



(12) 发明专利申请

(10) 申请公布号 CN 101702319 A

(43) 申请公布日 2010.05.05

(21) 申请号 200910209029.X

G11B 27/34(2006.01)

(22) 申请日 2006.08.08

G11B 27/32(2006.01)

(30) 优先权数据

G11B 27/10(2006.01)

2006-127502 2006.05.01 JP

G11B 27/034(2006.01)

60/706,871 2005.08.09 US

60/706,897 2005.08.09 US

60/706,937 2005.08.09 US

60/707,066 2005.08.09 US

(62) 分案原申请数据

200680029031.3 2006.08.08

(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 约瑟夫·麦克罗森 池田航

冈田智之 川上义雄

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 赵腾飞 王英

(51) Int. Cl.

G11B 20/10(2006.01)

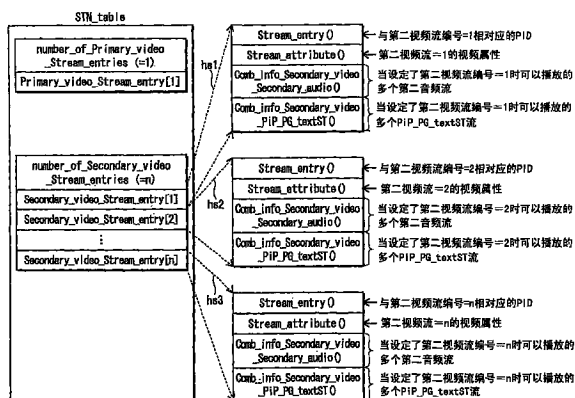
权利要求书 1 页 说明书 56 页 附图 67 页

(54) 发明名称

记录介质、重放设备、方法及程序

(57) 摘要

在本地存储器 200 中记录了多个视频流和一 STN_table。所述多个视频流中每一个都是要与主视频流一起进行播放的第二视频流，且包括表示要在画中画中进行显示的子图像的画面数据，所述画中画由父图像和所述子图像组成。在 STN_table 中，按照优先级的次序描述了允许播放的第二视频流的条目。



1. 一种重放设备,用于根据播放列表信息播放主视频流和第二视频流,其中,所述播放列表信息定义了所述主视频流的主重放部分,和所述第二视频流的子重放部分。

所述播放列表为多个数字流中的每一个定义了重放部分,并且包含主路径信息和子路径信息,

所述重放设备包括:

第一读出单元,用于按照所述主路径信息,读出构成所述主视频流的主重放部分的一个或多个分组;

第二读出单元,用于按照所述子路径信息,读出构成所述第二视频流的子重放部分的一个或多个分组;以及

控制单元,用于通过将一第二视频流所表示的画面合成到一主视频流所表示的画面中,来实现画中画,其中,

如果作为重放目标的主重放部分改变为另一主重放部分,则所述控制单元执行用于连接流的处理,但是如果作为重放目标的主重放部分没有改变为另一主重放部分而作为重放目标的子重放部分改变为另一子重放部分,则不执行所述连接流的处理。

记录介质,重放设备、方法及程序

技术领域

[0001] 本发明属于画中画技术领域。

背景技术

[0002] 画中画是用于显示合成视频的技术,其中将第二窗口叠置在主屏幕上。该技术还允许改变第二窗口在主屏幕上的位置,并在显示该合成视频与不显示该合成视频之间进行切换。

[0003] 注意,在以下专利文献 1 中描述了 DVD 播放器的标准模式的结构。

[0004] <专利文献 1> 日本专利公开 No. 2813245。

发明内容

[0005] [本发明要解决的问题]

[0006] 在 DVD 视频标准应用格式中实现这种画中画需要至少两个视频流。这是因为 DVD 视频播放器的标准模式没有被设计为具有以下功能:同时解码多个视频流,指定时间和位置,然后将两个或更多视频相互叠置地进行播放。为了在合成视频和非合成视频之间进行切换或者改变合成视频的位置,需要在 DVD 视频上单独记录合成流、非合成流、以及其中改变了合成位置的流。然后,命令 DVD 视频播放器播放它们之中的一个。

[0007] 另一方面,对于 BD-ROM 而言,已经进行了从 SDTV 到 HDTV 的视频转换,其结果是数据量上的增加。因此,如果试图在 BD-ROM 上分别记录合成流、非合成流、以及具有不同合成位置的流,则数据的总尺寸会变得过于庞大,用于画中画的视频流不能放在 BD-ROM 中。因此,在 BD-ROM 标准中,BD-ROM 播放器的标准模式已经具有了同时解码多个视频流来获得两个未压缩视频并将这些视频叠置的功能。

[0008] 由于该标准模式具有这样的功能,因此仅仅需要单独记录构成主视频的主视频流和构成第二视频的第二视频流。该记录格式在数据尺寸方面具有优势,因为在 BD-ROM 上没有冗余地记录主视频。另外,没有预先嵌入视频,因此能够根据用户的操作切换第二视频流的合成位置和时间,提高了多功能性。

[0009] 然而,虽然经常以 HDTV 来创建 BD-ROM 的主视频,但是有时是以 50 帧/秒的隔行扫描模式(50i)和 60 帧/秒的隔行扫描模式(60i)来创建第二视频。这是因为第二视频流是在屏幕中的小窗口中显示的,因此需要的分辨率是主视频的一半甚至更小。然而,并不是所有重放设备都能以 50i 和 60i 的模式进行播放,因为 50i 是 PAL 格式,60i 是 NTSC 格式。在每个广播区域中的重放设备通常具有仅按某种模式播放的能力,而在另一广播区域中的重放设备则不具有按该模式播放的能力。因此,在例如第二视频是以 PAL 格式创建的且记录了该第二视频的记录介质被放置在采用 NTSC 格式的重放设备上的情况下,该重放设备不能执行画中画。

[0010] 另外,当采用 HDTV 创建第二视频时,一些重放设备不能解码第二视频。这是因为尽管标准 BD-ROM 重放设备计划配备分别用于主视频和第二视频的解码器,但是第二视频

解码器不具有 HDTV 解码功能。为第二视频解码器提供这种功能意味着几乎没有为重放设备硬件结构的简化留下余地,并且对于产品的价格降低产生负面影响。

[0011] 如上所述的画中画操作可用性取决于视频素材类型的事实,可能变为生产运动画面的电影制作者和制造重放设备的制造商新的要担心的因素。

[0012] 本发明目的在于提供一种记录介质,其能够使重放设备即使是设备在能够解码的素材类型上有所不同的情况下也能够可靠地执行画中画操作。

[0013] [解决问题的手段]

[0014] 为了实现上述目的,本发明是一种其上记录了多个视频流以及一表的记录介质。在此,所述多个视频流中的每一个都是要与主视频流一起播放的第二视频流,并包括表示要在画中画中显示的子图像的画面数据,所述画中画是由父图像和该子图像组成的。所述表包含在所述多个第二视频流中允许播放的第二视频流的条目。

[0015] [本发明的有利效果]

[0016] 即使是在主视频流中存在不能播放的主视频流,通过执行选择下一个流的过程提供了“使用重放设备能够播放的主视频流”的可能性。因此,即使是在重放设备在其对第二视频流的重放能力上有所不同的情况下,也可以使得每个重放设备显示一些第二视频流,从而实现画中画。

[0017] 因此,本发明解决了一些重放设备能够执行画中画而另一些重放设备由于其能力上的差异而不能执行画中画的问题,并且重放设备制造商从而能够创建实现画中画的重放设备并将其投入市场,而不必有任何担心。

[0018] [附图简要说明]

[0019] 图 1 示出了根据本发明的记录介质的使用应用;

[0020] 图 2 示出了 BD-ROM 的内部结构;

[0021] 图 3 是示出了具有 .m2ts 扩展名的文件的结构的简化图;

[0022] 图 4 示出了直到将组成一个 AV 片段 (AVClip) 的 TS 分组写入 BD-ROM 中为止的处理;

[0023] 图 5 示出了 BD-ROM 的物理单元与构成一个文件扩展 (fileextent) 的源分组之间的关系;

[0024] 图 6 示出了被多路复用到一个 AV 片段中的多个基本流的类型;

[0025] 图 7 示出了用于被记录到 BD-ROM 上的多个基本流的 PID 分配映射表 (allocation map);

[0026] 图 8 示出了画中画的一个实例;

[0027] 图 9 示出了片段信息的内部结构;

[0028] 图 10 示出了用于运动画面的视频流的 EP_map 设置;

[0029] 图 11 示出了播放列表 (PlayList) 信息的数据结构;

[0030] 图 12 示出了 AV 片段与播放列表信息之间的关系;

[0031] 图 13 示出了本地存储器 200 的内部结构;

[0032] 图 14 示出了被多路复用到子片段 (SubClip) 中的多个基本流;

[0033] 图 15 示出了在 BD-ROM 标准中的 PID 分配映射表;

[0034] 图 16 示出了播放列表信息的数据结构;

- [0035] 图 17 示出了子路径 (Subpath) 信息的内部结构的展开；
- [0036] 图 18 示出了在本地存储器 200 中的子片段、在本地存储器 200 中的播放列表信息以及在 BD-ROM 上的主片段 (MainClip) 之间的关系；
- [0037] 图 19 示出了 STN_table 的内部结构；
- [0038] 图 20 示出了在图 19 中所示的 STN_table 的整体结构中,用于第二视频流的流条目 (Stream_entry)；
- [0039] 图 21A 示出了主视频流的“Stream_entry”和“Stream_attribute”；
- [0040] 图 21B 示出了第二视频流的“Stream_entry”和“Stream_attribute”；
- [0041] 图 22 示出了本发明的重放设备的内部结构；
- [0042] 图 23 示出了重放设备的输出级的结构；
- [0043] 图 24 功能性地示出了控制器 22；
- [0044] 图 25A 示出了 PSR14 中的比特分配；
- [0045] 图 25B 示出了 PSR29 中的比特分配；
- [0046] 图 26 示出了在 PSR14 中的第二视频流编号的状态转换；
- [0047] 图 27 是示出用于第二视频流的“在重放状态改变时的过程”的处理过程的流程图；
- [0048] 图 28 是示出为当前播放项 (PlayItem) 选择适合的流的处理过程的流程图；
- [0049] 图 29 是示出用于第二视频流的“当要求流改变时的过程”的处理过程的流程图；
- [0050] 图 30 示出了 PiP 元数据 (PiP_metadata) 的内部结构；
- [0051] 图 31 示出了 PiP_horizontal_position 和 PiP_vertical_position 在视频平面上可能采用的坐标；
- [0052] 图 32 示出了当 PiP_time_line_type = 1 时, PiP_metadata_time_stamp 在时间轴上的位置；
- [0053] 图 33 示出了当 PiP_time_line_type = 2 时, PiP_metadata_time_stamp 在时间轴上的位置；
- [0054] 图 34 示出了当 PiP_time_line_type = 3 时, PiP_metadata_time_stamp 在时间轴上的位置；
- [0055] 图 35 是示出基于画中画的重放控制过程的流程图；
- [0056] 图 36 是示出基于画中画的重放控制过程的流程图；
- [0057] 图 37 示出了在由一个播放项组成的主路径以及由两个子播放项组成的子路径之间的关系；
- [0058] 图 38 示出了基于播放列表信息的重放过程的流程图；
- [0059] 图 39 是示出子播放项无缝连接的处理过程的流程图；
- [0060] 图 40 是示出子播放项无缝连接的处理过程的流程图；
- [0061] 图 41 示出了 Primary_audio_stream_entry 和 Secondary_audio_stream_entry 的内部结构,以及 Comb_info_Secondary_video_Secondary_audio 的内部结构；
- [0062] 图 42A 示出了在 PSR1 中的比特分配；
- [0063] 图 42B 示出了在 PSR14 中的比特分配；
- [0064] 图 43 示出了在 PSR14 中的第二视频流编号的状态转换；

- [0065] 图 44 是示出第二视频流的“在重放状态改变时的过程”的处理过程的流程图；
- [0066] 图 45 是示出为当前播放项选择适合的第二音频流的过程的流程图；
- [0067] 图 46 是示出用于第二视频流的处理过程的流程图；
- [0068] 图 47 示出了 STN_table 的一部分,其具体与 PGtestST 流相关；
- [0069] 图 48A 示出了当前 PGtestST 流的编号能够采用的编号范围；
- [0070] 图 48B 示出了在 PSR2 中的比特分配；
- [0071] 图 49 示出了 PiP_PG_TextST 流编号的状态转换,PiP_PG_TextST 流编号是画中画 PGtestST 流的流编号；
- [0072] 图 50 是示出用于 PGtestST 流的“当要求流改变时的过程”的处理过程的流程图；
- [0073] 图 51 是为当前播放项选择适合的 PGTextST 流的处理过程的流程图；
- [0074] 图 52 是示出用于 PGTextST 流的“当要求流改变时的过程”的处理过程的流程图；
- [0075] 图 53 示出了实施例 6 的创作系统的内部结构；
- [0076] 图 54 是示出格式化处理的处理流程的流程图；
- [0077] 图 55 示出了如何对主视频流和第二视频流进行多路复用；
- [0078] 图 56 示出了一种归属关系,其示出了被多路复用的源分组序列属于主视频流和第二视频流中的哪一个 GOP；
- [0079] 图 57 示出了仅为主视频设定的 EP_map；
- [0080] 图 58 示出了多个 EP_map,其每一个都是为主视频和第二视频而设定的；
- [0081] 图 59 示出了播放列表信息的播放列表标记 (PlayListMark) 信息；
- [0082] 图 60 示出了将第二视频流所在的位置指定为章节的播放列表标志信息；
- [0083] 图 61A 示出了组成主视频和第二视频的视频场；
- [0084] 图 61B 示出了要进行合成的视频场彼此同相的情况下的组合；
- [0085] 图 61C 示出了要进行合成的视频场彼此反相的情况下的组合；
- [0086] 图 62A 示出了一种画中画图像,其中,主视频的视频场的偶数线与第二视频的视频场的偶数线匹配；
- [0087] 图 62B 示出了一种画中画图像,其中,主视频的视频场的偶数线与第二视频的视频场的奇数线匹配；
- [0088] 图 63 示出了在要将构成主视频的多个视频场之中的给定视频场 F_x 和来自构成第二视频的多个视频场之中的给定视频场 F_y 进行合成并输出的情况下,如何调节视频场 F_y 的位置和显示时间；
- [0089] 图 64 示出了将第二视频的场 F_y 与主视频的同相场一起进行显示的情况；
- [0090] 图 65 示出了将第二视频的场 F_y 与主视频的反相场一起进行显示的情况；
- [0091] 图 66 示出了将第二视频场与主视频的偶数线一起进行显示的情况；以及
- [0092] 图 67 示出了将第二视频场与主视频的奇数线一起进行显示的情况。
- [0093] [参考标记的解释]
- [0094] 1a BD-ROM 驱动器
- [0095] 1b、c 读出缓冲器
- [0096] 1a、b、c ATC 计数器
- [0097] 2a、d 源逆分组化器 (Source Depacketizers)

[0098]	2c、d	ATC 计数器
[0099]	3a、c	STC 计数器
[0100]	3b、d	PID 过滤器
[0101]	4a	传输缓冲器
[0102]	4c	基本缓冲器
[0103]	4d	视频解码器
[0104]	4e	重排序缓冲器
[0105]	4f	解码画面缓冲器
[0106]	4g	视频平面
[0107]	5a	传输缓冲器
[0108]	5c	基本缓冲器
[0109]	5d	视频解码器
[0110]	5e	重排序缓冲器
[0111]	5f	解码画面缓冲器
[0112]	5g	视频平面
[0113]	6a、b	传输缓冲器
[0114]	7a、b	缓冲器
[0115]	8a、b	音频解码器
[0116]	9a	混合器
[0117]	11a	传输缓冲器
[0118]	11b	交互图形解码器
[0119]	11c	交互图形平面
[0120]	12a	传输缓冲器
[0121]	12b	缓冲器
[0122]	12c	基于文本的字幕解码器
[0123]	13a	传输缓冲器
[0124]	13b	呈现图形解码器 (presentation graphics decoder)
[0125]	13c	现图形平面
[0126]	21	存储器
[0127]	22	控制器
[0128]	23	PSR 组
[0129]	24	PID 转换单元
[0130]	25	网络单元
[0131]	26	操作接收单元
[0132]	41	播放列表处理单元
[0133]	42	过程执行单元
[0134]	43	PiP 控制单元
[0135]	44	混合控制单元
[0136]	100	BD-ROM

[0137] 200 本地存储器

[0138] 300 重放设备

[0139] 400 电视

[0140] 500 AV 放大器

[0141] [实现本发明的最佳模式]

[0142] 实施例 1

[0143] 以下描述本发明的记录介质的实施例。首先,联系本发明的实现来描述一种使用应用。图 1 示出了本发明的记录介质的使用应用。在图 1 中,本发明的记录介质是本地存储器 200。本地存储器 200 用于为由重放设备 300、电视 400、AV 放大器 500 和扬声器 600 组成的家庭影院系统提供运动画面。

[0144] 接下来描述 BD-ROM 100、本地存储器 200、和重放设备 300。

[0145] BD-ROM 100 是其上记录了运动画面的记录介质。

[0146] 本地存储器 200 是内建在重放设备中的硬盘,用于存储从运动画面发行商的服务器所发行的内容。

[0147] 重放设备 300 是支持网络的数字家电,并具有播放 BD-ROM 100 的功能。重放设备 300 还能够经由网络从运动画面发行商的服务器设备下载内容,将所下载的内容存储在本地存储器 200 中,并且将该内容与记录在 BD-ROM 100 中的内容进行合并,以扩展 / 更新 BD-ROM100 的内容。使用称为“虚拟分组”的技术,可以将记录在 BD-ROM100 上的内容与存储在本地存储器 200 中的内容合并,将未记录在 BD-ROM 100 上的数据看作就好像是其记录在 BD-ROM 100 上一样。

[0148] 这样就结束了对本发明的记录介质的使用应用的描述。

[0149] 接下来描述本发明的记录介质的生产应用。通过对 BD-ROM 的文件系统进行改进,能够实现本发明的记录介质。

[0150] <BD-ROM 的概括描述 >

[0151] 图 2 示出了 BD-ROM 的内部结构。图中的层次 4 示出了 BD-ROM,层次 3 示出了 BD-ROM 上的轨道。该图采用横向绘制的形式示出了轨道,尽管实际上轨道是从 BD-ROM 的内侧向外侧螺旋缠绕地形成的。轨道包括导入区 (lead-in area)、数据区 (volume area) 和导出区 (lead-out area)。图中的数据区具有由物理层、文件系统层和应用层所组成的层模型。图中的层次 1 采用目录结构示出了 BD-ROM 的应用层的格式。在层次 1 中, BD-ROM 在根目录下具有 BDMV 目录。

[0152] 此外,三个子目录位于 BDMV 目录下:PLAYLIST 目录、CLIPINF 目录、以及 STREAM 目录。

[0153] PLAYLIST 目录包括扩展名为 mpls 的文件 (00001.mpls)。

[0154] CLIPINF 目录包括扩展名为 clpi 的文件 (00001.clip 和 00002.clip)。

[0155] STREAM 目录包括扩展名为 m2ts 的文件 (00001.m2ts 和 00002.m2ts)。

[0156] 因此,可以看到,根据以上的目录结构,在 BD-ROM 中布置了不同类型的多个文件。

[0157] <BD-ROM 结构 1:AV 片段 >

[0158] 首先,解释扩展名为“m2ts”的文件。图 3 示出了扩展名为“m2ts”的文件的示意性结构。扩展名为“m2ts”的文件 (00001.m2ts 和 00002.m2ts) 存储 AV 片段。AV 片段是采

用 MPEG2 传输流格式的数字流。该数字流是通过以下生成的：将数字化的视频和音频（上方层次 1）转换为由 PES 分组组成的基本流（上方层次 2）；将该基本流转换为 TS 分组（上方层次 3）；类似地，将用于字幕等等的呈现图形（PG）流和用于交互目的的交互流（IG）（下方层次 1 和下方层次 2）转换为 TS 分组（下方层次 3）；然后最终将这些 TS 分组进行多路复用。

[0159] 接下来，解释如何将具有以上所述结构的 AV 片段写入到 BD-ROM 中。图 4 示出了在将构成 AV 片段的 TS 分组写入到 BD-ROM 中之前要对其进行的处理。图中的层次 1 示出了构成 AV 片段的 TS 分组。

[0160] 如图 4 中层次 2 所示，将 4- 字节的 TS_extra_header（图中阴影部分）附加到构成 AV 片段的每个 188- 字节的 TS 分组上，生成每个 192- 字节的源分组。该 TS_extra_header 包括 Arrival_Time_Stamp，其是指示将 TS 分组输入到解码器的时刻的信息。

[0161] 在层次 3 中所示的 AV 片段包括一个或多个“ATC 序列”，其每一个都是源分组的序列。“ATC 序列”是源分组的序列，其中，由 ATC 序列中包含的 Arrival_Time_Stamp 所查询的 Arrival_Time_Clock 不包括“基于到达时刻的不连续性”。换言之，“ATC 序列”是源分组的序列，其中，在 ATC 序列中包含的 Arrival_Time_Stamp 所查询的 Arrival_Time_Clock 是连续的。

[0162] 这种 ATC 序列构成 AV 片段，并以文件名“xxxxx.m2ts”记录在 BD-ROM 上。

[0163] 如同普通计算机文件的情况一样，AV 片段被分割为一个或多个文件扩展（file extent），然后将这些文件扩展记录在 BD-ROM 的多个区中。层次 4 示出了如何将 AV 片段记录在 BD-ROM 上。在层次 4 中，构成该文件的每个文件扩展所具有的数据长度等于或者大于称为“展长（Sextent）”的预定长度。

[0164] 展长是每个文件扩展的最小数据长度，其中，AV 片段被分割为要记录的多个文件扩展。

[0165] 光盘读头（optical pickup）跳转到 BD-ROM 上的某个位置所需要的时间是采用以下公式获得的：

[0166] $T_{\text{jump}} = T_{\text{access}} + T_{\text{overhead}}$ 。

[0167] “Taccess”是与跳转距离（到跳转目的地物理地址的距离）相对应的所需时间。

[0168] 从 BD-ROM 中读出的 TS 分组存储在称为读出缓冲器的缓冲器中，然后被输出到解码器。“Toverhead”是在以称为“Rud”的比特速率进行对读出缓冲器的输入并且用 Secc 表示在 ECC 块中的扇区数量时采用以下公式获得的：

[0169] $T_{\text{overhead}} \leq (2 \times \text{Secc} \times 8) / \text{Rud} = 20 \text{ 毫秒}$ 。

[0170] 从 BD-ROM 中读出的 TS 分组以源分组的状态存储在读出缓冲器中，然后以称为“TS_Recording_rate”的传输速率提供给解码器。

[0171] 为了在将 TS 分组提供给解码器的同时保持传输速率 TS_Recording_rate，就必须在 Tjump 期间，将 TS 分组连续地从读出缓冲器输出到解码器。在此，从读出缓冲器中输出的是源分组而不是 TS 分组。因此，当 TS 分组与源分组在尺寸上的比率为 192/188 时，就必须在 Tjump 期间，将源分组以“ $192/188 \times \text{TS_Recording_rate}$ ”的传输速率从读出缓冲器中连续地输出。

[0172] 因此，不会造成下溢的读出缓冲器占用缓冲器容量用以下公式表示：

[0173] $B_{occupied} \geq (T_{jump}/1000 \times 8) \times ((192/188) \times TS_Recording_rate)$ 。

[0174] 输入到读出缓冲器的输入速率用 R_{ud} 表示,从读出缓冲器输出的速率用 $TS_Recording_rate \times (192/188)$ 表示。因此,通过执行“输入速率 - 输出速率”、并因此通过“ $(R_{ud} - TS_Recording_rate) \times (192/188)$ ”来获得读出缓冲器的占用速率。

[0175] 以“ $B_{occupied}$ ”占用读出缓冲器所需要的时间“ T_x ”是通过以下公式获得的:

[0176] $T_x = B_{occupied} / (R_{ud} - TS_Recording_rate \times (192/188))$ 。

[0177] 当从 BD-ROM 中读出时,必须在时间段“ T_x ”内连续地以比特速率 R_u 输入 TS 分组。从而,当将 AV 片段分割为多个要记录的文件扩展时,每个扩展的最小数据长度展长是通过以下公式获得的: $S_{extent} = R_{ud} \times T_x$

[0178] $= R_{ud} \times B_{occupied} / (R_{ud} - TS_Recording_rate \times (192/188))$

[0179] $\geq R_{ud} \times (T_{jump}/1000 \times 8) \times ((192/188) \times TS_Recording_rate)$

[0180] $/(R_{ud} - TS_Recording_rate \times (192/188))$

[0181] $\geq (R_{ud} \times T_{jump}/1000 \times 8) \times TS_Recording_rate \times 192$

[0182] $/(R_{ud} \times 188 - TS_Recording_rate \times 192)$ 。

[0183] 因此,

[0184] $S_{extent} \geq (T_{jump} \times R_{ud}/1000 \times 8) \times$

[0185] $(TS_Recording_rate \times 192 / (R_{ud} \times 188 - TS_Recording_rate \times 192))$

[0186] 如果构成 AV 片段的每个文件扩展所具有的数据长度都等于或者大于作为不会导致解码器下溢的值而计算得到的展长,则即使是构成 AV 片段的每个文件扩展在 BD-ROM 上位置离散,也会将 TS 分组连续地提供给解码器,从而在重放过程中连续地读出数据。

[0187] 图 5 示出了 BD-ROM 的物理单元与构成一个文件扩展的源分组之间的关系。如层次 2 所示,在 BD-ROM 上形成了多个扇区。如层次 1 所示,构成该文件扩展的源分组被划分为多个组,每个组由 32 个源分组组成。然后将每组源分组写入到三个连续扇区的集合中。32 个源分组的一个组是 6144 字节 (32×192),其等于三个扇区的尺寸 (2048×3)。存储在这三个扇区中的 32 个源分组称为“对齐单元 (Aligned unit)”。以对齐单元为单位来执行对 BD-ROM 的写入。

[0188] 在层次 3 中,为每个 32 扇区的块附加纠错码。带有纠错码的块被称为 ECC 块。只要以对齐单元为单位访问 BD-ROM,重放设备就能够获得 32 个完整的源分组。这样就结束了将 AV 片段写入到 BD-ROM 的写入处理的描述。

[0189] 以下将给出对于要被多路复用到主片段中的基本流的更为详细的描述。

[0190] 图 6 示出了被多路复用到主片段中的多个基本流。要被多路复用到主片段的 STC 序列中的基本流是:具有 PID 0x1011 的主视频流;具有 PID 0x1100 到 0x111F 的主音频流;具有 PID 0x1200 到 0x121F 的 32 个 PG 流;具有 PID 0x1400 到 0x141F 的 32 个 IG 流;以及具有 PID 0x1B00 到 0x1B1F 的 32 个第二视频流。

[0191] 以下描述这些视频流、音频流、PG 流和 IG 流。

[0192] < 主视频流 >

[0193] 主视频流是构成主电影的流,其由 SDTV 和 HDTV 的画面数据组成。该视频流采用 VC-1 视频流、MPEG4-AVC 或者 MPEG2 视频格式。当主视频流是采用 MPEG4-AVC 格式的视频流时,将诸如 PTS 和 DTS 之类的时间戳附加到 IDR、I、P 和 B 画面,并且以画面为单位进行

重放控制。视频流的一个单位是一个附加了 PTS 和 DTS 的、用于重放控制的单元,其被称为“视频呈现单元”。

[0194] < 第二视频流 >

[0195] 第二视频流是构成运动画面的解说等等的流,并且通过将第二视频流的重放视频与主视频流进行合成来实现画中画。该视频流是采用 VC-1 视频流格式、MPEG4-AVC 视频流格式或者 MPEG2 视频流格式的,并且包含“视频呈现单元”。第二视频流是采用 525/60 视频格式、625/50 视频格式、1920/1080 格式、或者 1280/720 格式的。

[0196] 图 8 示出了画中画的一个实例。该图示出了直到预定的时间点 t_p 为止,仅仅播放主视频。能够看出,在该时间点 t_p 之后,播放将第二视频叠置在主视频上的视频。因此,第二视频流由多个画面组成,每个画面都叠置到主视频的一个画面上。与图 6 中所示的第二视频流相同,要与主视频流多路复用到同一 AV 片段中的第二视频流,称为是“In_MUX 第二视频流”。在第二视频流中的、要被多路复用到与主视频流相同的 AV 片段中的普通基本流,称为是“In_MUX 流”。

[0197] < 主音频流 >

[0198] 主音频流是运动画面的主音轨的流,并且主音频流的格式包括 LPCM 音频流格式、DTS-HD 格式、DD/DD+ 音频流格式、以及 DD/MLP 音频流格式。将时间戳附加到音频流中的音频帧上,并且以音频帧为单位进行重放控制。音频流的一个单位是一个附加了时间戳的、用于重放控制的单元,其称为“音频呈现单元”。注意,尽管在此运动画面的子音轨没有记录在 BD-ROM 上,但是其被称为第二音频流。

[0199] < PG 流 >

[0200] PG 流是构成以某种语言书写的字幕的图形流。存在分别对应于多种语言的多个流,例如英语、日语和法语。PG 流包括诸如以下的功能段:PCS(呈现控制段)、PDS(调色板(pallet)定义段)、WDS(窗口定义段)、ODS(对象定义段)、以及 END(显示设置段的结束)。ODS(对象定义段)是定义图形对象的功能段,图形对象就是字幕。

[0201] WDS 是定义图形对象在屏幕上的绘制区的功能段。PDS(调色板定义段)是定义在绘制图形对象时的颜色的功能段。PCS 是定义在显示字幕时的页面控制的功能段。这种页面控制包括:突现/突逝(Cut-In/Out)、淡入/淡出(Fade-In/Out)、颜色改变、卷动、划入/划出(Wipe-In/Out)。采用 PCS 的页面控制可以实现某种显示效果,例如,在显示下一个字幕的同时将当前字幕淡出。

[0202] 除了呈现图形流之外,用于表示字幕(其没有被多路复用到与主视频流相同的 AV 片段中)的流包括文本字幕(textST)流。textST 流是用字符码表示字幕内容的流。在 BD-ROM 标准中,呈现图形流和 textST 流的组合被称为“PGTextST 流”。

[0203] < IG 流 >

[0204] IG 流是用于实现交互控制的图形流。IG 流所定义的交互控制是与 DVD 重放设备的交互控制兼容的交互控制。IG 流包括诸如以下的功能段:ICS(交互合成段)、PDS(调色板定义段)、以及 ODS(对象定义段)。ODS(对象定义段)是定义图形对象的功能段。在交互屏幕上的按钮是由多个这样的对象绘制的。PDS(调色板定义段)是定义在绘制图形对象时的颜色的功能段。ICS(交互合成段)是实现状态改变的功能段,其中,按钮状态根据用户操作而改变。ICS 包括当对某个按钮进行了确认操作时执行的按钮命令。

[0205] 图 7 示出了记录在 BD-ROM 上的基本流的 PID 分配映射表。该 PID 分配映射表的左列指示多个 PID 范围。右列指示分配给每个范围的基本流。当根据该描述来查询该分配表时,能够看到以下:program_map 被分配在 PID 范围 0x0100;PCR 在范围 0x1001 中;主视频流在范围 0x1011 中;主音频流在范围 0x1100 到 0x111F 中;PG 流在范围 0x1200 到 0x121F 中;IG 流在 0x1400 到 0x141F 中;In_MUX 第二视频流在范围 0x1B00 到 0x1B1F 中。

[0206] 如从该 PID 分配映射表中能够看到的,用于主视频流的范围和用于第二视频流的范围是不同的。

[0207] 用于构成画中画的第二视频流的 PES 分组所具有的高字节与主视频流的 PES 分组的 PID 的高字节不同。因此,即使是主视频流和第二视频流被多路复用到一个 AV 片段中,也能够通过查询 PID 的高字节来将构成每种类型的视频流的 PES 分组提供给不同的解码器。

[0208] <BD-ROM 结构 2:片段信息 >

[0209] 接下来描述扩展名为“clip”的文件。扩展名为“clip”的文件(00001.clip 和 00002.clip)存储片段信息。片段信息是对每个 AV 片段的管理信息。图 9 示出了片段信息的内部结构。如该图左侧所示,片段信息包括:

[0210] i) “ClipInfo()”,其中存储了关于 AV 片段的信息;

[0211] ii) “Sequence Info()”,其中存储了关于 ATC 序列和 STC 序列的信息;

[0212] iii) “Program Info()”,其中存储了关于节目序列的信息;以及

[0213] iv) “Characteristic Point Info(CPI())”。

[0214] “ClipInfo”包括“application_type”,其指示片段信息所指的 AV 片段的应用类型。对 ClipInfo 进行查询允许确认应用类型是主片段还是子片段、是否包含视频、或者是否包含静态画面(幻灯片)。另外,上述的 TS_recording_rate 是在 ClipInfo 中描述的。

[0215] “Sequence Info”是关于在 AV 片段中包含的一个或多个 STC 序列和 ATC 序列的信息。提供这些信息的原因是,将基于系统时间的不连续性和基于到达时间的不连续性预先通知给重放设备。就是说,如果存在这种不连续性,则存在在 AV 片段中出现具有相同值的 PTS 和 ATS 的可能性。这可能会造成错误重放。提供“Sequence Info”来指示 STC 或者 ATC 在传输流中从哪里到哪里是连续的。

[0216] “Program Info”是指示内容恒定的节目的一个部分(所谓的“节目序列”)。在此,“节目”是具有用于同步重放的公共时间轴一组基本流。提供“Program Info”的原因是:将节目内容改变时的点预先通知给重放设备。在此应该注意,节目改变时的点是例如视频流的 PID 发生改变时的点、或者视频流类型从 SDTV 改变为 HDTV 时的点。

[0217] 接下来描述“Characteristic Point Info”。图 9 中的引导线 cu2 指示 CPI 结构的展开。如引导线 cu2 所指示的,CPI 包括 Ne 个 EP_map_for_one_stream_PID(从 EP_map_for_one_stream_PID[0] 到 EP_map_for_one_stream_PID[Ne-1])。这些 EP_map_for_one_stream_PID 是属于该 AV 片段的多个基本流的 EP_map。EP_map 是与进入时刻(PTS_EP_start)相关联地指示在进入位置处的分组编号(SPN_EP_start)的信息,在所述进入位置处,在一个基本流中存在访问单元(Access Unit)。图中的引导线 cu3 指示 EP_map_for_one_stream_PID 的内部结构的展开。

[0218] 从该展开中了解到,EP_map_for_one_stream_PID 包括:Ne 个 EP_High(EP_High(0) 到 EP_High(Nc-1)) 和 Nf 个 EP_Low(EP_Low(0) 到 EP_Low(Nf-1))。在此,EP_High

用于指示访问单元（非 IDR I 画面、IDR 画面）的 SPN_EP_start 和 PTS_EP_start 的高位比特，EP_Low 用于指示访问单元（非 IDR I 画面、IDR 画面）的 SPN_EP_start 和 PTS_EP_start 的低位比特。

[0219] 图中的引导线 cu4 指示 EP_High 的内部结构的展开。如引导线 cu4 所示，EP_High(i) 包括：“ref_to_EP_Low_id[i]”，其是对 EP_Low 的查询值 (reference value)；“PTS_EP_High[i]”，其指示访问单元（非 IDR I 画面、IDR 画面）的 PTS 的高位比特；以及“SPN_EP_High[i]”，其指示访问单元（非 IDR I 画面、IDR 画面）的 SPN 的高位比特。在此，“i”是所给定的 EP_High 的标识符。

[0220] 图中的引导线 cu5 指示 EP_Low 的内部结构的展开。如引导线 cu4 所示，EP_Low(i) 包括：“is_angle_change_point(EP_Low_id)”，其指示相应的访问单元是否是 IDR 画面；“l_end_position_offset(EP_Low_id)”，其指示相应的访问单元的尺寸；“PTS_EP_Low[EP_Low_id]”，其指示访问单元（非 IDR I 画面、IDR 画面）的 PTS 的低位比特；以及“SPN_EP_Low[EP_Low_id]”，其指示访问单元（非 IDR I 画面、IDR 画面）的 SPN 的低位比特。在此，“EP_Low_id”是所给定的 EP_Low 的标识符。

[0221] < 片段信息解释 2 :EP_Map >

[0222] 在此，使用具体实例来解释 EP_map。图 10 示出了运动画面的视频流的 EP_map 设置。层次 1 示出了按照显示顺序排列的多个画面（在 MPEG4-AVC 中定义的 IDR 画面、I 画面、B 画面、P 画面）。层次 2 示出了这些画面的时间轴。层次 4 示出了在 BD-ROM 上的 TS 分组序列，层次 3 示出了 EP_map 的设置。

[0223] 在此假设，在层次 2 的时间轴中，在 t1 到 t7 的每个时间点上都存在 IDR 画面或者 I 画面。在时间点 t1 到 t7 中相邻两个时间点之间的间隔大约为 1 秒。用于运动画面的 EP_map 被设定为利用进入时刻 (PTS_EP_start) 指示 t1 到 t7，并且与进入时刻相关联地指示进入位置 (SPN_EP_start)。

[0224] < 播放列表信息 >

[0225] 接下来描述播放列表信息。扩展名为“mpls”的文件 (00001.mpls) 是其中存储了播放列表 (PL) 信息的文件。

[0226] 图 11 示出了播放列表信息的数据结构。如图中的引导线 mp1 所示，播放列表信息包括：主路径信息 (MainPath())，其定义主路径；播放列表标记信息 (PlayListMark())，其定义章节；以及其他扩展数据 (Extension_Data)。

[0227] < 播放列表信息解释 1 :主路径信息 >

[0228] 首先描述主路径。主路径是按照诸如主视频流之类的视频流和音频流而定义的。

[0229] 如箭头 mp1 所示，主路径由多个播放项信息定义：播放项信息 #1 到播放项信息 #m。播放项信息定义了构成主路径的一个或多个逻辑播放部分。

[0230] 在图中的引导线 hs1 指示了播放项信息的结构的展开。如引导线 hs1 所示，播放项信息包括：“Clip_Information_file_name”，其指示该重放部分的入点和出点所归属的 AV 片段的重放部分信息的文件名；“Clip_codec_identifier”，指示 AV 片段编码方法；“is_multi_angle”，指示播放项是否是多视角的；“connection_condition”，指示是否无缝连接当前播放项和先前播放项；“ref_to_STC_id[D]”，唯一性地指示作为该播放项的目标的 STC 序列；“In_time”，其是指示该重放部分的开始点的时间信息；“Out_time”，其是指示

