



(12) 发明专利

(10) 授权公告号 CN 101600122 B

(45) 授权公告日 2011. 10. 05

(21) 申请号 200910146619. 2

US 6538656 B1, 2003. 03. 25, 全文.

(22) 申请日 2005. 11. 30

JP 2004-334058 A, 2004. 11. 25, 摘要.

(30) 优先权数据

EP 1328114 A1, 2003. 07. 16, 第 [0001]、
[0031]、[0038]、[0080]、[0084]、[0149-0181], 图
1、9、10.

2004-349143 2004. 12. 01 JP

(62) 分案原申请数据

US 20030231259 A1, 2003. 12. 18, 全文.

200580041501. 3 2005. 11. 30

审查员 陈荣华

(73) 专利权人 松下电器产业株式会社

地址 日本大阪府

(72) 发明人 桥本敏史 大芦雅弘 冈田智之

池田航

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 陈萍

(51) Int. Cl.

H04N 9/804 (2006. 01)

(56) 对比文件

JP 2004-304767 A, 2004. 10. 28, 全文.

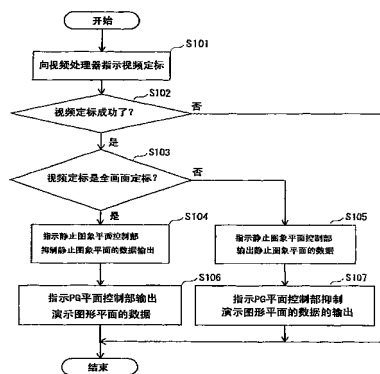
权利要求书 1 页 说明书 29 页 附图 30 页

(54) 发明名称

集成电路

(57) 摘要

一种集成电路,用于再现装置,一边执行被记录在记录介质中的应用,一边再现记录在该记录介质中的运动图像,其目的是提高合成与应用有关的图像和运动图像时的存储器总线的利用率。通过判断运动图像是否覆盖整个背景图像,来决定是否合成由应用使用的运动图像和菜单等 GUI 图形的背景中所使用的背景图像。在运动图像覆盖背景图像的情况下,没必要合成背景图像,所以能够将为了读取背景图像而使用的存储器总线,用于向存储器写入从记录介质读取的运动图像数据等,能够有效利用存储器总线带宽。



1. 一种集成电路,用于再现装置,该再现装置具有:读取单元,从记录介质读取数据;存储单元,具有视频平面、图形平面及静止图像平面的存储器区域,该视频平面用于存放与从所述记录介质读取的运动图像对应的第一数据,该图形平面用于存放与从所述记录介质读取的作为 GUI 显示的图像即 GUI 图像对应的第二数据,该静止图像平面用于存放与从所述记录介质读取的背景图像对应的第三数据;以及虚拟机部,执行与所述运动图像相关联的应用;所述集成电路具有:

运动图像存放单元,在所述视频平面存放与运动图像对应的第一数据;

图形存放单元,在所述图形平面存放与所述 GUI 图像对应的第二数据;

背景图像存放单元,在所述静止图像平面存放与所述背景图像对应的第三数据;以及

合成输出单元,在所述运动图像是能够完全遮蔽所述背景图像的规定大小的情况下,不从所述静止图像平面读取与所述背景图像对应的第三数据,而从所述视频平面读取与所述运动图像对应的第一数据,并且从所述图形平面读取与所述 GUI 图像对应的第二数据,并将表示重叠合成的合成图像的图像信号输出,在所述运动图像不是所述规定大小的情况下,从所述视频平面读取与所述运动图像对应的第一数据,从所述图形平面读取与所述 GUI 图像对应的第二数据,并且从所述静止图像平面读取与所述背景图像对应的第三数据,并将表示重叠合成的合成图像的图像信号输出;

所述规定大小是指所述运动图像在显示画面中以全屏显示的大小;

所述合成输出单元从由所述虚拟机部执行的应用,接受了对于存放以所述全屏显示的运动图像的视频平面的缩小指示的情况下,合成所述背景图像,在没有接受所述缩小指示的情况下,不合成所述背景图像。

2. 如权利要求 1 所述的集成电路,其特征在于,

所述合成输出单元、所述运动图像存放单元、所述图形存放单元以及所述背景图像存放单元共用与所述存储单元连接的存储器总线;

所述图形存放单元在判断为所述合成输出单元不进行静止图像平面的读取的情况下,将对静止图像平面的读取分配的所述存储器总线的带宽或时间,用于向图形平面的写入。

集成电路

[0001] 本申请是 2005 年 11 月 30 日提交的,中国专利申请号为 200580041501.3,发明名称为“再现装置、图像合成方法、图像合成程序及集成电路”的专利申请的分案申请。

技术领域

[0002] 本发明涉及一边再现运动图像一边执行应用的再现装置,是涉及合成图像时的存储器总线的活用方法的发明。

背景技术

[0003] 近年来开发了 DVD 播放器、BD 播放器等影像再现装置。在这些播放器中,能够从 DVD、BD 等记录介质读取运动图像流数据来再现运动图像。此时,在再现运动图像的同时,有时也执行记录在记录介质中的应用,有时除了显示运动图像之外还显示与应用关联的图像。例如,在运动图像的横向位置显示涉及运动图像的菜单 GUI(图形用户界面)等,或者显示字幕等。

[0004] 此时,播放器合成运动图像和由应用进行显示控制的图像而生成要显示的图像数据,并显示在显示器上。成为该合成的基础的数据依次从记录介质被读取,暂时被写入到播放器的存储器中的各自对应的区域。并且,播放器从该存储器读取图像数据,执行合成并输出的工作。在此,播放器的存储器中的各自对应的区域指的是,在存储器中存储运动图像的图像数据的视频平面、存储与运动图像一起显示的 GUI 菜单的图像数据的 IG(Interactive Graphics,交互图形)平面等。播放器在合成图像时需要访问各平面。

[0005] 在播放器中生成合成图像时需要高速地访问各平面,为了不延迟地再现运动图像,该访问中的存储器总线的有效活用成为重要课题。

[0006] 在对比文献 1 记载了与对存储器的写入有关的技术,根据该技术,在某个 GUI 对象中发生了再描绘的情况下,利用确定有必要修复的区域的检查单元、以及确定有必要再描绘的对象的生成单元,抑制对存储器的不必要的写入,所以能够对应于被抑制了写入的量,而有效地活用存储器总线带宽。

[0007] 但是,虽然播放器如上所述地进行对平面的写入和读取,但对存储器的写入和读取中使用的存储器总线,从成本和设置空间的问题考虑而被共用。例如,在要显示 HD 画质(1920×1080 像素)的图像的情况下,若假设 2B/像素、30fps,则在运动图像的写入中使用的存储器总线带宽成为约 120MB/sec(1920×1080×2×30),在读取时也同样要求约 120MB/sec 的性能。此外,关于背景图像的写入和读取,分别要求约 120MB/sec 的存储器总线带宽,关于 GUI 等的菜单等图形的写入和读取,若假设 4B/像素、30fps,则分别要求约 240MB/sec(1920×1080×4×30)的存储器总线带宽。此外,在有字幕的情况下,该数据也需要写入存储器并读取,依然分别要求约 120MB/sec 的存储器总线宽度。在运动图像再现时,读取和写入平行地进行,所以为了合成这些平面的图像数据,需要接近 2GB/sec 的带宽。根据这些也可以预测出,随着像质的高品质化需要更宽的存储器总线带宽,但是,同时这又成为妨碍播放器的低价格化的一个主要原因。

[0008] 专利文献 1 :日本特许公开 2004-10234 号公报

发明内容

[0009] 因此,本发明的目的在于提供一种新的再现装置,有助于实现图像数据的读取中使用的存储器总线带宽的有效活用。

[0010] 为了解决上述问题,本发明涉及一种集成电路,用于再现装置,该再现装置具有:读取单元,从记录介质读取数据;存储单元,具有视频平面、图形平面及静止图像平面的存储器区域,该视频平面用于存放与从所述记录介质读取的运动图像对应的第一数据,该图形平面用于存放与作为从所述记录介质读取的 GUI 显示的图像即 GUI 图像对应的第二数据,该静止图像平面用于存放与从所述记录介质读取的背景图像对应的第三数据;以及虚拟机部,执行与所述运动图像相关联的应用;所述集成电路具有:运动图像存放单元,在所述视频平面存放与运动图像对应的第一数据;图形存放单元,在所述图形平面存放与所述 GUI 图像对应的第二数据;背景图像存放单元,在所述静止图像平面存放与所述背景图像对应的第三数据;以及合成输出单元,在所述运动图像是能够完全遮蔽所述背景图像的规定大小的情况下,不从所述静止图像平面读取与所述背景图像对应的第三数据,而从所述视频平面读取与所述运动图像对应的第一数据,并且从所述图形平面读取与所述 GUI 图像对应的第二数据,并将表示重叠合成的合成图像的图像信号输出,在所述运动图像不是所述规定大小的情况下,从所述视频平面读取与所述运动图像对应的第一数据,从所述图形平面读取与所述 GUI 图像对应的第二数据,并且从所述静止图像平面读取与所述背景图像对应的第三数据,并将表示重叠合成的合成图像的图像信号输出;所述规定大小是指所述运动图像在显示画面中以全屏显示的大小;所述合成输出单元从由所述虚拟机部执行的应用,接受了对于存放以所述全屏显示的运动图像的视频平面的缩小指示的情况下,合成所述背景图像,在没有接受所述缩小指示的情况下,不合成所述背景图像。

[0011] 具有上述结构的再现装置,在不需要背景图像时、即运动图像遮盖了背景图像时,由于不需要合成背景图像而不需要从存储器读取,所以,此时不需要读取背景图像的数据。在运动图像以覆盖背景图像的规定大小例如全屏大小显示时,通过不读取背景数据,减轻了再现数据的负担,而且空出了用于读取的存储器总线带宽。

[0012] 这样,通过限制不必要的背景图像的读取,能够将背景图像的读取中使用的存储器总线用于其它目的、例如 GUI 图像向存储器的写入等,所以能够有效活用存储器总线带宽,还能够减轻向显示器的显示延迟的问题。

[0013] 通常,合成的平面是将相同大小的平面重叠合成,但也有运动图像从原来的全屏大小变更大小进行显示的情况下,即不能完全遮盖背景图像的情况,此时需要背景图像,所以进行合成。

[0014] 在如上所述的 HD 像质的情况下,如果再现装置将用于背景图像读取的 120MB/秒的存储器总线带宽例如用于运动图像的写入,则运动图像的写入速度会加倍,能够防止因在读取时运动图像的数据未被写入而不能读取,运动图像的再现出现延迟的问题。

[0015] 此外,也可以是,上述规定大小是指上述运动图像在显示画面中以全屏显示的大小;上述再现装置还具有执行与上述图像有关的应用的虚拟机部;上述合成输出单元接受上述应用对于上述视频平面的缩小指示,合成上述背景图像,在没有缩小指示的情况下,不

合成上述背景图像。

[0016] 在此,全屏是指,在以连接在显示装置上的外部设备的电视机等为代表的显示装置、或者内置的监视器等上,覆盖其显示画面整体的大小。

[0017] 再现装置根据正在执行的应用的指示进行运动图像的大小变更,该变更指示通常只对视频平面发出,但接受了该指示的合成输出单元判断是否执行来自静止图像平面的数据的读取。通过具备该结构,能够容易地判断来自静止图像平面的数据输出的必要性。

[0018] 此外,也可以是,上述静止图像平面还用于存储与上述运动图像不同的、根据时间切换的图像;上述再现装置还具备在上述静止图像平面存储根据上述时间切换的图像的图像存储单元;在上述运动图像是在显示画面中以全屏显示的大小的情况下,上述合成输出单元读取存储在上述视频平面中的运动图像、存储在上述图形平面中的 GUI 图像和存储在上述静止图像平面中的根据时间切换的图像,输出表示重叠合成的合成图像的图像信号,在上述运动图像不是以全屏显示的大小的情况下,上述合成输出单元读取存储在上述视频平面中的运动图像、存储在上述图形平面中的 GUI 图像和存储在上述静止图像平面中的背景图像,输出表示重叠合成的合成图像的图像信号。

[0019] 在此,根据时间切换的图像是与由运动图像存储单元存储的运动图像不同、且根据时间显示内容不同的图像,例如用于显示字幕的图像等。

[0020] 由此,在存储单元中,以前通常分别准备了静止图像平面和用于存储字幕的字幕平面的存储器区域能够被共用,从而空出了相当于一个平面的存储器区域,空出的存储器区域能够分配给其它数据的存储,所以,能够活用存储器。而且,同以前那样具有静止图像平面和字幕平面的情况相比,减少了一个平面,当然对应于减少的一个平面,存取的存储器总线带宽也能够以较小值实现。

[0021] 此外,也可以是,上述合成输出单元及上述各存储单元共用与上述存储单元连接的存储器总线;上述图形存储单元在判断为上述合成输出单元不进行静止图像平面的读取的情况下,将对静止图像平面的读取分配的上述存储器总线的带宽或时间,用于对图形平面的写入。

[0022] 由此,再现装置将分配给背景图像读取的存储器总线带宽用于图形平面的写入,可以说能够有效地利用存储器总线带宽。关于对图形平面的写入,GUI 图像同运动图像和背景图像相比是高品质的,其写入所需的时间变长、或者存储器总线带宽变宽,但是,通过将预定用于来自静止图像平面的数据读取用的存储器总线,用于对图形平面的写入,能够实现高速化。

[0023] 通过搭载该集成电路,再现装置在不必要的情况下不进行从静止图像平面的数据读取,所以,能够将存储器总线带宽相应地空出来。

附图说明

[0024] 图 1 是表示本发明涉及的再现装置的使用行为的图。

[0025] 图 2 是表示第一实施方式涉及的再现装置 100 的功能结构的方框图。

[0026] 图 3 是将由硬件和软件构成的部分置换为层结构进行描绘的图。

[0027] 图 4 是表示 Java(注册商标)虚拟机 36 的功能结构的图。

[0028] 图 5 是表示存储在 BD-ROM110 中的数据的目录构造的图。

- [0029] 图 6 是示意性地表示赋予了后缀 .m2ts 的文件如何构成的图。
- [0030] 图 7 是表示 Clip 信息的构成的图。
- [0031] 图 8 是表示对于电影的视频流的 EP_map 设定的图。
- [0032] 图 9 是表示 PlayList 信息的数据构造的图。
- [0033] 图 10 是表示 AVClip 和 PlayList 信息的关系的图。
- [0034] 图 11 是表示包含在 PlayList 信息中的 STN 表的图。
- [0035] 图 12(a) ~ (d) 是表示 entry-attribute 的详细内容的图。
- [0036] 图 13 是表示 PlayList 信息的 PlayListMark 信息的内部结构的图。
- [0037] 图 14 是表示由 PlayList 信息的 PlayListMark 信息进行的章节位置指定的图。
- [0038] 图 15 是表示 BD-J Object 的内部结构的图。
- [0039] 图 16 是表示由档案文件收容的程序和数据的图。
- [0040] 图 17(a) 是表示应用管理表的内部结构的图, (b) 是表示构成应用管理表的信息要素的含义内容的图。
- [0041] 图 18 是表示盘内容中的状态转移的图。
- [0042] 图 19(a) 是表示 BD-ROM 整体的时间轴的图, (b) 是表示 BD-ROM 整体的时间轴上的构成的图。
- [0043] 图 20(a)、(b) 是表示在 BD-ROM 整体的时间轴中, 由 BD-J Object 确定的标题再现区间的图。
- [0044] 图 21 是表示在图 20(b) 的时间轴上规定的、生存区间的典型例的图。
- [0045] 图 22 是表示包含正编标题、网络购物标题、游戏标题这三个标题的盘内容的图。
- [0046] 图 23(a)、(b) 是表示应用管理表、生存区间的一例的图。
- [0047] 图 24 是表示起动属性所能取的三种状态 (Present, AutoRun, Suspend) 和之前标题中的应用状态的三种状态 (非起动、起动中、Suspend), 能够取的组合的图。
- [0048] 图 25(a) 是表示播放列表管理表的内部结构的图, (b) 是表示构成播放列表管理表的信息要素的含义内容的图。
- [0049] 图 26 是由播放列表管理表、应用管理表规定的标题的具体例的图。
- [0050] 图 27 是表示当前标题所能取的三种状态 (没有播放列表管理表、有播放列表管理表且无指定、有播放列表管理表且自动播放 (AutoPlay)) 和之前标题中的 PL 的状态 (非再现状态、再现中状态) 能够取的 6 种组合的图。
- [0051] 图 28(a) 是表示播放列表管理表及应用管理表的记述例的图, (b) 是表示通过如图 27(a) 所示地记述的应用管理表、播放列表管理表, 播放列表再现和应用执行如何进行的图。
- [0052] 图 29(a) 是表示播放列表管理表的其他记述例的图, (b) 是表示基于图 28(a) 所示情况进行应用执行和播放列表再现的图。
- [0053] 图 30 是表示判断是否合成静止图像平面的再现装置 100 的动作的流程图。
- [0054] 图 31 是表示第二实施方式的再现装置 100 的功能结构的方框图。
- [0055] 图 32 是表示在第二实施方式中根据应用的指示和视频定标的指示进行来自 Stii 平面和 PG 平面的数据输出的控制的流程的流程图。

具体实施方式

[0056] < 第一实施方式 >

[0057] 下面,使用附图说明本发明的一实施方式的再现装置。

[0058] < 构成 >

[0059] < 使用方式的构成 >

[0060] 首先,图 1 中示出了关于再现装置的使用行为的方式。

[0061] 如图 1 所示,再现装置 100 安装 BD-ROM110 后再现被记录在 BD-ROM110 中的运动图像,显示在有线或无线连接的电视机 130 上。而且,再现装置 100 执行记录在 BD-ROM110 中的 Java(注册商标)应用,还显示基于该应用的图像。

[0062] 为了接受用户操作,设有遥控器 120,在该遥控器 130 上安装了用于控制再现装置 100 的各种键。该各种键中包括:接受再现开始(Play)、再现停止(Stop)、暂停(Pause ON)、暂停的解除(Pause OFF)、静止图像功能的解除(Still off)、带速度指定的快进(ForwardPlay(speed))、带速度指定的快退(Backward Play(speed))、声音切换(Audio Change)、副影像切换(Subtitle Change)、角度切换(AngleChange)的键,接受菜单操作时的焦点移动操作的 MoveUp 键、MoveDown 键、MoveRight 键、MoveLeft 键,接受菜单显示操作的 Pop-up 键,接受数值输入的 Numeric 键。

[0063] 以上是关于本发明涉及的再现装置 100 的使用行为方式的说明。

[0064] < 再现装置 100 的硬件结构 >

[0065] 接着,使用图 2 说明再现装置 100 的功能构成。

[0066] 本发明涉及的再现装置 100 是基于本图所示的内部结构而工业生产的。本发明涉及的再现装置 100 主要包括系统 LSI 和驱动装置这 2 个部件,通过将 these 部件安装在装置的壳体及基板上,能够工业化生产。系统 LSI 是集成了实现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 100 包括:BD-ROM 驱动器 1,读缓冲器 2,信号分离器 3,视频解码器 4,视频平面 5,声音处理器 6,声音处理器 7,混合器 8,声音控制器 9, D/A 变换器 10,交互图形(InteractiveGraphics)解码器 11,交互图形平面 12,演示图形(PresentationGraphics)解码器 13,演示图形平面 14, JPEG 解码器 15,静止图像平面 16,静止图像平面控制部 17,合成部 18a、18b、18c, STC-DELTA 附加部 19, ATC-DELTA 附加部 20, 局部存储器 21, 指令 ROM22, 用户事件处理部 23, PSR 组 24, CPU25, 脚本存储器 26, 局部存储器 27。

[0067] BD-ROM 驱动器 1 进行 BD-ROM 的装载 / 卸载,具有执行对 BD-ROM 的访问的功能。

[0068] 读缓冲器 2 是 FIFO(先进先出)存储器,具有依次存储从 BD-ROM 读取的 TS 包、并从先存储的包开始一次输出的功能。

[0069] 信号分离器(De-MUX)3 具有从读缓冲器 2 中取出资源(Source)包,将构成该资源包的 TS 包变换为 PES 包的功能。并且具有如下功能:从通过变换得到的 PES 包中的被记载在 STN Table 的包中,将具有 PID(Packet Identifier)的包输出给视频解码器 4、音频解码器 6、交互图形解码器 11、演示图形解码器 13 中的某一个的功能。关于资源包和 STN_Table 的详细内容,在后面记述。

[0070] 视频解码器 4 具有对从信号分离器 3 输出的多个 PES 包进行解码而得到非压缩格式的图片并写入到视频平面 5 上的功能。

[0071] 视频平面 5 是用于事先存储非压缩格式的图片的平面。平面指的是在再现装置 100 中用于存储相当于一个画面的像素数据的存储器区域。视频平面具有存储并输出由视频解码器写入的数据的功能。视频平面 5 中的析像度是 1920×1080 ，存储在该视频平面 5 中的图片数据，由用 16 比特的 YUY 值表现的图像数据构成。在视频平面 5 中，能够对视频流中的每个帧的再现图像进行定标。定标是指将每一个帧的再现图像变更为视频平面 5 整体的 $1/4$ （称为四分之一化）和 $1/1$ （称为全屏）中的哪一个。在 BD-J 模式中按照来自 CPU25 的指示来执行这样的定标，所以能够进行将视频流的再现图像缩在画面的一角、或者在整个画面显示的画面演出。

[0072] 声音处理器 6 包括：音频缓冲器 6a，具有在 PID 滤波器 3 输出了构成音频流的 PES 包时存储该 PES 包的功能；以及音频解码器 6b，具有对存储在该缓冲器中的 PES 包进行解码，输出 PCM (Pulse Code Modulation, 脉冲编码调制) 状态的音频数据的功能。

[0073] 声音处理器 7 包括：预加载缓冲器 7a，具有预加载从 BD-ROM 读出的文件 sound.bdmv 的功能；以及音频解码器 7b，具有对预加载的文件 sound.bdmv 中的多个声音数据中的由 CPU25 指示的声音数据进行解码，输出 PCM 状态的音频数据的功能。向预加载缓冲器 7a 的预加载，希望在 BD-ROM 的装载时或标题切换时进行。其理由是，如果要在 AVClip 的再现中读出文件 sound.bdmv，则会发生用于读取 AVClip 以外的其它文件的光拾取器的搜索。另一方面，在 BD-ROM 的装载时或标题切换时，很少会继续进行 AVClip 的再现，所以通过在这样的定时读取文件 sound.bdmv，能够保证不中断 AVClip 再现。

[0074] 声音混合器 8 具有混合从声音处理器 6 和声音处理器 7 输出的 PCM 状态的声音数据的功能。

[0075] 声音控制器 9 具有如下功能：对输出从声音混合器 8 输出的已解压状态的音频数据和不经声音处理器 6 的压缩状态的音频数据中的哪一个进行切换。在输出压缩状态的音频数据时，在其输出目的地的设备（电视机）中被解码。

[0076] D/A 变换器 10 具有对从声音合成器 8 输出的数字的音频数据进行 D/A 变换并输出模拟声音的功能。

[0077] 交互图形解码器 (IG 解码器) 11 具有对从 BD-ROM 或局部存储器 21 读出的 IG 流进行解码，将非压缩图形写入到交互图形平面 12 的功能。

[0078] 交互图形 (IG) 平面 12 中写入在 HDMV 模式下通过交互图形解码器 11 的解码而得到的非压缩图形。此外，在 BD-J 模式中，写入由应用描绘的字符和图形。

[0079] 演示图形 (PG) 解码器 13 具有对从 BD-ROM 或局部存储器 21 输出的 PG 流进行解码，将非压缩图形写入到演示图形 (PG) 平面 11 中的功能。通过合成由演示图形解码器 13 解码并被写入到演示图形平面 11 中的数据，字幕显示在画面上。

[0080] 演示图形平面 14 是具有相当于一个画面的区域的存储器，能够存储一个画面的非压缩图形。

[0081] JPEG 解码器 15 具有对被记录在 BD-ROM 或局部存储器 21 中的 JPEG 数据进行解码并写入到静止图像平面 16 的功能。

[0082] 静止图像平面 16 是存储有将 JPEG 数据解压缩而得到的非压缩的图形数据的平面。该图形数据作为 Java (注册商标) 应用所描绘的、GUI 框架的所谓“壁纸”而使用。

[0083] 静止图像平面控制部 17 具有控制从静止图像平面 16 的数据输出的功能。具体地

说,从CPU25接受对视频平面5的定标指示,在该定标指示是向电视机130整体的显示指示即全屏定标的情况下,抑制从静止图像平面16的数据读取。并且,在定标指示是全屏的1/4即四分之一定标的情况下,从静止图像平面16向合成部18c输出所存储的静止图像数据。

