(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101714384 A
(43)申请公布日 2010.05.26

(21)申请号 200910211505.1

(22)申请日 2005.05.10

(30)优先权数据
2004-141560 2004.05.11 JP

(62)分案原申请数据
200580015163.6 2005.05.10

(71)申请人 松下电器产业株式会社
地址 日本大阪府

(72)发明人 田中敬一 大芦雅弘 金丸智一

(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 徐殿军

(51)Int.Cl.
G11B 20/10(2006.01)
G11B 27/10(2006.01)
G11B 27/11(2006.01)
G11B 27/32(2006.01)
G11B 27/34(2006.01)
G11B 27/034(2006.01)
H04N 5/85(2006.01)

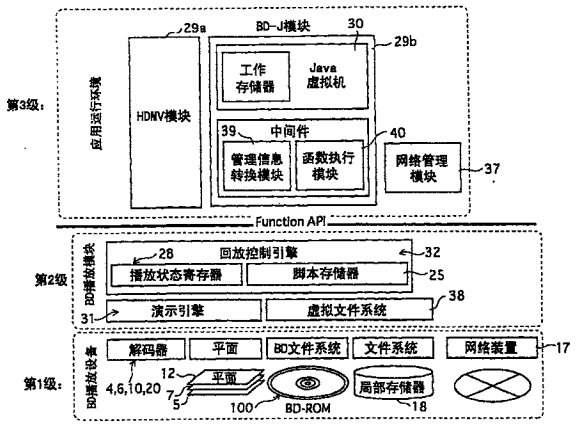
权利要求书 2 页 说明书 23 页 附图 31 页

(54)发明名称

再现装置、程序、再现方法

(57)摘要

本发明涉及再现装置、程序及再现方法。记录在BD_ROM及局部存储器(18)的多个AVClip构成一个流序列时，如果回放控制引擎(32)是构成流序列的一部分AVClip能识别的允许状态，即使其余的AVClip是不能识别的禁止状态，也开始进行再现。虚拟文件系统部(38)在对流序列中的一部分AVClip的再现处理继续的期间，按照应用程序对函数的调用，把其余AVClip的状态变更为回放控制引擎(32)能识别的允许状态、或不能识别的禁止状态。



CN 101714384 A

1. 一种再现装置，再现流序列，其特征在于，具备：

再现单元；

包管理单元，当用于存储构成上述流序列的各个数字流的存储场所在只读型记录介质和可重写型记录介质的各自中分开存在时，利用上述只读型记录介质上的文件管理信息和上述可重写型记录介质中的文件管理信息生成虚拟的卷标管理信息，通过将根据上述虚拟的卷标管理信息确定的数字流的状态设定为上述再现单元能识别的状态，使上述再现单元能够再现上述设定为能识别状态的数字流，或者通过将根据上述虚拟的卷标管理信息确定的上述可重写型记录介质上的数字流设定为上述再现单元不能识别的状态，使上述再现单元不能够再现上述设定为不能识别状态的数字流；以及

应用程序执行单元，执行应用程序，以执行对上述记录介质的存取控制；

上述再现单元能够识别上述只读型记录介质的文件布局，并且当构成上述流序列的一部分的数字流是能识别的状态时，即使其余的数字流是不能识别的状态，上述再现单元也开始再现，

在对上述流序列中的一部分数字流的再现处理继续进行期间，上述包管理单元按照来自应用程序执行单元的请求，将上述其余的数字流的状态变更为上述再现单元能识别的状态，或者将上述再现单元能识别状态的数字流中除了正在进行上述再现处理的一部分数字流之外的数字流变更为不能识别的状态。

2. 一种程序，使计算机执行再现流序列的处理，其特征在于，

使计算机执行如下步骤：

再现步骤；

包管理步骤，当用于存储构成上述流序列的各个数字流的存储场所在只读型记录介质和可重写型记录介质的各自中分开存在时，利用上述只读型记录介质上的文件管理信息和上述可重写型记录介质中的文件管理信息生成虚拟的卷标管理信息，通过将根据上述虚拟的卷标管理信息确定的数字流的状态设定为上述再现步骤能识别的状态，使上述再现步骤能够再现上述设定为能识别状态的数字流，或者通过将根据上述虚拟的卷标管理信息确定的上述可重写型记录介质上的数字流设定为上述再现步骤不能识别的状态，使上述再现步骤不能够再现上述设定为不能识别状态的数字流；以及

应用程序执行步骤，执行应用程序，以执行对上述记录介质的存取控制；

上述再现步骤能够识别上述只读型记录介质的文件布局，并且当构成上述流序列的一部分的数字流是能识别的状态时，即使其余的数字流是不能识别的状态，上述再现步骤也开始再现，

在对上述流序列中的一部分数字流的再现处理继续进行期间，上述包管理步骤按照来自上述应用程序执行步骤的请求，将上述其余的数字流的状态变更为上述再现单元能识别的状态，或者将上述再现步骤能识别状态的数字流中除了正在进行上述再现处理的一部分数字流之外的数字流变更为不能识别的状态。

3. 一种再现方法，再现流序列，其特征在于，具备：

再现步骤；

包管理步骤，当用于存储构成上述流序列的各个数字流的存储场所在只读型记录介质和可重写型记录介质的各自中分开存在时，利用上述只读型记录介质上的文件管理信息和

上述可重写型记录介质中的文件管理信息生成虚拟的卷标管理信息，通过将根据上述虚拟的卷标管理信息确定的数字流的状态设定为上述再现步骤能识别的状态，使上述再现步骤能够再现上述设定为能识别状态的数字流，或者通过将根据上述虚拟的卷标管理信息确定的上述可重写型记录介质上的数字流设定为上述再现步骤不能识别的状态，使上述再现步骤不能够再现上述设定为不能识别状态的数字流；以及

应用程序执行步骤，执行应用程序，以执行对上述记录介质的存取控制；

上述再现步骤能够识别上述只读型记录介质的文件布局，并且当构成上述流序列的一部分的数字流是能识别的状态时，即使其余的数字流是不能识别的状态，上述再现步骤也开始再现，

在对上述流序列中的一部分数字流的再现处理继续进行期间，上述包管理步骤按照来自上述应用程序执行步骤的请求，将上述其余的数字流的状态变更为上述再现单元能识别的状态，或者将上述再现步骤能识别状态的数字流中除了正在进行上述再现处理的一部分数字流之外的数字流变更为不能识别的状态。

再现装置、程序、再现方法

[0001] 本申请是申请日为2005年5月10日，申请号为200580015163.6，发明创造名称为再现装置、程序、再现方法的申请的分案申请。

技术领域

[0002] 本发明是属于虚拟包(Virtual Package)的技术领域的发明。

背景技术

[0003] 虚拟包是指，通过动态地组合记录在BD-ROM等只读型记录介质的数字流、和记录在硬盘等可重写型记录介质的数字流，构筑虚拟数据包，从而谋求只读型记录介质的容量扩张的技术。当记录在该BD-ROM的数字流构成电影作品的正篇，记录在硬盘的数字流构成电影作品的续篇时，通过构筑上述的虚拟包，能够将BD-ROM上的本篇和硬盘上的续篇作为一个长篇的电影作品，提供给再现。

[0004] 并且，作为关于虚拟包的已有技术，有记载在以下专利文献的技术。

[0005] 专利文献1：日本特开2002-369154号公报

[0006] 但是，在将记录在BD-ROM的数字流和记录在硬盘的数字流作为一个流序列进行再现时，在构成流序列的数字流中，有可能包含不稳定状态的内容。在此，不稳定状态的数字流是指，向硬盘的下载没结束、或者写入不完全这样的数字流。如果不顾包含不稳定状态的数字流，还要再现上述的流序列，则在再现单元的再现时刻刚到达不稳定流时，有可能产生再现紊乱及再现中断。

[0007] 在此，若再现装置内的再现单元（解码器等）能够正确地检验要再现的流是否处于不稳定状态，就可以避免这样的再现的紊乱和中断。但是，再现单元在每个1/24秒、1/29.4秒的帧期间专注于输出图像信号的处理，通常不存在用于进行那样的检验的余力。因此，无法避免如上所述的再现的紊乱和中断的发生。另一方面，在制造BD-ROM再现装置的民用设备的制造领域，对于图像再现要求严格的品质标准，即使在再现单元侧没有检验数字流的余力，也不能够允许如上所述的再现的紊乱和中断。

发明内容

[0008] 本发明的目的在于提供一种再现装置，即使在再现单元侧没有检验不稳定状态的数字流的余力的情况下，也可以防止将记录在BD-ROM的数字流和记录在硬盘的数字流作为一个流序列进行再现时的再现的紊乱和中断。

[0009] 为了实现上述目的，涉及本发明的再现装置，具备：再现单元；包管理单元，当用于存储构成上述流序列的各个数字流的存储场所在只读型记录介质和可重写型记录介质的各自中分开存在时，利用上述只读型记录介质上的文件管理信息和上述可重写型记录介质中的文件管理信息生成虚拟的卷标管理信息，通过将根据上述虚拟的卷标管理信息确定的数字流的状态设定为上述再现单元能识别的状态，使上述再现单元能够再现上述设定为能识别状态的数字流，或者通过将根据上述虚拟的卷标管理信息确定的上述可重写型记录

介质上的数字流设定为上述再现单元不能识别的状态，使上述再现单元不能够再现上述设定为不能识别状态的数字流；以及应用程序执行单元，执行应用程序，以执行对上述记录介质的存取控制；上述再现单元能够识别上述只读型记录介质的文件布局，并且当构成上述流序列的一部分的数字流是能识别的状态时，即使其余的数字流是不能识别的状态，上述再现单元也开始再现；在对上述流序列中的一部分数字流的再现处理继续进行期间，上述包管理单元按照来自应用程序执行单元的请求，将上述其余的数字流的状态变更为上述再现单元能识别的状态，或者将上述再现单元能识别状态的数字流中除了正在进行上述再现处理的一部分数字流之外的数字流变更为不能识别的状态。

[0010] 通过应用程序变更为可以识别数字流的状态、或者不能识别的状态，可以实现再现单元再现数字流、或者不再现的选择。由实现记录介质的存取的应用程序掌握使再现单元再现数字流、或者不再现的选择权，所以应用程序在不能确认数字流没有正确地写入在记录介质时，若将该数字流设为不能识别的状态，认为其数字流未曾存在地进行处理。由此，不会错误地再现不稳定的数字流。这样，不会出现图像的紊乱和中断。

[0011] 在此，在多个记录介质中有只读型记录介质和可重写型记录介质；上述再现单元只能识别只读型记录介质的文件布局；记录在可重写型记录介质的数字流以存储在文件中的状态被记录在可重写型记录介质；由上述包管理单元进行的向可识别状态的变化，是根据只读型记录介质中上的文件管理信息和可重写型记录介质中的文件管理信息来生成虚拟的卷标管理信息，并传递给再现单元而进行的；向不可识别状态的变化，是根据上述虚拟的卷标管理信息除去可重写型记录介质的文件管理信息进行的。

[0012] 由包管理单元进行的可识别状态的变化，还包含在可重写型记录介质中把存储了数字流的文件变更为只读属性的处理；向不能识别状态的变更，还包含在可重写型记录介质中把存储了数字流的文件变更为可重写性属性的处理。

[0013] 在此，一部分数字流的再现中，应用程序进行控制，以便把其余的数字流下载到记录介质，若该一部分数字流的再现结束，通过把该数字流设定为可识别的状态，能够使再现单元进行写入的数字流的再现。

[0014] 而且，若其余数字流的写入在一部分的数字流的再现期间内没有结束，通过把该数字流设定为不能识别的状态，可以从再现单元的再现中除去这样的数字流。数字流的再现可否的设定可以与下载处理的进度联动，所以，可以并行地执行构成流序列的数字流的再现和构成流序列的数字流的下载。在此，当下载在 BD-ROM 的数字流是构成电影本篇的数字流，下载到硬盘的数字流是电影的预告篇时，在本篇数字流的再现中下载预告篇数字流，若本篇数字流的再现结束，可以在其后马上开始预告篇数字流的再现。

附图说明

[0015] 图 1 是表示关于本发明涉及的再现装置的使用行为的方式的图。

[0016] 图 2 是表示 BD-ROM 的内部结构的图。

[0017] 图 3 是模式地表示 AVClip 如何构成的图。

[0018] 图 4 是表示 PL 信息的构成的图。

[0019] 图 5 是表示 AVClip 时间轴和 PL 时间轴之间关系的图。

[0020] 图 6 是表示由四个 Clip_Information_file_name 构成的成批指定的图。

[0021] 图 7 是表示 PLmark 信息的内部结构的图。

[0022] 图 8 是表示利用 PLmark 的章定义的图。

[0023] 图 9 是表示子路径信息的内部结构的图。

[0024] 图 10 是表示子播放条目 (SubPlayItem) 时间轴上的再现区间定义和同步指定的图。

[0025] 图 11(a) 是表示收纳在 Java(注册商标) 档案文件中的程序和数据的图。(b) 是表示类文件的内部结构的图。

[0026] 图 12 是表示本发明涉及的再现装置的内部结构的图。

[0027] 图 13(a) 是表示局部存储器 (Local Storage)18 中的目录结构的图。(b) 是表示流存储位置信息文件的内部结构的图。

[0028] 图 14 是将由存储在 ROM24 的软件和硬件构成的部分置换为层结构描述的图。

[0029] 图 15 是表示 Java(注册商标) 虚拟机 30 的内部结构的图。

[0030] 图 16 是表示虚拟文件系统 (Virtual File System) 部 38 的虚拟包信息结构的一例的图。

[0031] 图 17 是表示虚拟文件系统部 38 的状态管理的一例的图。

[0032] 图 18 是表示反映了状态管理的虚拟包信息的生成的图。

[0033] 图 19 是表示网络管理模块 37 的下载正在进行时的虚拟文件系统部 38 的虚拟包构筑的图。

[0034] 图 20(a) 是表示利用存储在局部存储器的 PL 信息定义了怎样的播放表 (PlayList) 再现时间轴的图。(b) 是表示前进播放表 (Progressive PlayList) 信息的概念的图。

[0035] 图 21 是表示管理信息转换模块 39 的实例生成的图。

[0036] 图 22 是表示播放表对象的 Play 函数的调用的图。

[0037] 图 23 是模式地表示 getMissingStreams 函数的询问结果的图。

[0038] 图 24 是模式地表示 enableClip 函数的调用的图。

[0039] 图 25 是模式地表示 disableClip 函数的调用的图。

[0040] 图 26 是表示虚拟包构筑时的处理的流程图。

[0041] 图 27(a) 是表示 enableClip 函数的调用时的处理顺序的流程图。(b) 是表示 disableClip 函数的调用时的处理顺序的流程图。

[0042] 图 28 是表示回放控制引擎 (Playback Control Engine)32 的再现顺序的流程图。

[0043] 图 29 是表示应用程序的处理顺序的流程图。

[0044] 图 30(a)、(b) 是表示网络管理模块 37 的下载进度和回放控制引擎 32 的再现进度的图。

[0045] 图 31(a)、(b) 是表示网络管理模块 37 的下载进度和回放控制引擎 32 的再现进度的图。

[0046] 图 32(a)、(b) 是表示网络管理模块 37 的下载进度和回放控制引擎 32 的再现进度的图。

[0047] 图 33 是表示网络管理模块 37 的下载进度和回放控制引擎 32 的再现进度的图。

具体实施方式

[0048] （第 1 实施方式）

[0049] 下面，说明本发明涉及的记录介质的实施方式。首先，说明本发明涉及的再现装置的实施行为中的使用行为的方式。图 1 是表示关于本发明涉及的再现装置的使用行为的方式的图。在图 1 中，本发明涉及的再现装置是再现装置 200。该再现装置 200 的用途是，向由遥控器 300、电视机 400 形成的家庭影院系统供给电影作品。

[0050] 以上说明了本发明涉及的再现装置的使用方式。接着，说明作为本发明涉及的再现装置所再现的对象的记录介质。由本发明涉及的再现装置再现的是 BD-ROM。图 2 是表示 BD-ROM 的内部结构的图。

[0051] 在该图的第 4 层表示 BD-ROM，在第 3 层表示 BD-ROM 上的轨道。该图的轨道是将从 BD-ROM 的内周至外周螺旋状地形成的轨道在横向上拉伸描绘的。该轨道由导入区、信息区和、导出区构成。该图的信息区具有称为物理层、文件系统层、应用层的层模型。如果使用目录结构表现 BD-ROM 的应用层格式（应用格式），就成为如图中的第 1 层。在该第 1 层中，在 BD-ROM 中的 Root 目录下，有 BDMV 目录。

[0052] 在 BDM 目录的下面，存在被称为 PLAYLIST 目录、CLIPINF 目录、STREAM 目录、BDJA 目录的四个子目录。

[0053] 在 STREAM 目录中，存在存储了成为所谓数字流主体的文件的目录，即赋予了后缀 M2TS 的文件 (00001.M2TS)。

[0054] 在 PLAYLIST 目录中，存在赋予了后缀 MPLS 的文件 (00001.MPLS)。

[0055] 在 CLIPINF 目录中，存在赋予了后缀 CLPI 的文件 (00001.CLPI)。下面，说明这些文件。

[0056] <AVClip>

[0057] 首先，说明赋予了后缀 .M2TS 的文件。图 3 是模式地表示赋予了后缀 .M2TS 的文件如何构成的图。赋予了后缀 .M2TS 的文件 (00001.M2TS，00002.M2TS，00003.M2TS) 存储 AVClip。AVClip（中层）是，将由多个视频帧（像素 pj1，2，3) 构成的视频流、由多个音频帧构成的音频流（上第 1 层）转换为 PES 包列（上第 2 层），进一步转换为 TS 包（上第 3 层），相同地，将字幕类的演示图形流（下第 1 层的 PG 流）及对话类的交互图形流（下第 2 层的 IG 流）转换为 TS 包（下第 3 层），然后对它们进行多路复用而构成。

[0058] AVClip 除了如图 3 所示的进行了多路复用的流以外，还有没进行多路复用的流。这样的 AVClip 被称为 SubClip，存在构成音频流、图形流、文本字幕流 (TextSTStream) 等的 AVClip。

[0059] <Clip 信息 >

[0060] 赋予了后缀“CLPI”的文件 (00001.CLPI) 是与各个 AVClip 一对一对应的 Clip 信息。由于是管理信息，Clip 信息持有 AVClip 中的流的编码方式、帧速率、比特率、分辨率等的信息，和表示 GOP 的起始位置位置的 EP_map。

[0061] <PlayList 信息 >

[0062] 赋予了后缀“MPLS”的文件 (00001.MPLS) 是存储了 PlayList(PL) 信息的文件。PL 信息是参照 AVClip 定义播放表的信息。图 4 是表示 PL 信息的结构的图，如该图的左侧所示，PL 信息由“MainPath 信息”、“PLMark 信息”、“子路径信息”构成。

[0063] MainPath 信息 (MainPath())，如点划线的箭头 mp1 所示地由多个播放条目信息 (PlayItem()) 构成。播放条目是指，在一个以上的 AVClip 时间轴上，通过指定 In_Time、Out_Time 定义的再现区间。通过配置多个播放条目信息，定义了由多个再现区间构成的播放表 (PL)。图中的点划线 mp2，特写播放条目信息的内部结构。如该图所示，播放条目信息，由表示对应的 AVClip 的“Clip_information_file_name”、“In_time”、和“Out_time”构成。图 5 是表示 AVClip 和 PL 的关系的图。第 1 层表示 AVClip 具有的时间轴，第 2 层表示 PL 具有的时间轴。PL 信息包含 PlayItem#1、#2、#3 的三个播放条目信息，由这些 PlayItem#1、#2、#3 的 In_time、Out_time，定义了三个再现区间。如果排列这些再现区间，就定义了与 AVClip 时间轴不同的时间轴。这是在第 2 层表示的 PL 时间轴。这样，通过播放条目信息的定义，可以定义与 AVClip 不同的时间轴。

[0064] 原则上对 AVClip 的指定是一个，但也可以进行对多个 AVClip 的成批指定。该成批指定是通过播放条目信息中的多个 Clip_Information_file_name 进行。图 6 是表示由四个 Clip_Information_file_name 形成的成批指定的图。在该图中，第 1 层～第 4_ 层表示四个 AVClip 时间轴 (AVClip#1，#2，#3，#4 的时间轴)，第 5 层表示 PL 时间轴。由播放条目信息持有的四个 Clip_Information_file_name，指定了这四个时间轴。通过这样做，根据播放条目所具有的 In_time、Out_time，择一地定义了可以再现的四个再现区间。由此，在 PL 时间轴上定义了由可切换的多个角度影像构成的区间 (所谓多角度区间)。

[0065] PLmark 信息 (PLmark()) 是将 PL 时间轴中的任意区间指定为章的信息。图 7 是表示 PLmark 信息的内部结构的图，如该图的引出线 pml 所示 PLmark 信息包含“ref_to_PlayItem_Id”和“Mark_time_stamp”。图 8 是表示利用 PLmark 的章定义的图。该图中的第 1 层表示 AVClip 时间轴，第 2 层表示 PL 时间轴。图中的箭头 pk1，2，表示 Plmark 中的播放条目指定 (ref_to_PlayItem_Id) 和某一时刻的指定 (mark_time_stamp)。通过这些指定，在 PL 时间轴上定义了三个章 Chapter#1、#2、#3)。以上是关于 PLmark 的说明。接着，说明子路径信息。

[0066] 子路径信息 (SubPath()) 是通过在 SubClip 的时间轴上指定 In_Time、Out_Time 来定义一个以上的再现区间的信息，具有图 9 所示的内部结构。该图所示的子路径信息如点划线的引出线 sh1 所示，由多个子播放条目信息 (SubPlayItem()) 构成。如使用点划线 sh2 进行特写那样，子播放条目信息由“Clip_information_file_name”、“In_time”、“Out_time”、“Sync_PlayItem_Id”、“Sync_start_Pts_of_PlayItem”构成。对于 SubClip 的时间轴上的 In_Time、Out_Time 的指定，由“Clip_information_file_name”、“In_time”、“Out_time”构成。“Sync_PlayItem_Id”及“Sync_start_Pts_of_PlayItem”，成为使 SubClip 时间轴上的再现区间和 PL 时间轴同步的同步指定。通过该同步指定，PL 时间轴和 SubClip 时间轴步进行。

[0067] 图 10 是表示 SubClip 时间轴上的再现区间定义定义和同步指定的图。在该图中，第 1 层表示 PL 时间轴，第 2 层表示 SubClip 时间轴。图中的 SubPlayItem.IN_time 表示再现区间的起始点，SubPlayItem.Out_time 表示再现区间的终点。由此可以知道，在 SubClip 时间轴上也定义着再现区间。在箭头 Sn1 中，Sync_PlayItem_Id 表示对播放条目的同步指定，在箭头 Sn2 中，sync_start_PTS_of_PlayItem 表示 PL 时间轴的播放条目的某时刻的指定。

[0068] 能够进行可切换多个 AVClip 的多角度区间、可以使 AVClip_SubClip 同步的同步区间的定义，是 BD-ROM 的 PL 信息的特征。以上的 Clip 信息及 PL 信息被分类为“静态脚本”。

[0069] 接着说明“动态脚本”。“动态”是指，通过再现装置 200 的状态变化或来自用户的键盘事件变更再现控制的内容。在 BD-ROM 中，利用与 Java（注册商标）应用程序相同的记述，可以记述该再现控制。即在 BD-ROM 中，Java（注册商标）应用程序担当动态脚本的作用。

[0070] <Java（注册商标）应用程序>

[0071] 说明 Java（注册商标）应用程序。Java（注册商标）应用程序由被装载到虚拟机的堆区域（还称为工作存储器）的一个以上的 xlet 程序构成。应用程序由被装载到该工作存储器的 xlet 程序及数据构成。以上是 Java（注册商标）应用程序的构成。

[0072] 相当于该 Java（注册商标）程序的实体的是存储在图 2 的 BDMV 目录下的 BDJA 目录的 Java（注册商标）档案文件 (00001.jar,00002.jar)。下面，参照图 11 说明 Java（注册商标）档案文件。

[0073] <Java（注册商标）档案文件>

[0074] Java（注册商标）档案文件（图 9 的 00001.JAR) 是将一个以上的类文件、一个以上的数据文件等归纳为一个而得到的文件。图 11(a) 是表示由档案文件收纳的程序、数据的图。该图的数据是用 Java（注册商标）档案归纳配置了框内所示的目录结构的多个文件的数据。框内所示的目录结构由根目录、Java（注册商标）目录、image 目录构成，在根目录配置着 common.pkg，在 Java（注册商标）目录是配置着类文件 (aaa.class,bbb.class)，在 image 目录配置着 menu.jpg。Java（注册商标）档案文件是用 Java（注册商标）档案归纳这些而得到的。这样的类文件及数据在从 BD-ROM 读取到高速缓存时被展开，在高速缓存上，作为配置在目录下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”这五位数值，表示应用程序的 ID(application ID)。在该 Java（注册商标）档案文件被读取到高速缓存时，通过参照该文件名中的数值，可以取出构成任意的 Java（注册商标）应用程序的程序及数据。

[0075] 该图中的类文件（图中的 aaa.class，bbb.class) 是对应上述 xlet 程序的类文件。Java（注册商标）工作环境的工作模式 (BD-J 模式）下的再现步骤，由相当于该类文件的是实例的 xlet 程序规定。Xlet 程序是可以利用 JMF(Java（注册商标）Media FrameWork) 方式的接口的 Java（注册商标）程序，按照 JMF 等的方式，基于键事件进行处理。Xlet 程序可以进行 JMF 方式的处理，所以，通过生成对于 MPLS 文件的 JMF 方式的实例（播放表对象），可以向再现装置 200 命令播放表再现。另外在 xlet 程序中，通过记载功能 API 的调用，可以执行 BD-ROM 再现装置 200 特有的处理。

[0076] 再者，xlet 程序还可以执行访问三维网页后下载内容的步骤。由此可以再现交换了下载内容和播放表再现的崭新的作品。

[0077] 说明 xlet 程序的类文件。图 11(b) 是表示类文件的内部结构的图。该图所示的类文件与通常的类文件相同，由“常量池”、“接口”、“函数 1,2,3……n”构成。在类文件的函数中，有预先注册了成为动作的触发的键事件的函数 (EventListner)、和调用 BD-ROM 再现装置 200 侧的功能 API 的函数。这些函数记载了通过使用分配给自身的局部变量、或调

用自身时的参数，进行运算等的步骤。以上说明了 Java（注册商标）档案文件。由此，结束对 BD-ROM 的说明。下面说明本发明涉及的再现装置的内部结构。

[0078] ＜再现装置＞

[0079] 图 12 是表示本发明涉及的再现装置的内部结构的图。本发明涉及的再现装置是基于该图所示的内部结构工业生产出来的。本发明涉及的再现装置，主要由系统 LSI 和称为驱动装置的两个配件构成，通过在装置的箱体及基板上实装这些配件，可以进行工业生产。系统 LSI 是集成了执行再现装置的功能的各种处理部的集成电路。这样生产的再现装置包括 BD-ROM 驱动器 1、读缓冲器 2、解复用器 3、视频解码器 4、视频平面 5、P 图形解码器 6、演示图形平面 7、合成部 8、字体生成器 9、I 图形解码器 10、开关 11、交互图形平面 12、合成部 13、CLUT 部 14、CLUT 部 15、音频解码器 16、网络装置 17、局部存储器 18、读缓冲器 19、解复用器 20、命令 ROM21、用户事件处理部 22、PSR 组 23、CPU24、脚本存储器 25、局部内存 26、开关 27。

