

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200610142906.2

[51] Int. Cl.

H04N 9/82 (2006.01)
G11B 27/034 (2006.01)
G11B 27/30 (2006.01)
G11B 27/32 (2006.01)
G11B 27/34 (2006.01)

[43] 公开日 2007年8月29日

[11] 公开号 CN 101026774A

[22] 申请日 2004.6.29

[21] 申请号 200610142906.2

分案原申请号 200480018700.8

[30] 优先权

[32] 2003.6.30 [33] US [31] 60/483,228

[71] 申请人 松下电器产业株式会社

地址 日本大阪府

[72] 发明人 矢羽田洋 约瑟夫·麦克罗森

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

代理人 王 英

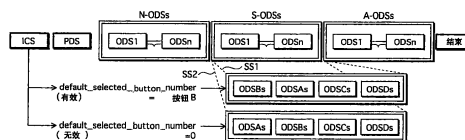
权利要求书 6 页 说明书 60 页 附图 54 页

[54] 发明名称

记录介质、再现装置、记录方法、程序和再现方法

[57] 摘要

一种包含 AV 剪辑的 BD - ROM, 所述 AV 剪辑是通过复用运动图像流和图形流而获得的。所述图形流表示要与所述运动图像流相结合以供显示的交互式屏幕, 并且包含三个按钮状态组。在该交互式屏幕上, 设置了多个按钮, 每个按钮根据用户操作从正常状态变为选中状态, 然后从选中状态变为活动状态。在所述图形流中的三个按钮状态组中, 第一组 (N - ODS) 包括多个表示按钮正常状态的图形数据; 第二组 (S - ODS) 包括多个表示按钮选中状态的图形数据; 第三组 (A - ODS) 包括多个表示按钮活动状态的图形数据。



1、一种记录介质，包括：

图形流，所述图形流表现要与运动图像相重叠的交互式显示，所述交互式显示包括多个图形按钮材料，其中：

所述图形流包括多个图形数据集合和控制信息，所述控制信息控制每个图形按钮材料的状态，每个图形按钮材料至少具有正常状态、选中状态和活动状态；以及

所述多个图形数据集合包括第一图形数据集合、第二图形数据集合和第三图形数据集合，其中，所述第一图形数据集合形成用于描绘所述正常状态的一组图形数据，所述第二图形数据集合形成用于描绘所述选中状态的一组图形数据，所述第三图形数据集合形成用于描绘所述活动状态的一组图形数据；

所述第一、第二和第三图形数据集合按顺序排列；

所述控制信息包括用于指示将所述图形按钮材料之一作为在所述交互式显示的初始显示时要以选中状态显示的默认选中按钮的信息；以及

用于描绘所述默认选中按钮的所述选中状态的所述图形数据排列在所述第二图形数据集合的开始处。

2、根据权利要求 1 所述的记录介质，还包括播放列表信息，其中：

所述播放列表信息包括主路径信息和子路径信息；

所述主路径信息将视频流指示为主流，并定义所述主流的再现部分；

所述子路径信息将所述图形流指示为与所述主流同步的子流，定义所述子流的再现部分，并且包括再现信息；

所述再现信息指示所述主流的再现时间轴上的同步点；以及

所述交互式显示与所述主流的所述再现部分中的所述视频流的图像相重叠地加以表现。

3、根据权利要求2所述的记录介质，还包括只读光盘，其中：
所述视频流记录在所述只读光盘上；以及
所述图形流和所述播放列表信息记录在所述记录介质上，所述记录介质是可重写的盘。

4、一种用于再现视频流和图形流的再现装置，所述再现装置包括：

图形解码器，用于对所述图形流进行解码，所述图形流表现要与运动图像相重叠的交互式显示，所述交互式显示包括多个图形按钮材料，其中：

所述图形流包括多个图形数据集合和控制信息，所述控制信息控制每个图形按钮材料的状态，每个图形按钮材料至少具有正常状态、选中状态和活动状态；以及

所述多个图形数据集合包括第一图形数据集合、第二图形数据集合和第三图形数据集合，其中，所述第一图形数据集合形成用于描绘所述正常状态的一组图形数据，所述第二图形数据集合形成用于描绘所述选中状态的一组图形数据，所述第三图形数据集合形成用于描绘所述活动状态的一组图形数据；

所述第一、第二和第三图形数据集合按顺序排列；

所述控制信息包括用于指示将所述图形按钮材料之一作为在所述交互式显示的初始显示时要以选中状态显示的默认选中按钮的信息；以及

用于描绘所述默认选中按钮的所述选中状态的所述图形数据排列在所述第二图形数据集合的开始处；以及

所述图形解码器包括：

图形处理器，用于对所述图形数据进行解码；

对象缓存器，用于存储通过所述解码而获得的解压缩图形数据；

图形平面，用于存储要与所述运动图像相重叠的所述解压缩

图形数据中的至少一些；以及

图形控制器，用于当所述图形处理器对排列在所述第二图形数据集合的开始处的图形数据进行解码时将所述解压缩图形数据写入所述图形平面。

5、根据权利要求 4 所述的再现装置，还包括用于对所述视频流进行解码并读取记录介质上记录的播放列表信息的视频解码器，其中：

所述播放列表信息包括主路径信息和子路径信息；

所述主路径信息将视频流指示为主流，并定义所述主流的再现部分，所述视频流包括多张图像；

所述子路径信息将所述图形流指示为与所述主流同步的子流，定义所述子流的再现部分，并且包括再现信息；

所述再现信息指示所述主流的再现时间轴上的同步点；以及

所述交互式显示与所述主流的所述再现部分中的所述视频流的图像相重叠地加以表现。

6、根据权利要求 5 所述的再现装置，其中：

所述视频流记录在只读光盘上；以及

所述图形流和所述播放列表信息记录在所述记录介质上，所述记录介质是可重写的盘。

7、一种用于向记录介质进行记录的记录方法，所述方法包括：

创建应用数据；以及

将所创建的应用数据记录到所述记录介质中；其中：

所述应用数据包括图形流；

所述图形流表现要与运动图像相重叠的交互式显示，所述交互式显示包括多个图形按钮材料；

所述图形流包括多个图形数据集合和控制信息，所述控制信息控制每个图形按钮材料的状态，每个图形按钮材料至少具有正常状态、

选中状态和活动状态；以及

所述多个图形数据集合包括第一图形数据集合、第二图形数据集合和第三图形数据集合，其中，所述第一图形数据集合形成用于描绘所述正常状态的一组图形数据，所述第二图形数据集合形成用于描绘所述选中状态的一组图形数据，所述第三图形数据集合形成用于描绘所述活动状态的一组图形数据；

所述第一、第二和第三图形数据集合按顺序排列；

所述控制信息包括用于指示将所述图形按钮材料之一作为在所述交互式显示的初始显示时要以选中状态显示的默认选中按钮的信息；以及

用于描绘所述默认选中按钮的所述选中状态的所述图形数据排列在所述第二图形数据集合的开始处。

8、一种包含在计算机可读介质上用于使计算机再现图形流的程序，所述程序包括使所述计算机执行以下步骤的代码：

对所述图形流进行解码；以及

将交互式显示与运动图像相重叠地进行显示，所述交互式显示包括多个图形按钮材料；

其中：

所述图形流包括多个图形数据集合和控制信息，所述控制信息控制每个图形按钮材料的状态，每个图形按钮材料至少具有正常状态、选中状态和活动状态；以及

所述多个图形数据集合包括第一图形数据集合、第二图形数据集合和第三图形数据集合，其中，所述第一图形数据集合形成用于描绘所述正常状态的一组图形数据，所述第二图形数据集合形成用于描绘所述选中状态的一组图形数据，所述第三图形数据集合形成用于描绘所述活动状态的一组图形数据；

所述第一、第二和第三图形数据集合按顺序排列；

所述控制信息包括用于指示将所述图形按钮材料之一作为在所述交互式显示的初始显示时要以选中状态显示的默认选中按钮的信

息；以及

用于描绘所述默认选中按钮的所述选中状态的所述图形数据排列在所述第二图形数据集合的开始处；以及

所述计算机包括：

图形处理器，用于对所述图形数据进行解码；

对象缓存器，用于存储通过所述解码而获得的解压缩图形数据；以及

图形平面，用于存储要与所述运动图像相重叠的所述解压缩图形数据中的至少一些；以及

其中，当所述图形处理器对排列在所述第二图形数据集合的开始处的图形数据进行解码时，将所述解压缩图形数据写入所述图形平面。

9、一种用于再现图形流的方法，所述图形流表现要与运动图像相重叠的交互式显示，所述交互式显示包括多个图形按钮材料，所述再现方法包括：

对视频流进行解码，其中：

所述图形流包括多个图形数据集合和控制信息，所述控制信息控制每个图形按钮材料的状态，每个图形按钮材料至少具有正常状态、选中状态和活动状态；以及

所述多个图形数据集合包括第一图形数据集合、第二图形数据集合和第三图形数据集合，其中，所述第一图形数据集合形成用于描绘所述正常状态的一组图形数据，所述第二图形数据集合形成用于描绘所述选中状态的一组图形数据，所述第三图形数据集合形成用于描绘所述活动状态的一组图形数据；

所述第一、第二和第三图形数据集合按顺序排列；

所述控制信息包括用于指示将所述图形按钮材料之一作为在所述交互式显示的初始显示时要以选中状态显示的默认选中按钮的信息；以及

用于描绘所述默认选中按钮的所述选中状态的所述图形数据排

列在所述第二图形数据集合的开始处；以及

所述解码包括：

对所述图形数据进行解码；

把通过所述解码而获得的解压缩图形数据存储到对象缓存器中；

把要与所述运动图像相重叠的所述解压缩图形数据中的至少一些存储到图形平面上；以及

当所述图形处理器对排列在所述第二图形数据集合的开始处的图形数据进行解码时，将所述解压缩图形数据写入所述图形平面。

记录介质、再现装置、记录方法、程序和再现方法

本申请是申请日为2004年6月29日、申请号为200480018700.8、名为“记录介质、再现装置、记录方法、程序和再现方法”的中国发明专利申请的分案申请。

发明领域

本发明涉及一种诸如BD-ROM之类的记录介质和一种再现装置。本发明尤其涉及一种交互式控制技术，用于将由多张图片构成的运动图像与由多个按钮构成的交互式显示进行组合，以及根据针对所述按钮的用户操作来实现再现控制。

技术背景

在接收关于选中待再现标题和章节、回答测验问题等的用户操作时，上面描述的交互式控制是再现装置的一项必备功能，并且，将其实现在DVD再现装置上是众所周知的。DVD再现装置的交互式控制是OSD（屏幕显示）技术的一种应用，通过它，在屏幕上显示按钮（即图形所附加的字符串），并根据用户操作来改变按钮边框的颜色。这样的颜色变化使得用户能够直观地理解交互式显示上的哪个按钮处于选中状态。

DVD上的交互式控制的缺点在于：几乎没有任何娱乐的因素。具体而言，在DVD交互式控制操作期间，用户很难觉得它有趣。鉴于这些批评，在BD-ROM（蓝光盘预记录格式）的交互式显示上尝试按钮的动画显示，BD-ROM的标准化还在进行中。更具体地讲，在BD-ROM呈现的交互式显示中，每个按钮以原始动画的形式进行呈现，并且与电影场景重叠，电影场景是主视频图像。此外，该动画显示的内容根据用户操作而改变。当该动画显示用于表示出现在电影作品上的人物时，用户可以通过他针对该交互式显示的操作来改变人物的表情和动作。通过这种动画创造的交互式显示连小孩子都会觉得有趣。

但是，按钮的动画显示会导致很大的解码负荷量，因此存在一个问题：在实现交互式显示的初始显示之前的等待时间变得很长。例如，

假设图 1 所示的交互式显示与电影作品的情景相重叠。该交互式显示有四个按钮，其中的每个按钮都有三个状态：正常状态、选中状态和活动状态。此外，假设一种情况，其中，以 2-3 秒的动画显示来呈现按钮的每个状态。即使一页图形数据在一个图像信号中以 5 帧的间隔显示，实现 2-3 秒的动画显示也需要大约 30 页的图形数据。不止这些，由于一个按钮有三个状态（正常状态、选中状态和活动状态），所以，总共将有 90 页（ 3×30 ）的图形数据。为了在交互式显示上设置四个按钮，需要对 360 页（ 4×90 ）进行解码，这是大量的图形数据。一页图形数据的解码负荷是很轻的。但是，为了对数百页的图形数据进行解码，需要花费大约几十秒。即便目的是使交互式显示有趣，让用户等待几十秒来显示各交互式显示也有些太过分，理应受到用户的批评。

发明内容

本发明的目的是提供一种记录介质，其利用动画实现交互式显示，而没有延迟。

为了实现上述目的，本发明提供了一种具有图形流的记录介质，其中：所述图形流表示将要与由多个图片构成的运动图像相重叠的交互式显示，所述交互式显示包括至少一个图形按钮材料；所述图形流包括多个图形数据，所述图形数据被分组在多个状态集合之下；以及，所述状态集合分别对应于所述至少一个按钮材料的不同状态，并且按顺序排列。

在上述结构下，假设有 360 页的图形数据用于呈现动画，并且每个按钮材料有三个状态。然后，将所述图形数据在三个状态集合之下进行分组（例如， $120+120+120$ 页）。接下来，设置每个状态集合，从而使得，在该动画中较早出现的集合被置于离头部较近，而较迟出现的集合被置于离头部较远。因此，在再现过程中，将较早出现的按钮状态集合先加载到再现装置中，而将较迟出现的按钮状态集合的加载推迟。这样，一旦整个图形数据的大约 $1/3-2/3$ 完成读取/解码，初始显示的准备工作的准备工作也已就绪，即便不是所有 360 页的图形数据都完成

解码。

这进一步意味着，在有大量的待读取/解码的图形数据的情况下，不会延迟初始显示的执行。据此，可以利用动画执行交互式显示，而没有延迟。

附图简述

图 1 示出了由动画构成的交互式显示；

图 2A 示出了使用根据本发明的记录介质的一个示例；

图 2B 示出了遥控器 400 上的按键设置，通过遥控器 400 可以接收针对交互式显示的用户操作；

图 3 是 BD-ROM 结构的示意图；

图 4 是 AV 剪辑结构的示意图；

图 5 是剪辑信息内部结构的示意图；

图 6 是播放列表信息（PL 信息）内部结构的示意图；

图 7 是借助于 PL 信息的间接引用的示意图；

图 8A 是交互式图形流的结构示意图；

图 8B 是 ICS 和 ODS 的内部结构的示意图；

图 9 是由各种功能段构成的逻辑结构的示意图；

图 10A 是用于定义图形对象的 ODS 的数据结构的示意图；

图 10B 是 PDS 的数据结构的示意图；

图 11 是交互式创作段（Interactive Composition Segment）的数据结构的示意图；

图 12 是 DS_n 中的多个 ODS 和 ICS 之间的关系示意图；

图 13 是在任意图像数据 pt1 的显示时机处的画面创作的示意图；

图 14 是 ICS 中的按钮信息的设定示例的示意图；

图 15 是按钮 A 到 D 的状态转变的一个示例的示意图；

图 16 是 ODS 11、21、31 和 41 的设计图像的一个示例的示意图；

图 17 是按钮 A 的 ODS 11-19 的设计图像的一个示例的示意图；

图 18 是包括在 DS 中的 ICS 和 ODS 的一个示例的示意图；

图 19 是属于一个显示集合的 ODS 的顺序以及按钮状态集合的示意图；

图 20 是图 19 的按钮状态集合所设置在的交互式显示的状态转变的示意图；

图 21 是一个显示集合中的 ODS 的顺序的示意图；

图 22 是 S-ODSD 中的 ODS 的顺序在当 default_selected_button_number 为“0”时和当它为“按钮 B”时的情形之间的差异的示意图；

图 23A 和图 23B 是在 N-ODS 包括构成按钮 A 到 D 的多个 ODS 并且 S-ODS 包括构成按钮 A 到 D 的多个 ODS 的情况下用于表示 $\sum \text{SIZE}(\text{DS}_n[\text{ICS.BUTTON}[i]])$ 的具体值的示意图；

图 24 是借助于 ICS 的同步显示时机的示意图；

图 25 是在多个 ODS 合成交互式显示的初始显示并且 default_selected_button_number 有效的情况下如何设定 DTS 和 PTS 的示意图；

图 26 是在多个 ODS 合成交互式显示的初始显示并且默认选中按钮无效的情况下如何设定 DTS 和 PTS 的示意图；

图 27 是根据本发明的再现装置的内部结构的示意图；

图 28 是与图形平面 8 相对照的对象缓冲器 15 的存储内容的示意图；

图 29 是在初始显示时由图形控制器 17 执行的处理的示意图；

图 30 是在根据第一个用户动作（右移）的交互式显示更新时图形控制器 17 的处理的示意图；

图 31 是在根据第一个用户动作（下移）的交互式显示更新时图形控制器 17 的处理的示意图；

图 32 是在根据第一个用户动作（激活）的交互式显示更新时图

形控制器 17 的处理的示意图；

图 33 是由再现装置执行的流水线处理的时序图；

图 34 是在默认选中按钮动态改变的情况下由再现装置执行的流水线处理的时序图；

图 35 是由控制单元 20 实现 LinkPL 功能的执行程序的流程图；

图 36 是段加载处理程序的流程图；

图 37 是一个复用示例的示意图；

图 38 是如何将 DS 10 加载到再现装置的编码数据缓冲器 13 中的示意图；

图 39 示出了执行正常再现的情况；

图 40 是当如图 39 所示执行正常再现时如何执行 DS 1、DS 10 和 DS 20 的加载的示意图；

图 41 的流程图示出了与图形控制器 17 的处理中的主程序相对应的处理；

图 42 的流程图示出了借助于时戳执行的同步控制的处理程序；

图 43 的流程图示出了针对图形平面 8 的写处理程序；

图 44 的流程图示出了如何自动激活默认选中按钮的处理程序；

图 45 的流程图示出了用于动画显示的处理程序；

图 46 的流程图示出了 UO 处理的处理程序；

图 47 的流程图示出了改变用于当前按钮的处理的程序；

图 48 的流程图示出了数字输入处理的程序；

图 49 是再现装置中基于 PDS 中 DTS 和的 PTS 的流水线的示意图；

图 50 是在再现装置的流水线处理期间 END 的意思的示意图；

图 51 的流程图示出了与第二实施例相关的 BD-ROM 的生产方法；

图 52 是第三个实施例涉及的 PL 信息的内容结构的示意图；

图 53 是根据“sync_PlayItem_id、syn_start_PTS_of_PlayItem”子 PlayItem 的同步的示意图；

图 54 是通过 HD 上的播放列表内的“Clip_Information_file_name”的文件识别。

具体实施方式

（第一个实施例）

下面描述与本发明相关的记录介质的一个实施例。在根据本发明的记录介质的实施例中，下面首先介绍如何使用该记录介质的一个示例。图 2A 是使用根据本发明的记录介质的示意图。在图 2A 中，本发明的记录介质是 BD-ROM 100。BD-ROM 100 用于为由再现装置 200、电视机 300 和遥控器 400 组成的家庭影院系统提供电影作品。在它们当中，遥控器 400 用于从用户接收操作，以改变交互式显示的状态，并且与本发明的记录介质密切相关。图 2B 示出了遥控器 400 的按键，针对交互式显示的用户操作是通过遥控器 400 接收的。如该图中所示，遥控器 400 包括上移键、下移键、右移键和左移键。这里，交互式显示上的按钮具有 3 种状态：正常状态；选中状态和活动状态。上移键、下移键、右移键和左移键用于接收用户的操作，以便将按钮的状态改变为正常状态→选中状态→活动状态。正常状态是仅提供显示的状态，而选中状态是根据用户操作给出聚焦（focus）但尚未收到确认的状态。活动状态是收到确认的状态。当交互式显示中的按钮处于选中状态中时，上移键用于将所选中按钮上方的按钮设定到选中状态。下移键用于将所选中按钮下方的按钮设定到选中状态。右移键用于将所选中按钮右侧的按钮设定到选中状态，而左移键用于将所选中按钮左侧的按钮设定到选中状态。

激活键用于将所选中按钮设定到活动状态（即，激活）。数字键“0”到“9”用于将对应数字所分配到的按钮设定到选中状态。“+”

10”键用于接收操作，以便将10加到已输入的数值中。在这里，应当注意的是，“0”键和“+10”键都用于接收不小于10个数字的数值的输入。因此，它们中的任何一个对于遥控器400都是足够的。

到此为止，已经描述了如何使用本发明的记录介质。

接下来介绍在本发明的记录介质的多个实施例中的生产示例。本发明的记录介质是通过改进BD-ROM的应用层而生产的。图3是BD-ROM结构的示意图。

在该图中，图中的第四行示出了BD-ROM，第三行中示出了BD-ROM上的轨道。该轨道实际上是该盘上的螺旋形，但图中被显示成在纵向延伸的线。该轨道包括导入区、容量区和导出区。该图中的容量区具有物理层、文件系统层和应用层。在该图的第一行中，使用目录结构的形式示出了BD-ROM的应用格式。正如该图中所示出的那样，该BD-ROM在根目录下有目录BDMV，该BDMV目录包括诸如XXX.M2TS，XXX.CLPI，和YYY.MPLS之类的文件。通过形成上述应用格式，可以生产根据本发明的记录介质。如果每一类型具有一个以上的文件，则优选在BDMV下提供名为STREAM、CLIPINF和PLAYLIST的三个目录，以便将具有相同扩展名的文件存储在一个目录中。具体而言，人们希望在STREAM中存储具有扩展名M2TS的文件，在CLIPINF中存储具有扩展名CLPI的文件，在PLAYLIST中存储具有扩展名MPLS的文件。

下面说明应用格式中的各个文件。首先，将给出有关AV剪辑（XXX.M2TS）的说明。

AV剪辑（XXX.M2TS）是通过复用—个视频流、至少—个音频流和—个呈现图形流以及—个交互图形流而获得的数字流，格式为MPEG-TS格式（TS表示传输流）。视频流表示电影的运动图像，音频流表示电影的声音，呈现图形流表示电影的字幕，交互图形流表示针对菜单的动态再现控制程序。图4是AV剪辑结构的示意图。

以下面的方式来构造 AV 剪辑（中间行）。将由多个视频帧（图像 pj1、pj2 和 pj3）组成的视频流和由多个音频帧（该图的最顶行）组成的音频流分别转换成 PES 数据包序列（从该图最顶行向下数第二行），然后再转换成 TS 数据包序列（从该图最顶行向下数第三行）。然后，将呈现图形流和交互图形流（该图的最底行）转换成 PES 数据包序列（从该图最底行向上数第二行），然后再转换成 TS 数据包序列（从该图最底行向上数第三行）。复用这些 TS 数据包序列，以获得 AV 剪辑。

以上述方式构成的 AV 剪辑被分为一个以上的块，与普通计算机文件一样，然后将其存储到 BD-ROM 中的区域中。该 AV 剪辑由一个或多个访问单元组成。每个访问单元都起到随机访问单元和以内部 (I) 图像开始的解码单元的作用。

剪辑信息 (XXX.CLPI) 是 AV 剪辑的管理信息。图 5 是剪辑信息的内部结构的示意图。由于 AV 剪辑是通过复用视频流和音频流而获得的，并且 AV 剪辑具有分别起到随机访问单元作用的访问单元，所以，剪辑信息管理的具体项目是：视频流和音频流中的每一个的属性；以及，随机访问点在 AV 剪辑中存在的位置。

在该图中，虚引线帮助阐述剪辑信息的结构。如引线 hn1 所示，剪辑信息 (XXX.CLPI) 包括：有关视频流和音频流的“属性信息”；以及“Ep_map”，其作为对访问单元进行随机访问时所使用的引用表。

如引线 hn2 所示，属性信息（“属性”）包括：“视频属性信息”，作为有关视频流的属性信息；“数量”，表明属性信息集合的数量；以及“音频属性信息 #1 至 #m”，其分别表示要被复用到 AV 剪辑上的多个音频流中的相应音频流的属性信息。如引线 hn3 所示，视频属性信息表示该视频流是采用何种压缩方法压缩的（编码）以及构成该视频流的各图像数据集合的分辨率（Resolution）、纵横比（Aspect）和帧速率（Framerate）。

如引线 hn4 所示，音频属性信息 #1 至 #m（有关音频流的属性信息）表示该音频流是采用何种压缩方法压缩的（编码）以及该音频流的信道（Ch.）、该音频流对应的语言（Lang）和该音频流的采样频率。

EP-map 是一张引用表，其使用时间信息对多个随机访问点的地址执行间接引用。如引线 hn5 所示，EP-map 包括：多组条目信息集合（访问单元#1 条目、访问单元#2 条目、访问单元#3 条目…）和条目编号（Number）。如引线 hn6 所示，每个条目都表明对应访问单元的再现开始时间和相关联的访问单元的地址（应当注意的是，可以在访问单元中输入第一 I 图像的尺寸（I-尺寸））。位于访问单元开始处的图像数据的时戳（呈现时戳）表示访问单元的再现开始时间。另外，相应 TS 数据包的序列号（SPN（源数据包编号））表示访问单元的地址。在编码中，采用变长编码压缩方法。因此，即使包括 GOP 的各访问单元之间有尺寸和再现时间的变化，也可能通过参考访问单元的条目，从任何再现时间对与目的再现时间相对应的访问单元的图像数据执行随机访问。

应当注意的是，XXX.CLPI 中的文件名 XXX 与剪辑信息所匹配的 AV 剪辑的名称相同。例如，在该图中，AV 剪辑的文件名是 XXX，因此该文件名对应于 AV 剪辑（XXX.M2TS）。接下来说明播放列表信息（PL 信息）。

YYY.MPLS（播放列表信息）是一张表，构成作为再现路径信息的播放列表，并且包括多条播放项目（PlayItem）信息（播放项目信息 #1、#2、#3…#n）和播放项目信息数（Number）。图 6 是播放列表信息的内部结构的示意图。一条播放项目信息定义了构成播放列表的一个或多个逻辑再现部分。图中的引线 hs1 帮助阐述了一条播放项目信息的结构。正如该引线所示，该播放项目信息包括：“Clip_Information_file_name”，用于表示再现部分的入点和出点所属

的 AV 剪辑的再现部分信息的文件名；“Clip_codec_identifier”，用于表示对 AV 剪辑进行编码所使用的编码方法；“IN-time”，作为表示再现部分开始点的时间信息；以及“OUT-time”，作为表示再现部分结束点的时间信息。

播放项目信息的特征在于其符号表示方法。具体而言，在播放项目信息中，将 EP-map 用作为引用表，借助于时间以间接的方式来定义再现部分。图 7 是表示这种间接引用的示意图。在该图中，AV 剪辑包括多个访问单元。剪辑信息的 EP-map 指定访问单元的地址，如箭头 ay1、2、3 和 4 所示。箭头 jy1、2、3 和 4 示意性地示出了如何用多条播放项目信息来引用访问单元。换言之，借助于所述多条播放项目信息（即箭头 jy1、2、3 和 4）所作的引用能够经由 EP-map 指定 AV 剪辑内访问单元的地址，并且，还可被视为借助于 EP-map 并使用时间概念的间接引用。

BD-ROM 上的再现部分被称为“播放项目”，它包括播放项目信息、剪辑信息和 AV 剪辑的组合。BD-ROM 上的逻辑再现单元被称为“播放列表”（缩写成 PL），它包括 PL 信息、剪辑信息和 AV 剪辑组合。在 BD-ROM 中记录的电影作品是由这种逻辑再现单元（即 PL）来构造的。因此，通过定义表示具有某特征的场景的 PL，可以很容易地产生另一由具有该特征的所有情景构成的电影作品。

因为 BD-ROM 上记录的电影作品具有上述逻辑结构，所以，可以高效地将构成一部电影作品场景的 AV 剪辑引用到另一部电影作品中。

接下来将描述交互式图形流。

图 8A 是交互式图形流的结构示意图。第一行表示构成 AV 剪辑的 TS 数据包序列。第二行表示构成图形流的 PES 数据包序列。从包括预定 PID 的第一行的 TS 数据包中提取出有效载荷，然后将所提取的有效载荷链接在一起，从而构造出第二行中的 PES 数据包序列。

应当注意的是，呈现图形流并不是本发明的重点，因此这里不再赘述。

第三行表示图形流的结构。该图形流由下面的功能段组成：ICS（交互合成段）、PDS（调色板定义段）、ODS（对象定义段）和END（显示集合段的结束）。在这些功能段中，ICS 被称为显示合成段，而 PDS、ODS 和 END 分别被称为定义段。PES 数据包和各功能段一一对应，或者，一个 PES 数据包对应于多个功能段。换言之，在将一个功能段转换为一个 PES 数据包之后，或者，将其划分为多个片断并转换为一个以上的 PES 数据包之后，将其记录到 BD-ROM 中。

图 8B 是通过转换功能段而获得的 PES 数据包的示意图。如图 8B 所示，PES 数据包由包头和有效载荷组成，其中有效载荷是功能段的实体内容。另外，包头包括与该功能段相对应的 DTS 和 PTS。下文中，包括在包头中的 DTS 和 PTS 被称为该功能段的 DTS 和 PTS。

这些多种功能段构成了图 9 所示的逻辑结构。在该图中，第三行表示功能段，第二行表示显示集合，而第一行表示时元（Epoch）。

第二行是多个显示集合（简称为“DS”），其中的每一个显示集合对应于组成一幅图形画面的图形。图中的虚线表示第三行的功能段所属的 DS。如从上述内容可以理解，“ICS-PDS-ODS-END”功能段序列构成一个 DS。再现装置可以通过从 BD-ROM 中读取构成一个 DS 的一系列功能段，来构造出一幅图形画面。

第一行中的各个时元表示在该 AV 剪辑再现时间轴上具有存储器管理连续性的一个时段，并且对应于分配给该时段的一个数据集合。这里，所假定的存储器是用于存储一幅图形画面的图形平面，或者是用于存储处于解压缩状态的图形数据的对象缓冲器。如果声明图形平面或对象缓冲器具有存储器管理连续性，这意味着：在由该时元表示的时段中，图形平面或对象缓冲器中不会出现闪烁，并且，仅在所述图形平面内的预定矩形区域中执行图形的删除 / 重绘（这里，闪烁意味着清除所述平面或缓冲器中存储的所有内容）。该矩形区域的尺寸

(长度 / 宽度) 和位置在整个时元内都是固定的。只要在所述图形平面的固定区域中执行图形的删除 / 重绘, 就能保证无缝的再现。也就是说, 可以将时元视为再现时间轴上的一个单元, 其中能够保证无缝的再现。如果用户想要改变区域, 那么, 他必须定义该再现时间轴上的改变时间, 并且, 创建与该变化时间之后的时间相对应的新时元。在这种情况下, 这两个时元之间无法保证无缝的再现。

应当注意的是, “无缝的再现” 意味着: 图形的删除 / 重绘将在预定的视频帧数量之内完成。对于交互式图形流来说, 该视频流帧数量是 4 到 5 个帧。该视频帧的数量取决于固定区域与整个图形平面的比值以及在对象缓冲器和图形平面之间的传输速率。

在该图中, 虚线 hk1、2 表示第三行中的功能段属于哪个 Epoch。从图中可以理解的是, 一系列的 DS (即 Epoch Start、Acquisition Point 和 Normal Case) 构成了第一行的 Epoch。“Epoch Start (时元开始)”、“Acquisition Point (采集点)” 和 “Normal Case (正常情况)” 分别是 DS 的类型。应当注意的是, “Acquisition Point” 和 “Normal Case” 的顺序仅是示例性的, 也可以颠倒过来。

“Epoch Start” 是用于产生 “新显示” 的显示效果的一个 DS, 并且, 它表示新时元的开始。因此, “Epoch Start” 必须包括用于合成下一画面所需的所有功能段, 并且, 它在 AV 剪辑中的位置是随机访问目标 (例如电影作品的章节) 的目的地。

“Acquisition Point” 是用于产生 “显示更新” 的显示效果的一个显示集合, 并且与前一 “Epoch Start” 相关。Acquisition Point 分为两种类型: 复制和继承。复制是与前一 “Epoch Start” 完全相同的一个显示集合, 而 “继承” 是从前一 “Epoch Start” 继承功能段但却具有与前一 “Epoch Start” 不同按钮命令的显示集合。尽管并不表示时元的开始时间, 但 Acquisition Point DS 却包括用于合成下一画面所需的全部功能段。因此, 如果对 Acquisition Point DS 执行随机访问, 则可

以保证图形显示。换言之，Acquisition Point DS 使得能够在进行中的时元期间合成画面。

Acquisition Point 显示集合所加入的位置是随机访问目标。该位置可以通过时间搜索来指定。时间搜索这样一种操作：从用户接收时间输入（分和秒），然后，随机访问与输入时间相对应的时间。执行时间搜索的单位大致是 10 分或 10 秒，因此，时间搜索可以以 10 分钟时间间隔或 10 秒钟时间间隔指定再现点。通过在可由时间搜索指定的位置处提供 Acquisition Point，有助于在时间搜索处进行图形流的再现。

“Normal Case”是用于产生“显示更新”的显示效果的 DS，并且，仅包括与现有画面合成的不同之处。例如，假定 DS_v 的按钮与之前的 DS_u 具有相同的图像设计，但却与 DS_v 处于不同的状态控制之下。在这种情况下，该 DS_v 要么仅包括 ICS，要么仅包括 ICS 和 PDS，并且，将 DS_v 设置为 Normal Case DS。这样，就没有必要提供重叠的 ODS，因此有助于减少 BD-ROM 中的占用区域。应当注意的是，Normal Case DS 本身不构成画面，因为它仅表示不同之处。

通过将 GUI 部件放置到屏幕上，来创建由这些 DS 定义的交互式显示。通过声明某一 DS 具有交互性，这意味着 GUI 部件的状态可根据用户操作而改变。在该实施例中，这种 GUI 部件（是用户的操作目标）被称为“按钮”。按钮的状态包括“一般状态”、“选中状态”和“活动状态”，每一种状态都是由多个处于解压缩状态的图形合成的。表示按钮状态的每个解压图形被称为“图形对象”。用多个解压图形来表示各个按钮状态，其原因是出于动画显示的目的。

接下来说明定义段（ODS、PDS）。

“Object_Definition_Segment”是定义图形对象的信息。下面将说明该图形对象。BD-ROM 中记录的 AV 剪辑特征在于，其高质量的图像足以与高清晰度电视的图像媲美。因此，图形对象的分辨率很

高，为 1920 * 1080 像素，从而产生高清晰度。至于像素的颜色，将一个像素的索引值设定为 8 比特长度。这里，索引值包括红色差值分量 (Cr_value)、蓝色差值分量 (Cb_value)、亮度分量 (Y_value) 和透明级 (T_value)。通过这种设定，可以从 16777216 种颜色 (全色) 中选择任意 256 种颜色，从而将 256 种颜色设定为像素的颜色。

ODS 具有图 10A 所示的数据结构，用于定义图形对象。ODS 包括：“segment_type”，表示其是 ODS；“segment_length”，表示 ODS 数据长度；“object_id”，唯一地标识与该时元中的 ODS 相对应的图形对象；“object_version_number”，表示该时元中的 ODS 版本；“last_in_sequence_flag”；以及连续字节长度数据“object_data_fragment”，是图形对象的一部分或全部。

“object_id”唯一地标识与该时元中的 ODS 相对应的图形对象。然而，当通过由多个 ODS 定义的多个图形对象构造动画时，分配给这多个 ODS 中的每一个 ODS 的“object_id”将是序列号。

接下来描述“last_in_sequence_flag”和“object_data_fragment”。有一种情况下，构成一个按钮的解压图形不能存储到一个 ODS 中，这是因为对 PES 数据包的有效载荷的约束的原因。在这种情况下，将按钮命令的划分 (片断) 被设定为“object_data_fragment”。当将一个图形对象存储到多个 ODS 中时，除最后一个片断之外，所有的片断必须具有相同的尺寸。即，最后一个片断的尺寸小于其他片断的尺寸。存储这些片断的多个 ODS 将以相同顺序在 DS 中出现。图形对象的最后一部分是由具有“last_in_sequence_flag”的 ODS 来指定的。ODS 的上述数据结构采用的存储方法中，针对新的 PES 包的存储直到当前 PES 包变满为止才开始。但是也可以采用下面的存储方法，即，采用该方法，在各 PES 包中将存在自由空间。

“palette_definition_segment (PDS)”是定义用于颜色转换的调色板的信息。图 10B 示出了 PDS 的数据结构。如图 10B 所示，PDS

包括：“segment_type”，表示其是 PDS；“segment_type”，表示 PDS 数据长度；“palette_id”，唯一地标识包括在 PDS 中的调色板；“palette_version_number”，表示该时元中的 PDS 的版本；“palette_entry”，是各条目的有关信息。对于各条目来说，“palette_entry”表示红色差值分量（Cr_value）、蓝色差值分量（Cb_value）、亮度分量（Y_value）和透明级（T_value）。

接下来说明显示集合段的 END。

显示集合段的 END 是一条索引，表示显示集合传送的结束，并且，紧接在最后一个 ODS 之后。该显示集合段的 END 的内部结构包括：segment_type，表示该功能段是显示集合段的 END；segment_length，表示功能段的数据长度。这些都不需要具体的解释。因此，该图没有显示内部结构。

接下来将说明交互式合成段（ICS）。ICS 是构成交互式显示的功能段。ICS 具有图 11 所示的数据结构。如该图所示，ICS 包括：segment_type；segment_length；composition_number；composition_state；command_update_flag；composition_timeout_PTS；selection_timeout_PTS；UO_mask_table；animation_frame_rate_code；default_selected_button_number；default_activated_button_number；以及按钮信息集合（button_info（1）（2）（3））”。

“composition_number”表示是否在该 ICS 所属的 DS 中执行更新，其取值为 0 到 15。

“composition_state”表示以该特定 ICS 开始的 DS 是 Normal Case、Acquisition Point 还是 Epoch Start。

“command_update_flag”表示该 ICS 中的按钮命令与前面的 ICS 的按钮命令相比是否发生了变化。例如，当一个特定 ICS 所属的 DS 是 Acquisition Point 时，原则上，该 ICS 具有与紧接着的前一 ICS 相同的内容。然而，当将该 command_update_flag 设为 ON 时，可以在

该 ICS 中设置与前一 DS 不同的按钮命令。通过该标记，可以挪用前一图形对象，但是，当指示其变化时，该命令就变成有效的。

“compostion_timeout_PTS”示出了借助于按钮的交互式显示的结束时间。在该结束时间处，交互式显示的显示不再有效，因此不会被呈现。优选情况下，以运动图像数据的再现时间轴的时间精度描述 copostion_timeout_PTS。

“selection_timeout_PTS”描述了有效按钮选中时段的结束时间。在 selection_timeout_PTS 显示的时间处，激活由 default_activated_button_number 表示的按钮。selection_timeout_PTS 等于 compostion_timeout_PTS 表示的时间或更短。以视频帧的时间精度描述 selection_timeout_PTS。

“UO_mask_table”表示与该 ICS 相对应的显示集合中允许 / 不允许用户操作。当在该掩码字段中设定“不允许”时，针对该再现装置的任何用户操作都是无效的。

“animation_frame_rate_code”表示要应用于在动画中显示的按钮的帧速率。动画帧速率是视频帧速率除以该字段值而得到的。如果该字段值是 00，那么，在定义所有按钮图形对象的 ODS 中，只显示由 start_object_id_xxx 标识的 ODS，并且以非动画方式。

“default_selected_button_number”表示要在交互式显示开始时被默认设为选中状态的按钮编号。当该字段为 0 时，分配有再现装置的寄存器中存储的按钮编号的按钮被自动设为活动状态。如果该字段不是 0，则将该字段解释为标识相应的按钮。

“ default_activate_button_number ” 表示：当用户在该 selection_timeout_PTS 定义的时间内未将任何按钮设置成活动状态的情况下，将要自动设置成活动状态的按钮。当 deflaut_activated_button_number 表示“ FF ”时，在 selection_timeout_PTS 定义的时间处，自动选择当前处于选中状态的

