



(12) 发明专利

(10) 授权公告号 CN 1754225 B

(45) 授权公告日 2012. 05. 30

(21) 申请号 200480004848. 6

(22) 申请日 2004. 02. 20

(30) 优先权数据
60/449, 136 2003. 02. 21 US

(85) PCT申请进入国家阶段日
2005. 08. 22

(86) PCT申请的申请数据
PCT/JP2004/002026 2004. 02. 20

(87) PCT申请的公布数据
W02004/074976 EN 2004. 09. 02

(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 小塚雅之 冈田智之 小川智辉

(74) 专利代理机构 永新专利商标代理有限公司
72002

代理人 夏青

(51) Int. Cl.
G11B 27/10 (2006. 01)

(56) 对比文件

EP 1381232 A1, 2004. 01. 14, 全文 .
US 2003/0028892 A1, 2003. 02. 06, 全文 .
JP 2002-369154 A, 2002. 12. 20, 全文 .
CN 1287358 A, 2001. 03. 14, 全文 .
W0 00/63860 A1, 2000. 10. 26, 全文 .
CN 1164916 A, 1997. 11. 12, 全文 .

审查员 宋洁

权利要求书 1 页 说明书 27 页 附图 39 页

(54) 发明名称

再现设备、记录方法以及再现方法

(57) 摘要

一种记录介质,当相同电影作品的不同版本被记录在单个记录介质上时,其能够根据这些不同版本的特性来执行菜单调用。AV 剪辑以及动态脚本被记录在 BD-ROM 100(记录介质)上。动态脚本是显示与视频数据有关的再现控制程序的命令串,并且具有附加到其上的属性信息。属性信息是显示当在 AV 剪辑再现期间用户请求菜单调用时的控制程序的信息,并且包括 resume_intension_flag。该 resume_intension_flag 显示是否期望在菜单调用结束之后恢复视频数据的再现。

Movie Object {		
resume_intention_flag [mobj_id]		属性信息
menu_call_mask [mobj_id]		
title_search_mask [mobj_id]		
Command [mobj_id] [0]		命令串
Command [mobj_id] [1]		
Command [mobj_id] [2]		
Command [mobj_id] [3]		
:		
:		
}		

1. 一种再现设备,该再现设备与一个记录介质有关,在该记录介质上记录有视频数据和动态脚本,包括:

用于执行所述动态脚本的模块;

再现单元,用于根据被执行的动态脚本中显示的再现控制程序来再现所述视频数据;

接收单元,用于在所述视频数据的再现期间接收用户操作;以及

管理器,用于如果所述接收的用户操作是菜单调用请求时,执行附加到所述动态脚本的属性信息中表示的控制程序,其中,

所述属性信息包括第一标记,

当所述第一标记是打开时由所述管理器执行的控制程序是这样一个控制,该控制用于在停止所述视频数据的再现之后执行所述菜单调用,并且在所述菜单调用已经结束之后,从在所述菜单调用请求时的再现位置恢复所述视频数据的再现,以及

当所述第一标记是关闭时由所述管理器执行的控制程序在所述菜单调用结束之后不恢复所述视频数据的再现的控制。

2. 一种用于在记录介质上进行记录的记录方法,包括以下步骤:

产生应用数据;以及

在所述记录介质上记录所述产生的应用数据,其中:

所述应用数据包括视频数据和动态脚本,

所述动态脚本是显示所述视频数据的再现控制程序的命令串,并且具有附加到其上的属性信息,

所述属性信息显示当在所述视频数据的再现期间用户请求菜单调用时的控制程序,并且还包括第一标记,以及

当所述第一标记是打开时由管理器执行的控制程序是这样一个控制,其用于在停止所述视频数据的再现之后执行所述菜单调用,并且在所述菜单调用结束之后,从所述菜单调用被请求时的再现位置恢复所述视频数据的再现,以及

当所述第一标记是关闭时由所述管理器执行的控制程序是用于在所述菜单调用结束之后不恢复所述视频数据的再现的控制。

3. 一种再现方法,与一个记录介质有关,该记录介质具有在其上记录的视频数据和动态脚本,该方法包括以下步骤:

执行所述动态脚本;

根据被执行的动态脚本中显示的再现控制程序,播放所述视频数据;

在所述视频数据的再现期间,接收用户操作;以及

如果所述接收的用户操作是菜单调用请求,则执行附加到所述动态脚本的属性信息中显示的控制程序,其中,

所述属性信息包括第一标记,

当所述第一标记是打开时由管理器执行的控制程序是这样一个控制,其用于在停止所述视频数据的再现之后执行所述菜单调用,并且在所述菜单调用结束之后,从所述菜单调用被请求时的再现位置恢复所述视频数据的再现,以及

当所述第一标记是关闭时由所述管理器执行的控制程序是用于在所述菜单调用结束之后不恢复所述视频数据的再现的控制。

再现设备、记录方法以及再现方法

技术领域

[0001] 本发明涉及诸如 BD-ROM 的用于发行电影作品的记录介质,以及用于播放这种记录介质的再现设备,并且特别是改进其中提供实现动态再现控制电影作品的方法。

背景技术

[0002] 随着 DVD-ROM 和 BD-ROM 的传播,可以按单个光盘售卖的电影作品(标题)的变化数量越大,产品的附加价值就越大。被称为静态脚本和动态脚本的脚本数据在增加标题变化的数量方面扮演了积极的角色。静态脚本是表示预先由光盘设计者定义的再现路径的信息。比较起来,动态脚本是根据设备的状态设置来动态地改变再现进程的脚本。

[0003] 图 1A-1C 表示了动态脚本。动态脚本根据在再现设备中的与语言设置来实现用于切换再现情景的“语言片头字幕”。在图 1A-1C 中,“PL”是播放列表的简略形式,其是再现路径,并且“PI”是播放项目的简略形式,是其再现章节。在图 1A-1C 中的动态脚本实现有条件的再现,以便如果再现设备中的语言设置 (SPRM(0)) 是“日语”(即“if (SPRM(0)) == Japanese”),则再现路径 PL#4 (PL#4, PI#1) 的再现章节 PI#1 被播放,并且如果在再现设备中的语言设置是除了 PL#4 的其他语言(即“else”),则再现路径 PL#2 (PL#2, PI#1) 的再现章节 PI#1 被播放。作为该有条件的再现的结果,经由取决于由用户所作的语言设置而不同的再现路径来执行再现。在图 1B 中的箭头 hb1 和 hb2 示意性地表示由动态脚本产生的条件转移。有关 DVD 再现控制的现有技术包括在日本专利申请 NO. 2856363 中公开的已知技术。

[0004] 然而,如果用户实施菜单调用,同时再现设备正根据内部的状态设置执行再现控制,则存在再现设备的状态设置将被改变的危险。菜单调用是按需型转移,用于转移通过用户按下菜单键而触发的再现设备中的状态设置例行程序。作为一个调用而不是跳转,菜单调用遵循处理(1),用于在执行状态设置例行程序之前保存在再现设备的寄存器中保持的一个值,并且遵循处理(2),用于在执行状态设置例行程序之后将保存的值重新存储在寄存器。被保存和重新存储的的寄存器保持的值表示再现时间中的当前点。这样,即使用户在再现路径中间请求一个菜单调用,由此启动状态设置例行程序,也可以在状态设置例行程序一旦结束从上述的再现位置紧接着之后恢复再现。

[0005] 在这里给出的举例中,再现设备中的语言设置是英语,并且在图 1A-1C 中的播放时间超过 PL#2,其是具体地用于英语的播放路径。如果在上述状态中实施菜单调用,并且再现设备中的状态设置从英语更新为日语,则再现设备失去了用于恢复再现的位置。这是因为根据菜单调用,当语言设置已经从英语变化为日语时恢复英语语言再现路径上的再现没有意义。同样,无意义的再现位置的设置的危险在于:当再现设备中执行软件时会引起暂停。

[0006] 这些困难可以通过一律禁止菜单调用而避开。然而,当电影作品的多个版本被记录在单个光盘上时,完全可以想象到不运行语言片头字幕的标题被记录在光盘上。在标题再现期间一律防止菜单调用显示了缺少对用户的考虑。

[0007] 本发明的目的在于提供一种记录介质，当电影作品的不同版本被记录在单个记录介质上时，其能够响应于各个标题的特性来执行菜单调用。

发明内容

[0008] 所提供的用于实现上述目的的记录介质具有在其上记录的视频数据和动态脚本，动态脚本是表示视频数据的再现控制程序的命令串，并且具有附加到其上的属性信息，属性信息表示用于当用户在视频数据再现期间请求菜单调用时的控制程序并且包括第一标记，并且第一标记指示当在视频数据再现期间菜单调用结束时是否从在菜单调用被请求时的再现位置恢复视频数据的再现。

[0009] 根据这种结构，在动态脚本等级，设置有关菜单调用的控制程序，动态脚本等级是分层模型的最高层，分层模型自下而上包括流，再现路径和动态脚本。当设计者特别想生成的标题实现语言片头字幕时，可以执行控制以接受菜单调用请求，而无需恢复再现。结果，即使当流和再现路径相同时，标题可以被容易地分为两种类型，即，关于哪些菜单调用被允许的标题，以及关于哪些菜单调用被禁止的标题。随着标题的产生，可以很轻松地增加具有不同的控制程序的变化数量，因为没有增加再现路径或者流的数量。

[0010] 日本专利申请 NO. 2856363 公开了用于基于流等级和再现路径设置用户操作容许性的技术。根据所公开的技术，相对于哪些菜单调用被允许或者禁止进行标题划分将使得流和再现路径的数量极大地增加，因为用户操作的容许性是基于流等级和再现路径来设置的。相反，采用本发明，没有增加流和再现路径的数量，因为在菜单调用完成之后的再现恢复的容许性是在动态脚本等级被设置的。因为没有在流和再现路径数量上的任何的增加，所以可以根据本发明防止错误，诸如关于哪些菜单调用应该被允许的标题与关于哪些菜单调用被禁止的标题相混淆。

附图说明

[0011] 图 1A-1C 显示动态脚本；

[0012] 图 2 显示关于本发明的记录介质的使用应用；

[0013] 图 3 显示 BD-ROM 的结构；

[0014] 图 4 显示利用目录结构的 BD-ROM 的应用格式；

[0015] 图 5 是显示根据功能性分类的在图 4 中的文件的分类图；

[0016] 图 6 显示将 BD-ROM 作为目标的分层模型；

[0017] 图 7 示意性地显示 AV 剪辑是如何构造的；

[0018] 图 8 显示剪辑信息的内部结构；

[0019] 图 9 显示 PL 信息的内部结构；

[0020] 图 10 概略显示利用 PL 信息的间接引用；

[0021] 图 11 显示与图 10 中定义的 (PLinfo#1) 不同的 PL 信息块 (PLinfo#2) 的例子；

[0022] 图 12 显示在分层模型的第四层的再现模式；

[0023] 图 13 显示电影对象的内部结构；

[0024] 图 14A 显示在其上附加有 resume_intension_flag、menu_call_mask、和 title_search_mask 的动态脚本；

- [0025] 图 14B 显示基于图 14A 中的电影对象的再现控制；
- [0026] 图 14C 显示正从标题的头部重新启动的再现；
- [0027] 图 15A-15C 显示再现设备侧的用于开始再现的重新启动的处理；
- [0028] 图 16A-16C 显示当由问题引起的转移被实现时电影对象的描述性例子；
- [0029] 图 17A-17C 显示当指示母锁定 (parental lock) 时的动态脚本的描述性例子；
- [0030] 图 18A-18b 显示 title_search_mask 的示范性设置；
- [0031] 图 19 显示关于本发明的再现设备的内部结构；
- [0032] 图 20 是一个显示由模块管理器 20 执行的处理过程的流程图；
- [0033] 图 21 是一个显示由模块管理器 20 执行的处理过程的流程图；
- [0034] 图 22 是一个显示由模块管理器 20 执行的处理过程的流程图；
- [0035] 图 23A-23C 显示 INFO. BD 的内部结构；
- [0036] 图 24A 显示具有多个记录在其上的动态脚本 (001. 电影、002. 电影、003. 电影、... 001. CLASS、002. CLASS、003. CLASS、...) 的 BD-ROM；
- [0037] 图 24B 显示当图 24A 中表示的动态脚本被列出时的索引表的描述性例子；
- [0038] 图 25A 显示当索引表如图 24B 中所示时在整个系统中的间接引用；
- [0039] 图 25B 显示在核心系统中的间接引用；
- [0040] 图 26 示意性地显示如何执行从电影对象转移到 Java 对象；
- [0041] 图 27 显示当具有在其上记录的图 18 中的脚本的 BD-ROM 被安装在核心系统再现设备中时如何执行转移；
- [0042] 图 28 显示由实施例 2 中的模块管理器 20 执行的处理过程；
- [0043] 图 29A 显示具有多个用于记录在其上的不同版本的索引表的 BD-ROM；
- [0044] 图 29B 假定图 29A 中的 BD-ROM 被安装在 0.1 版本再现设备中；
- [0045] 图 29C 假定图 29A 中的 BD-ROM 被安装在 1.1 版本再现设备中；
- [0046] 图 30 是一个显示由模块管理器 20 执行的处理过程的流程图；
- [0047] 图 31 显示由 BD-ROM 分级实现的菜单；
- [0048] 图 32 显示用于操作具有分级的菜单的电影对象；
- [0049] 图 33 是一个显示转移处理过程的流程图；
- [0050] 图 34 显示关于实施例 5 的播放项目的内部结构；
- [0051] 图 35 显示关于其再现控制由电影和 Java 对象执行的播放列表的分级结构；
- [0052] 图 36 显示根据播放项目 #3 和 #12 中的 Playable_PID_entries 如何执行滤波器规范；
- [0053] 图 37 显示由播放项目 #3 和 #12 中的 Playable_PID_entries 如何进行再现输出；
- [0054] 图 38 是一个显示由再现控制引擎 12 执行的 PLPlay 功能执行过程的流程图；以及
- [0055] 图 39 是一个显示用于 BD-ROM 的制造过程的流程图。

具体实施方式

[0056] 实施例 1

[0057] 下面描述本发明的记录介质的实施例。首先,描述与本发明的记录介质的实现相

关的使用动作。图 2 显示本发明的记录介质的使用动作。在图 2 中的 BD-ROM100 是与本发明有关的记录介质。BD-ROM100 用于在由再现设备 200、电视 300 以及遥控器 400 形成的家庭影院系统中提供电影作品。

[0058] 接下来,描述与本发明的记录介质的实现相关的产生动作。由于 BD-ROM 的应用层提高的结果,可以实现本发明的记录介质。图 3 表示 BD-ROM 的结构。

[0059] 在图 3 中的等级 4 表示 BD-ROM,第三等级表示 BD-ROM 的轨道。在等级 3 的轨道,以横向延长的形式描绘了从 BD-ROM 内部到外部螺旋的轨道。这些轨道由导入区、卷区、以及导出区组成。图 3 中的卷区具有分层模型,包括物理层、文件系统层、以及应用层。本发明的记录介质是通过在 BD-ROM 的应用层上形成图 3 中表示的数据格式来进行工业制造的。

[0060] 图 4 显示利用目录结构的 BD-ROM 的应用层格式(在下文中,简称为“应用格式”)。如图 4 中所示,在 BD-ROM 的根目录以下的是 BDMV 目录,并且在 BDMV 目录以下的是 JCLASS 目录和 BROWSER 目录。隶属 BDMV 目录的有以下文件:INFO. BD、XXX. M2TS、XXX. CLPI、YYY. MPLS、以及 ZZZ. 电影。隶属 JCLASS 目录的是 ZZZ. CLASS,并且隶属 BROWSER 目录的是 ZZZ. HTM。

[0061] 图 5 是一个当这些文件从功能性观点被分类时的分类图。在图 5 中,由第一、第二、第三和第四层形成的分层结构在图中示意性地显示分类。在图 5 中,XXX. M2TS 被包在第二层。XXX. CLPI 和 YYY. MPLS 被包在第三层(静态脚本)。隶属 BDMV 目录的 ZZZ. 电影、隶属 JCLASS 目录的 ZZZ. CLASS、以及隶属 BROWSER 目录的 ZZZ. HTM 被包在第四层。

[0062] 图 5 中的分类(第一到第四层)把诸如图 6 中表示的分层模型作为目标。参考图 6,以下描述在把 BD-ROM 作为目标的控制软件中的分层模型。

[0063] 图 6 中的第一层是物理层,其中实现关于目标用于处理的流的供给控制。如第一层中所示,目标流把 BD-ROM 和 HDD(硬盘驱动器)、存储卡、网络及其他种类的记录和通信介质作为它们的供给源。致力于这些 HDD、存储卡、和网络的控制(磁盘存取,卡存取,网络通信)在第一层被实现。

[0064] 第二层是一个解码格式层。这个第二层是其中定义了被用于解码由第一层供给的流的解码格式。MPEG-2 解码格式被用于当前的实施例。

[0065] 第三层(静态脚本)定义了流的静态脚本。静态脚本是由光盘设计者预先定义的再现路径信息和剪辑信息,第三层(静态脚本)是其中基于这些静态脚本的再现控制被定义的层。

[0066] 第四层用于实现流中的动态脚本。动态脚本是用于根据用户操作、设备状态等等动态地改变再现进程的脚本,第四层是其中定义了基于这些动态脚本的再现控制的层。下面,根据该分层模型,描述关于流、静态脚本、以及动态脚本的文件。

[0067] 首先,描述属于第二层的 AVClip(XXX. M2TS)。

[0068] AVClip(XXX. M2TS)是通过多路复用视频流、一个或多个音频流以及一个或多个作为呈现图形流以及交互图形流的图形流所获取的 MPEG-TS(传输流)格式数字流。视频流表示电影的运动图像部分,音频流表示电影的音频部分,呈现图形流表示电影的副标题,以及交互图形流表示涉及瞄准菜单的动态再现控制的过程。图 7 示意性地显示 AVClip 怎样组成。

[0069] AVClip(第四等级)是通过将包括多个视频帧(图像 pj1、pj2、pj3)的视频流和

包括多个音频帧（第一等级）的音频流转换为 PES 包串（第二等级）而形成的，PES 包串随后被转换为 TS 包（第三等级）。同样地，涉及副标题的呈现图形流和涉及对话的交互图形流（第七等级）被转换为 PES 包串（第六等级），其被转换为 TS 包（第五等级），并且随后 TS 包被多路复用。多路复用包括安排存储视频帧的 TS 包和存储音频帧的 TS 包，以便音频帧被放置于接近视频帧，视频帧与音频帧同时被从 BD-ROM 读取。

[0070] 通过上述处理产生的 AVClip 被分成多个宽度，并且被记录在 BD-ROM 的区域中，正如采用正常计算机程序的情况。AVClip 包括一个或多个访问单元 (ACCESS UNIT)，并且可以在这些访问单元中被提示。访问单元是最小的解码单元，包括单个 GOP（图像组）和与 GOP 同时读取的音频帧。GOP 包括利用时间相关特性采用在过去方向和将来方向播放的图像进行压缩的双向预测的 (B) 图像、利用时间相关特征采用在过去方向播放的图像进行压缩的预测的 (P) 图像，以及利用频率相关特征（即非时间相关特征）在个别帧的图像中进行压缩的内部 (I) 图像。

[0071] 此外，XXX.M2TS 中的文件名“XXX”提取附加在 BD-ROM 中的 AVClip 上的 3 位识别号。也就是说，在图 7 中的 AVClip 是利用“XXX”进行唯一地识别。由此完成流 (XXX.M2TS) 的描述。应当注意到在这里涉及的 3 位号仅仅是示范性的，并且可以是任何长度。

[0072] 静态脚本

[0073] 以下描述静态脚本文件 (XXX.CLPI, YYY.MPLS)。

[0074] 剪辑信息 (XXX.CLPI) 是与各个 AVClip 有关的管理信息。图 8 表示了剪辑信息的内部结构。AVClip 是通过多路复用视频流和音频流来获取的，并且因为 AVClip 可以在访问单元中被提示，所以剪辑信息的管理项目包括视频流和音频流的属性以及提示位置在 AVClip 中的位置。图 8 中的引出线突出了剪辑信息结构。如引出线 hn1 所示，剪辑信息 (XXX.CPLI) 包括与视频流和音频流有关的“属性信息”以及“EP_map”，“EP-map”是用于提示访问单元的参考表。

[0075] 如引出线 hn2 所示，属性信息（属性）包括关于视频流的属性信息（视频属性信息）、属性信息号（编号）以及与在 AVClip 上多路复用的多个音频流中的每一个有关的属性信息（音频属性信息 #1-#m）。如引出线 hn3 所示，视频属性信息指示用于压缩视频流的压缩格式（编码），以及构成视频流的各条图像数据的分辨率（分辨率），纵横比（纵横比）以及帧率（帧率）。

[0076] 另一方面，如引出线 hn4 所示，与音频流有关的音频属性信息 #1-#m 指示用于压缩各个音频流的压缩格式（编码）、以及各个音频流的信道号（信道）和相应的语言（语言）。

[0077] EP_map 是一个参考表，用于利用时间信息来间接地参考多个提示位置的地址，并且如引出线 hn5 所示，包括多条入口信息（访问单元 #1 入口，访问单元 #2 入口，访问单元 #3 入口，...）以及入口号（编号）。如引出线 hn6 所示，每个入口指示与访问单元中的头部 I- 图像的地址和大小 (I-Size) 相对应的相应的访问单元的再现开始时间。访问单元的再现开始时间可以表示为位于访问单元头部的图像数据的时间戳（呈现时间戳）。同样，访问单元中的地址由 TS 包的序列号（源包号或者“SPN”）表示。因为采用可变长度编码压缩格式，所以可以通过参考访问单元的入口来从任意的再现时间提示相应于该再现时间的访问单元中的一条图像数据，甚至当包括 GOP 的访问单元的大小和再现时间不一致时。此

外, XXX.CPLI 的文件名" XXX" 使用与剪辑信息相对应的 AVClip 相同的名称。换句话说, 图 8 中的剪辑信息的文件名" XXX" 与 AVClip 相对应 (XXX.M2TS)。这样结束剪辑信息的描述。接下来描述播放列表信息。

[0078] YYY.MPLS(播放列表信息) 是一个构成播放列表的表, 其是再现路径信息, 并且包括多条播放项目信息 (播放项目信息 #1、#2、#3、...、#n) 以及播放项目信息号 (编号)。图 9 显示 PL 信息的内部结构。播放项目信息是指针信息, 用于定义构成播放列表的一个或者多个再现逻辑段。播放项目信息的结构由引出线 hs1 突出显示。如引出线 hs1 所示, 播放项目信息包括: " Clip_information_filename", 用于指示与再现部分的入点 (In-point) 和出点 (Out-point) 所属的 AVClip 有关的再现部分信息的文件名; " Clip_codec_identifier", 用于表示用来编码 AVClip 的编码格式; " In_time", 用于表示再现部分开始的时间信息; 和 " Out_time", 用于表示再现部分结束的时间信息。

[0079] 播放项目信息的特征是评论。也就是说, 再现部分由使用 EP_map 作为参考表的间接引用格式来定义。图 10 概略显示了利用 PL 信息的间接引用。图 10 中的 AVClip 由多个访问单元构成。剪辑信息中的 EP_map 指定访问单元的扇区地址, 如箭头 ay1、ay2、ay3 以及 ay4 所示。图 10 中的箭头 jy1、jy2、jy3 和 jy4 示意性地表示了利用播放项目信息的访问单元的参考。换句话说, 这显示播放项目信息的参考 (jy1、jy2、jy3、jy4) 包括间接引用, 其中经由 EP_map 指定包括在 AVClip 中的访问单元的地址。

[0080] 由播放项目信息、剪辑信息和 AVClip 的组形成的 BD-ROM 上的再现部分被称作 " 播放项目"。由 PL 信息、剪辑信息和 AVClip 的组形成的 BD-ROM 上的再现单元被称作 " 播放列表" (缩写为 " PL")。记录在 BD-ROM 上的电影作品以这些逻辑再现单元 (PL) 构成。因为 BD-ROM 上的电影作品以逻辑再现单元构成, 所以不同于主要的电影作品, 可以例如通过定义指定情景的 PL, 从其中仅仅某一个特性出现的情景中很容易地生成电影作品。图 11 显示了定义与图 10 中表示的 PL (PL 信息 #1) 不同的 PL (PL 信息 #2) 时的例子。

[0081] 静态脚本的最大优点是能够增加电影制作人表达的范围, 因为简单地通过定义不同的 PL 信息段, 电影作品的变化增加。

[0082] 除了 PL 和播放项目, 在 BD-ROM 中还有被称为章节的再现单元。章节由一个、两个、或更多个播放项目构成。

[0083] 同样, PL 信息的文件名 " YYY" 提取附加在 BD-ROM 中的 PL 信息上的 3 位识别号。也就是说, 在图 11 中的 PL 信息是利用识别号 YYY 进行唯一地识别。将 PL 信息的识别号表示为 " YYY" 显示这个识别号是一个不同于 AVClip 的识别号 XXX 和剪辑信息的编号系统 (在这里使用的 3 位号仅仅是示范性的, 以及可以任何位数字)。

[0084] 由此结束静态脚本的描述。接下来描述动态脚本。

[0085] 动态脚本

[0086] 动态脚本是表示与 AVClip 有关的动态再现控制程序的命令串。动态再现控制程序响应于关于设备的用户操作而变化, 并且在性质上类似于计算机程序。在这里, 动态再现控制具有两种模式。两种模式的其中之一是用于播放记录在 BD-ROM 上的视频数据 (标准模式) 的模式, 另一个模式是用于在专用于 AV 设备的再现环境中提高记录在 BD-ROM 上的视频数据的附加价值的模式 (增强模式)。图 12 表示在分层模型的第四层的再现模式。在图 12 中的第四层描述一个标准模式和两个增强模式。被称为电影模式的标准模式是用

于类似 DVD 的环境的再现模式。两个增强模式的第一个被称为 Java 模式,其是一个主要以 Java 虚拟机方式使用的再现模式。第二个增强模式被称为浏览器模式,是一个主要以浏览器方式使用的再现模式。因为在第四层存在三个模式(即电影模式,Java 模式和浏览器模式),优选地描述了可以用来执行动态脚本的模式。当想要利用非常类似于面向 DVD 的命令的命令描述控制命令时,优选地描述电影模式的再现控制程序。利用这种方法,可以使得再现设备执行再现控制,所述再现控制非常类似于现有 DVD 再现设备中的再现控制。当利用页面描述语言来描述控制程序时,优选地描述浏览器模式的再现控制程序。同样地,可以描述用于访问网络位置、下载文件等等的控制程序。图 4 中的 ZZZ. CLASS 是 Java 模式的动态脚本,ZZZ. HTM 是浏览器模式的动态脚本,以及 ZZZ. 电影是电影模式的动态脚本。

[0087] 电影模式中的动态脚本

[0088] 以下描述涉及电影模式中的动态脚本。电影对象(ZZZ. 电影)是以类似于使用在 DVD 再现设备的命令描述的动态脚本。电影对象包括指示 PL 再现的再现命令,在 PL 再现之前被执行的命令(前命令),以及在 PL 再现之后被执行的命令(后命令)。具有 PL 的一个或多个动态脚本对被称为标题,在动态脚本中指示 PL 再现。标题是相应于 BD-ROM 上的整个电影作品的单元。应当注意到:“电影对象”有时在以下缩短为“M-OBJ”。

[0089] 用于描述脚本的技术

[0090] 上述动态脚本可以利用从第三层(静态脚本)提供的函数进行描述。以下描述涉及从第三层(静态脚本)提供的函数。

[0091] (a) 再现函数:从由第二自变量指定的位置开始由第一自变量指定的播放列表的再现。

[0092] 格式:PlayPL(第一自变量、第二自变量)

[0093] 第一自变量能够利用播放列表的编号指定用于进行再现的 PL。第二自变量能够利用包括在 PL 中的播放项目、以及 PL 中的任意时间、章节和标记指定再现开始位置。

[0094] 利用播放项目指定再现开始位置的 PlayPL 函数被称作“PlayPLatPlayItem()”,使用章节指定再现开始位置的 PlayPL 函数被称作“PlayPLatChapter()”,以及利用时间信息指定再现开始位置的 PlayPL 函数被称作“PlayPLatSpecified Time()”。

[0095] (b) 用于再现设备的状态获取和状态设置的函数

[0096] 再现设备的状态被表示在 32 个单独的播放器状态寄存器(这些寄存器的设定值被称作系统参数(SPRM))和 32 个单独的通用寄存器(这些寄存器的设定值被称作总参数(GPRM))中。

[0097] 电影对象、Java 对象和网页对象例如能够通过利用以下函数(i)至(iv)在这些寄存器中设置值以及从这些寄存器获取值。

[0098] (i)“得到播放器状态寄存器的值”的函数

[0099] 格式:得到播放器状态寄存器的值(自变量)

[0100] 这个函数获取利用自变量指定的播放器状态寄存器的设定值。

[0101] (ii)“设置播放器状态寄存器的值”的函数

[0102] 格式:设置播放器状态寄存器的值(第一自变量,第二自变量)

[0103] 这个函数使得利用第二自变量指定的值被设置在利用第一自变量指定的播放器

状态寄存器中。

[0104] (iii) “得到通用寄存器的值” 的函数

[0105] 格式 : 得到通用寄存器的值 (自变量)

[0106] 这个函数获取利用自变量指定的通用寄存器的设定值。

[0107] (iv) “设置通用寄存器的值” 函数

[0108] 格式 : 设置通用寄存器的值 (第一自变量, 第二自变量)