[0080] 首先，说明与记录在 BD-ROM 的 AVClip 再现有关的构成要素 (BD 驱动器 1 ～音频解码器 16)。

[0081] BD-ROM 驱动器 1 进行 BD-ROM 的装载 / 排出，执行对 BD-ROM 的存取。

[0082] 读缓冲器 2 是 FIFO 存储器，按先入先出方式存储了从 BD-ROM 读取的 TS 数据包。

[0083] 解复用器 (De-MUX)3 从读缓冲器 2 取出 TS 数据包，将构成该 TS 数据包的 TS 数据包转换成 PES 数据包。然后，将通过转换得到的 PES 数据包中的、具有由 CPU24 设定的 PID 的 PES 数据包，向视频解码器 4、P 图形解码器 6、I 图形解码器 10、音频解码器 16 的某一个输出。

[0084] 视频解码器 4 解码从解复用器 3 输出的多个 PES 数据包，得到非压缩形式的图像，写入到视频平面 5。

[0085] 视频平面 5 是用于存储非压缩形式的图像的平面。平面是指，在再现装置中用于存储一个画面的图像数据的存储区域。视频平面 5 的分辨率是 1920×1080，存储在该视频平面 5 的图像数据，由用 16 比特的 YUV 值表现的图像数据构成。在视频平面 5 中，可以对视频流中的每个帧的再现影像进行定标。定标是指，使每一帧的再现图像变化为视频平面 5 整体的 1/4、1/1（满定标）中的某一个。在 BD-J 模式下按照来自 CPU24 的指示执行这样的定标，所以，可以将视频流的再现图像缩到画面的角落、或者进行在整个画面输出的画面演出。

[0086] P 图形解码器 6 对从 BD-ROM 读取的演示图形流进行解码，向演示图形平面 7 写入非压缩图形。通过图形流的解码，字幕可以显示在画面上。

[0087] 演示图形平面 7 是具有相当于一个画面的区域的存储器，可以存储一个画面的非压缩图形。该平面的分辨率是 1920×1080，演示图形平面 7 中的非压缩图形的各像素用 8 比特的指示色表现。通过使用 CLUT(Color Lookup Table) 转换这样的指示色，存储在演示图像平面 7 的非压缩图形用于显示。

[0088] 合成部 8 将存储在视频平面 5 的非压缩状态的图像数据 (i) 和演示图形平面 7 的存储内容进行合成。

[0089] 字体生成器 9 使用字符字体将包含在 textST 流中的文本代码展开成位图，并写入演示图形平面 7。

[0090] I 图形解码器 10 在 HDMV 模式下对从 BD-ROM 或局部存储器 18 读取的 IG 流进行解码，将非压缩图形写入交互图形平面 12。

[0091] 开关 11 是将字体生成器 9 生成的字体列、和通过 P 图形解码器 6 的解码得到的图形的某一个有选择地写入演示图形平面 7 的开关。

[0092] 交互图形平面 12 中写入通过 I 图形解码器 10 的解码而得到的非压缩图形。而且，在交互图形平面 12 中，在 BD-J 模式下写入由应用程序描绘的字符或图形。

[0093] 合成部 13 合成交互图形平面 12 的存储内容和作为合成部 8 的输出的合成图像（合成了非压缩状态的图形数据和演示图形平面 7 的存储内容的图像）。通过这样的合成，可以在非压缩状态的图形数据上叠加应用程序写入 I 图形解码器 10 的字符及图形，进行显示。

[0094] CLUT 部 14 将存储在视频平面 5 的非压缩图形中的指示色转换为 Y、Cr、Cb 值。

[0095] CLUT 部 15 将存储在交互图形平面 12 的非压缩图形中的指示色转换为 Y、Cr、Cb 值。

[0096] 音频解码器 16 对从解复用器 3 输出的 PES 包进行解码，输出非压缩形式的音频数据。

[0097] 以上是涉及 AVClip 再现的构成要素。接着，说明涉及 BD-J 模式下的动作的构成要素（网络装置 17 ～解复用器 20）。

[0098] 网络装置 17 是实现再现装置的通信功能的设备，若在 BD-J 模式下由 Java（注册商标）应用程序赋予了 URL 指定，则确立与相当于该 URL 的 web 网页的 TCP 链接、FTP 链接等。通过这样的链接确立，使 Java（注册商标）应用程序进行自 web 网页的下载。

[0099] 局部存储器 18 是将通过由网络装置 17 确立的链接从 web 网页下载的内容等、从 DB-ROM 以外的记录介质及通信介质供给的内容，同，中间数据一起存储的硬盘。该中间数据是用于将下载内容绑定到局部存储器 18 进行管理的信息，通过对该局部存储器 18 进行存取，BD-J 模式下的应用程序可以进行利用了下载内容的各种处理。

[0100] 读缓冲器 19 是 FIFO 存储器，在存储在局部存储器 18 的下载内容中包含 SubClip 时，按先入先出方式存储构成该 SubClip 的 TS 数据包。

[0101] 解复用器 (De-MUX)20 从读缓冲器 19 取出 TS 数据包，将 TS 数据包转换为 PES 数据包。然后，将通过转换得到的 PES 数据包中的、具有希望的 PID 的 PES 数据包，输出给字体生成器 9、I 图形解码器 10、音频解码器 16。

[0102] 利用以上的网络装置 17 ～ De-mux20，可以与记录在 BD-ROM 的内容同样得再现 Java（注册商标）应用程序通过网络下载的内容。接着，说明实现再现装置中的综合控制的构成要素（命令 ROM21 ～开关 27）。

[0103] 命令 ROM21 存储着规定再现装置的控制的软件。

[0104] 用户事件处理部 22 根据对遥控器或再现装置的前面板的键操作，向 CPU24 输出进行该操作的用户事件。

[0105] PSR 组 23 是内置在再现装置中的寄存器，由 64 个播放状态寄存器 (PSR) 和 4096 个的通用寄存器 (GPR) 构成。播放状态寄存器的设定值 (PSR) 中的 PSR4 ～ PSR8 用于表现现在的再现时刻。

[0106] PSR4，通过被设定为 1 ～ 100 的值，表示现在的再现时刻所属的标题，通过被设定

为 0，表示现在的再现时刻是顶级菜单。

[0107] PSR5，通过被设定为 1 ～ 999 的值，表示现在的再现时刻所属的章序号，通过被设定为 0xFFFF，表示再现装置中章序号是无效的。

[0108] PSR6，通过被设定为 0 ～ 999 的值，表示现在的再现时刻所属的 PL（当前 PL）的序号。

[0109] PSR7，通过被设定为 0 ～ 255 的值，表示现在的再现时刻所属的播放条目（当前播放条目）的序号。

[0110] PSR8，通过被设定为 0 ～ 0xFFFFFFFF 的值，使用 45KHz 的时间精度表示现在的再现时刻（当前 PTM(Presentation Time)）。通过以上的 PSR4 ～ PSR8，在图 21(a) 的 BD-ROM 全体的时间轴中，可以特定现在的再现时刻在哪儿。

[0111] CPU24 执行存储在命令 ROM21 的软件，执行再现装置全体的控制。该控制的内容根据从用户事件处理部 22 输出的用户事件、及 PSR 组 23 的各 PSR 的设定值，动态地变化。

[0112] 脚本存储器 25 是用于存储当前的 PL 信息或当前的 Clip 信息的存储器。当前 PL 信息是指，存储在 BD-ROM 的多个 PL 信息中的成为当前处理对象的 PL 信息。当前 Clip 信息是指，记录在 DB-ROM 的多个 Clip 信息中的成为当前处理对象的 Clip 信息。

[0113] 局部内存 26 是，因从 BD-ROM 的读取是低速，所以用于暂时存储 BD-ROM 的记录内容的缓冲存储器。通过存在这样的局部内存 26，BD-J 模式下的程序执行效率化。

[0114] 开关 27 是将从 BD-ROM 及局部存储器 18 读取的各种数据有选择地投入到读缓冲器 2、读缓冲器 19、脚本存储器 25、局部内存 26 的某一个的开关。

[0115] 以上是本实施方式涉及的再现装置的硬件结构。接着，说明本实施方式涉及的再现装置的文件系统结构。

[0116] 图 13(a) 是表示局部存储器 18 的目录结构的图。

[0117] 在该图的目录结构中的根目录下，有“DLDAT”的子目录，在其下，有“PKG00001”、“PKG00002”的子目录。“PKG00001”、“PKG00002”分别是 BD-ROM 的卷标号，表示在局部存储器 18 中存在与具有“PKG00001”这样卷标号的 BD-ROM 对应的目录、和与具有“PKG00002”这样卷标号的 BD-ROM 对应的目录。其中，子目录“PKG00001”表示是与卷标号“PKG00001”的 BD-ROM 对应的、下载数据的放置位置。通过在对应各 BD-ROM 的目录下设置子目录，分别存储关于各 BD-ROM 的下载数据。在该子目录下，与存储在 BD-ROM 的信息相同地存储着播放表信息、Clip 信息、AVClip，另外存在 Java（注册商标）档案文件、和流存储位置信息文件。

[0118] 在此，记录在 BD-ROM 的 AVClip 对应于电影作品的本篇，记录在局部存储器 18 上的 AVClip 是在目前在剧场上映的电影作品的预告篇。预告篇的内容随时代的变迁而变化，所以目前在剧场上映的电影作品的预告篇，在局部存储器 18 上作为 AVClip 记录着。

[0119] 说明后者的流存储位置信息文件。该流存储位置信息文件是表示构成虚拟包的文件一览的信息。图 13(b) 是表示流存储位置信息文件的内部结构的图，流存储位置信息文件包括构成虚拟包的各 AVClip 及 Clip 信息的存储位置信息。各存处位置信息包括 AVClip 及 Clip 信息的“识别符”和表示该 AVClip 及 Clip 信息在局部存储器 18 中的存储位置的“文件路径”。通过参照这样的存储位置信息，在构筑虚拟包时，可以确认作为虚拟包的构成要素的 AVClip 及 Clip 信息是否齐备。

[0120] 以上是本实施方式涉及的再现装置的文件系统结构。下面说明本实施方式涉及的再现装置中的软件结构。

[0121] 图 14 是将由存储在 ROM24 的软件和硬件构成的部分置换为层结构描绘的图。如该图所示，再现装置的层结构包括以下的 a)、b)、c)。即，包括：

[0122] a)BD 播放装置的第 1 级；

[0123] b)BD 播放模块的第 2 级；

[0124] c) 应用程序运行环境 (Application Runtime Enviroment) 的第 3 级。

[0125] 在这些级中，图 32 所示的再现装置的硬件结构属于第 1 级。在该图的第 1 级“BD Player Device”中，包含：由图 12 所示的硬件结构中的视频解码器 4、P 图形解码器 6、I 图形解码器 10、音频解码器 16 构成的“解码器”，由视频平面 5、演示图形平面 7、交互图形平面 12 构成的“平面”，BD-ROM 及其文件系统、局部存储器 18 及其文件系统。

[0126] 第 2 级“BD 播放模块”，包括以下的 b1)、b2) 层。即，包括：

[0127] b2) 回放控制引擎 32 的层，

[0128] b1) 虚拟文件系统 38 及演示引擎 31 的层，

[0129] 并且，对比自身上位的级提供，提供功能 API。

[0130] 其中，图 12 所示的 PSR 组 23、脚本存储器 25，存在于回放控制引擎 32 内。

[0131] 第 3 级“应用程序运行环境”包括以下的 c1) 的堆栈级。即，包括：

[0132] c1)HDMV 模块 29a、BD-J 模块 29b 所存在的层。

[0133] 下面，说明该软件结构中的各结构要素。

[0134] <HDMV 模块 29a，BD-J 模块 29b>

[0135] HDMV 模块 29a 解读导航命令，基于解读结果执行对回放控制引擎 32 的功能调用。

[0136] BD-J 模块 29b 是所谓 Java（注册商标）平台，成为将

[0137] d1-1)Java（注册商标）虚拟机 30

[0138] d1-2) 用于 Java（注册商标）虚拟机进行的工作的中间件阶层化的构成。

[0139] <Java（注册商标）虚拟机 30>

[0140] Java（注册商标）虚拟机 30 将构成应用程序的 xlet 程序加载到工作存储器，解读 xlet 程序，按照解读结果进行对下层的控制。对下层的控制过程是，向中间件发行函数，置换成 BD 再现装置对应的功能调用，向回放控制引擎 32 发行置换后的功能调用。

[0141] <Java（注册商标）虚拟机 30 的内部结构>

[0142] 在此说明 Java（注册商标）虚拟机 30 的内部结构。图 15 是表示，Java（注册商标）虚拟机 30 的内部结构的图。该图所示的 Java（注册商标）虚拟机 30 包括图 12 所示的 CPU24、用户类装载器 52、函数区 53、工作存储器 54、线程 55a、b…n、Java（注册商标）堆栈 56a、b…。

[0143] 用户类装载器 52 从局部内存 26 等读取 BDJA 目录的 Java（注册商标）档案文件中的类文件，并存储到函数区 53。该用户类装载器 52 的类文件读出是通过应用程序管理器 36 向用户类装载器 52 指示指定了文件路径的读取而进行的。如果文件路径表示局部内存 26，用户类装载器 52 从局部内存 26 将构成应用程序的 Java（注册商标）档案文件中的类文件读取到工作存储器 54。如果文件路径表示文件系统上的目录，用户类装载器 52 从 BD-ROM 或局部存储器 18 将构成应用程序的 Java（注册商标）档案文件中的类文件。

[0144] 函数区 53 存储由用户类装载器 52 从局部内存 26 读取的类文件。

[0145] 工作存储器 54 是所谓的堆区域，存储各种各样类文件的对象。在工作存储器 54，存储常驻型的常驻应用程序、和与读取到函数区域 53 的类文件对应的实例。该实例是构成应用程序的 xlet 程序。通过在工作存储器 54 配置这样的 xlet 程序，应用程序成为可执行状态。

[0146] 线程 55a、b…n 是执行存储在工作存储器 54 的函数的逻辑上的执行主体，将存储在局部变量、操作数堆栈的参数作为操作数进行运算，将运算结果存储到局部变量或操作数堆栈。图中的箭头 ky1、ky2、kyn 象征地表示从工作存储器 54 向线程 55a、b…n 的函数供给。物理上的执行主体唯一地是 CPU，作为逻辑上的执行主体的线程可以存在于最大 64 个 Java（注册商标）虚拟机 30 内。在该 64 个这样的数值内，可以新创建线程、也可以删除已有的线程，线程的动作数在 Java（注册商标）虚拟机 30 的动作中可以增减。线程数可以适当增加，所以，可以由多个线程并行地执行一个实例，实现实例的高速化。

[0147] Java（注册商标）堆栈 56a、b…n 与线程 55a、b…n 按一对一的比例存在，在内部具有程序计数（图中的 PC）和一个以上的帧。“程序计数”表示在实例中当前在执行哪一部分。“帧”是向对函数的一次调用分配的堆栈式的区域，包括存储有其一次调用时的参数的“操作数堆栈”和被调用的函数使用的“局部变量堆栈（图中的局部变量）”。帧在每次进行调用时被堆积到 Java（注册商标）堆栈 56a、b…n 上，所以在某个函数递归调用自身时，该帧也被堆积一个。

[0148] 以上是 Java（注册商标）虚拟机的内部结构。这样构成的 Java（注册商标）虚拟机，成为被事件驱动的执行主体（事件驱动的执行主体）。以上说明了 Java（注册商标）虚拟机。

[0149] <演示引擎 (Presentation Engine)31>

[0150] 演示引擎 31 执行 AV 再现功能。再现装置的 AV 再现功能是指，从 DVD 播放机、CD 播放机继承的传统的功能群，是如下的功能：再现开始 (Play)、再现停止 (Stop)、暂停 (Pause On)、解除暂停 (PauseOff)、静音功能的解除 (still off)、带速度指定的快进 (ForwardPlay(speed))、带速度指定的快退 (Backword Play(speed))、声音切换 (Audio Change)、副图像切换 (Subtitle Change)、角度切换 (Angle Change)。为了实现 AV 再现功能，演示引擎 31 控制视频解码器 4、P 图形解码器 6、I 图形解码器 10、音频解码器 16，以便进行读取到读缓冲器 2 上的 AVClip 中的、希望时刻的部分的解码。作为希望时刻，进行 PSR8（目前 PTM）所示的位置的解码，在 AVClip 中可以进行任意时刻的再现。

[0151] <回放控制引擎 32>

[0152] 回放控制引擎 32 执行对播放表的再现控制功能 (i)、PSR 组 23 中的状态取得 / 设定功能 (ii) 这样的各功能。对 PL 的再现控制功能是指，按照当前 PL 信息及 Clip 信息进行演示引擎 31 所进行的 AV 再现功能中的再现开始、再现停止。这些功能 (i) ～ (ii) 根据来自 HDMV 模块 29a ～ BD-J 模块 29b 的功能调用执行。

[0153] 在此，说明回放控制引擎 32 的处理和 Java（注册商标）虚拟机的处理的同期。若功能被调用，回放控制引擎 32 基于 PL 信息执行处理步骤。如果要再现的 AVClip 具有 15 分、30 分的再现时间，则继续这些时间及上述的处理。在此成为问题的是，Java（注册商标）虚拟机 30 返回存取响应的时间、和回放控制引擎 32 实际上结束处理的时间之间的间隔。

Java(注册商标)虚拟机30是事件驱动的处理主体,所以在调用后立即返回表示再现成功、还是再现失败的响应,但是,回放控制引擎32结束AVClip及播放条目的再现是经过15分钟、30分钟后,因此,如果以向应用程序返回存取响应的时间为基准,不能检测出经过15分钟、30分后钟后的处理结束。如果在PL再现中进行快进、快退,该15分钟、30分钟的再现期间前后变动,处理结束的检测更为困难。因此,回放控制引擎32在播放条目的再现结束时、AVClip的再现结束时,向应用程序输出表示播放条目、AVClip的再现结束的事件。通过这样的输出,应用程序可以得知回放控制引擎32结束播放条目或AVClip的再现的时刻。

[0154] 以上是Java(注册商标)虚拟机、网络装置31、回放控制引擎32的说明。对于回放控制引擎32的Java(注册商标)虚拟机的控制通过虚拟包进行。为了实现通过虚拟包的、对回放控制引擎32的控制,在再现装置中存在如下所示的构成要素(网络管理模块37、虚拟文件系统部38、管理信息转换模块39、函数执行模块40)。下面说明这些构成要素。

[0155] <网络管理模块37>

[0156] 网络管理模块37按照来自应用程序的函数调用,从电影作品的提供者运营的三维网下载虚拟包构筑所需的数据。在虚拟包构筑所需的数据中,有流存储位置信息文件、博放表信息、Clip信息、AVClip。当由工作存储器(Work Memory)54上的应用程序发出了的下载请求时,网络管理模块37通过网络下载在虚拟包构筑所需的数据,写入到局部存储器18。

[0157] 该下载的写入步骤是:

[0158] 流存储位置信息文件→播放表信息→多个Clip信息→多个AVClip。

[0159] 在此,在播放表信息-多个Clip信息的写入结束的阶段,可以进行虚拟包的构筑。

[0160] <虚拟文件系统部38>

[0161] 虚拟文件系统部38是属于第2级的构成要素之一,按照来自应用程序的函数调用,构筑虚拟包。在虚拟包的构筑处理中,包含构成虚拟包的各AVClip的状态管理和虚拟包信息的生成处理。

[0162] 1.虚拟包信息

[0163] 虚拟包信息是扩展了BD-ROM中的卷标管理信息的信息。在此,卷标管理信息是规定某记录介质上存在的目录文件结构的信息,包括关于目录的目录管理信息、关于文件的文件管理信息。

[0164] 虚拟包信息是通过在表示BD-ROM的目录文件结构的卷标管理信息中追加新的文件管理信息,实现BD-ROM中的目录文件结构的扩展的信息。在此,追加到BD卷标管理信息的文件管理信息是指,关于存在于局部存储器18的播放表信息、Clip信息、AVClip的文件管理信息,通过生成追加了这样的文件管理信息的虚拟包并传递给回放控制引擎32,回放控制引擎32可以识别为位于局部存储器18上的播放表信息、Clip信息、AVClip存在于BD-ROM上。图16是表示由虚拟文件系统部38进行的虚拟包信息构筑的一例的图。该图的左上是BD_ROM中的目录文件构造,与图2所示的相同。该图的左下是局部存储器18中的目录文件构造,与图13(a)所示的相同。向BD-ROM的卷标管理信息追加关于该局部存储器18上的播放表信息、Clip信息、AVClip的文件管理信息。

[0165] 具体地说,

[0166] i)向BD卷标构成信息中的MPLS路径的路径管理信息,追加关于局部存储器18上

的播放表 (00002.MPLS) 的文件管理信息 ;

[0167] ii) 向 BD 卷标构成信息中的 CLPI 目录的目录管理信息,追加关于局部存储器 18 上的 Clip 信息 #2、Clip 信息 #3、#4(00002.CLPI,00003.CLPI,00004.CLPI) 的文件管理信息 ;

[0168] iii) 向 BD 卷标构成信息中的 STREAM 目录的目录管理信息,追加关于局部存储器 18 上的 AVClip#2、AVClip#3、#4(00002.M2TS,00003.M2TS,00004.M2TS) 的文件管理信息。

[0169] 通过这样做,可得到虚拟包信息。通过该追加得到的容量管理信息是虚拟包信息。

[0170] 向回放控制引擎 32 传递这样生成的虚拟包信息。由此,回放控制引擎 32,可以与 BD-ROM 上的播放表信息、Clip 信息、AVClip 对等地处理局部存储器 18 上的播放表信息、Clip 信息、AVClip。以上是对于虚拟包信息生成的说明。接着说明由虚拟文件系统部 38 进行的状态管理。

[0171] 3. 允许 (enable) 及禁止 (disable)

[0172] “状态管理”是指,应构筑虚拟包的 AVClip 是允许还是禁止的管理。在此,播放表信息中的一个播放条目信息是多角度对象时,构成多角度区间的各个 AVClip 成为是允许还是禁止的状态管理的对象。而且,在播放条目信息中存在子播放条目信息时,该子播放条目信息的记载了 Clip_information_file_name 的 AVClip 也成为是允许还是禁止的状态管理的对象。

[0173] 图 17(a) 是表示由虚拟文件系统部 38 进行的状态管理的一例的图。如该图所示,AVClip 的状态有“允许”、和“禁止”。“允许”是指,Clip 信息和与其对应的 AVClip 存在于局部存储器 18 上,该 AVClip 被设定成只读模式的状态。

[0174] 另一方面,“禁止”是指,虽然 Clip 信息存在于局部存储器 18 上,但与其对应的 AVClip 在局部存储器 18 中不存在 (损失流,MissingStream)、或者即使存在也是可以删除的状态。

[0175] 禁止是作为可删除的内容操作的,在局部存储器 18 上的容量不足时,禁止的 AVClip 成为删除对象。

[0176] 因此,在局部存储器 18 下载了 AVClip 时,只要虚拟文件系统部 38 不改变 AVClip 为允许,回放控制引擎 32 就把该下载的 AVClip 作为不存在的内容来处理。只要没有虚拟文件系统部 38 的作为,回放控制引擎 32 就不识别局部存储器 18 上的 AVClip,所以在不稳定的状态下,回放控制引擎 32 不可能使再现继续进行。

[0177] 3. 只读模式及其取消

[0178] 下面说明只读模式。向只读模式的变化是通过重写局部存储器 18 中的容量管理信息的文件管理信息进行的。该重写对象是局部存储器 18 中的 DLDAT 目录下的 PACKAGE00001 目录中的文件管理信息,通过将这样的文件管理信息中的文件属性全部重写到只读属性,局部存储器 18 上的 AVClip 不会被重写。因此,在回放控制引擎 32 的再现中,即使应用程序要存取局部存储器 18 上的 AVClip,AVClip 也不会被重写,所以保持了再现中的 AVClip 的同一性。

[0179] 使局部存储器 18 上的 AVClip 成为可以从回放控制引擎 32 识别的状态时,在局部存储器 18 中,将存储该 AVClip 的文件置为只读属性,所以 Java(注册商标) 虚拟机上的应用程序即使要直接对局部存储器 18 进行存取,其 AVClip 的内容也不会在中途改变。

[0180] 说明只读模式的取消。只读模式的取消，与向只读模式的变化同样地，通过重写局部存储器 18 中的卷标管理信息的文件管理信息进行的。该重写对象是局部存储器 18 中的 DLDAT 目录下的 PACKAGE00001 目录中的文件管理信息，通过将这样的文件管理信息中的文件管理信息从只读属性改写为可重写属性，局部存储器 18 上的 AVClip 任何时候都能被重写。因此，在回放控制引擎 32 的再现结束后，马上把局部存储器 18 中的文件管理信息改写成可重写属性，局部存储器 18 上的 AVClip 成为删除对象。以上是对状态管理的说明。

[0181] 4. 状态管理和虚拟包信息的关系

[0182] 该状态管理的内容可反映为虚拟包信息的制作。即，在状态管理中，关于被禁止的 AVClip 的文件管理信息，不追加到 BD-ROM 的卷标管理信息，只有关于被允许的 AVClip 的文件管理信息，被追加到 BD-ROM 的卷标管理信息。通过将这样生成的虚拟包信息传递给回放控制引擎 32，回放控制引擎 32 只识别设定为允许的 AVClip。存在于 BD-ROM 上的 AVClip#1、和存在于局部存储器 18 上的 AVClip#2、AVClip#3、AVClip#4 中，如果 AVClip#1、AVClip#2 是允许，AVClip#3、AVClip#4 是禁止，那么虚拟文件系统部 38 如图 18 所示地进行虚拟包信息的生成。图 18 是表示反映了状态管理的虚拟包信息的生成的图。在该图中，关于 AVClip#2 的文件管理信息被追加到 BD 卷标构成信息中的 STREAM 目录的目录管理信息，但是，关于 AVClip#3、AVClip#4 的文件管理信息不被追加到 BD 卷标构成信息中的 STREAM 目录的目录管理信息。在图 18 左下部的四个 AVClip 中，相当于 AVClip#2 的 00002.M2TS 是允许，相当于 AVClip#3、AVClip#4 的 00003.M2TS、00004.M2TS 是禁止。那么，如该图所示，在局部存储器 18 中，相当于 AVClip#2 的 0000.M2TS 的文件属性设定为只读属性。而且，相当于 AVClip#3、AVClip#4 的 00003.M2TS、00004.M2TS 的文件属性设定为可重写属性。通过向回放控制引擎 32 传递这样生成的虚拟包信息，回放控制引擎 32 不会识别 AVClip#3、AVClip#4。