[0084] 合成部18a、18b、18c具有分别合成存储在交互图形平面12中的数据、存储在演示图形平面11中的数据、存储在视频平面5中的数据和存储在静止图像平面16中的数据,并输出所得到的合成图像的图像信号的功能。并且,从平面未输出图像信号的情况下,直接输出没进行合成输出的信号。

[0085] STC_delta附加部18具有生成STC(System Time Clock,系统时钟)的功能。并且,在STC_Sequence的切换时,通过在至今为止的STC_Sequence中的STC值上加上称为STC_delta的偏移值,求出新的STC_Sequence的STC值(STC2),将至今为止的STC_Sequence中的STC值(STC1)和新的STC_Sequence的STC值(STC2)作为连续的值。

[0086] 设定在先行STC_Sequence中最后再现的图片的显示开始时刻为PTS1(1stEND)、图片的显示期间为Tpp、在后续STC_Sequence中最初显示的图片的开始时刻为PTS2(2ndSTART)的情况下,STC_delta表现为:

[0087] $STC_delta = PTS1(1stEND) + Tpp - PTS2(2ndSTART)$

[0088] 如上所述地求出STC_delta,向各解码器输出该STC_delta被加起来的时钟的计数值。由此,各解码器能够不中断地再现相当于2个STC_Sequence的数据流。如此,即使在1个AVClip中存在2个以上的STC_Sequence,而且应该连续再现的2个以上的AVClip的每个具有不同的STC_Sequence,也能够无间隙地执行这些STC_Sequence间的解码处理。

[0089] 并且,为了满足缓冲的连续性,希望满足以下的条件1)、2)。

[0090] 1) 满足 $STC2(2ndSTART) > STC2(1stEND)$;

[0091] 2) 由投影在相同时间轴上的STC1、STC2定义从TS1的TS包取出和从TS2的TS包取出,不会导致缓冲器的下溢(underflow)和溢出。

[0092] 并且,在1)中,STC2(1stEND)是将STC1(1stEND)投影在STC2的时间轴上的值,由 $STC2(1stEND) = STC1(1stEND) - STC_delta$ 这样的计算式给出。

[0093] ATC_delta附加部19具有生成ATC(Arrival Time Clock)的功能。并且,通过在STC_Sequence的切换时,在至今为止的STC_Sequence中的ATC值(ATC1)上加上称为ATC_delta的偏移值,使至今为止的STC_Sequence中的ATC值(ATC1)和新的STC_Sequence的ATC值(ATC2)成为连续的值。通过该加法运算, $ATC2 = ATC1 + ATC_delta$ 。ATC_delta指的是,从至今读出的传输流(TS1)的最后的TS包的输入时刻T1起,到新读出的传输流(TS2)的最初的TS包的输入时刻T2为止的偏移值,由“ $ATC_delta \geq N1/TS_recording_rate$ ”这样的计算式给出。在此,输入时刻T2意味着将TS2的最初的TS包的输入时刻投影到TS1的时间轴上的时刻。此外,N1是接在TS1的最后的视频PES包后面的TS包的包数。在BD-ROM中,这样的ATC_delta被记述在Clip信息中,所以通过使用它,能够计算ATC_delta。通过以上的计算,能够使至今为止的ATC_Sequence具有的ATC值(ATC1)和新的ATC_Sequence具有的ATC值(ATC2)成为连续的值。通过向信号分离器(De-MUX)3输出相加了ATC_delta的时钟的计数值,能够实现无中断的缓冲控制。

[0094] 以上是涉及AVClip的再现的构成要素。接着说明与BD-J模式下的动作有关的构成要素(局部存储器21~局部存储器27)。

[0095] 局部存储器 21 是具有将从网页下载的内容等、从 BD-ROM 以外的记录介质和通信介质供给的内容,同元数据一起存储的功能的硬盘。该元数据是用于将下载内容存到局部存储器 21 中进行管理的信息,通过访问该局部存储器 21,BD-J 模式下的应用能够进行利用了下载内容长度的各种处理。

[0096] 接着,说明实现再现装置中的综合控制的构成要素(指令 ROM22 ~ 局部存储器 26)。

[0097] 指令 ROM22 存储着对再现装置的控制进行规定的软件。

[0098] 用户事件处理部 23 根据对遥控器或再现装置的前面板的键操作,向 CPU25 输出执行该操作的用户事件。

[0099] PSR(Player Status Register,播放状态寄存器)装置 24 是内置于再现装置中的寄存器,包括 6 个 PSR 和 4096 个 GPR(General Purpose Register,通用寄存器)。播放状态寄存器的设定值(PSR)中,PSR4 ~ PSR8 用于表现当前的再现时刻。

[0100] PSR4 通过被设定为 1 ~ 100 的值,表示当前的再现时刻所属的标题,通过被设定为 0,表示当前的再现时刻是顶级菜单。

[0101] PSR5 通过被设定为 1 ~ 999 的值,表示当前的再现时刻所属的章节号,通过被设定为 0xFFFF,表示在再现装置中章节号是无效的。

[0102] PSR6 通过被设定为 1 ~ 999 的值,表示当前的再现时刻所属的播放列表(当前 PL)的号码。

[0103] PSR7 通过被设定为 1 ~ 255 的值,表示当前的再现时刻所属的 PlayItem(当前 PlayItem)的号码。

[0104] PSR8 通过被设定为 0 ~ 0xFFFFFFFF 的值,使用 45KHz 的时间精度来表示当前的再现时刻(当前 PTM(演示时刻))。利用以上的 PSR4 ~ PSR8,能够在图 18(a) 中的 BD-ROM 全体的时间轴上确定当前的再现时刻位于何处。

[0105] CPU25 具有执行存储在指令 ROM22 的软件,并执行再现装置整体的控制的功能。该控制内容根据从用户事件处理部 23 输出的用户事件、以及 PSR 组 24 中的各 PSR 的设定值,动态地变化。

[0106] 脚本存储器 26 是具有存储当前的 PL 信息和当前的 Clip 信息的功能的存储器。当前 PL 信息是指,记录在 BD-ROM 中的多个播放列表信息中的、成为当前处理对象的信息。当前 Clip 信息是指,记录在 BD-ROM 中的多个 Clip 信息中的、成为当前处理对象的信息。

[0107] 局部存储器 27 是因从 BD-ROM 的读取是低速的,所以用于暂时存储 BD-ROM 的记录内容的高速缓冲存储器。由于存在这样的局部存储器 27,BD-J 模式下的应用执行更有效。

[0108] 以上是本实施方式涉及的再现装置的硬件结构。

[0109] < 再现装置 100 的软件构成 >

[0110] 接着,说明本实施方式涉及的再现装置的软件构成。

[0111] 图 3 是将由存储在 ROM24 的软件和硬件构成的部分替换为层结构来描绘的图。如该图所示,再现装置的层结构包括以下的 a)、b)、c)。即,包括

[0112] a)BD Player Device 的第一层,

[0113] b)BD Player Model 的第二层,

[0114] c)Application Runtime Enviroment 的第三层。

[0115] 这些层中,图 3 所示再现装置的硬件结构属于第一层。本图的第一层“BD Player Device”中,包括图 2 所示硬件结构中的相当于视频解码器 4、交互图形解码器 11、音频解码器 6 等的“解码器”,相当于视频平面 5、交互图形平面 12 等的“平面”,BD-ROM110 及其文件系统,局部存储器 21 及其文件系统。

[0116] 第二层“BD Player Model”由以下的 b1)、b2) 的层构成。即,包括

[0117] b1) 虚拟文件系统 (Virtual File System)30 及演示引擎 (Presentation Engine)31 的层、

[0118] b2) 再现控制引擎 (Playback Control Engine)32 的层,对于比自身上位的层,提供函数 API。

[0119] 第三层“Application Runtime Enviroment”由以下的 c1)、c2) 的层构成。即,包括

[0120] c1) 模块管理器 34 存在的层,

[0121] c2)BD-J 平台 35 存在的层。

[0122] 首先,说明属于第二层的虚拟文件系统 30 ~ 模块管理器 34。

[0123] 虚拟文件系统 30 用于将存储在局部存储器 21 中的下载内容同 BD-ROM 中的盘内容一体地处理的假想的文件系统。其中,存储在局部存储器 21 的下载内容包括 SubClip、Clip 信息、播放列表信息。该下载内容中的播放列表信息不论是存在于 BD-ROM 和局部存储器 21 的哪一个中的 Clip 信息都能指定这一点,与 BD-ROM 上的播放列表信息不同。进行该指定时,虚拟文件系统 30 上的播放列表信息不需要用全路径指定 BD-ROM 或局部存储器 21 中的文件。这是因为,BD-ROM 上的文件系统或局部存储器 21 上的文件系统,能够被识别为假想的一个文件系统(虚拟文件系统 30)。因此,播放列表信息中的 Clip_Information_file_name 通过指定与存储了 Clip 信息的文件的文件主体相当的 5 位数值,能够指定虚拟文件系统 30、BD-ROM 上的 AVClip。通过虚拟文件系统 30 读取局部存储器 21 的记录内容,并且和 BD-ROM 的记录内容动态地组合,能够产生各种各样的再现的变化。组合局部存储器 21 和 BD-ROM 而形成的盘内容和 BD-ROM 中的盘内容对等地被处理,所以在本发明中,BD-ROM 中还包含局部存储器 21+BD-ROM 的组合所形成的假想的记录介质。

[0124] 演示引擎 31 执行 AV 再现函数。再现装置的 AV 再现函数指的是从 DVD 播放器、CD 播放器继承的传统的功能群,是再现开始 (Play)、再现停止 (Stop)、暂停 (Pause ON)、暂停的解除 (Pause OFF)、静止图像功能的解除 (Still off)、带速度指定的快进 (ForwardPlay(speed))、带速度指定的快退 (Backward Play(speed))、声音切换 (Audio Change)、副影像切换 (Subtitle Change)、角度切换 (AngleChange) 这样的功能。为了实现 AV 再现函数,演示引擎 31 控制视频解码器 4、演示图形解码器 13、交互图形解码器 10、音频解码器 6,以便进行被读取到读缓冲器 2 上的 AVClip 中的、相当于期望时刻的部分的解码。通过进行作为期望时刻由 PSR8(当前 PTM) 所示的部分的解码,能够进行任意时刻的再现。

[0125] 再现控制引擎 (Playback Control Engine(PEC))32 执行对于播放列表的再现控制函数 (i)、PSR 组 23 中的状态取得 / 设定函数 (ii) 这样的诸功能。对于播放列表的再现控制函数是指,按照当前 PL 信息和 Clip 信息,进行演示引擎 31 执行的 AV 再现函数中的再现开始和再现停止。这些功能 (i)、(ii) 按照来自 HDMV 模块 33 ~ BD-J 平台 35 的函数调用而执行。

[0126] 模块管理器 34 保持从 BD-ROM 读取的 Index.bdmv, 进行分支控制。该分支控制向构成当前标题的动态脚本发行 Terminate 事件, 向构成分支目的地标题的动态脚本发行 Activate 事件。

[0127] 以上说明了演示引擎 31 ~ 模块管理器 34。接着, 说明 BD-J 平台 35。

[0128] BD-J 平台 35 是所谓 Java(注册商标)平台, 是将 Java(注册商标)虚拟机 36 作为核心的结构。BD-J 平台 35 除了安装上述的 Java(注册商标)2Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0) 和 Globally Executable MHP specification(GEM[1.0.2])for package media targets 之外, 还安装了 BD-J Extention。BD-J Extention 包含为了将超越了 GEM[1.0.2] 的功能提供给 BD-J 平台而特化的各种处理包。

[0129] 首先, 说明成为 BD-J 平台 35 的核心的 Java(注册商标)虚拟机 36。

[0130] <Java(注册商标)虚拟机 36>

[0131] 图 4 是表示 Java(注册商标)虚拟机 36 的内部结构的图。如本图所示, Java(注册商标)虚拟机 36 包括图 33 所示的 CPU24 和用户类装载机 52、方法区域 53、工作存储器 54、线程 55a、b、...n、Java(注册商标)堆栈 56a、b...n。

[0132] 用户类装载机 52 从局部存储器 26 等读取 BDJA 目录的 Java(注册商标)档案文件中的类文件, 并存储到方法区域 53。由该用户类装载机 52 进行的类文件读取, 是通过应用管理器 37 向用户类装载机 52 指示指定了全路径的读取而进行的。如果全路径表示局部存储器 26, 则用户类装载机 52 将构成应用的 Java(注册商标)档案文件中的类文件, 从局部存储器 26 读取到工作存储器 54。如果全路径表示 VirtualFile System30 上的目录, 则用户类装载机 52 将构成应用的 Java(注册商标)档案文件中的类文件, 从 BD-ROM 或局部存储器 20 读取到工作存储器 54。应用的起动控制, 通过该用户类装载机 52 的类文件读取实现。在被指示了读取的类文件不存在于局部存储器 26 时, 用户类装载机 52 向应用管理器 37 通知读取失败。

[0133] 方法区域 53 存储由用户类装载机 52 从局部存储器 27 读取的类文件。

[0134] 工作存储器 54 是所谓的堆区域, 存储各种各样的类文件的实例。图 3 所示的应用管理器 37 是常驻在该工作存储器 54 中的常驻应用。在工作存储器 54 中, 除了这些常驻型的实例(instance) 以外, 还存储与被读取到方法区域 53 的类文件对应的实例。该实例是构成应用的 xlet 程序。通过将这样的 xlet 程序配置到工作存储器 54 中, 应用成为可执行状态。

[0135] 在图 3 的层模型中, 将该工作存储器 54 上的应用管理器 37 描绘在 Java(注册商标)虚拟机 36 上, 但这只不过是考虑到易于理解而做出的。现实的记述是, 应用管理器 37 及应用作为实例由线程 55a、b...n 执行。

[0136] 线程 55a、b...n 是执行存储在工作存储器 54 中的方法的逻辑上的执行主体, 将局部变量和存储在操作数堆栈中的变量作为操作数进行运算, 将运算结果存储在局部变量或操作数堆栈中。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 的方法供给。物理的执行主体是 CPU 这唯一的, 与此相比, 相当于逻辑的执行主体的线程能够存在于最大 64 个 Java(注册商标)虚拟机 36 内。在该 64 个这样的数值内, 既可以新制作线程, 也可以删除已存在的线程, 线程的动作数也可以在 Java(注册商标)虚拟机 36 的

动作中增减。由于能够适当增加线程数量,利用多个线程进行一个实例的并列执行,能够实现实例的高速化。本图中,CPU24 和线程的对应关系是 1 对多的关系,但在 CPU 为多个的情况下,CPU24 和线程的对应关系可以是多对多的关系。线程 55a、b...n 的方法执行,是通过将构成方法的字节代码变换为 CPU 的默认代码之后发行给 CPU24 而进行的。

[0137] Java(注册商标)堆栈 56a、b...n 和线程 55a、b...n 按照 1 对 1 的比例存在着,在内部具有程序计数器(图中的 PC)和 1 个以上的帧。“程序计数器”表示在实例中当前正在执行哪个部分。“帧”是向对于方法的一次调用分配的堆栈式的区域,包括存储其 1 次调用时的变量的“操作数堆栈”和被调用的方法所用的“局部变量堆栈(图中的局部变量)”。帧是在每进行一次调用时被堆积在 Java(注册商标)堆栈 56a、b...n 上,因此在某一个方法递归地调用自己的情况下,该帧也能够被堆积一个。

[0138] 以上说明了 Java(注册商标)虚拟机。

[0139] <应用管理器 37>

[0140] 接着,说明应用管理器 37。应用管理器 37 是在 Java(注册商标)虚拟机 36 内的工作存储器上动作的系统软件,每当发生标题分支时,虽然在分支前标题中不执行,但在新的标题中指示 Java(注册商标)虚拟机 36 起动的具有 AutoRun 的起动的属性的应用。与此同时,将在分支前标题中执行、但不把新的标题作为生存区间的应用。这些起动的控制和结束控制是在参照局部 BD-J Object 中的应用管理表的基础上进行的。

[0141] <BD-ROM 的构成>

[0142] 下面,说明 BD-ROM110 中存储的数据。

[0143] 首先,利用图 5 说明 BD-ROM 的数据的文件目录构造。

[0144] 如图 5 所示,在 BD-ROM110 中,在 Root 目录下有 BDM 目录。

[0145] 在 BDM 目录中,有赋予了后缀 bdmv 的文件(index.bdmv,MovieObject.bdmv)。并且,在该 BDM 目录之下,进一步存在称为 PLAYLIST 目录、CLIPINF 目录、STREAM 目录、BDBJ 目录、BDJA 目录、AUXDATA 目录的 6 个子目录。

[0146] 在 PLAYLIST 目录中,有赋予了后缀 mpls 的文件(00001.mpls,00002.mpls,00003.mpls)。

[0147] 在 CLIPINF 目录中,有赋予了后缀 clpi 的文件(00001.clpi,00002.clpi,00003.clpi)。

[0148] 在 STREAM 目录中,有赋予了后缀 m2ts 的文件(00001.m2ts,00002.m2ts,00003.m2ts)。

[0149] 在 BDBJ 目录中,有赋予了后缀 bobj 的文件(00001.bobj,00002.bobj,00003.bobj)。

[0150] 在 BDJA 目录中,有赋予了后缀 jar 的文件(00001.jar,00002.jar,00003.jar)。通过以上的目录构造,理解到在 BD-ROM 上配置有相互不同种类的多个文件。

[0151] 在 AUXDATA 目录中,存储了文件 sound.bdmv。

[0152] <BD-ROM 的构成之一、AVClip>

[0153] 首先,说明赋予了后缀 .m2ts 的文件。图 6 是示意性地表示赋予了后缀 .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...)存储着 AVClip。AVClip 是 MPEG2-TransportStream 格式的数字流。该数字流的构成是,把对

胶片影像、NTSC影像、PAL影像进行数字化而得到的数字视频,和对LPCM(Linear PulseCode Modulation,线性脉冲编码调制)音源、AC-3音源、DTS(DigitalSurround,数字环绕)音源进行数字化而得到的数字音频(上第一层),变换为由PES包构成的基本流(上第二层),进一步变换为TS包(上第三层),并且,同样地将字幕类的演示图形流(PG流)和对话类的交互图形流(IG流)(下第一层),变换为PES包串(下第二层),进一步变换为TS包(下第三层),并将它们复用而构成。

[0154] PG流是实现伴随运动图像再现的字幕显示的基本流,IG流是实现伴随着运动图像再现的GUI的基本流。这些IG流和PG流不是本发明的主要特征,因此省略说明。

[0155] 将视频流中的由1个PTS再现的再现单位(图片等)称为“VideoPresentation Unit”。将音频流中的由1个PTS再现的再现单位称为“Audio Presentation Unit”。

[0156] 在此,构成AVClip的PES包,构成1个以上的“STC_Sequence”。“STC_Sequence”是指,PES包的排列中,在其PTS、DTS参照的SystemTime Clock(STC)的值中不存在STC不连续点(system time-basediscontinuity)的构成。不存在STC不连续点是STC_Sequence的主要条件,所以,在构成一个STC_Sequence的PES包串中,从位于不连续点的正后面且包含了PCR(Program Clock Reference,节目时钟基准)的PES包到下一个STC不连续点之前,成为一个STC_Sequence。

[0157] <Clip信息>

[0158] 接着,说明赋予了后缀.clpi的文件。赋予了后缀.clpi的文件(00001.clpi,00002.clpi,00003.clpi...)存储着Clip信息。Clip信息是关于每个AVClip的管理信息。图7是表示Clip信息的内部结构的图。如本图的左侧所示,Clip信息包括:

[0159] 1) 存储了关于AVClip的信息的“ClipInfo()”,

[0160] 2) 存储了与ATC Sequence,STC Sequence有关的信息的“SequenceInfo()”,

[0161] 3) 存储了与节目时序(Program Sequence)有关的信息的“ProgramInfo()”,

[0162] 4) “Characteristic Point Info(CPI())”。

[0163] Sequence Info是包含在AVClip中的、关于1个以上的STC-Sequence、ATC-Sequence的信息。预先设置这些信息的意义是,为了预先向再现装置通知STC、ATC的不连续点。即,若存在这样的不连续点,则会在AVClip内有可能出现相同值的PTS、PCR,再现时发生不良情况。为了表示在传输流中的从何处到何处STC、ATC是连续的,而设置了Sequence Info。

[0164] Program Info是表示节目内容为固定的区间的信息。节目(Program)是互相共有用于同步再现的时间轴的传输流彼此的集合。预先设置节目时序的意义是,为了预先向再现装置通知节目内容的变化点。这里的节目内容的变化点指的是,视频流的PID变化、或者视频流的种类从SDTV(Standard Definition Tele Vision,标准清晰度数字电视)变化到HDTV(High-Definition Tele Vision,高清晰度数字电视)的点等。

[0165] 接着,说明Characteristic Point Info。图7中的引出线cu2示出了CPI的构成。如引出线cu2所示,CPI包括Ne个的EP_map_for_one_stream_PID(EP_map_for_one_stream_PID(0)~EP_map_for_one_stream_PID(Ne-1))。这些EP_map_for_one_stream_PID是关于属于AVClip的各个基本流的EP_map。EP_map是在一个基本流中,将Access Unit Delimiter所存在的入口位置的包序号(SPN_EP_start)和入口时刻(PTS_EP_start)对应

起来表示的信息。图中的引出线 cu3 示出 EP_map_for_one_stream_PID 的内部结构。

[0166] 从该图可知,EP_map_for_one_stream_PID 包含有 N_c 个的 EP_High (EP_High(0) ~ EP_High(N_c-1)) 和 N_f 个 EP_Low (EP_Low(0) ~ EP_Low(N_f-1))。在此,EP_High 具有表示 Access Unit (Non-IDR I 图片、IDR 图片) 的 SPN_start 及 PTS_EP_start 的高位比特的作用,EP_low 具有表示 Access Unit (Non-IDR I 图片、IDR 图片) 的 SPN_start 及 PTS_EP_start 的低位比特的作用。

[0167] 图 7 中的引出线 cu4 示出 EP_High 的内部结构。如该引出线所示,High(i) 包括作为对 EP_Low 的参考值的“ref_to_EP_Low[i]”、表示 Access Unit (Non-IDR I 图片、IDR 图片) 的 PTS 的高位比特的“PTS_EP_High[i]”和表示 Access Unit (Non-IDR I 图片、IDR 图片) 的 SPN 的高位比特的“SPN_EP_High[i]”。在此,i 是用于识别任意的 EP_High 的识别符。

[0168] 图 7 中的引出线 cu5 示出了 EP_Low 的构成。如引出线 cu5 所示,EP_Low 包括表示对应的 Access Unit 是否是 IDR 图片的“is_angle_change_point (EP_Low_id)”、表示对应的 Access Unit 的大小的“I_end_position_offset (EP_Low_id)”、表示对应的 Access Unit (Non-IDR I 图片、IDR 图片) 的 PTS 的低位比特的“PTS_EP_Low (EP_Low_id)”和表示对应的 Access Unit (Non-IDR I 图片、IDR 图片) 的 SPN 的低位比特的“SPN_EP_Low (EP_Low_id)”。在此,EP_Low_id 是用于识别任意的 EP_Low 的识别符。

[0169] <Clip 信息的说明之二 . EP_map>

[0170] 下面,通过具体例说明 EP_map。图 8 是表示对于电影视频流的 EP_map 设定的图。第一层表示按显示顺序配置的多个图片 (由 MPEG4-AVC 规定的 B 图片、IDR 图片、P 图片),第二层表示该图片中的时间轴。第四层表示 BD-ROM 上的 TS 包串,第三层表示 EP_map 的设置。

[0171] 在第二层的时间轴上,设在时刻 $t_1 \sim t_7$ 存在成为 Access Unit 的 IDR 图片。并且,如果假设这些 $t_1 \sim t_7$ 的时间间隔为 1 秒左右,则用于电影的视频流中的 EP_map 设定成,将 $t_1 \sim t_7$ 作为入口时刻 (PTS_EP_start) 表示,与此建立对应地表示入口位置 (SPN_EP_start)。

[0172] <PlayList 信息>

[0173] 接着,说明 PlayList 信息。赋予了后缀“mpls”的文件 (00001.mpls) 是存储了 PlayList (PL) 信息的文件。PlayList 信息是将称为 MainPath 的再现路径和与此对应的再现信息作为播放列表 (PL) 定义的信息。

[0174] 图 9 是表示 PlayList 信息的数据结构的图,如本图所示,PlayList 信息包括定义 MainPath 的 MainPath 信息 (MainPath()) 和定义章节信息的 PlayListMark 信息 (PlayListMark())。