按钮。当 `default_activated_button_number` 表示“00”时，不执行自动选择。当 `default_activated_button_number` 表示除“00”和“FF”之外的其他数字时，则将该字段解释为标识相应的按钮。

“按钮信息 (`button_info`)”是定义将要在交互式显示中重叠的各按钮的信息。在该图中，引线 `hp1` 帮助描述按钮信息 `i` 的内部结构，该按钮信息 `i` 是由 ICS 控制的第 `i` 个按钮 (按钮 `i`) 的有关信息。下面描述构成该按钮信息 `i` 的信息元素。

“`button_number`”是唯一地标识该 ICS 中的按钮 `i` 的数值。

“`numerically_selectable_flag`”是表示对于按钮 `i` 是否允许进行数值选择的标志。

“`auto_action_flag`”表示是否自动将按钮 `i` 设置为活动状态。当 `auto_action_flag` 被设为 ON (即比特值 1) 时，按钮 `i` 被设为活动状态而不是选中状态。当 `auto_action_flag` 被设为 OFF (即比特值 0) 时，即使实际选择了按钮 `i`，按钮 `i` 也仅仅处于选中状态。

“`object_horizontal_postion`”、“`object_vertical_postion`”分别表示交互式显示中按钮 `i` 的左上方像素的水平位置和垂直位置。

“`upper_button_nubmer`”表示当上移按键在按钮 `i` 处于选中状态期间被按下时处于选中状态的按钮编号，而不是按钮 `i`。如果在该字段中设置按钮 `i` 的编号，则忽略上移键的按压。

“`lower_button_number`”、“`left_button_number`”和“`right_button_number`”分别表示当上移键、左移键、右移键在按钮 `i` 处于选中状态期间被按下时处于选中状态的按钮编号，而不是按钮 `i`。当在该字段中设置按钮 `i` 的编号时，忽略这些键的按压。

当以动画来绘制处于正常状态的按钮 `i` 时，“`start_object_id_normal`”表示分配给构成动画的多个 ODS 的序列号的第一个编号。

当以动画来绘制处于正常状态的按钮 `i` 时，

“end_object_id_normal”表示作为分配给构成动画的多个 ODS 的序列号的 object_Id 的最后一个编号。如果该 end_object_id_normal 表示的 ID 与 start_object_id_normal 表示的 ID 相同，则与该 ID 标识的图形对象相对应的静止图像将是该按钮 i 的图像设计。

“repeated_normal_flag”表示对于正常状态下的按钮 i 是否重复地继续动画显示。

当以动画来绘制处于选中状态的按钮 i 时，“start_object_id_selected”表示分配给构成动画的多个 ODS 的序列号的第一个编号。如果 end_object_id_selected 表示的 ID 与该 start_object_id_selected 表示的 ID 相同，则与该 ID 标识的图形对象相对应的静止图像将是该按钮 i 的图像设计。

“end_object_id_selected”表示作为分配给构成动画的多个 ODS 的序列号的 object_Id 的最后一个编号。

“repeat_selected_flag”表示对于选中状态中的按钮 i 是否重复地继续动画显示。当 start_object_id_selected 与 nd_object_id_selected 的值相同时，在该字段中设置 00。

当以动画来绘制处于活动状态的按钮 i 时，“start_object_id_activated”表示分配给构成动画的多个 ODS 的序列号的第一个编号。

当以动画来绘制处于活动状态的按钮时，“end_object_id_activated”表示作为分配给构成动画的多个 ODS 的序列号的对象 id 的最后一个编号。

接下来将说明按钮命令。

“按钮命令 (button_command)”表示当按钮 i 进入活动状态时要执行的命令。

该按钮命令可以指示再现装置执行针对 PL 和播放项目的再现。指示再现装置执行针对 PL 和播放项目的再现的命令被称为“LinkPL

命令”。该命令可以根据第二自参数指定的位置，来启动由第一自变量指定的播放列表的再现。

格式：LinkPL（第一参数，第二参数）

所述第一参数借助于分配给播放列表的编号来指定要进行再现的 PL。第二参数借助于包括在 PL 中的播放项目、或包括在 PL 中的章节和标志来指定再现开始位置。

借助于 PlayItem 来指定再现开始位置的 LinkPL 函数被表示为“LinkPLatPlayItem()”。

借助于章节来指定再现开始位置的 LinkPL 函数被表示为“LinkPLatChapter()”。

借助于标志来指定再现开始位置的 LinkPL 函数被表示为“LinkPLatMark()”。

按钮命令用于向再现装置发出指示，以便执行状态获取和状态设置。64 个播放机状态寄存器（其设定值被称为 PSR）和 4096 个通用寄存器（其设定值被称为 GPR）表示再现装置的状态通过。所述按钮命令被分为下面的 (i) – (iv)，并对这些寄存器的值进行设定，以及从这些寄存器中获取值。

(i) “Get value of Player Status Register” 命令

格式：Get value of Player Status Register（参数）

在获取由参数指定的播放机状态寄存器的设定值时，使用该函数。

(ii) “Set value of Player Status Register” 命令

格式：Set value of Player Status Register（第一参数，第二参数）

在将由第二参数指定的值设定到由第一参数指定的播放机状态寄存器时，使用该函数。

(iii) “Get value of General Purpose Register” 命令

格式：Get value of General Purpose Register（参数）

在获取由该参数指定的通用寄存器的设定值时，使用该函数。

(iv) “Set value of General Purpose Register” 命令

格式：Get value of General Purpose Register (第一参数、第二参数)

在将由第二参数指定的值设定到由第一参数指定的通用寄存器时，使用该函数。

接下来将说明通过该 ICS 执行的交互式控制的一个具体示例。该具体示例采用图 12 中示出的 ODS 和 ICS。图 12 是表示 DS_n 中的 ODS 和 ICS 之间的关系的示意图。假设该 DS_n 包括 ODS 11-19、21-29、31-39 和 41-49。在这些 ODS 中，ODS 11-19 绘制了按钮 A 的各状态。ODS 21-29 绘制了按钮 B 的各状态。ODS 31-39 绘制了按钮 C 的各状态，ODS 41-49 绘制了按钮 D 的各状态（如图中的标记“）”所示）。另外，button_Info (1)、(2)、(3)、(4) 分别表示这些按钮 A 至按钮 D（参见该图中的箭头 bh1、2、3、4）的状态控制。

当借助于该 ICS 的控制的执行时机与任一图像数据 pt1（其在图 13 的图形中）的显示时机一致时，通过将由按钮 A 至 D 构成的交互式显示 tm1 重叠(gs1)到该图像数据 pt1 上来显示该 tm1 (gs2)。根据动画图像的内容来显示由多个按钮构成的交互式显示。因此，一个 ICS 可以通过使用按钮来实现逼真的效果。

图 14 示出了当执行按钮 A 到 D（图 15 所示）的状态变换时 ICS 的描述性示例。

在图 15 中，箭头 hh1 和 hh2 示意性地表示借助于按钮信息(1)的 neighbor_info()的状态转换。在按钮信息(1)的 neighbor_info()中，将 lower_button_number 设定为“按钮 C”，因此，如果在按钮 A 处于选中状态（图 15 的 up1）的同时出现下移键按压的 UO，则按钮 C 将处于选中状态（图 15 的 sj1）中。在按钮信息(1)的 neighbor_info()中，

将 `right_button_number` 设定为“按钮 B”，因此，如果在按钮 A 处于选中状态（图 15 的 `up2`）的同时出现下移键按压的 UO，则按钮 B 将处于选中状态（图 15 的 `sj2`）。

图 15 中的箭头 `hh3` 表示借助于按钮信息(3)的 `neighbor_info()` 进行的状态转换控制。在按钮信息(3)的 `neighbor_info()` 中，将 `upper_button_number` 设定为“按钮 A”，因此，如果在按钮 C 处于选中状态（`up3`）的同时出现上移键按压的 UO，则按钮 A 将返回到选中状态。

接下来将描述按钮 A 至 D 的图像设计。这里假定：ODS 11、21、31 和 41 具有如图 16 所示的图像设计，分配给按钮 A 的 ODS 11-19 具有如图 17 所示的图像设计。因为在该 ICS 的 `button_info(1)` 中的 `normal_state_info()` 中，`start_object_id_normal` 和 `end_object_id_normal` 指定了 ODS 11-13，所以，根据 ODS 11-13 在动画中呈现按钮 A 的正常状态。另外，在 `button_info(1)` 的 `selected_state_info()` 中，`start_object_id_selected` 和 `end_object_id_selected` 指定了 ODS 14-16，所以，根据 ODS 14-16 在动画中呈现按钮 A 的选中状态。当用户将按钮 A 输入到选中状态时，按钮 A 的图像设计将从根据 ODS 11-13 的设计改变到根据 ODS 14-16 的设计。这里，在 `normal_state_info()` 和 `selected_state_info()` 中，如果将 `repeat_normal_flag()` 和 `selected_state_info()` 设定为 1，则根据 ODS 11-13 的动画和根据 ODS 14-16 的动画将分别重复地继续，如图中“→ (A)”、“(A) →”和“→ (B)”、“(B) →”所示。

如果将动画中可呈现的多个 ODS 分配给按钮 A 至 D，并且在 ICS 中描述用于这些 ODS 的控制，则将会实现逼真的按钮状态控制（例如，根据各用户操作来改变人物的表情）。

接下来将描述 `numerically_selected_flag` 的应用。

图 18 是表示包括在 DS 中的 ICS 和 ODS 的一个示例的示意图。

在该图中，ODS 31-33 分别表示该图的顶行中示出的三个棒球运动员中对应的一个运动员的肖像、名字和队服号码。该 DS 所属的 ICS 包括三组按钮信息，其中，按钮信息（1）的 start_object_id 被设定为表示 ODS 31，按钮信息（2）的 start_object_id 被设定为表示 ODS 32，按钮信息（3）的 start_object_id 被设定为表示 ODS 33。另外，按钮信息（1）的按钮编号被设置为 99，按钮信息（2）的按钮编号被设置为 42，按钮信息（3）的按钮编号被设置为 94。此外，对于所有的按钮信息（1）-（3），numerically_selected_flag 被设定为 1。这意味着，与按钮信息（1）-（3）相对应的任何一个按钮都是可选择的。因此，如果用户通过遥控器 400 输入数值“99”，则对应于 Mr. Beginner's Luck 的按钮将被置于选中状态。可以通过顺序地按压“9”和“9”来接收数值“99”的输入。作为选择，也可通过“9”的一次按压与“+10”键的 9 次按压的组合来实现数值“99”的输入。当输入数值“42”时，用于 Mr. Careless Mistake 的按钮将被置于选中状态，并且当输入数值“94”时，用于 Mr. Dead Stock 的按钮将被置于选中状态。

当这些按钮信息（1）-（3）的 auto_action_flag 被设定为 1 时，这些条按钮信息（1）-（3）将进入活动状态，而不是进入选中状态，并且，按钮命令被包括在其中，即执行（LinkPL(PL#21)、LinkPL(PL#22)、LinkPL(PL#23)）。当包括在这些按钮信息中的按钮命令的链接目的地 PL#21、#22 和 #23 是相应运动员的打击 / 投掷场景时，然后，将再现这些打击 / 投掷场景，以响应与相应运动员队服号码相对应的数字编号的输入。由于可以使用熟知的队服号码直接选择按钮，所以，用户的可操作性将显著地得以改善。

接下来将描述显示集合中的 ODS 的顺序。如上所述，属于显示集合的各 ODS 由 ICS 来指定，以便显示按钮的一种状态。根据这样一种规范，即 ODS 表示那种按钮状态，判定显示集合中的 ODS 的顺

序。

更具体地讲，将显示集合中的 ODS 分为下面的集合：（1）表示正常状态的 ODS；（2）表示选中状态的 ODS；以及（3）表示活动状态的 ODS。表示特定按钮的一种状态的 ODS 集合被称为“按钮状态集合”。以“正常状态→选中状态→活动状态”的顺序来设置这些按钮状态集合。如上所示，根据 ODS 表示按钮的哪个状态来决定显示集合中的各 ODS 的次序。

图 19 是属于一个显示集合的 ODS 的顺序的示意图。在该图的第二行中，示出了该显示集合中的三个按钮状态集合。该图描绘的是：用于再现按钮的正常状态的 ODS 集合（用于正常状态的 ODS）；用于再现按钮的选中状态的 ODS 集合（用于选中状态的 ODS）；以及用于再现按钮的活动状态的 ODS 集合（用于活动状态的 ODS）。这些按钮状态集合的顺序如下：正常状态→选中状态→活动状态。该顺序的目的是：便于在早期读取构成交互式显示的最初显示的 ODS，并且在更新后推迟读取构成屏幕显示的 ODS。

图 19 的第一行表示图形对象 A_n 、 B_n 、 C_n 、 D_n 、 A_s 、 B_s 、 C_s 、 D_s 、 A_a 、 B_a 、 C_a 、 D_a ，它们将通过这些按钮状态集合来再现。分配给 A_n 、 B_n 、 C_n 、 D_n 的下标“n”表示对应按钮的正常状态。同样， A_s 、 B_s 、 C_s 、 D_s 的下标“s”表示对应按钮的选中状态， A_a 、 B_a 、 C_a 、 D_a 的下标“a”表示对应按钮的活动状态。图 19 的第二行示出了第一行的图形对象所属的按钮状态集合。应当注意的是，在该图中，为分别属于 N-ODS、S-ODS 和 A-ODS 的 ODS1 至 ODSn 集合分配彼此相同的编号，（即，存在属于 N-ODS 的 ODS1，同时存在属于 S-ODS 的 ODS1）。然而，各个集合彼此不同。这同样适用于下文中的各类似图。

图 20 是表示交互式显示的状态转换，在该交互式显示处设置有图 19 的按钮状态集合。

该图中的交互式显示具有多种状态，即“初始显示”、“通过第一次用户动作的更新显示”以及“通过第二次更新显示的更新显示”。该图中的箭头表示触发状态转换的用户动作。在该图中，四个按钮 A、B、C 和 D 分别具有三种状态“正常状态”、“选中状态”和“活动状态”。在这些状态中，初始显示所必需的是用于呈现正常状态的三个图形对象和用于呈现选中状态的一个图形对象。

即使仍然没有判定默认选中按钮并且不确定将按钮 A 到 D 中的哪一个置于选中状态，当对表示各个按钮的正常状态和选中状态的图形对象完成解码时，也可以实现初始显示。鉴于此，在该实施例中，以图 19 第二行所示的顺序“正常状态→选中状态→活动状态”来设置按钮状态集合。根据按钮状态集合的该设置，即使还没有完成构成活动状态的 ODS 的读取和解码时，也可以实现初始显示，并且，还可以缩短一个显示集合的读取开始和初始显示的完成之间的时间段。

接下来将说明图 16 和图 17 中所示的 ODS 应当被设置的顺序。图 21 是表示一个显示集合中的 ODS 的顺序的示意图。在该图中，用于正常状态的 ODS 包括 ODS 11-13、ODS 21-23、ODS 31-33、ODS 41-43。用于选中状态的 ODS 包括 ODS 14-16、ODS 24-26、ODS 34-36 和 ODS 44-46。最后，用于活动状态的 ODS 包括 ODS 17-19、ODS 27-29、ODS 37-39 和 ODS 47-49。ODS 11-13 用于呈现图 17 所示的人物的表情变化，ODS 21-23、ODS 31-33 和 ODS 41-43 也是如此。因此，通过将这些 ODS 设置在第一按钮状态集合中，即使在读取该显示集合期间也可以进行初始显示的准备。通过上述安排，可以没有延迟地执行加入了动画的交互式显示。

接下来将描述由多个按钮状态引用的 ODS 的顺序（多引用）。多引用（multi-reference）指的是：两个或多个 `normal_state_info`、`selected_state_info` 和 `activated_state_info` 指定一个 ODS 的 `object_Id`。通过这样的多引用，可以使用用于呈现一个按钮正常状态的图形对象

来呈现另一按钮选中状态。这意味着可以共享一个图形对象的设计图像。这种共享有助于减少 ODS 的数量。这里，将要进行多次引用的 ODS 应当属于哪一个按钮状态集合是存在问题的。

例如，假定通过一个 ODS 来绘制一个按钮的正常状态和另一个按钮的选中状态。那么，该 ODS 属于与正常状态相对应的按钮状态集合或与选中状态相对应的按钮状态集合是有问题。

原则上，仅将该 ODS 设置在与多个状态中最早出现的状态相对应的按钮状态集合中。

例如，当通过正常状态和选中状态多次引用一个 ODS 时，将该 ODS 设置在用于正常状态（N-ODS）的按钮状态集合中，而不是在用于选中状态（S-ODS）的按钮状态集合中。在一个不同的示例中，如果通过选中状态和活动状态多次引用 ODS，则将 ODS 设置在用于选中状态（S-ODS）的按钮状态集合中，而不是在用于活动状态（A-ODS）的按钮状态集合中。以这种方式，将多次引用的 ODS 仅设置在与最早出现的状态相对应的按钮状态集合中。

接下来将描述 S-ODS 中的 ODS 的顺序。在 S-ODS 中，哪个 ODS 先出现取决于默认选中按钮是静态确定的还是动态确定。静态确定的默认选中按钮是由 ICS 中的 `default_selected_button_number` 中的有效值（除 00 之外）集合指定的按钮。当将除 00 之外的有效值设定为默认选中按钮编号并且在 N-ODS 中不存在表示默认选中按钮的非 ODS 时，将表示默认选中按钮的 ODS 设置在 S-ODS 的开始处。

当 `default_selected_button_number` 表示 00 值时，默认被设置为选中状态的按钮将根据再现装置一侧的状态而动态地变化。

将 `default_selected_button_number` 设定为表示值 00 的情形包括：显示集合被复用到 AV 剪辑存在于多条再现路径的结合点处。例如，假定如下的情形：前面的再现路径分别为第一、第二和第三章节，并且，在所述结合点处的显示集合用于显示与第一、第二和第三章节相

对应的按钮。在这种情况下，默认通过 `default_selected_button_number` 将按钮预先确定为选中状态是不合适的。

在这种情况下，理想的是，在到达该显示集合之前，根据已经经过了前面多个再现路径中的哪一个来将按钮改变到选中状态（例如，当从第一章节到达时为第二章节按钮，当从第二章节到达时为第三章节按钮，当从第三章节到达时为第四章节按钮）。如果要处于选中状态的按钮根据前面的再现路径而改变，则将 `default_selected_button_number` 设定为表示“无效”，即，设定为值 0。由于要处于选中状态的按钮将发生变化，所以将不执行将特定 ODS 设置在按钮状态集合的开始处的方案。

图 22 示出了 S-ODS 中的 ODS 的顺序在 `default_selected_button_number` 表示“0”的情形和表示“按钮 B”的情形之间的差异。在该图中，虚线 `ss1` 表示当 `default_selected_button_number` 为按钮 B 时 S-ODS 中的 ODS 的顺序，虚线 `ss2` 表示当 `default_selected_button_number` 为 0 时 S-ODS 中的 ODS 的顺序。如图所示，当 `default_selected_button_number` 表示按钮 B 时，表示按钮 B 的选中状态的 ODSB 被设置在 S-ODS 的开始处，而其他按钮的 ODS 被设置在其之后。另一方面，当 `default_selected_button_number` 表示 0 时，表示按钮 A 的选中状态的 ODSA 被设置在 S-ODS 的开始处。以这种方式，`default_selected_button_number` 是有效还是无效并不影响 S-ODS 内的 ODS 的顺序。

接下来描述如何将具有这种 ICS 和 ODS 的显示集合分配到 AV 剪辑的再现时间轴。时元是存储器管理在再现时间轴上连续的一个时间段。由于时元由一个或多个显示集合组成，所以，如何将显示集合分配给 AV 剪辑的再现时间轴是一个问题。这里，AV 剪辑的再现时间轴用于定义各图形数据的解码 / 再现时机，图形数据构成复用到

AV 剪辑中的视频流。在该再现时间轴中，解码时机和再现时机由 90KHz 的时间精度来表示。分配给显示集合中的这些 ICS 和 ODS 的 DTS 和 PTS 表示在该再现时间轴上实现同步控制的时机。借助于分配给 ICS 和 ODS 的 DTS 和 PTS 的同步控制将显示集合分配给再现时间轴。

首先描述借助于分配给 ODS 的 DTS 和 PTS 执行何种同步控制。

DTS 表示 ODS 的解码开始时间，时间精度为 90 KHz。PTS 表示 ODS 的解码结束时间。

ODS 的解码不会立即结束，而是要花费一定的时间来完成。

为了阐述开始点 / 结束点，DTS 和 PTS 分别表示 ODS 的解码开始时间和解码结束时间。

由于 PTS 的值表示了结束时间，所以，有必要在 PTS 所示时间之前完成 ODS_j 的解码，再现装置的对象缓冲区中的解压缩图形对象的获取不应迟于此。

由于属于显示集合“n”的任一 ODS_j 的解码开始时间以 90KHz 的时间精度被表示为 DTS (DS_n [ODS_j])，所以，通过将解码所需的最大时间加上该解码开始时间而获得的时间将是所述显示集合的 ODS_j 的有保障的解码结束时间。

假定解压之后 ODS_j 的尺寸是 SIZE (DS_n [ODS_j]) 并且该 ODS 的解码速率是 R_d。那么，解码所需的最大时间将是 SIZE(DS_n[ODS_j]) //R_d。

应当注意的是，这里使用的运算符“//”是将小数升为下一整数的一种除法。

将该最大时间转变为 90KHz 的时间精度，并且加上 ODS_j 的 DTS。这样，就获得了由 PTS 表示的解码结束时间 (90KHz)。

属于 DS_n 的 ODS_j 的 PTS 表示如下：

$$PTS(DS [ODS_j]) = DTS(DS_n [ODS_j]) +$$

$$90000 * (\text{SIZE}(\text{DSn} [\text{ODSj}])) // \text{RD}$$

然后，在两个 ODS(ODSj,ODSj+1)之间应当满足下面的关系：

$$\text{PTS}(\text{DS} [\text{ODSj}]) \leq \text{DTS}(\text{DSn} [\text{ODSj+1}])$$

接下来将说明 ICS 的 PTS 值。

如果紧接着在 Epoch Start 之后，则在通过将下面的值添加到图形平面所获得的值之后来设置 ICS 的 PTS，这些值是：(1) 解码时间为构成 DS_n 的初始显示的 ODS 中的最后一个的 ODS 的 PTS 值；(2) 清除图形平面所需的时间；以及 (3) 写入由 ODS 的解码获得的图形对象所需的写入时间。

另一方面，如果在 Acquisition Point 处，则在通过将 (1) 和 (3) 相加所获得的值之后设置 IDS 的 PTS。

如果通过 ICS 来表示 default_selected_button_number，那么，当已经完成用于呈现所有按钮的正常状态的 ODS 的解码和用于呈现按钮的默认选中状态的 ODS 的解码时，初始显示的准备已就绪。用于呈现初始显示中出现的多个按钮的选中状态的 ODS 被称为“S-ODS”，而在解码时间完成最早的 S-ODS 中的 ODS 被称为“S-ODSsfirst”(在这种情况下，用于再现默认按钮的 ODS)。将 S-ODSsfirst 的 PTS 值设置为解码时间最后结束的 ODS 的 PTS 值，并且，将其用作 ICS 的 PTS 的引用值。

如果 default_selected_button_number 不是由 ICS 表示的，则哪个按钮将处于选中状态是未知的。因此，直到用于呈现所有按钮的正常状态和选中状态的准备工作就绪时，初始显示的准备才结束。在初始显示中所需的用于呈现多个按钮的选中状态的 S-ODS 中，将解码时间最后结束的 ODS 称为“S-ODSlast”。将该 S-ODSlast 的 PTS 值设置为解码时间最后结束的 ODS 的 PTS 值，并且，将其用作 ICS 的 PTS 的引用值。

假定 S-ODSsfirst 的解码结束时间被表示为 PTS (DS_n

[S-ODSsfirst])。那么,通过将(2)和(3)加上 $PTS(DSn[S-ODSsfirst])$ 来获得 $PTS(DSn[ICS])$ 。

假定在图形平面中可绘制的矩形区域的横向尺寸表示为“video_width”,而其纵向尺寸表示为“video_height”。同样假定针向图形平面的写入速率为 128Mbps。在这些假定的情况下,将清除图形平面所需的时间量表示为 $8*video_width*video_height / / 128,000,000$ 。如果以 90KHz 的时间精度进行表达,则(2)将清除图形平面所需的时间量表示为 $90,000*(8*video_width*video_height / / 128,000,000)$ 。

同样假定由包括在 ICS 中的所有按钮信息指定的图形对象的总尺寸表示为 $\sum SIZE(DSn[ICS.BUTTON[i]])$,向图形平面的写入速率为 128Mbps。那么,将写入图形平面所需的时间表示为 $\sum SIZE(DSn[ICS.BUTTON[i]]) / / 128,000,000$ 。如果以 90KHz 的时间精度进行表达,则清除图形平面所需的时间表示为 $90,000*(\sum SIZE(DSn[ICS.BUTTON[i]]) / / 128,000,000)$ 。

这里, $\sum SIZE(DSn[ICS.BUTTON[i]])$ 是用于表示一个按钮的多个图形对象的第一显示对象的总尺寸。该 $\sum SIZE(DSn[ICS.BUTTON[i]])$ 在判定默认选中按钮的情况及其动态变化的情况之间产生不同的值。当静态地判定默认选中按钮时, $\sum SIZE(DSn[ICS.BUTTON[i]])$ 将产生下面的总尺寸:表示默认选中按钮的选中状态的第一被显示 ODS;以及表示除默认选中按钮之外的按钮的正常状态的第一被显示 ODS。

接下来,当默认选中按钮动态变化时,不清楚哪一个按钮是选中的按钮。因此有必要假定具有最长写入时间的情形。这里,在表示任意按钮 x (ODSn1) 的正常状态的第一页的图形对象与表示按钮 x (ODSs1) 的选中状态的第一页的图形对象之间,将具有较大尺寸 ($Max(ODSn1,ODSs1)$) 的图形对象视为要首先为按钮 x 显示的图形对象。

通过将所有按钮的 $\text{Max}(\text{ODSn1}, \text{ODSs1})$ 相加到一起而获得的结果将是 $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 。

图 23A 和图 23B 是表示在 N-ODS 包括多个用于构成按钮 A 至 D 的 ODS 以及 S-ODS 包括多个用于构成按钮 A 至 D 的 ODS 的情况下 $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 的具体值的示意图。这里，当 `default_selected_button_number` 为有效值时， $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 将是由粗框所示的四个 ODS 的总尺寸。“As1” 是表示按钮 A 的选中状态的多个 ODS 中最先显示的那一个。“Bn1”、“Cn1” 和 “Dn1” 分别是表示按钮 B 到 D 的正常状态的多个 ODS 中最先显示的那一个。当这些尺寸用 “size ()” 表示时， $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 将被表示为 $\text{size}(\text{As1}) + \text{size}(\text{Bn1}) + \text{size}(\text{Cn1}) + \text{size}(\text{Dn1})$ 。

另一方面，当 `default_selected_button_number` 为 0 时，An1 和 As1 的较大 ODS；Bn1 和 Bs1 的较大 ODS；Cn1 和 Cs1 的较大 ODS；以及 Dn1 和 Ds1 的较大 ODS 的总和将成为 $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 。

从而， $\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]])$ 被表示如下。

$$\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) = \max((\text{size}(\text{An1}), \text{size}(\text{As1})) + \max((\text{size}(\text{Bn1}), \text{size}(\text{Bs1})) + \max((\text{size}(\text{Cn1}), \text{size}(\text{Cs1})) + \max((\text{size}(\text{Dn1}), \text{size}(\text{Ds1})))$$

于是，紧接着 Epoch Start 后的 $\text{PTS}(\text{DSn}[\text{ICS}])$ 将使用上面的表达式表示如下：

$$\begin{aligned} \text{PTS}(\text{DS}[\text{ICS}]) &\geq \text{PTS}(\text{DSn}[\text{S-ODSsfirst}]) \\ &+ 90,000 * (8 * \text{video_width} * \text{video_height} / / 128,000,000) \\ &+ 90,000 * (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) / / 128,000,000) \end{aligned}$$

另一方面，当 `default_selected_button_number` 无效时，仅仅需要用 $\text{PTS}(\text{DSn}[\text{S-ODSslast}])$ 代替 $\text{PTS}(\text{DSn}[\text{S-ODSsfirst}])$ 。具体的表达式如下：

$$\text{PTS}(\text{DS}[\text{ICS}]) \geq \text{PTS}(\text{DSn}[\text{S-ODSslast}])$$

$$+90,000*(8*video_width*video_height / / 128,000,000)$$

$$+90,000* (\sum SIZE(DSn[ICS.BUTTON[i]]) / / 128,000,000)$$

图 24 表示通过以上述方式设定 PTS 和 DTS 来实现同步显示的一个示例。在该图中，做出假设以使得在任意图像数据“py1”（在该图中的动画中）的显示时机处显示按钮。在这种情况下，应该设置 ICS 的 PTS 值，从而与该图像数据的显示时机相一致。

在构成 DS_n 的初始显示的多个 ODS 中解码时间最后完成的那个 ODS 的解码应当在通过从 ICS 的 PTS 中减去屏幕清除时间 cd1 和图形对象传送时间 td1 而获得的时间之前完成。因此，应当将该 ODS 的 PTS 值设置在该图中示出的时机（★1）处。此外，ODS 的解码花费时间 dd1，因此，应当将该 ODS 的 DTS 值设置在比该 PTS 早的 dd1 时间处。

在图 24 中，仅有一个要与运动图像重叠的 ODS，这是一种简化情形。如果通过多个 ODS 来实现与运动相重叠的交互式显示的初始显示，，则 ICS 的 PTS 和 DTS 以及 ODS 的 PTS 和 DTS 应该如图 25 所示进行设置。

图 25 是表示在多个 ODS 合成交互式显示的初始显示以及静态地判定默认选中按钮的情况下如何设置 DTS 和 PTS 的示意图。如果经过图中的时间 dd1 时 S-ODS_{sfirst}（用于实现初始显示的 ODS）中需要最长解码时间的 ODS 的解码完成，则将该 S-ODS_{sfirst} 的 PTS(DS_n[S-ODS_{sfirst}]) 设定为表示经过时段 dd1 的时间。

此外，为了实现初始显示，有必要执行屏幕清除和解码图形对象的传输。因此，应当将该 ICS (DS_n[ICS]) 的 PTS 设定在通过将执行屏幕清除所需的时间 (90,000*(8*video_width*video_height / / 128,000,000)) 和解码图形对象的传输时间 (90,000*(∑SIZE(DSn[ICS.BUTTON[i]]) / / 128,000,000)) 加上该 PTS 值 (DS_n[S-ODS_{sfirst}]) 而获得的时间之后。

图26是表示在多个ODS合成交互式显示的初始显示以及不对默认选中按钮进行判定的情况下如何设定DTS和PTS的示意图。如果S-ODSslast（其解码在用于实现初始显示的S-ODS中花费的时间最长）的解码经过在图中的时间dd2时完成，则将该S-ODSslast（DSn[S-ODSslast]）的PTS设定为表示经过时段dd2的时间。

此外，为了实现初始显示，有必要执行屏幕清除，并且传输解码的图形对象。因此，应当将该ICS的PTS（DSn[ICS]）设定在通过将执行屏幕清除所需的时间（ $90,000 \times (8 \times \text{video_width} \times \text{video_height} / / 128,000,000)$ ）与解码图形对象的传输时间（ $90,000 \times (\sum \text{SIZE}(\text{DSn}[\text{ICS.BUTTON}[i]]) / / 128,000,000)$ ）加上该PTS（DSn[S-ODSslast]）的值而获得的时间之后。

在DVD中，交互式控制有效的时段对应于VOBU（其起到视频流的GOP的作用）的周期。然而，在BD-ROM中，可以借助于包括在时元中的ICS中的PTS和DTS来任意设定这种有效时段。因此，借助于BD-ROM的交互式控制不会依赖于GOP。

应当注意的是，借助于ICS的PTS的同步控制不限于执行控制以便在再现时间轴上的确定时机处显示一个按钮，并且还包含执行控制以便能够在再现时间轴上的特定时段中显示弹出菜单。弹出菜单是通过按压遥控器400上提供的菜单按键而在屏幕上弹出来的菜单，借助于ICS的PTS的同步控制还包括在AV剪辑中的特定图形数据的显示时机处能够显示这种弹出菜单。对构成弹出菜单的ODS进行解码，并且将通过解码获得的图形对象写入图形平面中，就如同构成按钮的ODS一样。除非向图形平面的写入没有完成，否则，用户不可能响应菜单调用。鉴于此，在执行弹出菜单的同步显示中，可能进行弹出显示的时间通过ICS的PTS来表示。

上面描述的显示集合（ICS、PDS、ODS）的数据结构是用程序语言编写的类结构体的一个实例。进行创作的生产者能够通过描述这种结构体来获得 BD-ROM 上的结构。

接下来说明根据本发明的再现装置的一个实施例。图 27 是根据本发明的再现装置的内部结构的示意图。在工业上，基于该图中示出的内部结构来生产根据本发明的再现装置。本发明的再现装置主要包括三个部分：系统 LSI，驱动装置和微系统。通过将这些部件实现为再现装置的机壳和基板来进行工业生产。系统 LSI 是其上集成了多种处理单元的集成电路，这些处理单元具有再现装置的功能。以这种方式生成的再现装置包括：BD 驱动器 1，跟踪缓冲器 2，PID 滤波器 3，传输缓冲器 4a、b 和 c，外围电路 4d，视频解码器 5，视频平面 6，音频解码器 7，图形平面 8，CLUT 单元 9，加法设备 10，图形解码器 12，编码数据缓冲器 13，外围电路 13a，流图形处理器 14，对象缓冲器 15，合成缓冲器 16，图形控制器 17，UO 控制器 18，选手寄存器组 19 和控制单元 20。

BD-ROM 驱动器 1 通过执行 BD-ROM 的加载 / 读取 / 弹出来访问 BD-ROM。

跟踪缓冲器 2 是 FIFO 存储器，并且在先进先出的基础上存储从 BD-ROM 中读取的 TS 数据包。

PID 滤波器 3 对从跟踪缓冲器 2 输出的多个 TS 数据包执行滤波。通过 PID 滤波器 3 执行的滤波是仅将具有预期 PID 的 TS 数据包写入传输缓冲器 4a、b 和 c，而不需要进行缓冲。因此，输入到 PID 滤波器 3 中的 TS 数据包被写入传输缓冲器 4a、b 和 c 中，而不存在延时。

传输缓冲器 4a、b 和 c 分别是在先进先出的基础上存储从 PID 滤波器 3 输出的 TS 数据包的存储器。

外围电路 4d 是执行处理以将从传输缓冲器 4a 读取的 TS 数据包转换成功能段的线逻辑。通过该转换而获得的功能段存储到编码数据缓冲器 13 中。

视频解码器 5 对从 PDI 滤波器 3 输出的多个 TS 数据包进行解码，以获得解压缩的图片，并将它们写入视频平面 6。

视频平面 6 是用于动画图片的平面。

音频解码器 7 对从 PDI 滤波器 3 输出的 TS 数据包进行解码，以获得解压缩的音频数据，并且输出解压缩的音频数据。

图形平面 8 是具有与一个画面相对应的区域的存储器，并且能够存储用于一个画面的解压缩图形。

CLUT 单元 9 基于 PDS 所示的 Y、Cr、Cb 值，图形平面 8 中存储的解压缩图形的索引颜色进行转换。

加法设备 10 将从 CLUT 单元 9 处进行颜色转换所得的解压缩图形与 PDS 所示的 T 值（传输速率）相乘，并且将所得结果与视频平面 6 中存储的解压缩图形数据逐像素地进行相加，从而获得重叠的图像，并且输出重叠的图像。

图形解码器 12 对图形流进行解码，以获得解压缩图形，并且将该解压图形作为图形对象写入图形平面 8。通过对图形流进行解码，字幕和菜单将出现在屏幕上。该图形解码器 12 包括编码数据缓冲器 13、外围电路 13a、流图形处理器 14、对象缓冲器 15、合成缓冲器 16 和图形控制器 17。

编码数据缓冲器 13 是其中将功能段和 DTS 与 PTS 一起存储的缓冲器。通过去除传输缓冲器 4a 中存储的传输流中的各 TS 数据包的 TS 包头和 PES 包头，而后顺序地放置这些有效载荷，从而得到这些功能段。在被去除的 TS 包头和 PES 包头中，PTS/DTS 与对应 PES 包关联地存储。

外围电路 13a 是能够实现如下功能的线逻辑：在编码数据缓冲器 13 和流图形处理器 14 之间进行传输；以及在编码数据缓冲器 13 和合成缓冲器 16 之间进行传输。在这种变换处理中，当到达 ODS 的 DTS 所示的时间时，将 ODS 从编码数据缓冲器 13 传输到流图形处理器 14。另外，当到达 ICS 和 PDS 的 DTS 所示的时间时，将 ICS 和 PDS 传输到合成缓冲器 16。

流图形处理器 14 对 ODS 进行解码，并且将包括源自所述解码的索引颜色的解压缩图形作为图形对象写入对象缓冲器 15。通过流图形处理器 14 进行的解码在与 ODS 相关联的 DTS 处开始，并且在与 ODS 相关联的 PTS 处结束。针对上述图形对象的解码速率“Rd”对应于该流图形处理器 14 的输出速率。

从流图形处理器 14 的解码中获得的图形对象被存储在对象缓冲器 15 中。图 28 是与图形平面 8 相对照的对象存储器 15 的存储内容的示意图。该存储内容假定一种情形，其中，在图 16 和 17 所示的示例中的 ODS 被写入对象缓冲器 15。在图 16 和图 17 的示例中，通过 36 个 ODS (ODS11-49) 实现了四个按钮动画。在该示例中，表示该动画的所有帧的 ODS 被存储在对象缓冲器 15 中。相比之下，图形平面 8 中定义了对对象缓冲器 15 中存储的各 ODS 的显示位置。该显示位置通过各按钮信息的按钮水平位置和按钮垂直位置来定义。通过逐帧从对象缓冲器 15 传输 ODS 来实现动画，从而将 ODS 写入与图形平面 8 的对应的显示位置。

合成缓冲器 16 是设置 ICS 和 PDS 的存储器。

图形控制器 17 对合成缓冲器 16 中放置的 ICS 进行解码，并且根据 ICS 执行控制。该控制的执行时机基于分配给 ICS 的 PTS 的值。该图形控制器 17 最重要的功能之一是在初始显示处和交互式显示更新处的写处理。下面参照图 19 解释图形控制器 17 在初始显示处和更新处的写处理。图 29 是图形控制器 17 在初始显示处执行的处理的示

意图。如该图所示，图形控制器 17 执行控制，从而将属于按钮 A 的 S-ODS 的 ODS 写入由按钮 A 的按钮信息的 `button_horizontal_position` 和 `button_vertical_position` 定义的显示位置，并且将属于按钮 B、C 和 D 的 N-ODS 的 ODS 分别写入由按钮 B、C 和 D 的按钮信息的 `button_horizontal_position` 和 `button_vertical_position` 定义的位置（图中的箭头 W1、W2、W3 和 W4 示出了上述写入）。通过所述写入，实现了图 20 的初始显示。这里请注意，交互式显示的初始显示并不需要所有的 ODS。当属于默认选中的 S-ODS 的 ODS 和属于其他按钮的 N-ODS 的 ODS 都已经在对象缓冲器 15 上就绪时，初始显示得以实现。也就是说，如果已经完成了针对属于默认选中按钮的 S-ODS 的 ODS 的解码和针对属于其他按钮的 N-ODS 的 ODS 的解码，则图形控制器 17 就可以启动针对初始显示的写入。

图 30 是表示图形控制器 17 根据第一用户动作（右移）的交互式显示的更新处的处理的示意图。如该图所示，图形控制器 17 执行控制，从而将属于按钮 B 的 S-ODS 的 ODS 写入由按钮 B 的按钮信息的 `button_horizontal_position` 和 `button_vertical_position` 定义的位置，并且，将属于按钮 A 的 N-ODS 的 ODS 写入由按钮 A 的按钮信息的 `button_horizontal_position` 和 `button_vertical_position` 定义的位置（箭头 W5、W6、W7 和 W8 示出了上述写入）。通过所述写入，实现了如图 20 所示的状态转换。按钮 C 和 D 保持在正常状态，就如同交互式显示的初始显示中一样，但继续向图形平面 8 写入这些按钮，以继续该动画。