[0109] 这个函数使得利用第二自变量指定的值被设置在利用第一自变量指定的通用寄存器中。

[0110] 播放器状态寄存器的设定值 (SPRM) 具有以下含义。符号 “SPRM(x)” 在以下指的是第 x 个播放器状态寄存器的设定值。

[0111] SPRM(0) : 保留

[0112] SPRM(1) : 用于解码的音频流的流编号

[0113] SPRM(2) : 用于解码的图形流的流编号

[0114] SPRM(3) : 表示用户角度设置的编号

[0115] SPRM(4) : 当前用于再现的标题的编号

[0116] SPRM(5) : 当前用于再现的章节的编号

[0117] SPRM(6) : 当前用于再现的 PL 的编号

[0118] SPRM(7) : 当前用于再现的播放项目的编号

[0119] SPRM(8) : 表示当前再现时间的时间信息

[0120] SPRM(9) : 导航计时器的计数值

[0121] SPRM(10) : 当前在选择状态的按钮的编号

[0122] SPRM(11)–(12) : 保留

[0123] SPRM(13) : 用户的母等级的设置

[0124] SPRM(14) : 有关再现设备的视频再现的设置

[0125] SPRM(15) : 有关再现设备的音频再现的设置

[0126] SPRM(16) : 表示再现设备中的音频设置的语言码

[0127] SPRM(17) : 表示再现设备中的副标题设置的语言码

[0128] SPRM(18) : 用于呈现菜单的语言设置

[0129] SPRM(19)–(31) : 保留

[0130] 在这些 SPRM 中, 当由用户经由菜单操作选择标题时, SPRM(4) 被更新。无论何时当前再现时间前进时 SPRM(5)–(7) 被更新。

[0131] 也就是说, 如果当前的再现时间从一个播放项目移动到另一个播放项目, SPRM(7) 被更新, 如果一个 PL 被转到另一个 PL, SPRM(6) 被更新, 以及如果一个章节被转到另一个章节, SPRM(5) 被更新。

[0132] 以这种方法, 通过参考 SPRM(4)–(7), 显示正在播放的标题和 PL, 以及正在 PL 中播放的播放项目和章节。

[0133] 无论何时显示属于 AVClip 的图像数据时, 更新作为表示当前再现时间 (即时间点) 的时间信息的 SPRM(8)。也就是说, 如果再现设备显示新的图像数据, SPRM(8) 被更新为表示新的图像数据的显示开始时间 (呈现时间) 的值。

[0134] Java 对象和网页对象能够通过利用“得到播放器状态寄存器的值”函数和“得到通用状态寄存器的值”函数参考播放器状态寄存器来详细地找到再现设备的状态。

[0135] (c) 同样存在从一个动态脚本到另一个动态脚本的转移, 尽管这些不是从第三等级(静态脚本)提供的编程函数。用于执行从一个动态脚本到另一个动态脚本的转移的函数包括以下 JMP 和调用(CALL)函数。

[0136] JMP 函数

[0137] 格式: JMP 自变量

[0138] 调用函数

[0139] 格式: CALL 自变量

[0140] JMP 函数是一个转移, 用于在操作期间丢弃当前的动态脚本, 并且执行利用自变量执行的转移目标的动态脚本。JMP 命令包括直接引用命令, 用于直接指定转移目标的动态脚本; 以及间接引用命令, 用于间接地指定转移目标的动态脚本。

[0141] 调用函数是一个转移, 用于使得利用自变量指定的转移目标的动态脚本在暂停当前的动态脚本操作之后进行操作, 然后一旦结束转移目标的动态脚本, 恢复暂停的脚本的操作。恢复命令被放置在形成调用命令的转移目标的动态脚本的末端。恢复命令(所谓的子程序的返回命令)用来重新激活由于调用功能执行而处于暂停状态的动态脚本。调用命令, 正如 JMP 命令, 包括直接引用命令, 用于直接指定转移目标动态脚本; 以及间接引用命令, 用于间接地指定转移目标动态脚本。

[0142] 由此结束由第三层(静态脚本)提供的函数和变量的描述。

[0143] 图 13 显示电影对象的内部结构。图 13 中所示的电影对象包括属性信息和命令串。属性信息包括 resume_intension_flag、menu_call_mask 和标题 _search_mask。

[0144] 该“resume_intension_flag”表示当菜单调用被请求时电影对象应当执行何种控制。如果 resume_intension_flag 是关闭的(OFF), 则当用户请求菜单调用时, 状态设置例行程序被调用。在这时候, 当前执行的电影对象被丢弃, 因为如上所述的恢复操作没有执行。如果当前的电影对象在再现设备中被丢弃, 再现设备的再现需要被重新启动。有两种关于当重新启动再现时返回再现位置的方法。一种方法包括在转移之前立刻从当前的电影对象中的状态重新启动再现。

[0145] 另一种方法包括当再现设备已经执行一些命令时从多个构造当前电影对象的命令的头部重新启动再现。因为前一种的方法包括复杂的处理以重新产生预转移状态, 本发明采用后一种方法。

[0146] 通过初始化显示当前的电影对象的执行位置的参数以及显示当前的再现位置的参数, 后一种的重新启动再现的方法被执行。也就是说, 当因为菜单调用转移至状态设置例行程序时, 显示再现位置的 SPRM(5)-(8) 被初始化。如果 SPRM(4)-(8) 在初始化之后被保存, SPRM(4)-(8) 能够在状态设置例行程序结束之后执行的重新存储处理期间在原始的寄存器中被重置。因为 SPRM(5)-(8) 已经被初始化, 再现设备利用重置值来重新启动再现。

[0147] 另一方面, 如果 resume_intension_flag 是打开(ON), 用于菜单调用使用的电影对象被跳转到暂停当前的电影对象之后并且保存 SPRM。当用于菜单调用使用的电影对象的处理已经结束时, 利用当前电影对象的再现在重新保存 SPRM 之后被恢复。采用实现图 1A-1C 中所表示的语言片头字幕的电影对象, resume_intension_flag 优选地被设置为关

闭 (OFF)。这是因为：如果菜单调用被请求并且当再现设备在 PL#2 时语言设置从英语变化为日语，再现设备错过了再现恢复位置。

[0148] 通过将其中有如上所述的损失再现位置的危险的电影对象中的 属性信息段中的 resume_intension_flag 设置为关闭，设计者能够在再现被执行时防止操作上的错误出现在再现设备中。以这种方法，设计者可以确保生成根据 SPRM 设置执行再现控制的电影对象。

[0149] 因为利用 resume_intension_flag 的再现恢复或者重新启动在电影对象单元中是可以的，所以产生包括一个或者两个命令的电影对象和转移这些电影对象允许在一个或者两个命令的单元中执行再现恢复或者重新启动。也就是说，电影对象优选地根据其中再现恢复或者重新启动被执行的单元来生成。由此结束 resume_intension_flag 的描述。

[0150] " menu_call_mask " 是一个表示是否屏蔽菜单调用的标记。如果这个标记是 OFF，则允许用户的菜单调用请求，并且如果是 ON，则禁止。

[0151] " 标题 _search_mask " 是一个表示是否屏蔽标题搜索的标记。如果这个标记是 OFF，则允许用户的标题搜索请求，并且如果是 ON，则禁止。如果当前的电影对象例如用于播放片尾（预览视频）或者 FBI 的警告视频，可以通过设置电影对象的标题 _search_mask 为 ON 来确信用户观看和理解这个视频的内容。

[0152] resume_intension_flag 和标题 _search_mask 的特定示范性设置描述如下。图 14A-14C 显示当实现由这个示范性的描述引起的语言片头字幕和再现控制时电影对象的示范性的描述。

[0153] 在图 14A 所示的示范性的电影对象描述中，resume_intension_flag、menu_call_mask 以及标题 _search_mask 已经被附加到图 1A 中所示的示范性的电影对象描述中。resume_intension_flag、menu_call_mask 以及标题 _search_mask 全部已经被设置为 " 0 "。图 14B 表示基于图 14A 中描述的电影对象的再现控制。图 14A 中的示范性的描述实现有条件的再现，使得如果再现设备中的语言设置 (SPRM(0)) 是 " 日语 "，则 PL#4 被播放 (Link (PL#4, PI#1))，以及如果再现设备中的语言设置是除了图 15B 中的 日语以外的其他语言（即，其他语言），则 PL#2 被播放 (Link (PL#2, PI#1))。在此，如果在再现设备中的语言设置是英语，则再现设备处理 PL#2。

[0154] 假定当再现设备在 PL#2 (rg1) 上时，用户请求菜单调用。在这种情况下，在图 15A-15C 中重新启动再现的处理被执行，因为给定的电影对象中的 resume_intension_flag 被设置为 "0"。图 15A- 图 15C 显示在再现设备中用于重新开始再现的处理。如果 resume_intension_flag 被设定为 "0"，SPRM(4)-(8) 显示在初始化 SPRM(5)-(8) 图之后再现位置被保存到存储器（图 15B）。在 SPRM(4)-(8) 已经被保存之后执行到状态设置例行程序 rcl 的转移 br1。假定用户利用这个状态设置例行程序将语言设置从英语变为日语（在图 15B 中，英语→日语）。一旦状态设置例行程序的处理已经结束，再现设备将保存在存储器中的 SPRM(4)-(8) 重新存储到寄存器。因为显示再现位置的初始化的 SPRM(5)-(8) 被设置在寄存器中，所以再现位置被设置到图 14B 中的标题的头部。结果，执行从标题头部开始的重新启动。此外，在图 14C 的例子中，再现位置由于再现设备未初始化 SPRM(4) 而被设置为标题头部，其中 SPRM(4) 显示当前被播放的标题的编号。如果这个标题编号被初始化，从鼓励用户选择标题的标题菜单重新启动再现。

[0155] 在图 16A-16C 中显示了另一个示范性的描述。

[0156] 图 16A-16C 显示当实现由一个问题引起的转移时的电影对象的示范性的描述以及由这个示范性的描述引起的再现控制。图 16A 所示的示范性的电影对象描述不同于图 14A 所示的在于:图 16A 实现了对话再现控制,而图 14 实现了语言片头字幕。在图 16A 中, PL#1 是一个问题情景,而 PL#2 和 PL#4 是当响应于该问题分别选择答案 (1) 和 (2) 时出现的情景。选择的那个答案设置在 GPRM(0) 中。由 IF 语句引起的再现切换是根据 GPRM(0) 来执行的。响应于状态设置例行程序的设置,没有更新仅仅是一个通用寄存器值的 GPRM(0)。同样,在这个示范性的描述中, resume_intension_flag 被设置为 " 1 "。

[0157] 图 16B 显示基于如上所述的电影对象的再现控制。

[0158] 假定当再现设备在 PL#2 时用户请求菜单调用。在这种情况下,再现设备省略图 15A,以及执行图 15B 的处理,因为给定电影对象中的 resume_intension_flag 被设置为 " 1 "。也就是说,显示再现位置的 SPRM(4)-(8) 被从寄存器保存到存储器。在 SPRM(4)-(8) 已经被保存之后,执行到状态设置例行程序 rc1 的转移 br1。假定用户利用这个状态设置例行程序(图 16B 中的英语→日语)将语言设置从英语改变为日语。一旦状态设置例行程序的处理已经结束,再现设备执行将 SPRM(4)-(8) 从存储器重新存储到寄存器的处理。因为 SPRM(4)-(8) 根据重新存储而在寄存器中被设置,所以再现位置是如此以至再现从先前的再现位置被恢复。

[0159] 图 17A-17C 中显示了一个示范性的电影对象。如果作为再现设备中的母等级设置的 SPRM(13) 是图 17A 所示的电影对象中的 " kids " (if (SPRM(13) == " kids ")),则 PL#4(Link (PL#4, PL#1)) 被播放,以及如果再现设备中的母等级是除 " kids " 之外的任何其他设置(即, " else "),则 PL#2(Link (PL#2, PL#1)) 被播放。在这里,可以实现所谓的母锁定,因为再现取决于当 PL#2 和 PL#4 被分别假定为过激情景和面向孩子的情景时设置的 SPRM(13) 在过激情节和面向孩子的情节之间进行切换。因为 SPRM(13) 可以利用状态设置例行程序被改变,所以属性信息中的 resume_intension_flag 被设置为 " 0 "。

[0160] 图 17B 显示如上所述的电影对象的再现控制。这个再现控制用于在再现设备中设置 SPRM(13) 为 " kids "。PL#4 被如此播放,因为甚至再现设备中的 SPRM(13) 被设置以显示 " kids " (Link (PL#4, PL#1))。

[0161] 假定当再现设备在 PL#4 上时菜单调用被请求。因为在图 17A 中, resume_intension_flag 被设置为 " 0 ",所以在初始化 SPRM(5)-(8) (图 15A) 之后 SPRM(4)-(8) 被保存(图 15B)。到状态设置例行程序的转移然后被执行。

[0162] 在这里,假定在这个状态设置例行程序中一个操作被执行以更新 SPRM(13),并且状态设置例行程序已经结束。因为 SPRM(4)-(8) 在状态设置例行程序(图 15B) 结束以后执行的重新存储中被返回到再现设备中的寄存器,所以再现位置被设置为标题的头部,并且再现从这个位置重新启动(图 17C)。

[0163] 以上图 14, 16 和 17 所示的例子是涉及 resume_intension_flag 设置的例子。图 18A-18B 显示正在电影对象中设置的标题 _search_mask 的例子。

[0164] 在图 18A 中的电影对象 (0) 是在图 14A 所示的电影对象 (1) 之前执行的电影对象。在这个电影对象中, PL#6 是预览 (1), PL#7 是预览 (2), 以及 PL#5 是视频, 用于使用户选择 PL#6 和 PL#7 的其中之一。所选择的预览的电影对象被设置在 GPRM(0) 中。根据 GPRM(0)

执行 IF 语句的再现切换。Jump 电影对象 (1) 是切换之后执行的转移命令, 电影对象 (1) 是转移目标。因为电影对象 (1) 中的标题 `_search_mask` 被设置为 "1", 所以当电影对象正在执行再现控制时, 标题搜索要求被屏蔽。反之, 如果预览 (1) 和 (2) 的任何一个被观看 (图 18B), 则标题搜索将被激活。因为仅仅通过设置 1 比特标题 `_search_mask`, 控制被实现为 "禁止标题搜索, 直到预览 (1) 和 (2) 的任何一个被观看", 所以增加了描述控制的自由度。让我们与当利用日本专利申请 No. 2856363 执行与图 18B 相同的再现控制时进行比较。根据日本专利申请 No. 2856363, 用户操作的容许性是相对于单个再现路径进行设置的, 其意味着当有大量可以被交替播放的预览时, 设置为禁止用户操作的再现路径的数量必须等于该数量。同样地, 必须被提供的再现路径的数量随可播放的预览的数量一起增加, 如此导致复杂化。

[0165] 相反, 采用图 18A-18B 中的电影对象, 用于拒绝标题搜索直到预览被播放的再现控制可以通过设置电影对象中的标题 `_search_mask` 为 "1" 来简单地进行描述, 甚至当存在大量可交替播放的预览时。

[0166] 因为这个再现控制的简单描述, 当分配标题时, 本实施例的电影对象是有效的。

[0167] 如此结束有关本发明的记录介质的实施例的描述。下列描述涉及本发明的再现设备的实施例。图 19 表示本发明的再现设备的内部结构。关于本发明的再现设备包括两个主要的部分, 就是系统 LSI 和驱动设备, 并且可以通过将这些部分安装到设备的机柜和衬底来进行工业上地生产。系统 LSI 是集成电路, 用来集成用于执行再现设备功能的各种处理单元。如此生产的再现设备包括 DVD 驱动器 1、轨道缓冲器 2、PID 过滤器 4、视频解码器 5、图像平面 6、音频解码器 7、图形平面 8、图形解码器 9、加法器 10、静态脚本存储器 11、再现控制引擎 12、播放器寄存器 13、备份存储器 14、动态脚本存储器 15, 类似 DVD 的模块 16、Java 模块 17、浏览器模块 18、UO 控制器 19、模块管理器 20 和分配器 21。

[0168] BD-ROM 驱动器 1 执行 BD-ROM 的装入 / 弹出, 并且存取装入的 BD-ROM。

[0169] 轨道缓冲器 2 是一个先进 / 先出存储器, 用于存储在先进先出基础上从 BD-ROM 读取的访问单元。

[0170] PID 过滤器 4 从轨道缓冲器 2 检索访问单元, 并且将构造访问单元的 TS 包切换为 PES 包。作为切换的结果获得的期望的 PES 包被输出到视频解码器 5、音频解码器 7 和图形解码器 9 的其中之一。当参考 PES 包的 ID(PID) 时执行输出。其 PID 显示视频的 PES 包被输出到视频解码器 5, 其 PID 显示音频的 PES 包被输出到音频解码器 7, 以及其 PID 显示图形图像的 PES 包被输出到图形解码器 9。

[0171] 视频解码器 5 将通过解码从 PID 过滤器 4 输出的多个 PES 包而获得的未压缩格式图像写入到图像平面 6。

[0172] 图像平面 6 是一个用于存储未压缩格式图像的存储器。

[0173] 音频解码器 7 输出通过解码从 PID 过滤器 4 输出的 PES 包而获得的未压缩格式音频数据。

[0174] 图形平面 8 是一个具有单个屏幕容量区域的存储器, 该单个屏幕容量区域可以存储图形图像的单屏幕值。

[0175] 图形解码器 9 将通过解码图形流而获得的光栅图像写入到图形平面 8。作为解码图形流的结果, 副标题、菜单等等出现在屏幕上。

[0176] 加法器 10 输出将在图形平面 8 展开的图像与图像平面 6 中存储的未压缩格式图像数据合成的结果。

[0177] 静态脚本存储器 11 是用于存储当前的 PL 信息、剪辑信息等等的存储器。当前的 PL 信息是来自记录在 BD-ROM 上的多个 PL 信息当中当前用于处理的信息。当前的剪辑信息是来自记录在 BD-ROM 上的多个剪辑信息当中当前用于处理的信息。

[0178] 再现控制引擎 12 在再现设备中执行各种各样的功能,诸如 AV 再现功能 (1)、播放列表再现功能 (2) 以及状态获取 / 设置功能 (3)。再现设备中的 AV 再现功能包括类似于 DVD 和 CD 播放器中找到的功能组,其指的是响应于诸如播放、停止、暂时打开、暂时关闭、静止关闭、前向播放 (快速)、反向播放 (快速)、音频变化、副标题变化和角度变化等处理的用户操作的执行。PL 再现功能指的是根据 PL 信息进行播放、停止及其他 AV 再现功能的执行。再现控制引擎 12 通过执行 PL 再现功能而执行在分层模型中的第三层的功能 (基于静态脚本的再现控制)。另一方面,再现控制引擎 12 根据来自类似 DVD 模块 16、Java 模块 17 和浏览器模块 18 的功能调用来执行功能 (2) 至 (3)。也就是说,再现控制引擎 12 响应于来自用户操作的指令和来自分层模型的高级层的指令执行再现控制引擎 12 的功能。

[0179] 播放器寄存器 13 包括 32 个单独的系统参数寄存器和 32 个单独的通用寄存器。这些寄存器的被存储值被用来编程 SPRM 和 GPRM。因为系统参数寄存器和通用寄存器是由与模块 16 至 18 分开的再现控制引擎 12 进行管理的,所以例如甚至当再现模式发生变化时,在切换之后执行再现模式的模块也可以简单地通过参考在再现控制引擎 12 中的 SPRM(0)-(31) 和 GPRM(0)-(31) 来找出再现设备的再现状态。

[0180] 备份存储器 14 是一个堆栈存储器,用于当模块 16 至 18 的其中之一执行暂停时保存再现设备寄存器的存储值。当模块 16 至 18 的其中之一执行动态脚本中的恢复时,备份存储器 14 的存储值被重新存储为由再现设备拥有的寄存器的存储值。如果模块 16 至 18 的其中之一执行暂停两次或更多次,寄存器的存储值被以在先进先出的方式存储。如果存储值的数量大于或等于堆栈中的槽的数量,已经被保存的存储值被重写。保存至备份存储器 14 的 SPRM 包括当前正在播放的标题的编号 (标题编号)、当前正在播放的章节编号、当前正在播放的 PL 编号 (播放列表编号)、当前正在播放的播放项目编号 (播放项目编号)、选择状态中的按钮的编号 (选择的按钮)、以及表示当前的再现时间的时间信息。

[0181] 动态脚本存储器 15 是一个存储当前的动态脚本的存储器,并且由类似 DVD 模块 16、Java 模块 17 和浏览器模块 18 共同地进行处理。当前的动态脚本是来自记录在 BD-ROM 上的多个脚本当中的当前用于处理的动态脚本。

[0182] 类似 DVD 模块 16 是一个作为电影模式主执行体的 DVD 虚拟播放器,其执行读取至动态脚本存储器 15 的当前电影对象。

[0183] Java 模块 17 是一个由 java 虚拟机形成的平台、结构和配置文件。

[0184] Java 模块 17 从读取到动态脚本存储器 15 的 ZZZ.CLASS 文件产生当前的 Java 对象,并且执行当前的 Java 对象。java 虚拟机将利用 Java 语言描述的 Java 对象切换为用于再现设备中的 CPU 的本机码,并且使 CPU 执行这些本机码。

[0185] 浏览器模块 18 是一个作为浏览器模式的主执行体的浏览器,执行读取至动态脚本存储器 15 的当前的网页对象。

[0186] U0 控制器 19 检测相对于遥控器、再现设备的前面板等等执行的用户操作,并且输

出表示检测的用户操作的信息（以下“U0 信息”）至模块管理器 20。

[0187] 模块管理器 20 保持从 BD-ROM 读取的索引表, 执行模式管理和转移。由模块管理器 20 执行的模式管理参考模块的分配; 即, 模块 16 至 20 的哪个将执行哪个动态脚本。模块分配的原则是类似 DVD 模块 16 执行动态脚本。这个原则被支持, 即使在转移由内部模式（即, 相同模式内的转移）引起的情况下。一个例外是当模式间转移发生时（即, 模式之间的转移）。当从电影对象到 java 对象 / 网页对象的转移发生时, Java 模块 17 和浏览器模块 18 分别执行当前的对象。

[0188] 分配器 21 仅仅选择适合再现设备的当前模式的 U0, 并且将所选择的 U0 传递到用于执行当前模式的模块上。例如, 如果方向键或者激活 U0 在电影模式执行期间被接收, 则分配器 21 输出这些 U0 至执行电影模式的模块。这些 U0 仅仅是电影模式的菜单动作所需要的, 并且 Java 和浏览器模式不需要这些 U0。

[0189] 由此结束再现设备元件的描述。现在将详细地描述模块管理器 20。

[0190] 模块管理器 20 可以通过具有用于执行图 20 至 22 所示的处理过程的通用 CPU 读取程序来实现。图 20 至 22 是显示由模块管理器 20 执行的处理过程的流程图。由模块管理器 20 执行的转移控制现在将同时参考流程图进行描述。在图 20 的流程图中, 模块管理器 20 从索引表中的第一个播放索引开始检索文件名（步骤 S1）。索引表是有关电影对象的集成信息, 并且第一个播放索引是显示描述 BD-ROM 启动过程的电影对象的索引。

[0191] 一旦该文件名已经被检索, 模块管理器 20 设置当前模式为电影模式（步骤 S2）, 设置检索的文件名的动态脚本作为当前动态脚本（步骤 S3）, 将当前动态脚本 i 读取至存储器（步骤 S4）, 并且执行在存储器中的当前动态脚本（步骤 S5-S9）。

[0192] 每当当前动态脚本被重新设置时, 执行步骤 S4 至 S9。

[0193] 步骤 S5 至 S9 形成循环处理过程, 其中用于每个构造脚本的命令, 重复步骤 S6 至 S9 的处理。流程图中的“x”是一个变量, 用于从构造动态脚本的命令中识别处理目标。循环处理包括模块管理器 20 重复下列处理: 初始化变量 x（步骤 S5）, 使当前模式的模块执行包括在当前动态脚本 i 内的命令 x（步骤 S6）, 执行在步骤 S7 至 S8 定义的判断处理, 然后在返回步骤 S6 之前增加变量 x（步骤 S9）。对于所有的构造脚本的命令, 重复步骤 S6 至 S9 的处理。

[0194] 如果在执行循环处理期间 U0 发生（步骤 S7 = 是）, 模块管理器 20 在经过步骤 S10 至 S12 的判断处理之后将 U0 输出至执行当前模式的模块（步骤 S26）。

[0195] 步骤 S10 是一个用于判断用户操作是否是菜单调用的步骤。如果是菜单调用, 模块管理器 20 在经过步骤 S13 和 S14 的判断之后执行图 21 中的步骤 S15 和 S16 的其中之一保存处理。模块管理器 20 然后将用于执行状态设置的动态脚本设置为当前动态脚本 i（步骤 S17）, 并且返回到步骤 S4。因为用于执行状态设置的动态脚本作为步骤 S17 的结果变成当前动态脚本 i, 所以在步骤 S5 至 S9 执行用于状态设置的动态脚本。

[0196] 步骤 S13 是一个关于当前动态脚本 i 中的 menu_call_mask 是否是“1”的判断。如果是“1”, 模块管理器 20 无需执行任何处理返回到图 20 中的步骤 S8。

[0197] 步骤 S15 是暂停当前的动态脚本 i 并且在备份存储器 14 中保存变量 x 和 SPRM(4) 至 (8) 的处理。如果 resume_intension_flag 是“1”（步骤 S14 = 是）, 则执行步骤 S15。

[0198] 步骤 S16 是在初始化变量 x 和 SPRM(5) 至 (8) 之后暂停当前动态脚本 i 并且在

备份存储器 14 中保存变量 x 和 SPRM(4) 至 (8) 的处理。如果 resume_intension_flag 是 " 0 " (步骤 S14 = 否), 则执行步骤 S16。

[0199] 步骤 S11 是一个关于用户操作是否请求标题搜索的判断。如果标题搜索被请求, 模块管理器 20 在步骤 S18 判断当前动态脚本 i 的标题 _search_mask 是否是 " 1 "。如果是 " 1 " , 模块管理器 20 在步骤 S19 将用于执行标题搜索的动态脚本设置为当前动态脚本 i。

[0200] 步骤 S12 用于执行 U0 的分配处理。U0 的分配处理包括模块管理器 20 判断命令执行期间发生的 U0 是否是方向键或者激活操作 (步骤 S12), 以及如果当前模式是电影模式 (步骤 S20), 则输出 U0 至执行当前模式的模块。如果命令执行期间发生的 U0 并不是方向键或者激活操作, 则 U0 被简单地输出到执行当前模式的模块 (步骤 S26)。如果命令执行期间发生的 U0 是方向键或者激活操作, 但是当前模式不是电影模式, 则 U0 不被输出到模块。由此结束分配处理的描述。

[0201] 用于结束步骤 S4 至 S19 的循环处理的必要条件是在步骤 S8 的判断为是。如果命令 x 是动态脚本 i 中的最后的命令 (步骤 S8 = 是), 判断恢复命令是否是动态脚本 i 中的最后一个 (图 22 中的步骤 S21)。

[0202] 恢复命令是指示再现设备执行作为调用源的动态脚本的状态重新存储的命令。恢复命令被放置在用于状态设置的动态脚本的末端 (即, 状态设置例行程序)。

[0203] 如果恢复命令存在于动态脚本 i 的末端, 模块管理器 20 设置暂停的动态脚本作为动态脚本 i (步骤 S22), 设置动态脚本 i 的模式为当前模式 (步骤 S23), 将保存在备份存储器 14 的 SPRM 重新存储至寄存器 (步骤 S24), 并且将变量 x 返回为保存值 (步骤 S25)。

[0204] 在这里, 因为在 SPRM(4)-(8) 和变量 x 被设置为显示再现位置的值之后它们被保存至存储器 14, 直到如果 resume_intension_flag 是 " 1 " 的时间点, 所以作为在步骤 S24 执行的重新存储处理的结果, 播放器寄存器显示在调用状态设置例行程序之前的再现位置。恢复标题再现的处理被执行, 因为这些值设置在播放器寄存器中。

[0205] 另一方面, 因为如果 resume_intension_flag 是 " 0 " 的话, 在 SPRM(5)-(8) 和变量 x 已经被初始化之后, SPRM(4)-(8) 和变量 x 被保存至存储器 14, 所以播放器寄存器显示当前正在播放的标题的再现开始位置。执行重新启动标题的处理, 因为这些值设置在播放器寄存器中。应当注意到: 虽然在图 20 至 22 的流程图中, 从当前正在播放的标题执行重新启动, 但是可以通过过初始化在步骤 S16 显示再现位置的 SPRM 来相对于整个 BD-ROM 执行重新启动。由此结束由模块管理器 20 执行的处理过程的描述。

[0206] 根据如上所述的当前的实施例, 关于在最上层 (动态脚本) 的菜单调用的控制程序以分层模式设置, 分层模式自下而上包括流、再现路径和动态脚本。特别是, 当用户想要产生用于实现语言片头字幕的标题时, 可以实现其中菜单调用被接受但是再现没有恢复的控制。结果, 可以容易地产生两种标题, 就是允许菜单调用的标题和禁止菜单调用的标题, 即使采用相同的流和再现路径。因为利用标题的产生而没有增大再现路径和流的数量, 所以可以轻松地增加具有不同的控制程序的变化数量。

[0207] 实施例 2

[0208] 实施例 2 涉及待避免的允许再现设备中的停止和重新启动的改进。当任何下列三种情形在再现设备出现时, 再现设备中的停止和重新启动可以发生。

[0209] 1) 当对于相对应于 Java 模式和浏览器模式的 BD-ROM 被装入仅仅相对应于电影模式的再现设备的情况,到 Java 对象或者网页对象的转移发生 M 时。

[0210] 2) 当企图读取不存在的流,或者企图转移到从不存在的动态脚本构造的标题时。

[0211] 3) 当恢复发生在 Java 对象的错误是不可以的时。

[0212] 采用用于避免停止和重新启动的本实施例,与用于异常处理的标题相关的索引被提供在用于集成 / 管理动态脚本的信息中。

[0213] 图 4 所示的 INFO.BD 是用于集成 / 管理电影模式、Java 模式和浏览器模式中的动态脚本的信息。

[0214] 图 23A 显示了 INFO.BD 的内部结构。如图 23A 所示,INFO.BD 包括索引表。索引表是当从一个动态脚本转移到另一个动态脚本时引用的间接引用表,包括与多个标签一对一相应的索引。在每个索引中,描述了对应于索引标签的动态脚本的文件名。如图 23B 所示,每个文件名包括文件主体和扩展。标签包括标题 #1 ~ #m、标题 #m+1 ~ #n 和标题 #0。索引表也从三种模式的任何一个的动态脚本进行参考。当经由索引表时,从电影对象到 Java 对象或者从电影对象到网页对象的转移是唯一可以的。即,不可以从电影对象转移到 Java 或者网页对象,网页对象在索引表中没有索引。

[0215] 标题 #1 ~ #m 索引涉及记入在 BD-ROM 中的第一到第 m 个标题。在这些索引中,描述了当第一到第 m 个标题标号被选择时作为转移目标的电影对象的文件名。图 23B 显示标题 #1 ~ #m 的内容。如图 23B 所示,在标题 #1 ~ #m 索引中,描述了电影对象的文件名。每个文件名包括文件主体 (ZZZ) 和扩展 (.MOVIE)。

