(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 101826357 B
(45)授权公告日 2012.11.07

(21)申请号 201010163585.0

(22)申请日 2005.07.19

(30)优先权数据
2004-214918 2004.07.22 JP
2004-293042 2004.10.05 JP
2004-314663 2004.10.28 JP
2005-039832 2005.02.16 JP
2005-099409 2005.03.30 JP

(62)分案原申请数据
200580024565.2 2005.07.19

(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 大芦雅弘 田中敬一 大户英隆
杰尔马诺·莱克森林

(74)专利代理机构 永新专利商标代理有限公司
72002
代理人 王英 刘炳胜

(51)Int.Cl.
G11B 20/10(2006.01)

(56)对比文件
CN 1366669 A,2002.08.28,
JP 11282714 A,1999.10.15,
WO 2004030356 A1,2004.04.08,

审查员 张俊伟

权利要求书 2 页 说明书 38 页 附图 54 页

(54)发明名称

重放装置和重放方法

(57)摘要

在本地存储器(18)中,有多个文件,合并管理信息从所述多个文件中指定要与记录在只读记录记录介质上的内容结合使用的文件,以及签名信息用于判断所述合并管理信息的可靠性。虚拟文件系统单元(38)根据所述签名信息判断所述合并管理信息的可靠性。在所述合并信息被判断为是可信的情况下,虚拟文件系统单元(38)生成包信息,所述包信息指示通过将所述合并管理信息所指定的文件加入到所述只读记录介质的文件结构中而获得的新文件结构。

光盘ID	文件路径	散列值
disc#1	ROOT/organization#1/disc#1/00002.MPLS	XXXX
	⋮	⋮

CN 101826357 B

1. 一种用于重放相互结合的应用程序和数字流的重放方法，其中

在只读记录介质上记录有第一组文件，

在可读 / 可写记录介质上记录有 (i) 第二组文件，(ii) 合并管理信息，所述合并管理信息指示不需要判断其是否被窜改的数字流的存储位置、需要判断其是否被窜改的播放列表信息的存储位置，以及所述播放列表信息的散列值，其中所述数字流和所述播放列表信息中的每一个均为要与在所述只读记录介质上记录的数据组合使用的所述第二组文件中的文件，以及 (iii) 签名信息，所述签名信息包括所述合并管理信息的散列值，

所述重放方法包括 ：

第一判断步骤，其将所述播放列表信息的所述散列值与根据所述播放列表信息计算的散列值进行比较，并且如果这些散列值彼此相等，则判断所述播放列表信息未被窜改 ；

第二判断步骤，其将所述合并管理信息的所述散列值与根据所述合并管理信息计算的散列值进行比较，并且如果这些散列值彼此相等，则判断所述合并管理信息未被窜改 ；

包信息生成步骤，其 (a) 在所述合并管理信息被判断为未被窜改的情况下，生成包信息，所述包信息指示通过将其存储位置由所述合并管理信息指示的文件添加到所述只读记录介质的文件结构中而获得的新文件结构，以及 (b) 在所述合并管理信息被判断为已经被窜改的情况下，不生成指示所述新文件结构的包信息 ；

重放步骤，其在所述播放列表信息被判断为未被窜改的情况下，基于所述包信息生成步骤所生成的包信息、以所述播放列表信息指示的重放顺序来重放在所述只读记录介质或所述可读 / 可写记录介质上记录的所述数字流 ；以及

执行步骤，其基于所述包信息生成步骤所生成的包信息来执行在所述只读记录介质或所述可读 / 可写记录介质上记录的应用程序，其中

在所述合并管理信息被判断为未被窜改的情况下，在生成指示所述新文件结构的包信息之前，将其存储位置由所述合并管理信息指示的文件的属性设定为只读，并且

在生成所述包信息时，所述包信息生成步骤采用一种或多种颜色来执行显示。

2. 根据权利要求 1 所述的重放方法，还包括 ：

输出步骤，其将信息输出到显示装置，其中

在所述包信息生成步骤正在生成所述包信息时，所述输出步骤发出信号，以将所期望的信息输出到所述显示装置的显示屏幕。

3. 根据权利要求 2 所述的重放方法，其中

所期望的信息指示由所述包信息生成步骤进行的所述包信息生成的进度。

4. 一种用于重放相互结合的应用程序和数字流的重放装置，所述重放装置包括 ：

读出单元，其用于读出在安放于所述重放装置上的只读记录介质上记录的文件 ；

存储单元，其中存储 (i) 多个文件，(ii) 合并管理信息，所述合并管理信息指示不需要判断其是否被窜改的数字流的存储位置、需要判断其是否被窜改的播放列表信息的存储位置，以及所述播放列表信息的散列值，其中所述数字流和所述播放列表信息中的每一个均为要与在所述只读记录介质上记录的数据组合使用的所述多个文件中的文件，以及 (iii) 签名信息，所述签名信息包括所述合并管理信息的散列值 ；

第一判断单元，其用于将所述播放列表信息的所述散列值与根据所述播放列表信息计算的散列值进行比较，并且如果这些散列值彼此相等，则判断所述播放列表信息未被窜

改；

第二判断单元，其用于将所述合并管理信息的所述散列值与根据所述合并管理信息计算的散列值进行比较，并且如果这些散列值彼此相等，则判断所述合并管理信息未被窜改；

包信息生成单元，其用于 (a) 在所述合并管理信息被判断为未被窜改的情况下，生成包信息，所述包信息指示通过将其存储位置由所述合并管理信息指示的文件添加到所述只读记录介质的文件结构中而获得的新文件结构，以及 (b) 在所述合并管理信息被判断为已经被窜改的情况下，不生成指示所述新文件结构的包信息；

重放单元，其用于在所述播放列表信息被判断为未被窜改的情况下，基于所述包信息生成单元所生成的包信息、以所述播放列表信息指示的重放顺序来重放在所述只读记录介质或所述存储单元上记录的所述数字流；以及

执行单元，其用于基于所述包信息生成单元所生成的包信息来执行在所述只读记录介质或所述存储单元上记录的应用程序，其中

在所述合并管理信息被判断为未被窜改的情况下，在生成指示所述新文件结构的包信息之前，将其存储位置由所述合并管理信息指示的文件的属性设定为只读，并且

在生成所述包信息时，所述包信息生成单元以一种或多种颜色来执行显示。

5. 根据权利要求 4 所述的重放装置，包括

输出单元，其用于将信息输出到显示装置，其中

在所述包信息生成单元正在生成所述包信息时，所述输出单元发出信号，以将所期望的信息输出到所述显示装置的显示屏幕。

6. 根据权利要求 4 所述的重放装置，其中

所期望的信息指示由所述包信息生成单元进行的所述包信息生成的进度。

重放装置和重放方法

[0001] 本申请是基于申请日为 2005 年 7 月 19 日的申请号为 200580024565.2 的中国专利申请提出的分案申请。

技术领域

[0002] 本发明涉及“虚拟包”的重放技术。

背景技术

[0003] “虚拟包”是用于以下的技术 :(i) 生成完整地指示在只读记录介质 (例如 BD-ROM) 上记录的内容以及在可读写记录介质 (例如硬盘上) 记录的内容的信息,以及 (ii) 重放或执行记录在这些记录介质上的数字流和应用程序 (以下简称为应用程序),就好像其被记录在一个虚拟的包中一样。

[0004] 根据该技术,通过对记录在可重写硬盘上的数据进行更新,即使是在 BD-ROM 发行之后,也可以将产品的内容作为整个虚拟包而改变。例如,即使是记录有电影产品本身的 BD-ROM 发行之后,该电影产品的提供商也能够通过经由网络提供其他未发表电影产品的电影宣传片,向用户做最近电影产品的广告,不管什么时候发行该 BD-ROM。

[0005] 在以下专利文件中公开了涉及虚拟包的现有技术。

[0006] [专利文件 1]

[0007] 国际公开 WO 2004/030356A1。

发明内容

[0008] 本发明要解决的问题

[0009] 硬盘是可重写的事实暗示可以窜改硬盘上所记录的数据,并且存在提供商和用户受到不利情况的可能性。例如,这种不利情况包括,在 BD-ROM 上记录的数字流可能在违反提供商意愿的状态下被重放,从而可能降低产品的价值,并且 BD-ROM 上记录的数字流可能在执行硬盘上记录的非法应用程序的状态下被重放,从而使得重放装置遇到故障。

[0010] 本发明的目的在于提供一种重放装置,其能够防止在可重写记录介质上记录的非法数据被执行,或者与只读记录介质上记录的数据组合地被播放。

[0011] 解决问题的方法

[0012] 为了实现上述目的,本发明提供了一种重放装置,其重放相互结合的应用程序和数字流,所述重放装置包括 :读出单元,其用于读出安置在所述重放装置上的只读记录介质上记录的文件 ;存储单元,其中存储 (i) 多个文件, (ii) 合并管理信息,其从所述多个文件中指定要与在所述只读记录介质中记录的内容组合使用的文件,以及 (iii) 签名信息,其用于判断所述合并管理信息的可靠性 ;判断单元,其用于根据所述签名信息判断所述合并管理信息的可靠性 ;包管理单元,其用于 (a) 在所述合并管理信息被判断为可靠的情况下,生成包信息,所述包信息指示通过将所述合并管理信息所指定的文件加入到所述只读记录介质的文件结构中而获得的新文件结构,以及 (b) 在所述合并管理信息被判断为不可靠的

情况下，不生成指示所述新文件结构的包信息；重放单元，其用于根据由所述包管理单元所生成的包信息，重放记录在所述只读记录介质上或者存储在所述存储单元中的数字流；以及执行单元，其用于根据由所述包管理单元所生成的包信息，执行记录在所述只读记录介质上或者存储在所述存储单元中的应用程序。

[0013] 本发明的效果

[0014] 采用以上布置，根据本发明的重放装置根据签名信息验证合并管理信息的可靠性，并且在所述合并管理信息不能被确认为可靠时，所述重放装置禁止从所述存储单元中存储的文件生成包信息。

[0015] 采用该布置，可以防止在只读记录介质中记录的数据被与非法数据组合地执行或者重放。

附图说明

[0016] 图 1 示出了根据本发明的重放装置的使用形式；

[0017] 图 2 示出在 BD-ROM 上的文件目录结构；

[0018] 图 3 示意性地示出如何构建 AVClip；

[0019] 图 4 示出 PL 信息的结构；

[0020] 图 5 示出在 AVClip 时间轴与 PL 时间轴之间的关系；

[0021] 图 6 示出了采用 4 个 Clip_Information_file_name 的成批指定 (batch specification)；

[0022] 图 7 示出了 PLmark 信息的内部结构；

[0023] 图 8 示出了使用 PLmark 的章节定义；

[0024] 图 9 示出了 SubPath 信息的内部结构；

[0025] 图 10 示出了在 SubPlayItem 时间轴上的同步指定和重放间隔定义；

[0026] 图 11A 示出了 Java 档案 (archive) 文件中包含的程序和数据；

[0027] 图 11B 示出类文件的内部结构；

[0028] 图 12 示出了 BD-J 对象的内部结构；

[0029] 图 13 示出了 INDEX.BDMV 的内部结构；

[0030] 图 14 示出了本地存储器的目录结构；

[0031] 图 15 示出了由本地存储器中的 PL 信息所定义的 PlayList 重放时间轴类型；

[0032] 图 16A 示出了在 BD-ROM 上和在本地存储器中存储的应用程序和 AVClip；

[0033] 图 16B 示出了由在 BD-ROM 上和在本地存储器中存储的应用程序和 AVClip 所构成的标题；

[0034] 图 17 示出了合并管理信息文件的内部结构；

[0035] 图 18 示出了根据本发明的重放装置的内部结构；

[0036] 图 19 以层的形式示出了在指令 ROM 21 中存储的软件和硬件所构成的配置；

[0037] 图 20 示出了 Java 虚拟机 30 的内部结构；

[0038] 图 21 示出了重放状态的变换；

[0039] 图 22 示出了要由虚拟文件系统单元 38 执行的虚拟包信息的示例性结构；

[0040] 图 23A 示出了整个光盘的时间轴；

[0041] 图 23B 示出了整个光盘的时间轴的结构；

[0042] 图 24 示出如何由于标题变化而更新虚拟包信息；

[0043] 图 25 示意性地示出 Java 应用程序是如何请求服务器传送构成虚拟包的文件的；

[0044] 图 26 示意性地示出 Java 应用程序是如何将从服务器传送来的文件存储到本地存储器中的；

[0045] 图 27 示意性地示出 Java 应用程序是如何请求虚拟文件系统更新虚拟包信息的；

[0046] 图 28 示意性地示出是如何对虚拟包信息进行更新的；

[0047] 图 29 是示出模块管理器 33 执行的标题重放控制的处理的流程图；

[0048] 图 30 是示出重放控制引擎 32 所执行的 PlayList 重放处理的处理过程的流程图；

[0049] 图 31 是示出要由 Java 应用程序执行的、下载构成虚拟包的文件的过程的流程图；

[0050] 图 32 是示出虚拟文件系统单元 38 所执行的准备处理的流程图；

[0051] 图 33 是示出由虚拟文件系统单元 38 所执行的更新处理的流程图；

[0052] 图 34 是流程图，示出了当由于标题调用而将当前标题的重放临时暂停时由重放控制引擎所执行的处理过程，以及当原始播放的标题的重放继续时由重放控制引擎所执行的处理过程；

[0053] 图 35 是虚拟包管理表的示例；

[0054] 图 36 是示出根据第三实施例，用于构建虚拟包的处理过程的流程图；

[0055] 图 37 是示出控制对文件的写入访问的 API 的处理过程的流程图；

[0056] 图 38 示出了在构成虚拟包的文件之一具有错误的情况下，如何生成错误被校正的校正文件；

[0057] 图 39 是示出根据第四实施例，用于构建虚拟包的处理过程的流程图；

[0058] 图 40 是流程图，示出了在已经构建了虚拟包之后在对本地存储器中的文件进行写入访问的情况下，提供文件 I/O 功能的 API 的处理；

[0059] 图 41 示出了根据第五实施例的重放装置的内部结构；

[0060] 图 42 示出了根据第五实施例的虚拟包管理表的示例；

[0061] 图 43 是示出将下载的文件记录到本地存储器中的处理过程的流程图；

[0062] 图 44 是示出根据第五实施例，用于构建虚拟包的处理过程的流程图；

[0063] 图 45 示出了根据第六实施例的虚拟包构建的示例；

[0064] 图 46 是示出根据第六实施例，用于构建虚拟包的处理过程的流程图；

[0065] 图 47 是示出将文件移动到 ACTIVE 目录中的处理过程的流程图；

[0066] 图 48 是示出根据第六实施例，控制对文件的写入访问的 API 的处理过程的流程图；

[0067] 图 49 示出根据第七实施例的虚拟包的构建的示例；

[0068] 图 50 示意性地示出了是如何采用在 SHARED 目录下的 ACTIVE 目录和在 disc#1 目录下的 ACTIVE 目录构建虚拟包的；

[0069] 图 51 是示出了根据第七实施例，用于构建虚拟包的处理过程的流程图；

[0070] 图 52 描述了构建中间文件结构的过程；

[0071] 图 53 是详细示出图 51 中步骤 S184 中的处理的流程图；

[0072] 图 54 是示出虚拟文件系统单元 38 如何将本地存储器中的文件提供给其他模块的流程图；

[0073] 图 55 示意性地示出了在根据第九实施例的虚拟包构建过程中的显示；

[0074] 图 56 示出了在根据第九实施例的虚拟包构建处理中的失败显示；

[0075] 图 57 示意性地示出了在根据第九实施例的虚拟包构建处理中的成功显示；

[0076] 图 58 是示出用于对与虚拟包的构建有关的显示进行更新的处理过程的流程图；以及

[0077] 图 59 示出了根据第九实施例的在本地存储器中的内容列表的显示。

具体实施方式

[0078] 第一实施例

[0079] 以下描述与本发明有关的重放装置的实施例。首先，描述与本发明有关的重放装置的实现形式中的使用形式。图 1 示出了与本发明有关的重放装置的使用形式。在图 1 中，与本发明有关的重放装置是重放装置 200。重放装置 200 用于在包含重放装置 200、遥控器 300 和电视机 400 的家庭剧场系统中提供电影作品。

[0080] 在此完成了对与本发明有关的重放装置的使用形式的描述。接下来要描述的用于由与本发明有关的重放装置进行的重放的记录介质。与本发明有关的重放装置所播放的记录介质是 BD-ROM。图 2 示出了 BD-ROM 的内部结构。

[0081] BD-ROM 在图 2 的第四层级 (tier) 示出，而 BD-ROM 上的轨道 (trace) 在第三层级示出。图 2 所示的轨道源自从已经被画出各面的 BD-ROM 的内圆周螺旋至外圆周的轨道。该轨道包含导入区、卷区以及导出区。图 2 中的卷区具有包括物理层、文件系统层以及应用层的分层结构。采用目录结构表示的 BD-ROM 的应用格式给出了图 2 中的第一层级。BDMV 目录放置在 BD-ROM 中的第一层级的 ROOT 目录下。

[0082] 在 BDMV 目录下存在 INDEX.BDMV 文件和五个子目录，分别为 PLAYLIST 目录、CLIPINF 目录、STREAM 目录、BDJA 目录和 BOBJ 目录。

[0083] STREAM 目录存储形成主要数字流的文件，将扩展名“M2TS”指定给该文件 (00001.M2TS)

[0084] 在 PLAYLIST 目录中存在扩展名指定为“MPLS”的文件 (00001.MPLS)。

[0085] 在 CLIPINF 目录中存在扩展名为“CLPI”的文件 (00001.CLPI)。

[0086] 在 BDJA 目录中存在具有扩展名为“JAR”的文件 (00001.JAR)。

[0087] 在 BOBJ 目录中存在具有扩展名为“BOBJ”的文件 (00001.BOBJ)。

[0088] 接下来描述这些文件

[0089] <AVClip>

[0090] 首先描述具有扩展名“M2TS”的文件。图 3 示意性地示出了具有扩展名“M2TS”的文件是如何构建的。每个具有扩展名“M2TS”的文件 (00001.M2TS、00002.M2TS、00003.M2TS、...) 存储 AVClip。通过多路复用 TS 包来构成 AVClip(中间层级)，TS 包是通过将包含多个视频帧 (图像 pj1、pj2、pj3) 的视频流和多个音频帧 (上面第一层级) 首先转换为 PES 包 (上面第二层级)，然后转换为 TS 包 (上面第三层级)，以及以相同方式将字幕显示图形流 (下面第一层级的 PG 流) 和对话交互图形流 (下面第一层级的 IG 流) 转换为 TS

包（下面第三层级）而得到的。

[0091] 除了通过图 3 所示的多路复用获得的 AVClip 之外，还存在不是从多路复用得到的 AVClip。这些 AVClip 称为子剪辑 (SubClip)，包括组成音频流、图形流或者文本字幕流 (TextSTStream) 的 AVClip。

[0092] ＜剪辑信息＞

[0093] 具有扩展名“CLPI”的文件 (00001.CLPI) 是与 AVClip 相对应的一条剪辑信息。剪辑信息，即管理信息，包括 EP_map，EP_map 示出了在 AVClip 中的流的 GOP 的头位置和诸如编码格式、帧频、比特率和分辨率等等之类的信息。

[0094] ＜播放列表信息＞

[0095] 具有扩展名“MPLS”的文件 (00001.MPLS) 存储播放列表 (PL) 信息。PL 信息通过查询 AVClip 来定义播放列表。在图 4 的左侧示出了 PL 信息的结构，其包括主路径信息、PLMark 信息和子路径信息。

[0096] 主路径信息“MainPath()”包括播放项目信息“PlayItem()”，用箭头 mp1 指示。播放项目是通过在一个或多个 AVClip 时间轴上指定 In_time 和 Out_time 所定义的重放间隔。由多个播放间隔组成的播放列表 (PL) 是通过放置多条播放项目信息来定义的。图 4 中的箭头 mp2 示出了播放项目信息内部结构的特写 (close up)。如图 4 所示，播放项目信息包括 In_time、Out_time 和示出相应 AVClip 的 Clip_Information_file_name。图 5 示出了 AVClip 与 PL 之间的关系。第一层级示出了 AVClip 的时间轴，而第二层级示出了 PL 的时间轴。PL 信息包括 3 条播放项目信息 (PlayItem#1-#3)，PlayItem#1、#2 和 #3 的 In_time 和 Out_time 定义三个播放间隔。当三个重放间隔排列成线时，对 AVClip 定义了的不同的时间轴。这就是第二层级的 PL 时间轴。因此通过在播放项目信息中的定义，能够对 AVClip 定义不同的时间轴。

[0097] 通常，每次指定一个 AVClip，但是也可以进行多个 AVClip 的成批指定。使用播放项目信息中的 Clip_Information_file_name 执行 AVClip 的成批指定。图 6 示出了使用 4 个 Clip_Information_file_name 的 AVClip 成批指定。图中的第一层级到第四层级示出了 4 个 AVClip 时间轴 (AVClip#1-#4 的时间轴)，第五层级示出了 PL 时间轴。4 个时间轴采用在播放项目信息中包含的所述 4 个 Clip_Information_file_name 来指定。这允许由 In_time 和 Out_time 定义 4 个可选择播放的重放间隔。因此，在 PL 时间轴上定义由多条可转换角度的视频（所谓的多角度间隔）组成的间隔。

[0098] PLmark 信息“PLmark()”指定在 PL 时间轴上的任意间隔为章节。图 7 示出了 PLmark 结构的内部结构，其包括 ref_to_PlayItem_Id 和 Mark_time_stamp，如图 7 中的箭头 pm1 所指示。图 8 示出了使用 PLmark 的章节定义。图 8 中的第一层级示出了 AVClip 时间轴，而第二层级示出了 PL 时间轴。图 8 中的箭头 pk1 和 pk2 示出了播放项目 (ref_to_PlayItem_Id) 和 Plmark 中的时间点 (Mark_time_stamp) 的指定。作为这些指定的结果，在 PL 时间轴上定义了 3 个章节（章节 #1-#3）。在此完成了 PLmark 的描述。接下来描述子路径信息。

[0099] 子路径信息“SubPath()”通过指定子剪辑时间轴上的 In_time 和 Out_time 来定义一个或多个重放间隔，并且其具有图 9 所示的内部结构。子路径信息包括箭头 sh1 所指示的子播放项目信息“SubPlayItem()”。在箭头 sh2 标记的特写中，子播放项目信息包括 Clip_

Information_file_name、In_time、Out_time、sync_PlayItem_Id 和 sync_Start_PTS_of_PlayItem。采用 Clip_Information_file_name、In_time 和 Out_time 指定在子剪辑时间轴上的 In_time 和 Out_time。sync_PlayItem_Id 和 sync_Start_PTs_of_PlayItem 用于将在子剪辑时间轴上的重放间隔与 PL 时间轴进行同步。这就允许在子剪辑时间轴和 PL 时间轴两者上相互同步地进行处理。

[0100] 图 10 示出在子剪辑时间轴上的重放间隔的同步指定和定义。图 10 中的第一层级示出了 PL 时间轴，第二层级示出了子剪辑时间轴。图 10 中的 SubPlayItem.In_time 和 SubPlayItem.Out_time 分别示出了重放间隔的开始和结束。因此显然，也在子剪辑时间轴上定义重放间隔。由箭头 Sn1 标记的 sync_playItem_Id 示出了播放项目的同步指定，由箭头 Sn2 标记的 sync_Start_PTs_of_PlayItem 示出了在 PL 时间轴上的播放项目中的时间点的指定。

[0101] 在 BD_ROM 中的 PL 信息的特征在于，其使得可以定义使 AVClip 能够被切换的多角度间隔和使 AVClip 和子剪辑能够被同步的同步间隔。剪辑信息和 PL 信息被分类为“静态脚本 (static scenarios)”。

[0102] 以下描述“动态脚本”。在此，“动态”指由于在重放装置 200 中的用户键事件和和状态改变等等造成的重放控制内容的改变。采用 BD_ROM，能够采用与 Java 应用程序相同的描述来描述重放控制。即，采用 BD_ROM，Java 应用程序作为动态脚本。

[0103] <Java 应用程序 >

[0104] 以下描述 Java 应用程序。Java 应用程序包括加载在虚拟机的堆区域 (也称为工作存储器) 中的一个或多个 xlet 程序。应用程序由工作存储器中加载的 xlet 程序以及数据构成。在此完成了对 Java 应用程序结构的描述。

[0105] Java 应用程序的实体是在图 2 中的 BDMV 目录下的 BDJA 目录中存储的 Java 档案文件 (00001.jar，00002.jar)。以下参考图 11 描述 Java 档案文件。

[0106] <Java 档案文件 >

[0107] Java 档案文件 (图 9 中的 00001.JAR) 是一个或多个类文件和数据文件等的集合。图 11A 示出了在档案文件中收集的程序和数据。图 11A 中的数据是由 Java 存档器 (archiver) 收集并排列在帧中所示的目录结构中的多个文件。该目录结构包括 Root 目录、Java 目录和 image 目录，common.pkg 文件放置在 Root 目录中，类文件 (aaa.class，bbb.class) 放置在 Java 目录中，menu.jpg 文件放置在 image 目录中。Java 档案文件是 Java 存档器将这些文件收集在一起的结果。当将类文件和数据从 BD_ROM 读入到高速缓冲器中时，展开类文件和数据，并在高速缓冲器中将其看作存在于目录中的多个文件。在 Java 档案文件的文件名中的 5 位 (five-digit) 数值“zzzzz”示出了 Java 档案文件的 ID，BD-J 对象指使用这种值的 Java 档案文件。通过在已经将 Java 档案文件读入高速缓冲器时查询在文件名中的该数值，就可以提取构成任意 Java 应用程序的数据和程序。

[0108] 图 11A 中的类文件与上述 xlet 程序相对应。采用由 Java 操作环境所支持的操作模式 (BD-J 模式) 的重放过程规定使用与这些类文件的实例相对应的 xlet 程序。xlet 程序是能够采用 Java 多媒体框架 (JMF) 接口的 Java 程序，并且其根据 JMF 等等、基于键事件来执行播放列表重放处理。

[0109] 此外，xlet 程序还能够执行访问 WWW 站点和下载内容的过程。这实现了通过将所

下载内容与播放列表相混合创建的原始作品的重放。

[0110] 接下来描述 xlet 程序的类文件。图 11B 示出了类文件的内部结构。如图 11B 所示，该类文件与普通类文件类似，具有常数池、接口、方法 1、2、3、....n。在重放装置 200 中，在类文件中的方法包括由预先记载的键事件所触发的那些方法（时间监听器）和用于调用函数 API（应用程序接口）的那些方法。通过采用分配给给定方法的局部变量和在调用该方法时出现的自变量，来描述在这些方法中的计算过程等等。在此完成了对 Java 档案文件的描述。

[0111] 接下来描述具有扩展名“BOBJ”的文件。该文件存储 BD-J 对象。BD-J 对象是通过将在播放列表中定义的 AVClip 序列与应用程序相关联来定义标题的信息。图 12 示出了 BD-J 对象的内部结构。BD-J 对象示出了“应用程序管理表”和“播放列表信息参考值”。应用程序管理表通过枚举应用程序标识符（应用程序 ID）和构成该应用程序的档案文件的 ID，示出了其生命周期为标题的每个应用程序。换而言之，一个应用程序包括一个或多个 Java 档案文件。“播放列表信息参考值”示出了当标题开始时同时被重放的播放列表信息。

[0112] 在此完成了对具有扩展名“BOBJ”的文件的描述。

[0113] 应该注意的是，在以下列出的国际公开中，公开了对 BD-J 对象和应用程序管理表的描述，参考该公开以进行进一步详细描述：

[0114] 国际公开 WO 2005/045840

[0115] 接下来描述 INDEX.BDMV 文件。

[0116] INDEX.BDMV 是整个 BD-ROM 的管理信息，包括诸如组织 ID 和光盘 ID 之类的信息，其中组织 ID 是标识电影产品提供商的标识符，光盘 ID 是为由提供商所提供的每个 BD-ROM 指定的标识符。当将光盘插入重放装置时，首先读出 INDEX.BDMV，这样重放装置就一一对应地识别出光盘。另外，INDEX.BDMV 包括一个表格，其相互对应地示出了在 BD-ROM 中的多个可播放标题和规定各标题的 BD-J 对象。以下描述可以记录在 BD-ROM 中的标题类型，其包括 FirstPlayTitle、Top_menuTitle 和 Title#1、#2 和 #3。

[0117] 在进行任何其他事情之前，在加载 BD-ROM 时，FirstPlayTitle 承担播放 BD-ROM 的动态商标。因此，在加载 BD-ROM 时，FirstPlayTitle 实现播放象征电影作品的制片商和 / 或者发行者的动态商标。

[0118] Top_menuTitle 包括 AVClip 和用于播放在 BD-ROM 中的菜单分层的极顶部所放置的菜单的应用程序。

[0119] Title#1、#2 和 #3 是与普通电影作品相对应的标题。因此，INDEX.BDMV 是示出诸如 FirstPlayTitle、Top_menuTitle、Title#1 到 #3 之类的标题与单个 BD-J 对象的对应关系的文件。

[0120] 图 13 示出了 INDEX.BDMV 的内部结构。如图所示，INDEX.BDMV 包括多条标题信息，例如“FirstPlayTitle 信息”、“Top_menuTitle 信息”、“Title#1 信息”、“Title#2 信息”和“Title#3 信息”。每条标题信息示出在标题号与规定标题的 BD-J 对象之间的对应关系。采用这样一条标题信息，就可以指定定义标题的 BD-J 对象，采用这种 BD-J 对象，就可以使得播放列表信息与应用程序相互结合地工作。因此完成对 INDEX.BDMV 的描述。

[0121] 在以下列出的国际公开中，公开了对 INDEX.BDMV 的详细描述，参考该公开以进行进一步详细描述：

[0122] 国际公开 WO 2004/025651

[0123] BD-ROM 不是用于本发明的重放装置的唯一记录介质。磁记录装置（本地存储器），诸如重放装置中内置的硬盘，也可用于本发明的重放装置。以下描述了在这种本地存储器上记录的数据。

[0124] 图 14 示出了在本地存储器中的目录结构。

[0125] 在这种目录结构中，子目录“organization#1”位于 ROOT 目录下，并且在该目录下是子目录“disc#1”和“disc#1”。将“organization#1”目录分配给特定的电影作品提供商。将“disc#1”和“disc#1”目录分别分配给由该提供商所提供的 BD-ROM。在每个 BD-ROM 的 INDEX.BDMV 中指示的组织 ID 和光盘 ID 的值用于目录名。

[0126] 在与特定提供商相对应的目录中提供与 BD-ROM 相对应的目录，允许分别存储与单个 BD-ROM 相关的下载数据。在这些子目录下存储播放列表信息、剪辑信息和 AVClip，与 BD-ROM 中所存储的类似。还额外存在 Java 档案文件、合并管理信息文件和签名信息文件。

[0127] 与在 BD-ROM 上的播放列表信息相比，在 BD-ROM 上播放列表仅仅指 AVClip，而在本地存储器中的播放列表信息（在虚拟包中新加入的播放列表信息）指在 BD-ROM 上的 AVClip 和本地存储器 18 中的 AVClip。

[0128] 在此，描述了本地存储器中的播放列表信息由 4 条播放列表信息（播放项目信息 #1-#4）构建的情况。在头条信息指在 BD-ROM 上的剪辑信息而剩余三条信息指在本地存储器中的信息的情况下，该播放列表信息从在 BD-ROM 上的 AVClip 和在本地存储器中的 AVClip 来定义单一流序列，如图 15 所示。

[0129] 图 15 示出了由存储在本地存储器中的 PL 信息所定义的播放列表重放时间轴的种类。第一层级示出了在 BD-ROM 上存储的 AVClip 的重放时间轴，第二层级示出了在本地存储器中存储的 PL 信息中定义的播放列表的重放时间轴。第三层级示出了在本地存储器中存储的 AVClip#3 的重放时间轴。第四层级示出了在本地存储器中存储的 AVClip#4 的重放时间轴。第五层级示出了在本地存储器中存储的 AVClip#5 的重放时间轴。

[0130] 在播放列表信息中的播放项目中，在播放项目信息 #2、#3 和 #4 指定 AVClip#2、#3 和 #4 作为重放间隔的情况下，播放列表信息能够将在 BD-ROM 上的 AVClip 和在本地存储器 18 中的 AVClip 两者规定为单一流序列。

[0131] 如上所述，可以将在 BD-ROM 上的 AVClip 和在本地存储器 18 中的 AVClip 两者规定为单一流序列，并且通过将该流序列与在 BD-ROM 上或者本地存储器中的应用程序组合，可以从这些 AVClip 和从在 BD-ROM 上或者在本地存储器中记录的应用程序构建单一标题。在如图 16A 所示，在 AVClip#1 被记录在 BD-ROM 上并且 AVClip#2 到 #4 和应用程序被记录在本地存储器中的情况下，可以将这些 AVClip 和应用程序看作单一标题，如图 16B 所示。

[0132] 接下来描述合并管理信息文件。该合并管理信息文件总体示出了构成虚拟包的所有这些文件，并且其在本地存储器中的 disc#1 和 disc#2 目录中。

[0133] 图 17 示出了合并管理信息文件的示例性内部结构。合并管理信息文件包括与在本地存储器中的构成虚拟包的文件相关的存储位置信息。对于每个文件而言，该存储位置信息包括指示在本地存储器 18 中的文件的存储位置的文件路径，和指示从在该文件中的数据确定的并使用单向函数计算的文件的散列值的“散列值”。

[0134] 接下来描述签名信息文件。签名信息文件示出了在合并管理信息文件上的提供商

的电子签名。通常采用的电子签名是通过计算需要窜改验证 (tamper-proof) 的信息的散列值并使用某种密钥对该散列值进行加密来获得的。在根据本实施例的签名信息文件中，使用与重放装置持有的合并证书中的公钥相对应的密钥，对合并管理信息文件的散列值进行加密。

[0135] 应该注意的是，合并证书是用于验证合并管理信息文件的证书，并且其包括提供商所发布的公钥。预先将提供商所提供的合并证书并入重放装置中。作为示例，对于合并证书的文件格式而言，可以使用 X.509。X.509 的详细说明写入在国际电报电话咨询委员会公布的 CCITT Recommendation X.509(1988)，“The Directory-AuthenticationFramework”中。

[0136] 应该注意的是，尽管以上描述示出预先将合并证书并入重放装置中，但是将合并证书记录在 BD-ROM 上也是可接受的。可替换地，从通过因特网提供合并证书的服务器装置中下载并获得合并证书也是可接受的。

[0137] 在此完成了对记录介质的描述。以下描述与本发明相关的重放装置的内部结构。

[0138] < 重放装置 >

[0139] 图 18 示出了本发明的重放装置的内部结构。本发明的重放装置可以根据图中所示的内部结构进行工业制造。本发明的重放装置主要包括两个部分，例如系统 LSI 和驱动装置，并且其可以通过将这些部分安放在重放装置的箱体中和安放在基板上进行工业制造。系统 LSI 是集成电路，在该集成电路中集成了用于实现重放装置功能的各种处理单元。采用这种方式制造的重放装置包括 :BD-ROM 驱动器 1、读缓冲器 2、多路分解器 3、视频解码器 4、视频平面 5、P 图形解码器 6、显示图形平面 (presentation Graphic plane)7、组合单元 8、字体生成器 9、I 图形解码器 10、开关 11、交互图形平面 12、合成单元 13、CLUT 单元 14、CLUT 单元 15、音频解码器 16、网络设备 17、指令 ROM 21、用户事件处理单元 22、PSR 组 23、CPU 24、脚本存储器 25、本地存储器 26 和开关 27。

[0140] 首先描述与记录在 BD-ROM 上的 AVClip 的重放有关的组成组件 (BD-ROM 驱动器 1 至音频解码器 16)。

[0141] BD-ROM 驱动器 1 加载 / 退出，以及访问 BD-ROM。

[0142] 读缓冲器 2 是先进先出 (FIFO) 存储器，在该存储器中，根据先进先出存储从 BD-ROM 或者本地存储器 18 中读出的传输流 (TS) 包。

[0143] 多路分解器 (De-MUX)3 从读缓冲器 2 中取出 TS 包，并将这些 TS 包转换为 PES 包。然后，在通过该转换获得 PES 包中，将具有由 CPU 24 设定的 PID 的 PES 包输出到视频解码器 4、P 图形解码器 6、I 图形解码器 10 以及音频解码器 16 中的一个。

[0144] 视频解码器 4 对从多路分解器 3 输出的 PES 包进行解码，以获得未压缩格式的图像，并将这些图像写入视频平面 5。

[0145] 视频平面 5 用于存储未压缩图像。平面是在重放装置中的存储区，用于存储一个屏幕大小的像素数据。视频平面 5 具有 1920×1080 分辨率，所存储的图像数据由采用 16 比特 YUV 表示的像素数据构成。在视频平面 5 中，在视频流中的重放视频图像能够对于每帧进行缩放。缩放包括将每帧的重放视频图像改变为完整视频平面 5 的 1/4(四分之一) 或者 1/1(全比例)。缩放是根据来自 CPU 24 的指令采用 BD-J 模式执行的，实现了屏幕的产生，从而将视频流的重放图像缩小到屏幕的角落或者投影到整个屏幕。

[0146] P 图形解码器 6 对从 BD-ROM 中读出的显示图形流进行解码，并将未压缩图形写入显示图形平面 7。作为对图形流解码的结果，字幕出现在屏幕上。

[0147] 显示图形平面 7 是具有用于一屏大小数据的空间的存储器，其能够存储一屏大小的未压缩图形。该平面具有 1920×1080 的分辨率，在显示图形平面 7 中的未压缩图形的像素由 8 比特索引颜色来表示。在通过采用 CLUT（颜色查询表）转换索引颜色，来将显示图形平面 7 中存储的的未压缩图形用于显示。

[0148] 合成单元 8 将在视频平面 5 中存储的未图像数据 (i) 与显示图形平面 7 中所存储的进行合成。

[0149] 字体生成器 9 使用字符字体扩展在位图中的 textST 流中包含的文本代码，并将所扩展的代码写入显示图形平面 7。

[0150] I 图形解码器 10 采用 DVD-like 模式对从 BD-ROM 或者本地存储器 18 中读出的 IG 流进行解码，并且将未压缩图形写入交互图形平面 12。

[0151] 开关 11 选择性地将由字体生成器 9 生成的字体序列和由 P 图形解码器 6 的解码得到的图形中的一个写入到显示图形平面 7。

[0152] 交互图形平面 12 被写有由 I 图形解码器 10 解码得到的未压缩图形。将由应用程序画出的字符和图形采用 α RGB 全彩色并采用 BD-J 模式写入到交互图形平面 12 中。

[0153] 合成单元 13 将在交互图形平面 12 中所存储的内容与从合成单元 8 输出的合成图形进行合成（即，未压缩图像数据与在显示图形平面 7 中所存储的内容的合成）。该合成使得由应用程序写入到 I 图形解码器 10 中的字符和图形覆盖到未压缩图像数据上并进行显示。

[0154] CLUT 单元 14 将在视频平面 5 中存储的未压缩图形中的索引颜色转换为 Y、Cr、Cb 值。

[0155] 当采用 DVD-like 模式工作时，CLUT 单元 15 将在交互图形平面 12 中存储的未压缩图形中的索引颜色转换为 Y、Cr、Cb 值。当采用 BD-J 模式工作时，CLUT 单元 15 将 α RGB 全彩色转换为 Y、Cr、Cb。

[0156] 音频解码器 16 对从多路分解器 3 输出的 PES 包进行解码，并输出未压缩音频数据。

[0157] 在此完成了对与 AVClip 重放有关的构成组件的描述。以下描述与 BD-J 模式工作有关的构成组件（网络设备 17 至 De-MUX 20）

[0158] 网络设备 17 实现了在重放装置中的通信功能。在应用程序采用 BD-J 模式指定 URL 的情况下，网络设备 17 与该 URL 指示的站点建立 TCP 连接、FTP 连接等等。作为所述连接被建立的结果，使得 Java 应用程序能够从该站点进行下载。

[0159] 本地存储器 18 是硬盘，用于存储元数据以及从通信和 BD-ROM 之外的记录介质提供的内容，例如通过由网络设备 17 建立的连接从站点下载的内容。元数据是用于绑定和管理在本地存储器 18 中的所下载内容的信息。通过访问本地存储器 18，采用 BD-J 模式的应用程序能够使用所下载内容执行各种处理。

[0160] 读缓冲器 19 是 FIFO 存储器，其在子剪辑包含在 BD-ROM 上或者在本地存储器 18 中所存储的内容中的情况下，基于先进先出存储组成子剪辑的 TS 包。

[0161] 多路分解器 (De-MUX)20 从读缓冲器 19 中取出 TS 包，并将所读出的 TS 包转换为

PES 包。然后，在通过该转换获得的 PES 包中，将具有所期望 PID 的 PES 包输出到字体生成器 9、P 图形解码器 6 和音频解码器 16。

[0162] 上述组件，即网络设备 17 到 De-MUX 20 使得 Java 应用程序通过网络下载的内容能够采用与在 BD-ROM 中所记录的内容相同的方式进行重放。以下描述了在重放装置中用于实现集体控制的构成组件（指令 ROM 21- 开关 27）。

[0163] 指令 ROM 21 存储规定与重放装置相关的控制的软件。

[0164] 用户事件处理单元 22 响应于遥控器或者重放装置的前面板的键操作，将用于执行这些操作的用户事件输出到 CPU 24。

[0165] PSR 组 23 是重放装置中内置的电阻器，其包括 64 个独立的播放器状态寄存器 (PSR) 和 4096 个独立的通用寄存器 (GPR)。PSR4 到 PSR8 是表示当前重放时间点的 PRS。

[0166] 通过将 PSR4 设定为 1-100 的值，来指示当前重放点的标题。当将 PSR4 设定为“0”时，意味着当前重放点为顶部菜单。

[0167] 通过将 PSR5 设定为 1-999 的值，来指示当前重放点的章节号。当 PSR5 设定为“0xFFFF”时，意味着在重放装置中的章节号为空。

[0168] 通过将 PSR6 设定为 0-999 的值，来指示当前重放点所属的 PL（当前 PL）号。

[0169] 通过将 PSR7 设定为 0-255 的值，来指示当前重放点所属的播放项目（当前播放项目）号。

[0170] 通过将 PSR8 设定为 0-0xFFFFFFFF 的值，来指示使用 45KHz 时间精度的当前重放点（当前显示时间或者“PTM”）。PSR4 到 PSR8 使得能够在整个 BD-ROM 的时间轴上指定重放的当前时间点。

[0171] CPU 24 运行在指令 ROM 21 中存储的软件，以执行与整个重放装置相关的控制。这些控制根据从用户事件处理单元 22 输出的用户事件和在 PSR 组 23 中的 PSR 设定值而动态改变。

[0172] 脚本存储器 25 用于存储当前 PL 信息和当前剪辑信息。当前 PL 信息是当前用于处理的 BD-ROM 中所记录的该条 PL 信息。当前剪辑信息是当前用于处理的 BD-ROM 中所记录的该条剪辑信息。

[0173] 本地存储器 26 是高速缓存器，用于在从 BD-ROM 的读出速度慢时，暂时存储在 BD-ROM 上所记录的内容。本地存储器 26 的提供允许采用 BD-J 模式的应用程序高效运行。