该重放部分的结束点的时间信息;“U0_mask_table”,指示该播放项应该屏蔽哪种用户操作;“PlayItem_random_access_flag”,指示是否允许对该播放项中的中间点进行随机访问;“Still_mode”,指示在该播放项的重放结束之后是否继续最后画面的静态显示;以及“STN_table”。在这些之中,指示重放部分的开始点的时间信息“In_time”和指示重放部分的结束点的时间信息“Out_time”构成重放路径。重放路径信息由“In_time”和“Out_time”组成。

[0231] 图 12 示出了在 AV 片段与播放列表信息之间的关系。层次 1 示出了播放列表信息的时间轴(播放列表时间轴)。层次 2 到 5 示出了 EP_map 所指的视频流。

[0232] 该播放列表信息包括两个播放项信息:播放项信息 #1 和播放项信息 #2。在播放项信息 #1 和播放项信息 #2 中分别包含的“In_time”和“Out_time”定义了两个重放部分。当排列这些重放部分时,定义了一个与 AV 片段时间轴不同的时间轴。这就是在层次 1 中所示的播放列表时间轴。因此,可以通过定义播放项信息来定义与 AV 片段不同的重放路径。

[0233] 这样就结束了对 BD-ROM 100 的描述。

[0234] <本地存储器 200>

[0235] 以下描述了本地存储器 200,其是本发明的记录介质。图 13 示出了本地存储器 200 的内部结构。如图所示,本发明的记录介质能够通过改进应用层来生产。

[0236] 图中的层次 4 示出了本地存储器 200,层次 3 示出了本地存储器 200 的轨道。该图采用横行绘制形式画出了轨道,尽管在实际中轨道是从本地存储器 200 的内侧向外侧螺旋缠绕地形成的。轨道由导入区、数据区、以及导出区组成。图中的数据区具有由物理层、文件系统层和应用层所组成的层模型。图中的层次 1 采用目录结构示出了本地存储器 200 的应用层的结构。

[0237] 在图 13 中所示的目录结构中,在根目录下存在子目录“organization#1”。此外,在目录“organization#1”下存在子目录“disk#1”。目录“organization#1”被指定给运动画面的具体提供商。目录“disk#1”被指定给由该提供商所提供的每个 BD-ROM。

[0238] 采用这种指定给具体提供商的目录包含与 BD-ROM 相对应的目录的结构,独立地存储每个 BD-ROM 的下载数据。与存储在 BD-ROM 中的信息相类似,在子目录“disk#1”下,存储了以下信息:播放列表信息(“00002.mpls”);片段信息(“00003.clpi”和“00004.clpi”);以及 AV 片段(“00003.m2ts”和“00004.m2ts”)。

[0239] 以下将描述本地存储器 200 中的组件:播放列表信息、片段信息和 AV 片段。

[0240] <本地存储器 200 结构 1:AV 片段>

[0241] 在本地存储器 200 中的 AV 片段(00003.m2ts 和 00004.m2ts)组成了子片段。子片段是通过对四种类型的基本流-视频流、音频流、PG 流和 IG 流-进行多路复用而形成的。以下给出了被多路复用到 AV 片段中的基本流类型的详细描述。

[0242] 图 13 示出了本地存储器 200 的内部结构。要被多路复用到子片段中的基本流是:具有 PID 0x1800 的 textST 流;具有 PID 0x1A00 到 0x1A1F 的主音频流;具有 PID 0x1B00 到 0x1B1F 的 32 个 Out_of_MUX 第二视频流;具有 PID 0x1200 到 0x121F 的 32 个 PG 流;以及具有 PID 0x1400 到 0x141F 的 32 个 IG 流。作为图 14 中所示的第二视频流,要与主视频流多路复用到不同的 AV 片段中的第二视频流,称为“Out_of_MUX 第二视频流”。除了第二视频流之外,要被多路复用到与主视频流不同的 AV 片段的普通基本流,称为是“Out_MUX 流”。

[0243] 图 15 示出了被多路复用到子片段中的基本流的 PID 分配映射表。该 PID 分配映射表的左列指示多个 PID 范围。右列指示分配给每个范围的基本流。当根据该描述查询该映射表时,能够看到以下:program_map 被分配在 PID 范围 0x0100 中;PCR 在范围 0x1001 中;PG 流在范围 0x1200 到 0x121F 中;IG 流在 0x1400 到 0x141F 中;textST 流在范围 0x1800 中;第二音频流在范围 0x1A00 到 0x1A1F 中;第二视频流在范围 0x1B00 到 0x1B1F 中。

[0244] <本地存储器 200 结构 2:播放列表信息>

[0245] 接下来描述本地存储器 200 中的播放列表信息。扩展名为“mpls”的文件(00002.mpls)是定义了通过将称为主路径和子路径的两类重放路径绑定作为播放列表(PL)而得到的一个组的信息。图 16 示出了播放列表信息的数据结构。如图所示,播放列表信息包括:主路径信息(MainPath()),其定义主路径;播放列表标记信息(PlayListMark()),其定义章节;子路径信息,其定义子路径;以及其他扩展数据(Extension_Data)。播放列表信息和播放项的内部结构与在 BD-ROM 中的播放列表信息和播放项相同,因此在此省略了对其的描述。以下描述子路径信息。

[0246] <播放列表信息解释 1:子路径信息>

[0247] 主路径是为主片段(其是主视频)而定义的重放路径,子目录是为与主路径同步的子片段而定义的重放路径。

[0248] 图 17 示出了子路径信息的内部结构的展开。如图中箭头 hc0 所示,每个子路径包括指示子片段类型的“SubPath_type”以及一个或多个子播放项信息(...SubPlayItem(...))。

[0249] 图中的引导线 hc1 指示子播放项信息的结构的展开。

[0250] 子播放项定义了与主路径相独立的一个或多个基本流重放路径,并且用于表示出这些重放路径如何与主路径进行同步的类型。当子播放项使用主视频、PG、IG、第二音频和第二视频的子路径时,这些子播放项使用在播放列表中的播放项与主路径进行同步。用于基本流重放的子路径所使用的基本流被多路复用到子片段,即,与主路径的播放项所使用的主片段相独立的片段。

[0251] 接下来描述子播放项的内部结构。如图中的箭头 hc1 所示,子播放项信息包括:“Clip_information_file_name”、“Clip_codec_identifier”、“ref_to_STC_id[0]”、“SubPlayItem_In_time”、“SubPlayItem_Out_time”、“sync_PlayItem_id”、以及“sync_start_PTS_of_PlayItem”。

[0252] “Clip_information_file_name”是通过描述片段信息的文件名来唯一性地指定与该子播放项相对应的子片段的信息。

[0253] “Clip_codec_identifier”指示 AV 片段的编码系统。

[0254] “ref_to_STC_id[0]”唯一性地指示子播放项所针对的 STC 序列。

[0255] “SubPlayItem_In_time”是指示在子片段的重放时间轴上的子播放项开始点的信息。

[0256] “SubPlayItem_Out_time”是指示在子片段的重放时间轴上的子播放项结束点的信息。

[0257] “sync_PlayItem_id”是用于从组成主路径的多个播放项中唯一性地指定该子播放项与之同步的播放项的信息。“SubPlayItem_In_time”存在于由 sync_PlayItem_id 所指

定的播放项重放时间轴上。

[0258] “sync_start_PTS_of_PlayItem”以 45KHz 的时间精度指示了 SubPlayItem_In_time 所指定的子播放项的开始点在 sync_PlayItem_id 所指定的播放项的重放时间轴上的位置。在子播放项定义了第二视频流上的重放部分并且子播放项的 sync_start_PTS_of_PlayItem 指示了在播放项时间轴上的时间点的情况下,子播放项实现了“同步画中画”。

[0259] 另外,能够在 sync_start_PTS_of_PlayItem 中设定一个未指定值 (0xFFF)。该未指定值使得在由 sync_PlayItem_id 所指定的播放项时间轴上的、用户执行了锁定操作的时间点,成为与由 sync_PlayItem_id 所指定的播放项同步的点。在 sync_start_PTS_of_PlayItem 中设定该未指定值并且子播放项指示第二视频流的重放的情况下,子播放项实现“异步画中画”。

[0260] <子路径信息细节 1:SubPath_type>

[0261] 这样结束对子路径信息的描述。接下来描述 SubPath_type。当在 SubPath_type 中设定了范围为 0 到 255 的值时,SubPath_type 指示由该子路径信息所定义的子路径是哪种重放路径。以下解释 SubPath_type 所表示的重放路径的 3 种典型类型。

[0262] Subpath_Type = 5

[0263] :画中画呈现路径的 Out-of-mux 与同步类型

[0264] 在该情况下,子播放项定义了与 Out-of-mux 同步播放的画中画重放路径。在该路径中所使用的基本流被多路复用到子片段,即,与由主路径的播放项所使用的主片段相独立的片段。

[0265] Subpath_Type = 6

[0266] :画中画呈现路径的 Out-of-mux 与异步类型

[0267] 在该情况下,子播放项定义了与 Out-of-mux 流不同步播放的画中画重放路径。这就是说,在该路径中所使用的基本流被多路复用到子片段,即,与由播放项所使用的片段不同的片段。当 Subpath = 6 时,子路径仅仅包含一个子播放项。

[0268] 当子播放项的 SubPath_type 设定为“6”时, sync_PlayItem_id 和 sync_start_PTS_of_PlayItem 变为无效并且分别设定为 0x0000 和 0x00000000。

[0269] Subpath_Type = 7

[0270] :画中画呈现路径的 In_Mux 与同步类型

[0271] 在该情况下,子播放项定义了与 In_mux 流同步播放的画中画重放路径。即,当 Subpath_Type = 7 时,在该路径中所使用的基本流与要同步播放的播放项所使用的基本流一起被多路复用到同一片段中。

[0272] 这样就结束了对 SubPath_type 的描述。

[0273] <子路径信息细节 2:3 个对象之间的关系>

[0274] 在此,3 个对象指的是本地存储器 200 中的子片段、本地存储器 200 中的播放列表信息以及在 BD-ROM 中的主片段。

[0275] 图 18 示出了本地存储器 200 中的子片段、本地存储器 200 中的播放列表信息以及在 BD-ROM 中的主片段之间的关系。图中的层次 1 指示在本地存储器 200 中存在的子片段。如层次 1 中所示,存在不同类型的子片段:第二视频流、第二音频流、PG 流和 IG 流。其任何一个都用作同步重放的子路径。

[0276] 层次 2 指示由播放列表信息所定义的两个时间轴。在层次 2 中,下面的时间轴是由播放项信息定义的播放项时间轴,上面的时间轴是由子播放项定义子播放项时间轴。

[0277] 如图所示,能够看出:子播放项信息的 SubPlayItem_Clip_information_file_name 在子片段选择中起作用,从多路复用到存储在 STREAM 目录中的 .m2ts 文件中的多个 Out-of-Mux 流中,选出作为重放部分指定的目标的 Out-of-Mux 流。

[0278] 还能够看出:SubPlayItem.In_time 和 SubPlayItem.Out_time 在定义重放部分的开始点和结束点中起作用。

[0279] 箭头 Sync_PlayItem_Id 在指定哪个播放项与子播放项同步中起作用。sync_start_PTS_of_PlayItem 在确定在播放项时间轴上的 SubPlayItem_In_time 的时间点中起作用。

[0280] 这样就结束了对子路径信息的描述。

[0281] <STN_table>

[0282] STN_table 是在 BD-ROM 和本地存储器 200 中的播放列表信息的一个组成部分。以下描述在本地存储器 200 中的播放列表信息。

[0283] STN_table 示出了在被多路复用到由播放项信息的 Clip_Information_file_name 所指定的片段中的多个基本流以及由子播放项信息的 Clip_Information_file_name 所指定的 OUT_of_MUX 流之中,允许进行重放的流。更具体而言,在 STN_table 中,多路复用到主片段中的多个 In_MUX 流以及多路复用到子片段中的 Out_of_MUX 流的流条目 (Stream_entry) 分别与流属性 (Stream_attribute) 相关联。

[0284] 图 19 示出了 STN_table 的整体结构。图 20 示出了在图 19 中所示的 STN_table 的整体结构中,第二视频流的流条目。如图所示,该 STN_table 包括 n 个 Secondary_video_stream_entry (Secondary_video_stream_entry[1] 到 Secondary_video_stream_entry[n]) 和第二视频流的数量 (number_of_Secondary_video_stream_entries (= n))。

[0285] 引导线 hs1 指示了 Secondary_video_stream_entry[1] 的内部结构的展开。这就是说,Secondary_video_stream_entry[1] 到 Secondary_video_stream_entry[n] 是从同一个类结构中生成的多个实例,并且其每一个都具有与引导线 hs1 所指示的相同的内部结构。在每个 Secondary_video_stream_entry 的方括号 [] 中的数字指示该 Secondary_video_stream_entry 在 STN_table 中的次序。

[0286] 如引导线 hs1 所示,Secondary_video_stream_entry[1] 包括:“Stream_entry”,将与第二视频流编号 = 1 相对应的 PID 提供给重放设备;“Stream_attribute”,指示与第二视频流编号 = 1 相对应的视频属性;“Comb_info_Secondary_Video_Secondary_Audio”,指示当在第二视频流编号中设定为“1”时变为可播放的第二音频流;以及“Comb_info_Secondary_Video_PiP_PG_textST()”,指示当在第二视频流编号中设定为“1”时变为可播放的 PG 流或者 textST 流。

[0287] 如引导线 hs2 所示,Secondary_video_stream_entry[2] 包括:“Stream_entry”,将与第二视频流编号 = 2 相对应的 PID 提供给重放设备;“Stream_attribute”,指示与第二视频流编号 = 2 相对应的视频属性;“Comb_info_Secondary_Video_Secondary_Audio”,指示当在第二视频流编号中设定为“2”时变为可播放的第二音频流;以及“Comb_info_Secondary_Video_PiP_PG_textST()”,指示当在第二视频流编号中设定为“2”时变为可播

放的 PG 流或者 textST 流。

[0288] 对于引导线 hs3 也是相同的。因此,位于 STN_table 中的第 x 个位置处的 Secondary_video_stream_entry[x] 为重放设备指示以下信息:当为第二视频流编号设定为“x”时与“x”相对应的第二视频流的 PID;该第二视频流的视频属性;以及第二音频流与 PGTextST 流的可用组合。

[0289] 图 21A 示出了主视频流的“Stream_entry”和“Stream_attribute”。该“Stream_entry”包括“ref_to_stream_PID_of_mainClip”,其指示构成该主视频流的 PES 分组的分组标识符。

[0290] 该“Stream_attribute”包括指示视频流显示格式的“Video_format”和指示视频流显示频率的“frame_rate”。

[0291] 图 21B 示出了第二视频流的“Stream_entry”。如图所示,第二视频流的“Stream_entry”包括:“ref_to_Sub_Path_id”,指示涉及该第二视频流的子路径信息;以及“ref_to_stream_PID_of_mainClip”,指示构成该第二视频流的 PES 分组的分组标识符。

[0292] 这样就结束了对本发明的记录介质的描述。除了这些组件之外,BD-ROM 还包括例如电影对象 (Movie Object)、BD-J 对象和 Index.bdmv。

[0293] 电影对象采用与 DVD 视频中所用的类似的描述来命令重放设备进行动态控制过程,而 BD-J 采用以 Java™ 应用程序进行的描述来命令重放设备进行动态控制过程。Index.bdmv 是一个表,其指示构成标题的电影对象或者 BD-J 对象。

[0294] 这些组件并不是本发明的重点,因此在本实施例中省略了对其的详细描述。这样就结束了对本发明的记录介质的描述。

[0295] < 重放设备 >

[0296] 图 22 示出了本发明的重放设备的内部结构。本发明的重放设备是基于图中所示的内部结构而商业制造的。该重放设备主要包括两个两个部件:系统 LSI 和驱动装置,并且通过在该设备的箱体和底座上安装这些部件来商业制造该重放设备。这样制造的重放设备包括:BD-ROM 驱动器 1a;读出存储器 1b 和 1c;ATC 计数器 2a 和 2c;源逆分组化器 2b 和 2d;ATC 计数器 2b 和 2d;STC 计数器 3a 和 3c;PID 过滤器 3b 和 3d;传输缓冲器 (TB) 4a;基本缓冲器 (EB) 4c;视频解码器 4d、重排序缓冲器 4e;已解码画面缓冲器 4f;视频平面 4g;传输缓冲器 (TB) 5a;基本缓冲器 (EB) 5c;视频解码器 5d、重排序缓冲器 5e;已解码画面缓冲器 5f;视频平面 5g;缓冲器 6a 和 6b;缓冲器 7a 和 7b;音频解码器 8a 和 8b;混合器 9a;切换器 10a、10b、10c、10d 和 10e;传输缓冲器 (TB) 11a;交互图形解码器 11b;交互图形平面 11c;传输缓冲器 (TB) 12a、缓冲器 12b;基于文本的字母解码器 12c;传输缓冲器 (TB) 13a、呈现图形解码器 13b;呈现图形平面 13c;存储器 21;控制器 22;PSR 组 23;PID 转换单元 24;网络单元 25;操作接收单元 26;以及本地存储器 200。注意,在图中未示出该重放设备的输出级。以下采用另一个示出了内部结构的附图来描述该输出级。

[0297] BD-ROM 驱动器 1a 加载 / 弹出 BD-ROM,并进行对 BD-ROM 的访问。

[0298] 读出缓冲器 (RB) 1b 累积从 BD-ROM 读出的源分组序列。

[0299] 读出缓冲器 (RB) 1c 累积从本地存储器 200 读出的源分组序列。

[0300] 通过使用在构成主片段的多个源分组之中位于重放部分开始处的源分组的 ATS 来重置 ATC 计数器 2a,然后将 ATC 输出到源逆分组化器 2b。

[0301] 源逆分组化器 2b 从构成主片段的源分组中取出 TS 分组,并发送该 TS 分组。在发送时,源逆分组化器 2b 根据每个 TS 分组的 ATS 来调节输入到解码器中的时刻。更具体而言,当 ATC 计数器 2a 所生成的 ATC 的值变为与源分组的 ATS 的值相同时的时刻,源逆分组化器 2b 以 TS_Recording_Rate 仅将该 TS 分组传输给 PID 过滤器 3b。

[0302] 通过使用在构成子片段的多个源分组之中位于重放部分开始处的源分组的 ATS 来重置 ATC 计数器 2c,然后将 ATC 输出到源逆分组化器 2d。

[0303] 源逆分组化器 2d 从构成子片段的源分组中取出 TS 分组,并发送该 TS 分组。在发送时,源逆分组化器 2d 根据每个 TS 分组的 ATS 来调节输入到解码器中的时刻。更具体而言,当 ATC 计数器 2c 所生成的 ATC 的值变为与源分组的 ATS 的值相同时的时刻,源逆分组化器 2d 以 TS_Recording_Rate 仅将该 TS 分组传输给 PID 过滤器 3d。

[0304] 用主片段的 PCR 来重置 STC 计数器 3a,并输出 STC。

[0305] PID 过滤器 3b 是主片段的去复用单元,其输出在从源逆分组化器 2b 输出的多个源分组之中的、具有由 PID 转换单元 24 通知给视频解码器 4d 和 5d、音频解码器 8a、交互图形解码器 11b 和呈现图形解码器 13b 的 PID 查询值的源分组。这些解码器中的每一个都接收经由 PID 过滤器 3b 的基本流,并根据主片段的 PCR 执行从解码处理到重放处理的处理过程。因此,基于主片段的 PCR,对在经由 PID 过滤器 3b 传递之后输入到每个解码器中的基本流进行解码和重放。

[0306] 用子片段的 PCR 来重置 STC 计数器 3c,并输出 STC。PID 过滤器 3d 执行关于该 STC 的去复用。

[0307] PID 过滤器 3d 是子片段的去复用单元,其输出在从源逆分组化器 2d 输出的多个源分组之中的、具有与由 PID 转换单元 24 通知给音频解码器 8b、交互图形解码器 11b 和呈现图形解码器 13b 的 PID 查询值的源分组。因此,基于子片段的 PCR,对在经由 PID 过滤器 3d 传递之后输入到每个解码器中的基本流进行解码和重放。

[0308] 传输缓冲器 (TB) 4a 是用于在从 ID 过滤器 3b 输出属于主视频流的 TS 分组时暂时在其中对这些 TS 分组进行累积的缓冲器。

[0309] 基本缓冲器 (EB) 4c 是其中存储了处于编码状态的画面 (I 画面、B 画面、和 P 画面) 的缓冲器。

[0310] 解码器 (Dec) 4d 通过在每个预定解码时间周期 (DTS) 中对构成主视频的各个画面进行解码来获得多个帧图像,并将这些帧图像写入到视频平面 4。

[0311] 已解码画面缓冲器 4e 是用于存储从解码器 4d 的解码处理中获得的未压缩画面的缓冲器。

[0312] 重排序缓冲器 4f 是用于将已解码画面的顺序从已解码顺序改变为显示顺序的缓冲器。

[0313] 主视频平面 4g 是用于存储主视频的一个画面的像素数据的存储区域。该像素数据由 16 比特的 YUV 值表示,并且视频平面 4g 中存储了分辨率为 1920x1080 的像素数据。

[0314] 传输缓冲器 (TB) 5a 是在将属于第二视频流的 TS 分组从 PID 过滤器 3b 输出时在其中对这些 TS 分组进行累积的缓冲器。

[0315] 基本缓冲器 (EB) 5c 是其中存储了处于编码状态的画面 (I 画面、B 画面、和 P 画面) 的缓冲器。

[0316] 解码器 (Dec) 5d 通过在每个预定解码时间周期 (DTS) 中对构成第二视频的各个画面进行解码来获得多个帧图像, 并将这些帧图像写入到视频平面 5。

[0317] 已解码画面缓冲器 5e 是用于从解码器 5d 的解码处理中获得的未压缩画面的缓冲器。

[0318] 重排序缓冲器 5f 是用于将已解码画面的顺序从解码顺序改变为显示顺序的缓冲器。

[0319] 第二视频平面 5g 是用于存储第二视频的一个画面的像素数据的存储区域。该像素数据由 16- 比特的 YUV 值表示, 并且视频平面 5g 中存储了分辨率为 1920x1080 的像素数据。

[0320] 缓冲器 6a 以先进先出方式存储了在从去复用器 3a 输出的多个 TS 分组之中的、构成主音频流的 TS 分组, 并将这些 TS 分组提供给音频解码器 7a。

[0321] 缓冲器 6b 以先进先出方式存储了在从去复用器 3b 输出的多个 TS 分组之中的、构成第二音频流的 TS 分组, 并将这些 TS 分组提供给音频解码器 7b。

[0322] 音频解码器 7a 将存储在缓冲器 6a 中的 TS 分组转换为 PES 分组, 解码这些 PES 分组以获得 LPCM 状态的未压缩音频数据, 并输出所获得的音频数据。这就实现了主音频流的数字输出。

[0323] 音频解码器 7b 将存储在缓冲器 6b 中的 TS 分组转换为 PES 分组, 解码这些 PES 分组以获得 LPCM 状态的未压缩音频数据, 并输出所获得的音频数据。这就实现了第二音频流的数字输出。

[0324] 混合器 9a 将从音频解码器 7a 输出的 LPCM 状态的数字音频与从音频解码器 7b 输出的 LPCM 状态的数字音频进行混合。

[0325] 切换器 10a 选择性地将从 BD-ROM 读出的 TS 分组或者从本地存储器 200 读出的 TS 分组提供给第二视频解码器 5d。

[0326] 切换器 10b 选择性地将从 BD-ROM 读出的 TS 分组或者从本地存储器 200 读出的 TS 分组提供给呈现图形解码器 13b。

[0327] 切换器 10c 选择性地将从 BD-ROM 读出的 TS 分组或者从本地存储器 200 读出的 TS 分组提供给交互图形解码器 11b。

[0328] 切换器 10d 是对于是否将由去复用器 3a 去复用的主音频流的 TS 分组与由去复用器 3b 去复用的主音频流的 TS 分组之中的一个提供给音频解码器 8a 进行切换的切换器。

[0329] 切换器 10e 是对于是否将由去复用器 3a 去复用的第二音频流的 TS 分组与由去复用器 3b 去复用的第二音频流的 TS 分组之中的一个提供给音频解码器 8b 进行切换的切换器。

[0330] 传输缓冲器 (TB) 11a 是暂时在其中累积属于 IG 流的 TS 分组的缓冲器。

[0331] 交互图形 (IG) 解码器 11b 对从 BD-ROM 100 或者本地存储器 200 读出的 IG 流进行解码并将未压缩图形写入 IG 平面 12 中。

[0332] 交互图形 (IG) 平面 11c 是其中存储了构成了通过 IG 解码器 11b 的解码所获得的未压缩图形的像素数据的平面。

[0333] 传输缓冲器 (TB) 12a 是在其中暂时累积属于 textST 流的 TS 分组的缓冲器。

[0334] 缓冲器 (TB) 12b 是在其中暂时累积构成 textST 流的 PES 分组的缓冲器。

[0335] 对于从 BD-ROM 100 或者本地存储器 200 读出的 textST 流,基于文本的字幕解码器 12c 使用字符码将所呈现的字幕转换为位图,并将该位图写入 PG 平面 13c 中。由于存储在 BD-ROM 100 或者本地存储器 200 中的字体数据用于转换,对 textST 流进行解码需要预先读取这种字体数据。

[0336] 传输缓冲器 (TB) 13a 是在其中暂时累积属于 PG 流的 TS 分组的缓冲器。

[0337] 呈现图形 (PG) 解码器 13b 对从 BD-ROM 100 或者本地存储器 200 读出的 PG 流进行解码,并将未压缩图形写入呈现图形平面 14。通过 PG 解码器 13b 的解码,在屏幕上显示字幕。

[0338] 呈现图形 (PG) 平面 13c 是具有一个屏幕面积的存储器,其中存储了一个屏幕的未压缩图形。

[0339] 存储器 21 是用于在其中存储当前播放列表信息和当前片段信息的存储器。当前播放列表信息是在存储在 BD-ROM 中的多个播放列表信息之中的当前正在处理的播放列表信息。当前片段信息是在存储在 BD-ROM/本地存储器中的多个片段信息之中的当前正在处理的片段信息。

[0340] 控制器 22 通过执行播放列表的重放来实现 BD-ROM 的重放控制 (即,根据当前播放列表信息进行重放控制)。

[0341] PSR 组 23 是内建在重放设备中的寄存器,其包括 64 个播放器设置 / 状态寄存器 (PSR) 以及 4096 个通用寄存器 (GPR)。在播放器设置 / 状态寄存器 (PSR) 中设置的值 (PSR) 之中,PSR4 到 PSR8 用于表示当前重放点。

[0342] PID 转换单元 24 基于 STN_table 将存储在 PSR 组 23 中的流编号转换为 PID 查询值,并且将转换所得到的 PID 查询值通知给 PID 过滤器 3b 和 3d。

[0343] 网络单元 25 实现重放设备的通信功能。当指定了 URL 时,通信单元 25 建立与该指定 URL 的网站之间的 TCP 连接或者 FTP 连接。这种连接的建立允许从网站进行下载。

[0344] 操作接收单元 26 接收用户在遥控器上进行的操作的指定,并向控制器 22 通知用于指示用户指定的操作的用户操作信息。采用该用户操作信息,指定用户希望选择的流的编号 (编号 x)。

[0345] 这样就结束了对播放设备的内部结构的描述。接下来描述播放设备的输出级的内部结构。图 23 示出了播放设备的输出级的结构。如图所示,播放设备的输出级包括: $1-\alpha^3$ 乘法单元 15a、缩放与定位单元 15b、 α^3 乘法单元 15c、加法单元 15d、 $1-\alpha^1$ 乘法单元 15e、 α^1 乘法单元 15f、加法单元 15g、 $1-\alpha^2$ 乘法单元 15h、 α^2 乘法单元 15i、加法单元 15j、以及 HDMI 发送与接收单元 16。

[0346] $1-\alpha^3$ 乘法单元 15a 将构成存储在视频解码器 4g 中的未压缩数字画面的像素的亮度乘以传递系数 (transmittance) $1-\alpha^3$ 。

[0347] 缩放与定位单元 15b 放大或者缩小 (即,缩放) 存储在视频平面 5g 中的未压缩数字画面,并改变位置 (即定位)。该放大和缩小基于元数据的 PiP_scale,位置的改变基于 PiP_horizontal_position 和 PiP_vertical_position 的。

[0348] α^3 乘法单元 15c 将构成已经由缩放与定位单元 15b 进行了缩放与定位的未压缩画面的像素的亮度乘以传递系数 α^3 。

[0349] 乘法单元 15d 将通过由 α^3 乘法单元 15c 将每个像素的亮度乘以传递系数 α^3 而

创建的未压缩数字画面与由 $1-\alpha 3$ 乘法单元 15a 将每个像素的亮度乘以传递系数 $1-\alpha 3$ 而创建的未压缩数字画面进行组合,从而获得合成画面。

[0350] $1-\alpha 1$ 乘法单元 15e 将构成由乘法单元 15d 所创建的合成数字画面的像素的亮度乘以传递系数 $1-\alpha 1$ 。

[0351] $\alpha 1$ 乘法单元 15f 将构成存储在呈现图形解码器 13c 中的压缩图形的像素的亮度乘以传递系数 $\alpha 1$ 。

[0352] 乘法单元 15g 将通过由 $1-\alpha 1$ 乘法单元 15e 将每个像素的亮度乘以传递系数 $1-\alpha 1$ 而创建的未压缩数字画面与由 $\alpha 1$ 乘法单元 15f 将每个像素的亮度乘以传递系数 $\alpha 1$ 而创建的未压缩图形进行组合,从而获得合成画面。

[0353] $1-\alpha 2$ 乘法单元 15h 将构成由乘法单元 15g 创建的数字画面的像素的亮度乘以传递系数 $1-\alpha 2$ 。

[0354] $\alpha 2$ 乘法单元 15i 将构成存储在交互图形解码器 1c 中的未压缩图形的像素的亮度乘以传递系数 $\alpha 2$ 。

[0355] 乘法单元 15j 将通过由 $1-\alpha 2$ 乘法单元 15h 将每个像素的亮度乘以传递系数 $1-\alpha 2$ 而创建的未压缩数字画面与由 $\alpha 2$ 乘法单元 15i 将每个像素的亮度乘以传递系数 $\alpha 2$ 而创建的未压缩图形进行组合,从而获得合成画面。

[0356] HDMI 发送与接收单元 16 从经由 HDMI (高清晰度多媒体接口) 连接的另一设备接收关于该设备的信息,并将通过乘法单元 15j 的合成而获得的数字未压缩视频与由混合器 9a 所混合的音频数据一起发送给该经由 HDMI 连接的设备。

[0357] 这样就结束了对本实施例的重放设备的硬件结构的描述。接下来描述本实施例的重放设备的软件结构。

[0358] 在图 24 中功能性地示出了图 22 中的控制器 22。图 24 功能性地示出了控制器 22, 如图所示, 控制器 22 包括: 播放列表处理单元 41、过程执行单元 42、PiP 控制单元 43 和混合控制单元 44。

[0359] 这些组件基于 PSR 组 23 来执行处理。以下解释 PSR1、PSR14 和 PSR31。PSR 的字长为 32 比特。构成 PSR 的一个字 (32 个比特) 的各个比特数据的比特位置称为 b0 到 b31。在此, PSR 的最高有效比特称为 b31, 最低有效比特称为 b0。

[0360] <PSR14>

[0361] 图 25A 示出了 PSR14 中的比特分配。

[0362] 如图所示, 在 PSR14 的 32 个比特中的 b8 到 b15 表示第二视频流的流编号, 标识在当前播放项的 STN_table 中写入了其条目的多个第二视频流中的一个。当在 PSR14 中设定的值改变时, 重放设备播放与改变后的设定值相对应的第二视频流。PSR14 的第二视频流的流编号设定为初始值“0xFF”, 然后可以由重放设备设定为范围从“1”到“32”的值。值“0xFF”是未指定值, 指示没有第二视频流或者还没有选择第二视频流。当 PSR14 被设定为范围为“1”到“32”的值时, 该设定值被解释为第二视频流的流编号。

[0363] PSR14 的 b31 是 disp_v_flag, 指示重放设备是否能够播放第二视频流 (1b: 能够进行第二视频的呈现; 0b: 不能进行第二视频的呈现)。在此, HDTV 指的是分辨率为 1920x1080 的视频, 或者分辨率为 1280x720 的视频, 比特 b31 示出了重放设备是否能够解码该视频并输出所解码的视频。

[0364] <PSR29>

[0365] 图 25B 示出了在 PSR29 中的比特分配。

[0366] PSR29 的 b0 是 HD 第二视频能力,指示重放设备是否能够播放 50&25Hz 的视频,即,采用 50 帧 / 秒模式的 PAL 格式视频流或者采用 25 帧 / 秒模式的 PAL 格式视频流 (1b :能够播放 HD 第二视频),或者重放设备是否能够播放 HDTV 第二视频流 (0b :不能播放 HD 第二视频)。

[0367] PSR29 的 b1 是 50&25Hz 视频能力,指示重放设备是否能够播放 50&25Hz 的视频,即, PAL 格式的视频流 (1b :能够播放 50&25Hz 的视频 ;0b :不能播放 50&25Hz 的视频)。

[0368] 这样就结束了对 PSR 组 23 的描述。

[0369] 从现在开始,将描述播放列表处理单元 41、过程执行单元 42、和过程执行单元 43。

[0370] 由于 PiP 控制单元 43 具有各种处理,因此在实施例 2 中给出了描述。混合控制单元 44 与本实施例的要点几乎没有关系,因此在实施例 4 中给出了解释。

[0371] <功能结构细节 1 :播放列表处理单元>

[0372] 播放列表处理单元 41 实现了 PL 重放。播放列表处理单元 41 播放在主视频流中与播放项的从 In_time 到 Out_time 的部分相对应的部分。与此同步地,播放列表处理单元 41 使视频解码器 5b 播放第二视频流中与子播放项信息的从 Sub_PlayItem_In_time 到 Sub_PlayItem_Out_time 的部分相对应的部分。

[0373] <功能结构细节 2 :过程执行单元 42>

[0374] 当需要将一个播放项信息改变到另一个时,或者当用户改变流编号时,过程执行单元 42 执行预定的流选择过程,将新的第二视频流的流编号写入 PSR14。重放设备根据写入到 PSR14 中的流编号来播放第二视频流。因此 PSR14 的设置使得能够选择一个第二视频流。

[0375] 为什么要在改变播放项信息时执行该流选择过程的原因如下。由于 STN_Table 是相对于每个播放项信息提供的,因此可能发生的是,根据一个播放项信息能够播放的第二视频流根据另一个播放项信息不能被播放。

[0376] 过程执行单元 42 在图 26 所示的状态中使得 PSR14 发生改变。图 26 示出了在 PSR14 中的第二视频流编号的状态改变。在该图中,术语“有效”表示 PSR14 的值等于或小于在播放项的 STN_table 中写入的条目数量,并且是可解码的。

[0377] 术语“无效”表示:(a)PSR14 的第二视频流编号为“0”,(b)PSR14 的第二视频流编号大于在播放项的 STN_table 中写入的条目数量,或者(c)即使是在播放项的 STN_table 中写入的条目数量在从“1”到“32”的范围之内时也不能进行解码。

[0378] 图 26 中的虚线框示意性地指示用于在状态改变时确定 PSR 的值的过​​程。用于将第二视频流编号设定给 PSR14 的过程包括“在重放状态改变时的过程”和“当要求流改变时的过程”。

[0379] “在重放状态改变时的过程”是在重放设备的状态由于在重放设备中发生的某些事件而发生了改变时要执行的过程。

[0380] “当要求流改变时的过程”是当用户要求改变流时要执行的过程。

[0381] “在重放状态改变时的过程”和“当要求流改变时的过程”是流选择过程,并且在以下参考流程图进行了详细描述。

[0382] 图 26 中的箭头示意性地示出了在 PSR14 中设定的第二视频流编号的状态改变。

[0383] 附加在箭头上的注释指示触发每个状态改变的事件。这就是说,图 26 指示了如果发生了诸如“加载盘”、“改变流”、“开始播放列表的重放”、“穿过播放项边界”或者“终止播放列表的重放”之类的事件,则发生在 PSR14 中的第二视频流编号的状态改变。通过参考图 26,可以看到:当发生“无效 ->无效”或者“有效 ->无效”的状态改变时不执行上述过程。另一方面,“无效 ->有效”或者“有效 ->有效”的状态改变经过该虚线框。这就是说,当在 PSR14 中的第二视频流编号被设定为“有效”时,执行上述的“在重放状态改变时的过程”或者“当要求流改变时的过程”。

[0384] 以下描述触发状态改变的事件。

[0385] 事件“加载盘”指示 BD-ROM 已经加载到重放设备中。在该加载时,在 PSR14 中的第二视频流编号被设定为未指定值 (0xFF) 一次。

[0386] 事件“开始播放列表的重放”指示基于播放列表的重放处理已经开始。当该事件发生时,执行“在重放状态改变时的过程”,并且将 PSR14 中的第二视频流编号设定为“有效”。

[0387] 事件“终止播放列表的重放”指示基于播放列表的重放处理已经结束。要理解的是,当该事件发生时,并不执行“在重放状态改变时的过程”,并且将 PSR14 中的第二视频流编号设定为“无效”。

[0388] 事件“改变流”指示用户要求改变流。如果在 PSR14 中的第二视频流编号为“无效”时该事件发生(图 26 中用“cj1”指示),则将 PSR14 设定为所要求的值。即使是这样所设定的值指示有效流编号,也会将 PSR14 中设定的值看作是“无效”值。这就是说,在由事件“改变流”所触发的状态改变中,PSR 不会从无效变为有效。

[0389] 另一方面,如果在 PSR14 中的第二视频流编号为“有效”时发生“改变流”的事件(图 26 中用“cj2”指示),则执行“当要求流改变时的过程”,并将 PSR14 设定为新的值。在此,当执行“当要求流改变时的过程”时所设定的值可能不是用户想要的值。这是因为“当要求流改变时的过程”具有排除无效值的功能。如果在 PSR14 为“有效”时发生“改变流”的事件,则 PSR14 不会从有效变为无效。这是因为“当要求流改变时的过程”假设 PSR14 不会变为“无效”。

[0390] 事件“穿过播放项边界”指示已经经过了播放项的边界。在此,播放项边界是两个连续播放项之间的边界,即在两个连续播放项之间的、先前播放项的结尾与后续播放项的开始之间的位置。如果在 PSR14 中的第二视频流编号为“有效”时发生“穿过播放项边界”的事件,则执行“在重放状态改变时的过程”。在执行了“在重放状态改变时的过程”之后,PSR14 的状态或者返回为“有效”,或者变为“无效”。由于 STN_table 是相对于每个播放项提供的,因此如果播放项改变了,则可播放的基本流也改变。状态改变的目的是通过在每次开始播放一个播放项时执行“在重放状态改变时的过程”,来将 PSR14 设定为对于每个播放项而言最适当的值。

