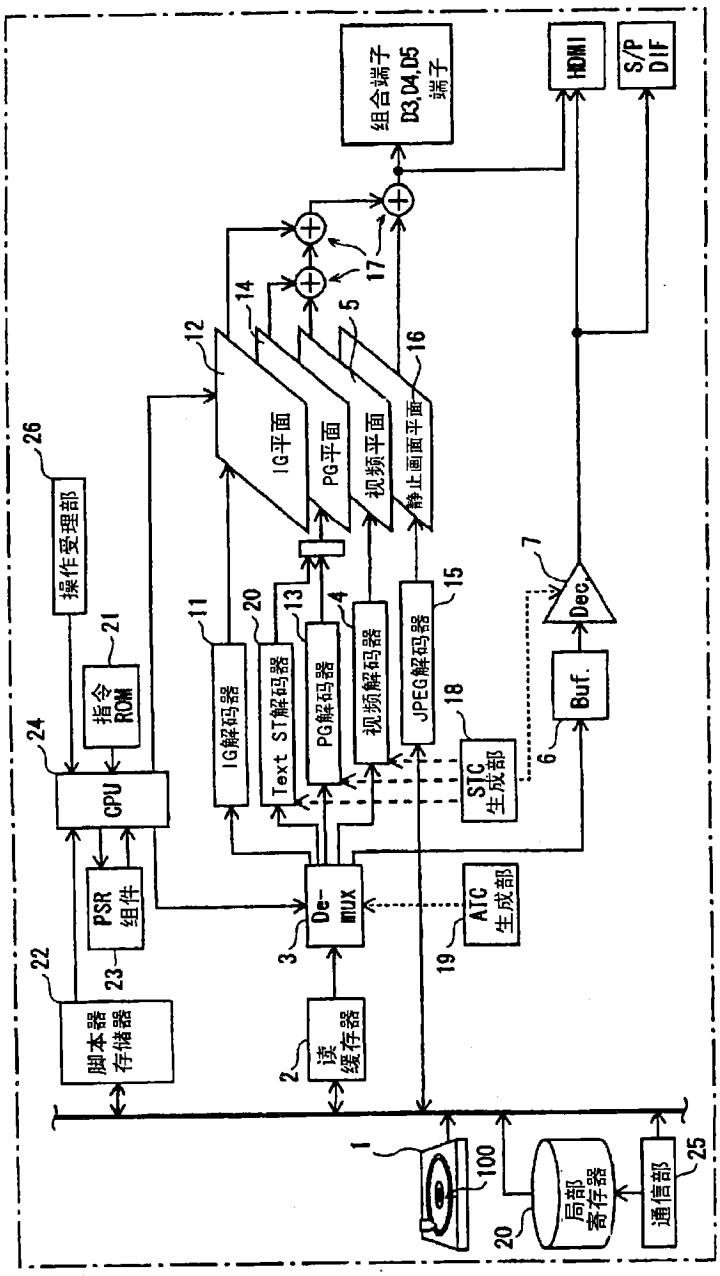

组合端子
D3,D4,D5
端子
HDMI
S/P
DIF
操作受理部
指令
ROM
CPU
PSR
组件
脚本器
存储器
读
缓存器
De-
MUX
IG解码器
Text ST解码器
PG解码器
视频解码器
JPEG解码器
STC
生成部
ATC
生成部
IG平面
PG平面
视频平面
静止画面平面
Dec
Buf.
局部
寄存器
通信部
1
2
3
4
5
6
7
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
100