[0174] 开关 27 选择性地将从 BD-ROM 和本地存储器 18 读出的数据传递到读缓冲器 2、读缓冲器 19、脚本存储器 25 和本地存储器 26 中的一个。

[0175] 因此完成对本发明的重放装置的硬件配置的描述。以下将描述本实施例的重放装置中的软件结构。

[0176] 图 19 采用层结构形式示出了在指令 ROM 21 中存储的软件和硬件构成的配置。如图所示，重放装置的层结构包括以下：

[0177] (a) 第一层，用于 BD 播放器设备

[0178] (b) 第二层，用于 BD 播放器模型以及

[0179] (c) 第三层，用于应用程序运行时间环境

[0180] 在这些层中，图 18 所示的重放装置的硬件配置属于第一层。图 18 所示的硬件结构中，图“BD 播放器设备”中所示的第一层包括“解码器”，其包括视频解码器 4、P 图形解码

器 6、I 图形解码器 10、音频解码器 16 ;“平面”,其包括视频平面 5、显示图形平面 7、交互图形平面 12 ;BD-ROM 及其文件系统 ;以及本地存储器 18 和及其文件系统。

[0181] 第二层“BD 播放器模型”包括以下两个层 (b1) 和 (b2) :

[0182] (b2) 用于重放控制引擎 32 的层 ;以及

[0183] (b1) 用于虚拟文件系统单元 38 和显示引擎 31 的层。

[0184] 由第二层提供函数 API,用于位于其上的层。

[0185] 在这些层中,图 18 所示的 PSR 组 23 和脚本存储器 25 在重放控制引擎 32 中。

[0186] 第三层“应用程序运行时间环境”包括以下栈层 (c2) 和 (c1) :

[0187] (c2) 存在模块管理器 33 的层

[0188] (c1) 存在 DVD-like 模块 29a 和 Java 平台 29b 的层。

[0189] 以下描述在该软件结构中的构成组件。

[0190] <DVD-like 模块 29a 和 Java 平台 29b>

[0191] DVD-like 模块 29a 对导航命令进行解码,并根据解码结果执行对于重放控制引擎 32 的函数调用。

[0192] 在以下国际申请中公开,

[0193] 在以下列出的国际公开中,公开了与 DVD-like 模块有关的现有技术和由该 DVD-like 模块所执行的 Movie 对象,参考该公开以进行进一步详细描述 :

[0194] 国际公开 WO 2004/074976

[0195] Java 平台 29b 是所谓的 Java 平台,具有将以下内容按层排列的结构 :

[0196] (d1-1)Java 虚拟机 30

[0197] (d1-2) 中间设备,用于使得 Java 虚拟机能够工作

[0198] <Java 虚拟机 30>

[0199] Java 虚拟机 30 加载将应用程序构建在工作存储器中的 xlet 程序,对该 xlet 程序进行解码,并根据解码结果对位于其下的层进行控制。对位于其下的层进行控制,是通过向中间设备发布方法以便将该方法替换为 BD 重放装置所对应的函数调用,并且在该替换之后将该函数调用发布到重放控制引擎 32 来实现的。

[0200] <Java 虚拟机 30 的内部结构 >

[0201] 以下描述 Java 虚拟机 30 的内部结构。图 20 示出了 Java 虚拟机 30 的内部结构。如图所示, Java 虚拟机 30 包括 CPU 24、用户类加载器 52、方法区 53、工作存储器 54、线程 55a、55b、...55n 以及 Java 栈 56a、56b、...56n。

[0202] 用户类加载器 52 从本地存储器 26 等等中的 BDJA 目录中的 Java 档案文件中读出类文件,并将所读出的类文件存储在方法区 53 中。当应用程序管理器 36 指定文件路径并命令用户类加载器 52 执行读出时,用户类加载器 52 读出该类文件。在该文件路径指示本地存储器 26 的情况下,用户类加载器 52 将从本地存储器 26 中的构成应用程序的 Java 档案文件中的类文件读出到工作存储器 54 中。在该文件路径指示在文件系统中的目录的情况下,用户类加载器 52 将从 BD-ROM 或者本地存储器 18 中的构成应用程序的 Java 档案文件中的类文件读出到工作存储器 54 中。

[0203] 方法区 53 中存储由用户类加载器 52 从本地存储器 26 中读出的类文件。

[0204] 工作存储器 54 是所谓的堆区域,其中存储各种类文件的实例。工作存储器 54 中

存储与称为常驻应用程序的常驻类型应用程序和读入到方法区 53 中的类文件相对应的实例。这些实例中的每一个都是构成应用程序的 xlet 程序。当这样的 xlet 程序位于工作存储器 54 中时，应用程序变为可执行。

[0205] 线程 55a、55b、...55n 是执行的逻辑对象，以执行在工作存储器 54 中存储的方法。线程 55a、55b、...55n 使用局部变量或者操作数栈中存储的自变量作为操作数执行计算，并将计算结果存储在局部变量或者操作数栈中。图中的箭头 ky1、ky2 和 kyn 示意性示出从工作存储器 54 向线程 55a、55b、...55n 提供的方法。尽管执行的物理对象仅仅是 CPU，但是存在以下可能性，即在 Java 虚拟机 30 中存在多达 64 个作为执行的逻辑对象的线程。在 64 或者小于其的范围中，可以新生成线程和删除已有线程。当 Java 虚拟机 30 工作时，工作中的线程数量可以增加或降低。由于适当时线程数量可以增加，因此可以采用多个线程并行地执行一个实例，从而以较高速度执行该实例。

[0206] Java 栈 56a、56b、...56n 和线程 55a、55b、...55n 按照一对一成比例存在。Java 栈 56a、56b、...56n 中的每一个在自身中具有程序计数器（图中称为“PC”）和一个或多个帧。“程序计数器”示出了当前正在执行实例的哪个部分。“帧”是分配给对于方法的一个调用的栈类型区，并且其包括其中存储一个调用中的自变量的“操作数栈”，和由所调用方法使用的“局部变量栈”（在图中称为“局部变量”）。在每次进行调用时，将一帧栈入到 Java 栈 56a、56b、...56n 上；类似地，当方法进行自身递归调用时，栈入一帧。

[0207] 因此完成了对 Java 虚拟机的内部结构的描述。采用这种结构的 Java 虚拟机用作由事件驱动的执行对象（即事件驱动的执行对象）。因此完成对 Java 虚拟机的描述

[0208] <显示引擎 31>

[0209] 显示引擎 31 执行 AV 重放功能。重放装置的 AV 重放功能是源自 DVD 播放器和 / 或者 CD 播放器的传统功能，例如开始重放（播放）、停止重放（停止）、暂停重放（暂停开启），取消重放暂停（暂停关闭）、取消静音功能（静音关闭）、以指定速度向前（向前播放（速度）），以指定速度回倒（向后播放（速度））、改变音频（音频改变），改变子图像（字幕改变）、以及改变角度（角度变化）。为了实现 AV 重放功能，显示引擎 31 控制视频解码器 4、P 图形解码器 6、I 图形解码器 10 以及音频解码器 16，以便对已经输入到读缓冲器 2 的并且与期望时间相对应的 AVClip 的部分进行解码。对于这种期望时间，可以通过对由 PSR8 指示的部分进行解码，来重放与任意时间相对应的 AVClip 的一部分。

[0210] <重放控制引擎 32>

[0211] 重放控制引擎 32(PCE 32) 执行各种功能，例如 (i) 对播放列表的重放控制功能，和 (ii) 获得 / 设定 PSR 组 23 的状态的功能。PL 的重放控制功能使在由显示引擎 31 所执行的 AV 重放功能中，开始和停止根据当前 PL 信息和剪辑信息所执行的重放。根据来自 DVD-like 模块 29a-Java 平台 29b 的函数调用来执行这些功能 (i) 和 (ii)。

[0212] 以下描述由重放控制引擎 32 执行的处理与由 Java 虚拟机执行的处理之间的同步。当调用函数时，重放控制引擎 32 根据 PL 信息执行处理过程。在要重放的 AVClip 具有 15 分钟或者 30 分钟的重放时间段时，上述处理在该持续时间内持续。问题是，在 Java 虚拟机 30 返回成功响应的时刻与重放控制引擎 32 实际完成处理的时刻之间存在间隙。由于 Java 虚拟机 30 是时间驱动的处理的对象，因此其在调用之后立即返回一个指示重放是成功还是失败的响应；然而，重放控制引擎 32 在 15 或 30 分钟之后完成 AVClip 和播放项目的

重放。从而，如果将成功响应返回到应用程序的时刻用作参考点，则不可能检测 15 或 30 分钟之后所发生的处理的结束。在 PL 重放过程中执行快速向前或倒带的情况下，这种 15 或 30 分钟的重放时间段向前或向后移动，并且检测处理的结束变得更加困难。为了解决该问题，当播放项目的重放完成时，或者当 AVClip 的重放完成时，重放控制引擎 32 输出事件到应用程序，该事件指示播放项目或 AVClip 的重放完成。由于该输出，应用程序能够找到重放控制引擎 32 已经完成播放项目或 AVClip 的重放的时刻。

[0213] <模块管理器 33>

[0214] 模块管理器 33 读出 INDEX.BDMV 并从在 INDEX.BDMV 中包含的多条标题信息中选择一条标题信息，作为一条当前标题信息。然后模块管理器 33 读出该当前标题信息所指示的 BD-J 对象，并控制重放控制引擎以便根据写入在该 BD-J 对象中的播放列表信息来执行重放控制。此外，模块管理器 33 控制 Java 虚拟机，以便读出并执行写入在该 BD-J 对象中的 Java 档案文件。

[0215] 在此，在完成根据播放列表信息的数字流的重放的情况下，或者在用户调用菜单的情况下，模块管理器 33 读出定义另一标题的另一条标题信息，以便将该条标题信息选择作为一条当前标题信息。根据数字流的重放或者用户的菜单调用来选择另一条标题信息作为该条当前标题信息的处理将称为“标题转换”。

[0216] 当这种“标题转换”重放执行时，就可以实现图 21 所示的状态变换 (state shifting)。在图中拉长的圆表示标题。

[0217] 标题包括当加载 BD-ROM 时重放的“FirstPlayTitle”、构成顶部菜单的“TOP_menu Title”、和其他作为普通标题的“Title”。图中的箭头 jh1、jh2、jh3、jh4、jh5、jh6、jh7、和 jh8 用符号指示标题之间的转移。该图中的状态变换是当加载 BD-ROM 时重放“FirstPlayTitle”，转移到“TOP_menu Title”，然后等待在顶部菜单中的选择。

[0218] 光盘内容所特有的状态变换是，重复进行以下处理直到 BD-ROM 被退出为止：当用户执行操作来从菜单中进行选择时，根据该选择重放相应的标题，然后过程返回到 TOP_menu Title。这种状态变换是在上述的模块管理器 33 的控制下实现的。

[0219] 在以下列出的国际公开中，公开了标题和当前标题的选择，参考该公开以进行进一步详细描述：

[0220] 国际公开 WO 2005/036555

[0221] 国际公开 WO 2005/036540

[0222] 国际公开 WO 2005/036547

[0223] 因此完成了对 Java 虚拟机 30、显示引擎 31、重放控制引擎 32 和模块管理器 33 的描述。

[0224] Java 虚拟机对重放控制引擎 32 的控制是通过虚拟包施加的。为了通过虚拟包实现对重放控制引擎 32 的控制，重放装置包括诸如网络管理模块 37、虚拟文件系统单元 38、管理信息转换模块 39 和方法执行模块 40 之类的构成组件。以下将描述这些构成组件。

[0225] <网络管理模块 37>

[0226] 网络管理模块 37 根据来自应用程序的方法调用，从由电影提供商运作的 WWW 站点下载构建虚拟包所必需的数据。构建虚拟包所必需的数据包括合并管理信息文件、签名信息文件、替代 BD-ROM 上文件的文件或者添加到 BD-ROM 上的文件（例如，播放列表信息、剪

辑信息、AVClip 和 Java 档案文件）的文件。当在工作存储器 54 中的应用程序提出下载请求时，网络管理模块 37 通过网络下载虚拟包的构建所必需的数据，并将所下载的数据写入到本地存储器 18 中。

[0227] 〈虚拟文件系统单元 38〉

[0228] 虚拟文件系统单元 38 是属于第二层的构成组件之一，并且其根据来自应用程序的方法调用来构建虚拟包。虚拟包构建处理包括对构建该虚拟包的 AVClip 的状态管理，和虚拟包信息的生成处理。

[0229] 1. 虚拟包信息

[0230] 虚拟包信息是通过扩展在 BD-ROM 上记录的卷管理信息获得的。在此所述的卷管理信息是用于定义在记录介质上存在的目录文件结构的信息，其包括关于目录的目录管理信息和关于文件的文件管理信息。

[0231] 虚拟包信息通过将新文件信息添加到指示 BD-ROM 上目录文件结构的卷管理信息中，来扩展在 BD-ROM 上的目录文件结构。要添加到 BD 卷管理信息中的新的文件管理信息是存储在本地存储器 18 中的播放列表信息、剪辑信息、AVClip 和 Java 档案文件的文件管理信息。由于已经将这种文件管理信息添加到其中的虚拟包信息被生成并提供给重放控制引擎 32，因此重放控制引擎 32 能够进行识别，就如同存储在本地存储器 18 中的播放列表信息、剪辑信息、AVClip 和 Java 档案文件是存在于 BD-ROM 上。图 22 示出了由虚拟文件系统单元 38 执行的虚拟包信息的构建的示例。图的左上方是在 BD-ROM 上的目录文件结构，其与图 2 所示的相同。图的左下方是本地存储器 18 的目录文件结构，其与图 14 所示的相同。根据合并管理信息，将存储在本地存储器 18 上的播放列表信息、剪辑信息、AVClip 和 Java 档案文件的文件管理信息添加到 BD-ROM 上的卷管理信息中。

[0232] 更具体而言，执行以下过程：

[0233] (i) 将在本地存储器 18 中的播放列表的文件管理信息 (00002.MPLS) 添加到 BD 卷管理信息中的 MPLS 目录的目录管理信息中。

[0234] (ii) 将在本地存储器 18 中的剪辑信息 #2、剪辑信息 #3 和剪辑信息 #4(00002.CLPI、00003.CLPI、00004.CLPI) 的文件管理信息添加到 BD 卷管理信息中的 CLPI 目录的目录管理信息中。

[0235] (iii) 将在本地存储器 18 中的 AVClip#2、AVClip#3 和 AVClip#4 的文件管理信息 (00002.M2TS、00003.M2TS、00004.M2TS) 添加到 BD 卷管理信息中的 STREAM 目录的目录管理信息中。

[0236] 应该注意的是，在 BD-ROM 上存在与合并管理信息文件所指定的虚拟包中的文件具有相同文件路径的文件的情况下，将在本地存储器中的文件管理信息写在卷管理信息中。采用该安排，将优先在虚拟包中使用存储在本地存储器中并由合并管理信息所引用的文件。

[0237] 因此，已经获得了虚拟包信息。通过对卷管理信息进行这种添加，获得了虚拟包信息。

[0238] 将采用这种方式生成的虚拟包信息提供给重放控制引擎 32。从而，重放控制引擎 32 通过指定在访问文件时指示虚拟包的定位器，能够将存储在本地存储器 18 中的播放列表信息、剪辑信息和 AVClip 等同地看作记录在 BD-ROM 上的播放列表信息、剪辑信息和

AVClip。

[0239] 更具体而言，重放控制引擎 32 使用由虚拟包信息所指示的虚拟包的定位器来请求文件访问。响应于该请求，虚拟文件系统单元 38 根据合并管理信息文件，判断作为访问请求目标的文件的实体是记录在 BD-ROM 上还是在本地存储器 18 中。在判断结果为文件的实体记录在本地存储器 18 中的情况下，虚拟文件系统单元 38 将访问请目标转换为由合并管理信息文件指定并存储在本地存储器 18 中的文件的文件路径。

[0240] 因此完成对虚拟包信息的生成的描述。

[0241] 以下将解释虚拟包信息更新的定时。

[0242] 对按照图 21 的箭头 jh1、jh2、jh3、jh4 等等所指示的参考数字的数值的顺序来执行转移，然后退出 BD-ROM 的情况进行解释。在该情况下，从加载 BD-ROM 时到退出 BD-ROM 时的连续时间线能够看作时整个光盘的时间轴。图 23A 示出了整个光盘的时间轴。图 23B 示出了该时间轴的配置。如图 23B 所示，整个光盘的时间轴包括重放 FirstPlayTitle 的间隔、重放 Top_menu Title 的间隔、重放标题 #1 的间隔等等。这些标题的重放间隔是根据以下思路进行定义的：由于每个标题是由单个 BD-J 对象构成的，因此将特定 BD-J 对象有效的时间段看作一个标题的重放间隔。在标题的重放间隔之间的过渡部分，换而言之，从一个标题转换到另一个标题（以下称为标题变化）处的部分，是更新虚拟包信息的时间段。

[0243] 图 24 示出了当发生标题变化时如何更新虚拟包信息。

[0244] 图 24 的第一层级示出了在时间轴上的标题的播放间隔。第二层级示出了构成该标题的 Java 应用程序的执行状态。第三层级示出了数字流的重放状态。第四层级示出了虚拟文件系统单元 38 的状态。

[0245] 在第一层级中所示的标题 #1 的重放时间间隔中，运行和重放由 BD-J 对象所指定的 Java 应用程序和 AVClip，如第二层级和第三层级所示。在标题的这种重放间隔中更新虚拟包信息的情况下，标题的构成组件在执行处理或者重放处理过程中可以相互改变位置。当构成组件相互改变位置时，重放装置工作异常，并且存在可能发生诸如重放图像中断之类的不可恢复的情况的可能性。

[0246] 为了处理该情况，当在标题 #1 的重放过程中请求虚拟包信息的更新时，虚拟文件系统单元 38 进入“准备中状态”，在其中，在构建虚拟包所必需的处理过程中，仅仅执行一些处理，这些处理在标题的重放时间间隔过程中能够并行处理，而不改变虚拟包的构建组件。在准备中阶段中的处理完成之后，虚拟文件系统单元 38 变换到“准备”状态并且等待直到发生标题变化。当发生标题变化时，虚拟文件系统单元 38 进入“更新”状态，在下一个标题的重放间隔之前，通过将由合并管理信息文件所引用的文件的文件管理信息添加到 BD 卷管理信息中来执行更新虚拟包信息的处理。在更新完成之后，虚拟文件系统单元 38 进入“稳定”状态，下一个标题的重放开始。在这种状态中，在选择从标题变化后的 Top_menuTitle 跳转回到标题 #1 的情况下，将重放标题 #1 以反映所更新的信息。

[0247] 如上所述，对于标题变化，由于虚拟包信息被更新，即使是当标题 #1 正在重放时请求对虚拟包信息的更新，标题的构成组件也不会在标题 #1 正在重放时改变；因此，在重放过程中就不可能发生异常。

[0248] 以下将参考图 25 到 28 描述虚拟包的构建的细节。

[0249] 由例如提供商所安装的服务器来提供构成虚拟包的文件。在 Java 应用程序的控

制下，通过诸如因特网之类的网络下载这些文件。应该注意的是，在下载中，使用因特网中通常使用的通信协议，例如 HTTP 或者 HTTPS。

[0250] 图 25 示出了 Java 应用程序如何是如何请求服务器发送构建虚拟包的文件。在图的中间所示的 ROOT 目录下的目录树指示在本地存储器中的目录结构。Java 应用程序请求服务器应该通过发送记录在本地存储器中的合并管理信息文件来下载构成虚拟包的文件。

[0251] 应该注意的是，当将第一次将 BD-ROM 安放在重放装置中时，本地存储器中没有记录合并管理信息文件，并且没有虚拟包被构建。在这种要第一次构建虚拟包的情况下，Java 应用程序通过向服务器通知光盘 ID 来请求服务器应该下载构成虚拟包的文件。

[0252] 接下来，图 26 示出了 Java 应用程序是如何将从服务器发送的文件记录在本地存储器中的。

[0253] 如图 26 所示，Java 应用程序在 disc#1 目录下生成与该光盘 ID 相对应的新目录（新 MF 目录），并将在响应于该请求而从服务器发送的文件中的新的合并管理信息文件和与该管理信息相对应的签名信息文件，记录到新生成的目录中。其他文件立即记录在 disc#1 目录下。应该注意的是，通过使用提供文件 I/O 功能的 API 将定位器指定到本地存储器中，来执行文件的记录。应该注意的是，如果直接在 disc#1 目录下的已有合并管理信息文件和签名信息文件没有与要新下载的合并管理信息文件和签名信息文件相同的文件名，则也可以将所下载文件直接记录在 disc#1 目录下而不生成新的目录。

[0254] 图 27 示出了 Java 应用程序是如何请求虚拟文件单元应该更新虚拟包信息的。如图所示，已经获得构成虚拟包的文件的 Java 应用程序，使用在新 MF 目录中记录的新合并管理信息文件的文件路径、签名信息文件的文件路径和光盘 ID 作为自变量，来调用更新请求方法，从而更新虚拟包信息。

[0255] 图 28 示出虚拟包信息是如何更新的。

[0256] 当调用更新请求方法时，虚拟文件系统单元 38 转换为准备中状态，执行能够在不改变与当前正在重放的标题相关的虚拟包的构成组件的情况下执行的一些处理。图的左侧所示的目录树示出了在准备中状态中的本地存储器。

[0257] 更具体而言，在准备中状态下执行以下处理：

[0258] (1) 判断由作为自变量的文件路径所指示的新合并管理信息文件是否可靠。

[0259] (2) 判断由新合并管理信息文件所引用的文件是否存在于指定存储位置处。

[0260] (3) 将由新合并管理信息文件所引用的文件和新合并管理信息文件的文件属性改变为只读属性。

[0261] 当完成在准备中状态下执行的这些处理时，虚拟文件系统单元 38 转换为准备状态。应该注意的是，将要被映射到虚拟包上的在本地存储器中的文件的属性设定为只读属性；因此，在 Java 应用程序通过使用提供文件 I/O 功能的 API 将定位器指定到本地存储器中，来请求应该将数据写入到这些文件中的一些中的情况下，API 将返回指示写入不被允许的异常。

[0262] 当其后标题变化已经发生时，虚拟文件系统单元 38 将存在于在 disc#1 目录中的文件替换为新合并管理信息文件和签名信息文件，并在替换之后，通过根据在 disc#1 目录中的合并管理信息文件将文件管理信息添加到 BD 卷管理信息中，来将记录在本地存储器中的文件映射到虚拟包上。此时，在本地存储器中的由旧合并管理信息文件引用但不由新

合并管理信息文件引用的一些文件的属性从只读属性改变为可读 / 可写属性。因此完成对虚拟包信息的构建的详细描述。

[0263] 以下参考流程图，描述根据本发明的重放装置所执行的重放处理的细节。图 29 是示出由模块管理器 33 所执行的标题重放控制的处理的流程图。该流程图示出在将 BD-ROM 安放在重放装置之后和将其退出之前要执行的处理过程。在步骤 S11 中，将在 INDEX.BDMV 文件中指定的 FirstPlayTitle 设定为当前标题，在执行步骤 S11 的处理之后，执行步骤 S12 到 S21 的循环处理。

[0264] 在步骤 S12 到 S21 的循环处理中，在步骤 S12 到 S15 中的处理是重放当前标题的过程。在步骤 S12，从 INDEX.BDMV 中获得与当前标题相对应的一条标题信息，将在所获得的标题信息中指定的 BD-J 对象作为当前 BD-J 对象。在步骤 S13 中，控制重放控制引擎 32，以便根据其参考值被写入到当前 BD-J 对象中的一条播放列表信息来执行 PL 重放。在步骤 S14，控制 Java 平台 29b，以便运行其标识符在当前 BD-J 对象的应用程序管理表中被指定的 Java 应用程序。在步骤 S15，控制 Java 平台 29b，以便终止其标识符没有在当前 BD-J 对象的应用程序管理表中被指定的其他 Java 应用程序。

[0265] 在步骤 S16 到 S19 中的处理是用于判断标题是否已经结束的过程。在以下情况之一已经发生的情况下判定标题已经结束 ：完成根据播放列表的 PL 重放 (步骤 S16) ；标题调用，例如由于用户操作造成的菜单调用指令，已经发生 (步骤 S17) ；诸如脚本转移之类的标题跳转已经发生 (步骤 S18) ；以及作为标题的主要组件的应用程序已经结束 (步骤 S19)。

[0266] 在判定标题已经结束的情况下，执行在步骤 S20 和 S21 中的处理。根据在步骤 S16 到 S19 中的哪一个判定用于确定标题结束，来指定下一个标题 (步骤 S20)，并将所指定的标题设定为新的当前标题 (步骤 S21)。作为上述处理的结果，在 BD-ROM 的插入与退出之间执行标题重放。

[0267] 以下描述由重放控制引擎 32 所执行的播放列表重放的细节。图 30 是流程图，示出了由重放控制引擎 32 所执行的播放列表重放处理的处理过程。该流程图示出了根据由模块管理器 33 所命令的一条播放列表信息，从由播放列表所指定的 AVClip 的开始到结束的重放的处理过程。

[0268] 首先，在作为指定播放列表信息的实体的 MPLS 文件是在本地存储器中的已经根据合并管理信息文件进行映射的文件的情况下，检查所指定文件是否被窜改 (步骤 S31)。在步骤 S31 中判定所指定文件已经被窜改的情况下，过程退出流程图，从而终止处理。在判定所指定文件没有被窜改的情况下，继续进行步骤 S32 中和该步骤之后的处理。

[0269] 在步骤 S32 中执行如下处理，将在播放列表信息中的第一条播放项目信息设定为要进行处理的一条播放项目信息 (以下简称为“播放项目信息 i”)，在执行步骤 S32 的处理之后，执行从步骤 S33 到 S41 的循环处理。

[0270] 在从步骤 S33 到 S41 的循环处理中，控制变量是变量 i，在步骤 S33 到 S41 中的处理被执行后，控制变量 i 递增 (步骤 S41)，直到变量 i 超过播放项目的数量 (“播放项目的数量”)

[0271] 在从步骤 S33 到 S41 的循环处理中，从步骤 S33 到 S35 中的处理是用于重放主路径的处理。在步骤 S33 的处理中，将写入到播放项目信息 i 中的 Clip_Information_file_name 中的 AVClip 作为用作重放目标的 AVClip j。在步骤 S34 的处理中，在作为与 AVClip

j 相对应的一条剪辑信息的实体的 CLPI 文件是在本地存储器中的已经根据合并管理信息文件进行映射的文件的情况下，检查在本地存储器中的这一文件是否已经被窜改。当一个或多个文件已经被窜改时，过程退出流程图，从而终止处理。在步骤 S35 中的处理中，命令驱动装置和解码器重放 AVClip j 中从 PlayItem.In_time 到 PlayItem.Out_time 的部分。

[0272] 在步骤 S36 到步骤 S39 中的处理是用于重放子路径的过程。在步骤 S34 中，判断是否存在将播放项目信息 i 指定为 Sync_PlayItem_id 的子播放项目 k。如果其不存在，则过程进入步骤 S40。

[0273] 如果其存在，则在步骤 S37 中的处理中，将写入到该子播放项目 k 的 Clip_Information_file_name 中的 AVClip 作为 AVClip h。在步骤 S38 的处理中，在作为与 AVClip h 相对应的该条剪辑信息的实体的 CLPI 文件是在本地存储器中的已经根据合并管理信息文件进行映射的文件的情况下，检查在本地存储器中的这一文件是否已经被窜改。当这一文件已经被窜改时，过程退出流程图，从而终止处理。在步骤 S39 的处理中，命令驱动装置和解码器重放 AVClip h 中从 sync_Start_PTS_of_PlayItem 到 Out_time 的部分，然后过程进入步骤 S40。

[0274] 对构成播放列表信息的所有条播放项目重复上述处理。因此，执行由播放列表所定义的流序列的重放。

[0275] 应该注意的是，采用以下过程执行在步骤 S31、步骤 S34 和步骤 S38 中对文件是否已经被窜改的检查，具体而言：

[0276] (a) 从合并管理信息文件中读出要检查的文件的散列值。

[0277] (b) 使用要检查的文件计算散列值。

[0278] (c) 如果在 (a) 和 (b) 中获得的值相等，则判定文件没有被窜改。如果在 (a) 和 (b) 中获得的值不相等，则判定文件已经被窜改。

[0279] 可接受的是忽略对流文件 (xxxxx.M2TS) 的窜改的检查。原因是因为流文件的文件大小通常比其他文件大，根据文件大小，需要极其长的时间来计算散列值。另外，通常使用记录在 BD-ROM 上的密钥对该流文件本身进行加密，与其他文件相比，其更加难以被窜改。

[0280] 以下参考流程图，描述由根据本发明的重放装置所执行的构建虚拟包的过程的细节。将虚拟包的构建划分为三个阶段，例如由 Java 应用程序下载构成虚拟包的文件、虚拟文件系统单元 38 进行的准备、以及虚拟文件系统单元 38 进行的更新。

[0281] 首先，详细描述要由 Java 应用程序所执行的构建虚拟包的文件的下载的过程。图 31 是示出要由 Java 应用程序所执行的构建虚拟包的文件的下载的过程的流程图。

[0282] 首先，Java 应用程序通过将记录在本地存储器中的、与 BD-ROM 的光盘 ID 相对应的目录中的合并管理信息文件发送到服务器，来请求应该发送文件 (步骤 S291)。在步骤 S292，Java 应用程序等待直到从服务器接收到文件。应该注意的是，有时在本地存储器中没有记录合并管理信息文件，例如当该 BD-ROM 是首次安放在该重放装置中时。在该情况下，可接受的是具有以下安排：其中在步骤 S291 的处理中，Java 应用程序通过将 BD-ROM 上的光盘 ID 通知给服务器，来请求服务器应该发送文件。

[0283] 当在步骤 S292 的待命过程中从服务器接收到文件数据时，在步骤 S293 和 S294 中将文件记录在本地存储器中。在步骤 S293 中，在与该光盘 ID 相对应的目录中生成新目录，

并将所接收的合并管理信息文件和签名信息文件记录在该新目录中。在步骤 S294，将已经下载的 AVClip、剪辑信息、播放列表信息和 Java 档案文件写入与该光盘 ID 相对应的目录中。

[0284] 当已经完成将文件记录在本地存储器中时，Java 应用程序使用在新目录中的合并管理信息文件和签名信息文件的文件路径作为自变量，调用更新请求方法（步骤 S295）。在步骤 S296 中，Java 应用程序等待从该方法返回的值。

[0285] 当返回值为 FALSE 的情况下（步骤 S296 ：是），停止虚拟包的构建过程。当返回值为不是 FALSE 的情况下（步骤 S296 ：否），虚拟文件系统单元 38 进入准备状态，并且当转换标题时更新虚拟包（步骤 S297）。

[0286] 以下详细描述要由虚拟文件系统单元 38 执行的准备处理。图 32 是示出要由虚拟文件系统单元 38 执行的准备处理的流程图。

[0287] 该流程图示出在已经调用上载请求方法的情况下已经转换为准备中状态的虚拟文件系统单元 38 要执行的处理过程。

[0288] 在步骤 S51 中，使用用作调用该方法的自变量的文件路径读出合并管理信息文件和签名信息文件，在步骤 S51 的处理之后，执行步骤 S52 到 S54 中的处理，在处理中，判断构建虚拟包是否是可接受的。在步骤 S52 中，判断在步骤 S51 中读出的合并管理信息文件是否已经被窜改。在该合并管理文件判定为已经被窜改的情况下，将不允许进行虚拟包的构建。

[0289] 更具体而言，执行以下过程 ：

[0290] (a) 使用重放装置所持有的合并证书中的公钥来解密在签名信息文件中的加密散列值。

[0291] (b) 计算合并管理信息文件的散列值。

[0292] (c) 如果通过 (a) 中的解密过程获得的值等于在 (b) 中计算得到的值，则判定该合并管理信息文件没有被窜改并且是可靠的。如果通过 (a) 中的解密过程获得的值不等于在 (b) 中计算得到的值，则判定该合并管理信息文件已经被窜改并且是不可靠的。

[0293] 步骤 S53 是用于判断是否允许调用该方法的 Java 应用程序来更新虚拟包的处理。在不允许调用者更新虚拟包的情况下，将不允许虚拟包的构建。

[0294] 步骤 S54 是用于判断由该合并管理信息文件指定的文件是否实际存在于本地存储器中的处理。在不存在所指定文件的情况下，将不允许虚拟包的构建。

[0295] 在步骤 S52 到 S54 中的一个中判断结果为不允许虚拟包的构建的情况下，将返回值“FALSE”返回到调用该方法的 Java 应用程序，然后处理终止。

[0296] 另一方面，在步骤 S52 到 S54 中的一个中判断结果为允许虚拟包的构建的情况下，则要执行步骤 S55 中的处理。

[0297] 步骤 S55 是用于读出合并管理信息文件、签名信息文件和由该合并管理信息文件所指定的文件并将它们的属性改变为只读属性的处理。因此完成对准备处理的细节描述。

[0298] 应该注意的是，在步骤 S52 到 S54 中的一个中判断结果为不允许虚拟包的构建的情况下，可以接受的是具有一种安排，其中，将 BD-ROM 的卷管理信息提供给调用该更新请求方法的 Java 应用程序作为虚拟包信息，并且终止该更新请求方法的处理。采用该安排，如果合并管理信息文件已经被窜改并且处于非法状态下时，重放装置能够禁止包含本地存

储器中的文件的虚拟包的构建，并且重放装置还能够采用与访问虚拟包相同的方式来访问 BD-ROM 上的文件。

[0299] 以下将详细描述由虚拟文件系统单元 38 执行的更新处理。图 33 是示出由虚拟文件系统单元 38 执行的更新处理的流程图。

[0300] 该流程图示出在由于更新请求方法而完成准备处理之后发生标题变化的情况下，由虚拟文件系统单元 38 所执行的处理过程。

[0301] 首先，将由作为该方法自变量的文件路径所指定的合并管理信息文件和签名信息文件，移动到与当前正在重放的 BD-ROM 的光盘 ID 相对应的本地存储器中的目录中。此时，在该合并管理信息文件和签名信息文件已经存在于作为移动目的地的目录中的情况下，采用新的合并管理信息文件和签名信息文件改写已有文件（步骤 S61）。随后，将由在本地存储器 18 中的合并管理信息文件所指定的一条播放列表信息的文件管理信息，添加到在 BD 卷管理信息中的 PLAYLIST 目录中的目录管理信息中（步骤 S62）。

[0302] 随后，执行步骤 S63 到步骤 S67 的循环处理。在该循环处理中，对在本地存储器 18 中存储的多条剪辑信息中的每一个重放在步骤 S64 到步骤 S66 中的处理（步骤 S63、步骤 S67）。

[0303] 在该循环处理中，将由合并管理信息文件所指定的多条剪辑信息中的一个作为剪辑信息 x，将该剪辑信息 x 的文件管理信息添加到在 BD 卷管理信息中的 CLIPINF 目录中的目录管理信息中（步骤 S64）。此外，指定与该剪辑信息 x 相对应的 AVClip x（步骤 S65），并且将 AVClip x 的文件管理信息添加到 BD 卷管理信息中的 STREAM 目录的目录管理信息中（步骤 S66）。

[0304] 当对所有条剪辑信息和所有 AVClip 重复上述处理时，将这些条剪辑信息和 AVClip 的文件管理信息添加到 BD 卷管理信息中。将作为该添加的结果而获得的这种 BD 卷管理信息作为虚拟包信息。将这种虚拟包信息提供给调用该更新请求方法的 Java 应用程序（步骤 S68），然后处理完成。因此完成对更新处理的细节描述。

[0305] 如迄今所述，根据本发明，根据签名信息来验证合并管理信息的可靠性。在合并管理信息确认为不可靠的情况下，则防止使用记录在本地存储器中的文件来构建虚拟包。采用该安排，即使是本地存储器中的某些数据已经被窜改而虚拟包的构建没有被执行时，也可以避免执行被窜改的非法数据，和结合记录在只读记录介质上的数据对其进行重放的情况。因此，就不可能遇到由于数据的非法使用所造成的不利情况。

[0306] 此外，当虚拟包被构建时，将作为可读 / 可写记录介质的本地存储器中的文件的属性锁定为只读属性；因此，就可以防止在构建了虚拟包之后，由于应用程序对未准备好的文件的访问造成的文件被重写。采用该安排，就可以防止虚拟包信息与在本地存储器中的文件实体不一致的情况。

[0307] 第二实施例

[0308] 第二实施例涉及对在执行标题调用时的改进。标题调用要暂时暂停当前标题的重放，重放所调用的标题，以及在对所调用标题的执行完成之后，继续对原始播放的标题的重放。

[0309] 由于将继续原始播放的标题的重放作为前提，因此当执行标题调用时，重放控制引擎 32 将存储在 PSR 中并用于重放控制的系统参数保存在备份 PSR 中，在所调用的标题被

重放之后，重放控制引擎 32 将在保存在备份 PSR 中的参数恢复到 PSR 中。

[0310] 以下是存储在 PSR 中的系统参数列表。PSR(0) 到 PSR(12) 中存储指示重放状态的系统参数。PSR(13) 到 PSR(19) 中存储由播放器设定的作为优选的系统参数。PSR(20) 到 PSR(32) 是用于备份的 PSR。

[0311] PSR(0) :IG 流号

[0312] PSR(1) :音频流号

[0313] PSR(2) :PG 流 / 文本字幕流号

[0314] PSR(3) :角度号

[0315] PSR(4) :当前正在重放的标题号

[0316] PSR(5) :当前正在重放的章节号

[0317] PSR(6) :当前正在重放的播放列表标识符

[0318] PSR(7) :当前正在重放的播放项目标识符

[0319] PSR(8) :重放时间信息

[0320] PSR(9) :导航计时器

[0321] PSR(10) :选择键信息

[0322] PSR(11) :IG 流中的当前页标识符

[0323] PSR(12) :PG 流中的用户样式标识符和文本字幕流

[0324] PSR(13) :父母级别 (paretal level)

[0325] PSR(14) :字幕支持信息

[0326] PSR(15) :播放器设定值 (音频)

[0327] PSR(16) :音频流的语言代码

[0328] PSR(17) :PG 流和文本字幕流的语言代码

[0329] PSR(18) :菜单的语言代码

[0330] PSR(19) :播放器的版本信息

[0331] PSR(20) :PSR(0) 的备份

[0332] PSR(21) :PSR(1) 的备份

[0333] PSR(22) :PSR(2) 的备份

[0334] PSR(23) :PSR(3) 的备份

[0335] PSR(24) :PSR(4) 的备份

[0336] PSR(25) :PSR(5) 的备份

[0337] PSR(26) :PSR(6) 的备份

[0338] PSR(27) :PSR(7) 的备份

[0339] PSR(28) :PSR(8) 的备份

[0340] PSR(29) :PSR(9) 的备份

[0341] PSR(30) :PSR(10) 的备份

[0342] PSR(31) :PSR(11) 的备份

[0343] PSR(32) :PSR(12) 的备份

[0344] 在此，如果在执行标题调用之后对虚拟包信息进行更新，则在标题调用之前与之后，在虚拟包信息中将会有些不同。

[0345] 当要继续调用其他标题（即，被调用标题）的标题（即，调用标题）时，虚拟包信息已经改变；因此，就存在以下可能性：当试图采用备份值来重放调用标题时，可能出现错误，因为备份值保持与以前相同。因此，当 Java 应用程序请求应该更新虚拟包信息时，将可以通过清除备份 PSR 来避免这种问题情况。然而，可能存在根据在虚拟包信息中的合并管理信息文件，所述改变不会造成任何影响的一些情况；因此，可接受的是，将在系统参数中的值是否应该被清除的问题留给 Java 应用程序。

[0346] 图 34 是流程图，示出当暂时暂停当前标题的重放以便执行标题调用时由重放控制引擎所执行的处理过程，和当原始播放标题的重放继续时由重放控制引擎所执行的处理过程。

[0347] 为了暂时暂停当前标题的重放，在步骤 S71 中将 SPRM(0) 到 SPRM(12) 保存到 SPRM(20) 到 SPRM(32) 中。

[0348] 当在完成被调用标题的重放之后继续原始播放标题的重放时，在步骤 S81 中判断在暂停过程中虚拟包信息是否已经被更新。

[0349] 在虚拟包信息没有被更新的情况下，将 SPRM(20) 到 SPRM(32) 恢复到 SPRM(0) 到 SPRM(12)（步骤 S83）。在虚拟包信息已经被更新的情况下，将 SPRM(20) 到 SPRM(32) 初始化（步骤 S82），并且将执行步骤 S83 中的处理。

[0350] 如迄今所述，根据本实施例，在执行了标题调用之后虚拟包信息被更新的情况下，将备份 PSR 的内容初始化；因此就不存在重放控制引擎根据标题调用之前的 PSR 而执行错误的重放处理的可能性。因此，就可以实现重放控制引擎的稳定工作。

[0351] 应该注意的是，采用以下安排也是可接受的：其中，当虚拟包信息被更新时，在系统端强制将系统参数的值清除，而不是将在是否应该清除备份 PSR 中的值的问题留给 Java 应用程序。

[0352] 第三实施例

[0353] 在第一实施例中，已经描述以下配置：其中，将在本地存储器中并相应于组织 ID 和 BD-ROM 的光盘 ID 的目录用作用于存储构建虚拟包的文件的位置。在第三实施例中，描述了以下配置：其中，采用记录在本地存储器中的任意目录中的文件来构建虚拟包。

[0354] 通过 Java 应用程序指定光盘 ID 和在本地存储器中的目录路径实现这种虚拟包构建，在本地存储器中，将用于虚拟包构建的文件记录为用于更新请求方法的自变量。

[0355] 在根据第三实施例的重放装置中，虚拟文件系统单元 38 保持虚拟包管理表。图 35 示出了虚拟包管理表的示例。虚拟包管理表示出了光盘 ID，其与指示在更新请求方法中所指定的本地存储器中的目录的路径相对应，来标识 BD-ROM。当响应于更新请求而生成虚拟包信息时，虚拟文件系统单元 38 能够根据在虚拟包管理表中记载的信息，指定在虚拟包中使用的文件的存储目的地。

[0356] 以下描述由根据本实施例的虚拟文件系统单元 38 所执行的虚拟包构建处理。图 36 是示出由根据第三实施例的虚拟文件系统单元 38 所执行的虚拟包构建处理过程的流程图。

[0357] 该流程图示出了在 Java 应用程序已经发布更新请求方法的情况下，由虚拟文件系统单元 38 所执行的处理过程。

[0358] 在步骤 S91，判断指定为更新处理方法的自变量的光盘 ID 是否已经在虚拟包管理

表中记载。

[0359] 在所指定的光盘 ID 已经在虚拟包管理表中记载的情况下（步骤 S91 :是），在执行步骤 S92 到 S94 中的处理过程之后，执行在步骤 95 到 S99 中的处理过程。另一方面，在所指定的光盘 ID 没有在虚拟包管理表中记载的情况下，过程进而执行步骤 S95 到 S99 的处理过程。

[0360] 在步骤 S92 到 S94，从虚拟包管理表中删除具有所指定光盘 ID 的条目 (entry)（步骤 S92），并判断由所指定光盘 ID 所标识的 BD-ROM 是否插入了 BD-ROM 驱动器 1 中（步骤 S93）。在由所指定光盘 ID 所标识的 BD-ROM 已经插入的情况下（步骤 S93 :是），过程进入步骤 S94。在其他情况下（步骤 S93 :否），过程进入步骤 S95。在步骤 S94，与所指定光盘 ID 相对应的虚拟包的构建被暂停。