图 31 和图 32 分别表示图形控制器 17 在第一用户动作是下移且激活的各种情况下的处理。在交互式屏幕更新时，涉及除默认选中按钮之外的按钮的 S-ODS 和 A-ODS，因此，所有这些 ODS 都需要在对象缓冲器 15 中存在。

UO 控制器 18 检测针对遥控器和再现装置的前面板的用户操作，并且将对应于用户操作（下文中“UO(用户操作)”）的信息输出到控制单元 20。

选手寄存器组 19 是嵌入在控制单元 20 中的寄存器，并且包括 32 个选手状态寄存器和 32 个通用寄存器。下面描述选手状态寄存器（PSR）的设定值的含义。在下文中，符号“PSR(x)”表示第 X 个选手状态寄存器的设定值。

PSR(0): 保留

PSR(1): 待解码的音频流的流编号

PSR(2): 待解码的子图像流的流编号

PSR(3): 表示用户角度设定的编号

PSR(4): 当前再现的标题号

PSR(5): 当前再现的章节号

PSR(6): 当前再现的 PL 号

PSR(7): 当前再现的播放项目号

PSR(8): 表示当前再现点的时间信息

PSR(9): 在导航计时器处的计数值

PSR(10): 当前处于选中状态的按钮编号

PSR(11)–(12): 保留

PSR(13): 由用户设定的父级

PSR(14): 再现装置的视频再现设置

PSR(15): 再现装置的音频再现设置

PSR(16): 表示再现装置的音频再现设置的语言代码

PSR(17): 表示再现装置的字幕设置的语言代码

PSR(18): 用于菜单呈现的语言设置

PSR(19)-(63): 保留

每次在显示 AV 剪辑的各图像数据时更新 PSR(8)。换句话说，当再现装置显示一条新的图像数据时，将 PSR(8)更新为由所述新的图像数据的显示开始时间（呈现时间）表示的值。通过参考该 PSR(8)，可以获知当前再现点。

控制单元 20 通过与图形解码器 12 的双向通信执行集中控制。从控制单元 20 到图形解码器 12 的通信要将在 UO 控制器 18 处接收的 UO 输出到图形解码器 12。从控制单元 20 到图形解码器 12 的通信要将包括在 ICS 中的按钮命令输出到控制单元 20。

在上述构造的再现装置中，各构成元件使用流水线操作来执行解码处理。

图 33 示出了由再现装置执行的流水线操作处理的时序图。第四行表示 BD-ROM 中的显示集合，第三行表示分别用于将 ICS、PDS 和 ODS 读取到编码数据缓冲器 13 的读取时间。第二行表示由流图形处理器 14 针对 ODS 执行的解码的时间。第一行表示图形控制器 17 的处理时间。图中的 DTS11、DTS12 和 DTS13 表示各个 ODS 的解码开始时间。在 DTS11 之前，完成 ODS (N-ODS [ODS1]) (是 N-ODS 中的第一个 ODS) 到编码数据缓冲器 13 的存储，并且，在 DTS12 之前，完成 ODS (N-ODS [ODSn]) (作为 N-ODS 中的最后一个 ODS) 到编码数据缓冲器 13 的存储。以这种方式，各 ODS 在它 DTS 所示的时间之前已经被读取到编码数据缓冲器 13 中。

另一方面，在图中，PTS 11、PTS 12、PTS 13 表示各 ODS 的解码结束时间。PTS 11 之前，完成由流图形处理器 14 执行的 N-ODS (ODS1) 的解码，并且，在 PTS12 之前，完成 N-ODS (ODSn) 的解码。以上述方式，各 ODS 在该 ODS 的 DTS 所示时间之前被读取到编码数据缓冲器 13 中，并且，在该 ODS 的 PTS 所示时间之前，完成对读取到编码数据缓冲器 13 中的 ODS 的解码并且将它们写入对

象缓冲器 15 中。流图形处理器 14 使用流水线操作来执行所描述的处理序列。

当静态地判定默认选中按钮时，针对交互式显示的初始显示的所有图形对象的就绪时间是：与正常状态的按钮状态集合相对应的第一个 ODS 的解码和与选中状态的按钮状态集合相对应的第一个 ODS 的解码结束的时候。在该图中，在 PTS13 所示的时间处，交互式显示的初始显示缩必需的所有图形对象都已就绪。

在图中，第一行中的时段 cd1 表示通过图形控制器 17 清除图形平面 8 所需的时间。另外，时段 td1 是将在对象缓冲器 15 中获得的、与交互式显示的第一页相对应的图形对象写入图形平面 8 所需的时间。ICS 中的按钮水平位置和按钮垂直位置表示图形平面 8 中的写入目的地。换言之，在通过将屏幕清除时段“cd1”和写入图形对象（通过解码“td1”所获得的）所需的写入时间加上 ODS 的 PTS13 而计算出的时间处，将构成交互式显示的解压图形获取到图形平面 8。通过使 CLUT 单元 9 对上述提到的解压图形执行颜色转换、并且使附加设备 10 执行将所述结果与视频平面 6 中存储的解压图形的组合，可以获得重叠图像。

与通过对包括在显示集合中的所有 ODS 进行解码来执行初始显示的情形相对照的是，在图中的时段 hy1 之前，执行该实施例中的初始显示，因为，初始显示是在与选中状态相对应的按钮状态集合和与活动状态相对应的按钮状态集合的解码完成之前执行的。

应当注意的是，在该图中，存在多个 ODS1-ODSn 的集合。然而，这些集合分别属于 N-ODS、S-ODS 和 A-ODS，因此，相同的 ODS1 在内容上彼此不同。这也适用于本发明中类似的附图。

即使当图形解码器 12 执行图形平面 8 的清除和写入时，流图形处理器 14 也继续对剩余的 ODS 进行解码（即，在第二行中的针对 ODS 的解码时间、针对 ODS1 的解码时间和针对 ODSn 的解码时间）。

基于此，剩余 ODS 的解码将及早完成。这样可以较早地为更新交互式显示作准备，因此借助于剩余的 ODS 可以响应有关交互式显示更新的用户操作，而没有延迟。如上所述的该流水线操作处理能够实现初始显示和交互式显示的更新的准时执行。

图 33 假定了静态地判定默认选中按钮的情形，而图 34 是在默认选中按钮动态变化的情况下由再现装置执行的流水线处理的时序图。如果默认选中按钮动态变化，那么，当对所有属于分别与 N-ODS 和 S-ODS 相对应的按钮状态集合的 ODS 进行解码并且在图形平面 8 中获得对应的图形对象时，初始显示所必需的图形对象将就绪。与必须对显示集合中的所有 ODS 进行解码以执行初始显示的情形相对照，在与活动状态相对应的按钮状态集合的解码结束之前，进行该实施例中的初始显示是可能的。因此，初始显示的执行比该图中的时段 hy2 要早。

接下来描述如何实现控制单元 20 和图形解码器 12。控制单元 20 创建用于执行图 35 和图 36 中所示处理的程序，该程序由通用 CPU 来执行。下面将参照图 35 和图 36 来描述控制单元 20 执行的处理。

图 35 的流程图示出了由控制单元 20 执行的 LinkPL 功能的执行程序。对包括 LinkPL 功能的命令进行解码时，控制单元 20 遵循该图的流程图。

这里，假定在该流程图中的处理目标播放项目被称为“PI_y”，而处理目标访问单元被称为“访问单元 v”。在该流程图中，读取由 LinkPL 的参数表示的当前 PL 信息 (.mpls) (步骤 S1)，并且将在当前 PL 信息中的第一条 PI 信息设置为 P_{iy} (步骤 2)。然后，读取由 P_{iy} 的剪辑信息文件名指定的剪辑信息 (步骤 3)。

一旦读取了剪辑信息，就使用当前剪辑信息的 EP-map，将 P_{iy} 的 In-time 转换为地址 (步骤 S4)。然后，将由该地址指定的访问单元设为访问单元 v (步骤 S5)。使用当前剪辑信息的 EP-map，将 P_{iy}

的 Out-time 转换为地址（步骤 S6）。然后，将由该地址指定的访问单元设为访问单元 w（步骤 S7）。

以这种方式，一旦判定访问单元 v、w，就向 BD 驱动器发出一条指令，以便从访问单元 v 到访问单元 w 进行读取（步骤 S8），并且，向视频解码器 5、音频解码器 7 和图形解码器 12 发出从 In_time 到 Out_time 进行解码 / 输出 Piy 的指令（步骤 S9）。

步骤 S11 用于执行流程图的结束判断。具体而言，判断 Piy 是否到达最后的 PI。如果步骤 S11 的结果为 YES，则流程图结束。否则，将下一个播放项目设为 Piy（步骤 S12），同时操作返回步骤 S3。下文中，重复步骤 S1 到步骤 S10 的处理，直到步骤 S11 的结果为是（YES）为止。

步骤 S10 是用于将功能段加载到编码数据缓冲器 13 以响应访问单元读取的步骤。

图 36 的流程图示出了用于加载功能段的处理的过程。在该流程图中，“SegmentK”表示对应于与访问单元一同读取的各段（ICS、ODS、PDS）的变量。忽略标志用于在 SegmentK 的忽略和加载之间进行切换。该流程图具有循环结构，其中，当把忽略标志设为“1”并且进行了初始化时，对于所有 SegmentK，重复执行步骤 S21 到 S24 和步骤 S27 到 S35 的处理（步骤 S25、S26）。

步骤 S21 用于判断 SegmentK 是否为 ICS。如果 SegmentK 是 ICS，则执行步骤 S27 和步骤 S28 的判断。

步骤 S27 用于判断 ICS 的 Segment_Type 是不是 Acquisition Point。如果 SegmentK 是 Acquisition Point，则操作转移到步骤 S28。相反，如果 SegmentK 是 Epoch Start 或 Normal Case，则操作转移到步骤 S33。

当步骤 S27 为 Yes 时，步骤 S28 用于判断前一 DS 位于图形解码器 12 内哪个缓冲器中（在编码数据缓冲器 13、流图形处理器 14、对象缓冲器 15 和合成缓冲器 16 之间）。当在图形解码器 12 中没有发现

DS 时，这意味着已经执行了随机访问。在这种情况下，显示应当从作为 Acquisition Point 的 DS 开始。从而，操作转移到步骤 S30（步骤 S28：否）。

当在图形解码器 12 中发现了前一 DS 时（步骤 S28：是），将忽略标志设为“1”（步骤 S29），并且，操作转移到步骤 S31。

步骤 S31 用于判断是否将 `command_update_flag` 设为 1。如果为 1（步骤 S31：是），则仅将按钮信息中的按钮命令加载到编码数据缓冲器 13 中，而忽略其他命令（步骤 S32）。如果为 0，则操作转移到步骤 S22。以这种方式，将忽略表明 Acquisition Point 的 ICS（步骤 S24）。

如果将忽略标志设为 1，则步骤 S22 为否，从而忽略所有属于 Acquisition PointDS 的功能段。

步骤 S33 用于判断 ICS 中的 `Segment_Type` 是否表示 Normal Case。当 `SegmentK` 为 Epoch Start 时，在步骤 S30 将忽略标志设为 0。

当忽略标志为 0 时（步骤 S22：是），将 `SegmentK` 加载到编码数据缓冲器 13 中（步骤 S23）。

当 `SegmentK` 是 Normal Case 时，操作转移到步骤 S34。步骤 S34 在内容上与步骤 S28 相同，用于判断是否在图形解码器 12 中发现了前面的 DS。如果判断结果为肯定，则将忽略标志设为“0”（步骤 S30）。如果判断结果为否定，则将忽略标志设为“1”，因为获得足够数量的用于构成交互式显示的功能段在本质上是是不可能的（步骤 S35）。根据上述标志的设定，当没有在图形解码器 12 中找到之前的 DS 时，将忽略构成 Normal Case 的功能段。

接下来说明在如图 37 所示那样在执行涉及 DS 的复用的情形下如何执行 DS 的读取。在图 37 的示例中，3 个 DS 已经与动画复用，该动画由多张图片构成。在这 3 个 DS 中的第一个 DS（DS1）中，

Segment_Type 表明 Epoch_Start, Command_update_flag 被设为 0, 并且包括 LinkPL (PL#5) 的按钮命令。

DS 10 是 DS 1 的副本, DS 10 的 Segment_Type 表明 Acquisition Point, 并且 Command_update_flag 被设为 0, 并且, 其中包括 LinkPL (PL#5) 的按钮命令。

DS 20 是 DS1 的继承, DS20 的 Segment_Type 表明 Acquisition Point。与 DS1 的唯一不同之处在于按钮命令 (LinkPL(PL#10))。为了表明存在差别, 将 Command_update_flag 设定为 1。

在上述 3 个 DS 已经与运动图像进行了复用的 AV 剪辑中, 假定在图像数据 pt10 处执行随机访问。在这种情况下, 最接近于该随机访问点的 DS10 将成为图 36 所示流程图处理的目标。在步骤 S27 中, 判定 Segment_Type 为 Acquisition Point, 但在图形解码器 12 中找不到在先的 DS。因此, 忽略标志被设为 0, 并且, 将 DS10 加载到再现装置中的编码数据缓冲器 13 中, 如图 38 所示。另一方面, 在当前显示集合 10 所在点之后存在随机访问点时 (图 37 中的虚线 hst1), 将显示集合 10 之后的显示集合 20 (图 38 中的 hst2) 读取到编码数据缓冲器 13 中。

图 40 表示当如图 39 所示执行正常再现时如何执行 DS1、DS10 和 DS20 的加载。在这 3 个 DS 中, 将 ICS 中的 segment_type 为 Epoch Start 的 DS1 如实加载到编码数据缓冲器 13 中 (步骤 S23)。然而, 将 ICS 中 Segment_Type 为 Acquisition Point 的 DS10 的忽略标志设为 1 (步骤 S29), 因此, 忽略构成 DS10 的功能段, 并且不将其加载到编码数据缓冲器 13 中 (步骤 S24)。就 DS20 来说, ICS 的 Segment_type 表示 Acquisition Point, 但 Command_update_flag 被设为 1。因此, 步骤 S31 的结果为是, 因此仅加载按钮命令。因此, 就编码数据缓冲器 13 的 DS 来说, 仅有 ICS 中的按钮命令被替换为该被加载的按钮命令

(步骤 S32)。这里，由于忽略标志仍然表示 1，所以，忽略并不加载除该按钮命令以外的其他元素。

尽管具有与 DS1 相同的显示内容，但在到达 DS20 时，该按钮命令从 DS1 的 LinkPL(#5)到 DS20 的 LinkPL(#10)已经被替代。这种替代方便了控制，以在再现过程中改变按钮命令的内容。接下来描述图形控制器 17 的处理过程。图 41 的流程图中描述了与图形控制器 17 的处理中的主程序相对应的处理。在该流程图中，重复执行这 3 种处理：时戳同步处理（步骤 S35）；动画显示处理（步骤 S36）；和 UO 处理（步骤 S37）。

图 42 是借助于时戳执行同步控制的处理过程的流程图。在该流程图中，判断是否存在步骤 S41、S43 到步骤 S47 的任何一个现象。如果任何一个现象存在，则形成执行相应操作的子程序，而后，操作返回到主程序。

步骤 S41 用于判断当前再现时间是由 S-ODSsfirst 的 PTS 表示的任何一个时间和由 S-ODSslast 的 PTS 表示的时间。如果判断结果为肯定，则在步骤 S42 计算时段 α 。时段 α 是通过将 (2) 清除图形平面所需的时间和 (3) 将通过解码 ODS 而获得的图形对象写入所述图形平面上所需的时间相加所获得的时段。

在步骤 S42，图形控制器 17 查询 ICS 的 Segment_Type，如果该 Segment_Type 表示 Epoch Start，则将 α 设置为 (2) 平面清除时间 + (3) 平面写入时间。如果该 Segment_Type 表示 Acquisition Point，则将 α 设置为 (3) 平面写入时间。为了计算平面写入时间 (3)，当 default_selected_button_numer 表示一个有效值时，使用图 23A 的计算。另一方面，当 default_selected_button_numer 表示 0 时，使用图 23B 的计算，来计算平面写入时间 (3)。在以上述方式计算出时段 α 之后，操作返回到循环处理。

步骤 S43 用于判断当前再现时间是不是 ICD 的 PTS — α 表示的时间。如果判断结果为肯定，则执行向图形平面 8 的写入处理，同时操作返回到主程序。

步骤 S45 用于判断当前再现时间是否为 ICS 的 PTS。如果判断结果为肯定，则启动图形平面 8 的存储内容的输出。该存储内容的输出目的地是 CLUT 单元 9。在 CLUT 单元 9 中进行颜色转换之后，交互式显示将与视频平面 6 的存储内容重叠。基于此来执行初始显示（步骤 S52）。然后，将变量“动画 (p) (p=1、2、3…n)”设为 0（步骤 S53），而后操作返回到主程序。这里，变量“动画 (P)”是一个全局变量，该全局变量表示帧序列中的当前帧的顺序位置，其在多个流程图中保持有效。在步骤 S53 中，所有按钮中的按钮 (p) 将被设为 0。

步骤 S46 和步骤 S47 用于判断当前再现时间是否到达在 ICS 中描述的时间信息。

步骤 S46 用于判断当前再现时间是不是 selection_timeout_PTS 表示的时间。如果判断结果为肯定，则执行用于激活由 default_activated_button_number 表示的按钮的处理，然后，操作返回到主程序（步骤 S54）。

步骤 S47 用于判断当前再现时间是不是 Compostion_TimeOut_PTS。如果判断结果为肯定，则执行屏幕清除，然后操作返回到主程序（步骤 S55）。在借助于时戳的上述同步处理中，步骤 S51 和步骤 S54 分别被创建为子程序。下面参照图 43 来说明在步骤 S51 的子程序中执行的处理。

图 43 是将菜单的初始显示写入图形平面 8 的处理过程的流程图。步骤 S64 用于判断 ICS 的 Segment_type 是否表示 Epoch Start。如果判断结果为肯定，则在步骤 S65 中清除图形平面 8，并且，执行步骤 S66 到步骤 S73 的处理。清除图形平面 8 所需的时间对应于图 25 和

图 26 中的时段 cd1。如果判断结果为否定，则跳过步骤 S65，并且，执行步骤 S66 到步骤 S73 的处理。

步骤 S66 到步骤 S73 构成循环处理，对于 ICS 中的各按钮信息重复执行（步骤 S66，步骤 S67）。在该循环处理中，将作为处理目标的一条按钮信息称为“按钮信息（p）”。

步骤 S67 用于判断 default_selected_button_number 是否表示一个有效编号。步骤 S68 用于判断按钮信息（p）是否对应于由 default_selected_button_number 表示的默认选中按钮。

如果 button_info（p）不对应于默认选中，则从对象缓冲器 15 中找到由 button_info（p）的 normal_state_info 表示的 start_object_id_normal 的图形对象，并且，将其设为图形对象(p)（步骤 S69）。

如果 button_info（p）对应于默认选中按钮，则从对象缓冲器 15 中找到 start_object_id_selected 的图形对象（步骤 S70），并且将按钮（p）设置为当前按钮（步骤 S71）。该当前按钮是当前显示的交互式显示中处于选中状态的按钮，并且，再现装置将该当前按钮的标识符存储为 PSR（10）。

在通过步骤 S69 到步骤 S70 的识别之后，将图形对象（p）写入到图形平面 8 中由 button_info（p）的 button_horizontal_postion 和 button_horizontal_postion 表示的位置处（步骤 S72）。通过对 ICS 的各条按钮信息重复该处理，将第一图形对象写入图形平面 8。完成图形平面 15 中的那些图形对象中并且对于初始显示是必须的图形对象的该处理（）所需的时间对应于在图 25 和图 26 中所示的时段 td1。

当 default_selected_button_number 表示 0 并且该默认选中按钮动态变化时，步骤 S67 将为否，在步骤 S73 中，判断 button_info（p）是否对应于当前按钮。如果判断结果为肯定，则操作进入步骤 S70。如果判断结果为否定，则操作进入步骤 S69。

接下来参照图 44 说明步骤 S54 的子程序的处理过程。

图 44 是表示如何自动激活默认选中按钮的处理过程的流程图。首先，判断该 default_activated_button_number 是 00 还是 FF。如果该 default_activated_button_number 为“00”，则操作返回到主程序而不执行任何处理。如果该 default_activated_button_number 为“FF”，则将当前按钮 i 改变到活动状态（步骤 S77）。然后将作为对应于当前按钮 i 的变量的动画 (i) 设置为 0，同时操作返回到主程序（步骤 S78）。

如果 default_activated_button_number 既不是 00 也不是 FF，则将 default_activated_button_number 表示的按钮设定为当前按钮（步骤 S76），并且将当前按钮 i 改变为活动状态（步骤 S77），并且将作为对应于当前按钮 i 的变量的动画 (i) 设置为 0，同时操作返回到主程序（步骤 S78）。

根据上述处理，在预定时间之后，将处于选中状态的按钮改变到活动状态。

接下来描述菜单中的动画（步骤 S36）。图 45 是表示动画显示处理程序的流程图。

这里，在各 button_info 中，通过将由 normal_state_info 的 start_object_id_normal 和由 selected_state_info 的 start_object_id_selected 指定的图形对象写入图形平面 8 来实现初始显示。每当步骤 S35 到步骤 S37 的循环操作完成一个循环时，对动画进行处理以利用各个按钮中的任意图像（即对应于第 q 个图像的图形对象）来重写该图形平面 8。具体而言，通过将由 button_info 中的 normal_state_info 和 selected_state_info 指定的图形对象逐一写入图形平面 8 并返回到主程序来实现该更新处理。这里，“q”是用于指定由各按钮信息中的 button_info 的 normal_state_info 和 selected_stated_info 指定的各图形对象的变量。

参照图 45 来说明实现这种动画显示的处理。为简单起见，创建该流程图基于这样一种假设：将 `repeat_normal_flag` 和 `repeat_selected_flag` 设定为表示这种重复是必要的。

步骤 S80 用于判断是否已完成初始显示。如果还没有完成初始显示，则该操作返回，而不执行任何处理。如果已完成了初始显示，则将步骤 S81 到步骤 S93 设计为构造一个循环处理，其中，针对 ICS 中的各条 `button_info` 重复步骤 S81 到步骤 S93 的处理（步骤 S81，步骤 S82）。

步骤 S83 将动画（`p`）设置为变量 `q`（对应于按钮信息（`p`）的变量）。以这种方式，变量 `q` 表明在帧序列中当前帧的顺序位置，其对应于 `button_info`（`p`）。

步骤 S84 用于判断按钮信息（`p`）是否对应于当前处于选中状态的按钮（当前按钮）。

如果 `button_info`（`p`）对应于与当前按钮不同的按钮，则将一个通过将变量 `q` 加上 `button_info`（`p`）的 `normal_state_info` 的 `start_object_id_normal` 而获得的标识符设置为 `ID(q)`（步骤 S85）。

如果 `button_info`（`p`）对应于当前按钮，则执行步骤 S86 的判断。

步骤 S86 用于判断当前按钮是否处于活动状态。如果判断结果是肯定的，则将一个通过将变量 `q` 加上 `button_info`（`p`）的 `actioned_stated_info` 的 `start_object_id_actioned` 而获得的标识符设置为 `ID(q)`（步骤 S87）。然后，执行包括在 `button_info`（`p`）中的一个按钮命令（步骤 S88）。

如果判断当前按钮不处于活动状态，则将一个通过将变量 `q` 加上 `button_info`（`p`）的 `selected_stated_info` 的 `start_object_id_selected` 而获得的标识符设置为 `ID(q)`（步骤 S89）。

以这种方式，一旦判定了 `ID(q)`，就在由 `button_info`（`p`）的 `button_horizontal_position` 和 `button_vertical_postion` 指定的位置处，将

具有 ID(q)且存在于对象缓冲器 15 中的图形对象 (p) 写入图形平面 8 中 (步骤 S90)。

根据上述循环处理, 将所述多个图形对象中的第 q 个图像 (其分别构成当前按钮的选中状态 (或活动状态) 和其他按钮的正常状态) 写入图形平面 8。

步骤 S91 用于判断 `start_object_id_normal + q` 是否达到 `end_object_id_normal`。如果判断结果为否定, 则变量“动画 (p)”将增加变量 q (步骤 S92)。如果判断结果为肯定, 则对变量“动画 (p)”进行初始化成具有值“0” (步骤 S93)。对 ICS 中的所有 `button_info` 重复上述处理 (步骤 S81, 步骤 S82)。在完成对 ICS 中的所有 `button_info` 的处理之后, 操作返回到主程序。

通过执行步骤 S80 到步骤 S93, 每次完成步骤 S35 到步骤 S37 的循环时, 都将交互式显示中的各按钮的设计图像更新为新的图形对象。如果多次执行步骤 S35 到步骤 S37 的循环, 则实现了所谓的动画。在所述动画中, 图形控制器 17 执行时间调整, 以便将一帧图形对象的显示时间间隔设定为由 `animation_frame_rate_code` 表示的值。

在上述说明中, `button_info (p)` 中的按钮命令是依次执行德。然而, 也可以在对应于活动状态的所有图形对象全部显示之后, 再执行包括在 `button_info (p)` 中的所有按钮命令。接下来参照图 46 来说明在主程序中的图 37 的 UO 处理的处理程序。

图 46 是表示 UO 处理的处理程序的流程图。在该流程图的处理中, 判断是否出现步骤 S100 到步骤 S103 的任何一个现象, 如果判断结果为肯定, 则执行相应的处理, 并且操作返回到主程序。步骤 S100 用于判断是否将 `UomaskTable` 设置为 1。如果判断结果为肯定, 则操作返回到主程序, 不执行任何处理。

步骤 S101 用于判断是否按压了上移 / 下移 / 左移 / 右移键中的任一个。如果判断结果为肯定, 则改变当前按钮 (步骤 S104), 并且

判断当前按钮的 `auto_action_flag` 是否为 01（步骤 S108）。如果判断结果为否定，则操作返回到主程序。如果判断结果为肯定，则操作移到步骤 S105。

步骤 S102 用于判断是否已按压了活动按键。如果判断结果为肯定，则将当前按钮 `i` 改变为活动状态（步骤 S105）。而后，将变量“动画 (`i`)”设置为 0（步骤 S106）。

步骤 S103 用于判断是否有数值输入。如果判断结果为肯定，则执行数值输入处理（步骤 S107），并且操作返回到主程序。在图 46 的处理程序中，步骤 S104 和步骤 S107 分别具有与子程序相同的结构。具体的子程序处理程序在图 47 和图 48 中示出。下面将说明这些流程图。

图 47 是表示当前按钮变化处理的过程的流程图。首先，在当前按钮的 `neighbor_info` 中，识别对应于被压按键的 `upper_button`、`lower_button_number`、`left_button_number` 和 `right_button_number` 其中之一（步骤 S110）。

然后，将当前按钮设置为“按钮 `i`”，并将新的当前按钮设置为“按钮 `j`”（步骤 S111）。步骤 S112 用于判断在步骤 S111 设置的按钮 `j` 是否与按钮 `i` 相同。如果它们相同，则操作返回到主程序，而不执行任何处理。如果它们彼此不同，则将按钮 `j` 设定为当前按钮（步骤 S113），并且将变量“动画(`i`)”和“动画(`j`)”设置为 0。然后，操作返回到主程序（步骤 S114）。

图 48 是表示数值输入处理程序的流程图。判断是否存在具有与输入数值相对应的 `button_number` 的按钮信息 `j`（步骤 S121），并且判断按钮信息 `j` 的 `numerically_selectable_flag` 是否为 1（步骤 S122）。如果步骤 S121 和步骤 S122 为是，则将当前按钮改变到正常状态，并且将按钮 `j` 设定为当前按钮（步骤 S123），并且将变量“动画(`i`)”和“动

画(j)”设置为 0(步骤 S124)。然后,判断按钮信息 j 的 auto_action_flag 是否为 1(步骤 S125)。如果不为 1,则操作返回到主程序。

如果为 1,则在步骤 S126 将当前按钮改变到活动状态,然后操作返回到主程序。

如果步骤 S121 到 S123 中之一为否,则操作返回到主程序。

当试图使用作为触发器(例如弹出显示)的用户操作来执行交互式屏幕显示时,流图形处理器 14 和图形控制器 17 执行下面的处理,这与同步显示执行的处理相同。基于此,在图形平面 8 中获得图形对象。以这种方式获得图形对象之后,一直等待,直到当前再现时间到达分配给 ICS 的 PTS 所示的时间为止。而后在再现时间之后,一旦控制器 18 接收到表示菜单调用的 UO,就从图形平面 8 将图形对象输出到 CLUT 单元 9,以便进行组合。通过执行这种与 UO 同步的输出,实现了根据该菜单调用按压的弹出显示。

上面说明了如何设置 ICS 中的 PTS 以及 ODS 中的 DTS 和 PTS,其中 ICS 和 ODS 属于 DS_n。然而,仍然没有说明 ICS 中的 DTS、PDS 中的 PTS 以及 END 中的 DTS、PTS。下面解释这些时戳。应当在 DS_n 中的第一 ODS (ODS1) 的解码开始时间(即 DTS(DS_n) [ODS1]) 之前,并且在 DS_n 中的第一 PDS (PDS1) 变得有效(即 PTS(DS_n) [PDS1]) 之前,将 ICS 加载到合成缓冲器 16。具体而言,应当满足下面的关系。

$$DTS(DS_n [ICS]) \leq DTS(DS_n [ODS1])$$

$$DTS(DS_n [ICS]) \leq PTS(DS_n [PDS1])$$

下面说明属于 DS_n 中的各个 PDS 中的 DTS、PTS 的设置。

在将 ICS 加载到合成缓冲器 16 的时间(DTS(DS_n) [ICS]) 和第一 ODS 的解码开始时间(DTS(DS_n) [ODS1]) 之间,属于 DS_n 的各个 PDS 应当在 CLUT 单元 9 中有效。因此,对于 PDS(PDS1-PDSlast) 的各个 PTS 值来说,应当满足下面的关系。

$$DTS(DS_n [ICS]) \leq PTS(DS_n [PDS1])$$

$$PTS(DS_n [PDS_j]) \leq PTS(DS_n [PDS_{j+1}]) \leq PTS(DS_n [PDS_{last}])$$

$$PTS(DS_n [PDS_{last}]) \leq DTS(DS_n [ODS1])$$

应当注意的是，即使没有在再现时提到，也可以将 PDS 的 DTS 设置为与对应 PTS 的值相同，以便符合 MPEG2 标准。

下面说明在对 DTS 和 PTS 进行设置以满足上述关系时再现装置的流水线中的 DTS 和 PDS 的含义。图 49 是基于 ICS 中的 DTS 和 PDS 中的 PTS 来表示再现装置中的流水线处理的示意图。图 49 是基于图 33 来创建的。图 33 的第三行中的“读取到编码数据缓冲器 13”对应于该图中的第五行。另外，第二行中的“通过流图形处理器 14 进行解码”对应于该图中的第四行。此外，设置 ICS 和 PTS，以便满足上面给出的关系。

图 49 中的第二行是 CLUT 单元 9 的 PDS 设置。第三行表示合成缓冲器 16 的存储内容。ICS 中的 DTS 被设置为比 PDS 的 DTS 或 ODS 的 DTS 要早的时间。因此，如该图中的箭头 up1 所示，首先执行 ICS 到合成缓冲器 16 的加载。另外，在 ICS 传送之后和 ODS1 解码之前，执行对 CLUT 单元 9 的 PDS1-last 的设置，如图中的箭头 up2 和 up3 所示。

从上述内容可以清楚地看出，在 ODS 解码之前进行 ICD 的加载及 PDS 的设置。

接下来描述属于 DS_n 的“显示集合段 END”的 PTS 的设置。 DS_n 的 END 表明 DS_n 的结束，因此表明 DS_n 的最后一个 ODS (ODS_{last}) 的解码结束时间。该解码结束时间用 $ODS_{last} (PTS(DS_n [ODS_{last}]))$ 的 PTS 来表示。因此，END 中的 PTS 应当满足下面的关系。

$$PTS(DS_n [END]) = PTS(DS_n [ODS_{last}])$$

就属于 DS_n 和 DS_{n+1} 的 ICS 来说，在第一 ODS ($ODS1$) 的加载时间之前，将 DS_n 中的 ICS 加载到合成缓冲器 16 中，因此 END

的 PTS 应当在属于 DS_n ($DTS(DS_n) [ICS]$) 的 ICS 的加载时间之后, 并且在属于 DS_{n+1} ($DTS(DS_{n+1}) [ICS]$) 的 ICS 的加载时间之前。

$$DTS(DS_n [ICS]) \leq PTS(DS_n [END]) \leq DTS(DS_{n+1} [ICS])$$

另一方面, 第一 ODS (ODS_1) 的加载时间是在最后一个 PDS (PDS_{last}) 的加载时间之后。因此 END 的 PTS ($PTS(DS_n[END])$) 应当在属于 DS_n (其属于 DS_n ($PTS(DS_n [PDS_{last}])$)) 的 PDS 的加载时间之后。因此, END 的 PTS 应当满足下面的关系。

$$PTS(DS_n [PDS_{last}]) \leq PTS(DS_n [END])$$

接下来说明再现装置的流水线中的 END 的 PTS 的含义。图 50 是表示在再现装置的流水线操作期间 END 的含义的示意图。该图是根据图 33 创建的, 并且其中几乎所有的行都与图 33 中的行相同, 除了第一行表示合成缓冲器 16 的存储内容之外。另外, 图 50 呈现了两个显示集合, 即 DS_n 和 DS_{n+1} 。在 DS_n 中, ODS_{last} 是 A-ODS 的最后一个 ODS_n , 因此将 END 的 PTS 设置为表示该 ODS_n 的 PTS。由该 END 的 PTS 表示的时间早于 DS_{n+1} 中的 ICS 的 DTS 表示的时间。

根据该 END 的 PTS, 在再现过程中, 可以什么时候完成针对 DS_n 的 ODS 的加载。

应当注意的是, 即使没有在再现时提到, 也可以将 END 的 DTS 设置为与对应 PTS 的值相同, 以便符合 MPEG2 标准。

在上述结构中, 其中设置了 DTS 和 PTS 的 ICS、PDS 和 ODS 是预先结合到 AV 剪辑中的。这对于描述交互式控制是很方便的, 该交互式控制使再现装置执行特定处理, 在屏幕上出现一帧运动图像, 这种交互式控制与运动图像的内容密切同步。另外, 在上述结构中, 将 ICS、PDS 和 ODS 复用到 AV 剪辑本身。因此, 即使作为再现控制目标的部分的数量为几百个时, 也没有必要存储所有相应的 ICS、PDS 和 ODS。将 ICS、PDS 和 ODS 从 BD-ROM 中与视频包一同读

取。随后的处理是读取用于运动图像部分的 ICS、PDS 和 ODS，这些运动图像部分要在存储器中再现，并且在完成该运动图像部分的再现之后，从存储器中删除 ICS、PDS 和 ODS，然后将对应于下一个运动图像部分的 ICS、PDS 和 ODS 的新的集合存储到所述存储器中。根据上述结构，即使在 ICS、PDS 和 ODS 的数量变为几百个时，存储器的占用区域也被限制到最小。

如上所述，根据本发明，有 360 页用于呈现动画的 ODS。在假定存在三个按钮状态集合的情况下，将这些 ODS 分组到三个按钮状态集合（例如 120+120+120 页）下。设置各个按钮状态集合，使得在动画中较早出现的集合离顶点较近，而较晚出现的集合离顶点较远。因此，在再现过程中，将较早出现的按钮状态集合在早期加载到再现装置中，而推迟较晚出现的按钮状态集合的加载。基于此，当一旦完成整个 ODS 的大约 1/3 到 2/3 的读取/解码时，初始显示的准备已就绪，即便并非 360 页的 ODS 完成解码。这进一步意味着，在存在大量待读取/解码 ODS 的情况下，不会延迟初始显示的执行。基于此，可以在没有延迟的情况下利用动画执行交互式显示。

（第二个实施例）

该实施例涉及 BD-ROM 的生产方法。图 51 是表示与第二实施例相关的 BD-ROM 的生产方法的流程图。

BD-ROM 的生产方法包括：材料生成过程 S201，其中执行诸如运动图像拍摄和录音的材料创建；创作过程 S202，其中使用创作装置来生成应用格式；以及印制过程 S203，其中通过印制和层压来创建 BD-ROM 的主盘，从而完成 BD-ROM。

在这些过程中，针对 BD-ROM 的创作过程包括下面的步骤，步骤 S204 到步骤 S209。

首先，在步骤 S204 中，在多种游程编码方法的多个图形数据中，逐帧地创建与按钮状态相对应的帧的动画。

在步骤 S205 中，将创建的多个图形数据分组到按钮状态下。然后创建 ICS，其中每条按钮信息指定一个图形数据的标识符。在步骤 S206 的创建过程中，将默认选中按钮的设置和如何改变按钮状态写入 ICS 中。在步骤 S207 中，将 ICS 和分组的图形数据结合到图形流中。一旦生成图形流，就在步骤 S208 中，将图形流和独立于图形流而创建的视频流和音频流进行复用，以获得 AV 剪辑。在步骤 S209（静态情况、活动情况）中，一旦获得了 AV 剪辑，就使 AV 剪辑遵循 BD-ROM 格式，从而完成一种应用格式。

如上所述，根据该实施例，通过在创作过程中执行图形数据的分组来产生第一实施例中所述的图形流。对于在第一实施例中描述的 BD-ROM 来说，这是一种很容易生成应用格式的方法。

（第三个实施例）

第一个实施例的交互式图形流通过覆盖到 AV 剪辑上而记录到 BD-ROM 上。但是，在第三个实施例中，交互式图形流作为独立于 AV 剪辑的子剪辑而记录到 BD-ROM 上，播放列表信息用于把 AV 剪辑和子剪辑关联起来。

在播放列表信息中，由播放项目信息形成的再现路径被称为“主路径”。相比之下，表示子剪辑的信息被称为子路径信息，其定义的再现路径被称为“子路径”。

图 52 是第三个实施例相关的 PL 信息的内部结构的示意图。

图 52 中的箭头 hc1 聚焦在每条子路径信息上。如箭头 hc1 所示，一条子路径信息由一个或多个子播放项目构成。如图中的箭头 hc1 所示，每个子播放项目包括“Clip_information_file_name”、“Clip_codec_identifier”、“Sub PlayItem_In_Time”、“Sub PlayItem_Out_Time”、“Sync_PlayItem_id”和“Sync_start_PTS_of PlayItem”。

“Clip_information_file_name”描述剪辑信息的文件名，它唯一地标识与该子播放项目相对应的子剪辑。

“Clip_codec_identifier”表示该 AV 剪辑是用哪种编码方法编码的。

“Sub_PlayItem_In_Time”表示该子播放项目在该子剪辑的再现时间轴上的开始点。

“Sub_PlayItem_Out_Time”表示该子播放项目在该子剪辑的再现时间轴上的结束点

“Sync_PlayItem_id”唯一地标识构成主流的一个播放项目，该子播放项目要与之同步。“Sub_PlayItem_In_Time”存在于该 Sync_PlayItem_id 标识的播放项目的再现时间轴上。

“Sync_start_PTS_of_PlayItem”表示由该 Sub_PlayItem_In_Time 标识的子播放项目的开始点处于由 Sync_PlayItem_id 标识的播放项目的再现时间轴上哪个位置。在该播放项目的再现过程中，在当前再现时间到达该 Sync_start_PTS_of_PlayItem 时，该子播放项目的再现开始。