图24

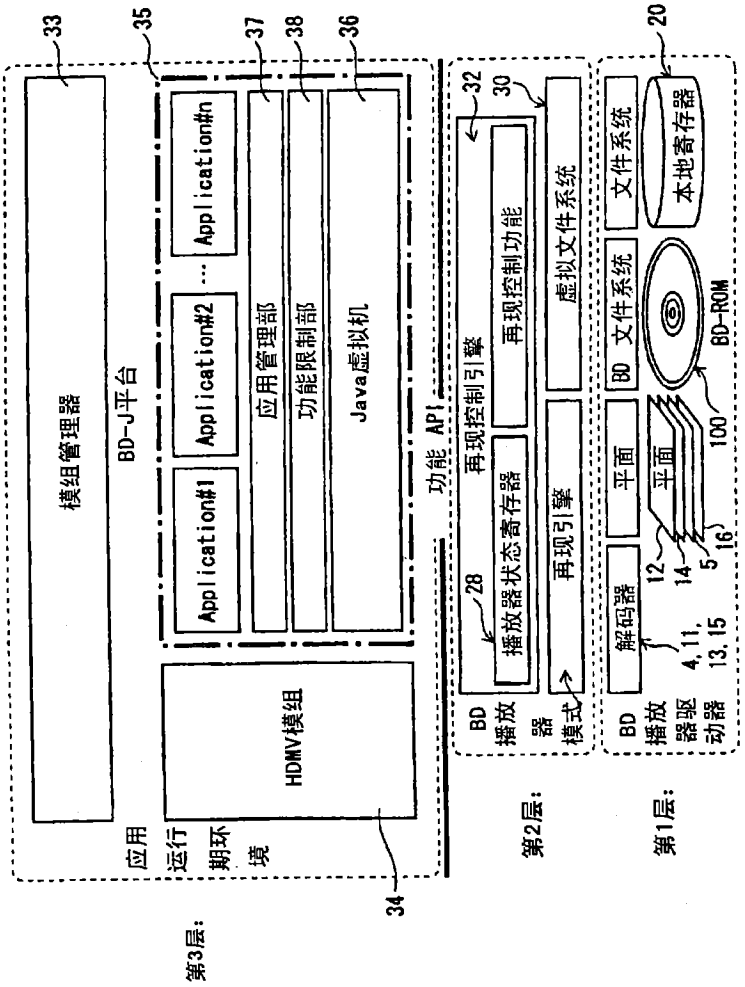

第3层：
应用运行期环境
模组管理器
BD-J平台
HDMV模组
Application#1
Application#2
Application#n
应用管理部
功能限制部
Java虚拟机
功能 API
34
33
35
37
38
36
第2层：
BD播放器模式
播放器状态寄存器
再现引擎
再现控制引擎
再现控制功能
虚拟文件系统
28
32
30
第1层：
BD播放器驱动器
解码器
平面
平面
BD 文件系统
文件系统
本地寄存器
BD-ROM
12
14
5
16
4, 11, 13, 15
100
20

图25

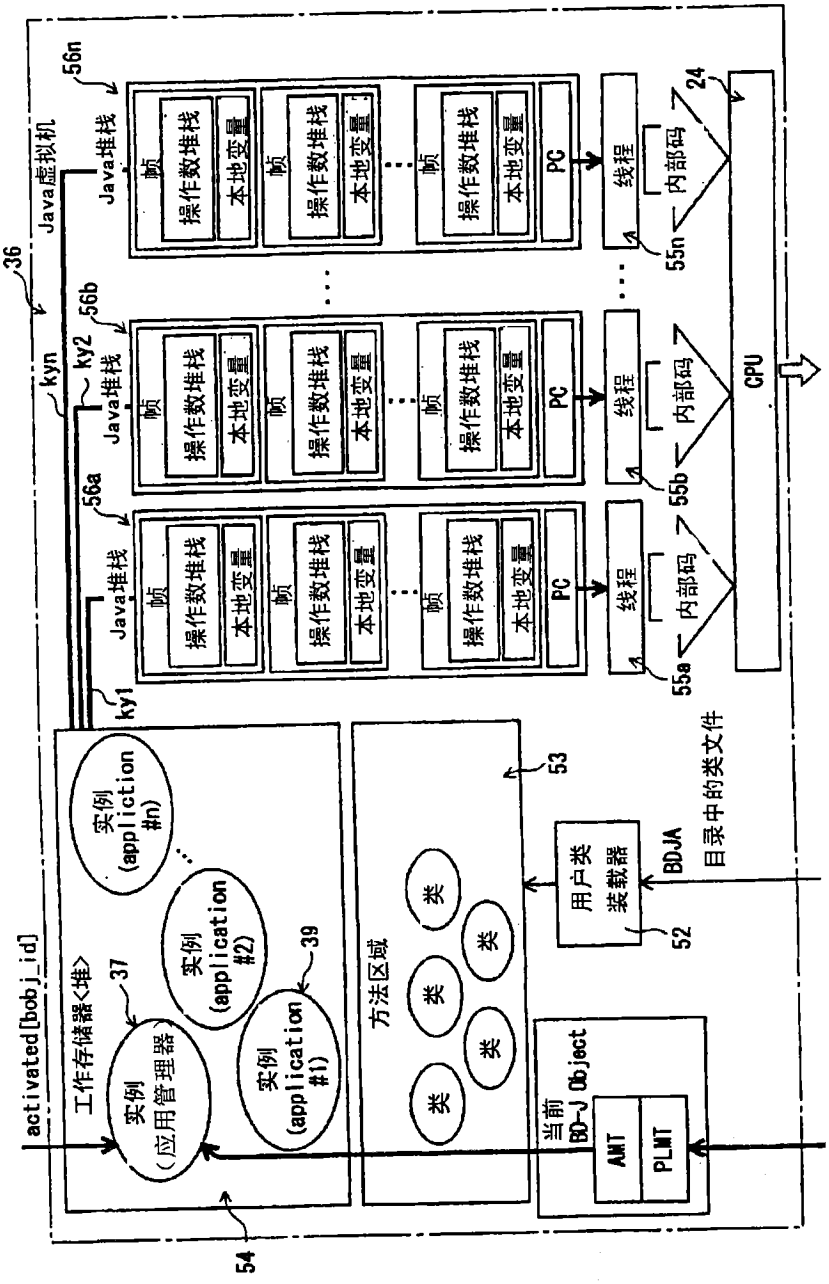

activated[bobj_id]
工作存储器<堆>
37
实例（应用管理器）
实例(application #1)
39
实例(application #2)
实例(application #n)
54
方法区域
类
53
用户类装载器
52
BDJA
目录中的类文件
当前BD-J Object
AMT
PLMT
36
Java虚拟机
kyn
ky1
ky2
56a
56b
56n
Java堆栈
帧
操作数堆栈
本地变量
PC
线程
55a
55b
55n
内部码
24
CPU