[0183] 在此，虽然 Clip 信息存在，但对应的 AVClip 不在局部存储器 18 上时，其 AVClip 被设定为禁止。但是，虽然对于 Clip 信息的文件路径在流存储位置信息文件中，可是，不仅 AVClip、连 Clip 信息也不在局部存储器 18 时，虚拟包的构筑不可能。即，在虚拟包中本来应有的 Clip 信息不存在，意味着波及整个虚拟包。

[0184] 5. 虚拟包信息的传递

[0185] 说明通过以上的 1. ～ 4. 生成的虚拟包信息如何被传递。

[0186] 在此，如图 18 所示，假设了播放表信息、Clip 信息 #2 ～ Clip 信息 #4、AVClip#2 的下载虽然结束，但 AVClip#3、AVClip#4 的下载没有结束的状态。在这样的状态下若请求虚拟包构筑，虚拟文件系统部 38 进行如 19 所示的动作。图 19 是表示通过网络管理模块 37 进行的下载正在中途时的、虚拟文件系统部 38 的虚拟包构筑的图。在该图的状况下若请求了虚拟包构筑（⊙1 VPCreate），则从 BD-ROM 及局部存储器 18 读取目录管理信息及文件管理信息（⊙2,3），进行虚拟包的构筑，向回放控制引擎 32 传递虚拟包信息（⊙4），并且传递给应用程序（⊙5）。通过这样做，回放控制引擎 32 可以通过虚拟包利用 BD-ROM、及局部存储器 18 上的 AVClip 构筑一个流序列。

[0187] 而且，在传递给应用程序的虚拟包信息中，Clip 信息 #3、Clip 信息 #4 存在而 AVClip#3、AVClip#4 不存在，所以应用程序认为虽然 AVClip#3、AVClip#4 构成数据流列，但它们是不存在的流（损失流）。

[0188] 以上是对于虚拟文件系统部 38 的说明。

[0189] <管理信息转换模块 39>

[0190] 管理信息转换模块 39 是 BD-J 模块 29b 内的一个中间件，在应用程序请求了存在于虚拟包中的播放表信息的实例时，在 Java（注册商标）虚拟机的工作存储器 54 生成 JMF 形式的播放表信息的实例（在图 16 的工作存储器 54 内表示的播放表对象）。

[0191] 该实例（播放表对象）具有作为播放表信息的“数据部”和多个“函数部”。

[0192] 播放表对象中的多个函数，使回放控制引擎 32 进行再现开始 (Play)、再现停止 (Stop)、暂停 (Pause On)、暂停解除 (Pause Off)、静音功能的解除 (still off)、带速度指定的快进 (ForwardPlay(speed))、带速度指定的快退 (Backword Play(speed))、声音切换 (Audio Change)、副图像切换 (Subtitle Change)、角度切换 (Angle Change) 这样的各种再现处理。由此，通过应用程序调用该函数，能够再现由该播放表规定的流序列、或者停止这样的再现。

[0193] 在此，说明由播放表对象再现的播放表信息。该播放表信息是存在于局部存储器 18 上的播放表信息。

[0194] BD-ROM 上的播放表信息只参照 BD-ROM 上的 AVClip，相对于此，局部存储器 18 上的播放表信息被称为“前进播放表信息”，参照 BD-ROM 上的 AVClip 和局部存储器 18 上的 AVClip。

[0195] 在此，假设前进播放表信息由播放条目信息 #1 ～播放条目信息 #5 这样的五个播放条目信息构成。在其中的三个播放条目信息 #1 ～播放条目信息 #3 参照 BD-ROM 上的 Clip 信息，剩余两个（播放条目信息 #4、播放条目信息 #5）参照局部存储器 18 上的 Clip 信息的时，该前进播放表信息如图 20(a) 所示地可以利用 BD-ROM 上的 AVClip 和局部存储器 18 上的 AVClip 定义一个流序列。

[0196] 图 20(a) 是表示由存储在局部存储器的 PL 信息定义了怎样的播放表再现时间轴的图。第 1 层表示记录在 BD-ROM 的 AVClip 中的再现时间轴，第 2 层表示由存储在局部存储器的 PL 信息定义的播放表再现时间轴。第 3 层表示存储在局部存储器的 AVClip#2 的再现时间轴，第 4 层表示存储在局部存储器的 AVClip#3 中的再现时间轴，第 5 层表示存储在局部存储器的 AVClip#4 中的再现时间轴。

[0197] 播放表信息中的播放条目中，播放条目信息 #4、播放条目信息 #5、播放条目信息 #6 指定 AVClip#2、AVClip#3、AVClip#4 作为再现区间，前进播放表信息可以将 BD-ROM 上的 AVClip 及局部存储器 18 上的 AVClip 规定为一个流序列。

[0198] 在此，前进播放表信息的“顺序”是指，在由构成播放表信息的播放条目信息参照的 AVClip 中，若一部分是允许，即使其余的是禁止，也可以开始流序列的再现。

[0199] 图 20(b) 是表示前进播放表信息的概念的图。该图的第 2 层～第 5 层表示图 20(a) 的第 2 层、第 3 层、第 4 层、第 5 层。在图 20(a) 的第 3 层、图 20(b) 的第 3 层的 AVClip 是允许，第 4 层～第 5 层的 AVClip 是禁止时，由前进播放表信息参照的四个 AVClip 中，虽然 AVClip#3、AVClip#4 是不能再现的，但 AVClip#1、AVClip#2 可以再现，所以，可以开始流序列的再现。这样，如果是一部分的 AVClip 则可以投机地开始再现，这是前进播放表信息的特征。

[0200] 在虚拟包构筑由虚拟文件系统部 38 进行的阶段，工作存储器 54 上的应用程序请

求了关于如图 20 那样的前进播放表信息的实例生成时，管理信息转换模块 39 进行如图 21 所示的处理。图 21 是表示由管理信息转换模块 39 进行的实例生成的图。

[0201] 当应用程序调用请求存在于虚拟包的播放表信息的实例生成的生成函数时（⊙ 1），管理信息转换模块 39 从脚本存储器 25 读取播放表信息（⊙ 2），在工作内存 54 生成将该前进播放表信息作为数据部，并附加了函数部的播放表对象（⊙ 3）。由于这样的播放表对象可以在工作内存 54 得到，所以通过应用程序调用该播放表对象的函数，可以向回放控制引擎 32 命令由播放表信息规定的流序列的再现。

[0202] 图 22 是表示播放表对象中的 Play 函数的调用的图。通过该 Play 函数的调用（⊙ 4），对回放控制引擎 32 进行 PL 再现的功能调用（⊙ 5）。这样，回放控制引擎 32 读取由请求了再现的播放表信息参照的 AVClip(BD-ROM 上的 AVClip#1、局部存储器 18 上的 AVClip#2)（⊙ 6,7），依次供给解码器。以上是对于管理信息转换模块 39 的说明。

[0203] ＜函数执行模块 40＞

[0204] 函数执行模块 40 是 BD-J 模块 29b 内的中间件中的一个构成要素，按照来自存在于 Java（注册商标）虚拟机 30 的应用程序的函数调用，执行各种函数。在由函数执行模块 40 执行的函数中，有各种各样的函数，但 getMissingStreams 函数、enableClip 函数、disableClip 函数、getDisableClips 函数是本申请的特征。下面，说明这些函数。

[0205] 1. getMissingStreams 函数

[0206] getMissingStreams 函数是指，在由网络管理模块 37 进行的下载正在继续时，向虚拟文件系统部 38 询问是否还存在成为损失流的 AVClip 的函数。

[0207] 在此，假设如图 19 所示的虚拟包信息被传递给工作内存 54 上的应用程序。在虚拟包构筑时判明的损失流是 AVClip#3、AVClip#4，之后，应用程序调用 getMissingStreams 函数，由此向虚拟文件系统部 38 询问了最新的损失流。此时的函数执行模块 40 的处理如图 22 所示。

[0208] 图 23 是模式地表示该询问的结果的图。在该图中如⊙ 1 所示，进行 getMissingStreams 函数的调用，通过该调用，虚拟文件系统部 38 作为返回值返回 AVClip#4。如果这样，在调用 getMissingStreams 函数之前，损失流是 AVClip#3、AVClip#4，在 getMissingStreams 函数的调用后成为 AVClip#4，这样可以判定为 AVClip#3 不是损失流（下载已结束）。通过这样做，随着下载的进行，可以知道损失流解决到什么程度。

[0209] 2. enableClip 函数

[0210] enableClip 函数是指，在虚拟包的构筑时是损失流，但是在其后，当存在不是损失流的 AVClip 时，用于设定该 AVClip 为允许的函数。

[0211] 在此，明确了在图 23 中 AVClip#3 不是损失流，所以调用将该 AVClip#3 作为对象的 enableClip 函数。此时，函数执行模块 40 进行如图 24 所示的处理。图 24 是模式地表示 enableClip 函数的调用的图。当进行以该 AVClip#3 为对象的 enableClip 函数的调用时（⊙ 1），应用程序将 AVClip#3 变更为允许，向应用程序返回 Success 响应（⊙ 2）。通过这样做，可以在虚拟包的构筑中将不能成为损失流的 AVClip 变更为允许。设 AVClip 为允许时，函数执行模块 40 向应用程序及回放控制引擎 32 传递包含新设为允许的 AVClip 的虚拟包信息。由此，回放控制引擎 32 及应用程序可以识别并存取新成为允许的 AVClip。

[0212] 3. disableClip 函数

[0213] disableClip 函数是指，把处于允许的 AVClip 设置为禁止的函数。在此，图 20 所示的前进播放表信息的再现中，表示 AVClip#2 的再现结束的事件由回放控制引擎 32 通知。此时，若通知了 AVClip#2 的再现已结束，进行如图 25 的处理。图 25 是模式地表示 disableClip 函数的调用的图。在该图中，若伴随回放控制引擎 32 的再现的进行，如果由 end of Clip 事件通知了 AVClip#2 的再现已结束（⊙ 1），通过调用以该 AVClip#2 为对象的 disableClip 函数（⊙ 2），应用程序可以把 AVClip#3 改变为禁止。通过这样做，可以把再现已结束的 AVClip 变更为禁止。通过这样做，AVClip#2 被设定为禁止，所以 AVClip#2 成为删除对象。在构成预告篇的 AVClip 时时刻刻被下载时，只要把由前进播放表信息规定的流序列中的、再现已结束的依次设置为禁止，则可以将再现已结束的 AVClip 设为禁止而成为删除对象。把 AVClip 设为禁止时，函数执行模块 40 向应用程序及回放控制引擎 32 传递除去了新设置为禁止的 AVClip 之外的虚拟包信息。由此，回放控制引擎 32 不会识别成为禁止的 AVClip。而且，在删除局部存储器 18 上的 AVClip 时，中间件只检验要删除的应用程序的写入权限，然后执行其删除。这是为了防止因不注意的删除引起的 AVClip 的丢失。

[0214] 4. getDisableClips 函数

[0215] getDisableClips 函数是询问是否存在成为禁止的 AVClip 的函数。通过该函数的调用，应用程序使用该函数可以得到目前是否有成为禁止的 AVClip 的信息。

[0216] 作为使用例，例如在前进播放表再现中接受了来自用户的章跳越请求时，应用程序使用该函数判断跳跃对象的 AVClip 是否是禁止。并且，若是禁止，不进行章跳跃，显示“正在下载，还不能跳跃到其它章”，唤起注意。通过这样做，可以在前进播放表的再现中执行称为章跳跃的特殊再现。

[0217] 通过以上说明，结束关于由函数执行模块 40 执行的函数的说明。

[0218] 参照流程图详细说明之前的各构成要素的处理顺序。

[0219] 图 26 是表示虚拟包构筑时的处理的流程图。在该流程图中，先判断与存储在流存储位置信息文件中的 AVClip 对应的 Clip 信息在局部存储器 18 中是否完备（步骤 S1），若在 Clip 信息中有不完备的，对于从应用程序的虚拟包调用，进行错误返回（步骤 S10）。即使对于 Clip 信息的文件路径存在于流存储位置信息文件，AVClip 甚至 Clip 信息都不在局部存储器时，虚拟包的构筑不可能。

[0220] 另一方面，如果记述在流存储位置信息文件中的所有 Clip 信息存在于局部存储器 18（步骤 S1 的“是”），向 BD 卷标管理信息中的播放表目录的目录管理信息追加局部存储器 18 中的播放表信息的文件管理信息（步骤 S2），向 BD 卷标管理信息中的 CLPINF 目录的目录管理信息追加局部存储器 18 中的 Clip 信息的文件管理信息（步骤 S3）。之后，执行步骤 4 ～步骤 S8 的循环处理。该循环处理是，对存在于局部存储器 18 的各个 Clip 信息，重复步骤 S6 ～步骤 S8 的处理的处理（步骤 S4、步骤 S5）。在此，把成为循环处理的对象的一个 Clip 信息作为 Clip 信息 x。

[0221] 在该循环处理中，确定对应 Clip 信息 x 的 AVClip（步骤 S6），判断该 AVClip 是允许还是禁止（步骤 S7）。若是允许，向 BD 卷标管理信息的 STREAM 目录的目录管理信息，追加局部存储器 18 中的 AVClipx 的文件管理信息（步骤 S8）。另一方面，若 AVClip 是禁止，跳过步骤 S8。如果对于所有 Clip 信息及 AVClip 重复以上的处理，关于允许的 AVClip 的文件管理信息就被追加到 BD 卷标管理信息上。通过这样的追加得到的 BD 卷标管理信息成

为虚拟包信息。向虚拟包调用的调用源的应用程序及回放控制引擎 32 传递该虚拟包信息（步骤 S9），结束处理。

[0222] 图 27(a) 是表示 enableClip 函数的调用时的处理顺序的流程图。若进行该函数的调用，根据调用时的参数确定应设为允许的 AVClip y（步骤 S14），并且依次进行步骤 S15、步骤 S16 的判断。步骤 S15 是判断 AVClip y 是否存储在局部存储器 18 的判断，步骤 S16 判断 AVClipy 的文件路径是否被记载在流存储位置信息文件。若这些判断步骤的某一个是“否”，就向调用源返回“错误”，结束该流程图的处理（步骤 S19）。另一方面，如果步骤 S15- 步骤 S16 的双方是“是”，就将 AVClip y 设为允许，在局部存储器 18 中，将存储了 AVClip y 的文件的文件属性设为只读，将 AVClipy 设为只读模式（步骤 S17）。之后，把“成功”作为返回值返回（步骤 S18）。

[0223] 图 27(b) 是表示 disableClip 函数的调用时的处理顺序的流程图。若进行了该函数的调用，根据调用时的参数确定应成为禁止的 AVClipy（步骤 S21），并且依次进行步骤 S22、步骤 S23 的判断。步骤 S22 判断 AVClip y 是否是再现中，步骤 S23 判断 AVClip y 的文件路径是否记载在流存储位置信息文件中。若这些判断步骤的某一个是“是”，向调用源返回“错误”，结束该流程图的处理（步骤 S26）。另一方面，如果步骤 S22- 步骤 S23 的双方是“是”，将 AVClip y 设为禁止，在局部存储器 18 中，把存储了 AVClipy 的文件的文件属性设为重写属性，取消 AVClipy 的只读模式（步骤 S24）。之后，把“成功”作为返回值返回（步骤 S25）。

[0224] 图 28 是表示回放控制引擎 32 的再现顺序的流程图。该流程图是，以播放表信息中的起始播放条目信息作为播放条目，执行步骤 S33 ～步骤 S42 的循环处理。步骤 S33 ～步骤 S42 的循环处理的控制变量是变量 i，执行步骤 S33 ～步骤 S40，之后，在变量 i 超过播放条目值（播放条目值的序号）之前执行控制变量 I 加 1 的处理（步骤 S41）。

[0225] 说明步骤 S32 ～步骤 S40 的处理。这些处理中，把记述在播放条目信息 i 的 Clip_information_file_name 的 AVClip 作为 AVClip（步骤 S33），成为再现的对象。步骤 S34 判断 AVClip 是否是允许，若 AVClip 不是允许，在该时刻中止本流程图的处理。若是允许，向驱动器装置及解码器发出指示，再现 AVClipj 中的从 PlayItem.In_time 到 PlayItem.Out_time 的部分（步骤 S35）。

[0226] 步骤 S36 判断把播放条目信息指定为 Sync_PlayItem_id 的子播放条目 k 是否存在。若不存在，直接转移到步骤 S41。若存在，将记述在子播放条目 k 的 Clip_information_file_name 的 AVClip 设为 AVClip h（步骤 S38），判断其 AVCliph 是否是允许（步骤 S39）。若是允许，向驱动器装置及解码器发出指示，以便再现其 AVCliph 中的从 Sync_Start_PTS_of_PlayItem 到 Out_time，转移到步骤 S41（步骤 S40）。

[0227] 如果其 AVCliph 不是允许，则跳过步骤 S40。通过对构成播放表信息的所有播放条目信息重复以上的处理，进行由播放表信息规定的流序列的再现。图 29 是表示应用程序的处理顺序的流程图。首先，请求流存储位置信息和播放表信息、Clip 信息、AVClip 的下载（步骤 S1）。然后进行虚拟包的构筑请求（步骤 S52）。步骤 S53 判断构筑请求是否是成功了，在虚拟包的构筑成功之前，虚拟包的构筑请求重复多次。

[0228] 若虚拟包的构筑成功，执行步骤 S55。步骤 S55 判断损失流是否存在于虚拟包信息中。具体地说，在 Clip 信息存在但对应的 AVClip 不存在时，将该 AVClip 作为损失流取得

(步骤 S56)。

[0229] 之后,在步骤 S57 调用前进播放表信息的 Play 函数,然后进行 Play 函数是否是成功的判定(步骤 S58)。若是错误,结束该流程图的处理。若是成功,执行由步骤 S59 ～步骤 S61 构成的循环处理。

[0230] 步骤 S59 是 getMissingStreams 函数的调用。步骤 S60 是在步骤 S59 的调用,若被通知了损失流,判断在以前取得的损失流中是否存在下载已结束的。

[0231] 步骤 S61 判断是否发生了表示 AVClip 的再现结束的事件。应用程序在进行回放控制引擎 32 的再现期间,重复执行该步骤 S59 ～步骤 S61 的处理。

[0232] 在上述处理的重复中,若有结束了下载的 AVClip(在步骤 S60 的“是”),对其下载结束的损失流调用 enableClip 函数(步骤 S62)。通过这样做,可以早期地将下载已结束的 AVClipe 设为允许。

[0233] 而且,若有结束再现的 AVClip(在步骤 S61 “是”),判定其 AVClip 是否是预告篇,若是预告篇,对于其 AVClip,调用 disableClip 函数。根据这样做,可以早期地将结束再现的预告篇的 AVClip 为 disable。

[0234] <动作>

[0235] 说明再现图 20 所示的播放表信息时的再现装置的动作。图 30 ～图 33 是表示由网络管理模块 37 进行的下载进度和由回放控制引擎 32 进行的再现进度的图。第一层表示由网络管理模块 37 进行的下载的进度,第二层表示由回放控制引擎 32 进行的再现的进度。第三层是现状的时间轴。第四层表示由应用程序进行的函数的调用。

[0236] 如第一层所示,由网络管理模块 37 进行的下载包括图 20 所示的播放表信息的下载、Clip 信息 #1 ～ Clip 信息 #4 的下载、AVClip#1 ～ AVClip#4 的下载。

[0237] 图 30(a) 的目前时刻＝时刻 t0

[0238] 时刻 t0 是播放表信息、Clip 信息 #1 ～ Clip 信息 #4 的下载结束的时刻。因此,虚拟包的构筑成为可能。而且,虚拟包中的播放表信息参照 AVClip#1,该 AVClip#1 至少是允许,所以流序列的再现从 t0 开始。

[0239] 图 30(b) 的目前时刻＝时刻 t1

[0240] 时刻 t1 是由播放表 #1 参照的 AVClip#1 的再现途中,但其是 AVClip#2 的下载结束的时刻。由于 AVClip#2 的下载已结束,所以 AVClip#2 不是禁止。因此,应用程序通过调用 enableClip 函数,可以将 AVClip#2 置为允许。通过该调用,相当于 AVClip#2 的播放条目 #4 成为再现对象。

[0241] 图 31(a) 的目前时刻＝时刻 t2

[0242] 时刻 t2 是由播放表 #1 参照的 AVClip#2 的再现途中,但其是 AVClip#3 的下载已结束的时刻。由于 AVClip#3 的下载已结束,所以 AVClip#3 不是禁止。因此,应用程序通过调用 enableClip 函数,可以把 AVClip#3 置为允许。通过该调用,相当于 AVClip#3 的播放条目 #5 成为再现对象。

[0243] 图 31(b) 的目前时刻＝时刻 t3

[0244] 时刻 t3 是 AVClip#2 的再现结束的时刻。AVClip#2 的再现已结束,AVClip#2 是预告篇,所以由播放条目参照的 AVClip#2 成为删除对象。因此,通过调用将该 AVClip#2 作为对象的 disableClip 函数,将 AVClip#2 设为删除对象。通过这样做,回放控制引擎 32 可以

像 AVClip#2 未曾存在似地进行处理。

[0245] 图 32(a) 的目前时刻＝时刻 t4

[0246] 时刻 t4 是由播放表 #1 参照的 AVClip#3 的再现途中，AVClip#4 的下载结束的时刻。由于 AVClip#4 的下载已结束，所以 AVClip#4 不是禁止。因此，应用程序通过调用 enableClip 函数，可以将 AVClip#4 设为允许。通过该调用，相当于 AVClip#4 的播放条目 #6 成为再现对象。

[0247] 图 32(b) 的目前时刻＝时刻 t5

[0248] 时刻 t5 是 AVClip#3 的再现结束的时刻。AVClip#3 的再现已结束，AVClip#3 是预告篇，所以由播放条目 #5 参照的 AVClip#3 成为删除对象。因此，通过调用将该 AVClip#3 作为对象的 disableClip 函数，将 AVClip#3 作为删除对象。通过这样做，回放控制引擎 32 可以像 AVClip#3 未曾存在似地进行处理。

[0249] 图 33 的目前时刻＝时刻 t6

[0250] 时刻 t6 是 AVClip#4 的再现结束的时刻。AVClip#4 的再现已结束，AVClip#4 是预告篇，所以由播放条目 #6 参照的 AVClip#4 成为删除对象。因此，通过调用将该 AVClip#4 作为对象的 disableClip 函数，将 AVClip#4 作为删除对象。通过这样做，回放控制引擎 32 可以像 AVClip#4 未曾存在似地进行处理。

[0251] 根据如上所述的本实施方式，并行地进行由网络管理模块 37 进行的下载和由回放控制引擎 32 进行的再现，通过把进行了下载的 AVClip 从禁止设为允许，可以提供给回放控制引擎 32 的再现。由此，可以实现推进再现处理的“追赶再现”，以便赶上下载的进度。

[0252] 而且，通过将下载的数字流中的已再现的数据流从允许变更为禁止，可以将已进行了再现的数字流作为删除对象提供，所以，可以实现推进删除处理的“追赶删除”，以便赶上再现处理的进度。

[0253] 由于能够进行推进再现处理的“追赶再现”，以便追赶下载的进度，或者进行推进删除处理的“追赶删除”，以便追赶再现处理的进度，所以，即使在连结再现装置和再现装置的通信线路的通信速率低，且内置在再现装置中的局部存储器 18 的容量小的情况下，也可以把记录在 BD-ROM 的数字流和下载到局部存储器 18 的数据流作为一个流序列，进行连续再现。

[0254] 以上说明并不是表示本发明的所有实施行为的方式。根据实施了下述 (A)(B)(C)(D)……的变更的实施行为的方式，可以实施本发明。本申请的权利要求涉及的各发明是以上记载的多个实施方式及扩展了那些变形方式的记载，甚至一般化的记载。扩展甚至一般化的程度，基于本发明的技术领域的申请时的技术水平的特性。

[0255] (A) 在上述的虚拟文件系统部 38 的说明中，没有提及目录管理信息及文件管理信息的具体数据结构，在 BD-ROM 中的文件系统是由 ISO/IEC13346 规定的情况下，目录管理信息也可以由“目录文件”及“文件入口”构成。目录文件包括记述了位于自身下面的目录的目录名的文件识别描述符、和记述了位于自身下面的文件的文件名的文件识别描述符。该“文件入口”包括描述符标签、ICB 标签、分布描述符长度、扩展属性、分布描述符。

[0256] 文件管理信息也可以由文件入口构成。分布描述符表示构成文件的各范围的记录位置。

[0257] 此时，虚拟包信息的生成过程是，在与 BD-ROM 的卷标管理信息中的 STREAM 目录相

当的目录文件中，追加局部存储器上的 AVClip 的文件名的文件识别描述符，在 BD-ROM 的卷标管理信息中的 STREAM 目录下，追加关于局部存储器上的 AVClip 的文件入口。

[0258] (B) 在所有实施方式中，把记录介质作为 BD-ROM 实施，只要是能够记录 AVClip 及播放表信息的记录介质，也可以是任何记录介质。例如，也可以是 DV-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R、CD-RW 等的光盘，PD、MO 等的磁光盘。而且，也可以是小型闪存（注册商标）卡、智能存储卡 (smart media)、存储棒 (memory stick)、多媒体卡、PCM-CIA 卡等半导体存储卡。也可以是软盘、超级盘 (SuperDisk)、Zip、Clik ! 等的磁记录盘 (i)，ORB、Jaz、SparQ、SyJet、EZFley、微驱动器等的可移动硬盘 (ii)。还可以是，设备内置型的硬盘。

[0259] (C) 所有实施方式中的再现装置，对记录在 BD-ROM 的 AVClip 进行解码后向 TV 输出，可以只把 BD-ROM 驱动器作为再现装置，此外的构成要素由 TV 具备，此时，可以把再现装置和 TV 组装到通过 IEEE1394 连接的家庭网络中。而且，实施方式的再现装置是与电视机连接并利用的类型，但也可以是与显示器成为一体的再现装置。再者，在各实施方式的再现装置中，也可以只把构成处理的本质部分的部分作为再现装置。这些再现装置都是记载在本申请说明书中的发明，所以不论是这些中的哪一个方式，都基于在各实施方式所示的再现装置的内部结构，制造再现装置的行为成为记载在本申请的说明书的发明的实施行为。各实施方式所示的由再现装置的有偿及无偿的转让（有偿时是买卖、无偿时是赠予）、租借、进口的行为也是本发明的实施行为。通过店内展示、目录劝导、传单发布，向一般用户请求这些的转让和租借的行为也是本再现装置的实施行为。

[0260] (D) 利用各流程图所示的程序进行的信息处理，是使用硬件资源具体实现的，在上述流程图中表示了处理顺序的程序是以单体作为发明来成立的。所有实施方式是组装到再现装置的方式，表示了对于本发明涉及的程序的实施行为的实施方式，也可以从再现装置分离，实施各实施方式所示的程序单体。在程序单体的实施行为中，有生产这些程序的行为 (1)，通过有偿及无偿转让程序的行为 (2)，进行租借的行为 (3)，进口行为 (4)，经由双向的电子通信线路向公众提供的行为 (5)，通过店内展示、目录劝导、传单发布，向一般用户请求程序的转让和租借的行为 (6)。