[0216] 标题 #1 ~ #n 索引涉及记入 BD-ROM 中的第一到第 m+1 个标题。在这些索引中,描述了当 m+1 到第 n 个标题编号被选择时作为转移目标的网页对象 / Java 对象的文件名。图 23C 显示标题 #m+1 ~ #n 索引的内部结构。如图 23C 所示,在索引标题 #m+1 ~ #n 中的每一个中存储 Java 对象的文件主体 (ZZZ) 和扩展 (.CLASS) 或者网页对象的文件主体 (ZZZ) 和扩展 (.HTM)。应当注意:索引格式可以如图 23D 所示。图 23D 中的索引格式具有显示转移目标标题的属性的属性区域,被构造来在属性区域中显示相应的转移目标标题的动态脚本是电影模式 ("00" 设置),Java 模式 ("01" 设置) 还是浏览器模式 ("10" 设置)。

[0217] 标题 #0 索引涉及异常处理标题,而且存储电影模式脚本的文件名。当上述三种情形的任何一种发生时,在这里描述的异常处理被执行。其中增强模式执行不可以用于这些三中情况的任何一个的再现设备被称作核心系统。另一方面,其中利用 java 虚拟机或者浏览器的程序执行可以的再现设备被称作完全系统。由核心系统和完全系统进行的 BD-ROM 的间接引用描述如下,同时参考图 24A-24B。间接引用的描述假定 BD-ROM 上记录了多个动态脚本 (001. 电影、002. 电影、003. 电影、...、001. CLASS、002. CLASS、003. CLASS、...),如图 24A 所示。图 24B 显示了当图 24A 所示的多个动态脚本被记录在 BD-ROM 上时索引表的示范性的描述。在图 24B 所示的示范性的描述中,在标题 #1 索引到标题 #m 索引中被描述电影模式脚本的文件名 (001. 电影、002. 电影、003. 电影、...)。另一方面,在标题 #m+1 索引到标题 #n 索引中被描述增强模式脚本的文件名 (001. CLASS、002. CLASS、003. CLASS、...)。

[0218] 图 25A 显示了当索引表被描述为图 24B 中那样时完全系统中的间接引用。因为索引表被描述成这样,所以当执行指定标签标题 #1 到标题 #m 作为转移目标的转移命令时,从标题 #1 索引到标题 #m 索引中检索文件名 "001. 电影、002. 电影、003. 电影、...",并且

当执行指定标签标题 #m+1 到标题 #n 作为转移目标的转移命令时,被从标题 #m+1 索引到标题 #n 索引中检索文件名" 001. CLASS、002. CLASS、003. CLASS、..."。用这些文件名表示的动态脚本然后被读取到存储器并且被执行。由此结束完全系统的间接引用的描述。

[0219] 图 25B 显示核心系统中的间接引用。当执行指定标签标题 #1 到标题 #m 作为转移目标的转移命令时,从标题 #1 索引到标题 #m 索引中检索文件名" 001. 电影、002. 电影、003. 电影、..."。然而,当执行指定标签标题 #m+1 到标题 #n 作为转移目标的转移命令时,从标题 #0 索引而不是标题 #m+1 索引到标题 #n 索引中检索文件名" 000. 电影"。再现设备然后执行用这些文件名表示的动态脚本。由此结束完全系统和核心系统两者的间接引用的描述。

[0220] 图 26 示意地显示从电影对象到 Java 对象的转移是怎样被执行的。图 26 中的电影对象包括其中 GPRM(0) 被设置为" 0" 的预命令、指示再现设备执行 PL 再现的命令(PlayPL(PL#1))、以及指示再现设备执行到另一个动态脚本的转移的后命令(IF(GPRM(0) = 0){JmpTilte#m}else{Jmp Title#m+1})。作为该预命令的结果,GPRM(0) 在 PL 再现之前被初始化。同时,作为该后命令的结果,当初始化时如果 GPRM(0) 显示" 0",则到电影 object#m+1 的转移被执行。另一方面,当菜单被显示而且 GPRM(0) 被设置为除了" 0" 之外的值时,如果按钮选择被执行,则到另一个标题(标题 #m)的转移被执行。

[0221] 用于实现如下所述的对话处理的交互图形流被多路复用到 AVClip。

[0222] 交互图形流是显示相应于字符 A、B 和 C 的按钮的流,当字符 A 被确定时 GPRM(0) 被设置为" 1",当字符 B 被确定时被设置为" 2",以及当字符 C 被确定时被设置为" 3"。

[0223] 图 26 中的箭头 jn1 和 jn2 示意性地指示从电影对象到 Java 对象的转移。图 26 中的 Jmp 标题 #m+1 是 Java 对象中的转移命令,以及经由标签标题 #m+1 的索引,利用间接引用格式来指定 Java 对象作为转移目标。Java 对象的文件名在标签标题 #m+1 的索引中被描述,通过参考这个索引再现设备能够找出哪个文件被读取作为 Java 对象。

[0224] 在 Java 对象中," A.drawCharacter();"意味着字符 A 的对象利用类" 字符 A" 的方法的其中之一在屏幕上被提取(即,图 26 中的 drawCharacter 功能)。同样地," B.drawCharacter();"和" C.drawCharacter();"分别意味着字符 B 和 C 的对象利用类" 字符 B" 和" 字符 C" 的方法的其中之一在屏幕上被提取(即,图 26 中的 drawCharacter 功能)。

[0225] 因为完全取决于 GPRM(0) 的值来执行" A.drawCharacter();、" B.drawCharacter();"和" C.drawCharacter();" (图 26 中的 IF 语句),所以如果 GPRM(0) 是" 1",则字符 A 的 CG 被提取,如果 GPRM(0) 是" 2",则字符 B 的 CG 被提取,如果 GPRM(0) 是" 3",则字符 C 的 CG 被提取。

[0226] 图 27 显示当图 26 所示的脚本记录在其上的 BD-ROM 被装入核心系统再现设备中时哪种转移被执行。利用图 27 中的虚线 hs1 描述图 26 中的箭头显示图 26 中的转移不再有效,因为核心系统缺少用于执行 Java 对象的元件。图 27 中的箭头 js1 显示在执行的异常处理中使用的转移,而不是无效的转移。用于异常处理的转移是经由标题 #0 的索引的间接引用。电影对象 sg1 的文件名被存储在标题 #0 的索引中,电影对象 sg1 正在由再现设备读取并且在这个转移中执行。因为当 BD-ROM 被装入仅仅具有核心系统的再现设备时显示电影对象中的视频,所以可以避免停止和重新启动。

[0227] 由此结束有关实施例 2 中的 BD-ROM 的改进的描述。现在将描述再现设备侧的改进。

[0228] 实施例 2 中的模块管理器 20 的特征是转移控制。转移控制将识别为转移目标的动态脚本读取到存储器,以及使类似 DVD 模块 16, Java 模块 17 和浏览器模块 18 的其中之一执行动态脚本。当利用间接引用格式指定转移目标动态脚本时,识别是必需的。通过参考转移命令的转移目标标签和从相应于该标签的索引中检索文件名,执行识别。结合该识别,判断模式切换是否是必需的。通过参考对应于转移目标标签的索引以便确定存储的文件扩展或者由存储的属性信息显示的模式,来执行模式切换判断。索引的存储内容揭示模式切换是否是必需的。如果模式切换是必需的,转移目标动态脚本被读取到存储器,并且模式转变请求被输出到执行后切换模式的模块。作为模式转变请求输出的结果,执行后切换模式的模块执行存储器中的转移目标动态脚本。

[0229] 作为模块管理器 20 执行上述转移的结果,实施例 2 中的模块管理器 20 的处理过程如图 28 所示。图 28 基于图 20 所示的流程图,描绘了两个流程图之间的差异。

[0230] 尽管当前动态脚本 i 中的命令通过重复步骤 S6 至 S9 一次一个地被执行,但是步骤 S30 已经新近被加到图 28 流程图中的步骤 S6 至 S9 的循环处理中。

[0231] 步骤 S30 是一个关于命令 x 是否是转移命令的判断。如果步骤 S30 为是,在步骤 S31 至 S43 设置当前动态脚本为新的动态脚本之后,模块管理器 20 返回到步骤 S4。结果,新的动态脚本被读取至存储器并且被执行。

[0232] 下列描述涉及步骤 S31 至 S43 中的处理。这个处理包括转移控制,并且取决于步骤 S31、S34、S39 和 S42 的判断结果而不同。步骤 S31 是关于转移命令的转移目标是否利用标题标签进行描述的判断。如果为是,模块管理器 20 在经过步骤 S42 判断之后获得转移目标标签标题 j(步骤 S32),并且从索引表中的标题 j 的索引 I 中检索文件名 j(步骤 S33)。如果为否,模块管理器 20 检索显示转移目标的文件名 j(步骤 S41)。

[0233] 步骤 S34 是关于转移命令是调用命令还是 Jmp 命令的判断。如果是调用命令,模块管理器 20 在暂停当前的动态脚本 i 之后保存变量 x 和 SPRM(步骤 S35)。如果是 Jmp 命令,模块管理器 20 丢弃当前的动态脚本 i(步骤 S36)。

[0234] 已经经过上述处理,模块管理器 20 设置从文件名 j 识别的动态脚本作为当前动态脚本 i(步骤 S37),并且设置从检索的扩展识别的再现模式作为再现模式 k(步骤 S38)。在这些设置之后,模块管理器 20 执行步骤 S39。步骤 S39 是关于再现模式 k 是否是当前再现模式的判断。如果不相同,模块管理器 20 设置再现模式 k 作为当前再现模式(步骤 S40),并且转入步骤 S4。之后,步骤 S4 至 S9 的处理相对于新近设置的当前的动态脚本被重复。步骤 S42 是关于再现设备是核心系统还是完全系统的判断,并且如果是核心系统,模块管理器 20 从标题 #0 的索引中检索文件名,并将其设置为转移目标(步骤 S43)。

[0235] 因为当由于某种原因增强模式遇到困难时,执行再现设备被设置为核心系统,并且转移被执行,同时参考用于异常处理的索引表中的索引,所以可以根据如上所述的实施例避免停止、重新启动等等。

[0236] 实施例 3

[0237] 实施例 3 涉及当各种规范的再现设备和 BD-ROM 被引入时的改进。当存在快速商业化 BD-ROM 和再现设备的强劲的压力时,具有很少可支持的功能的 BD-ROM 版本,诸如仅仅支

持电影模式的版本 1.0 和支持电影模式和增强模式的版本 1.1 停止被使商业化,并且被从投放市场。在这种情况下,市场具有许多版本的再现设备,诸如版本 1.0 和版本 1.1BD-ROM,以及版本 1.0 和版本 1.1 再现设备。因此,从电影模式中的电影对象到增强模式中的电影对象的转移例如可以发生在装入 1.0 版本再现设备中的 1.1 版本的 BD-ROM。在这种情况下,不可以执行增强模式中的电影对象,因为版本 1.0 再现设备仅仅具有用于电影模式的模块。由此采用当前的实施例,有关所有可利用的版本的索引表被记录在 BD-ROM 上。图 29A 是一个版本 1.1 的 BD-ROM。版本 1.1 索引表和版本 1.0 索引表被记录在图 29A 中的 BD-ROM 上。标题 #1 索引至标题 #m 索引存在于版本 1.0 索引表中。如实施例 2 所示,当转移至电影模式的动态脚本时,这些索引被参考。

[0238] 标题 #1 索引至标题 #m 索引、标题 #m+1 索引至标题 #n 索引,以及标题 #0 索引存在于版本 1.1 索引表。如实施例 2 所示,当转移至电影模式的动态脚本,增强模式的动态脚本以及用于异常处理的动态脚本时,这些索引被参考。

[0239] 当这些版本的 BD-ROM 的其中之一被装入再现设备时,再现设备利用在有关记录在 BD-ROM 上的多个版本的索引表当中与再现设备的版本相匹配的索引表选择电影对象。

[0240] 图 29B 假定其中图 29A 所示的 BD-ROM 被装入版本 1.0 再现设备的情况。因为图 29B 中的再现设备是版本 1.0,所以当转移发生时,转移目标电影对象通过参考版本 1.0 以及 1.1 索引表当中的版本 1.0 索引表被识别。

[0241] 图 29C 假定其中图 29A 所示的 BD-ROM 被装入版本 1.1 再现设备的情况。因为图 29C 中的再现设备是版本 1.1,所以当转移发生时,转移目标电影对象通过参考版本 1.0 以及 1.1 索引表当中的版本 1.1 索引表被识别。

[0242] 为了执行上述处理,根据实施例 3 的再现设备中的模块管理器 20 基于图 30 中的流程图执行处理。当 BD-ROM 被装入再现设备时,模块管理器 20 获得设备中的版本号(步骤 S45),读取记录在 BD-ROM 上的多个索引表中与获取的版本号相匹配的索引表,并且保持读取的索引表(步骤 S46)。模块管理器 20 然后执行步骤 S1 至 S42 的处理,同时参考保存的索引表。与实施例 2 所示的处理相同的步骤 S1 至 S42 的处理的描述在这里被省略。

[0243] 根据如上所述的本实施例,甚至当各种版本的再现设备和 BD-ROM 能在市场上见到时,通过选择匹配再现设备版本的索引表和执行关于这个索引表的再现,可以保证与老版本的再现设备的兼容性。

[0244] 实施例 4

[0245] 本实施例涉及当实现与 BD-ROM 上的 DVD 类似的菜单控制时的改进。图 31 表示由 BD-ROM 实现的菜单分级结构。图 31 中的菜单分级结构被构造成在最高等级放置顶端菜单,并且能够从顶端菜单选择所隶属标题菜单、副标题菜单和音频菜单。图 31 中的箭头 sw1, sw2 和 sw3 示意性地显示由按钮选择切换的菜单。顶端菜单安排用于接收音频选择、副标题选择和标题选择的哪一个将执行的按钮(图 31 中的按钮 sn1、sn2、sn3)。

[0246] 标题菜单安排用于接收电影作品选择(诸如选择电影作品的影院版本(标题)、编导的剪辑版本或者游戏版本)的按钮。音频菜单安排用于接收音频再现是否将是日语或者英语的按钮,并且副标题菜单安排用于接收副标题是否以日语或者英语显示的按钮。

[0247] 在图 32 中示出了用于操作具有这样的分级结构的菜单的电影对象。

[0248] 第一播放对象(FirstPlayOBJ)是描述当在再现设备中装入 BD-ROM 时的启动过程

的动态脚本。正方形方框表示表示用于执行这个设置过程的命令的第一播放对象。第一播放对象的最后的命令是转移命令,该转移目标是顶端菜单对象。

[0249] 顶端菜单对象 (TOPMenu OBJ) 是用于控制顶端菜单性质的动态脚本。顶端菜单对象是当用户请求菜单调用时调用的对象,并且等同于实施例 1 中提到的状态设置例行程序。表示顶端菜单对象的正方形方框示意出了表示这个控制程序的各个命令。这些命令包括:用于响应于用户的操作改变顶端菜单中的按钮的状态的命令、以及用于响应于按钮的激活进行转移的转移命令。转移命令实现从顶端菜单至标题菜单,从顶端菜单至副标题菜单,以及从顶端菜单至音频菜单的菜单切换。

[0250] 音频菜单对象 (AudioMenu OBJ) 是用于控制音频菜单性质的动态脚本。构造音频菜单对象的正方形方框示意出了表示这个控制程序的各个命令。这些命令包括:用于响应于用户的操作改变音频菜单中的按钮的状态的命令、以及用于响应于按钮的激活更新用于音频设置的 SPRM 的命令。

[0251] 副标题菜单对象 (OBJ) 是用于控制副标题菜单性质的动态脚本。构造副标题菜单对象的正方形方框示意出了表示这个控制程序的各个命令。这些命令包括:用于响应于用户的操作改变副标题菜单中的按钮的状态的命令、以及用于响应于按钮的激活更新用于音频设置的 SPRM 的命令。

[0252] 标题菜单对象 (TitleMenu OBJ) 是用于控制标题菜单的性质的动态脚本。标题菜单对象是当用户请求标题搜索时调用的对象,并且等同于实施例 1 中提到的用于标题搜索的动态脚本。构造标题菜单对象的正方形方框示意出了表示这个控制程序的各个命令。这些命令包括:用于响应于用户的操作改变标题菜单中的按钮的状态的命令,以及用于响应于按钮的激活进行转移的转移命令。转移命令实现至各个标题的转移。

[0253] 诸如在 DVD 中实现的菜单性质可以通过这些供菜单使用的电影对象来实现。由此结束有关菜单控制的电影对象的描述。

[0254] 现在将描述本实施例中的索引表的改进。第一播放索引、顶端菜单索引、音频菜单索引、副标题菜单索引以及标题菜单索引被加到本实施例中的索引表。如实施例 1 中所述,这些索引还通过与三种模式的每一个有关的动态脚本被参考。

[0255] 第一播放索引在 BD-ROM 启动期间被参考。第一播放对象的文件名描述在这个索引中。

[0256] 当进行用户操作以直接调用音频菜单、副标题菜单以及标题菜单时,顶端菜单索引、音频菜单索引、副标题菜单索引以及标题菜单索引被参考。通过用户按压下遥控器上的音频选择键、副标题选择键或者标题选择键,实施用户的直接调用。

[0257] 由此结束在本实施例中的电影对象的改进的描述。现在将描述本实施例中的再现设备的改进。为了操作诸如这些电影对象,模块管理器 20 需要执行图 33 的流程图所示的处理过程。

[0258] 在本实施例中,最初执行菜单控制的模块管理器 20 利用图 33 所示的处理过程执行转移。这个流程图不同之处在于:在步骤 S30 和 S31 之间插入步骤 S50。如果在步骤 S50 为是,模块管理器 20 执行步骤 S51 至 S54 处理,并且返回到步骤 S4。步骤 S51 至 S54 涉及用于实施菜单控制的脚本设置作为当前动态脚本。也就是说,如果转移命令的转移目标是 xxxMenu(步骤 S50 = 是),则模块管理器 20 暂停当前动态脚本 i,保存 SPRM 和变量 x(步

骤 S52), 从对应于转移目标菜单的索引中检索文件名 (步骤 S52), 将检索的文件名的动态脚本设置为当前动态脚本 (步骤 S53), 并且将当前模式返回到电影模式 (步骤 S54)。然后模块管理器 20 继续执行当前动态脚本。

[0259] 因为经由索引表的索引, 利用间接引用来实现至用于菜单控制的动态脚本的转移, 所以根据如上所述的本实施例, 可以转移到用于菜单控制的动态脚本, 甚至当 Java 模式或者浏览器模式的执行期间菜单键被按下时。可以进行从 Java 虚拟机和浏览器模式切换的音频和副标题, 由此甚至当利用 java 虚拟机或者浏览器模式执行再现时, 也可以实现类似于正常 DVD 的音频和副标题切换。

[0260] 实施例 5

[0261] 实施例 5 涉及用于通过为电影模式提供的数据防止任何可以施加于其它模式上的不利影响的改进。通过电影对象和多路复用到 AVClip 上的交互图形流中的命令 (按钮命令), 可以执行电影模式中的控制。

[0262] 当激活通过图形流描述的按钮时, 执行按钮命令。将按钮命令并入 AVClip 便于再现控制的描述, 其中该再现使得再现设备于根据特定的运动图像的各个帧出现在屏幕上的定时执行特定处理; 也就是说, 再现控制与运动图像内容接近同步。同样, 因为按钮命令多路复用到在实际的 AVClip 上, 所以没有必要将对应于 AVClip 的所有按钮命令都存储在存储器中, 甚至当有几百个分段想要执行再现控制时。因为对于每个访问单元, 按钮命令是连同视频包一起从 BD-ROM 中读取的, 优选地, 使得对应于用于当前再现的运动图像部分的按钮命令驻留在存储器中, 然后当再现这个运动图像部分时从存储器删除这些按钮命令, 并且在存储器中存储相应于下一个运动图像部分的按钮命令。因为按钮命令多路复用到 AVClip 上, 所以可以减少安装的存储器至最低的要求数量, 甚至当例如有几百个按钮命令时。

[0263] 当按钮命令被嵌入流中时, 问题出现于 Java 模式的动态脚本的干扰。例如, 如果当以 Java 模式执行再现控制时, 嵌入流中的按钮命令被提供给模块, 则 Java 模式动态脚本和按钮命令同时停止执行, 引起播放器错误。采用解决这个问题的本实施例, 播放项目被提供有滤波器规范功能。

[0264] 滤波器规范涉及多路复用在 AVClip 上的可播放的和不可播放的基本流之间进行区别。

[0265] 图 34 显示了关于实施例 5 的播放项目的内部结构。" Playable_PID_entries" 已经被加入图 34。图 34 中的引出线 hp1 突出了 Playable_PID_entries 的结构。正如以下所揭示的, Playable_PID_entries 列举用于再现的 PID 基本流。

[0266] 下列描述涉及通过播放项目中的滤波器规范实现哪一个再现控制。图 35 显示了其中通过 Java 对象执行再现控制的 PL 的分级结构。图 35 中的等级 4 处的电影对象包括用于播放 PL#1 的命令 (PlayPL(PL#1))。构造 PL#1 的三个播放项目 s#1, #2 和 #3 中的播放项目 #3 包括 Playable_PID_entries, 意思是说滤波器规范是可以的。

[0267] 图 35 中的等级 4 处的 Java 对象包括用于播放 PL#2 的命令 (PlayPL(PL#2))。构造 PL#2 的两个播放项目中的播放项目 #12 包括 Playable_PID_entries, 意思是说滤波器规范是可以的。

[0268] 图 36 显示了通过在播放项目 #3 和 #12 中的 Playable_PID_entries 执行的滤波器

规范。在图 36 中,构造 AVClip 的访问单元被显示在底部,以及两个播放项目 s#3 和 #12 被显示在顶部。一个视频流、三个音频流、两个呈现图形流和一个交互图形流在访问单元中被多路复用。" Video_PID" PID 被附加到视频流," Audio_PID" PID 被附加到音频流," P. Graphics_PID" PID 被附加到呈现图形流,以及" I. Graphics_PID" PID 被附加到交互图形流。三个音频流中,具有附加的" Audio_PID1" 的一个是英语音频 (0:英语),具有附加的" Audio_PID2" 的一个是日语音频 (1:日语),并且具有附加的" Audio_PID3" 的是评论音频 (2:评论)。两个呈现图形流中,具有附加的" P. Graphics_PID1" 的一个是英语音频 (0:英语) 并且具有附加的" P. Graphics_PID2" 的一个是日语音频 (1:日语)。

[0269] 在图 36 顶部的播放项目 #3 和 #12 具有不同的滤波器规范。播放项目 #3 和 #12 中的正方形的列举是 Playable_PID_entries 的实际的内容,播放项目 #3 被设置为允许 Video_PID 视频流、Audio_PID1 和 Audio_PID2 音频流、P. Graphics_PID1 和 P. Graphics_PID2 呈现图形流、以及 I. Graphics_PID 交互图形流的再现。播放项目 #12 被设置为允许 Video_PID 视频流以及 Audio_PID3 音频流的再现。当播放播放项目 #3 时,播放项目 #3 中的 Playable_PID_entries 被设置为再现设备中的 PID 过滤器 4。结果,PID 过滤器 4 输出 Video_PID 视频流至视频解码器 5,输出 Audio_PID1 以及 Audio_PID2 音频流至音频解码器 7,并且输出 P. Graphics_PID1 以及 P. Graphics_PID2 呈现图形流和 I. Graphics_PID 交互图形流至图形解码器 9。因为播放项目 #3 被设置,以便所有的图形流是可播放的,所以 AVClip 上多路复用的所有的图形流的再现是可以的。

[0270] 另一方面,因为播放项目 #12 被设置,以便并非所有的图形流是可播放的,所以利用 Java 语言的控制是可以的,同时没有来自 Java 模式的动态脚本的干扰。

[0271] 图 37 显示了由播放项目 s#3 以及 #12 引起的可以的再现输出。因为 Video_PID 视频流、Audio_PID1 以及 Audio_PID2 音频流、P. Graphics_PID1 和 P. Graphics_PID2 呈现图形流、以及 I. Graphics_PID 交互图形流的再现采用播放项目 #3 是可以的,所以可以采用利用电影对象的再现执行视频流的再现输出,接着 Audio_PID1 音频流 (即,图 37 中的叙述" She'sacaptiveofherownlies")、P. Graphics_PID1 呈现图形流 (日语副标题" 彼女は自分のうそにいしれた") 以及 I. Graphics_PID 交互图形流 (继续? ●是◎否) 的再现输出。

[0272] 播放项目 #12 被设置,以便并非所有的图形流是可播放的,使得可以仅仅执行两个流的再现输出;即,Video_PID 视频流以及 Audio_PID3 音频流。如果指示这个播放项目再现的 Java 对象提取虚拟演播室 (即,包含图 37 中的照相机,椅子以及灯光的房屋),则用于执行提取的 Java 对象将不接收来自包括在图形流中的干扰。由此可以实现 Java 模式的特定处理,同时避免来自包括在图形流中的干扰。由播放项目 #12 设置为可播放的 Audio_PID3 音频流是电影编导的评论 (即,行" It ake my hat off to her outstanding actingability"),并且通过在演播室播放编导的这样评论,可以产生电影设置的气氛。

[0273] 作为这个 Java 对象的结果,可以倾听电影编导的评论,同时播放电影情节作为在电影演播室中仿制的房间中的背景图像。

[0274] 通过在 BD-ROM 上记录这个标题作为收益 (bonus) 轨道标题,可以增加 BD-ROM 的产值。利用播放项目中的滤波器规范以在 BD-ROM 上 记录收益轨道标题导致下列优点。

[0275] 世界闻名的电影编导的评论对影迷来说肯定有兴趣,并且存在于现行的 DVD

上,来增加电影作品附加价值。

[0276] 虽然能够倾听编导的评论是这个标题的最大的吸引力,播放作为背景图象的电影情节同样促进增加标题的吸引力。换句话说,能够倾听有关电影产品的幕后的交谈,同时从电影中观看精彩场面情节增加了评论的气氛。在这种情况下的问题变成如何处理有关评论的音频流的问题。传统方法是提供人们想要用作从主要特征分开的背景图象的电影情节,并且将这些与音频流进行多路复用,以便产生收益轨道。然而,这个方法意味着用作背景图象的电影情节需要与主要特征分开记录在 BD-ROM 上,这增加记录项目的数量并且产生有关容量的问题。

[0277] 另一个可能的方法涉及在用于主要特征的视频流上连同用于主要特征的音频流一起多路复用用于评论的音频流。这允许来自主要特征的情节被用作评论的背景图象,尽管在这里危险是当播放主要特征时评论数据也被听到。同样地,设置构造主要特征标题的播放项目中的滤波器规范,以便仅仅评论的音频流被关闭,任何剩余的音频流是打开的。另一方面,设置构造收益标题的播放项目中的滤波器规范,以便仅仅评论的音频流被打开,任何剩余的音频流是打开被关闭。通过做这些,优选地,在记录在 BD-ROM 上的单个 AVClip 上一起多路复用所有的有关主要特征和评论的音频流。

[0278] 因为没有必要产生用于评论和主要特征的分开的 AVClip(即,仅仅用于主要特征的音频流的 AVClip,和仅仅用于评论的音频流的 AVClip),可以减少用于记录在 BD-ROM 上的 AVClip 的数量,并且使得编辑更简单。

[0279] 由此结束有关实施例 5 中的 BD-ROM 的改进的描述。现在将描述实施例 5 中的再现设备的改进。

[0280] 实施例 5 中的再现设备执行的处理是通过执行图 38 中的处理过程的再现控制引擎 12 实现的。

[0281] 图 38 是一个显示由再现控制引擎 12 执行的 PLPlay 函数的执行过程的流程图。在该流程图中,PLx 是用于处理的 PL,PIy 是用于处理的 PI,并且访问单元 v 是用于处理的访问单元。这个流程图包括下列过程:设置由 PLPlay 函数的自变量 PLx 指定的 PL;将 PLx 读取到存储器中(步骤 S61),识别用于处理的 PI(步骤 S62 至 S64),以及读取构造该 PI 的访问单元(步骤 S65 至 S76)。

[0282] 步骤 S62 是一个关于是否有 PI 自变量规范的判断。如果有自变量规范,再现控制引擎 12 将 PIy 设置为自变量指定的 PI,而且设置 PIz 为相同的自变量指定的 PI(步骤 S63)。PIz 是定义读取范围末端的 PI。PIy 和 PIz 被设置为自变量指定的 PI,因为在 PI 由自变量指定的情况下,才需要读取该 PI。

[0283] 如果没有自变量规范,再现控制引擎 12 将 PIy 设置为 PLx 中的头部 PI,以及设置 PIz 为 PLx 中的最后 PI(步骤 S64)。

[0284] 步骤 S65 至 S76 显示构造 PIy 的访问单元的读取和解码过程。这个过程涉及将在 PIy 中的 Playable_PID_entries 设置为 PID 过滤器 4(步骤 S65),从 EP_map 设置包括 PIy 中的入点视频帧的访问单元 v(步骤 S66),指示 BD-ROM 驱动器 1 读取访问单元 v(步骤 S67),然后,在经过步骤 S68 至 S69 的判断之后,命令视频解码器 5 解码包括在访问单元 v 中的视频帧(步骤 S70),以及设置访问单元 v 为下一个访问单元(步骤 S71)。在那之后,对于属于 Piy 的所有访问单元,重复步骤 S67 至 S71 的处理。

[0285] 步骤 S68 是关于访问单元 v 是否包括入点视频帧的判断。如果包括入点视频帧 (步骤 S68 = 是), 再现控制引擎 12 指示视频解码器 5 对从访问单元 v 中的入点视频帧到最后的视频帧进行解码 (步骤 S72), 以及移到步骤 S70。

[0286] 步骤 S69 是关于访问单元 v 是否包括出点视频帧的判断。如果包括出点视频帧 (步骤 S69 = 是), 再现控制引擎 12 指示视频解码器 5 对从访问单元 v 中的头部视频帧到出点视频帧进行解码 (步骤 S73), 以及从 PID 过滤器 4 释放 PIy 中的 Playable_PID_entries (步骤 S74)。结果, PIy 的滤波器规范被设置为关闭。然后执行步骤 S75 的判断。步骤 S75 是流程图中最后的判断, 其判断 PIy 是否现在是 PIz。如果步骤 S75 为是, 再现控制引擎 12 结束流程图, 并且如果为否, 在返回到步骤 S65 之前, 将 PIy 设置为下一个 PI。在那之后, 重复步骤 S65-S77 的处理, 直到在步骤 S75 的判断为是。由此结束再现控制引擎 12 执行的处理过程的描述。