图26

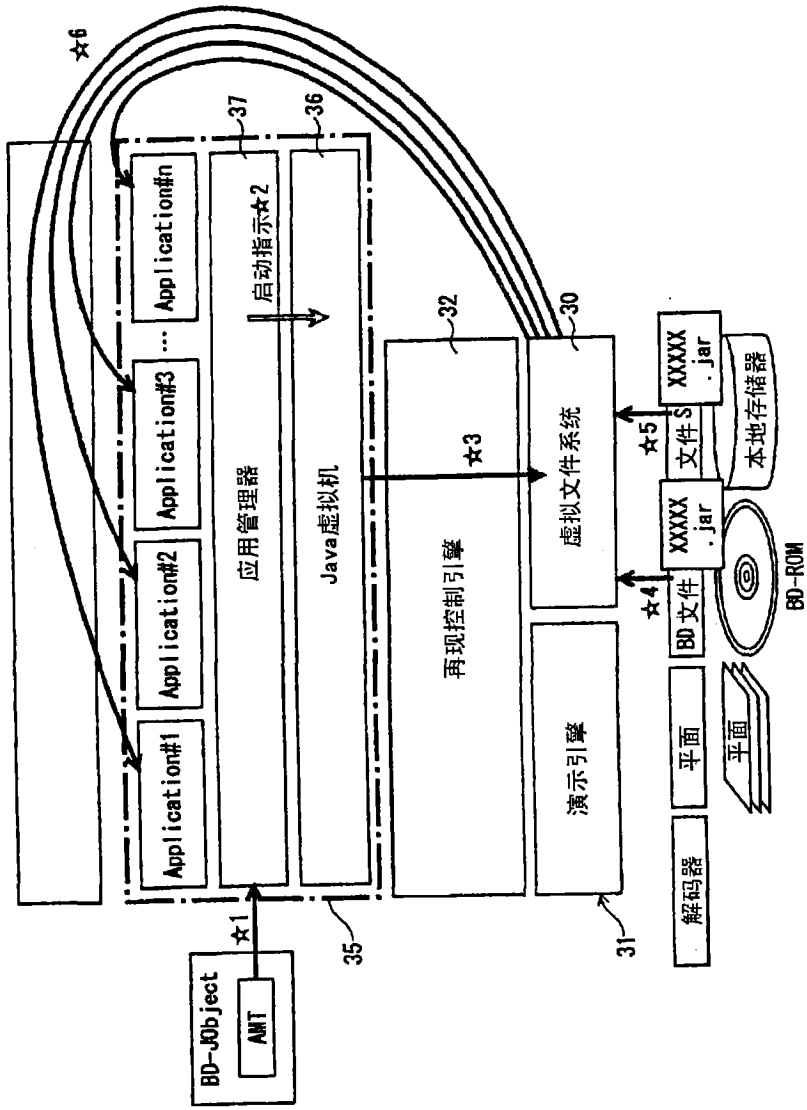

☆6
37
36
Application#n
...
Application#3
Application#2
Application#1
应用管理器
启动指示☆2
Java虚拟机
32
☆3
再现控制引擎
30
虚拟文件系统
演示引擎
31
35
☆1
BD-JObject
AMT
☆5
☆4
XXXXX.jar
文件S
XXXXX.jar
BD文件
本地存储器
BD-ROM
平面
平面
解码器

图27

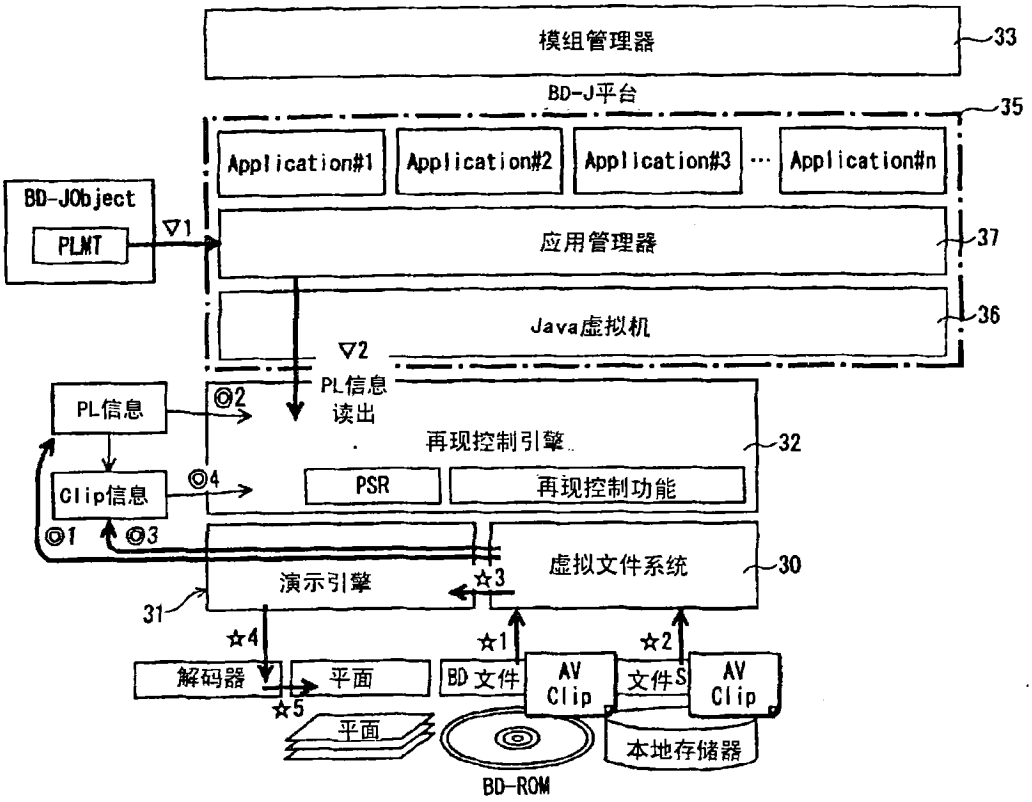

模组管理器
33
BD-J平台
35
Application#1
Application#2
Application#3
Application#n
BD-JObject
PLMT
▽1
应用管理器
37
Java虚拟机
36
▽2
PL信息
读出
PL信息
◎2
再现控制引擎
32
Clip信息
◎4
PSR
再现控制功能
◎1
◎3
虚拟文件系统
30
演示引擎
☆3
31
☆4
☆1
☆2
解码器
平面
BD文件
AV Clip
文件S
AV Clip
☆5
平面
本地存储器
BD-ROM