图 53 是子播放项目根据 Sync_PlayItem_id 和 Sync_start_PTS_of_PlayItem 的同步的示意图。该图中的主路径包括播放项目#1、#2 和#3。虚线“wh1”画出的框内示出了构成该子路径的子播放项目的 Sync_PlayItem_id 和 Sync_start_PTS_of_PlayItem。具体而言，将该 Sync_PlayItem_id 设置为表示播放项目#1，而将 Sync_start_PTS_of_PlayItem 设置为表示该播放项目在再现时间轴上的时间 t1。以这种方式，在当前再现时间到达播放项目的再现时间轴上的 t1 时，将再现该 Clip_information_file_name 标识的子剪辑的一部分（交互式图形流），该部分具体对应于从 Sub_PlayItem_In_time 到 Sub_PlayItem_Out_time 的部分。通过以上方式执行再现，交互式图形流中从 Sub_PlayItem_In_time 到 Sub_PlayItem_Out_time 的对应部分将与该 AV 剪辑同步再现。

此外，可以将作为子剪辑的交互式图形流记录在再现装置中包括的硬盘上。在这种情况下，如果硬盘上的播放列表信息对应于

BD-ROM 上的 AV 剪辑和硬盘的子剪辑，那么，可以同步地再现 AV 剪辑和子剪辑。图 54 示出了硬盘上的播放列表信息中的“Clip_information_file_name”标识的文件。箭头 rf1、rf2 和 rf3 示出了播放列表信息中的播放项目的 Clip_information_file_name 的表示。箭头 pf1、pf2、pf3 示出了播放列表信息中的子播放项目的 Clip_information_file_name 的表示。根据 Clip_information_file_name 的表示，为 HD 上的子剪辑定义的子路径与为 BD-ROM 上的主剪辑定义的主路径同步再现。

下面描述当播放列表信息中存在要与当前播放项目同步的子播放项目时控制单元 20 执行的处理。控制单元 20 控制子播放项目表示的子剪辑与主剪辑同步再现。该同步再现如下执行。从主剪辑中读取与子播放项目的 Sync_start_PTS_of_PlayItem 相对应的图像数据。然后，将子播放项目的 Sub_PlayItem_In_time 到 Sub_PlayItem_Out_time 之间存在的数据进行再现。

(注意)

上面的描述并没有给出本发明的所有实施例。不必说，还可通过多种修改，包括下面的 (A)、(B)、(C)、(D) 等，来实现本发明。本发明权利要求书相关的各发明是上述实施例或它们的修改实例的扩展或概括性描述。扩展和概括程度取决于提交本发明时本发明技术领域的技术发展水平。请注意，权利要求书中的各个发明反映了解决传统技术的技术问题的手段，因此本发明的范围不会超过本领域普通技术人员可识别的范围。因此，在本发明权利要求书中的各个发明实际上与本发明的说明书相关联。

(A) 上述所有实施例描述都基于这样的假设：根据本发明的记录介质是 BD-ROM。但是，本发明的记录介质的主要特征在于其中记录的图形流，该特征并不依赖于 BD-ROM 的物理特性。因此，本发明同样适用于能够记录活动情形和图形流的任何记录介质。例如，这种记录介质可以是光盘，如 DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R 或 CD-RW。该记录介质也可以

是磁光盘，如 PD 和 MO。该记录介质还可以是半导体存储卡，如 CF 卡、Smartmedia 卡、记忆棒、多媒体卡或 PCM-CIA 卡。此外，该记录介质也可以是：(i) 磁记录盘，如软盘、SuperDisk、Zip 和 Click!；以及(ii)可移动硬盘，如 ORB、Jaz、SparQ、SyJet、EZFley 和 Microdrive。当然，该记录介质还可以是作为装置的构成部件的硬盘。

(B) 在上述所有实施例中，再现装置首先对 BD-ROM 上的 AV 剪辑进行解码，然后将其输出到电视。但是也可以将再现装置设计成 BD 驱动器，其他组成部件由电视来提供。在这种情况下，可以将再现装置和电视集成到通过 IEEE 1394 连接器相连的家庭网络中。此外，根据这些实施例的再现装置连接到电视而被使用。但是，再现装置也可以与显示器集成在一起。此外，可以只将各实施例的再现装置的实质性部分视为本发明的再现装置。因为本发明说明书中描述了所有这样的再现装置，基于第一个实施例的再现装置的内部结构生产任意这些再现装置的行为构成实施说明书所述的本发明的行为。不管是否有偿地转让（如果有偿，即为销售，如果无偿，则为礼物）、出租和进口第一个实施例所涉及的再现装置也是实施本发明的行为。此外，同样，通过橱窗展示、商品目录拉客或分发宣传册来许诺转让和出租的行为也是实施本再现装置的行为。

(C) 使用上述流程图所示的程序的信息处理实际上是用硬件资源实现的。因此，描述这些流程图中所示操作过程的程序本身也是一项发明。在上述实施例所描述的情形中，程序是集成到再现装置中的，但是，这些程序的使用也可以独立于再现装置。实施这些程序的行为包括：(1) 制造行为；(2) 有偿或无偿转让的行为；(3) 出租行为；(4) 进口行为；(5) 经由双向电子通信网络提供给公众的行为；(6) 使用橱窗展示、商品目录或宣传册许诺销售或出租的行为。

(D) 实际上使用硬件资源来实现利用流程图中所示程序的信息处理。因此，描述流程图所示操作过程的程序本身就是发明。上述实施例描述的情形中，将程序结合到再现装置中，但是，可以独立于再现装置来使用所述程序。实施本发明的行为包括：(1) 制造行为；(2) 有偿或无偿转让的行为；(3) 出租行为；(4) 进口行为；(5) 经由双

向电子通信网络提供给公众的行为；(6) 使用橱窗展示、商品目录或宣传册许诺销售或出租的行为。

(E) 当把 AV 剪辑记录到 BD-ROM 上时，可以向 AV 剪辑中的各个 TS 包添加扩展头。该扩展头被称为 TP_extra_header，其包括 arrival_time_stamp 和 copy_permission_indicator，并且数据长度为 4 个字节。具有 TP_extra_header 的 TS 数据包(下文称之为“EX TS 包”)以 32 个数据包为单位进行分组，并且将每组写入 3 个扇区中。由 32 个 EX TS 数据包构成的一组具有 6144 个字节 (=32x192)，这相当于 3 个扇区的尺寸 (6144 个字节= (2048x3))。包含在这 3 个扇区中的这 32 个 EX TS 数据包被称为一个对准单元 (aligned unit)。

在通过 IEEE1394 连接器相连的家庭网络中，再现装置 200 以下面的方式发送对准单元。该再现装置将对准单元中的 32 个 EX TS 数据包中的每一个都删除 TP_extra_header，并且根据 DTCP 规范加密各 TS 数据包的主体，然后输出加密的 TS 数据包。当输出 TS 数据包时，该再现装置将一个同步包插入相邻的 TS 数据包之间。根据 TP_extra_header 的 arrival_time_stamp 所示的时间来确定插入同步包的位置。该再现装置 200 输出 DTCP_descriptor 以及 TS 数据包。DTCP_descriptor 对应于 TP_extra_header 中的 copy_permission_indicator。通过提供表明“禁止复制”的 DTCP_descriptor，当在 IEEE 连接器连接的家庭网络中使用 TS 数据包时，可以防止将 TS 数据包记录到其他设备上。

(F) 在上述实施例所描述的情形中，将 BD-ROM 格式的 AV 剪辑用作数字流，但本发明还可以用 DVD 视频格式或 DVD 视频记录格式的 VOB (视频对象) 来实现。VOB 是符合 ISO/IEC13818-1 标准的节目流，它是通过复用视频流和音频流而获得的。同样，AV 剪辑中的视频流也可以是 MPEG4 视频流或 WMV 视频流。此外，AV 剪辑中的音频流可以是线性 PCM 音频流、Dolby AC-3 音频流、MP3 音频流或 MPEG-AAC 音频流。

(G) 上述实施例中描述的视频编辑可以是对经过模拟广播传输的模拟图像信号进行编码而获得的。

或者，可以通过对记录在录像带上的模拟/数字图像信号进行编码来获得内容。此外，还可以通过对视频照相机直接捕获的模拟/数字图像信号进行编码来获得内容。也可以采用通过分发服务器分发的数字作品。

(H) 上述实施例中描述的图形对象是游程编码的栅格数据。游程编码用于压缩和编码图形对象，因为游程编码适于压缩和解压缩字幕。字幕的属性在于，水平方向上的相同像素值的连续长度较长。因而，通过使用游程编码进行压缩，可以获得高压缩率。此外，游程编码降低了解压缩的负担，因此适于用软件实现解密。但是，图形对象使用游程编码对于本发明不是实质性的。例如，图形对象可以是 PNG 数据。此外，图形对象不必是栅格数据，也可以是矢量数据。此外，图形对象可以是透明图案。

(I) 如果多个再现路径彼此相连，并且默认选中按钮根据要采用的再现路径而变化，则下面的安排是优选的。即，描述动态情形中的再现控制，以使得：经过各再现路径时，在再现装置的寄存器中设置再现通道的特征值，并且描述再现处理，以使得根据所述寄存器中设置的值，将按钮设置到选中状态。通过这种安排，可以根据要经过的再现路径来改变要处于选中状态的按钮。

(J) 当 BD-ROM 或硬盘具有多个交互式图形流时，其中每个图形流都是子剪辑，则可以在播放列表信息的播放项目信息内提供 STN_Table，该 STN_Table 表示应该选择哪一个交互式图形流。此外，还可以使用播放列表信息内的 STN_Table 中的条目，用于在这些交互式图形流之间设置优先级。

工业应用性

根据本发明的记录介质和再现装置实现了对电影作品的交互式控制，因此有助于为市场提供高附加值的电影作品，这有助于为电影市场和消费品市场注入活力。因此，根据本发明的记录介质和再现装置在电影工业和消费品工业中具有很高的实用性。