[0287] 因为播放项目被提供有滤波器规范, 其中滤波器规范设置在 AVClip 中多路复用的多个基本流的哪一个可播放, 哪一个不可播放, 所以作为在选择兼容的播放项目的每个模式中的动态脚本的结果, 可以根据本实施例避免在 AVClip 上多路复用的基本流中的按钮命令施加的任何影响。同样地, Java 模块 17 不再接收来自按钮命令的任何干扰, 其有助于再现设备的稳定运行。

[0288] 实施例 6

[0289] 本实施例涉及 BD-ROM 制造处理。图 39 是一个显示实施例 6 的 BD-ROM 制造处理的流程图。

[0290] BD-ROM 制造处理包括: 材料制造处理 S101, 用于产生诸如运动图像记录和音频记录的材料; 著作处理 S102, 用于产生应用格式; 以及压制处理 S103, 用于产生 BD-ROM 母片以及压制 / 层叠, 以形成 BD-ROM。

[0291] 在这项处理中, 瞄准 BD-ROM 的著作处理包括步骤 S104 至 S109 的处理。

[0292] 脚本编辑处理 S104 用于将在计划阶段产生的纲要转换为再现设备可理解的格式。产生脚本编辑结果, 作为 BD-ROM 脚本。同样, 在脚本编辑中也产生多路复用参数, 以便实现多路复用。

[0293] 一旦在这些处理中, 已经完成动态脚本, 则在步骤 S105 设置每个动态脚本的 resume_intension_flag、menu_call_mask、Title_search_mask。根据由动态脚本施加于再现控制上的 SPRM 的效果, 执行这些设置。作为这些设置的结果, 防止了由再现期间的菜单调用和标题搜索引起的不利影响。

[0294] 材料编码处理 S106 是用于分别编码视频、音频和次视频材料以获得视频、和音频以及图形流的任务。

[0295] 在多路复用处理 S107 中, 交织多路复用作为材料编码的结果获得的视频、音频以及图形流, 并且结果被转换成单个数字流。

[0296] 在格式化处理 S108, 基于面向 BD-ROM 的脚本, 产生各种类型的信息, 并且脚本和数字流适合于 BD-ROM 格式。

[0297] 仿真处理 S109 用于证实著作结果是否是正确的。

[0298] 因为能够利用 Java 和标记语言描述 Java 对象和网页对象, 所以可以在如上所述的著作处理中利用与在标准面向计算机的软件的发展过程中应用的相同可理解性来开发

Java 对象和网页对象。因此,本实施例具有增加脚本创作效率的效果。

[0299] 备注

[0300] 以上描述并没有显示本发明所有配置的实现。根据执行下列修改 (A)、(B)、(C)、(D),... 的配置的实现,本发明的实现仍然是可以的。由本申请的权利要求书相关的发明的范围是从上述公开的多个实施例及其修改的配置的展开的公开内容到概括的公开内容。展开或者概括的程度基于申请时的本发明的技术领域中的技术标准的特性。

[0301] 然而,因为与权利要求书有关的本发明反映了解决有关现有技术的技术问题的手段,所以与权利要求书有关的本发明的技术范围没有超出本领域技术人员相对于解决有关现有技术的技术问题所辨别的 技术范围。同样地,与本申请的权利要求书有关的本发明具有与详细说明中公开内容相对应的材料。

[0302] (A) 在所有的实施例中,关于本发明的光盘是作为 BD-ROM 被实现的。然而,本发明的光盘的特点在于记录的动态脚本和索引表,这些特征不取决于 BD-ROM 的物理性能。任何形式的记录介质是适用的,只要有记录动态脚本和索引表的容量。例如,诸如 DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R、CD-RW 等等的光盘,以及诸如 PD、MO 等等的光磁盘是可适用的。像致密闪存片、PCM-CIA 卡等等这样的半导体卡同样是可适用的,诸如 (i) 磁记录盘、诸如软盘、超级盘,Zip、Clik! 等等以及 (ii) 可移动的硬盘驱动,诸如 ORB、Jaz、SparQ、SyJet、EZFley、microdrive 等等也是同样适用的。此外,记录介质同样可以是内置硬盘。

[0303] 动态脚本、索引表以及播放列表信息可以记录在不同于与记录 AVClip 和流管理信息不同的介质上。然后,这些信息可以被并行读取并作为单个视频编辑被播放。

[0304] (B) 尽管所有实施例中的再现设备在解码之后输出记录在 BD-ROM 上的 AVClip 到 TV,但是再现设备可以仅仅包括 BD-ROM 驱动,并且 TV 可以设置所有的其它元件。在这种情况下,再现设备和 TV 可以被并入利用 IEEE1394 连接的本地网络。同样,尽管实施例中的再现设备属于在连接电视之后使用的类型,但是整体的显示-再现设备同样是可适用的。此外,再现设备可以仅仅是执行处理的基本部分的实施例的再现设备的一部分。因为这些再现设备是在本申请的说明书中公开的所有的发明,所以涉及基于实施例 1 至 6 所示的再现设备的内部结构的再现设备制造的动作是在本身请的说明书中公开的发明的实现。涉及实施例 1 至 6 所示的再现设备的传送(当涉及成本时的零售;当没有涉及成本时的赠送)、出租、或者进口的动作同样是本发明的实现。涉及借助于橱窗显示、目录请求、小册子发布等等到一般用户的传送、出租等等的动作同样是这些再现设备的实现。

[0305] (C) 因为图 20-22、图 28、图 30、图 33 和图 38 的流程图所示的计算机程序的信息处理是明确地利用硬件资源实现的,所以显示流程图中的处理过程的计算机程序以他们自己的权限形成了发明。尽管所有的实施例以在再现设备并入的形式显示了涉及关于本发明的计算机程序的实现的实施例,但是实施例 1 至 6 所示的计算机程序可以合理地权限被实现,而与再现设备分开。合理地计算机程序的实现包括以下动作:(1) 程序的生产,(2) 程序的转移,免费或者付费,(3) 程序的出租,(4) 程序的进口,(5) 经由双向电子通信电路公开地提供程序,以及(6) 借助于橱窗显示、目录请求、小册子发布等等接近一般用户的转移、出租等等。

[0306] (D) 考虑与在图 20-22、图 28、图 30、图 33 以及图 38 的流程图中的时间序列中执

行的步骤有关的“时间”的元素是用于指定本发明的要求项目。如果是这种情况,则然后,由流程图表示的处理过程可以被理解为公开再现方法的使用配置。通过执行在时间序列中的步骤的处理来执行流程图中的处理以便完成本发明的原始的目的并且产生动作不必说是关于本发明的记录方法的实现。

[0307] (E) 采用实施例 5,用于显示章节目录的菜单(章节菜单)和用于控制这菜单动作的电影对象可以被记录在 BD-ROM 上,并且从顶部菜单转移启动。同时,这些菜单可以通过按下遥控器上的章节键而被调用。

[0308] (F) 当记录在 BD-ROM 上时,扩展的头部优选地被附加到构造 AVClip 的 TS 包。被称为 TP_extra_header 的扩展头部包括“Arrival_Time_Stamp”和“copy_permission_indicator”,并且具有 4 字节的数据长度。附上 TP_extra_header 的 TS 包(以下,缩写成“附上 EX 的 TS 包”)被安排在 32 个包的组中,并且被写入三个扇区。包括 32 个附上 EX 的 TS 包的每个组是 6,144 字节长($= 32 \times 192$),并且匹配 6,144 字节长度的三个扇区($= 2048 \times 3$)。包含在三个扇区中的 32 个附上 EX 的 TS 包的组被称为是“对准的单元”。

[0309] 当用于经由 IEEE1394 连接的本地网络时,再现设备 200 在如下所述的传递过程中传送对准单元。也就是说,在发送者侧的设备从包括在对准单元中的 32 个附上 EX 的 TS 包的每一个除去 TP_extra_header,并且在基于 DTCP 标准编码 TS 包主体之后输出 TS 包。当输出 TS 包时,同步包被插入在所有相邻的 TS 包之间。同步包的定位基于每个 TP_extra_header 中的 Arrival_Time_Stamp 所示的时间。再现设备 200 输出 DTCP_Descriptor,之后输出 TS 包。DTCP_Descriptor 显示在每个 TP_extra_header 中的复制容许性设置。在这里,如果 DTCP_Descriptor 被描述以便显示“复制禁止”,则当用于经由 IEEE1394 连接的本地网络时,TS 包不会被记录在其它设备上。

[0310] (G) 尽管记录在实施例中的记录介质上的数字流是 AVClip,但是数字流可以是遵守 DVD 视频标准或者 DVD 视频记录标准的 VOB(视频对象)。VOB 是遵从通过多路复用视频和音频流获得的 ISO/IEC13818-1 的程序流。同时,在 AVClip 中的视频流可以是 MPEG-4 格式、WMV 格式等等。此外,音频流可以是线性 PCM 格式,杜比 AC3 格式、MP3 格式、MPEG-AAC 格式、Dts 或者 WMA(窗口介质音频)。

[0311] (H) 在再现设备的结构中,仅仅当前动态脚本被存储在动态脚本存储器 15,并且仅仅当前流管理信息和当前 PL 信息被存储在静态脚本存储器 11。然而,多个脚本、流管理信息和 PL 信息可以被预先存储,如同超高速缓冲存储器的情况一样。由此,直到从 BD-ROM 读取这数据为止的时滞可以被缩短。同样,尽管备份存储器 14 以堆栈形式保存寄存器的存储值,但是当考虑与存储器大小的关系时,安排用于保存在一个等级上的存储值是现实的。

[0312] (I) 实施例中的电影作品可以通过编码利用模拟广播而广播的模拟视频信号来获得,或者可以是由通过数字广播而广播的传输流构成的流数据。

[0313] 同样,内容可以通过编码记录在录像磁带上的模拟/数字视频信号获得。此外,内容可以通过编码直接地从摄像机取得的模拟/数字视频信号获得。替换地,内容可以从分发服务器分发的数字版权作品。

[0314] (J) Java 模块 17 可以是为了传送卫星广播安装在设备中的 Java 平台。如果 Java 模块 17 是该 Java 平台,则关于本发明的再现设备共享作为 MHP-use STB 的处理。

[0315] 此外,Java 模块 17 可以是为了执行移动电话处理控制而安装在设备中的 Java 平

台。如果 Java 模块 17 是该 Java 平台,则关于本发明的再现设备将共享作为移动电话的处理。

[0316] 同时, BROWSER 模块 18 可以是计算机安装的浏览器软件, 诸如微软公司的 Internet Explorer 等等。

[0317] (K) 在图所示的分层模型中, 浏览器模式和电影模式可以在 Java 模式上被处理。特别是因为基于动态脚本的控制程序的执行的再现设备的轻负荷, 电影模式中的动态脚本的解释等等, 即使当在 Java 模式上执行电影模式时, 没有问题出现。同样, 当开发再现设备和电影作品时, 操作保证可以以单个方式处理。

[0318] 此外, Java 模式处理可以仅仅以 Java 模式执行, 没有提供三种方式。如实施例 2 所示, 因为与 PL 再现同步的再现控制是可能的, 甚至在 Java 模式中, 所以提供电影模式的必要性被排除。此外, 动态脚本中的控制可以仅仅是电影模式或者仅仅是浏览器模式。

[0319] 工业实用性

[0320] 关于本发明的记录介质和再现设备能够给予电影作品交互控制, 由此可以使得具有高附加价值的电影作品供应给市场, 并且刺激电影、消费品市场等等。同样地, 关于本发明的记录介质和再现设备非常适用于电影和消费品工业。

动态脚本

```
function{PlayPL(PL#1,PI#1)
  if(SPRM(0)=="Japanese"){
    PlayPL(PL#4,PI#1);
  }else{
    PlayPL(PL#2,PI#1);
    PlayPL(PL#3,PI#1);
  }
}
```

图1A

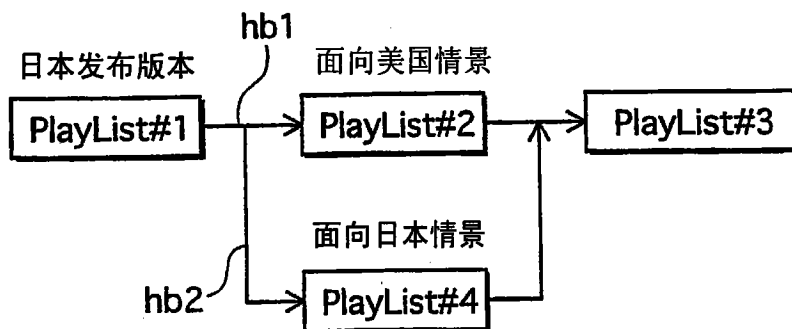


图1B

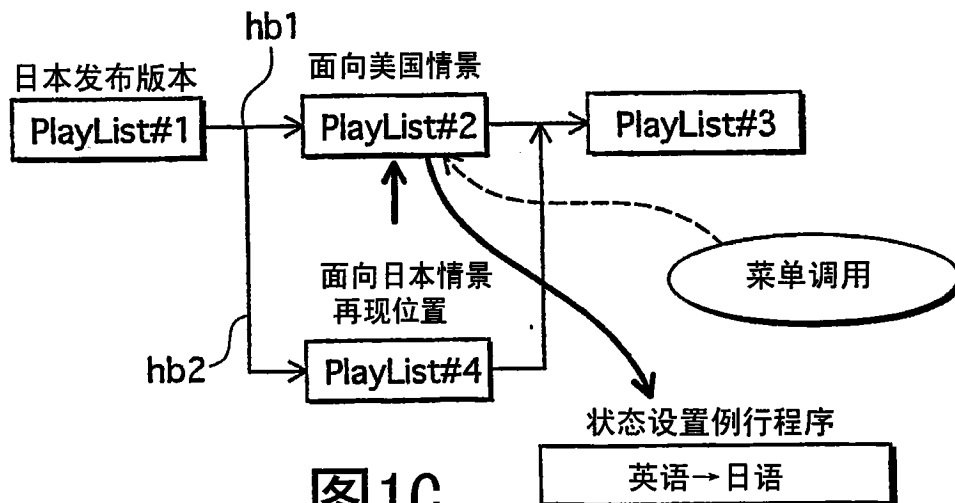
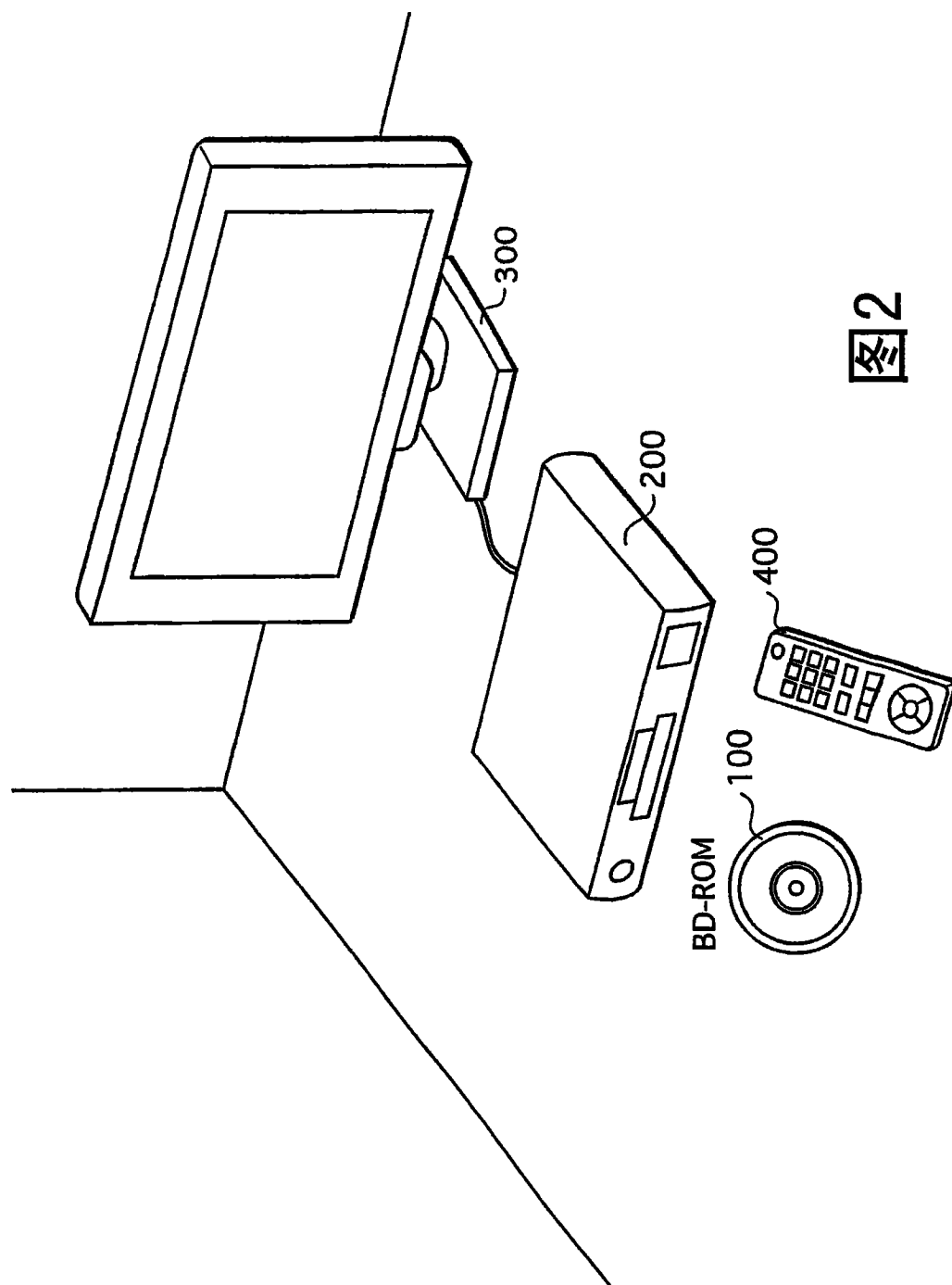