[0175] MainPath 信息是主要在 AVClip 上定义的再现路径。另一方面,Subpath 是在 SubClip 上定义的再现路径。在此,SubClip 是不包含作为运动图像的视频流的 AVClip。

[0176] <PlayList 信息的说明之一 . MainPath 信息>

[0177] 首先说明 MainPath 信息。MainPath 信息对于作为主影像的视频流和音频流定义的再现路径。

[0178] MainPath 是如箭头 mp1 所示地由多个 PlayItem 信息 #1...#m 定义。PlayItem 信息

定义构成 MainPath 的一个以上的逻辑的再现区间。PlayItem 信息的结构由引出线 hs1 示出。如该引出线所示,PlayItem 信息包括表示再现区间的 IN 点及 Out 点所属的 AVClip 的再现区间信息的文件名“Clip_Information_file_name”、表示 AVClip 的编码方式的“Clip_code_identifier”、表示再现区间的始点的时间信息“IN_time”、表示再现区间的终点的时间信息“OUT_time”和“STN_table”。

[0179] 图 10 是表示 AVClip 和 PlayList 信息之间关系的图。第一层表示 PlayList 信息所具有的时间轴。从第二层到第五层表示通过 EP_map 参考的视频流(与图 6 所示的相同)。

[0180] PlayList 信息包含 PlayItem 信息 #1、#2 这样的 2 个 PlayItem 信息,通过这些 PlayItem 信息 #1、#2 的 In_time、Out_time,定义了 2 个再现区间。若排列这些再现区间,就定义了与 AVClip 时间轴不同的时间轴。这就是示于第一层的 PlayItem 时间轴。如此地,通过 PlayItem 信息的定义,能够定义不同于 AVClip 的时间轴。

[0181] <STN_table>

[0182] STN_table 是表示被复用在由 Play_Item 的 Clip_Information_file_name 指定的 AVClip 中的多个基本流中,能够再现的基本流的表。具体来说,STN_table 是将与多个基本流的每个有关的 entry 和 attribute 对应起来而构成。

[0183] 图 11 是表示 STN_table 的内部结构的图。如该图所示,STN_table 包含多个 STN_table 中的 entry 和 attribute 的组合(entry-attribute),成为表示这些 entry-attribute 组合的个数(number_of_video_stream_entries, number_of_audio_stream_entries, number_of_PG_textST_stream_entries, number_of_IG_stream_entries)的数据结构。

[0184] entry-attribute 的组合如图中的括号“{”所示,分别与 PlayItem 中可再现的视频流、音频流、PG_textST_stream、IG 流相对应。

[0185] 说明 entry-attribute 的详细内容。图 11 是表示 entry-attribute 的详细内容的图。

[0186] 图 12(a) 是表示与视频流对应的 entry-attribute 的组合的图。

[0187] 视频流中的 entry 包含表示在对 AVClip 进行多路分离时用于该视频流的提取的 PID 的“ref_to_stream_PID_mainClip”。

[0188] 视频流中的 attribute 包括被设定为 0x02 的“stream_coding_type”和表示视频流的显示速率的“Frame_rate”等。

[0189] 图 12(b) 是表示与音频流对应的 entry-attribute 的组合的图。

[0190] 音频流中的 entry 包含表示在对 AVClip 进行多路分离时用于该音频流的提取的 PID 的“ref_to_stream_PID_mainClip”。

[0191] 音频流中的 attribute 包括:通过被设定为 0x80(LinearPCM)、0x81(AC-3)、0x82(DTS) 中的某一个,表示音频流的编码类型的“stream_coding_type”、表示对应的音频流声道且表示能否多声道输出的“audio_presentation_type”、表示对应的音频流的语言属性的“audio_language_code”等。

[0192] 多声道中除了包含 5.1CH 的环绕声音以外,还包含立体声,但在下面的说明中,以多声道的含义为只包含 5.1CH 的环绕声音进行说明。

[0193] 图 12(c) 是表示与 PG 流对应的 entry-attribute 的组合的图。

[0194] PG 流中的 entry 包含表示在对 AVClip 进行多路分离时用于该 PG 流的提取的 PID 的“ref_to_stream_PID_of_mainClip”。

[0195] PG 流中的 attribute 包括通过被设定为 0x90 来表示 PG 流的编码译码器的“stream_coding_type”和表示对应的 PG 流的语言属性的“PG_language_code”。

[0196] 图 12(d) 是表示与 IG 流对应的 entry-attribute 的组合的图。

[0197] IG 流中的 entry 包含表示在对 AVClip 进行多路分离时用于该 IG 流提取的 PID 的“ref_stream_PID_of_mainClip”。

[0198] IG 流中的 attribute 包括通过设定为 0x91 表示 IG 流的编码译码器的“stream_coding_type”和表示对应的 IG 流的语言属性的“language_code”。

[0199] <PlayList 信息的说明之二.PlayListMark>

[0200] 以上说明了本实施方式涉及的 PlayItem 信息。接着,说明 PlayListMark 信息。

[0201] 图 13 是表示 PlayList 信息的 PlayListMark 信息的内部结构的图。如该图的图中的引出线 pm0 所示, PlayListMark 信息包括多个 PLMark 信息 (#1 ~ #n)。PLMark 信息 (PLMark()) 是将 PL 时间轴中的任意区间指定为章节节点的信息。如引出线 pm1 所示, PLMark 信息包含表示作为章节指定的对象的 PlayItem 的“ref_to_PlayItem_Id”和利用时间标记表示该 PlayItem 中的章节位置的“mark_time_stamp”。

[0202] 图 14 是表示 PlayList 信息的、利用 PLMark 信息进行的章节位置的指定的图。本图的第二层到第五层表示图 10 所示的 EP_map 和 AVClip。

[0203] 本图的第一层表示 PLMark 信息和 PL 时间轴。在第一层,存在 2 个 PLMark 信息 #1 ~ #2。箭头 kt1、2 表示通过 PLMark 信息的 ref_to_PlayItem_Id 进行的指定。从该箭头可知, PLMark 信息的 ref_to_PlayItem_Id 指定着 PlayItem 信息的每个。此外, Mark_time_Stamp 表示 PlayItem 时间轴中的应成为 Chapter#1、#2 的时刻。如此地, PLMark 信息能够在 PlayItem 时间轴上定义章节节点。

[0204] 能够定义可使 AVClip-SubClip 同步的同步区间,是 BD-RO 中的播放列表信息的特征。以上的 Clip 信息和播放列表信息被分类为“静态脚本”。其理由是,由以上的 Clip 信息和播放列表信息来定义作为静态的再现单位的播放列表。至此,结束关于静态脚本的说明。

[0205] 接着,说明“动态脚本”。动态脚本是动态地规定 AVClip 的再现控制的脚本数据。“动态地”是指再现装置中的状态变化和通过来自用户的键事件来改变再现控制的内容。BD-ROM 中,作为该再现控制的动作环境,假想了 2 个模式。第一个是与 DVD 再现装置的动作环境很相似的动作环境,是基于指令的执行环境。第二个是 Java(注册商标)虚拟机的动作环境。这 2 个动作环境中的第一个,称为 HDMV 模式,第二个称为 BD-J 模式。由于存在这 2 个动作环境,动态脚本是假想了其中一个动作环境而记述的。假想了 HDMV 模式的动态脚本称为 MovieObject。另一方面,假想了 BD-J 模式的动态脚本称为 BD-J Object。

[0206] 首先说明 Movie Object。

[0207] <Movie Object>

[0208] Movie Object 被存储在如图 5 所示的 MovieObject.bdmv 这样的文件中,包含导航指令串。

[0209] 导航指令串包括实现条件分支、再现装置中的状态寄存器的设定、状态寄存器的设定值取得等的指令串。在 MovieObject 中可记述的指令示于下面。

[0210] 1) PlayPL 指令

[0211] 格式 : Play (第一变量, 第二变量)

[0212] 第一变量是播放列表的号码, 能够指定要再现的播放列表。第二变量可以使用该播放列表中包含的 PlayItem 或该播放列表中的任意时刻、chapter、Mark 来指定再现开始位置。

[0213] 将利用 PlayItem 指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatPlayItem(),

[0214] 将利用 Chapter 指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatChapter(),

[0215] 将利用时刻信息指定了 PL 时间轴上的再现开始位置的 PlayPL 函数称为 PlayPLatSpecifiedTime()。

[0216] 2) JMP 指令

[0217] 格式 : JMP 变量

[0218] JMP 指令是废弃 (discard) 当前的动态脚本而执行变量所示的分支目的地动态脚本的分支。JMP 指令的形式, 有直接指定了分支目的地动态脚本的直接参照的形式和间接参照分支目的地动态脚本的间接参照的形式。

[0219] Movie Object 中的导航指令的记述同 DVD 中的导航指令的记述方式很相似, 所以, 能够有效地进行将 DVD 上的盘内容移植到 BD-ROM 的作业。关于 MovieObject, 存在被记载于以下的国际公开公报中的现有技术。关于详细内容, 请参考该国际公开公报。

[0220] 国际公开公报 WO 2004/074976

[0221] 至此, 结束关于 Movie Object 的说明。接着, 说明 BD-J Object。

[0222] <BD-J Object>

[0223] BD-J Object 是在 Java(注册商标) 编程环境下记述的 BD-J 模式的动态脚本, 被存储在 00001 ~ 00003.bobj 这样的文件中。

[0224] 图 15 是表示 BD-J Object 的内部结构的图。包含应用管理表 (AMT)、播放列表管理表 (PLMT) 而构成。与 Movie Object 的不同点在于, 在 BD-J Object 中未直接记述指令。即, 在 Object 中的控制步骤是利用导航指令直接记述的。与此相比, 在 BD-J Object 中, 通过应用管理表中记载对于 Java(注册商标) 应用的指定, 间接地规定了控制步骤。通过这样的间接的规定, 能够有效地及进行在多个动态脚本中使控制步骤共同化的所谓控制步骤共同化。

[0225] 此外, MovieObject 中的播放列表再现是通过命令播放列表再现的导航指令 (PlayPl 指令) 的记述而实现的, 但是, BD-J Object 中的播放列表再现, 是通过将表示播放列表再现顺序的播放列表管理表组装到 BD-J Object 中而记述可能。

[0226] 说明该 BD-J 模式中的 Java(注册商标) 应用。在此, 假想了 BD-J 模式的 Java(注册商标) 平台, 完全安装了 Java(注册商标) 2Micro_Edition(J2ME) Personal Basis Profile(PBP 1.0) 和 GloballyExecutable MHP specification(GEM 1.0.2) for package mediatargets。

[0227] 该BD-J模式下的Java(注册商标)应用是通过xlet接口由应用管理器控制。Xlet接口具有“loaded”、“paused”、“active”、“destroyed”的四个状态。

[0228] 上述的Java(注册商标)平台包含用于显示JFIF(JPEG)和PNG、及其他图形数据的标准(注册商标)程序库。因此,Java(注册商标)应用能够实现与在HDMV模式下通过IG流实现的GUI不同的GUI框架。Java(注册商标)应用中的GUI框架,包含由GEM1.0.2规定的HAVi(HomeAudio/Video interoperability)框架,包含GEM1.0.2中的遥控机构。

[0229] 由此,Java(注册商标)应用能够实现将基于HAVi框架的按钮显示、文本显示、联机显示(BBS的内容)这样的显示和运动图像的显示进行了组合的画面显示,利用遥控能够进行对该画面显示的操作。

[0230] 相当于该Java(注册商标)应用的实体的是,图2中的BDMV目录下的BDJA目录中存储的Java(注册商标)档案文件(00001.jar,00002.jar)。下面,参照图16说明(注册商标)档案文件。

[0231] <Java(注册商标)档案文件>

[0232] Java(注册商标)档案文件(图2的00001.jar,00002.jar)是将1个以上的类文件、1个以上的数据文件等归纳为1个而得到的文件,构成在BD-J模式下应动作的Java(注册商标)应用。

[0233] 图16是表示由档案文件收容的程序、数据的图。该图中的程序、数据是用Java(注册商标)档案库归纳了配置有框内表示的目录构造的多个文件。框内表示的目录构造包括Root目录、Java(注册商标)1、2、3目录、Image1、2、3目录,在Root目录中配置有common.pkg,在Java(注册商标)1、Java(注册商标)2、Java(注册商标)3目录中配置有类文件(00001.class~00007.class),在Image1、Image2、Image3目录中配置有00001.JPEG~00003.JPEG、00004.PNG~00006.PNG。Java(注册商标)档案文件是用Java(注册商标)档案库归纳了这些文件而得到。这样的类文件及数据在从BD-ROM读取到高速缓存时被解压缩,在高速缓存上作为被配置在目录中的多个文件来处理。Java(注册商标)档案文件的文件名中“zzzzz”这5位数值,应用的ID(applicationID)。本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,能够取出构成任意的Java(注册商标)应用的程序、数据。

[0234] 而且,在本实施方式中构成应用的程序、数据被归纳在Java(注册商标)档案文件中,但也可以是LZH文件、zip文件。

[0235] 以上,说明了BD-J模式中的动态脚本。

[0236] <Index.bdmv>

[0237] Index.bdmv是表示构成标题的Movie Object或BD-J Object的表。

[0238] 在标题中,定义称为某一标题的构成要素的MovieObject是哪个、或者成为某一标题的构成要素的BD-J Object是哪个。

[0239] 关于Index.bdmv的详细内容记载在以下的国际公开公报中。关于详细内容请参考该公报。

[0240] 国际公开公报WO 2004/025651 A1公报

[0241] 下面,进一步详细地分别说明图15所示的应用管理表、播放列表管理表。

[0242] <应用管理表>

[0243] 说明应用管理表 (AMT)。应用管理表 (AMT) 是安装上述的 GEM1.0.2for package media targets 中的“应用 信令”的表。“应用信令”是指,在 GEM1.0.2 规定的 MHP (Multimedia Home Platform) 中,将“服务”作为生存期间进行应用的起动的控制。本实施方式中的应用管理表,代替该“服务”,而是将 BD-ROM 中的“标题”作为生存区间,来实现应用的起动的控制。

[0244] 图 17(a) 是表示应用管理表的内部结构的图。如该图所示,应用管理表包括“life_cycle”、“apli_id_ref”、“run_attribute”和“run_priority”。

[0245] 图 17(b) 表示构成应用管理表的信息要素的含义内容。

[0246] “life_cycle”表示应用的“生存区间”。

[0247] “apli_id_ref”表示通过记述对于“应用识别符”的参照值而具有左侧记载的生存区间的应用是哪一个。应用识别符在 Java(注册商标)档案文件中用作为文件名赋予的 5 位数值 zzzzz 表现。“apli_id_ref”中记载了该 5 位数值。

[0248] “run_attribute”记述了该生存区间中的应用的“起动的属性”。起动的属性有 AutoRun、Present、Susped 这样的种类。

[0249] “run_priority”记述了该生存区间中的应用的“起动的优先级”。在 BD_J Object 中,使用这些信息控制应用的动作。

[0250] <生存区间>

[0251] 在此,说明在应用管理表中规定的信息中的生存区间。

[0252] 生存区间表示,在记录在 BD-ROM 中的内容全体的时间轴上,应用能够在虚拟机的工作存储器上能够生存的区间。工作寄存器中的“生存”是说,构成该应用的 xlet 程序被读取到 Java(注册商标)虚拟机内的工作存储器中、且能够由 Java(注册商标)虚拟机执行的状态。

[0253] 在 Java(注册商标)虚拟机中使应用动作的情况下,明确地规定从时间轴的何处开始应用的服务、并在时间轴的何处结束应用的服务这样的“服务的开始点及结束点”是很重要的。规定该服务的开始点及结束点的是应用管理表中的生存区间。

[0254] 另一方面,用 DVD-Video 那样的读取专用盘供给的盘内容,成为将顶级菜单标题作为核心的构造。从该顶级菜单标题向各个著作物分支后进行再现,然后再返回到顶级菜单标题,具有这样独特的状态转移。图 18 是表示盘内容中的状态转移的图。该图中的四边框是标题 (Title)。标题是指在盘内容特有的状态转移中相当于 1 个“状态”的再现单位,该标题作为 Java(注册商标)应用的生存区间而处理。

[0255] 标题中,有在 BD-ROM 的装载时首先再现的“FirstPlayTitle”、构成顶级菜单的“Top_menuTitle”、除此以外的一般的“Title”。此外,图中的箭头 jh1、2、3、4、5、6、7、8 象征地表示 Title 间的分支。该图所示的状态转移是,在 BD-ROM 装载时再现“FirstPlayTitle”,并发生向“Top_menuTitle”的分支,成为对顶级菜单选择的等待。

[0256] 如果由用户进行了对顶级菜单的选择操作,则根据选择进行该 Title 的再现,并再次返回到 TopMenu Title,在 BD-ROM 的卸载之前一直重复进行上述的处理,就是盘内容特有的状态转移。

[0257] 那么,在形成图 18 所示的状态转移的盘内容中,Title 是怎样作为生存区间被规定的。在进行 BD-ROM 的装载之后,按照在图 18 中用箭头 jh1、2、3、4...表示的参考标记的

数值顺序进行分支, BD-ROM 被卸载。如果这样, 可以将从 BD-ROM 被装载之后到卸载的连续时间带同视为一根时间轴。将该时间轴作为盘整体的时间轴。图 19(a) 是表示盘整体的时间轴的图, 图 19(b) 表示该时间轴上的构成。如图 19(b) 所示, 盘整体的时间轴包括再现 FirstPlayTitle 的区间、再现 Top_menuTitle 的区间、再现 title#1 的区间等。这些 Title 的再现区间是这样规定的, 即 Title 是由唯一的 BD-J Object 构成的, 因此可以将任一个 MovieObject 或 BD-J Object 为有效的期间作为 Title 的再现区间。

[0258] 即, FirstPlay Title、TopMenu Title、其他的 Title 都由动态脚本构成, 所以, 可以将构成 Title 的 BD-J Object 中的某一个作为当前 BD-J Object 被激活、并在再现装置内供于解读及执行的期间定义为 Title 的再现区间。图 20(a) 时表示在 BD-ROM 整体的时间轴上, 从由识别符 bobj_id 确定的 BD-J Object 中确定的再现区间的图。在此, 如果由识别符 bobj_id 确定的 BD-J Object 构成 1 个 Title, 则可以将由该识别符 bobj_id 确定的 BD-J Object 为有效的 BD-ROM 时间轴上的一区间考虑为 Title 的再现区间。

[0259] 在此, BD-J Object 被激活的期间的结束时间, 一直到发生 Title 分支为止。即, 在发生 Title 分支之前, 成为执行对象的动态脚本作为当前 BD-J Object 被处理, 所以将在该 BD-J Object 中发生 JumpTitle 之前的一个区间作为 Title 区间来处理。

[0260] 接着, 说明 Title 区间和 PL 时间轴的关系。如上所述, 在 MovieObject、BD-J Object 中, 可以将播放列表再现步骤作为一个处理步骤记述。如果有播放列表再现步骤的记述, 则上述的 PL 时间轴的全部或部分归属于 Title 区间。在图 20(a) 的一例中, 假设在 BD-J Object 记述了播放列表管理表。该情况下, 如图 20(b) 所示, PL 时间轴归属在与 BD-J Object 对应的 Title 区间中。在该 PL 时间轴上还可以定义多个章节 (Chapter#1、#2、#3), 因此在 BD-ROM 上的时间轴上存在 BD-ROM 全体 -Title- 播放列表章节这样的函数。使用这些函数, 能够记述应用的生存区间。而且, 播放列表再现和应用执行同时进行, 所以在播放列表再现的途中有时会发生 Title 分支。该情况下, 在一个 Title 再现区间内只归属播放列表时间轴的一部分, 而不是播放列表时间轴全体。即, 在一个 Title 的再现区间中是归属播放列表时间轴的全部、还是归属其一部分, 是根据 Title 分支何时发生而变更的。

[0261] 图 21 是表示在图 20(b) 的时间轴上被规定的生存区间的典型例子的图。如该图所示, 在应用中, 存在将 Title 作为生存区间的“标题边界应用”、将 Title 内的章节作为生存区间的“章节边界应用”和将 BD-ROM 整体的时间轴作为生存区间的“标题无边界应用”这三个典型类型。

[0262] 其中, 标题边界应用的生存空间可以用该标题的识别符定义。此外, 章节边界应用的生存空间可以用章节所属的标题的识别符和该章节识别符的组合进行定义。

[0263] 即使平台在工作, 若 Title 或章节这样的生存区间结束, 则能够从应用回收资源。由于保证资源回收的机会, 能够使平台的动作稳定化。

[0264] 举具体例子说明在不远的将来会实施的将盘内容选作题材而进行的应用管理表中的生存区间记述。在此, 作为题材的盘内容包含构成影像正编的正编标题 (title#1)、构成网络购物的网络购物标题 (title#2)、构成游戏应用的游戏标题 (title#3) 这样的性格不同的三个标题。图 22 是表示包含正编标题、网络购物标题、游戏标题这三个标题的盘内容的图。在该图的右侧记述着 Index. bdmv, 在左侧记述着 3 个标题。

[0265] 右侧的虚线框表示各个应用属于哪个标题的归属关系。三个标题中的 title#1

包括 application#1、application#2、application#3 这三个应用。title#2 包括 application#3、application#4 这 2 个应用，title#3 包括 application#5。图 22 的一例中，application#3 在 title#1 和 title#2 的双方被起动的。

[0266] 如果根据图 22 的虚线所示的归属关系将各应用的生存区间图形化，则得到图 23(a)。该图中，横轴是标题再现区间，在纵轴方向配置了各个应用的生存区间。在此，由于 application#1、application#2 仅归属于 title#1，所以这些生存区间留在 title#1 内。由于 application#4 仅归属于 title#2，所以这些生存区间留在 title#2 内。由于 application#5 仅归属于 title#3，所以这些生存区间留在 title#3 内。由于 application#3 归属于 title#1 及 title#2，所以这些生存区间横跨 title#1-title#2。若根据该生存区间记述应用管理表，则 title#1、#2、#3 的应用管理表成为图 23(b)。如果如此地记述了应用管理表，在 title#1 的再现开始时将 application#1、application#2、application#3 装载到工作存储器中。然后，进行在 title#2 的再现开始时将 application#1、application#2 从工作存储器中删除而仅留下 application#3 的控制。与此同样地，能够进行在 title#2 的再现开始时将 application#4 装载到工作存储器中，在 title#3 的开始时将 application#3、application#4 从工作存储器中删除的控制。

[0267] 再者，能够进行在 title#3 的再现中将 application#5 装载到工作存储器中，在 title#3 的再现结束时将 application#5 从工作存储器中删除的控制。

[0268] 即使是发生了标题间分支的情况下，在分支源 - 分支目的地中生存的应用预先存储在工作寄存器上，将不存在于分支源而仅存在于分支目的地的应用读入到工作寄存器中即可，所以将应用读入工作寄存器的次数成为所需要的最低次数。如此地，通过减少读入次数，能够实现不用意识标题的边界的应用、即无边界的应用。

[0269] 接着，进一步详细说明应用的起动的属性。起动的属性中，有表示自动起动的“AutoRun”，表示不是自动起动的对象但可以放置在虚拟机的工作存储器中的“Present”，位于虚拟机的工作存储器中但不能分配 CPU 功率的“Suspend”。

[0270] “AutoRun”是表示与对应的标题的分支同时地将该应用读入到工作存储器且执行的属性。如果发生从某一标题向其他标题的分支，则进行应用管理的管理主体（应用管理器）在其分支目的地中生存着，并且将起动的属性被设定为 AutoRun 的应用读入到工作存储器执行。由此，该应用同标题分支一起自动地被起动的。

[0271] 起动的属性“Present”是继续属性，表示继续分支源 title 中的应用的状态。此外，是表示可以执行对应的应用的属性。在起动的属性为“Present”的情况下，被赋予了该起动的属性的应用被允许从其他应用的调用。如果从起动的应用发生了调用，则进行应用管理的管理主体（应用管理器）判断该应用的 applicationID 是否被记述在应用管理表中，起动的属性是否是“Present”。如果是“Present”，将该应用装载到工作存储器。另一方面，在该调用方应用的 applicationID 未记述在应用管理表的情况下，该应用不被装载到工作存储器。通过应用进行的调用，限于被赋予了该“Present”的应用。“Present”是在未明确地指定起动的属性的情况下赋予的缺省的起动的属性，所以，当某一应用的起动的属性是无指定“——”的情况下，意味着该应用的起动的属性是该“Present”。

[0272] “Suspend”是指，应用被放置在虽然分配了资源但未分配 CPU 功率的状态。这样的 Suspend，例如实现在游戏标题的执行中经由侧路径的处理时是有意义的。