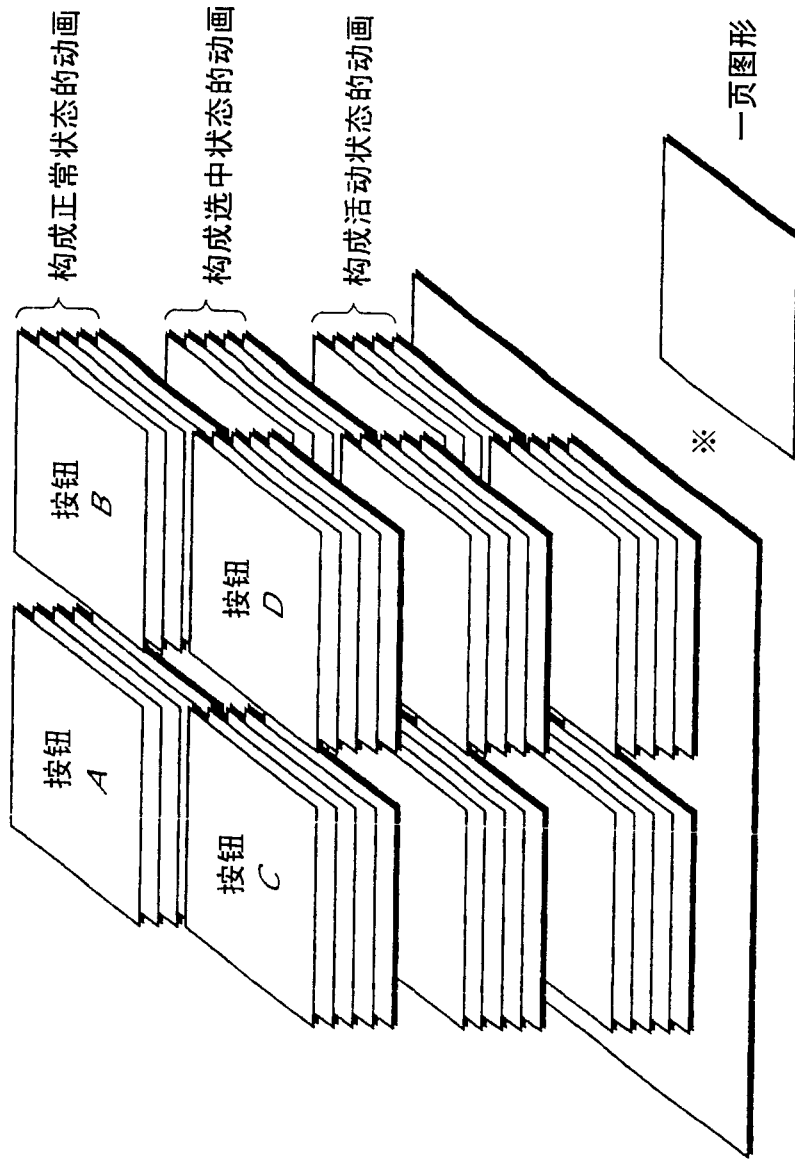


图1

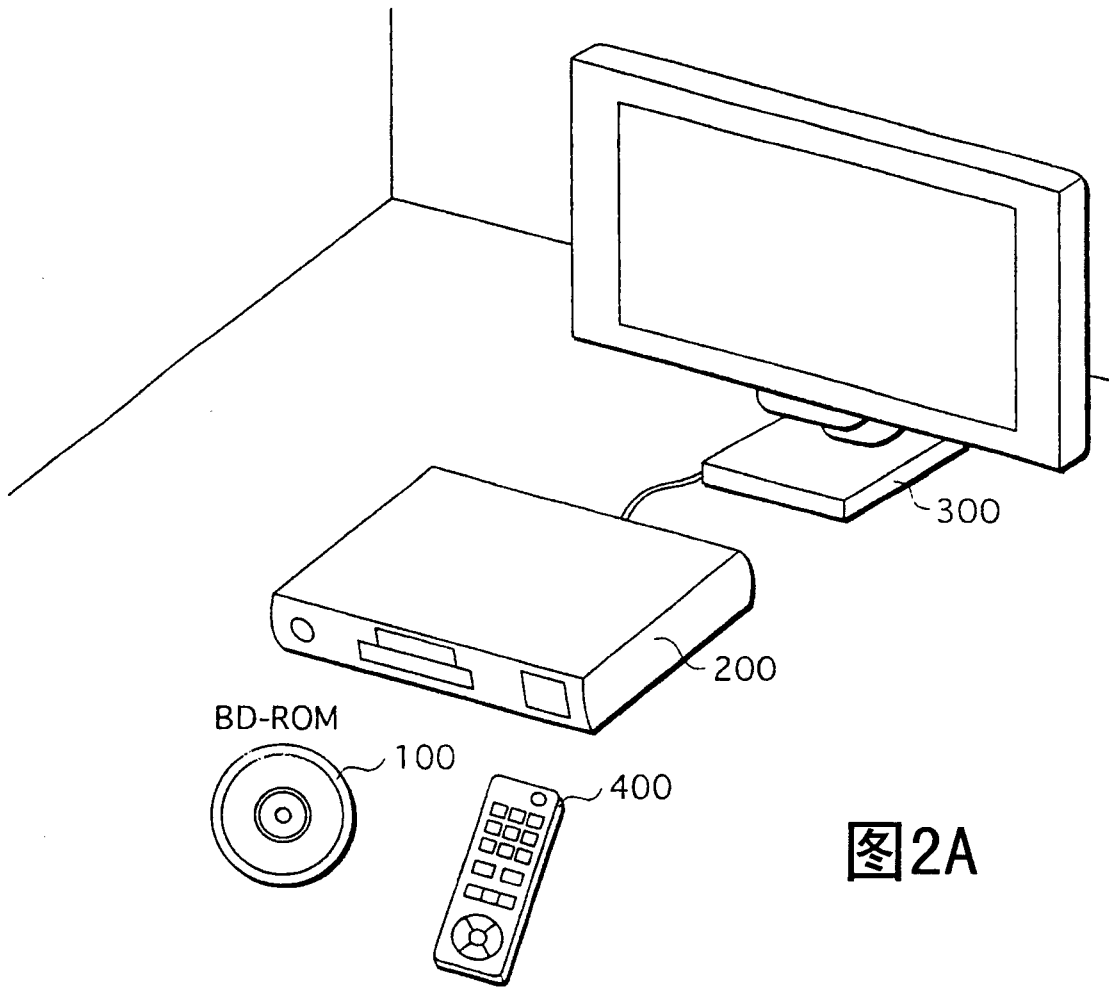


图2A

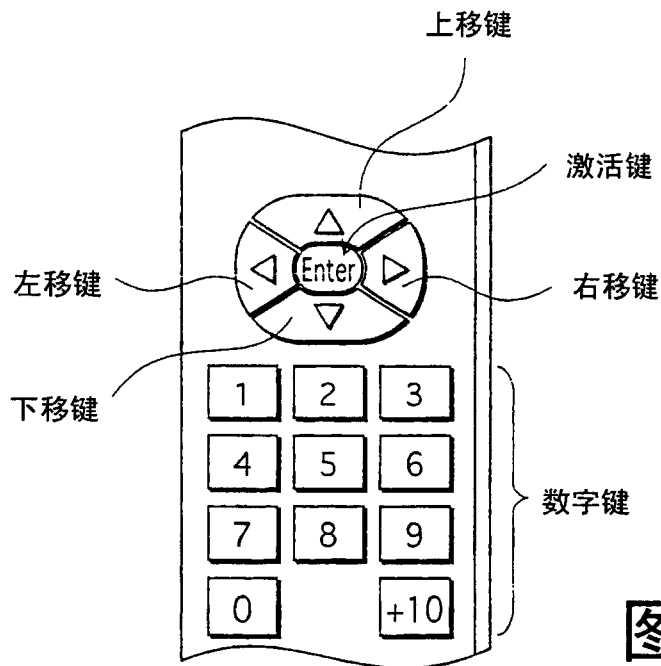


图2B

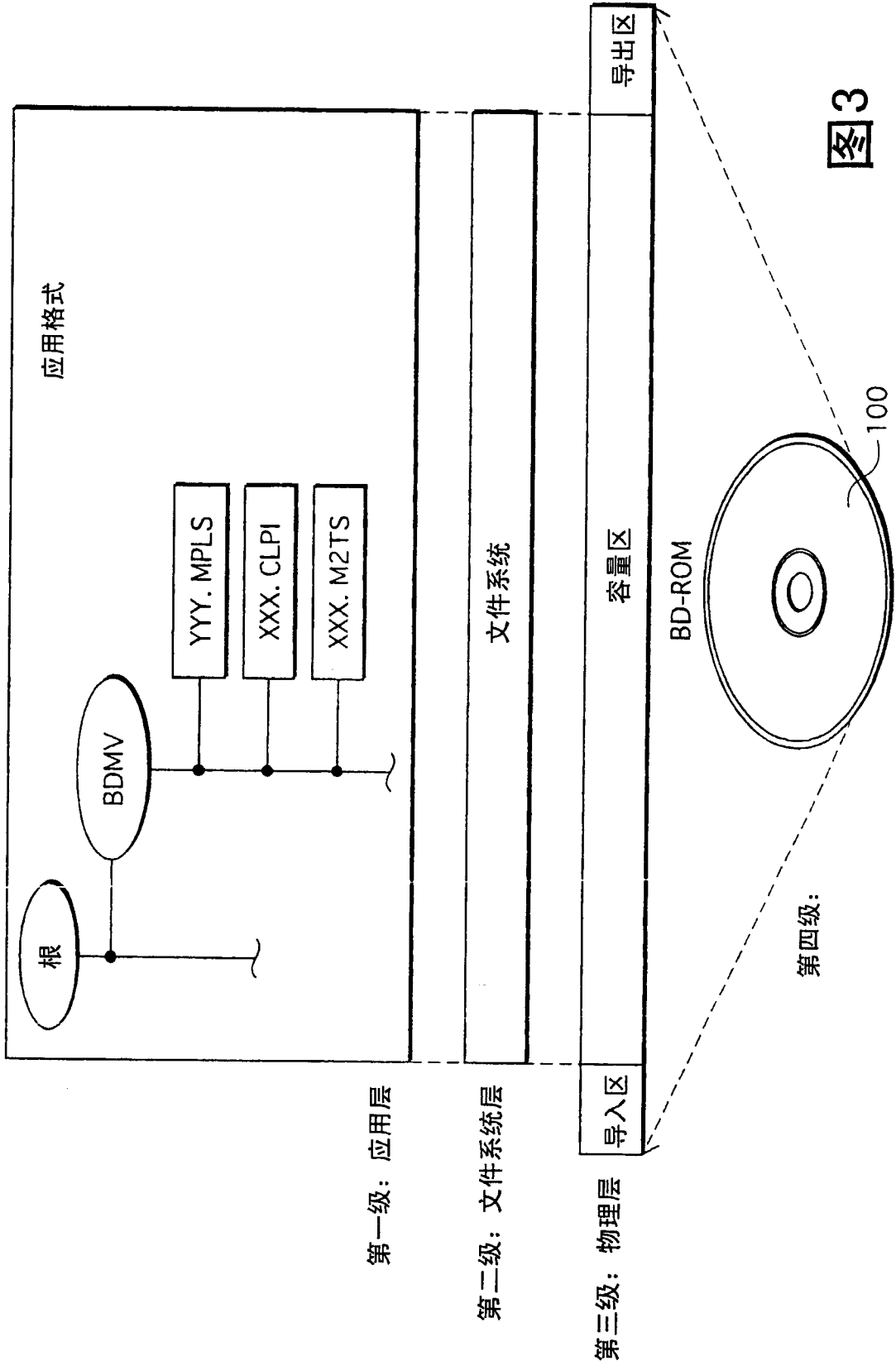


图3

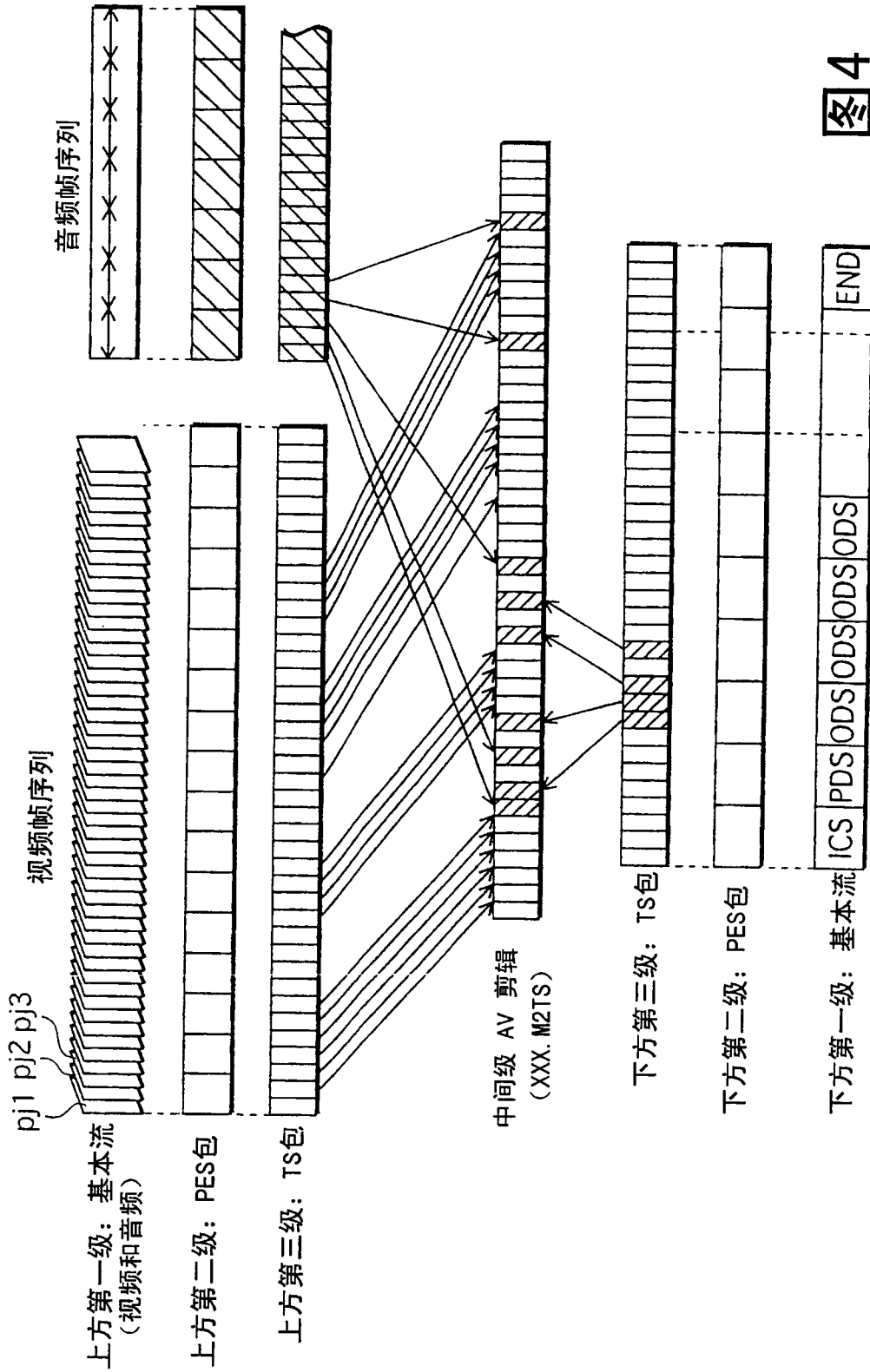


图4

交互图形流

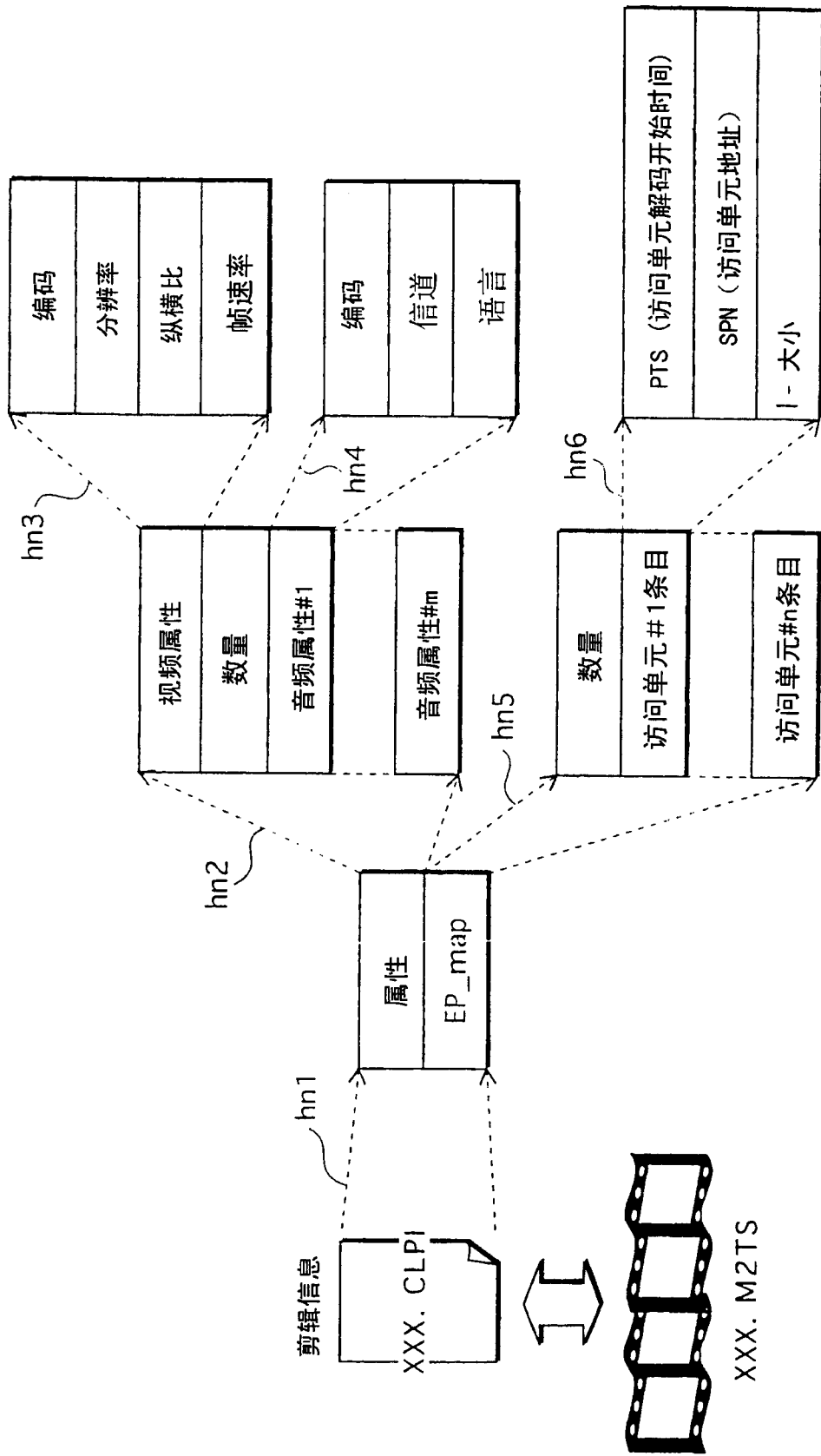


图5

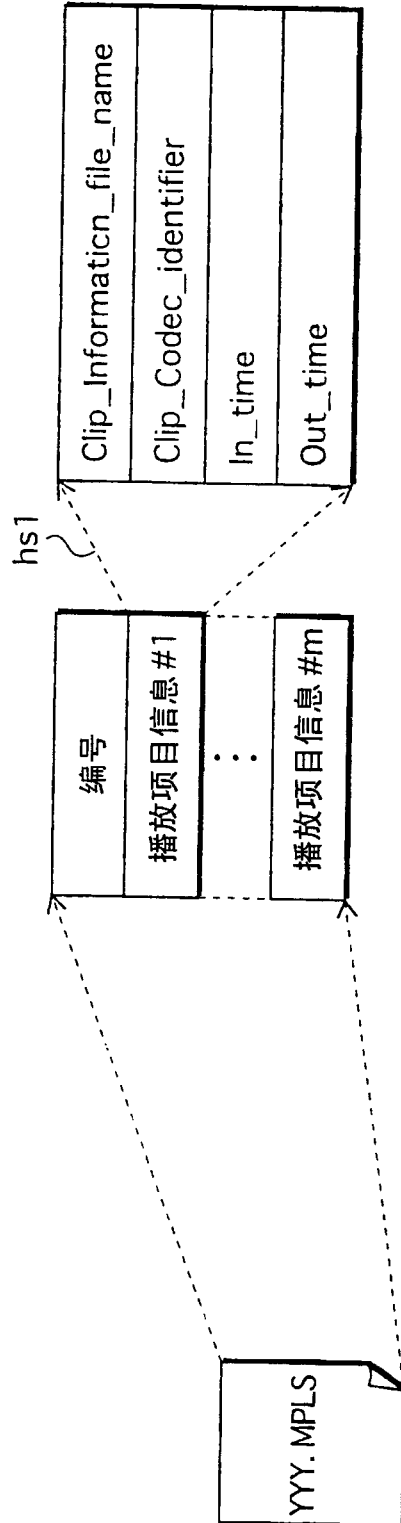


图6

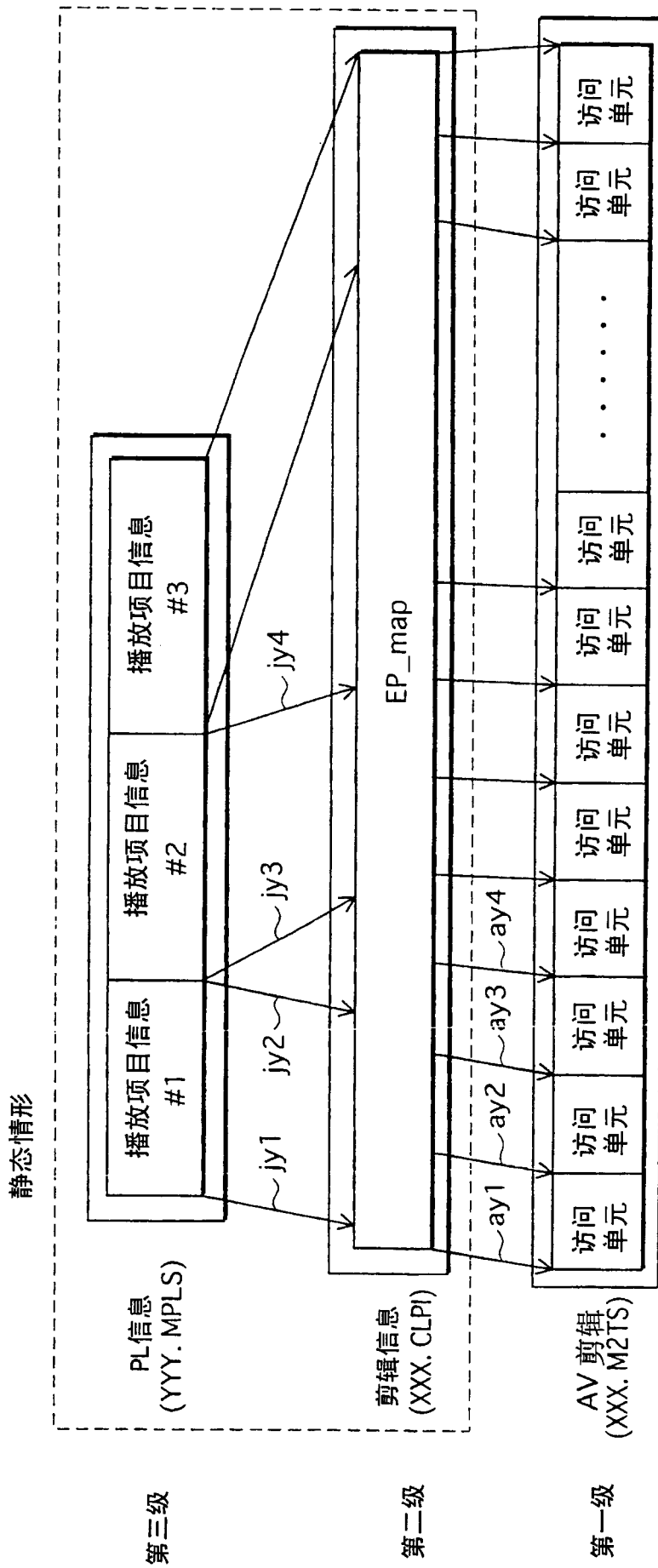


图7

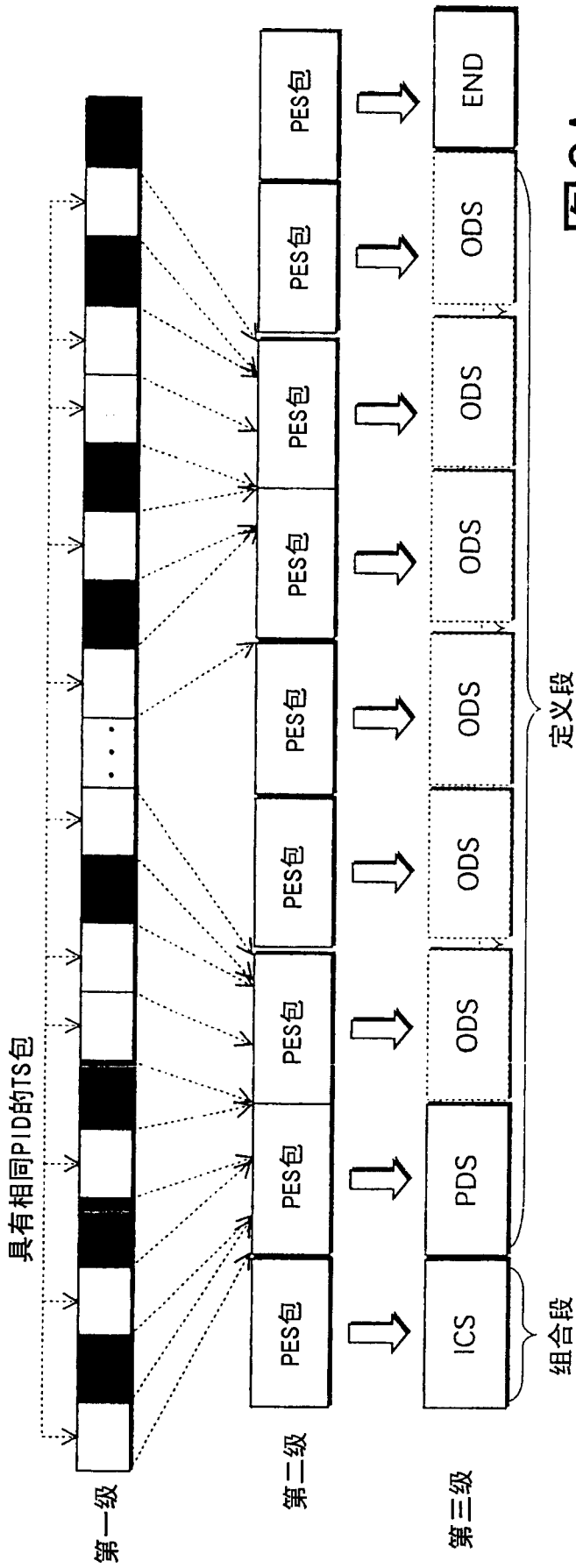


图8A

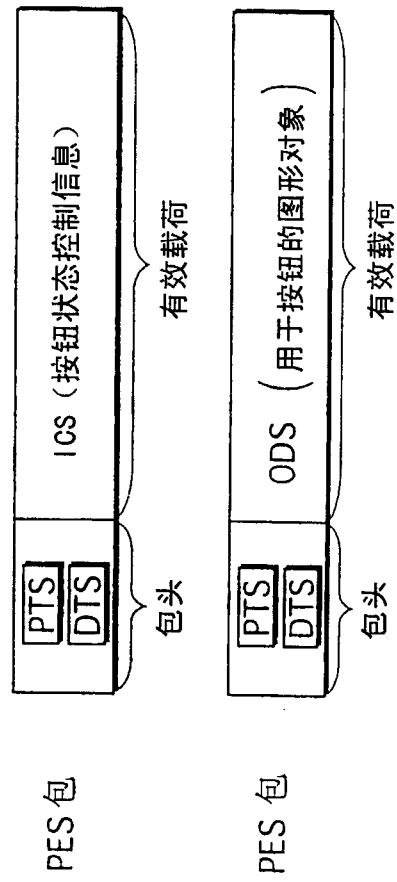


图8B

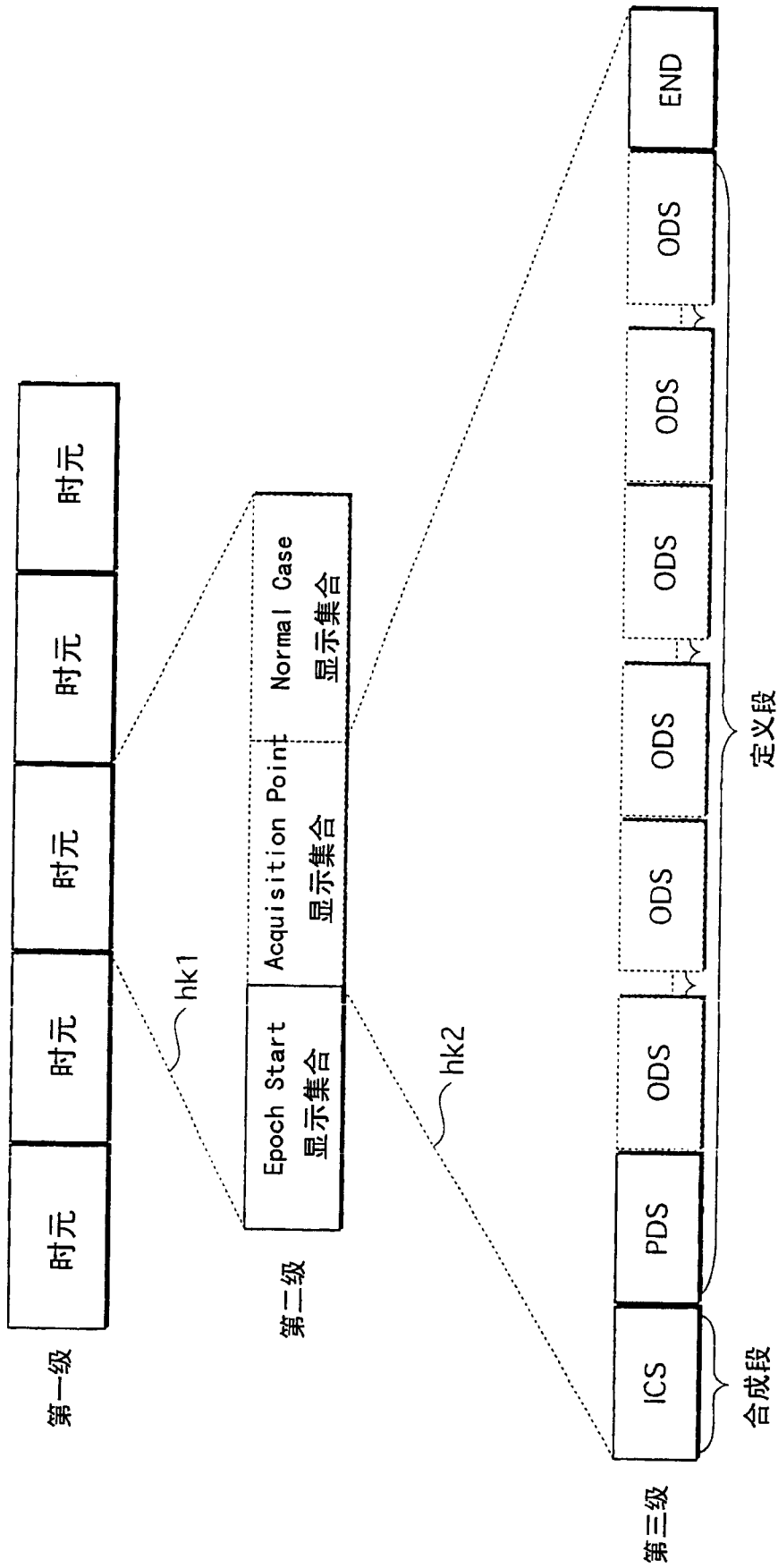


图9

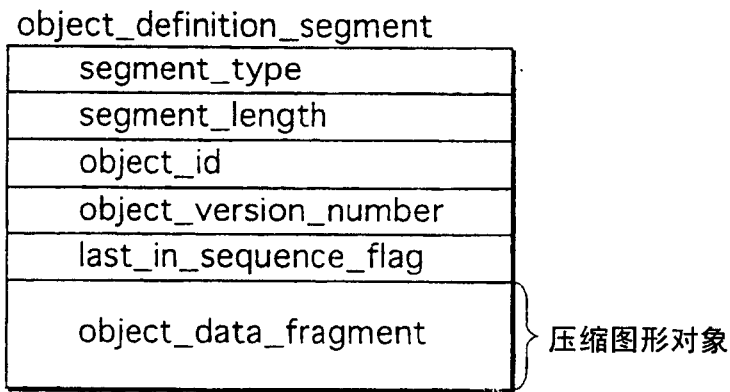


图10A

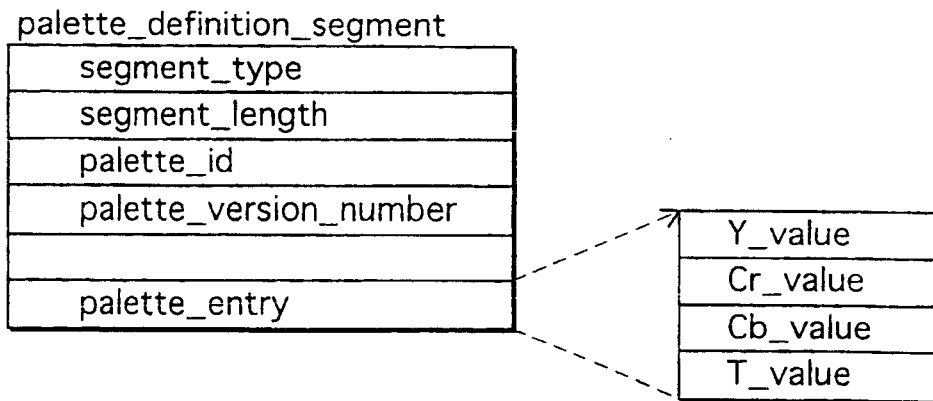


图10B

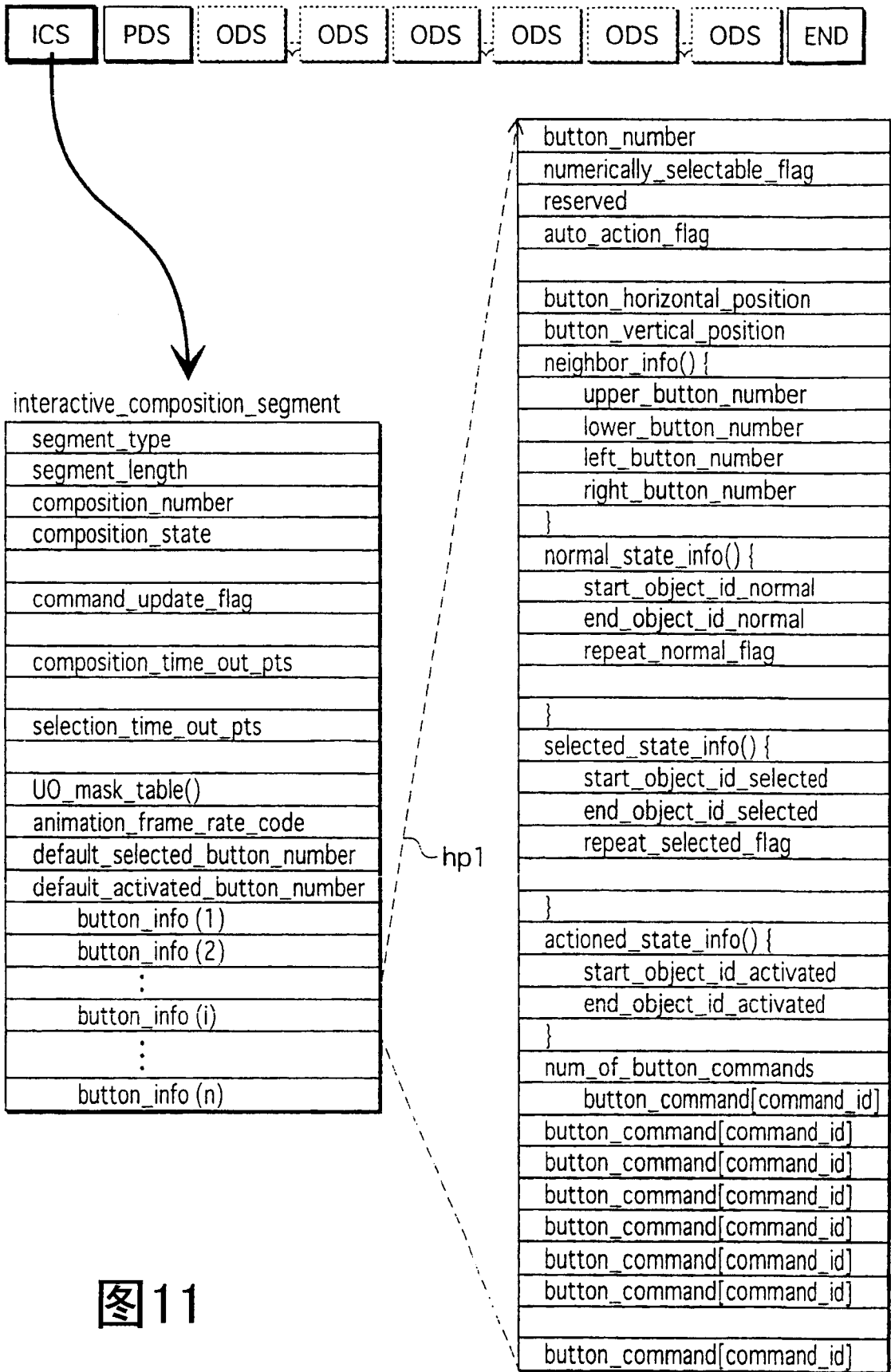


图 11

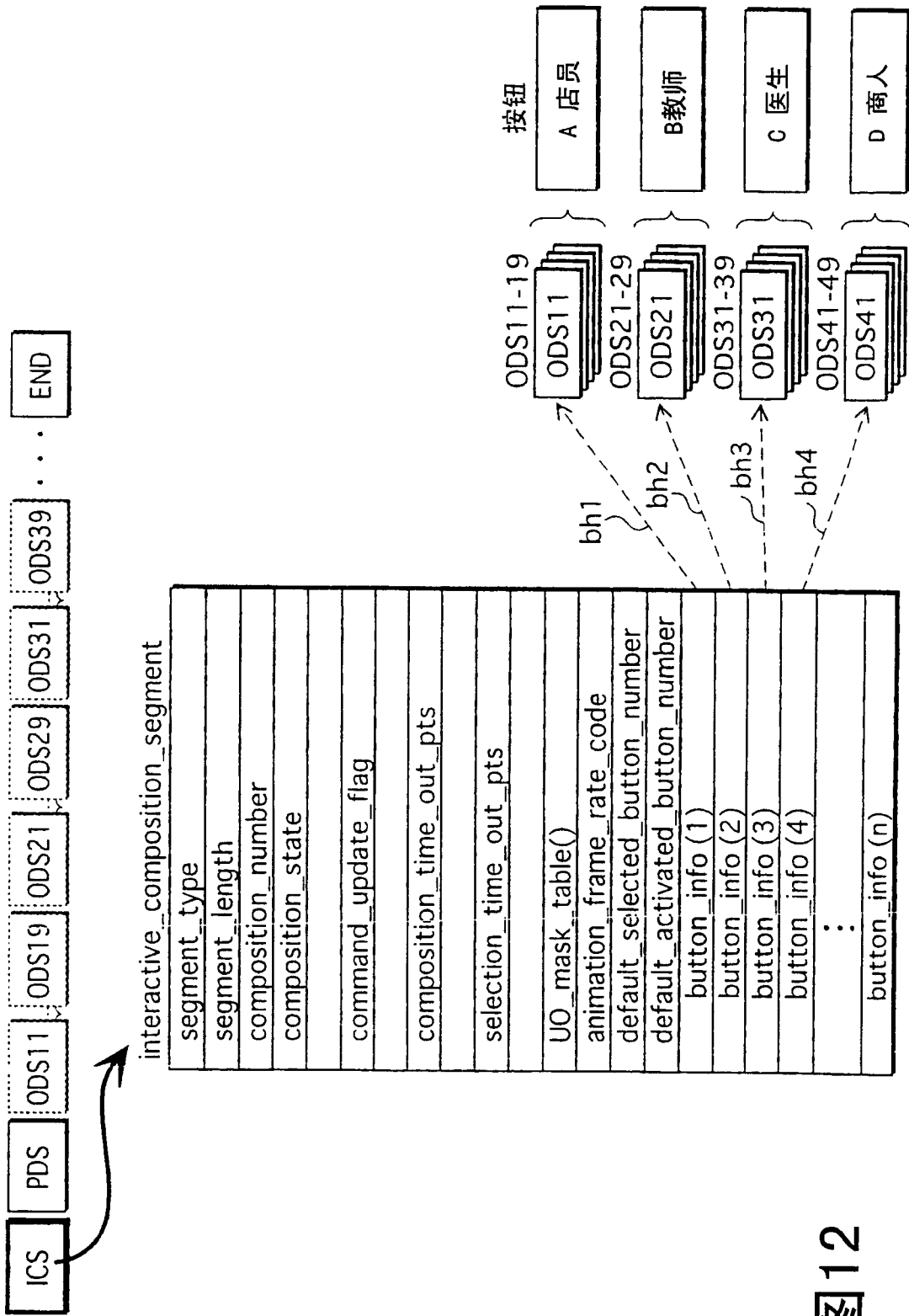


图12

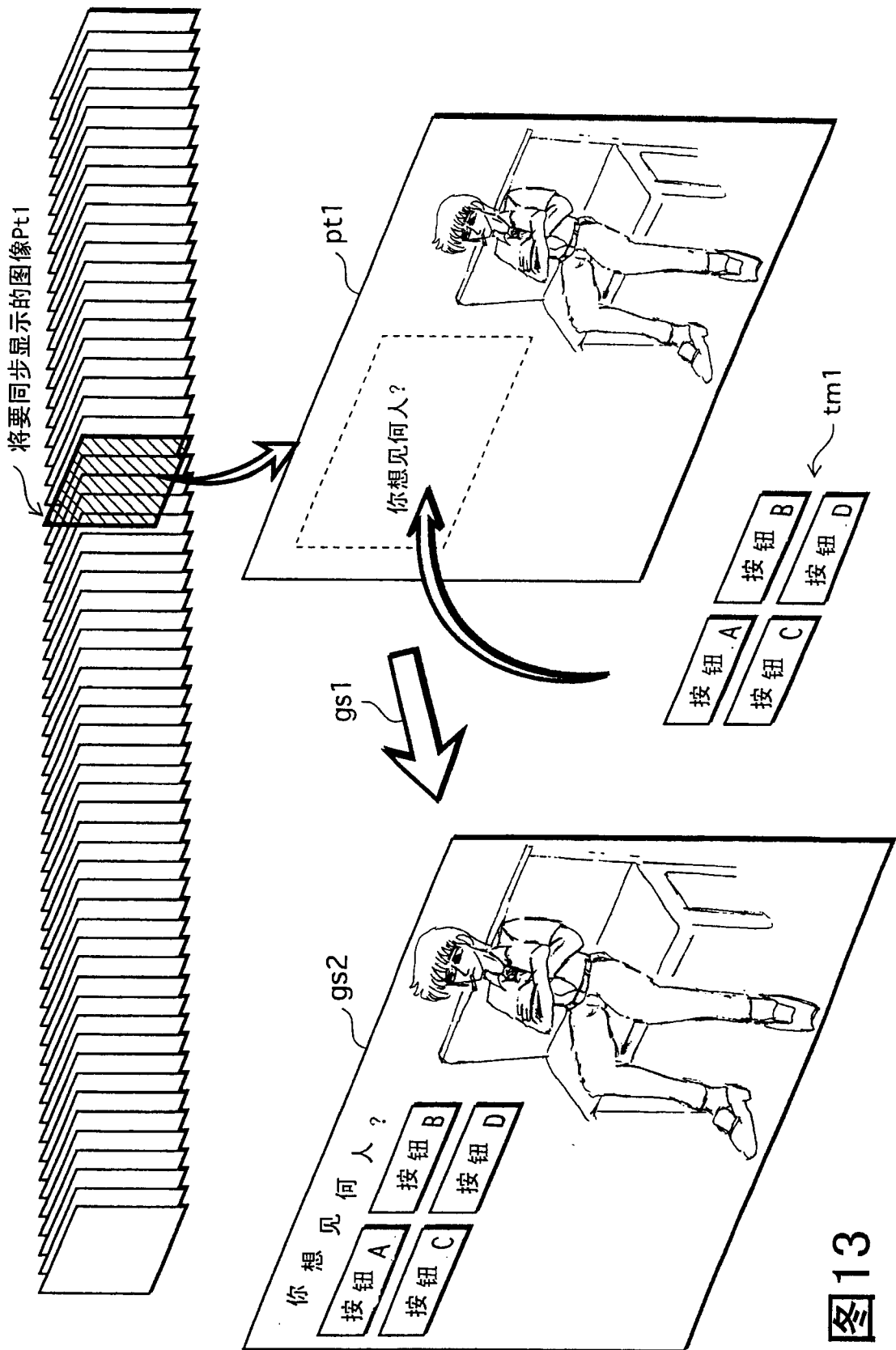