[0391] 图 27 是示出第二视频流的“在重放状态改变时的过程”的处理过程的流程图。

[0392] 在步骤 S51 中,检查具有在 PSR14 中所设定的流编号的第二视频流是否满足以下条件 (A) 和 (B):

[0393] 条件 (A):根据视频格式和帧速率与 HD 第二视频能力或者 50&25Hz 视频能力之间比较,重放设备能够播放由存储在 PSR14 中的编号所指定的第二视频流。

[0394] 条件 (B) :第二视频流的 SubPath_Type 为“6(即,异步画中画)”。

[0395] “HD 第二视频能力”指的是对 HDTV 视频流进行解码并将其输出以进行显示的能力,而“50&25Hz 视频能力”指的是能够对 50 帧/秒或者 25 帧/秒画面序列进行解码并将其输出以进行显示的能力。另外,在 STN_table 中,视频格式和帧速率被写入到第二视频流的 stream_attribute 中。在 PSR29 的 b0 和 b1 中示出了是否具有 HD 第二视频能力和 50&25Hz 视频能力。针对在 PSR29 中的 b1 的值来检查在 STN_table 中的这些设置能够确定是否满足条件 (A)。

[0396] 在执行了步骤 S51 之后,执行步骤 S52 和 S53。

[0397] 步骤 S52 是判断步骤,用于判断 STN_table 中的当前播放项的第二视频流的流条目数量是否为 0。在此,STN_table 中的当前播放项的第二视频流的流条目数量为 0 意味着没有允许重放的第二视频流。当该数量为 0 时,则保持在 PSR14 中的第二视频流编号(步骤 S53)。这是因为,只要在当前播放项中没有允许播放的第二视频流,就应该保持 PSR14 的当前值。

[0398] 步骤 S54 是判断步骤,其在步骤 S53 中判定 STN_table 中的第二视频流的流条目数量不为 0 时执行。在步骤 S54 中,判断以下:存储在 PSR14 中的编号 x 是否等于或小于在 STN_table 中的流条目总数量;以及具有编号 x 的第二视频流是否满足条件 (A)。如果在步骤 S54 中为“是”,则通过执行以下所述的如图 28 的流程图中所示的过程来为当前播放项选择最适当的流(步骤 S55)。

[0399] 如果步骤 S54 中为“否”,则执行步骤 S56 的判断步骤。该判断步骤是判断具有编号 x 的第二视频流是否满足条件 (B)。当满足条件 (B) 时,在步骤 S58 中将 PSR14 设定为 0xFE。值 0xFE 指示尽管在 PSR14 中的第二视频流编号是有效的,但是还没有选择第二视频流。如果在执行异步画中画时在 PSR14 中设定该值,则将响应于用户操作,执行“当要求流改变时的过程”。然而,如果 PSR14 的流编号是无效的,则即使是在执行用户操作时也不会执行“当要求流改变时的过程”,并且将不会播放第二视频流。为了避免发生这种情况,设计为在执行异步画中画时在 PSR14 中设定 0xFE。

[0400] 当虽然没有满足条件 (B) 但是已经在 PSR14 中设定了有效的第二视频流编号时,保持该编号(步骤 S57)。

[0401] 这样就结束了对于第二视频流的“在重放状态改变时的过程”的描述。

[0402] 图 28 是示出用于为当前播放项选择适合的流的处理过程的流程图。

[0403] 在步骤 S61 到 S63 中,对在 STN_table 中的所有流条目执行对于是否满足以下条件 (a) 和 (b) 的检查。

[0404] 条件 (a) :根据视频格式和帧速率与 HD 第二视频能力或者 50&25Hz 视频能力之间的比较,重放设备能够播放由存储在 PSR14 中的编号所指定的第二视频流。

[0405] 条件 (b) :第二视频流的 SubPath_Type 为“6(即,异步画中画)”。

[0406] 当对于在 STN_table 中所有允许重放的第二视频流都完成了这些检查时,过程执行单元 42 执行步骤 S64。

[0407] 步骤 S64 是判断步骤,用于判断是否存在满足条件 (a) 的第二视频流。如果在步骤 S64 中为“是”,则在 PSR14 中设定 0xFFF 作为第二视频流的流编号(步骤 S65)。

[0408] 如果存在至少一个满足条件 (a) 的第二视频流,则步骤 S64 为“是”,并且执行步骤

S66。在步骤 S66 中,判断在满足条件 (a) 的第二视频流中,在 STN_table 中按顺序排列的第一个第二视频流是否满足条件 (b)。如果其满足条件 (b),则在步骤 S67 中在 PSR14 中设定 0xFF 作为第二视频流的流编号。

[0409] 如果不满足条件 (b),则从满足条件 (a) 的第二视频流中选出流条目在 STN_table 中排第一的第二视频流,并且将所选择的第二视频流的流编号设定在 PSR14 中 (步骤 S68)。通过该过程,就将对于当前播放项而言最适合的第二视频流存储在 PSR14 中。这样就结束了用于选择适合的流编号的选择过程。

[0410] 图 29 是示出第二视频流的“当要求流改变时的过程”的处理过程的流程图。

[0411] 假设在操作接收单元 26 上进行的用户操作指定了第二视频流编号 x。在该情况下,在步骤 S71 中检查编号 x 的第二视频流是否满足以下条件 (A) 和 (B)。

[0412] 条件 (A):根据视频格式和帧速率与 HD 第二视频能力或者 50&25Hz 视频能力之间的比较,重放设备能够播放由存储在 PSR14 中的编号所指定的第二视频流。

[0413] 条件 (B):第二视频流的 SubPath_Type 为“6(即,异步画中画)”。

[0414] 在完成这些检查之后,执行步骤 S72 的判断步骤。在步骤 S72 中,判断在 STN_table 中的用于当前播放项的第二视频流的流条目数量是否为 0。在步骤 S72 中为“0”意味着不存在允许重放的流,并且保持在 PSR14 中的第二视频流编号 (步骤 S73)。

[0415] 当在步骤 S72 中不为“0”时,执行步骤 S74 的判断步骤。在步骤 S74 中,判断用户操作所指定的第二视频流编号 x 是否等于或小于在 STN_table 中的流条目的总数量以及该编号 x 是否满足条件 (A)。如果其满足条件 (A),则选择编号 x 所指定的流并将其设定在 PSR14 中 (步骤 S75)。

[0416] 当在步骤 S74 中为“否”时,执行步骤 S76 的判断步骤。该判断步骤是判断编号 x 是否为 0xFF。在此,编号 x “0xFF”是“委托编号 (entrustment number)”,其表示将为当前播放项选择适合的第二视频流的选择决定留给重放设备进行判断。如果编号 x 不是 0xFF,则保持存储在 PSR14 中的第二视频流编号 (步骤 S77)。如果编号 x 是 0xFF,则开始从步骤 S78 到 S80 的循环处理。在该循环处理中,对在 STN_table 的流条目中写入的所有第二视频流检查是否满足以下条件。

[0417] 条件 (a):根据采用视频格式和帧速率与 HD 第二视频能力或者 50&25Hz 视频能力之间的比较,重放设备能够播放由存储在 PSR14 中的编号所指定的第二视频流。

[0418] 在执行了该循环处理之后,在步骤 S81 中判断是否存在至少一个满足该条件 (a) 的第二视频流。如果其存在,则从满足条件 (a) 的第二视频流中选出其条目在 STN_table 中排第一的第二视频流,并且将所选择的第二视频流的流编号设定在 PSR14 中 (步骤 S82)。这样就结束了对于第二视频流的“当要求流改变时的过程”的描述。

[0419] 根据本实施例,即使是在记录在 BD-ROM 或者本地存储器中的第二视频流中存在不可播放的第二视频流,也存在一个选择,即,通过执行选择后续流的过程来使用“对于画中画,重放设备能够播放的视频”。因此,即使是第二视频的总尺寸彼此不同,并且重放设备之间在其播放第二视频的能力方面有所不同,也可以使重放设备播放某种类型的第二视频或者另一种第二视频,并执行画中画。

[0420] 实施例 2

[0421] 本发明描述了使重放设备执行画中画的一种具体数据结构,以及根据该数据结构

的重放设备的处理过程。使重放设备执行画中画的具体数据结构存在于图 11 和 16 中所示的 mpls 文件的内部结构中的扩展数据中,并且被称为 PiP 元数据 (PiP_metadata)。图 30 示出了 PiP 元数据的内部结构。引导线 hm1 指示 PiP 元数据的内部结构的展开。如引导线 hm1 所示, PiP 元数据包括 :number_of_metadata_block_entries、n1 个 metadata_block_header、以及 n2 个 PiP_metadata_block。

[0422] 引导线 hm2 指示 metadata_block_header 的内部结构的展开。即,这些 metadata_block_header 是从同一类结构中生成的多个实例,并且其每一个都具有与引导线 hm2 所示相同的内部结构。以下描述了构成 metadata_block_header 的每个字段。

[0423] ref_to_PlayItem_id[k] :

[0424] 这是用于指示要作为画中画目标的 PlayItem[k] 的 PlayItem_id 的字段。由 ref_to_PlayItem_id[k] 所指示的播放项的 STN_table 中必须存储 ref_to_secondary_video_stream_id[k] 的查询值 (secondary_video_stream_id)。另外,在 PiP_metadata() 中的 ref_to_PlayItem_id 必须以查询值的升序进行重新排列。这是为了在当前播放项改变时,迅速地找到具有指定了改变后的播放项的 ref_to_PlayItem_id 的 metadata_block_header。

[0425] ref_to_secondary_video_stream_id[k] :

[0426] 这是用于示出在由 ref_to_PlayItem_id[k] 所指的播放项的 STN_table 中所定义的多个 secondary_video_stream_id 中的、指示用于画中画重放的第二视频流的字段。该字段用于指定使用相关的 PiP_metadata_block[k] () 的第二视频流。更具体而言,通过参考 secondary_video_stream_id 能够查询到在 STN_table 中的 stream_entry(), 并且通过参考该流条目能够得到与 PiP_metadata_block[k] () 所使用的第二视频流相对应的子路径。

[0427] 注意,在此,在 pip_metadata() 中必须不能存在两个或更多的这样的对 :该对的 ref_to_PlayItem_id[k] 与 ref_to_secondary_video_stream_id[k] 的值相同。

[0428] pip_timeline_type[k] :

[0429] 这指示 Sync_Start_PTS_of_PlayItem 在播放项时间轴上的映射点应该用作执行画中画的参考时刻,或者用作 SubPlayItem_In_time 的位置。

[0430] 当 Pip_time_line_type = 1 时, pip_metadata_time_stamp 使用 由 ref_to_PlayItem_id[k] 所指的播放项时间轴作为执行同步画中画时的基础。在该情况中,由 ref_to_secondary_video_stream_id[k] 所指定的子路径的 SubPath_type 必须被设定为 5 或者 7。

[0431] 当 Pip_time_line_type = 1 时, pip_metadata_time_stamp 使用子路径时间轴作为执行异步画中画时的基础。在该情况中,由 ref_to_secondary_video_stream_id[k] 所指定的子路径的 SubPath_type 必须被设定为 6。

[0432] 当 Pip_time_line_type = 3 时, pip_metadata_time_stamp 使用 由 ref_to_PlayItem_id[k] 所指的播放项时间轴作为执行异步画中画时的基础。在该情况下,由 ref_to_secondary_video_stream_id[k] 所指定的子路径的 SubPath_type 必须被设定为 6。

[0433] 因此,通过理想地使用播放项侧或者子播放项侧中的任何一个作为基础,能够实现优选的画中画重放。

[0434] `is_luma_key` :

[0435] 当该标志为 1 时,根据 `upper_limit_luma_key` 而将 `luma_keying` 应用于相应的第二视频流。`luma_keying` 是这样的处理:当构成第二视频流的每个画面包含物体和背景时,从该画面中提取出物体,并提供该物体用于与主视频进行合成。当将 `luma_keying` 应用于第二视频时,在实施例 1 中所示出的 α_3 乘法单元 15c 的 α_3 设定为 0 或 1。当不采用 `luma_keying` 时, α_3 设定为 1。

[0436] `trick_playing_flag` :

[0437] 这是示出内容提供商对于在主视频特技重放 (`trick playback`) 期间画中画的窗口是保持打开还是关闭的意图。该标志仅仅在同步画中画时变为有效。通过将标志设定为 0,内容提供商示出了使得在主视频特技重放期间画中画的窗口 (PiP 窗口) 关闭的意图。在此,特技重放是由重放设备制造商针对每个重放设备定义的。例如,在一些重放设备中,1.2x 前进播放可以定义为不是特技播放,而 2x 后退播放或者更高的情况可以定位是特技重放。

[0438] `upper_limit_luma_key` :

[0439] 这是用于为 `luma_keying` 定义相应第二流的亮度 (Y) 上限的字段。当将主视频平面和第二视频平面相互置顶放置时,在画中画合成时对在第二视频平面上亮度 (Y) 值为 0 或更大但不超过 `upper_limit_luma_key` 的像素实现了完整传输。例如,当第二视频由人的图像和背景组成,并且希望仅仅使用该人的图像作为合成目标时,在 `upper_limit_luma_key` 中设定一个亮度值,使得在构成第二视频的多个像素中具有该亮度值或者更小亮度值的像素被看作是背景。

[0440] 接下来描述 `PiP_metadata_block` 的内部结构。引导线 hm2 指示 `PiP_metadata_block` 的结构展开。如该引导线所示,`PiP_metadata_block[1]` 包括:k 个 `PiP_metadata_entry[1]` 到 [k],以及 `number_of_pipmetadata_entries`。

[0441] 引导线 hm3 指示 `PiP_metadata_entry` 的内部结构。即,这些 `PiP_metadata_entry` 是从同一类结构生成的多个实例,并且其每一个都具有相同的内部结构,并且包括:`pip_metadata_time_stamp[i]` 和 `pip_composition_metadata()`。

[0442] `pip_metadata_time_stamp[i]` :

[0443] 这是用于指示在其中 `pip_composition_metadata()` 为有效的时间间隔的开始点。

[0444] 除了最后一个 `pip_composition_metadata()` 之外,在第 k 个 `PiP_metadata_block[k]` 中的第 i 个 `pip_composition_metadata()` 在不小于 `pip_metadata_time_stamp[i]` 且不大于 `pip_metadata_time_stamp[i+1]` 的时间间隔内变为有效。在 `PiP_metadata_block[k]()` 中的最后一个 `pip_metadata_time_stamp` 的最后一个 `pip_composition_metadata()`,在不小于最后一个 `pip_metadata_time_stamp` 且不大于由 `ref_to_secondary_video_stream_id[k]` 所指定的子路径的显示结束时间的间隔内变为有效。另外,在两个相邻 `pip_metadata_time_stamp` 之间的最小时间间隔为小于一秒。

[0445] `pip_composition_metadata()` 包括以下字段:

[0446] `pip_horizontal_position[i]` :

[0447] 该字段指示第二视频流左上角像素在主视频平面上的水平位置。当用 `video_`

width 表示视频平面的水平宽度时,由 pip_horizontal_position 指定的水平位置的范围从 0 到 video_width-1。

[0448] pip_vertical_postion[i] :

[0449] 该字段指示第二视频流左上角像素在主视频平面上的垂直位置。当用 video_Height 表示视频平面的垂直宽度时,由 pip_vertical_postion 指定的垂直位置的范围从 0 到 video_height-1。图 31 示出了 pip_horizontal_position 和 pip_vertical_postion 在视频平面上采用的坐标。该图示出了一个坐标系,其中,图像的左上角为原点,x 轴和 y 轴的正方向在图中分别为向右和向下。在该坐标系中,PiP_horizontal_position 表示 x 坐标,PiP_vertical_postion 表示 y 坐标。

[0450] pip_scale[i] :

[0451] 这是用于指示第二视频的缩放类型的字段。缩放类型如下 :

[0452] 0 :重置

[0453] 1 :不缩放 (x1)

[0454] 2 :1/2 缩放 (x1/2)

[0455] 3 :1/4 缩放 (x1/4)

[0456] 4 :1.5x 缩放 (x1.5)

[0457] 5 :全屏缩放

[0458] 以下描述在 pip_timeline_type 和 pip_metadata_time_stamp 之间的关系。

[0459] pip_timeline_type[k] = 1

[0460] 在该情况下,对于从 SubPlayItem_In_Time 在由 ref_to_PlayItem_id[k] 所指的播放项时间轴上的映射点开始并且对应于子播放项的时间长度的时间长度,画中画变为有效。在情况下,PiP_metadata_time_stamp 以 45KHz 的时间精度表示在画中画的有效时间段内的任何一个时间点。在此,将 pip_metadata_time_stamp[0] 放置在通过将 SubPlayItem_In_Time 映射到由 ref_to_PlayItem_id[k] 所指的播放项时间轴上所得到的映射点上。

[0461] 图 32 示出了当 PiP_time_line_type = 1 时,如何将 PiP_metadata_time_stamp 放置在该时间轴上。在该图中,层次 3 示出了子播放项时间轴,层次 2 示出了播放项时间轴。层次 1 示出了构成主视频的多个画面。

[0462] 在此假设第二视频是购物视频并且提供对在主电影中出现的演员所穿的衣服的销售。在此,屏幕的左侧是在播放项时间轴上的时间点 t1 处是空白的,屏幕的右侧在时间点 t2 处是空白的。在此情况下,将 PiP_metadata_time_stamp[i] 设定为 t1,并且将 PiP_horizontal_position[i] 和 PiP_vertical_postion[i] 设定为屏幕的左侧。另外,将 PiP_metadata_time_stamp[i+1] 设定为 t2,并且将 PiP_horizontal_position[i+1] 和 PiP_vertical_postion[i+1] 设定为屏幕的右侧。

[0463] 以此,将在时间点 t1 和 t2 处在空白中显示购物视频。当 pip_time_line_type = 1 时,能够通过将第二视频显示在其不会打断主电影的观看的位置处来按照主电影的画面放置第二视频。

[0464] pip_timeline_type[k] = 2

[0465] 图 33 示出了在 PiP_time_line_type = 2 时如何将 PiP_metadata_time_stamp 放置在时间轴上。图中的层次 1、2 和 3 与图 32 中的相同。

[0466] 对于从图中层次 3 子路径时间轴上的 SubPlayItem_In_Time 开始并且对应于子播放项的时间长度的时间长度内,画中画变为有效。PiP_metadata_time_stamp 以 45KHz 的时间精度表示在画中画的有效时间段内的任何一个时间点。在此,将 pip_metadata_time_stamp[0] 放置在子播放项时间轴上由 In_Time 所指定的时间点处。

[0467] 将 PiP_metadata_time_stamp[0] 放置在子路径时间轴上的原因是:当在第二视频中绘制的事物是希望从屏幕左侧移动到右侧的运动物体时,定义运动轨迹。这是因为这种运动轨迹能够与主电影的画面无关地进行定义。因此,将 PiP_metadata_time_stamp[0] 放置在子路径信息的重放轴上。

[0468] 在图 33 中,假设 PiP_metadata_time_stamp[i] 指示在子播放项时间轴上的 t1,并且 PiP_horizontal_position[i] 和 PiP_vertical_postion[i] 指示屏幕的左上部分。另外, PiP_metadata_time_stamp[i+1] 指示在子播放项时间轴上的 t2,并且 PiP_horizontal_position[i+1] 和 PiP_vertical_postion[i+1] 指示屏幕的右下部分。

[0469] 在该情况下,在子播放项时间轴上由 PiP_metadata_time_stamp[i] 所指示的时刻处,在屏幕的左上部分处显示第二视频。然后在子播放项时间轴上由 PiP_metadata_time_stamp[i+1] 所指示的时刻处,在屏幕的右下部分处显示第二视频。在此,在 Sync_Start_PTS_of_PlayItem 为 0xFF 的情况下,将 Sync_Start_PTS_of_PlayItem 理解为是在用户执行锁定操作时的点。因此,如果用户执行锁定操作,则将顺序地显示第二视频。以此,能够实现这样的屏幕效果:在第二视频中绘制的物体(图中的流星)响应于用户的操作而在主视频中四处移动。

[0470] pip_timeline_type[k] = 3

[0471] 在由 ref_to_PlayItem_id[k] 所指的播放项时间轴上从 PlayItem_In_Time 处开始并在 PlayItem_Out_Time 处结束的时间段内,画中画变为有效。PiP_metadata_time_stamp 以 45KHz 的时间精度表示在画中画的执行时间段内的任何一个时间点。在此,将 pip_metadata_time_stamp[0] 放置在由 ref_to_PlayItem_id[k] 所指的 PlayItem_In_Time 处。

[0472] 图 34 示出了当 PiP_time_line_type = 3 时,如何将 PiP_metadata_time_stamp 放置在时间轴上。在图中的每个层次都与图 32 中的相同。另外,在主视频中的空白位置与图 32 中的相同。当 pip_timeline_type[k] = 3 时,将 PiP_metadata_time_stamp[0] 放置在由播放项时间轴原点所指示的时间点处,即由 In_Time 所指示的时间点。因此,可以通过使用基于播放项时间轴原点的坐标表示 t1 并将 t1 设定给 PiP_metadata_time_stamp[i],同时使用基于播放项时间轴原点的坐标表示 t2 并将 t2 设定给 PiP_metadata_time_stamp[i],来实现与图 32 相同的屏幕效果。如果 PiP_time_line_type 设定为 3,则由基于播放项时间轴原点的坐标来定义应该执行画中画的时间段。

[0473] 这样就结束了对本实施例的记录介质的改进。接下来描述在本实施例的重放设备中的改进。在本实施例的重放设备中的改进是:PiP 控制单元 43 基于 metadata_block_header 和 PiP_metadata_block 执行重放控制。图 35 和 36 是示出基于画中画的重放控制过程的流程图。以下参考这些流程图描述重放控制。

[0474] 步骤 S91 形成了一个事件等待循环,其等待选择过程的执行。当执行该过程时,步骤 S91 为“是”,并且将执行从步骤 S92 开始的执行。在步骤 S92 中,判断是否存在将当前播放项指定为 ref_to_PlayItem_id 并且将与在 PSR14 中的第二视频流编号相对应的 PID 指

定为 Ref_to_secondary_video_stream_id 的 metadata_block_header。

[0475] 如果存在, 则将该 metadata_block_header 设定为当前的 metadata_block_header (步骤 S93)。步骤 S94 到 S96 用于根据 metadata_block_header 的 PiP_time_line_type 的值, 来选择性地执行步骤 S97、S98 和 S99。当 metadata_block_header 的 PiP_time_line_type 为 1 时, 步骤 S94 为“是”, 并且将 PiP_metadata_block.PiP_metadata_time_stamp[0] 放置在 Sync_Start_PTS_of_PlayItem 在播放列表重放时间轴上的映射点处 (步骤 S97)。

[0476] 当 metadata_block_header 的 PiP_time_line_type 为 2 时, 步骤 S95 为“是”, 将 PiP_metadata_block.PiP_metadata_time_stamp[0] 放置在子播放项时间轴上的 SubPlayItem.In_Time 处。(步骤 S98)

[0477] 当 metadata_block_header 的 PiP_time_line_type 为 3 时, 步骤 S96 为“是”, 将 PiP_metadata_block.PiP_metadata_time_stamp[0] 放置在 PlayItem.In_Time 处 (步骤 S99)。

[0478] 在步骤 S100 中, 基于 PiP_metadata_time_stamp[0] 的位置将 PiP_metadata_time_stamp[0][1]... [n] 转换为 STC 时间轴上的坐标。随后, 开始步骤 S101 到 S105 的循环处理。在该循环处理中的计数器变量是变量 i, 其在步骤 S101 中被初始化为“0”。在每次执行步骤 S103 和 S104 时, 在步骤 S105 中将计数器变量累加 1。

[0479] 步骤 S103 用于判断当前重放时间点 (当前呈现时间 (PTM)) 是否已经到达了 PiP_metadata_block.PiP_metadata_time_stamp[i]。当当前 PTM 到达该点时, 命令 α 3 乘法单元 15c 和加法单元 15d 在由 (PiP_horizontal_position 和 PiP_vertical_position) 所指示的坐标处、以由 PiP_scale[i] 所指示的尺寸、显示由 metadata_block_header.Ref_to_secondary_video_stream_id 所指示的第二视频 (步骤 S104)。

[0480] 在此, 如果 metadata_block_header 的 is_luma_key 设定为 1, 则命令 α 3 乘法单元 15c 和加法单元 15d 使得亮度 (Y) 值为 0 或更大但是不超过 upper_limit_luma_key 的像素在画中画的合成时具有完整的传输。以此, 能够从第二视频中仅仅提取出人的图像, 并将所提取的图像与主视频进行组合。

[0481] 结束该循环处理的必要条件是在步骤 S102 中获得 $i = \text{number_of_pip_metadata_entries} - 1$ 。除非满足该必要条件, 否则步骤 S103 到 S105 的处理将会重复进行。

[0482] 当要在播放列表信息的中间开始特殊重放并且当前执行的画中画是同步 PiP 时, 检查内容提供商对于在主视频的特殊重放过程中 PiP 的窗口是否保持打开还是关闭的意图。当该标志为 0 时, 内容提供商意欲关闭 PiP 窗口, 从而该窗口被关闭。当该标志为 1 时, 内容提供商意欲保持 PiP 窗口打开, 该窗口保持打开。

[0483] 因此, 根据本实施例, 基于 PiP_time_line_type 判断主路径时间轴和子路径时间轴中哪一个要用作开始画中画的参考。因此, 可以在重放设备的操作上, 反应创作开发者对于是按照主视频的画面来决定第二视频位置还是响应于用户操作而移动第二视频的意图。

[0484] 实施例 3

[0485] 本实施例涉及的改进是: 多个第二视频流与子播放项一对一地相关联, 并且根据子播放项来定义一个画中画重放路径。

[0486] 在该情况下, 一个子播放项的重放部分的长度可以比相应播放项的重放部分的长

度短。因此,在子路径中的连续的子播放项在时间上不能交叠。即使是在播放项之间存在边界,子播放项也不能交叠。另一方面,允许子路径中的连续的子播放项之间具有时间间隙。这种时间间隙是 3 秒或者更长。

[0487] 图 37 示出了由一个播放项组成的主路径与由两个子播放项组成的子路径之间的关系。层次 1 示出了构成主视频流的多个视频呈现单元。层次 2 示出了多个组,其每一个都包括 sync_PlayItem_id 和 sync_start_PTS_of_PlayItem。层次 3 示出了子片段时间轴,层次 4 示出了子播放项的 In_Time 和 Out_Time。层次 5 示出了构成第二视频流的多个视频呈现单元。

[0488] 层次 2 中的多组 sync_PlayItem_id 与 sync_start_PTS_of_PlayItem 指定了 SubPlayItem_IN_times 在主路径时间轴上进行同步的时刻。

[0489] 在层次 5 中的由 SubPlayItem_IN_times 所指示的第二视频的视频呈现单元 vp1 和 vp2,与由 sync_start_PTS_of_PlayItem 所指示的主视频的视频呈现单元 vp3 和 vp4 分别位于主路径时间轴上的相同的时间点上。

[0490] 在层次 3 中的子路径时间轴上定义了两个子播放项(子播放项 #1 和 #2)。在此,在子路径时间轴上,在子播放项 #1 的 Out_Time 与子播放项 #2 的 In_Time 之间可能存在时间间隙(在图中所示的时间间隙)。如果在一个子路径中的连续的子播放项无时间间隙地相互连接,则子播放项的连接状态必须用 sp_connection_condition 来表示。

[0491] 在图 37 中,在两个或更多个的如层次 3 中所示的子播放项之间具有 3 秒的时间间隙,因此不会将过多的用于控制第二视频流的负载加到控制器 22 上。已经为子路径时间轴进行了这种考虑,可以基于实现基于两个连续的子播放项执行子片段重放且不会增加重放设备的操作时钟。

[0492] 这样就结束了在本实施例的记录介质中的改进。接下来描述在本实施例的重放设备中的改进。

[0493] 本实施例的重放设备中的改进是播放列表处理单元 41 执行如图 38、39 和 40 所示的流程图的处理过程。

[0494] 图 38 是示出基于播放列表信息的重放过程的流程图。该流程图示出了一个循环结构,在该结构中,读入构建该播放列表信息的 .mpls 文件(步骤 S11),将在播放列表信息开始处的播放项设定为当前播放项(步骤 S12),并且对当前播放项重复步骤 S13 到 S25。该循环结构以步骤 S23 作为结束条件。播放列表处理单元 41 命令 BD-ROM 驱动器读取与当前播放项的 In_Time 到 Out_Time 相对应的访问单元(步骤 S13),判断在当前播放项中是否存在先前播放项(步骤 S14),并根据该判断结果选择性地执行步骤 S15 或者步骤 S16 到 S21 的处理。更具体而言,如果当前播放项不具有先前播放项(步骤 S14:否),则播放列表处理单元 41 命令解码器执行从 PlayItem_In_Time 到 PlayItem_Out_Time 的重放(步骤 S15)。

[0495] 如果当前播放项具有先前播放项(步骤 S14:是),则播放列表处理单元 41 判断在当前播放项的连接状态字段中的值是否为 5(cc = 5)(步骤 S16)。当 cc = 5 时(步骤 S16:是),播放列表处理单元 41 执行步骤 S17 到 S20 的处理。

[0496] 当以上的先前播放项存在时,切换主视频中的 ATC 序列。为了该 ATC 序列的切换,播放列表处理单元 41 计算主片段的偏移值,其被称为 ATC_delta1(步骤 S17),并通过将该 ATC_delta1 加到原始 ATC 序列的 ATC 值(ATC1)上来获得新的 ATC 序列的 ATC 值(ATC2)

(步骤 S18)。

[0497] 另外,当以上的先前播放项存在时,切换主视频中的 STC 序列。为了该 STC 序列的切换,播放列表处理单元 41 计算被称为 STC_delta1 的偏移值(步骤 S19),并通过将该 STC_delta1 加到原始 STC 序列的 STC 值上来获得新的 STC 序列的 STC 值(STC2)(步骤 S20)。

[0498] 在播放列表处理单元 41 命令音频解码器 9 消除音频重叠(audioOverlap)之后,命令解码器执行从 PlayItem_In_Time 到 PlayItem_Out_Time 的重放(步骤 S21)。在当前播放项不是 cc = 5 时,播放列表处理单元 41 执行 cc = 1 和 cc = 6 的处理。

[0499] 在执行了步骤 S15 的处理和步骤 S16 到 S21 的处理之中的任何一个处理之后,播放列表处理单元 41 执行步骤 S22 和 S23。

[0500] 在步骤 S22,播放列表处理单元 41 判断是否存在与当前播放项同步的子播放项且当前重放点(当前 PTM(呈现时间))已经到达该子播放项与下一个子播放项之间的边界。如果步骤 S22 为“是”,则播放列表处理单元 41 执行图 39 中流程图的步骤 S30。

[0501] 在步骤 S23 中,播放列表处理单元 41 判断在 AV 片段时间轴上的当前 PTM 是否已经到达当前播放项的 Out_Time。

[0502] 如果步骤 S23 为“是”,则播放列表处理单元 41 移动到步骤 S24。在步骤 S24 中,播放列表处理单元 41 判断当前播放项是否是播放列表信息中的最后一个播放项。如果不是最后一个播放项,则播放列表处理单元 41 使播放列表信息中的下一个播放项成为当前播放项(步骤 S25),并经由步骤 S26 移动到步骤 S13。根据上述处理,对播放列表信息中的所有播放项都执行步骤 S13 到 S24 的处理。

[0503] 图 39 是示出子播放项的无缝连接的处理过程的流程图。

[0504] 当在步骤 S26 中按照播放项的切换选择了一个新的第二视频流时,播放列表处理单元 41 在步骤 S30 中选择与该新的第二视频流相对应的子播放项。

[0505] 随后,在步骤 S31 中,播放列表处理单元 41 将在步骤 S30 中所选择的子播放项设定为当前子播放项。

[0506] 然后,播放列表处理单元 41 对当前子播放项执行步骤 S32 到 S41 的处理。首先,播放列表处理单元 41 命令本地存储器 200 读取与子播放项的 In_Time 到 Out_Time 相对应的访问单元(步骤 S32)。然后播放列表处理单元 41 判断当前播放项是否具有先前子播放项(步骤 S33),并基于该判断结果选择性地执行步骤 S34 和 S35 的处理或者步骤 S36 到 S41 的处理。更具体而言,如果当前播放项不具有先前子播放项(步骤 S33:否),则播放列表处理单元 41 进行等待,直到当前 PTM 已经到达 Sync_Start_Pts_of_PlayItem(步骤 S34)。当其已经到达 Sync_Start_Pts_of_PlayItem 时,播放列表处理单元 41 命令解码器执行从 SubPlayItem_In_Time 到 SubPlayItem_Out_Time 的重放(步骤 S35)。

[0507] 如果当前播放项具有先前子播放项(步骤 S33:是),则播放列表处理单元 41 判断是否当前播放项为 SP_CC = 5(步骤 S36)。当 SP_CC = 5(步骤 S36:是)时,播放列表处理单元 41 执行步骤 S37 到 S41。

[0508] 在当前播放项具有先前子播放项时,切换 ATC 序列。为了该 ATC 序列的切换,播放列表处理单元 41 计算主视频的偏移值,其被称为 ATC_delta2(步骤 S37),并通过将该 ATC_delta1 加到原始 ATC 序列的 ATC 值(ATC1)上来获得新的 ATC 序列的 ATC 值(ATC2)(步骤 S38)。

[0509] ATC_delta 指的是一个偏移值,该偏移值表示从最初已经读出的传输流 (TS1) 的最后一个 TS 分组的输入时间点 T1 到最新读出的传输流 (TS2) 的最后一个 TS 分组的输入时间点 T2 的偏移量。ATC_delta 满足“ $ATC_delta \geq N1/TS_recording_rate$ ”,其中,N1 是随着 TS1 的最后一个视频分组之后的 TS 分组数量。

[0510] 另外,当存在以上的先前播放项时,切换 STC 序列。为了该 STC 序列的切换,播放列表处理单元 41 计算 STC_delta2(步骤 S39),并通过将该 STC_delta2 加到原始 STC 序列的 STC 值上来获得新的 STC 序列的 STC 值 (STC2) (步骤 S40)。

[0511] 假设在该先前 STC 序列中最后播放的画面的显示开始时间为 PTS1(1stEND),该画面的显示时间段为 TPP,并且在随后的 STC 序列中最初显示的画面的开始时间为 PTS2(2ndSTART)。在此,对于 $cc = 5$,由于必须将 PTS1(1stEND)+TPP 的时刻与 PTS2(2ndSTART) 的时刻进行匹配,因此通过以下公式能够计算 STC_delta2:

[0512] $STC_delta2 = PTS1(1stEND) + TPP - PTS2(2ndSTART)$

[0513] 在播放列表处理单元 41 命令音频解码器 9 消除音频重叠之后,命令解码器执行从 PlayItem_In_Time 到 PlayItem_Out_Time 的重放 (步骤 S41)。

[0514] 控制器 22 执行如上所述的 STC 切换处理,并且该处理是在解码器处于自由运行状态时采用普通实现在重放设备中执行的。自由运行状态指的是解码器不执行同步控制的状态。随后,当 STC 返回到能够设置 STC 时间轴的状态时,解码器从自由运行状态转换到进行与 STC 的同步控制。另一方面,当在步骤 S36 中判定当前播放项不是 $cc = 5$ 时 (步骤 S36: 否),播放列表处理单元 41 执行 $cc = 1$ 和 $cc = 6$ 的处理。

[0515] 图 38 的步骤 S26 是判断步骤,用于判断是否按照播放项的切换选择了新的第二视频流。当步骤 S26 为“是”时,播放列表处理单元 41 执行图 40 的步骤 S45 到 S47。

[0516] 在步骤 S45 到 S47 中,重放从一个播放项中两个连续子播放项中的一个子播放项切换到另一个子播放项,并且播放列表处理单元 41 在该切换之后将该子播放项设定为当前子播放项 (步骤 S45)。

[0517] 然后,播放列表处理单元 41 命令本地存储器 200 读取与当前子播放项的 In_Time 到 Out_Time 相对应的访问单元 (步骤 S46),并且命令解码器执行当前 SubPlayItem_In_Time 到当前 SubPlayItem_Out_Time 的重放 (步骤 S47)。

[0518] 因此,根据本实施例,在一个播放项中的当前子播放项与先前子播放项之间提供了时间间隙,因此在子播放项之间的边界不需要步骤 S37 到 S41 的无缝连接处理过程。从而,能够减少要执行的无缝连接过程的数量,这就消除了以高时钟频率操作重放设备的需求,并导致能够以低成本制造重放设备。

[0519] 实施例 4

[0520] 本实施例涉及如何实现画中画中的音频重放。音频重放是在播放列表中的 STN_table 中指定的。以下描述在 STN_table 中的用于音频的流条目。将以上所述的主音频流和第二音频流分别分配给主视频流和第二视频流。即,将主音频流用作主视频流的音频源,将第二音频流用作第二视频的音频源。在 STN_table 中的流条目定义了允许重放用于主视频流和第二视频流的音频中的哪一个。

[0521] 图 41 示出了 Primary_audio_stream_entry 和 Secondary_audio_stream_entry 的内部结构以及 Comb_info_Secondary_video_Secondary_audio 的内部结构。图中的引导

线 ha1 和 ha2 指示 Primary_audio_stream_entry 和 Secondary_audio_stream_entry 的内部结构的展开。这些内部结构是相同的并且都包括 stream_entry(流条目)和 stream_attribute(流属性)。

[0522] 引导线 ha3 示出了 stream_entry 的内部结构的展开。如该引导线所示,主/第二音频流的 stream_entry 包括“ref_to_Sub_Path_id”,指示关于第二音频流的子路径信息;“ref_to_Sub_Clip_entry_id”,指定主/第二音频流要被多路复用到其中的子片段;以及“ref_to_stream_PID_of_Sub_Clip”,指示构成主/第二音频流的 PES 分组的 PID 的查询值。

[0523] 引导线 ha1 指示 stream_attribute 的内部结构的展开。如该引导线所示,主/第二音频流的 stream_attribute 包括:“stream_coding_type”,指示音频流的编码系统;“audio_presentation_type”,指示相应音频流的声道结构;“Sampling_frequency”,指示相应音频流的采样频率;以及“audio_language_code”,指示音频流的语言属性。

[0524] 如引导线 ha5 所示,第二视频流的 stream_entry 具有 Comb_info_Secondary_video_Secondary_audio,作为与第二音频流相对应的信息。