[0273] 图 24 是表示起动属性可取的三种状态 (Present、AutoRun、Suspend) 和之前标题中的应用状态的三种状态 (非起动、起动中、Suspend) 能够取的组合方式的图。在之前状态是“非起动”的情况下,若起动属性是“AutoRun”,则在分支目的地标题中该应用被起动。

[0274] 若之前状态是“非起动”、起动属性是“Present”、“Suspend”,在分支目的地标题中该应用什么也不做而继续其状态。

[0275] 在之前状态是“起动中”的情况下,若起动属性是“Present”、“AutoRun”,在分支目的地标题中该应用什么也不做而继续其状态。

[0276] 如果起动属性是“Suspend”,应用的状态被暂停。在之前状态是“Suspend”的情况下,若分支目的地标题的起动属性是“Suspend”,则维持暂停。若是“Present”或“AutoRun”,则在分支目的地标题中该应用重启。通过在应用管理表中定义生存区间及起动属性,可以沿着标题再现区间的行进,进行使 Java(注册商标)应用动作的同步控制,可以向人们提供伴随影像再现和程序执行的各种各样的应用。

[0277] 而且,在之前状态是“Suspend”、分支目的地标题的起动属性为“Present”的情况下,也可以维持之前状态即暂停状态。

[0278] 最后,说明对于各应用的“起动优先级”。

[0279] 该起动优先级取 0 ~ 255 的值,是应用管理器进行如下处理时使用的判断材料:在存储器资源枯竭时、或者 CPU 负载变高时,强制地结束哪个应用,或者从哪个应用夺取资源。该情况下,应用管理器进行结束起动优先级低的应用的动作,继续起动优先级高的应用的动作的处理。

[0280] 此外,起动优先级还用于对再现中播放列表的请求发生了竞争时的应用间的调解。在此,假设某一个应用正在进行某一播放列表的块进。这时,如果别的应用对相同的播放列表发出了暂停请求,则比较向这些应用赋予的起动优先级。然后,如果命令块进的应用的起动优先级高,则继续进行这样的应用进行的块进。相反,如果命令了暂停的应用的起动优先级高,则进行块进中播放列表的暂停。

[0281] 利用以上的生存区间、起动属性、起动优先级,在编制时能够预先进行规定,将能够在虚拟机上动作的应用的数量限制在规定数以下。因此,能够保证应用的稳定动作。

[0282] < 播放列表管理表 >

[0283] 以上说明了应用管理表。接着,说明播放列表管理表 (PLMT)。是表示在应用的生存区间应该和各应用执行同时进行的再现控制的表。应用的动作是不稳定的,会发生起动的失败和异常结束,在此,作为发生了起动失败、异常结束时的故障保护机构,本实施方式在应用的每个生存区间设置了播放列表管理表。播放列表管理表是规定在某一应用的生存区间开始时应该与此同时进行的再现控制的信息。该再现控制是基于播放列表信息的 AVClip 再现,通过与基于播放列表信息的再现控制同时进行,同时进行应用执行和播放列表再现。播放列表管理表虽然在应用的每个生存区间设置,但设有播放列表管理表的应用仅限于标题边界的应用。其原因是,标题无边界应用将整个标题作为生存区间,所以在应用执行的同时进行播放列表再现的控制,是不合适的。

[0284] 章节边界应用是在从一个播放列表内的章节起开始执行应用的前提下规定了生存区间,所以不需要规定播放列表再现。从上述内容可知,播放列表管理表被定义为由 1 个以上的标题构成的生存区间。

[0285] 图 25(a) 是表示播放列表管理表的内部结构的图。如该图所示,播放列表管理表包括“PL_id_ref”和“Playback_attribute”。

[0286] 图 25(b) 表示构成播放列表管理表的信息要素的含义内容。

[0287] “PL_id_ref”通过记述对于播放列表识别符的“参考值”,表示在应用的生存区间可以再现的播放列表是哪一个。播放列表识别符用文件 YYYYY.MPLS 中的、作为文件名赋予的 5 位数值 YYYYY 表现。通过记述该 YYYYY,“PL_id_ref”表示在对应的标题中可以再现的播放列表是哪一个。

[0288] “Playback_Attribute”是模仿了应用管理表中的起动属性的属性,是规定在标题开始时如何再现被记述在“PL_id_ref”的播放列表的再现属性。对于播放列表的再现属性,有“AutoPlay”和“Present”的 2 种。

[0289] “AutoPlay”是表示与对应的标题的分支同时地再现该播放列表的含义的属性。当发生从某一标题向其他标题的分支时,进行应用管理的管理主体(应用管理器),开始进行在其分支目的地标题中可以再现、且再现属性设定为 AutoPlay 的播放列表的再现。由此,起动属性被设定为 AutoPlay 的播放列表同标题分支一起自动地被起动。

[0290] “Present”和起动属性中的 Present 一样是继续属性,表示继续分支源标题中的播放列表的状态。而且,是表示可以再现对应的播放列表的属性。例如有被连续再现的 2 个标题,在前面的标题侧的播放列表管理表中,某一播放列表的再现属性被设定为 AutoPlay,在当前标题侧的播放列表管理表中,该播放列表的再现属性被设定为 Present。在此,播放列表的再现时间是 2 小时,假设在其中经过了 1 小时的时刻发生了分支。该情况下,在当前标题中再现属性被设定为 Present,所以在当前标题中,该播放列表从 1 个小时的已再现区间之后开始进行再现。若如此地将再现属性设定为 Present,在发生标题间的分支的情况下,也能从其剩余部分开始进行播放列表再现。由此,在相互分支的一连串的标题中,能够容易地实现再现共通的播放列表的所谓“标题间的播放列表再现的共通化”。此外,在分支目的地标题为多个的情况下,若将这些多个标题的再现属性都设为 Present,则不论分支到多个标题中的哪个,都能够继续进行一个共通的播放列表再现。

[0291] 而且,标题的边界也可以不保证无间隙再现,所以,如上所述地在多个标题间要再现 1 个播放列表的情况下,允许在分支前后中断播放列表再现。

[0292] 此外,再现属性是“Present”的情况下,被赋予了该再现属性的播放列表通过来自其他应用的再现请求进行再现。当从起动中的应用发出了播放列表的再现请求,则进行应用管理的管理主体(应用管理器)判断接受了请求的播放列表的 PL_id_ref 是否记述在播放列表管理表中,判断再现属性是不是“AutoPlay”和“Present”中的一个。如果是“AutoPlay”和“Present”中的某一个,则再现该播放列表。另一方面,接受了请求的播放列表的 PL_id_ref 没有被记述在播放列表管理表中的情况下,不再现该播放列表。基于应用的请求进行的播放列表再现,限于被赋予了该“AutoPlay”和“Present”中的某一个的播放列表。由于“Present”是在没有明确地指定再现属性时被赋予的缺省的再现属性,因此,如果某一播放列表的再现属性是“——”,则意味着该播放列表的再现属性是该 Present。

[0293] 图 26 表示由播放列表管理表、应用管理表规定的标题的具体例子。图 26 的第一层表示标题的再现影像,第二层表示时间轴,第三层表示由 PLMT 规定了再现的播放列表,第四层表示应用执行。第四层中, application#1 在标题的开始时同时被起动,然后,在时

刻 t1 成为动作状态。另一方面, PlayList#1 在标题的开始时同时开始再现。PlayList#1 的再现在与标题的再现相同的时刻开始, 所以如第一层的左侧所示, 在标题的再现开始之后起到应用成为动作状态为止的起动的延迟中, 播放列表的再现图向 gj1 全屏显示。通过将播放列表管理表的再现属性设定为“AutoPlay”, 即使在 Java(注册商标)应用成为动作状态之前需要 5 ~ 10 秒的时间, 该期间也处在“姑且显示某一画面的状态”。利用该“姑且显示某一画面的状态”, 能够补偿标题执行开始时的起动的延迟。

[0294] 另一方面, application#1 在时刻 t1 成为动作状态, 所以在时刻 t1 显示将播放列表再现图像作为子画面、将应用的执行图像作为母画面的合成图像 gj2。应用的执行图像是配置了开始按钮、继续按钮、电源指示器的游戏用 GUI 框架, Java(注册商标)应用执行这样的 GUI 框架的描绘处理。

[0295] 如此地, 能够构成组合了播放列表的再现影像和 Java(注册商标)应用的 GUI 框架的、形成再现影像的标题, 这是 PLMT 的特征。

[0296] 图 27 是表示当前标题可取的三种状态(无播放列表管理表(i), 有播放列表管理表且 AutoPlay(ii), 有播放列表管理表且无指定(iii))和之前标题中的播放列表的状态(非再现状态, 再现状态)能够组合的 6 种组合的图。

[0297] 在该图中的 6 种组合中, “之前状态=非再现状态”和“当前标题=有播放列表管理表, 且当前标题的再现属性=AutoPlay”的组合中, 自动地开始分支目的地标题中的播放列表的再现。

[0298] 此外, 在“之前状态=再现中状态”和“当前标题=无播放列表管理表”的组合中, 自动地停止分支目的地标题中的播放列表的再现。

[0299] 并且, 在除了这 2 个组合之外的其他所有组合中, 继续前面的标题的状态。基于播放列表管理表的播放列表再现的开始, 在分支源标题中是非显示状态, 限于在分支目的地标题中被赋予了 AutoPlay 属性的情况, 所以, 不需要在每次发生标题的分支时开始播放列表再现。即使发生了多次标题间的分支, 也能够将开始播放列表再现的次数设为所需的最低次数。

[0300] 参考图 28(a) 说明播放列表管理表和应用管理表的记述例。在此假想的具体例是, 2 个连续的 Title(title#1, title#2), 在其中的 title#1 中, 作为 AutoRun 应用记述了 application#1、application#2。在 title#2 中, 作为 AutoRun 应用记述了 application#2、application#3。另一方面, 在 title#1 的播放列表管理表中, 作为 AutoPlay 播放列表记述了 PlayList#1, 在 title#2 的播放列表管理表中, 作为 AutoPlay 播放列表记述了 PlayList#2。图 28(b) 是表示利用如图 28(a) 那样记述的应用管理表、播放列表管理表, 如何进行播放列表再现及应用执行的图。

[0301] 在 title#1 中如上所述地设定了应用管理表、播放列表管理表, 所以在 title#1 的开始时 application#1、application#2 自动被起动的, 自动地开始 PlayList#1 的再现。

[0302] 在 title#2 中如上所述地设定了应用管理表、播放列表管理表, 所以, 虽然在 title#1 侧有记载但在 title#2 侧无记载的 application#1 的执行停止。同样地, 虽然在 title#1 侧有记载但在 title#2 侧无记载的 PlayList#1 的再现也停止。

[0303] 虽然在 title#1 侧有记载但在 title#2 侧无记载的 PlayList#2、application#3 自动地开始再现及执行。如果有分支, 以该分支为契机, 能够将应再现的播放列表切换为其

他的播放列表。通过如此地使用应用管理表、播放管理表,并以分支为契机,能够在编制阶段预先规定切换播放列表再现的处理。

[0304] 此外,在图 28 中,对 application#1、application#2、application#3 分别赋予 200、128、200 的起动优先级。通过赋予这些起动优先级,能够进行对于 PlayList#1、PlayList#2 的控制请求产生了竞争时的调解。在此,假设 application#1 对 PlayList#1 命令了快进。另一方面,设 application#2 进行了暂停请求。该情况下,由于在应用管理表中规定了对于各应用的起动优先级,所以按照该起动优先级进行对两用户的起动优先级。其结果,能够在编制时预先规定拒绝 application#2 的请求而继续 application#1 的控制的处理。通过同时利用起动优先级和播放列表管理表,可以使再现装置还能够进行对于播放列表的控制发生了竞争时的调解。

[0305] 说明播放列表管理表记述的其他具体例。图 29(a) 是表示播放列表管理表的其他记述例的图。在该图中假定的是,在 2 个连续的标题 (title#1, title#2) 中,在 title#1 侧的播放列表管理表中,作为 AutoPlay 播放列表记述了 PlayList#1,作为可再现的播放列表记述了 PlayList#2,在 title#1 的应用管理表中,记述了作为 AutoPlay 应用的 application#1 和作为可执行的应用的 application#2。另一方面,在 title#2 侧的播放列表管理表中,作为可再现的播放列表记述了 PlayList#2、PlayList#3,在应用管理表中,作为 AutoRun 应用记述了 application#3。图 29(b) 是表示基于图 29(a) 的实例的应用执行和播放列表再现的进行过程的图。在 title#1 的应用管理表中,作为 AutoRun 应用记述了 application#1,所以在 title#1 的开始时, application#1 自动地起动。另一方面,在 title#1 的应用管理表中,作为可执行的应用记述了 application#2,因此,通过来自 application#1 的调用 yd1, application#2 被起动。

[0306] 在 title#2 侧的应用管理表中, application#1、application#2 处于非生存,代替它,作为 AutoRun 应用记述了 application#3。因此,在 title#1-title#2 的边界部,进行停止 application#1、application#2,自动地起动 application#3 的处理。如果参照播放列表管理表, title#1 侧的播放列表管理表被记述成 PlayList#1、PlayList#2 可以再现,其中 PlayList#1 成为 AutoRun 属性。因此, PlayList#1 在 title#1 的开始时自动地进行再现。

[0307] 在 Title#1 侧的播放列表管理表中,除了 PlayList#1 以外, PlayList#2 被记述成可以再现,所以 application#1 通过停止 PlayList#1 的再现、代之请求 PlayList#2 的再现,能够执行播放列表的交替。

[0308] 在 title#2 侧的播放列表管理表中,作为可再现的播放列表记述了 PlayList#2、PlayList#3。并且,不存在被赋予 AutoPlay 属性的播放列表。因此,即使假设在 title#1 开始时自动再现的 PlayList#1 的再现继续到 title#2, PlayList#1 的再现也自动结束。

[0309] 但是,如果在 PlayList#2 的再现继续的状态下到达 title#2, PlayList#2 的再现在 title#2 开始以后也继续。在 title#2 的播放列表管理表中,作为可再现的播放列表记述了 PlayList#2、PlayList#3。因此,在 title#2 成为执行中的 application#3,通过停止 PlayList#2 的再现、代之请求 PlayList#3 的再现,能够交替再现中的播放列表。

[0310] 如上所述,如果将播放列表管理表的再现属性设定为“AutoPlay”,即使 Java(注册商标)应用的起动需要 5~10 秒钟的时间,在执行该起动的期间,也成为“姑且显示某个图像的状态”。在标题执行开始时,即使应用起动需要时间,画面也成为“姑且显示某个图像

的状态”。由此,能够补偿应用起动需要时间所造成的起动延迟的长时间化。

[0311] 通过定义应用管理表和播放列表管理表,能够沿着标题再现区间的进行,实现使 Java(注册商标)应用动作的同步控制,能够向人们提供伴随影像再现和程序执行的各种各样的应用。

[0312] <动作>

[0313] 下面,利用图 30 所示的流程图,说明本实施方式的再现装置 10 的、装载 BD-ROM110 再现运动图像或者显示菜单之前的动作。该动作还是通过应用管理器 37 进行的处理。

[0314] 如图 30 所示,首先,判断是否进行了 Title Jump(步骤 S1)。

[0315] 在进行了 Title Jump 的情况下(步骤 S1 的“是”),执行标题切换(步骤 S7),接着,查看在对应于当前标题的 BD-J Object 是否存在 PLMT(步骤 S8)。

[0316] 当存在 PLMT 的情况下(步骤 S8 的“是”),开始在前面的标题中未记载、但在当前标题中被记载在 PLMT 且赋予了 AutoPlay 属性的播放列表的再现。在不存在 PLMT 的情况下(步骤 S8 的“否”),停止在前面的标题中被记载在 PLMT、但在当前标题中未记载于 PLMT 且被赋予了 AutoPlay 属性的播放列表的再现。

[0317] 接着,判断是否有对应于当前标题的应用管理表(步骤 S11)。

[0318] 在有应用管理表的情况下(步骤 S11 的“是”),起动未将前面的标题作为生存区间但将当前标题作为生存区间的 Java(注册商标)应用中的、被赋予了 AutoRun 属性的应用。在没有应用管理表的情况下(步骤 S11 的“否”),结束将前面的标题作为生存区间但未将当前标题作为生存区间的应用。

[0319] 接着,判断应用的起动是否成功(步骤 S14)。当应用的起动成功的情况下(步骤 S14 的“是”),对视频平面 5 及视频平面控制部 17 执行被赋予了 AutoRun 属性的播放列表的再现图像的四分之一化的指示。此外,接受了该指示的静止图像平面控制部 17,执行来自静止图像平面 16 的静止图像数据的输出(步骤 S15)。然后,在电视机 130 上显示还合成了静止图像平面 16 的静止图像数据的图像。在应用的起动失败的情况下(步骤 S14 的“否”),转移到步骤 S23,执行以后的处理。

[0320] 在没发生 Title Jump 的情况下(步骤 S1 的“否”),检测主应用是否结束(步骤 S2)。

[0321] 在主应用已结束的情况下(步骤 S2 的“是”),判断该应用是否正常地结束(步骤 S5)。在正常地结束的情况下(步骤 S5 的“是”),返回到步骤 S1,执行以后的处理。

[0322] 在主应用异常结束的情况下(步骤 S5 的“否”),查看是不是 AutoPlay PL 的再现中(步骤 S21)。是再现中的情况下(步骤 S21 的“是”),CPU25 将视频平面 5 的数据以全屏大小输出,以便将 Auto Play PL 的再现图像全屏化。而且,对静止图像平面控制部 17 也发出同样的指示,接受了该指示的 Stii 平面控制部 17 制止存储在静止图像平面 16 中的静止图像数据的输出(步骤 S22)。然后,在电视机 130 上显示未合成静止图像数据的合成图像。

[0323] 然后,判断再起动计数器是否为 0(步骤 S23)。在此,再起动计数器是用于规定应用的再起动次数的控制变量。在再起动计数器是 0 的情况下(步骤 S23 的“是”),返回到步骤 S1,执行以后的处理。在再起动计数器不是 0 的情况下(步骤 S23 的“否”),使再起动计数器递减,转移到步骤 S12 并执行以后的处理。通过执行步骤 S23、S24 的过程,保证应用

的起动。并且,再起动计数器在该流程的起动时被复位。

[0324] 在主应用没结束的情况下(步骤 S2 的“否”),接着判断被赋予了 AutoPlay 属性的播放列表的再现是否结束(步骤 S3)。在被赋予了 AutoPlay 属性的播放列表的再现结束的情况下(步骤 S3 的“是”),返回到步骤 S1 执行以后的处理。在被赋予了 AutoPlay 属性的播放列表的再现没结束的情况下(步骤 S3 的“否”),检测 BD 驱动器 1 中是否有 BD-ROM(步骤 S4)。

[0325] 在 BD 驱动器 1 中仍存在 BD-ROM 的情况下(步骤 S4 的“是”),返回到步骤 S1 执行以后的处理。在 BD 驱动器 1 中不存在 BD-ROM 的情况下(步骤 S4 的“否”),执行全部应用的结束指示(步骤 S6),并结束。

[0326] 以上是从 BD-ROM110 安装到再现装置 100 起到取出为止的动作。如上所述,当再现图像全屏显示时和四分之一屏显示时,决定来自静止图像平面 16 的数据输出的可否。在全屏的情况下,不需要合成静止图像平面 16 的数据,所以不输出,这样做,能够增大静止图像平面 16 的读取中使用的存储器总线带宽。

[0327] <第二实施方式>

[0328] 下面,参照附图说明本发明涉及的第二实施方式。

[0329] 在第二实施方式示出了如下内容:在 BD 应用中,为了实现更丰富的交互性,导 Java(注册商标)那样的程序环境的同时,通过控制自静止图像平面或 PG 平面的数据输出,来有效活用存储器总线带宽。基本上是基于第一实施方式的内容,下面说明追加或不同的部分。

[0330] <再现装置 100 的硬件结构>

[0331] 关于再现装置 100,是在第一实施方式的基础上,追加了 PG 平面控制部 28。在此,说明 PG 平面控制部 28 的功能。其他各部分,与第一实施方式中示出的内容相同。

[0332] PG 平面控制部 28 具有控制来自 PG 平面的数据输出的功能。具体地说,接受从 CPU25 发送给视频平面 5 的定标指示,在该定标指示是在电视机 130 的全体上显示的显示指示即全屏显示指示的情况下,输出来自 PG 平面 14 的数据。并且,在定标指示是全屏的 1/4 即四分之一化的情况下,抑制来自 PG 平面 14 的数据读取。

[0333] <动作>

[0334] 图 32 是表示在第二实施方式中根据应用的请求,进行来自 PG 平面 14 和静止图像平面 16 的数据的输出控制的流程图。

[0335] 如图 32 所示,首先从应用请求视频定标,CPU25 向视频处理器(未图示)指示视频定标(步骤 S101)。

[0336] 接着,视频处理器向 CPU25 通知视频定标是否成功(步骤 S102)。在视频定标成功的情况下(步骤 S102 的“是”),判断该定标指示是不是向电视机 130 的显示画面全体上显示的显示指示即全屏显示指示(步骤 S103)。

[0337] 在是全屏显示指示的情况下(步骤 S103 的“是”),CPU25 向静止图像平面控制部 17 指示抑制来自静止图像平面 16 的数据输出(步骤 S104)。此外,CPU25 指示 PG 平面控制部 28 执行来自 PG 平面 14 的数据输出(步骤 S106),并结束。

[0338] 在不是定标指示的情况下(步骤 S103 的“否”),即视频定标的指示是全屏显示的 1/4 即四分之一化定标的情况下,CPU25 指示静止图像控制部 17 执行来自静止图像平面 16

的数据输出（步骤 S105）。此外，CPU25 指示 PG 平面控制部 28 抑制来自 PG 平面 14 的数据输出，并结束。

[0339] 并且，在步骤 S102，即使在视频定标失败的情况下（步骤 S102 的“否”），结束该流程。

[0340] 如上所述地，应用对全画面的视频定标成功的情况下，能够抑制来自静止图像平面 16 的数据的读取，并执行来自 PG 平面 14 的数据的读取。此外，在应用对全画面的 1/4 大小的视频定标成功的情况下，能够执行来自静止图像平面 16 的数据的读取，抑制来自 PG 平面 14 的数据的读取。

[0341] 对于 PG 平面 14，也如上所述地进行基于视频定标的读取控制，在视频定标是四分之一定标的情况下，通过抑制数据的输出，能够空出应该用于该输出的存储器总线带宽。

[0342] < 第三实施方式 >

[0343] 在第一实施方式中，示出了进行静止图像平面 16 的输出控制，在不需要静止图像平面 16 的数据输出的情况下，能够相应地空出存储器总线带宽。

[0344] 在第三实施方式中，示出能够有效活用存储器总线带宽和存储器区域的其他方法。在此，只说明不同于第一实施方式的部分。

[0345] 如第一实施方式及第二实施方式所示，对于来自静止图像平面 16 的数据输出，在视频定标是全屏比例的情况下，抑制来自静止图像平面 16 的数据输出，在视频定标是四分之一大小的情况下，执行来自静止图像平面 16 的数据输出。

[0346] 此外，在第二实施方式中记载了对于 PG 平面 14 也执行基于视频定标的输出控制。如上所述，关于 PG 平面，是同静止图像平面 16 的情况相反地，在视频定标是全屏的情况下，执行数据的输出，在视频定标是四分之一大小的情况下，抑制数据的输出。

[0347] 即，关于来自 PG 平面 14 和静止图像平面 16 的数据输出是排他性的，具有若一方执行输出则另一方不执行输出的结构。

[0348] 因此，在第三实施方式中，不同于第一实施方式，共用在 PG 平面 14 和静止图像平面 16 使用的存储器区域。该情况下，当来自应用的关于视频定标的指示是全屏定标的情况下，该存储器区域作为 PG 平面 14 用的存储器区域使用。并且，在关于视频定标的指示是四分之一大小的情况下，该存储器区域作为静止图像平面 16 使用。具体地说，PG 平面 14 和静止图像平面 16 共用存储器的相同地址的区域。

[0349] 这样做，能够实质上只开辟相当于一个平面的存储器区域，其他的存储器区域用于其它的用途。例如，可以具有如下结构：将应显示在静止图像平面 16 上的 JPEG 数据以压缩状态存储，在需要来自静止图像平面 16 的数据输出的情况下，对压缩的 JPEG 数据进行解压并写入静止图像平面。

[0350] 此外，在数据读取时实质上访问的存储器区域相当于减少一个平面的量，可以说存储器总线带宽也相应地产生空余。

[0351] < 补充 >

[0352] 根据上述实施方式说明了本发明涉及的再现装置，但本发明的实施方式不限于上述结构。下面，说明其变形例。

[0353] (1) 在上述实施方式中，将再现装置作为 BD-ROM 再现装置进行了说明，但并不是特别限定于 BD-ROM 再现装置，也可以是 DVD 播放器等。