图13

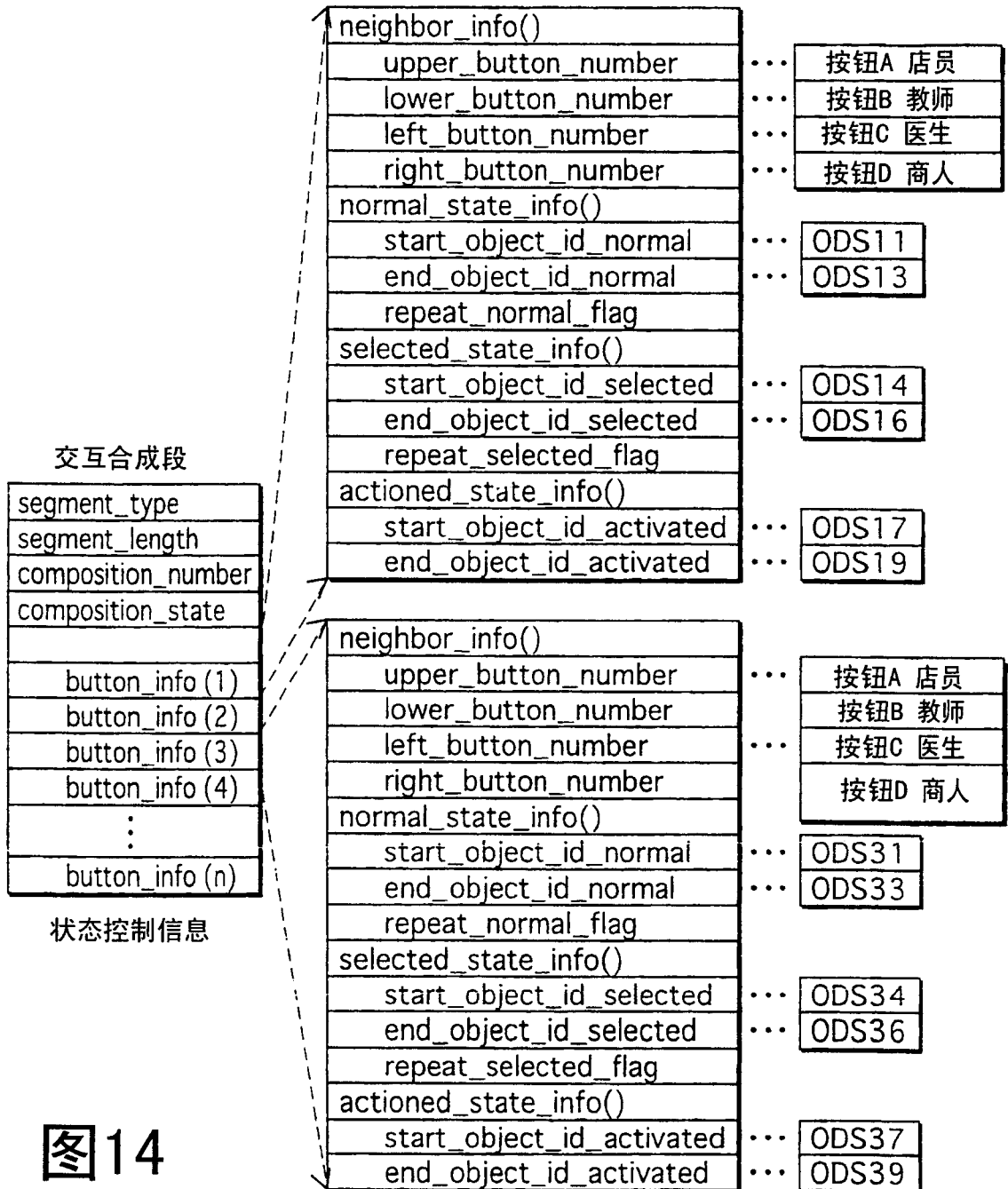


图14

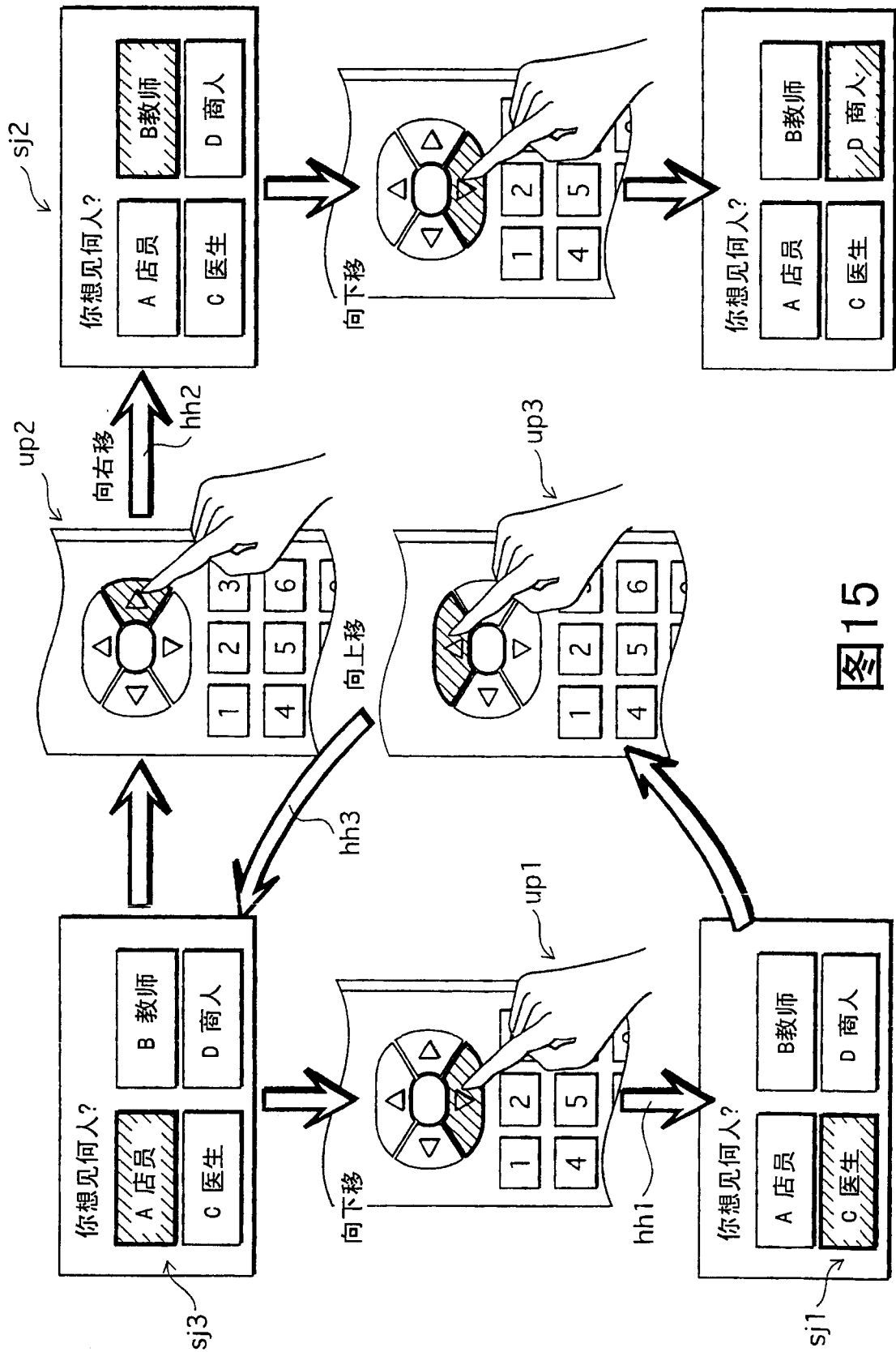


图15



图16

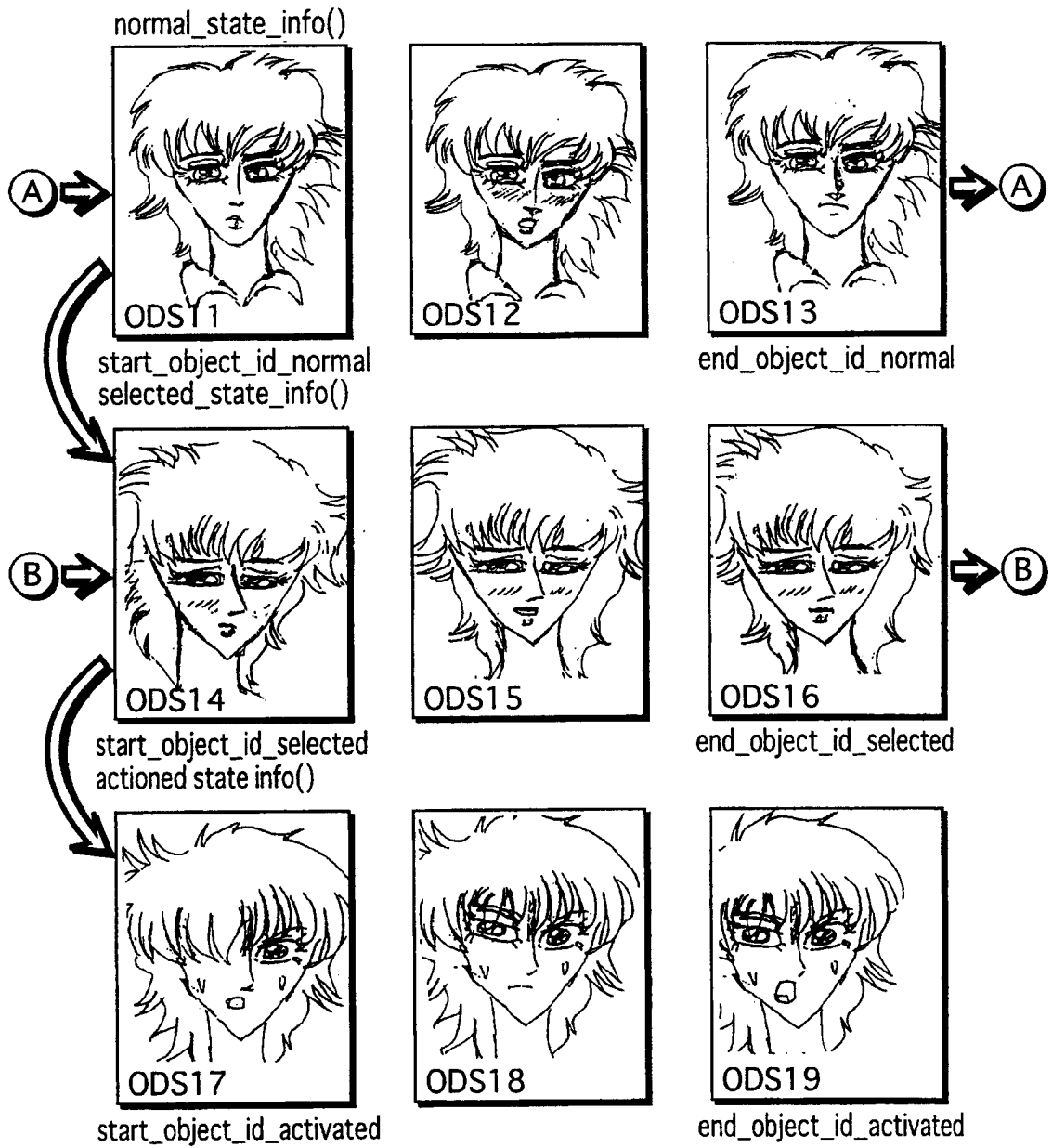


图17

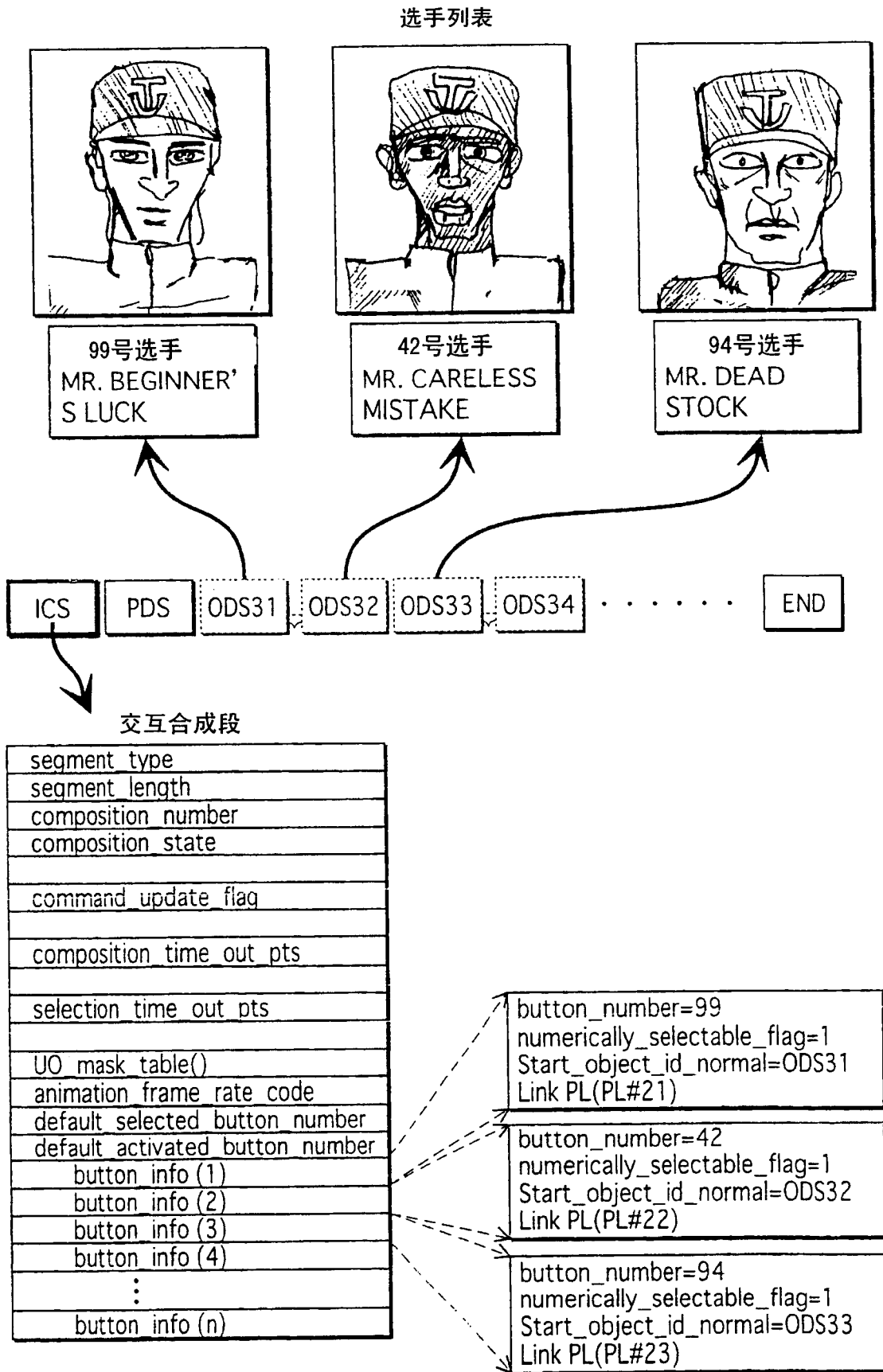


图18

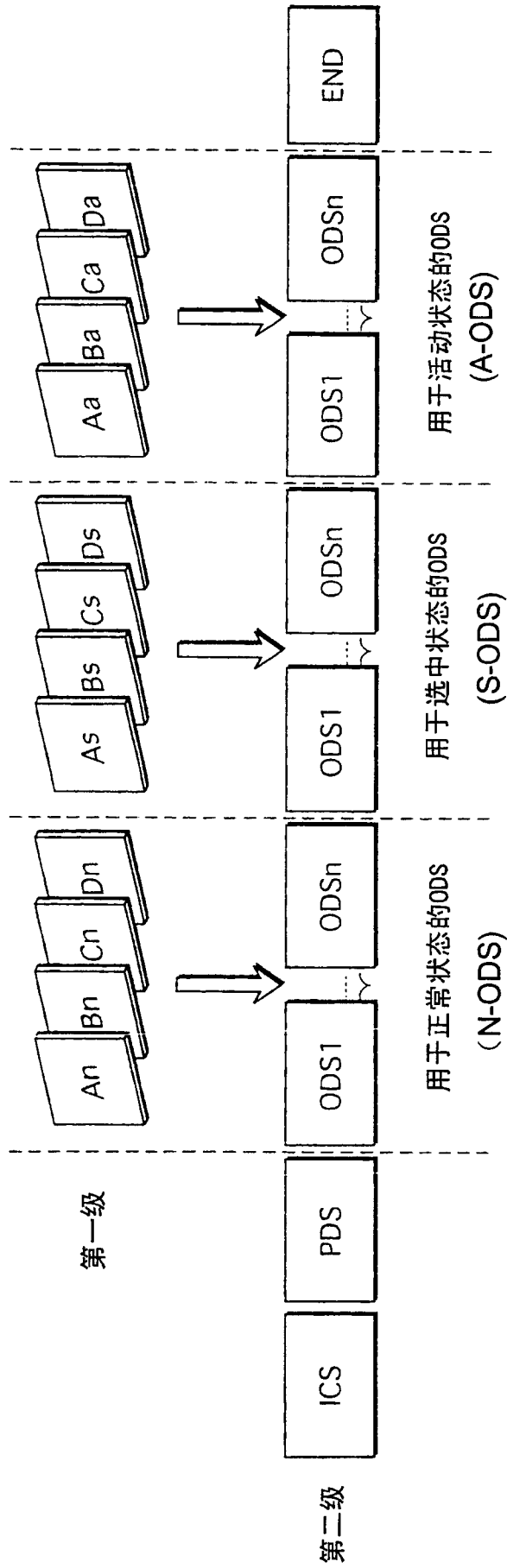


图19

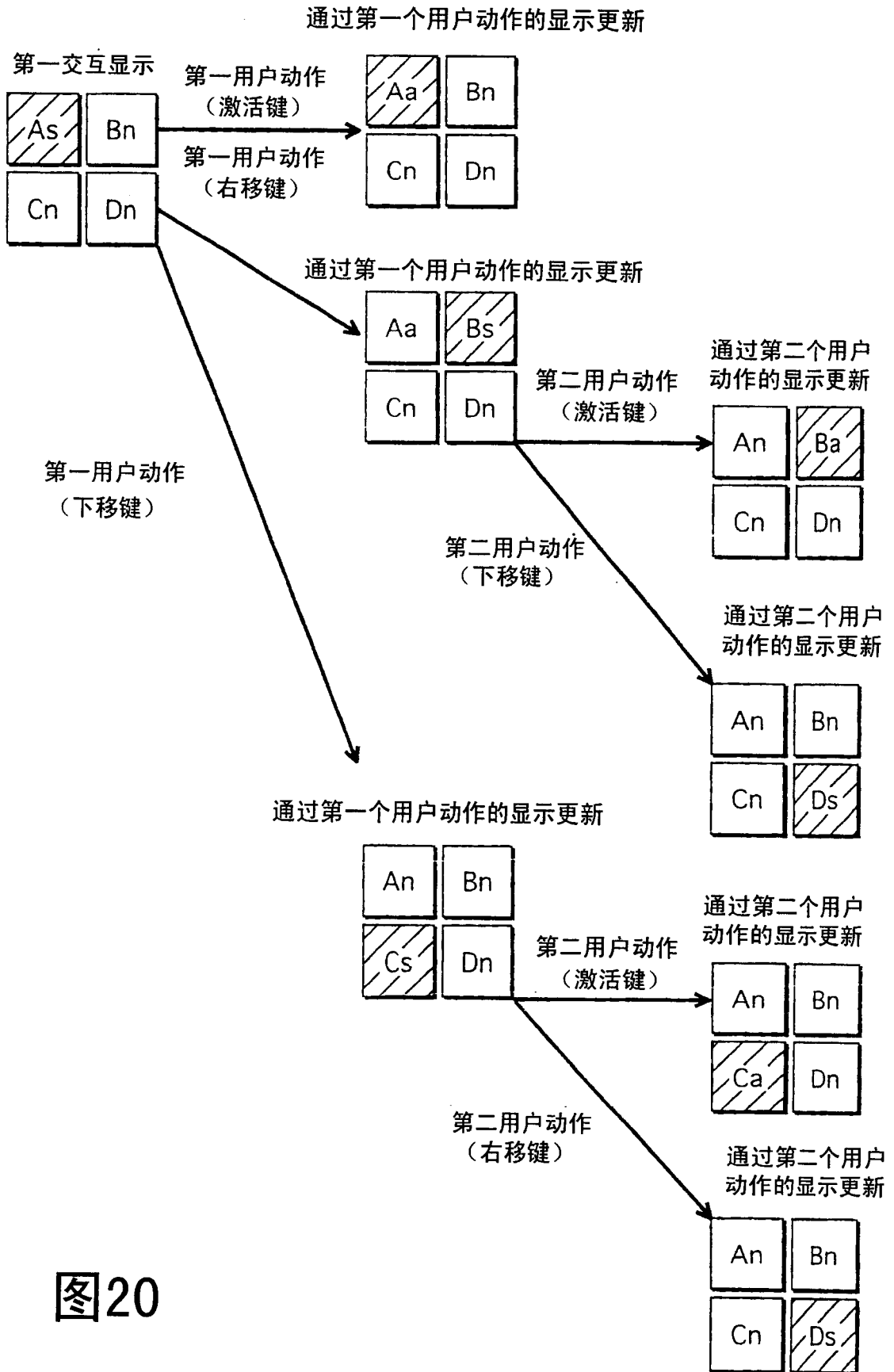


图20

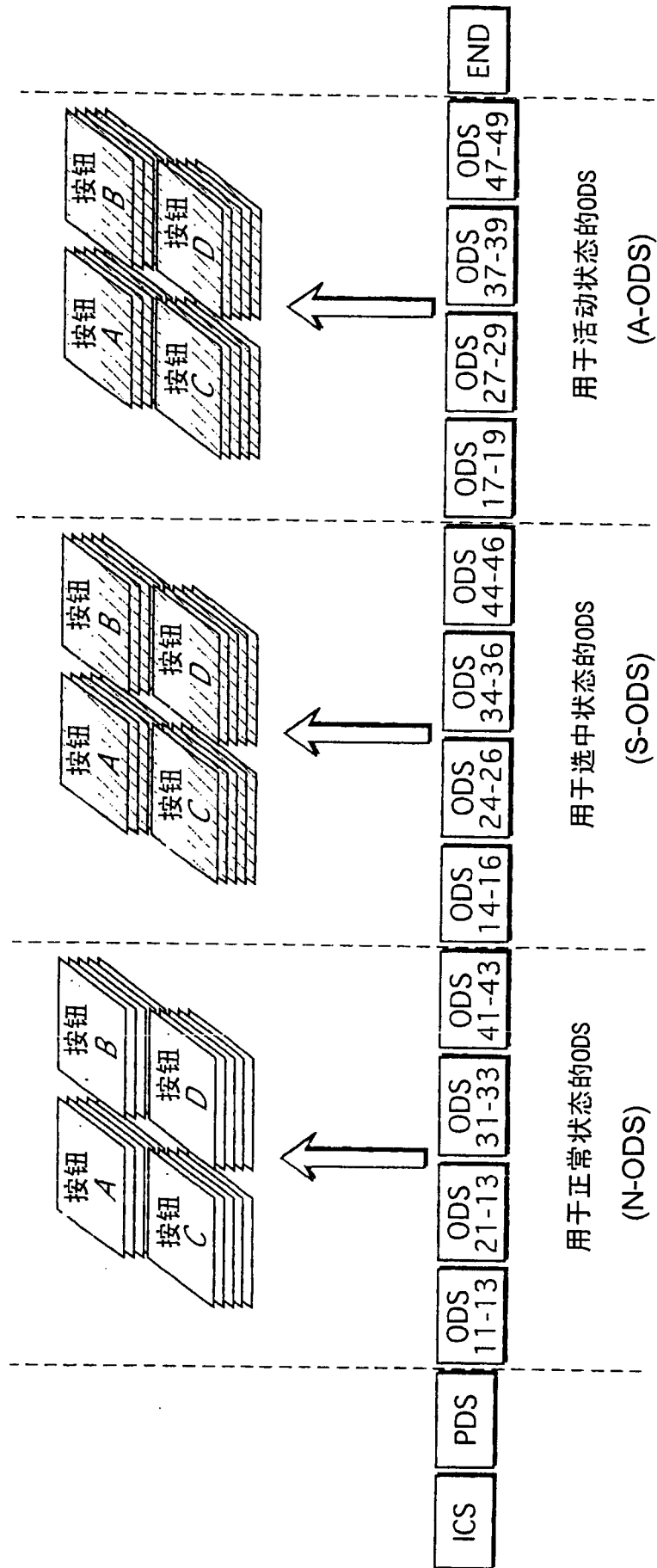


图21

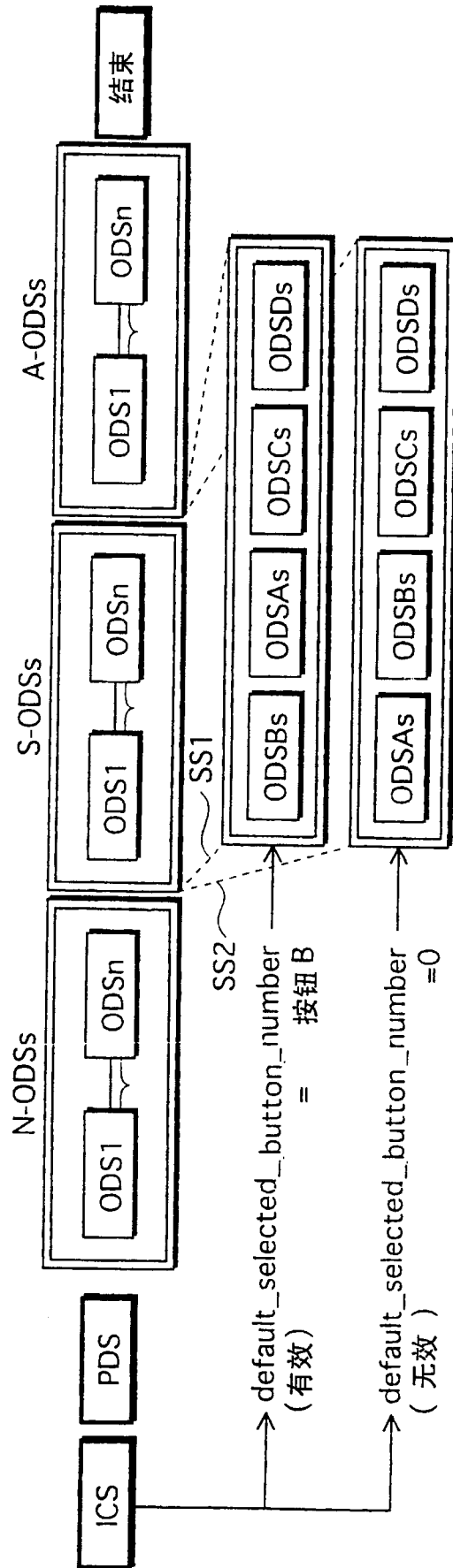


图22

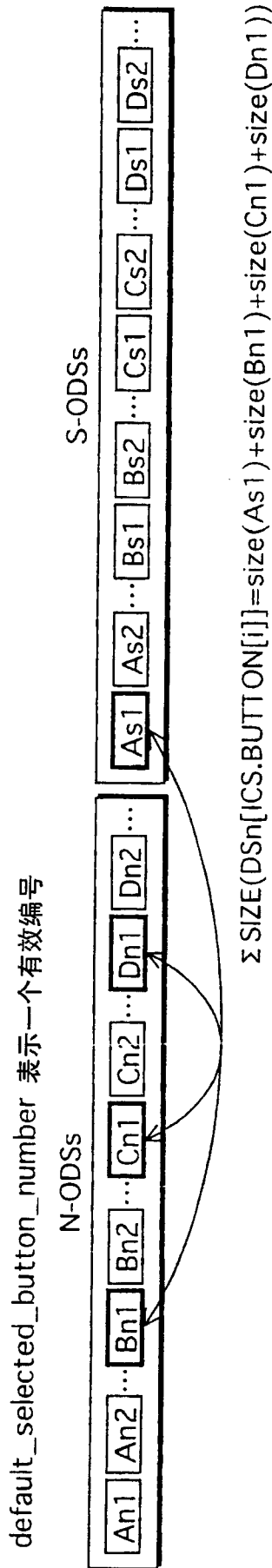


图 23A

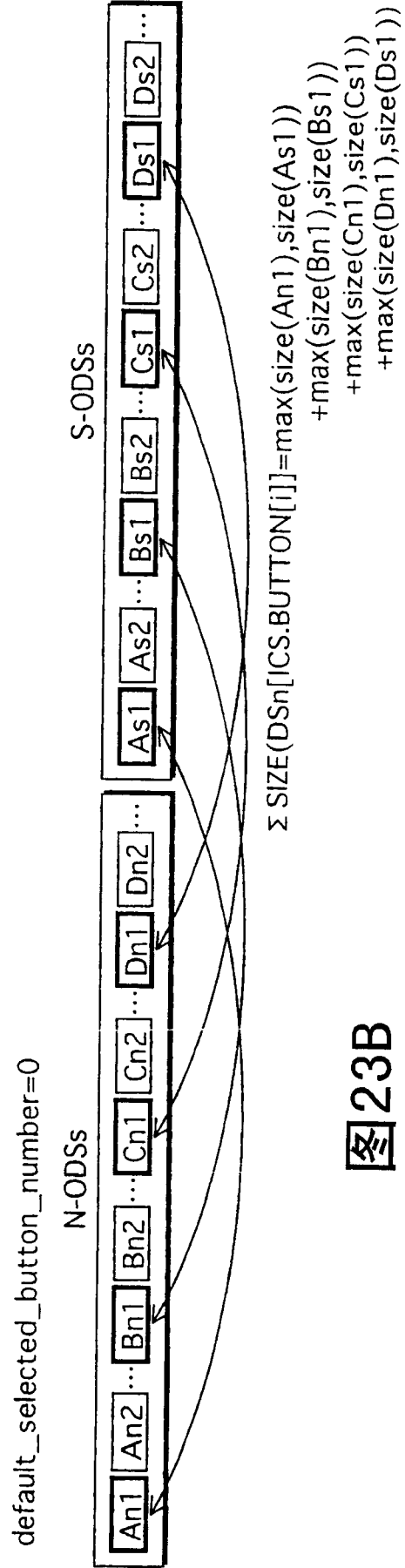


图 23B

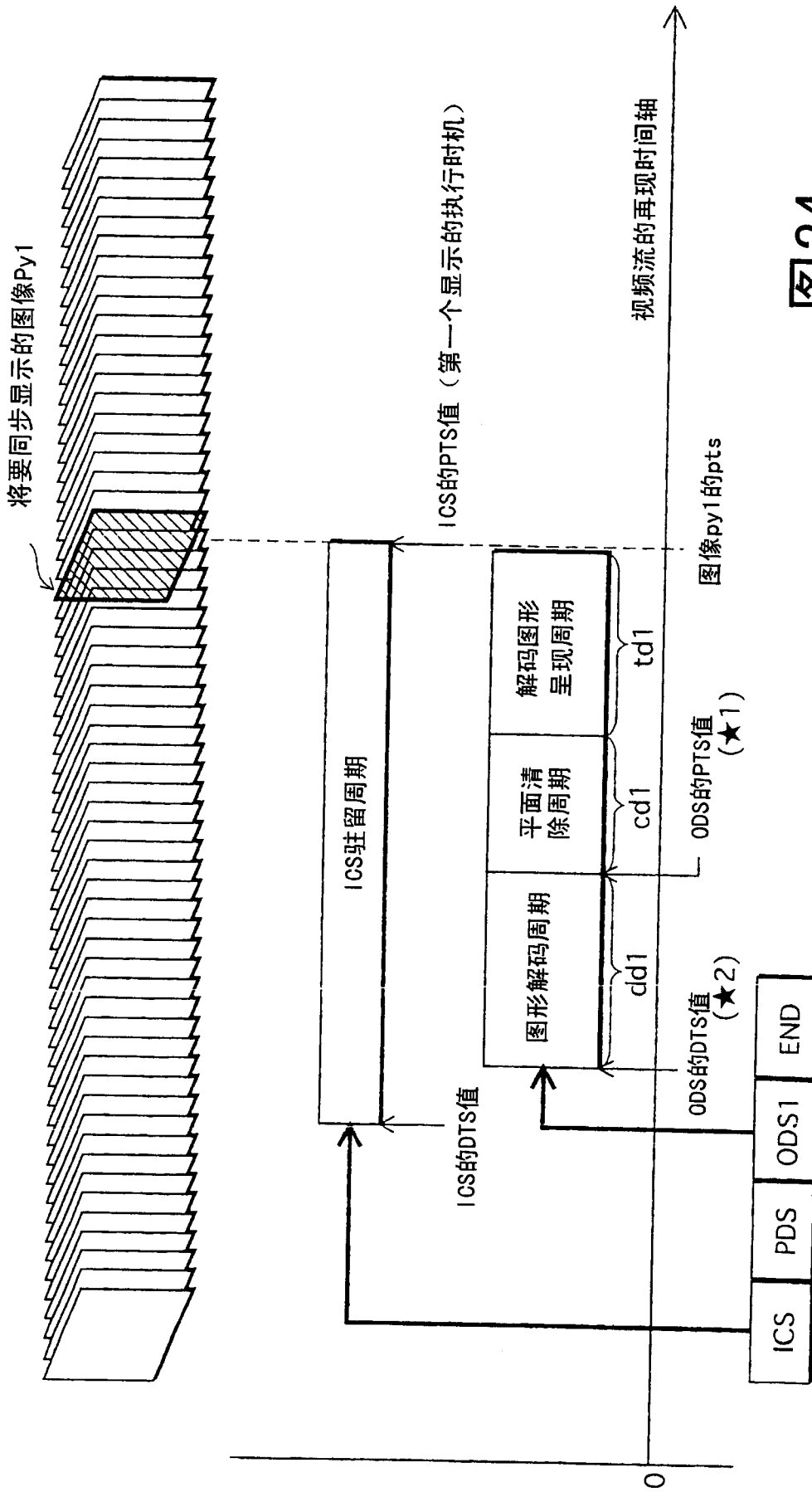


图24

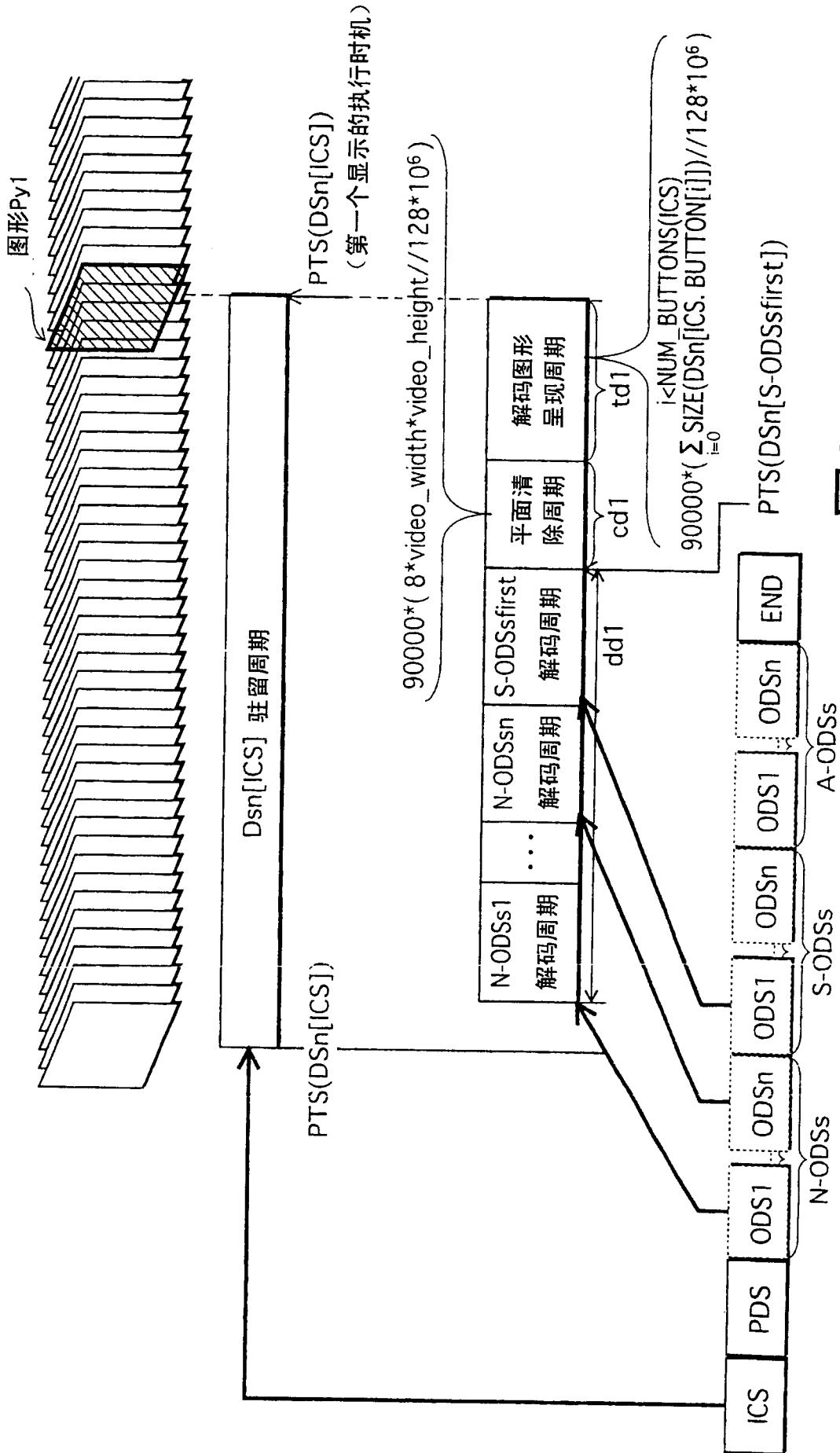


图25

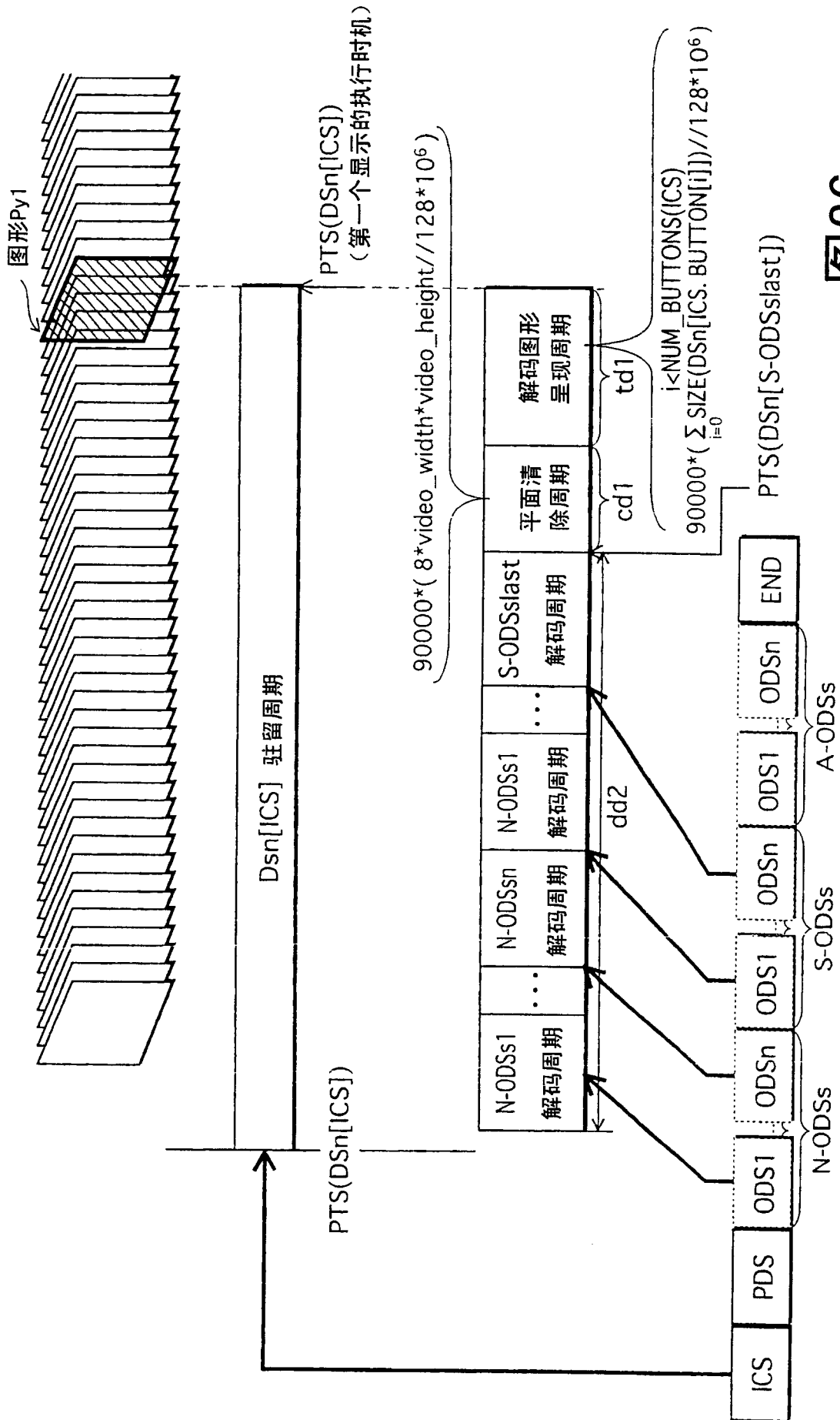


图26

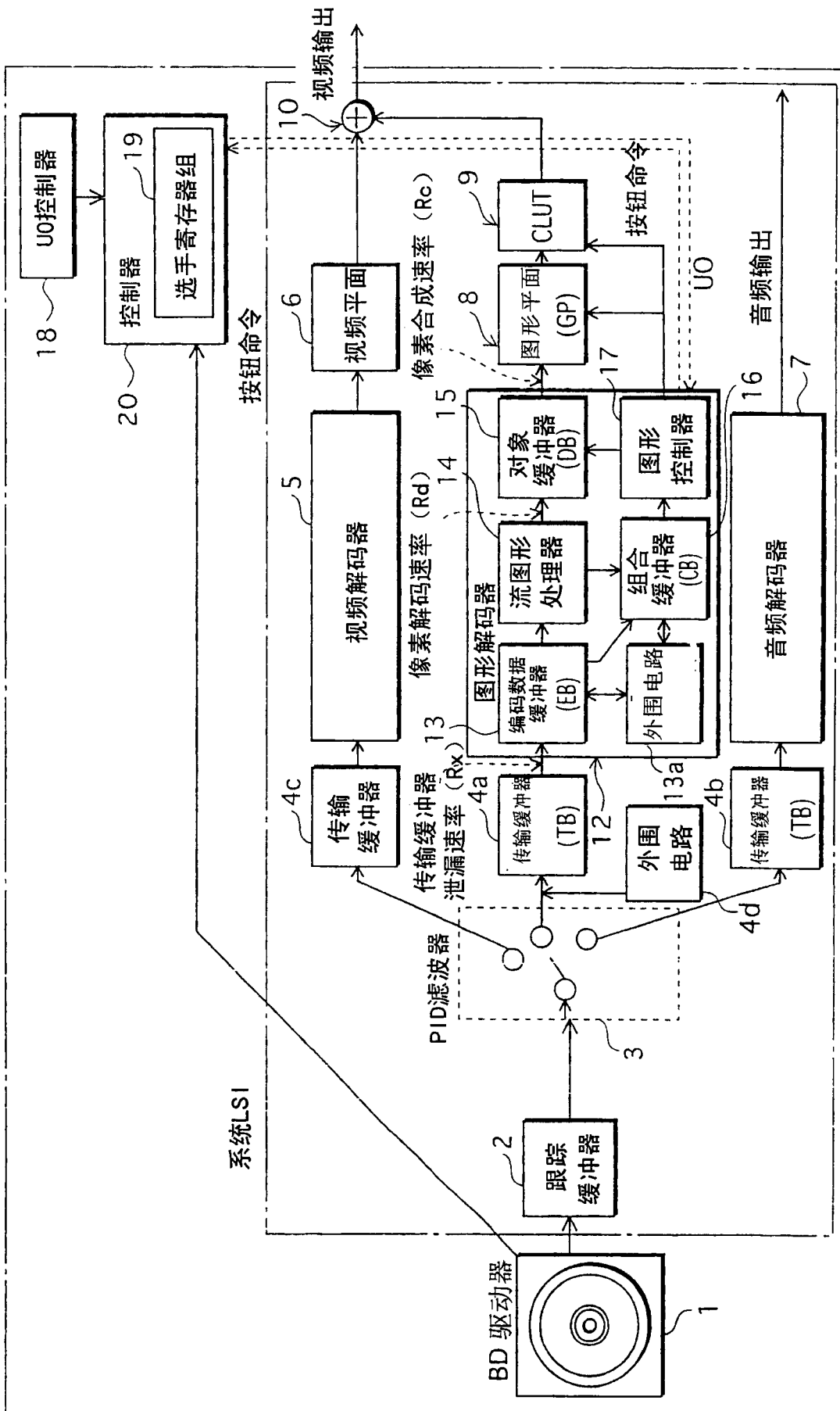
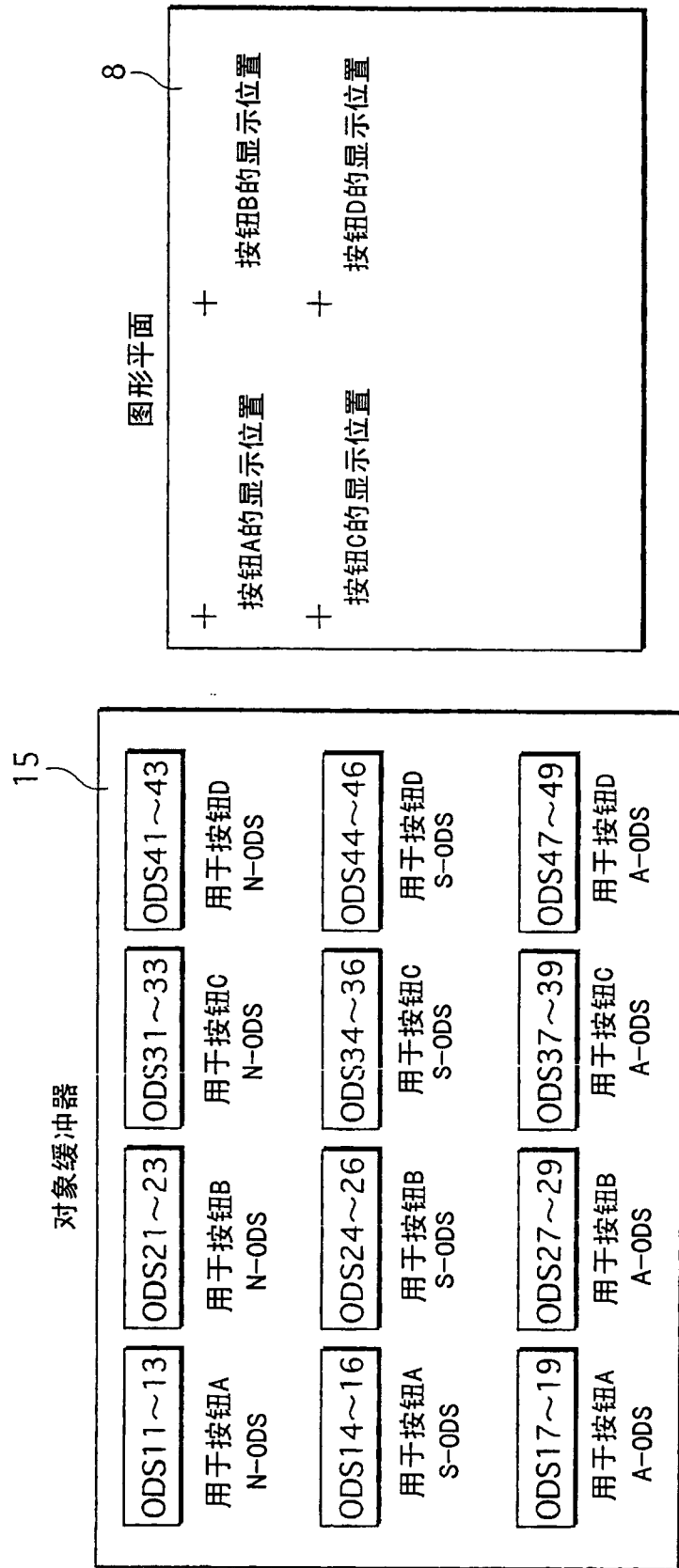