图28

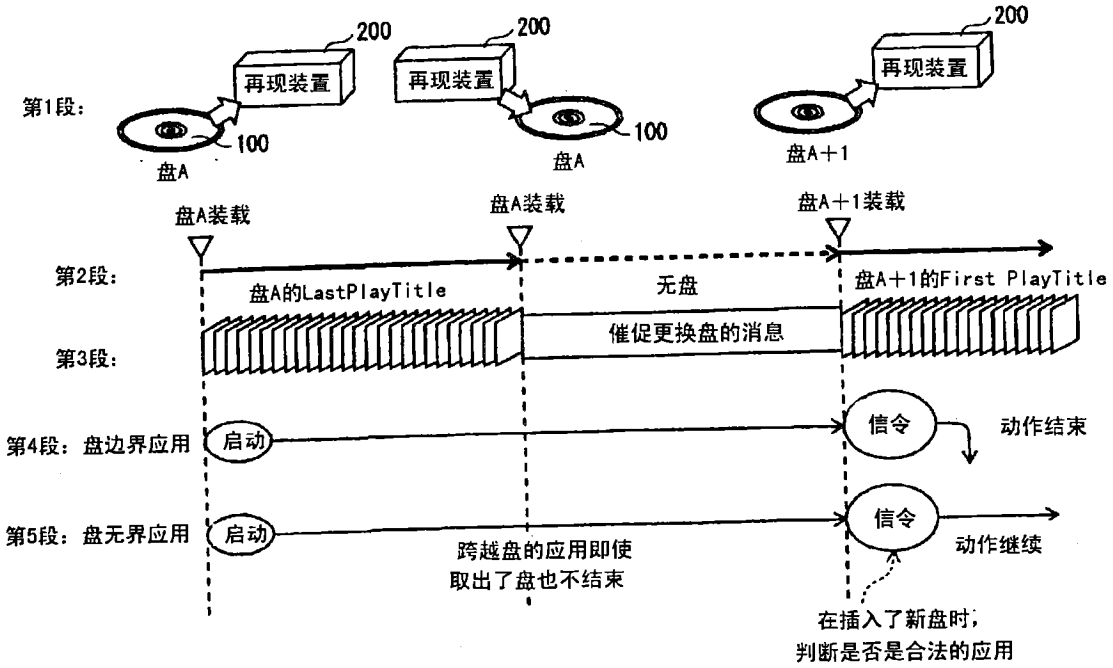

第1段：
200
再现装置
100
盘A
200
再现装置
100
盘A
200
再现装置
盘A+1
盘A装载
盘A装载
盘A+1装载
第2段：
盘A的LastPlayTitle
无盘
盘A+1的First PlayTitle
第3段：
催促更换盘的消息
第4段：盘边界应用
启动
信令
动作结束
第5段：盘无界应用
启动
信令
动作继续
跨越盘的应用即使
取出了盘也不结束
在插入了新盘时，
判断是否是合法的应用