[0354] (2) 在上述实施方式中,静止图像平面控制部 17 根据是否进行视频平面的定标变更,来判断是否进行来自静止图像平面的静止图像数据的读取,但是,也可以根据在用于存储字幕数据的 PG 平面中是否有要合成的数据来进行判断。

[0355] 字幕数据基本上以进行全屏显示时的字体大小存储在 BD-ROM 中,变更比例而缩小的情况下,字会变形,所以在合成 PG 平面的情况下,必须进行全屏显示。这样,静止图像平面控制部 17 在有 PG 平面时判断为进行全屏显示,此时,不进行来自静止图像平面的静止图像数据的读取。

[0356] (3) 在上述第一实施方式中记载了,在运动图像全屏显示的情况下,本来预定在静止图像平面的读取中使用的存储器总线带宽空出来了。这空出来的存储器总线带宽当然也可以用于其它的用途。

[0357] 例如,代替静止图像平面的读取,可以在对视频平面的运动图像用图像数据的写入中使用,或者在对 IG 平面的 GUI 图像的写入中使用。通过具备上述结构,再现装置能够缩短写入和读取的周期,能够尽可能地抑制电视机上的图像显示的延迟。

[0358] (4) 在上述实施方式中,视频流是 BD-ROM 标准的 AVClip,也可以是 DVD-Video 标准、DVD-Video Recording 标准的 VOB (Video Object)。VOB 是通过将视频流、音频流多路复用而得到的、基于 ISO/IEC13818-1 标准的程序流。而且,AVClip 中的视频流可以是 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 方式。

[0359] (5) 上述实施方式中,基于 MPEG4-AVC (称为 H. 264 或 JVT) 进行了说明,但也可以是 MPEG2 视频流,或者,即使在其它格式 (VC-1 等) 的图像的情况下,只要是能够单独解码的图像,就能容易地应用。

[0360] (6) 在上述实施方式中,仅在视频平面是全屏时不合成静止图像平面,但这是因为,静止图像平面和视频平面通常都是以全屏大小准备的。但是,视频平面和静止图像平面也可以不都是全屏大小,只要视频平面的数据遮盖静止图像平面的全部数据,也可以具备不执行静止图像平面的合成的结构。

[0361] (7) 上述实施方式中,根据视频平面、即进行运动图像的定标时成为全屏还是成为四分之一屏,控制了来自静止图像平面 16 的数据的输出,但是,也可以不在视频定标时进行来自静止图像平面 16 的数据的输出控制。

[0362] 此情况下,使应用具有抑制或执行对静止图像平面控制部 17 的数据输出的功能。通过这样做,不仅在视频定标时,也可以在其它时刻执行来自静止图像平面 16 的数据的输出控制。

[0363] 此外,在第二实施方式中,关于来自 PG 平面 14 的数据的输出控制,不仅在视频定标时,也可以使应用具有执行对于 PG 平面控制部 28 的输出控制的功能。通过这样做,不仅在视频定标时,在其它时刻也能够执行来自 PG 平面 14 的数据的输出控制。

[0364] (8) 在上述实施方式中,以基于应用的指示的方式进行了静止图像平面 16 的数据的输出控制,但是,也可以使 BD-ROM 持有与静止图像数据的输出控制有关的标志,根据该标志进行输出控制。

[0365] (9) 在上述实施方式中,在 PG 平面 14 仅存储字幕,不仅是字幕的图像数据,也可以存储与存储在视频平面中的运动图像不同的运动图像。这样做,能够例如将显示画面分割

为 2 个区域,并分别显示不同的运动图像。

[0366] (10) 本发明也可以是上述实施方式中示出的静止图像平面的静止图像数据的合成控制中表示的方法,还可以是使再现装置的计算机执行该方法所示的处理步骤的计算机程序。此外,该计算机程序也可以记录在软磁盘、硬盘、CD、MO、DVD、BD、半导体存储器等。