[0261] (E) 把在各流程图中按时间序列执行的各步骤的 [时] 的要素，作为确定发明的必须的事项考虑。这样，可以知道这些流程图的处理顺序公开了再现方法的使用方式。如果通过按时间序列进行各步骤的处理进行这些流程图的处理，以便实现本发明的本来目的，产生作用及效果，则毋庸置疑，相当于本发明涉及的记录方法的实施行为。

[0262] (F) 在各实施方式中，记录在记录介质的数字流是 AVClip，但也可以是 DVD-Video 格式、DVD-Video Recording 格式的 VOB（视频对象）。VOB 是对视频流、音频流进行复用而得到的按照 ISO/IEC13818-1 规格的程序流。而且，AVClip 中的视频流，也可以是 MPEG4 或 WMV 方式。再者，音频流也可以是 Linear-PCM 方式、Dolby-AC3 方式、MP3 方式、MPEG-AAC 方式、Dts、WMA(Windows（注册商标）media audio)。

[0263] (G) 各实施方式中的图像作品也可以是通过对由模拟广播播放的模拟图像信号进行编码而得到的。也可以是由通过数字广播播放的传输流构成的流数据。

[0264] 而且，也可以是对记录在录像带上的模拟 / 数字图像信号进行编码而得到内容。再者，还可以对从摄像机直接读取的模拟 / 数字的图像信号进行编码而得到内容。另外，也

可以是通过发送服务器发布的数字著作物。

[0265] (H)BD-J 模块 29b 也可以是为了卫星广播接收而组装到设备中的 Java(注册商标)平台。若是涉及 BD-J 模块 29b 的 Java(注册商标)平台,则本发明涉及的再现装置兼用作 MHP 用 STB 的处理。

[0266] 再者,也可以是为了便携式电话的处理控制而组装到设备中的 Java(注册商标)平台。若 BD-J 模块 29b 是这样的 Java(注册商标)平台,本发明涉及的再现装置可兼用作便携式电话的处理。

[0267] (I) 在矩阵模式中,也可以在 BD-J 模块之上配置 HDMV 模块。尤其是,HDMV 模式下的动态脚本的解释、和基于动态脚本的控制顺序的执行,对于再现装置的负担轻,所以在 BD-J 上执行 HDMV 模式也不会产生任何问题。而且,在再现装置和电影作品的开发时,动作保证可以用一个模式就可以。

[0268] 再者,可以只在 BD-J 模式下执行再现处理。在 BD-J 模式下也可以进行与 PL 的再现同步的再现控制,所以,基于即使不设置较强的 HDMV 模式也可以的理由。

[0269] (J) 在局部存储器 18 下载的 AVClip,也可以是特别图像、制作者、演员的采访。

[0270] (K) 作为利用应用程序得知某 Clip 是损失流或可重写的哪一状态的方法,在上述实施方式中调用 getMissingStream 函数,但是,也可以使用 java(注册商标).net,java(注册商标).io 等的 java(注册商标)程序库实现。此时,使用以下 a),b),c) 的方法是现实的。

[0271] a) 利用 XML Parser 解析 Binding Unit Manifest File(流位置存储信息)的内容。

[0272] Binding Unit Manifest File 的例:

[0273] <Progressive>

[0274] <ProgressiveAsset VPFilename =”BDMV/STREAM/01000.m2ts”>

[0275] ←识别符

[0276] <BUDAFile name =”1/1/streams/0.m2ts”></BUDAFile>

[0277] ←文件路径

[0278] </ProgressiveAsset>

[0279] <ProgressiveAsset VPFilename =”BDMV/STREAM/01001.m2ts”>

[0280] <BUDAFile name =”1/1/streams/1.m2ts”></BUDAFile>

[0281] </ProgressiveAsset>

[0282] <ProgressiveAsset VPFilename =”BDMV/STREAM/01002.m2ts”>

[0283] <BUDAFile name =”1/1/streams/2.m2ts”></BUDAFile>

[0284] </ProgressiveAsset>

[0285] </Progressive>

[0286] ※”BUDA”是 Binding Unit Data Area 的简称,

[0287] Binding Unit Data Area 是存储在虚拟包利用的文件的局部存储器内的区域

[0288] b) 使用 java(注册商标).io,确认在记载在 Binding UnitManifest File 的文件路径(流存储位置)中是否存在 xxxxx.m2ts。

[0289] 例:

[0290] new java(注册商标).io.File(“1/1/streams/0.m2ts”).exists()；

[0291] //错误时是损失流

[0292] c)下载是否结束的判断是使用java(注册商标).net,java(注册商标).io,确认服务器上的文件大小和局部存储器上的文件大小是否一致。

[0293] 例：

[0294] new java(注册商标).net.URL(http://xxx.com/vp_data/0.m2ts).openConnection().getContentLength() == new java(注册商标).io.File(“1/1/streams/0.m2ts”).length()；

[0295] //若一致,可以认为下载已结束。

[0296] 生产商的利用可能性

[0297] 本发明涉及的再现装置,如在家庭影院系统的利用那样,可以以个人用途利用。但是,本发明在上述实施方式中公开了内部结构,基于该内部可以进行批量生产,所以在资质上可以工业利用。由此本发明涉及的再现装置,具有产业上的利用可能性。

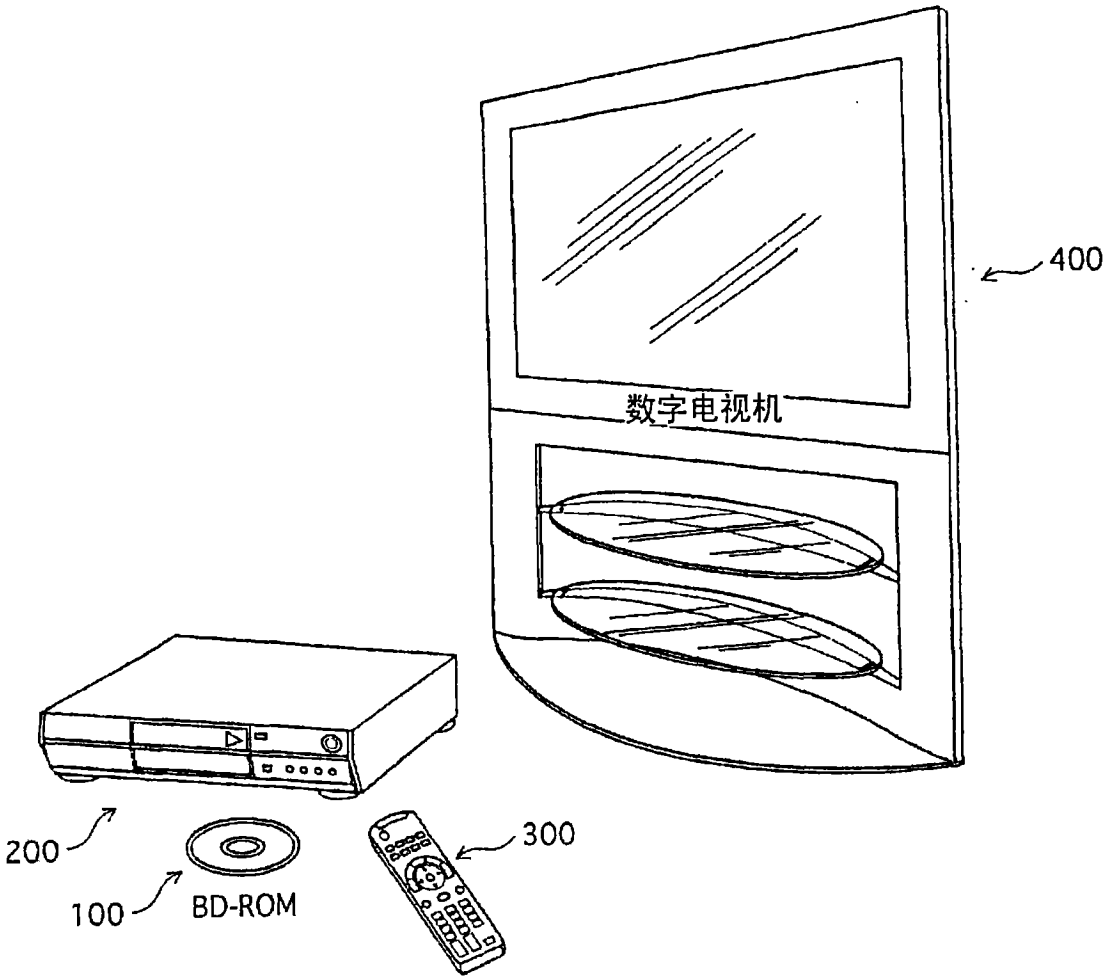


图 1

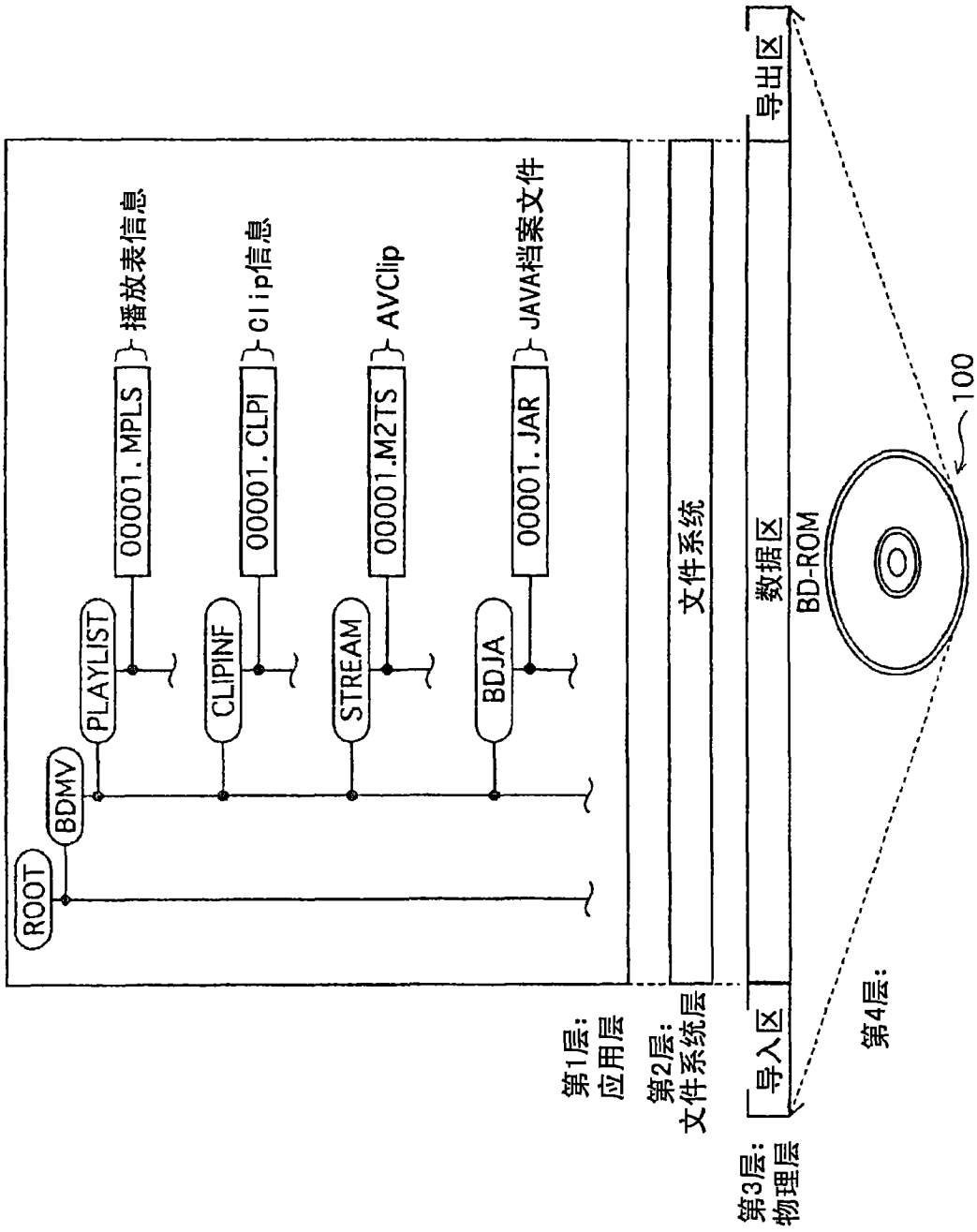

ROOT
BDMV
PLAYLIST
00001.MPLS
播放表信息
CLIPINF
00001.CLPI
clip信息
STREAM
00001.M2TS
AVClip
BDJA
00001.JAR
JAVA档案文件
文件系统
导出区
数据区
BD-ROM
导入区
100
第1层:
应用层
第2层:
文件系统层
第3层:
物理层
第4层:

图 2

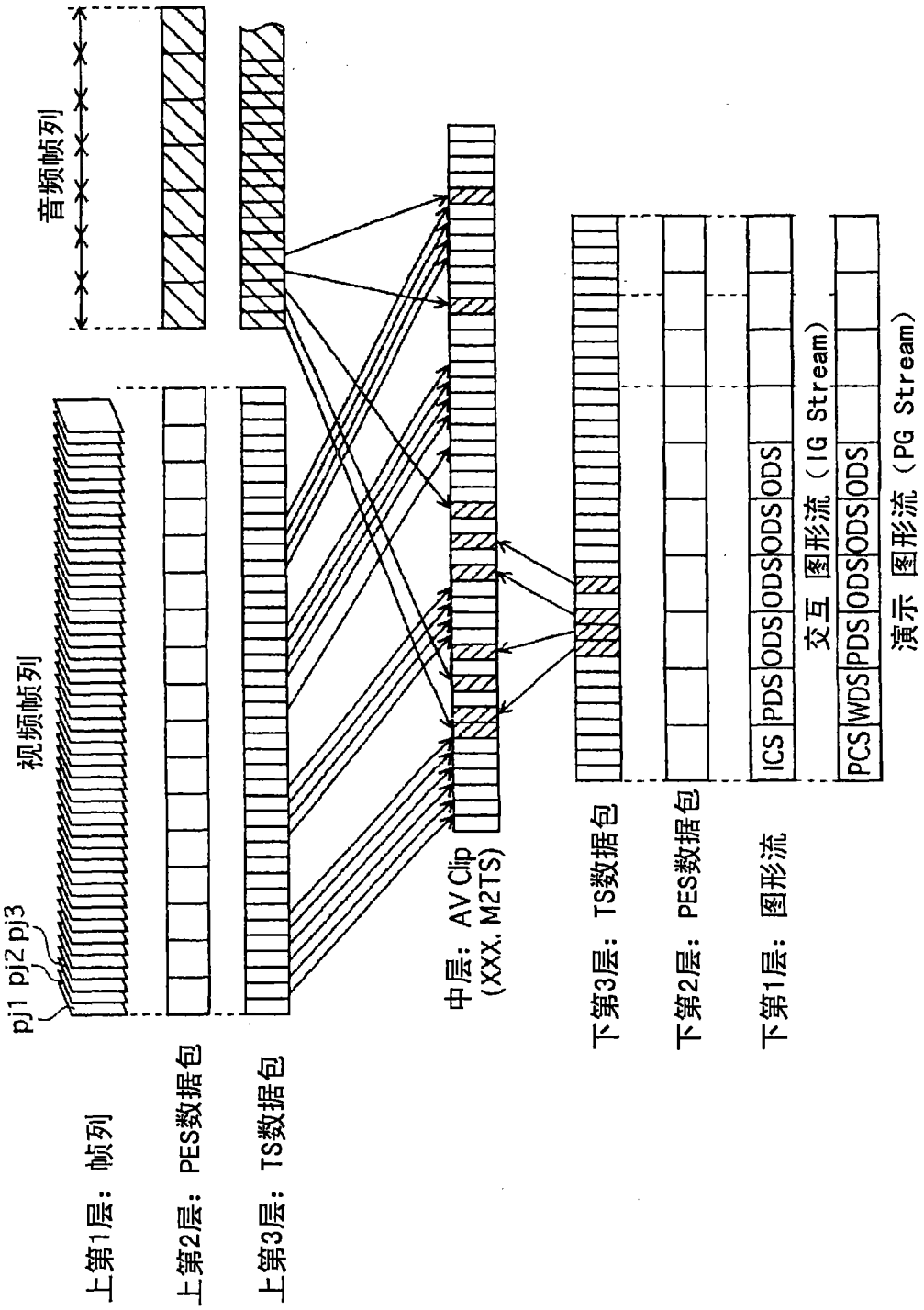

音频帧列
视频帧列
pj1 pj2 pj3
上第1层：帧列
上第2层：PES数据包
上第3层：TS数据包
中层：AV Clip
(XXX. M2TS)
下第3层：TS数据包
下第2层：PES数据包
下第1层：图形流
ICS PDS ODS ODS ODS ODS
交互 图形流（IG Stream）
PCS WDS PDS ODS ODS ODS
演示 图形流（PG Stream）

图 3

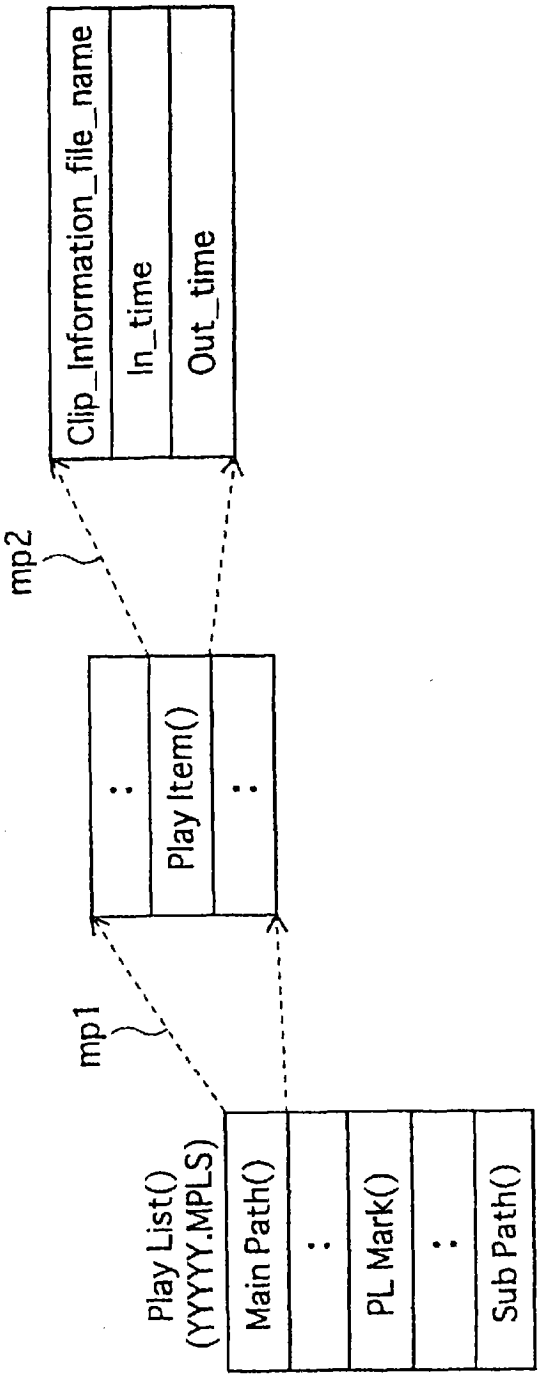

Clip_Information_file_name
In_time
Out_time
mp2
Play Item()
mp1
Play List()
(YYYYY.MPLS)
Main Path()
PL Mark()
Sub Path()

图 4

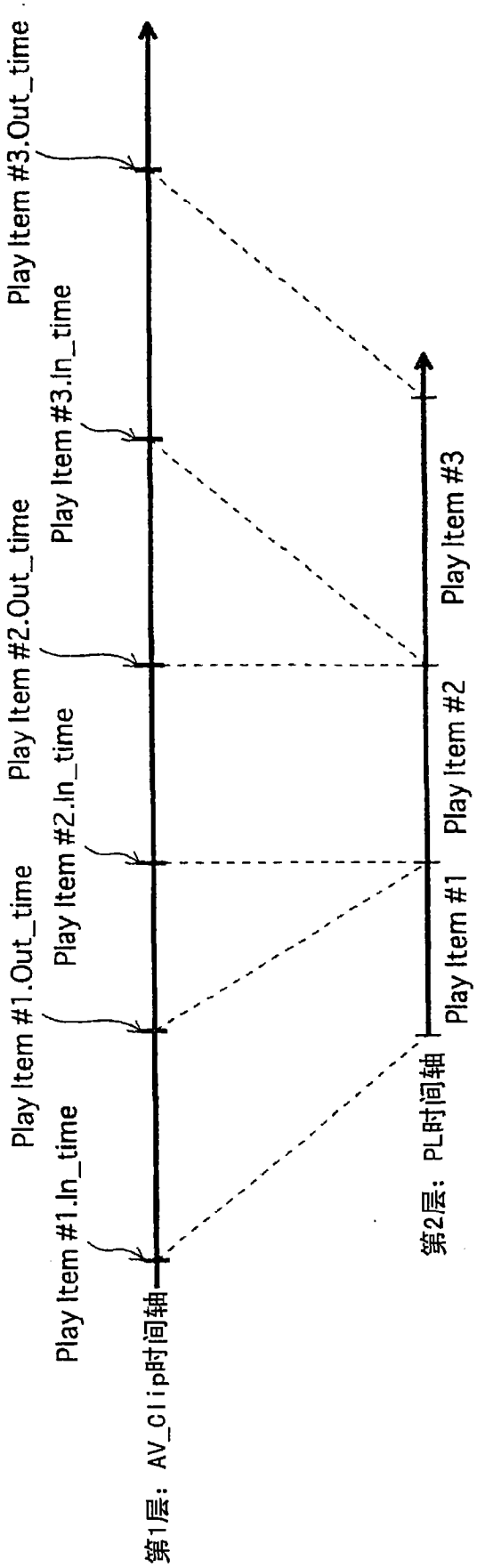

Play Item #3.Out_time
Play Item #3.In_time
Play Item #2.Out_time
Play Item #2.In_time
Play Item #1.Out_time
Play Item #1.In_time
第1层：AV_Clip时间轴
第2层：PL时间轴
Play Item #3
Play Item #2
Play Item #1

图 5

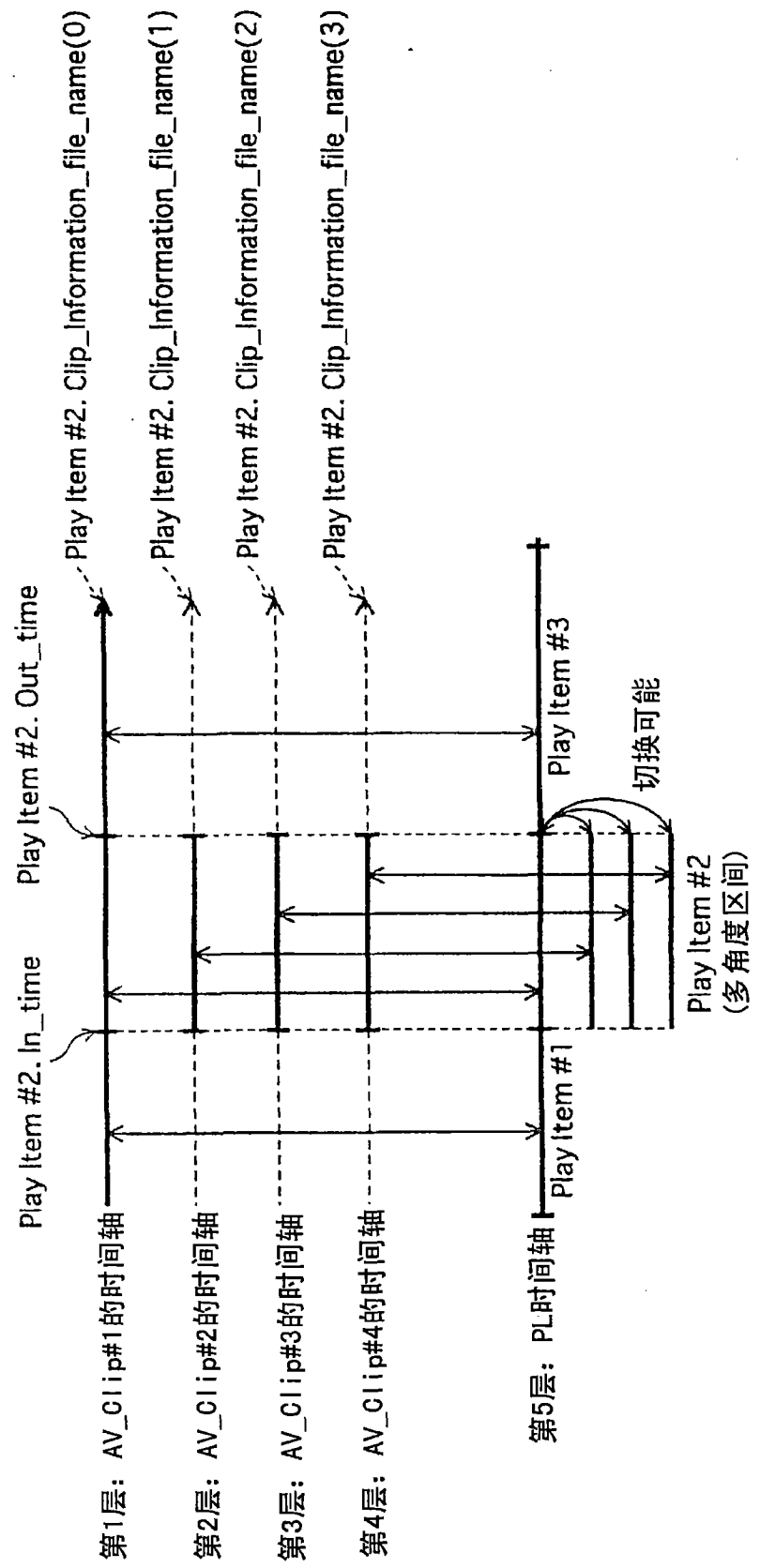

Play Item #2. Clip_Information_file_name(0)
Play Item #2. Clip_Information_file_name(1)
Play Item #2. Clip_Information_file_name(2)
Play Item #2. Clip_Information_file_name(3)
Play Item #2. Out_time
Play Item #2. In_time
第1层：AV_Clip#1的时间轴
第2层：AV_Clip#2的时间轴
第3层：AV_Clip#3的时间轴
第4层：AV_Clip#4的时间轴
第5层：PL时间轴
Play Item #1
Play Item #3
切换可能
Play Item #2
(多角度区间)