图29

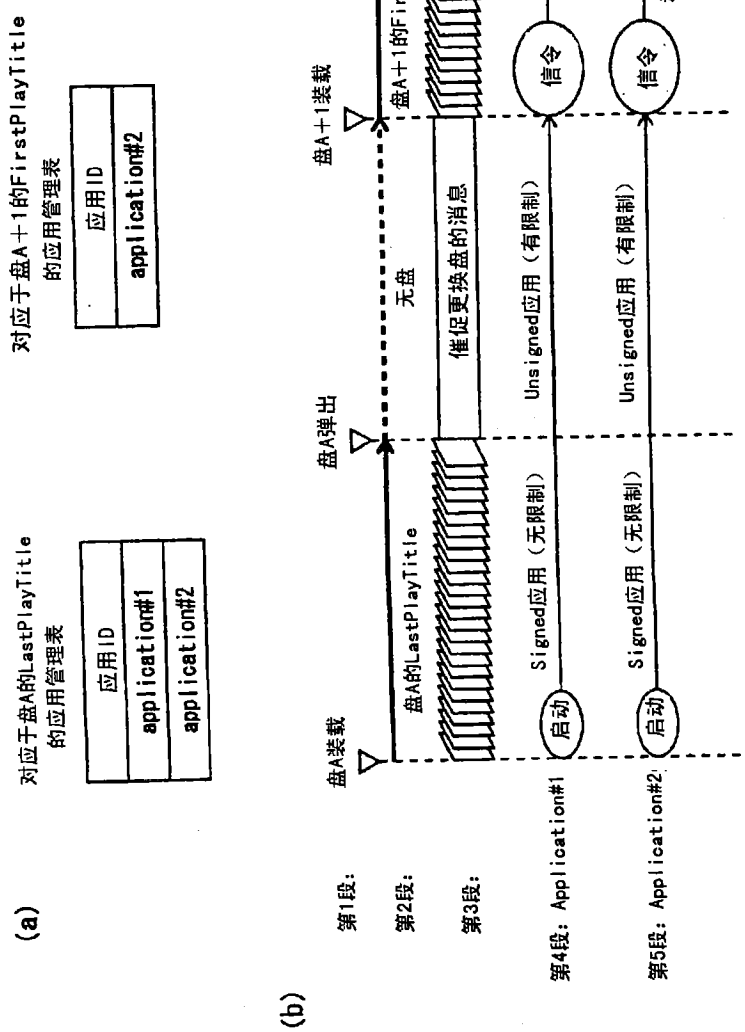

(a)
对应于盘A的LastPlayTitle的应用管理表
应用ID
application#1
application#2
对应于盘A+1的FirstPlayTitle的应用管理表
应用ID
application#2
(b)
第1段:
第2段:
第3段:
第4段: Application#1
第5段: Application#2
盘A装载
盘A的LastPlayTitle
盘A弹出
无盘
催促更换盘的消息
盘A+1装载
盘A+1的First PlayTitle
启动
Signed应用（无限制）
Unsigned应用（有限制）
信令
动作结束
启动
Signed应用（无限制）
Unsigned应用（有限制）
信令
Signed应用
动作继续

图30

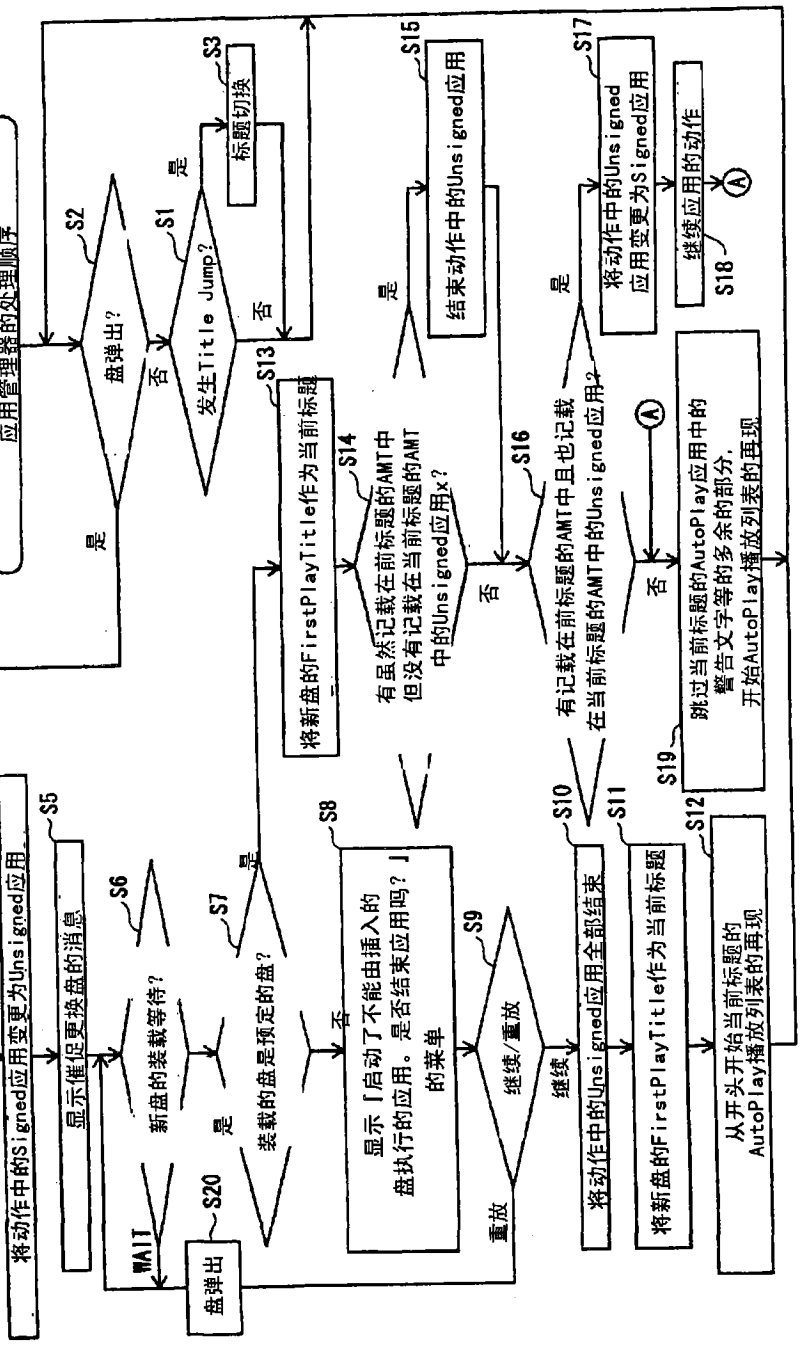

应用管理器的处理顺序
S1
发生Title Jump?
是
否
S2
盘弹出?
是
否
S3
标题切换
S4
将动作中的Unsigned应用变更为Unsigned应用
S5
显示催促更换盘的消息
S6
新盘的装载等待?
WAIT
S7
装载的盘是预定的盘?
是
否
S8
显示「启动了不能由插入的盘执行的应用。是否结束应用吗?」的菜单
S9
继续/重放
继续
重放
S10
将动作中的Unsigned应用全部结束
S11
将新盘的FirstPlayTitle作为当前标题
S12
从开头开始当前标题的AutoPlay播放列表的再现
S13
将新盘的FirstPlayTitle作为当前标题
S14
有虽然记载在前标题的AMT中但没有记载在当前标题的AMT中的Unsigned应用x?
是
否
S15
结束动作中的Unsigned应用
S16
有记载在前标题的AMT中且也记载在当前标题的AMT中的Unsigned应用2?
是
否
S17
将动作中的Unsigned应用变更为Signed应用
S18
继续应用的动作
A
S19
跳过当前标题的AutoPlay应用中的警告文字等的多余的部分,开始AutoPlay播放列表的再现
S20
盘弹出