[0367] (11) 在上述实施方式中,再现装置 100 由系统 LSI 实现,但是再现装置 100 的各功能也可以由多个 LSI 实现。

[0368] 本发明涉及的再现装置,可以作为从 BD-ROM 读取数据、一边执行应用一边将图像显示在再现装置上的、没有显示延迟的再现装置而活用。

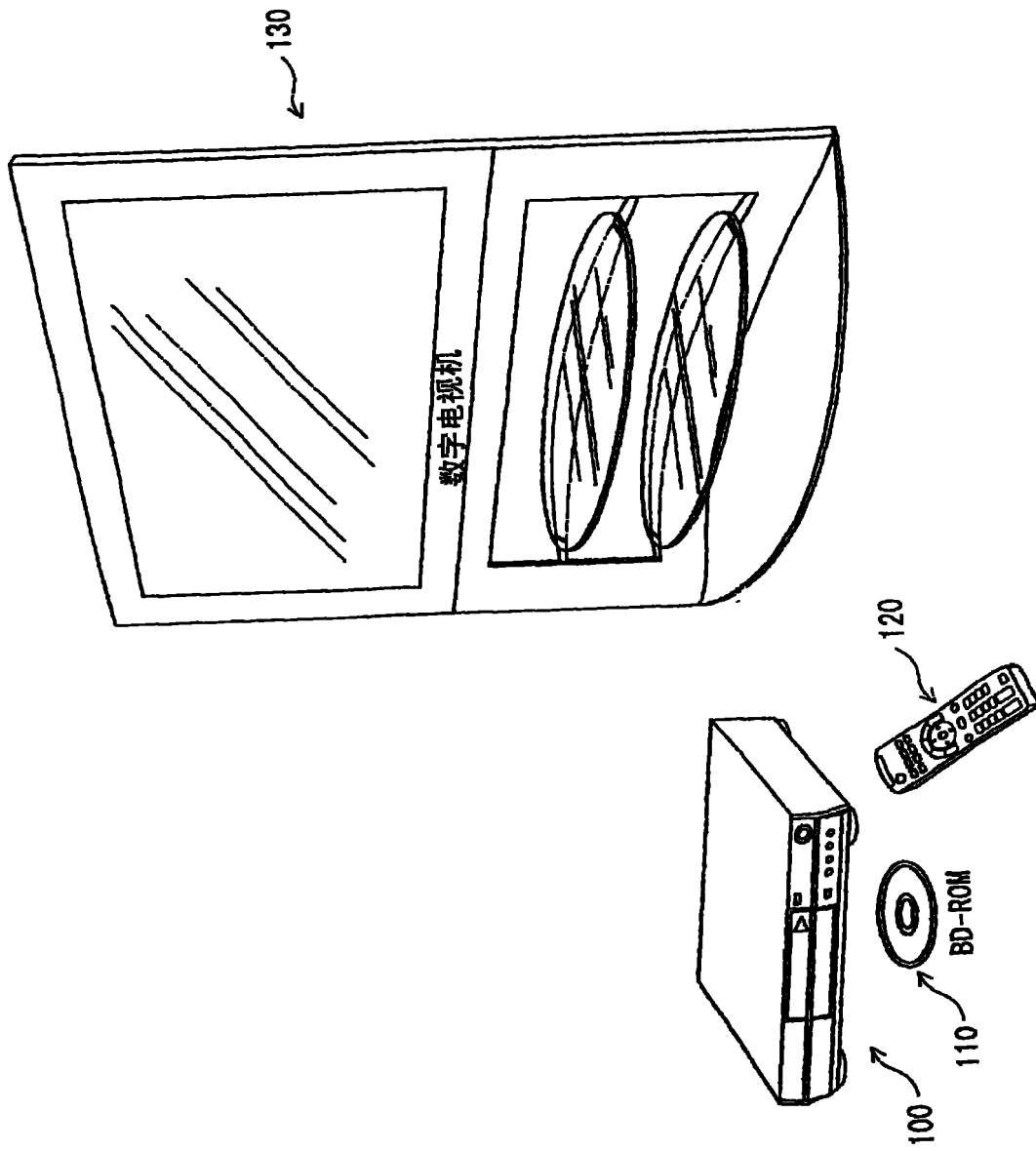


图 1

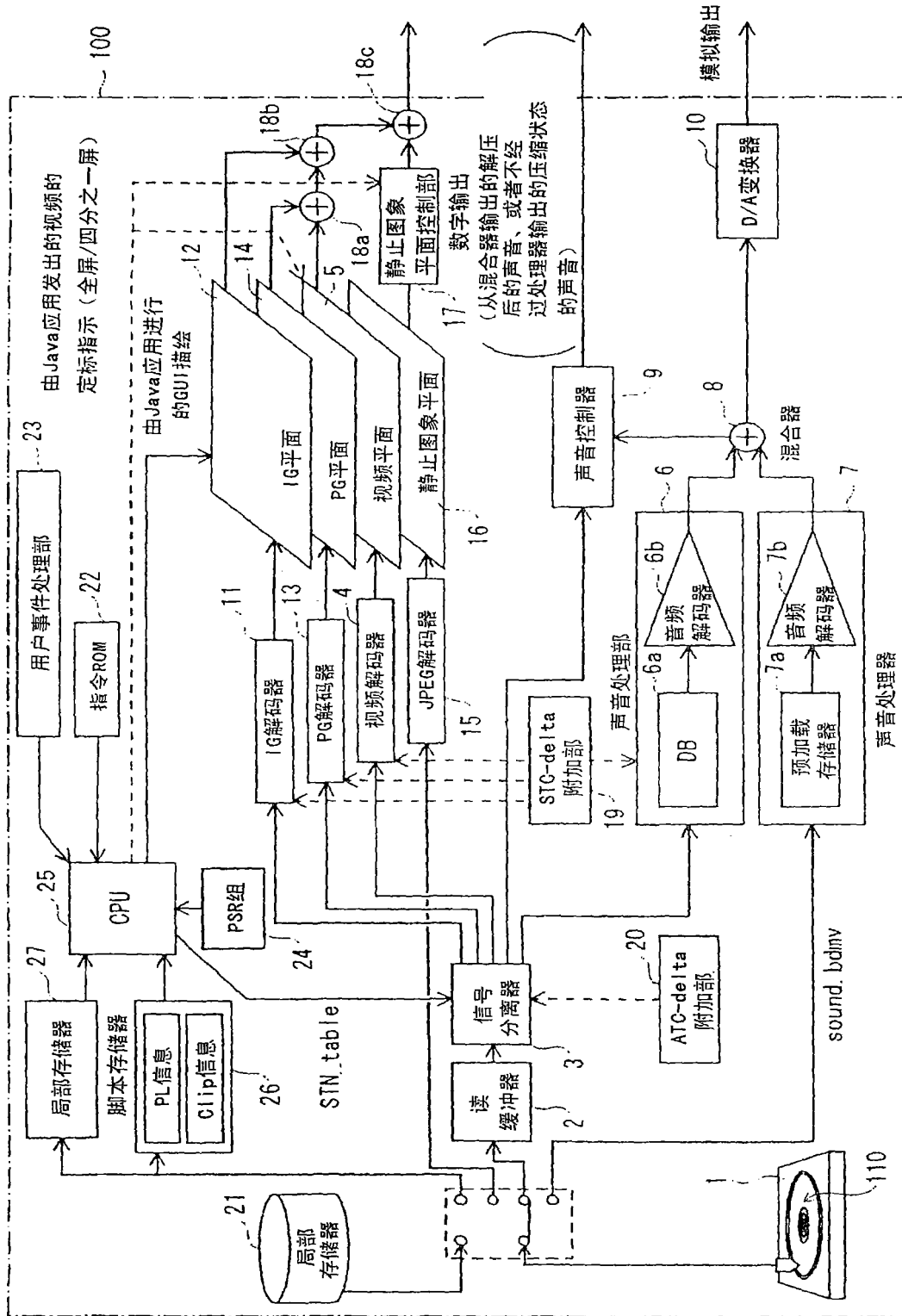


图 2

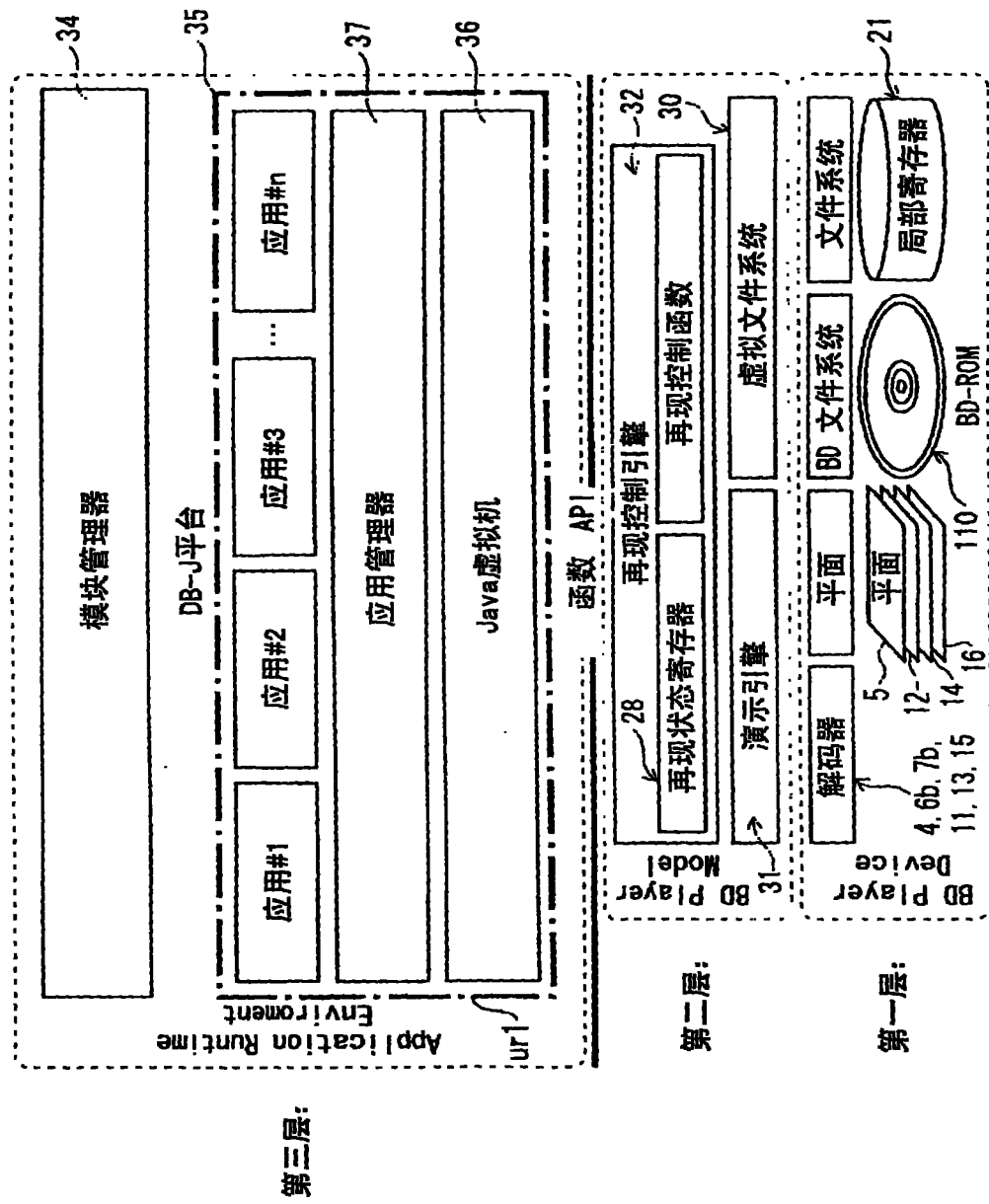


图 3

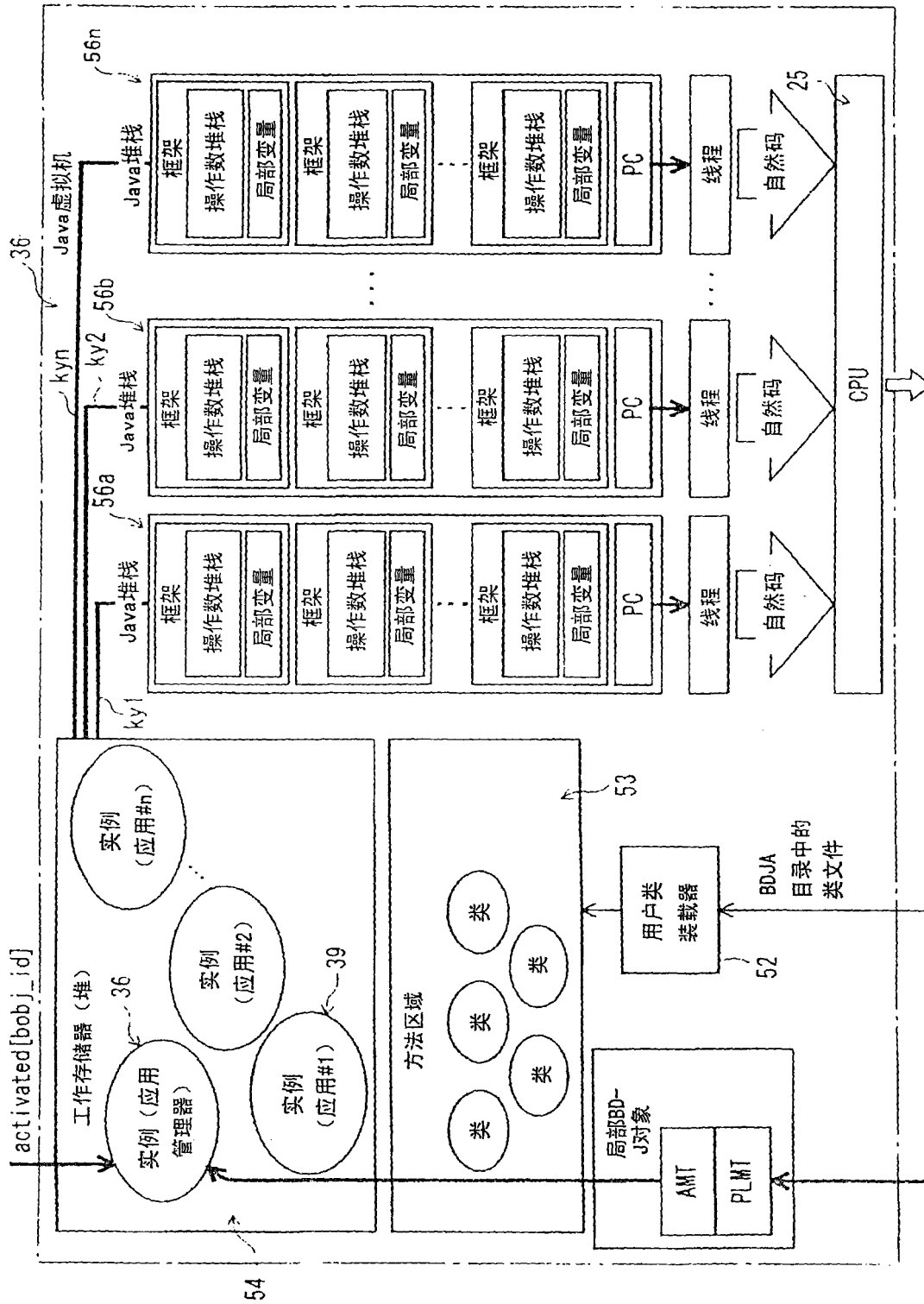


图 4

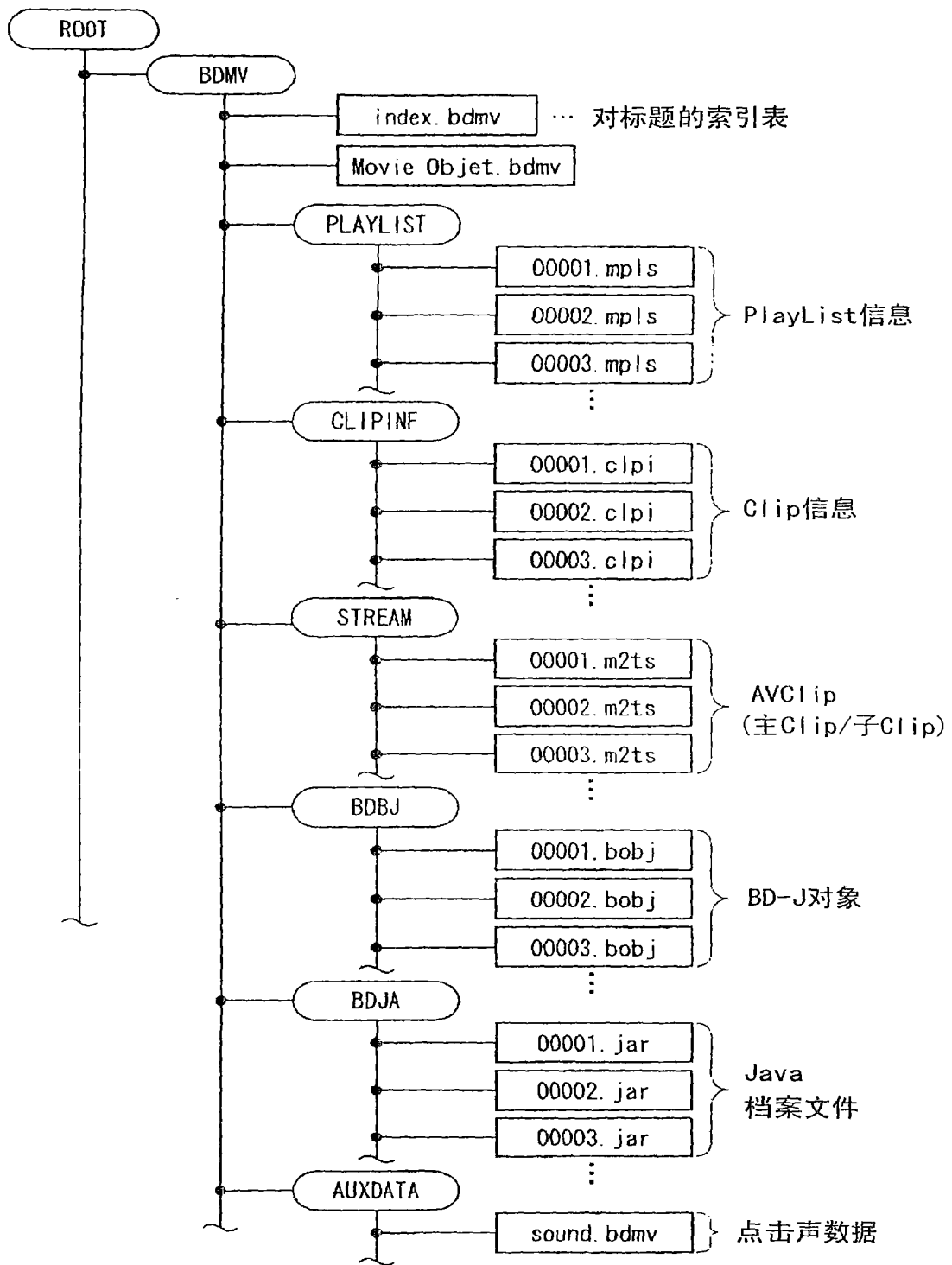


图 5

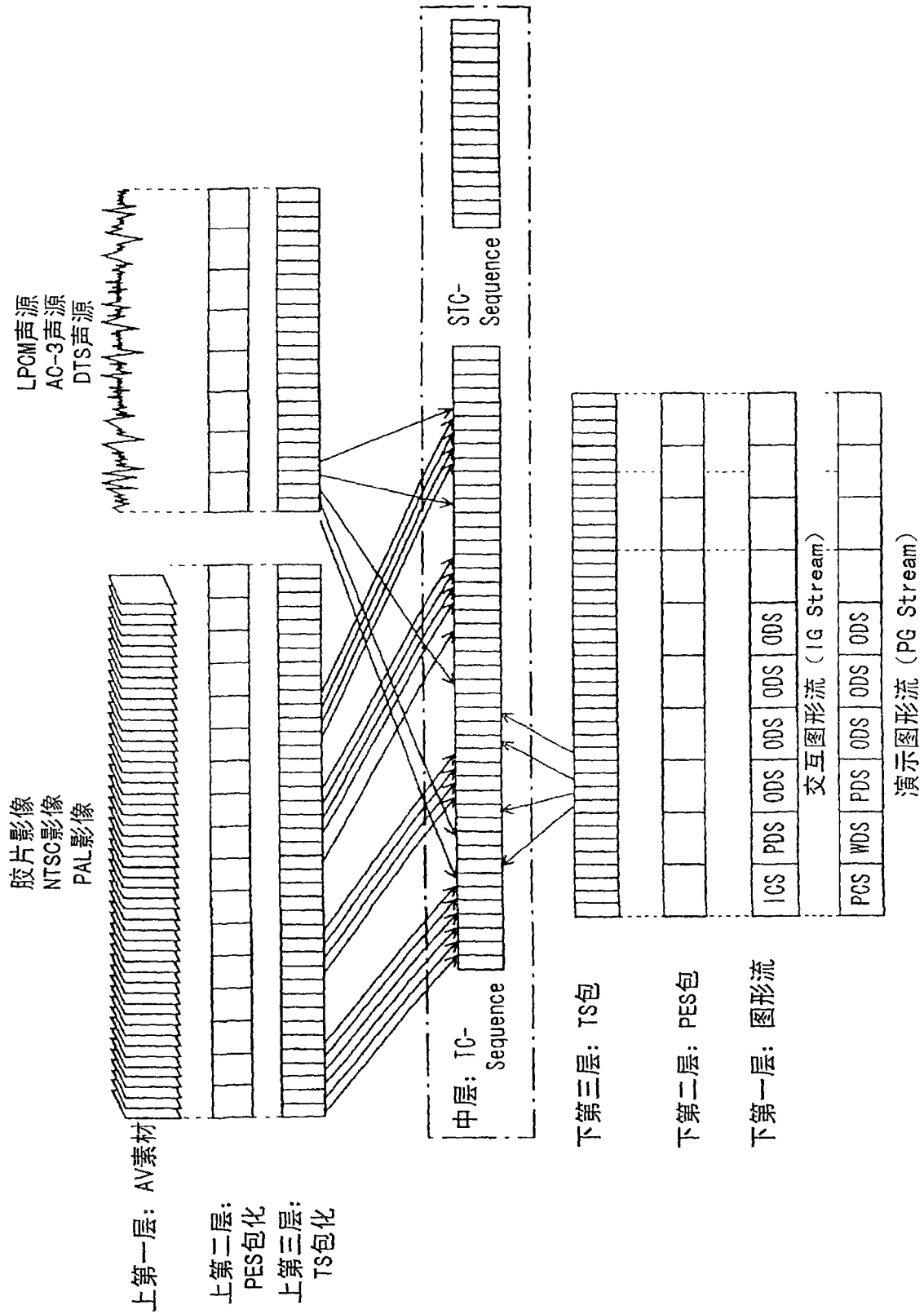


图 6

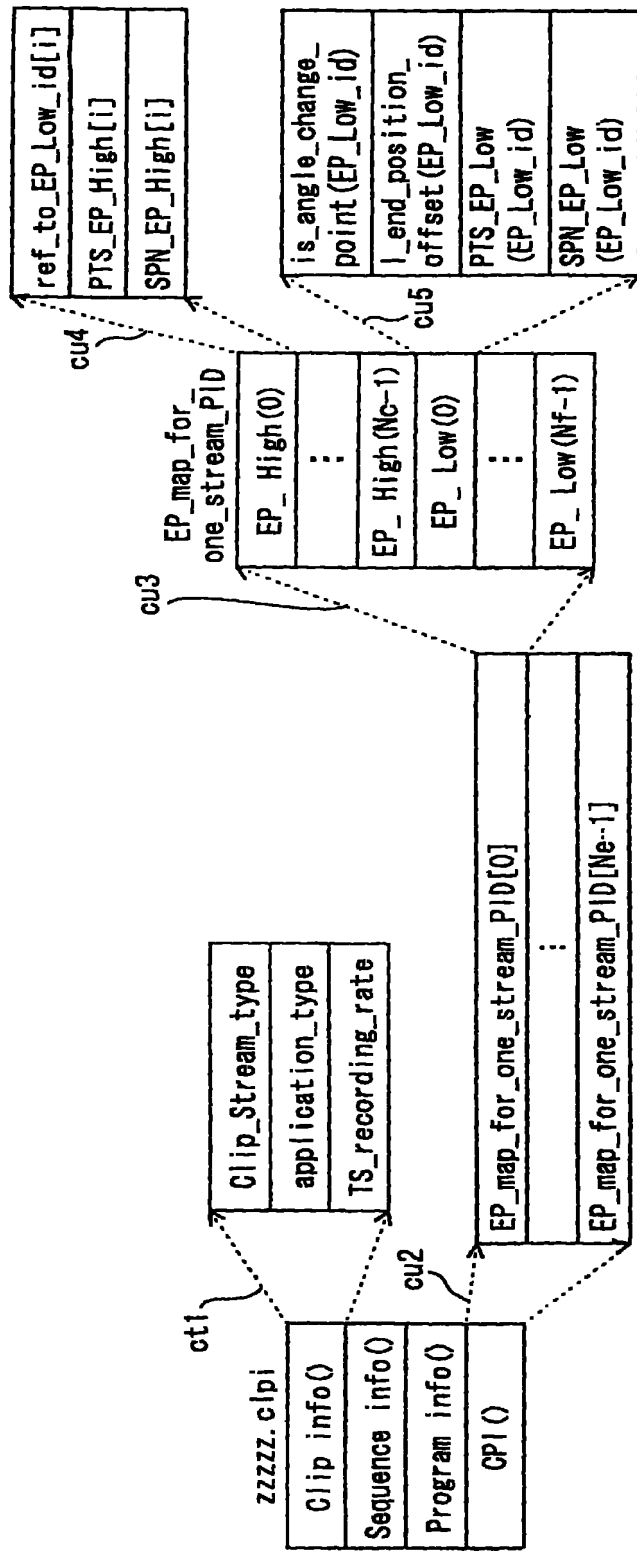


图 7

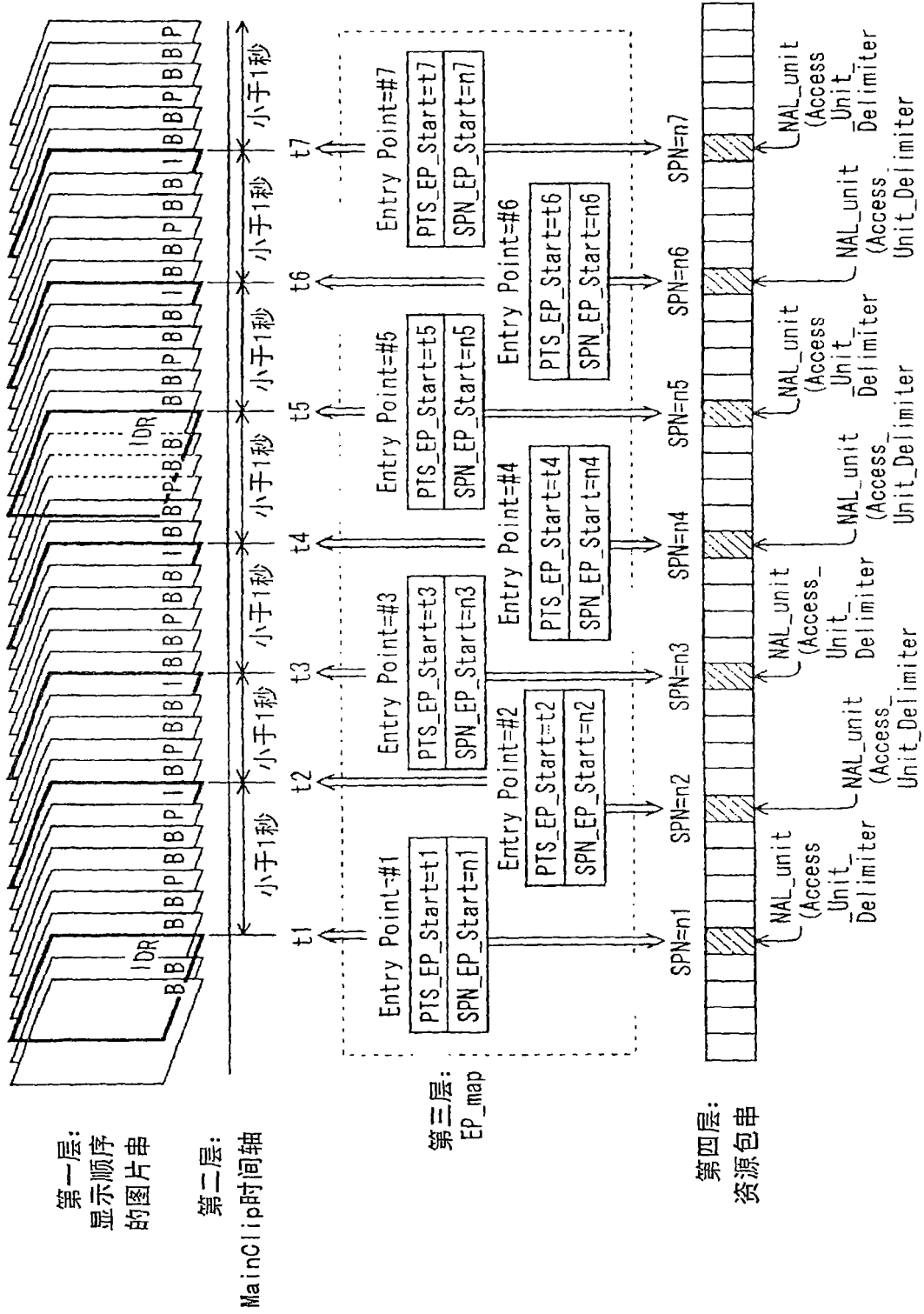


图 8

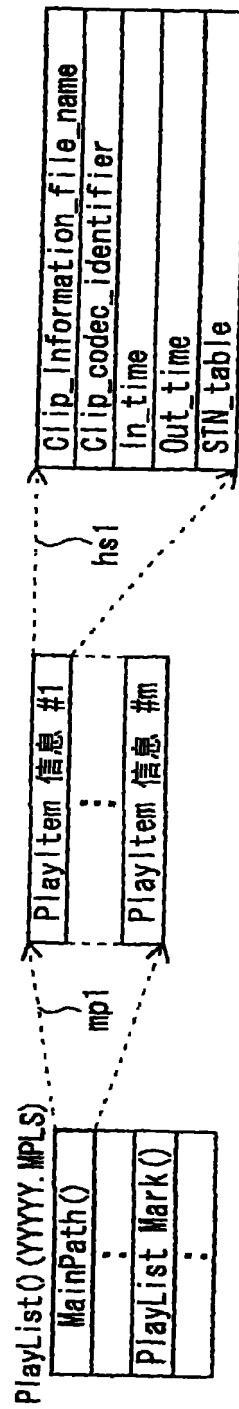


图 9

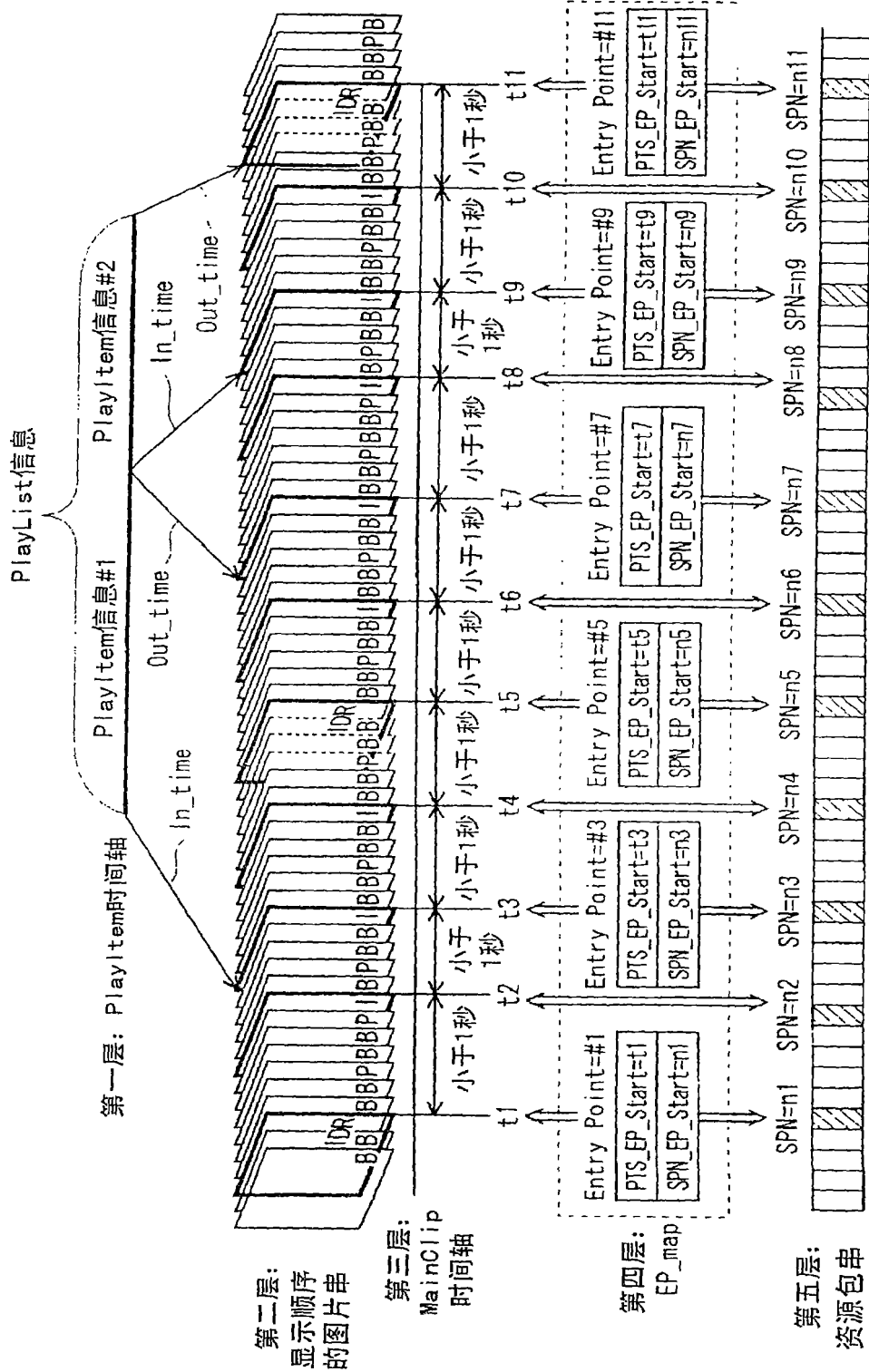


图 10

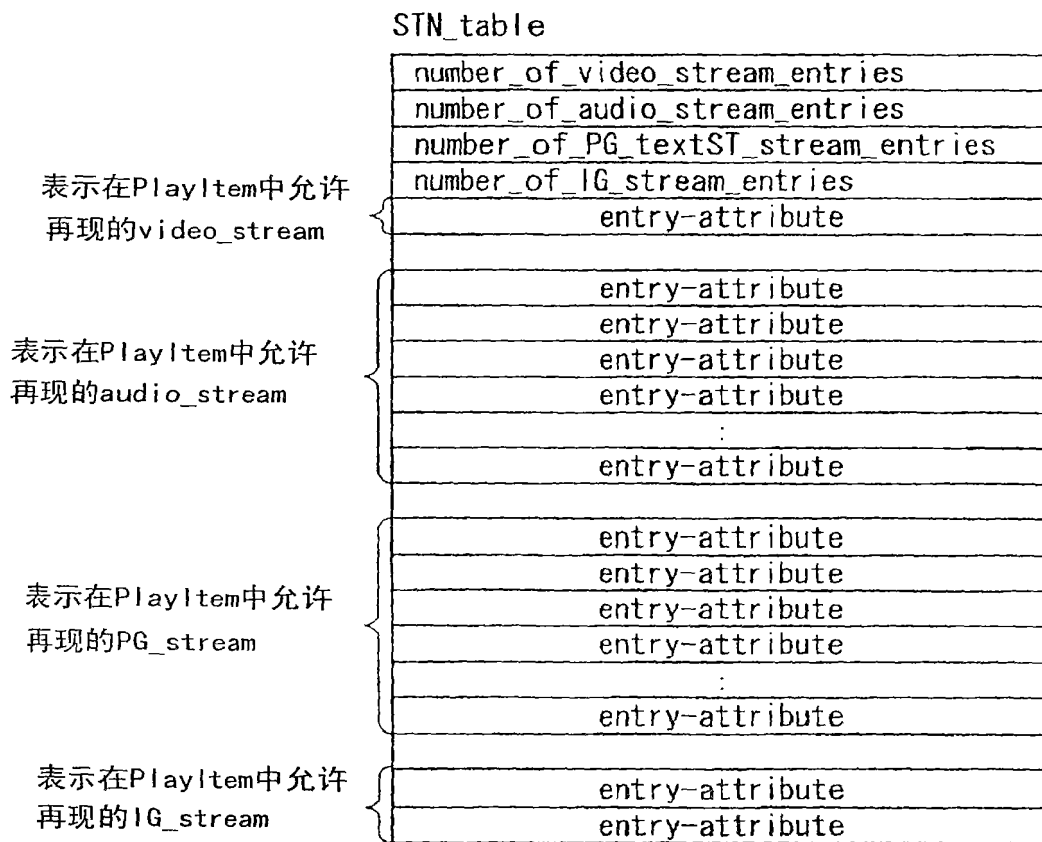
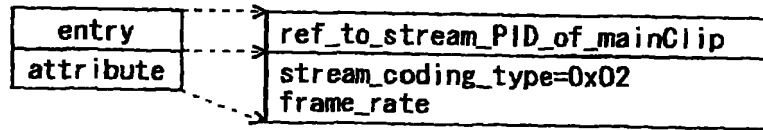
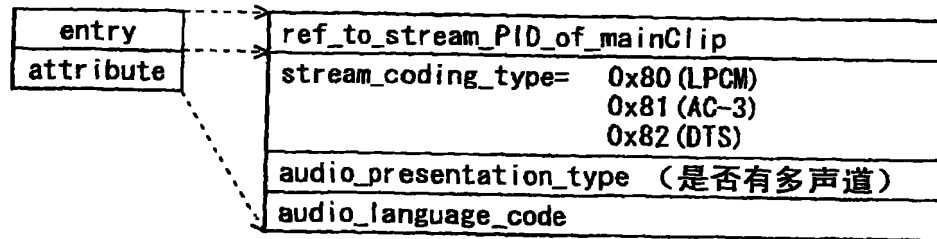