[0361] 步骤 S95 是用于判断所指定目录的签名是否可靠的处理过程。在当前示例中，用于判断所指定目录的签名是否可靠的处理是 :(a) 检查在所指定目录下的合并管理信息文件是否未被窜改，以及然后 (b) 检查由合并管理信息文件所指定的文件是否未被窜改。

[0362] 为了确定合并管理信息文件没有被窜改，执行以下过程 :

[0363] (a-1) 对虚拟包管理表计算散列值。

[0364] (a-2) 使用由重放装置持有的合并证书中的公钥来对在所指定目录下的签名信息文件中的加密散列值进行解密。

[0365] (a-3) 将所计算的散列值与解密的签名值进行比较，如果所述值相同，则确定合并管理信息文件没有被窜改。

[0366] 为了确定由合并管理信息文件所指定的文件没有被窜改，执行以下过程 :

[0367] (b-1) 对由合并管理文件所指定的每个文件计算散列值。

[0368] (b-2) 将所计算的散列值与写入到合并管理信息文件中的散列值进行比较，如果所述值相同，则确定由合并管理信息文件所指定的文件没有被窜改。

[0369] 作为步骤 S95 的判断结果，在所指定目录的签名是可靠的情况下（步骤 S95 :是），过程进入步骤 S96。在其他情况下，过程终止。

[0370] 在步骤 S96 中，使用被指定为更新请求方法的自变量的光盘 ID 和目录路径，将新的条目添加到虚拟包管理表中。在步骤 S97，判断由所指定光盘 ID 所标识的 BD-ROM 是否插入了 BD-ROM 驱动器 1。在由所指定光盘 ID 所标识的 BD-ROM 已经插入的情况下（步骤 S97 :是），过程进入步骤 S98。在其他情况下（步骤 S97 :否），处理终止。在步骤 S98 中，将其路径作为自变量而被指定的本地存储器中的目录下的内容读出，并且将属性设定为只读属性。在步骤 S99 中，采用其文件路径被指定的本地存储器和 BD-ROM 中的目录来构建虚拟包。因此完成根据第三实施例的用于构建虚拟包的处理过程的描述。

[0371] 应该注意的是，上述处理过程是用于响应于更新请求而构建虚拟包的过程 ;然而，对于在 BD-ROM 新插入重放装置的情况下构建虚拟包的处理过程而言，从虚拟包管理表中搜索与所插入的 BD-ROM 的光盘 ID 相对应的目录路径，并且采用所检测到的目录路径作为指定目录路径来执行在图 36 的步骤 95 中和该步骤之后的处理过程，从而能够构建虚拟包。

[0372] 此外，在响应于更新请求而构建虚拟包的处理过程中，作为在步骤 S95 中用于判断所指定目录的签名是否可靠的处理，首先 (a) 检查在所指定目录下的合并管理信息文件是否未被窜改，以及然后 (b) 检查由合并管理信息文件所指定的文件是否未被窜改 ;然而，

对于在 BD-ROM 新插入重放装置的情况下构建虚拟包的处理过程而言，还可接受的是仅仅通过以下处理来在步骤 S95 中判断所指定目录的签名是否可靠：(a) 检查在所指定目录下的合并管理信息文件是否未被窜改，以便通过在步骤 S96 到 S99 中的处理构建虚拟包。在这种情况下，当 Java 应用程序作出对其文件路径已经写入到合并管理信息文件中的文件的访问请求时，执行如下处理 (b) 检查由合并管理信息文件所指定的文件是否未被窜改。仅仅在所述文件确认没有被窜改的情况下，才允许 Java 应用程序访问所指定文件。采用以下安排，减少了在插入并加载 BD-ROM 之后所必需的处理数量：其中，在访问文件时而不是在构建虚拟包时执行散列值计算处理部分。因此，可以实现使启动更加快速的效应。

[0373] 以下描述正在构建虚拟包时，Java 应用程序为本地存储器指定定位器，并访问文件以将数据写入该文件的情况下要执行的操作。

[0374] 图 37 是示出用于控制对文件的写入访问的 API 的处理过程的流程图。当 Java 应用程序采用指定本地存储器定位器请求写入访问时，判断目录属性是否因为正在构建虚拟包而被设定为只读属性（步骤 S101）。在包含所指定文件的目录的属性被锁定在只读属性时（步骤 S101：是），将指示不允许写入该文件的异常返回到 Java 应用程序，并终止处理（步骤 S102）。在其他情况下（步骤 S101：否），过程进入步骤 S103。

[0375] 在步骤 S103，执行对本地存储器中的文件进行写入访问的处理。

[0376] 根据上述处理过程，如果指定为访问目标的文件是构建虚拟包的文件，并且当作出访问请求时该虚拟包正在构建，则不允许访问该文件。

[0377] 在此，即使是在当正在采用在 contsID#1 目录的指定构建虚拟包时，图 38 所示的 contsID#1 目录下的文件有错误时，由于在正在构建虚拟包时 contsID#1 目录的属性被设定为只读属性，因此不可能校正该文件的错误。

[0378] 在这种情况下，具有文件校正功能的 Java 应用程序采用提供文件 I/O 功能的 API 生成称为 contsID#2 的目录，如图 38 所示，并且将在 contsID#1 目录下的文件复制到 contsID#2 目录中，从而校正文件中的错误。此时，如果用于 AV 数据的 M2TS 文件的内容中没有改变，则可接受的是在 contsID#1 目录下生成对 M2TS 文件的符号链接，而不是在 contsID#2 目录下生成复制文件。

[0379] 在 contsID#2 下生成文件之后，Java 应用程序使用 contsID#2 目录的路径（在图 38 中所示的示例中为“ROOT/organization#1/disc#1/contsID#2”）作为自变量发布更新请求方法。

[0380] 因此，根据图 36 中所示的流程图，将采用在 contsID#1 目录下的文件的虚拟包的构建暂停，然后采用在 contsID#2 目录下的文件构建虚拟包。当已经构建了新的虚拟包时，将在本地存储器中的 contsID#2 的属性和其下内容的属性设定为只读属性。

[0381] 应该注意的是，尽管在本实施例中使用符号链接，采用非直接引用的机制也是可以接受的，在该机制中，“ROOT/organization#1/disc#1/contsID#2/00002.M2TS”文件的内容为诸如“../contsID#2/00002.M2TS”之类的字符串。

[0382] 如迄今所述，根据本实施例，可以采用在本地存储器中的任意目录的指定来构建虚拟包。此外，在通过将记录在多个记录介质上的文件进行合并来构建虚拟包的情况下，采用以下安排可以防止在虚拟包构建过程中将文件重写：其中，将在作为可读 / 可写记录介质的本地存储器中所存储的文件的属性锁定为只读属性。因此，可以防止由该应用程序本

身或者其他应用程序所造成的、在文件中的非法改变可能导致的错误。

[0383] 此外，即使是在虚拟包没有创建时实体文件被窜改，也可以通过判断签名是否可靠来检测实体文件的窜改，并且避免虚拟包的构建。因此，可以在由于采用非法文件而可能导致的故障发生之前，将其阻止。

[0384] 应该注意的是，本实施例包括在步骤 S95 中的处理，在该步骤中，作为当构建虚拟包时要使用的标准判断，判断所指定目录的签名是否是可靠的；然而，也可接受的是，不对签名的可靠性进行判断而构建虚拟包。

[0385] 应该注意的是，在本实施例中，在构建虚拟包之后，将在本地存储器中的所指定目录的属性和其下内容的属性设定为只读属性；然而，采用以下安排也是可接受的：其中，位于所指定目录下的目录和文件的属性也可以递归的设定为只读属性。

[0386] 应该注意的是，在本实施例中，当虚拟包被构建时，将在本地存储器中的路径，例如“ROOT/organization#1/disc#1/contsID#1”，指定为更新请求方法的自变量；然而，采用以下安排也是可接受的：其中仅仅指定 contsID，以便从 BD-ROM 中获得组织 ID 和光盘 ID。

[0387] 第四实施例

[0388] 在第三实施例中，描述了以下配置：其中，通过将被指定为更新请求方法的自变量的目录的属性设定为只读属性来防止构建虚拟包的、本地存储器中记录的文件的改变，以便避免虚拟包信息与在本地存储器中的实际目录 - 文件结果之间不一致的情况。

[0389] 在第四实施例中，讨论了一种改进，采用所述改进，可以避免虚拟包信息与在本地存储器中的实际目录 - 文件结果之间不一致的同时，对在构成虚拟包的本地存储器中的文件进行写入访问。

[0390] 图 39 是示出根据第四实施例用于构建虚拟包的处理过程的流程图。在该流程图与图 36 的流程图之间的差异在于，分别将步骤 S98 和 S99 改变为步骤 S111 和 S112。在步骤 S111 中，将在本地存储器中并被指定为更新请求方法的自变量的目录和其下内容复制到另一记录介质中。在步骤 S112 中，使用已经复制到所述另一记录介质与 BD-ROM 中的目录和其下内容来构建虚拟包。因此完成对构建虚拟包的处理过程的描述。

[0391] 以下描述 Java 应用程序进行的写入文件访问的处理过程。图 40 是流程图，示出了在虚拟包已经被构建之后对本地存储器中的文件进行写入访问的情况下，提供文件 I/O 功能的 API 的处理。该流程图与图 37 所示的流程图之间的差异在于，删除了步骤 S101 和 S102。当采用对本地存储器定位器的指定来请求对在本地存储器中的文件进行写入访问时，在步骤 S103 对本地存储器中的文件进行写入访问，然后处理完成。

[0392] 因此完成对应用程序进行的对文件的写入访问的处理过程的描述。

[0393] 如迄今所述，根据本实施例，当通过对记录在多个记录介质上的文件进行合并来构建虚拟包时，将在作为可读 / 可写记录介质的本地存储器中的文件复制到另一记录介质中，即使是当构建虚拟包时在本地存储器中的文件被重写，虚拟包也可以查询已经被复制到该记录介质上并反映在重写之前的状态的文件。因此，即使是该应用程序本身或者其他应用程序执行了非法的文件改变，也可以防止可能发生的故障。

[0394] 应该注意的是，在本实施例中，在本地存储器中的文件复制目的地是另一存储介质；然而，以下安排也是可以接受的：其中，将文件复制到在本地存储器中的不能从 Java 应

用程序直接引用的目录中，或者仅仅能够进行只读访问的目录中。

[0395] 应该注意的是，在本实施例中，将在本地存储器中的指定目录和其下内容复制到另一记录介质上；然而，存储有 AV 流等等的文件的大小是很大的，复制需要时间，并且占用所述记录介质的存储空间。为了解决该情况，还可以接受的是，如图 38 的示例所示，对这种具有很大大小的文件使用符号链接。

[0396] 第五实施例

[0397] 第五实施例描述了一种改进，其用于在将要构建虚拟包时防止文件被非法改变。

[0398] 图 41 示出了根据第五实施例的重放装置的内部结构。根据本实施例的重放装置具有以下配置：其中将加密器 41 和解密器 42 添加到图 18 所示的配置中。与图 18 中的相同的构成组件具有相同的参考符号，在此将省略对其的描述。

[0399] 加密器 41 采用记录在插入到 BD-ROM 驱动器 1 中的 BD-ROM 的 BCA(Burst CuttingArea 烧录区）中的对于该光盘而言唯一的密钥信息，选择性地对要写入到本地存储器中的文件进行加密。加密器 41 要加密的文件，是在用于构建虚拟包并已经从服务器下载的文件中的用于 AV 数据的 M2TS 文件。在此所使用的加密方法为，例如，诸如 AES、DES 等等的传统加密方法。BCA 是光盘中的导入区内侧的特定区域，所述光盘仅能够由驱动装置读出。由于没有应用程序能够从 BCA 区域中读出数据，因此 BCA 区域经常用于版权保护技术。

[0400] 解密器 42 采用记录在插入到 BD-ROM 驱动器 1 中的 BD-ROM 的 BCA 中的对于该光盘而言唯一的密钥信息，对文件进行解密。解密器 42 所要解密的数据是从本地存储器中读出的 M2TS 文件。解密器 42 获得 AV 数据的访问单位 (ACCESS UNIT) 并对其进行解密，将被解密数据输出到重放机构。在此所使用的加密方法与加密器 41 所使用的加密方法相同，并用于对由加密器 41 加密的文件进行解密。

[0401] 另外，根据本实施例的重放装置，在虚拟文件系统单元 38 中所保持的虚拟包管理表的结构与第三实施例中不同。图 42 示出了第五实施例的虚拟包管理表的示例。本实施例的虚拟包管理表与图 35 所示不同之处在于，其具有为其添加的签名数据。在虚拟包管理表中的签名数据是用于验证在用于构建虚拟包的目录中，在文件被记录到本地存储器中之后没有被重写。

[0402] 在签名数据 607 中的多个注册信息中，分别与具有小文件大小的 BDMV 文件、MPLS 文件和 CLPI 文件相对应的这些条注册信息 601、602 和 603 包括在虚拟包中使用的文件名和记录在本地存储器中的文件实体的散列值。当文件从服务器下载时，已经计算了在此设定的散列值。应该注意的是，散列值是采用用于获得从在文件中存储的数据中唯一确定的值的单向函数来计算。单向函数的一个示例是 MD5，其是传统方法。另一方面，在签名数据中的多条注册信息中，用于具有很大文件大小的 M2TS 文件的注册信息 606 仅仅包括在虚拟包中使用的文件名，而不包括散列值。签名数据 607 是采用记录在 BD-ROM 的 BCA 中并对于光盘而言唯一的密钥信息进行加密的。

[0403] 以下参考图 43 和 44 描述第五实施例的重放装置的操作的流程。首先，描述将从服务器下载的文件记录到本地存储器中的处理。图 43 是示出用于将所下载文件记录到本地存储器中的处理过程的流程图。

[0404] 当 Java 应用程序从服务器下载要在构建虚拟包时使用的文件并将所下载文件写入到本地存储器中时，执行在本流程图中所示的处理。

[0405] 当 Java 应用程序采用本地存储器定位器的指定来请求写入访问时，判断是否由于现在正在构建虚拟包而将目录的属性设定为只读属性（步骤 S121）。在包含所指定文件的目录的属性被锁定到只读属性的情况下（步骤 S121 ：是），将指示不允许写入该文件的异常返回给 Java 应用程序，并终止处理（步骤 S122）。在其他情况下（步骤 S121 ：否），过程进入步骤 S123。

[0406] 在步骤 S123，判断所指定的文件是否是用于 AV 数据的 M2TS 文件。

[0407] 在所指定文件是 M2TS 文件的情况下（步骤 S123 ：是），由加密器 41 采用记录在插入到 BD-ROM 驱动器 1 中的 BD-ROM 的 BCA 中的对于该光盘而言唯一的密钥信息对该文件进行加密（步骤 S124）。然后，将仅包含要在虚拟包中使用的指定文件的文件名的一个注册信息添加到签名数据中（步骤 S125），并且执行在步骤 S127 中和该步骤之后的处理。

[0408] 当所指定文件不是 M2TS 文件的情况下（步骤 S123 ：否），计算该指定文件的散列值。然后，将包含所计算的散列值和要在虚拟包中使用的指定文件的文件名的一条注册信息添加到签名数据中（步骤 S126），并且执行在步骤 S127 中和该步骤之后的处理。

[0409] 在步骤 S127，由加密器 41 采用记录在 BD-ROM 的 BCA 中的对于该光盘而言唯一的密钥信息对签名数据进行加密。在步骤 S128，对本地存储器进行写入访问，将在步骤 S127 中加密的签名数据连同所指定文件记录到本地存储器中。因此完成对将从服务器下载的文件记录到本地存储器中的处理的描述。

[0410] 以下描述要由虚拟文件系统单元 38 执行的构建虚拟包的处理。图 44 是示出构建虚拟包的处理过程的流程图。该流程图示出在 Java 应用程序使用光盘 ID 和其中存储用于构建虚拟包的数据的目录的路径作为自变量，发布更新请求方法的情况下要执行的处理过程。

[0411] 在步骤 S131 中，判断是否已经为由所指定光盘 ID 所标识的 BD-ROM 构建了虚拟包。

[0412] 在已经为由所指定光盘 ID 所标识的 BD-ROM 创建了虚拟包的情况下（步骤 S131 ：是），过程进入步骤 S132。在其他情况下，过程进入步骤 S135。在步骤 S132，将所指定光盘 ID 的条目从虚拟包管理表中删除。在步骤 S133 中，判断由所指定光盘 ID 标识的 BD-ROM 是否已经插入到 BD-ROM 驱动器 1 中。在已经插入由所指定光盘 ID 标识的 BD-ROM 的情况下（步骤 S133 ：是），过程进入步骤 S134。在其他情况下（步骤 S93 ：否），过程进入步骤 S135。在步骤 S134，将与所指定光盘 ID 相对应的虚拟包的构建暂停。

[0413] 在步骤 S135，将指定为自变量的目录路径和记录在所指定目录中的签名数据，添加到指定为自变量的光盘 ID 的条目中。

[0414] 在步骤 S136，判断是否将由所指定光盘 ID 标识的 BD-ROM 插入 BD-ROM 驱动器 1 中。在由所指定光盘 ID 标识的 BD-ROM 已经插入的情况下（步骤 S136 ：是），过程进入步骤 S137。在其他情况下，处理终止。

[0415] 在步骤 S137，解密器 42 采用记录在 BD-ROM 的 BCA 中的对于该光盘而言唯一的密钥信息，对虚拟包管理表的签名数据进行解密。

[0416] 在步骤 S138，判断被解密的签名数据是否是可靠的。当以下 3 个条件都满足时，判定签名数据可靠：

[0417] (a) 对具有已经设定的散列值的签名数据的所有文件中的每一个计算散列值，所

计算的值等于在签名数据中所包括的散列值；

[0418] (b) 在本地存储器中存在其的签名数据不具有已经设定的散列值的 M2TS 文件，并且其未被删除；

[0419] (c) 在本地存储器中的所指定目录中不存在没有在签名数据中记载的文件。

[0420] 在签名数据判定为是可靠的情况下（步骤 S138：是），过程进入步骤 S139。在其他情况下，处理终止。在步骤S139，将在本地存储器中的所指定目录的属性和其下内容的属性设定为只读属性。在步骤S140，使用自变量指定的本地存储器中的目录和 BD-ROM 来构建虚拟包。因此完成对构建虚拟包的处理的描述。

[0421] 此外，在将光盘新插入重放装置的情况下，从虚拟包管理表中搜索与所插入 BD-ROM 的光盘 ID 相对应的目录路径，并使用所检测的目录路径作为指定目录路径执行步骤 S136 中和该步骤之后的处理，以便构建虚拟包。

[0422] 应该注意的是，在本实施例中，仅仅对包含在虚拟包中的 M2TS 文件进行加密 / 解密；而然，以下安排也是可以接受的：对其他文件进行加密 / 解密，例如 CLPI 文件、BDMV 文件和 MPLS 文件。此外，虽然对在经过加密之后将所下载的 M2TS 文件写入到本地存储器中的情况进行了描述；然而，以下安排也是可接受的：其中，下载已经采用对该光盘而言唯一的密钥预先加密的 M2TS 文件。

[0423] 应该注意的是，所述描述说明 M2TS 文件的散列值没有添加到签名数据中，以便当要执行重放时忽略计算散列值的处理，这是因为 M2TS 文件的大小是很大的；然而，在采用具有足够高的处理能力的重放装置的情况下，将 M2TS 文件的散列值添加到签名中也是可以接受的。另外，以下安排也是可以接受的：其中，对其他文件进行加密、解密，例如 CLPI 文件、BDMV 文件和 MPLS 文件，从而使得这些文件的散列值计算处理也如同 M2TS 文件一样被忽略。

[0424] 应该注意的是，在本实施例中，当将文件写入到本地存储器中生成签名数据；然而，也可接受的是，当构建虚拟包时下载签名数据本身，并且指定所下载的签名数据。

[0425] 在本实施例中，将对于该光盘而言唯一的密钥信息用作加密 / 解密的密钥；然而，也可接受的是，使用对于该光盘而言唯一的密钥信息和从对于终端装置而言唯一的密钥信息中生成的信息，作为加密 / 解密的密钥。

[0426] 如迄今所述，根据本实施例，当通过将记录在多个记录介质上的文件进行合并来构建虚拟包时，采用对于该光盘而言唯一的密钥来对在作为可写 / 可读记录介质的本地存储器中记录的文件进行加密。可以防止，例如，在本地存储器中的实体文件被另一光盘上的应用程序所重写，或者防止通过将该本地存储器插入到诸如个人计算机的另一装置中进行的窜改。此外，由于采用对于该光盘而言唯一的密钥对签名数据进行加密并将其存储，因此可以检测对本地存储器中的文件的窜改，并可以防止采用这些文件构建虚拟包。因此，可以防止由于已经被非法窜改的文件的使用导致的故障，和防止提供商不期望的非法使用。

[0427] 第六实施例

[0428] 第六实施例讨论了当正在构建虚拟包时，将文件从本地存储器移动到另一记录区域中的方法。

[0429] 图 45 示出了根据第六实施例，由虚拟文件系统单元 38 所执行的虚拟包构建的示例。在该图的左上方，是在 BD-ROM 中的目录 - 文件结构。在该图的左下方是在构建虚拟包

之前的本地存储器中的目录 - 文件结构。

[0430] 在本实施例中的虚拟文件系统单元 38 将由更新请求方法的自变量所指定的目录（在图中的示例中为“ROOT/organization#1/disc#1/contsID#1”）下的文件移动到 ACTIVE 目录下，并从 BD-ROM 和 ACTIVE 目录中构建虚拟包。在该图的右下方是在移动所述文件之后在本地存储器中的目录 - 文件结构。

[0431] ACTIVE 目录是与由更新请求方法的自变量所指定的光盘 ID 标识的 BD-ROM 一起构建虚拟包的目录。应用程序不能直接将数据写入到 ACTIVE 目录中。

[0432] 以下描述根据第六实施例的重放装置的操作。首先，描述构建虚拟包的处理过程。

[0433] 图 46 是示出根据第六实施例，构建虚拟包的处理过程的流程图。在该流程图与图 36 所示流程图之间的差异在于，分别将步骤 S98 和 S99 改变为步骤 S151 和 S152。在步骤 S151，执行用于将在本地存储器中由自变量指定的目录下的文件移动到 ACTIVE 目录中的处理。在步骤 S152，从 ACTIVE 目录中和 BD-ROM 中的内容构建虚拟包。因此完成对构建虚拟包的处理过程的描述。

[0434] 以下描述在步骤 S151 中的处理的细节。图 47 是示出将文件移动到 ACTIVE 目录中的处理过程的流程图。

[0435] 在该流程图中，判断在更新请求方法的自变量所指定的本地存储器中的目录下是否存在一个或多个文件（步骤 S161），如果存在一个或多个文件（步骤 S161 :是），则使用在所指定目录中的所述一个或多个文件作为处理目标，执行步骤 S162 到 S164 的处理。

[0436] 在步骤 S162，判断在目标文件的文件名中的第一个字符是否为“_”。在第一个字符为“_”的情况下（步骤 S162 :是），将具有通过从该目标文件的文件名中提取出第一个字符“_”而获得的文件名的文件，从 ACTIVE 目录中删除，并且将该目标文件本身也从所指定目录中删除（步骤 S163）。在第一个字符不是“_”的情况下（步骤 S162 :否），将该目标文件从所指定目录移动到 ACTIVE 目录中（步骤 S164）。

[0437] 对在所指定目录下存在的所有文件执行步骤 S162 到步骤 S164 的处理，并且当在所指定目录下不存在文件时（步骤 S161 :否），终止处理。从而完成对将在本地存储器中的指定目录下的文件移动到 ACTIVE 目录中的处理的细节描述。

[0438] 应该注意的是，替代根据目标文件的文件名是否为“_”来从 ACTIVE 目录中删除文件，也可以接受的是，提供进行删除预定 (deletion reservation) 的 API。进行删除预定的 API 用于使得 Java 应用程序能够对在 ACTIVE 目录中的文件进行删除预定；然而，即使是该 API 进行了删除预定，也不会立即删除在 ACTIVE 目录中的文件。虚拟文件系统单元 38 将文件删除预定的内容存储到存储器中，并且当执行在 AV 目录中的文件的移动处理时，通过删除由所述删除预定所指定的 ACTIVE 目录中的文件，来执行步骤 S162 和 S163 中的替换处理。

[0439] 以下描述由 Java 应用程序进行的写入文件访问的处理过程。

[0440] 图 48 是示出控制对文件的写入访问的 API 的处理过程的流程图。当 Java 应用程序使用本地存储器定位器的指定来请求写入访问时，判断该写入访问是否是对在 ACTIVE 目录下的文件之一进行的写入访问（步骤 S171）。

[0441] 当该写入访问是对在 ACTIVE 目录下的文件之一进行的写入访问时（步骤 S171 :是），将指示不允许对文件进行写入的异常返回给 Java 应用程序，然后处理终止（步骤

S172)。在其他情况中（步骤 S171 ：否），过程进入步骤 S173。

[0442] 在步骤 S173，对本地存储器中的文件进行写入访问。因此完成对文件的写入访问的处理过程的描述。

[0443] 根据上述处理过程，在作为访问目标的文件是构成虚拟包的文件，并且如果当作出访问请求时正在构建虚拟包的情况下，将不会允许对该文件的访问。

[0444] 应该注意的是，以下安排也是可以接受的：其中，在虚拟包构建处理过程中允许 Java 应用程序将数据写入到 ACTIVE 目录中，以便 Java 应用程序执行将文件移动到 ACTIVE 目录中的处理。更具体而言，在图 47 的流程图中，虚拟文件系统单元 38 将在由 Java 应用程序指定的目录下的文件移动到 ACTIVE 目录中；然而，以下安排也是可接受的：其中，在虚拟包构建处理过程中允许 Java 应用程序写入 ACTIVE 目录，以便将开始把文件移动到 ACTIVE 目录的处理的事件通知给 Java 应用程序，并且 Java 应用程序自己将文件移动到 ACTIVE 目录。

[0445] 如迄今所述，根据本实施例，在通过将记录在多个记录介质中的文件进行合并来构建虚拟包的情况下，将在作为可读 / 可写记录介质的本地存储器中存储的一个或多个文件移动到 ACTIVE 目录中，应用程序不允许对 ACTIVE 目录进行写入访问；因此，可以防止当正在构建虚拟包时，采用来自该应用程序本身或者其他应用程序的非法文件来改变文件。

[0446] 第七实施例

[0447] 第七实施例描述了在来自相同提供商的 BD-ROM 之间共享本地存储器中的内容的方法。在第七实施例的描述中，与在第六实施例中的相同部分的描述将被省略。

[0448] 图 49 示出了根据第七实施例的虚拟包的构建的示例。在该图的左上方是在 BD-ROM 上的目录 - 文件结构，在该图的左下方是通过将构建虚拟包的文件移动到 ACTIVE 目录中以便构建虚拟包，而获得的在本地存储器中的目录 - 文件结构。

[0449] 图 49 与图 48 的不同之处在于，ACTIVE 目录不仅存在于 ROOT/organization#1/disc#1 下，还存在于 ROOT/organization#1/SHARED 下。

[0450] SHARED 目录是能够由任何被验证的并且具有公共组织 ID 的应用程序从中读出数据和将数据写入的目录。

[0451] 在需要将从服务器下载的文件作为公共使用的文件，来用于为具有不同光盘 ID 的多个 BD-ROM 构建虚拟包的情况下，将所下载文件保存在 ROOT/organization#1/SHARED 目录下而不是 ROOT/organization#1/disc#1 目录下。在 SHARED 目录下的目录结构与 disc ID 目录下的目录结构相同。对 ACTIVE 目录的操作也与第六实施例中所述相同。然而，应该注意的是，当要构建虚拟包时，不仅仅将在 disc#1 目录下的 ACTIVE 目录中的文件，而且将在 SHARED 目录下的 ACTIVE 目录下的文件合并来构建虚拟包。

[0452] 本实施例的虚拟文件系统单元 38 将同时存在于本地存储器的 SHARED 目录下和本地存储器的 disc ID 目录下的一个或多个文件，作为在虚拟包中使用的文件记载到虚拟包信息中。另外，在具有相同名称的文件存在于以下的任意组合的情况下：(i)BD-ROM 上，(ii) 本地存储器的 SHARED 目录下，以及 (iii) 本地存储器的 disc ID 目录下，（例如，0002.CLPI 同时存在于 BD-ROM 上和本地存储器的 disc#1 目录下），虚拟文件系统单元 38 使用本地存储器中的文件作为用于虚拟包构建的文件。在具有相同名称的文件存在于以下全部中的情况下：(i)BD-ROM 上，(ii) 本地存储器的 SHARED 目录下，以及 (iii) 本地存储器的 disc

ID 目录下，虚拟文件系统单元 38 使用在 discID 目录下的文件进行虚拟包构建。递归地对每个子目录重复进行这种合并处理。

[0453] 图 50 示意性地示出了如何使用在 SHARED 目录下的 ACTIVE 目录和在 disc#1 目录下的 ACTIVE 目录来构建虚拟包。当虚拟包被构建时，将在 SHARED 目录下的 ACTIVE 目录中的文件与在 BD-ROM 上的文件进行合并，从而生成中间文件结构。该合并操作与在第四实施例中所述相同。随后，将该中间文件结构与在 disc#1 目录下的 ACTIVE 目录中的文件进行合并。从而通过将 BD-ROM、在 SHARED 目录下的 ACTIVE 目录中的文件、以及在 disc#1 目录下的 ACTIVE 目录中的文件进行合并来构建虚拟包。该中间文件结构与在 disc#1 目录下的 ACTIVE 目录中的文件进行合并，是通过在将该中间文件结构看作 BD-ROM 的同时执行第四实施例中的合并操作来实现的。换而言之，在具有相同名称的文件同时存在于中间文件结构和 disc ID 下的 ACTIVE 目录中的情况下，disc#1 目录下的 ACTIVE 目录具有最终优先级别。因此，优先级别按以下顺序升高 :BD-ROM、在 SHARED 目录下的 ACTIVE 目录、以及在 disc#1 目录下的 ACTIVE 目录。

[0454] 可替换地，也可接受的是，将 SHARED 目录下的 ACTIVE 目录的优先级别降低，来构建虚拟包 ;即，优先级别按以下顺序升高 :在 SHARED 目录下的 ACTIVE 目录、BD-ROM、以及在 disc#1 目录下的 ACTIVE 目录。这意味着在具有相同的名称的文件同时存在于 BD-ROM 上和在本地存储器的 SHARED 目录下的情况下，使用在 BD-ROM 上的文件。参考图 49 进行解释，尽管 00002.JAR 和 00002.CLPI 同时存在于 BD-ROM 上和本地存储器中，对于 00002.JAR，由于其在本地存储器中的 SHARED 目录下的 ACTIVE 目录中，因此使用在 BD-ROM 上的 00002.JAR ;对于 00002.CLPI，因为其在本地存储器中的 disc#1 目录下的 ACTIVE 目录中，因此使用在本地存储器中的 00002.CLPI。在对 SHARED 目录下的 ACTIVE 目录的访问限制微弱，并且存在第三方企图通过将恶意文件放置在 SHARED 目录下来非法地替换在 BD-ROM 上的文件的情况下，采用这种方式设定优先级别是有效的。

[0455] 通过将在 SHARED 目录下的 ACTIVE 目录的优先级别降低到 BD-ROM 之下，可以禁止 BD-ROM 上的文件被替换，从而仅仅允许添加新文件。因此，可以增强安全级别。

[0456] 应该注意的是，也可以接受的是，在 BD-ROM 上或者在 SHARED 目录下的 ACTIVE 目录中，提供定义有关在 SHARED 目录下的 ACTIVE 目录的优先级别是高于还是低于 BD-ROM 的优先级别的优先级别信息的文件。

[0457] 上述虚拟包的构建是通过发布更新请求方法来实现的，更新请求方法指定以下作为自变量 :(i) 其中记录有要在多个 BD-ROM 的虚拟包之间进行共享的文件的目录路径（在图 49 的示例中为“ROOT/organization#1/SHARED/contsID#3”），(ii) 其中记录有对于由光盘 ID 所标识的 BD-ROM 的虚拟包而言唯一的文件的目录路径（在图 49 中的示例中为“ROOT/organization#1/disc#1/contsID#1”），以及 (iii) 光盘 ID。

[0458] 以下描述根据本发明的第七实施例的重放装置的操作。首先，解释构建虚拟包的处理过程。

[0459] 图 51 是示出根据第七实施例的构建虚拟包的处理过程的流程图。图 51 的流程图与图 36 的流程图之间的差异在于，将步骤 S98 和 S99 改变为步骤 S181 到 S184。

[0460] 在步骤 S181，执行以下处理，将在由自变量指定的、作为要在多个 BD-ROM 之间进行共享的文件的文件记录目的地的本地存储器中的目录下所存储的文件，移动到在 SHARED

目录下的 ACTIVE 目录中。采用与图 47 所示的过程相同的方式执行移动文件的处理。在步骤 S182，使用在 SHARED 目录下的 ACTIVE 目录中和 BD-ROM 中的内容构建中间文件结构。在步骤 S183，执行以下处理，将由自变量指定的、作为唯一性文件的文件记录目的地的目录下所存储的文件，移动到在 discID 目录下的 ACTIVE 目录中。采用与图 47 所示的过程相同的方式执行移动文件的处理。在步骤 S184，将在 discID 目录下的 ACTIVE 目录中所存储的内容与在步骤 S182 中构建的中间文件结构进行合并，从而构建虚拟包。因此完成对虚拟包创建的处理过程的描述。

[0461] 应该注意的是，以下的顺序可以颠倒：(i) 在步骤 S182 中的使用在 SHARED 目录下的 ACTIVE 目录和 BD-ROM 构建中间文件结构的步骤，以及 (ii) 在步骤 S183 中的将文件移动到 discID 目录下的 ACTIVE 目录中的步骤。

[0462] 接下来，以下参考图 52 详细描述图 51 所示的步骤 S182 中的处理。当将在 SHARED 目录下的 ACTIVE 目录中的文件与在 BD-ROM 上的文件进行合并时，则首先在步骤 S191，将或者存储在 BD-ROM 上或者存储在 discID 目录中，并且在其中写入优先级别信息的文件读出，以便检查 SHARED 目录和 BD-ROM 的优先级别。在 SHARED 目录的优先级别较高的情况下（步骤 S191 ：是），过程进入步骤 S192，获得示出在 SHARED 目录下的 ACTIVE 目录中的文件的文件列表。随后，对在 BD-ROM 上记录的所有文件中的每一个重复执行步骤 S193 到 S195 的循环处理。

[0463] 在步骤 S194，将记录在 BD-ROM 上的、还没有被处理的文件中的一些的文件名作为处理目标进行检查，判断在步骤 S192 中获得的文件列表是否包含与每个处理目标具有相同文件名的文件（步骤 S193）。在文件列表不包含与处理目标具有相同文件名的文件的情况下（步骤 S193 ：否），将该处理目标文件的文件名添加到文件列表（步骤 S 195）。对在 BD-ROM 上记录的所有文件中的每一个都进行在步骤 S194 和 S195 中的处理，并且当检查完所有文件时（步骤 S193 ：是），将所获得的文件列表作为中间文件结构，完成处理。

[0464] 另一方面，在步骤 S191 中 BD-ROM 的优先级别比 SHARED 目录高的情况下，过程进入步骤 S196，获得记录在 BD-ROM 上的文件的列表。随后，对在 SHARED 目录下的 ACTIVE 目录中的所有文件的每一个都重复执行在步骤 S197 到 S199 中的循环处理。

[0465] 在步骤 S198 中，在 SHARED 目录下的 ACTIVE 目录中的文件中，将还没有被处理的文件中的一些的文件名作为处理目标进行检查，判断在步骤 S196 中获得的文件列表是否包含与每个处理目标具有相同文件名的文件。在文件列表不包含与处理目标具有相同文件名的文件的情况下（步骤 S198 ：否），将该处理目标文件的文件名添加到文件列表（步骤 S199）。对在 SHARED 目录下的 ACTIVE 目录中所记录的所有文件中的每一个都进行在步骤 S198 和 S195 中的处理，并且当检查完所有文件时（步骤 S197 ：是），将所获得的文件列表作为中间文件结构，完成处理。因此完成用于构建中间文件结构的过程。

[0466] 应该注意的是，也可接受的是，预先确定缺省优先顺序，从而如果不存在指示优先级别信息的文件时，使用该缺省优先顺序。

[0467] 图 53 是更详细地示出图 51 的步骤 S184 中执行的处理的流程图。首先，在步骤 S201，获得在 discID 目录下的 ACTIVE 目录中的文件的文件名列表。在步骤 S203，检查在图 51 中的步骤 S182 中获得的中间文件结构的文件列表中所包含的文件名。在没有具有相同文件名的文件的情况下，将中间文件结构中的文件的文件名添加到步骤 S204 中的列表中。

对在中间文件结构中的所有文件中的每一个进行步骤 S203 和 S204 的处理，当在步骤 S202 判定检查完所有文件时，将所获得的文件列表作为虚拟包的文件结构列表，然后处理结束。因此完成对采用中间文件结构构建虚拟包的处理过程的描述。

[0468] 如迄今所述，根据本实施例，在通过将记录在多个记录介质上的文件进行合并来构建虚拟包的情况下，将记录在作为可读 / 可写记录介质的 HDD 上的文件移动到 ACTIVE 目录中；因此，就不会存在以下可能性：在正在构建虚拟包时，由于该应用程序本身或者其他应用程序进行的非法文件访问，导致在本地存储器中所存储的实体文件被改变。此外，通过在属于同一提供商的 Java 应用程序能够公共访问的位置处提供 ACTIVE 目录，就可以使得不同 BD-ROM 能够共享在本地存储器中所存储的内容。因此，本地存储器不需要存储具有很大文件大小的、相同实体的多个 AV 流等等。因此，就可以在 BD-ROM 之间有效地共享和利用数据。

[0469] 第八实施例

[0470] 第八实施例描述了在本地存储器中所存储的用于构建虚拟包的文件被窜改的情况，用于检测窜改的一种方法。在第八实施例中，与第三实施例相同部分的描述被省略。将仅仅描述与第三实施例不同的部分。

[0471] 当要用于构建虚拟包的本地存储器中的文件被窜改时，存在当重放虚拟包时执行提供商不期望的操作的可能性。因此，必须检测窜改。

[0472] 为了检测窜改，根据本实施例的重放装置在将要构建虚拟包时，检查合并管理信息文件是否被窜改，并且当读出文件时，检查在 HDD 上的文件是否被窜改。

[0473] 在构建虚拟包之后，在虚拟包的重放过程中，虚拟文件系统单元 38 有时可能提供存储在本地存储器中的一些文件用于 DVD-like 模块 29a、Java 平台 29b、重放控制引擎 32 等等。因为这些文件存储在本地存储器中，因此在所述文件被所述模块使用之前，必须检查其是否未被窜改。可以将构建虚拟包的本地存储器中的文件按照窜改检查的必要性分为两种类型。

[0474] 第一种类型是每个都具有小的文件大小的数据文件，其通常在响应重放装置时不会造成任何问题，即使是对整个文件检查窜改也是如此。例如，通常将“合并管理信息文件”分类到这种类型。

[0475] 第二种类型是每个都具有很大文件大小的数据文件，其在响应重放装置时可能造成问题，这是因为对整个文件进行检查需要很长时间。例如，通常将 MPEG 流中的“AV 数据 (M2TS 文件)”分类到这种类型。

[0476] 以下描述在被分类到第一种类型的文件中的窜改检查。在结尾，将描述与对分类到第二种类型的文件执行检查的情况之间的差异。

[0477] 图 54 是示出虚拟文件系统单元 38 将在本地存储器中的文件提供给另一模块的流程图。在步骤 S211，检查所请求文件是否存在于本地存储器中。在所请求文件在本地存储器中时（步骤 S211：是），在步骤 S212 中从本地存储器中获得该文件。随后，在步骤 S213 中，检查目标文件的散列值是否被写入到合并管理信息文件中。在散列值被写入到合并管理信息文件中的情况下（步骤 S213：是），在步骤 S214 计算该文件的散列值，并将所计算的值与在合并管理信息文件中所写入的值进行比较，以判断所述值是否彼此相等。

[0478] 在所计算的值与在合并管理信息文件中所写入的值不相等的情况下（步骤 S214：

否），在实际中将不执行该文件的加载。在步骤 S211 的判断结果是该文件不存在于本地存储器中时（步骤 S211 ：否），在步骤 S215 从 BD-ROM 获得该文件。随后，实际在步骤 S216 中加载该文件，从而使得其他模块能够使用该文件。类似地，在散列值没有写入合并管理信息文件（步骤 S213 ：否）和所计算的值等于在合并管理信息文件中所写入的值的情况下（步骤 S214），实际在步骤 S216 中加载该文件，从而使得其他模块能够使用该文件。

[0479] 应该注意的是，在虚拟文件系统单元 38 多次从本地存储中读出文件的情况下，可能在每次读出该文件时都执行窜改检查。还可以接受的是，在执行窜改之后，锁定该文件，从而使得在本地存储器中的信息不会被改变。

[0480] 此外，以下安排也是可以接受的 ：其中，虚拟文件系统单元 38 对要被其他模块所使用的一些或者全部文件预先执行窜改检查，并将被检查的文件存储在虚拟文件系统单元 38 或者锁定被检查文件。只要作出如下安排 ：在从本地存储器中读出文件之后，在其他模块使用该文件之前计算该文件的散列值以检查窜改，就可以接受以任何方式实现该安排。

[0481] 当检测到要被其他模块所使用的文件已经被窜改时，虚拟文件系统单元 38 暂停重放，从而使得将不会发生故障。在该情况下，可以接受的是，除了暂停重放之外，还通知用户或者通过 Java 平台 29b 通知 Java 应用程序，或者仅仅使用 BD-ROM 再次执行重放。可替换地，将这些处理中任意处理进行组合也是可以接受的。

[0482] 对于每一个都具有很大文件大小，并且在响应重放装置时由于对整个文件进行检查需要很长时间的文件而言，使用一种方法来加密整个文件，从而使得在该文件被窜改之后不能被解密。在该情况下，以下安排也是可接受的 ：其中，即使是将被加密文件的散列值写入到合并管理信息文件中，虚拟文件系统单元 38 也将其忽略，这是因为在被加密文件的一部分被窜改时，不可能将该文件解密为可靠文件。

[0483] 可替换地，还可接受的是，仅仅对该文件的一部分检查窜改，或者仅仅检查该文件的大小。在对整个文件进行加密的情况下，存在整个文件可能被另一文件所替换的可能性。在这种情况下，通过检查该文件的大小，可以容易地执行窜改检查。还可以接受的是设定一种情况，在该情况下，虚拟文件系统单元 38 忽略所写入的散列值。例如，在该文件的大小大于特定值的情况下，或者在合并管理信息文件中所包含的文件名中的扩展名是特定值，例如 .M2TS，的情况下，或者在合并管理信息文件中所包含的文件名是在特定目录下，例如 STREAM 目录，的情况下，忽略所写的散列值。

[0484] 如迄今所述，根据本实施例，当通过对记录在多个记录介质上的文件进行合并来构建虚拟包时，如果在作为可读 / 可写记录介质的本地存储器中的实体文件已经被窜改，则可以判断散列值是否正确并检测实体文件的窜改，以便暂停重放。因此，可以在可能由于使用非法文件造成的故障发生之前，将其避免。