图1C



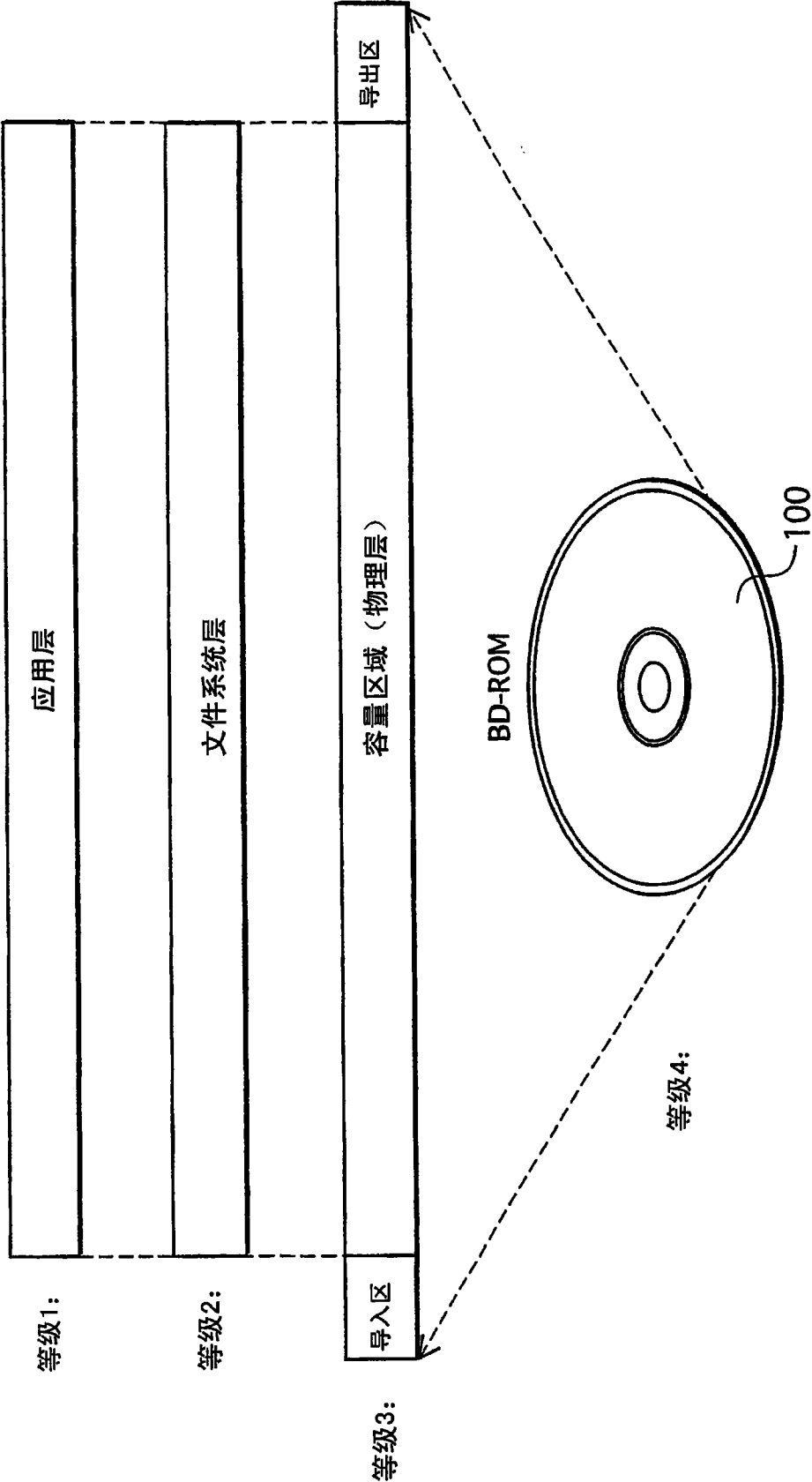


图3

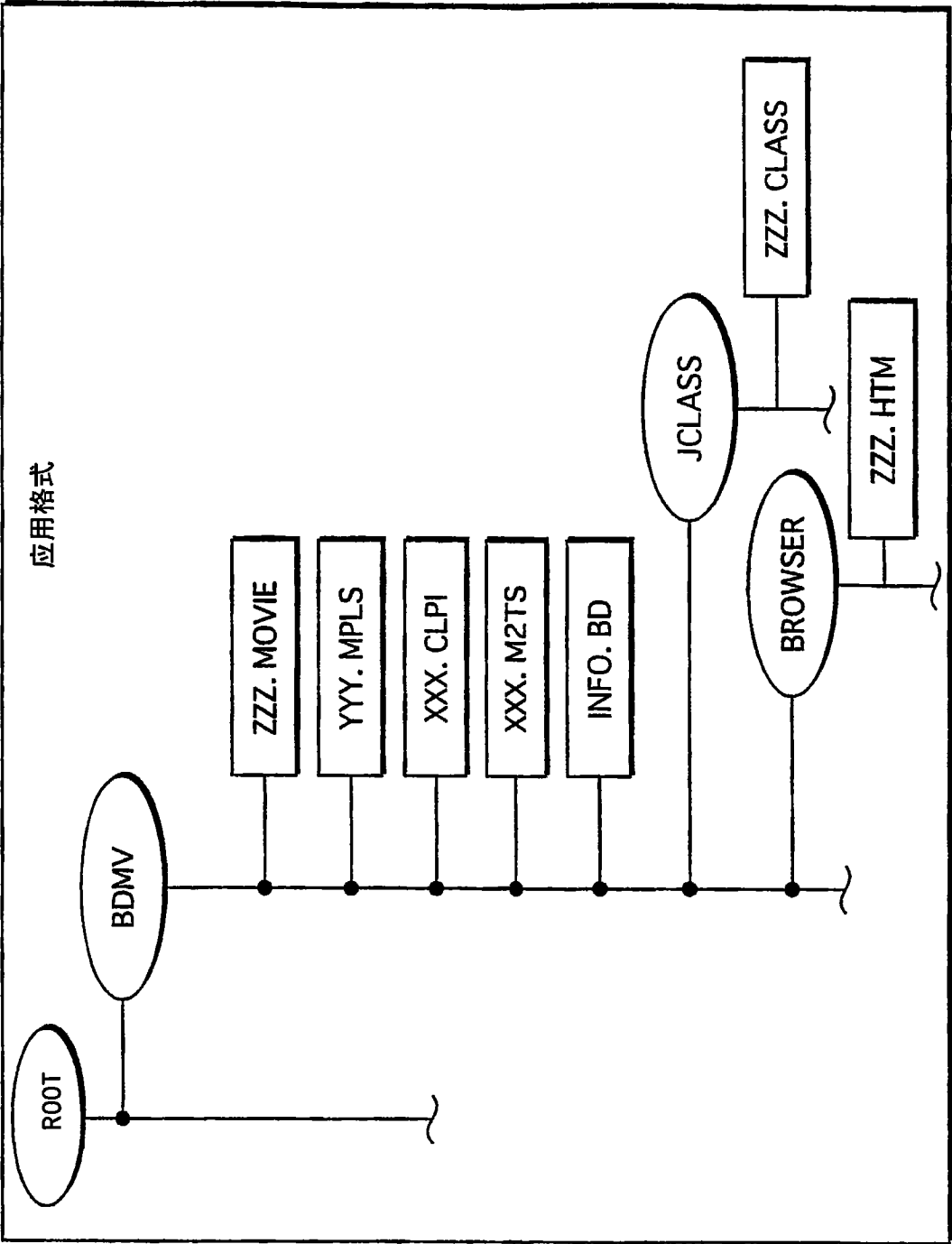


图4

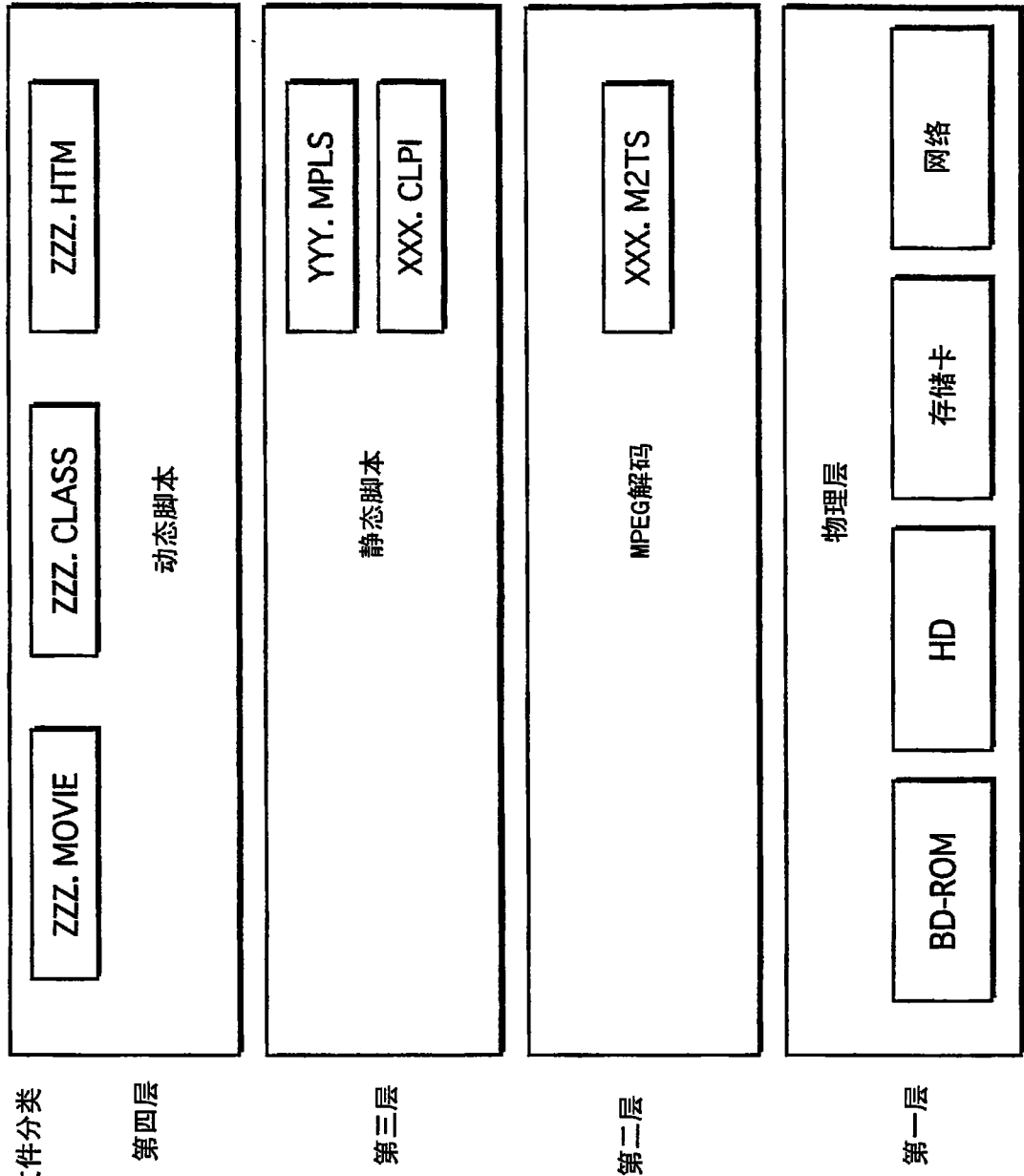


图5

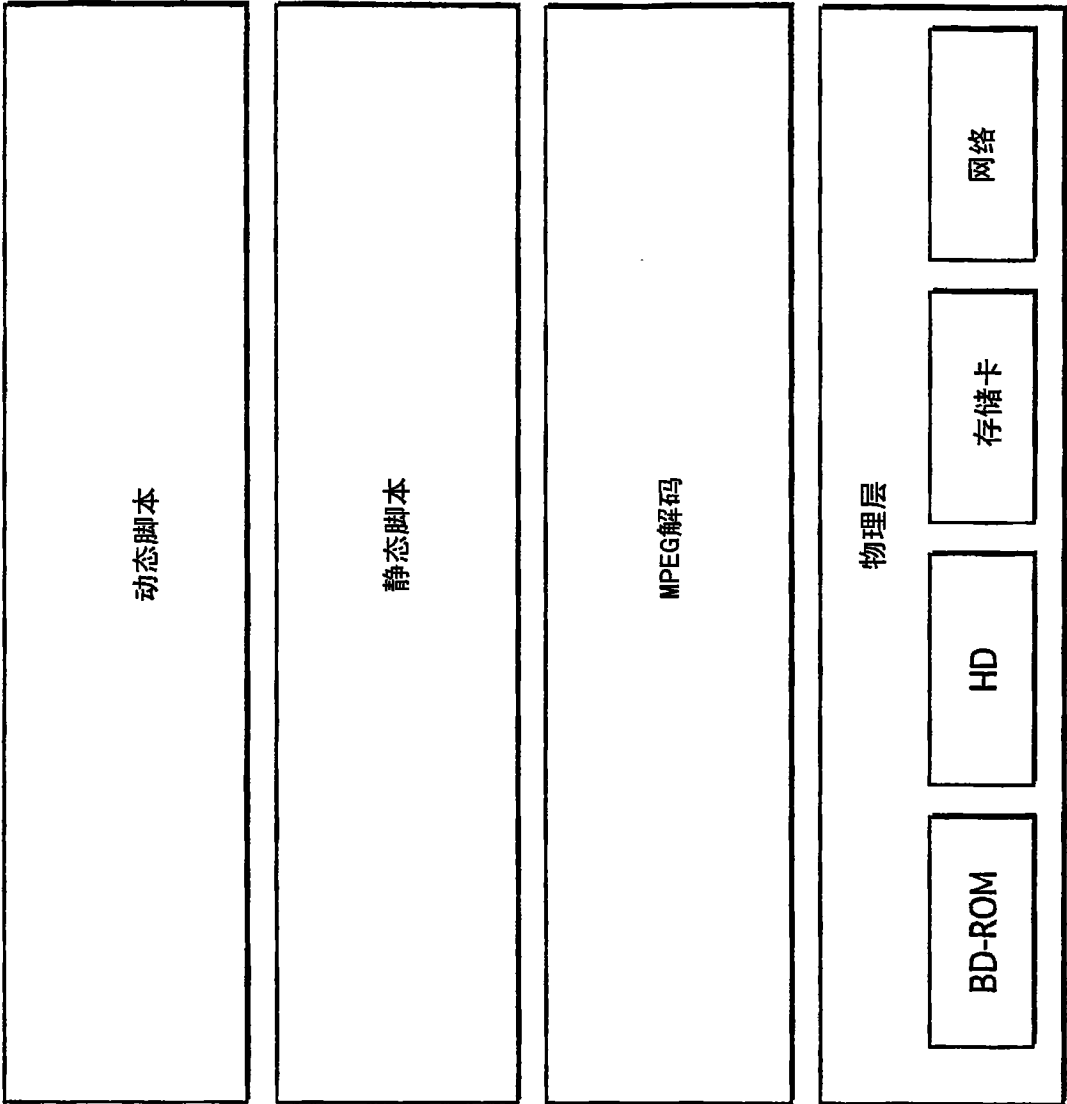


图6

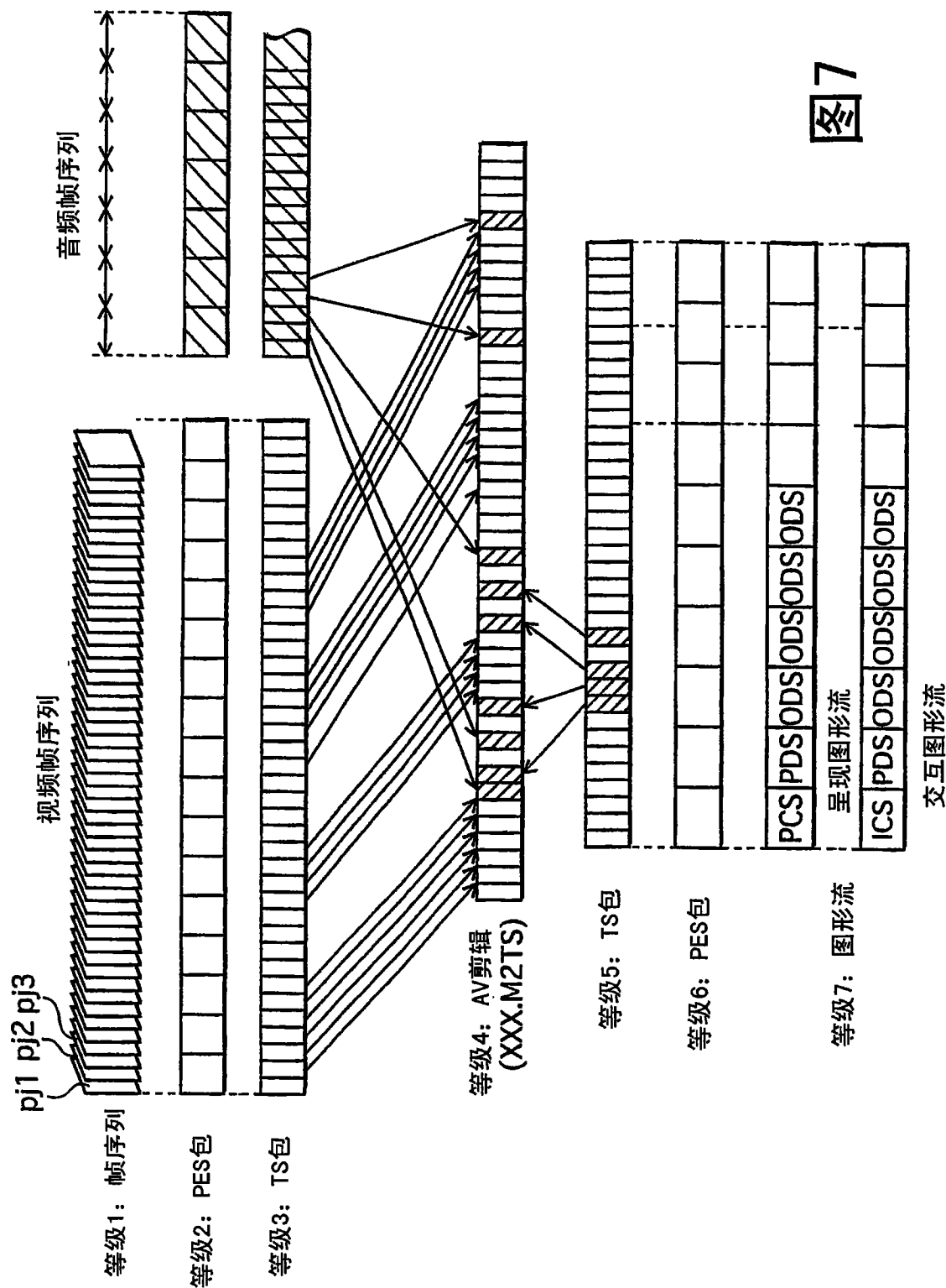
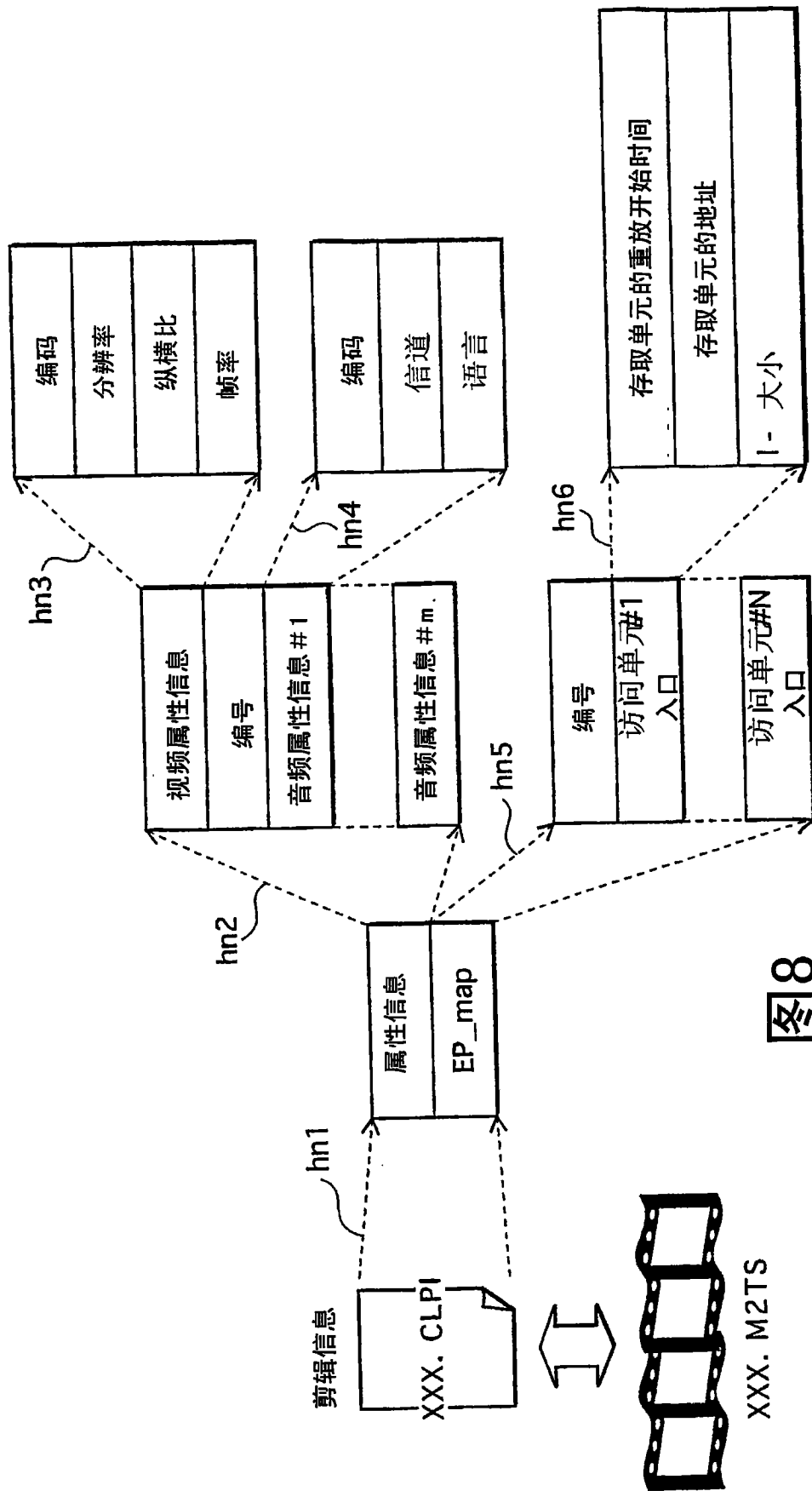


图7



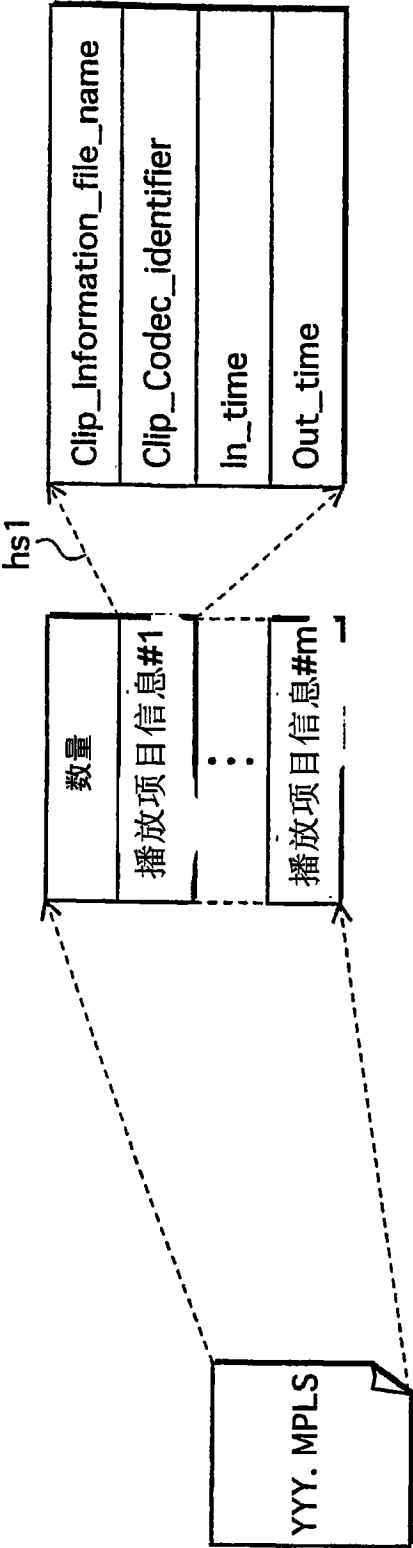
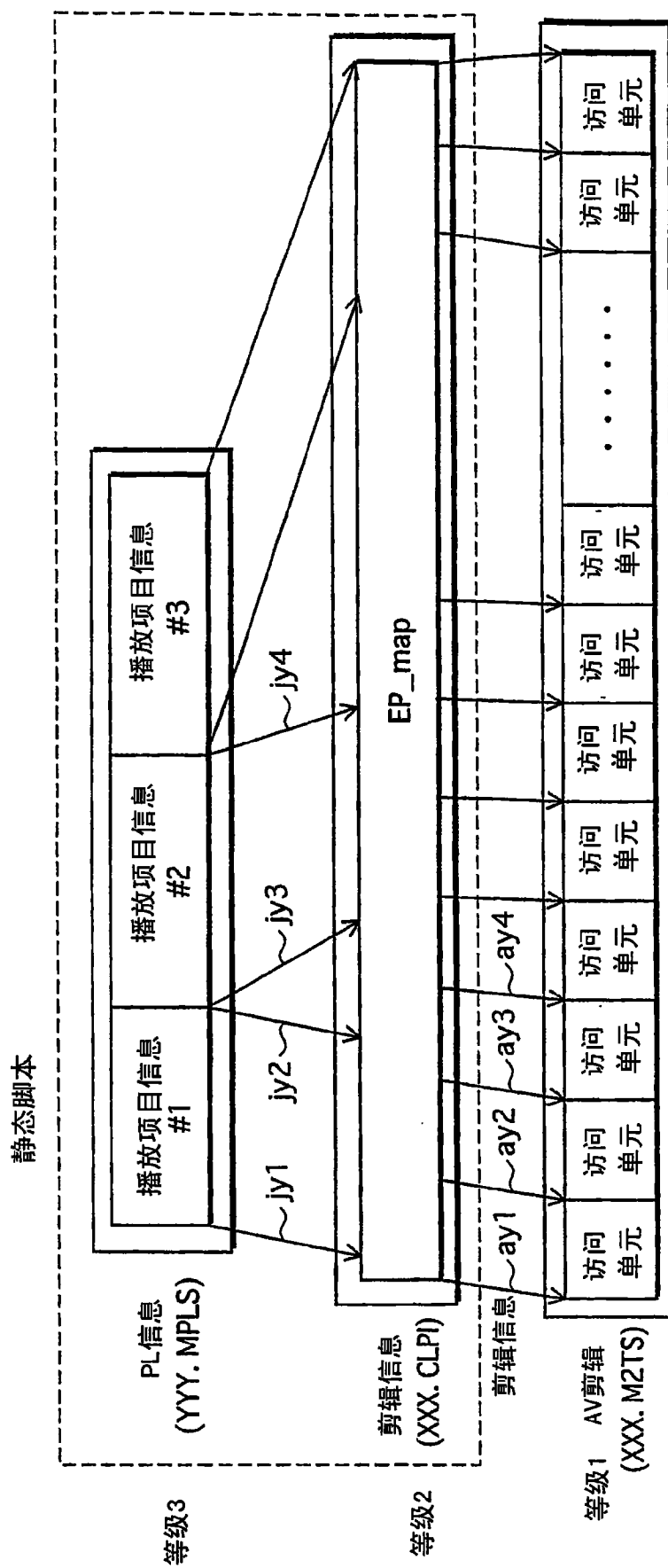


图9



10

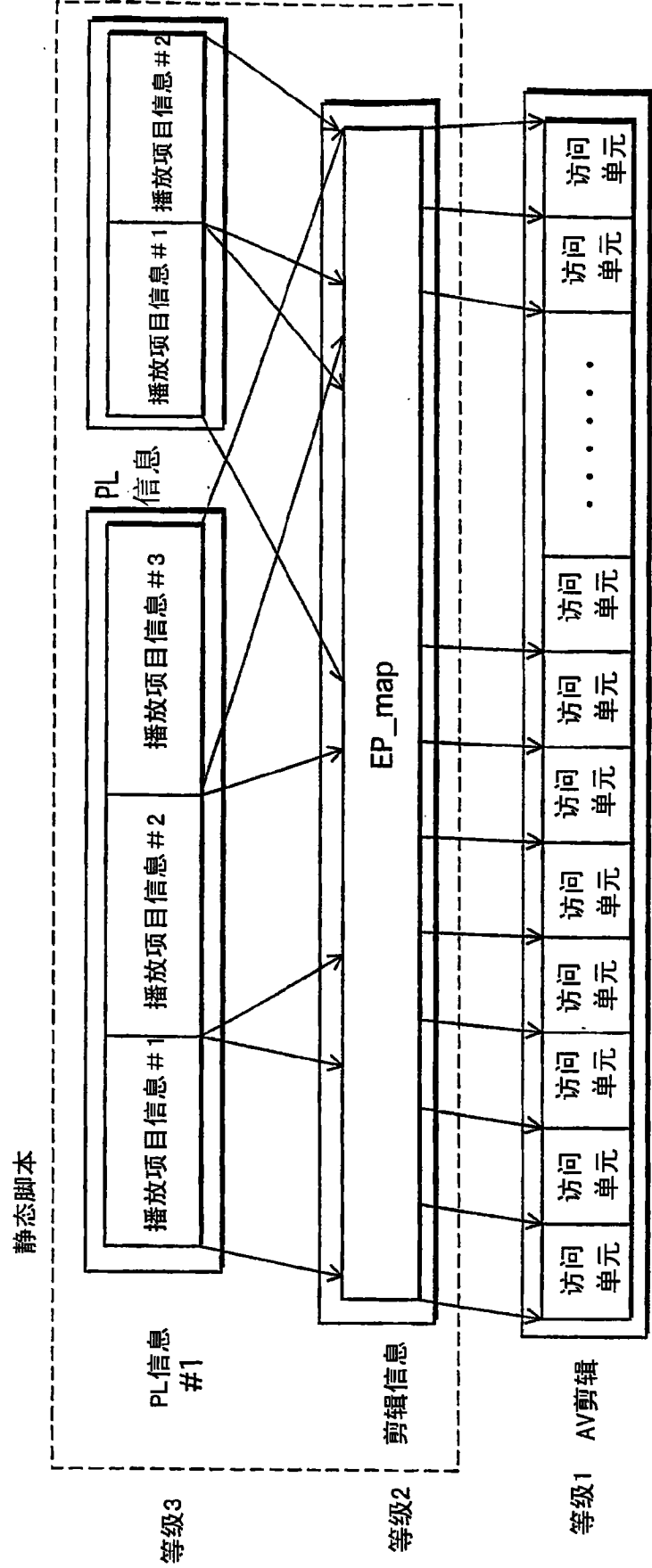


图11

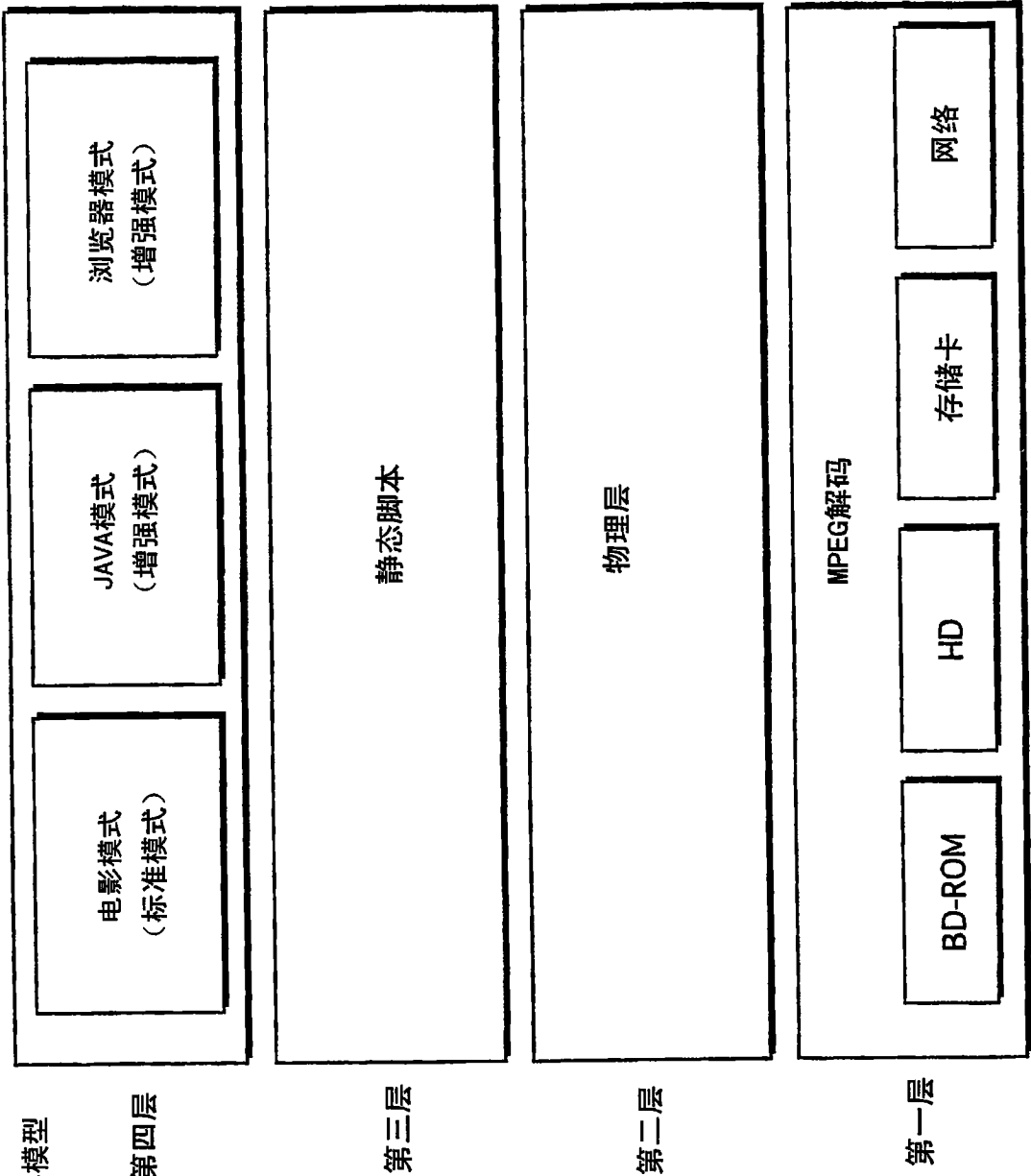


图12

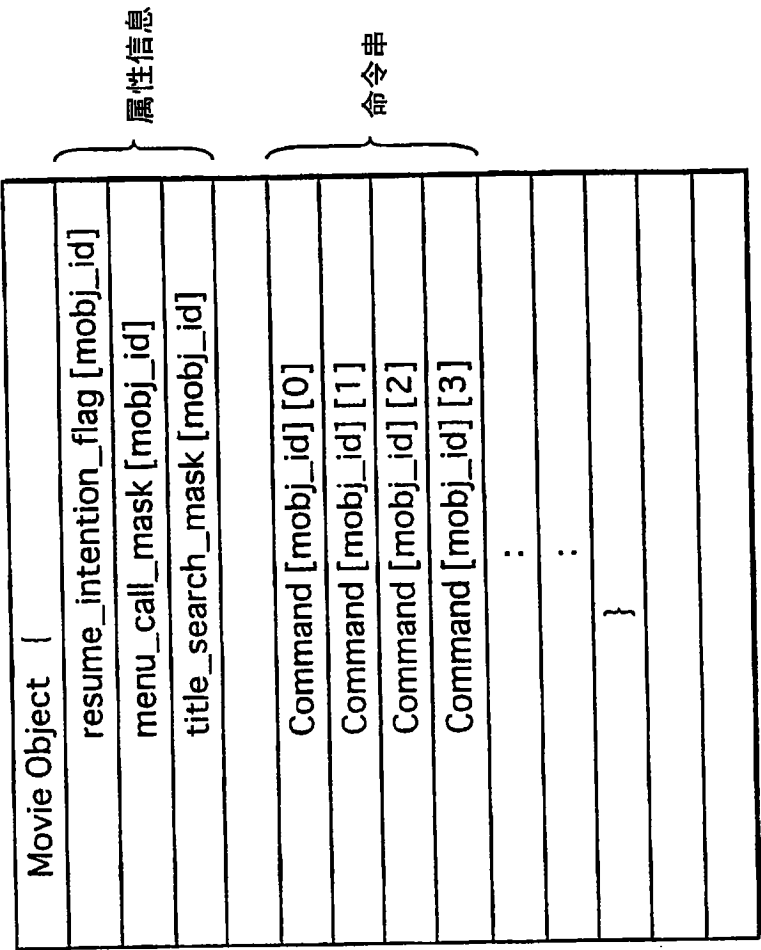


图13

```
MovieObject(1){  
  resume_intension_flag=0  
  menu_call_mask=0  
  Title_search_mask=0  
  function{PlayPL(PL#1,PI#1)  
    if(SPRM(0)=="Japanese"){  
      PlayPL(PL#4,PI#1);  
    }else{  
      PlayPL(PL#2,PI#1);  
      PlayPL(PL#3,PI#1);  
    }  
  }  
}
```

图14A

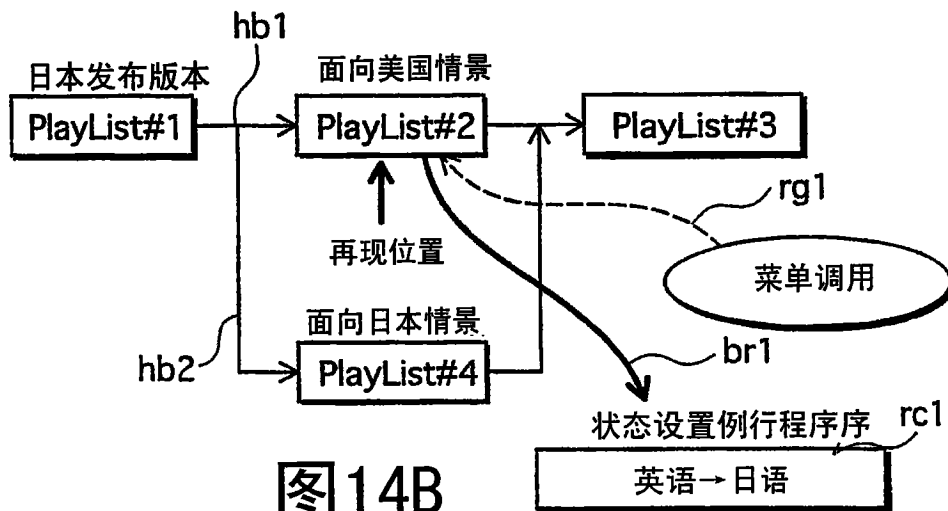


图14B

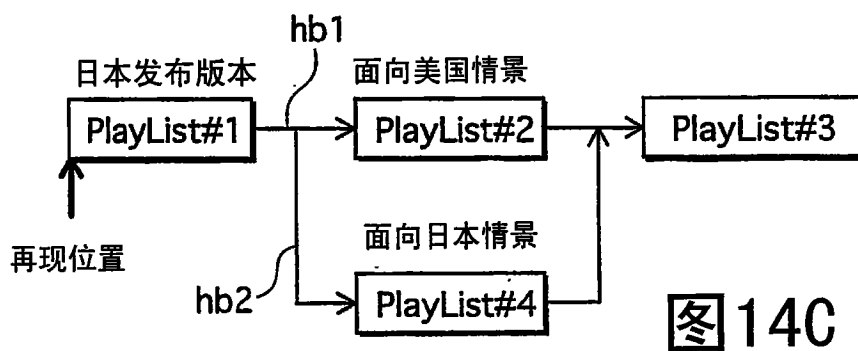


图14C

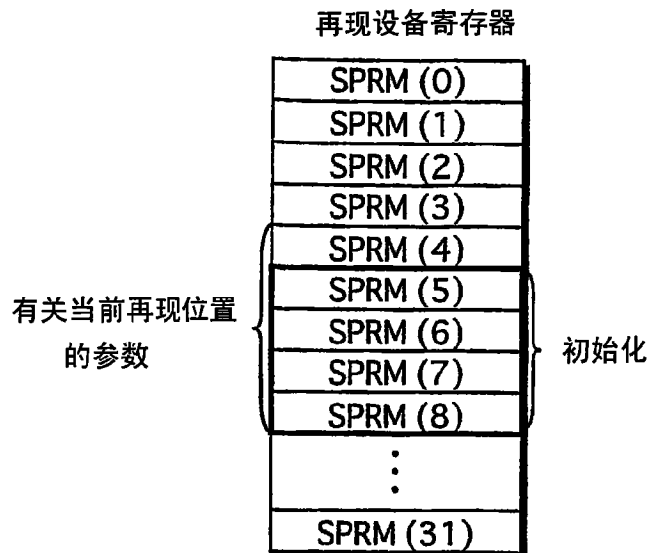


图15A

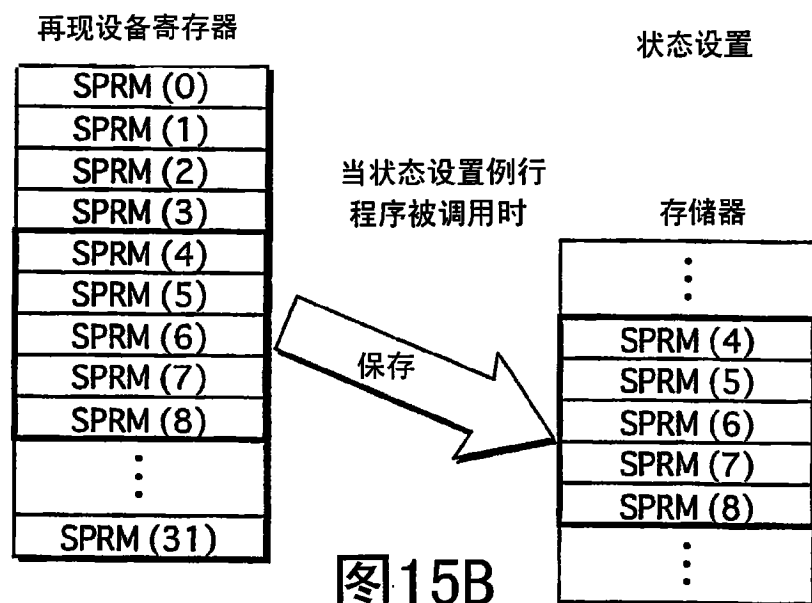


图15B

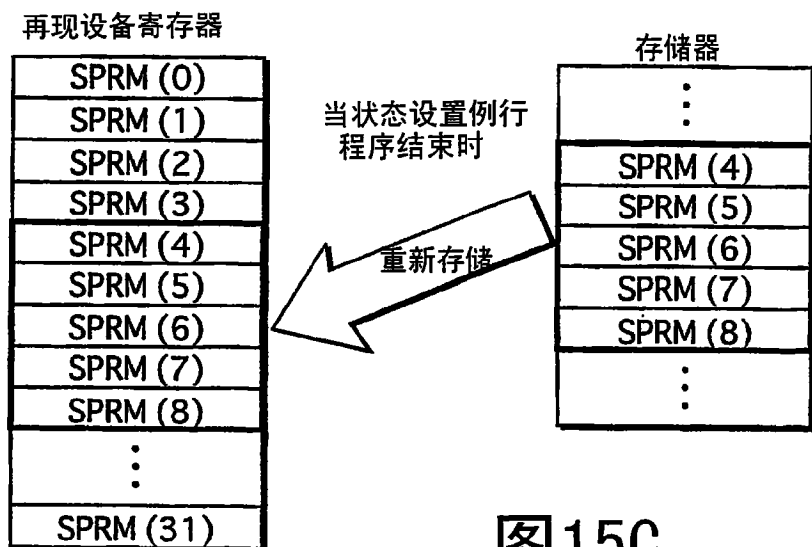


图15C

```

MovieObject(1){
  resume_intension_flag=1
  menu_call_mask=0
  Title_search_mask=0
  function{PlayPL(PL#1,PI#1)
    if(GPRM(0)=="Answer1"){
      PlayPL(PL#4,PI#1);
    }else{
      PlayPL(PL#2,PI#1);}
    PlayPL(PL#3,PI#1);
  }
}