[0525] 引导线 ha6 示出 Comb_info_Secondary_video_Secondary_audio 的内部结构的展开。Comb_info_Secondary_video_Secondary_audio 包括:“number_of_Secondary_audio_stream_ref_entries”,指示每一个都能与第二视频流进行组合的第二音频流的总数量;以及“Secondary_audio_stream_id_ref[0]到[n]”,指示要被组合用于第二视频流的重放的第二音频流的流编号。

[0526] 因此,能够看到,STN_table 包括用于作为主视频流的音频的主音频流的流条目和用于作为第二视频流的音频的第二音频流的流条目。另外,在第二视频流的流条目中,相对于每个第二视频流描述都要与第二视频流进行组合的每个第二音频流(Comb_info_Secondary_video_Secondary_audio)。

[0527] 这样就结束了记录介质的用于音频流重放的特征的描述。以下描述重放设备的用于音频流重放的特征。

[0528] 对于音频重放,在重放设备中,将当前主音频流的流编号和当前第二音频流的流编号分别存储在 PSR 组 23 中。

[0529] <PSR1>

[0530] 图 42A 示出了在 PSR1 中的比特分配。

[0531] 如图 42A 所示,在 PSR1 的 32 个比特中的低 8 个比特(b0-b7)表示流编号,标识在当前播放项的 STN_table 中写入了其条目的多个主音频流之一。当在 PSR1 中设定的值改变时,重放设备播放与改变后的值相对应的主音频流。PSR1 被设定为初始值“0xFF”,然后可以由重放设备设定为范围从“1”到“32”的值。值“0xFF”是未指定值,指示没有主音频流或者还没有选择主音频流。当 PSR1 被设定为范围为“1”到“32”的值时,该设定值被解释为主音频流的流编号。

[0532] <PSR14>

[0533] 图 42B 示出了 PSR14 中的比特分配。

[0534] 如图 42B 所示,在 PSR14 的 32 个比特中低 8 个比特(b0-b7)表示流编号,标识在当前播放项的 STN_table 中写入了其条目的多个第二音频流中的一个。当在 PSR14 中设定

的值改变时,重放设备播放与改变后的设定值相对应的第二音频流。PSR14 被设定为初始值“0xFF”,然后可以由重放设备设定为范围从“1”到“32”的值。值“0xFF”是未指定值,指示没有第二音频流或者还没有选择第二音频流。当 PSR14 被设定为范围为“1”到“32”的值时,该设定值被解释为第二音频流的流编号。

[0535] 由过程执行单元 42 对这些第二音频流编号进行设定和更新。在 PSR14 中的第二音频流编号示出了由过程执行单元 42 进行的、如图 43 所示的状态转换。图 43 示出了在 PSR14 中的第二音频流编号的状态转换。在该图中,术语“有效”表示 PSR14 的值等于或小于在播放项的 STN_table 中写入的条目数量,是可编码的。

[0536] 术语“无效”表示:(a)PSR14 的第二音频流编号为“0”,(b)PSR14 的第二音频流编号大于在播放项的 STN_table 中写入的条目数量,或者(c)即使是在播放项的 STN_table 中写入的条目数量在从“1”到“32”的范围之内时也不能进行解码。

[0537] 图 43 中的虚线框示意性地指示用于在状态改变时确定 PSR 的值的過程。用于设定 PSR 的过程包括“在重放状态改变时的过程”和“当要求流改变时的过程”。这些过程与用于实施例 1 中的第二视频流的过程相同。

[0538] 状态转换由诸如“加载盘”、“改变流”、“开始播放列表的重放”、“穿过播放项边界”或者“终止播放列表的重放”之类的事件触发。这些事件也与用于实施例 1 中的第二视频流的事件相同。

[0539] 以下描述第二音频流的“在重放状态改变时的过程”的过程。

[0540] 图 44 是示出了第二音频流的“在重放状态改变时的过程”的处理过程的流程图。在步骤 S111 中,过程执行单元 42 从 PSR14 获得第二音频流编号,并且在步骤 S112 中判断所获得编号的第二音频流是否满足以下条件(A)。

[0541] 条件(A):根据在 Comb_info_Secondary_video_Secondary_audio 中所指定的第二音频流编号与从 PSR14 获取的第二音频流编号之间的比较,允许将具有该编号的第二音频流与当前第二视频流进行组合。

[0542] 在该判断之后,过程执行单元 42 执行步骤 S113。步骤 S113 是判断步骤,用于判断当前播放项的第二音频流的流条目数量是否为 0。当在 STN_table 中的流条目数量为 0 时,过程执行单元 42 保持在 PSR14 中的第二音频流编号(步骤 S116)。如果在 STN_table 中的流条目数量不是 0,则过程执行单元 42 执行步骤 S114 的判断。

[0543] 在步骤 S114 中,过程执行单元 42 判断该第二音频流编号是否等于或者小于在当前播放项的 STN_table 中的流条目数量,以及具有该编号的第二音频流是否满足条件(A)。如果在步骤 S114 中为“否”,则认为在 PSR14 中存在有效第二音频流的编号,并且保持该编号(步骤 S117)。如果在步骤 S114 中为“是”,则过程执行单元 42 为当前播放项选择适合的第二音频流(步骤 S115)。

[0544] 这样就结束了对第二音频流的“在重放状态改变时的过程”的描述。

[0545] 图 45 是示出了用于为当前播放项选择适合的第二音频流的过程的流程图。

[0546] 在步骤 S121 到 S123 中,过程执行单元 42 检查在 STN_table 的流条目中所写入的所有流是否满足以下条件。

[0547] 条件(a):根据在 Comb_info_Secondary_video_Secondary_audio 中所指定的第二音频流编号与第二音频流编号 i 之间的比较,允许将第二音频流 i 与当前第二视频流进

行组合。

[0548] 条件 (b) : 当前主音频流的 audio_language_code 与第二音频流的 audio_language_code 匹配。

[0549] 当对在 STN_table 中的所有允许重放的第二音频流完成了这些检查之后, 过程执行单元 42 执行步骤 S124。

[0550] 步骤 S124 是判断步骤, 用于判断是否不存在满足条件 (a) 的第二音频流。如果在步骤 S124 中为“是”, 则在 PSR14 中设定 0xFF 作为第二音频流的流编号 (步骤 S125)。

[0551] 当存在满足条件 (a) 的第二音频流时, 过程执行单元 42 执行步骤 S126 的判断。步骤 S126 用于判断是否存在至少一个满足条件 (a) 和 (b) 两者的第二音频流。如果在步骤 S126 中为“是”, 则过程执行单元 42 从满足条件 (a) 和 (b) 的流中选择一个其相应的流条目位于 STN_table 顶部的流, 并且将所选择的第二音频流的流编号设定在 PSR14 中 (步骤 S127)。

[0552] 当没有满足条件 (a) 和 (b) 的第二音频流时, 过程执行单元 42 执行步骤 S128 的判断。S128 用于判断是否存在一个或多个满足条件 (a) 的第二音频流。当在步骤 S128 中为“是”时, 过程执行单元 42 从满足条件 (a) 的第二视频流中选择一个其相应的流条目在 STN_table 中排第一的流, 将所选择的第二音频流的流编号设定在 PSR14 中 (步骤 S129)。

[0553] 这样就结束了对用于选择适合的第二音频流的过程的描述。

[0554] 图 46 是示出第二音频流的处理过程的流程图。

[0555] 在步骤 S131 中, 过程执行单元 42 检查与用户操作相对应的编号 (编号 x) 所指定的第二音频流是否满足条件 (A)。

[0556] 条件 (A) : 根据在 Comb_info_Secondary_video_Secondary_audio 中所指定的第二音频流编号与由编号 x 所确定的第二音频流编号之间的比较, 允许将具有该编号 x 的第二音频流与当前第二视频流进行组合。

[0557] 如果满足条件 (A), 则过程执行单元 42 执行步骤 S132 的判断。步骤 S132 用于判断该编号 x 是否等于或者小于在 STN_table 中的流条目的总数量, 以及与该编号 x 相对应的第二音频流是否满足条件 (A)。当在步骤 S132 中为“是”时, 过程执行单元 42 选择由该编号 x 所确定的第二音频流, 并将编号 x 设定在 PSR14 中 (步骤 S142)。如果步骤 S132 中为“否”, 则过程执行单元 42 执行步骤 S133 的判断。步骤 S133 用于判断编号 x 是否为 0xFF。“0xFF”的意思与实施例 1 中的编号 x 的意思相同, 即“委托编号”。如果编号 x 不是 0xFF, 则过程执行单元 42 保持 PSR14 中的当前第二音频流编号 (步骤 S143)。

[0558] 如果编号 x 是 0xFF, 则过程执行单元 42 在步骤 S134 到 S136 中检查在 STN_table 的流条目中所写入的所有第二音频流是否满足以下条件, 从而为当前重放项选择适合的第二音频流。

[0559] 当待检查的第二音频流是第二音频流 i 时, 按如下指定条件 (a) 和 (b)。

[0560] 条件 (a) : 根据在 Comb_info_Secondary_video_Secondary_audio 中所指定的第二音频流编号与第二音频流编号 i 之间的比较, 允许将第二音频流 i 与当前第二视频流进行组合。

[0561] 条件 (b) : 当前主音频流的 audio_language_code 与第二音频流的 audio_language_code 匹配。

[0562] 在执行了该循环处理之后,过程执行单元 42 执行步骤 S137 的判断。步骤 S137 用于判断是否存在至少一个满足条件 (a) 和 (b) 的流。当在步骤 S137 中为“是”时,过程执行单元 42 从满足条件 (a) 和 (b) 的第二音频流中选择一个其相应的流条目在 STN_table 中排第一的流,并将所选择的第二视频流的流编号设定在 PSR14 中。

[0563] 如果在步骤 S137 中为“否”,则过程执行单元 42 在步骤 S139 中判断是否存在一个或多个满足条件 (a) 的第二音频流。当在步骤 S139 中为“是”时,过程执行单元 42 在步骤 S140 中从满足条件 (a) 的第二音频流中选择一个其相应的流条目在 STN_table 中排第一的流,并将所选择的第二视频流的流编号设定在 PSR14 中。如果步骤 S139 中为“否”,则过程执行单元 42 保持 PSR14 中的第二音频流编号(步骤 S141)。

[0564] 这样就结束了对第二音频流的“当要求流改变时的过程”的描述。

[0565] 这样就结束了对本实施例的过程执行单元 42 所执行的过程的描述。以下描述混合控制单元 45,其是音频流重放的功能部件。

[0566] 混合控制单元 45 控制混合器 9a 来将音频解码器 8a 的主音频流的重放输出和音频解码器 8b 的第二音频流的重放输出进行混合。

[0567] 当在播放项时间轴中的当前重放时间点在于播放项信息的从 In_time 到 Out_time 的期间内时,并且在当前播放项信息中的 STN_table 中将第二音频流设定为“允许重放”时,混合控制单元 45 控制混合器 9a 来将音频解码器 7a 的重放输出与音频解码器 7b 的重放输出进行混合,因为音频解码器 7b 正在对具有存储在 PSR14 中的流编号的第二音频流进行解码。

[0568] 如果主音频流的属性为“环绕”,则能够在执行缩混(downmixing)之后混合第二音频流的重放输出,以使得在诸如 L、R、C、LS、RS、LR、RR 和 LFE 之类的分量中仅仅保持所希望的分量。在此,在第二音频流是例如电影导演的解说时,通过将要与第二音频流进行混合的主音频流的声音从 L 改变为 C 并改变为 R,可以给予用户好像导演在该用户周围走动的感觉。这种混合称为移位(panning)。在移位中,使用声道数量比主音频流少的第二音频流(例如单声道)的声音/语音。

[0569] 因此,根据本实施例,在 STN_table 中定义了用于主视频流的主音频流与用于第二视频流的第二音频流之间的允许重放的组合。因此,通过从来自 STN_table 中的流条目的主音频流和第二音频流的每一个中选择一个来执行所述混合。由此,在实现画中画时,可以在将电影导演的解说添加到主电影音频中位置处提供合成音频。

[0570] 例如,假设第二视频是这样的视频:其中,仅仅出现了导演和/或演员,并且他们在动作,例如,就好像指向主视频的视频内容。通过将视频内容的第二视频与在画中画中的电影导演的解说的第二视频进行组合,可以实现逗笑的屏幕效果,其中,电影导演和/或者演员在给出解说的同时指向在电影重放视频中的内容。

[0571] 实施例 5

[0572] 本实施例涉及在画中画字幕重放中的改进。字幕重放是在播放列表信息中的 STN_table 中指定的。以下描述在 STN_table 中用于字幕的流条目。尽管分别将主音频流和第二音频流分配给主视频流和第二视频流,但是字幕并没有被分类为用于主视频流的字幕和用于第二视频流的字幕,仍然能够被区分为呈现图形流和 textST 流。

[0573] 以下描述用于 PgtestST 流的流条目。图 47 示出了 STN_table 的一部分,尤其

是与 PgttestST 流相关的部分。根据该图, STN_table 中具体与字幕显示相关的组成部分为:“number_of_PG_textST_streams_entries[1] 到 [n]”;“number_of_PiP_PG_textST_stream_entries_plus”;“PgtextST_stream_entry”;以及在“Secondary_video_Stream_entry”中的“Comb_info_Secondary_video_PiP_PG_textST”。

[0574] “number_of_PG_textST_streams_entries”指示在不执行画中画时,作为选择目标的 PgttestST 流的数量。

[0575] “number_of_PiP_PG_textST_stream_entries_plus”指示应该与用于画中画应用的 STN_table 中所定义的 number_of_PiP_PG_textST_stream_entries 相加的呈现图形流和 textST 流的数量。number_of_PG_textST_streams_entries 和 number_of_PiP_PG_textST_stream_entries_plus 的和为 255 或更小。

[0576] 在此,当 number_of_PG_textST_streams_entries 为 n1 时,从 1 到 n1 的流编号变为 PgttestST 流的范围。另一方面,当 number_of_PiP_PG_textST_stream_entries_plus 为 n2 时,从 1 到 n1+n2 的流编号变为 PgttestST 流的范围。即,如果 number_of_PiP_PG_textST_stream_entries_plus 被写入到 STN_table 中,则 PgttestST 流的流编号范围就会被拓宽了在 number_of_PiP_PG_textST_stream_entries_plus 中所写入的数量。

[0577] 当没有执行画中画时,在 PgttestST 流的流条目之中的 1 到 n1 变为选择目标。当执行画中画时,在 PgttestST 流的流条目之中的 1 到 n1+n2 变为选择目标。注意,即使是执行画中画,也不会将用于主视频的字幕和用于第二视频的字幕分别显示。在画中画中仅仅显示主视频字幕和第二视频字幕中的一个。这是基于两个原因:(i) 如果在 BD-ROM 中分别记录主视频字幕和第二视频字幕,则播放设备需要两个图形解码器,这导致硬件成本的增加;以及(ii) 如果将第二视频缩小,则将会难以看清字幕。

[0578] 引导线 hp1 指示 PG_testST_stream_entry 的内部结构的展开图。如该引导线所示,PG_testST_stream_entry 包括:Stream_entry 和 Stream_Attribute,Stream_entry 包括允许重放的 PgttestST 流的 PID。

[0579] 引导线 hp2 指示“Secondary_video_Stream_entry[1]”的内部结构的展开。在该内部结构的 Comb_info_Secondary_video_PiP_PG_textST 中,唯一性地指示了允许与第二视频流进行合成的 PgttestST 流。

[0580] 引导线 hp3 指示 Comb_info_Secondary_video_PiP_PG_textST 的内部结构。Comb_info_Secondary_video_PiP_PG_textST 包括:“number_of_PG_textST_streams_ref_entries”,指示每一个都能与第二视频流组合的 PiP_PG_textST 流的总数量;以及“PiP_PG_textST_stream_id_ref[0] 到 [n]”,其指示能够被组合用于重放的 PiP_PG_textST 流的流编号。

[0581] 这样就结束了对于在本实施例的记录介质中的改进的描述。以下将描述在本实施例的重放设备中的改进。对于字幕重放,在重放设备中,将用于画中画执行的当前 PgttestST 流的流编号与当前用于画中画不执行的当前 PgttestST 流的流编号分别存储在 PSR 组 23 中。

[0582] 图 48A 示出了当前 PgttestST 流的流编号能够采用的流编号范围。如果 PiP_PG_text_ST_stream_number 为 n1,则在不执行画中画时当前 PgttestST 流的流编号采用范围从 1 到 n1 的值,如该图上方部分所示。

[0583] 另一方面,用于画中画执行的当前 PGtestST 流的编号称为 PiP_PG_TextST 流编号,如果在 STN_table 中所写入的 PG_text_ST_stream_number 为 n1 并且 number_of_PiP_PG_textST_stream_entries_plus 为 n2,则 PiP_PG_TextST 流编号采用范围从 1 到 n1+n2 的值,如该图下方部分所示。

[0584] 图 48B 示出了 PSR2 的比特分配。以下解释 PSR2 的每个比特。

[0585] disp_s_flag :

[0586] PSR2 的 b31 是 disp_s_flag,在该比特中所设定的值按照如下解释 :

[0587] 0b :禁止“PG textST 流”和“PiP PG textST 流”的显示 ;以及

[0588] 1b :允许“PG textST 流”和“PiP PG textST 流”的显示。

[0589] PiP_PG_textST_valid_flag

[0590] PSR2 的 b30 是 PiP_PG_textST_valid_flag,在该比特中所设定的值按照如下解释 :

[0591] 0b :在当前播放项的显示期间显示第二视频流时,使用在 PSR2 中所定义的 PG TextST 流编号 ;以及

[0592] 1b :当在当前播放项的显示期间显示第二视频流时,使用在 PSR2 中定义的 PiP PG TextST 流编号。

[0593] 在当前播放项的显示期间不显示第二视频流时, PiP_PG_textST_valid_flag 对 BD-ROM 重放设备的显示不起作用。在该情况下,使用在 PSR2 中所定义的 PG TextST 流编号。

[0594] PG TextST 流编号 :

[0595] PSR2 的 b0 到 b11 是 PG TextST 流编号,并且在这 12 个比特中所设定的值按照如下解释 :

[0596] 在当前播放项的显示期间不显示第二视频流时 (“当不显示第二视频流时”意味着 PSR14 的 disp_v_flag 被设定为 0b),使用在 PSR2 中的 PG TextST 流编号的值来确定显示在当前播放项的 STN_table 中的 PGTextST 流中的 PG 流和 TextST 流中的哪一个。

[0597] PiP PG TextST 流编号 :

[0598] PSR2 的 b16 到 b27 是 PiP PG TextST 流编号,在这 12 个比特中所设定的值按照如下解释 :

[0599] 在当前播放项的播放期间显示第二视频流并且 PSR2 的 PiP_PG_textST_valid_flag 设定为 1b 时 (“当显示第二视频流时”意味着 PSR14 的 disp_v_flag 被设定为 1b,并且在当前播放项的显示期间显示由 PSR14 所确定的第二视频流),使用在 PSR2 中的 PiP PGTextST 流编号的值来确定显示在当前播放项的 STN_table 中的 PGTextST 流中的 PG 流和 TextST 流中的哪一个。

[0600] 用于画中画不执行的 PGtestST 流的流编号及其处理不是本发明的重点,因此在本实施例中省略了对其的描述。

[0601] 图 49 是 PiP_PG_TextST 流编号的转换转换, PiP_PG_TextS 流编号是画中画 PGtestST 流的流编号。在该图中,术语“有效”表示 PSR2 的 PiP_PG_TextST 流的值等于或者小于播放项的 STN_table 中所写入的条目数量,是可编码的。

[0602] 术语“无效”表示 (a)PSR2 的 PiP_PG_TextST 流编号为“0”, (b)PSR2 的 PiP_PG_

TextST 流编号大于在播放项的 STN_table 中所写入的条目数量,或者 (c) 即使是在播放项的 STN_table 中所写入的条目数量在从“1”到“23”的范围之内,解码也是无效的。

[0603] 图 49 中的虚线框示意性地指示用于在状态改变时确定 PSR 的值的过​​程。用于将 PGTextST 流编号设定给 PSR2 的过程包括“在重放状态改变时的过程”和“当要求流改变时的过程”。

[0604] 虚线框所指示“在重放状态改变时的过程”和“当要求流改变时的过程”与实施例 1 中的第二视频流的过程相同。

[0605] 状态转换由诸如“加载盘”、“改变流”、“开始播放列表的重放”、“穿过播放项边界”或者“终止播放列表的重放”之类的事件触发。这些事件也与用于实施例 1 中的第二视频流的事件相同。

[0606] 以下描述“当要求流改变时的过程”的处理过程。

[0607] 图 50 是示出了 PGtextST 流的“当要求流改变时的过程”的处理过程的流程图。

[0608] 过程执行单元 42 在步骤 S151 中从 PSR2 获得当前 PiP_PG_TextST 流编号,并且在步骤 S152 中判断当前 PiP_PG_TextST 流编号是用于呈现图形流还是 textST 流。在步骤 S153 中,过程执行单元 42 判断与当前 PiP_PG_TextST 流编号相对应的呈现图形流是否满足条件 (A)、(B) 和 (C)。

[0609] 在此,(A)、(B) 和 (C) 具体如下:

[0610] 条件 (A):重放设备能够解码由当前 PiP_PG_TextST 流编号的编号所指定的呈现图形流;

[0611] 条件 (B):重放设备能够播放所指定的语言;以及

[0612] 条件 (C):根据 PSR14 中的第二视频流编号与 comb_info_Secondary_video_PiP_textST() 的 PGtextST 流编号的比较,允许将当前第二视频流与由当前 PiP_PG_TextST 流编号所指定的 PGtextST 流进行组合。

[0613] 另一方面,在步骤 S154 中,过程执行单元 42 判断与当前 PiP_PG_TextST 流编号相对应的 textST 流是否满足条件 (A)、(B) 和 (C)。

[0614] 条件 (A):重放设备能够将当前 PiP_PG_TextST 流编号的 textST 流的字符码转换为位图。这种重放能力在 PSR 组 23 的 PSR30 中指示。

[0615] 条件 (B):重放设备能够支持当前 PiP_PG_TextST 流编号的 textST 流的语言属性。这种支持能力在 PSR 组 23 的 PSR48 到 PSR61 中指示。

[0616] 条件 (C):根据在 PSR14 中的第二视频流编号与 comb_info_Secondary_video_PiP_textST() 的 PGtextST 流编号的比较,允许存储在 PSR14 中的当前第二视频流与由当前 PiP_PG_TextST 流编号所指定的 PGtextST 流进行组合。

[0617] 在此,当采用某种语言的字幕是用 textST 流表示时,重放设备必须具有能够将该字符码转换为位图的能力,以及支持该语言属性的支持能力,以便“能够解码”采用该语言的 textST 流。

[0618] 在此,以英语、日语和阿拉伯语为例。对于英语字幕,仅仅在重放设备具有“水平书写”、“字母紧排 (kerning)”和“双字母 / 连字 (double letter/ligature)”的能力时,才认为其支持该语言属性。

[0619] 对于日语,仅仅在重放设备具有“水平写入”、“垂直书写”、“行尾换行 (line end

wrap)”和“加注音 (rudy)”的能力时,才认为其支持该语言属性。

[0620] 对于阿拉伯语,仅仅在重放设备具有“从右向左书写”和“双字母/连字 (double letter/ligature)”的能力时,才认为其支持该语言属性。

[0621] 当重放设备能够将采用某种语言的 textST 流转换为位图并且支持该语言属性时,就认为满足上述条件 (A) 和 (B)。当重放设备能够将采用该语言的 textST 流转换为位图却不能支持该语言属性时,则认为没有满足条件 (B) 而仅仅满足条件 (A)。

[0622] 在完成了以上判断之后,过程执行单元 42 执行步骤 S155。步骤 S155 是用于判断重放设备是否满足条件 (Z) 的步骤。

[0623] 在此,条件 (Z) 是:用户意图重放采用不支持的语言的字幕。该意图在 PSR 组 23 的 PSR30 中指示。

[0624] 随后,过程执行单元 42 执行步骤 S156 的判断步骤。这是用于判断在当前播放项的 STN_table 中的 number_of_PG_textST_streams_entries 与 number_of_PiP_PG_textST_stream_entries_plus 之和是否为“0”的步骤。当在 STN_table 中没有允许重放的 PGTextST 流时,过程执行单元 42 保持 PSR2 的 PGTextST 流编号 (步骤 S157)。当在当前 STN_table 中存在至少一个允许重放的 PGTextST 流时,过程执行单元 42 执行步骤 S158。这是用于检查当前 PiP_PG_TextST 流编号是否有效的步骤,过程执行单元 42 判断当前 PiP_PG_TextST 流编号是否等于或者小于在 STN_table 中的流条目总数量以及当前 PiP_PG_TextST 流编号是否满足条件 (A)、(B) 和 (C)。

[0625] 当在步骤 S158 中为“是”时,过程执行单元 42 保持在 PSR2 中的值,认为在 PSR2 中已经设定了有效的 PiP_PG_TextST 流编号。(步骤 S159)

[0626] 如果在步骤 S158 中为“否”,则过程执行单元 42 执行步骤 S160 的判断步骤。这是用于判断当前 PiP_PG_TextST 流编号是否等于或者小于在 STN_table 中的流条目总数量以及当前 PiP_PG_TextST 流编号是否满足条件 (A) 的步骤。如果在步骤 S160 中为“是”,则过程执行单元 42 保持在 PSR2 中的值,因为尽管在 PSR2 中设定了采用不支持语言的文本字幕的 PiP_PG_TextST 流编号,但是用户意图重放该不支持的语言 (步骤 S161)。当在步骤 S160 中为“否”时,过程执行单元 42 选择为当前播放项选择的适合的流 (步骤 S162)。

[0627] 这样就结束了对 PGTextST 流的“在重放状态改变时的过程”的描述。

[0628] 图 51 是示出了用于为当前播放项选择适合的 PGTextST 流的过程的流程图。

[0629] 过程执行单元 42 在步骤 S190 中检查所有 PGTextST 流是否满足条件 (a) 到 (d)。

[0630] 当待检查的呈现图形流是呈现图形流 i 时,条件 (a) 到 (d) 具体如下:

[0631] 条件 (a):重放设备能够解码呈现图形流 i;

[0632] 条件 (b):重放设备能够播放呈现图形流 i 所指定的语言;

[0633] 条件 (c):根据在 PSR14 中的第二视频流编号与 comb_info_Secondary_video_PiP_textST() 的 PGtextST 流编号的比较,允许当前第二视频流与呈现图形流 i 进行组合;以及

[0634] 条件 (d):呈现图形流 i 的 PG_language_code 与重放设备上设定的语言匹配。

[0635] 当待检查的 textST 流是呈现图形流 i 时,条件 (a) 到 (d) 具体如下:

[0636] 条件 (a):重放设备能够将 textST 流 i 的字符码转换为位图;

[0637] 条件 (b):重放设备能够支持 textST 流 i 的语言属性;

[0638] 条件 (c) :根据在 PSR14 中的第二视频流编号与 comb_info_Secondary_video_PiP_textST() 中的 PGTextST 流 i 的编号的比较,允许当前第二视频流与由 PiP_PG_TextST 流编号所指定的 textST 流 i 进行组合 ;以及

[0639] 条件 (d) :textST 流 i 的 textST_language_code 与在重放设备上的语言设定匹配。

[0640] 在完成了以上判断之后,过程执行单元 42 在步骤 S191 中判断重放设备是否满足在先前流程图中所描述的条件 (Z) (即,不支持语言的重放)。当在步骤 S191 中为“否”时,过程执行单元 42 在步骤 S192 中判断是否有一个或多个 PGTextST 流满足条件 (a) 到 (d)。如果在步骤 S192 中为“是”,则过程执行单元 42 从满足条件 (a) 到 (d) 的 PGTextST 流中,选择其对应的流条目在 STN_table 中排第一的流,并将所选择的 PiP_PG_TextST 流编号的流编号设定在 PSR2 中 (步骤 S193)。

[0641] 当在步骤 S192 中为“否”时,过程执行单元 42 判断是否有一个或多个 PGTextST 流满足较少的条件。在此,较少的条件指的是满足 3 个条件 (a)、(b) 和 (c),过程执行单元 42 在步骤 S194 中判断是否有一个或多个 PGTextST 流满足所述较少的条件。如果在步骤 S194 中为“是”,则过程执行单元 42 从满足条件 (a)、(b) 和 (c) 的 PGTextST 流中选择其对应的流条目在 STN_table 中排第一的流,并将所选择的 PiP_PG_TextST 流编号设定在 PSR2 中 (步骤 S196)。

[0642] 当在步骤 S194 中为“否”时,过程执行单元 42 将 0xFFF 设定给 PSR2,作为 PiP_PG_TextST 流编号 (步骤 S195)。当在步骤 S191 中判定重放设备满足条件 (Z) 时,过程执行单元 42 在步骤 S197 中判断是否有一个或多个 PGTextST 流满足另一较少的条件。在此,另一较少的条件指的是满足条件 (a)、(c) 和 (d),过程执行单元 42 在步骤 S198 中判断是否有一个或多个 PGTextST 流满足所述较少的条件。

[0643] 如果在步骤 S197 中为“是”,则过程执行单元 42 从满足条件 (a)、(c) 和 (d) 的 PGTextST 流中选择其对应的流条目在 STN_table 中排第一的流,并将所选择的 PiP_PG_TextST 流编号的流编号设定在 PSR2 中 (步骤 S198)。

[0644] 如果在步骤 S197 中为“否”,则过程执行单元 42 在步骤 S199 判断是否有一个或多个 PGTextST 流满足满足条件 (a) 和 (c)。如果在步骤 S199 中为“是”,则从满足条件 (a) 和 (c) 的 PiP_PGtestST 流中选择其对应的流条目在 STN_table 中排第一的流,并将所选择的 PiP_PG_TextST 流的编号设定在 PSR2 中 (步骤 S200)。当在步骤 S199 中为“否”时,过程执行单元 42 将 0xFFF 设定给 PSR2 (步骤 S201)。

[0645] 这样就结束了对用于选择适合的 PGTextST 流的过程的描述。

[0646] 图 52 是示出 PGTextST 流的“当要求流改变时的过程”的处理过程的流程图。

[0647] 在步骤 S171 中,过程执行单元 42 判断响应于用户操作而获得的编号 x 是指示用于呈现图形流的 PiP_PG_TextST 流编号还是用于 textST 流的 PiP_PG_TextST 流编号。在步骤 S172 中,过程执行单元 42 判断与该编号 x 相对应的呈现图形流是否满足以下条件 (A)、(B) 和 (C)。

[0648] 条件 (A) :重放设备能够解码编号 x 所指定的呈现图形流 ;

[0649] 条件 (B) :重放设备能够播放作为所指定呈现图形流的属性的语言 ;以及

[0650] 条件 (C) :根据在 PSR14 中的第二视频流编号与 comb_info_Secondary_video_

PiP_textST() 的 PGtextST 流编号的比较, 允许当前第二视频流与由编号 x 所指定的 PGtextST 流进行组合。

[0651] 在步骤 S173 中, 过程执行单元 42 检查与编号 x 相对应的 textST 流是否满足以下条件 (A)、(B) 和 (C)。

[0652] 条件 (A): 重放设备能够将与编号 x 相对应的 textST 流的字符码转换为位图;

[0653] 条件 (B): 重放设备能够支持与编号 x 相对应的 textST 流的语言属性; 以及

[0654] 条件 (C): 根据在 PSR14 中的第二视频流编号与 comb_info_Secondary_video_PiP_textST() 的 RGtextST 流编号的比较, 允许当前第二视频流与由编号 x 所指定的 PGtextST 流进行组合。

[0655] 过程执行单元 42 在步骤 S174 中检查重放设备是否满足条件 (Z), 并执行步骤 S175 的判断。该判断是用于判断该编号是否小于或等于在 STN_table 中的流条目总数量并且该编号是否满足条件 (A)。当在步骤 S175 中为“是”时, 过程执行单元 42 选择该 PiP_PG_TextST 流编号 (即编号 x) 的 PGTextST 流, 并将该编号设定在 PSR2 中 (步骤 S176)。

[0656] 当在步骤 S175 中为“否”时, 过程执行单元 42 执行步骤 S177 的判断。该判断是用于判断该编号是否小于或等于在 STN_table 中的流条目总数量并且该编号是否满足条件 (A)、(C) 和 (Z)。当在步骤 S177 中为“是”时, 过程执行单元 42 选择与编号 x 相对应的 PiP_PG_TextST 流, 并将该 PiP_PG_TextST 流编号设定在 PSR2 中 (步骤 S178)。

[0657] 如果在步骤 S177 中为“否”, 则过程执行单元 42 执行步骤 S179 中的判断。该判断是用于判断编号 x 是否为 0xFFF。当在步骤 S179 中为“否”时, 过程执行单元 42 保持在 PSR2 中的值, 假设在 STN_table 中没有允许播放的 PGTextST 流 (步骤 S180)。

[0658] 如果该编号 x 为 0xFFF, 则过程执行单元 42 为当前播放项选择适合的 PGTextST 流 (步骤 S181)。对适合的 PGTextST 流的选择与图 51 中所示的相同。

[0659] 这样就结束了对 PGTextST 流的“当要求流改变时的过程”的描述。

[0660] 因此, 根据本实施例, 当画中画有效时, 从通过将 number_of_PG_textST_streams_entries 和 number_of_PiP_PG_textST_stream_entries_plus 相加而得到的数字范围中选择 PiP_PG_TextST 流编号, 并播放与该 PiP_PG_TextST 流编号相对应的呈现图形流或者 textST 流。从而, 可以在一个实例中与多个视频一起播放适合于主视频流的字幕, 并可以在另一实例中与视频一起播放适合于第二视频流的字幕, 这导致了字幕选择范围的扩大。

[0661] 实施例 6

[0662] 本实施例给出了对在上述实施例中的 BD-ROM 的生产和商业制造的详细描述。

[0663] <BD-ROM 的生产>

[0664] 首先, 执行计划过程。在该过程中, 决定使用什么脚本 (scenarios) 来播放 BD-ROM。

[0665] 接下来, 执行素材创建过程。在该过程中, 创建用于视频记录、音频记录等等的素材。

[0666] 然后, 执行格式化过程。在该过程中, 根据在计划过程中所创建的脚本以及素材, 获得要记录在 BD-ROM 的数据区中的数据 (通常称为“卷数据 (volume data)”) 的概观。

[0667] 采用编程语言描述的类结构的实例是根据本发明的记录介质的应用层的格式。通过基于在 BD-ROM 标准中指定的语法描述该类结构的实例, 能够创建片段信息、播放列表信

息、等等。在该情况下,能够使用编程语言的“for”语句来定义采用表格式的数据,使用“if”语句来定义在特定情况下所需要的数据。

[0668] 最后,执行压制 (press) 过程。在该压制过程中,将卷数据转换为物理数据序列,并通过使用该物理数据序列创建模板盘 (masterdisk),来生产模板盘刻纹 (cutting)。

[0669] 通过压制设备创建模板,然后商业批量生产BD-ROM。所述生产包括各种过程,主要包括基片成型、反射膜涂敷、保护膜涂敷、层压和印刷标签。

[0670] 通过完成这些过程,能够创建在以上每个实施例中所述的记录介质 (BD-ROM)。

[0671] < 附加内容创建过程 >

[0672] 当运动画面由 BD-ROM 内容和附加内容组成时,执行以上所述的计划过程到格式化过程。然后,获得构成一块卷数据的 AV 片段、片段信息和播放列表信息。通过存档器程序等等从所获得的 AV 片段、片段信息和播放列表信息中去掉将会由 BD-ROM 提供的信息,并将剩余的信息聚集到一个文件中作为附加内容。当在这些过程之后获得了这种附加内容之后,将附加内容提供给 www 服务器并在要求时发送给重放设备。

[0673] < 创作系统 (authoring system) >

[0674] 在以上多个过程之中,格式化过程是 BD-ROM 生产的核心,其是使用称为创作系统的专用系统来管理的。该系统是在生产技术室中建立的,并且被提供给用户。图 53 示出了实施例 6 的创作系统的内部结构。以下参考该图描述创作系统的内部结构。

[0675] 如图中所示,创作系统是通过经由内部网络将以下设备相互连接而构建的:标题配置创建设备 51、卷本 (reel) 设置编辑设备 52、BD 脚本生成设备 53、Java™ 编程设备 54、素材创建 / 导入设备 55、盘创建设备 56、验证设备 57、模板创建单元 58。

[0676] 1) 标题配置创建设备 51

[0677] 标题配置创建设备 51 确定构成由 Index.bdmv 所指示的每个标题的内容。该设备是通过创建标题配置信息来决定的。标题配置信息是使用树结构来指定在标题、电影对象、BDJ 对象和播放列表之间的关系的信息。具体而言,标题配置信息指定与待生产的 BD-ROM 的“盘名称”相对应的节点、与在 BD-ROM 上的 Index.bdmv 中能够播放的“标题”相对应的节点、与构成标题的“电影对象和 BDJ 对象”相对应的节点、以及通过构成电影对象和 BDJ 对象的命令而播放的“播放列表”的节点,然后通过将这些节点与边缘相连接来指定在标题、电影对象、BDJ 对象和播放列表之间的关系。在标题配置信息中,不使用诸如 00001.mpls 和 00002.mpls 之类的文件名来描述播放列表信息,而是使用诸如主播放列表 (MainPlaylist) 和菜单播放列表 (MenuPlaylist) 之类的抽象名称。该设备通过基于与用户的交互操作创建这种树结构,来完成标题配置信息。

[0678] 2) 卷本设置编辑设备 52

[0679] 卷本设置编辑设备 52 确定构成一个完成的电影的多个基本流 (例如视频、音频、字幕和按钮) 之间的关系。例如,当单个电影由一个视频流、两个音频流、3 个字幕流和一个按钮流组成时,卷本设置编辑设备指定一个电影由这些基本流形成,并且其具有将具有部分不同的图像的导演剪辑分配给主电影以及设置具有不同角度的多视角场景的功能。从卷本设置编辑设备 52 输出的卷本设置文件输出是上述信息的汇编。

[0680] 3) BD 脚本生成设备 53

[0681] BD 脚本生成设备 53 包括菜单编辑单元 53a 和脚本编辑单元 53b。

[0682] < 菜单编辑单元 53a >

[0683] 菜单编辑单元 53a 根据经由 GUI 的用户操作来放置构成交互图形流的按钮, 并创建诸如要附着到按钮上的命令、按钮动画等等的功能。

[0684] < 脚本编辑单元 53b >

[0685] 脚本编辑单元 53b 通过根据经由 GUI 的用户操作对由标题配置创建设备 51 所创建的标题配置信息执行编辑过程, 来创建并输出脚本。在此, 脚本指的是使得重放设备以标题为单位执行重放的信息。在 BD-ROM 中, 被定义为索引表、电影对象和播放列表的信息就是脚本。BD-ROM 脚本数据包括构成流的素材信息、重放路径信息、菜单屏幕排列、以及关于从菜单进行转移的信息, 并且用户使用脚本生成设备来进行脚本编辑操作, 直到完成了对这些信息的验证为止。在脚本编辑操作中, 脚本编辑单元 53b 设定标题配置信息的播放列表的内容。通过在脚本编辑操作中将实施例 1、4 和 5 中所示的 STN_table 和在实施例 2 中的 PiP_metadata 定义为播放列表的组成部分, 将这些组成部分结合到 BD-ROM 脚本数据中。