[0485] 第九实施例

[0486] 第九实施例描述了用于在屏幕上显示虚拟包构建处理的状态的一种方法。在第九实施例的描述中，与第三实施例相同部分的描述被省略。将仅仅描述与第三实施例不同的部分。

[0487] 图 55 示意性地示出了根据第九实施例，在虚拟包构建处理过程中的显示。

[0488] 重放装置 610 示意性地表示重放装置的外部面貌。内置在重放装置 610 中的 BD-ROM 驱动器 611 加载和退出 BD-ROM 以及访问 BD-ROM。在重放装置 610 的前面提供了荧

光管显示设备 612，其用于示出正在执行虚拟包构建处理，并且用作彩色显示单元，并且其通过显示颜色（例如，用橙色显示）来指示正在执行虚拟包构建处理。在重放装置 610 的前面提供了荧光管显示设备 613，其用于通过改变所显示区域的比例来示出在与虚拟包构建相关的处理中，例如在验证处理中，的进度，并且其通过改变由来自荧光管的光所显示的区域的比例来指示虚拟包构建处理的进度。通过有线等等将诸如 TV 之类的显示装置 614 的输入单元（图中未示出）连接到重放装置 610 的输出单元（图中未示出），并在屏幕上显示从重放装置 610 输出的图像。重放装置 610 将指示正在执行虚拟包构建的屏幕显示 615 输出到显示装置 614，作为弹出消息面板等等，从而将其覆盖在内容重放显示屏幕上来进行显示。

[0489] 作为弹出显示的屏幕显示 615 包括屏幕显示 616，其通过改变所显示区域的比例，来显示与虚拟包构建有关的处理，例如验证处理，的进度，所述所显示区域通过改变在显示装置 614 的屏幕上的条形的比例，来视觉性地指示虚拟包构建处理的进度。

[0490] 图 56 示意性地示出根据第九实施例，在虚拟包构建处理中的一个失败的显示。与图 55 相同的部分的描述将省略。将仅仅描述与图 55 不同的部分。在重放装置 610 的前面提供了荧光管显示 617，其用于指示在虚拟包构建处理中的失败，并且通过显示颜色（例如，用红色显示）来指示虚拟包构建处理已经失败。在虚拟包构建处理已经失败的情况下，重放装置 610 将指示虚拟包构建处理失败的屏幕显示 618 输出到显示装置 614，作为弹出消息面板等等，从而将其覆盖在内容重放显示屏幕上来进行显示。

[0491] 图 57 示意性地示出了根据第九实施例，在虚拟包构建处理中的成功的显示。与图 55 相同的部分的描述将省略。将仅仅描述与图 55 不同的部分。在重放装置 610 的前面提供了荧光管显示 619，其用于指示在虚拟包构建处理中的成功，并且通过显示颜色（例如，用绿色显示）来指示虚拟包构建处理已经成功。在虚拟包构建处理已经成功的情况下，重放装置 610 将指示虚拟包构建处理成功的屏幕显示 620 输出到显示装置 614，作为弹出消息面板等等，从而将其覆盖在内容重放显示屏幕上来进行显示。

[0492] 以下描述当正在构建虚拟包时，在屏幕上显示构建处理的状态的过程。图 32 是示出对与虚拟包构建处理相关的显示进行更新的处理过程的流程图。

[0493] 该流程图是通过从图 44 所示的流程图中取出在步骤 S131 到 S135 中的处理过程，并加入在步骤 S222 和 S223 中的处理过程而获得的。

[0494] 在步骤 S221，将显示转换到图 55 所示的正在执行虚拟包处理的显示。在步骤 S222，将显示转换到图 56 所示的虚拟包处理已经失败的显示。在步骤 S223，将显示切换到图 27 所示的虚拟包处理已经成功的显示。还可以接受的是，在执行步骤 S222 中的处理或者在步骤 S223 中的处理之后，在指示虚拟包构建的状态的屏幕显示的输出完成之前，将该显示在屏幕上示出一段预定时间，并且执行正常的内容重放。

[0495] 图 59 示出了根据第九实施例，在本地存储器中所存储的内容的列表的显示。与图 55 相同的部分的描述将省略。将仅仅描述与图 55 不同的部分。在本地存储器中所存储的内容列表的屏幕显示 621，是用于在显示装置 614 上显示在虚拟包管理表中所管理的本地存储器中的内容列表的图像。示出了在该内容列表的虚拟包构建处理中已经失败的内容的屏幕显示 622 是用于通过在屏幕显示 621 中的图标，来向用户指示在签名验证中内容已经失败，所述屏幕显示 621 用于本地存储器中的内容列表的屏幕显示。屏幕显示 623 示出了

在该内容列表的虚拟包构建处理中已经成功的内容，所述屏幕显示 623 是用于通过在屏幕显示 621 中的图标，来向用户指示在签名验证中内容已经成功，所述屏幕显示 621 用于本地存储器中的内容列表的屏幕显示。

[0496] 应该注意的是，在本实施例中，将在虚拟包构建处理中已经失败的内容通知给用户；然而，以下安排也是可以接受的：其中，系统自动删除这种内容。

[0497] 其他修改示例

[0498] 迄今已经根据上述实施例示例描述了本发明。然而，不言而喻，本发明并不局限于上述实施例示例。本发明包括如下所述的其他示例。

[0499] (1) 本发明可以构建为上述的方法。可替换地，本发明可以构建为计算机程序，所述计算机程序使用从这种计算机程序转换的计算机或者数字信号来实现所述方法。

[0500] 另外，可接受的是，本发明构建为计算机可读记录介质，例如软盘、硬盘、CD-ROM、MO、DVD、DVD-ROM、DVD-RAM、BD（蓝光光盘）或者半导体存储器，所述计算机可读记录介质上记录有这种计算机程序和这种数字信号；或者本发明构建为所述记录介质上所记录的这种计算机程序或者这种数字信号。

[0501] 此外，可以通过经由通信网、无线或有线通信线、诸如因特网之类的网络等等传输这种计算机程序或者这种数字信号，来实现本发明。

[0502] 此外，可接受的是，认为本发明是包括微处理器和存储器的计算机系统，其中，所述存储器中存储上述计算机程序，并且所述微处理器根据所述计算机程序工作。

[0503] 此外，可接受的是，通过输送记录在所述记录介质上的程序或者数字信号，或者经由上述网络等等进行输送，来基于计算机系统执行所述程序或数字信号。

[0504] (2) 在第一到第九实施例中，将 Java（注册商标）用作虚拟机的编程语言。然而，采用用作 UNIX™ 的 OS 的 B-shell、Perl 脚本、ECMA 脚本等等来取代 Java™ 编程语言也是可以接受的。换而言之，本发明并不局限于 Java™。

[0505] (3) 在第一到第九实施例中，在用于存储文件的本地存储器中的结构具有与 BD-ROM 上相同的目录结构；然而，只要清晰地表示文件之间的对应关系，本发明就不局限于采用相同的目录结构的情况。

[0506] (4) 在第一到第九实施例中，将 HDD 用作可读 / 可写记录介质；然而，使用闪存等等作为可读 / 可写记录介质也是可接受的。

[0507] (5) 在第一到第九实施例中，描述了具有对记录介质进行重放功能的重放装置；然而，不言而喻，本发明可以应用于不仅仅具有重放功能还具有记录功能的重放装置。

[0508] (6) 在第一到第九实施例中，描述了重放 BD-ROM 的重放装置；然而，不言而喻，在以上实施例所述的记录在 BD-ROM 上的必要数据被记录在可写光学记录介质上的情况下，也能够实现相同的效果。

[0509] 此外，不言而喻，在以上实施例所述的记录在 BD-ROM 上的必要数据被记录在取代光学记录介质的便携式记录介质（例如，诸如 SD 卡、或者小型闪存 (CF) 之类的半导体存储器）上的情况下，也能够实现相同的效果。

[0510] (7) 可以接受的是，将任意所述实施例和修改示例进行组合。

[0511] 工业适用性

[0512] 能够应用本发明的示例包括，诸如 BD-ROM 播放器之类的重放装置，在所述重放装

置上安装了 HDD，并且所述重放装置通过采用记录在可读 / 可写记录介质上的内容对记录在只读记录介质上的内容进行扩展，来执行重放。

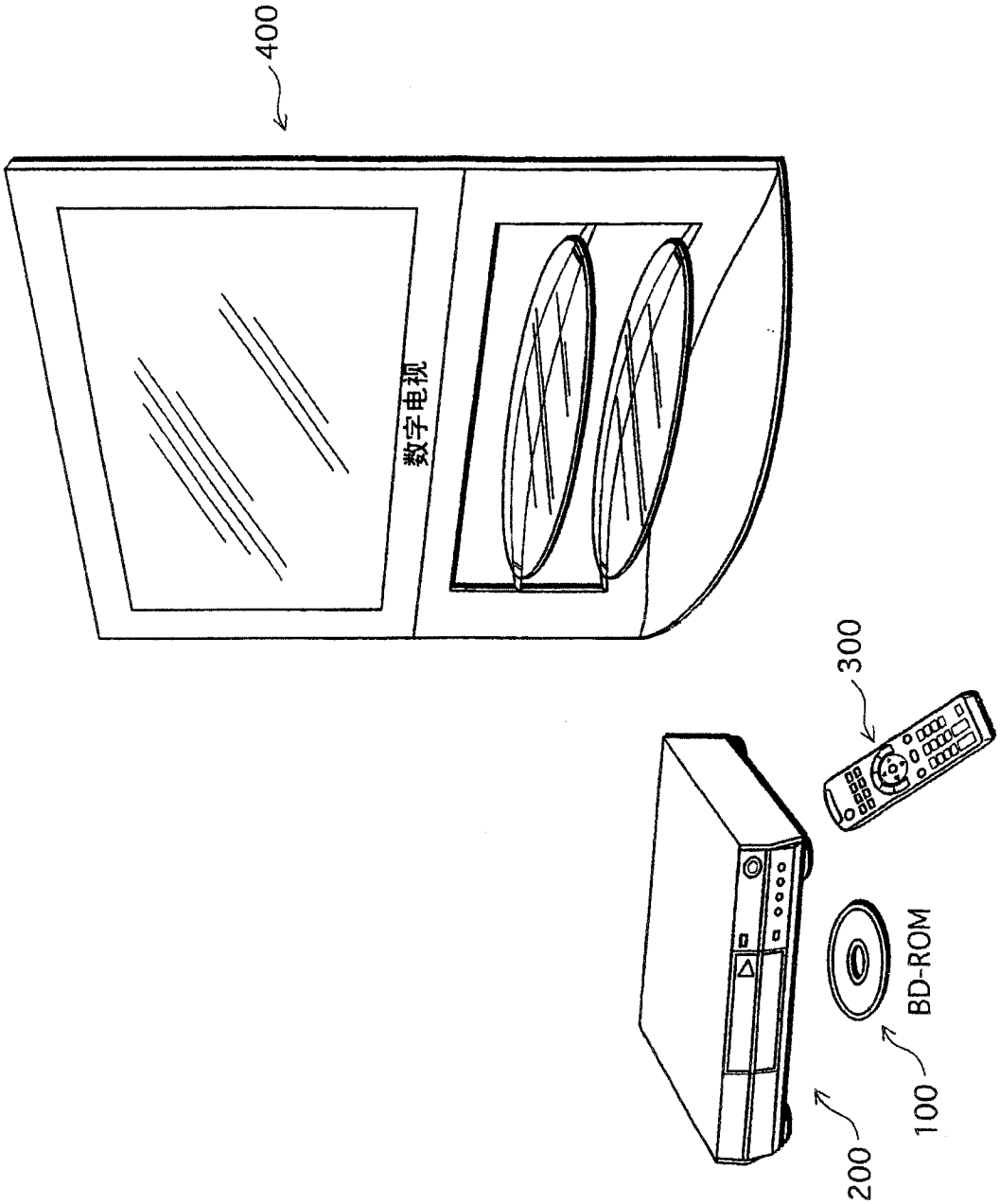

图 1

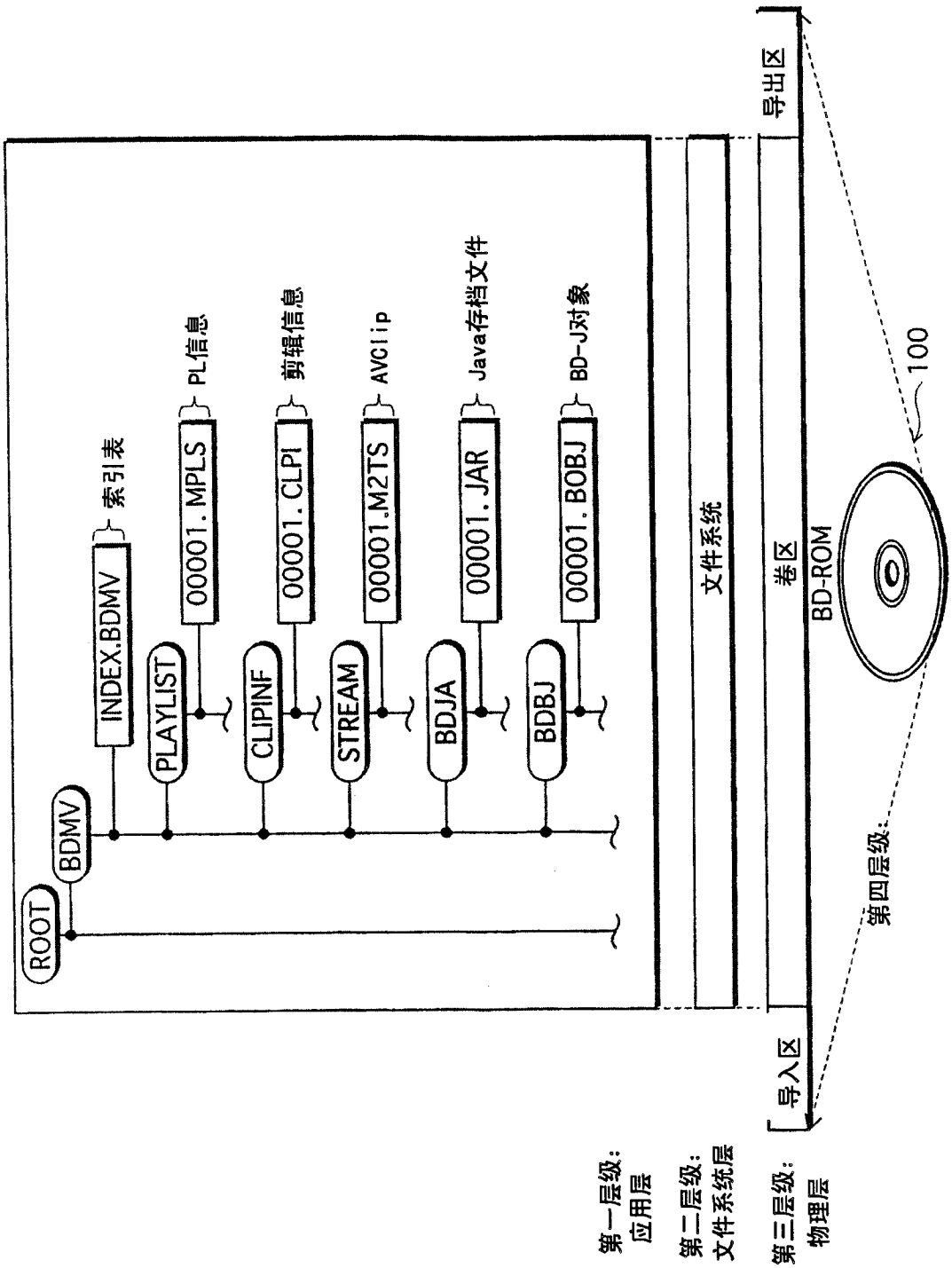
ROOT
BDMV
INDEX.BDMV
索引表
PLAYLIST
00001.MPLS
PL信息
CLIPINF
00001.CLPI
剪辑信息
STREAM
00001.M2TS
AVClip
BDJA
00001.JAR
Java存档文件
BDBJ
00001.BOBJ
BD-J对象
文件系统
导出区
卷区
BD-ROM
导入区
100
第四层级:
第一层级:
应用层
第二层级:
文件系统层
第三层级:
物理层

图 2

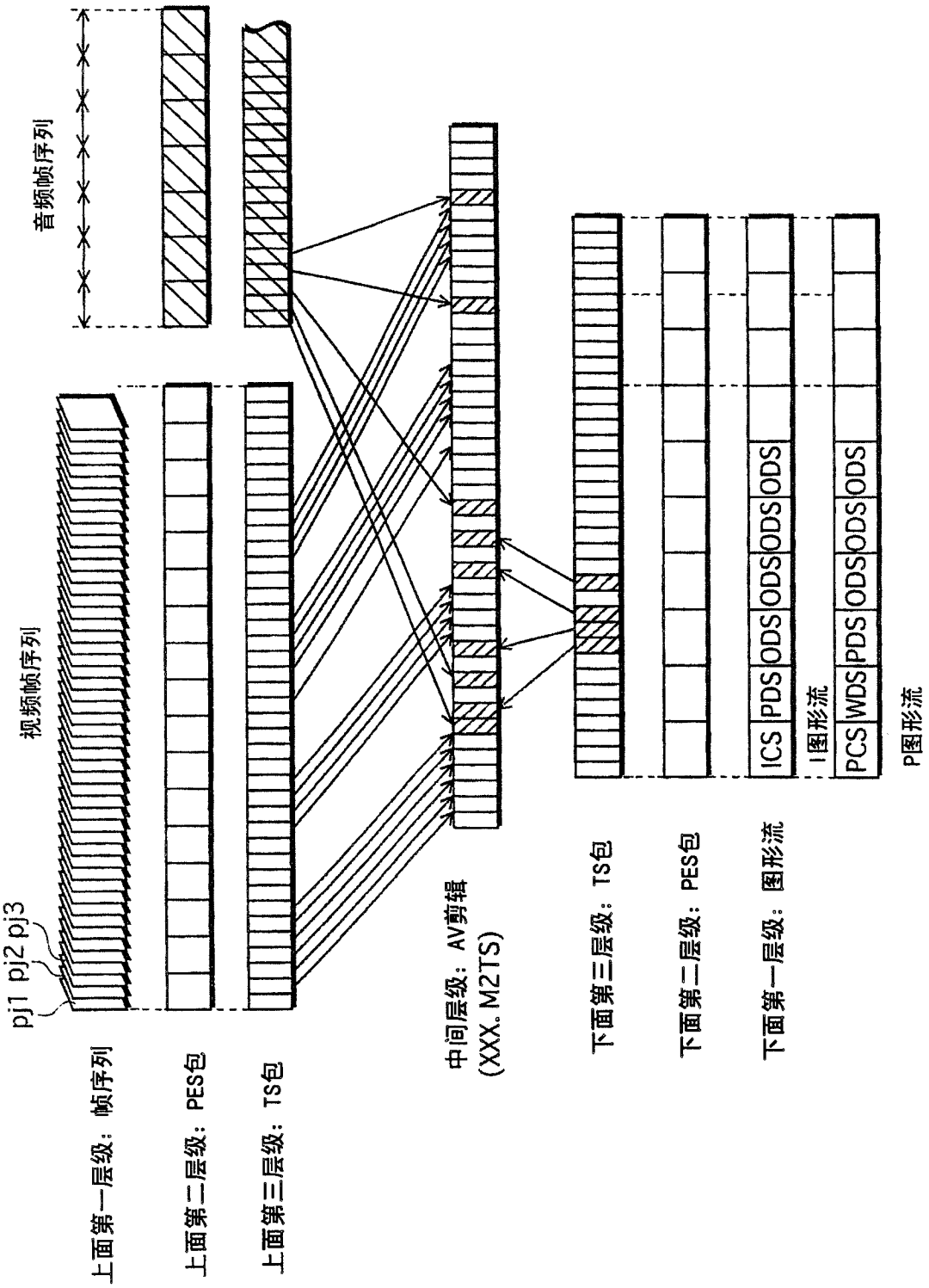
音频帧序列
视频帧序列
pj1 pj2 pj3
上面第一层级：帧序列
上面第二层级：PES包
上面第三层级：TS包
中间层级：AV剪辑
(XXX. M2TS)
下面第三层级：TS包
下面第二层级：PES包
下面第一层级：图形流
ICS
PDS
ODS
ODS
ODS
ODS
I图形流
PCS
WDS
PDS
ODS
ODS
ODS
P图形流

图 3

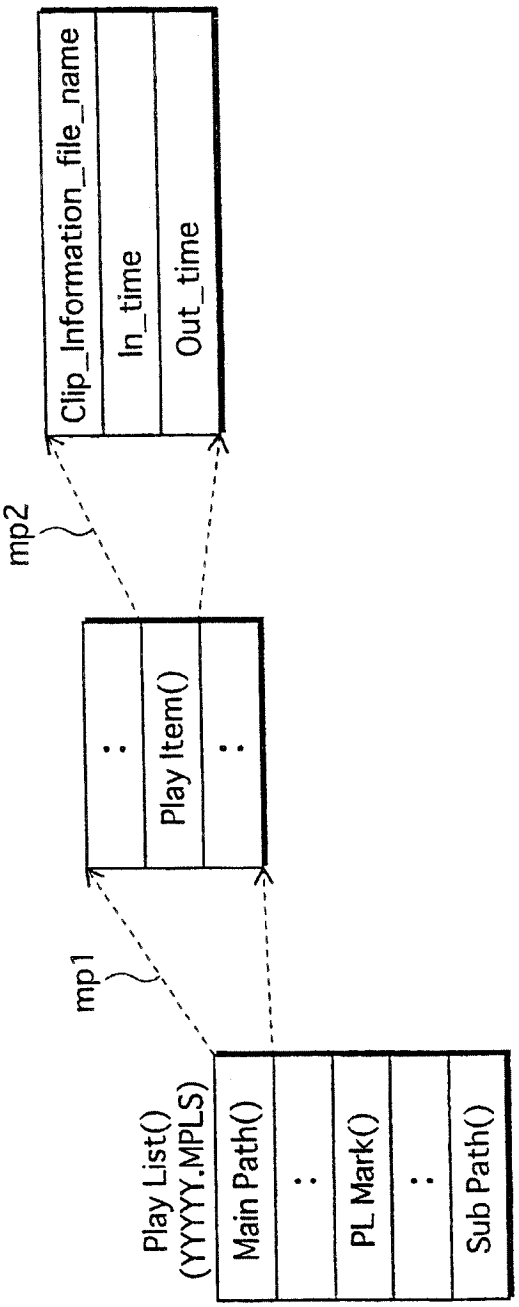
Clip_Information_file_name
In_time
Out_time
mp2
Play Item()
mp1
Play List()
(YYYYY.MPLS)
Main Path()
PL Mark()
Sub Path()

图 4

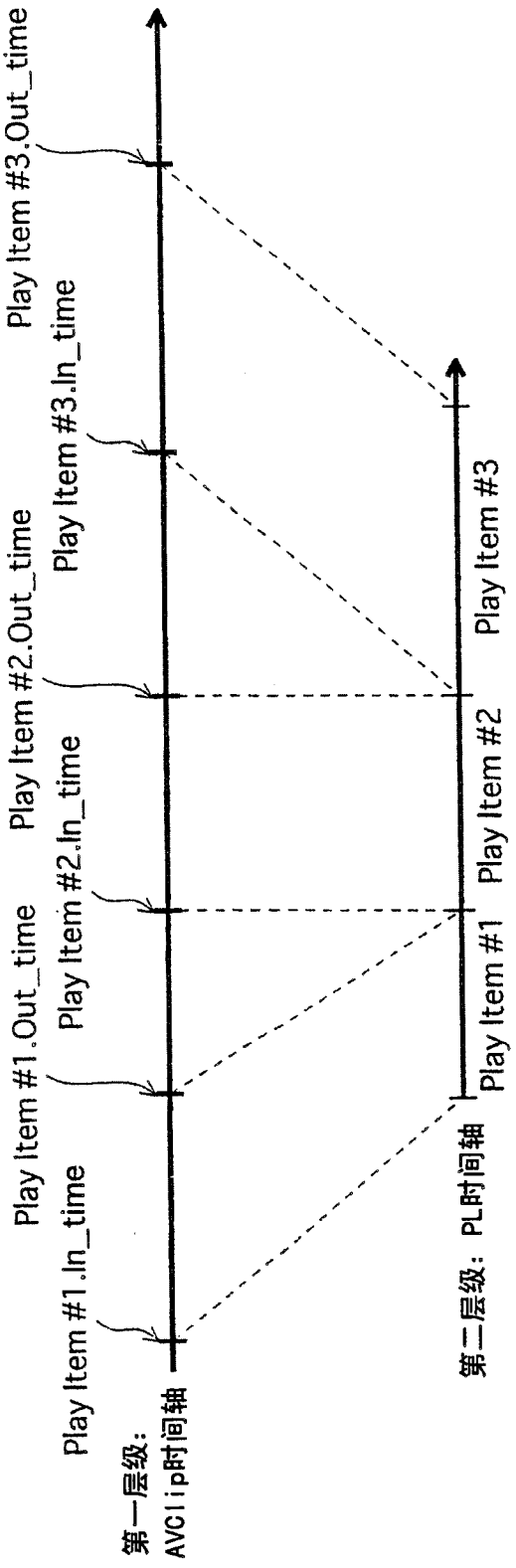
Play Item #3.Out_time
Play Item #3.In_time
Play Item #2.Out_time
Play Item #2.In_time
Play Item #1.Out_time
Play Item #1.In_time
第一层级：
AVClip时间轴
第二层级：PL时间轴
Play Item #1
Play Item #2
Play Item #3

图 5

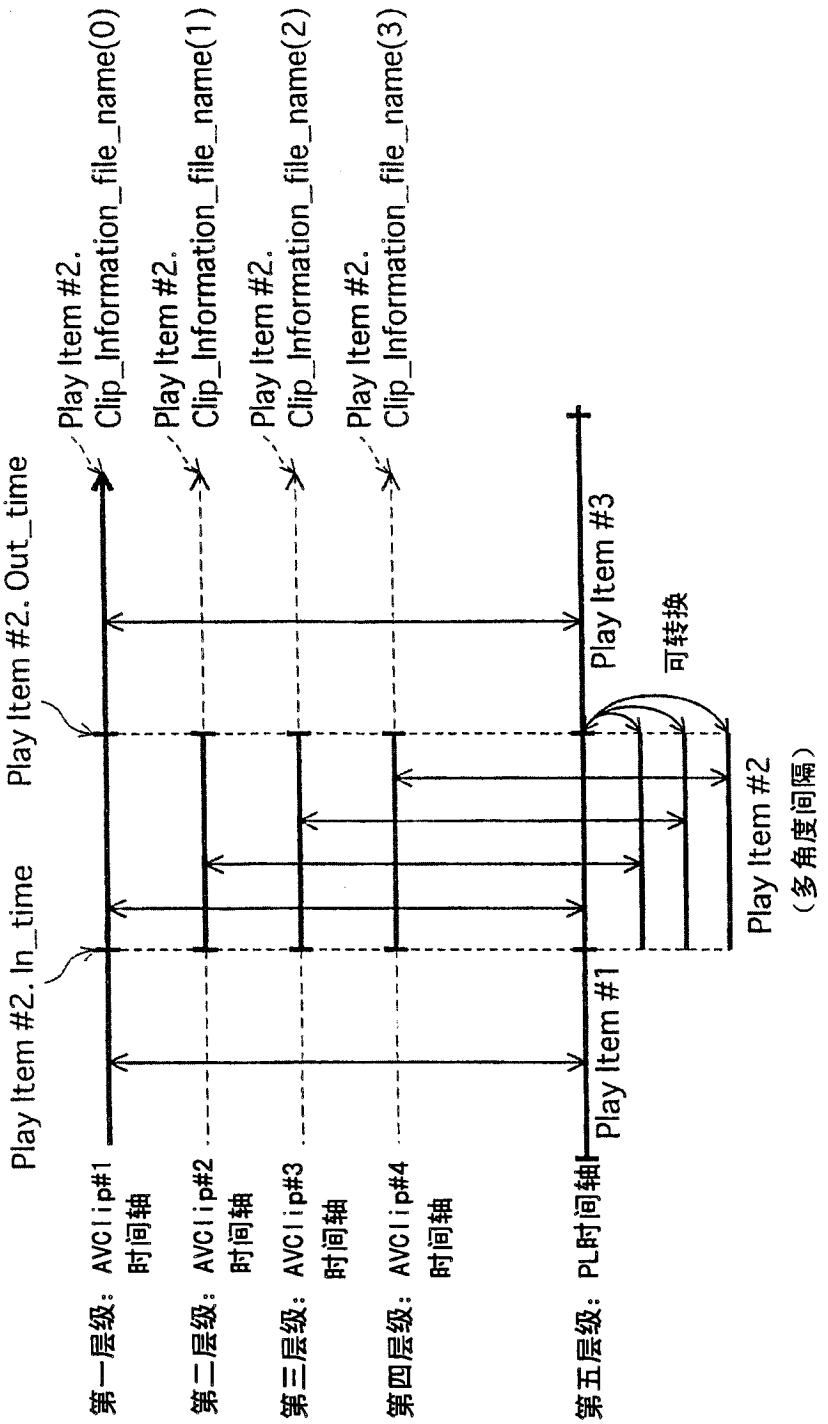
Play Item #2. In_time
Play Item #2. Out_time
Play Item #2. Clip_Information_file_name(0)
Play Item #2. Clip_Information_file_name(1)
Play Item #2. Clip_Information_file_name(2)
Play Item #2. Clip_Information_file_name(3)
第一层级：AVClip#1 时间轴
第二层级：AVClip#2 时间轴
第三层级：AVClip#3 时间轴
第四层级：AVClip#4 时间轴
第五层级：PL时间轴
Play Item #1
Play Item #3
可转换
Play Item #2
（多角度间隔）

图 6

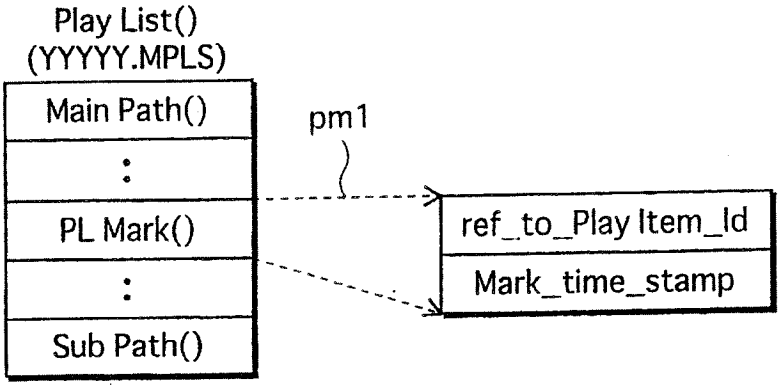
Play List()
(YYYYY.MPLS)
Main Path()
:
PL Mark()
:
Sub Path()
pm1
ref_to_Play Item_Id
Mark_time_stamp

图 7

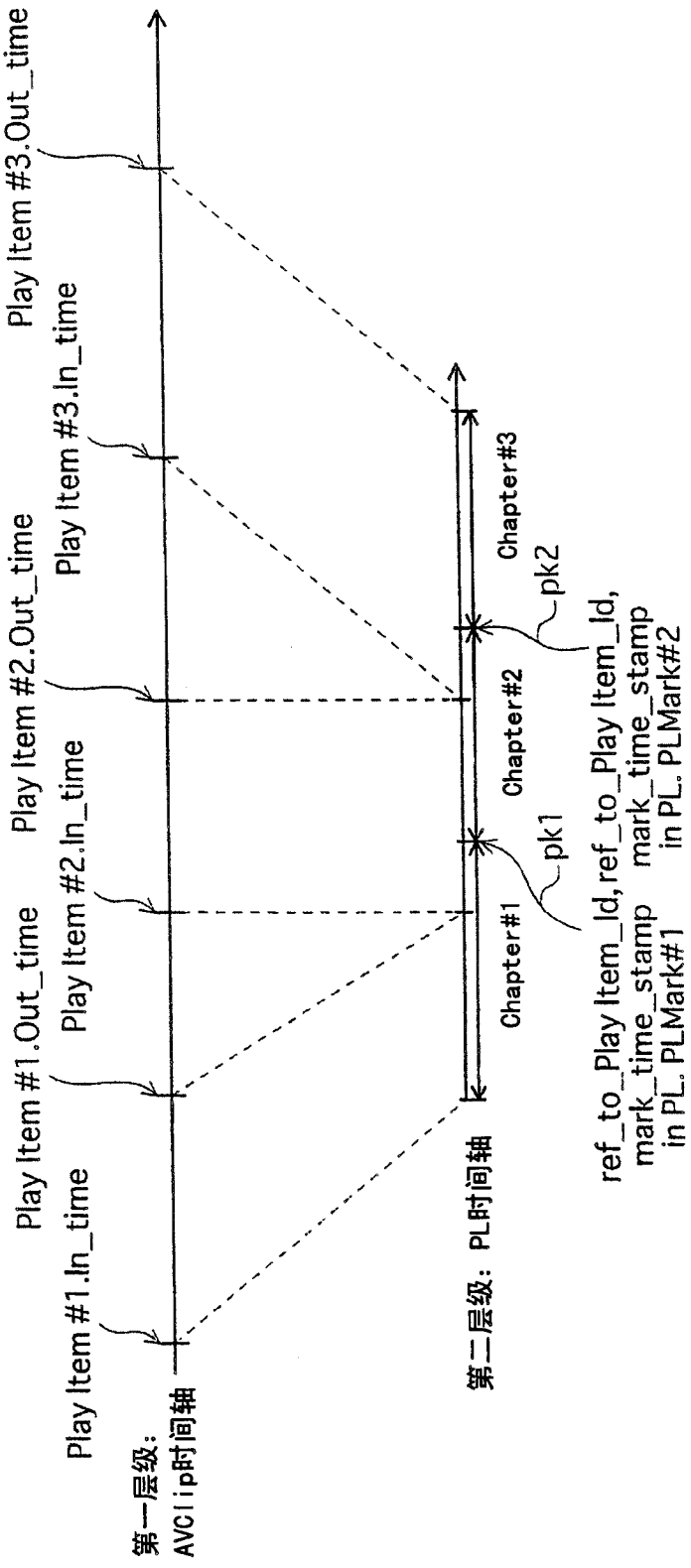
Play Item #3.Out_time
Play Item #3.In_time
Play Item #2.Out_time
Play Item #2.In_time
Play Item #1.Out_time
Play Item #1.In_time
第一层级：
AVClip时间轴
第二层级：PL时间轴
Chapter#3
Chapter#2
Chapter#1
pk2
pk1
ref_to_Play Item_Id,
mark_time_stamp
in PL. PLMark#1
ref_to_Play Item_Id,
mark_time_stamp
in PL. PLMark#2

图 8

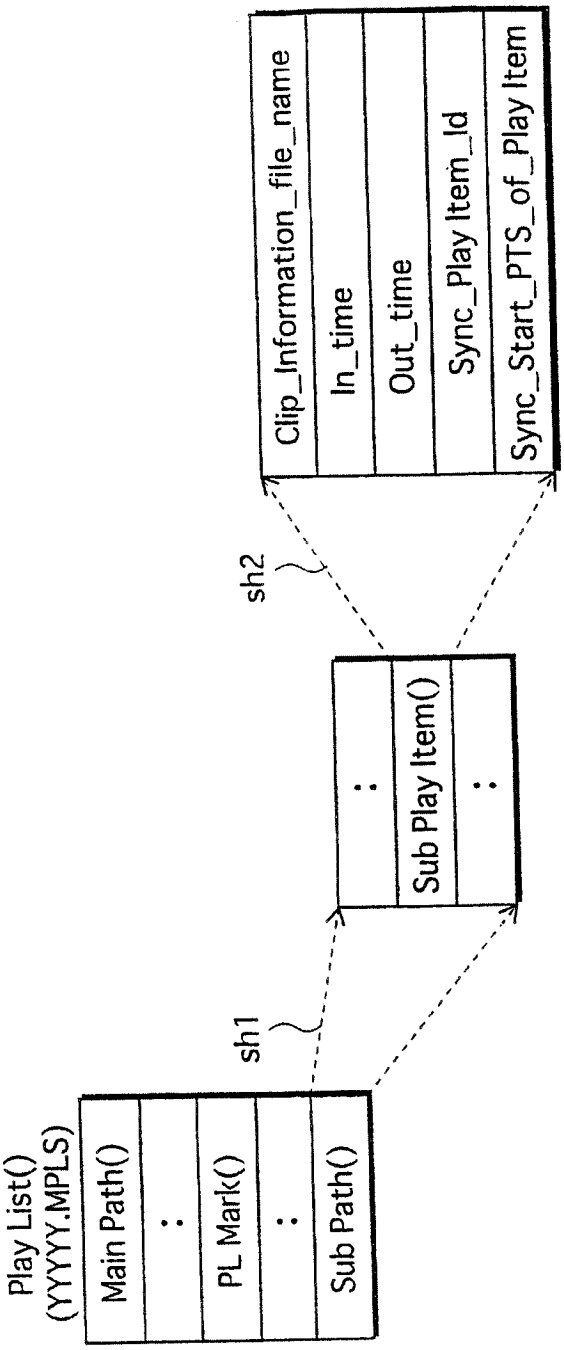
Play List()
(YYYYY.MPLS)
Main Path()
PL Mark()
Sub Path()
sh1
Sub Play Item()
sh2
Clip_Information_file_name
In_time
Out_time
Sync_Play Item_Id
Sync_Start_PTS_of_Play Item

图 9

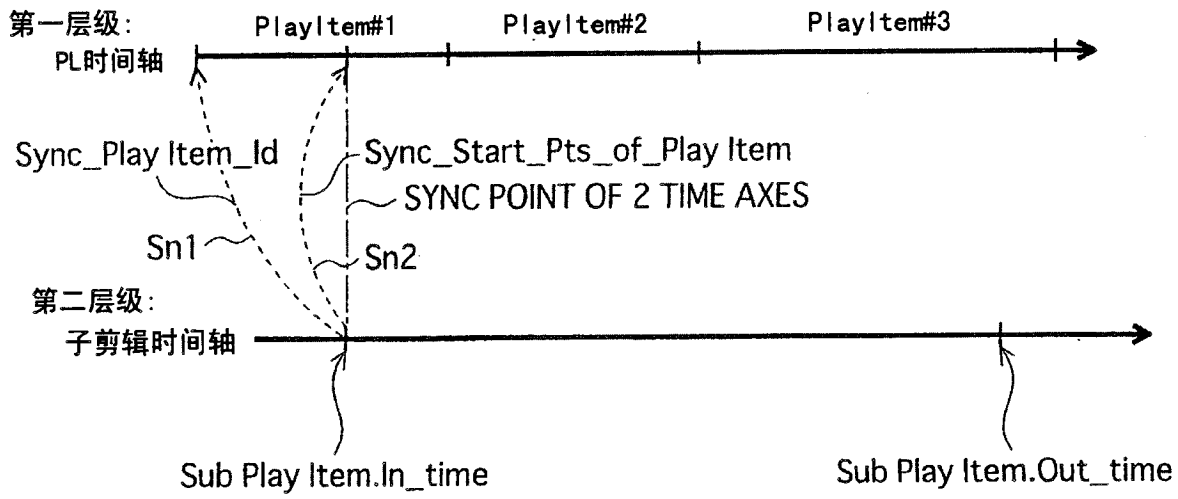
第一层级:
PL时间轴
PlayItem#1
PlayItem#2
PlayItem#3
Sync_Play Item_Id
Sync_Start_Pts_of_Play Item
SYNC POINT OF 2 TIME AXES
Sn1
Sn2
第二层级:
子剪辑时间轴
Sub Play Item.In_time
Sub Play Item.Out_time

图 10

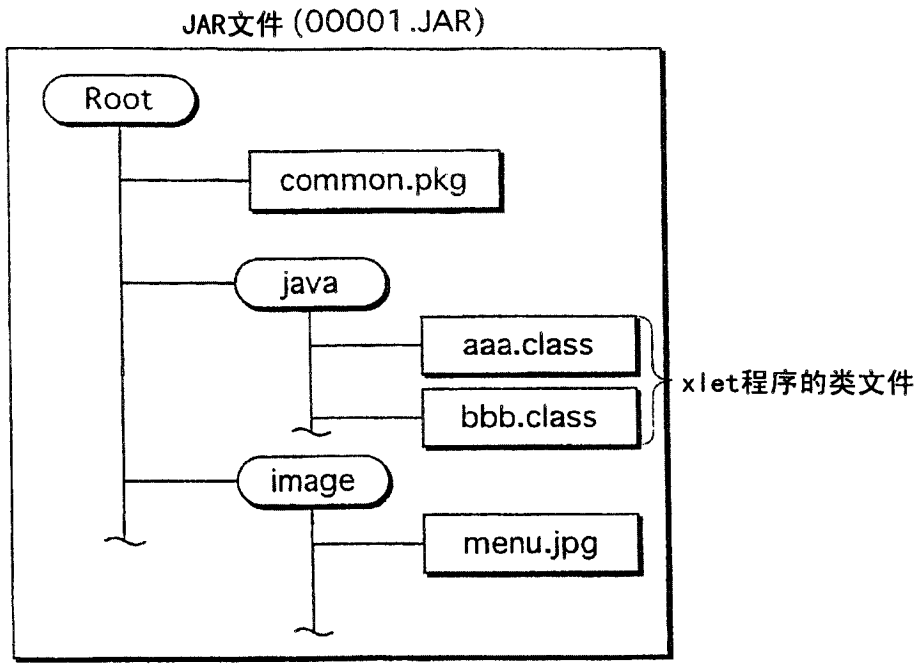
JAR文件 (00001.JAR)
Root
common.pkg
java
aaa.class
bbb.class
image
menu.jpg
xlet程序的类文件

图 11A

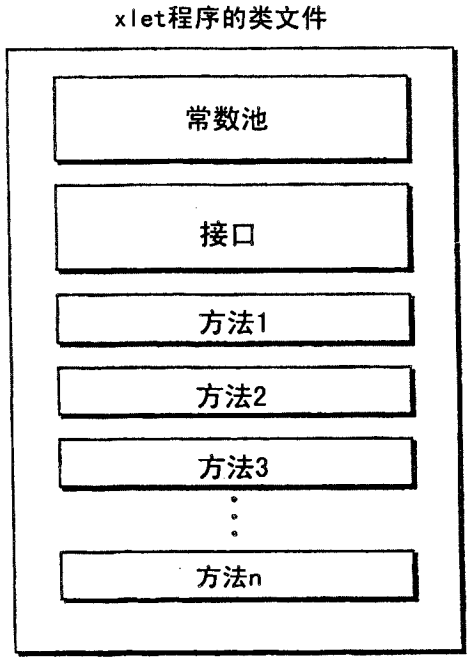
xlet程序的类文件
常数池
接口
方法1
方法2
方法3
方法n

图 11B

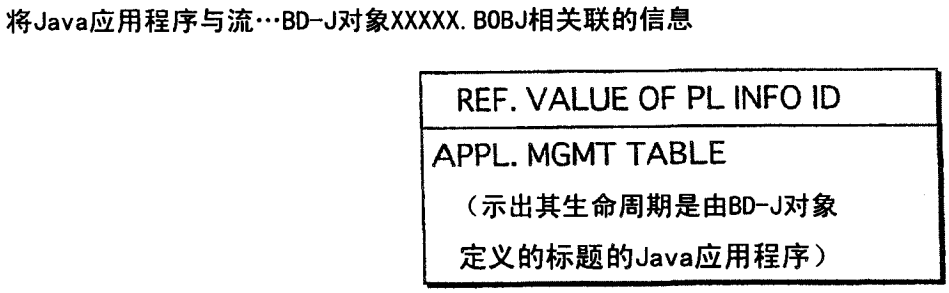
将Java应用程序与流…BD-J对象XXXXX.BOBJ相关联的信息
REF. VALUE OF PL INFO ID
APPL. MGMT TABLE
（示出其生命周期是由BD-J对象
定义的标题的Java应用程序）