图 6

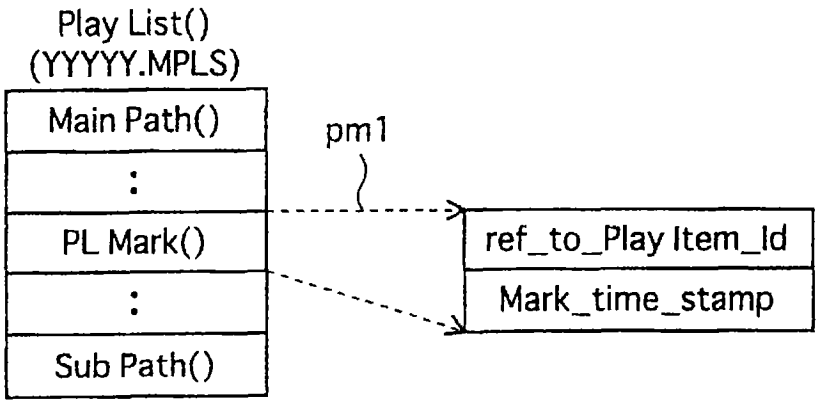

Play List()
(YYYYY.MPLS)
Main Path()
:
PL Mark()
:
Sub Path()
pm1
ref_to_Play Item_Id
Mark_time_stamp

图 7

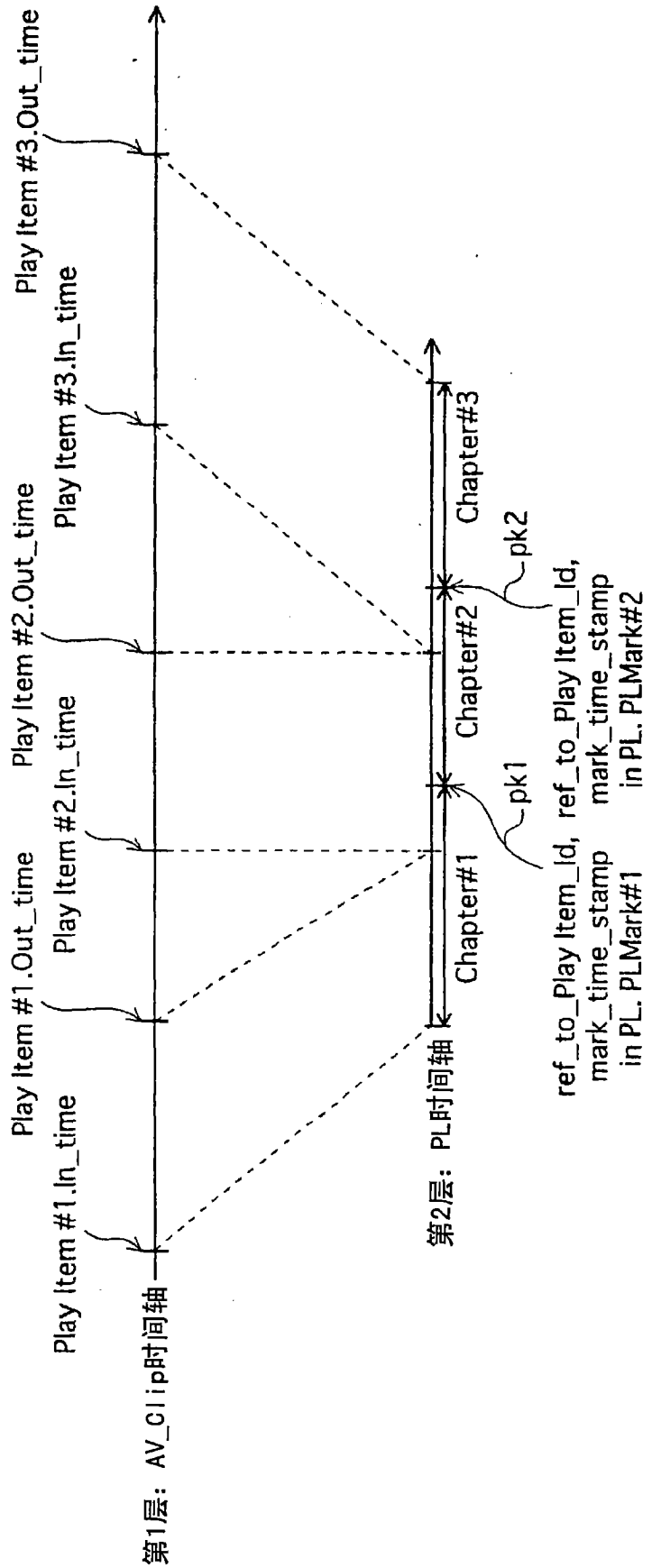

第1层：AV_Clip时间轴
Play Item #1.In_time
Play Item #1.Out_time
Play Item #2.In_time
Play Item #2.Out_time
Play Item #3.In_time
Play Item #3.Out_time
第2层：PL时间轴
Chapter#1
Chapter#2
Chapter#3
pk1
pk2
ref_to_Play Item_Id,
mark_time_stamp
in PL. PLMark#1
ref_to_Play Item_Id,
mark_time_stamp
in PL. PLMark#2

图 8

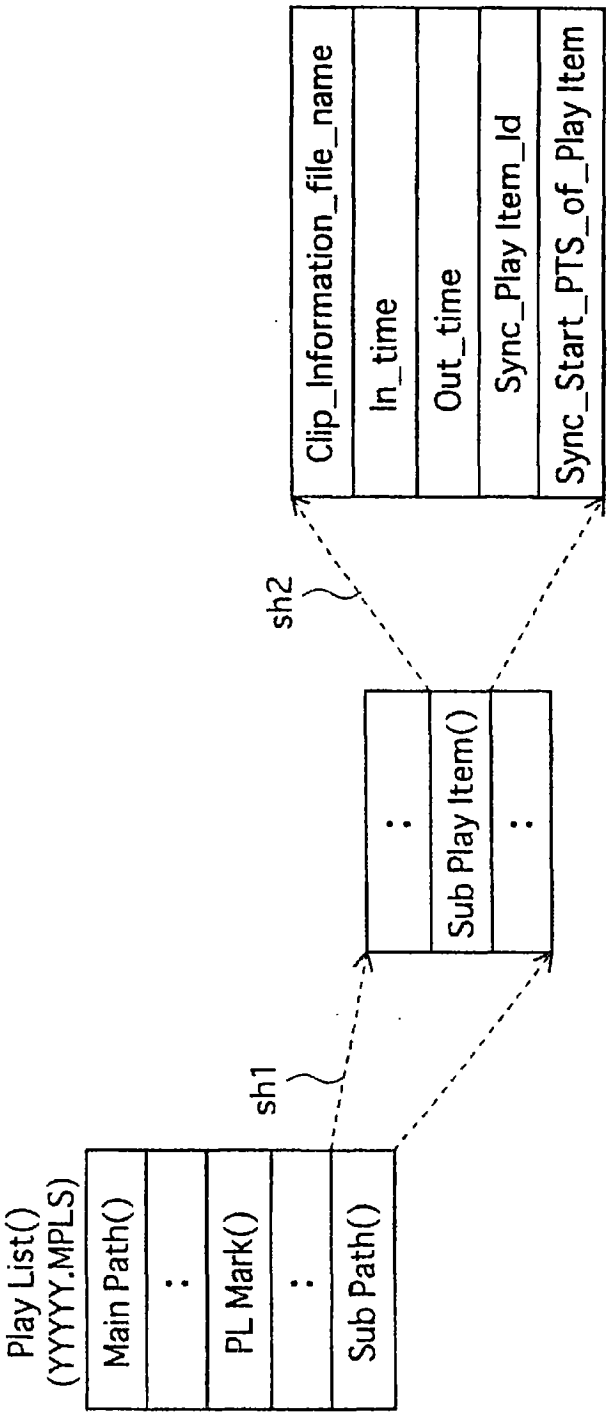

Play List()
(YYYYY.MPLS)
Main Path()
PL Mark()
Sub Path()
sh1
Sub Play Item()
sh2
Clip_Information_file_name
In_time
Out_time
Sync_Play Item_Id
Sync_Start_PTS_of_Play Item

图 9

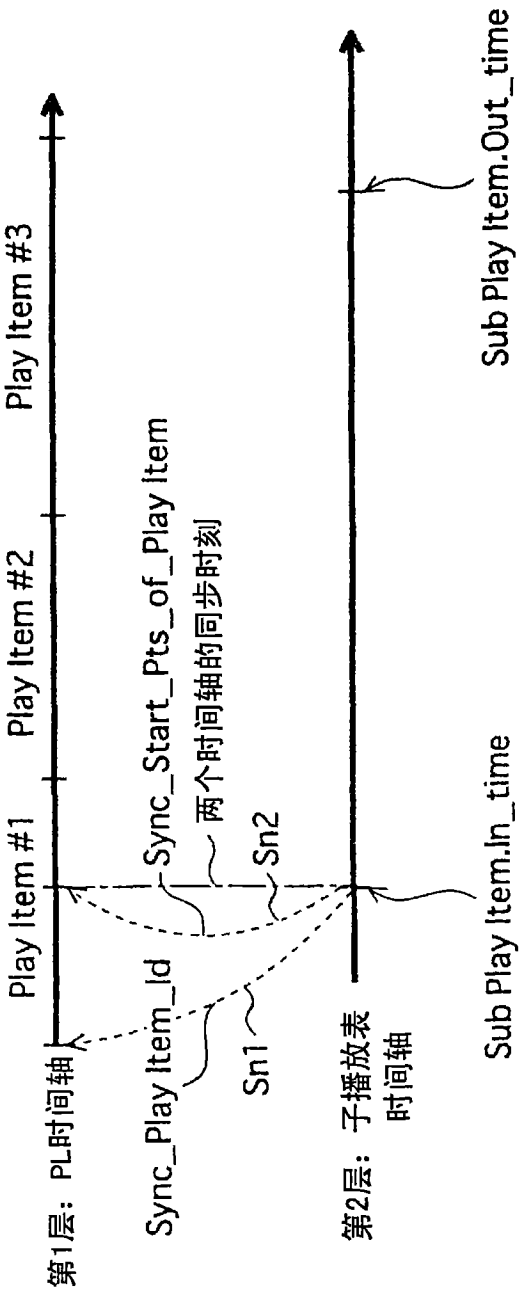

Play Item #1
Play Item #2
Play Item #3
第1层：PL时间轴
Sync_Play Item_Id
Sync_Start_Pts_of_Play Item
两个时间轴的同步时刻
Sn1
Sn2
第2层：子播放表
时间轴
Sub Play Item.In_time
Sub Play Item.Out_time

图 10

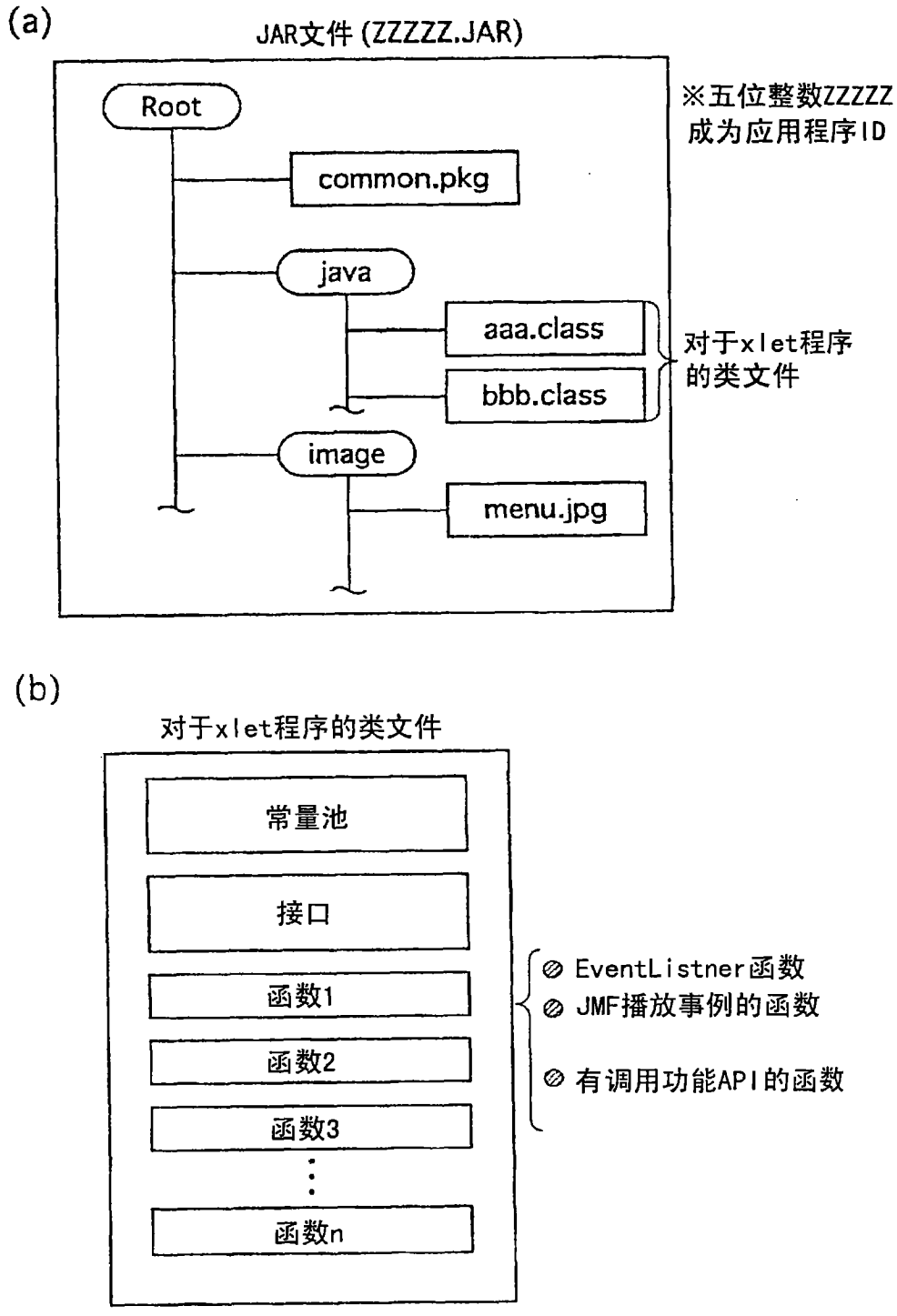

(a)
JAR文件 (ZZZZZ.JAR)
Root
common.pkg
java
aaa.class
bbb.class
image
menu.jpg
※五位整数ZZZZZ
成为应用程序ID
对于xlet程序
的类文件
(b)
对于xlet程序的类文件
常量池
接口
函数1
函数2
函数3
函数n
EventListner函数
JMF播放事例的函数
有调用功能API的函数

图 11

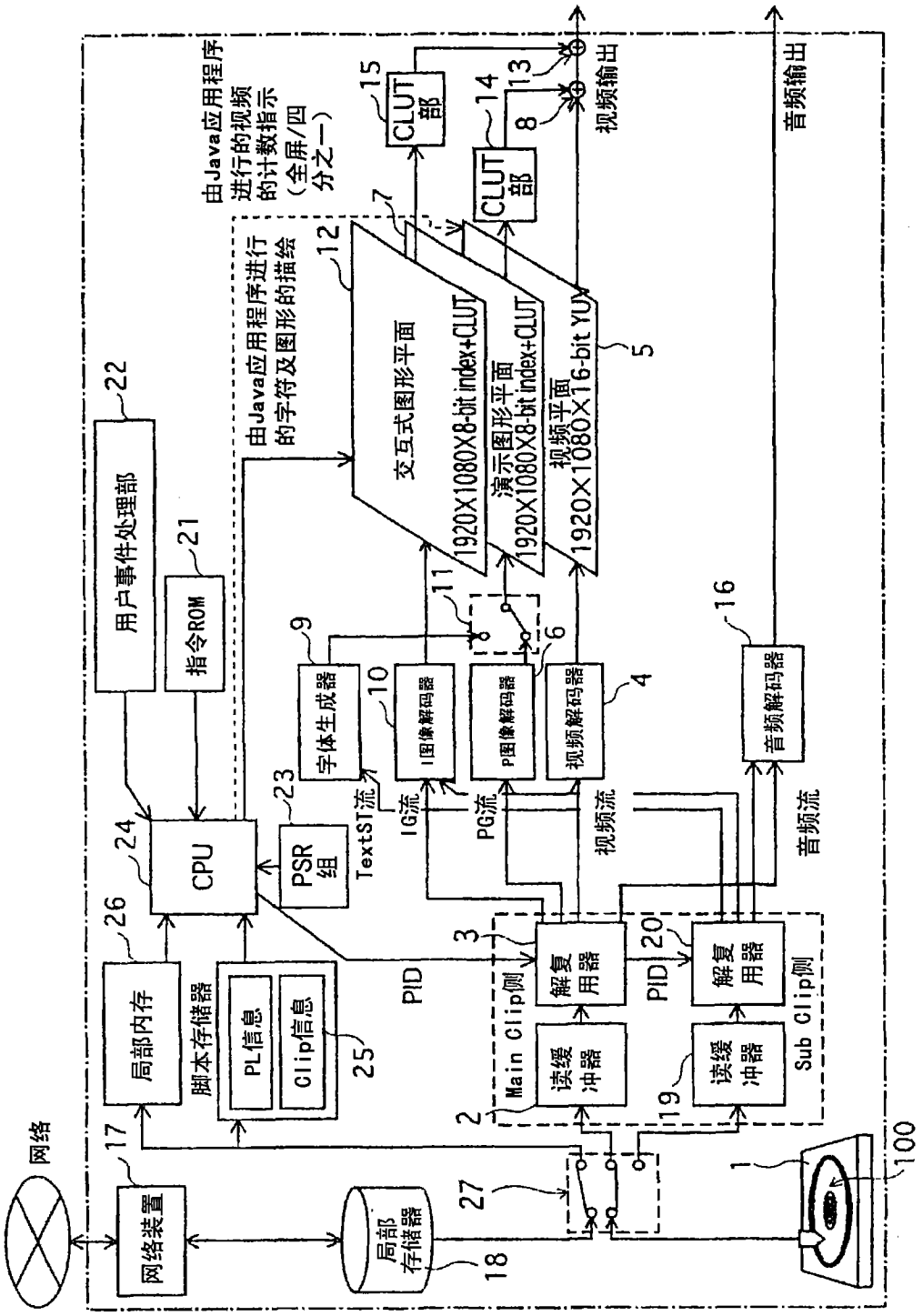

网络
网络装置
局部存储器
局部内存
脚本存储器
PL信息
Clip信息
用户事件处理部
指令ROM
CPU
PSR组
字体生成器
TextST流
IG流
PG流
视频流
音频流
PID
Main Clip侧
Sub Clip侧
读缓冲器
解复用器
I图像解码器
P图像解码器
视频解码器
音频解码器
交互式图形平面
演示图形平面
视频平面
1920X1080X8-bit index+CLUT
1920X1080X8-bit index+CLUT
1920X1080X16-bit YUV
CLUT部
CLUT部
视频输出
音频输出
由Java应用程序进行的计数指示（全屏/四分之一）
由Java应用程序进行的字符及图形的描绘
100

图 12

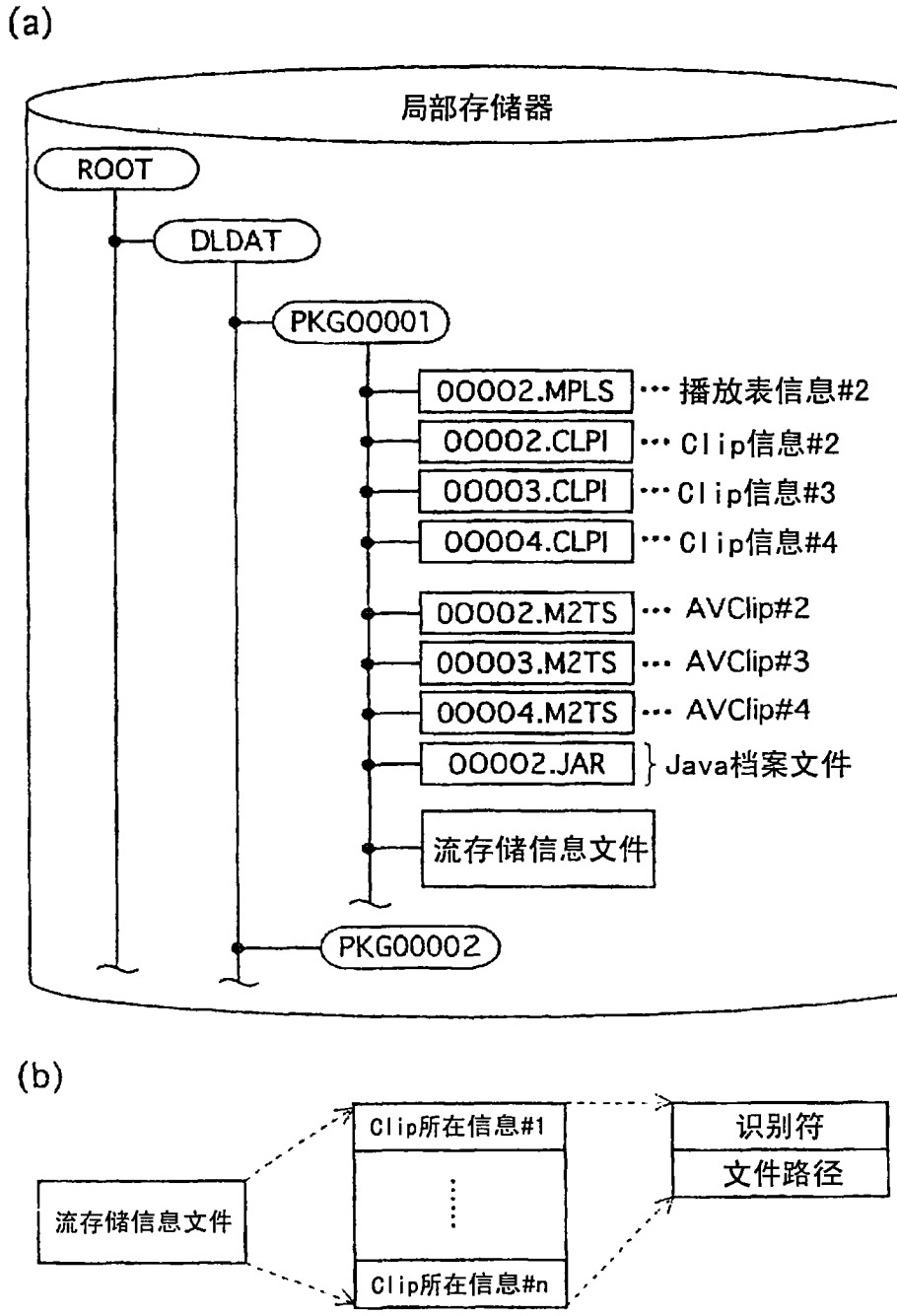

(a)
局部存储器
ROOT
DLDAT
PKG00001
00002.MPLS
播放表信息#2
00002.CLPI
Clip信息#2
00003.CLPI
Clip信息#3
00004.CLPI
Clip信息#4
00002.M2TS
AVClip#2
00003.M2TS
AVClip#3
00004.M2TS
AVClip#4
00002.JAR
Java档案文件
流存储信息文件
PKG00002
(b)
流存储信息文件
Clip所在信息#1
Clip所在信息#n
识别符
文件路径

图 13

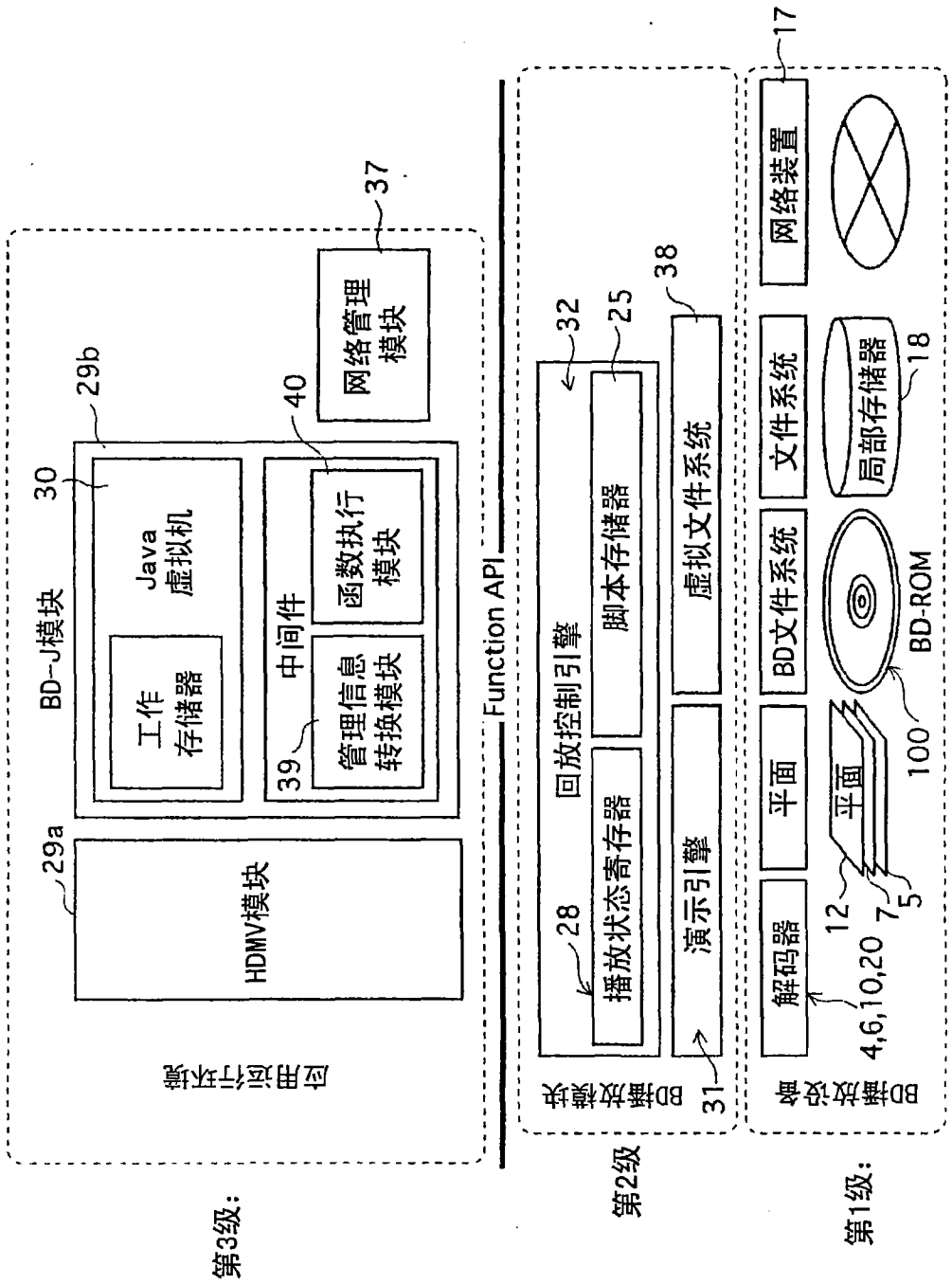

第3级：
第2级
第1级：
应用运行环境
BD播放模块
BD播放设备
BD-J模块
30
29b
工作存储器
Java虚拟机
中间件
40
39
管理信息转换模块
函数执行模块
网络管理模块
37
HDMV模块
29a
Function API
回放控制引擎
32
播放状态寄存器
28
脚本存储器
25
演示引擎
31
虚拟文件系统
38
解码器
平面
平面
12
7
5
4,6,10,20
BD文件系统
文件系统
局部存储器
18
网络装置
17
100
BD-ROM