图 11

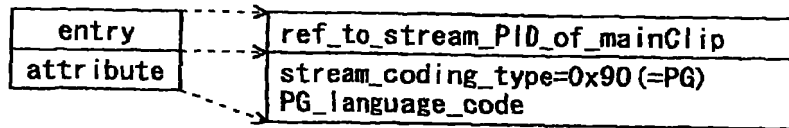
(a) video_streamのentry-attribute



(b) audio_streamのentry-attribute



(c) PG streamのentry-attribute



(d) IG_streamのentry-attribute

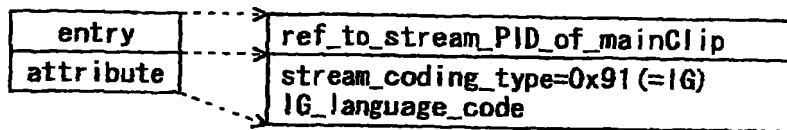


图 12

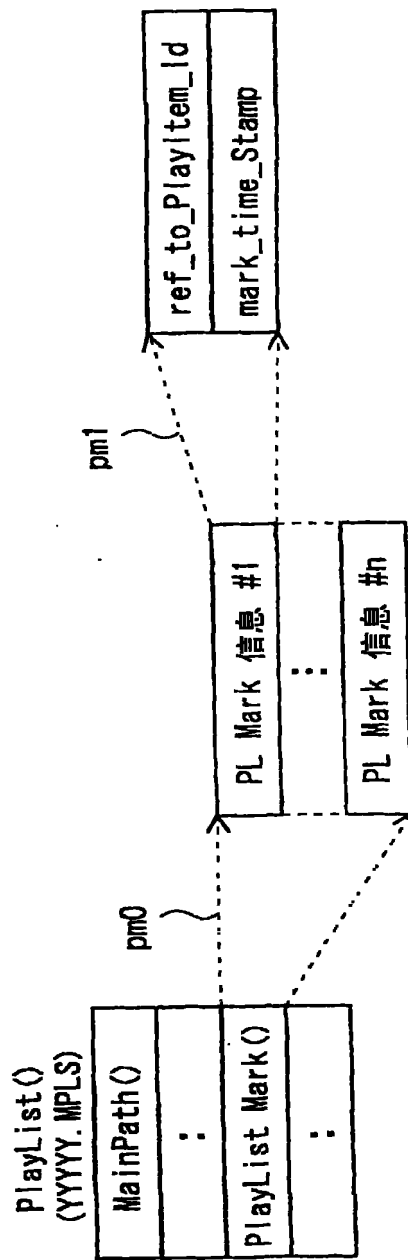


图 13

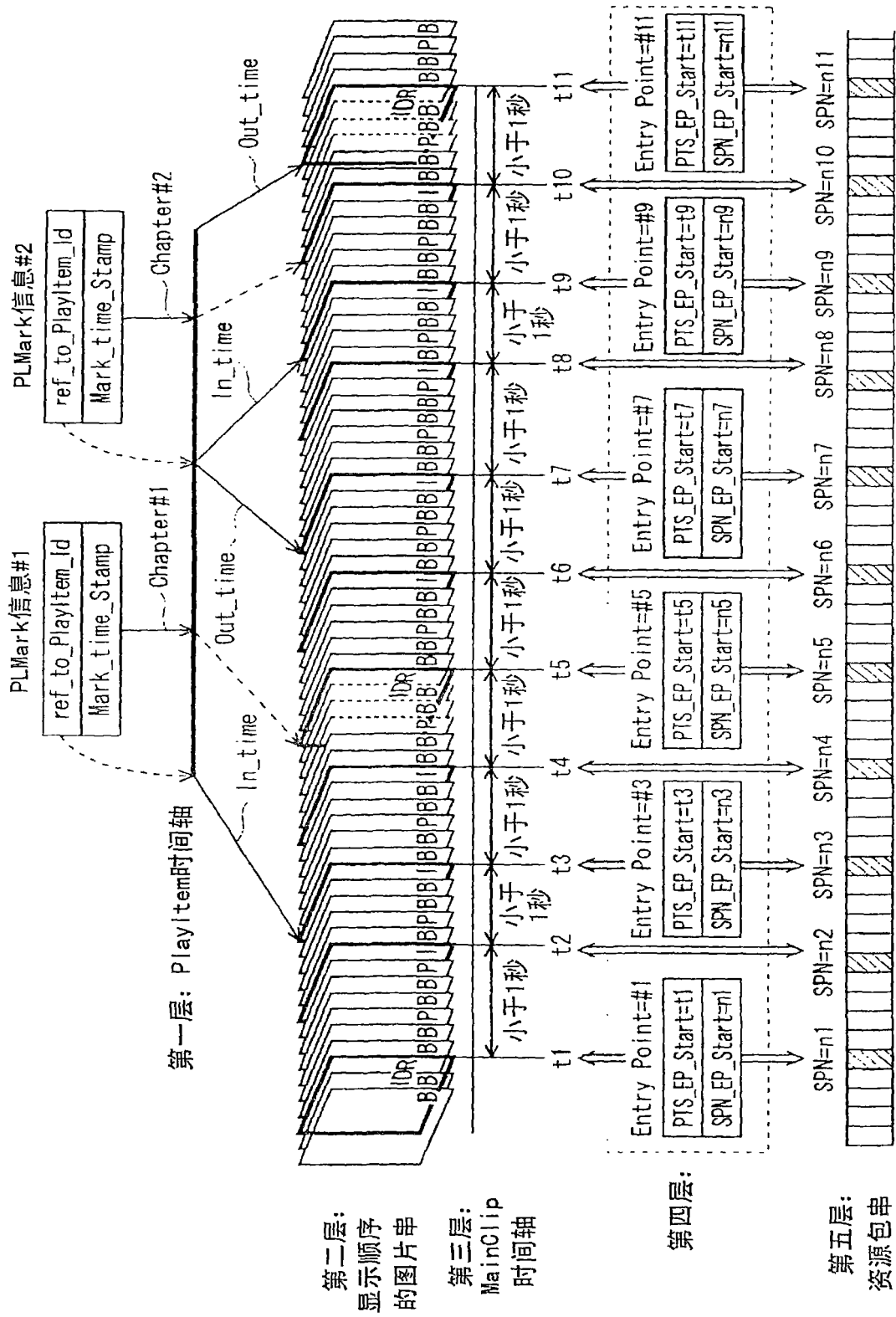


图 14

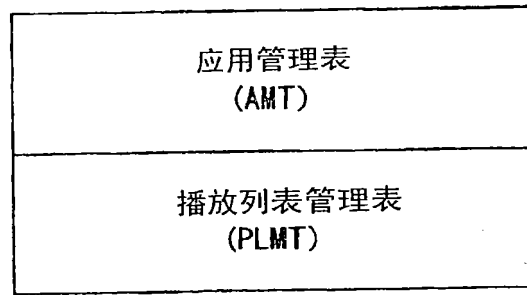


图 15

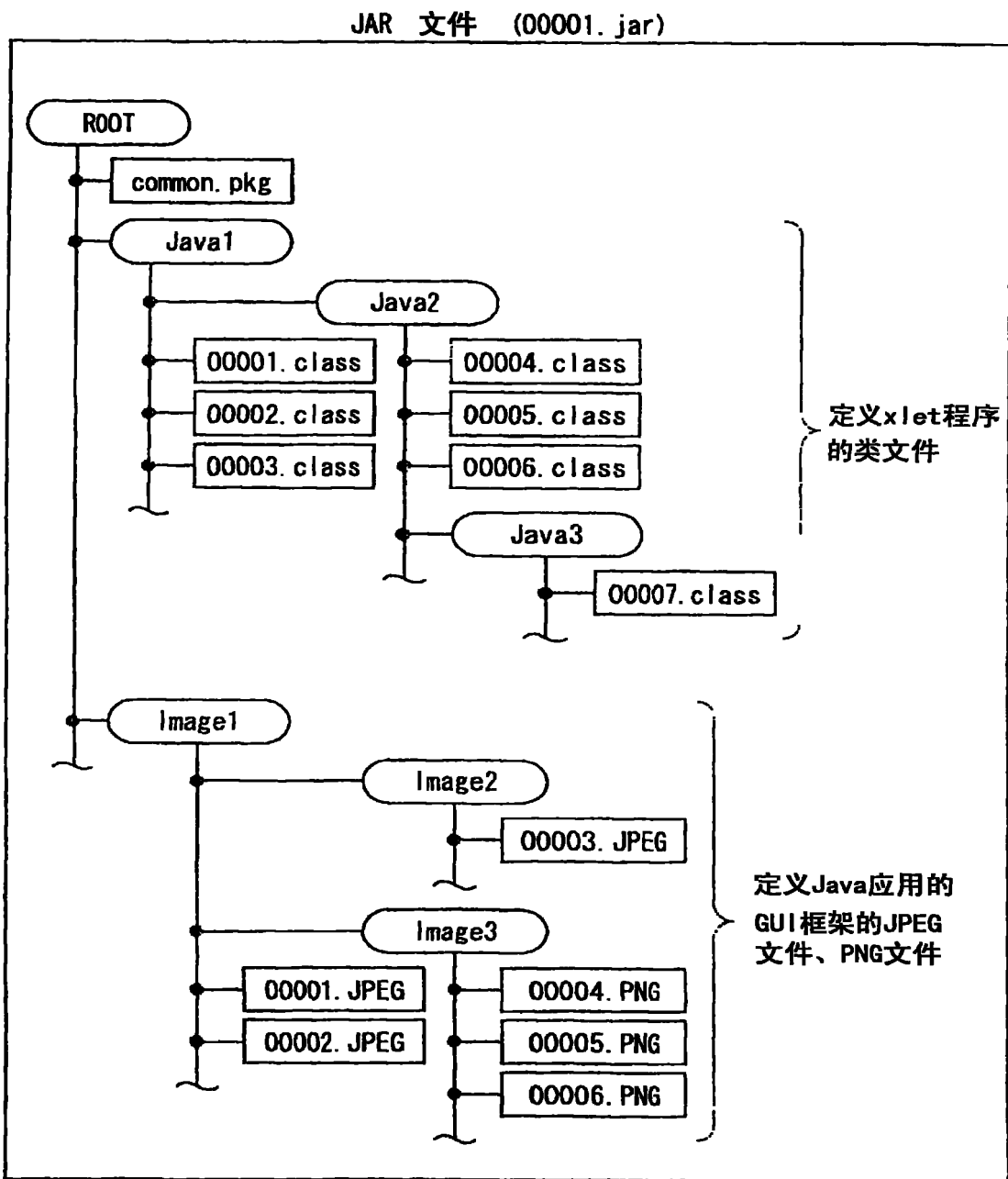


图 16

(a)

life_cycle
apli_id_ref
run_attribute
run_priority

(b)

life_cycle	apli_id_ref	run_attribute	run_priority
使用标题、播放列表、章节表现的应用的“生存区间”	对JAR的文件名赋予的5位整数ZZZZZ, 即“应用ID”的参照值	<ul style="list-style-type: none"> - Auto Run - Present (无指定) - Suspend 中的某一个即“启动属性”	取0~255的值的“启动优先级”

图 17

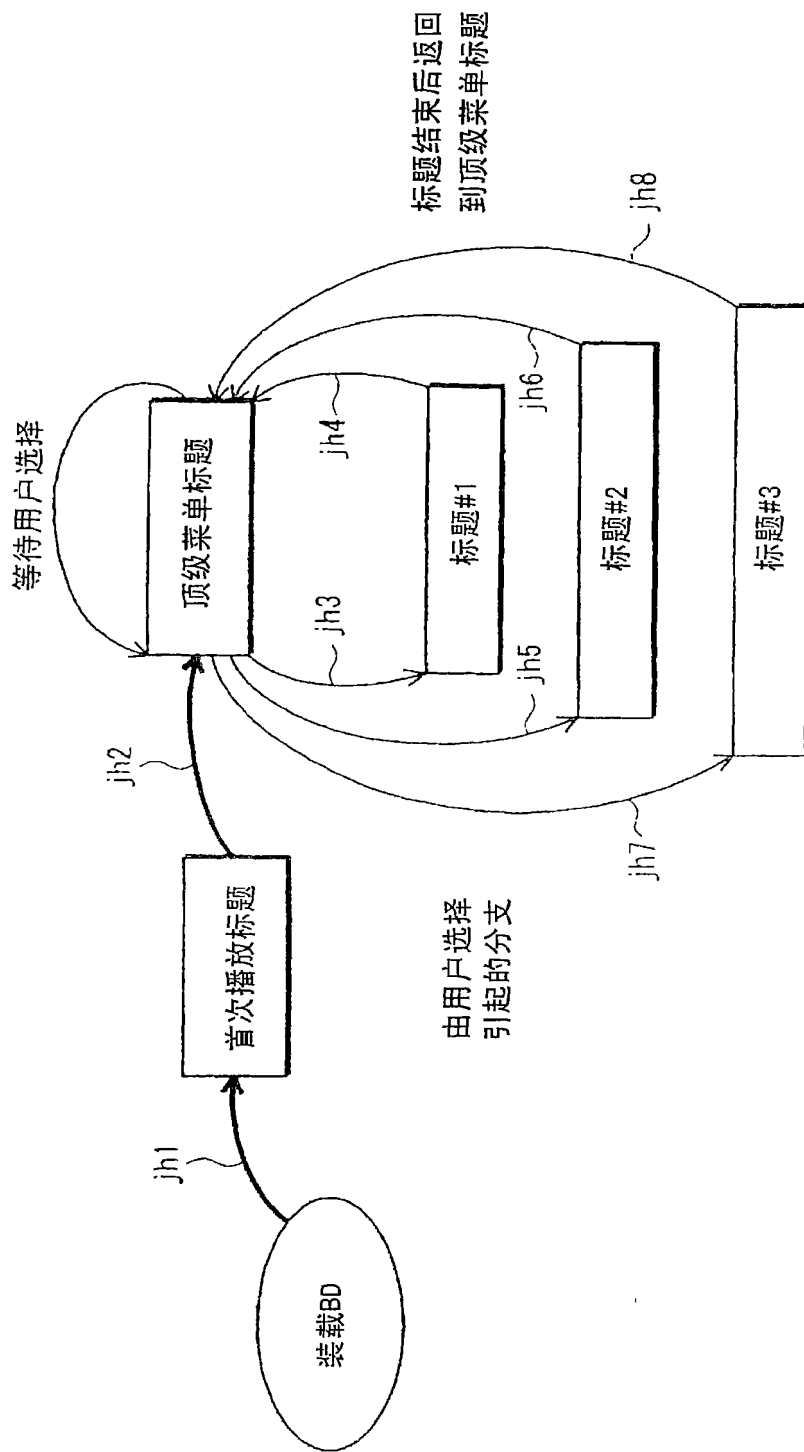


图 18

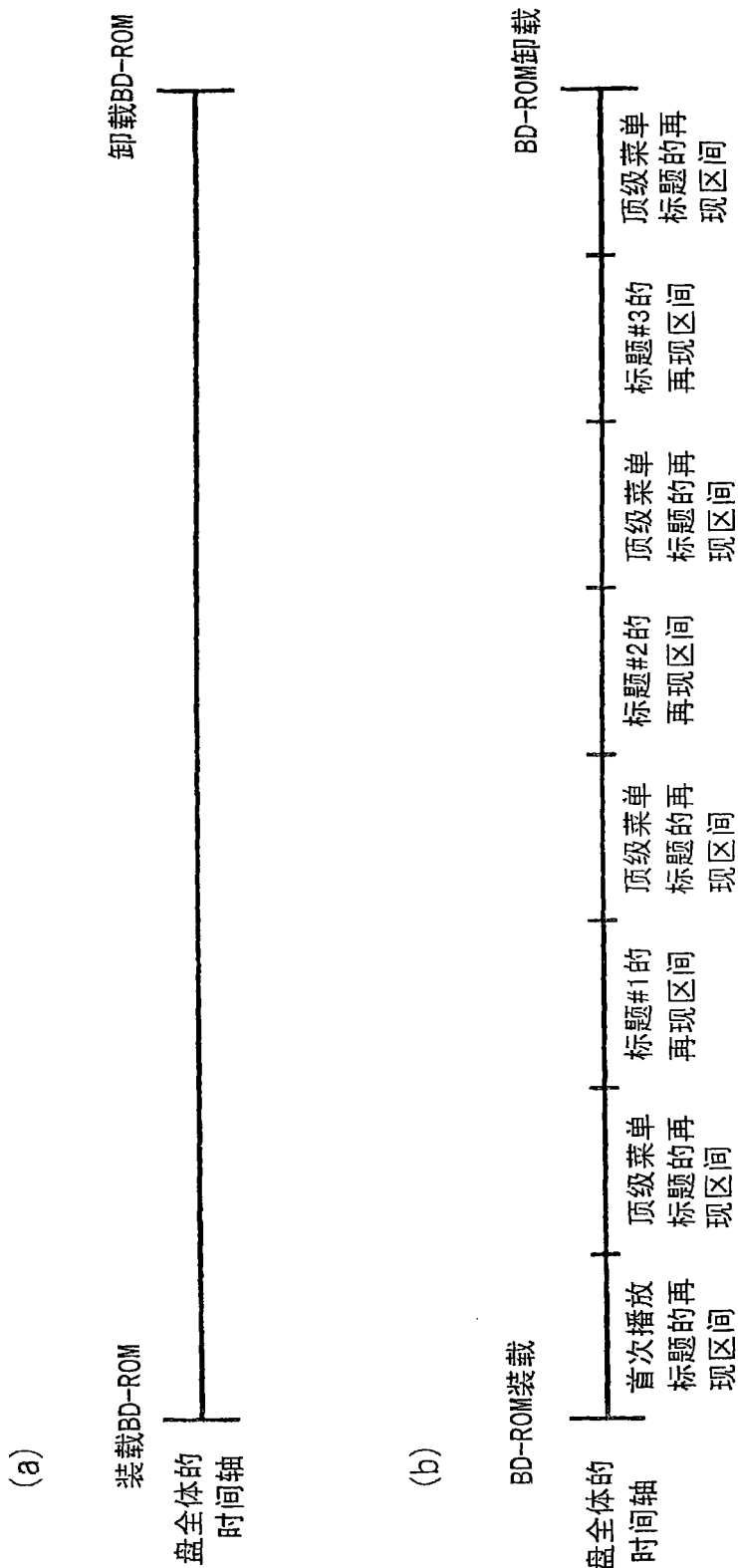


图 19

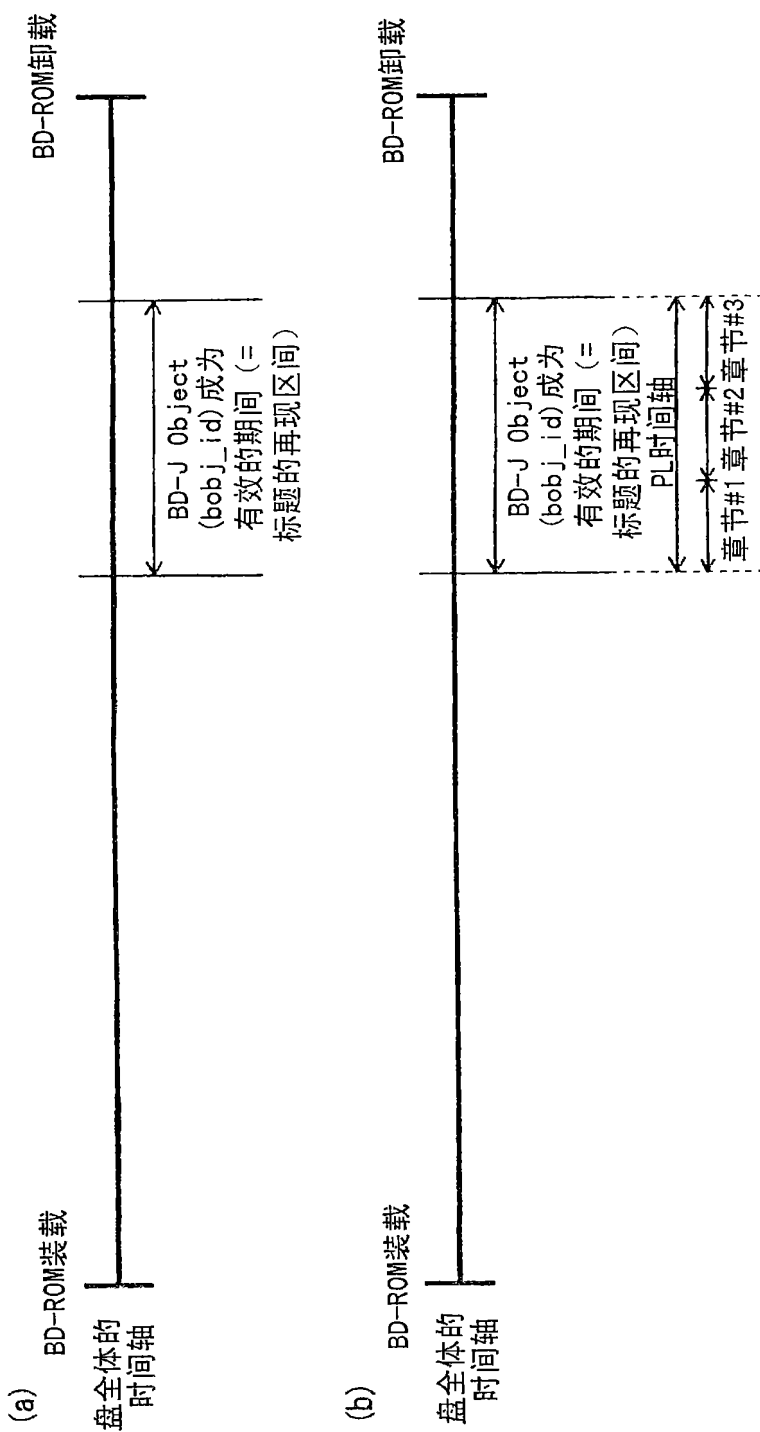


图 20

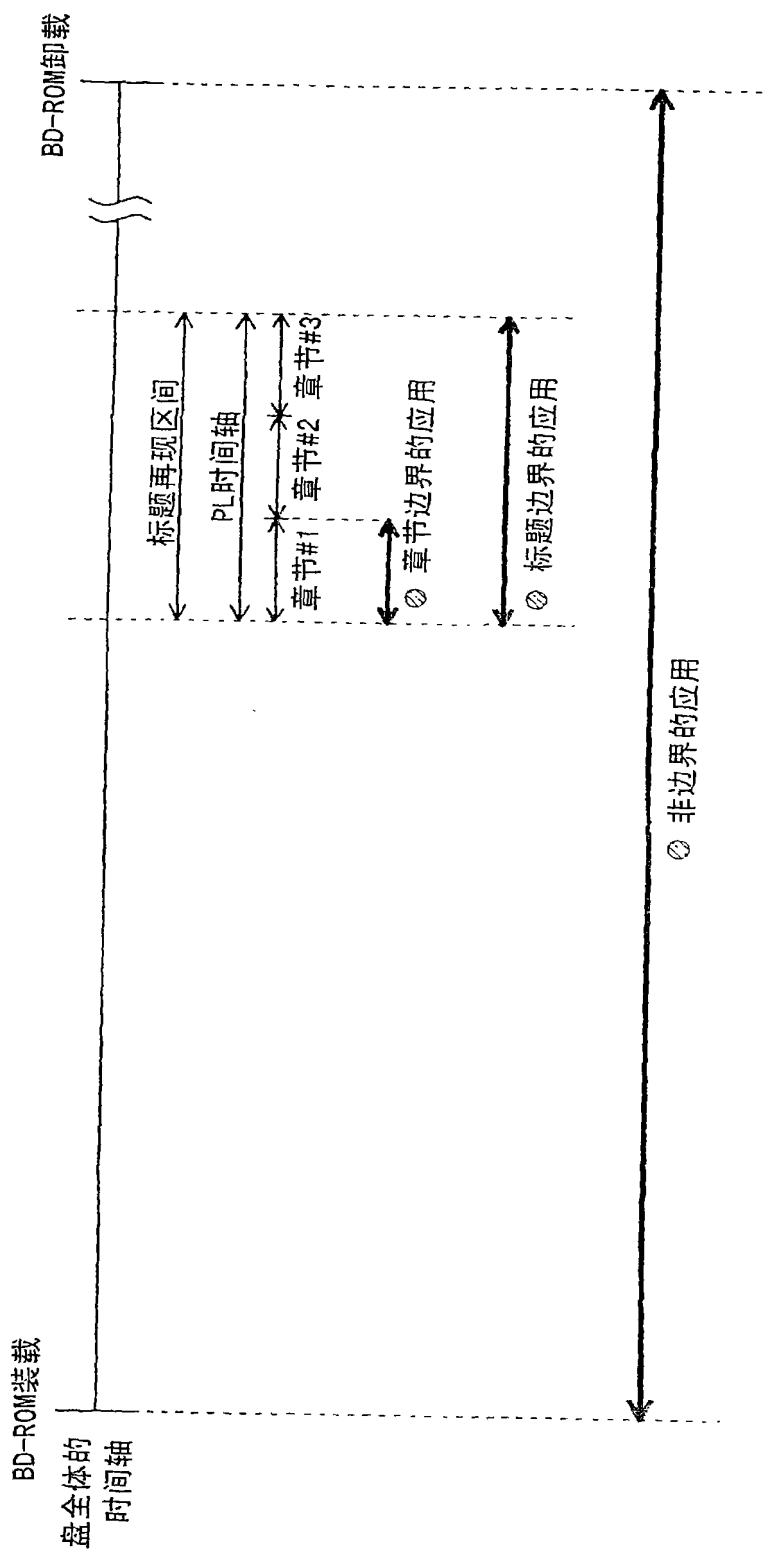


图 21

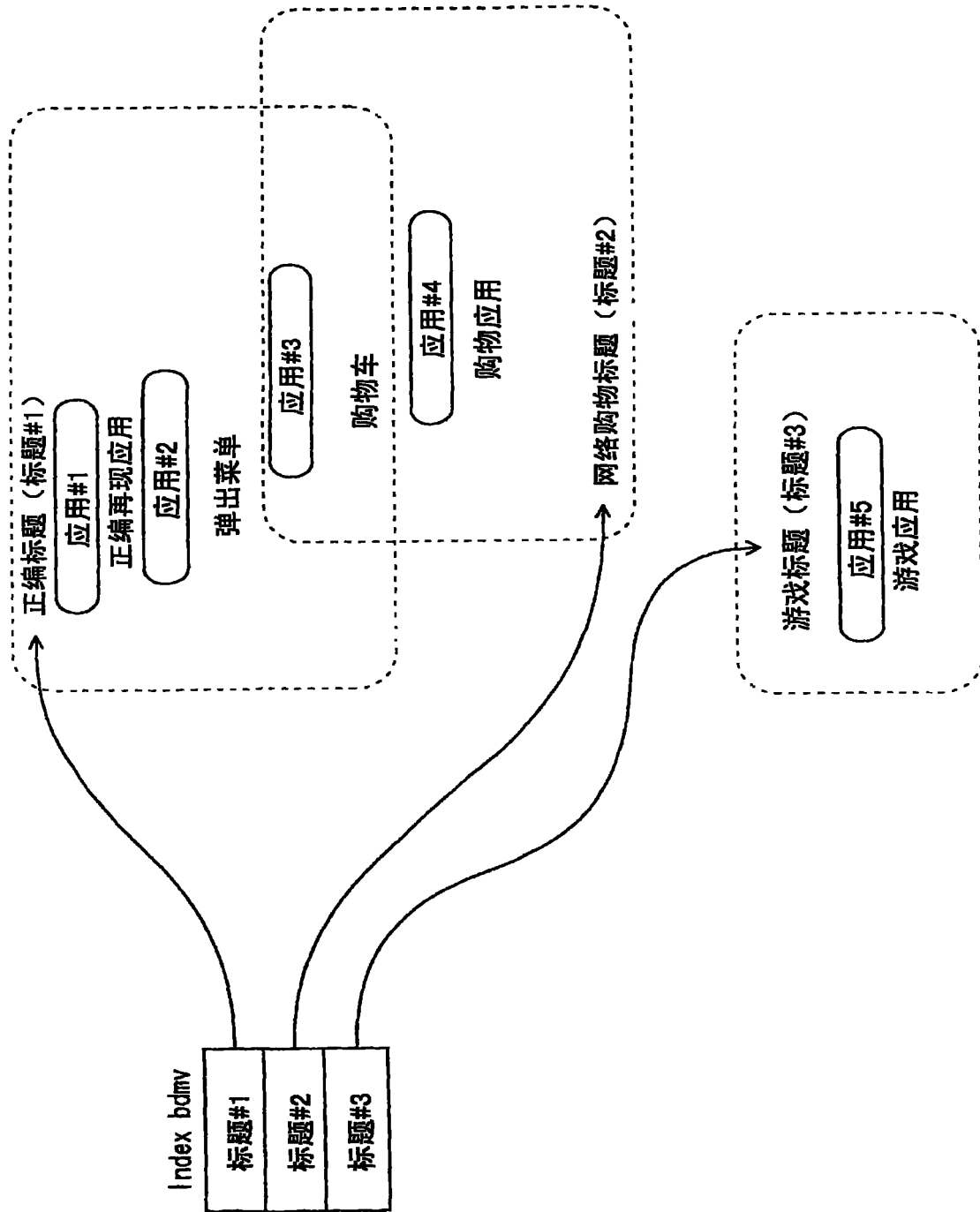
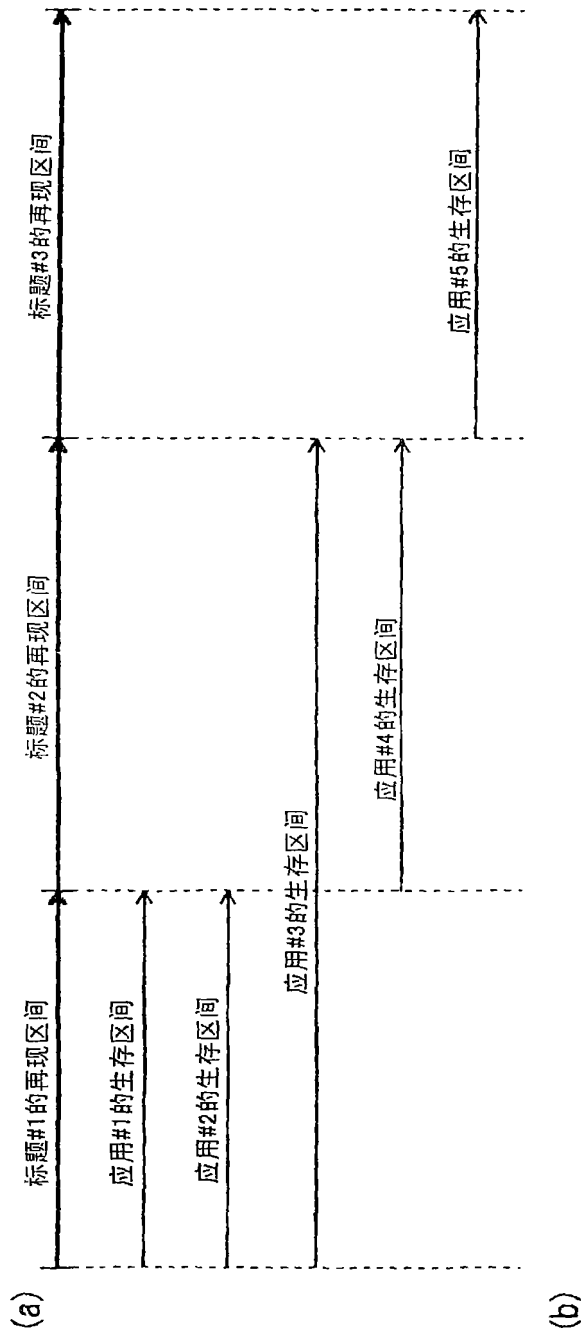