图 12

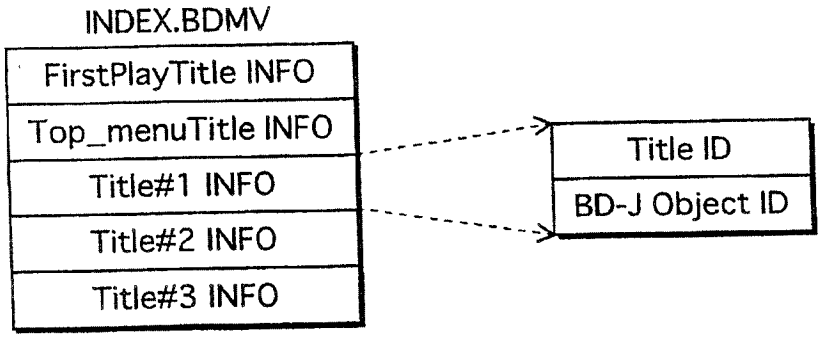
INDEX.BDMV
FirstPlayTitle INFO
Top_menuTitle INFO
Title#1 INFO
Title#2 INFO
Title#3 INFO
Title ID
BD-J Object ID

图 13

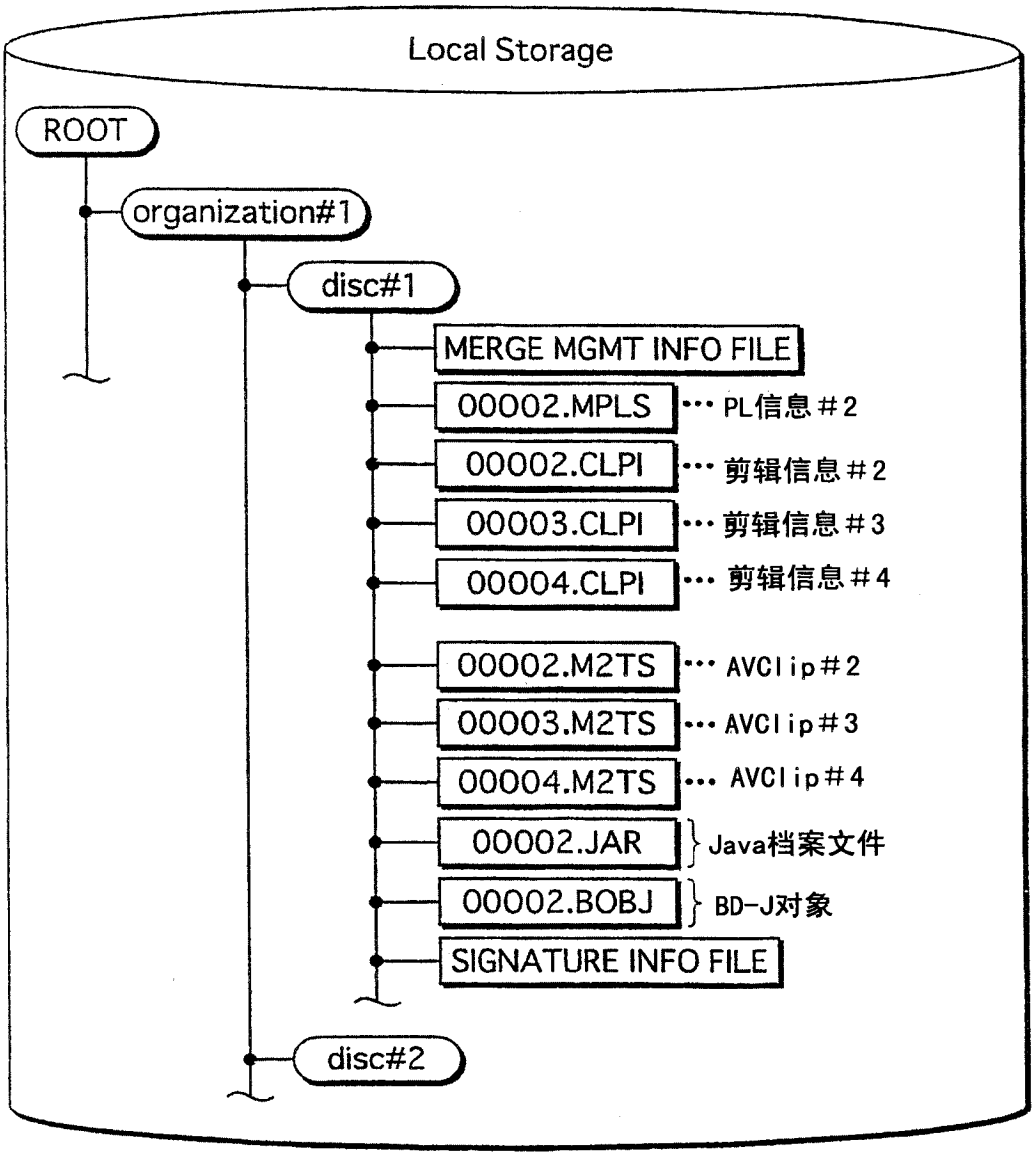
Local Storage
ROOT
organization#1
disc#1
MERGE MGMT INFO FILE
00002.MPLS
PL信息＃2
00002.CLPI
剪辑信息＃2
00003.CLPI
剪辑信息＃3
00004.CLPI
剪辑信息＃4
00002.M2TS
AVClip＃2
00003.M2TS
AVClip＃3
00004.M2TS
AVClip＃4
00002.JAR
Java档案文件
00002.BOBJ
BD-J对象
SIGNATURE INFO FILE
disc#2

图 14

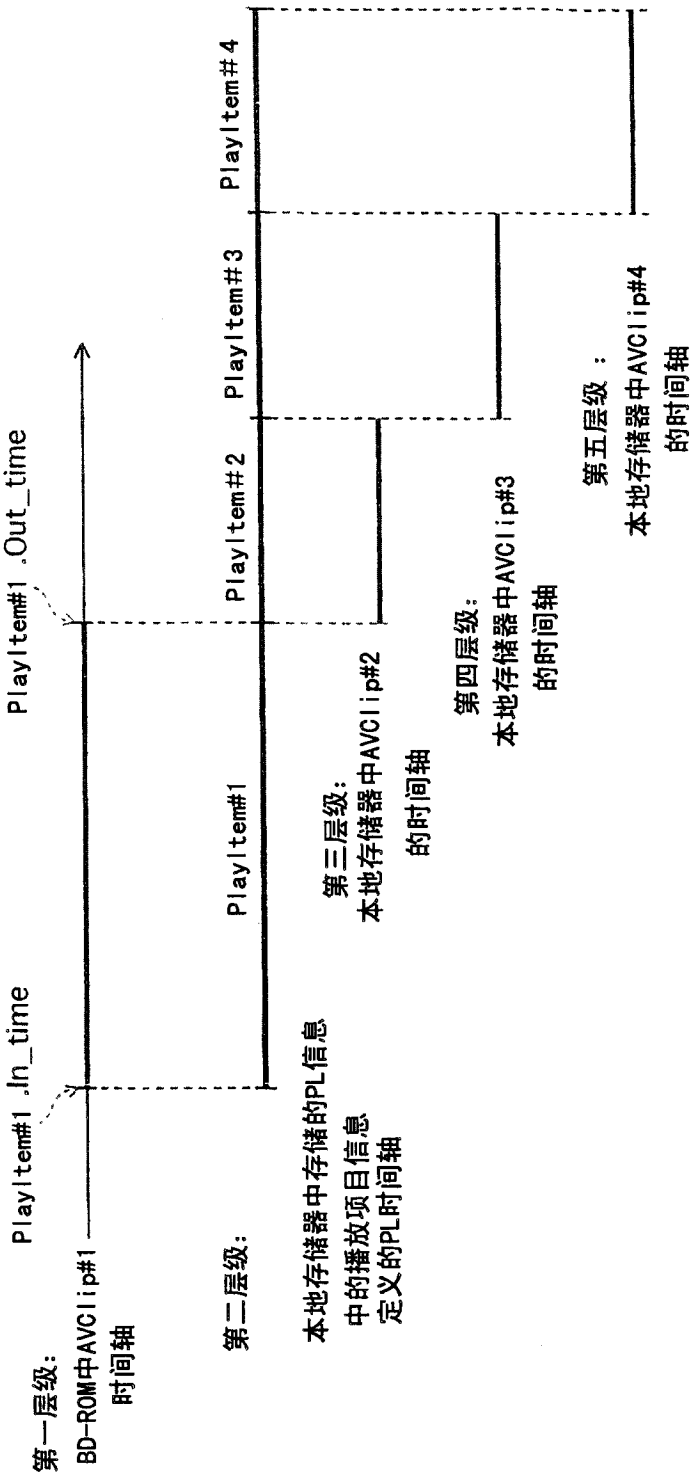
第一层级：
BD-ROM中AVClip#1
时间轴
Playltem#1 .In_time
Playltem#1 .Out_time
第二层级：
本地存储器中存储的PL信息
中的播放项目信息
定义的PL时间轴
Playltem#1
Playltem#2
第三层级：
本地存储器中AVClip#2
的时间轴
Playltem#3
第四层级：
本地存储器中AVClip#3
的时间轴
Playltem#4
第五层级 ：
本地存储器中AVClip#4
的时间轴

图 15

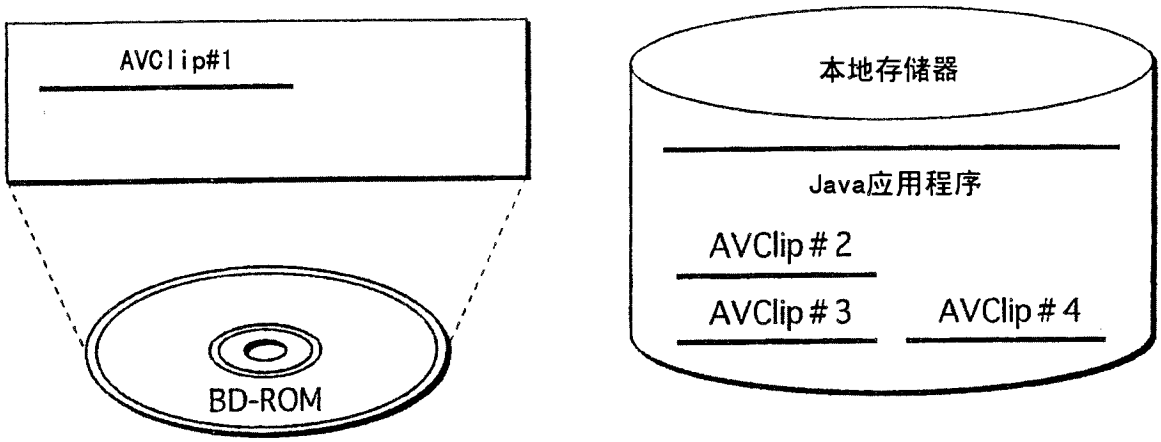

图 16A

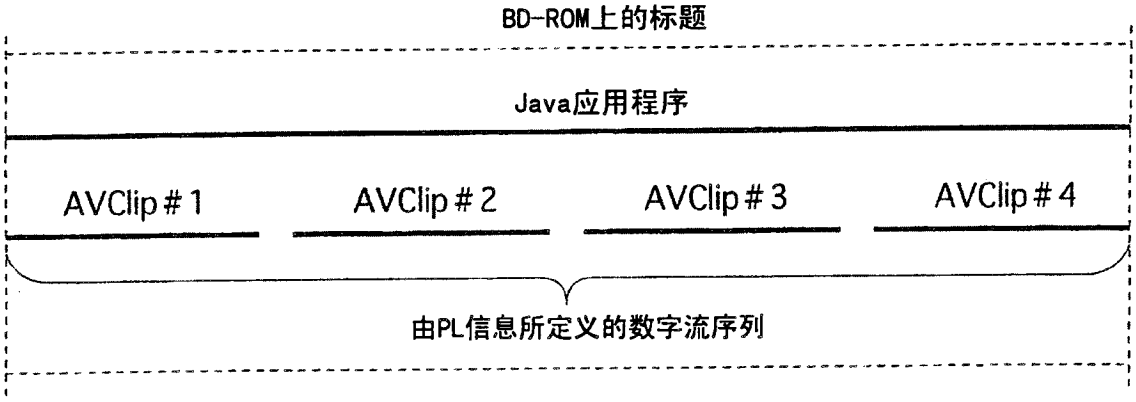

图 16B

光盘ID	文件路径	散列值
disc#1	ROOT/organization#1/disc#1/00002.MPLS	××××
	⋮	⋮

图 17

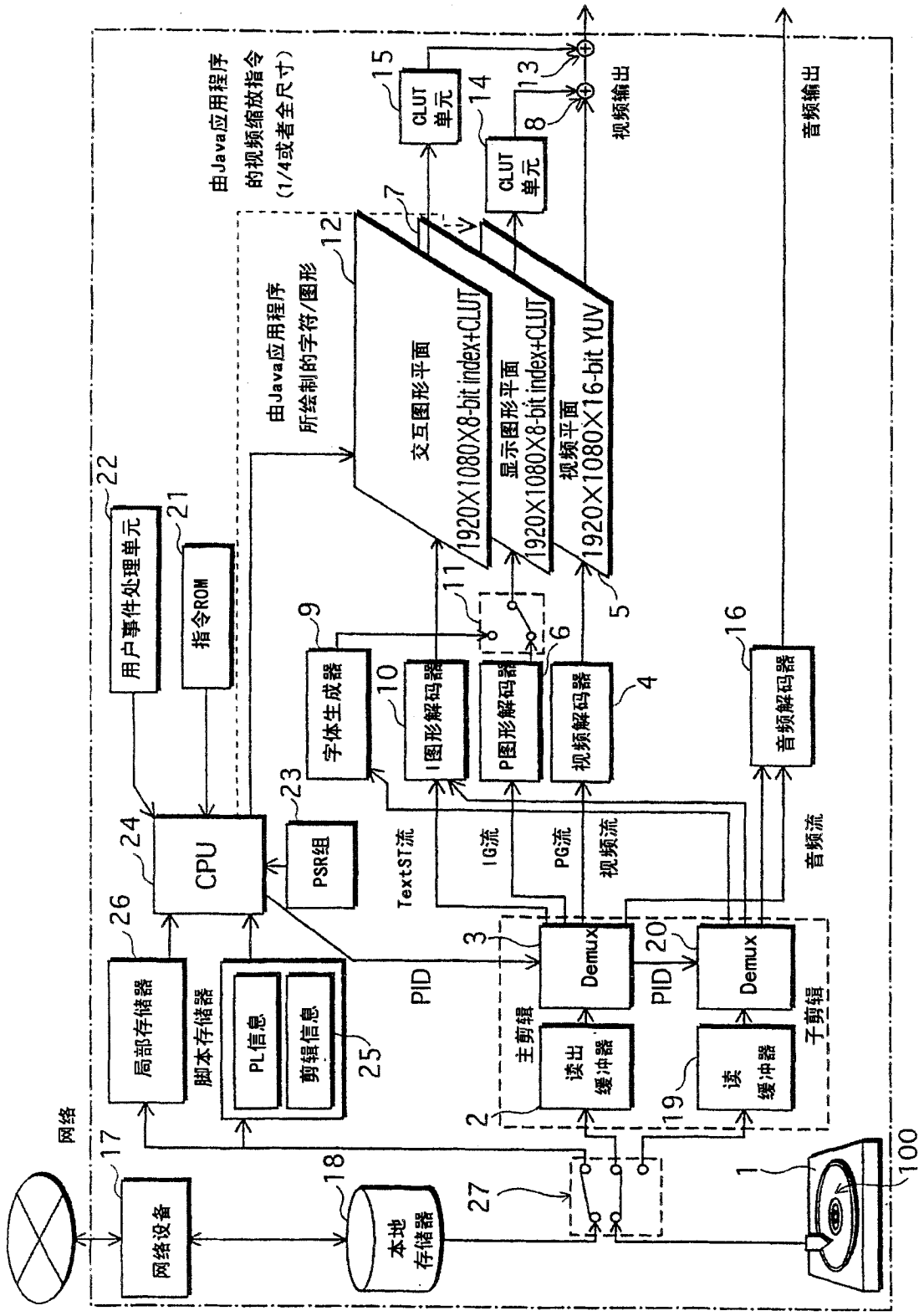
网络
网络设备
本地存储器
局部存储器
脚本存储器
PL信息
剪辑信息
用户事件处理单元
指令ROM
CPU
PSR组
字体生成器
I图形解码器
P图形解码器
视频解码器
音频解码器
Demux
Demux
读出缓冲器
读缓冲器
主剪辑
子剪辑
PID
PID
TextST流
IG流
PG流
视频流
音频流
由Java应用程序的视频缩放指令(1/4或者全尺寸)
由Java应用程序所绘制的字符/图形
交互图形平面
1920X1080X8-bit index+CLUT
显示图形平面
1920X1080X8-bit index+CLUT
视频平面
1920X1080X16-bit YUV
CLUT单元
CLUT单元
视频输出
音频输出
100

图 18

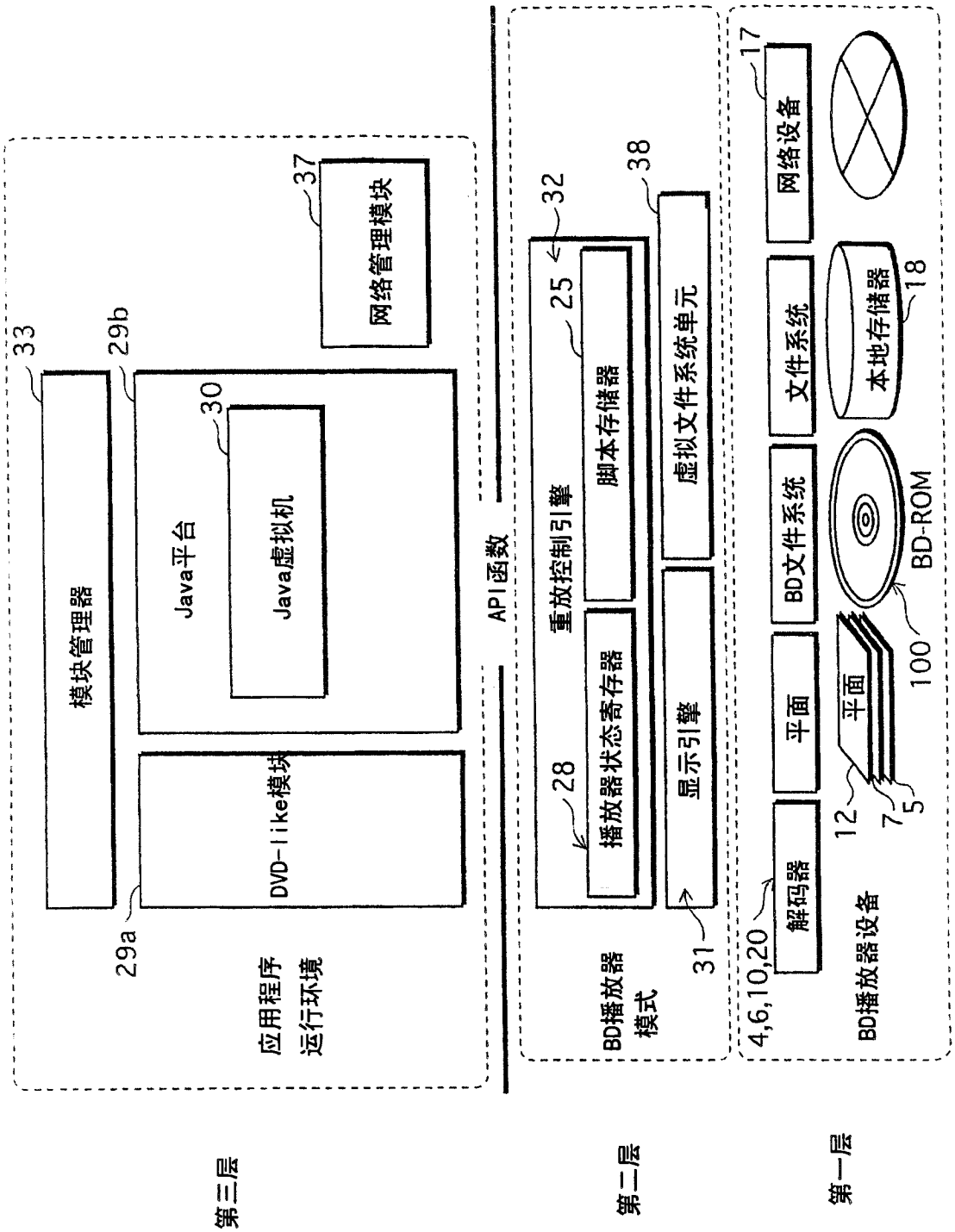
第三层
第二层
第一层
应用程序
运行环境
BD播放器
模式
BD播放器设备
DVD-like模块
Java平台
Java虚拟机
模块管理器
网络管理模块
API函数
重放控制引擎
播放器状态寄存器
脚本存储器
显示引擎
虚拟文件系统单元
解码器
平面
平面
BD文件系统
文件系统
本地存储器
网络设备
BD-ROM
29a
29b
30
33
37
28
25
32
38
31
4,6,10,20
12
7
5
100
17
18

图 19

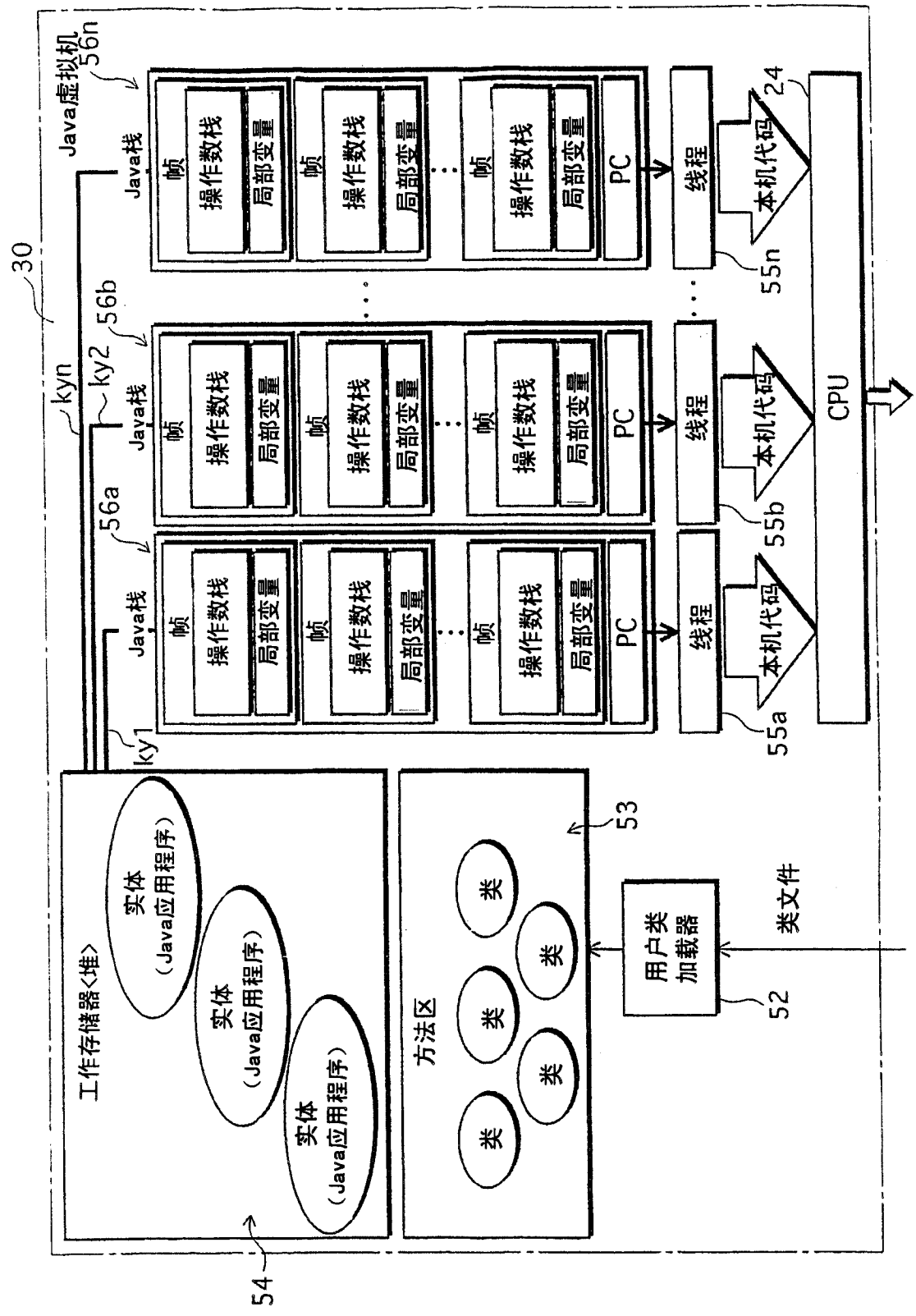
30
Java虚拟机
56n
56b
56a
Java栈
帧
操作数栈
局部变量
PC
线程
本机代码
24
55n
55b
55a
CPU
kyn
ky2
ky1
工作存储器<堆>
实体
（Java应用程序）
方法区
类
53
用户类
加载器
52
类文件
54

图 20

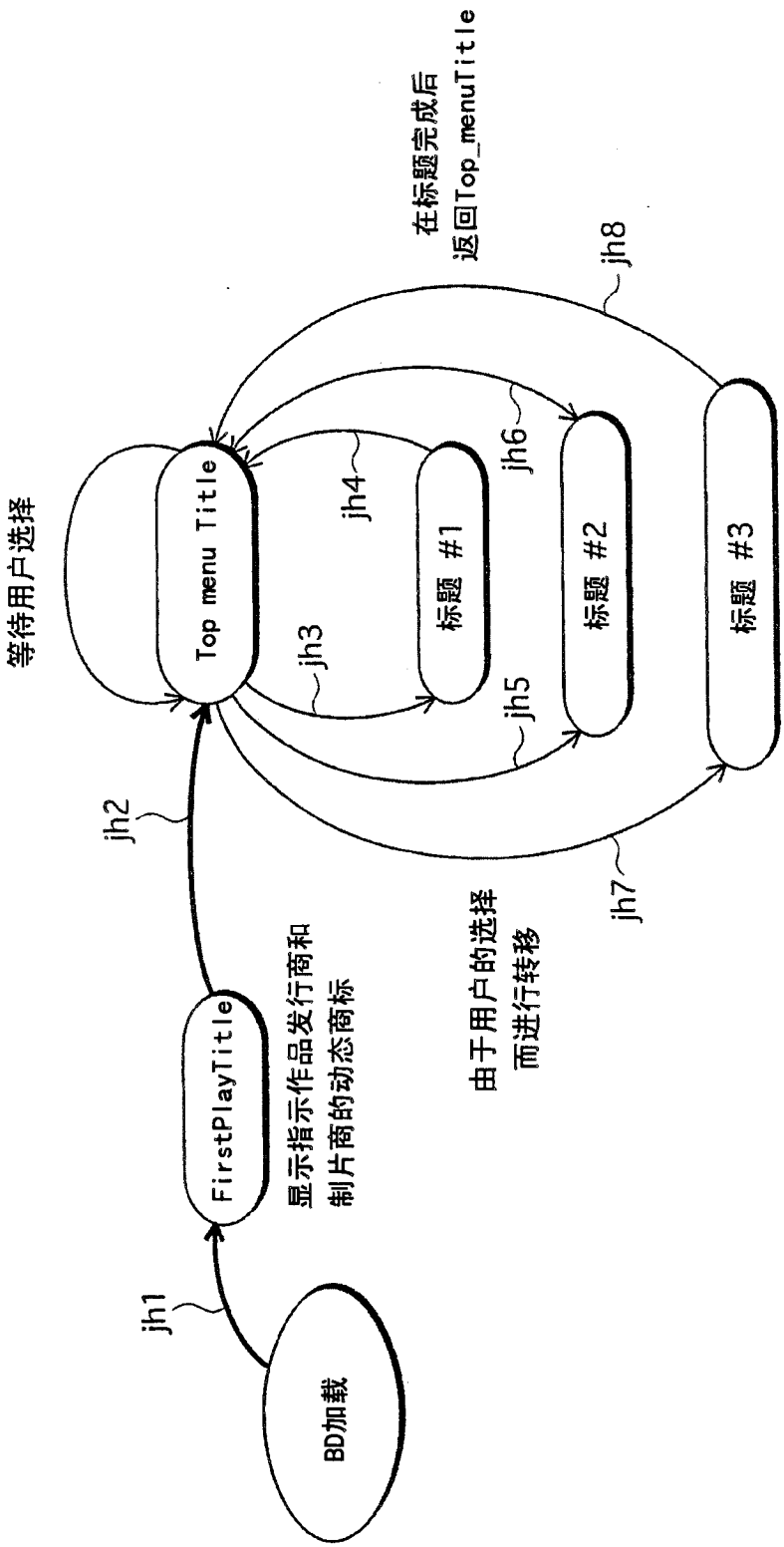
BD加载
jh1
FirstPlayTitle
显示指示作品发行商和
制片商的动态商标
jh2
Top menu Title
等待用户选择
jh3
jh4
标题 #1
jh5
jh6
标题 #2
由于用户的选择
而进行转移
jh7
标题 #3
jh8
在标题完成后
返回Top_menuTitle

图 21

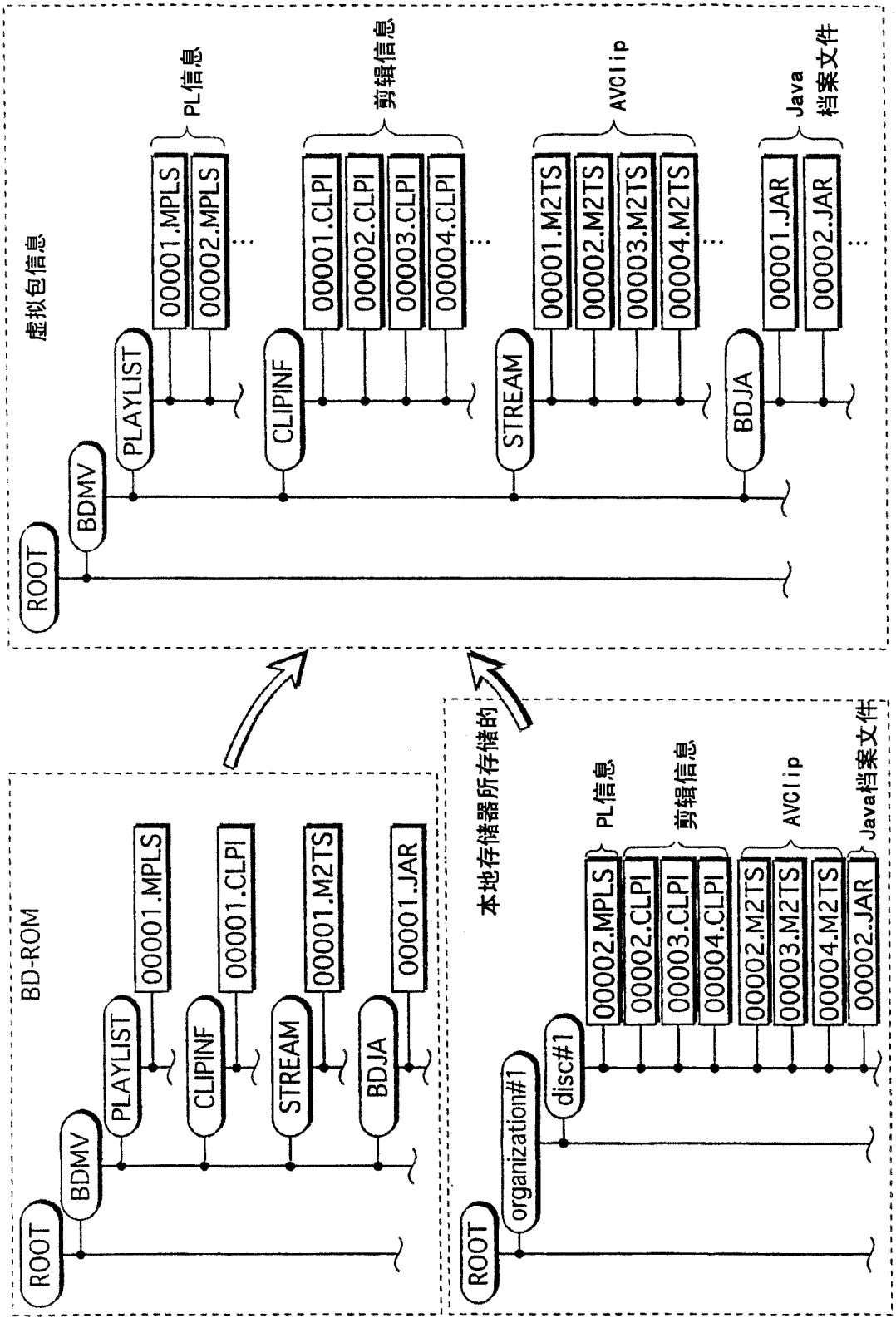
虚拟包信息
ROOT
BDMV
PLAYLIST
CLIPINF
STREAM
BDJA
00001.MPLS
00002.MPLS
00001.CLPI
00002.CLPI
00003.CLPI
00004.CLPI
00001.M2TS
00002.M2TS
00003.M2TS
00004.M2TS
00001.JAR
00002.JAR
PL信息
剪辑信息
AVClip
Java
档案文件
BD-ROM
本地存储器所存储的
organization#1
disc#1
Java档案文件

图 22


BD-ROM被退出
BD-ROM被加载
整个光盘的时间轴

图 23A

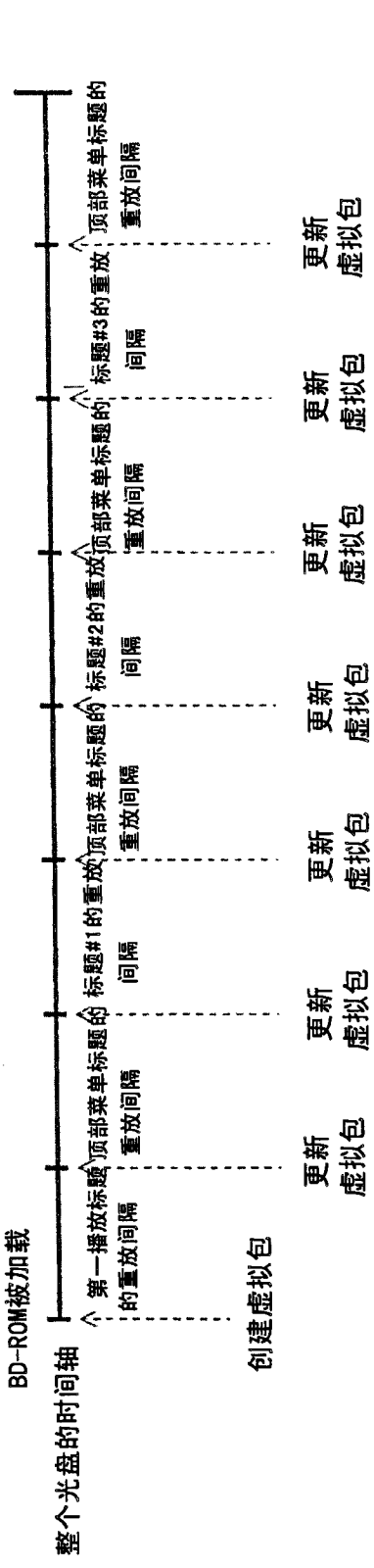
BD-ROM被退出
BD-ROM被加载
整个光盘的时间轴
第一播放标题的重放间隔
顶部菜单标题的重放间隔
标题#1的重放间隔
顶部菜单标题的重放间隔
标题#2的重放间隔
顶部菜单标题的重放间隔
标题#3的重放间隔
顶部菜单标题的重放间隔
创建虚拟包
更新虚拟包
更新虚拟包
更新虚拟包
更新虚拟包
更新虚拟包

图 23B

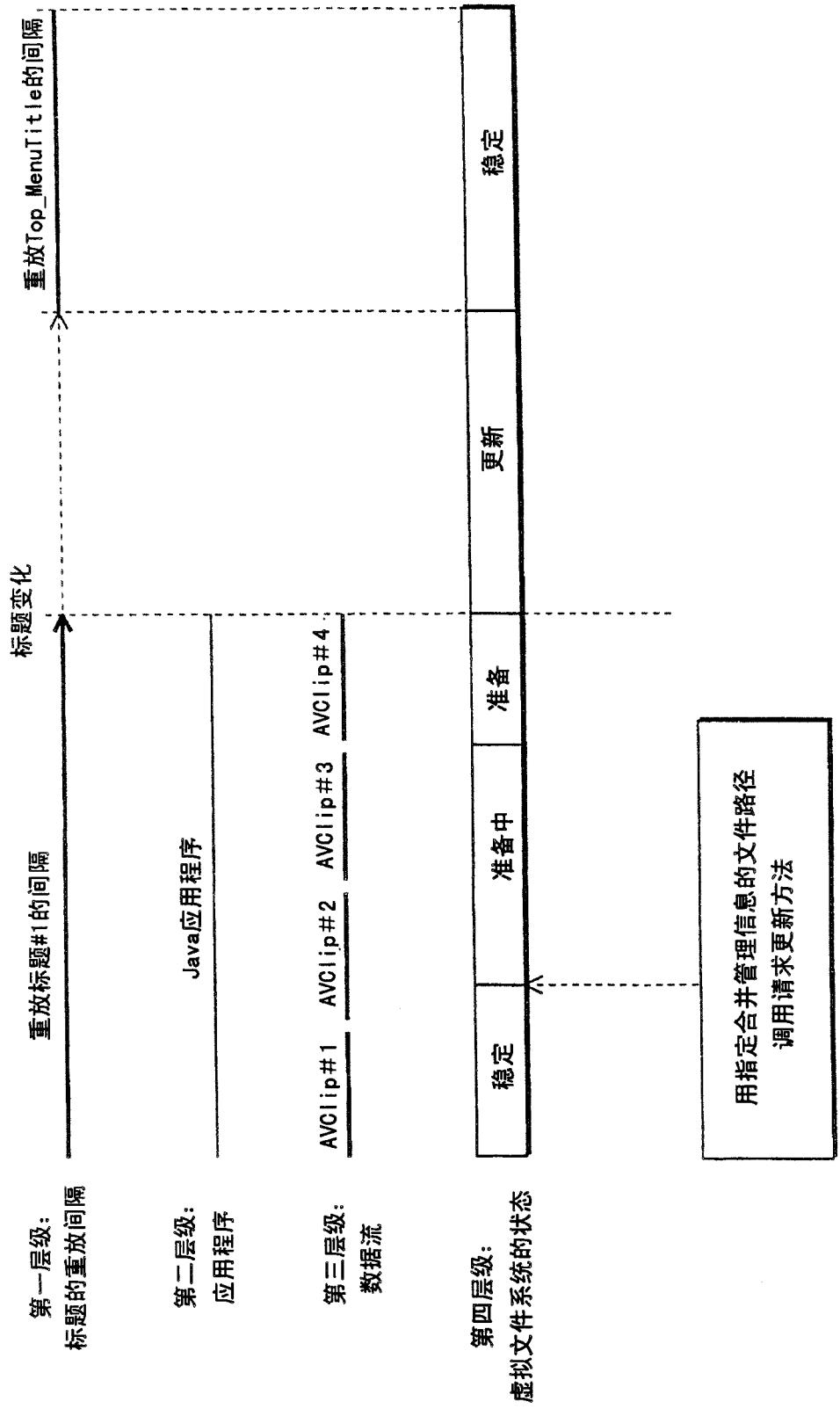
第一层级：
标题的重放间隔
第二层级：
应用程序
第三层级：
数据流
第四层级：
虚拟文件系统的状态
标题变化
重放标题#1的间隔
重放Top_MenuTitle的间隔
Java应用程序
AVClip#1
AVClip#2
AVClip#3
AVClip#4
稳定
准备中
准备
更新
稳定
用指定合并管理信息的文件路径
调用请求更新方法

图 24

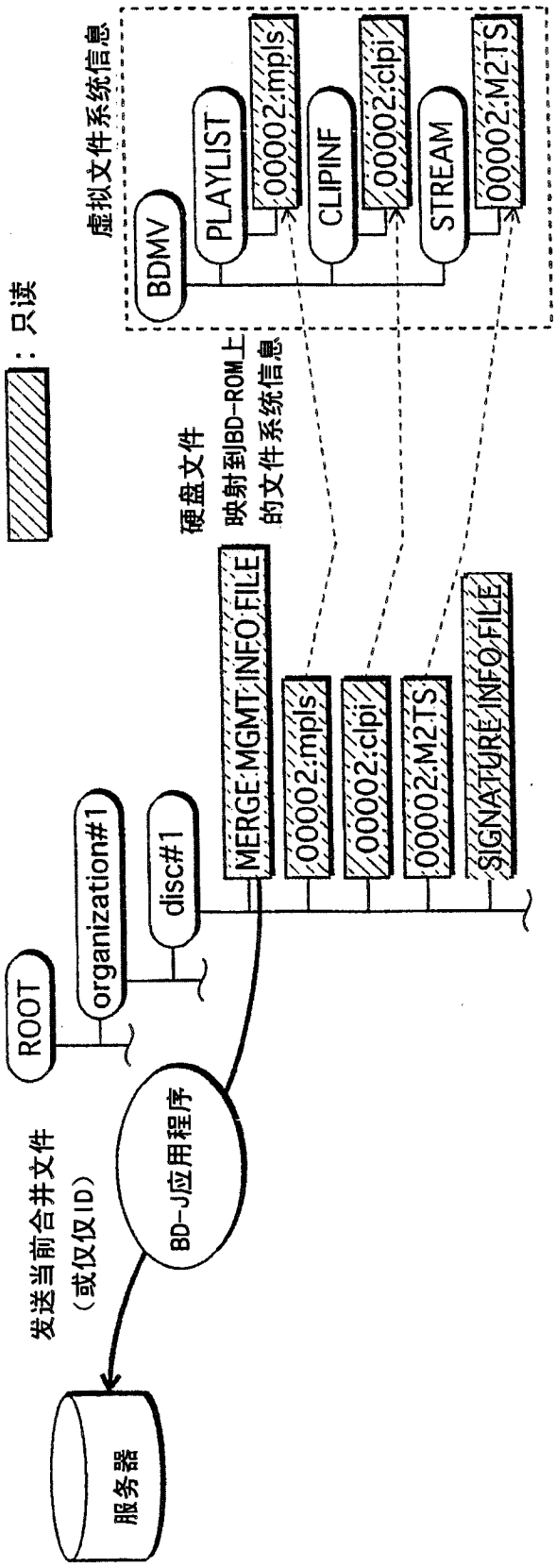
虚拟文件系统信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00002.clpi
STREAM
00002.M2TS
：只读
硬盘文件
映射到BD-ROM上
的文件系统信息
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE.INFO.FILE
ROOT
organization#1
disc#1
BD-J应用程序
发送当前合并文件
（或仅仅ID）
服务器