图 14

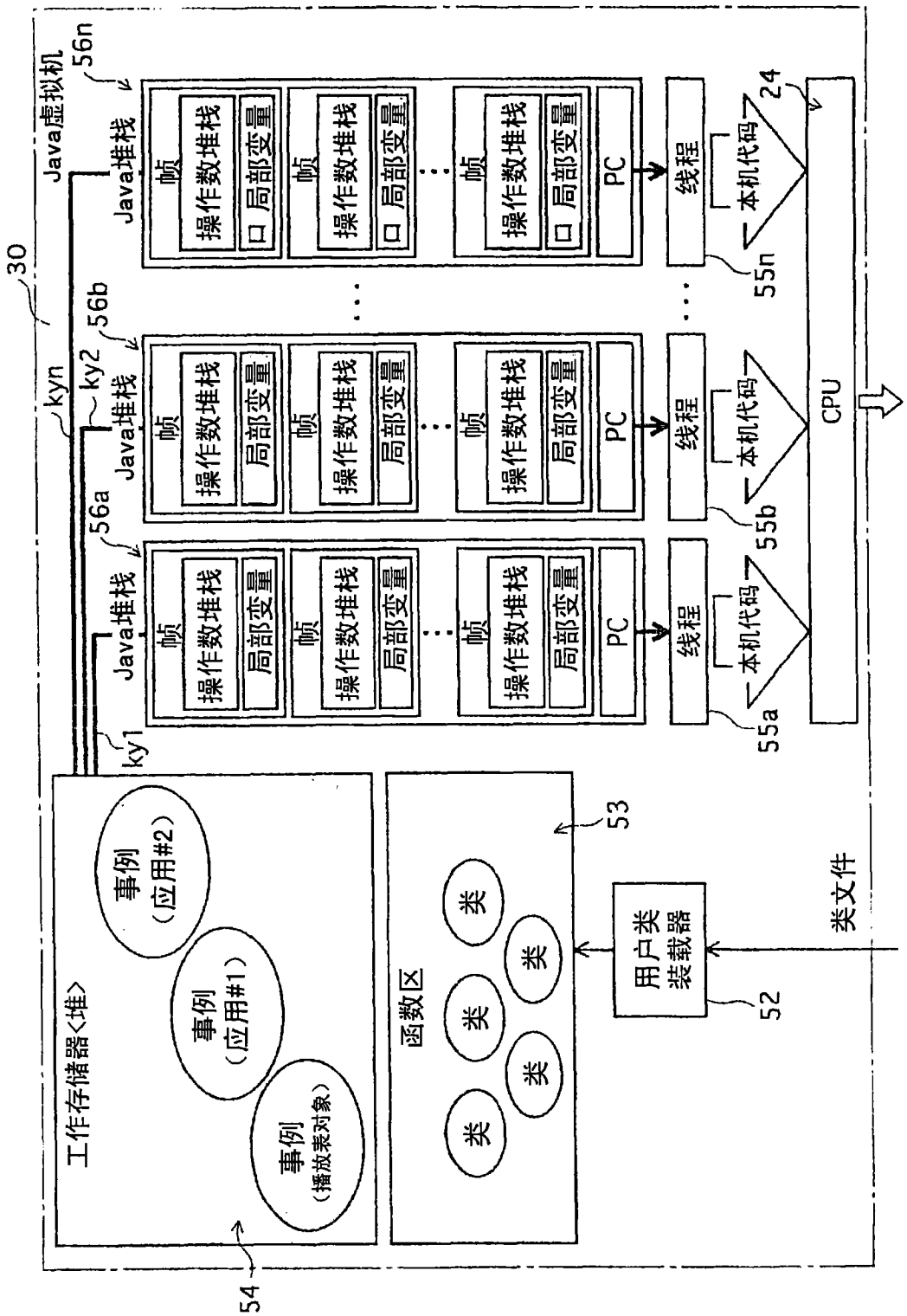

30
Java虚拟机
56n
56b
56a
Java堆栈
帧
操作数堆栈
局部变量
PC
线程
本机代码
24
55n
55b
55a
CPU
kyn
ky2
ky1
工作存储器<堆>
事例（应用#2）
事例（应用#1）
事例（播放表对象）
54
函数区
类
53
用户类装载器
52
类文件

图 15

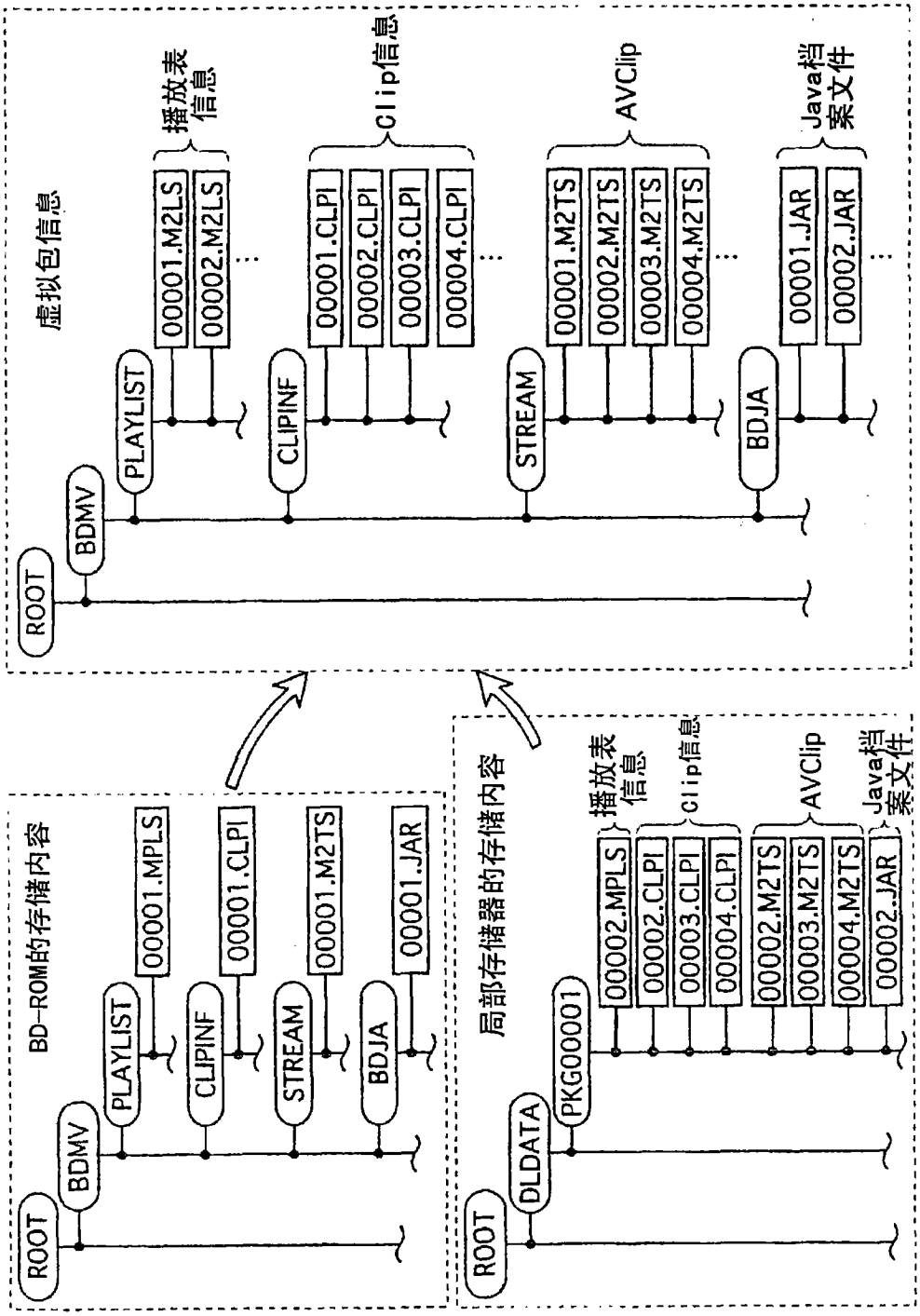

虚拟包信息
播放表信息
Clip信息
AVClip
Java档案文件
ROOT
BDMV
PLAYLIST
CLIPINF
STREAM
BDJA
00001.M2LS
00002.M2LS
00001.CLPI
00002.CLPI
00003.CLPI
00004.CLPI
00001.M2TS
00002.M2TS
00003.M2TS
00004.M2TS
00001.JAR
00002.JAR
BD-ROM的存储内容
00001.MPLS
局部存储器的存储内容
DLDATA
PKG00001
00002.MPLS
00002.CLPI
00003.CLPI
00004.CLPI
00002.M2TS
00003.M2TS
00004.M2TS
00002.JAR

图 16

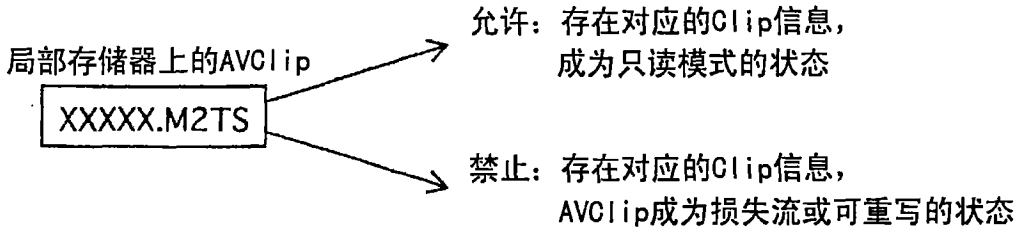

局部存储器上的AVClip
XXXXX.M2TS
允许：存在对应的Clip信息，
成为只读模式的状态
禁止：存在对应的Clip信息，
AVClip成为损失流或可重写的状态

图 17

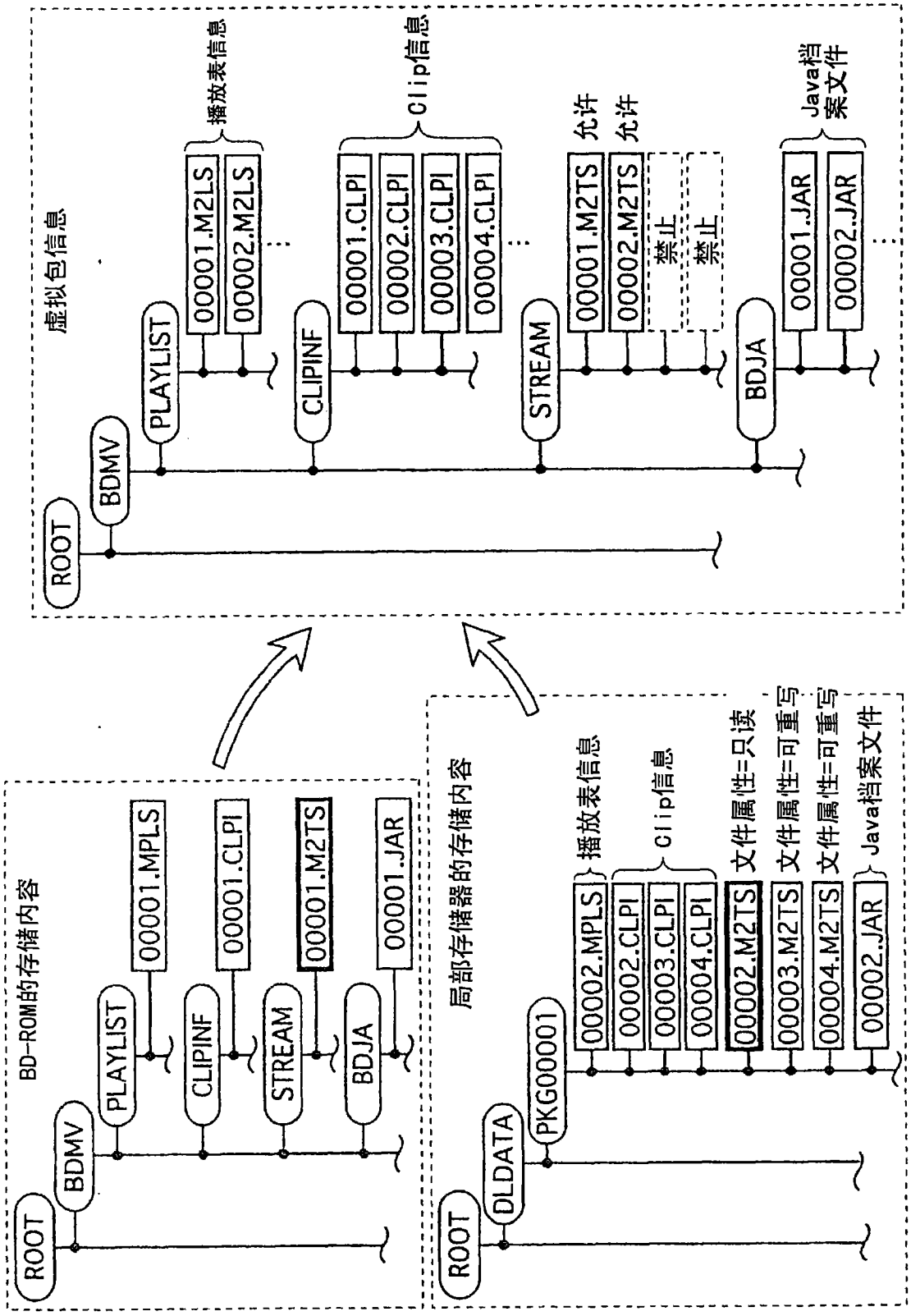

虚拟包信息
ROOT
BDMV
PLAYLIST
00001.M2LS
00002.M2LS
播放表信息
CLIPINF
00001.CLPI
00002.CLPI
00003.CLPI
00004.CLPI
Clip信息
STREAM
00001.M2TS
00002.M2TS
允许
允许
禁止
禁止
BDJA
00001.JAR
00002.JAR
Java档案文件
BD-ROM的存储内容
ROOT
BDMV
PLAYLIST
00001.MPLS
CLIPINF
00001.CLPI
STREAM
00001.M2TS
BDJA
00001.JAR
局部存储器的存储内容
ROOT
DLDATA
PKG00001
00002.MPLS
播放表信息
00002.CLPI
00003.CLPI
00004.CLPI
Clip信息
00002.M2TS
文件属性=只读
00003.M2TS
文件属性=可重写
00004.M2TS
文件属性=可重写
00002.JAR
Java档案文件

图 18

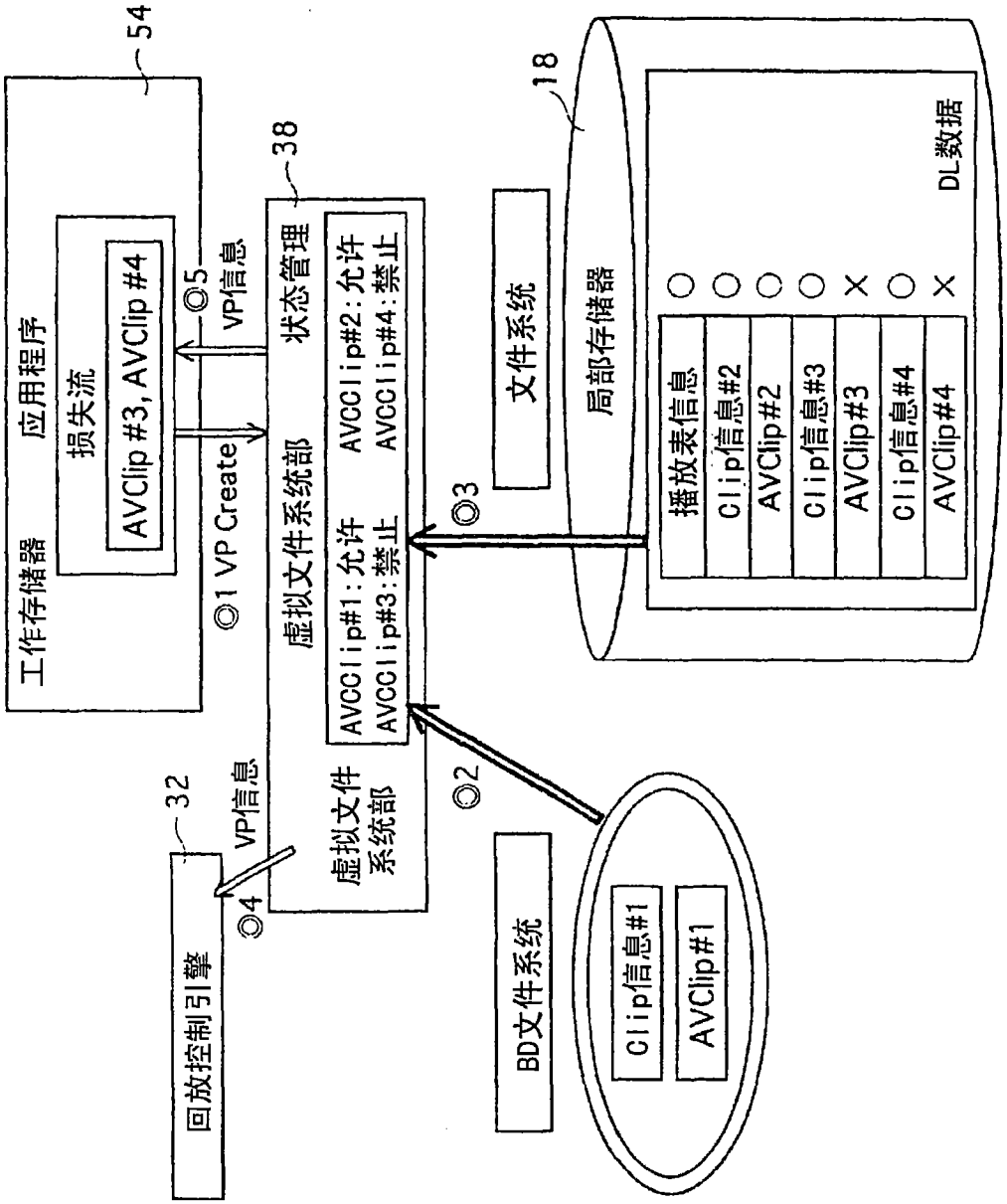

54
工作存储器
应用程序
损失流
AVClip #3, AVClip #4
◎5
VP信息
◎1 VP Create
38
虚拟文件系统部
状态管理
AVCClip#1:允许
AVCClip#3:禁止
AVCClip#2:允许
AVCClip#4:禁止
◎3
18
文件系统
局部存储器
播放表信息
Clip信息#2
AVClip#2
Clip信息#3
AVClip#3
Clip信息#4
AVClip#4
DL数据
32
回放控制引擎
VP信息
◎4
虚拟文件系统部
◎2
BD文件系统
Clip信息#1
AVClip#1

图 19

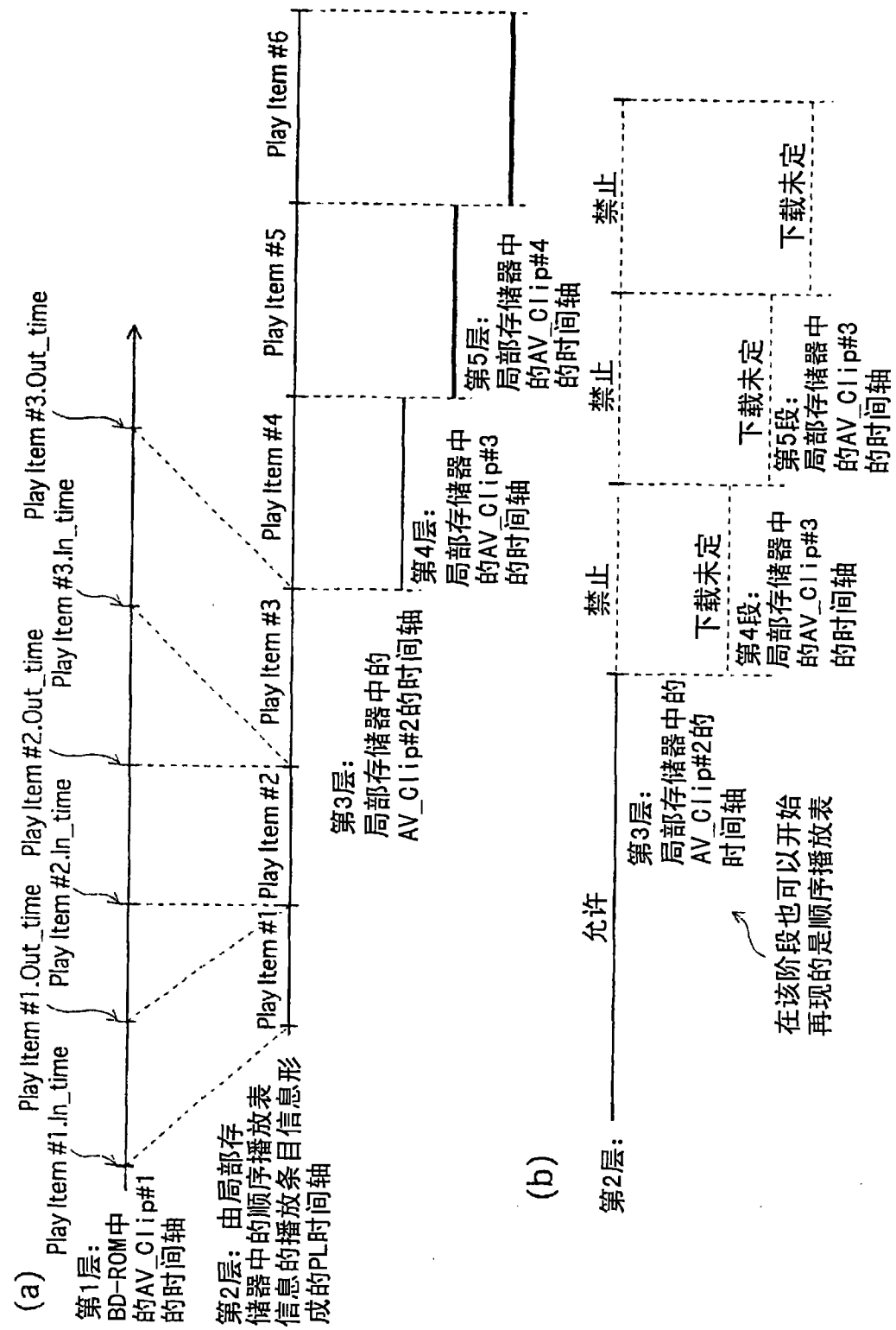

(a)
第1层：
BD-ROM中
的AV_Clip#1
的时间轴
Play Item #1.In_time
Play Item #1.Out_time
Play Item #2.In_time
Play Item #2.Out_time
Play Item #3.In_time
Play Item #3.Out_time
第2层：由局部存
储器中的顺序播放表
信息的播放条目信息形
成的PL时间轴
Play Item #1
Play Item #2
Play Item #3
Play Item #4
Play Item #5
Play Item #6
第3层：
局部存储器中的
AV_Clip#2的时间轴
第4层：
局部存储器中
的AV_Clip#3
的时间轴
第5层：
局部存储器中
的AV_Clip#4
的时间轴
(b)
第2层：
允许
禁止
禁止
禁止
第3层：
局部存储器中的
AV_Clip#2的
时间轴
下载未定
第4段：
局部存储器中
的AV_Clip#3
的时间轴
下载未定
第5段：
局部存储器中
的AV_Clip#3
的时间轴
下载未定
在该阶段也可以开始
再现的是顺序播放表

图 20

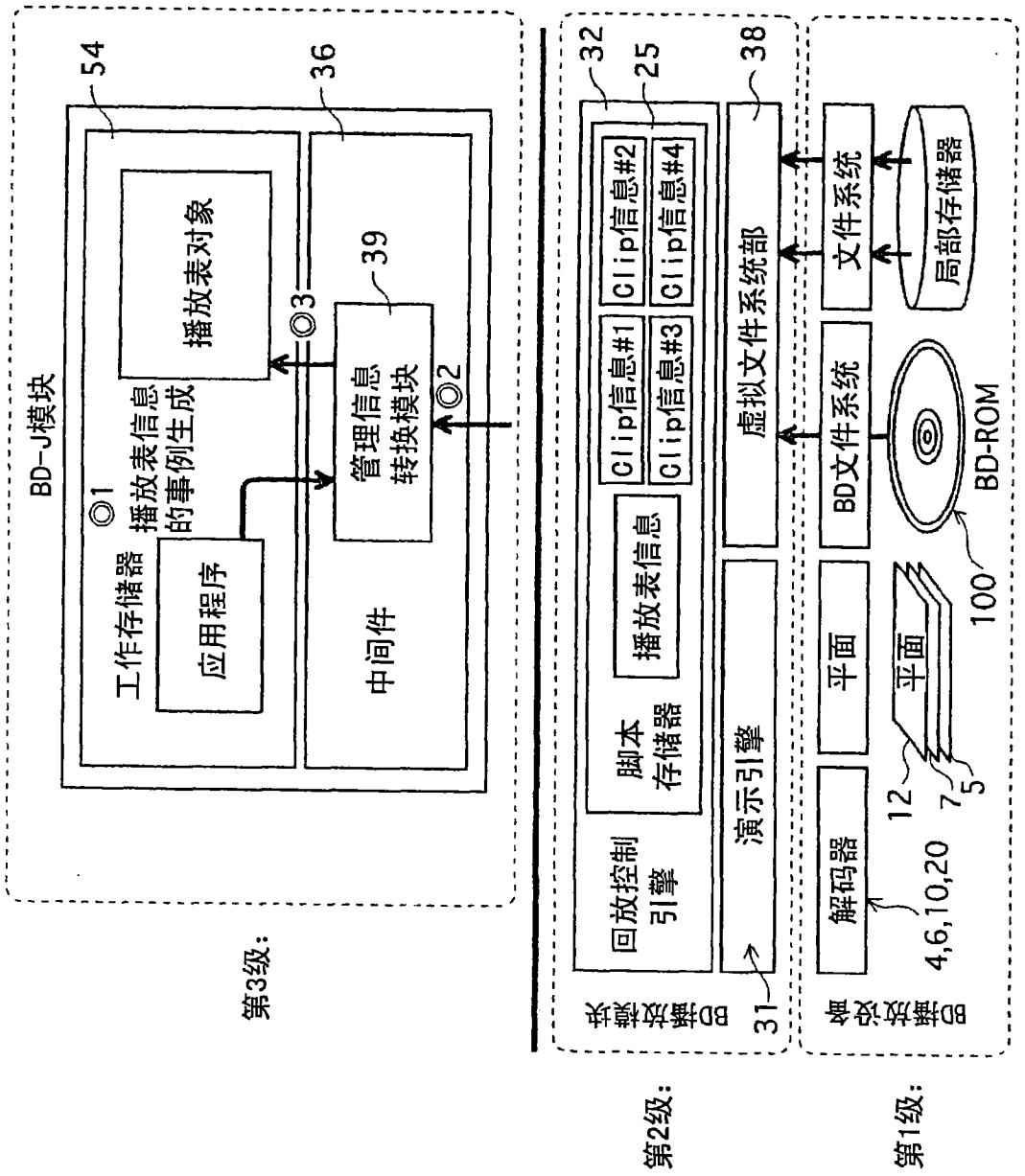

BD-J模块
54
36
工作存储器
应用程序
播放表信息的事例生成
播放表对象
中间件
管理信息转换模块
39
◎1
◎2
◎3
第3级：
32
25
38
Clip信息#1
Clip信息#2
Clip信息#3
Clip信息#4
脚本存储器
播放表信息
回放控制引擎
演示引擎
虚拟文件系统部
BD播放模块
31
解码器
平面
平面
BD文件系统
文件系统
局部存储器
BD-ROM
100
12
7
5
4,6,10,20
BD播放装置
第2级：
第1级：