图31

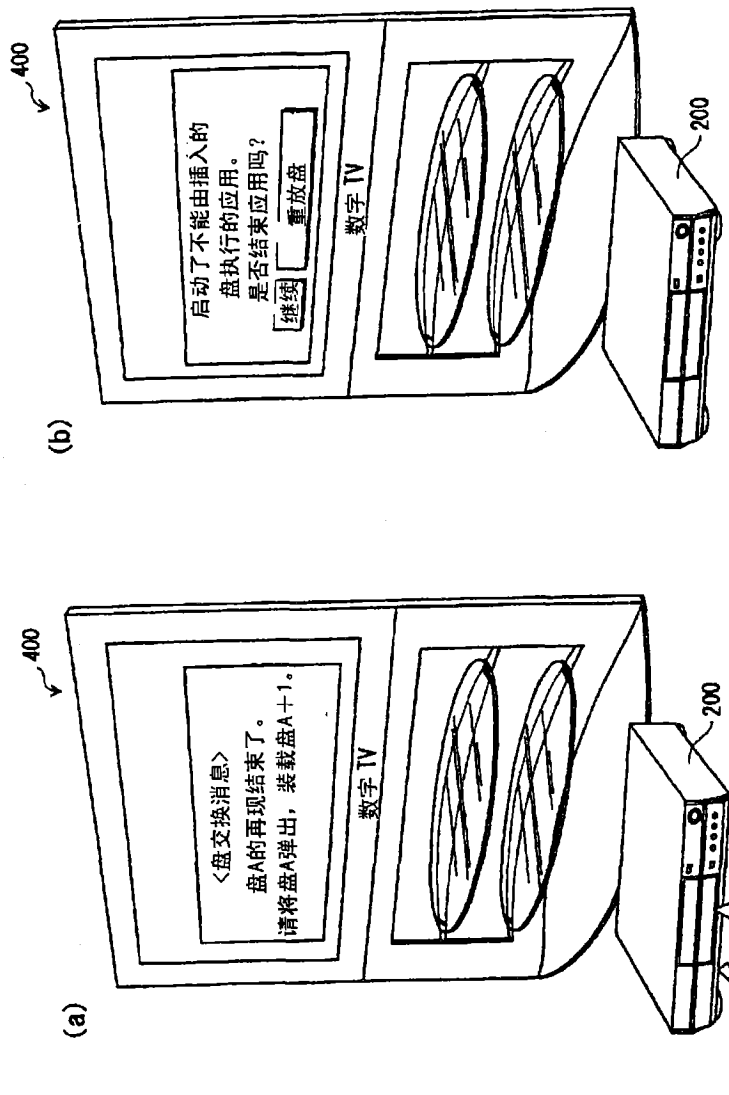

400
启动了不能由插入的
盘执行的应用。
是否结束应用吗?
继续
重放盘
数字 TV
200
(b)
400
<盘交换消息>
盘A的再现结束了。
请将盘A弹出，装载盘A+1。
数字 TV
200
盘B
盘A
(a)