```

图16A

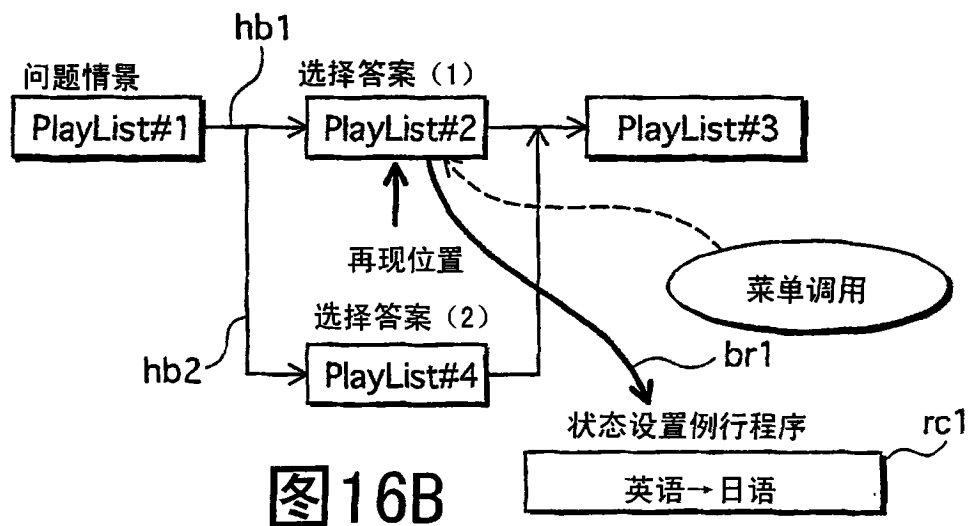


图16B

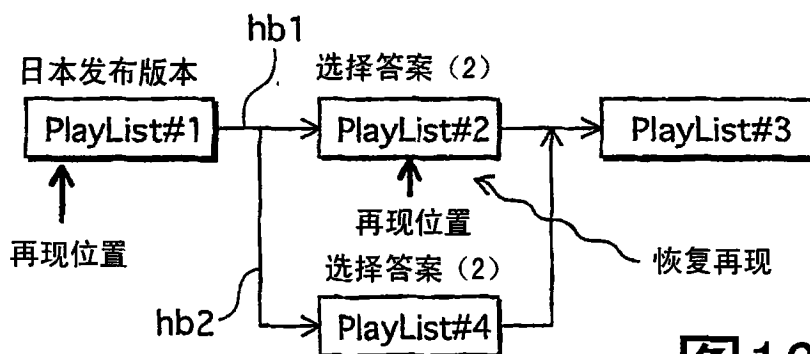


图16C

```

MovieObject(1){
  resume_intension_flag=0
  menu_call_mask=0
  Title_search_mask=0
  function{PlayPL(PL#4,PI#1)
    PlayPL(PL#1,PI#1);
    if(SPRM(13)=="kids"){
      PlayPL(PL#4,PI#1);
    }else{
      PlayPL(PL#2,PI#1);}
    PlayPL(PL#3,PI#1);
  }
}

```

图 17A

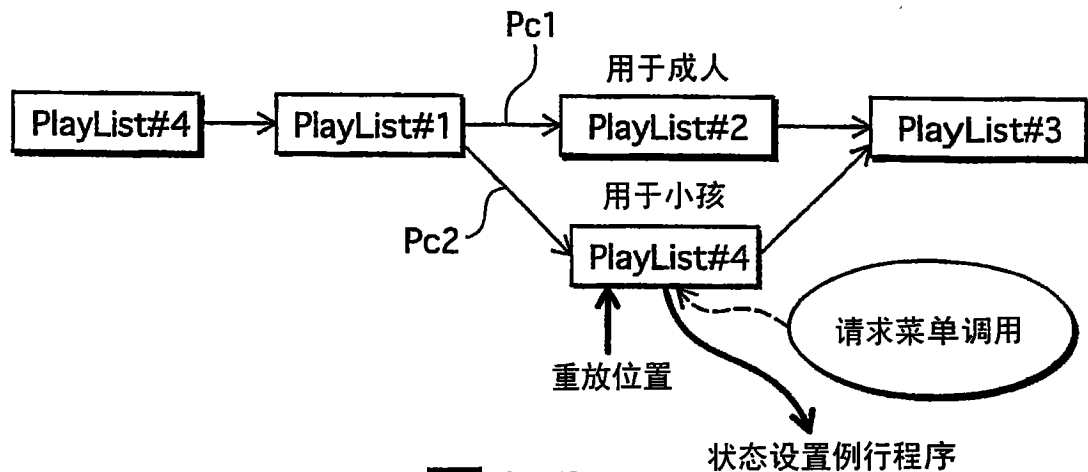


图 17B

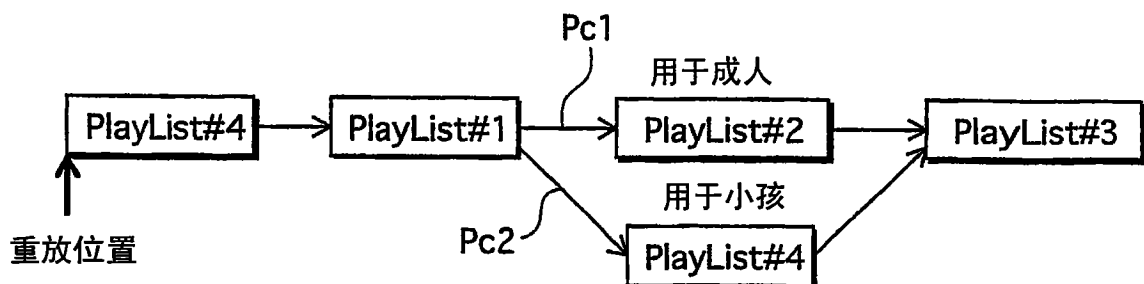


图 17C

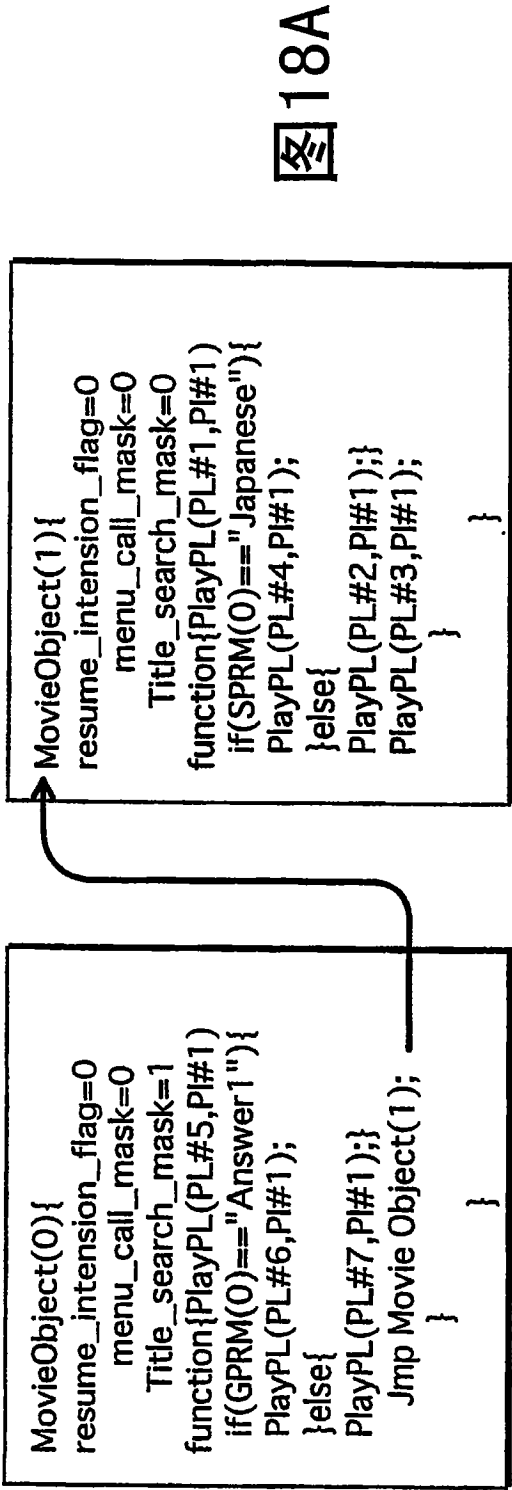


图18A

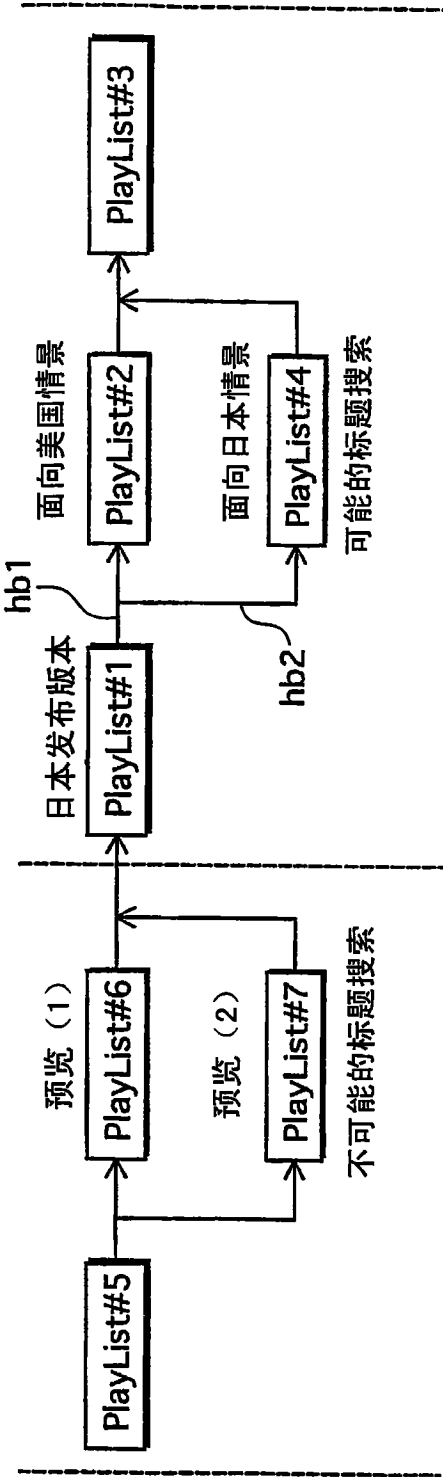


图18B

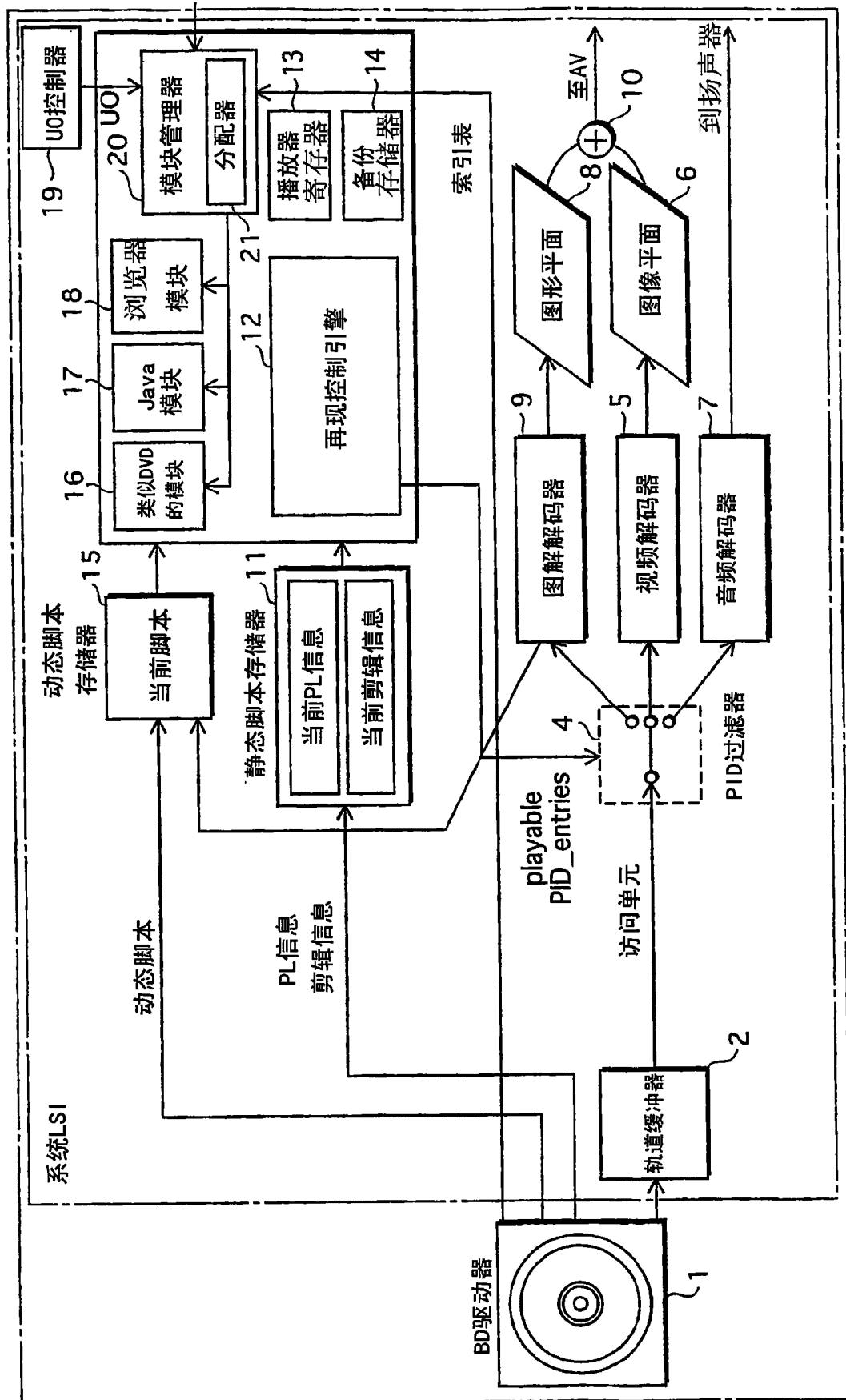


图19

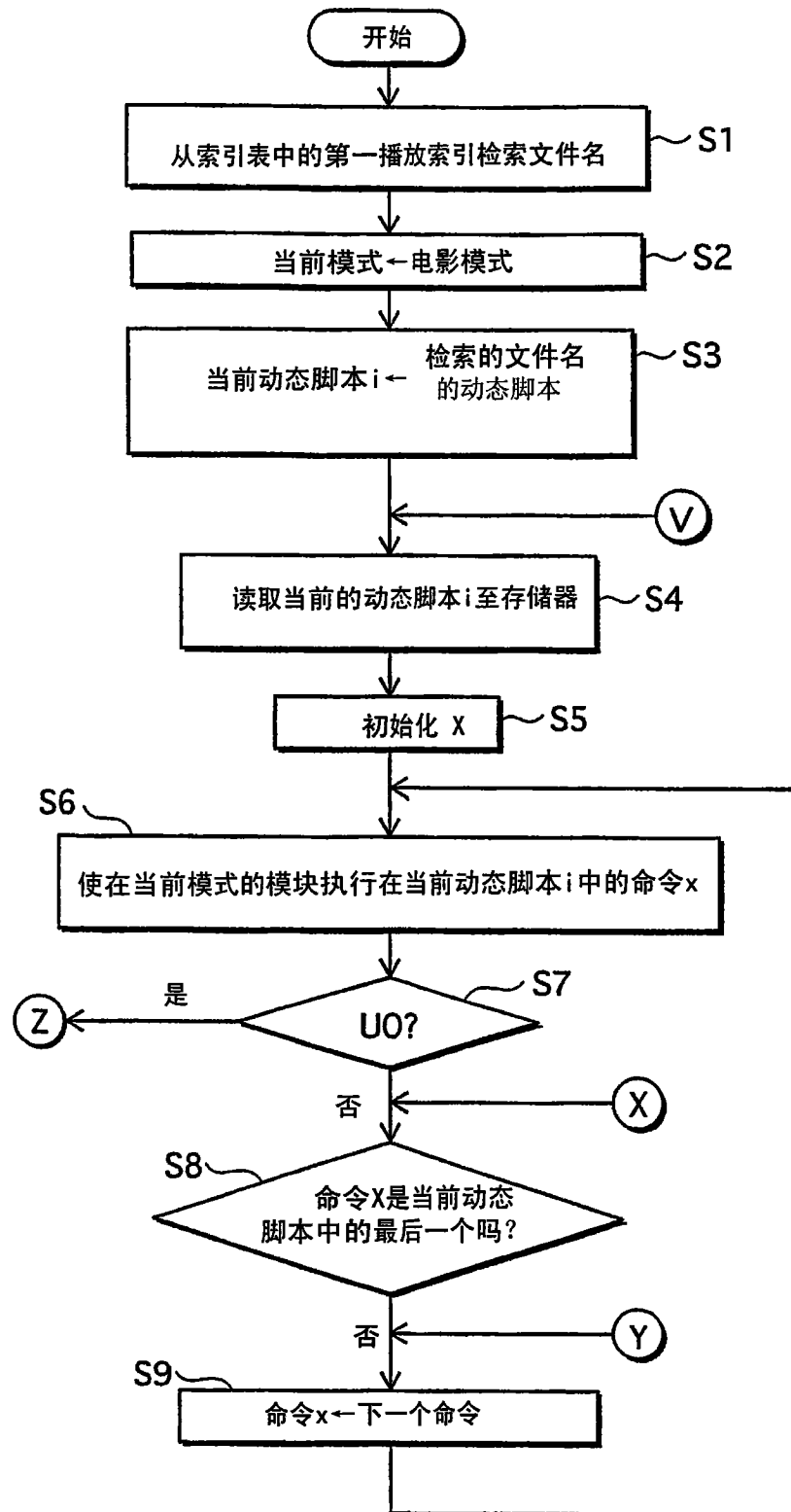
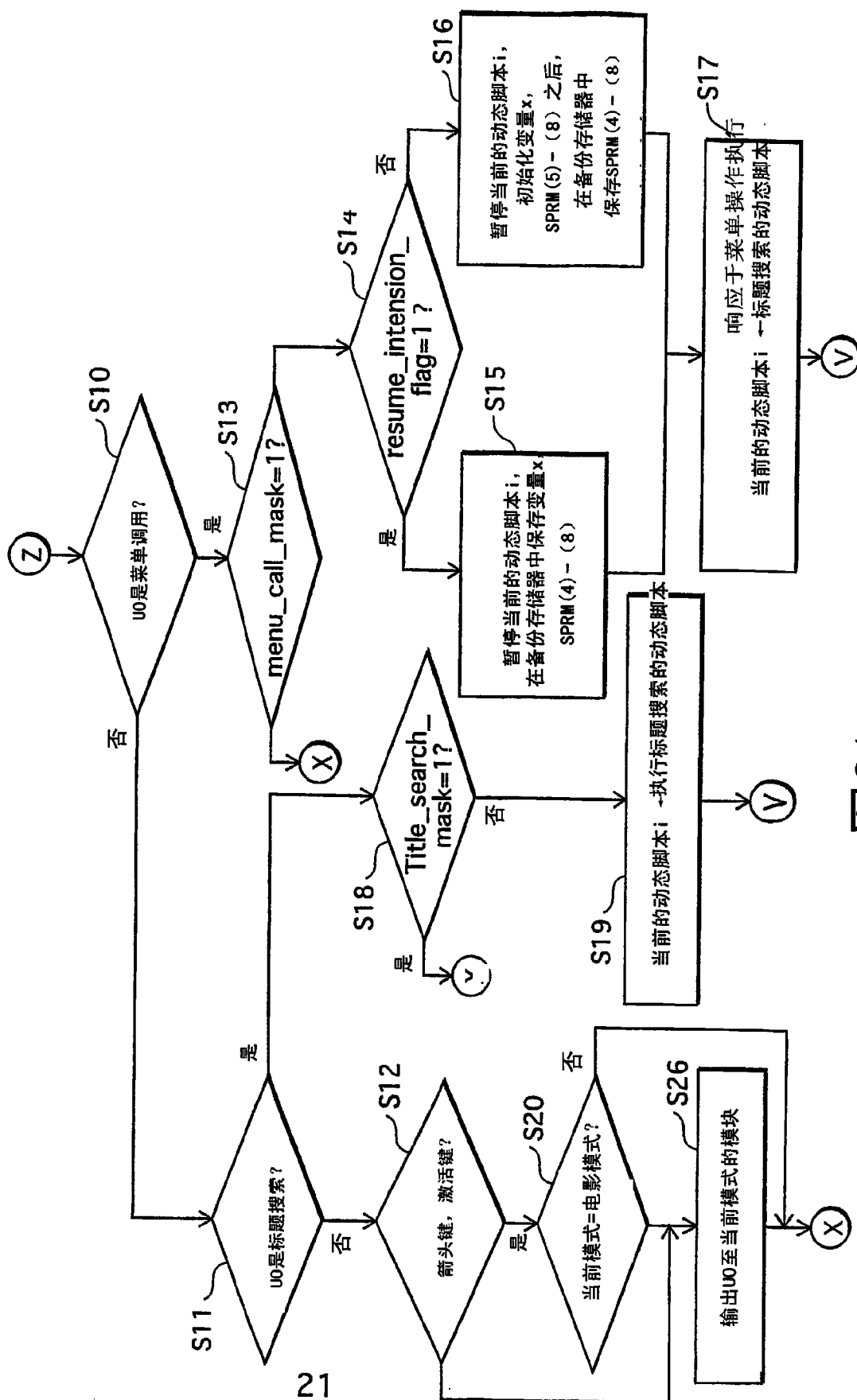
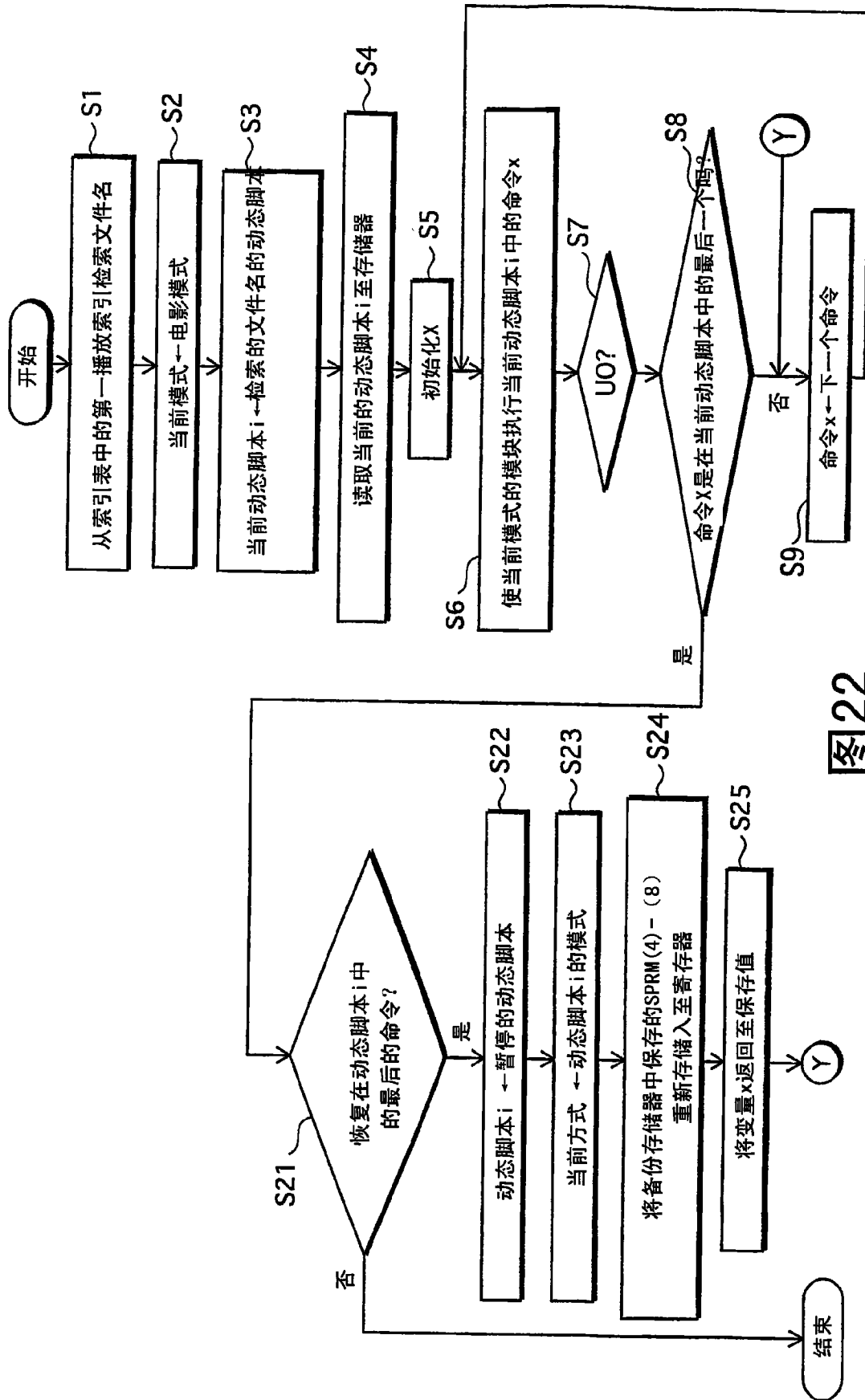