图 21

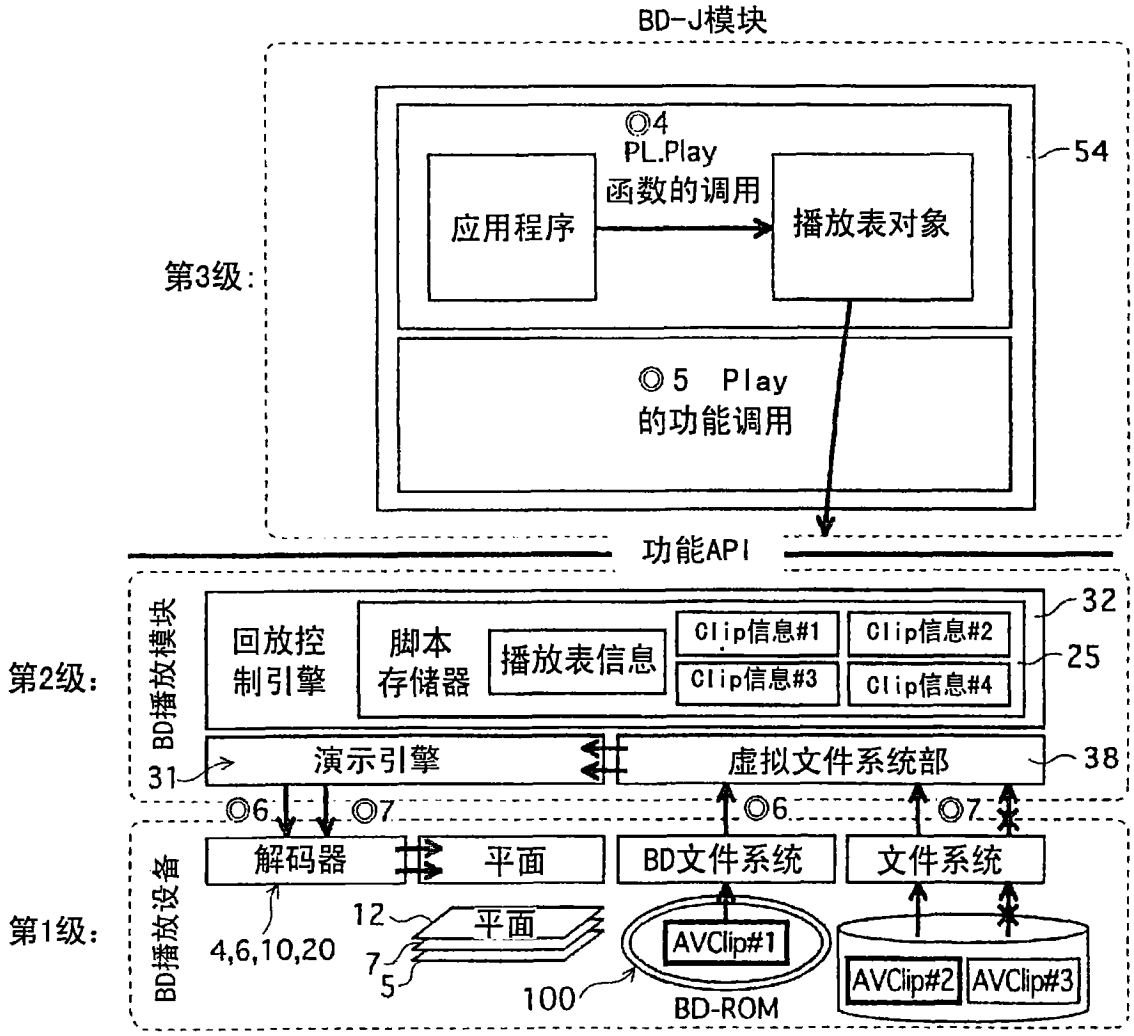

BD-J模块
第3级：
◎4
PL.Play
函数的调用
应用程序
播放表对象
54
◎5 Play
的功能调用
功能API
第2级：
BD播放模块
回放控制引擎
脚本存储器
播放表信息
Clip信息#1
Clip信息#2
Clip信息#3
Clip信息#4
32
25
31
演示引擎
虚拟文件系统部
38
◎6
◎7
◎6
◎7
第1级：
BD播放设备
解码器
平面
BD文件系统
文件系统
4,6,10,20
12
7
5
平面
AVClip#1
100
BD-ROM
AVClip#2
AVClip#3

图 22

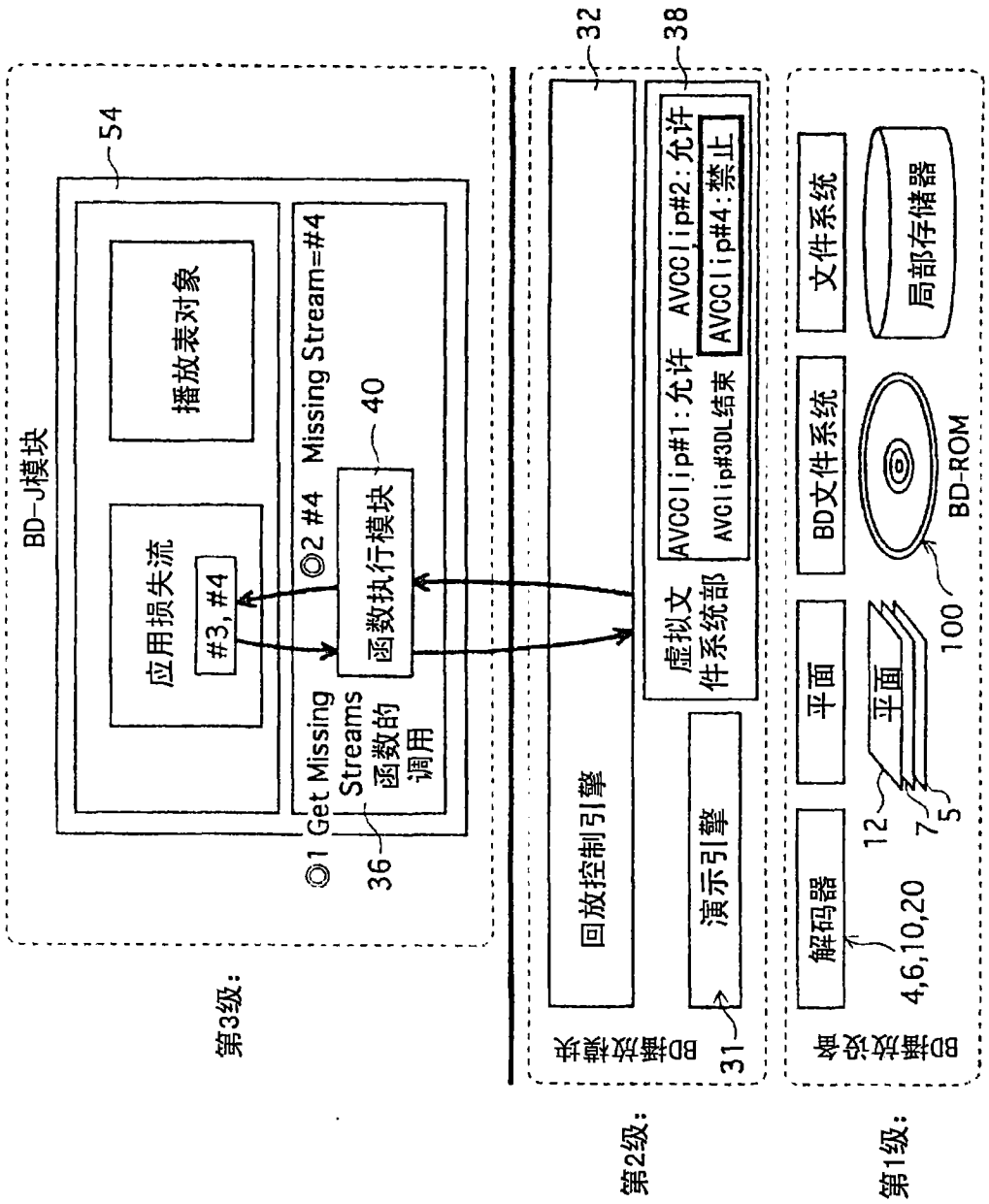

第3级:
第2级:
第1级:
BD-J模块
54
播放表对象
应用损失流
#3, #4
◎2 #4 Missing Stream=#4
函数执行模块
40
◎1 Get Missing Streams 函数的调用
36
回放控制引擎
32
虚拟文件系统部
AVClip#1:允许
AVClip#3DL结束
AVClip#2:允许
AVClip#4:禁止
38
演示引擎
31
BD播放模块
解码器
平面
平面
12
7
5
4,6,10,20
BD文件系统
文件系统
局部存储器
BD-ROM
100
BD播放设备

图 23

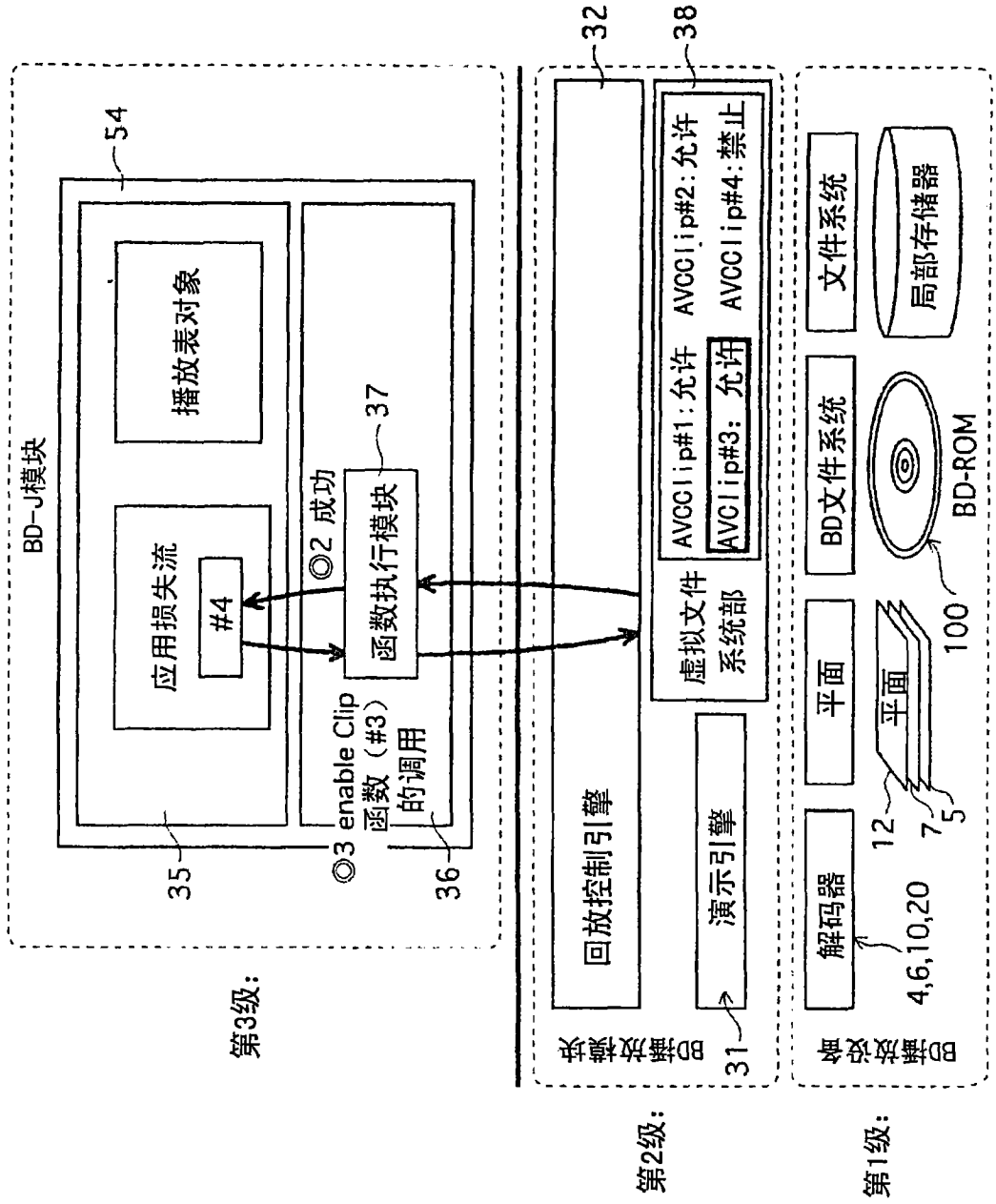

第3级：
第2级：
第1级：
BD-J模块
54
播放表对象
应用损失流
#4
35
◎3 enable Clip 函数（#3）的调用
36
◎2 成功
函数执行模块
37
回放控制引擎
32
演示引擎
31
虚拟文件系统部
AVCClip#1：允许
AVCClip#3：允许
AVCClip#2：允许
AVCClip#4：禁止
38
BD播放模块
解码器
4,6,10,20
平面
12
平面
7
5
BD文件系统
100
BD-ROM
文件系统
局部存储器
BD播放装置

图 24

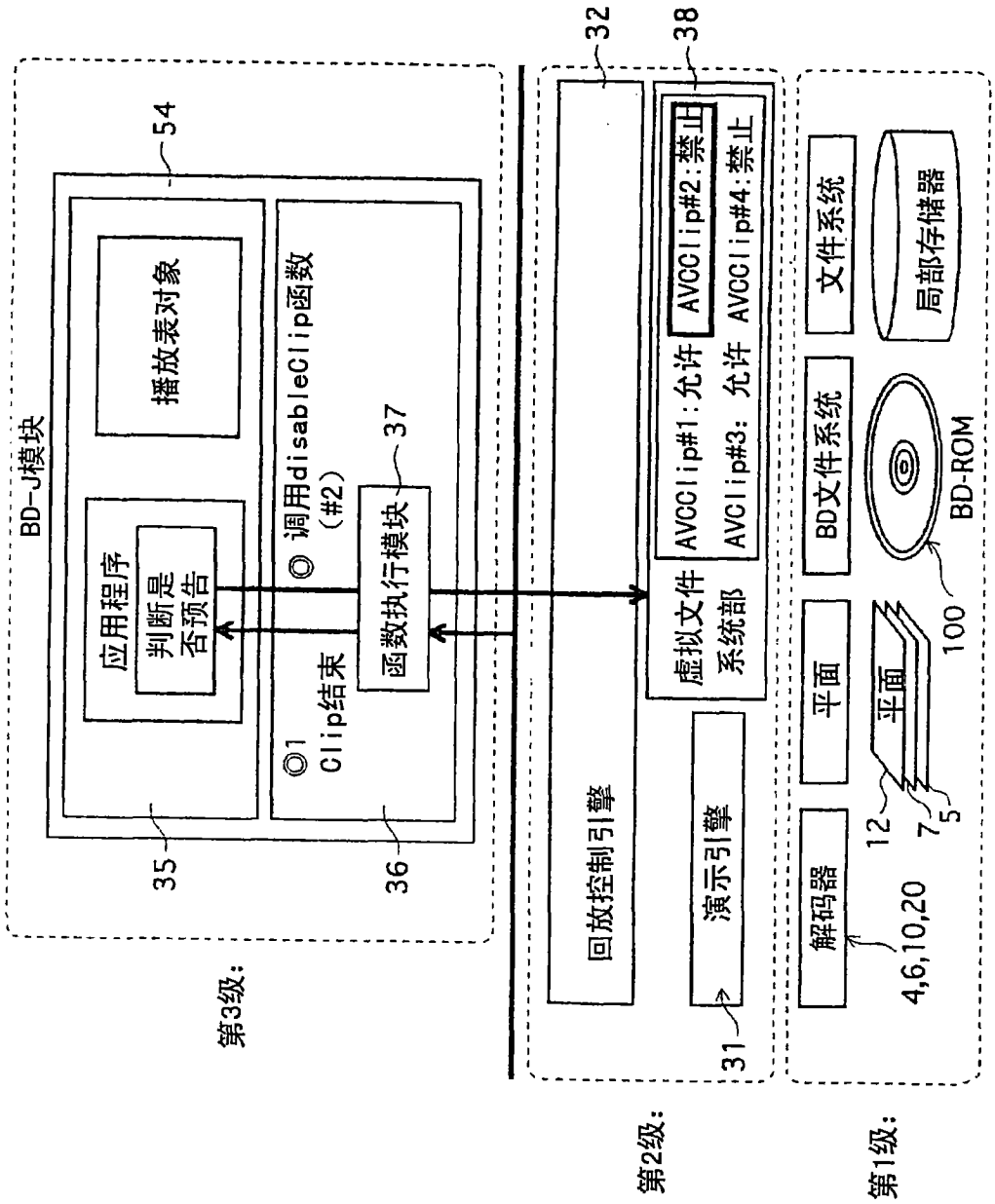

BD-J模块
54
播放表对象
应用程序
判断是否预告
35
◎调用disableClip函数（#2）
◎1 Clip结束
函数执行模块
37
36
第3级：
32
回放控制引擎
演示引擎
31
38
虚拟文件系统部
AVCClip#1：允许
AVCClip#2：禁止
AVClip#3：允许
AVClip#4：禁止
第2级：
解码器
平面
平面
12
7
5
4,6,10,20
BD文件系统
文件系统
局部存储器
BD-ROM
100
第1级：

图 25

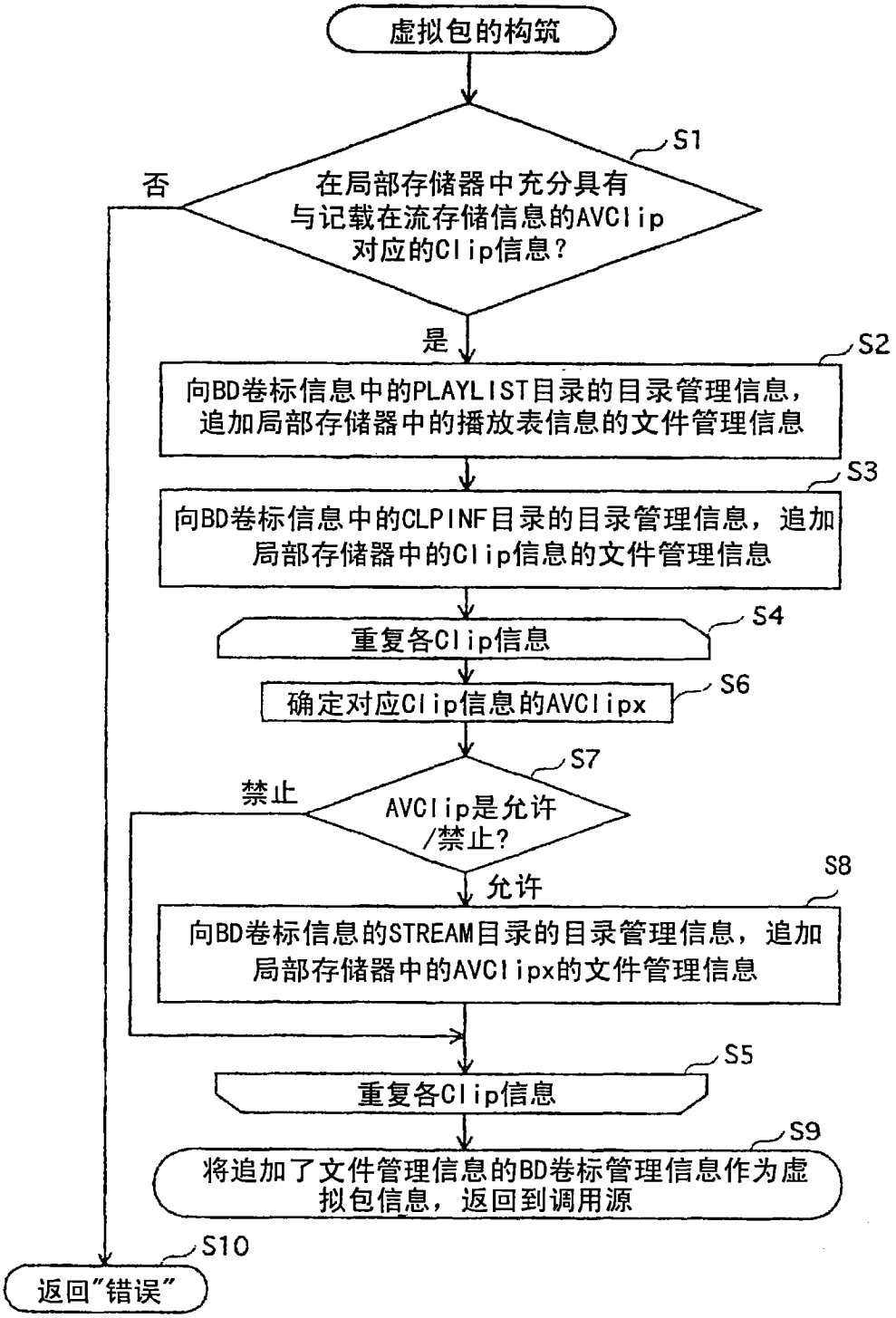

虚拟包的构筑
S1
否
在局部存储器中充分具有
与记载在流存储信息的AVClip
对应的Clip信息？
是
S2
向BD卷标信息中的PLAYLIST目录的目录管理信息，
追加局部存储器中的播放表信息的文件管理信息
S3
向BD卷标信息中的CLPINF目录的目录管理信息，追加
局部存储器中的Clip信息的文件管理信息
S4
重复各Clip信息
S6
确定对应Clip信息的AVClipx
S7
禁止
AVClip是允许
/禁止?
允许
S8
向BD卷标信息的STREAM目录的目录管理信息，追加
局部存储器中的AVClipx的文件管理信息
S5
重复各Clip信息
S9
将追加了文件管理信息的BD卷标管理信息作为虚
拟包信息，返回到调用源
S10
返回"错误"

图 26

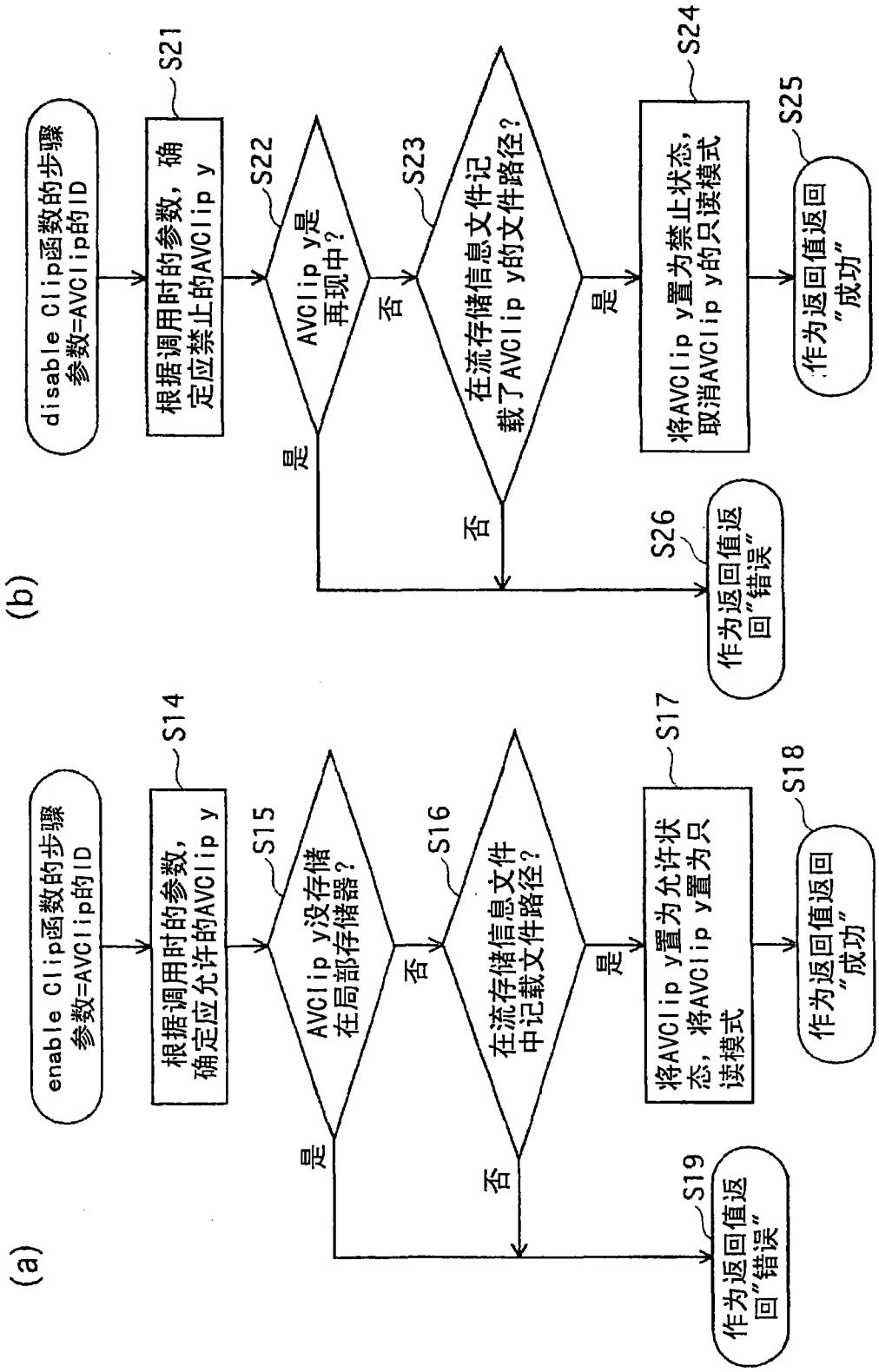

(a)
enable Clip函数的步骤
参数=AVClip的ID
S14
根据调用时的参数，确定应允许的AVClip y
S15
AVClip y没存储在局部存储器?
是
否
S16
在流存储信息文件中记载文件路径?
是
否
S17
将AVClip y置为允许状态，将AVClip y置为只读模式
S18
作为返回值返回"成功"
S19
作为返回值返回"错误"
(b)
disable Clip函数的步骤
参数=AVClip的ID
S21
根据调用时的参数，确定应禁止的AVClip y
S22
AVClip y是再现中?
是
否
S23
在流存储信息文件记载了AVClip y的文件路径?
是
否
S24
将AVClip y置为禁止状态，取消AVClip y的只读模式
S25
作为返回值返回"成功"
S26
作为返回值返回"错误"