[0686] 另外, 由 BD 脚本生成设备 53 输出的 BD-ROM 脚本数据包括用于在以下所述的多路复用器 56e 中实现多路复用的参数。

[0687] BD 脚本生成设备 53 能够创建用于上述无缝视频菜单的数据结构的脚本。用户通过经由 GUI 的操作, 使用菜单编辑单元 53a 来选择希望进行无缝播放的视频作为菜单的背景视频。脚本编辑单元 53b 创建与该无缝视频菜单的数据结构相一致的播放列表。脚本编辑单元 53b 调节播放列表中的播放项数量以便与 AV 片段的数量一致, 并将它们输出, 作为 BD-ROM 脚本数据。在这时, 脚本编辑单元 53b 设定用以实现在多路复用器 56e 中的多路复用的参数, 从而无缝地播放每个 AV 片段。

[0688] 4) Java™ 编程设备 54

[0689] Java™ 编程设备 54 包括 ID 类创建单元 54a、Java™ 程序编辑单元 54b、和 BDJ 对象创建单元 54c。

[0690] < ID 类创建单元 54a >

[0691] ID 类创建单元 54a 使用由标题配置创建设备 51 创建的标题配置信息来创建 ID 类源代码。ID 类源代码是用于访问 Index_bdmv 和播放列表信息的 Java™ 类库的源代码, 通过该源代码, 最终在盘上创建 Java™ 程序。在此, 将 Java™ 类库是从 ID 类源代码中形成的汇编, 其被称为 ID 类库。

[0692] ID 类源代码被设计和实现为, 使得其每一个的构造函数都通过指定播放列表编号而从盘中读出预先定义的播放列表文件, 并且使用通过执行该构造函数而创建的实例, 能够执行 AV 片段的重放。通过使用由标题配置信息所定义的播放列表节点的名称, 例如主播放列表和菜单播放列表, 来定义 ID 类库的变量名称。在该处所使用的播放列表名称可以是虚号 (dummy number)。

[0693] < Java™ 程序编辑单元 54b >

[0694] Java™ 程序编辑单元 54b 响应于经由用户接口 (例如 GUI) 的用户请求, 创建 Java™ 程序的源代码, 并输出该 Java™ 程序源代码。在 Java™ 程序中, 由 ID 类库来查询 Index_bdmv 和播放列表。

[0695] < BDJ 对象创建单元 54c >

[0696] BDJ 对象创建单元 54c 根据由 Java™ 程序编辑单元 54b 所创建的 Java™ 程序源代

码和由 ID 类创建单元 54a 创建的 ID 类源代码,来创建 BDJ 对象创建信息。BDJ 对象创建信息是要作为最终记录在 BD-ROM 上的 BDJ 对象的形式的信息,BDJ 对象创建信息不是通过诸如 00001.mpls 和 00002.mpls 之类的具体文件名来指定要播放的播放列表的,而是通过在 ID 类库中定义的变量名来指定的。

[0697] 5) 素材创建 / 导入设备 55 包括字幕创建单元 55a、音频导入单元 55b、视频导入单元 55c、和 Java™ 导入单元 55d。素材创建 / 导入设备 55 将输入的视频素材、音频素材、字幕素材、Java™ 程序源代码等等按照 BD-ROM 标准进行转换,并将其发送到盘创建设备 56。

[0698] < 字幕创建单元 55a >

[0699] 字幕创建单元 55a 根据字幕信息文件,创建符合 BD-ROM 标准的字幕数据,所述字幕信息文件包括字幕、显示时刻信息、以及诸如淡入 / 淡出之类的字幕效果。

[0700] < 音频导入单元 55b >

[0701] 音频导入单元 55b 在所输入的音频已经被压缩为 AC-3 格式的情况下,向输入的音频中添加用于相应的视频的時刻信息和 / 或者从输入的音频中删除无用数据,并输出结果。当输入未压缩的音频时,在输出之前将其转换为用户指定的格式。

[0702] < 视频导入单元 55c >

[0703] 视频导入单元 55c 在所输入的视频已经被压缩为 MPEG2、MPEG4-AVC 或者 VC-1 格式的情况下,在将其输出之前,如果需要的话,就删除无用信息。当输入未压缩的视频文件时,将该视频文件输入到视频编码器,根据用户指定的参数对其进行压缩,然后将其输出。

[0704] Java™ 导入单元 55d 发送以下数据到盘创建设备 56:由 Java™ 编程设备 54 创建的 Java™ 程序源代码、程序辅助信息、ID 类源代码、BDJ 对象生成信息。Java™ 导入单元 55d 使用标题配置信息,将由导入的 Java™ 程序源代码、程序辅助信息、ID 类源代码和 BDJ 对象生成信息组成的文件组与相应的 BDJ 对象相关联,并为标题配置信息的 BDJ 对象节点设置 BDJ 对象生成信息。

[0705] 6) 盘创建设备 56

[0706] 盘创建设备 56 包括 ID 转换单元 56a、静态图像编码器 56b、数据库生成单元 56c、Java™ 程序建立 (building) 单元 56d、多路复用器 56e、格式化单元 56f 和盘镜像创建单元 56g。

[0707] < ID 转换单元 56a >

[0708] ID 转换单元 56a 将从 Java™ 导入单元 55d 发送至盘创建设备 56 的 ID 类源代码转换为标题编号和播放列表编号。ID 转换单元 56a 还转换 BDJ 对象生成信息,从而使得在 BDJ 对象中定义的播放列表名与在盘上实际的播放列表编号相匹配。

[0709] < 静态图像编码器 56b >

[0710] 静态图像编码器 56b 在输入的 BD-ROM 脚本数据包含静态图像或者存储有静态图像的区域的情况下,从所输入的静态图像中选出适合的静态图像,并将所选的静态图像转换为符合 BD-ROM 标准的 MPEG2、MPEG4-AVC 和 VC-1 格式中的一种格式。

[0711] < 数据库生成单元 56c >

[0712] 数据库生成单元 56c 根据所输入的 BD-ROM 脚本数据和从 ID 转换单元 56a 发送的 BDJ 对象生成信息,生成符合 BD-ROM 标准的用于脚本数据的数据库。在此,术语“数据库”是在上述的 BD-ROM 中所定义的 Index、bdmv、电影对象、播放列表、BDJ 对象的集合术语。

[0713] <Java™ 程序建立单元 56d>

[0714] Java™ 程序建立单元 56d 对由 ID 转换单元 56a 所转换的 ID 类源代码和 Java™ 程序源代码执行编译处理,并输出 Java™ 程序。

[0715] <多路复用器 56e>

[0716] 多路复用器 56e 对用于实现在 BD-ROM 脚本数据中所描述的视频、音频、字幕和菜单的多个基本流进行多路复用,获得采用 MPEG2-TS 格式的数字流,其被称为 AV 片段。另外,多路复用器 56e 将 AV 片段与具有与该 AV 片段相关的信息的片段信息一起输出。

[0717] 具体而言,多路复用器 56e 在为 BD-ROM 所生成的数字流中检测:(i) 在所包含的视频基本流是采用 MPEG2 的情况下,I 画面所在的位置;(ii) 在该流是采用 MPEG4-AVC 的情况下,I 画面和 IDR 画面所在的位置;以及(iii) 在该流是 VC1 的情况下,I 画面所在的位置。然后,多路复用器 56e 通过将上述画面的显示时间与 MPEG2-TS 的 AV 片段的 TS 分组相关联,来生成 EP_map,在 TS 分组中,存储了该画面的顶部数据。多路复用器 56e 通过从卷本设置文件中删除多路复用器 56e 已经生成的 EP_map 和指示每个数字流的音频和视频的属性的信息,来创建片段信息。

[0718] 由多路复用器 56e 创建 EP_map 的原因是:EP_map 是与从多路复用器 56e 输出的 MPEG2-TS 格式的 AV 视频紧密相关的信息。另外,为在 BD-ROM 中使用而创建的 AV 片段可能具有非常大的文件尺寸,因此,需要时间来创建 EP_map,这是因为在创建 AV 片段之后,为了创建 EP_map,则必须再次读取具有庞大文件尺寸的 AV 片段。另一方面,在 AV 片段的创建过程中创建 EP_map 减少了 EP_map 创建的时间,因为不需要两次读取庞大的 AV 片段文件。

[0719] 多路复用器 56e 还使用包含在 BD-ROM 脚本数据中的、用于多路复用器 56e 的参数,来改变多路复用方法。例如,当将参数设定为将待多路复用的第一和第二 AV 片段进行无缝连接时,多路复用器 56e 使用对第一 AV 片段进行解码之后的缓冲器状态作为初始值来多路复用第二 AV 片段,以便不会破坏如上所述的缓冲器模型。

[0720] <格式化单元 56f>

[0721] 格式化单元 56f 使用上述数据库、AV 片段和 Java™ 程序作为输入以及使用适合于 BD-ROM 格式的数据结构,来执行文件分配处理。格式化单元 56f 创建指定 BD-ROM 的应用层的目录结构,并将每个文件分配到适当的位置。在这时,格式化单元 56f 通过定义一个树结构,将 Java™ 程序与 AV 片段相关联。根据该关联,将用于重放的多个文件分类到称为块的多个单元中。该设备通过根据与用户的交互操作来创建这种树结构,来实现文件的关联。

[0722] <盘镜像创建单元 56g>

[0723] 盘镜像创建单元 56g 通过使用上述数据库和 AV 片段并将其分配到适合于 BD-ROM 格式的地址,来获得卷镜像。

[0724] 为了创建卷镜像,盘镜像创建单元 56g 将相关的文件组分配为使其在物理上连续,这从而实现了在重放时对盘的有效读取。关于无缝标志为“开启”的块,盘镜像创建单元 56g 将文件组分配为使得能够无缝地播放属于该块的 AV 片段。具体而言,盘镜像创建单元 56g 使得在盘中的分配满足作为实现上述无缝重放的物理条件的最小扩展尺寸和最大跳转距离。

[0725] 7) 验证设备 57

[0726] 验证设备 57 包括模拟器单元 57a 和验证器单元 57b。

[0727] 模拟器单元 57a 使用卷镜像作为输入来播放实际电影内容,并且验证例如生产者想要的操作(例如,从菜单转换转到主电影)是否正确执行、字幕改变和音频改变是否按照意图进行操作、以及视频和音频是否具有想要的质量。

[0728] 验证器单元 57b 使用上述卷镜像,验证所生产的数据是否符合 BD-ROM 标准。

[0729] 为了采用 Out_of_MUX 流实现画中画,在 STN_table 中允许同时播放的多个基本流中的 TS 分组的总比特速率必须限制在 48M 比特/秒或者更低。为了检查是否满足该限制,验证器单元 57b 判断在 ATC 时间轴上任意的一秒时间段内的比特数量是否等于或者小于该限制。该一秒的单位时间称为“窗口”,其能够位于 ATC 序列时间轴上的任何位置处。这就是说,在任何一秒时间段内所解码的基本流的比特数量必须为 48M 比特或者更小。

[0730] 在进行创作时,验证器单元 57b 在每次将窗口在源分组序列上移动一个分组的同时,检查在一秒时间段上的 TS 分组的比特数量是否为 48M 比特或者更小。当满足该限制时,验证器单元 57b 将窗口移动至下一个 TS 分组。如果不满足该限制,则验证器单元 57b 判定其违反了 BD-ROM 标准。当窗口的 Out_Time 在重复这种移动之后到达了最后一个源分组时,验证器单元 57b 判定该源分组符合 BD-ROM 标准。

[0731] 因此,由模拟器单元 57a 和验证器单元 57b 验证了卷镜像,当发现错误时就返回到适当的过程来再次进行该操作。在这两个验证过程之后,卷镜像经过模板创建单元 58,其从而创建用于 BD-ROM 压制的的数据。进而,将用于 BD-ROM 压制的的数据发送到用于盘生产的压制过程。

[0732] 接下来,参考图 54 描述格式化过程的处理流程图。

[0733] 在步骤 S301 中,用户使用标题配置创建设备 51 来设置 BD-ROM 的标题配置。

[0734] 在步骤 S302 中,用户使用 BD 脚本生成设备 53 来创建具有无缝视频菜单结构的脚本数据。从而,在 BD-ROM 脚本数据中创建用于无缝视频菜单的播放列表。

[0735] 在步骤 S303 中,用户准备用于标题的视频、音频、静态图像和字幕信息,随后使用素材创建/导入设备 55 将这些信息导入到盘创建设备 56。

[0736] 在步骤 S304 中,用户使用 Java™ 编程设备 54 创建用于 Java™ 标题的 Java™ 程序源代码、程序辅助信息、以及 ID 类源代码。

[0737] 在步骤 S305 中,用户使用 Java™ 导入单元 55d 将在步骤 S4 中所创建的 Java™ 程序源代码、程序辅助信息、以及 ID 类源代码导入盘创建设备 56。

[0738] 在步骤 S306 中,盘创建设备 56 使用 ID 转换单元 56a 将 ID 类源代码和 BDJ 对象生成信息的描述转换为在实际盘上的标题编号和播放列表编号。

[0739] 在步骤 S307 中,Java™ 程序建立单元 56d 使用在步骤 S306 中输出的源代码,通过编译过程创建 Java™ 程序。注意,当标题配置信息不包含 Java™ 标题时,能够跳过步骤 S306 和 S307。

[0740] 在步骤 S308 中,静态图像编码器 56b 在 BD-ROM 脚本数据包含静态图像或者存储了静态图像的区域的情况下,将适当的静态图像转换为符合 BD-ROM 标准的 MPEG2、MPEG4-AVC 和 VC-1 格式中的一种格式。

[0741] 在步骤 S309 中,多路复用器 56e 根据 BD-ROM 脚本数据对多个基本流进行多路复用,并创建 MPEG2-TS 格式的 AV 片段。

[0742] 在步骤 S310 中,数据库生成单元 56c 根据 BD-ROM 脚本数据创建符合 BD-ROM 标准

的数据库信息。

[0743] 在步骤 S311 中,格式化单元 56f 使用在步骤 S307 中创建的 Java™ 程序、在步骤 S309 中创建的 AV 片段和在步骤 S310 中创建的数据库作为输入,执行符合 BD-ROM 标准的文件分配。在这时,格式化单元 56f 将 Java™ 程序与 AV 片段相关联,并创建文件分配信息。

[0744] 在步骤 S312 中,盘镜像创建单元 56g 使用在步骤 S311 中创建的文件组同时参考文件关联信息,来创建适合于 BD-ROM 格式的卷镜像。

[0745] 在步骤 S313 中,验证单元 57 验证在步骤 S312 中所创建的盘镜像。当发现错误时,验证单元 57 返回到适当的过程来再次进行该操作。

[0746] 以下描述主视频流和第二视频流的多路复用以及 EP_map 的创建。

[0747] 为了对主视频流和第二视频流进行多路复用,将指示每个画面的显示时刻的 PTS 与指示每个画面的解码时刻的 DTS 进行比较。然后,执行多路复用,以使得构成主视频的画面和构成第二视频的画面大致位于同一位置或者彼此相邻。

[0748] 接下来描述如何对主视频流和第二视频流进行多路复用。

[0749] 在图 55 中的层次 1 和层次 2 示出了在同一 ATC 时间轴上所分配的构成主视频流的 TS 分组序列和构成第二视频流的 TS 分组序列。

[0750] 如实施例 1 所示,主视频流和第二视频流分别被转换为 PES 分组序列和 TS 分组序列。将 ATS 附加在 TS 分组上,从而使得这样获得的 TS 分组序列在单个 ATS 时间轴上位置连续。

[0751] 注意,并非在 ATC 时间轴上的所有坐标都被构成主视频流的 TS 分组所占据,在 GOP 附近的坐标是空的。这是因为,在设计上是要将与主视频画面同步播放的一种不同类型的数据放在这些空闲坐标中。将时间戳附加在 TS 分组上,从而使得第二视频流的 TS 分组被分配给空闲坐标,即没有被主视频流的 TS 分组所占据的坐标。层次 3 指示通过多路复用而获得的传输流。

[0752] 因此,通过附加指示在主视频的 ATC 时间轴上未占据的坐标的 ATS 并连续地排列 TS 分组,将主视频流与第二视频流进行多路复用。在多路复用中连续排列的 TS 分组是通过序列号来标识的。这些序列号被称为 SPN,由 SPN 来指示源分组在 BD-ROM 中的位置。

[0753] 当已经以这种方式多路复用了主视频流和第二视频流时,通过按照从顶部开始的次序读出单个传输流,能够在需要的时刻取出所需的主视频流和第二视频流。然而,需要不从流的顶部开始重放,而是根据采用时间规格的跳过操作或者跳转操作而在流中间开始重放时,就必须要考虑主视频和第二视频的 GOP 边界。

[0754] < 同步重放 >

[0755] 主视频和第二视频是作为基本流的独立流,然而,当第二视频流是 IN_MUX 流时,在重放时,通过按照从顶部开始的次序读出单个传输流,能够在需要的时刻取出所需的主视频流和第二视频流。然而,需要不从流的顶部开始重放,而是根据采用时间规格的跳过操作或者跳转操作来在流中间开始重放时,就必须要考虑主视频和第二视频的 GOP 边界。

[0756] 图 58 示出了一种归属关系,其指示每个被多路复用的源分组属于主视频流或者第二视频流的哪一个 GOP。图中层次 1 中的框示出了一种归属关系,其指示 AV 片段的每个 TS 分组属于包含在主视频流中的多个 GOP 中哪一个。可以看到,采用这些框,从地址 n1 到地址 n2 之前紧邻位置处的多个源分组属于 GOP-1,从地址 n2 到地址 n3 之前紧邻位置处的

多个源分组属于 GOP-2,从地址 n3 开始的多个源分组属于 GOP-3。

[0757] 在层次 2 中的框示出了一种归属关系,其用于指示 AV 片段的每个 TS 分组属于包含在第二视频流中的多个 GOP 中哪一个。可以看到,采用这些框,从地址 u1 到地址 u2 之前紧邻位置处的多个源分组属于 GOP-1,从地址 u2 到地址 u3 之前紧邻位置处的多个源分组属于 GOP-2,从地址 u3 开始的多个源分组属于 GOP-3。

[0758] 在该情况下,当从 SPNn1 和 SPNn2 中读出源分组时,第二视频流的 GOP SPNu1 和 SPNu2 的顶部在 SPNn1 和 SPNn2 之后,因此不仅能够读出主视频的 GOP,而且还能够读出第二视频的 GOP,从而完成了画中画。即,当要从作为主视频的第二个 GOP 的(主)GOP-2 开始进行重放时,如果从作为包含在(主)GOP-2 中的第一个分组的 SPNn2 开始进行重放,则也读出(第二)GOP-2-即与主视频进行同步的第二视频的第二个 GOP。因此,可以毫无困难地从重放开始时起就与第二视频同步地播放主视频。

[0759] 然而,当从 SPNn3 开始读出源分组时,第二视频的 GOP 在 n3 之前,因此不能读出第二视频的 GOP。即,当由于希望从(主)GOP-3 开始重放而从 SPNn3 开始进行重放时,如果(第二)GOP-3 在(主)GOP-3 之前,则即使是从 SPNn3 开始读出传输流,也不能从顶部读出在(第二)GOP-3 中包含的 GOP 中的数据。从而,不能对 GOP 中包含的视频进行完全解码。在这种情况下,虽然是从 GOP-3 的视频开始进行主视频的重放的,但是是从 GOP-4 的视频开始延迟地进行第二视频的重放,或者主视频的 GOP-3 的视频重放是以该误时的第二视频流的重放开始的。

[0760] 能够考虑两种方法来使得主视频和第二视频从指定的时刻或者 GOP 开始彼此相一致地进行重放:按照多路复用的次序进行控制;以及确定在重放设备中的开始点。

[0761] 当 GOP 边界在流上对齐时,重要的是从哪个点开始重放。在重放开始点由时钟时间指定的情况下,重放设备将时间信息转换为 SPN,并通过查询 EP_map 来确定重放开始点。

[0762] 图 57 示出了仅用于主视频的 EP_map。层次 1 示出了构成主视频和第二视频的画面,层次 2 示出了 EP_map。层次 3 示出了主视频流和第二视频流的 GOP。在该情况下,时间信息是主视频的 GOP 的重放开始时间,SPN 是 GOP 的顶部地址。由于重放设备从由该地址所指示的点开始读出数据,因此第二视频的数据需要在主视频的 GOP 之后开始。这是按照多路复用的次序进行控制的方法。

[0763] 在从存储在 EP_map 中的时间开始进行重放的情况下,通过使主视频和第二视频的 GOP 具有相同的时间宽度并且始终将主视频的 GOP 的开始放置在第二视频的 GOP 的开始之前,即使是使得重放跳转到任何时间点,也可以从所跳转的时间点开始对主视频和第二视频一起进行显示。

[0764] 图 58 示出了分别为主视频和第二视频所设定的 EP_map。层次 1 和层次 3 与上一个图相同。层次 2 指示分别为主视频和第二视频所设定的 EP_map。通过如图所示地设置 EP_map,当以时钟时间来提供重放开始点时,能够找到用于主视频和第二视频的相应的 EP_map。然后从与时间信息相对应的多个 SPN 中的先前点地址开始进行重放,从而可以从指定时刻的主视频和第二视频的 GOP 的开头处将其读出。这是在重放设备端确定开始点的方法。通过将同一时间信息登记到各个 EP_map 中,就能够在指定了某个时刻时,获得与主视频和第二视频相对应的开始地址。

[0765] 更具体而言,重放设备使用用于主视频流的 EP_map 将时间信息转换为主视频流

上的 SPN, 还使用用于第二视频流的 EP_map 将时间信息转换为第二视频流上的 SPN。

[0766] 随后, 对采用这种方式所获得的 SPN 进行比较。当 SPN1 小于 SPN2 时, 将 SPN1 转换为 BD-ROM 的地址。当 SPN2 小于 SPN1 时, 将 SPN2 转换为 BD-ROM 的地址。

[0767] 因此, 将单个时间信息转换为在主视频流和第二视频流上的 SPN, 并且将这些 SPN 中较小的一个转换为地址。因此, 如果主视频流的 GOP 在第二视频流的 GOP 之前, 或者如果第二视频流的 GOP 在主视频流的 GOP 之前, 都能够读出两个 GOP。

[0768] 注意, 当存在多个主视频和第二视频时, 能够从需要进行显示的多个视频流的多个重放开始点之中选择具有最早点的开始地址, 并且不必对不进行显示的流执行地址搜索。

[0769] 实施例 7

[0770] 本实施例涉及用于使得重放设备实行逐个章节的重放的改进。经常的情况是, 在一个电影标题中定义了称为章节的部分 (其是通过根据电影情景或内容对电影进行分割来创建的), 并且能够从用户所选择的章节的开头处开始进行重放。考虑对画中画采用这种章节选择应用。

[0771] 章节的开头是情景中的中断或者内容中的改变, 优选地是, 当从章节开头处播放该章节时, 在起始情景时刻, 不仅开始主视频的重放, 而且还开始第二视频的重放。

[0772] 图 59 示出了播放列表信息的播放列表标记 (PlayListMark) 信息。在该图中的层次 2 和层次 3 示出了主视频流和第二视频流的 GOP, 层次 1 示出了播放列表标记信息。播放列表标记信息是通过将多个章节编号中的每一个与时间信息相关联而建立的。

[0773] 章节的位置由时间信息表示。因此, 为了根据章节确定重放开始地址, 重放设备通过上述使用 EP_map 的方法来确定该地址, 并从该地址开始数据的读出。

[0774] 当仅为主视频设定 EP_map 时, 为主视频提供标记, 并且要同步播放的第二视频的起始点应该放置在比主视频的起始点靠后的位置。

[0775] < 跳过点 (skip point) >

[0776] 与章节边界类似, 通过使用能够指定在标题时间轴上的一个点的标记, 能够通过用户作出的跳过操作来指定跳转点 (jump point)。对跳过点的 GOP 定位的控制以及从跳过点开始的重放方法与用于章节开始的控制和方法相同。

[0777] 现在, 要考虑的情况是, 为多个第二视频分别设置跳过点。

[0778] 可能的是, 在主视频设置跳过点时, 用户能够指定跳过点并且不论是否显示第二视频都使重放点跳转到该点。这是在例如当在主电影的情景发生改变的点处提供跳过点时使用的。另一方面, 对于第二视频 (其例如, 导演解说的视频) 的跳过点, 如果这些跳过点有效, 除了正在显示相应的视频之外都会令用户困惑。

[0779] 图 60 示出了用于指定第二视频流所在位置作为章节的播放列表标记信息。在该图中, 层次 2、3 和 4 分别示出了主视频流和两个第二视频流, 而层次 1 示出了播放列表标记信息的内容。

[0780] 第二视频流 #1 是导演解说的视频, 其具有跳过点, 因为该解说的内容在中间发生了改变。在该情况下, 如果正在显示第二视频流 #1, 则用户能够通过指定跳过点 (标记编号 2) 来进行跳转。然而, 应该设计为, 在没有显示第二视频流 #1 时或者在正在显示第二视频流 #2 时, 用户不能进行该跳转。

[0781] 因此,在本实施例中,播放列表标记信息的每个标记信息都与称为“流说明”的字段相关联。每个标记信息最初包含标记编号和相应的时间信息,并且现在为标记信息提供了流说明字段。

[0782] 如果流说明字段被设定为“无说明(全部)”,则允许对该标记信息点的章节跳转且与第二视频流的显示无关。如果在流说明字段中设定了“第二视频流 #1”,则仅仅在播放第二视频流 #1 时才允许对该标记信息点的章节跳转。

[0783] 如果在流说明字段中设定了“第二视频流 #2”,则仅仅在播放第二视频流 #2 时才允许对该标记信息点的章节跳转。因此,重放设备使得用户能够使用适合于所显示视频的内容的跳过点。

[0784] 因此,通过在各个流上设置跳过点,能够提供允许用户根据所显示内容而跳转到适当位置的机制。

[0785] 注意,在希望通过使用 GUI 强制性地指定标记编号来将重放开始点移动到标记 2 的位置处的情况下,即使是没有显示第二视频流 #1 或者正在显示另一第二视频流时,也能够将显示强制性地改变到第二视频流 #1,并且将重放点移动到该跳过点。

[0786] 实施例 8

[0787] 本实施例涉及对于主视频和第二视频是隔行扫描视频的情况的改进。画面数据由以下要描述的前场(TOP field)和后场(BOTTOMfield)组成。前场是构成一个帧的多个行中的奇数行的汇集。后场是构成一个帧的多个行中的偶数行的汇集。

[0788] 图 61A 示出了构成主视频和第二视频的视频场。在画中画中,构成主视频的视频场和构成第二视频的视频场按照一对一的方式合成。因此,待合成的主视频和第二视频的视频场可以是彼此同相或者反相的。

[0789] 图 61B 示出了待合成的视频场是彼此同相的情况下的组合。同相组合意思是,如图所示,主视频和第二视频都是前场或者都是后场。

[0790] 图 61C 示出了待合成的视频场是彼此反相的情况下的组合。反相组合意思是,如图所示,主视频是前场而第二视频是后场,或者主视频是后场而第二视频是前场。

[0791] 因此两种不同的类型的组合都是可能的,当合成主视频和第二视频时,能够创建如图 62A 和 62B 所示的两种画中画图像。图 62A 示出了主视频的视频场的偶数行与第二视频的视频场的偶数行匹配的画中画图像。这种画中画图像能够输出并显示在 TV 上。

[0792] 图 62B 示出了主视频的视频场的偶数行与第二视频的视频场的奇数行匹配的画中画图像。这种画中画图像不能够输出并显示在 TV 上。

[0793] 为了避免得到不能显示的画中画图像,即图 62B 所示的合成图像,在本实施例的记录介质中,采用以下方式来调节第二视频的位置的显示点。

[0794] 图 63 示出了在对在构成主视频的多个视频场中的给定视频场 F_x 和在构成第二视频的多个视频场中的给定视频场 F_y 进行合成并输出的情况下,如何调节视频场 F_y 的位置和显示时间。

[0795] 在此,当由 PiP_metadata_block 中的 PiP_horizontal_position 和 PiP_vertical_position 所指定的坐标是 (P_x, P_y) 时,通过将坐标 (P_x, P_y) 向上或者向下滑动,如箭头 $sd1$ 所示,来调节 P_y 的位置。

[0796] 当在重放时间轴上的主视频的场数据的显示时刻为 T_y ,通过在时间的未来方向和

过去方向上滑动该显示时刻,如箭头 sd2 所示,来调节由 PiP_metadata_time_stamp 所指定的显示点。

[0797] 图 64 示出了将第二视频的场 Fy 与主视频的同相场一起进行显示的情况。在此,从主视频场中的多个偶数行中的一个行(图中为偶数 1)中选出坐标 (Px, Py) 的 Py,其应该由 PiP_metadata_block 的 PiP_horizontal_position 和 PiP_vertical_position 指定。因此,能够获得可播放的画中画图像。

[0798] 图 65 示出了将第二视频的场 Fy 与主视频的反相场一起进行显示的情况。在此,从主视频场中的多个偶数行中的一个行(图中为奇数 1)中选出坐标 (Px, Py) 的 Py,其应该由 PiP_metadata_block 的 PiP_horizontal_position 和 PiP_vertical_position 指定。

[0799] 图 66 示出了在主视频的偶数行上显示第二视频场的情况。在此,从应该显示主视频的同相场的多个时间点中选出时间点 Ty,其应该由 PiP_metadata_block 的 PiP_metadata_time_stamp 指定。因此,能够获得可播放的画中画图像。

[0800] 图 67 示出了在主视频的奇数行上显示第二视频场的情况。在此,从应该显示主视频的反相场的多个时间点中选出时间点 Ty,其应该由 PiP_metadata_block 的 PiP_metadata_time_stamp 指定。因此,能够获得可播放的画中画图像。

[0801] 根据要实现的内容(即,第二视频的时间点或者第二视频的坐标),能够创建可播放的组合。因此,在创作阶段,确保将会正确地执行重放输出。

[0802] 实施例 9

[0803] 本实施例涉及在创作阶段没有进行对 PiP_metadata_block(其在先前实施例中进行了描述)的调节的情况下,在重放设备端进行的补偿。

[0804] 在本实施例中的重放设备检查画中画的合成模式是图 62A 中所示的模式还是图 62B 中所示的模式。当其是图 62B 中所示的模式时,重放设备进行在图 64 到 67 中所示的调节。因此,即使是在创作过程中没有调节用于采用实例创建画中画的 PiP_metadata_time_stamp、PiP_horizontal_position 和 PiP_vertical_position,也能够实现画中画。

[0805] 根据本实施例,即使是在创作时没有进行用于实行画中画的调节,重放设备也能够对此进行补偿。

[0806] 实施例 10

[0807] 本实施例涉及对以下情况的改进:通过在每个实施例的重放设备上安装用于封装介质目标的 Java™ 2Micro_Edition(J2ME)Personal Basis Profile(PBP1.0)和全局可执行 MHP 说明书(GEM1.0.2)来构建 Java™ 平台,并使得重放设备执行 BD-J 应用程序。

[0808] 播放列表信息包括如实施例 1 中所述的主路径信息和子路径信息,主路径信息指定主视频流,子路径信息的子播放项指定第二视频流。因此,Java 应用程序能够通过命令 Java 虚拟机生成用于播放多个播放列表的 JMF 播放器实例,来使得重放设备执行画中画重放。JMF 播放器实例的生成优选地是通过称为 JMF A"BD://00001.mpls"的方法来实现的。

[0809] 注意,可以使 Java 虚拟机生成指定在主视频流中的重放部分的播放项信息的 JMF 播放器实例和指定在第二视频流中的重放部分的子播放项信息的 JMF 播放器实例,然后可以使重放设备根据这两个播放器实例来执行重放。

[0810] <补充注意事项>

[0811] 已经描述了在提交本申请时申请人所知道的用于实现本发明的最佳方式。然而,

在以下技术主题上,能够对本发明进行进一步的改进或者修改。在此应该注意,是否进行这种改进和修改是可选的,取决于本发明的实现情况。

[0812] < 标题 >

[0813] 优选的是,在重放设备中创建“模块管理器”,其根据 BD-ROM 的加载、用户操作、或者设备状态,来选择标题。在 BD-ROM 重放设备中的解码器根据“模块管理器”的标题选择,基于播放列表信息来执行 AV 片段的重放。

[0814] 当“模块管理器”选择一个标题时,应用程序管理器使用与先前标题相对应的应用程序管理表 (AMT) 和与当前标题相对应的 AMT 来进行信号发送。所述信号发送进行如下控制:终止在先前标题的 AMT 中描述了但是在当前标题的 AMT 中未描述的应用程序的操作,同时开始在先前标题的 AMT 中未描述但是在当前标题的 AMT 中描述了的应用程序的操作。

[0815] < 本地存储器中的目录结构 >

[0816] 优选地,在与 BD-ROM 的盘的根证书相对应的目录下,创建在每个实施例中所描述的在本地存储器中的各个区域。

[0817] 盘的根证书是根证书授权中心 (root certificate authority) 发布并由 BD-ROM 的创建者指定给 BD-ROM 的证书。盘的根证书是采用例如 X.509 编码的。X.509 的说明书已经由国际电报电话咨询委员会发布,并在 CCITT Recommendation X.509 (1988), "The Directory-Authentication Framework" 中进行了描述。

[0818] 另外,优选的是,使用先进访问内容系统 (Advanced AccessContent System, AAC) 来编码记录在 BD-ROM 和本地存储器中的内容,并且对其附加签名信息,并在许可文件中指定使用授权。

[0819] < 控制过程的实现 >

[0820] 在上述实施例中流程图所解释的控制过程以及在上述实施例中所解释的功能组件的控制过程满足“程序发明”的要求,因为上述控制过程具体上是采用硬件资源实现的,并且是利用自然规律的技术方案的创建。

[0821] ● 本发明的程序的产生

[0822] 本发明的程序是能够在计算机上执行的目标程序。该目标程序是包括使计算机执行在流程图中的每个步骤或者功能组件的每个过程的一个或多个程序代码。存在各种类型的程序代码,例如处理器的自然语言、JAVA™ 字节代码。还存在实现程序代码的步骤的各种形式。例如,当能够使用外部函数实现每个步骤时,使用 call 语句来调用外部函数。实现一个步骤的程序代码可以属于不同的目标程序。在指令类型受到限制的 RISC 处理器中,可以通过将算术运算指令、逻辑运算指令、分支指令等等进行组合,来实现流程图中每个步骤。

[0823] 本发明的程序能够按照如下产生。首先,软件开发者使用编程语言编写实现每个流程图和功能组件的源代码。在该编写中,软件开发者使用符合她 / 他所使用的编程语言的语句结构的类结构、变量、数组变量、对外部函数的调用等等。

[0824] 将所编写的源程序作为文件发送到编译器。编译器转换源代码,生成目标程序。

[0825] 编译器所执行的转换包括诸如语句结构分析、优化、资源分配、和代码生成之类的过程。在语句结构分析中,对字符和短语、语句结构、以及源程序的含义进行分析,并将源程序转换为中间程序。在优化中,对中间程序进行诸如基本块设置、控制流分析、以及数据流分析之类的处理。在资源分配中,为了适合于目标处理器的指令集,将中间程序中的变量分

配给目标处理器的寄存器或者存储器。在代码生成中,将在中间程序中的每个中间指令转换为程序代码,获得目标程序。

[0826] 在生成了目标程序之后,程序员启动连接器。连接器为目标程序和相关的库程序分配存储器空间,并将其连接在一起,来生成装入模块(load module)。所生成的转入模块是基于以下假设的:其已经被计算机读出,并使得计算机执行在流程图中所指示的过程和功能组件的过程。这样就产生了本发明的程序。

[0827] 能够按照如下来使用本发明的程序。当将本发明的程序用作嵌入程序时,将作为该程序的装入模块与基本输入/输出系统(BIOS)程序和各種中间件(操作系统)一起写入指令ROM中。在将该指令ROM嵌入到控制单元中并由CPU执行时,本发明的程序用作重放设备300的控制程序。

[0828] 当重放设备是自举模块时,将基本输入/输出系统(BIOS)程序嵌入指令ROM中,并且将各种中间件(操作系统)预先装入第二记录介质中,例如硬盘。通常,在重放设备中提供引导ROM,用于启动来自第二记录介质的系统。在该情况下,经由可移动记录介质和/或者网络而仅将装入模块提供给重放设备,并将其作为应用程序安装在第二记录介质上。这使得重放设备能够执行引导ROM的自举,以启动操作系统,然后使得CPU执行作为应用程序而安装的装入模块,从而能够使用本发明的程序。

[0829] 如上所述,当重放设备是自举模块时,本发明的程序能够用作一个应用程序。因此,可以经由网络独立地运输、出借、提供本发明的程序。

[0830] <控制器 22>

[0831] 控制器 22 能够实现为一个系统 LSI。

[0832] 该系统 LSI 是通过在高密度基板上实现承载芯片(bear chip)并对它们进行封装而获得的。系统 LSI 还通过在高密度基板上实现多个承载芯片并对它们进行封装从而这些承载芯片具有一个 LSI 的外观(这种 LSI 称为多芯片模块)来获得的。

[0833] 系统 LSI 具有 QFP(四侧引脚扁平封装)类型和 PGA(引脚网格阵列)类型。在 QFP 类型的系统 LSI 中,引脚附着在封装的 4 个边上。在 PGA 类型的系统 LSI 中,多个引脚附着在整个底部。

[0834] 这些引脚充当与其他电路的接口。系统 LSI 通过作为接口的这种引脚与其他电路相连,起到了重放设备 300 的核心的作用。

[0835] 如以上“用作嵌入程序”中所述,将作为程序的装入模块、基本输入/输出系统(BIOS)和各种中间件(操作系统)写入指令ROM。本实施例的主要改进是通过作为程序的装入模块来实现的。因此,本发明的系统 LSI 可以通过在其中装入存储了作为程序的装入模块的指令ROM作为承载芯片来生成。

[0836] 具体的生成过程如下。首先,基于在每个实施例中的结构图,创建要成为系统 LSI 的部件的电路图,并使用电路元件和 IC 或者 LSI 来实现在结构图中的组件。