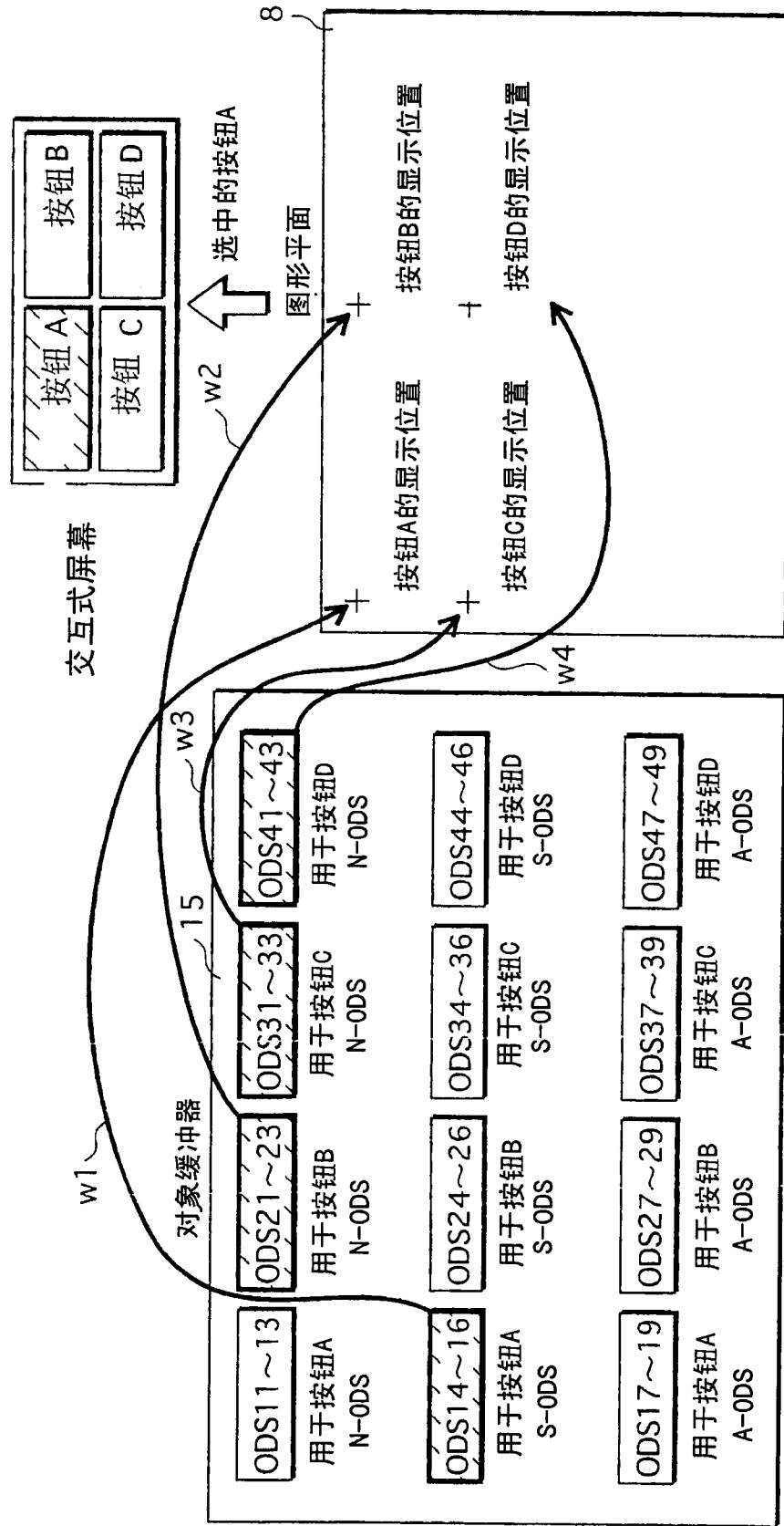
图27



按钮的显示位置：通过按钮信息button_horizontal_position、
button_vertical_position定义的显示位置

图28

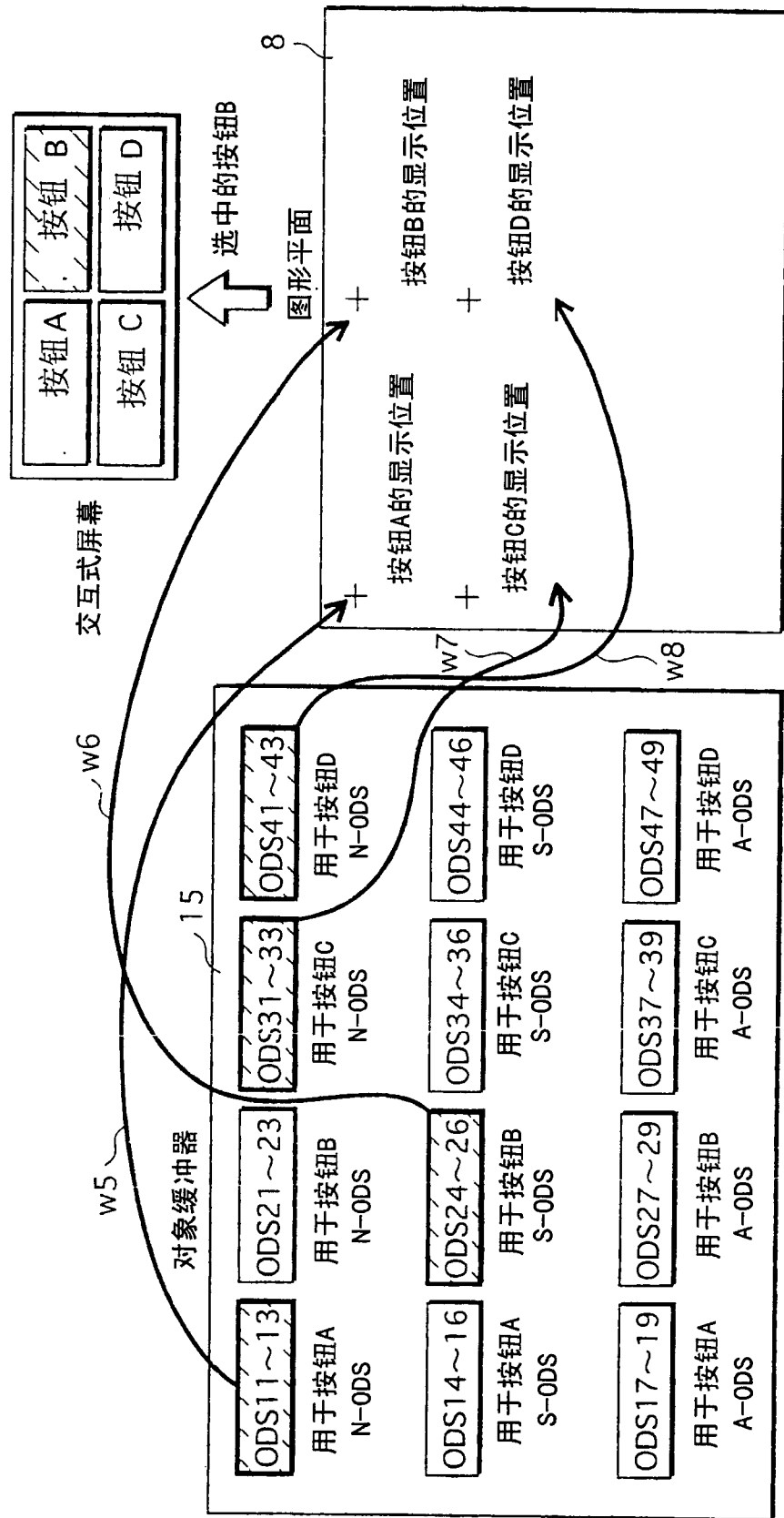
初始显示处的图形控制器的写操作



按钮的显示位置: 通过按钮信息button_horizontal_position, button_vertical_position定义的显示位置

图29

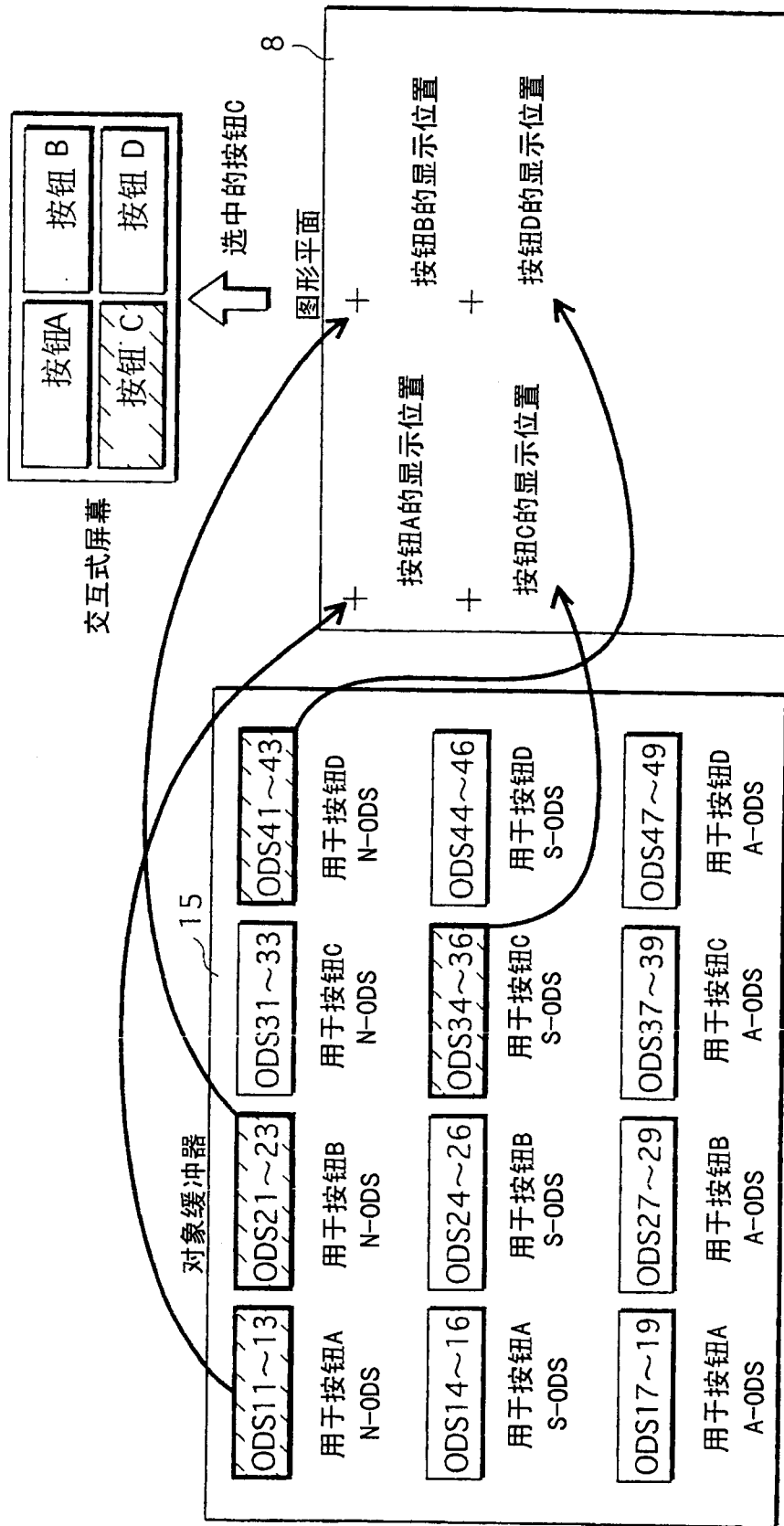
根据第一个用户动作（右移）的交互屏幕更新处的图形控制器的写操作



按钮的显示位置：通过按钮信息button_horizontal_position、button_vertical_position定义的显示位置

图30

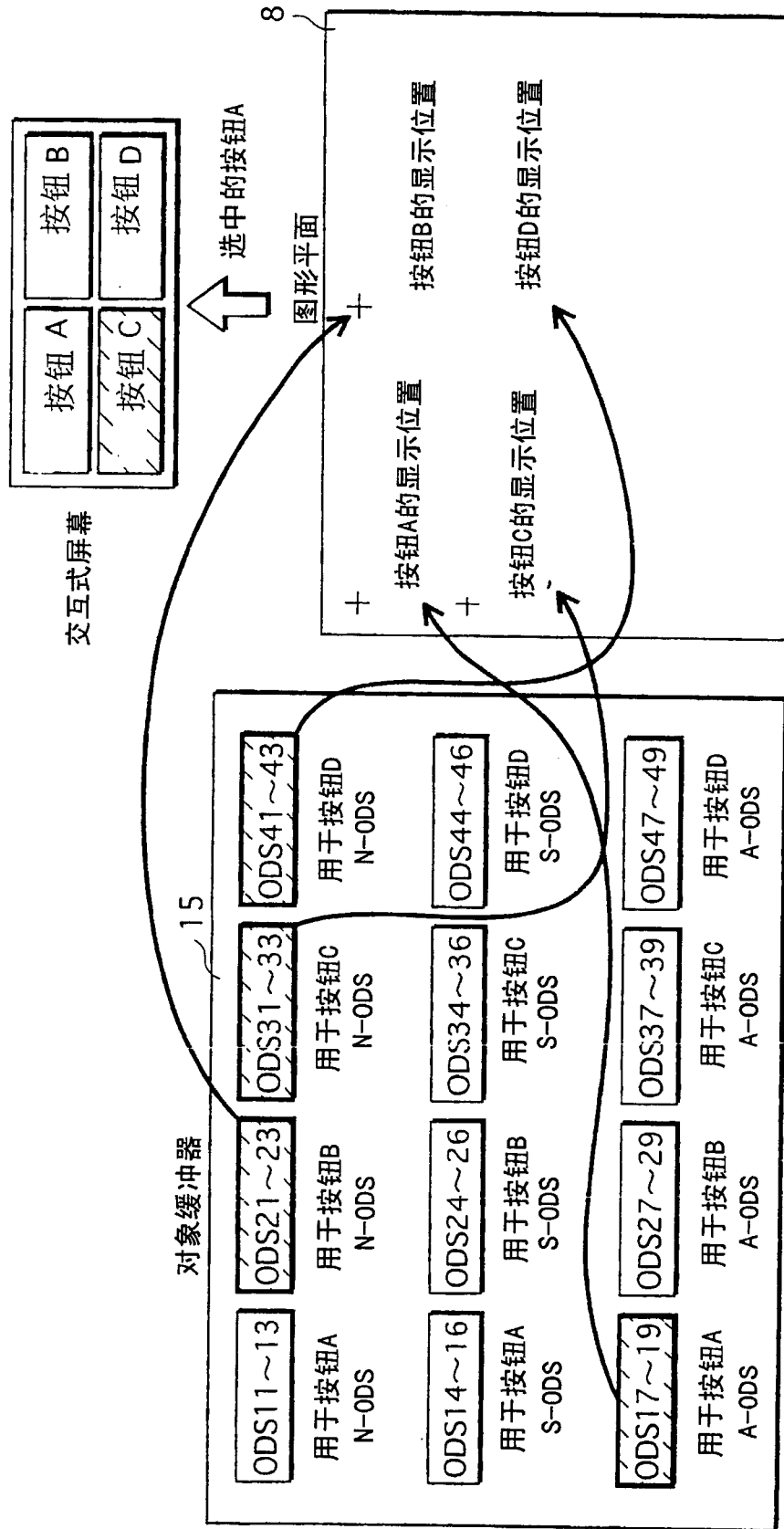
根据第一个用户动作（下移）的交互式屏幕更新处的图形控制器的写操作



按钮的显示位置: 通过按钮信息button_horizontal_position、
button_vertical_position定义的显示位置

图 31

根据第一个用户动作（激活）的交互屏幕更新处的图形控制器的写操作



按钮的显示位置：通过按钮信息button_horizontal_position、button_vertical_position定义的显示位置

图32

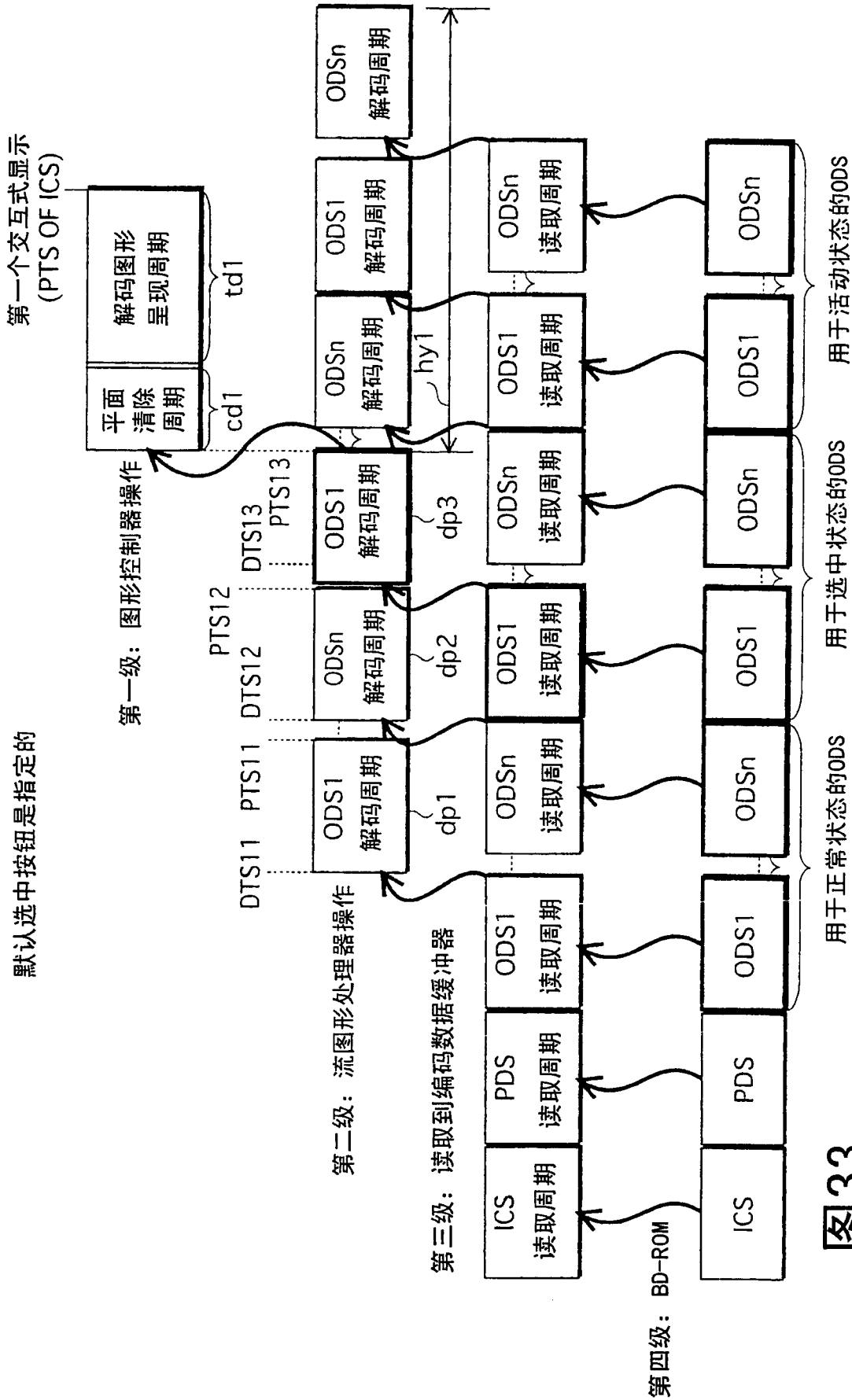


图 33

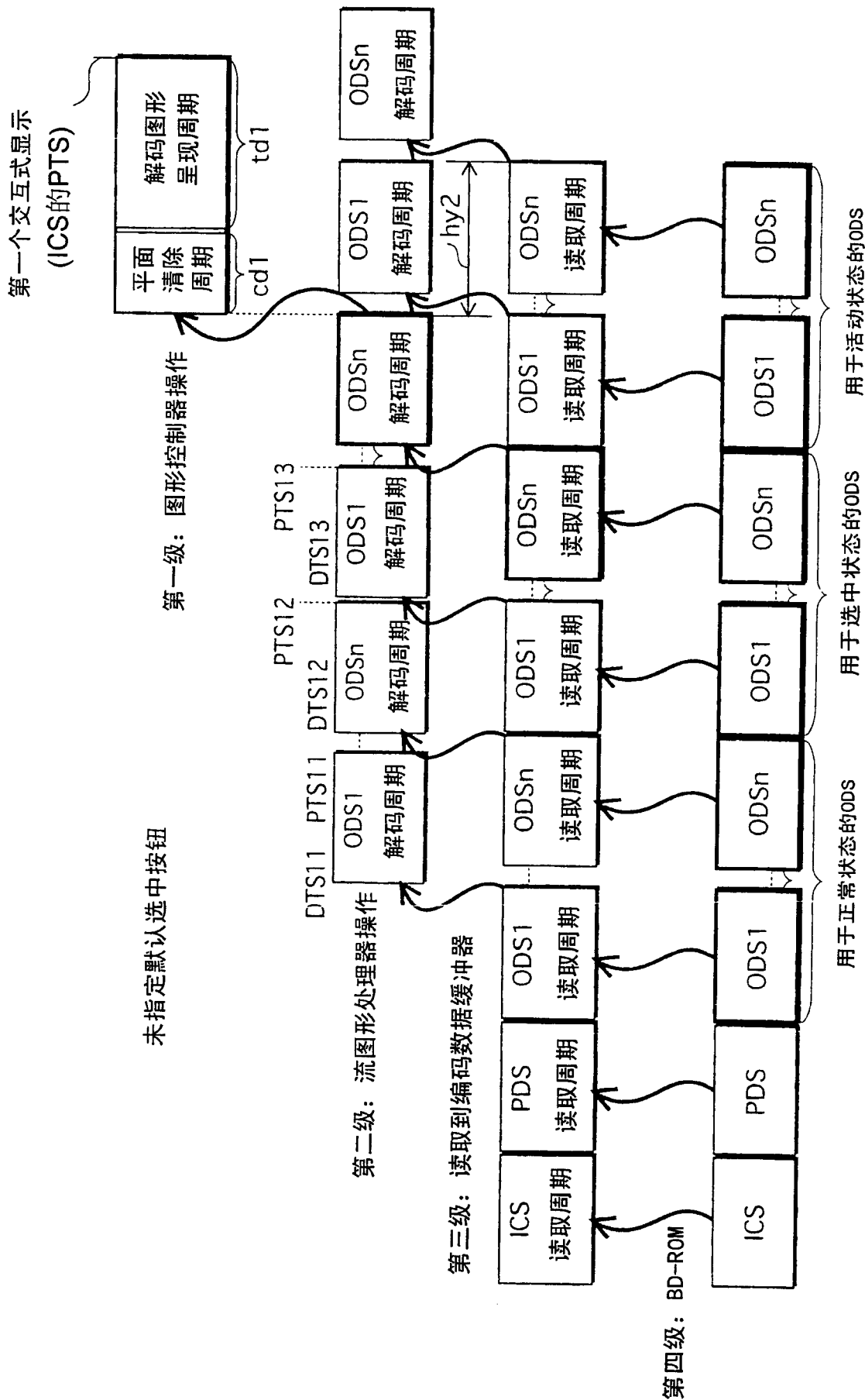


图 34

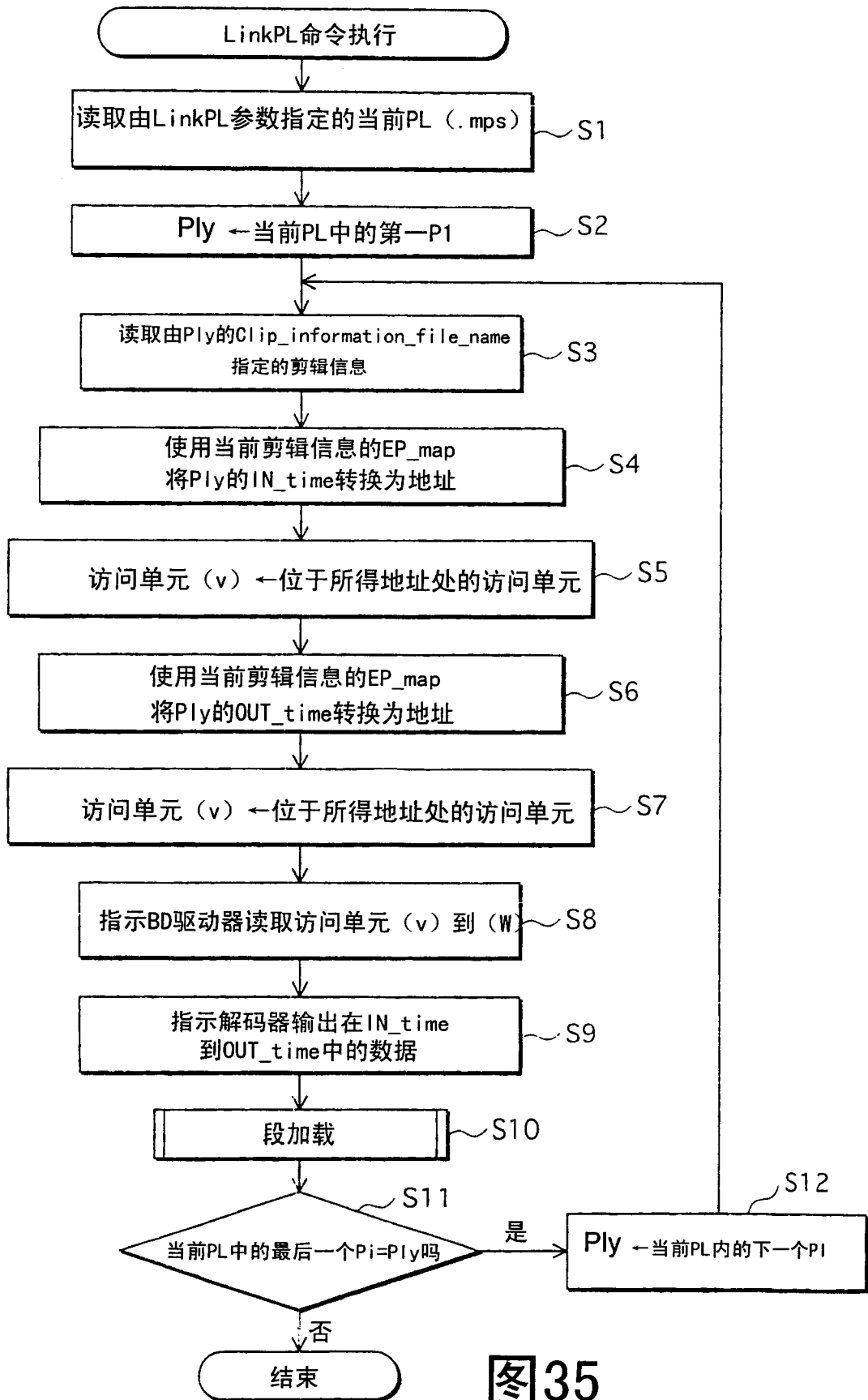


图35

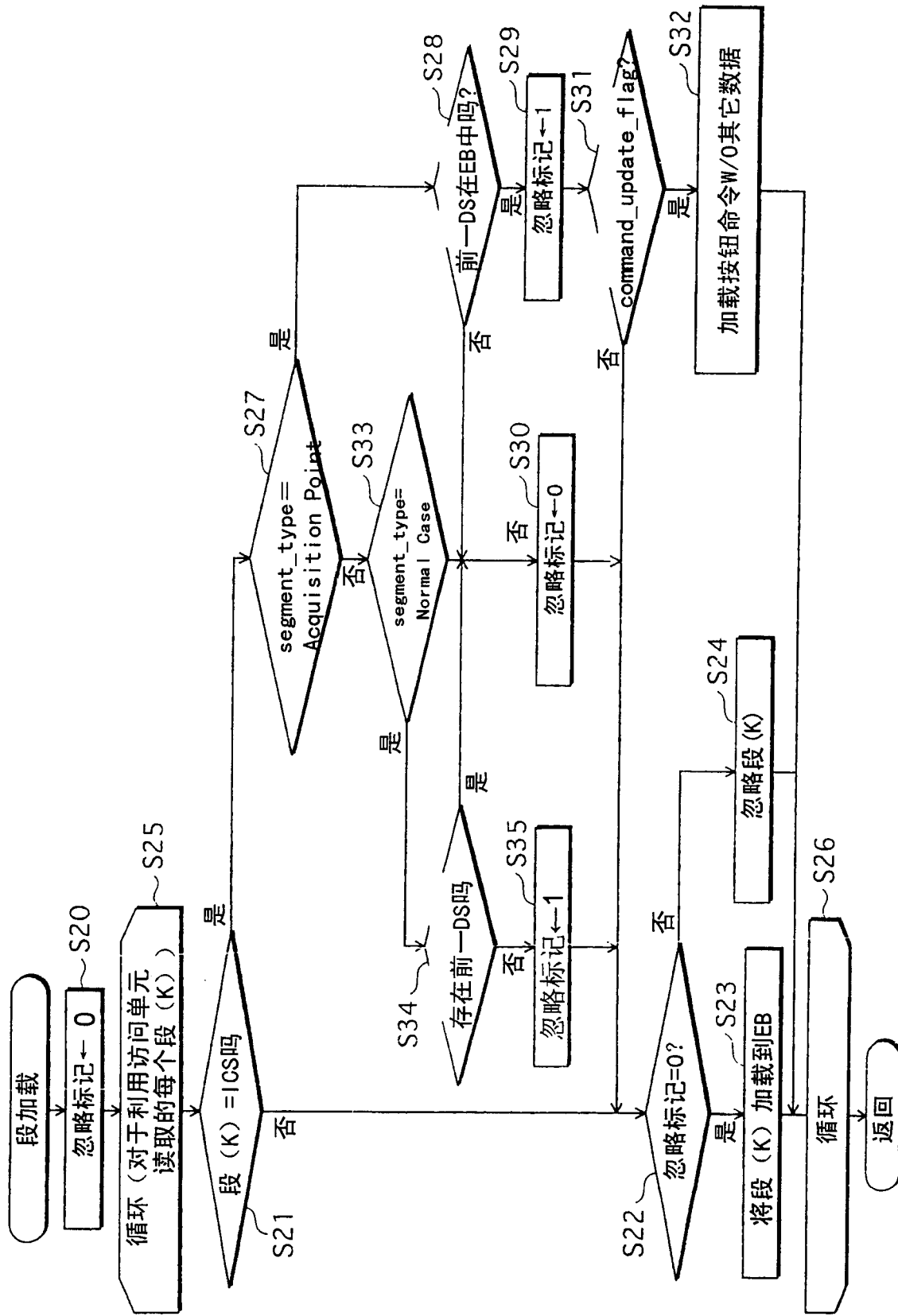