图 27

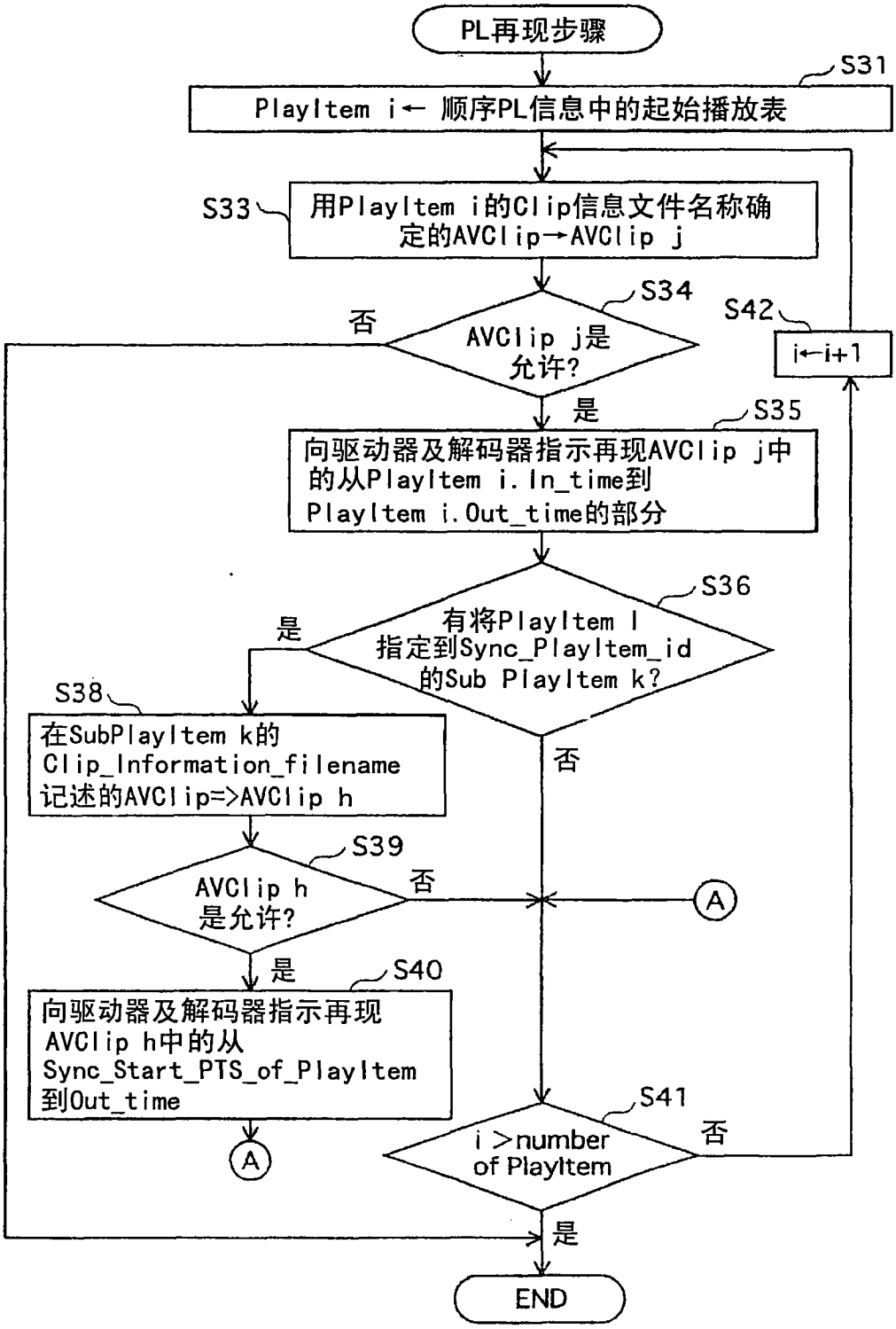

PL再现步骤
S31
PlayItem i← 顺序PL信息中的起始播放表
S33
用PlayItem i的Clip信息文件名称确定的AVClip→AVClip j
S34
AVClip j是允许?
否
是
S42
i←i+1
S35
向驱动器及解码器指示再现AVClip j中的从PlayItem i.In_time到PlayItem i.Out_time的部分
S36
有将PlayItem I指定到Sync_PlayItem_id的Sub PlayItem k?
是
否
S38
在SubPlayItem k的Clip_Information_filename记述的AVClip=>AVClip h
S39
AVClip h是允许?
否
是
A
S40
向驱动器及解码器指示再现AVClip h中的从Sync_Start_PTS_of_PlayItem到Out_time
A
S41
i >number of PlayItem
否
是
END

图 28

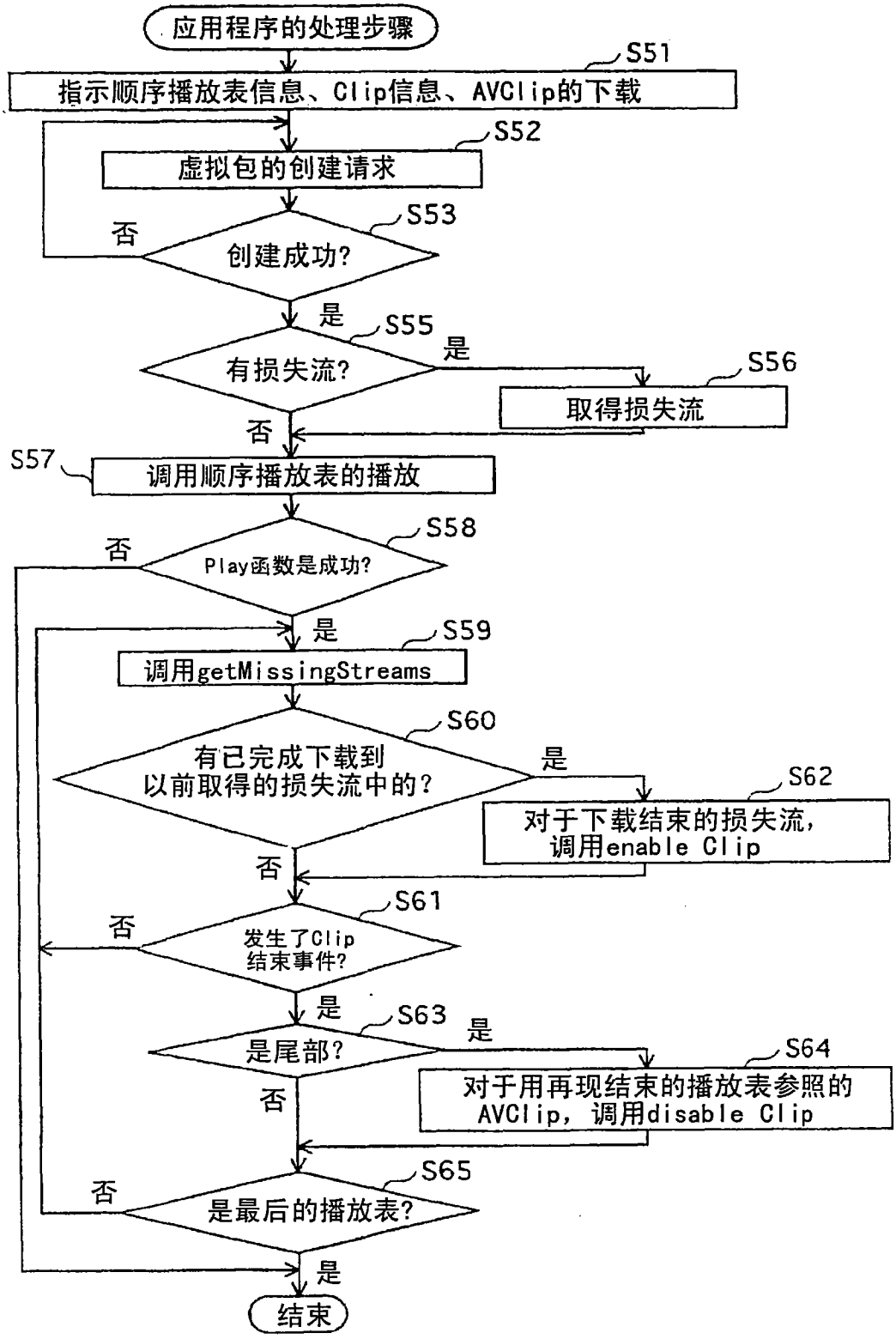

应用程序的处理步骤
S51
指示顺序播放表信息、Clip信息、AVClip的下载
S52
虚拟包的创建请求
S53
否
创建成功?
是
S55
有损失流?
是
S56
取得损失流
否
S57
调用顺序播放表的播放
S58
否
Play函数是成功?
是
S59
调用getMissingStreams
S60
有已完成下载到
以前取得的损失流中的?
是
S62
对于下载结束的损失流,
调用enable Clip
否
S61
否
发生了Clip
结束事件?
是
S63
是尾部?
是
S64
对于用再现结束的播放表参照的
AVClip, 调用disable Clip
否
S65
否
是最后的播放表?
是
结束

图 29

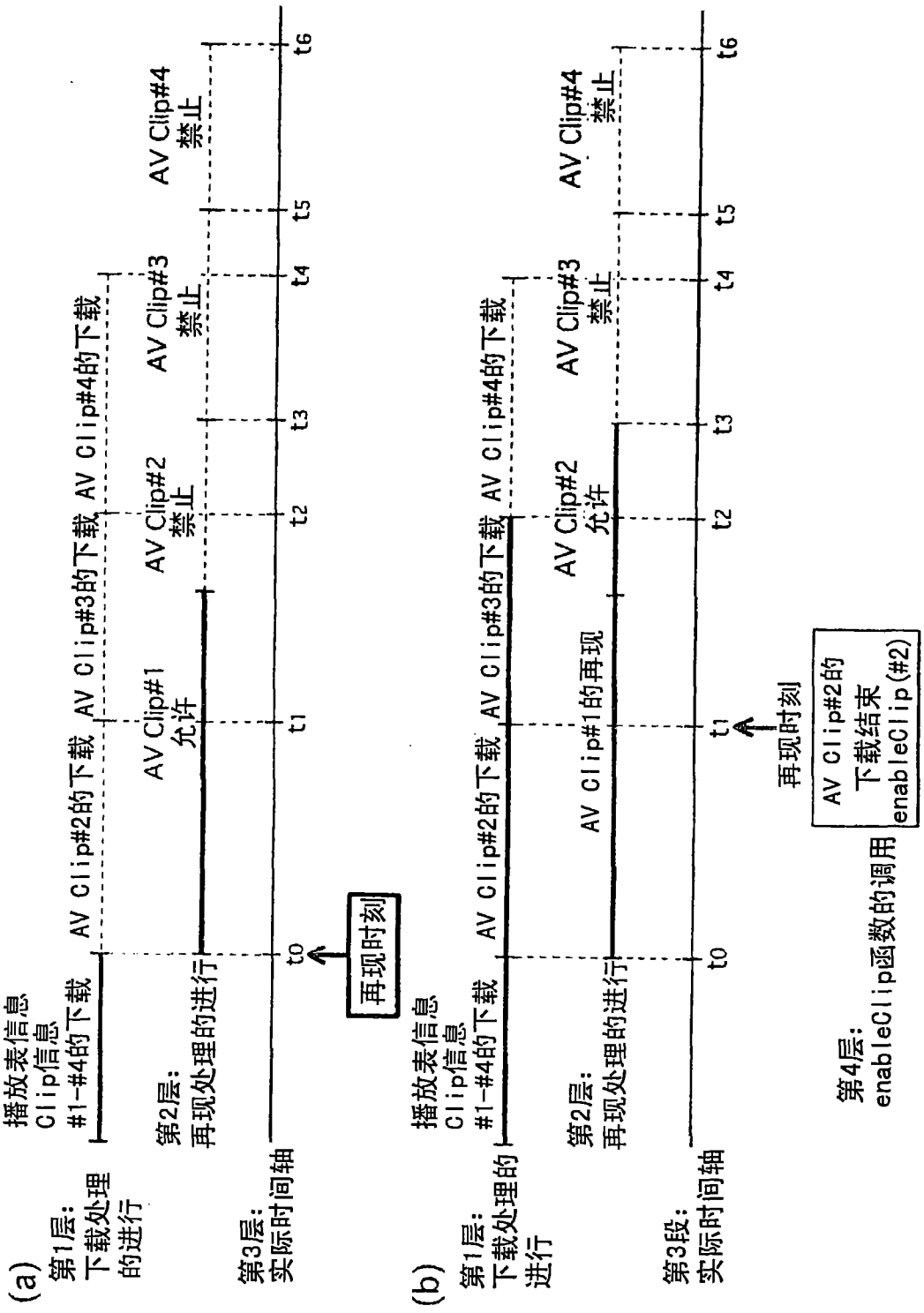

(a)
第1层：
下载处理的进行
播放表信息
Clip信息
#1-#4的下载
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
第2层：
再现处理的进行
AV Clip#1
允许
AV Clip#2
禁止
AV Clip#3
禁止
AV Clip#4
禁止
第3层：
实际时间轴
t0
t1
t2
t3
t4
t5
t6
再现时刻
(b)
第1层：
下载处理的进行
播放表信息
Clip信息
#1-#4的下载
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
第2层：
再现处理的进行
AV Clip#1的再现
AV Clip#2
允许
AV Clip#3
禁止
AV Clip#4
禁止
第3段：
实际时间轴
t0
t1
t2
t3
t4
t5
t6
再现时刻
第4层：
enableClip函数的调用
AV Clip#2的
下载结束
enableClip(#2)

图 30

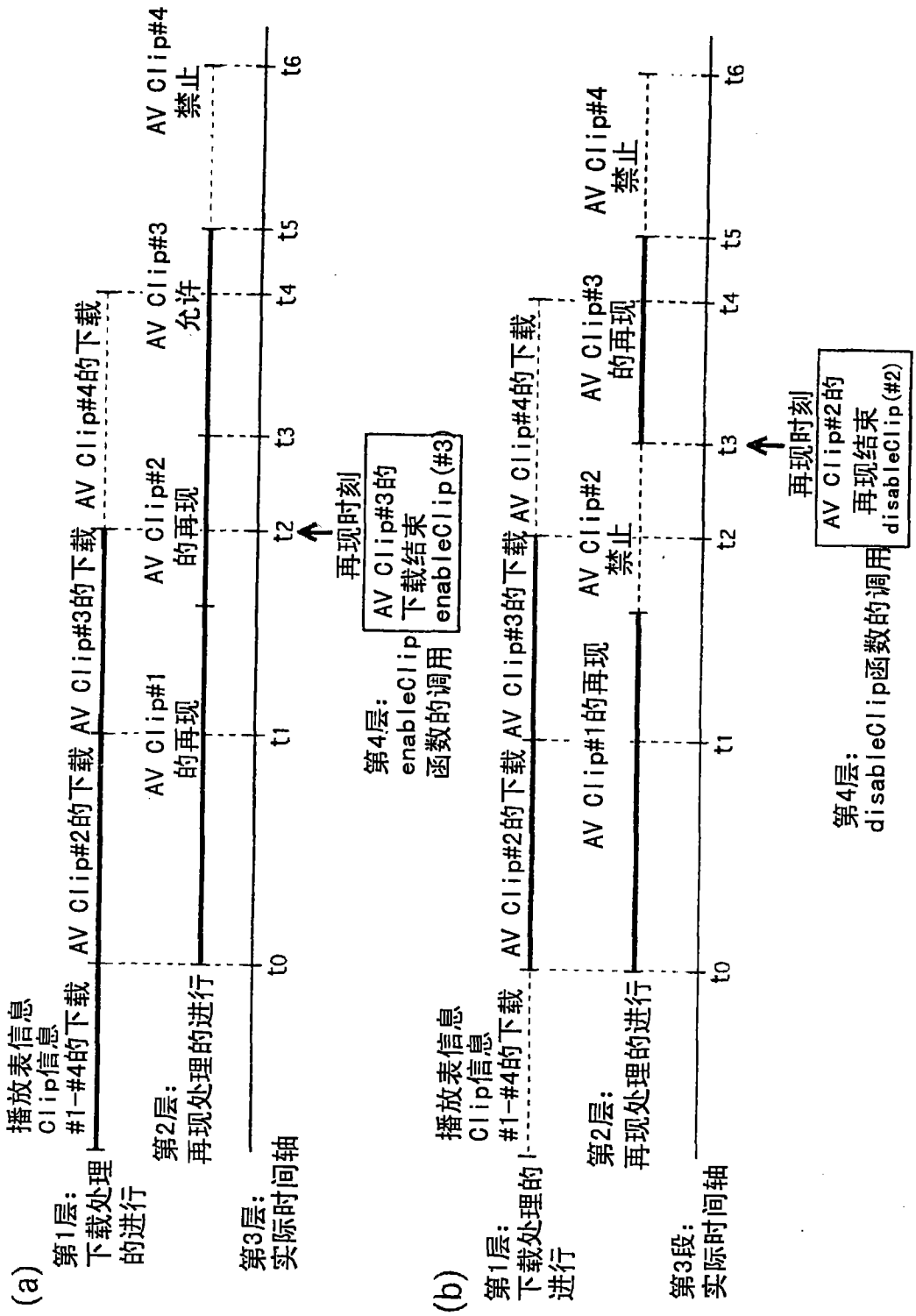

(a)
播放表信息
Clip信息
#1-#4的下载
第1层:
下载处理的进行
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
第2层:
再现处理的进行
AV Clip#1
的再现
AV Clip#2
的再现
AV Clip#3
允许
AV Clip#4
禁止
第3层:
实际时间轴
t0
t1
t2
t3
t4
t5
t6
再现时刻
第4层:
enableClip
函数的调用
AV Clip#3的
下载结束
enableClip(#3)
(b)
播放表信息
Clip信息
#1-#4的下载
第1层:
下载处理的进行
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
第2层:
再现处理的进行
AV Clip#1的再现
AV Clip#2
禁止
AV Clip#3
的再现
AV Clip#4
禁止
第3段:
实际时间轴
t0
t1
t2
t3
t4
t5
t6
再现时刻
第4层:
disableClip函数的调用
AV Clip#2的
再现结束
disableClip(#2)

图 31

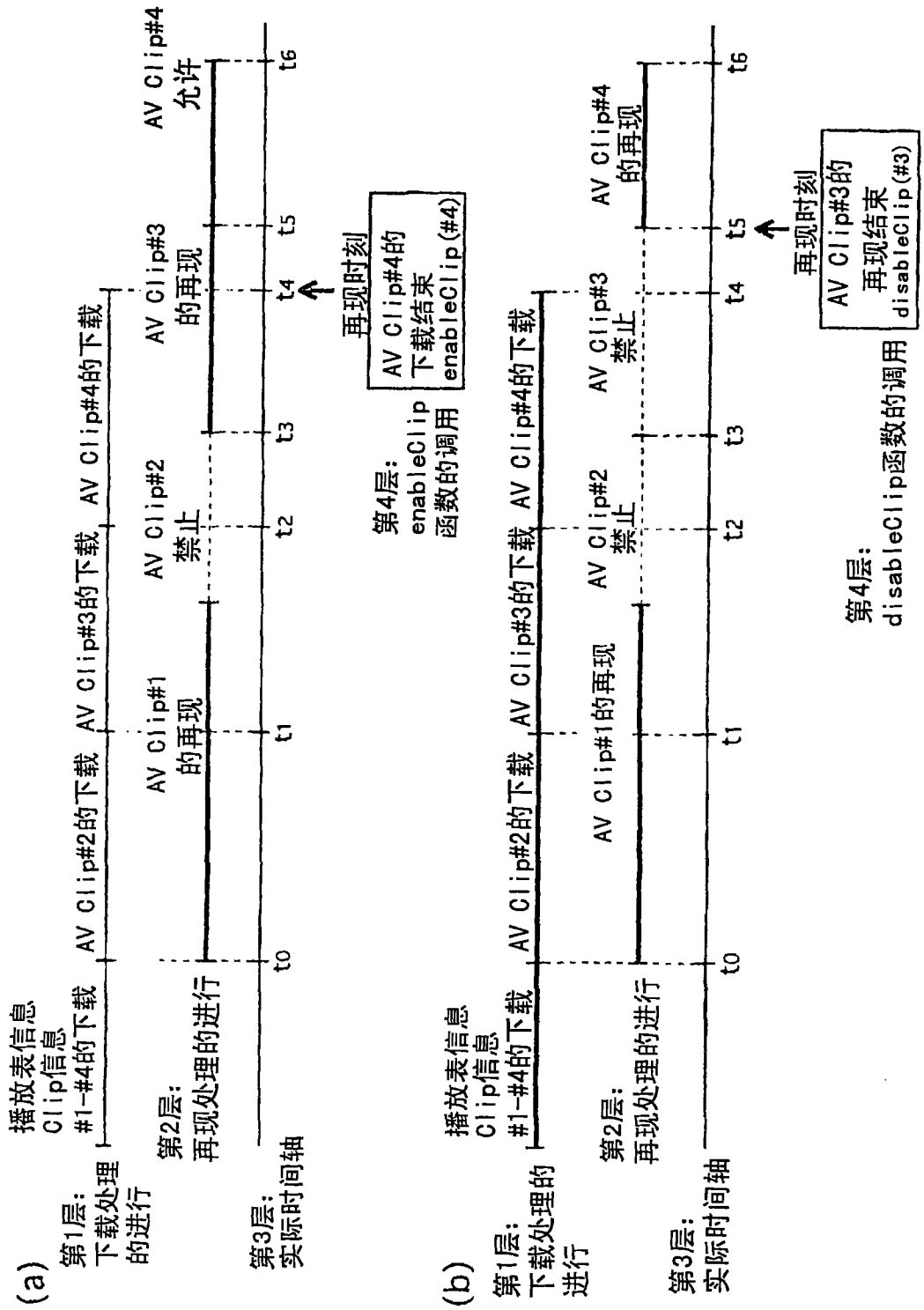

(a)
播放表信息
Clip信息
#1-#4的下载
第1层:
下载处理的进行
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
AV Clip#1
的再现
AV Clip#2
禁止
AV Clip#3
的再现
AV Clip#4
允许
第2层:
再现处理的进行
t0
t1
t2
t3
t4
t5
t6
第3层:
实际时间轴
再现时刻
第4层:
enableClip
函数的调用
AV Clip#4的
下载结束
enableClip(#4)
(b)
播放表信息
Clip信息
#1-#4的下载
第1层:
下载处理的进行
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
AV Clip#1的再现
AV Clip#2
禁止
AV Clip#3
禁止
AV Clip#4
的再现
第2层:
再现处理的进行
t0
t1
t2
t3
t4
t5
t6
第3层:
实际时间轴
再现时刻
第4层:
disableClip函数的调用
AV Clip#3的
再现结束
disableClip(#3)

图 32

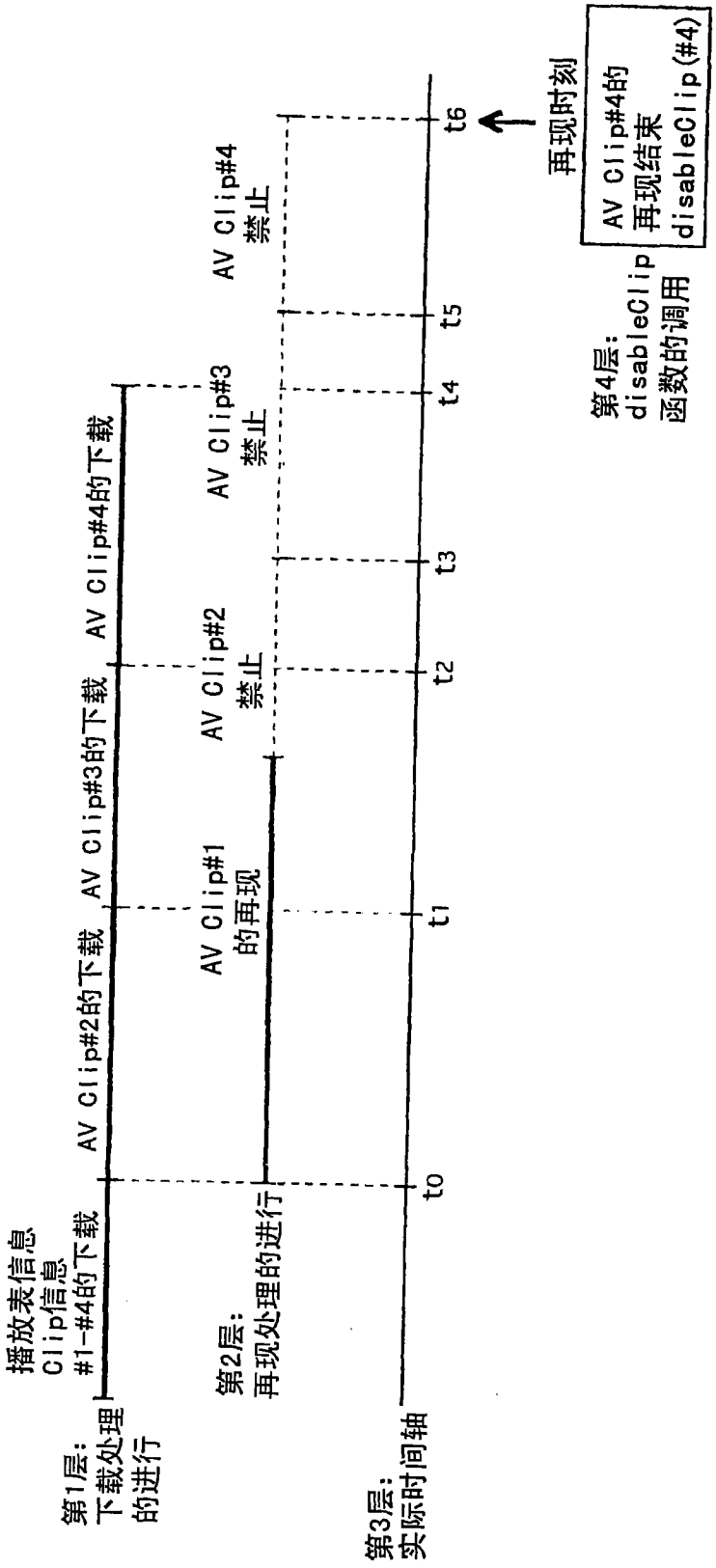

播放表信息
Clip信息
#1-#4的下载
第1层：
下载处理
的进行
AV Clip#2的下载
AV Clip#3的下载
AV Clip#4的下载
AV Clip#1
的再现
AV Clip#2
禁止
AV Clip#3
禁止
AV Clip#4
禁止
第2层：
再现处理的进行
第3层：
实际时间轴
t0
t1
t2
t3
t4
t5
t6
再现时刻
第4层：
disableClip
函数的调用
AV Clip#4的
再现结束
disableClip(#4)

图 33