[0837] 因此实现了每个组件,然后指定在电路元件和 IC 或者 LSI、外围电路、与外部的接口等等之间进行相连的总线。除此之外,还指定连接线、电源线、地线、时钟信号线等等。在这些说明中,考虑到 LSI 的规格来调节每个组件的操作时刻,并且通过进行调节,例如为每个组件分配所需带宽,来完成电路图。

[0838] 在完成了电路图之后,进行实现设计。实现设计是用于以下的工作:创建基板布局

以便决定在电路设计中所创建的电路图中的组件（电路元件和 IC 或者 LSI）放置在基板的什么位置，或者电路图上的连接线在基板上是如何进行走线的。

[0839] 在此，该实现设计包括自动放置和自动走线，

[0840] 在使用 CAD 设备的情况下，通过使用称为“质心法”的具体算法，能够实现这种自动放置。自动走线定义了用于使用金属薄片和通孔连接电路图组件的引脚的连接线。当使用 CAD 设备时，通过使用称为迷宫 (maze) 算法和线搜索算法的具体算法能够实现该走线过程。

[0841] 这样就进行了实现设计，一旦决定了在基板上的布局，就将实现设计的结果转换为 CAM 数据并输出到诸如 NC 机器工具之类的装置上。NC 机器工具根据 CAM 数据执行 SoC 实现或者 SiP 实现。SoC（芯片上系统）实现是将多个电路烧制到一个芯片上的技术。SiP（封装内系统）实现是使用树脂将多个芯片放置在一个封装中的技术。通过以上过程，根据在每个实施例中所述的重放设备的内部结构图，能够生产本发明的系统 LSI。

[0842] 在此应该注意，尽管在此使用了术语 LSI，但是其可以称为 IC、LSI、超大 LSI、极大 LSI 等等，这取决于集成程度。

[0843] 此外，每个重放设备的组件中的一部分或者全部可以实现为一个芯片。集成电路并不局限于 SoC 实现或者 SiP 实现，而是可以使用专用电路或者通用处理器来实现。也可以通过使用 FPGA（现场可编程门阵列）或者可重构处理器来实现该集成电路，其中，FPGA 在制造之后能够进行再次编程，可重构处理器能够重新配置在 LSI 内部的电路单元的连接和设置。此外，随着半导体技术的改进或者分支到另一种技术，在即将到来的未来可能出现替代 LSI 的集成电路技术。在该情况下，可以将该新技术结合到如上所述的、构成本发明的功能模块的集成中。这种可能的技术包括生物工程学。

[0844] < 架构 >

[0845] 由于假设本发明的系统 LSI 是内建到 BD-ROM 重放设备中的，因此，系统 LSI 优选地符合 UniPhier 架构。

[0846] 服从 UniPhier 架构的系统 LSI 包括以下电路模块。

[0847] ● 数据并行处理器 DPP

[0848] 这是一种 SIMD 处理器，在其中，多个单元处理器 (elementprocessor) 并行地操作，通过用一条指令并行地操作内建在各个单元处理器中的计算单元，来将构成一个画面的多个像素的解码处理并行化。实现这种并行化实现了对视频流的解码。上述视频解码器被实现为这种数据并行处理器。

[0849] ● 指令并行处理器

[0850] 其包含：指令 RAM、指令高速缓存、数据 RAM、由数据高速缓存组成的“本地存储器控制器”、指令读取单元、解码器、执行单元、由寄存器文件 (register file) 构成的“处理单元”、以及使得处理单元执行多个应用程序的并行化的“虚拟多处理器单元”。除了视频解码器之外的图形解码器和音频解码器在系统 LSI 中被实现为指令并行处理器 IPP。

[0851] ● CPU 模块

[0852] 其包含：ARM 内核、外部总线接口 (BUS 控制单元：BCU)、DMA 控制器、定时器、诸如矢量中断控制器之类的外围电路、UART、GPIO（通用输入输出）、以及诸如同步串行接口之类的外围接口。上述控制器在系统 LSI 中实现为该 CPU 模块。

[0853] ●流 I/O 模块

[0854] 其执行与经由 USB 接口或者 ATA 分组接口连接的外部总线上的 BD-ROM 驱动设备、硬盘驱动设备和 SD 存储卡驱动设备之间的数据输入输出。

[0855] ● AVI/O 模块

[0856] 其包含音频输入输出、视频输入输出和 OSD 控制器,并执行 TV 和 AV 放大器的数据输入输出。

[0857] ●存储器控制模块

[0858] 其是实现经由外部总线连接的 SD-RAM 的读写的模块,并且包括:内部总线连接单元,其控制在各个模块之间的内部连接;访问控制单元,其执行与连接到系统 LSI 外部的 SD-RAM 之间的传输;以及访问调度单元,其调节每个块的对于访问 SD-RAM 的请求。

[0859] 关于符合这种架构的系统 LSI 的生产,希望采用从低到上(bottom-up)的布局技术,该技术通过为每个电路块(例如 IPP 和 DPP)进行布局,并在对每个电路块的性能进行优化之后建立每个块,来完成一个芯片布局。

[0860] 工业可应用性

[0861] 本发明的记录介质和重放设备能够根据在以上实施例中所示的其内部结构而批量制造。这样,本发明的重放设备具有工业可应用性。

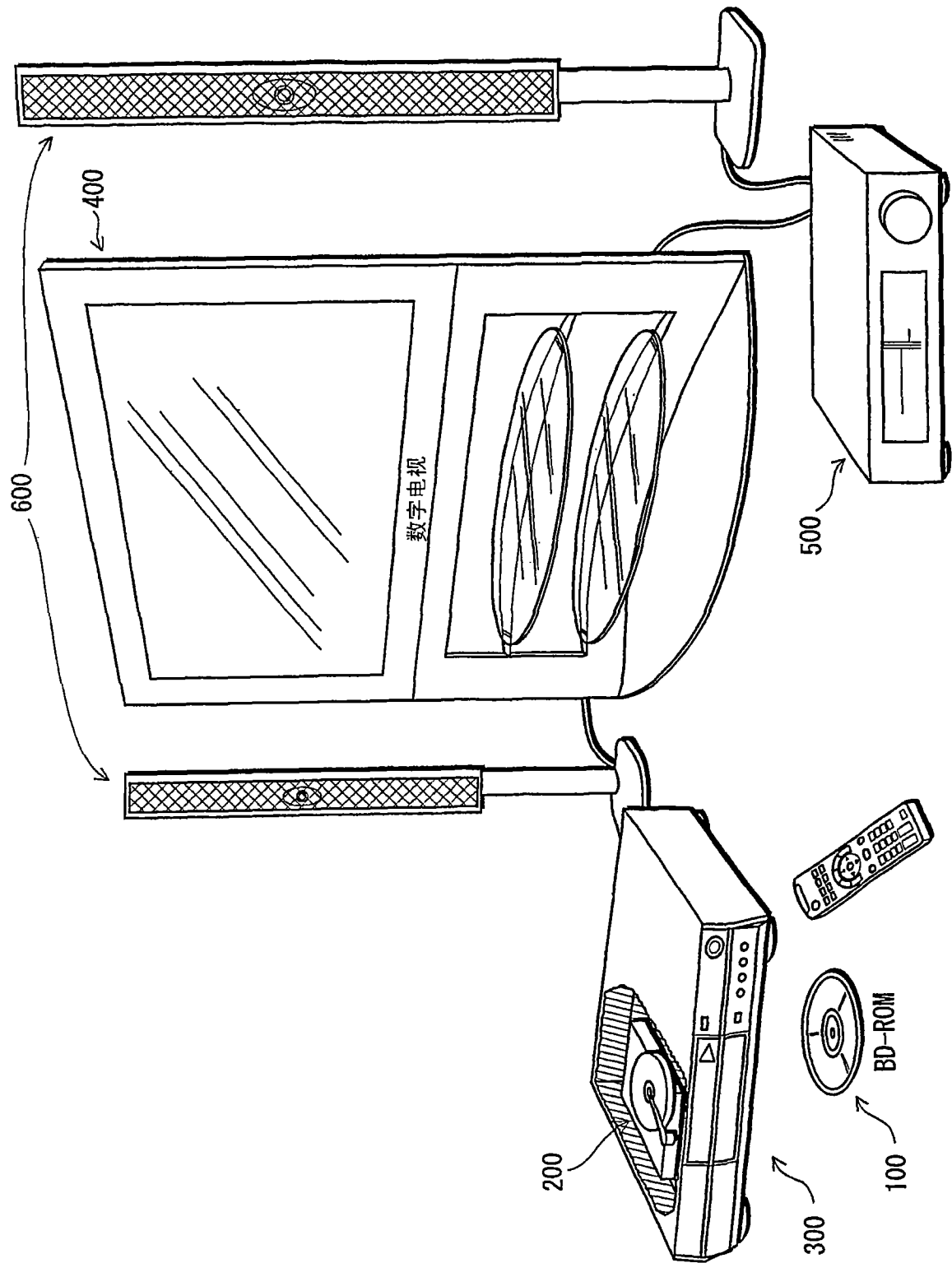


图 1

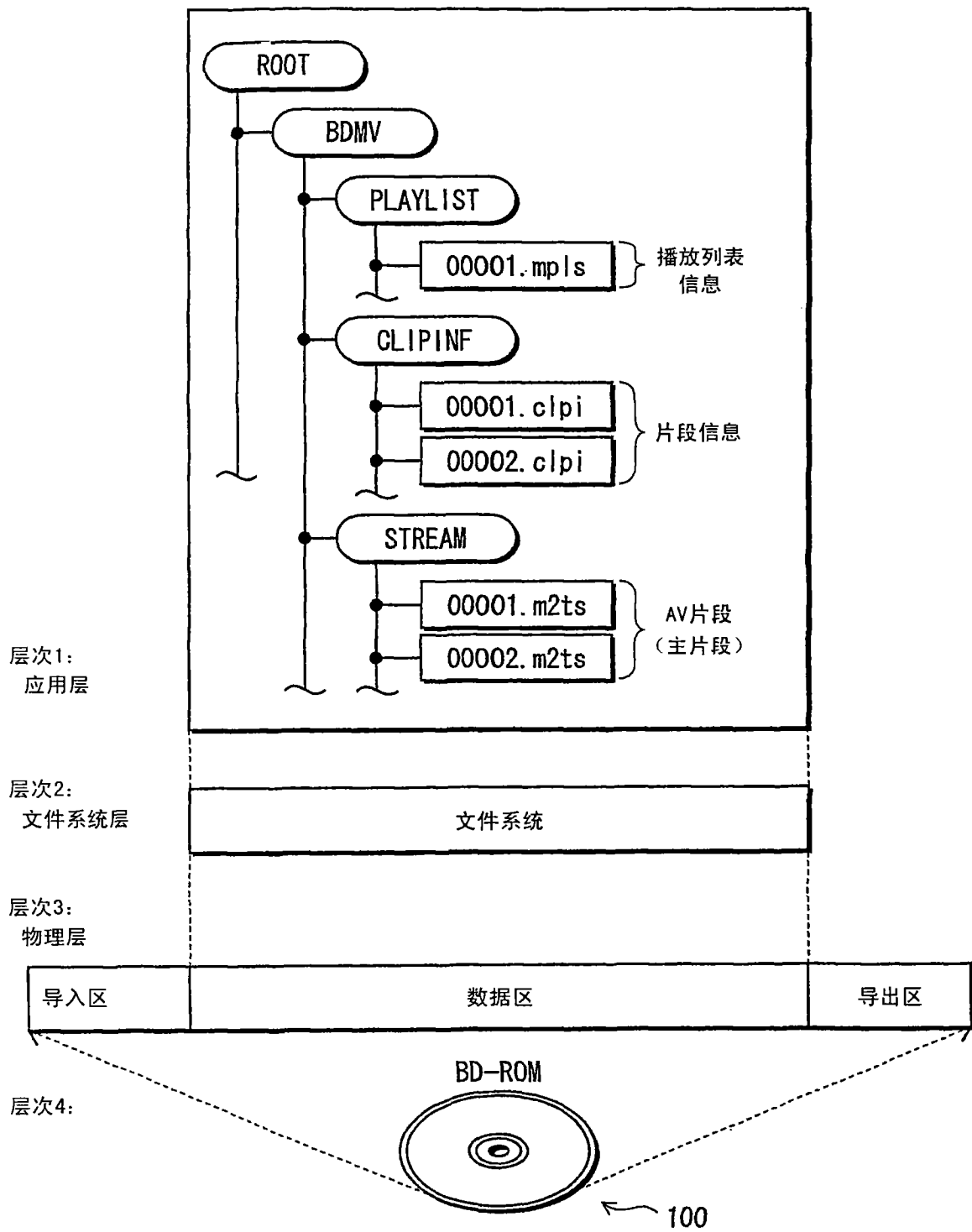


图 2

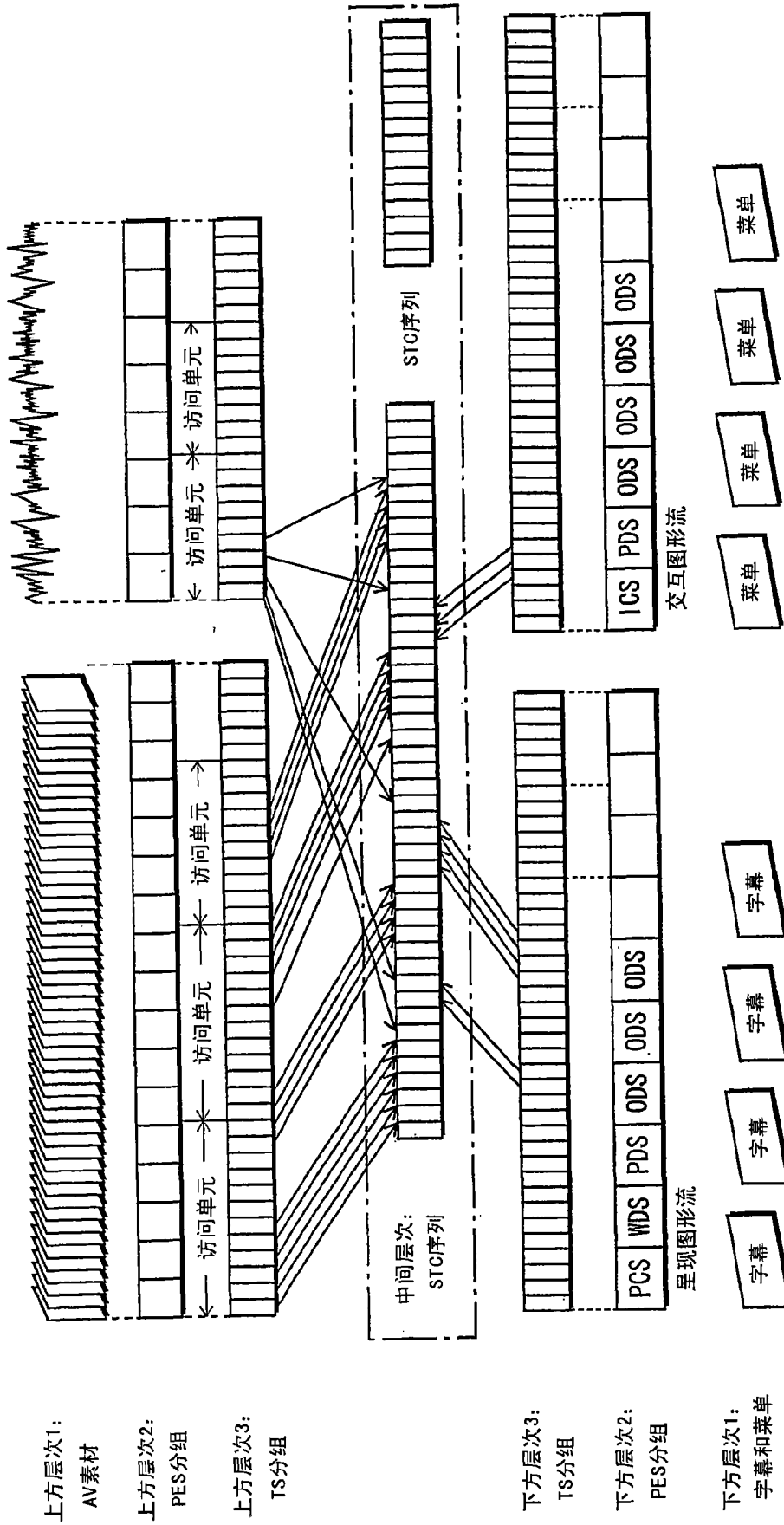


图 3

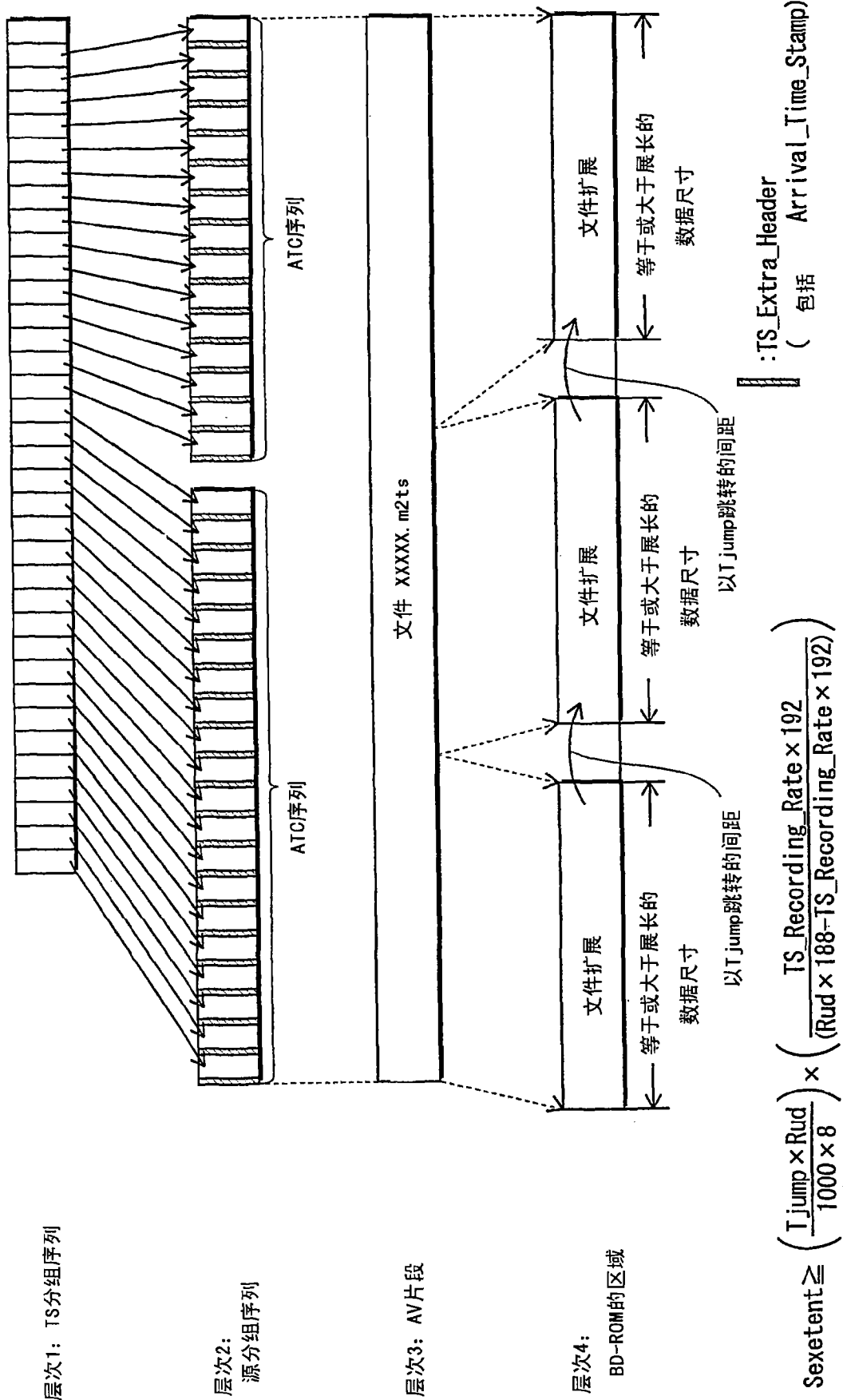
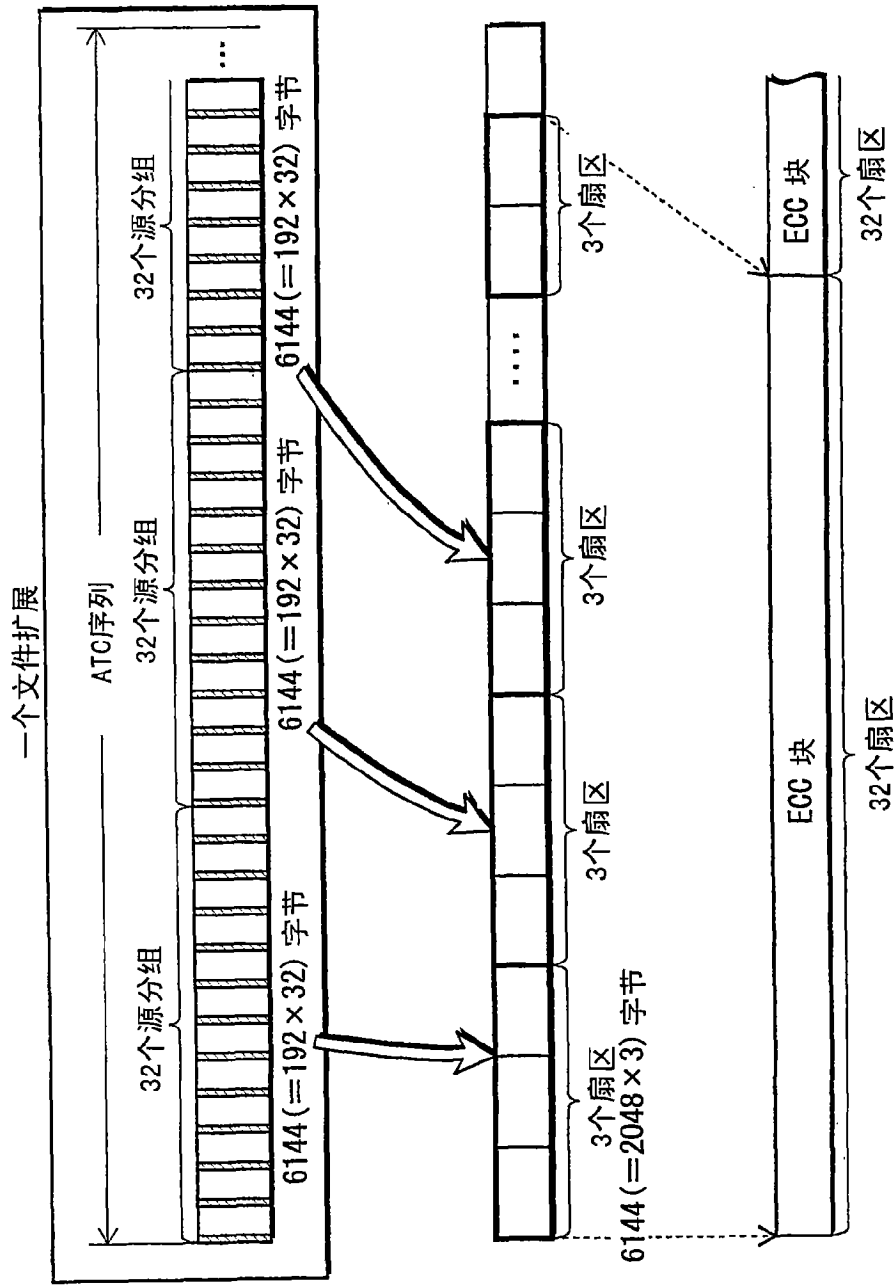


图 4



层次1：
构成文件扩展的
对齐单元

层次2：
在BD-ROM上的扇区序列

层次3：在BD-ROM上的ECC块

图 5

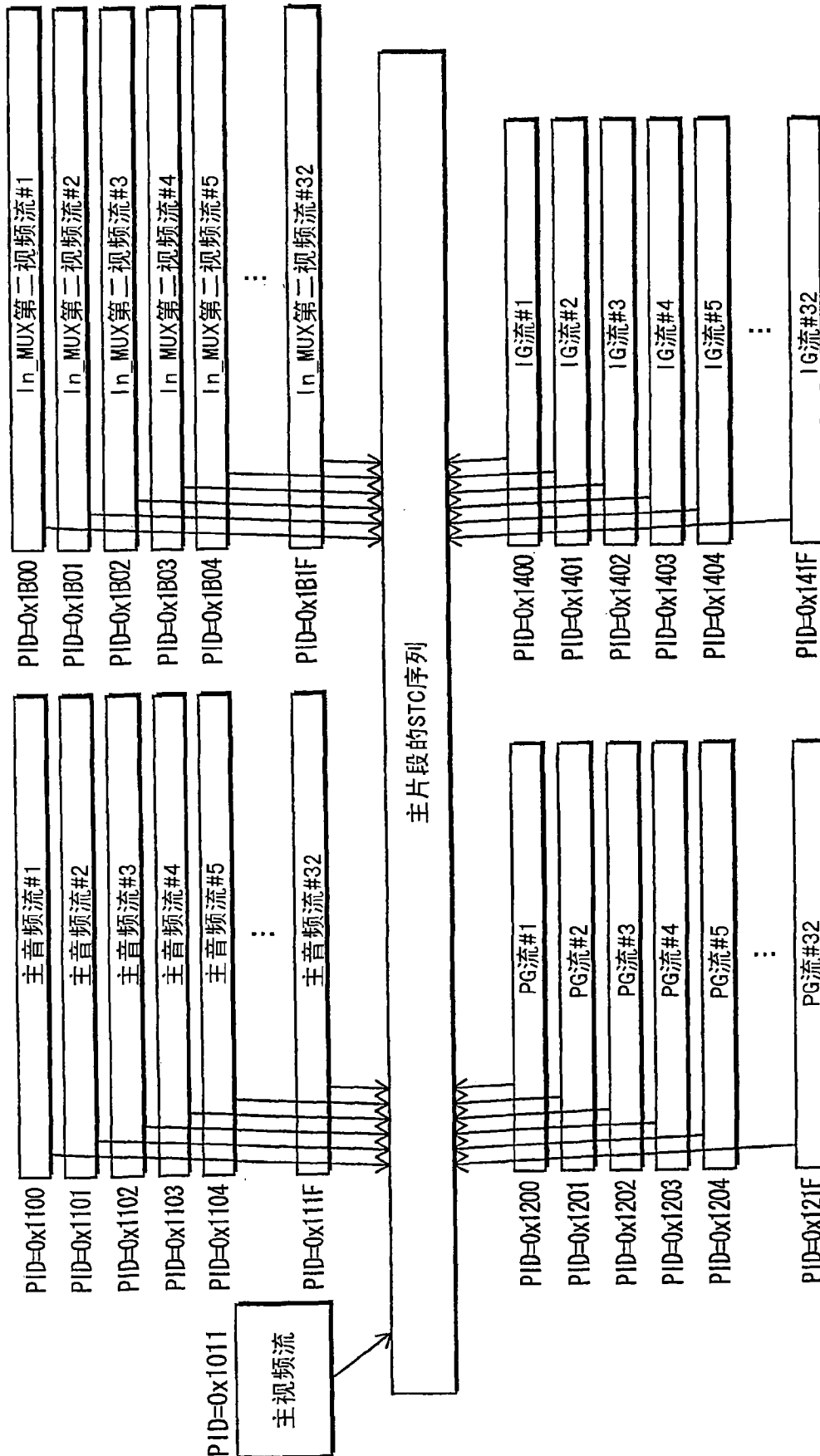


图 6

PID的范围	分配给该范围的流
0x0100	program_map
0x1001	PCR
0x1011	主视频流
0x1100~0x111F	主音频流
0x1200~0x121F	PG流
0x1400~0x141F	IG流
0x1B00~0x1B1F	In_MUX第二视频流

图 7

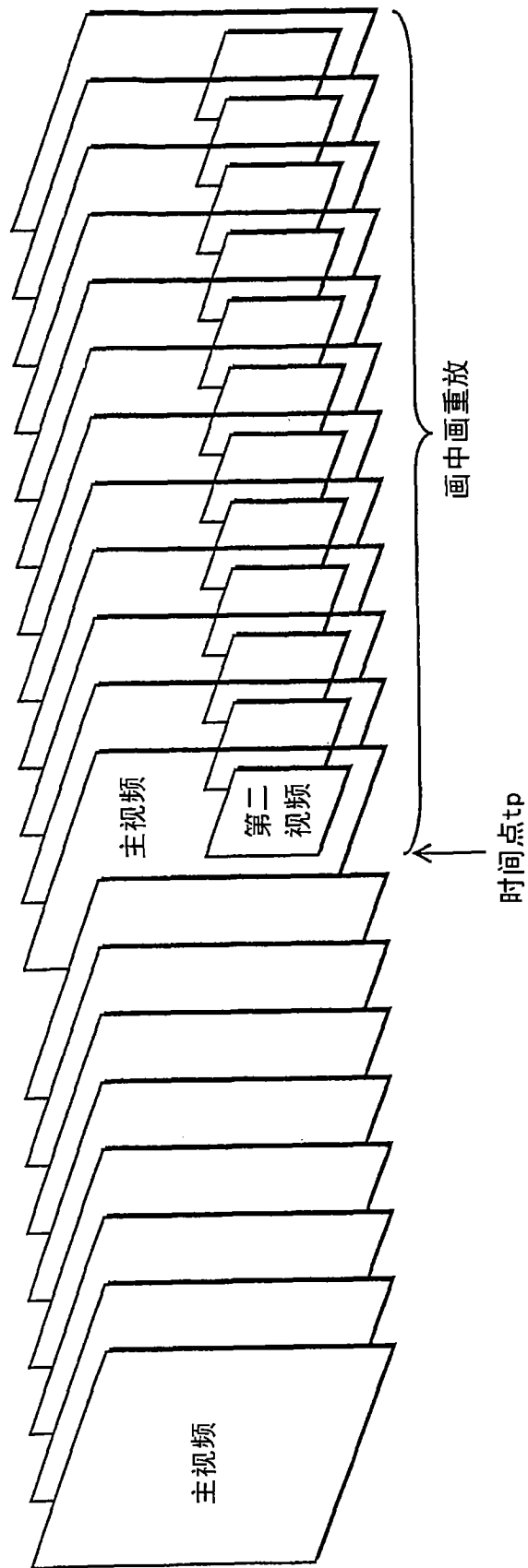


图 8

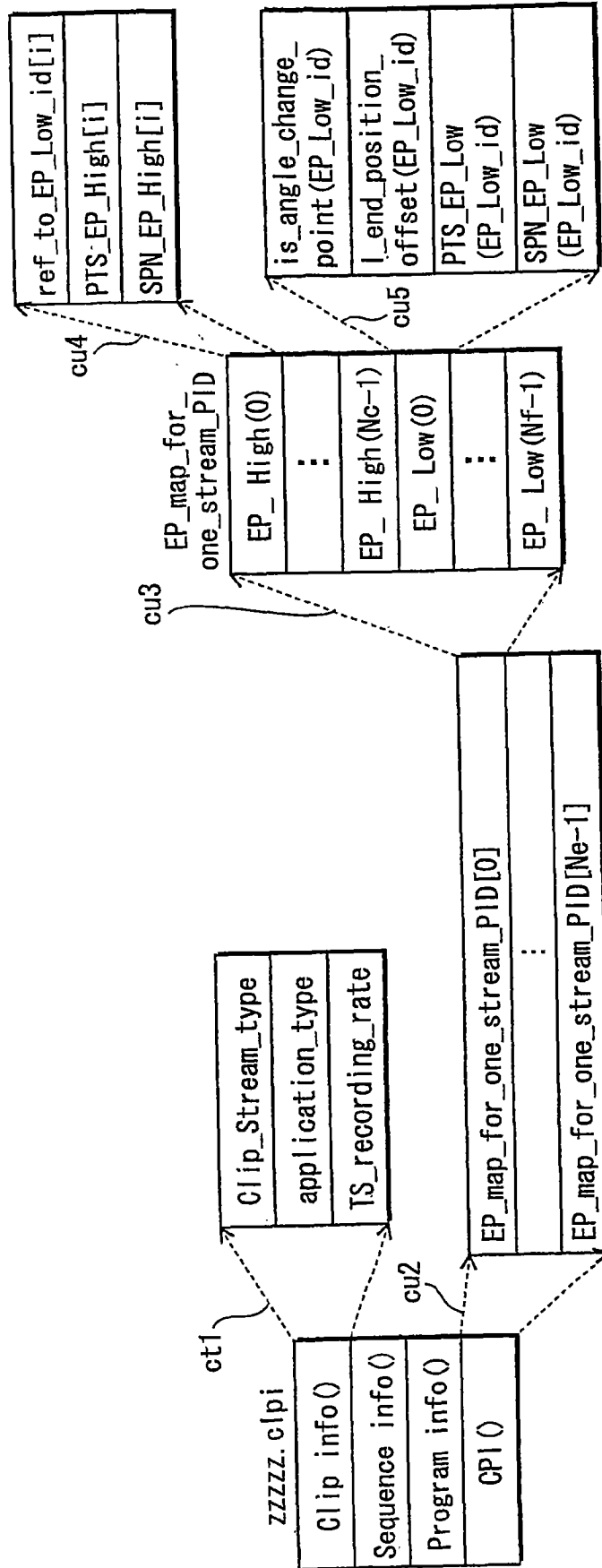
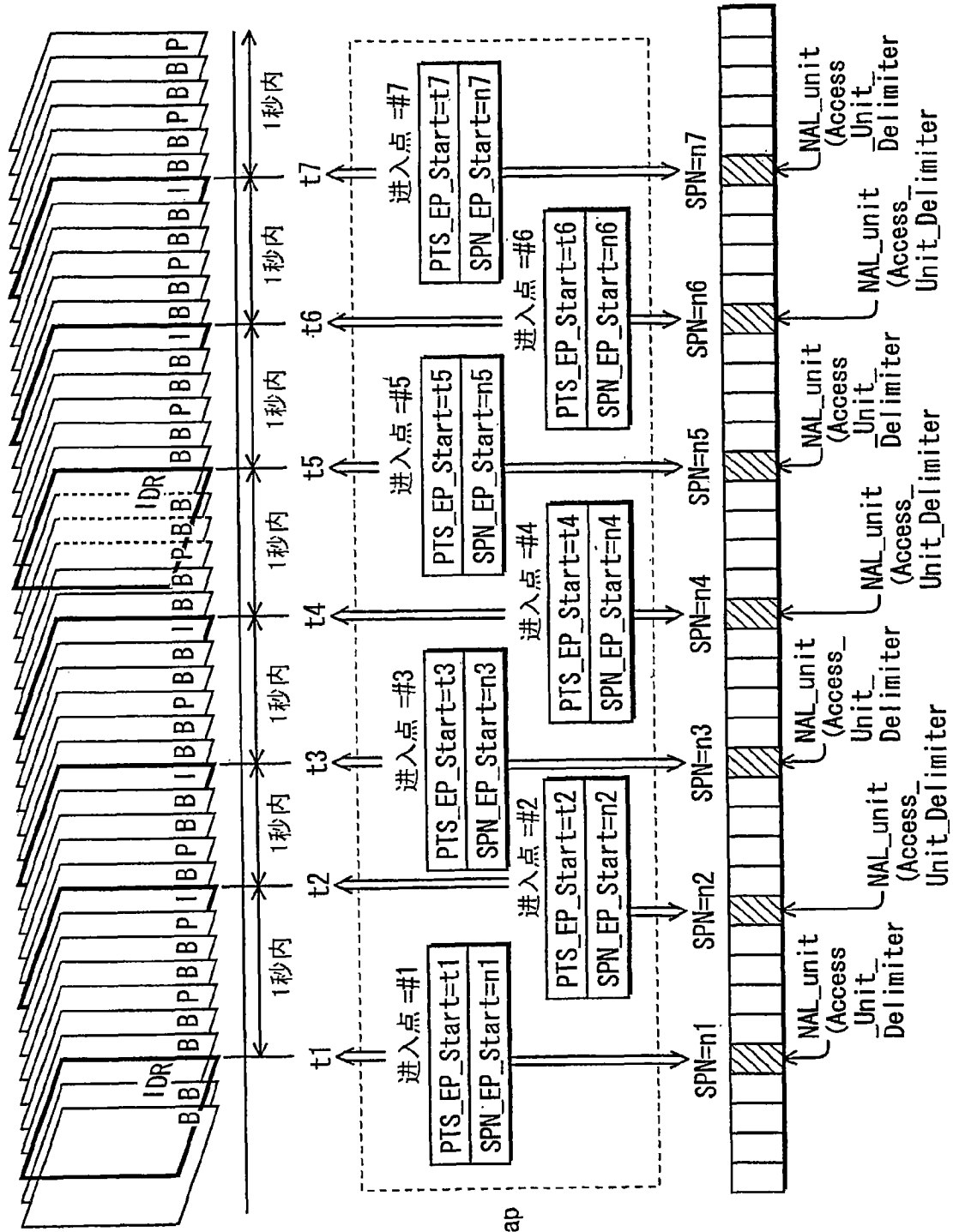