图 22



(b)

生存区间	应用ID的参照值	启动属性	启动优先级
标题#1	应用#1		
标题#1	应用#2		
标题#1	应用#3		

生存区间	应用ID的参照值	启动属性	启动优先级
标题#2	应用#3		
标题#2	应用#4		

生存区间	应用ID的参照值	启动属性	启动优先级
标题#3	应用#5		

图 23

之前标题中的 应用的状态	非启动	启动属性		
		Present	AutoRun	
		Suspend		
	启动中	什么也不做、 状态继续	启动应用	什么也不做、 状态继续
	暂停	什么也不做、 状态继续	什么也不做、 状态继续	暂停
		恢复	恢复	什么也不做、 状态继续

图 24

(a)

PL_id_ref
Playback_Attribute

(b)

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

图 25

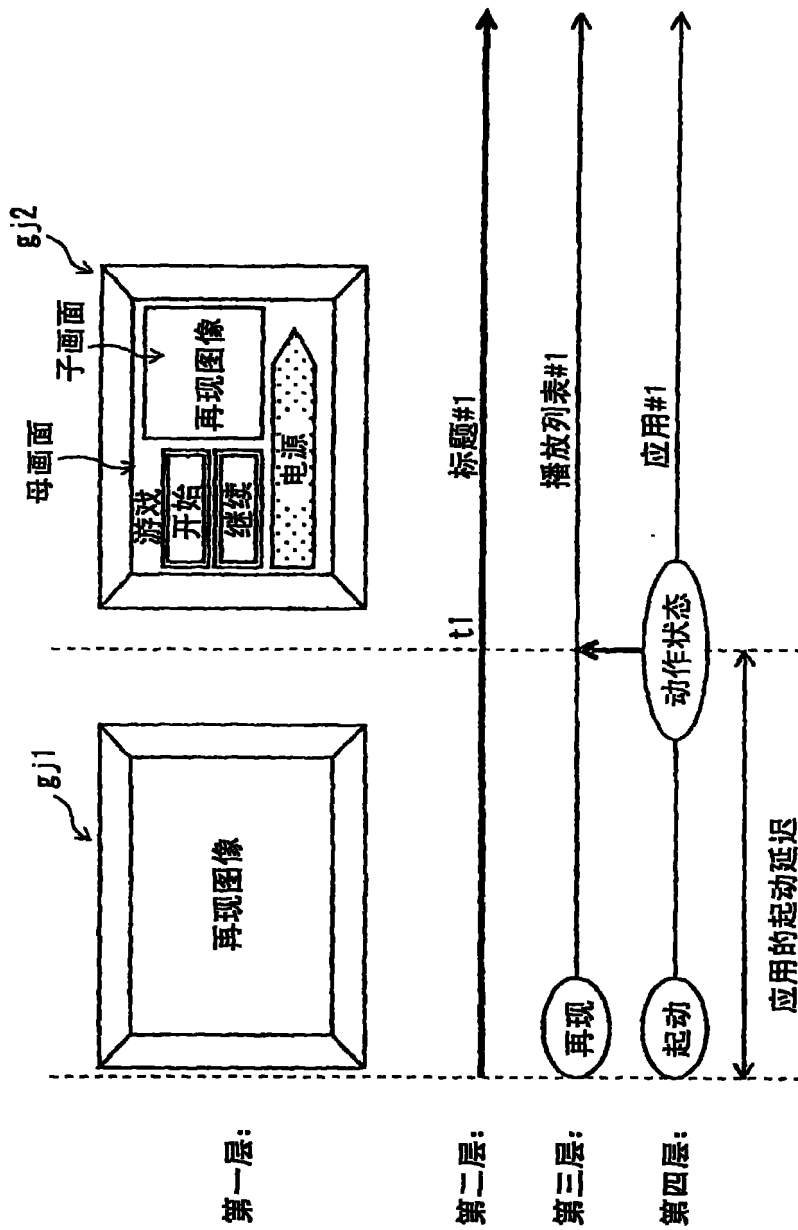


图 26

分支目的地标题中 没有PLMT	再现停止	状态继续	分支目的地标题中有PLMT	再现属性: Present	状态继续
	状态继续	自动再现开始	再现属性: Auto Play	状态继续	状态继续
分支源标题是非再现状态	再现停止	状态继续	分支源标题是非再现状态	状态继续	状态继续
分支源标题是再现中状态	状态继续	状态继续	分支源标题是再现中状态	状态继续	状态继续

图 27

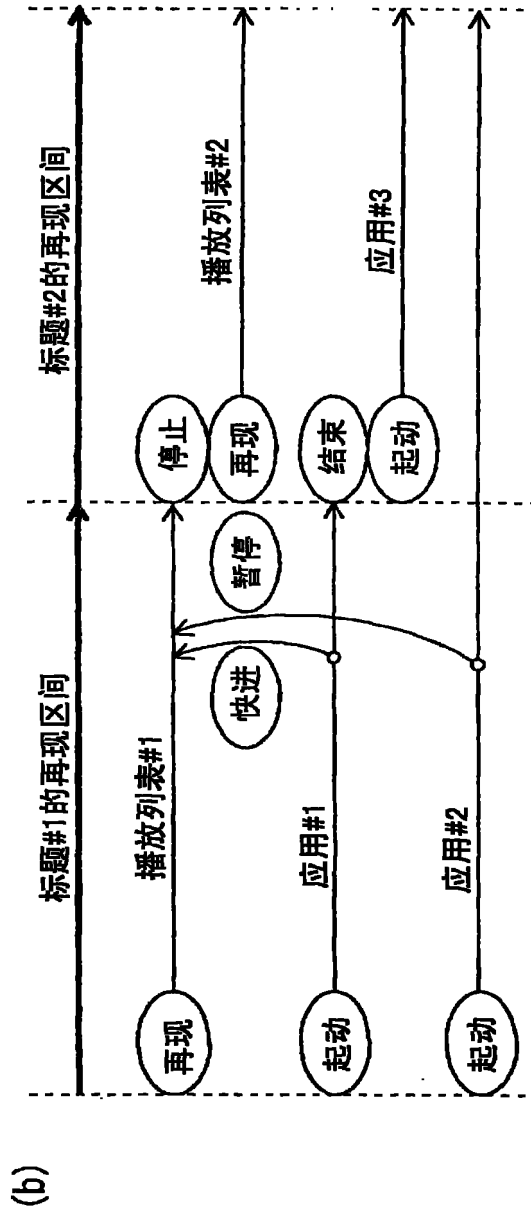
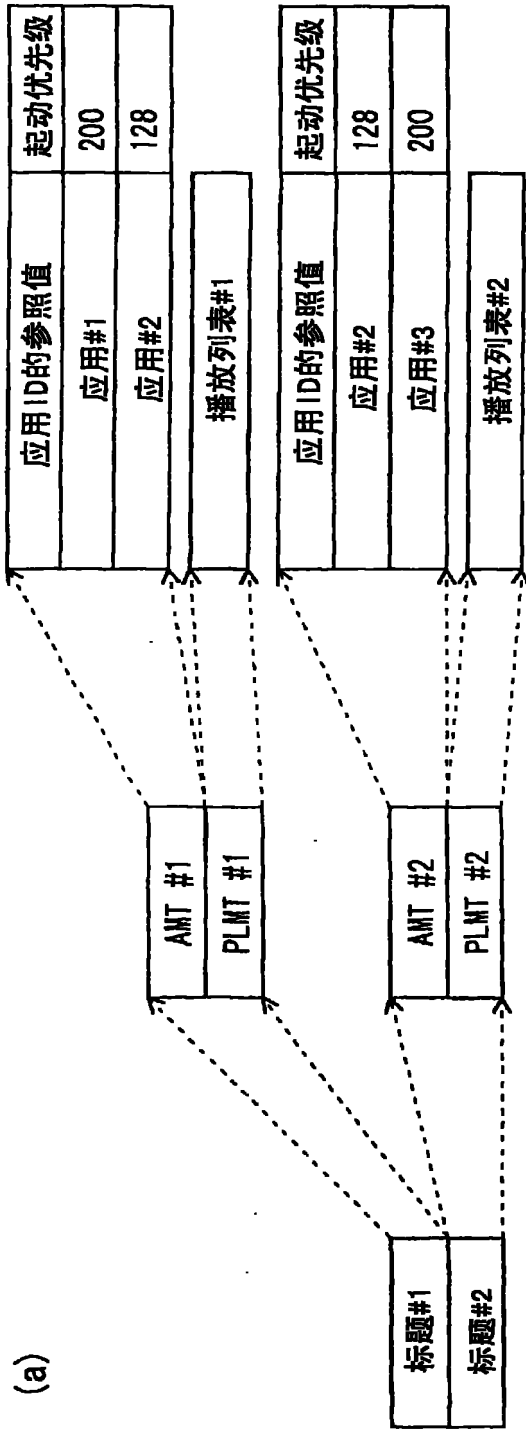


图 28

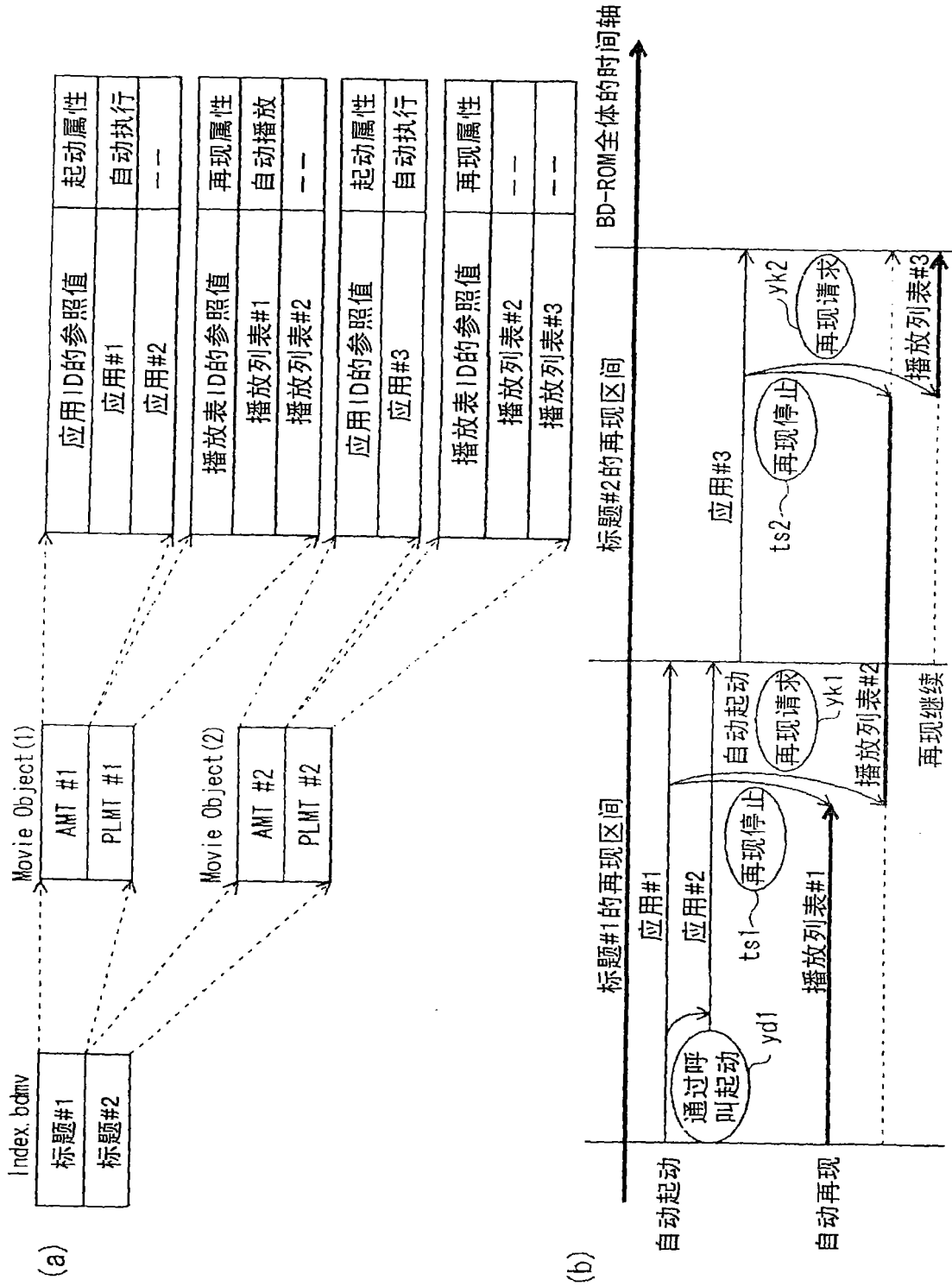


图 29

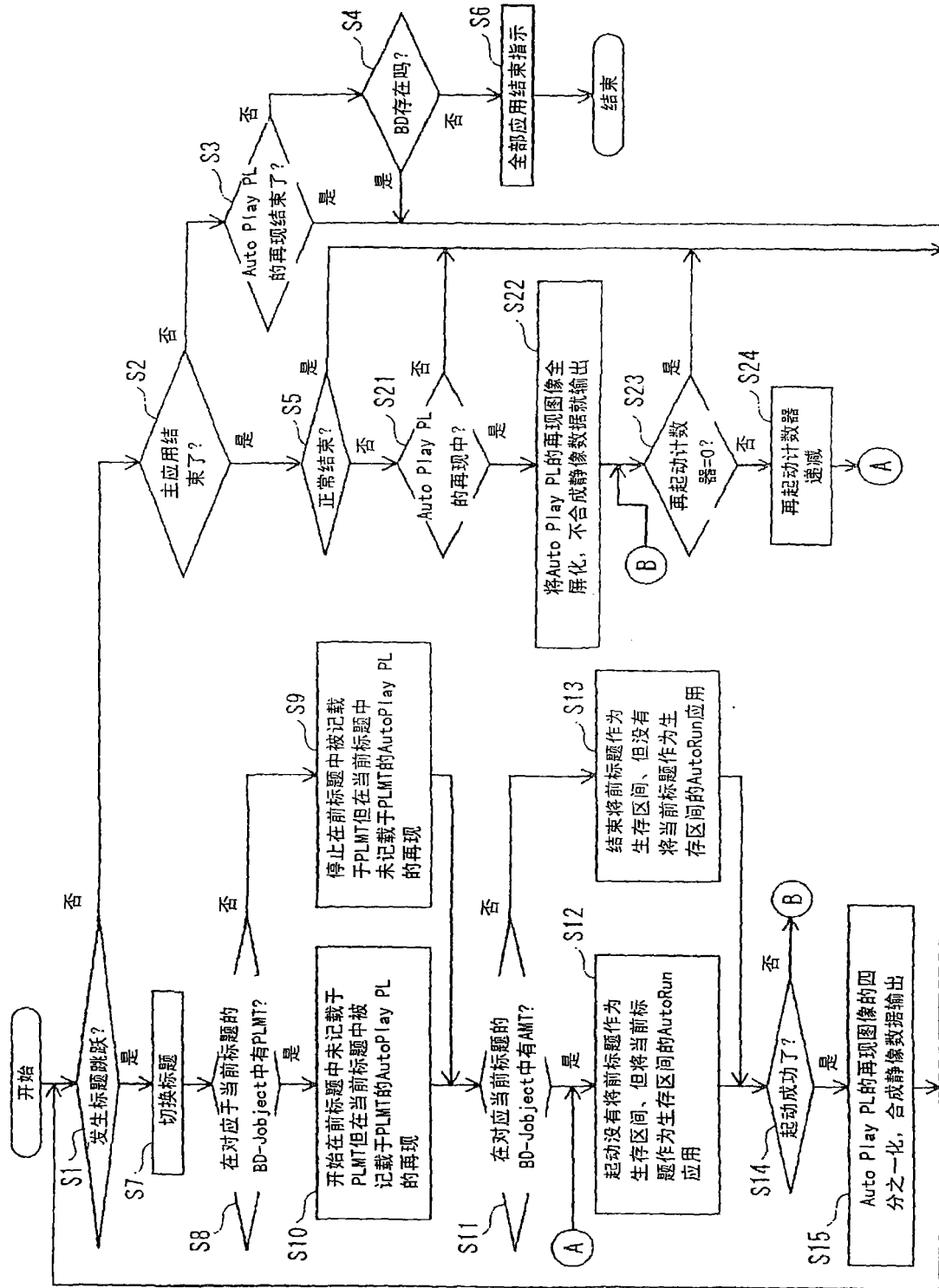


图 30

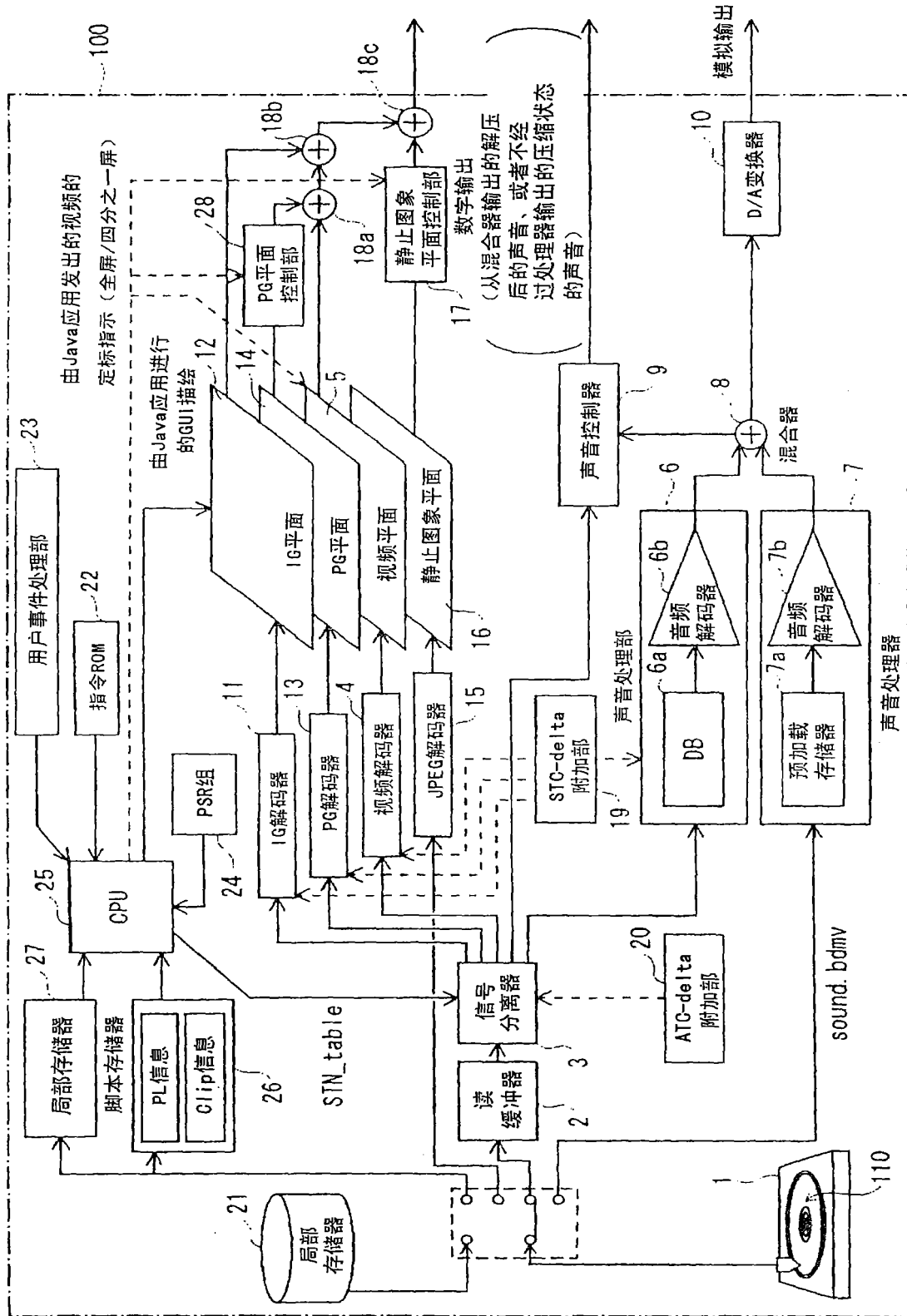


图 31

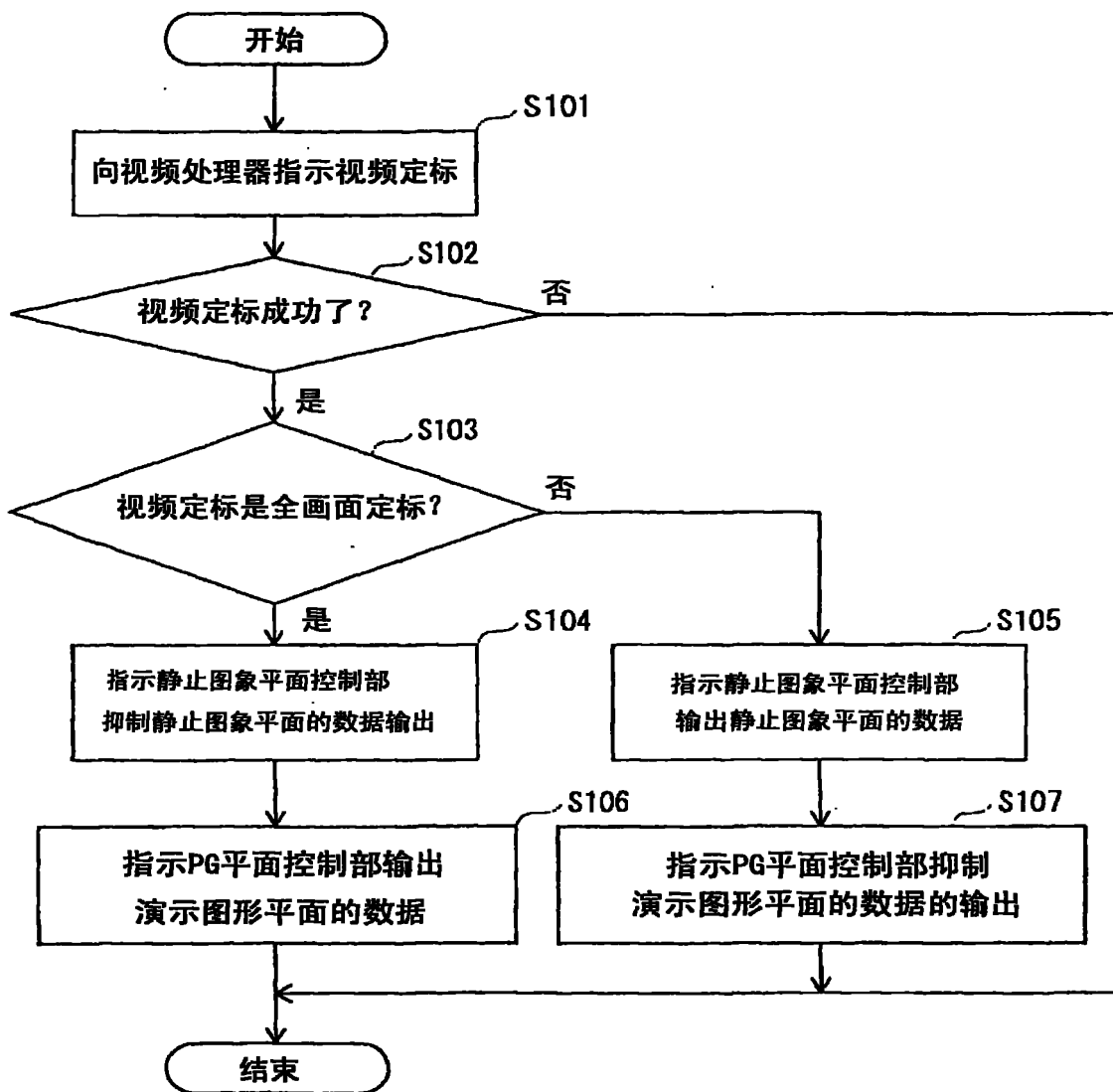


图 32