图 25

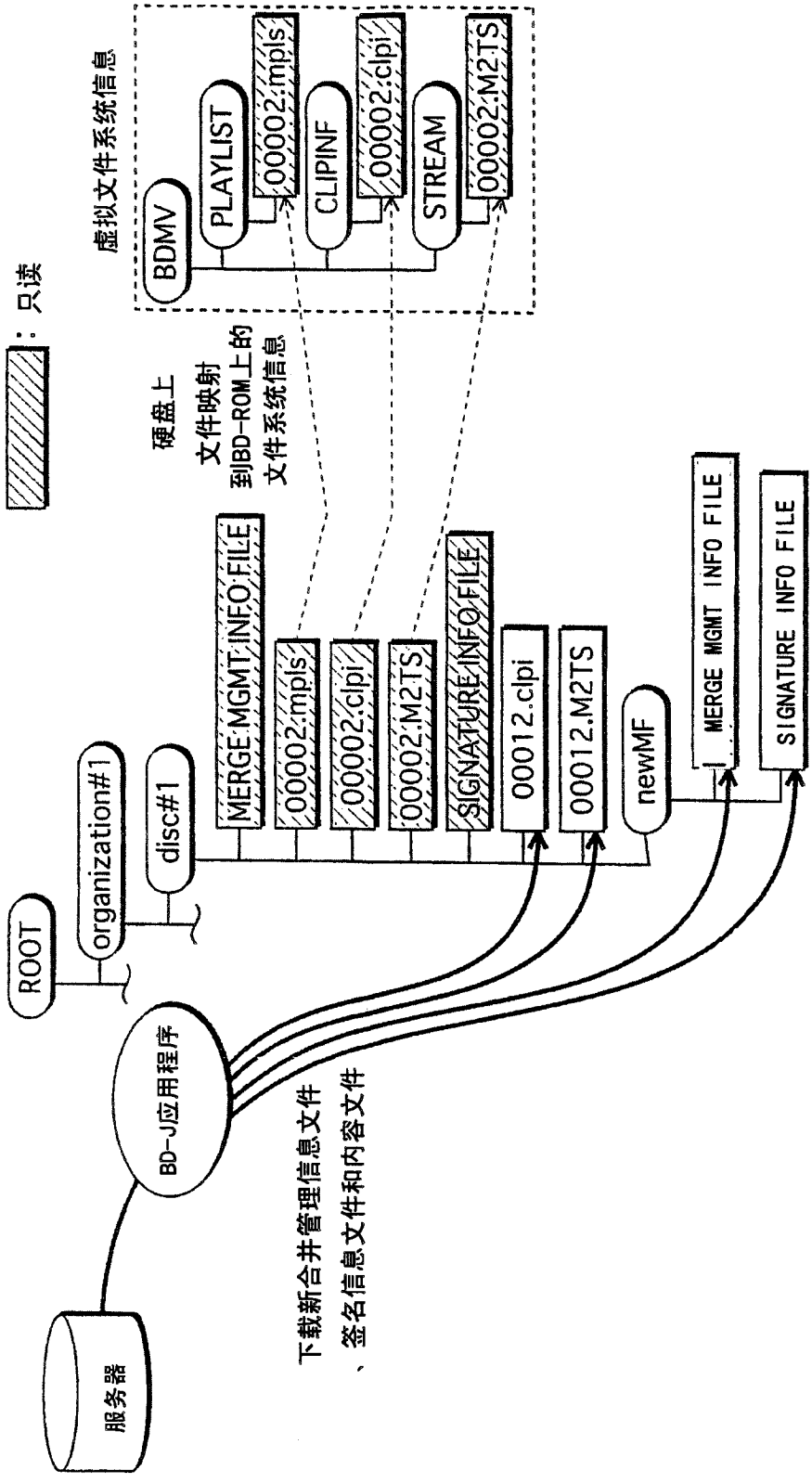
虚拟文件系统信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00002.clpi
STREAM
00002.M2TS
：只读
硬盘上
文件映射
到BD-ROM上的
文件系统信息
ROOT
organization#1
disc#1
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE.INFO.FILE
00012.clpi
00012.M2TS
newMF
MERGE MGMT INFO FILE
SIGNATURE INFO FILE
BD-J应用程序
服务器
下载新合并管理信息文件
、签名信息文件和内容文件

图 26

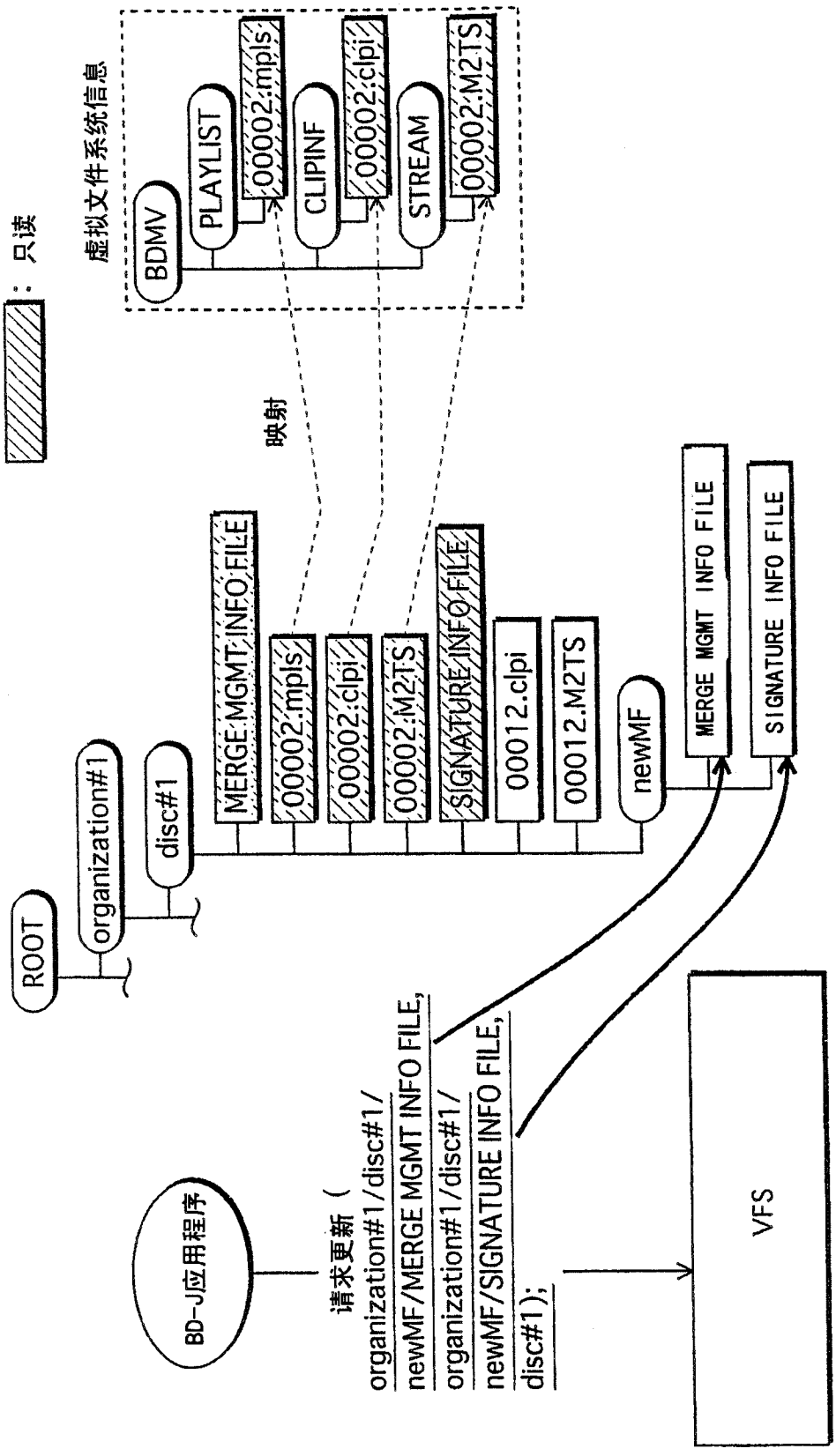
虚拟文件系统信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00002.clpi
STREAM
00002.M2TS
映射
: 只读
ROOT
organization#1
disc#1
MERGE MGMT INFO FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE INFO FILE
00012.clpi
00012.M2TS
newMF
MERGE MGMT INFO FILE
SIGNATURE INFO FILE
BD-J应用程序
请求更新（
organization#1/disc#1/
newMF/MERGE MGMT INFO FILE,
organization#1/disc#1/
newMF/SIGNATURE INFO FILE,
disc#1);
VFS

图 27

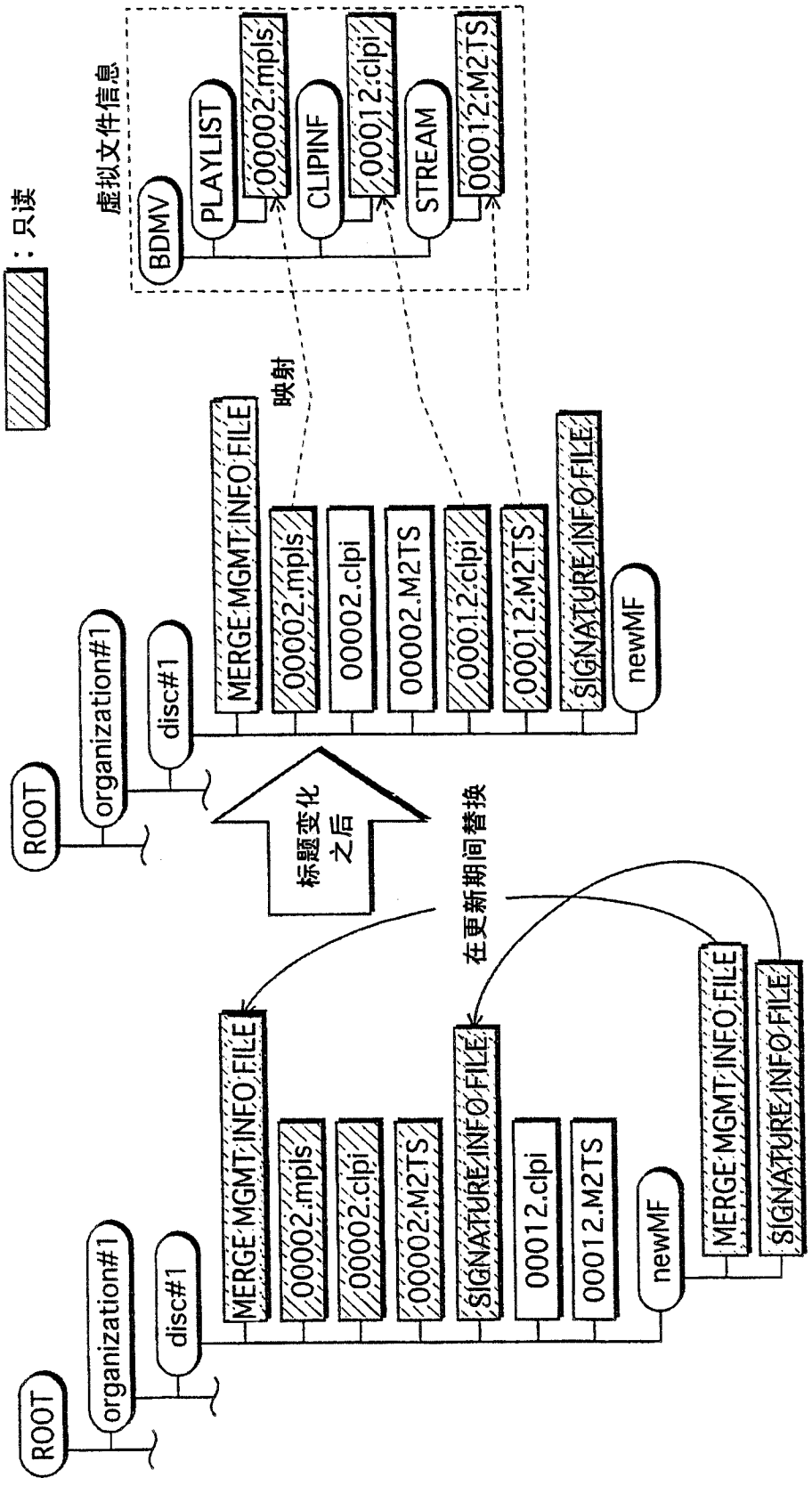
：只读
虚拟文件信息
BDMV
PLAYLIST
00002.mpls
CLIPINF
00012.clpi
STREAM
00012.M2TS
映射
ROOT
organization#1
disc#1
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
00012.clpi
00012.M2TS
SIGNATURE.INFO.FILE
newMF
标题变化之后
在更新期间替换
ROOT
organization#1
disc#1
MERGE.MGMT.INFO.FILE
00002.mpls
00002.clpi
00002.M2TS
SIGNATURE.INFO.FILE
00012.clpi
00012.M2TS
newMF
MERGE.MGMT.INFO.FILE
SIGNATURE.INFO.FILE

图 28

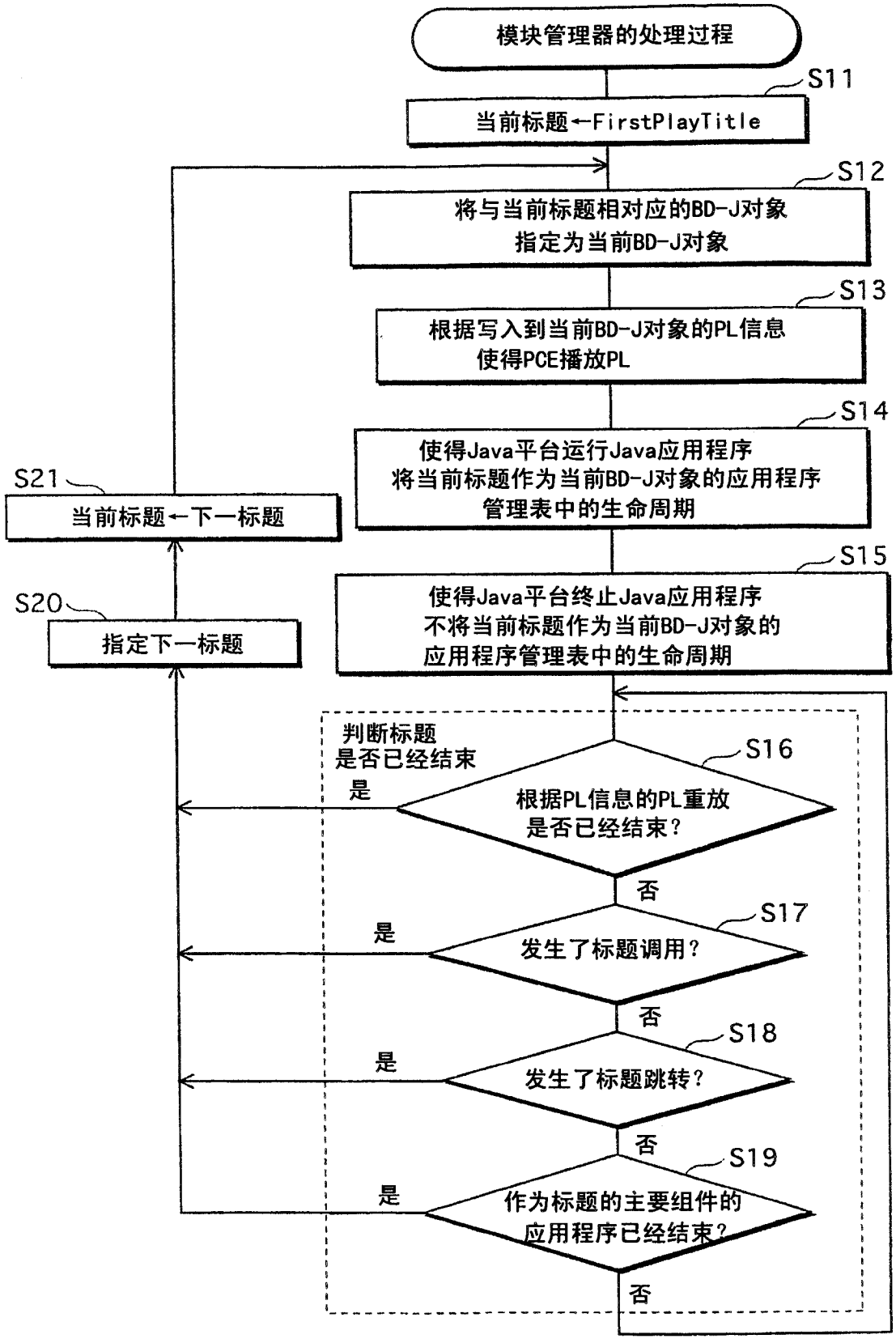
模块管理器的处理过程
S11
当前标题←FirstPlayTitle
S12
将与当前标题相对应的BD-J对象
指定为当前BD-J对象
S13
根据写入到当前BD-J对象的PL信息
使得PCE播放PL
S14
使得Java平台运行Java应用程序
将当前标题作为当前BD-J对象的应用程序
管理表中的生命周期
S15
使得Java平台终止Java应用程序
不将当前标题作为当前BD-J对象的
应用程序管理表中的生命周期
判断标题
是否已经结束
S16
根据PL信息的PL重放
是否已经结束?
是
否
S17
发生了标题调用?
是
否
S18
发生了标题跳转?
是
否
S19
作为标题的主要组件的
应用程序已经结束?
是
否
S20
指定下一标题
S21
当前标题←下一标题

图 29

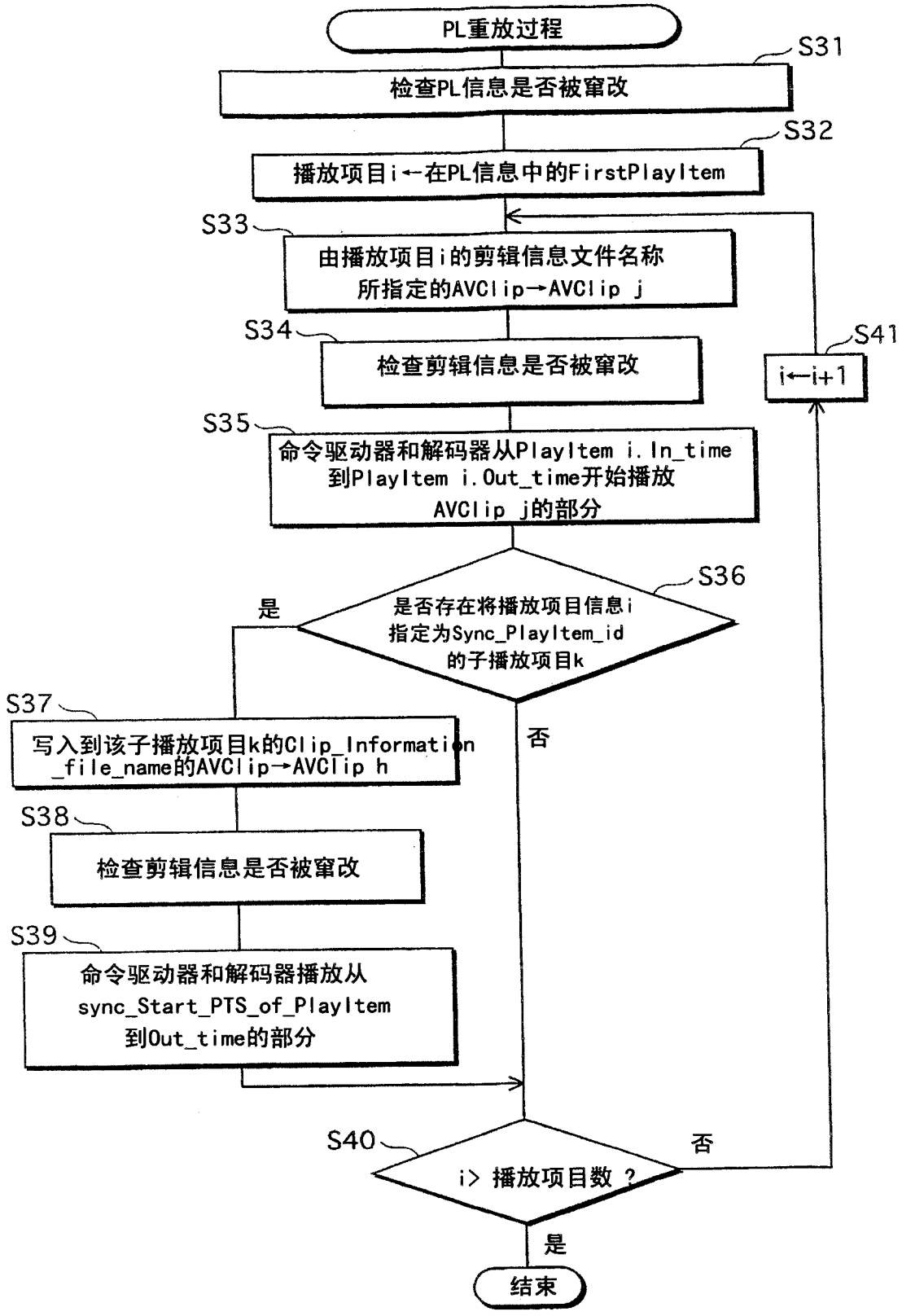
PL重放过程
S31
检查PL信息是否被窜改
S32
播放项目i←在PL信息中的FirstPlayItem
S33
由播放项目i的剪辑信息文件名称
所指定的AVClip→AVClip j
S34
检查剪辑信息是否被窜改
S41
i←i+1
S35
命令驱动器和解码器从PlayItem i.In_time
到PlayItem i.Out_time开始播放
AVClip j的部分
S36
是否存在将播放项目信息i
指定为Sync_PlayItem_id
的子播放项目k
是
否
S37
写入到该子播放项目k的Clip_Information
_file_name的AVClip→AVClip h
S38
检查剪辑信息是否被窜改
S39
命令驱动器和解码器播放从
sync_Start_PTS_of_PlayItem
到Out_time的部分
S40
i> 播放项目数 ?
否
是
结束

图 30

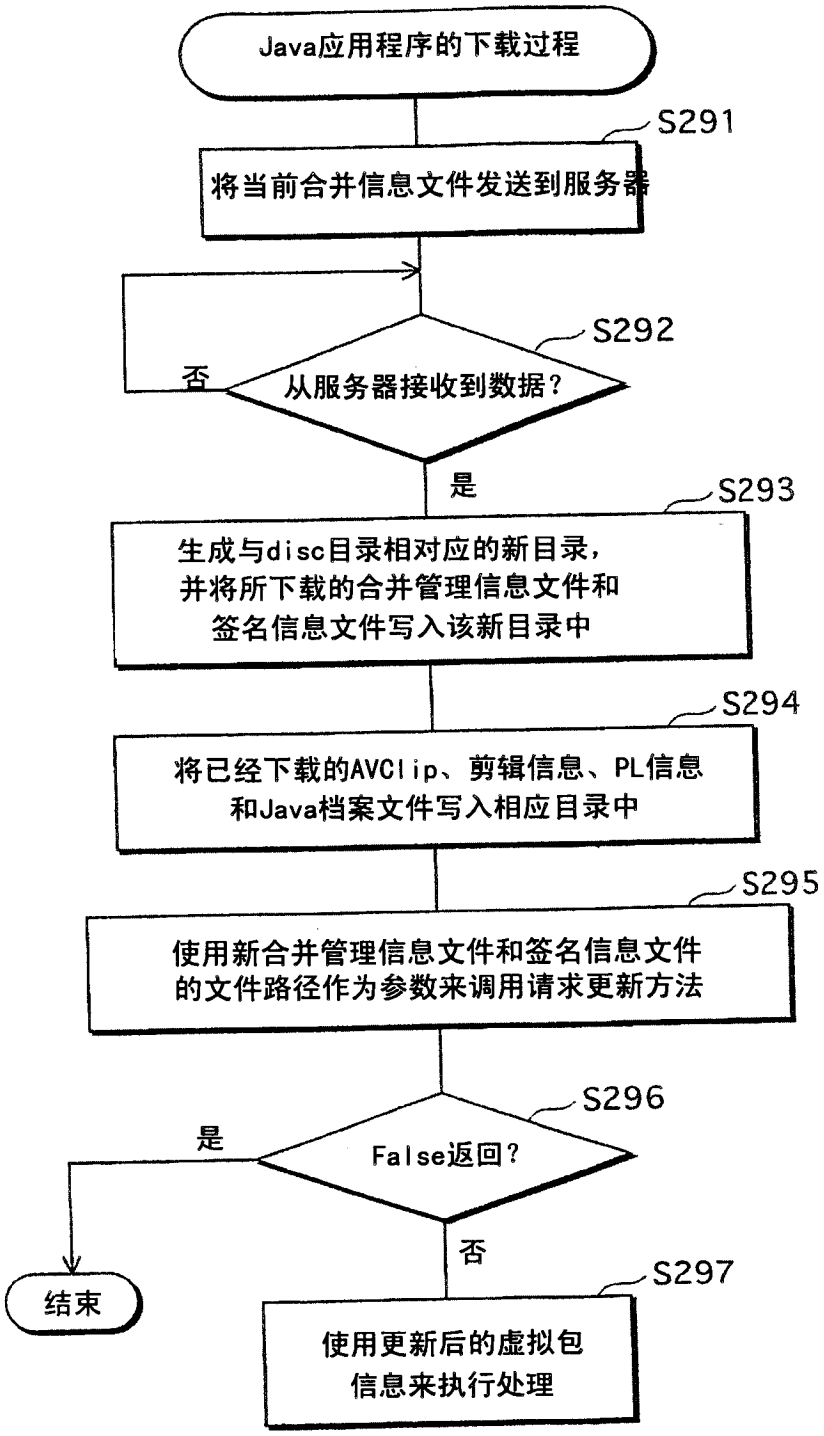
Java应用程序的下载过程
S291
将当前合并信息文件发送到服务器
S292
否
从服务器接收到数据？
是
S293
生成与disc目录相对应的新目录，
并将所下载的合并管理信息文件和
签名信息文件写入该新目录中
S294
将已经下载的AVClip、剪辑信息、PL信息
和Java档案文件写入相应目录中
S295
使用新合并管理信息文件和签名信息文件
的文件路径作为参数来调用请求更新方法
S296
是
False返回？
否
结束
S297
使用更新后的虚拟包
信息来执行处理

图 31

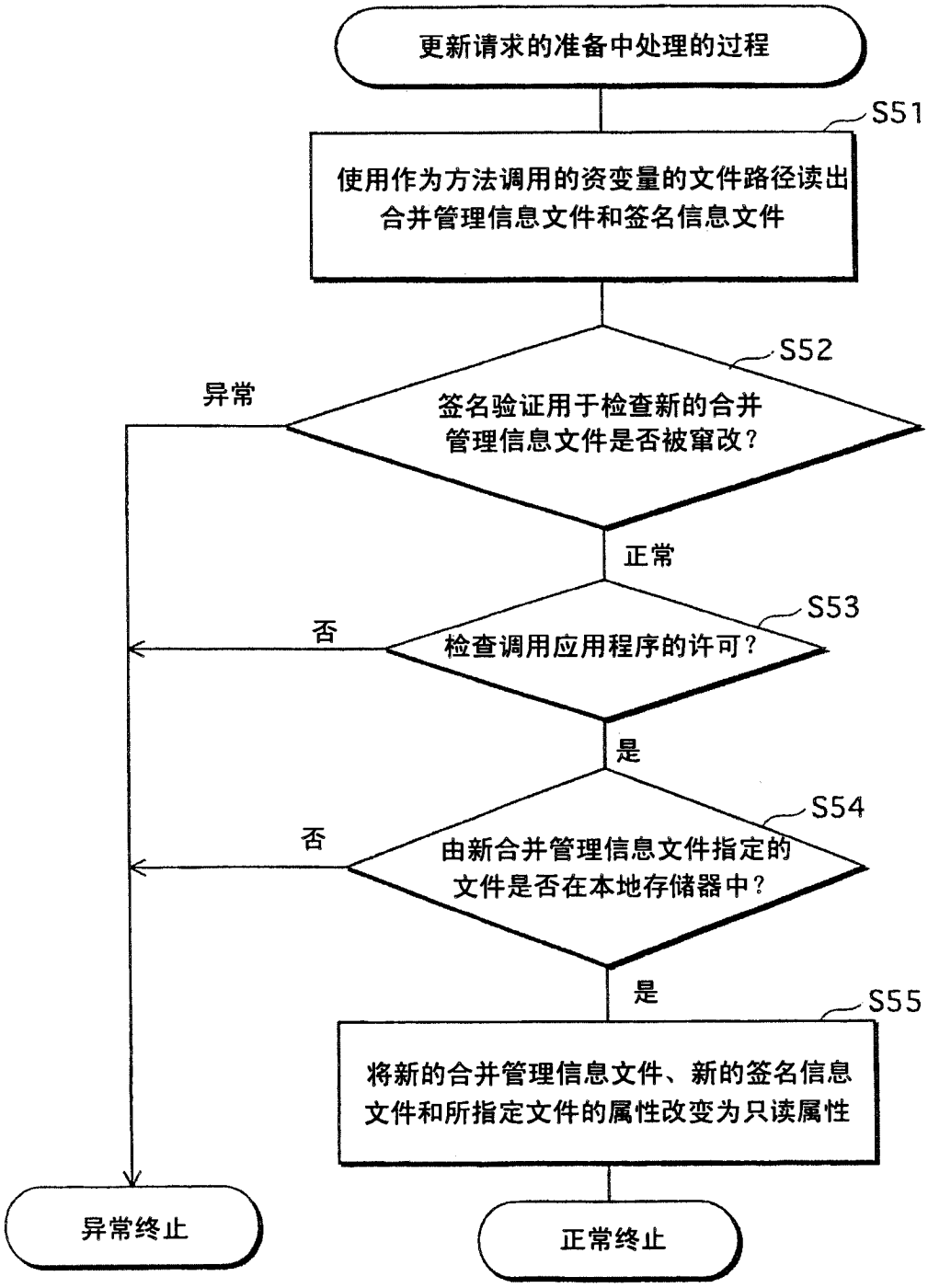
更新请求的准备中处理的过程
S51
使用作为方法调用的资变量的文件路径读出合并管理信息文件和签名信息文件
S52
签名验证用于检查新的合并管理信息文件是否被窜改？
异常
正常
S53
检查调用应用程序的许可？
否
是
S54
由新合并管理信息文件指定的文件是否在本地存储器中？
否
是
S55
将新的合并管理信息文件、新的签名信息文件和所指定文件的属性改变为只读属性
异常终止
正常终止

图 32

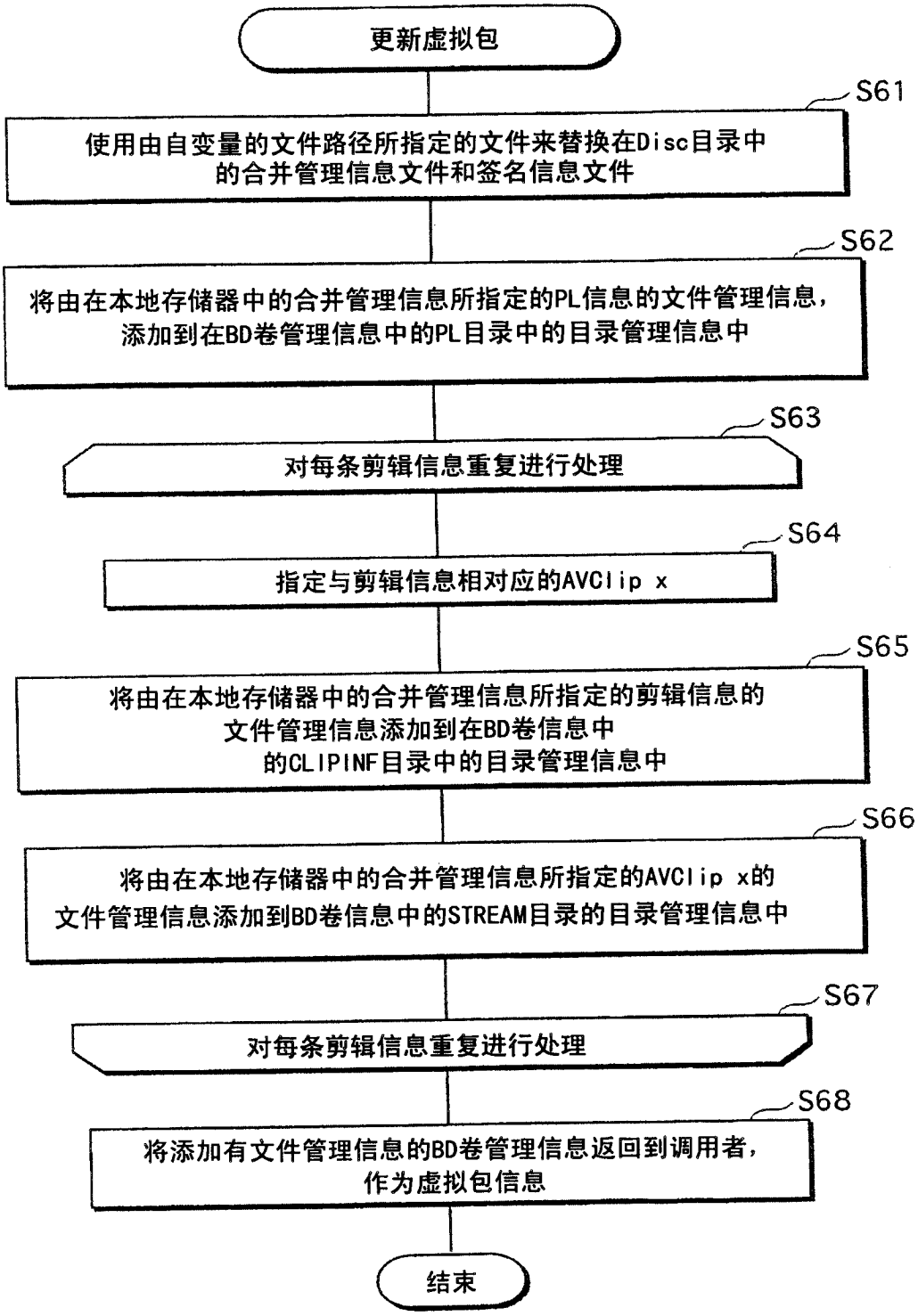
更新虚拟包
S61
使用由自变量的文件路径所指定的文件来替换在Disc目录中的合并管理信息文件和签名信息文件
S62
将由在本地存储器中的合并管理信息所指定的PL信息的文件管理信息，添加到在BD卷管理信息中的PL目录中的目录管理信息中
S63
对每条剪辑信息重复进行处理
S64
指定与剪辑信息相对应的AVClip x
S65
将由在本地存储器中的合并管理信息所指定的剪辑信息的文件管理信息添加到在BD卷信息中的CLIPINF目录中的目录管理信息中
S66
将由在本地存储器中的合并管理信息所指定的AVClip x的文件管理信息添加到BD卷信息中的STREAM目录的目录管理信息中
S67
对每条剪辑信息重复进行处理
S68
将添加有文件管理信息的BD卷管理信息返回到调用者，作为虚拟包信息
结束

图 33

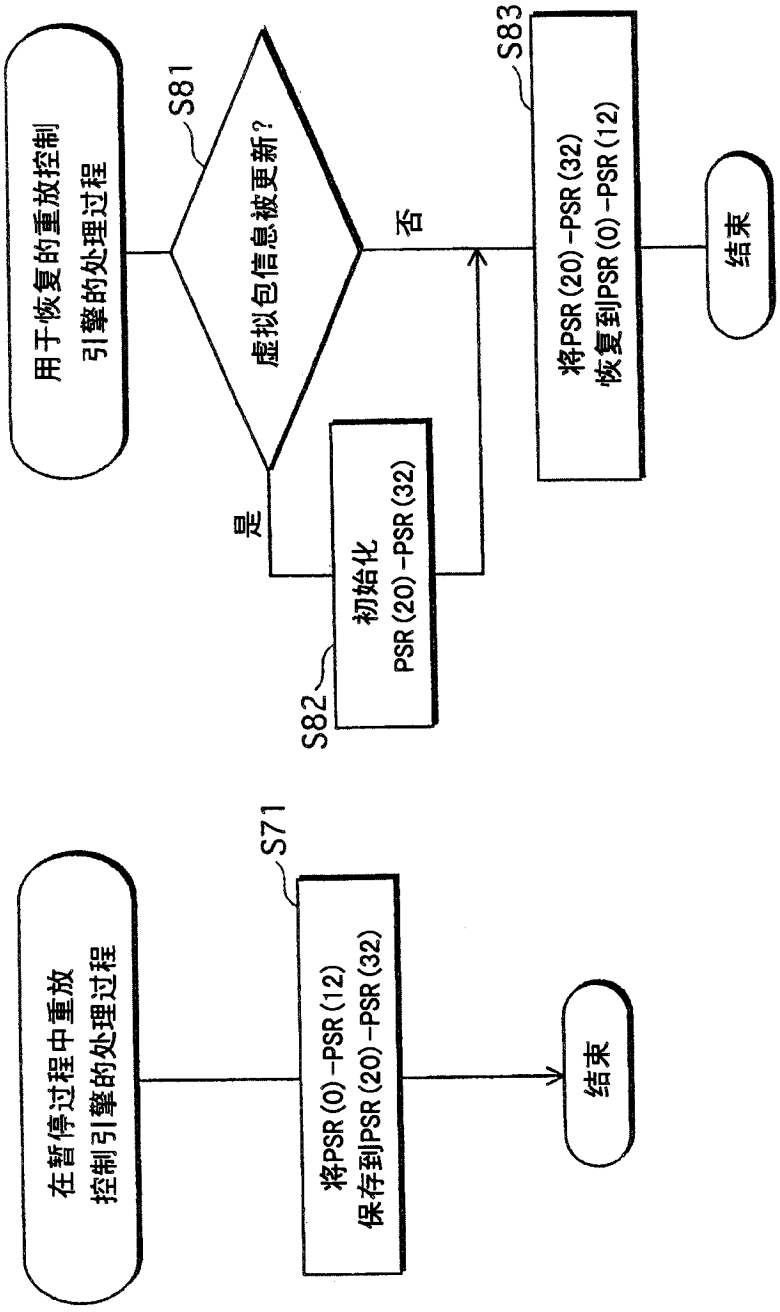

图 34

光盘ID	合并点
disc#1	ROOT/organization#1/disc#1/conts ID#1
⋮	⋮

图 35

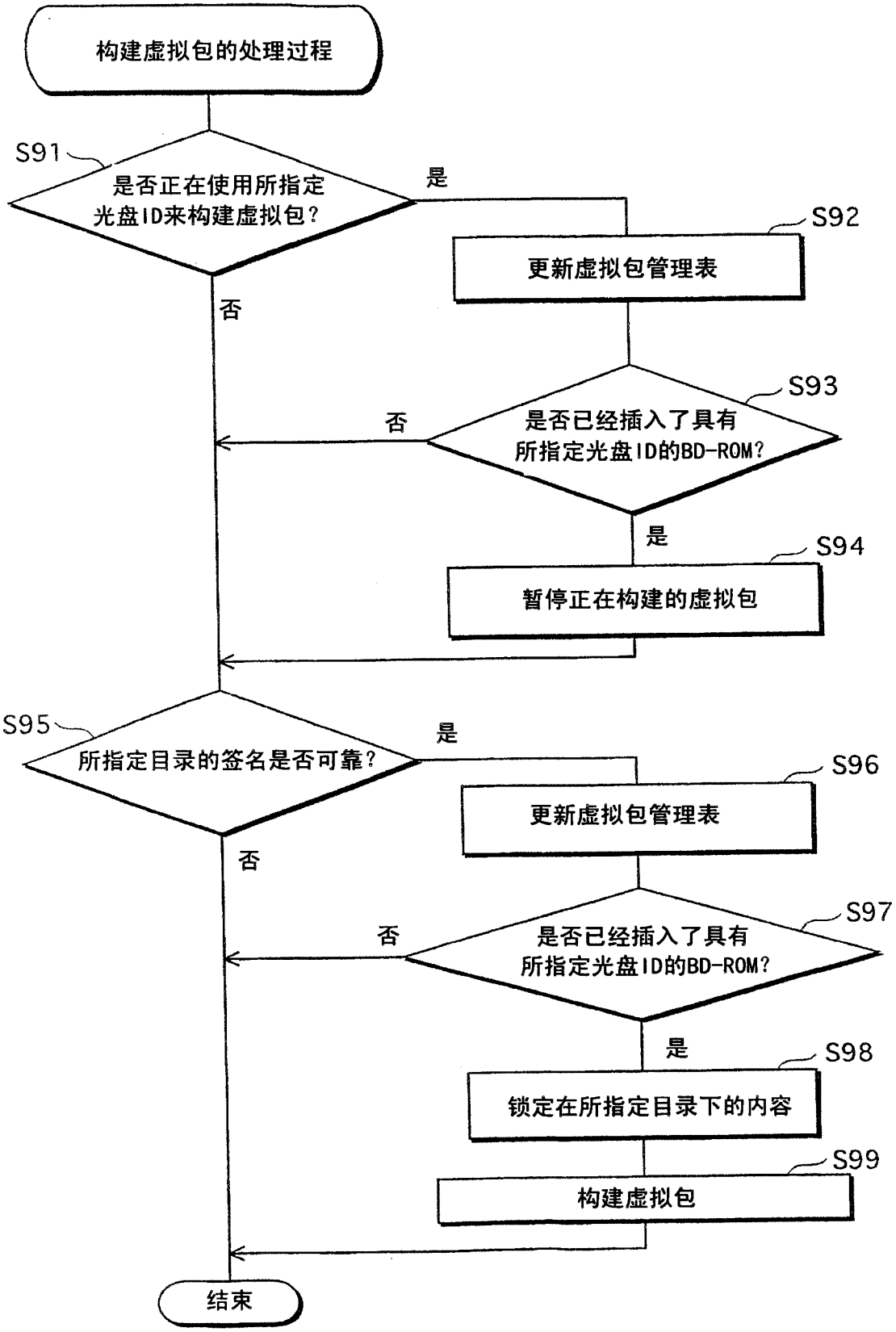
构建虚拟包的处理过程
S91
是否正在使用所指定光盘ID来构建虚拟包？
是
否
S92
更新虚拟包管理表
S93
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S94
暂停正在构建的虚拟包
S95
所指定目录的签名是否可靠？
是
否
S96
更新虚拟包管理表
S97
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S98
锁定在所指定目录下的内容
S99
构建虚拟包
结束