图 9



层次1:
按照显示次序的
画面序列

层次2:
主片段时间轴

层次3:
EP_map

层次4:
源分组序列

图 10

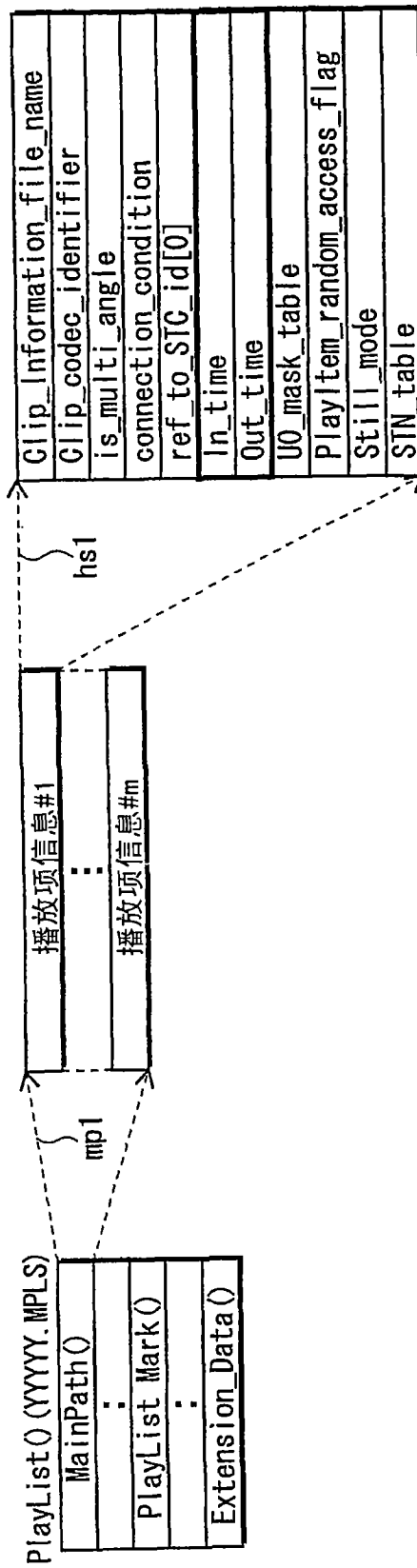


图 11

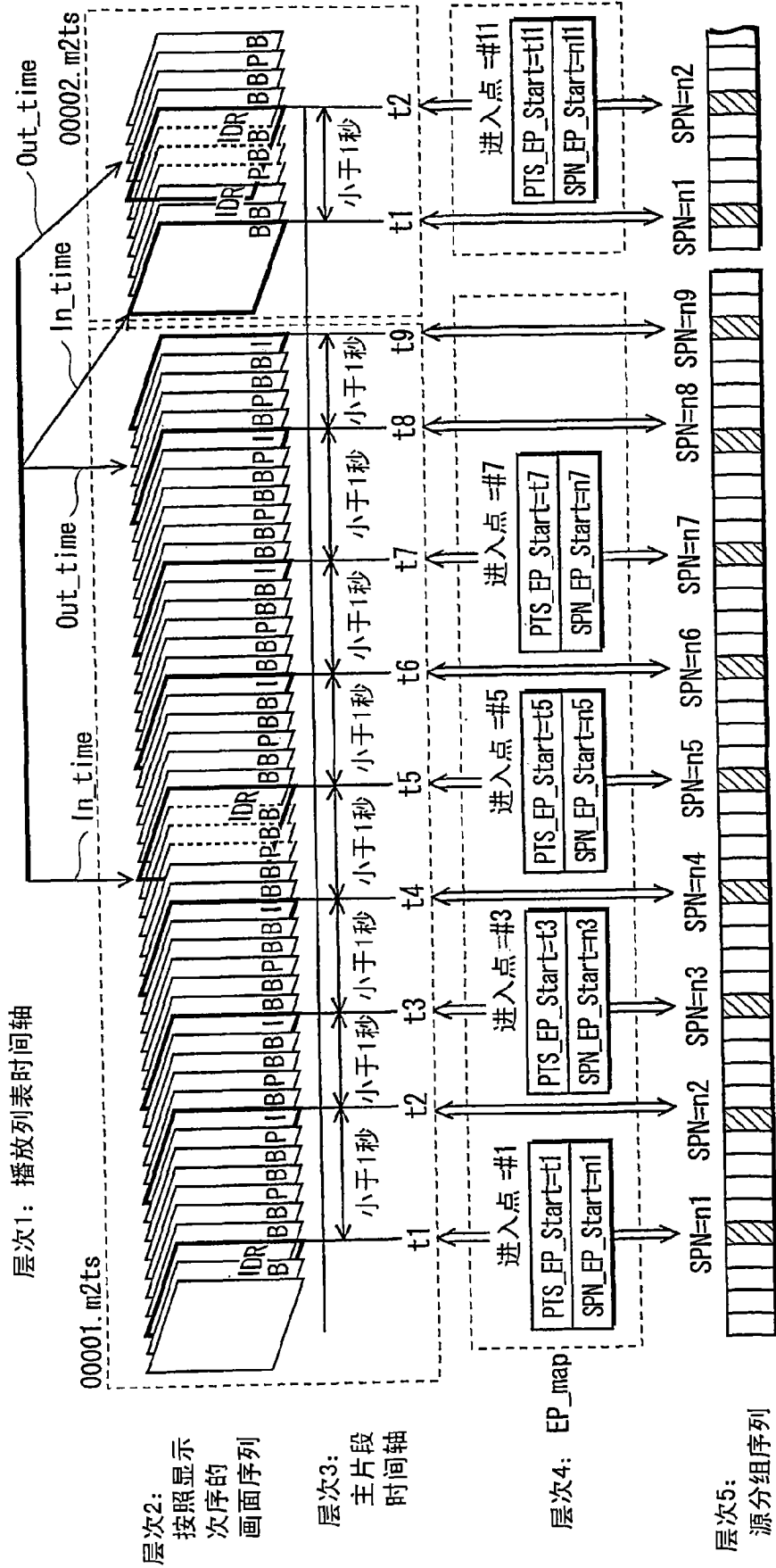


图 12

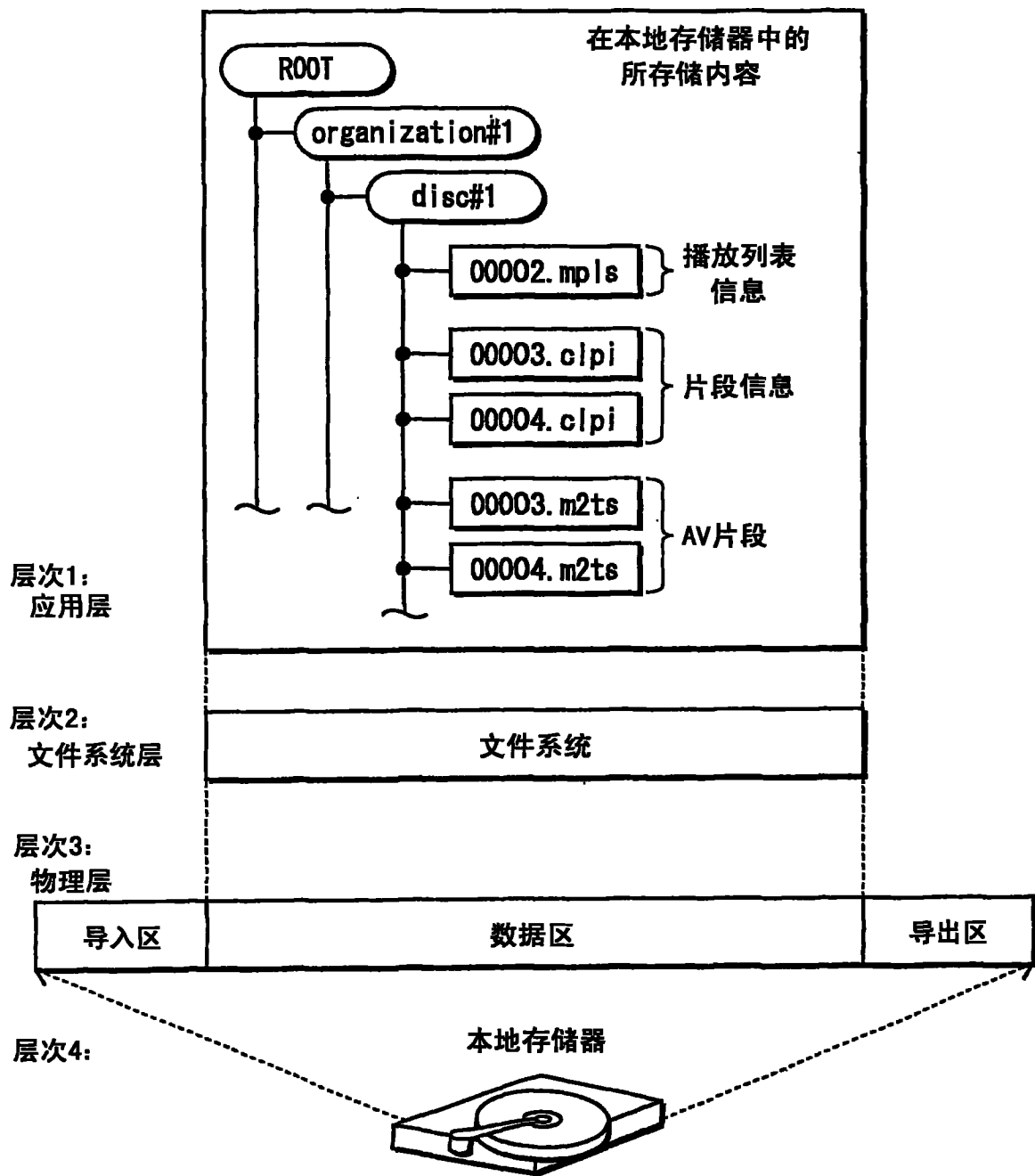


图 13

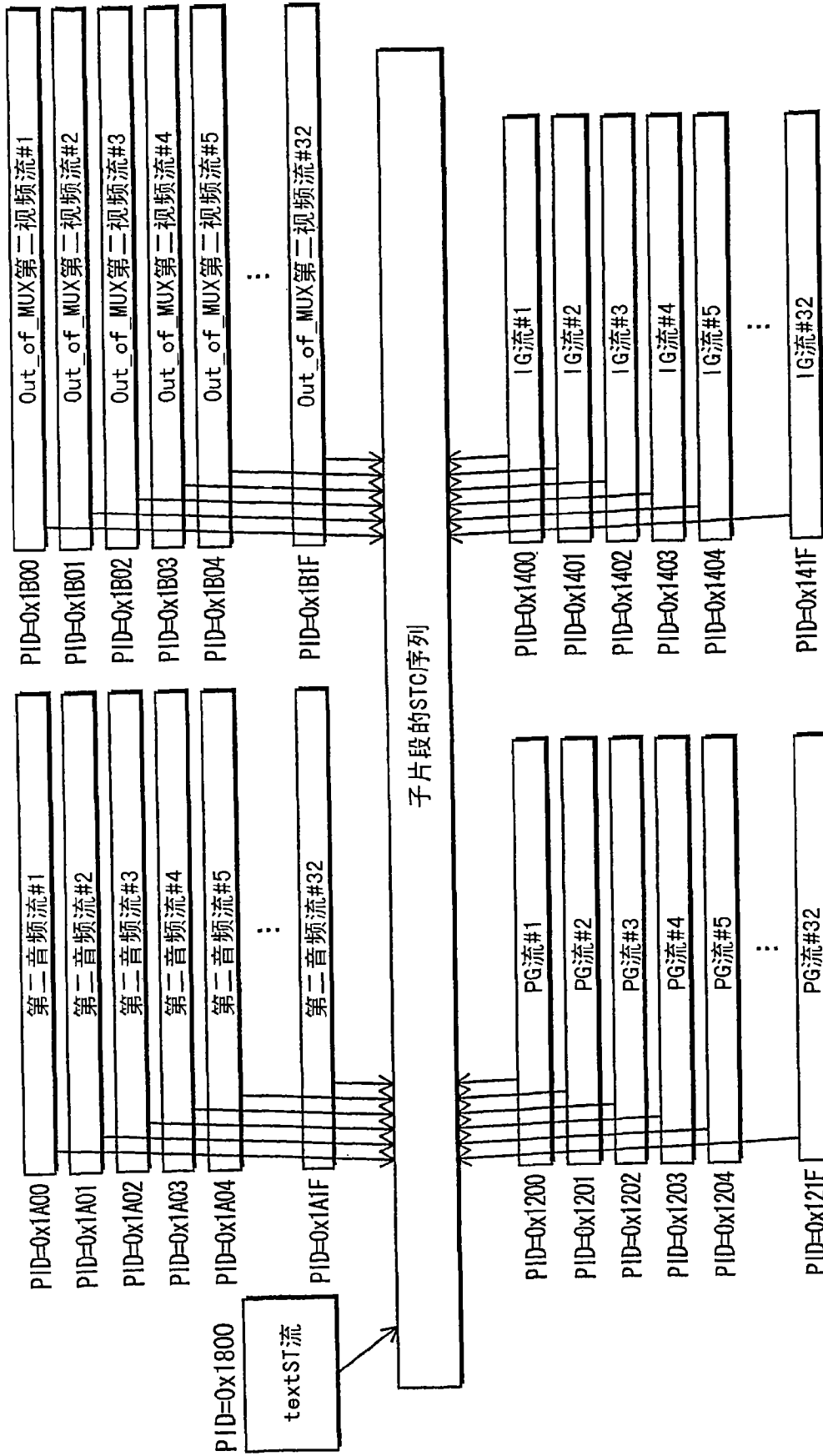


图 14

PID的范围	分配给该范围的流
0x0100	program_map
0x1001	PCR
0x1200~0x121F	PG流
0x1400~0x141F	IG流
0x1800	textST流
0x1A00~0x1A1F	第二音频流
0x1B00~0x1B1F	第二视频流

图 15

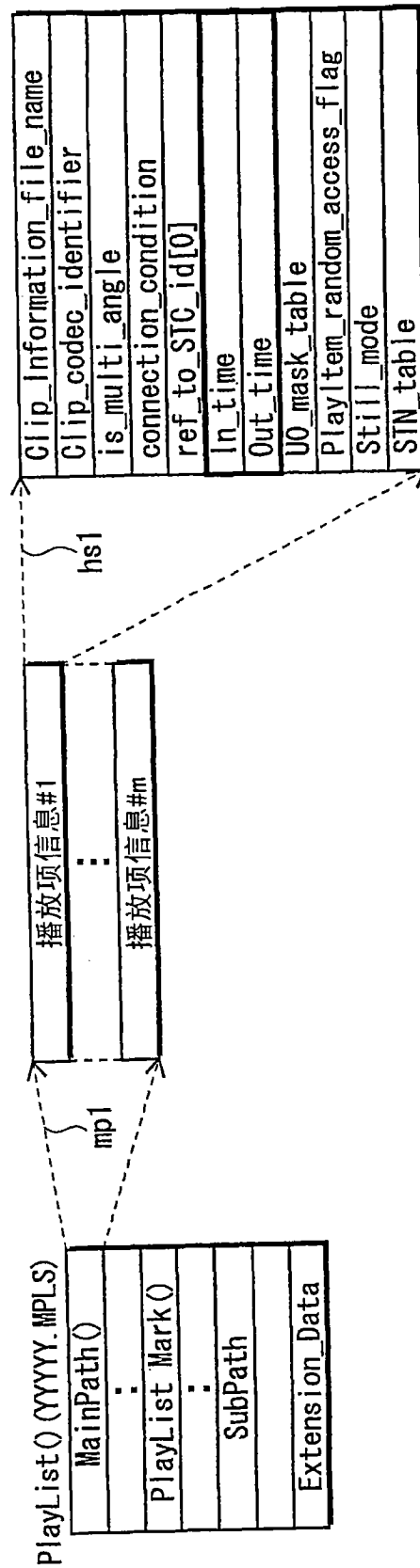


图 16

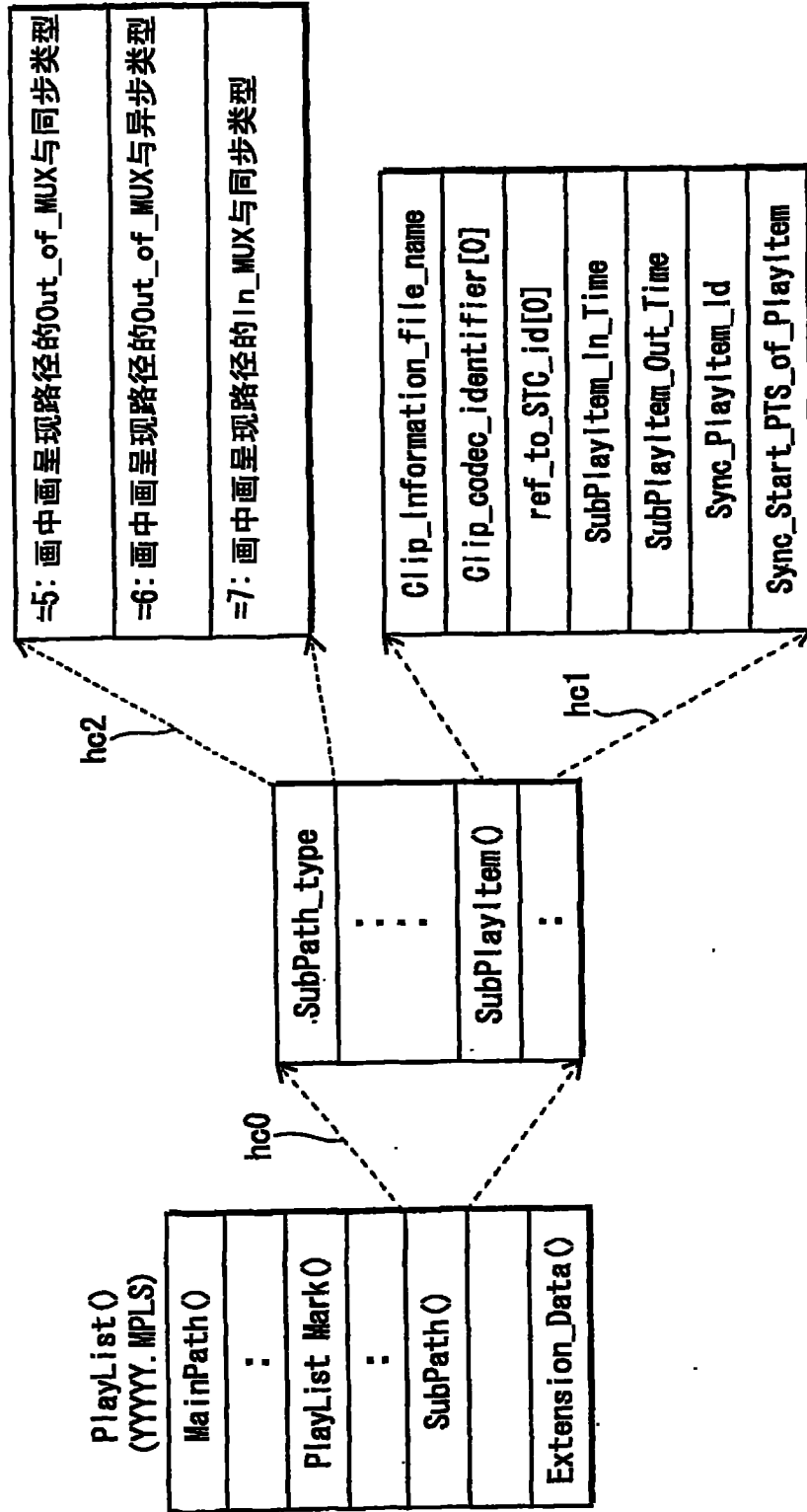


图 17

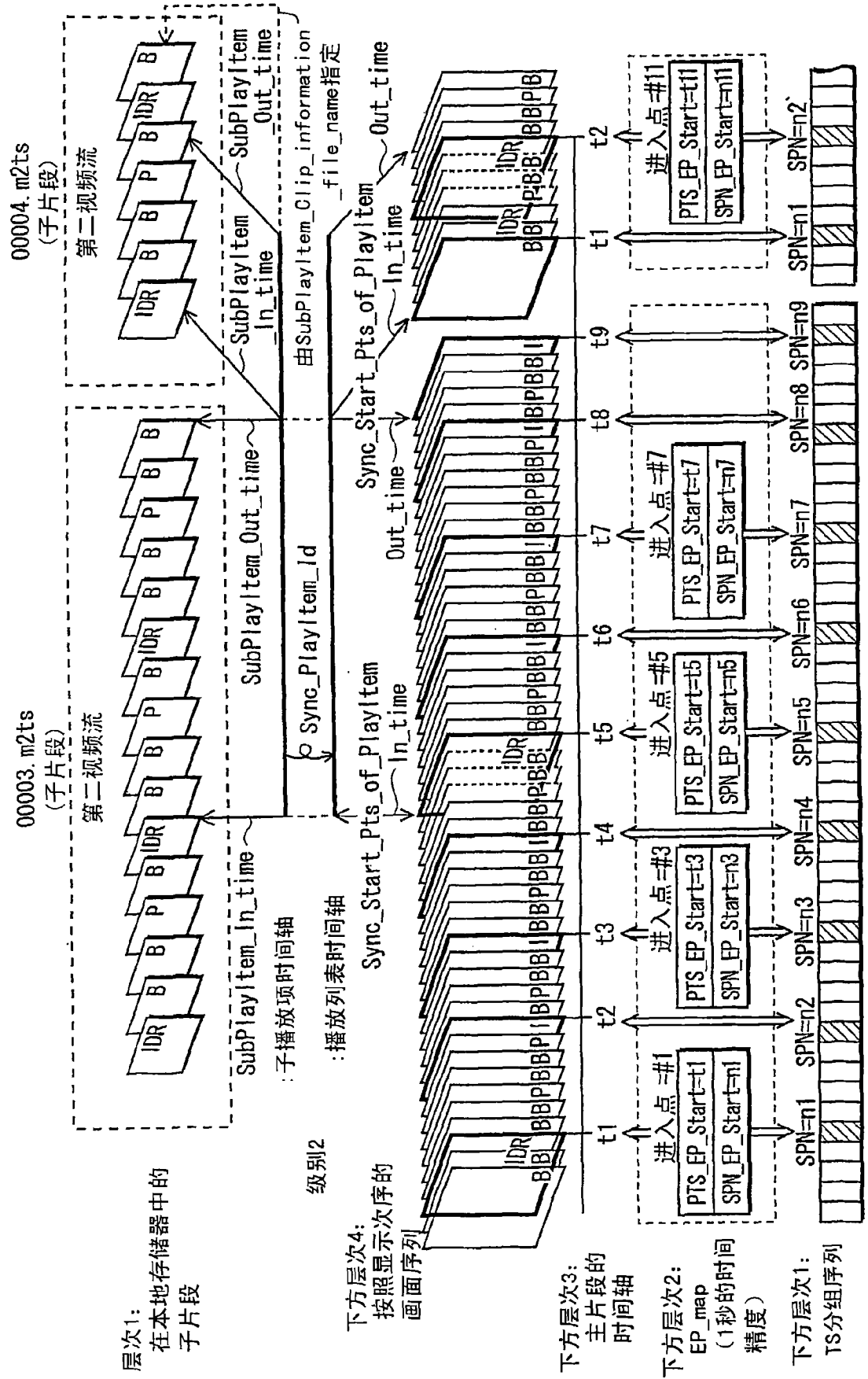


图 18

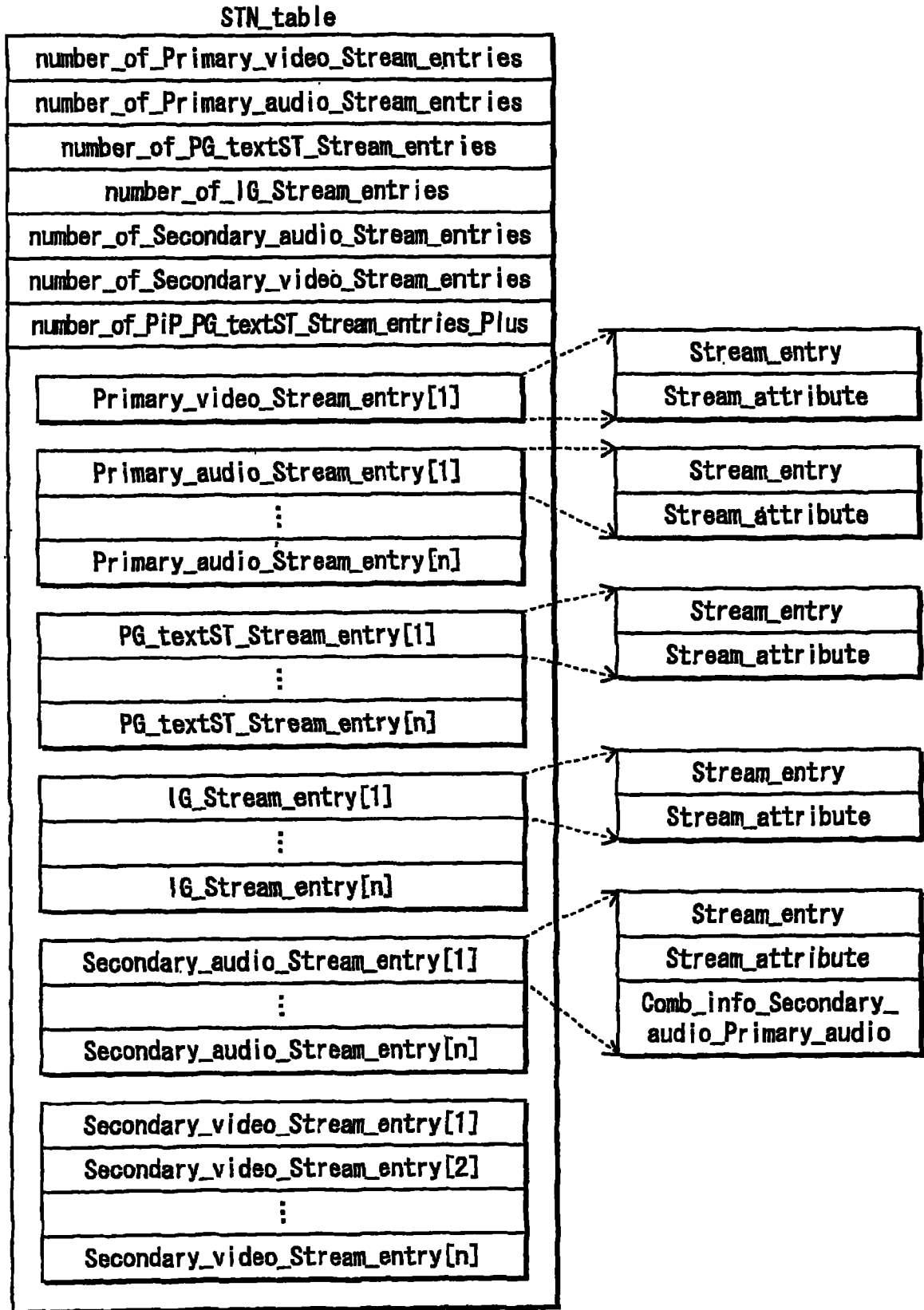


图 19

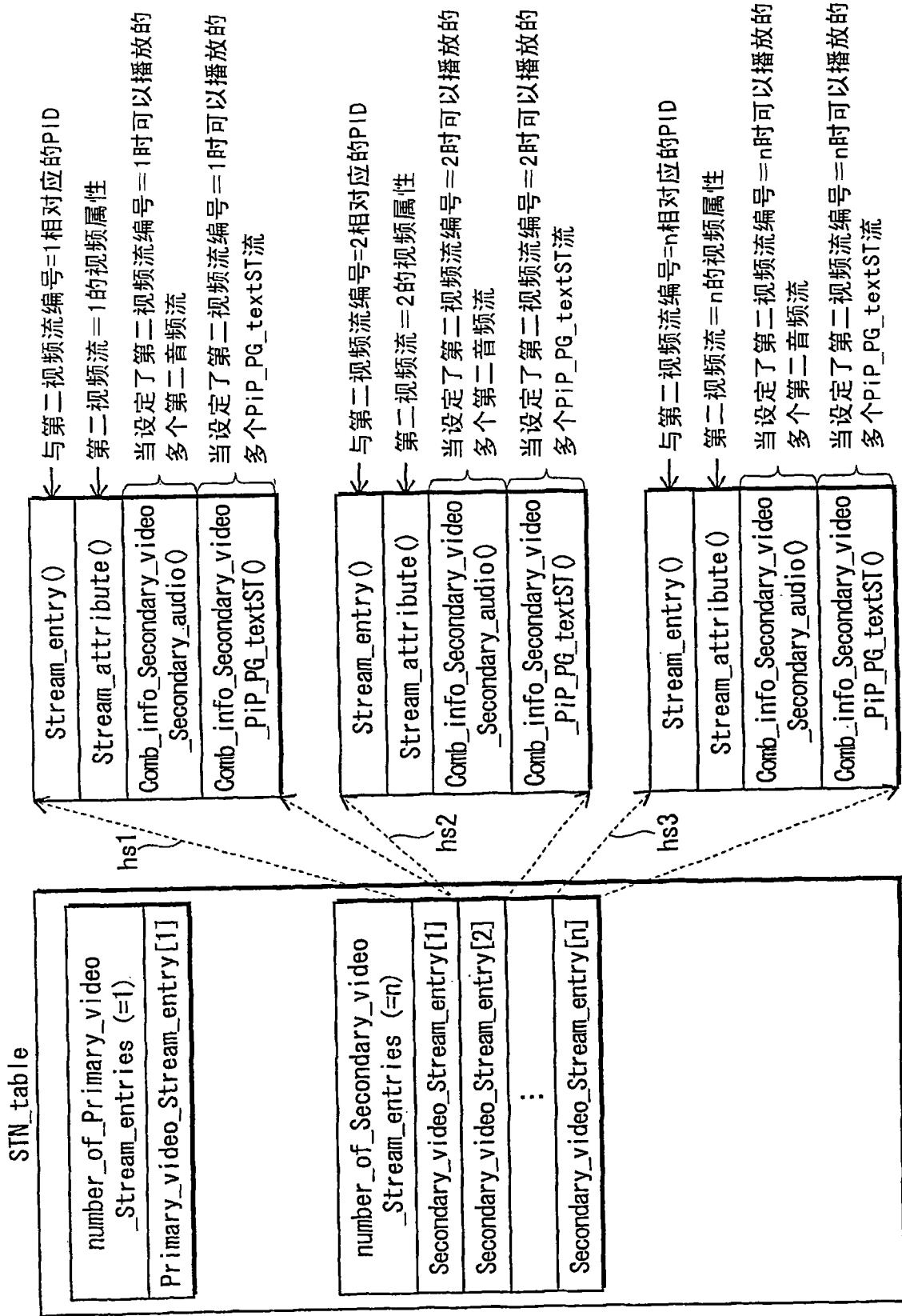


图 20

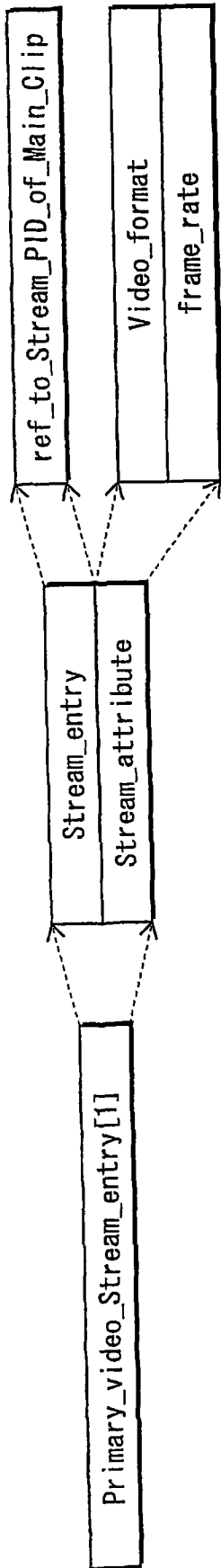


图 21A

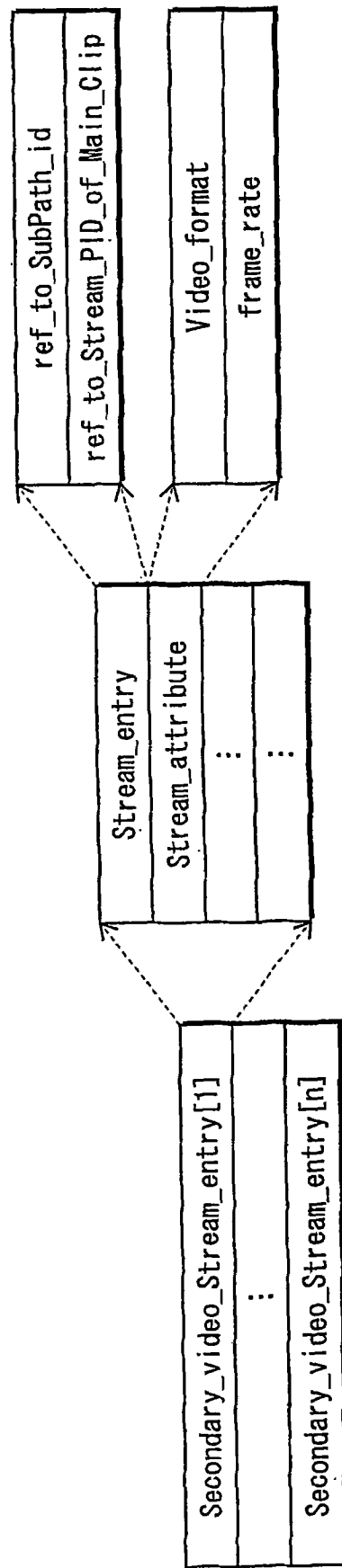


图 21B

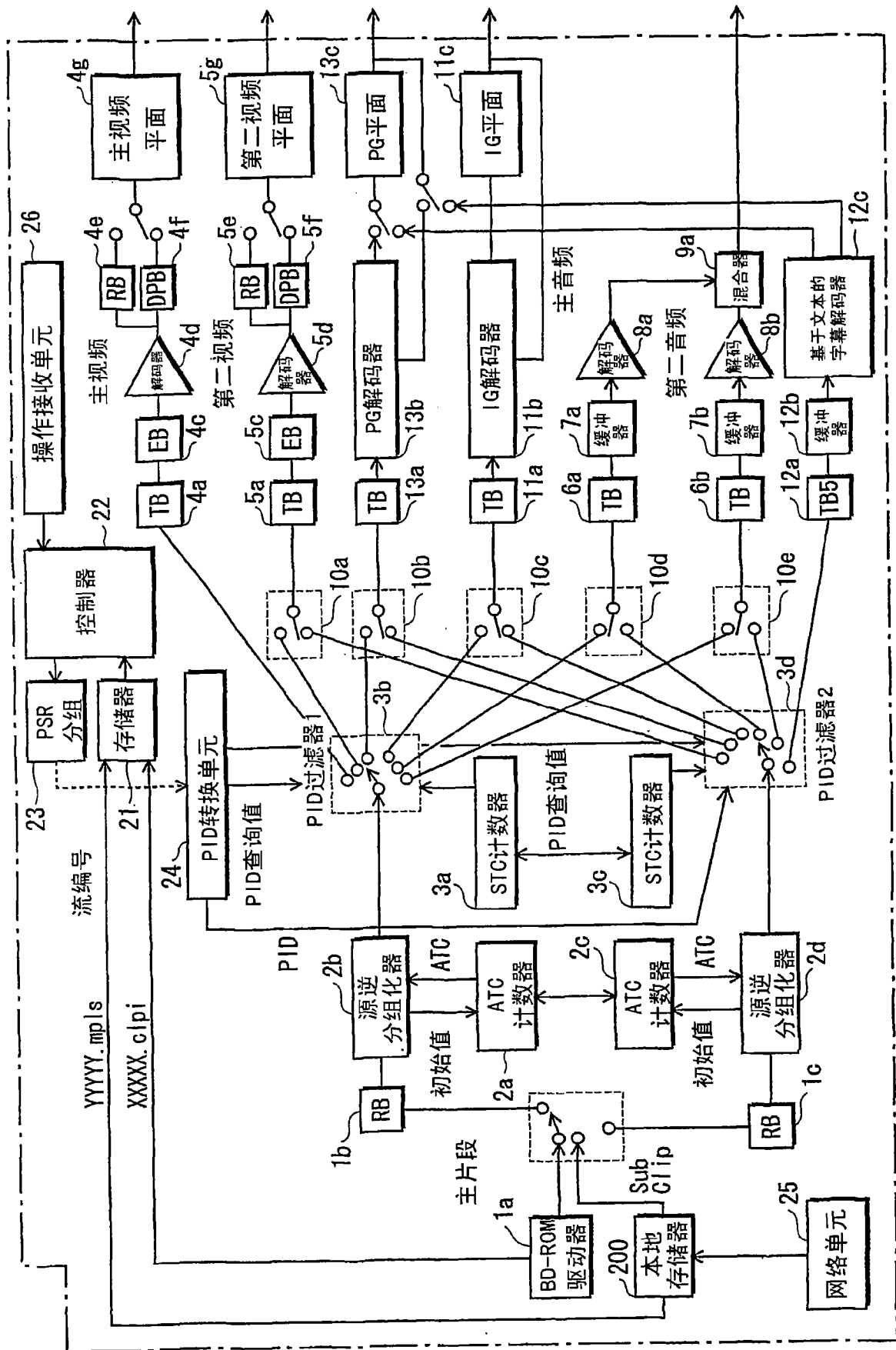


图 22

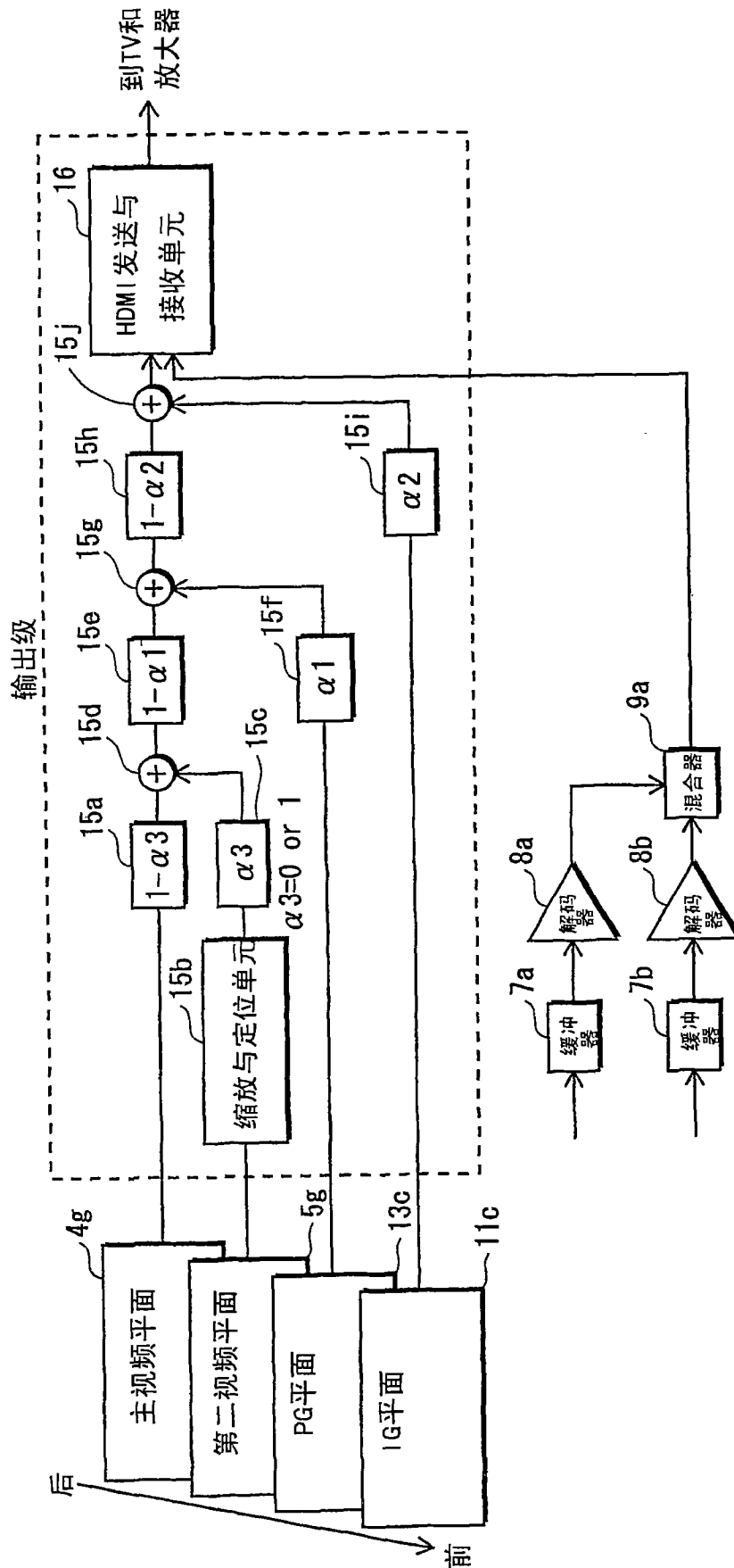


图 23

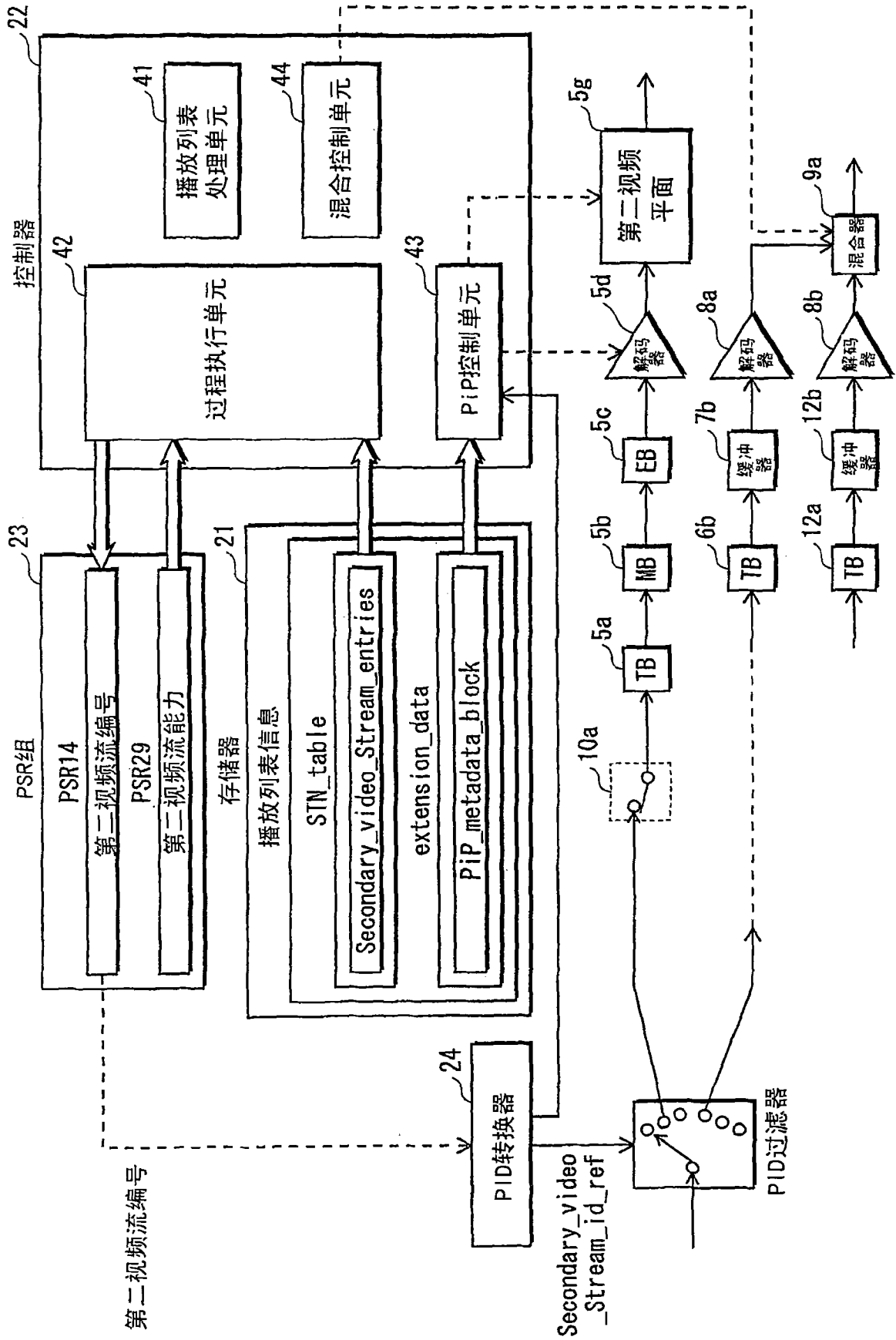
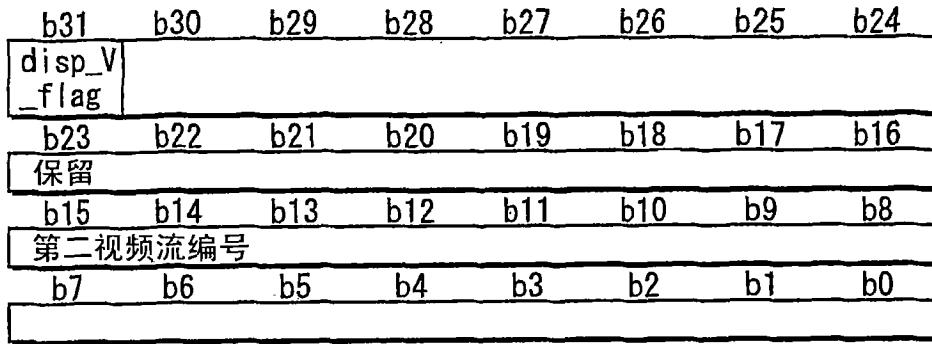