图 36

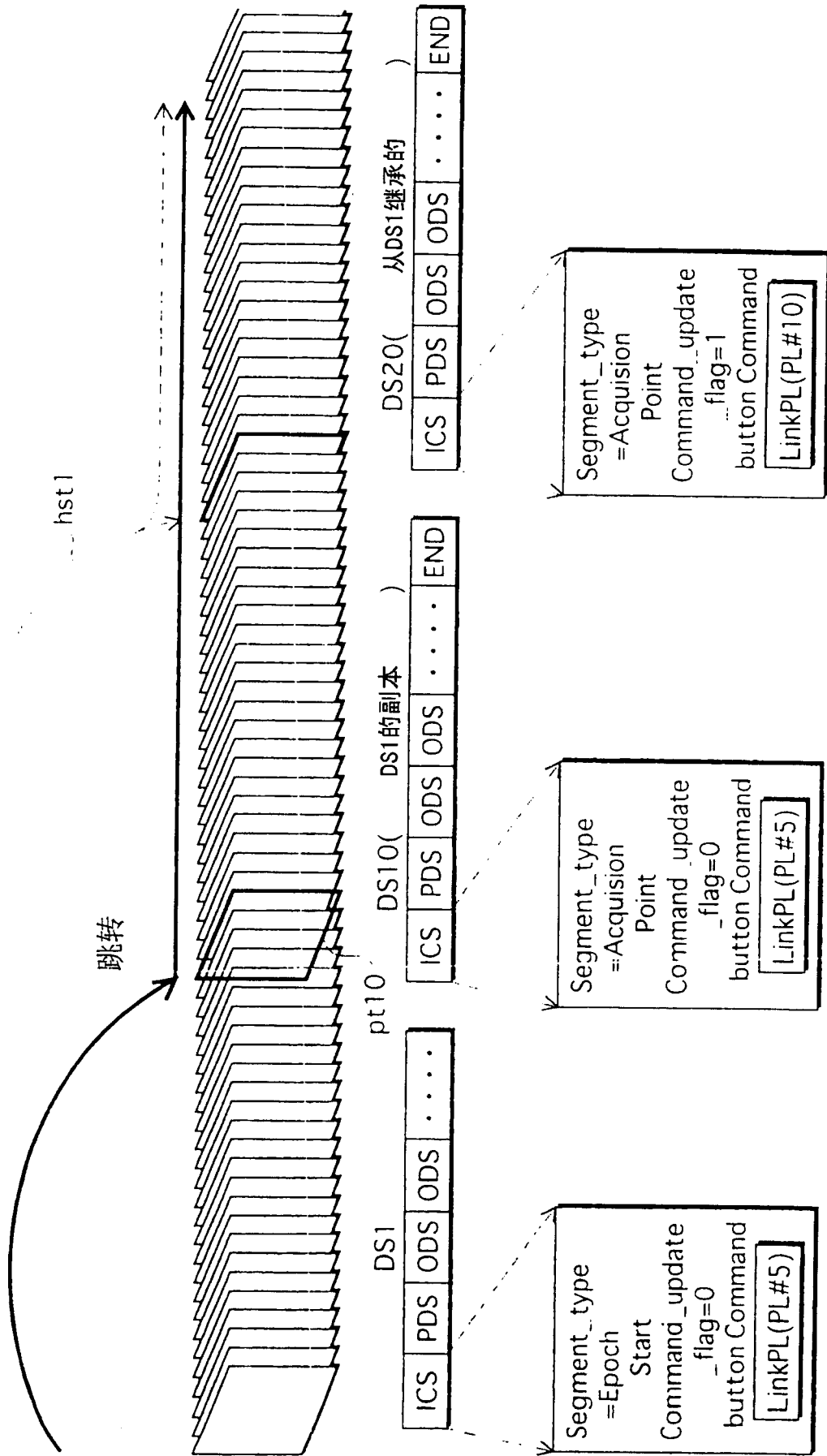


图 37

再现装置的编码数据缓冲器

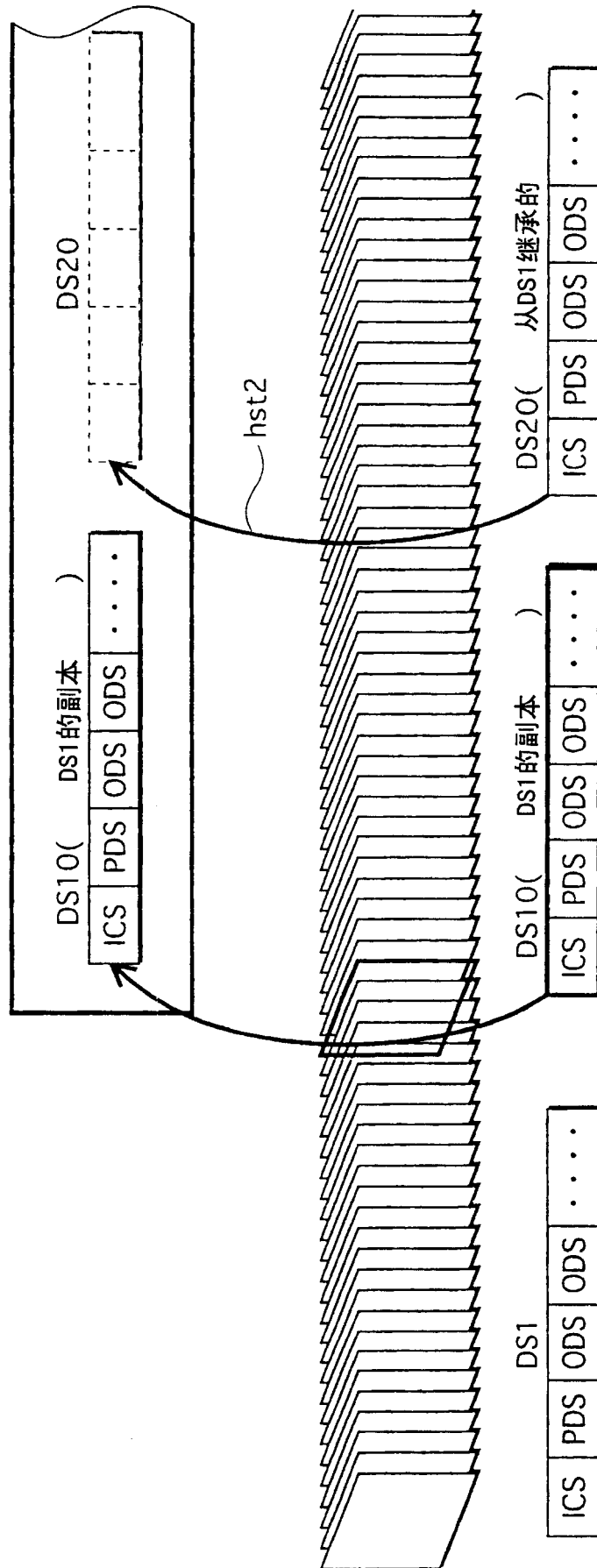


图38

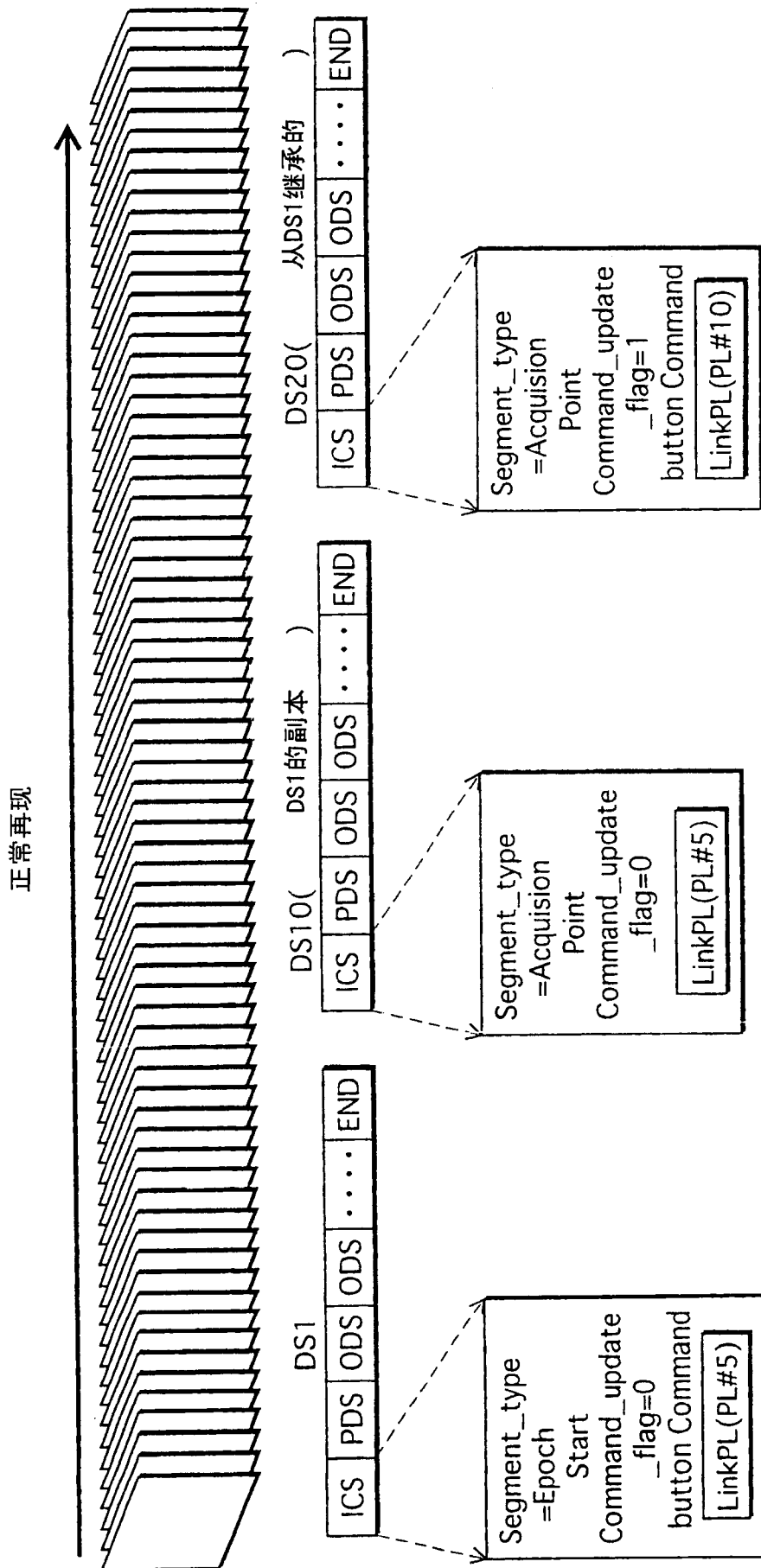


图 39

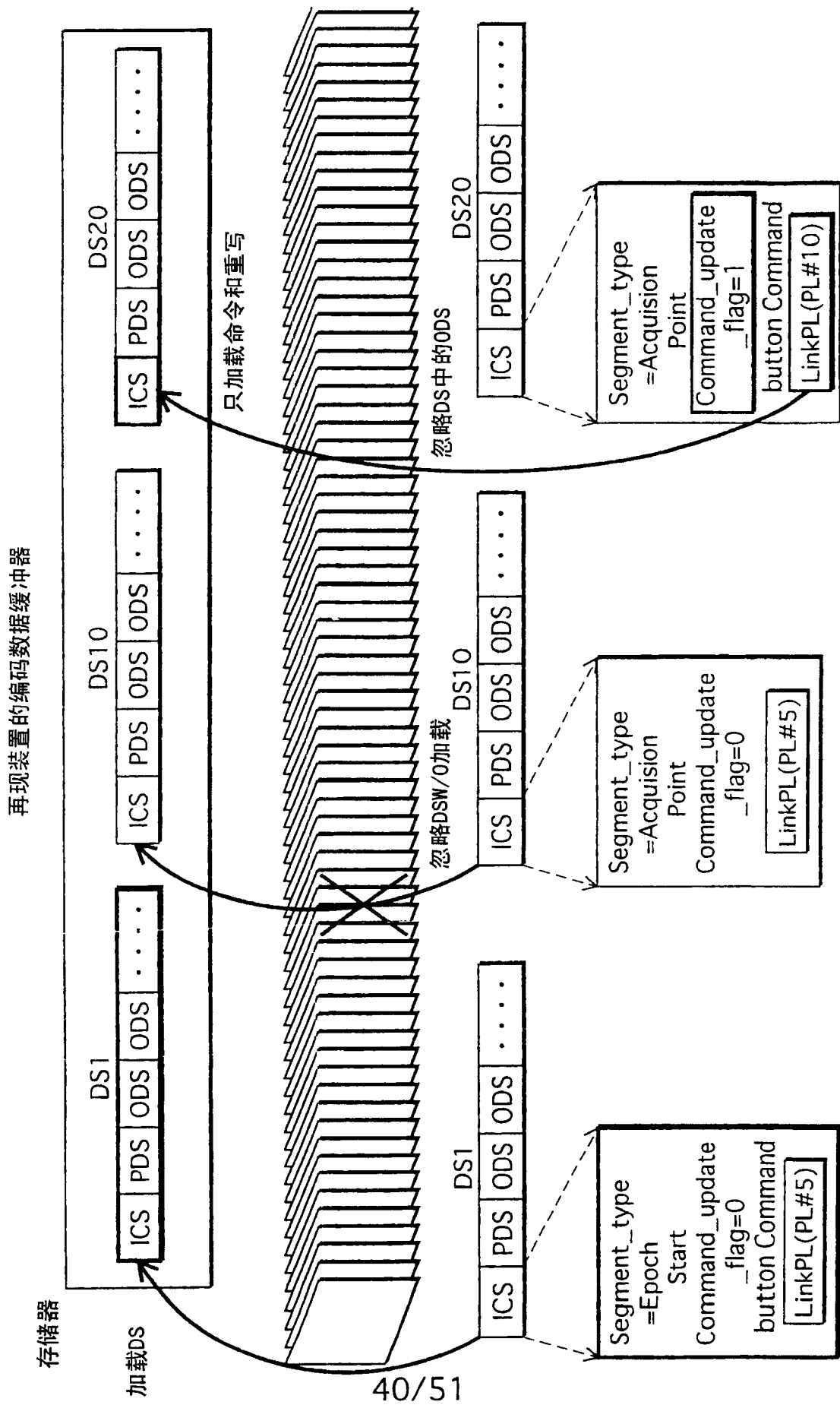


图40

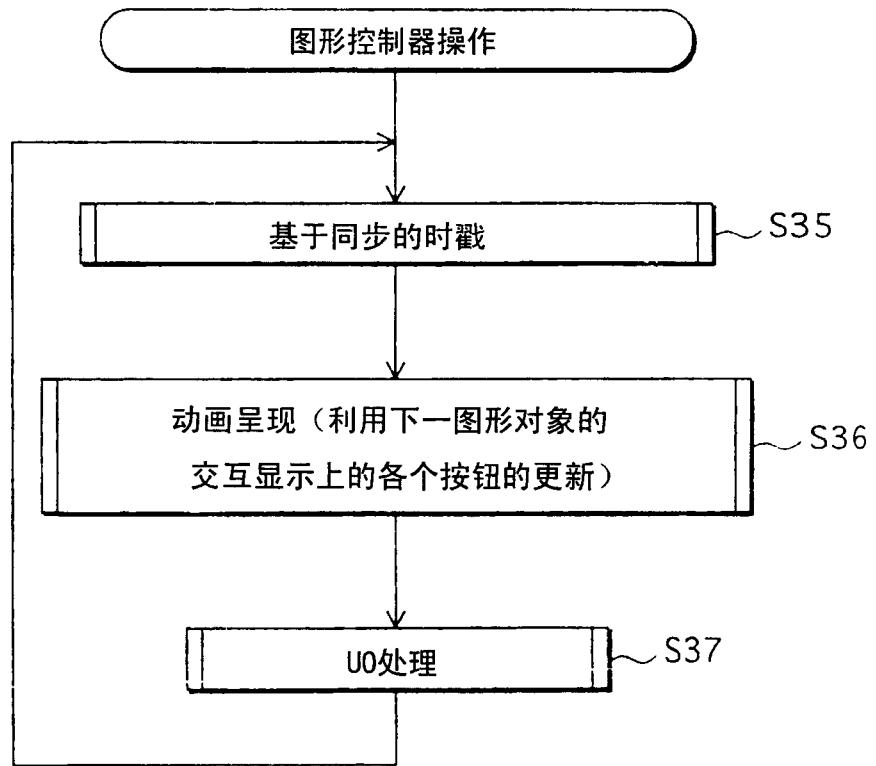


图41

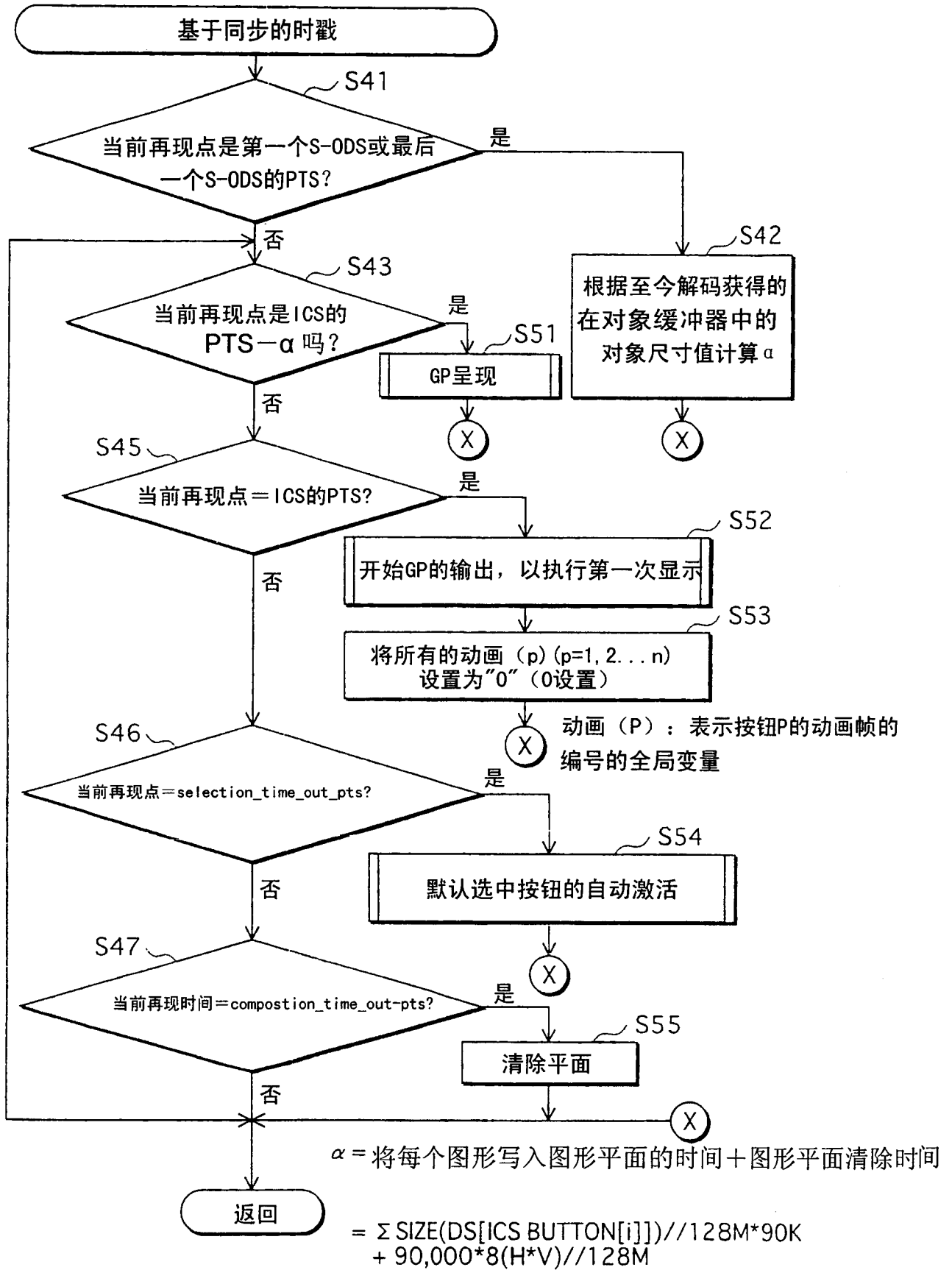


图42

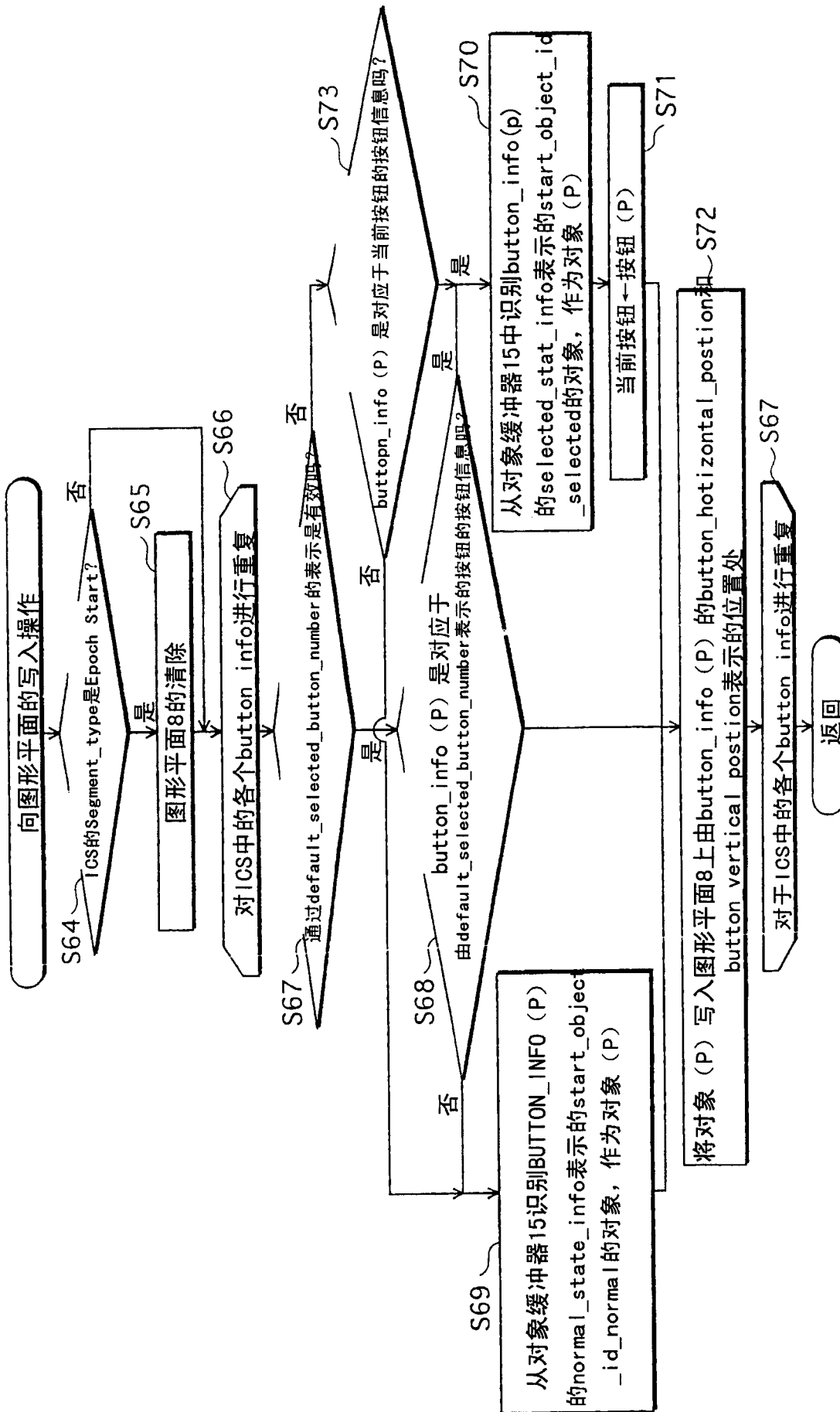


图43

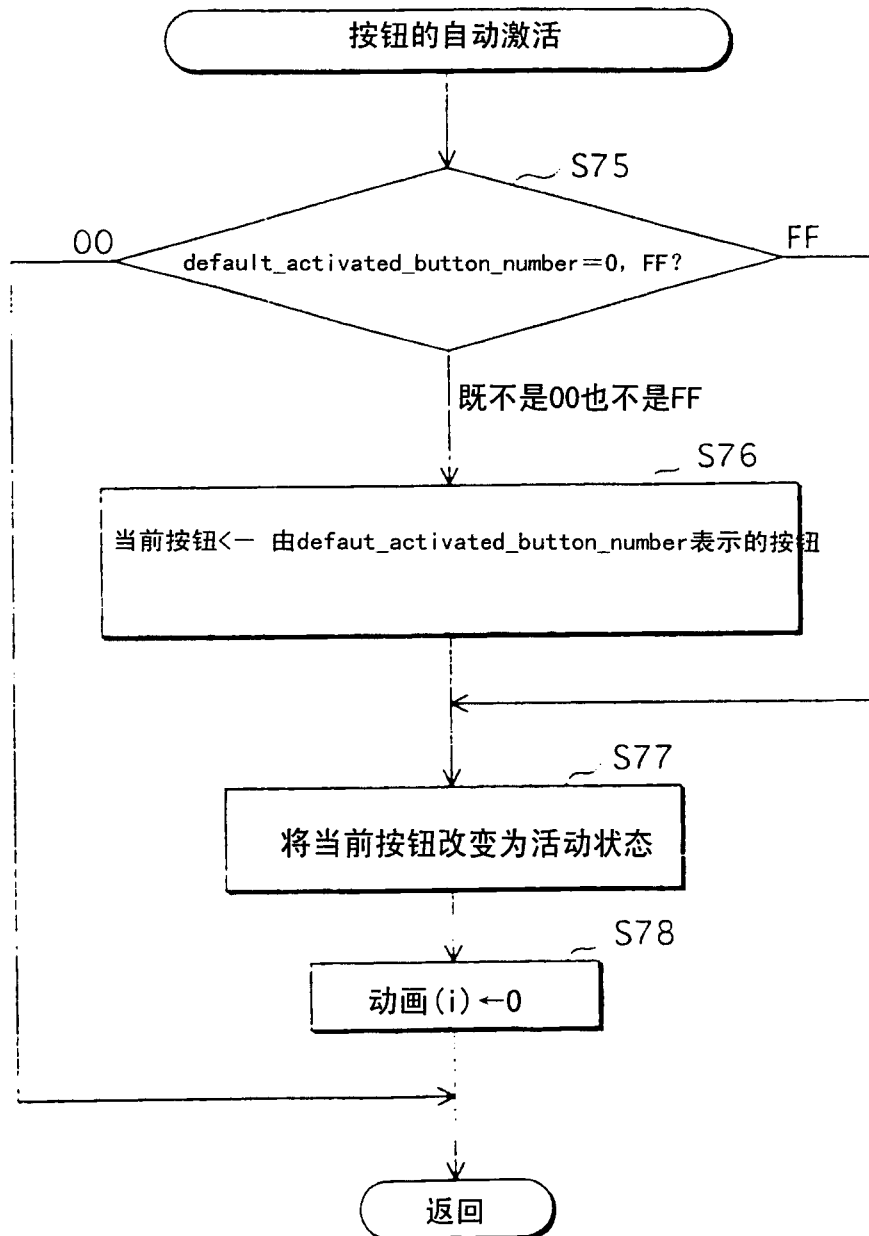


图44

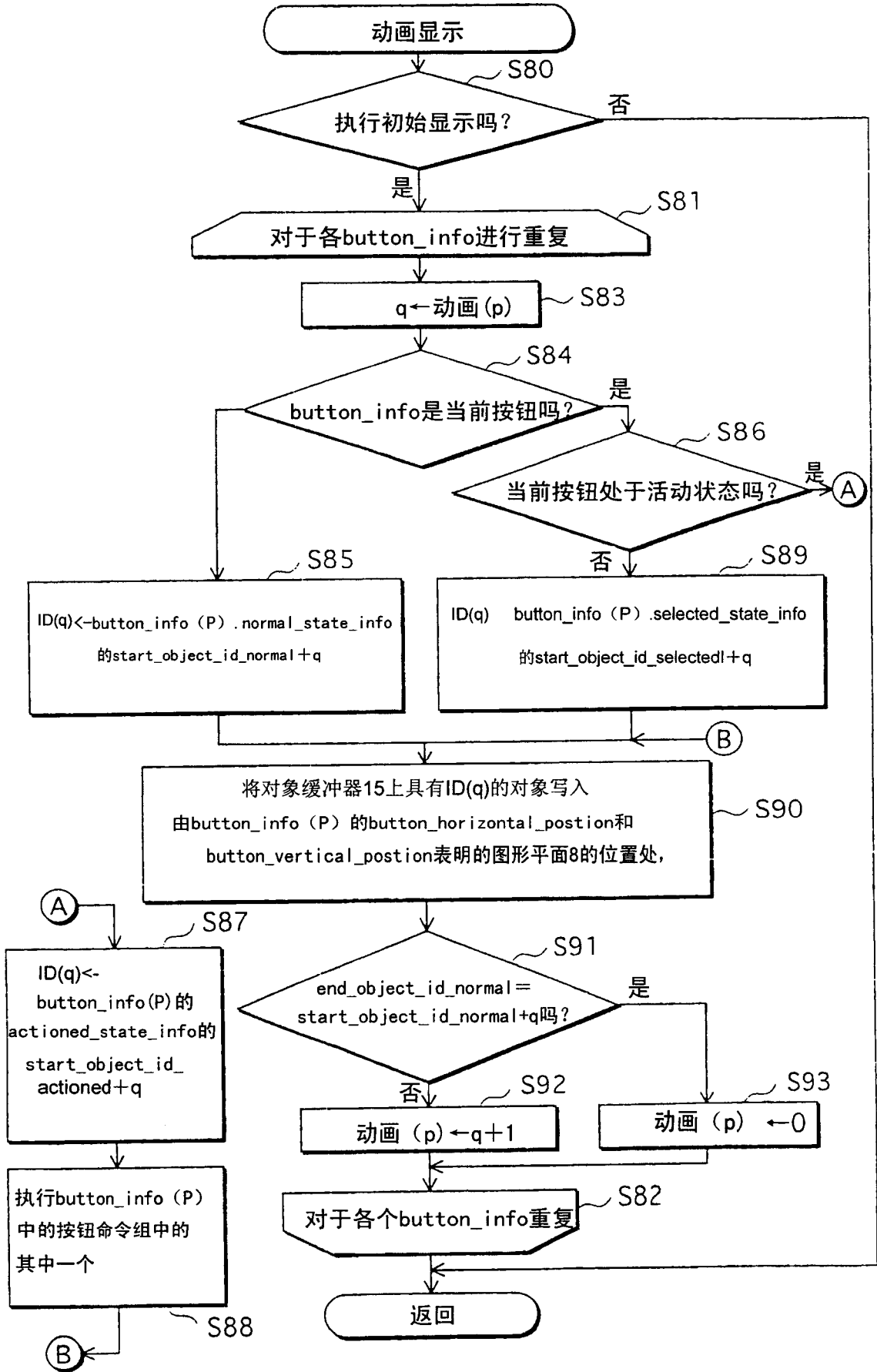


图45

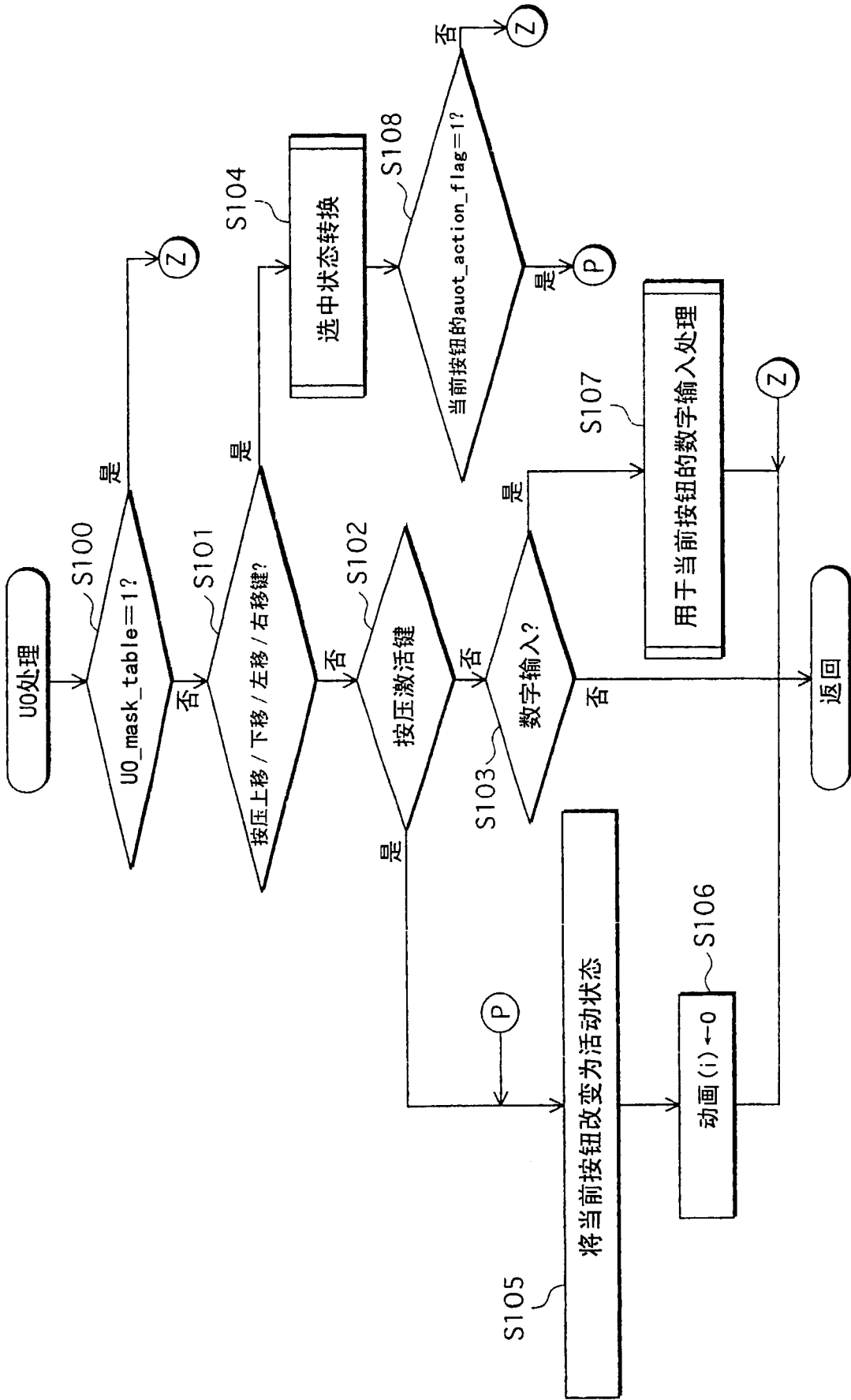


图46

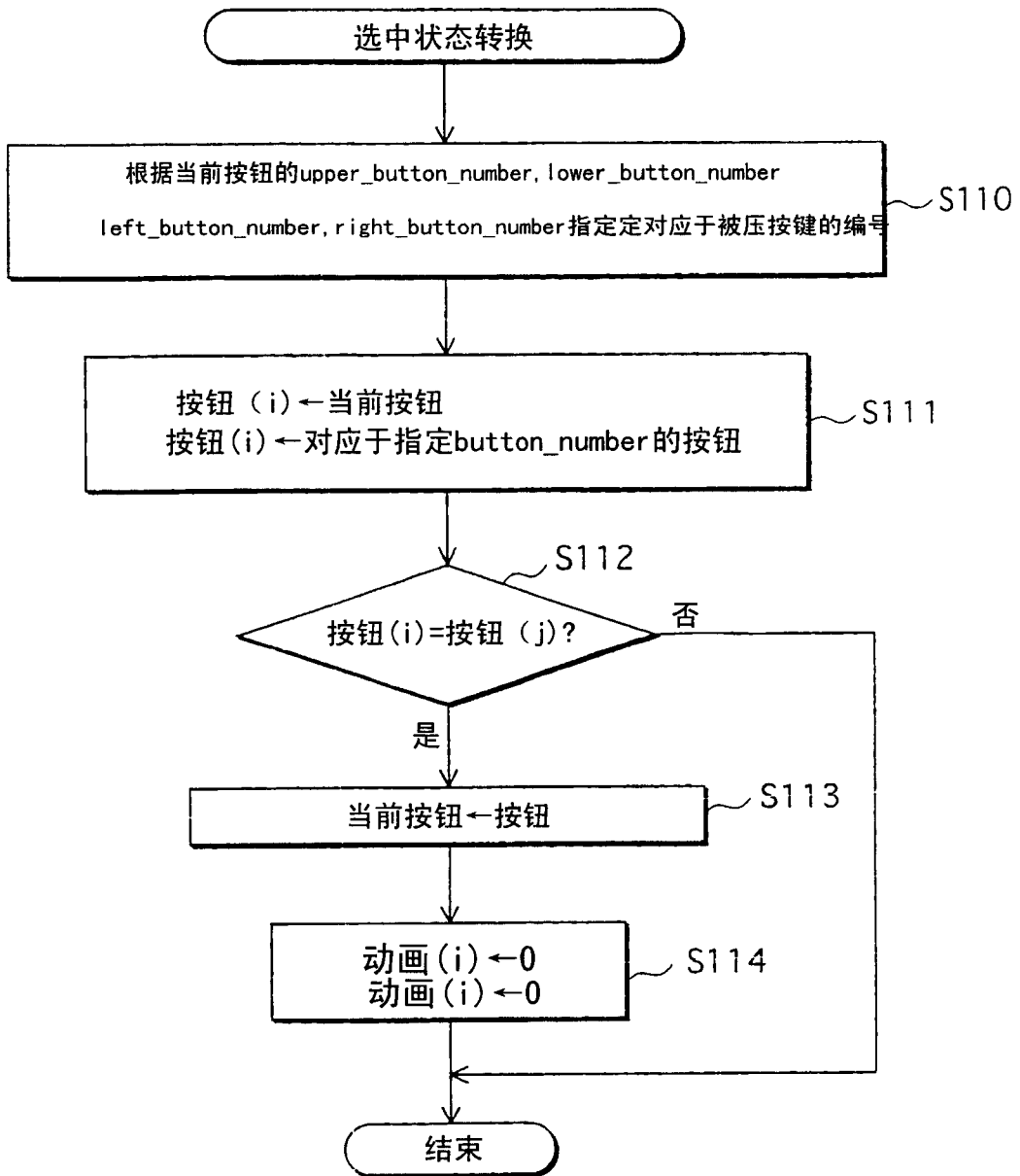


图47

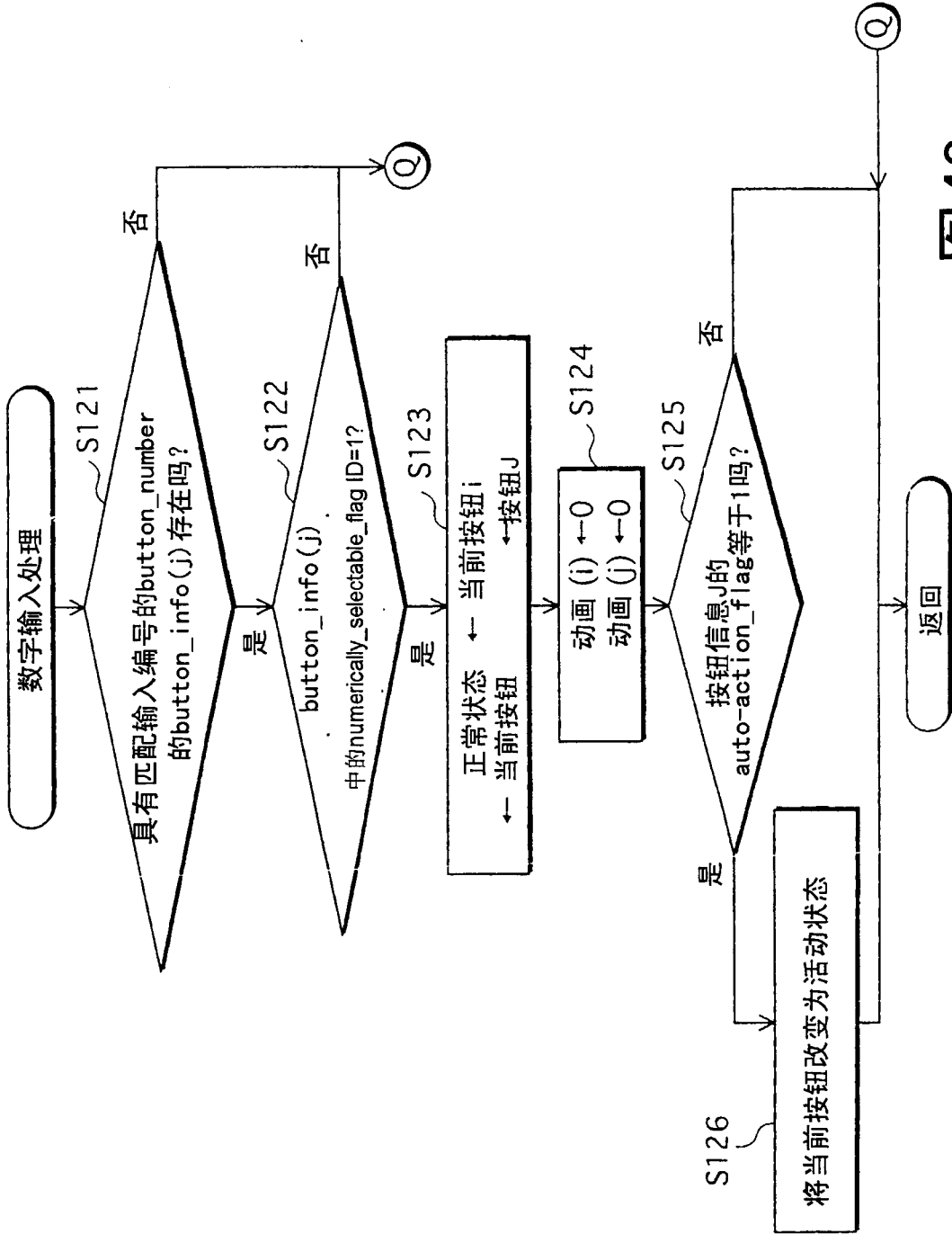


图48

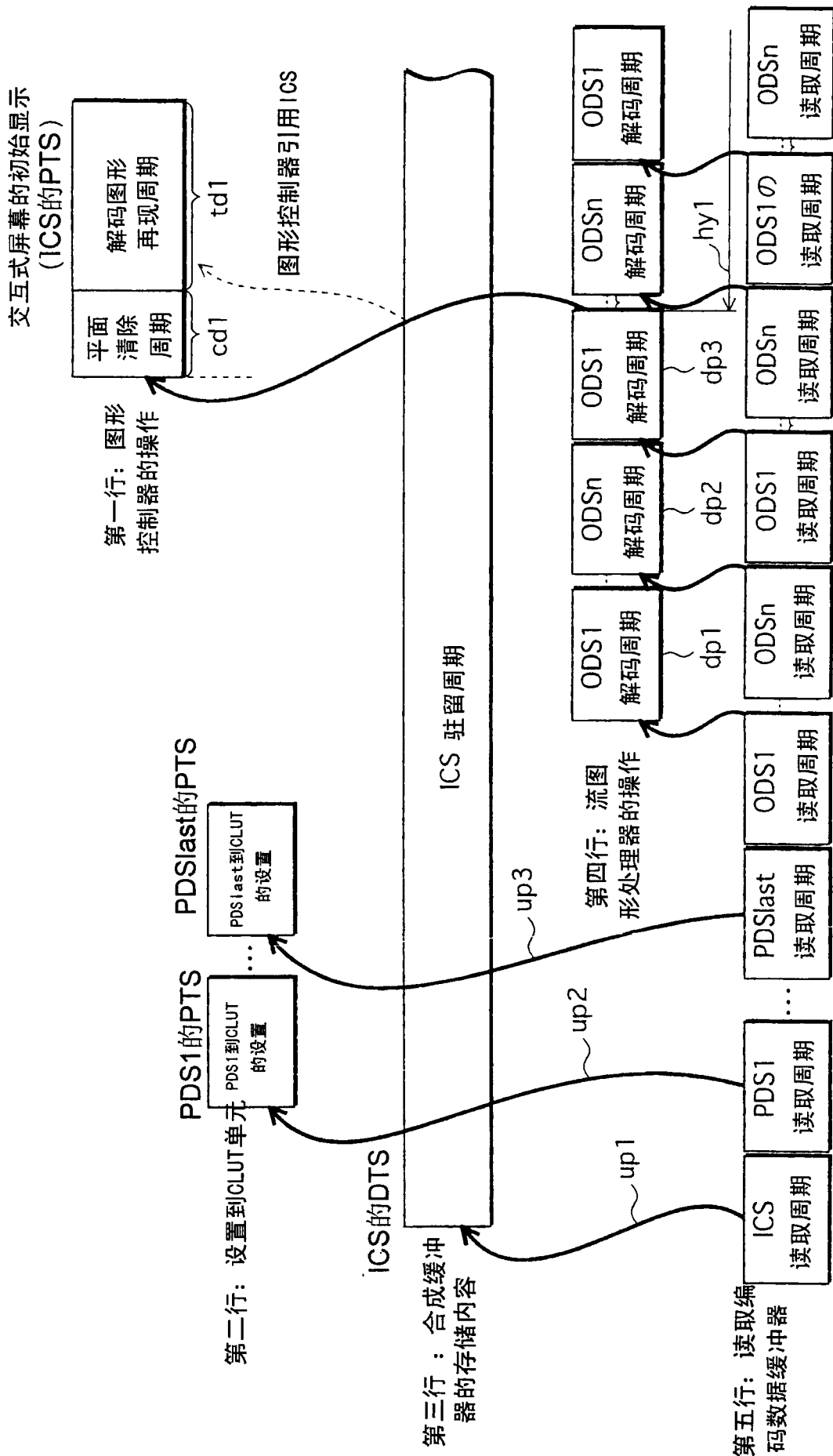


图 49

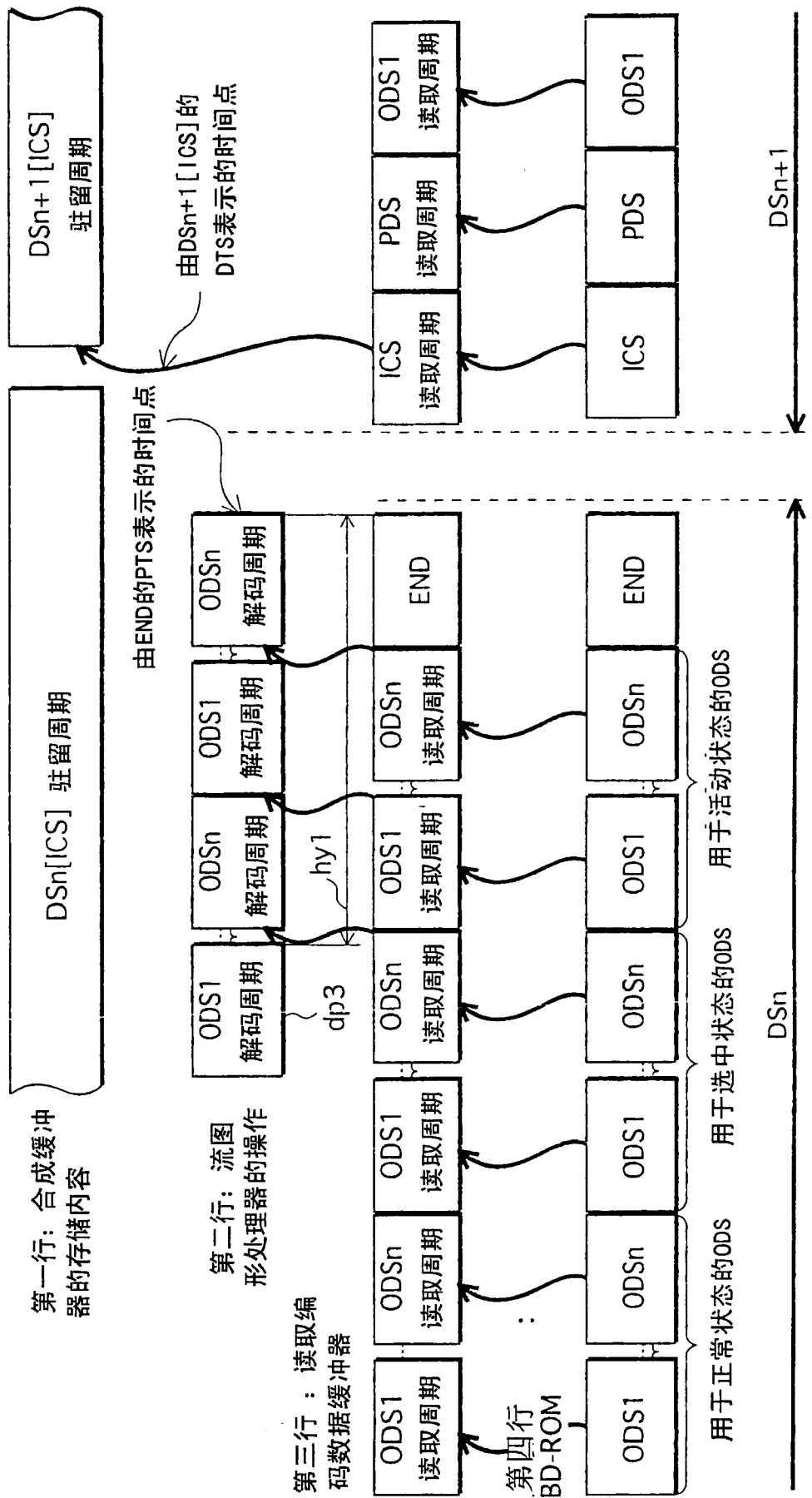


图50

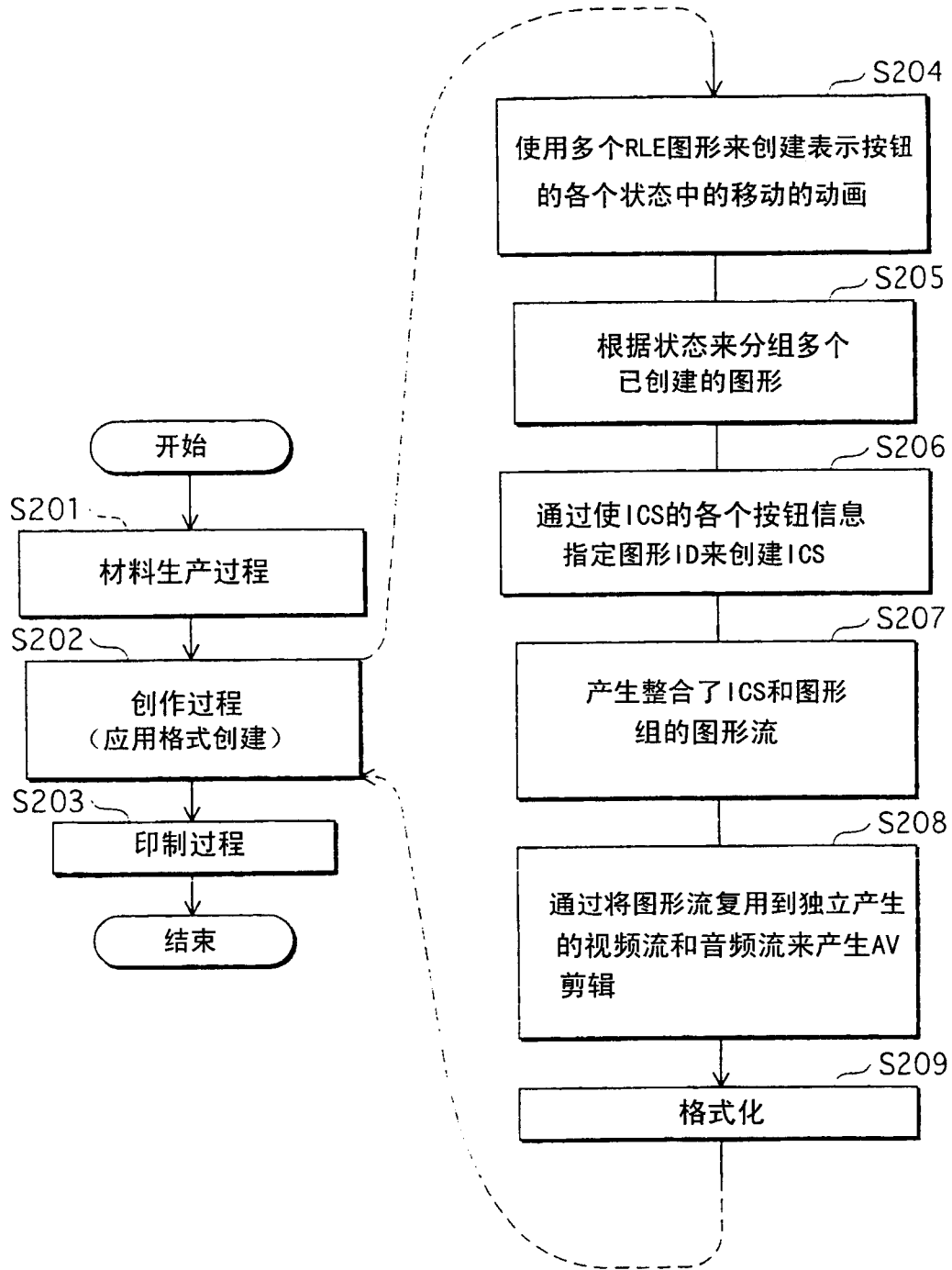


图51

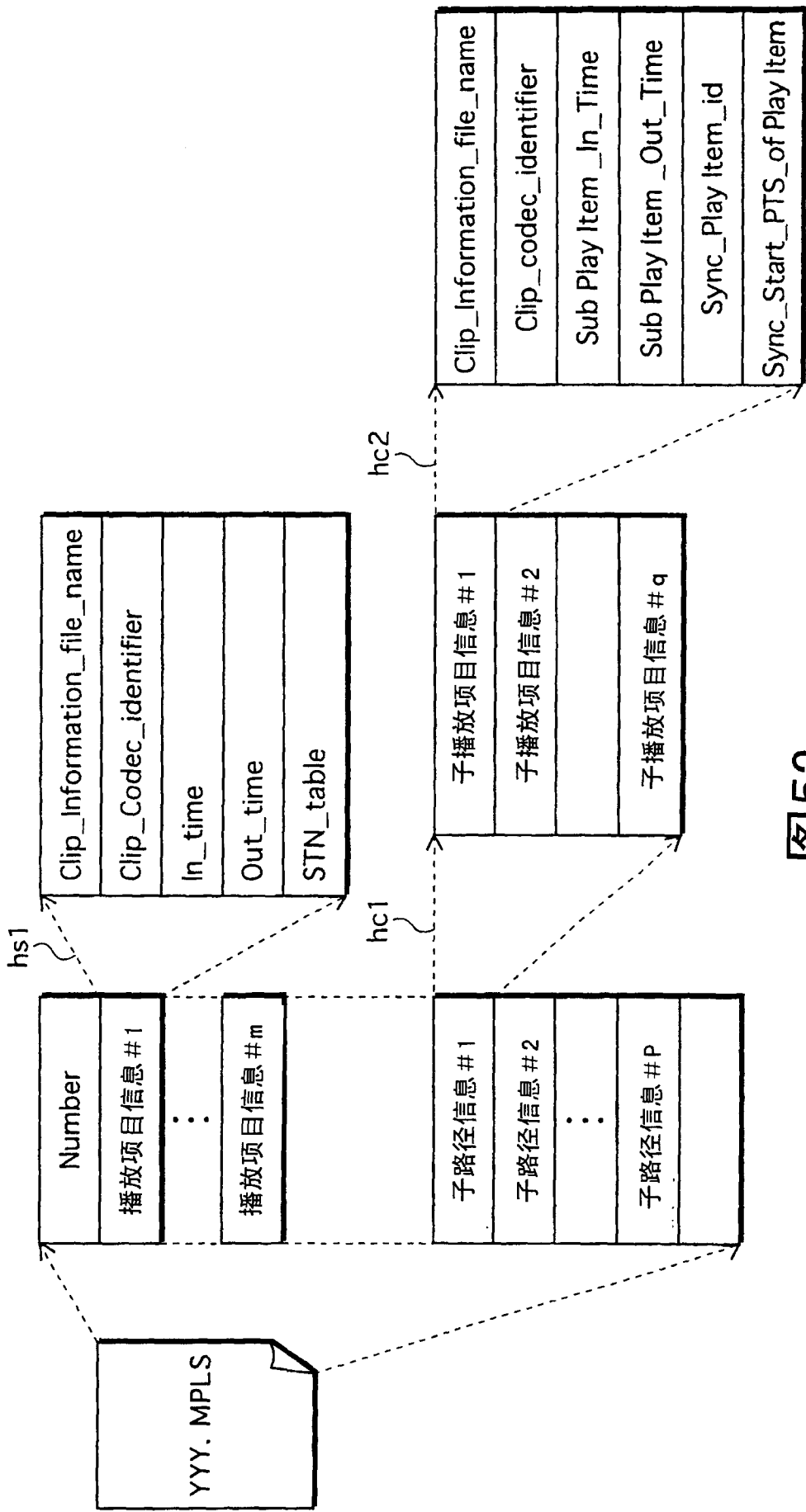


图52

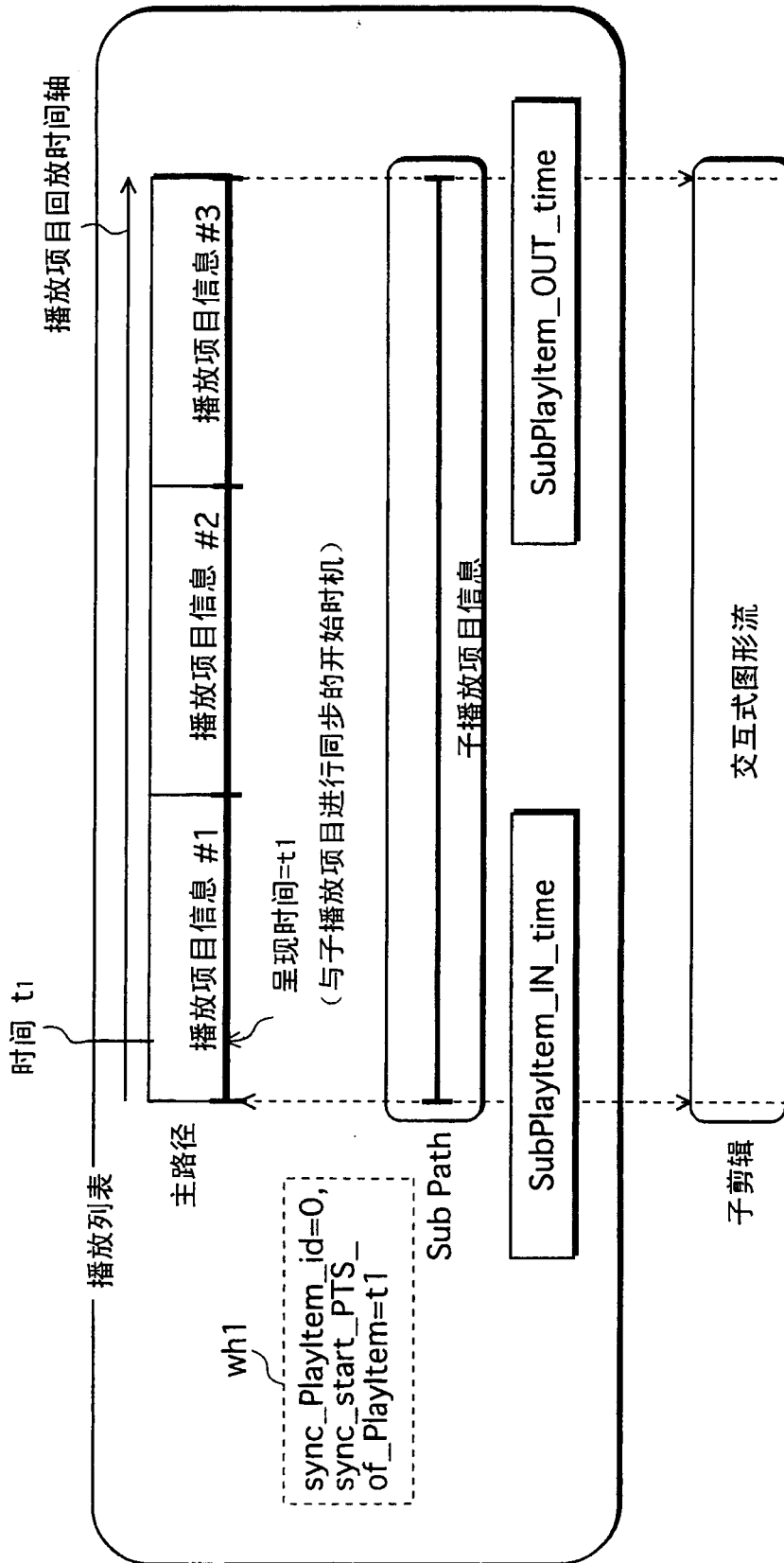


图53

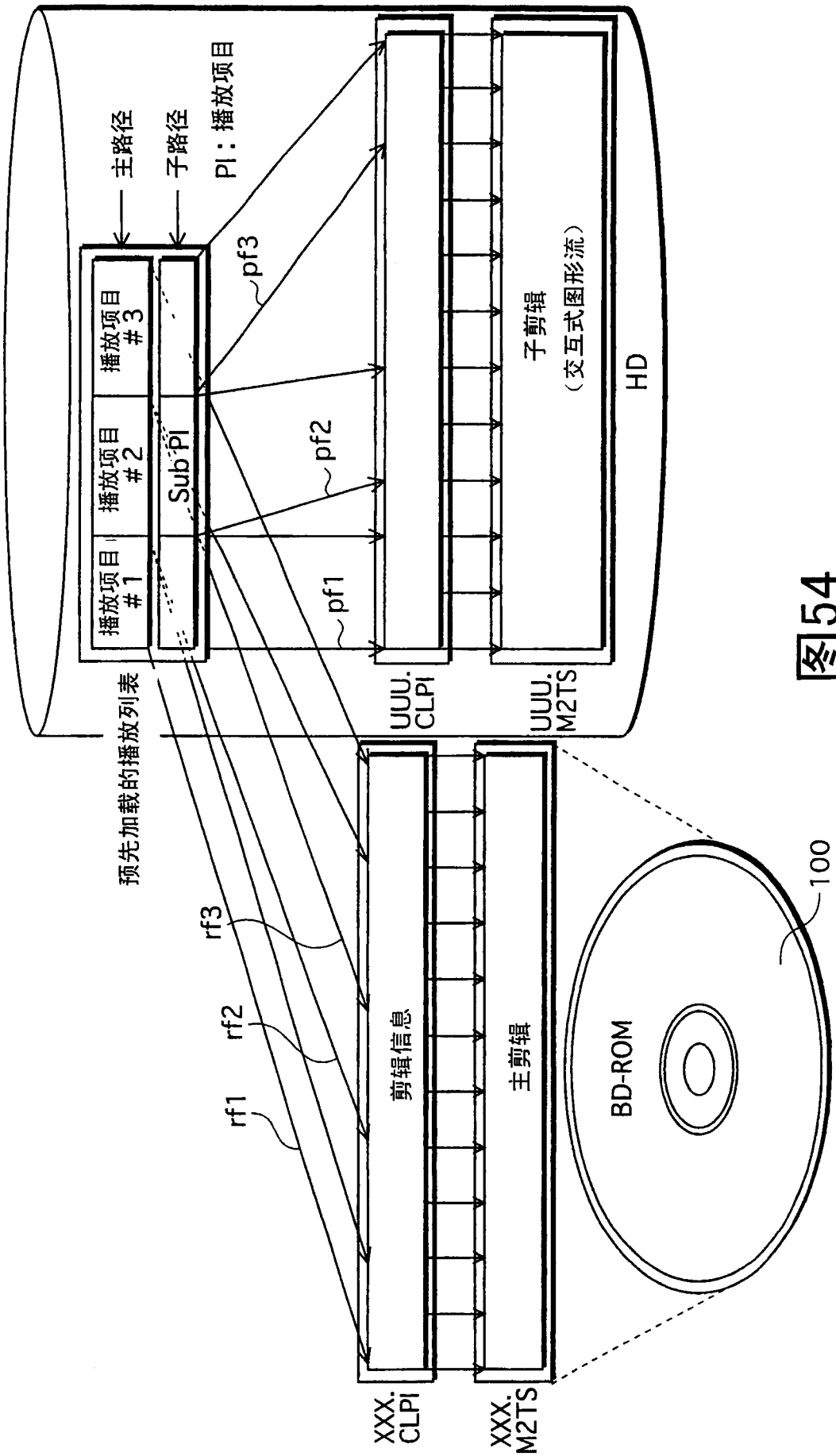


图54