图 20





索引表

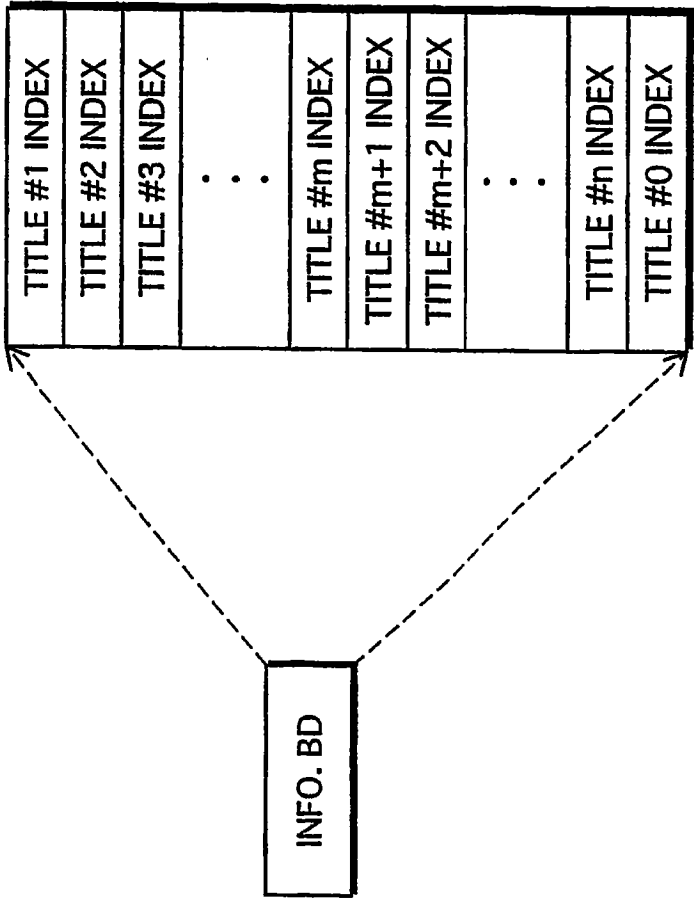


图 23A

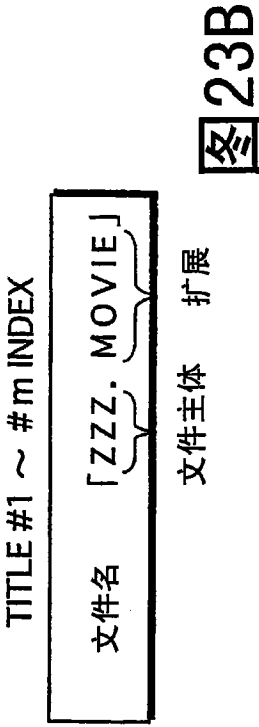


图 23B

TITLE #m+1 ~ #n INDEX

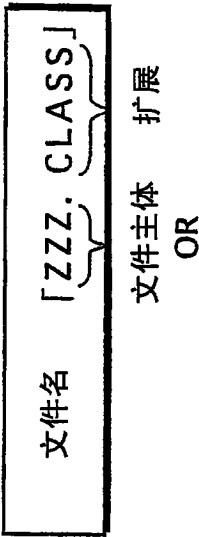


图 23C

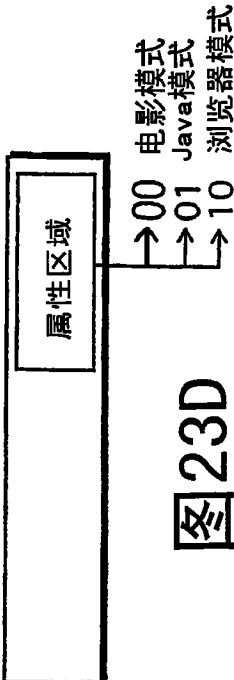
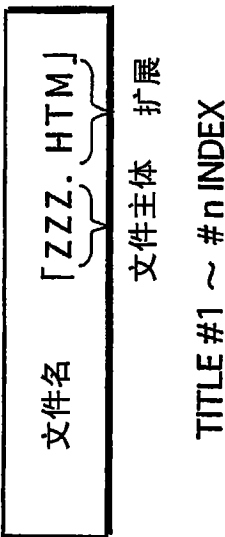
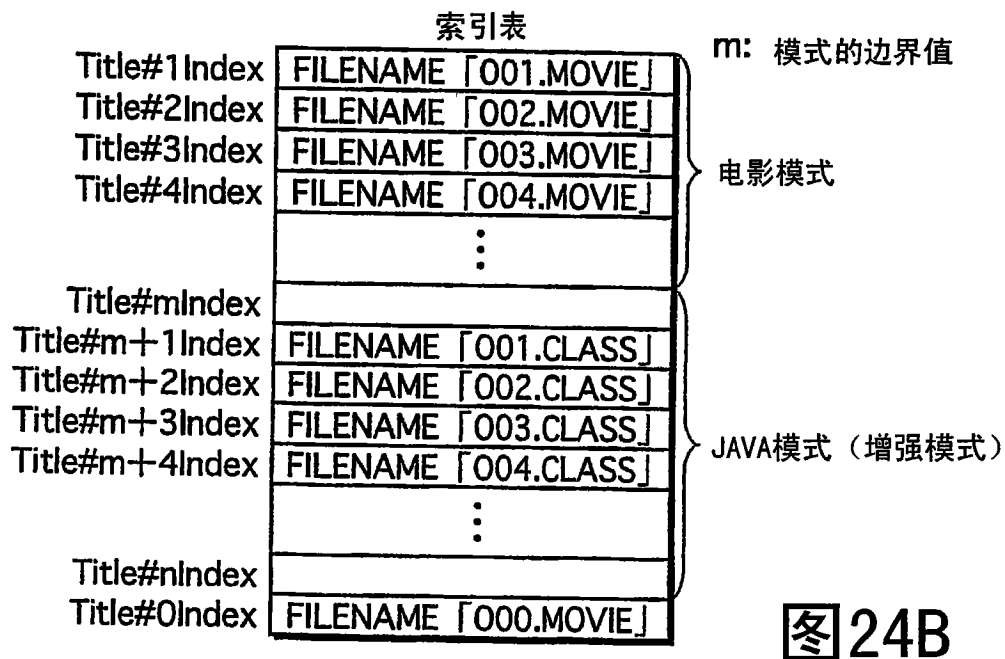
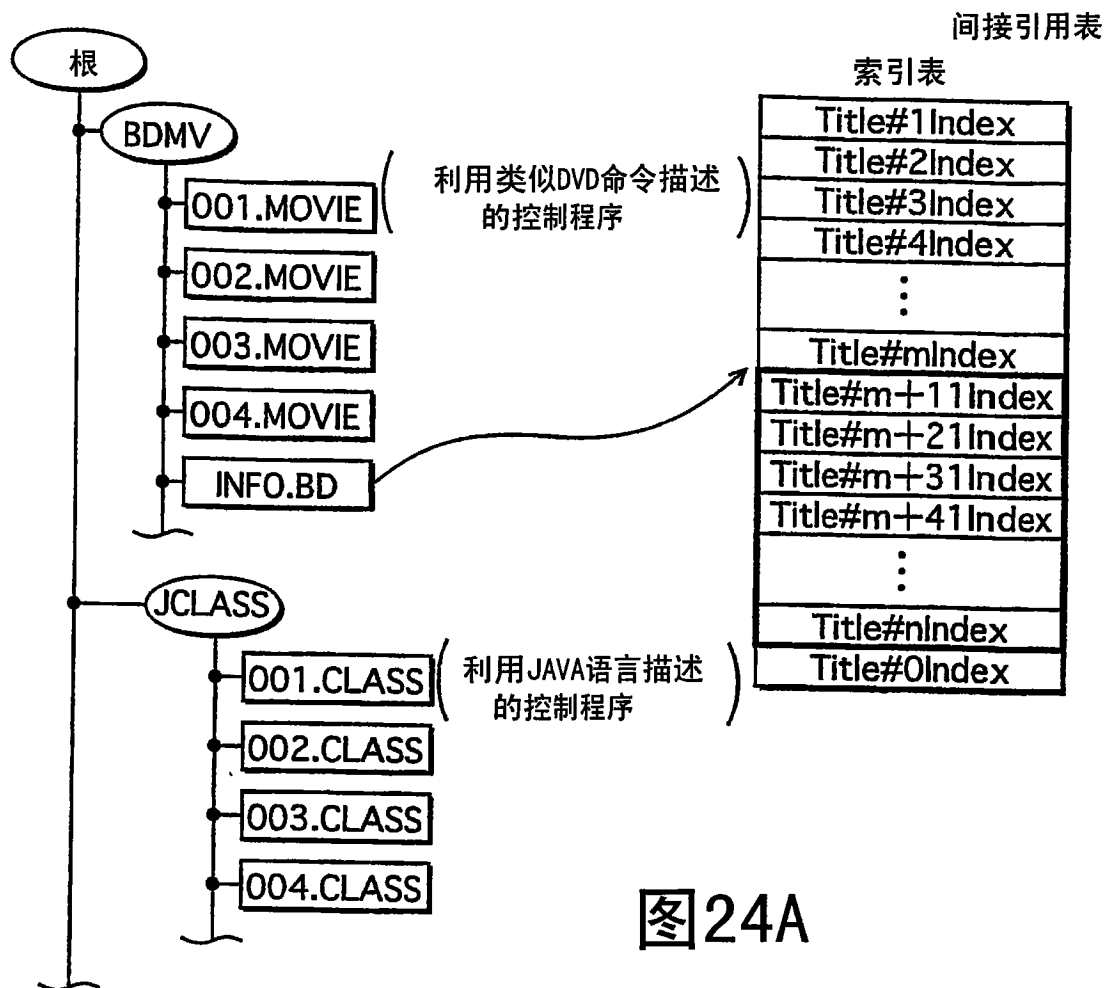


图 23D



电影模式+JAVA模式可能（完全系统）

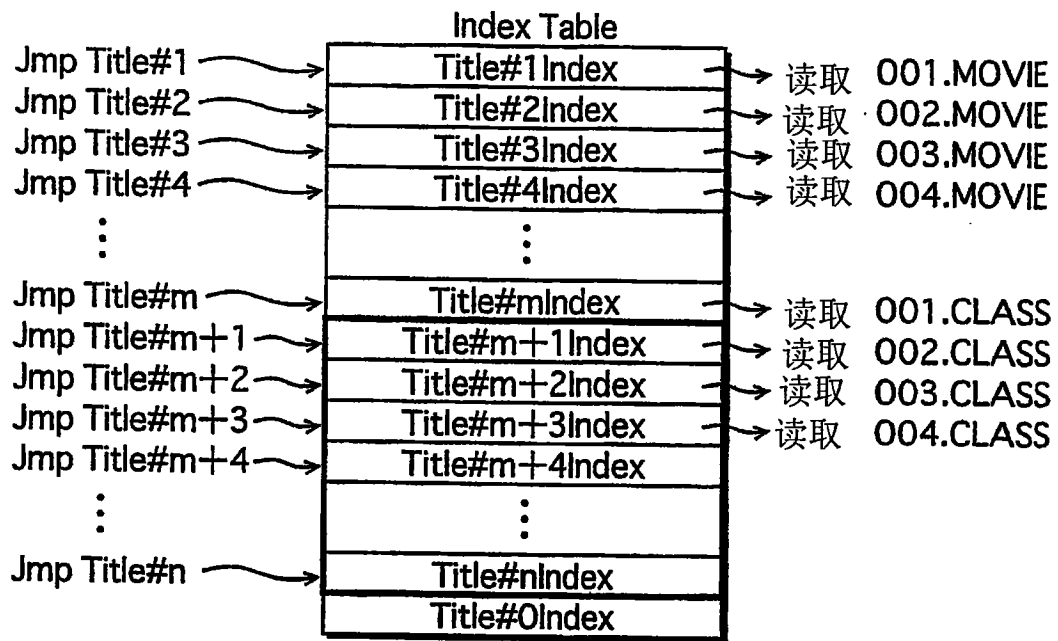


图 25A

仅有电影模式可能（核心系统）

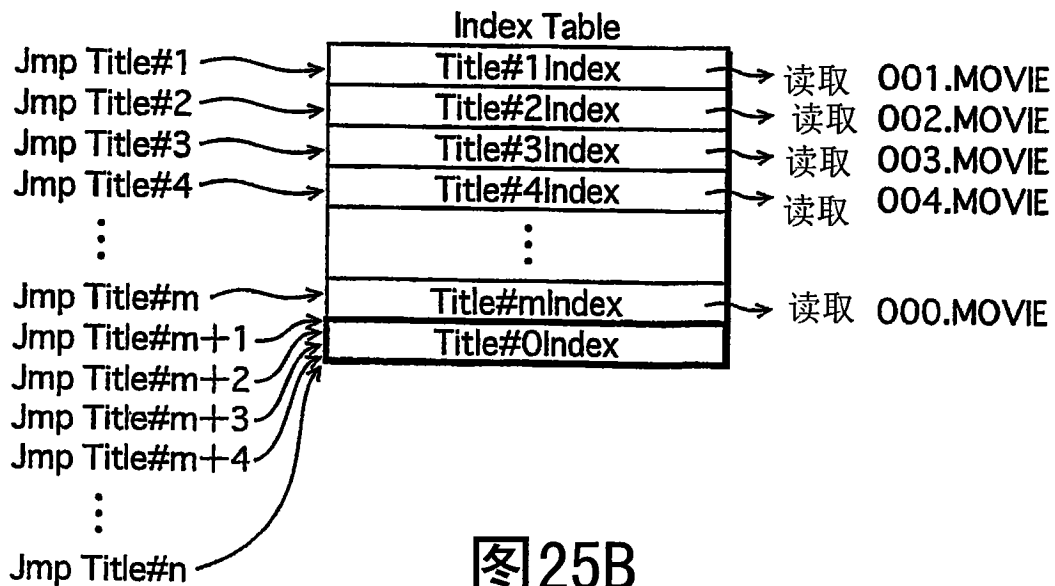


图 25B

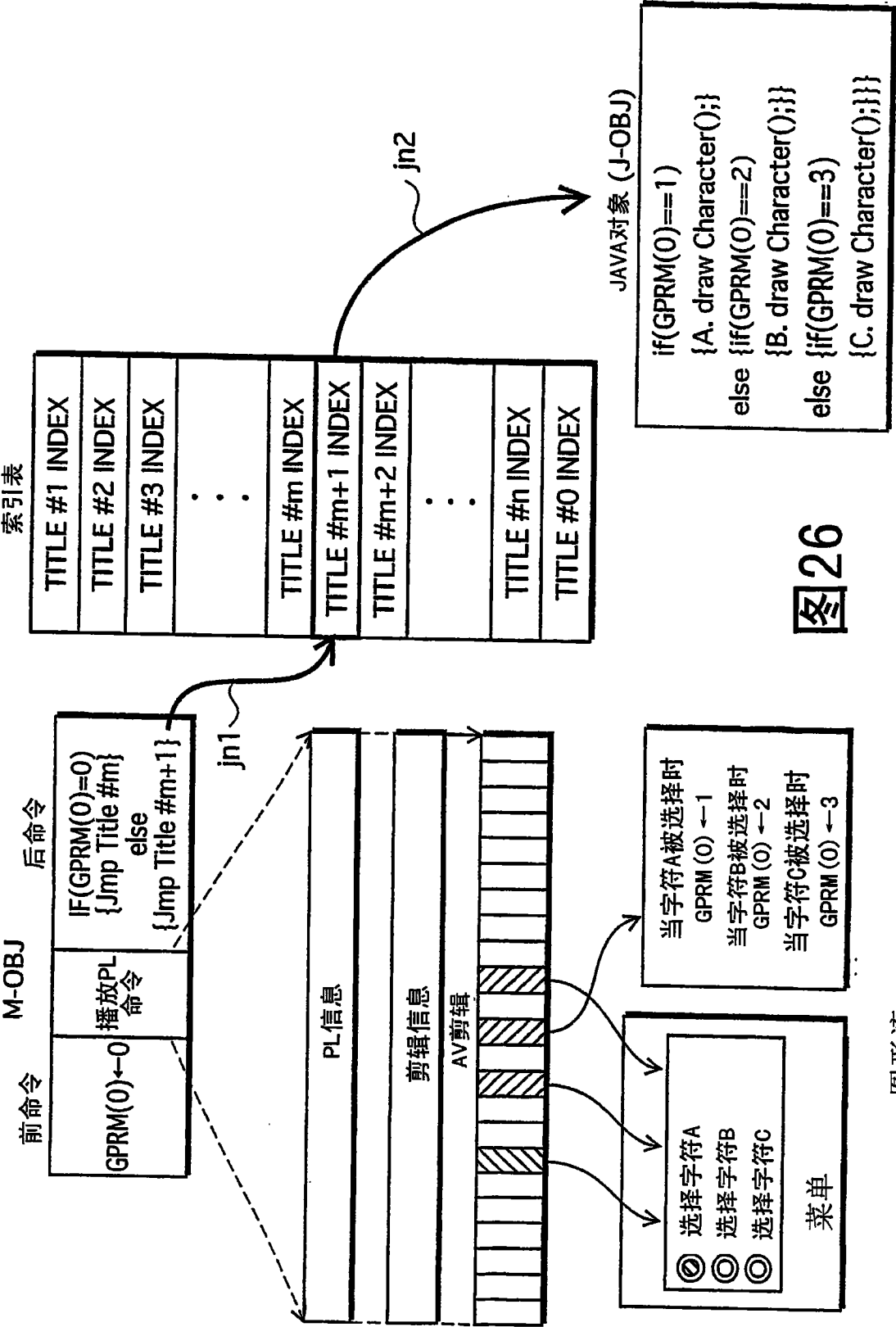


图26

图形流

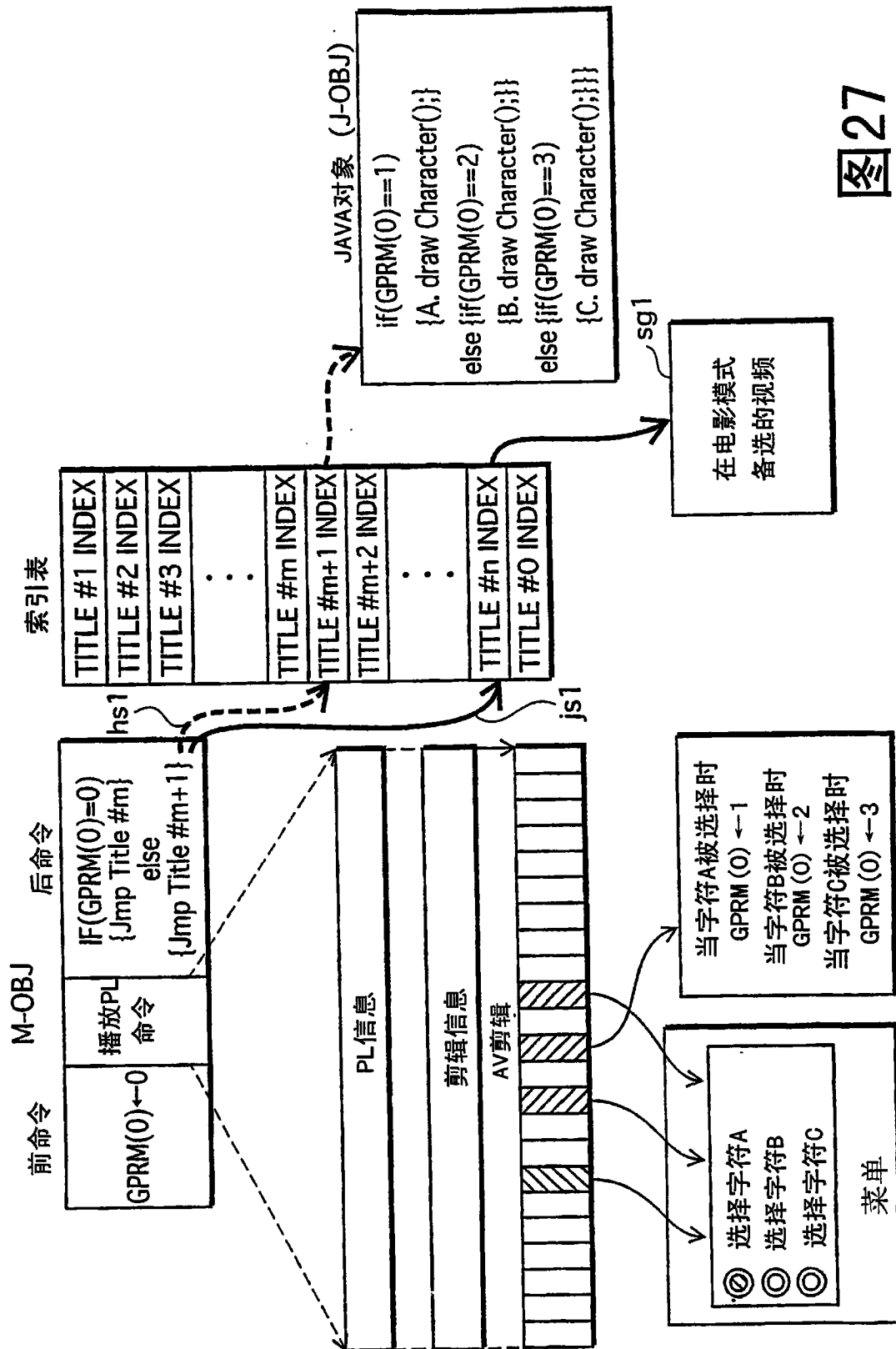


图27

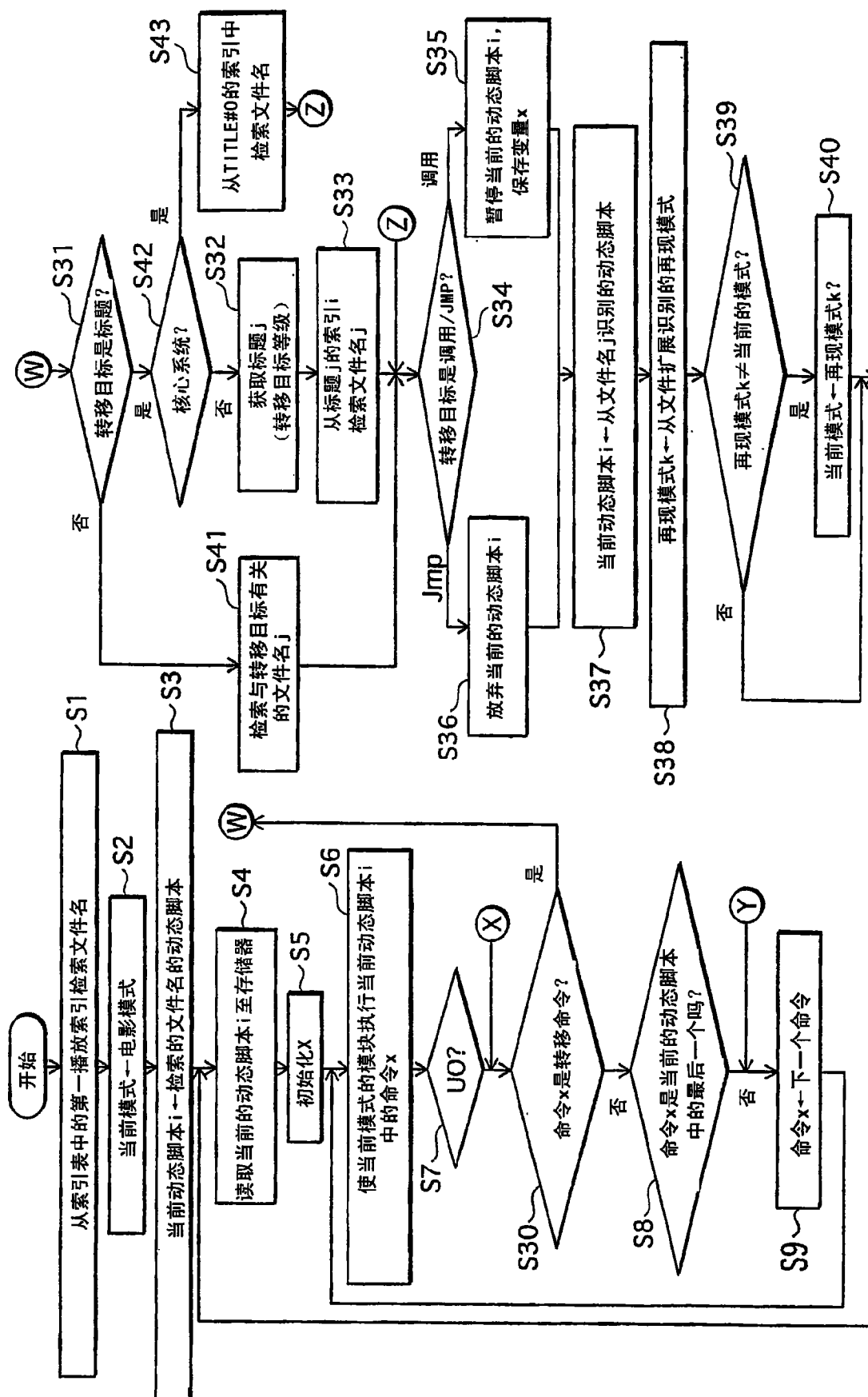
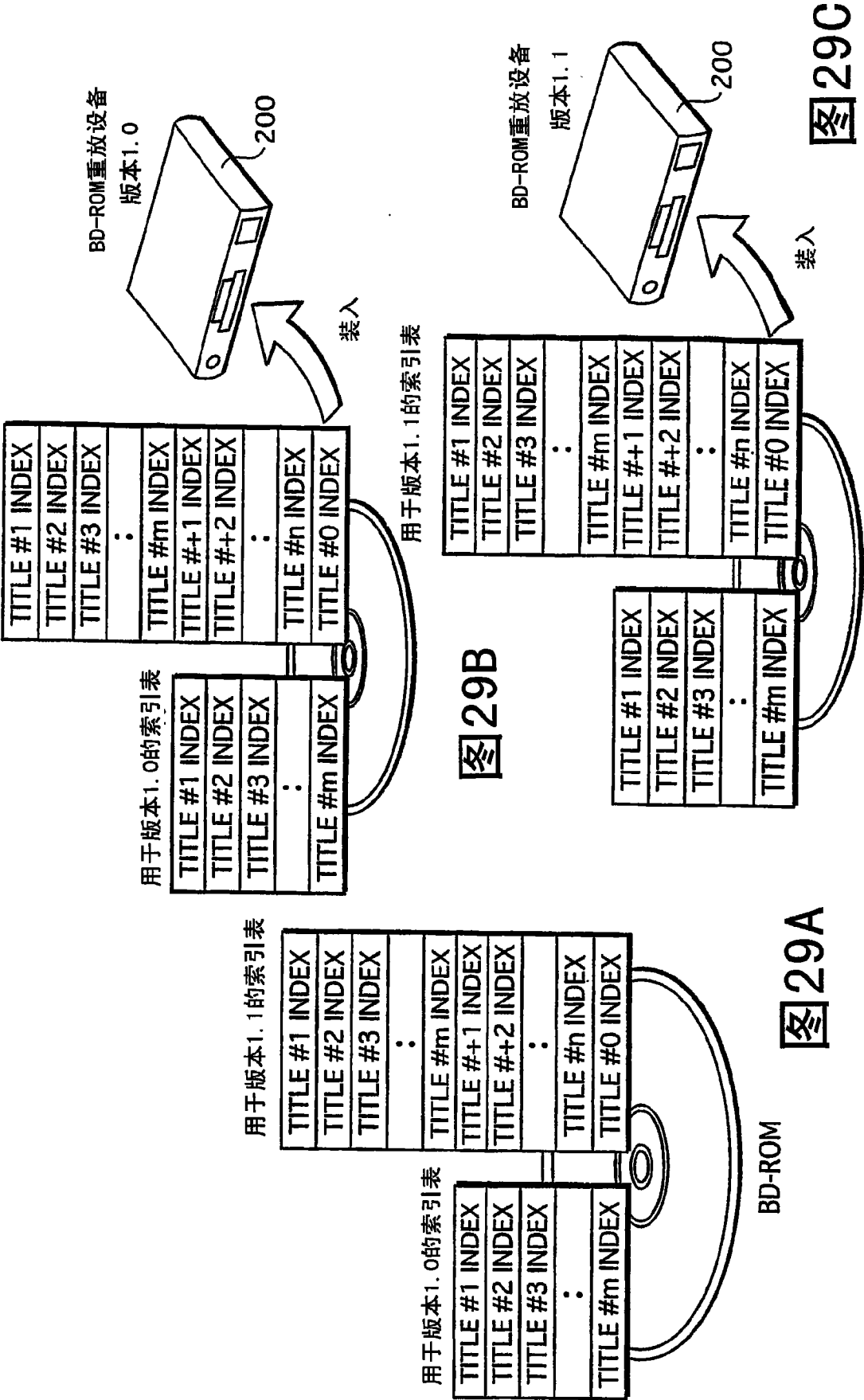