图 36

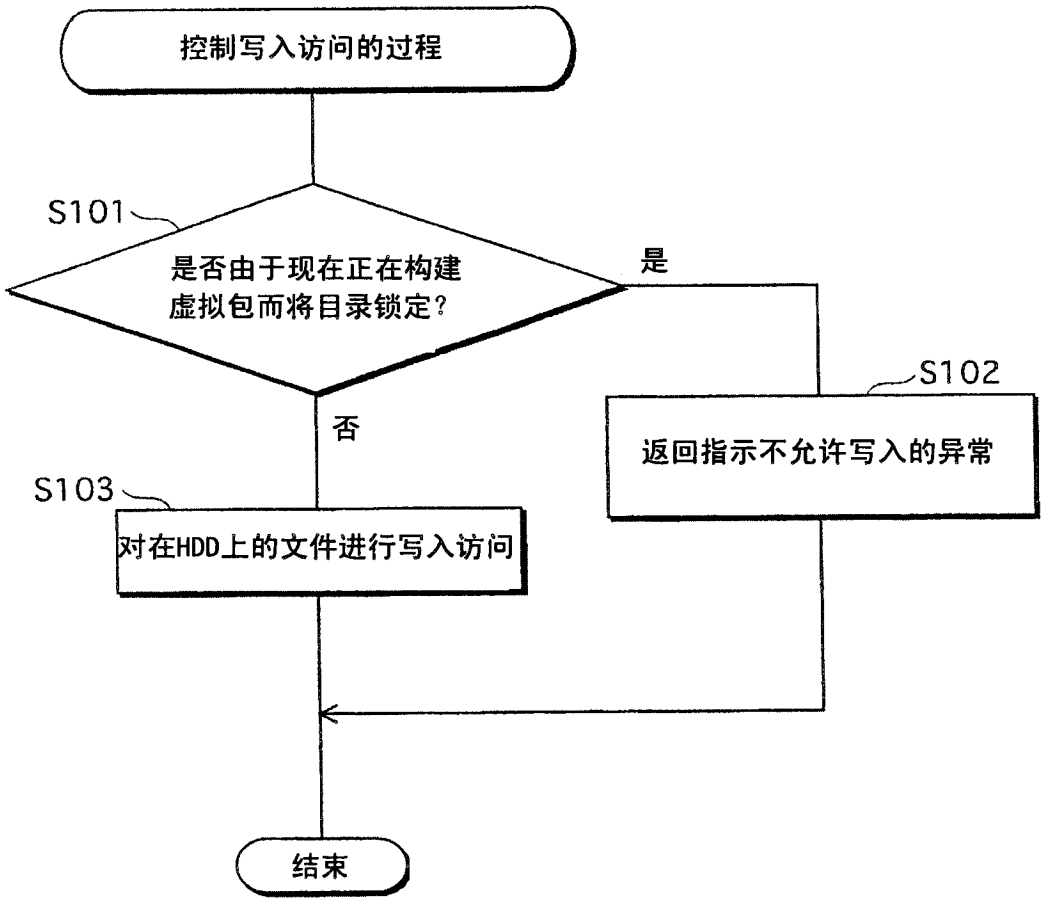
控制写入访问的过程
S101
是否由于现在正在构建
虚拟包而将目录锁定？
是
否
S102
返回指示不允许写入的异常
S103
对在HDD上的文件进行写入访问
结束

图 37

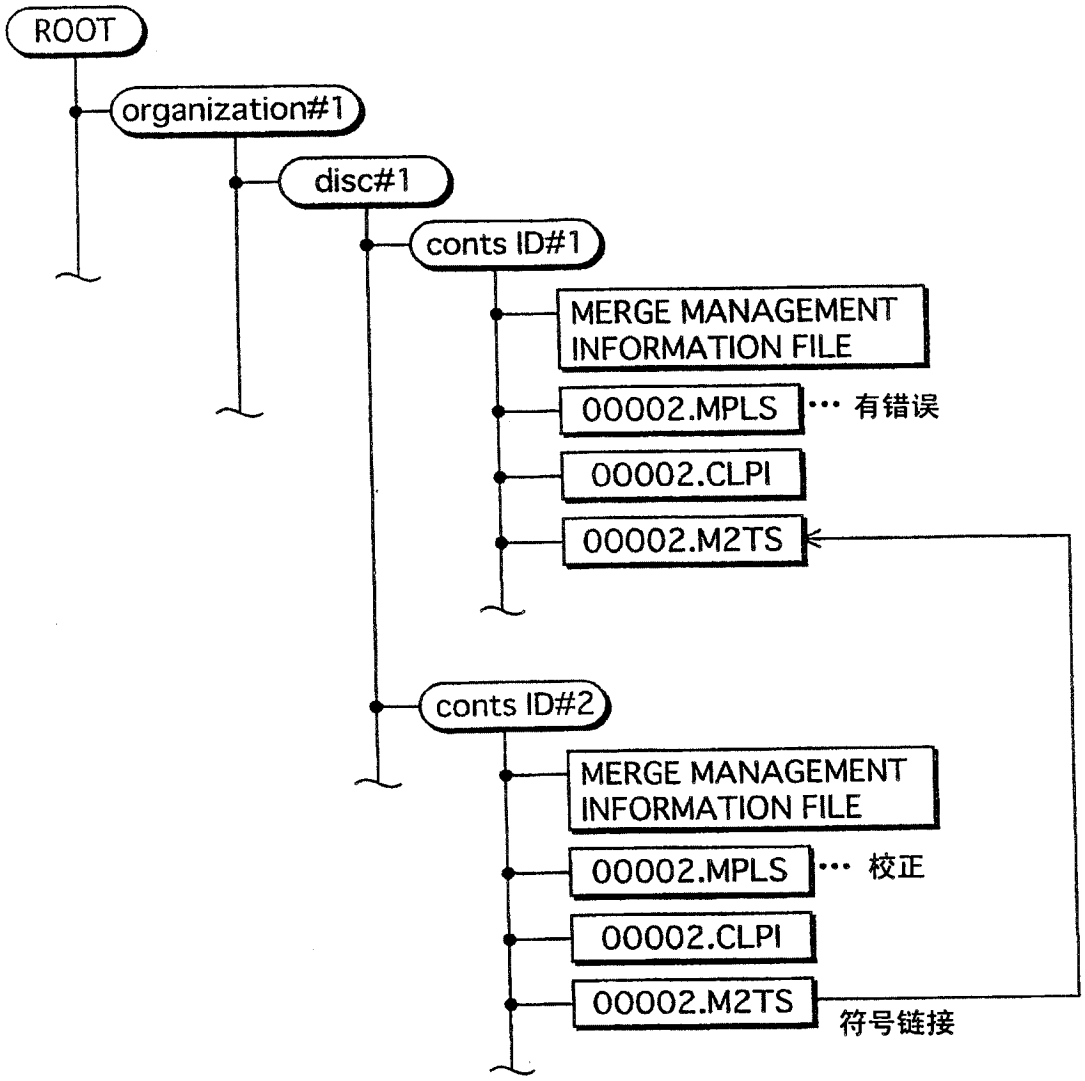
ROOT
organization#1
disc#1
conts ID#1
MERGE MANAGEMENT INFORMATION FILE
00002.MPLS
··· 有错误
00002.CLPI
00002.M2TS
conts ID#2
MERGE MANAGEMENT INFORMATION FILE
00002.MPLS
··· 校正
00002.CLPI
00002.M2TS
符号链接

图 38

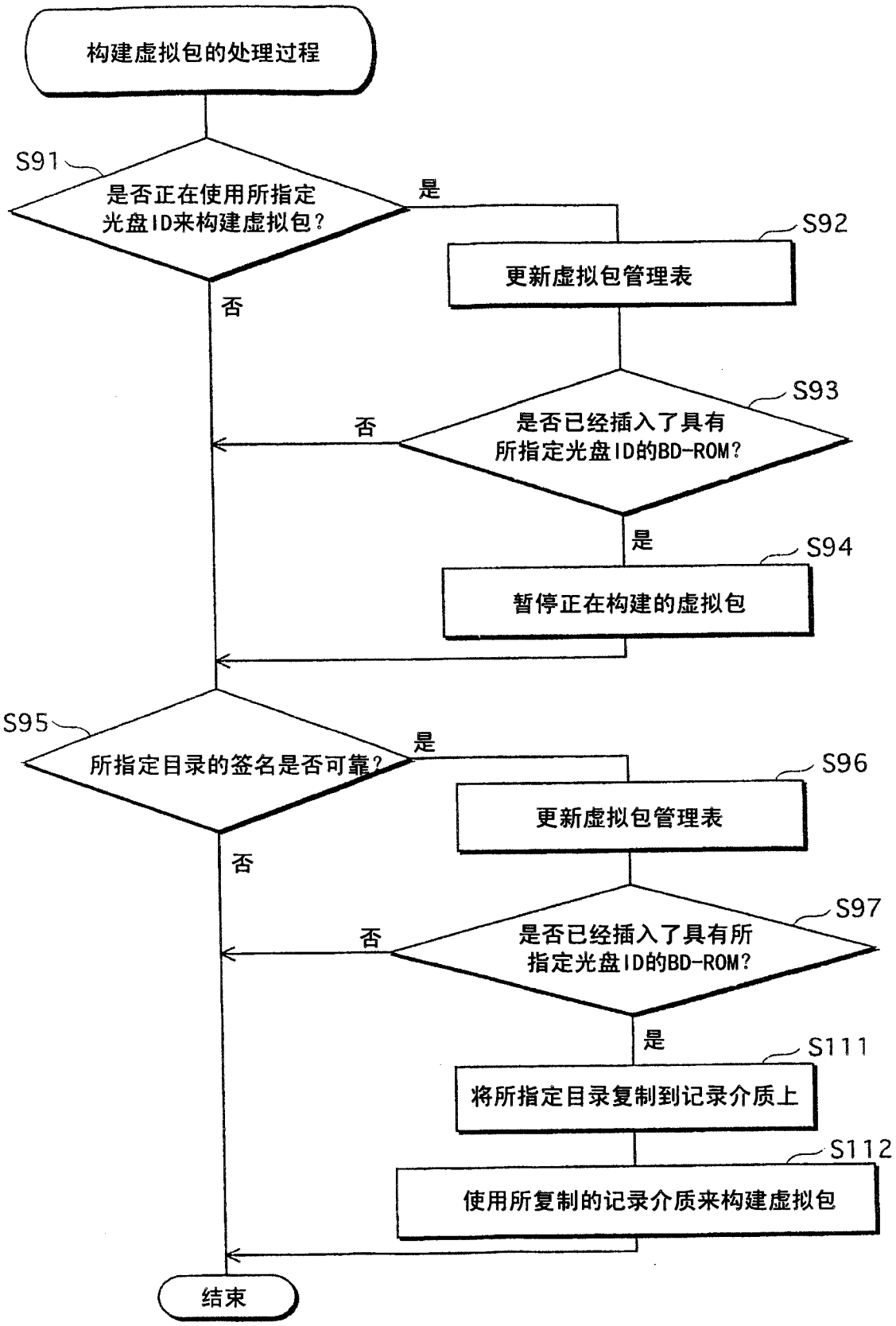
构建虚拟包的处理过程
S91
是否正在使用所指定光盘ID来构建虚拟包？
是
否
S92
更新虚拟包管理表
S93
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S94
暂停正在构建的虚拟包
S95
所指定目录的签名是否可靠？
是
否
S96
更新虚拟包管理表
S97
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S111
将所指定目录复制到记录介质上
S112
使用所复制的记录介质来构建虚拟包
结束

图 39

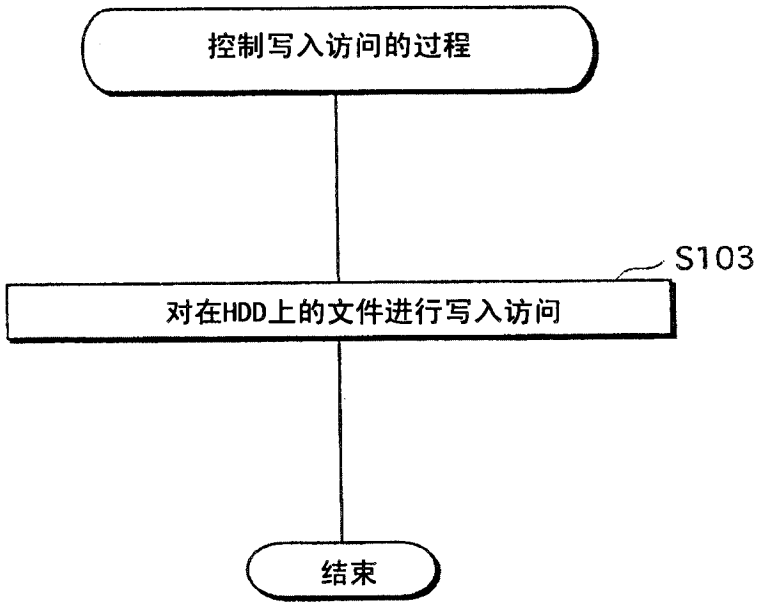

图 40

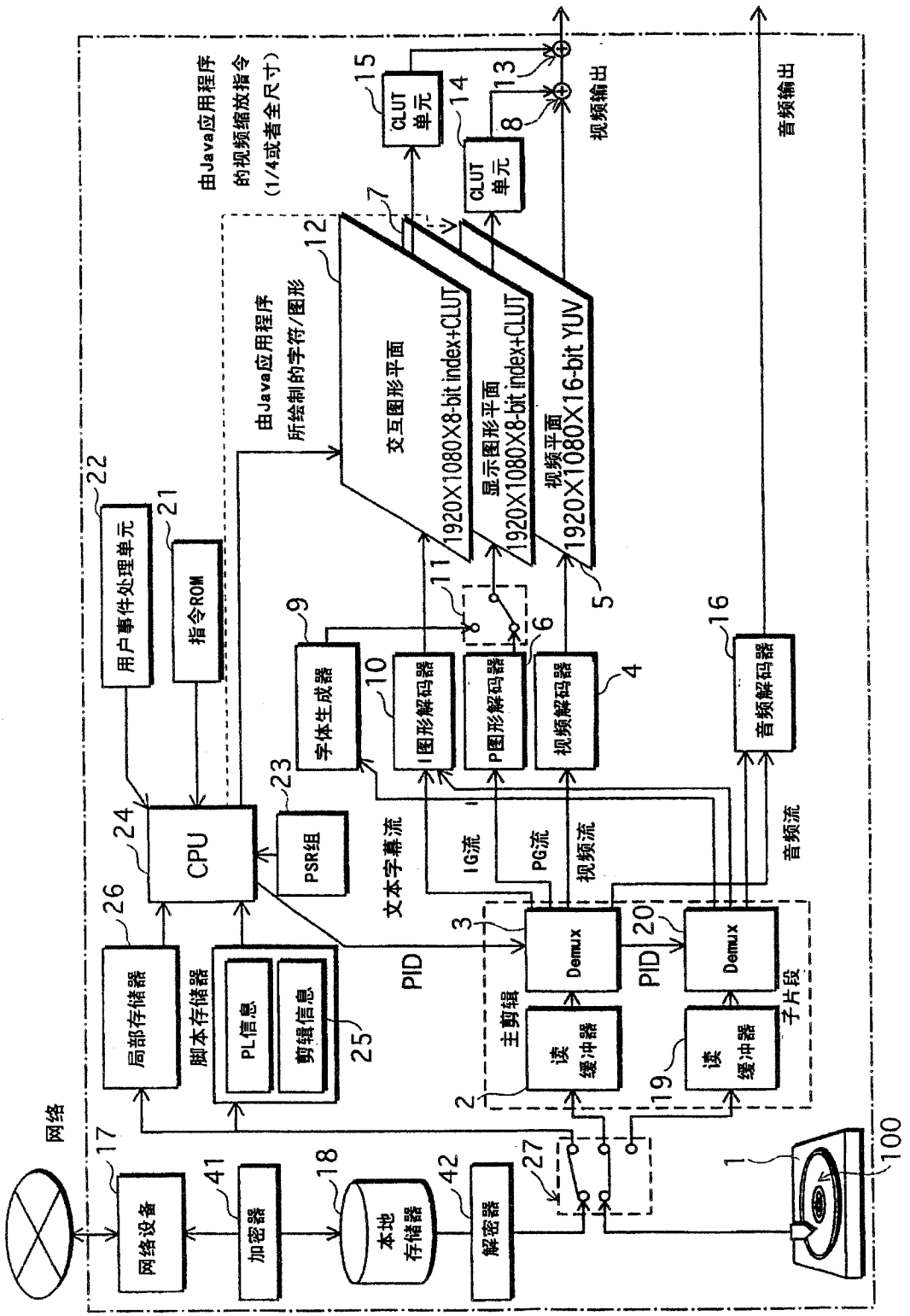
网络
100
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
41
42
网络设备
加密器
本地存储器
解密器
读缓冲器
主剪辑
Demux
PID
子片段
局部存储器
脚本存储器
PL信息
剪辑信息
用户事件处理单元
指令ROM
CPU
PSR组
字体生成器
I图形解码器
P图形解码器
视频解码器
音频解码器
CLUT单元
CLUT单元
由Java应用程序的视频缩放指令(1/4或者全尺寸)
由Java应用程序所绘制的字符/图形
交互图形平面
1920X1080X8-bit index+CLUT
显示图形平面
1920X1080X8-bit index+CLUT
视频平面
1920X1080X16-bit YUV
视频输出
音频输出
文本字幕流
IG流
PG流
视频流
音频流

图 41

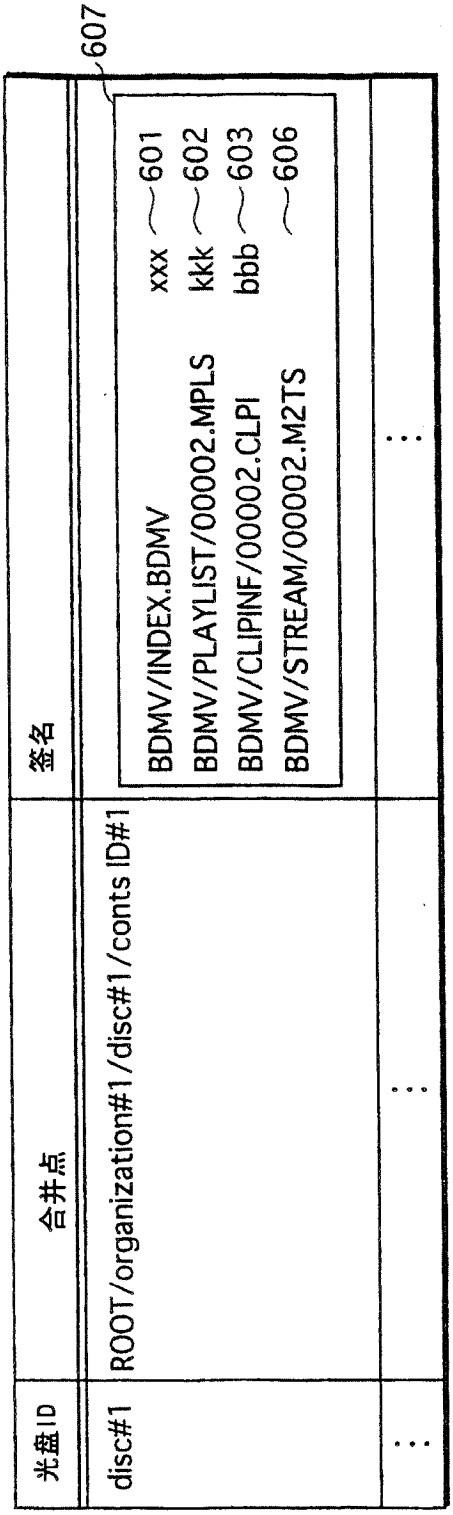
607
光盘ID
合并点
签名
disc#1
ROOT/organization#1/disc#1/conts ID#1
BDMV/INDEX.BDMV xxx 601
BDMV/PLAYLIST/00002.MPLS kkk 602
BDMV/CLIPINF/00002.CLPI bbb 603
BDMV/STREAM/00002.M2TS 606

图 42

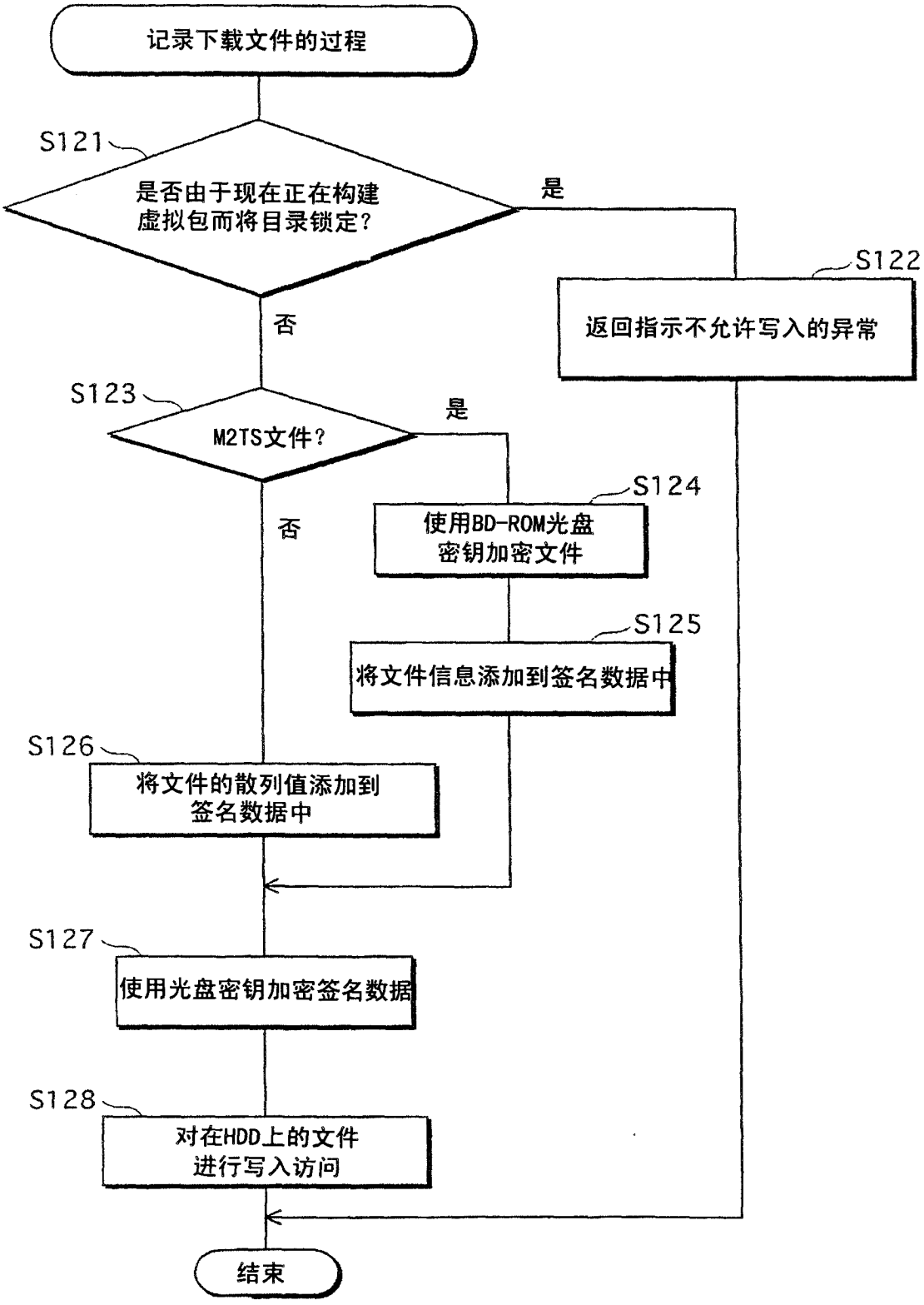
记录下载文件的过程
S121
是否由于现在正在构建虚拟包而将目录锁定?
是
否
S122
返回指示不允许写入的异常
S123
M2TS文件?
是
否
S124
使用BD-ROM光盘密钥加密文件
S125
将文件信息添加到签名数据中
S126
将文件的散列值添加到签名数据中
S127
使用光盘密钥加密签名数据
S128
对在HDD上的文件进行写入访问
结束

图 43

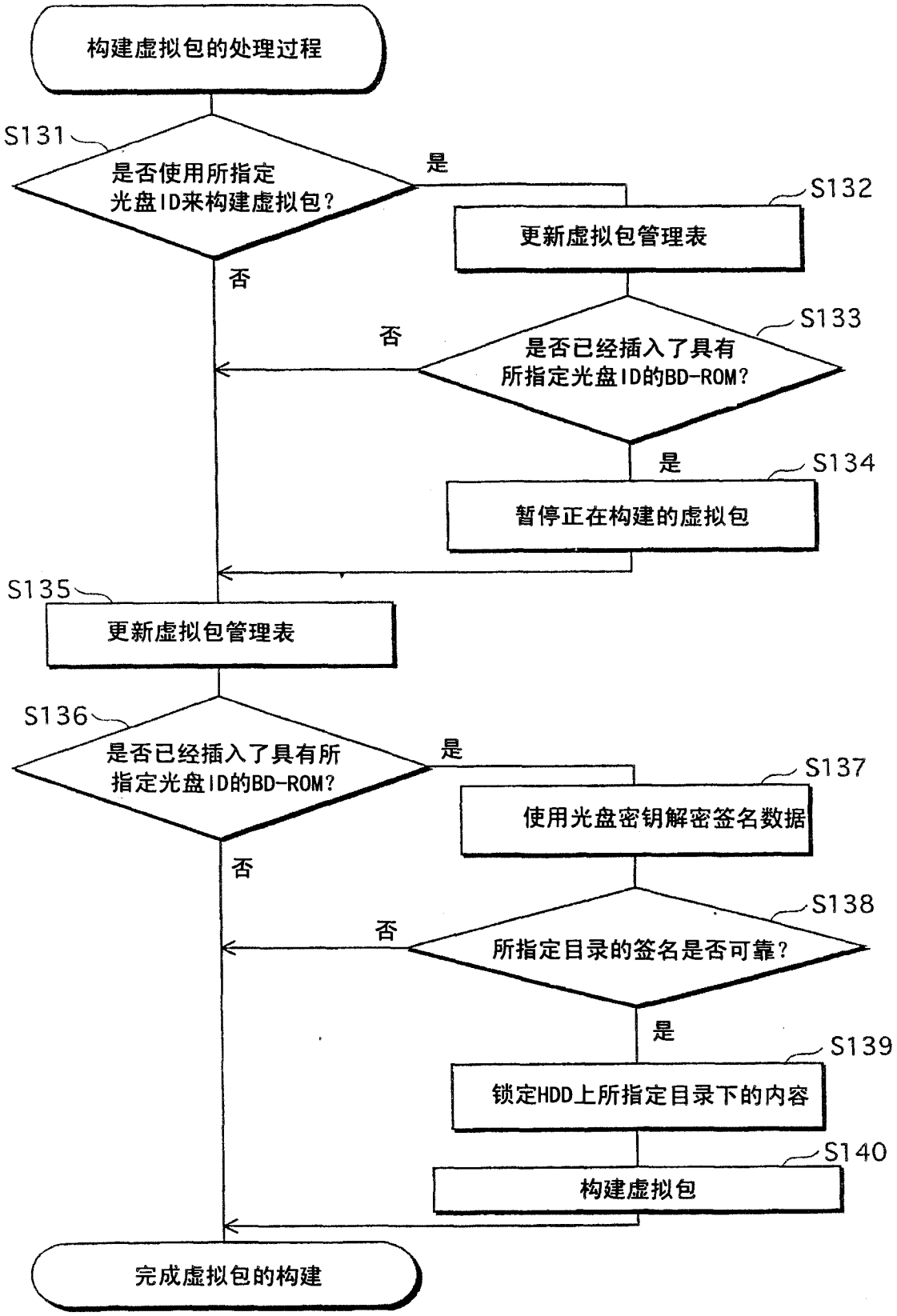
构建虚拟包的处理过程
S131
是否使用所指定光盘ID来构建虚拟包?
是
S132
更新虚拟包管理表
否
S133
否
是否已经插入了具有所指定光盘ID的BD-ROM?
是
S134
暂停正在构建的虚拟包
S135
更新虚拟包管理表
S136
是否已经插入了具有所指定光盘ID的BD-ROM?
是
S137
使用光盘密钥解密签名数据
否
S138
否
所指定目录的签名是否可靠?
是
S139
锁定HDD上所指定目录下的内容
S140
构建虚拟包
完成虚拟包的构建

图 44

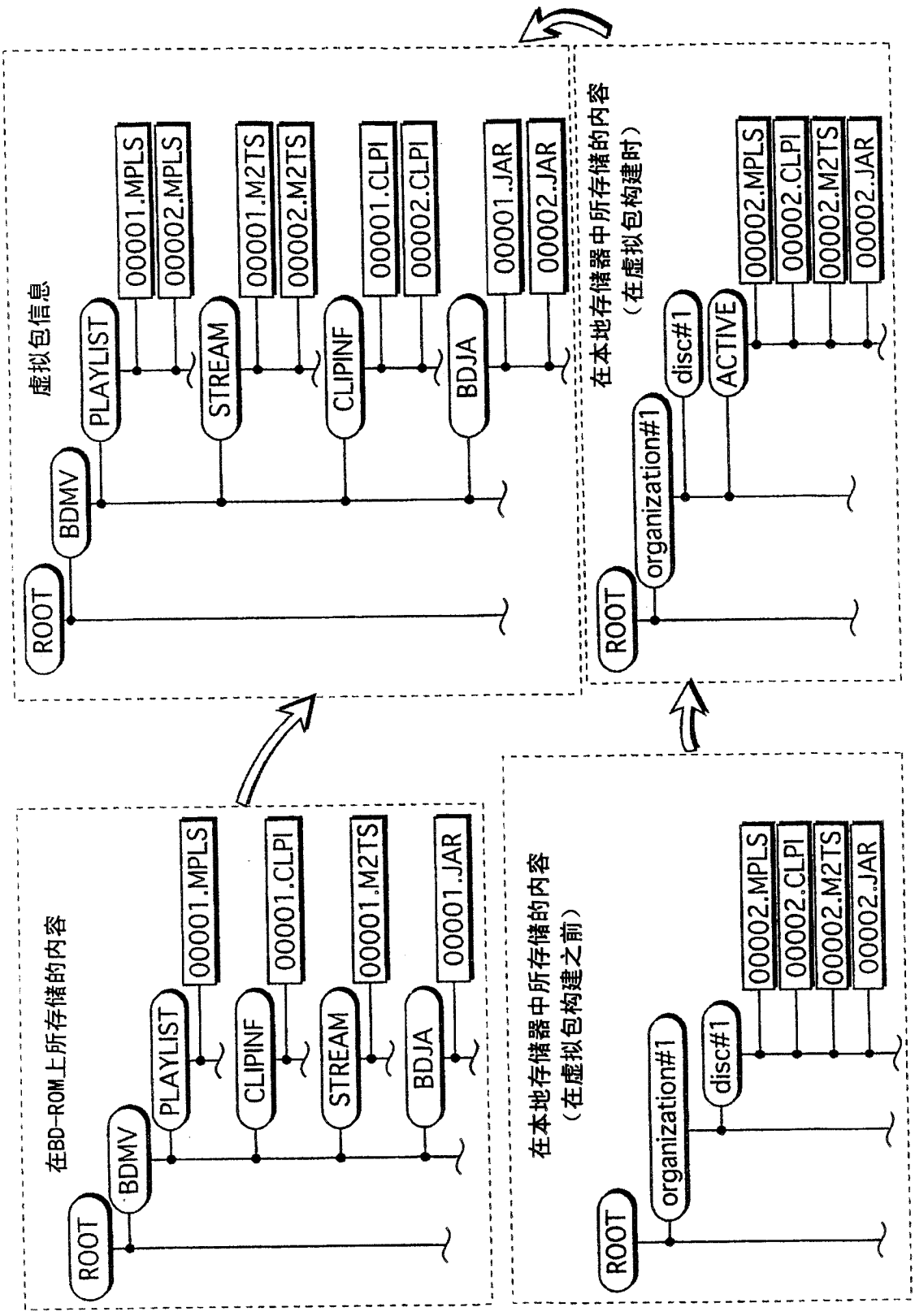
虚拟包信息
ROOT
BDMV
PLAYLIST
00001.MPLS
00002.MPLS
STREAM
00001.M2TS
00002.M2TS
CLIPINF
00001.CLPI
00002.CLPI
BDJA
00001.JAR
00002.JAR
在本地存储器中所存储的内容
（在虚拟包构建时）
ROOT
organization#1
disc#1
ACTIVE
00002.MPLS
00002.CLPI
00002.M2TS
00002.JAR
在BD-ROM上所存储的内容
ROOT
BDMV
PLAYLIST
00001.MPLS
CLIPINF
00001.CLPI
STREAM
00001.M2TS
BDJA
00001.JAR
在本地存储器中所存储的内容
（在虚拟包构建之前）
ROOT
organization#1
disc#1
00002.MPLS
00002.CLPI
00002.M2TS
00002.JAR

图 45

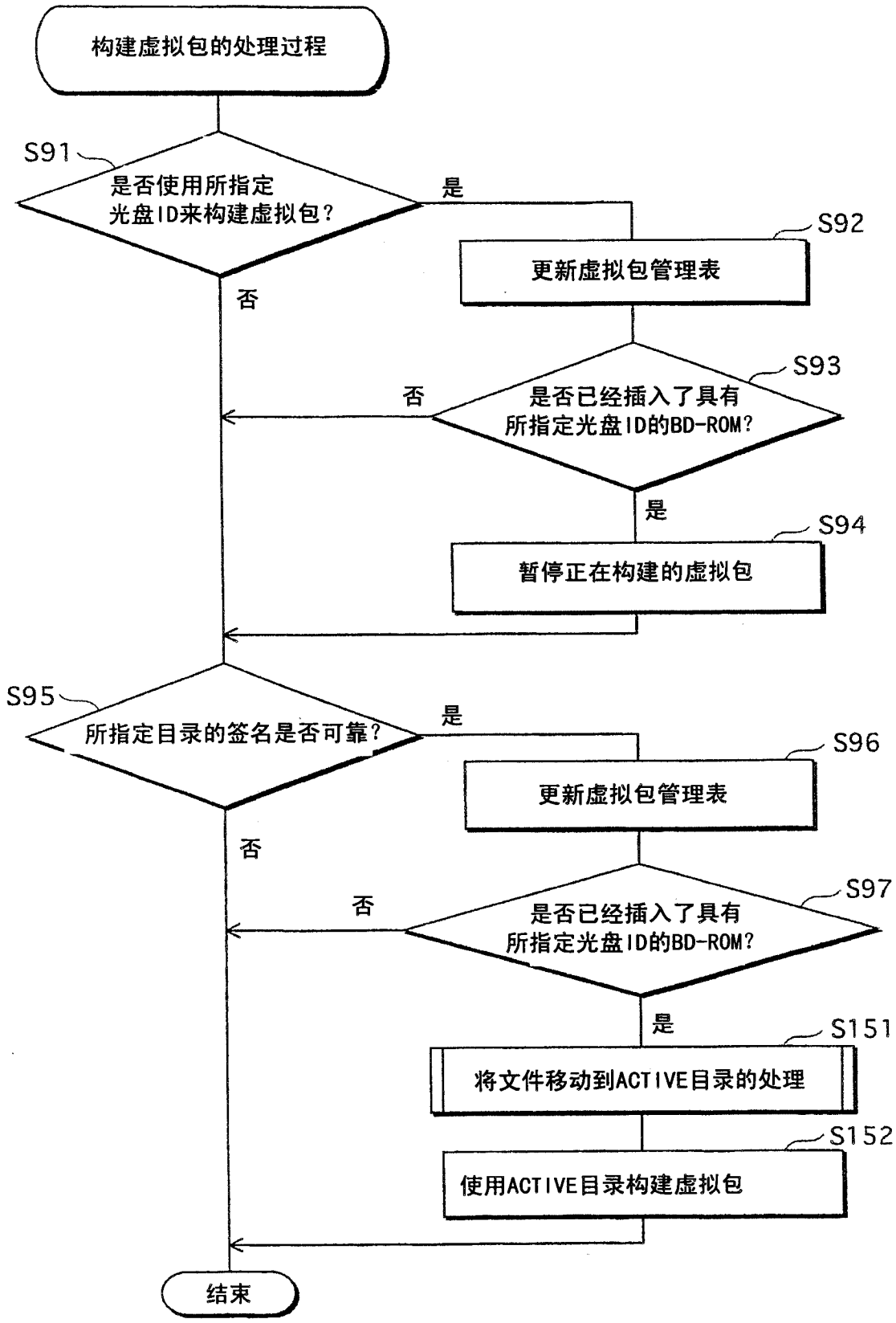
构建虚拟包的处理过程
S91
是否使用所指定
光盘ID来构建虚拟包？
是
S92
更新虚拟包管理表
否
S93
是否已经插入了具有
所指定光盘ID的BD-ROM？
否
是
S94
暂停正在构建的虚拟包
S95
所指定目录的签名是否可靠？
是
S96
更新虚拟包管理表
否
S97
是否已经插入了具有
所指定光盘ID的BD-ROM？
否
是
S151
将文件移动到ACTIVE目录的处理
S152
使用ACTIVE目录构建虚拟包
结束

图 46

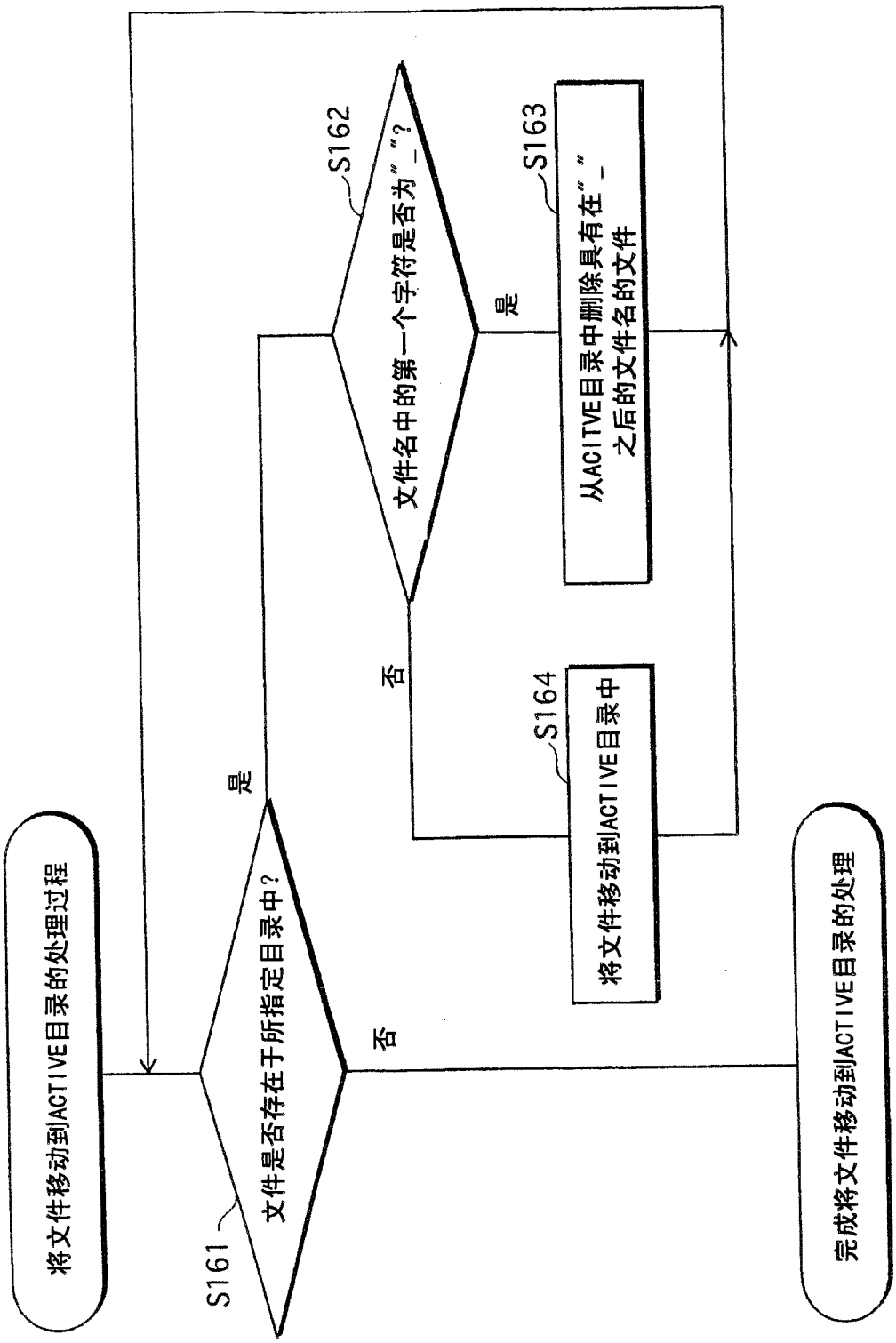
将文件移动到ACTIVE目录的处理过程
S161
文件是否存在于所指定目录中？
是
否
S162
文件名中的第一个字符是否为"_"？
是
否
S163
从ACTIVE目录中删除具有在"_"之后的文件名的文件
S164
将文件移动到ACTIVE目录中
完成将文件移动到ACTIVE目录的处理

图 47

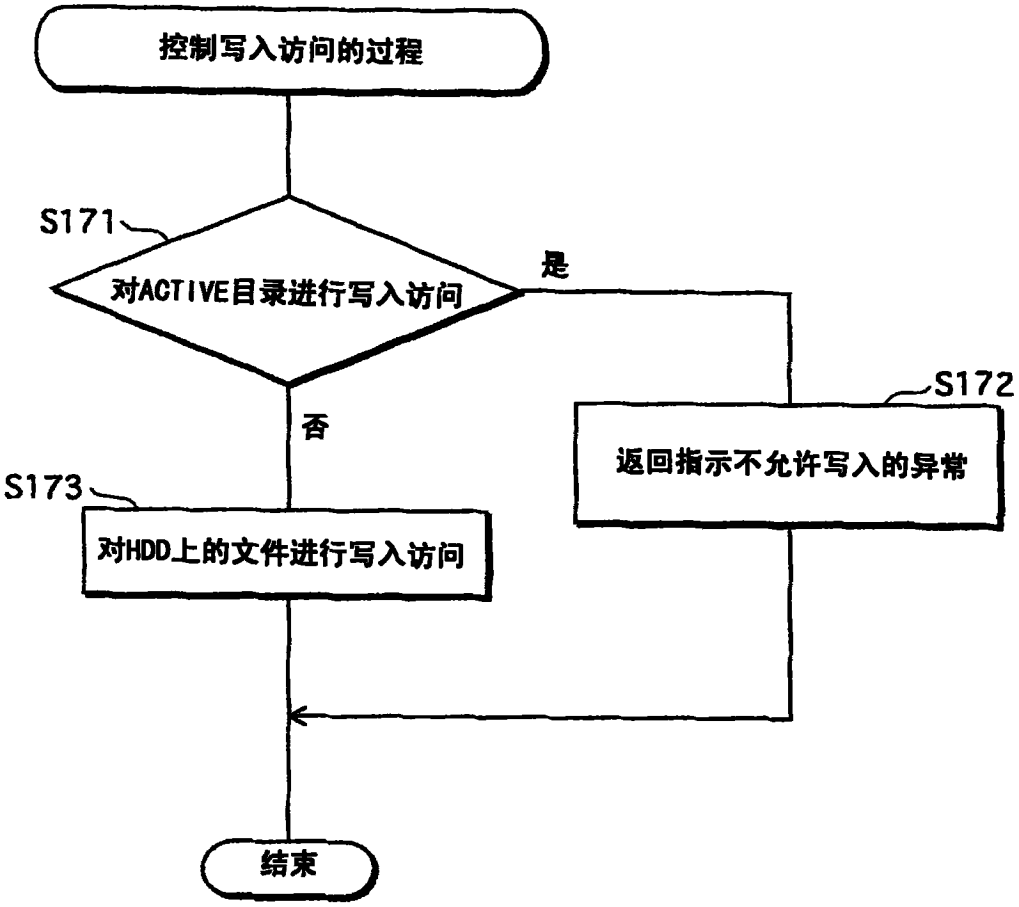
控制写入访问的过程
S171
对ACTIVE目录进行写入访问
是
否
S172
返回指示不允许写入的异常
S173
对HDD上的文件进行写入访问
结束