图32

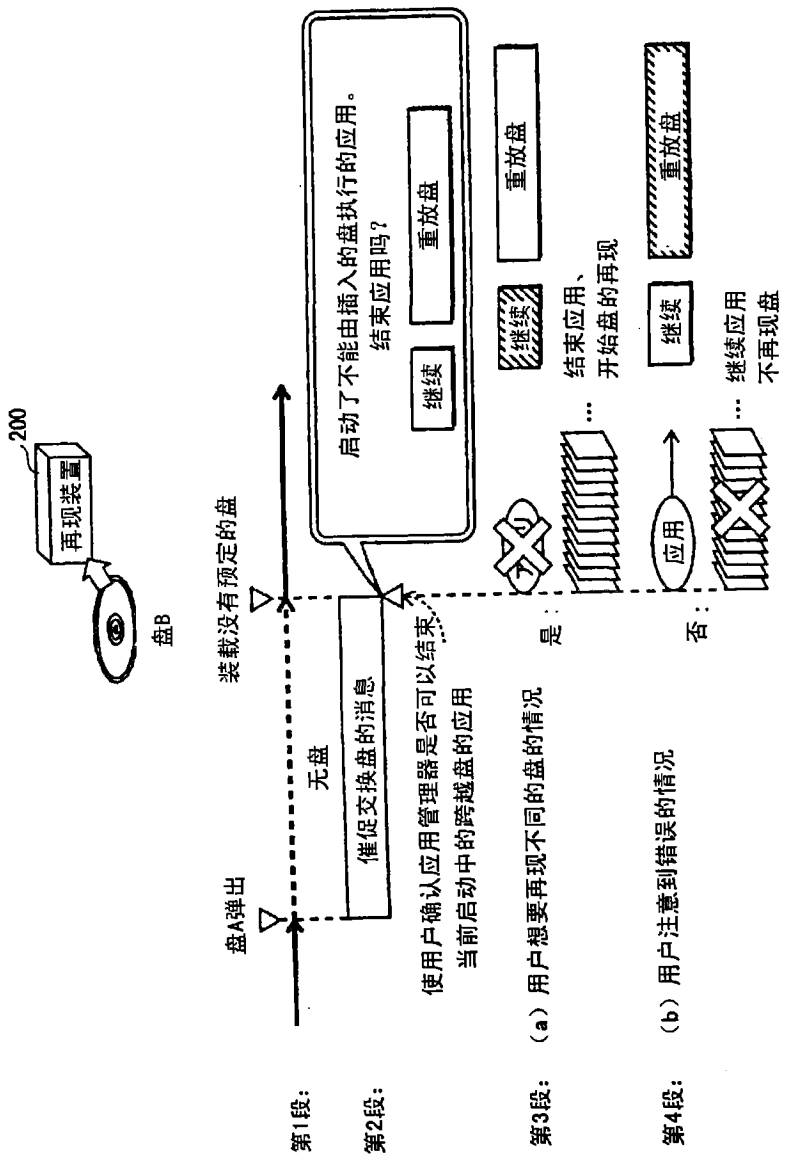

200
再现装置
盘B
装载没有预定的盘
盘A弹出
第1段:
无盘
催促交换盘的消息
第2段:
使用户确认应用管理器是否可以结束
当前启动中的跨越盘的应用
启动了不能由插入的盘执行的应用。
结束应用吗?
继续
重放盘
第3段: (a) 用户想要再现不同的盘的情况
是:
否:
继续
重放盘
结束应用、
开始盘的再现
第4段: (b) 用户注意到错误的情况
应用
继续
重放盘
继续应用
不再现盘

图33

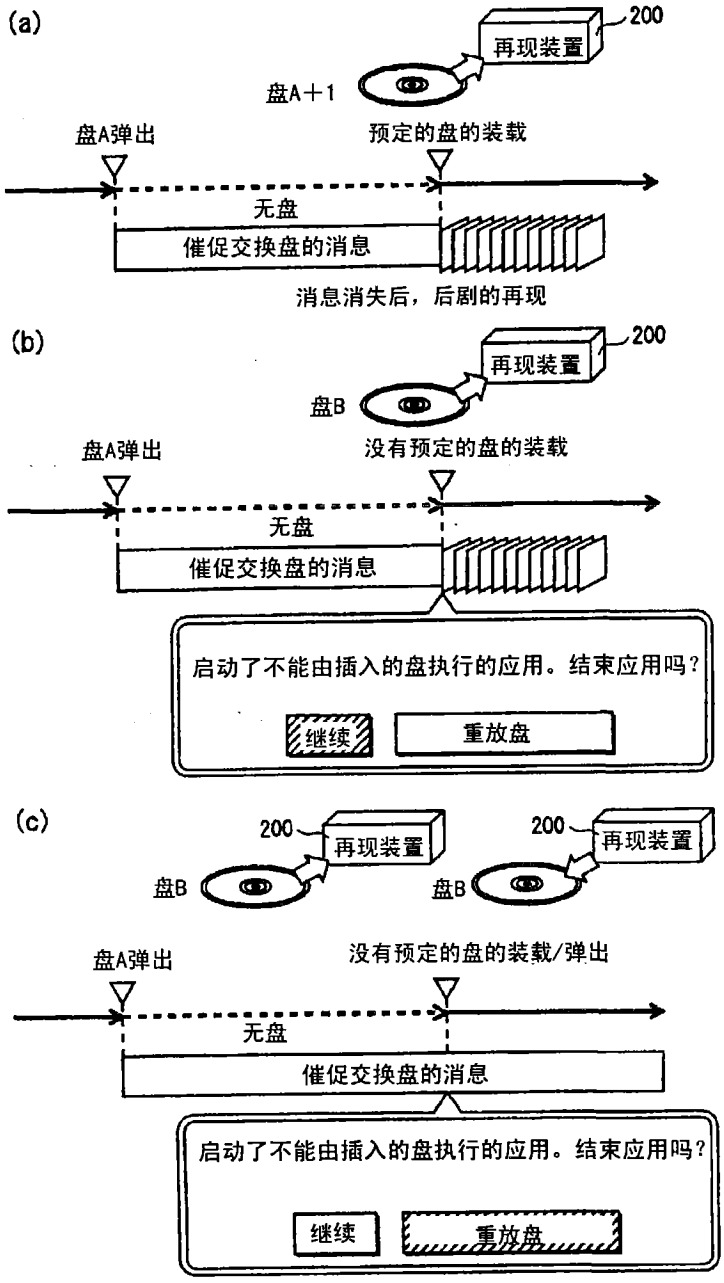

(a)
200
再现装置
盘A+1
盘A弹出
预定的盘的装载
无盘
催促交换盘的消息
消息消失后，后剧的再现
(b)
200
再现装置
盘B
盘A弹出
没有预定的盘的装载
无盘
催促交换盘的消息
启动了不能由插入的盘执行的应用。结束应用吗？
继续
重放盘
(c)
200
再现装置
盘B
200
再现装置
盘B
盘A弹出
没有预定的盘的装载/弹出
无盘
催促交换盘的消息
启动了不能由插入的盘执行的应用。结束应用吗？
继续
重放盘