图28



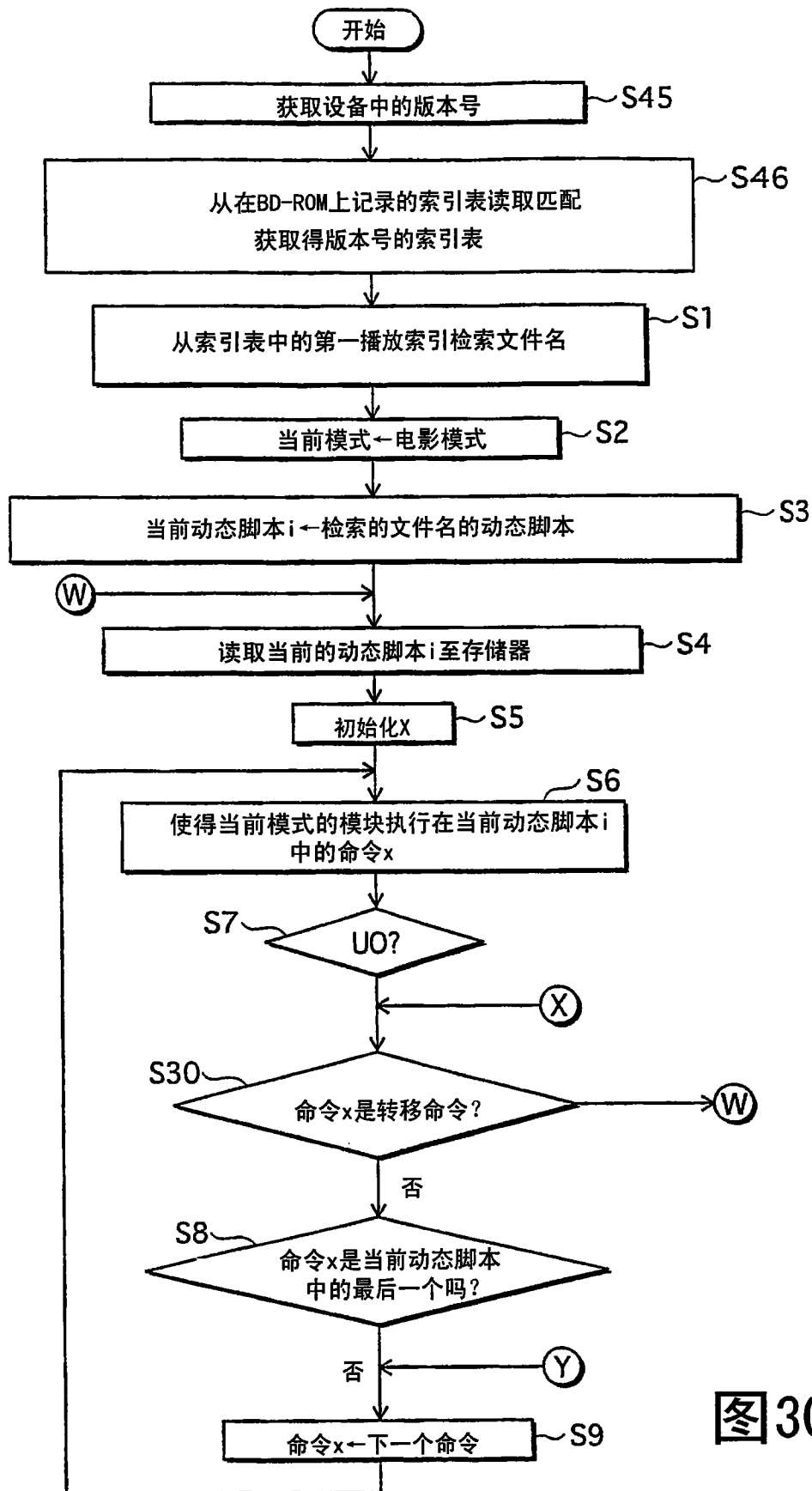


图30

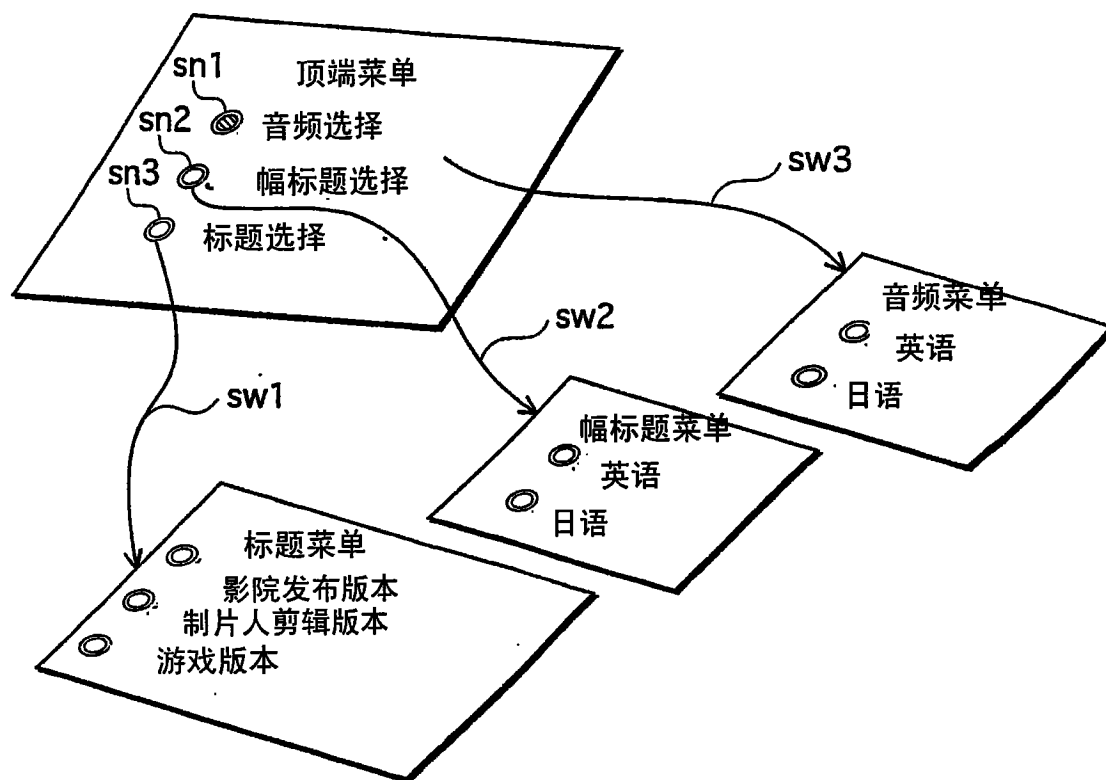


图 31

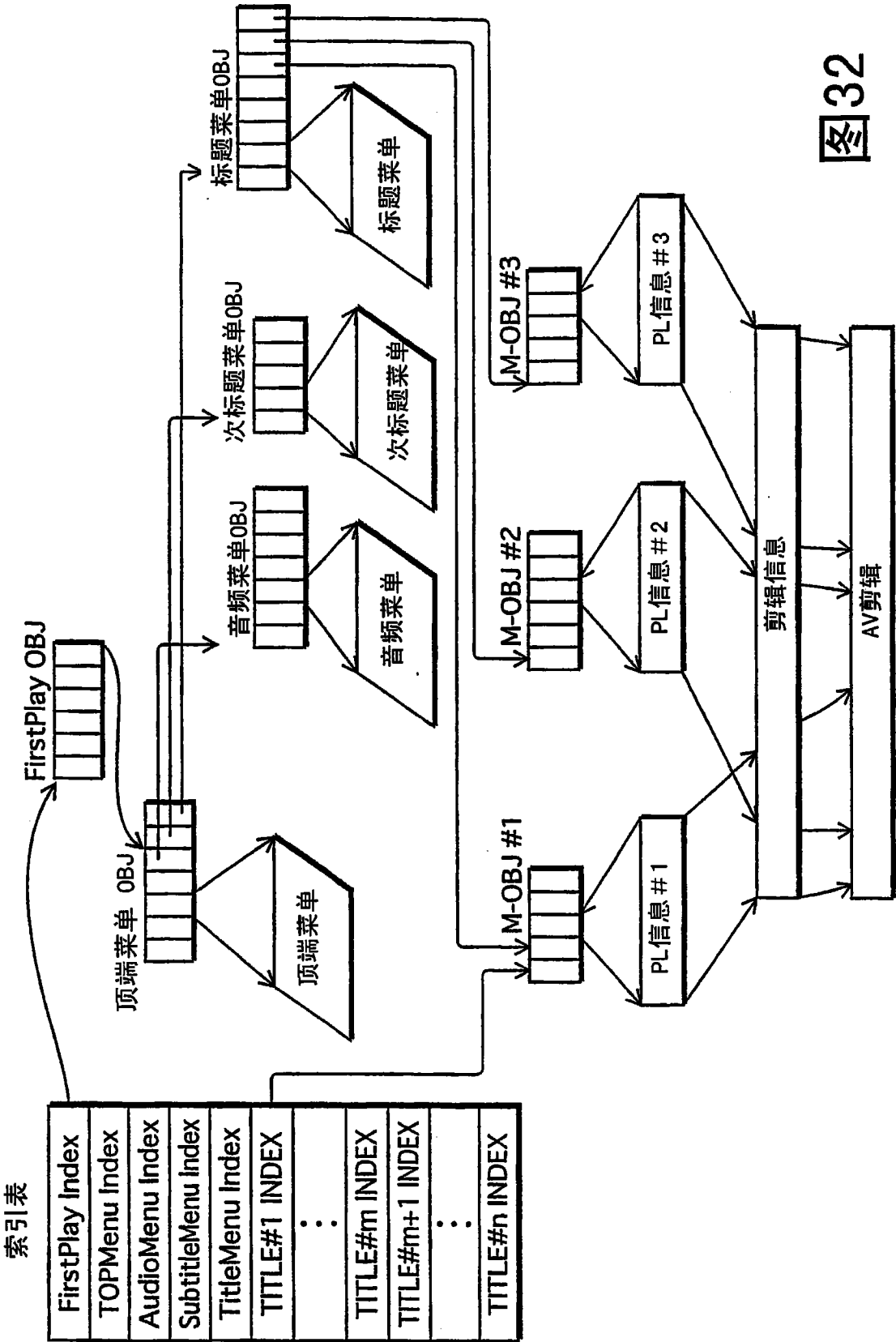
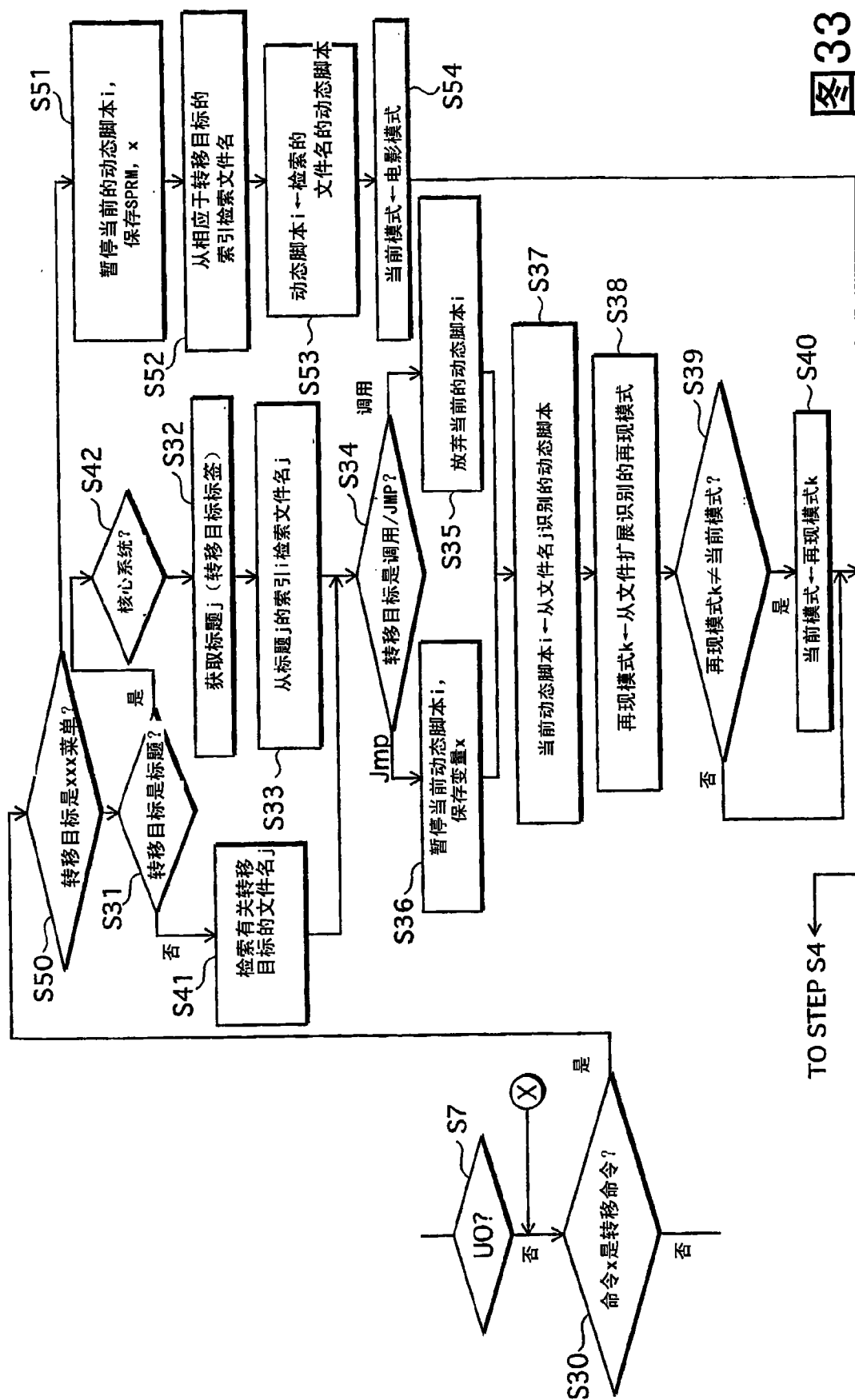


图32



33

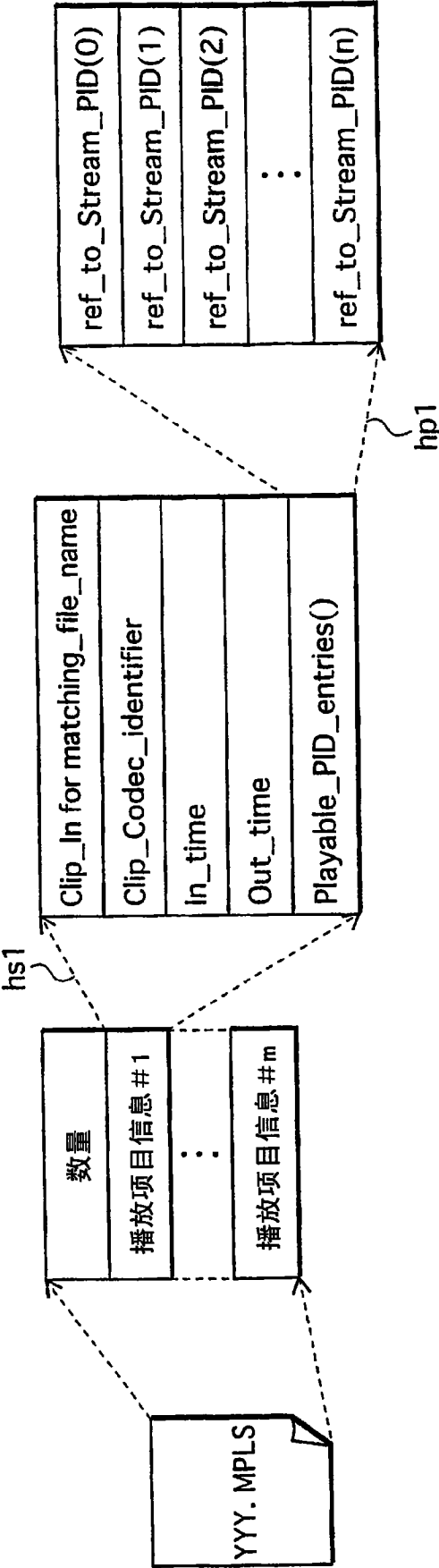


图34

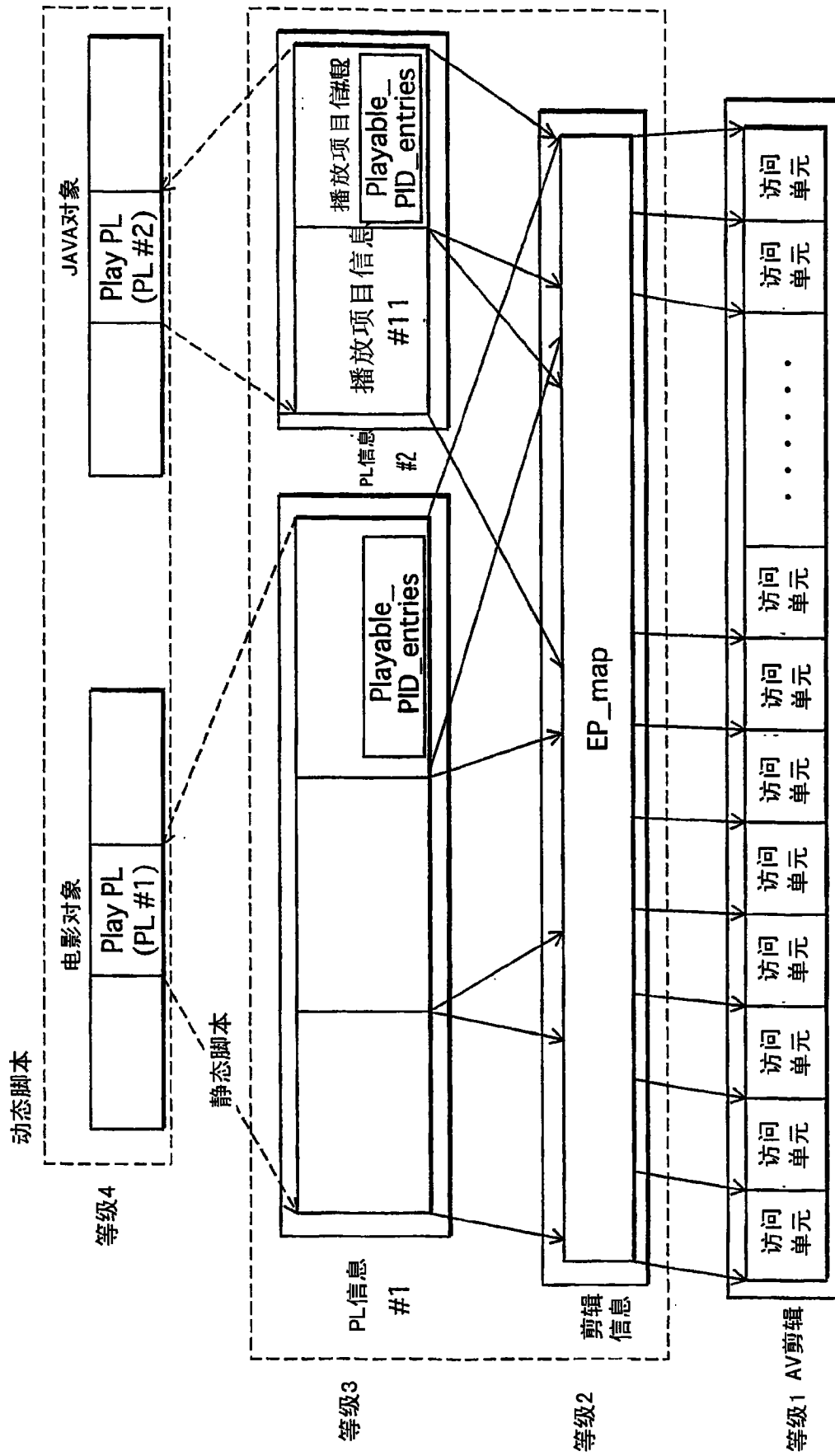


图35

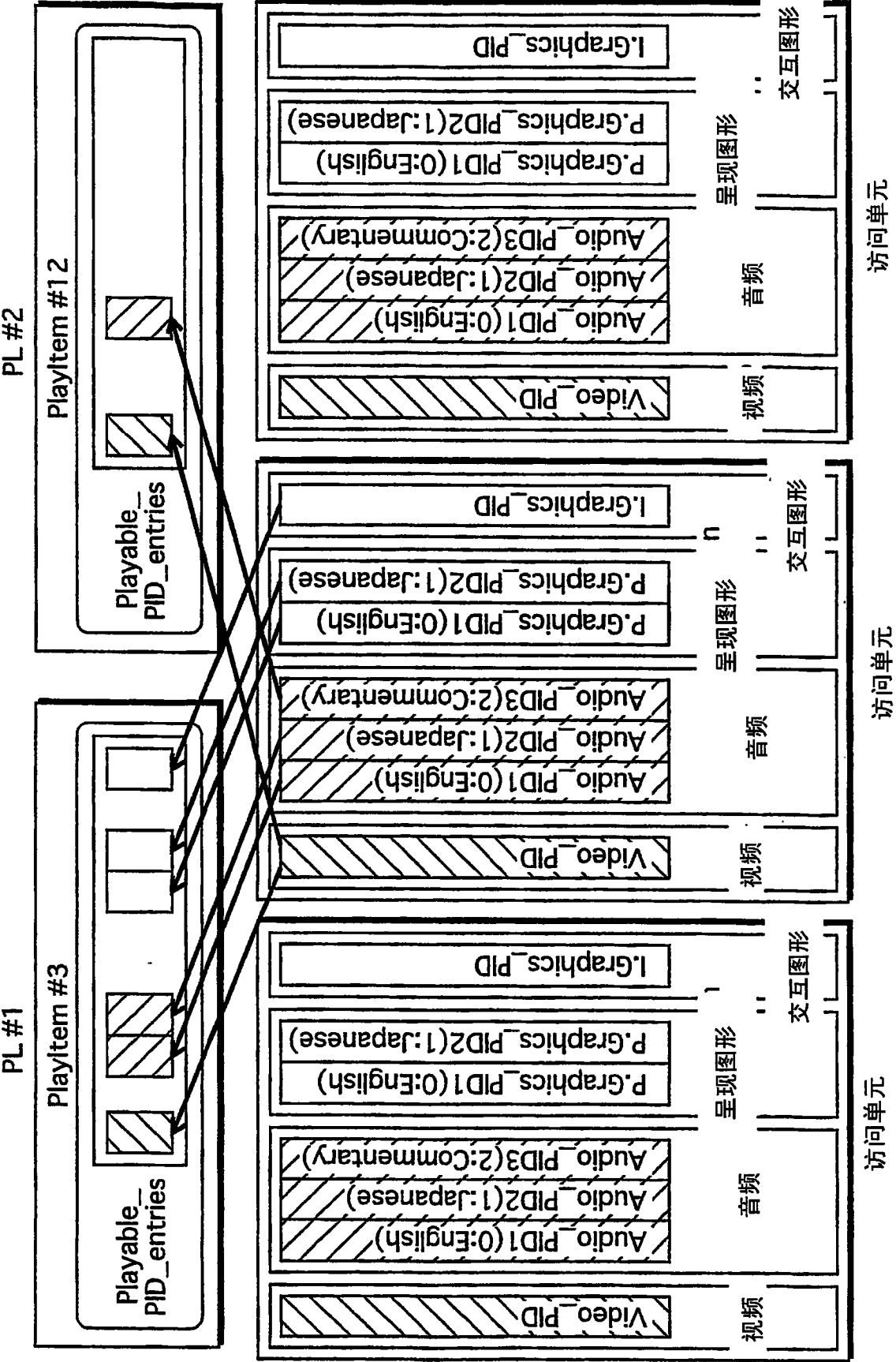


图 36

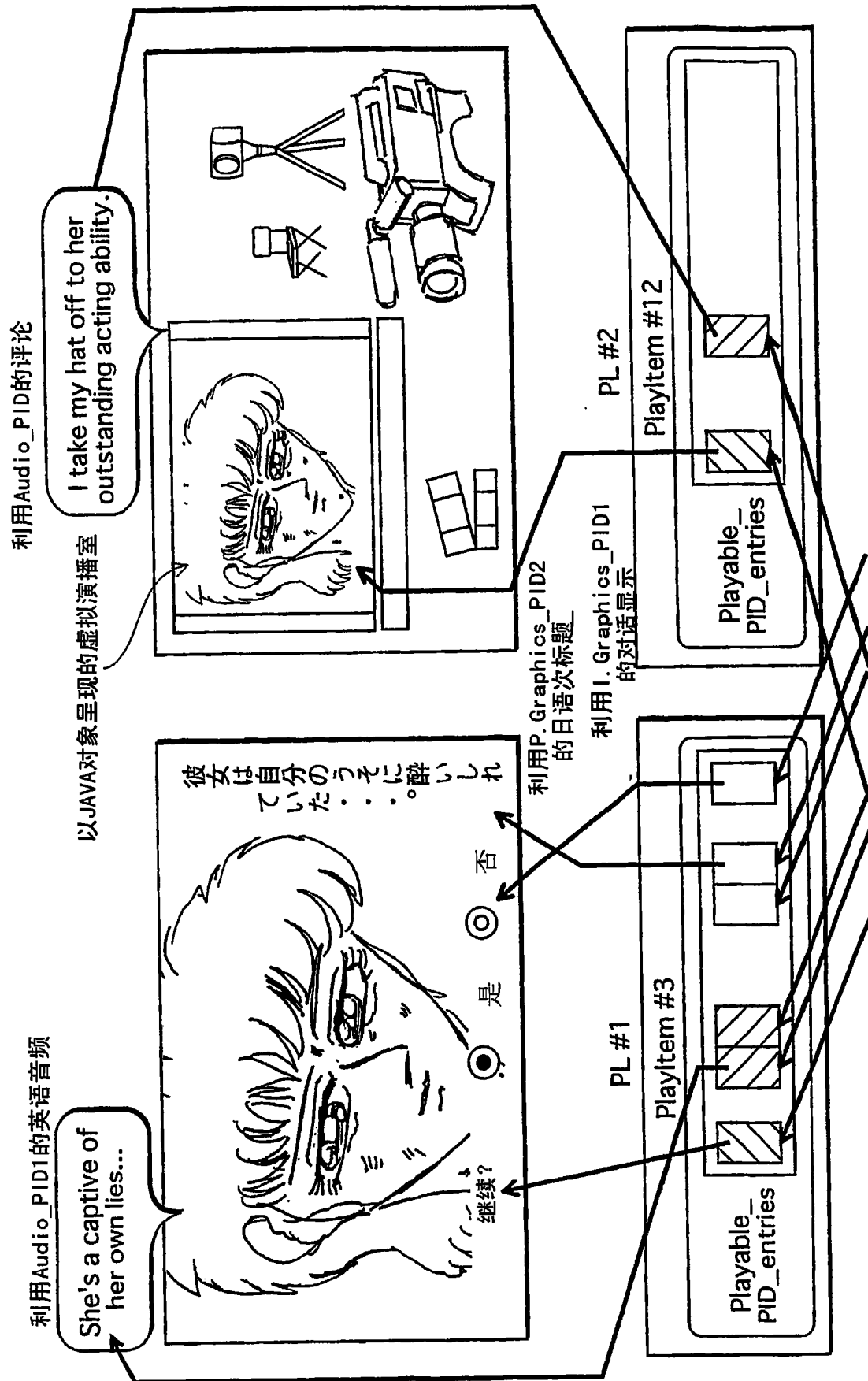
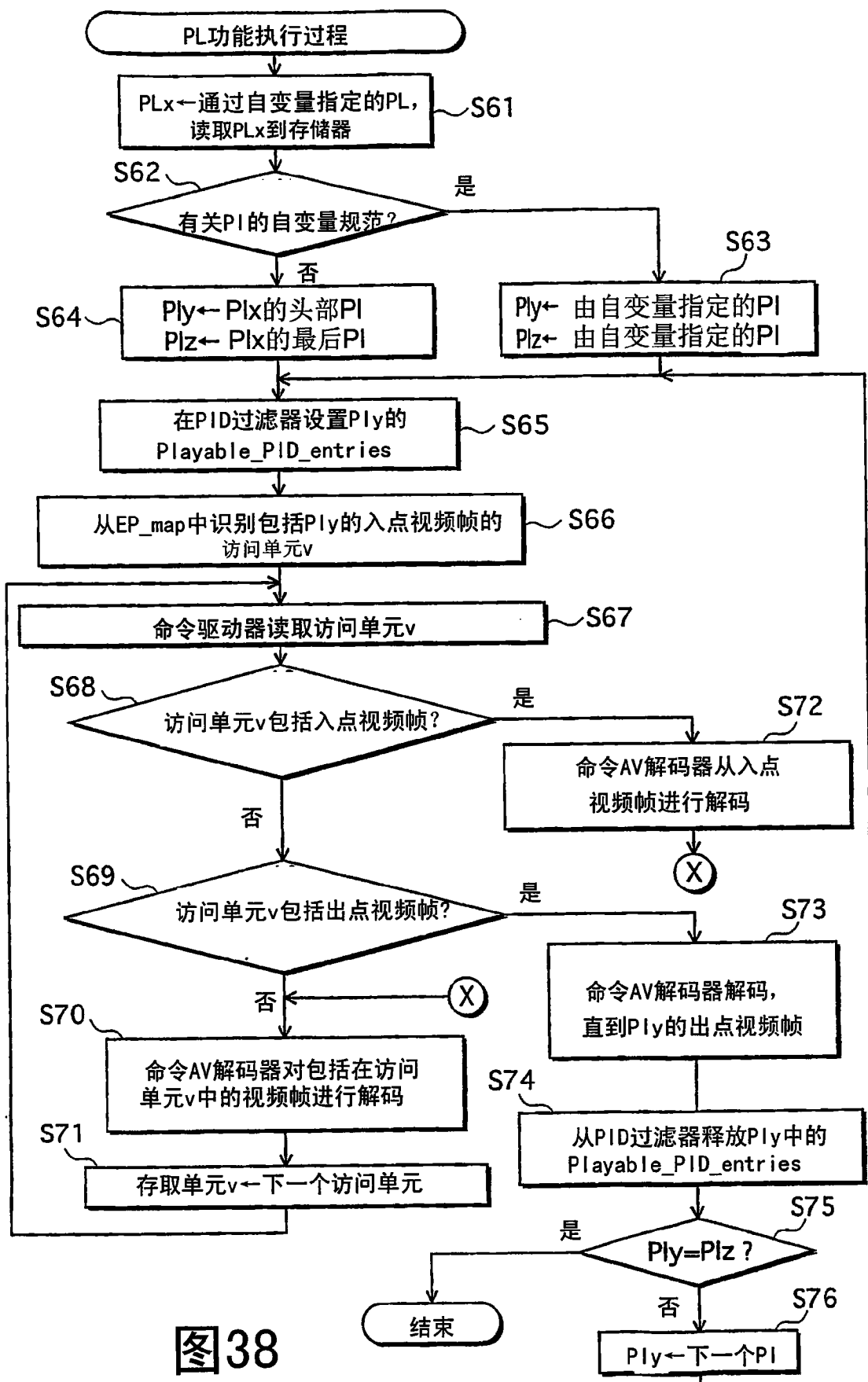


图37



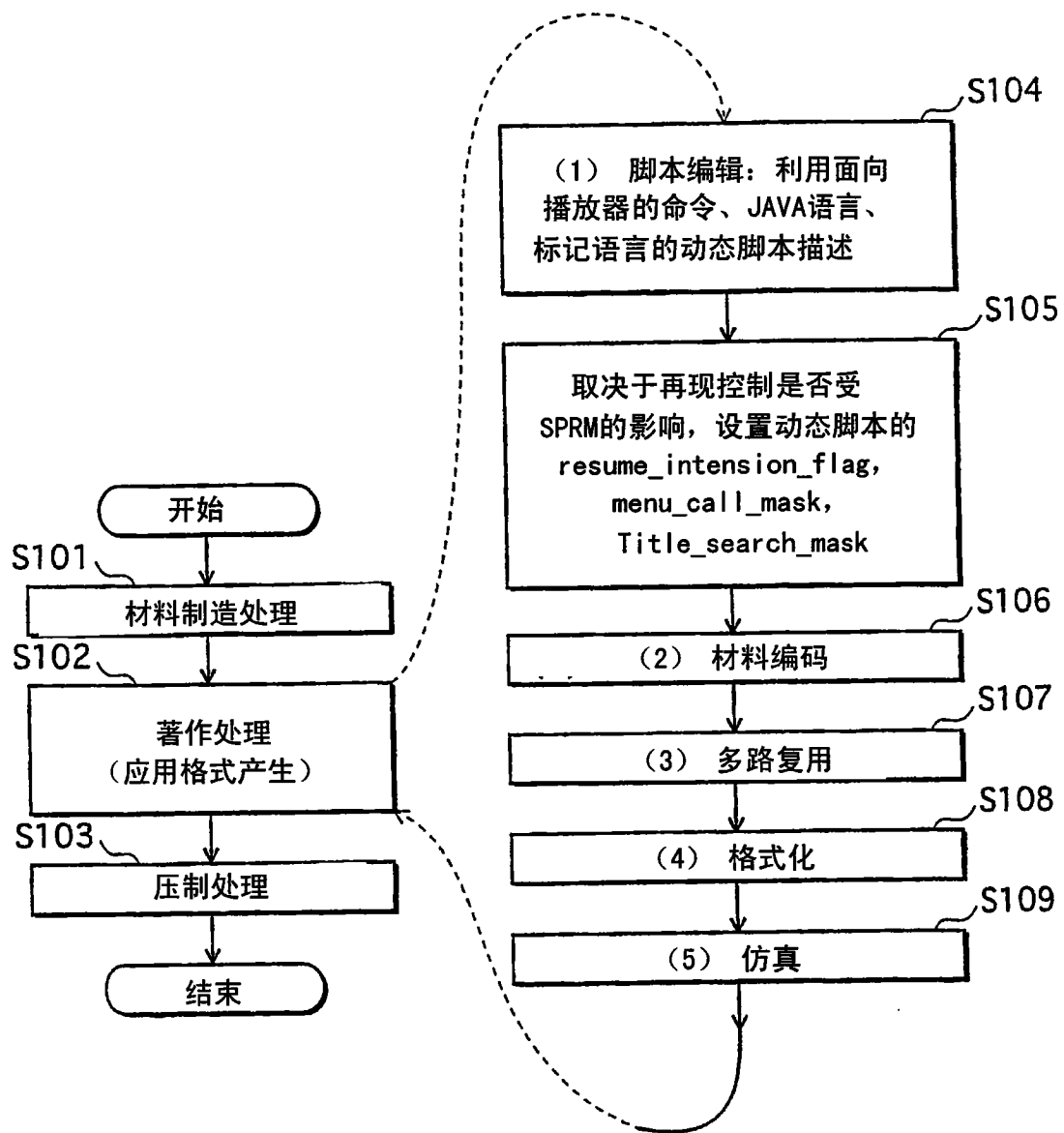


图 39