图 48

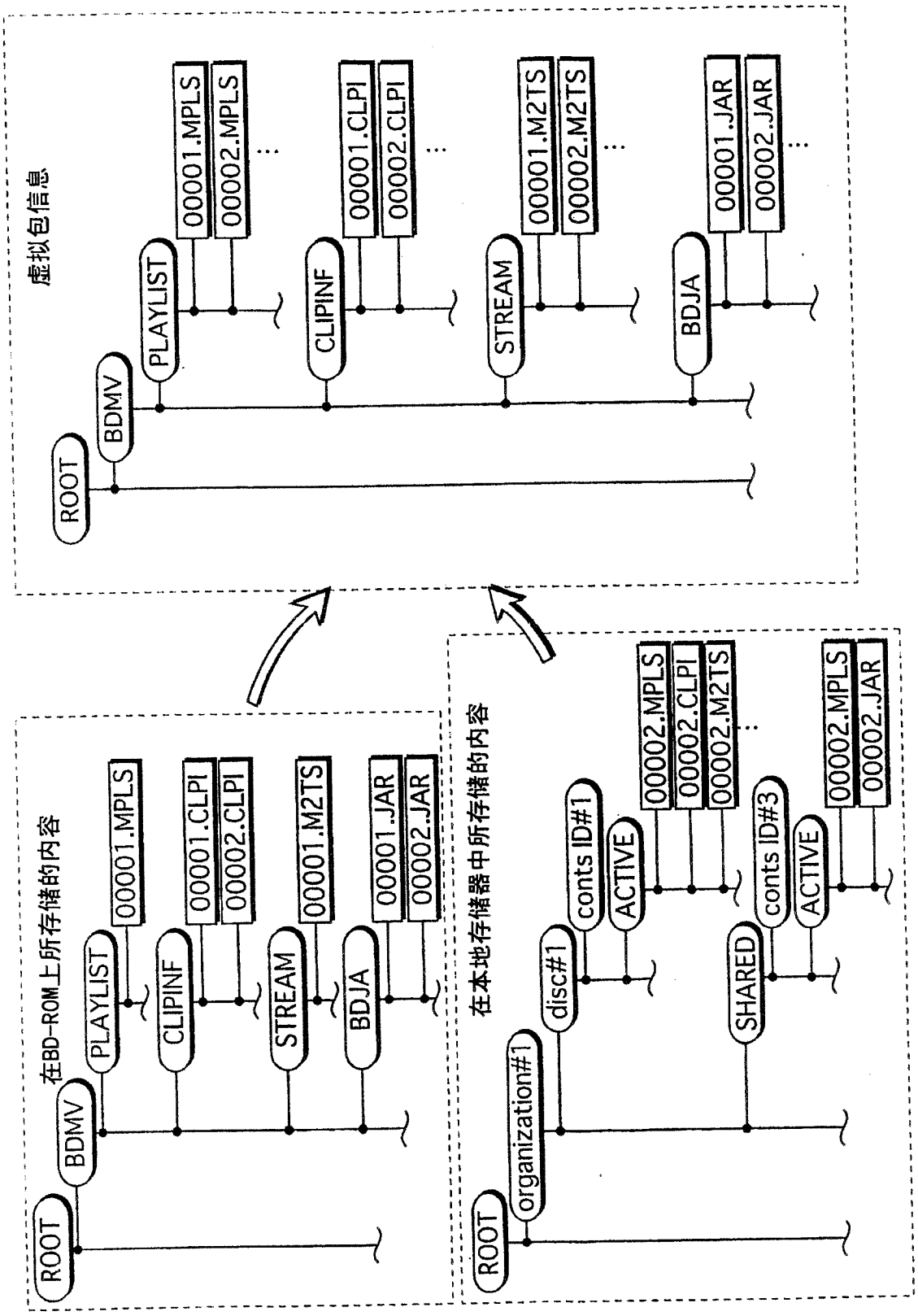
虚拟包信息
ROOT
BDMV
PLAYLIST
00001.MPLS
00002.MPLS
CLIPINF
00001.CLPI
00002.CLPI
STREAM
00001.M2TS
00002.M2TS
BDJA
00001.JAR
00002.JAR
在BD-ROM上所存储的内容
在本地存储器中所存储的内容
organization#1
disc#1
cnts ID#1
ACTIVE
SHARED
cnts ID#3

图 49

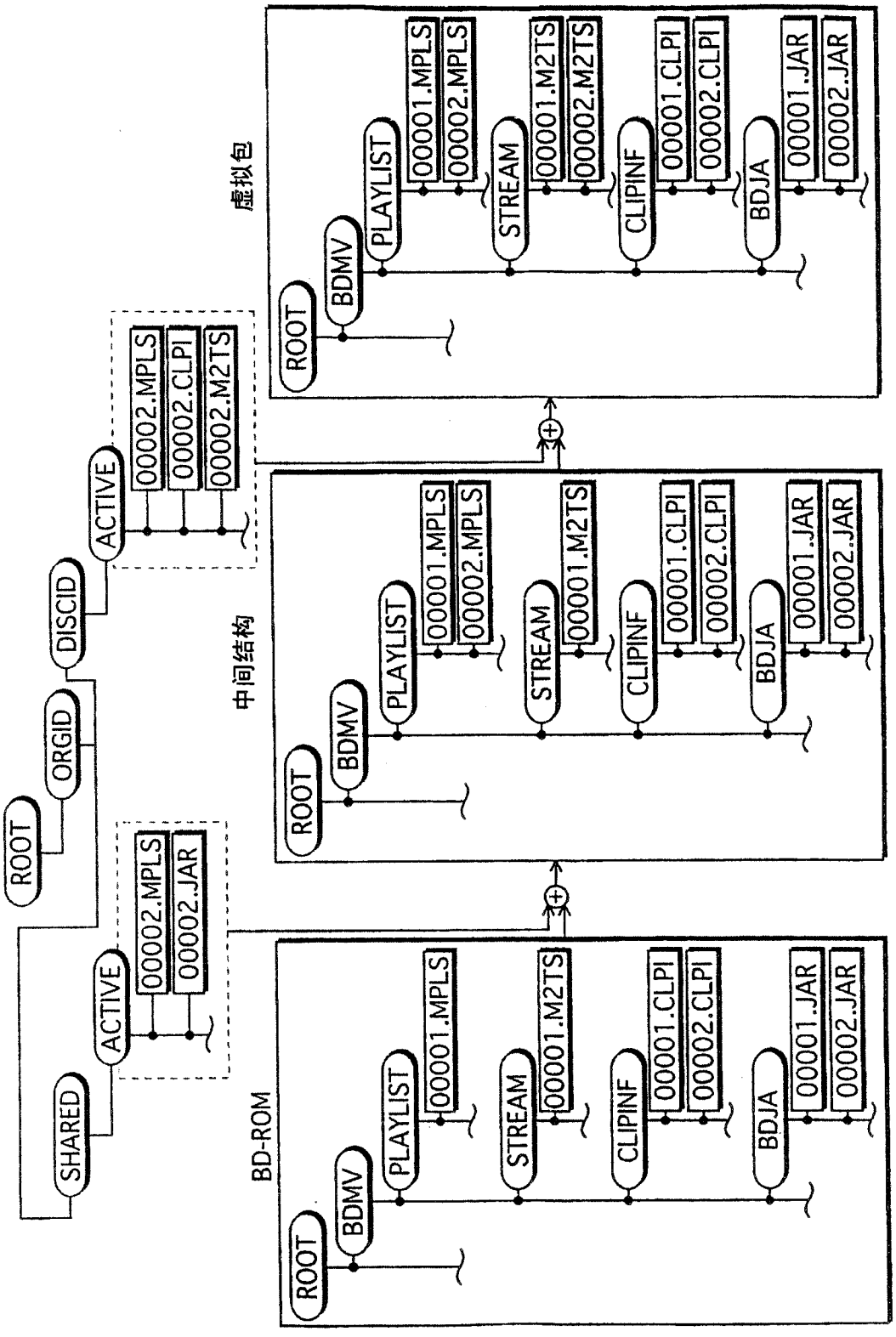
虚拟包
中间结构
BD-ROM
ROOT
BDMV
PLAYLIST
00001.MPLS
00002.MPLS
STREAM
00001.M2TS
00002.M2TS
CLIPINF
00001.CLPI
00002.CLPI
BDJA
00001.JAR
00002.JAR
ROOT
ORGID
DISCID
ACTIVE
00002.MPLS
00002.CLPI
00002.M2TS
SHARED
ACTIVE
00002.MPLS
00002.JAR

图 50

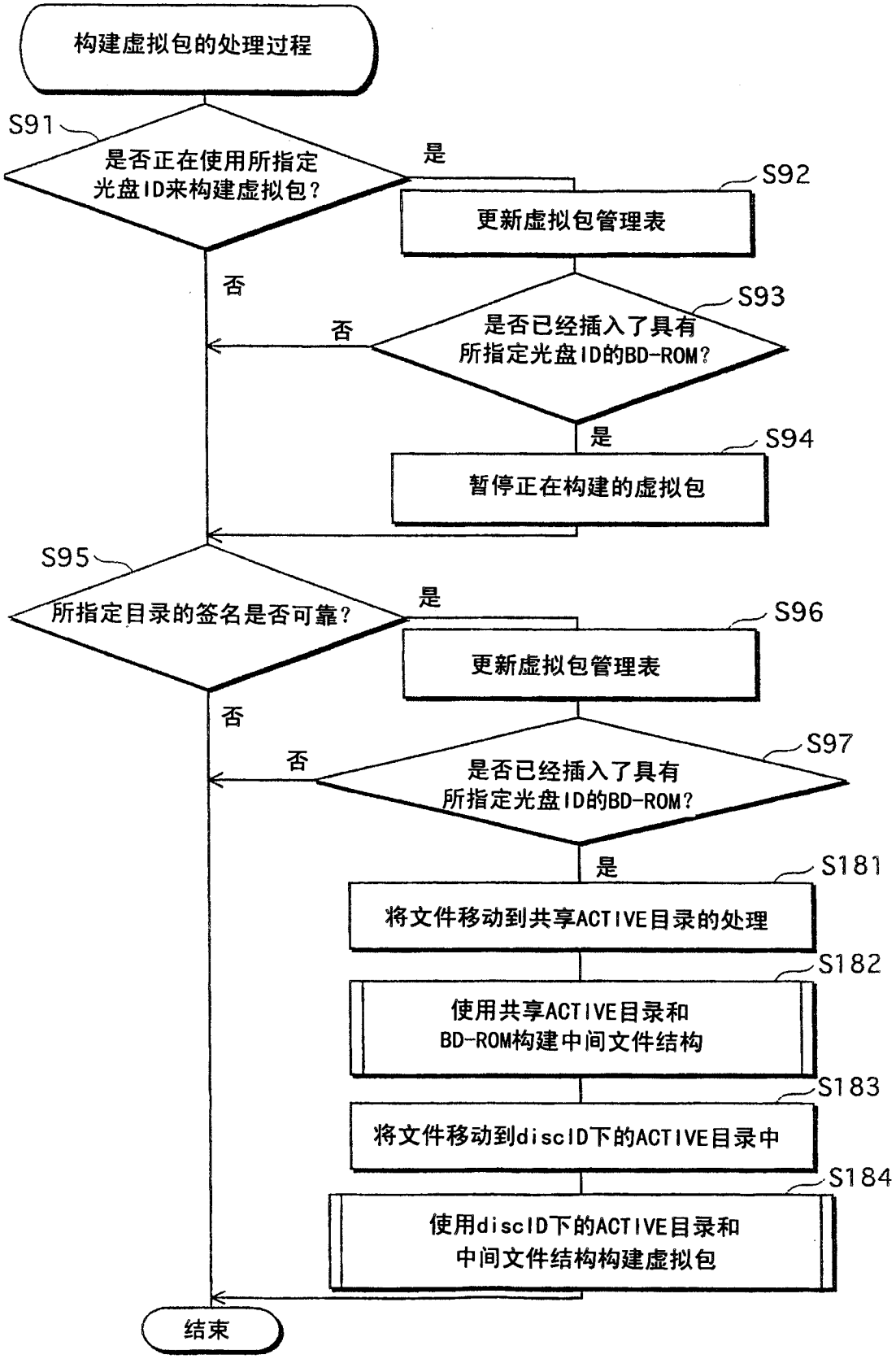
构建虚拟包的处理过程
S91
是否正在使用所指定光盘ID来构建虚拟包？
是
S92
更新虚拟包管理表
否
S93
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S94
暂停正在构建的虚拟包
S95
所指定目录的签名是否可靠？
是
S96
更新虚拟包管理表
否
S97
是否已经插入了具有所指定光盘ID的BD-ROM？
否
是
S181
将文件移动到共享ACTIVE目录的处理
S182
使用共享ACTIVE目录和BD-ROM构建中间文件结构
S183
将文件移动到discID下的ACTIVE目录中
S184
使用discID下的ACTIVE目录和中间文件结构构建虚拟包
结束

图 51

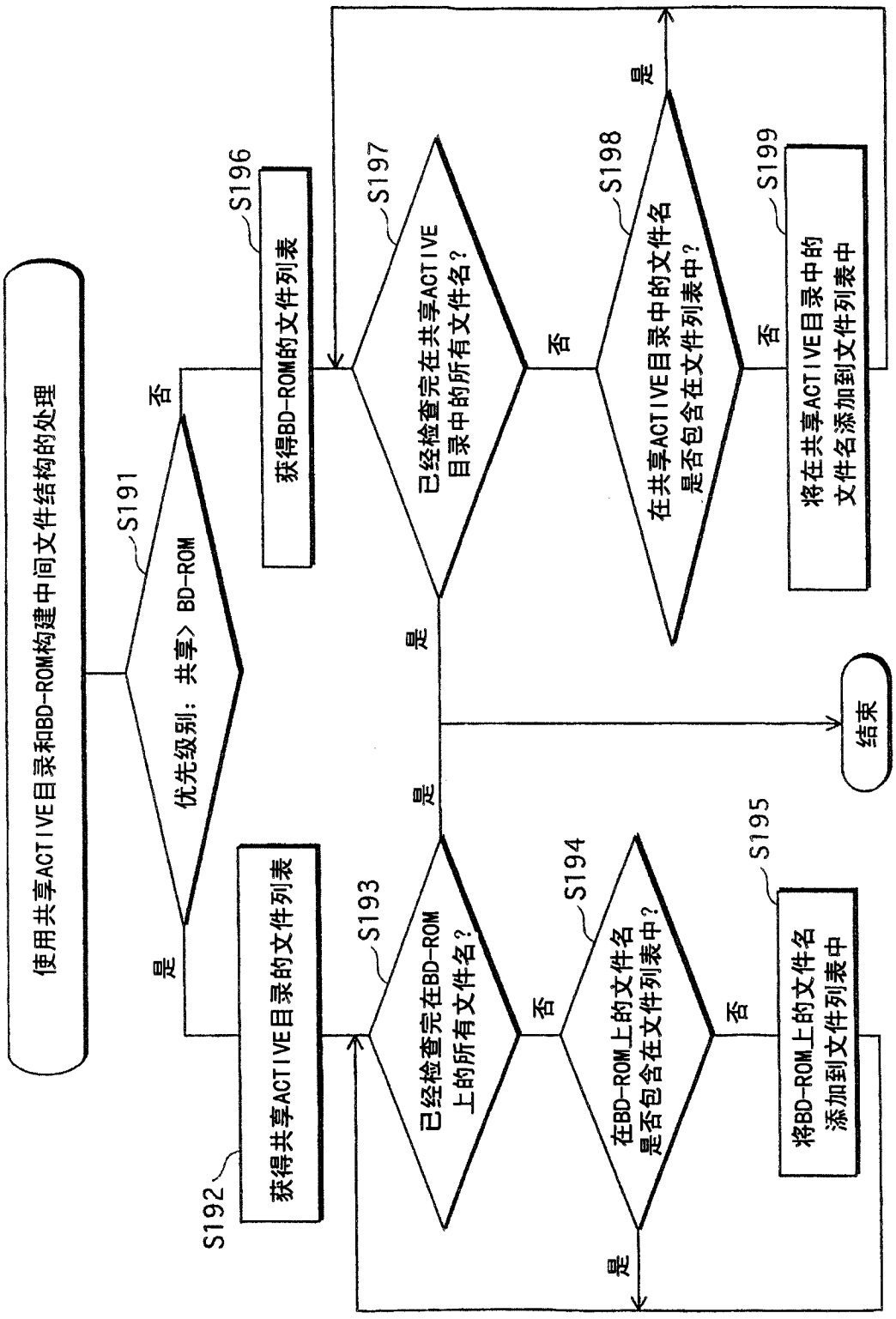
使用共享ACTIVE目录和BD-ROM构建中间文件结构的处理
S191
优先级别：共享> BD-ROM
是
否
S192
获得共享ACTIVE目录的文件列表
S193
已经检查完在BD-ROM上的所有文件名？
是
否
S194
在BD-ROM上的文件名是否包含在文件列表中？
是
否
S195
将BD-ROM上的文件名添加到文件列表中
S196
获得BD-ROM的文件列表
S197
已经检查完在共享ACTIVE目录中的所有文件名？
是
否
S198
在共享ACTIVE目录中的文件名是否包含在文件列表中？
是
否
S199
将在共享ACTIVE目录中的文件名添加到文件列表中
结束

图 52

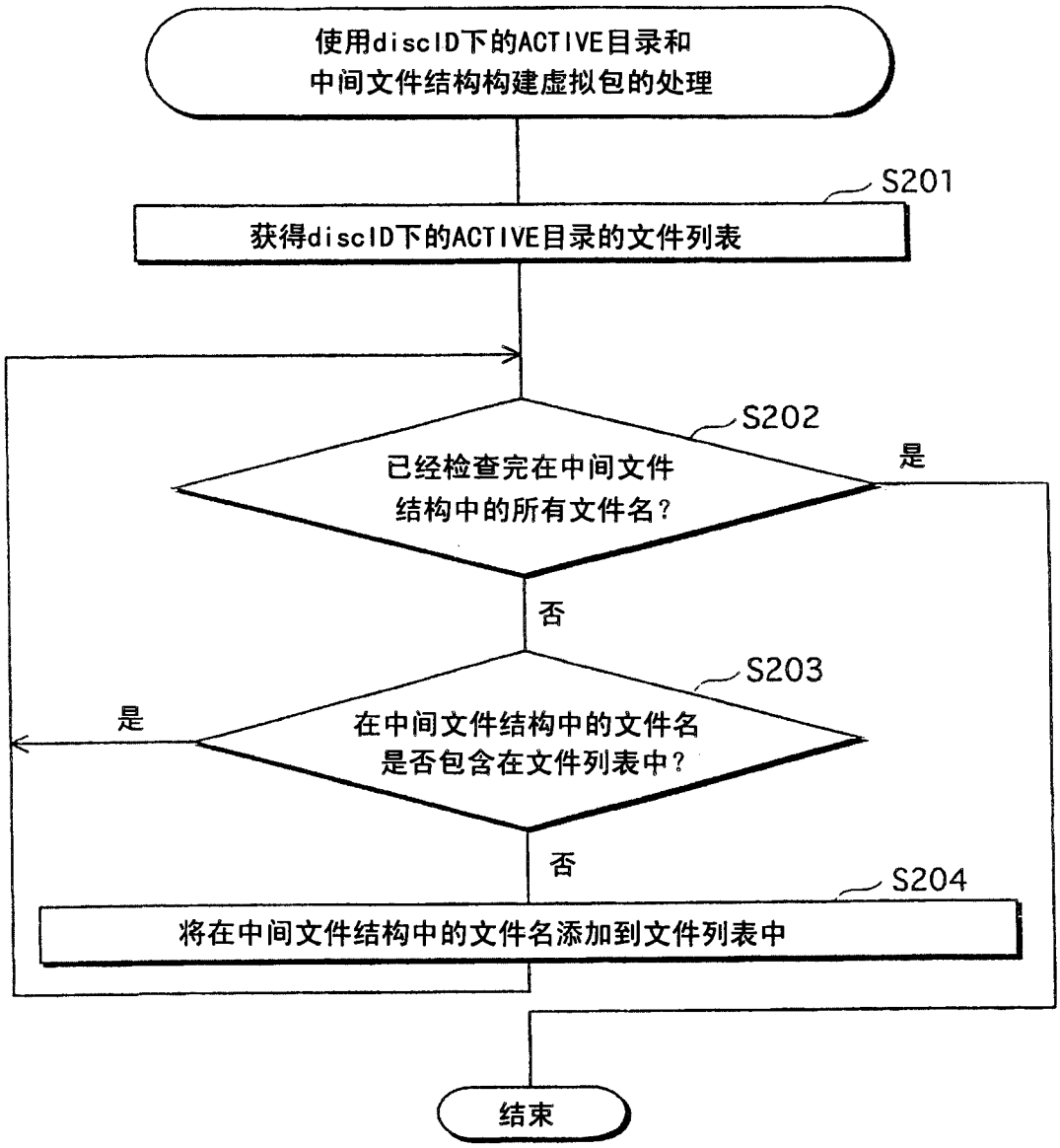
使用discID下的ACTIVE目录和
中间文件结构构建虚拟包的处理
S201
获得discID下的ACTIVE目录的文件列表
S202
已经检查完在中间文件
结构中的所有文件名？
是
否
S203
在中间文件结构中的文件名
是否包含在文件列表中？
是
否
S204
将在中间文件结构中的文件名添加到文件列表中
结束

图 53

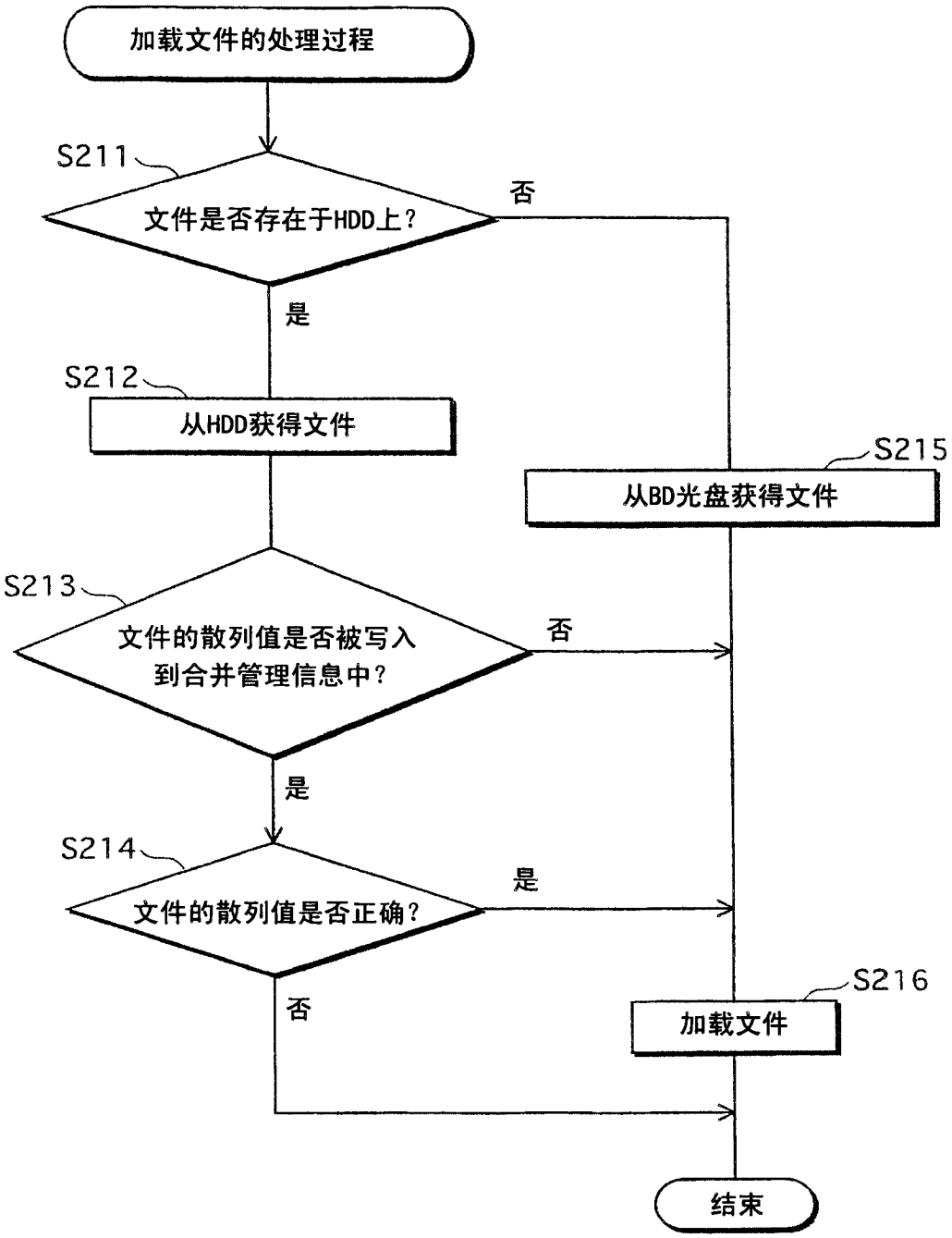
加载文件的处理过程
S211
文件是否存在于HDD上？
否
是
S212
从HDD获得文件
S215
从BD光盘获得文件
S213
文件的散列值是否被写入到合并管理信息中？
否
是
S214
文件的散列值是否正确？
是
否
S216
加载文件
结束

图 54

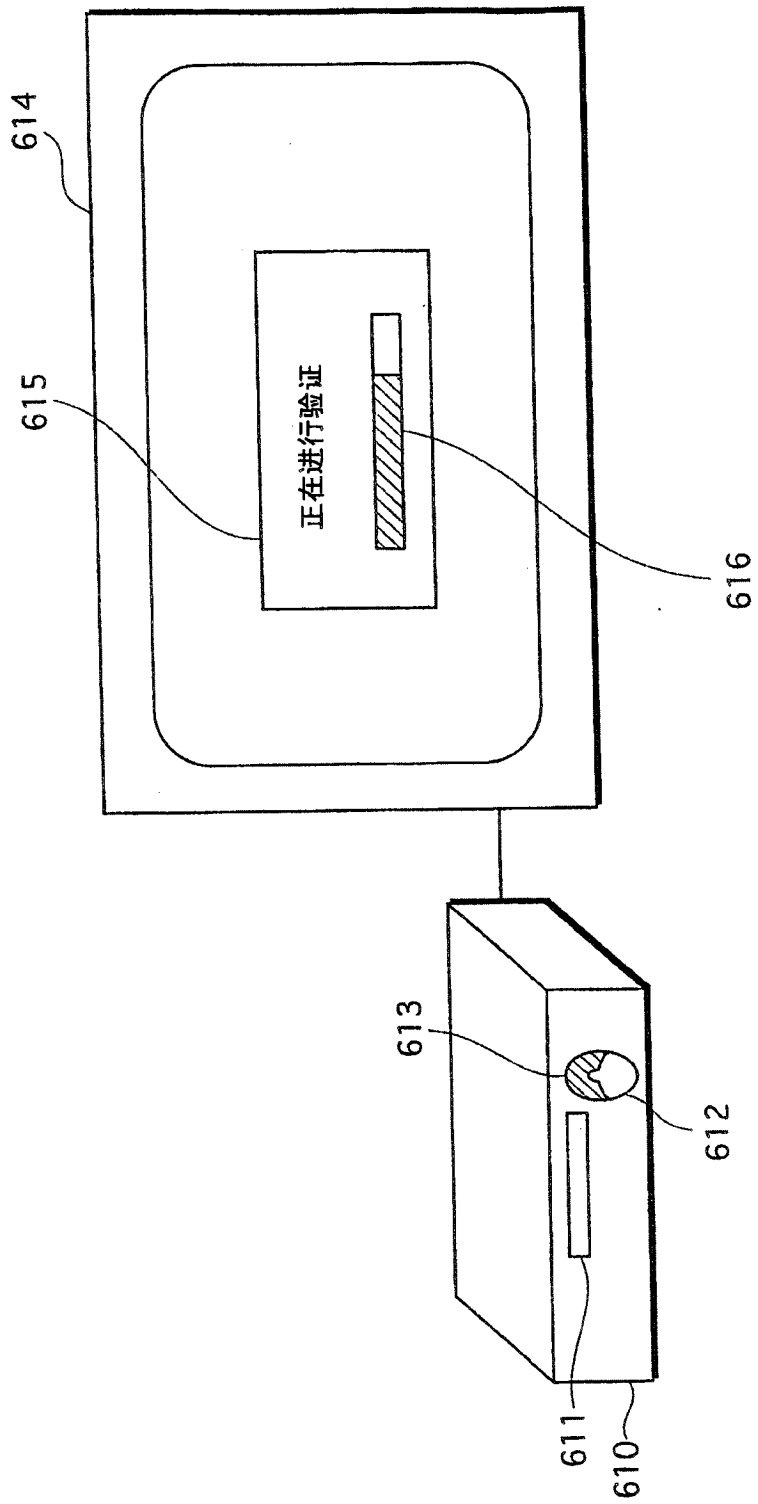

图 55

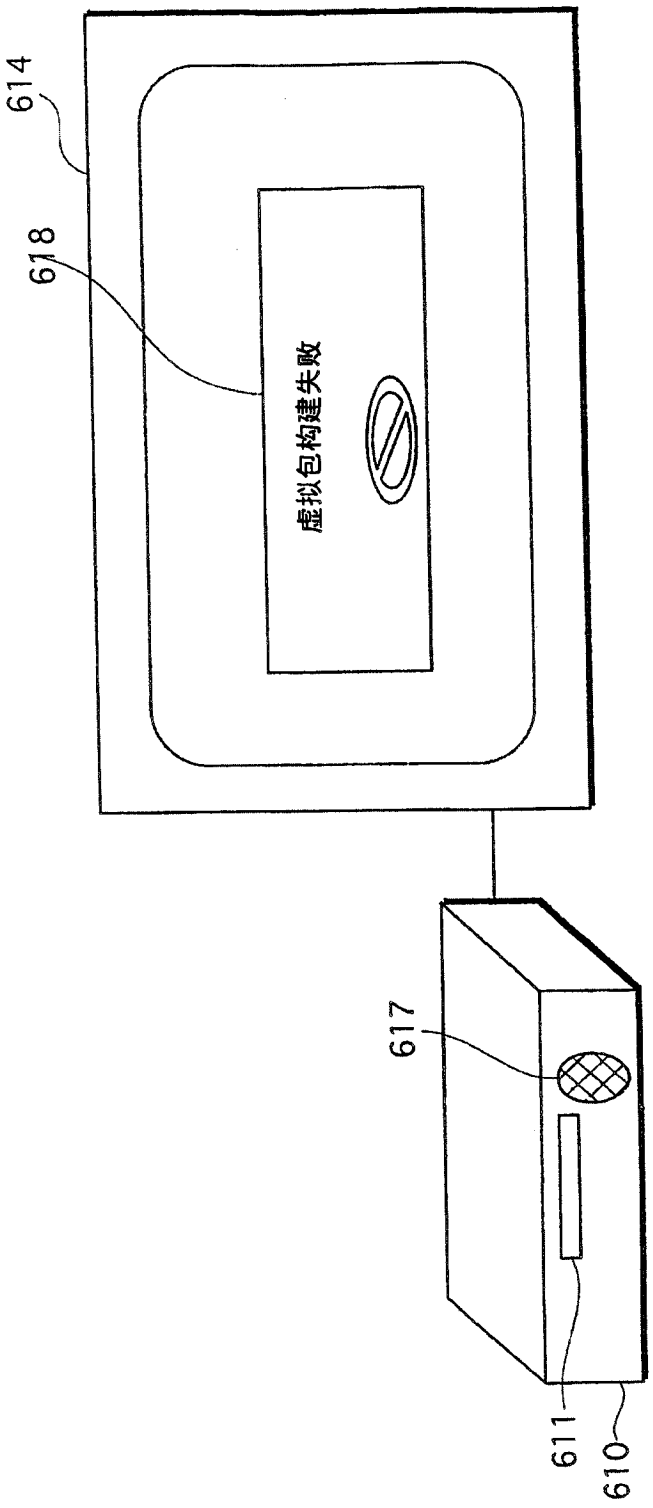

图 56

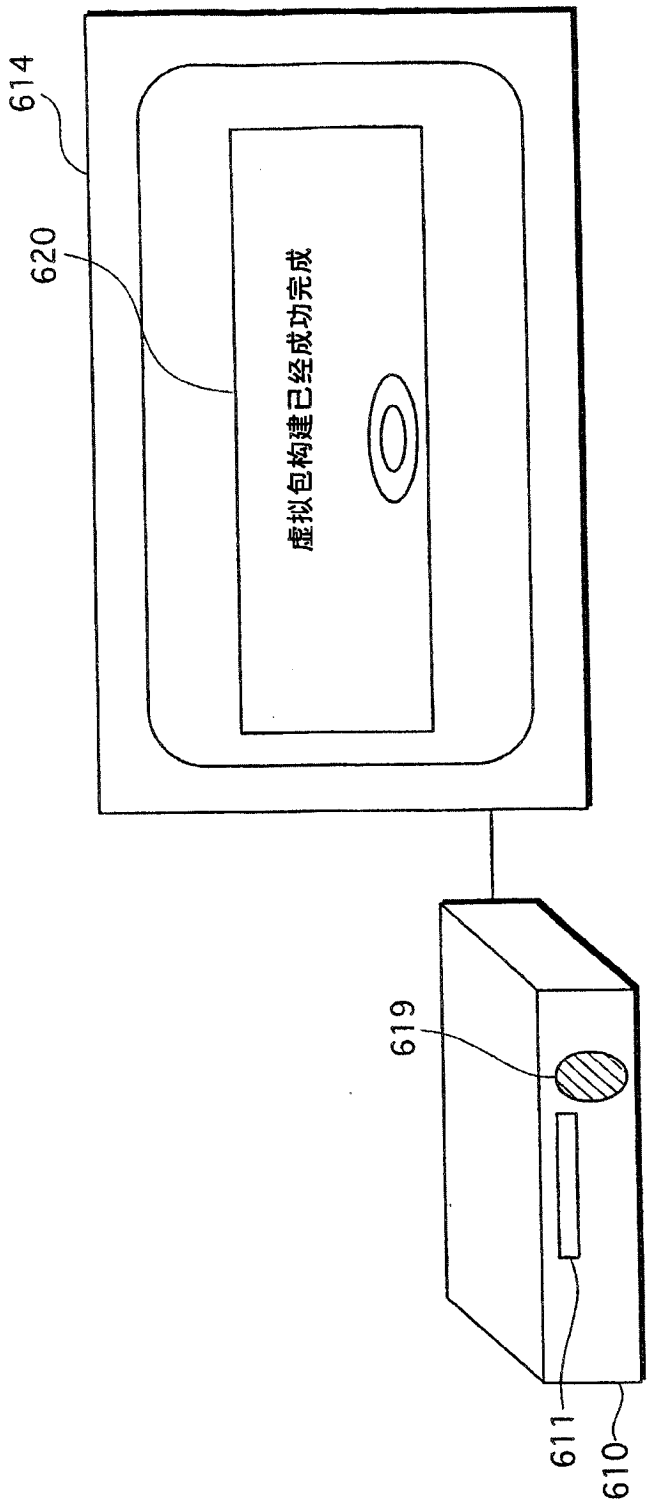

图 57

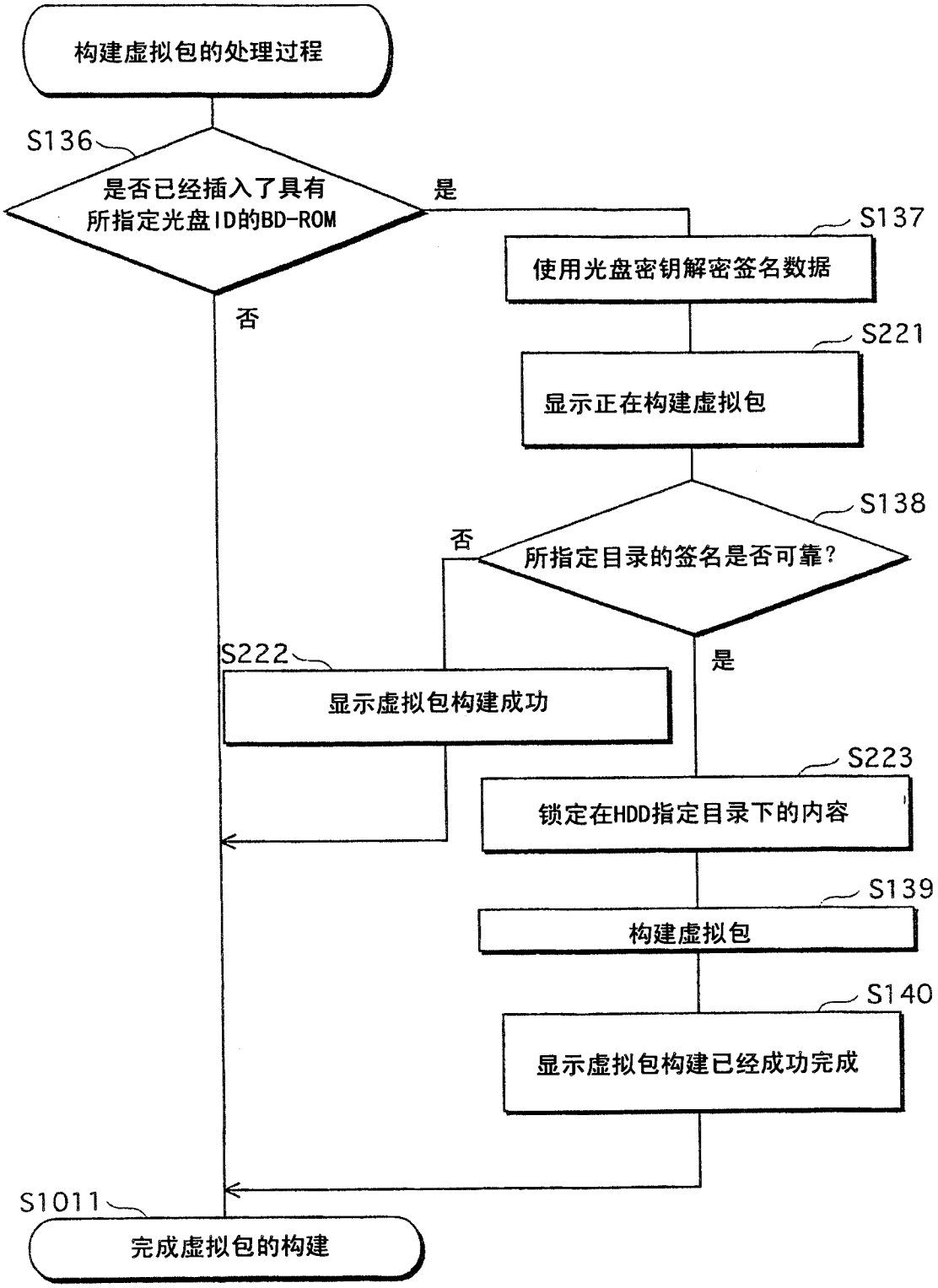
构建虚拟包的处理过程
S136
是否已经插入了具有所指定光盘ID的BD-ROM
是
否
S137
使用光盘密钥解密签名数据
S221
显示正在构建虚拟包
S138
所指定目录的签名是否可靠？
否
是
S222
显示虚拟包构建成功
S223
锁定在HDD指定目录下的内容
S139
构建虚拟包
S140
显示虚拟包构建已经成功完成
S1011
完成虚拟包的构建

图 58

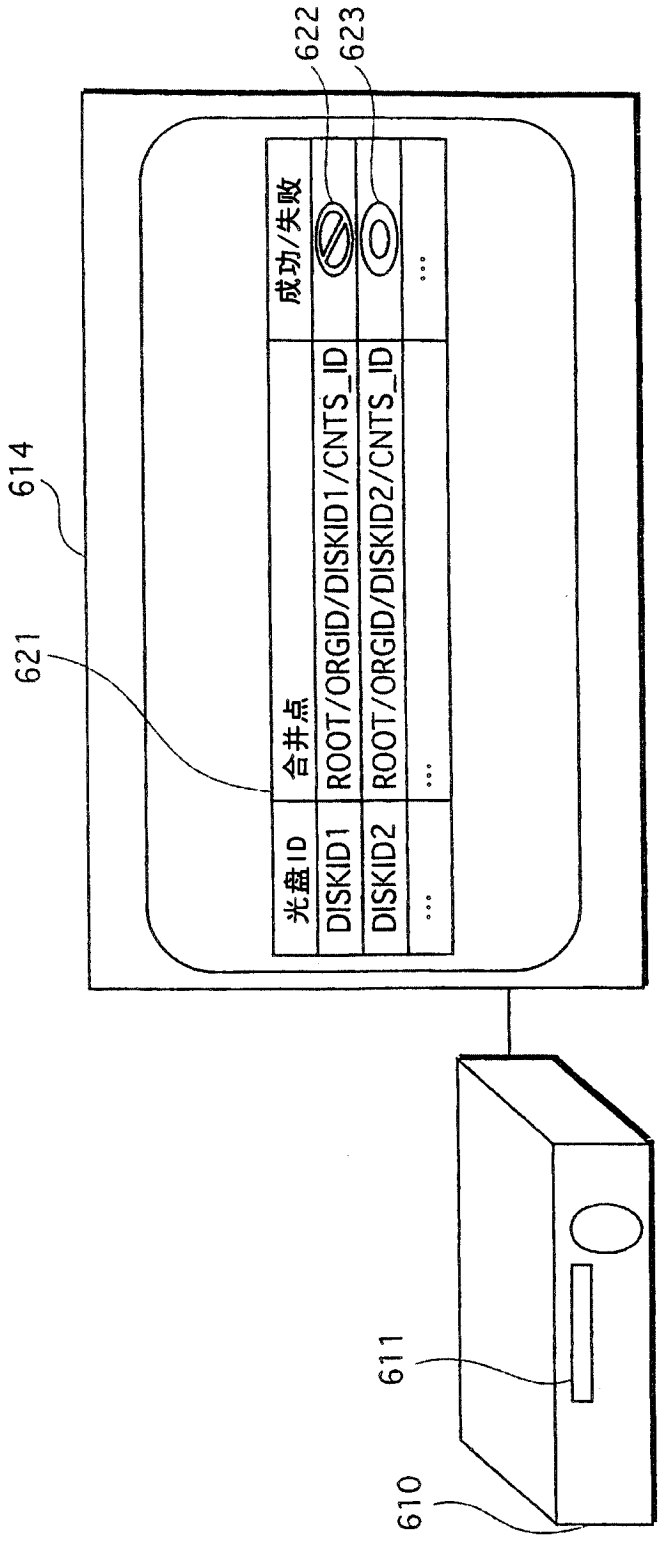
622
623
614
621
光盘ID
合并点
成功/失败
DISKID1
ROOT/ORGID/DISKID1/CNTS_ID
DISKID2
ROOT/ORGID/DISKID2/CNTS_ID
611
610

图 59