图 34

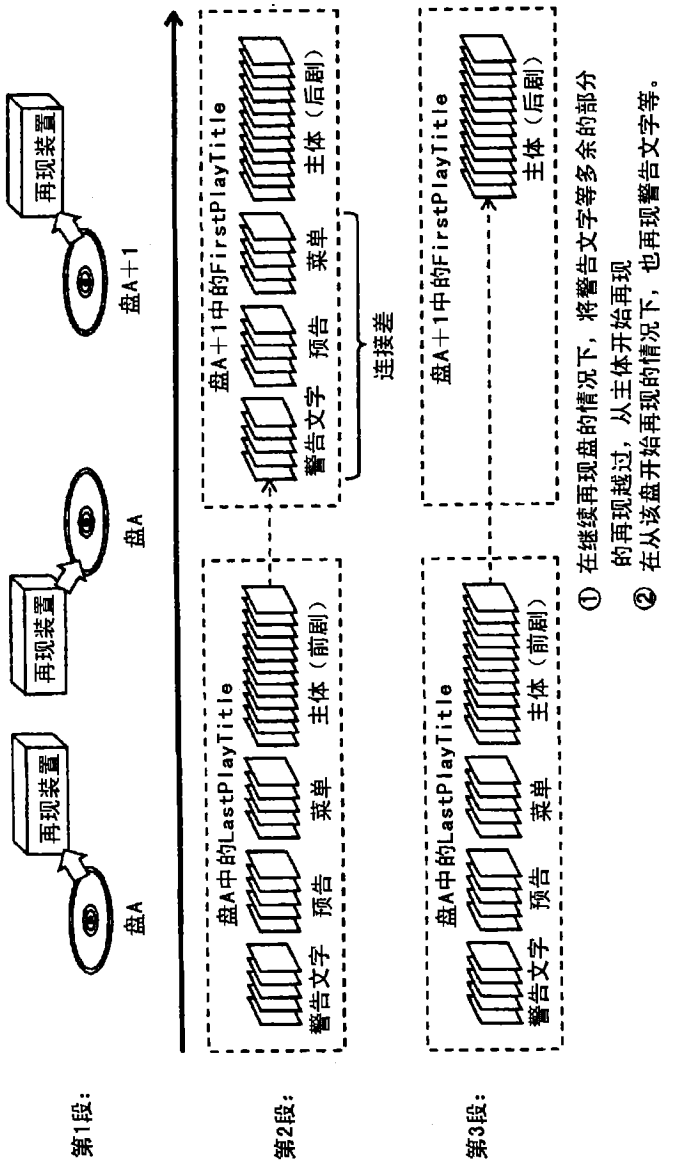

第1段:
第2段:
第3段:
再现装置
盘A
盘A+1
盘A中的LastPlayTitle
警告文字 预告 菜单 主体（前剧）
盘A+1中的FirstPlayTitle
警告文字 预告 菜单 主体（后剧）
连接差
① 在继续再现盘的情况下，将警告文字等多余的部分的再现越过，从主体开始再现
② 在从该盘开始再现的情况下，也再现警告文字等。

图35

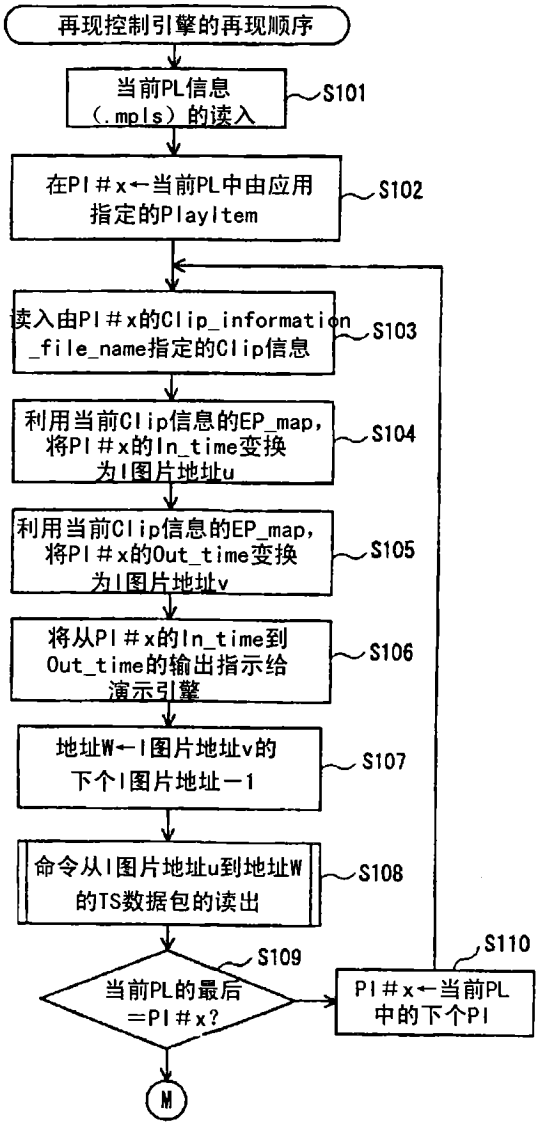

再现控制引擎的再现顺序
当前PL信息（.mpls）的读入
S101
在PI#x←当前PL中由应用指定的PlayItem
S102
读入由PI#x的Clip_information_file_name指定的Clip信息
S103
利用当前Clip信息的EP_map，将PI#x的In_time变换为I图片地址u
S104
利用当前Clip信息的EP_map，将PI#x的Out_time变换为I图片地址v
S105
将从PI#x的In_time到Out_time的输出指示给演示引擎
S106
地址W←I图片地址v的下个I图片地址－1
S107
命令从I图片地址u到地址W的TS数据包的读出
S108
当前PL的最后=PI#x?
S109
PI#x←当前PL中的下个PI
S110
M

图 36