图 24

PSR14: 第二视频流编号



disp_V_flag ... 0b: 不能进行第二视频的呈现
1b: 能够进行第二视频的呈现

第二视频流

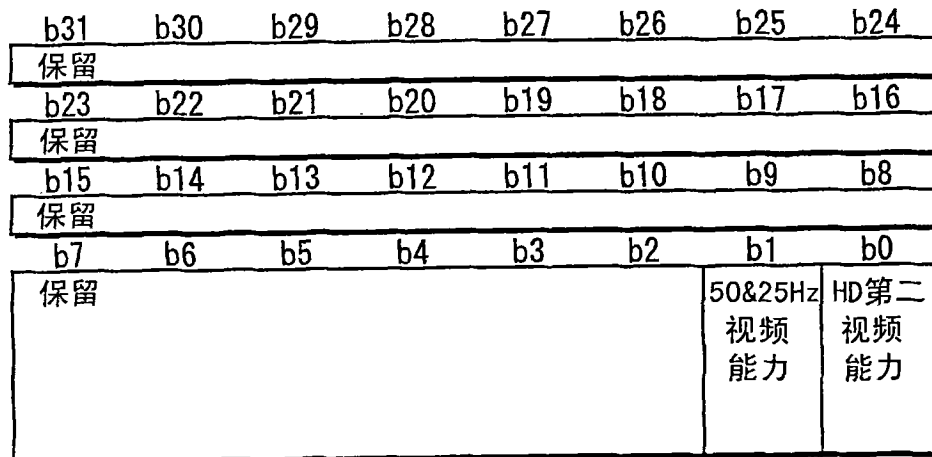
编号 ... 0: 保留
1-32: 第二视频流编号
0xFF: 没有选择第二视频流或者没有第二视频流
其他: 保留

图 25A

PSR29: 对于视频的播放器能力

PSR29寄存器表示播放器的视频能力。

导航命令和BD-J应用程序不能直接改变PSR29中包含的值



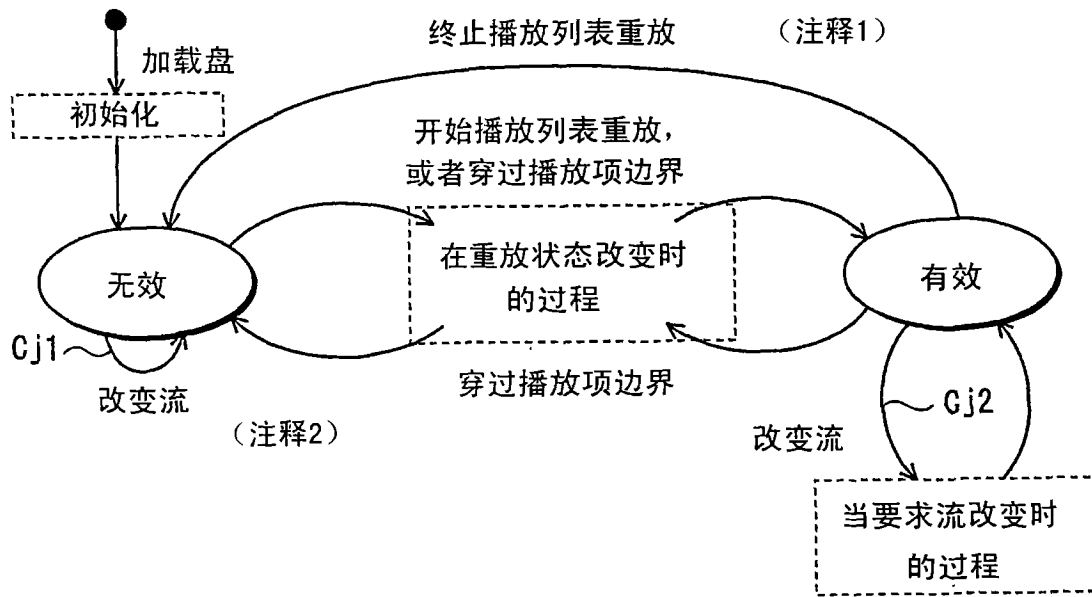
HD第二视频能力

... 0b: 不能播放HD第二视频
1b: 能够播放HD第二视频

HD50&25Hz

视频能力 ... 0b: 不能播放50&25Hz视频
1b: 能够播放50&25Hz视频

图 25B



注释1：当状态从“有效”变为“无效”时保持PSR的最后的值

注释2：当状态处于“无效”时，简单地覆写PSR的值

图 26

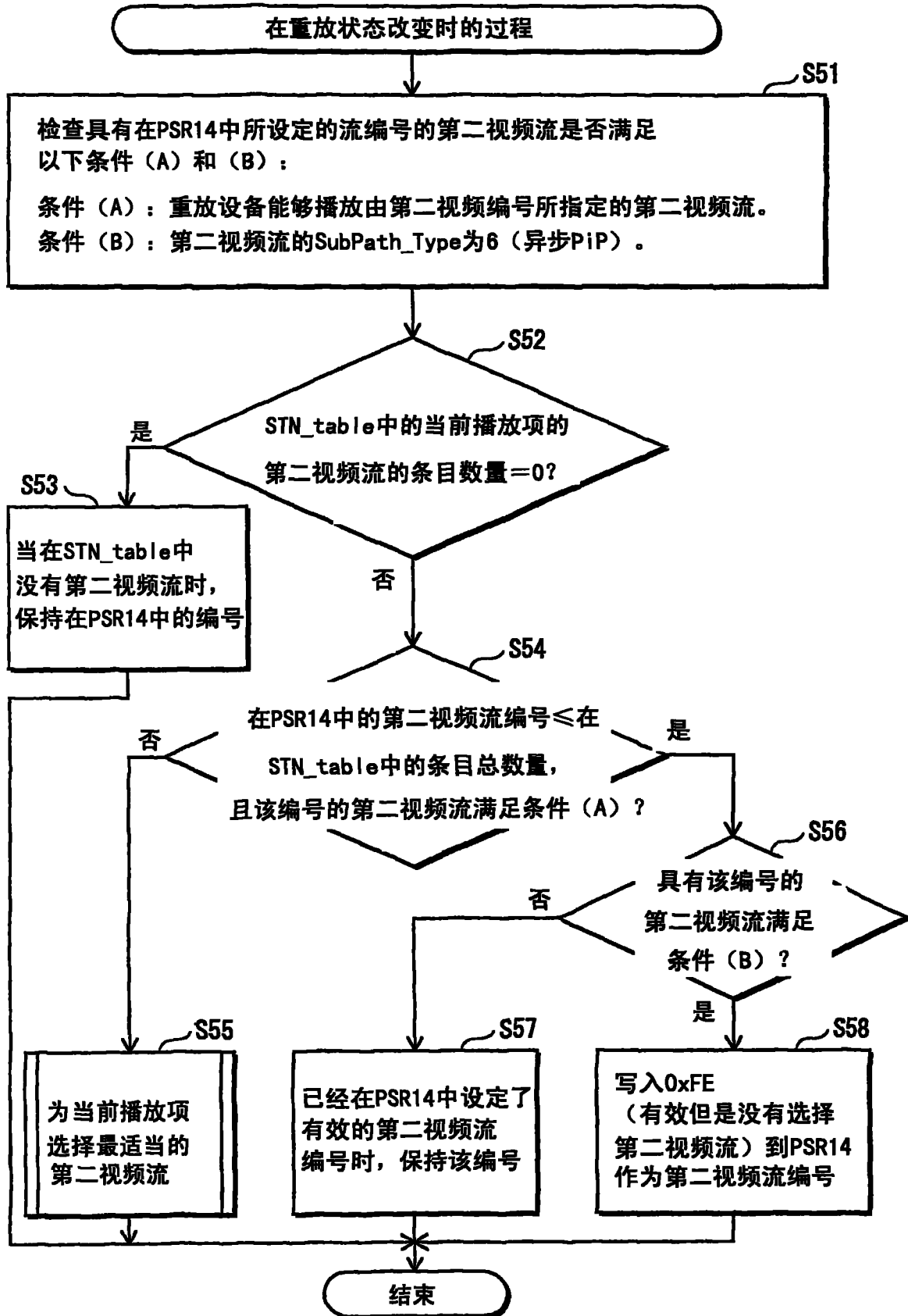


图 27

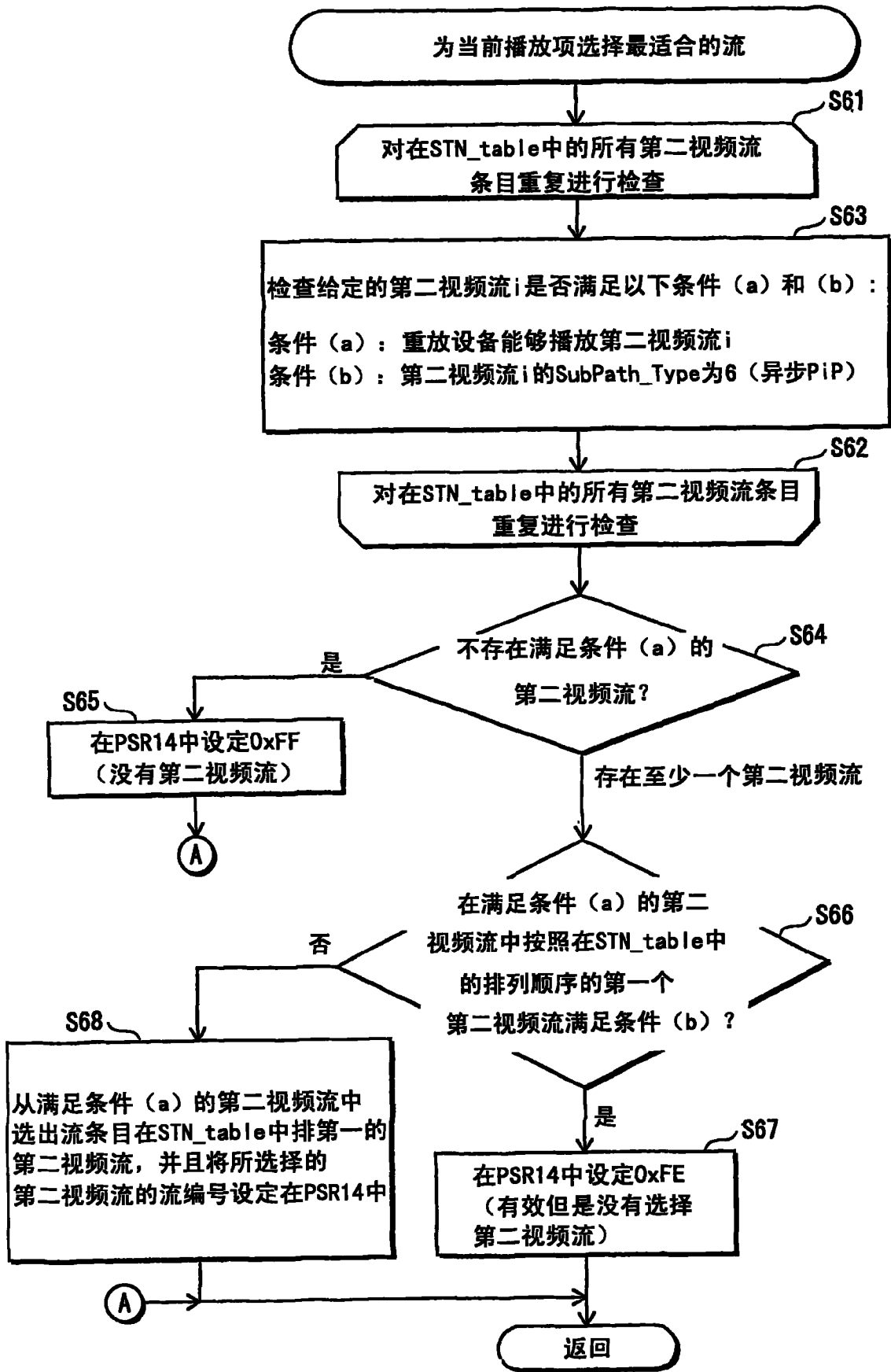


图 28

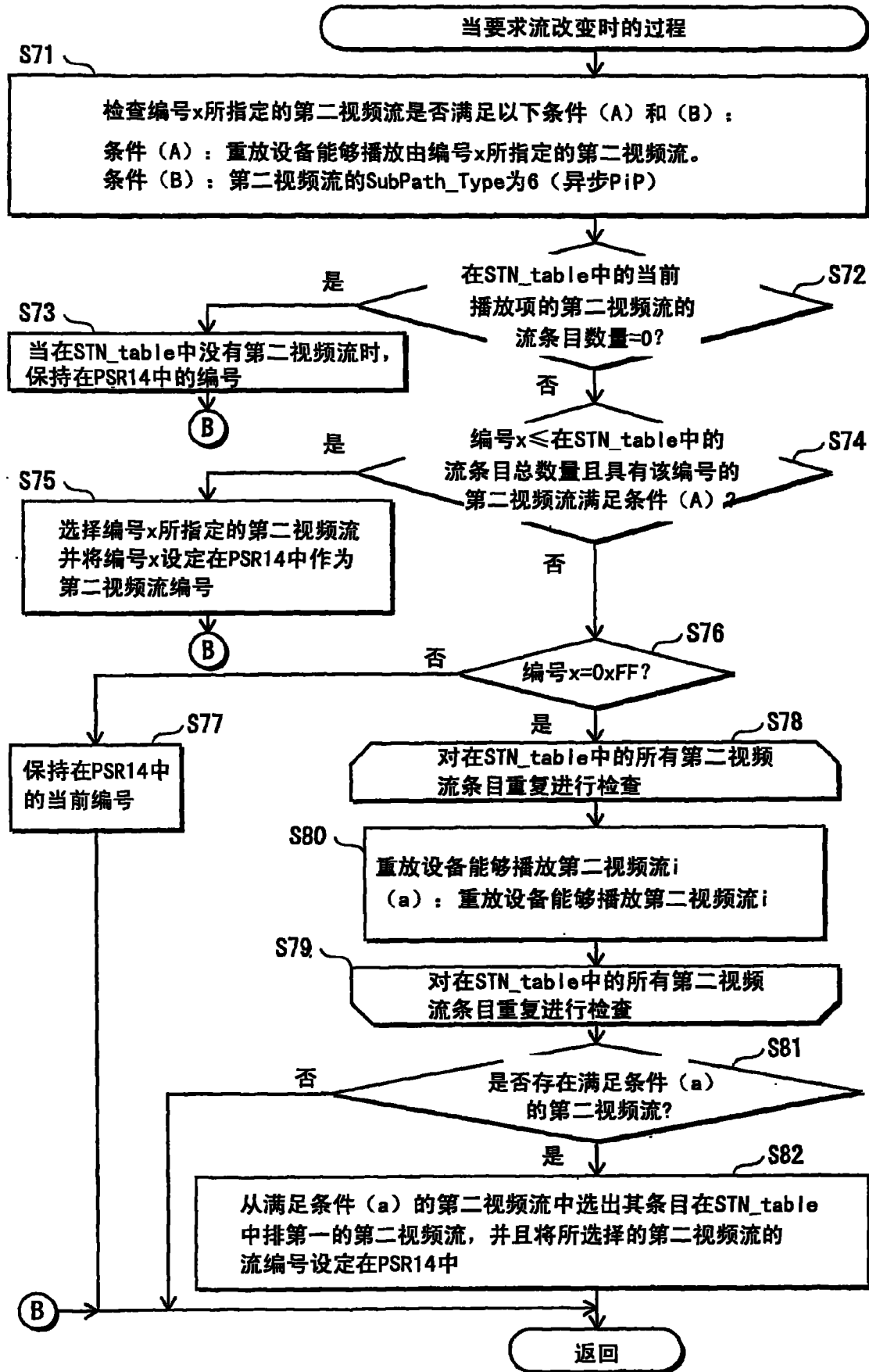


图 29

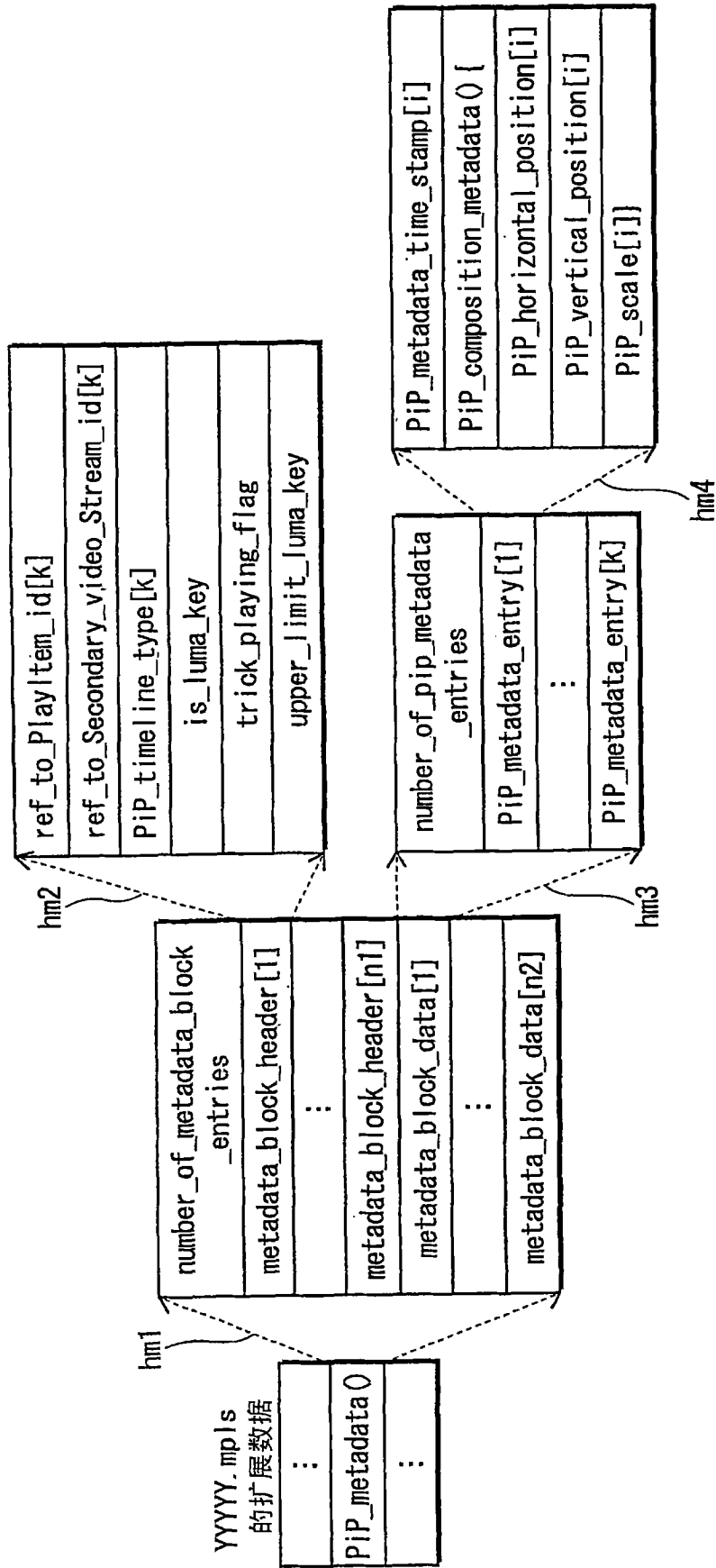


图 30

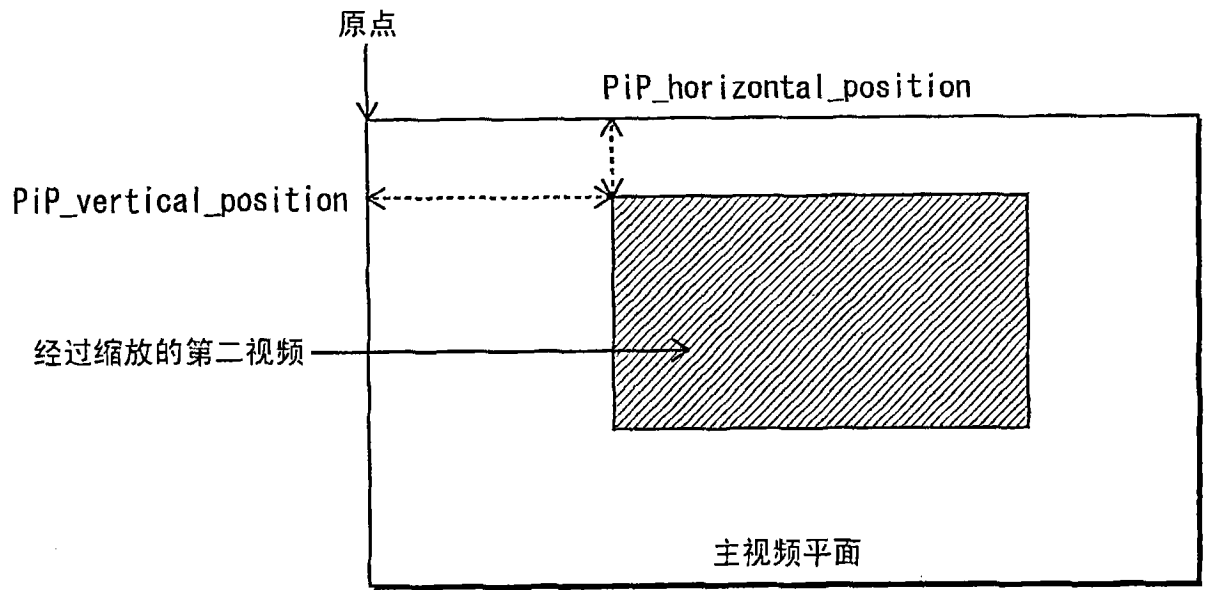


图 31

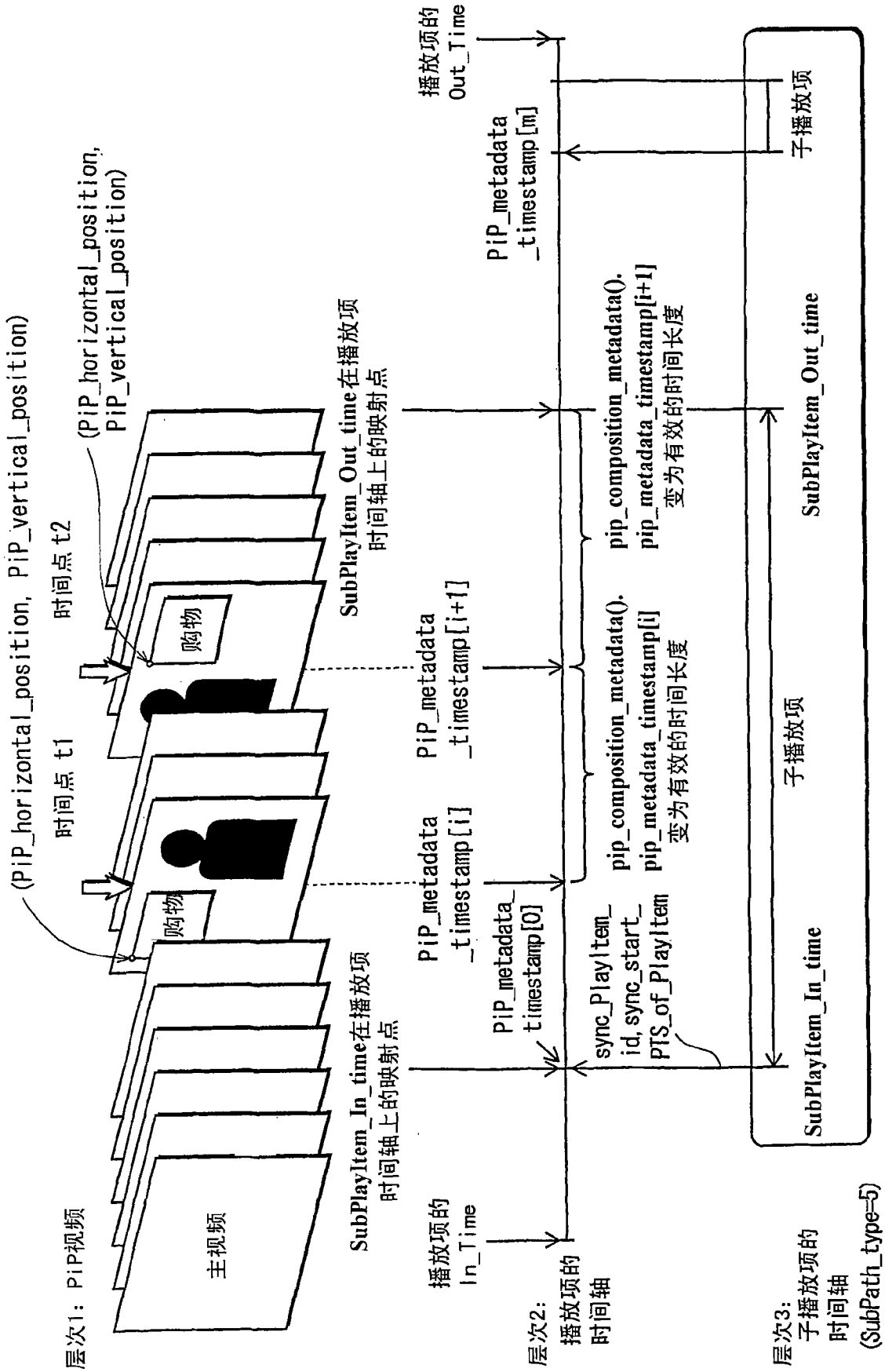


图 32

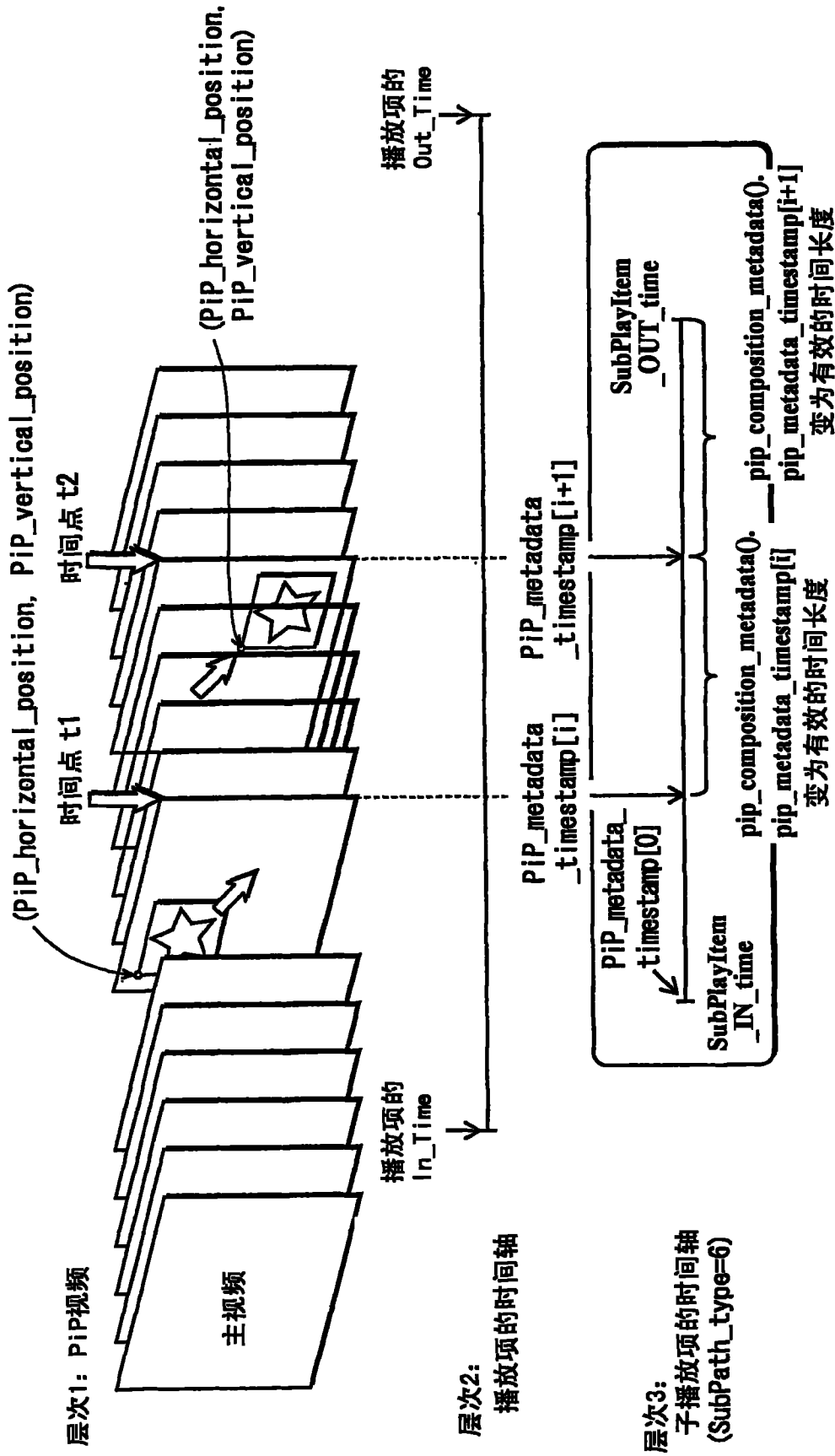


图 33

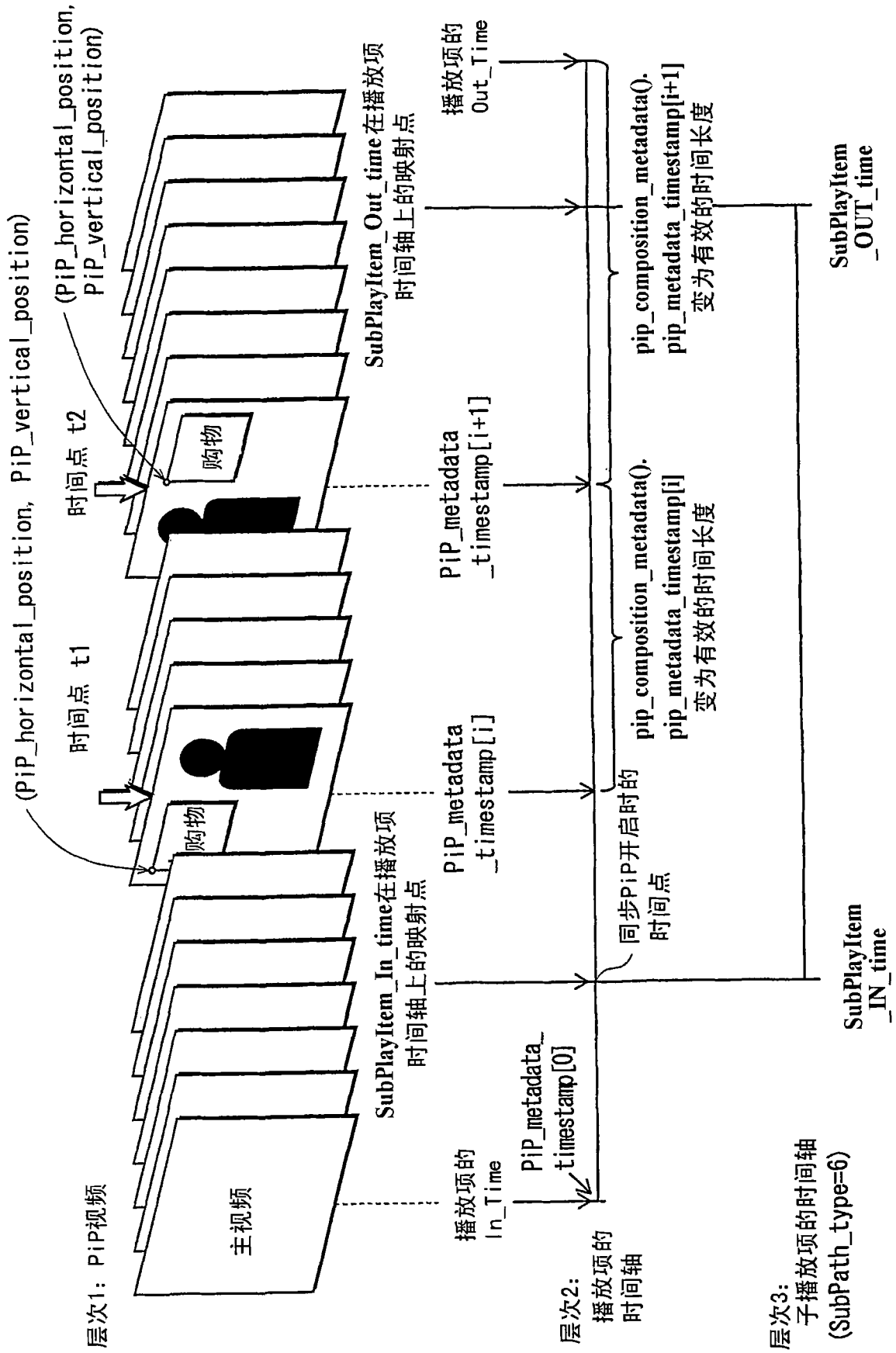


图 34

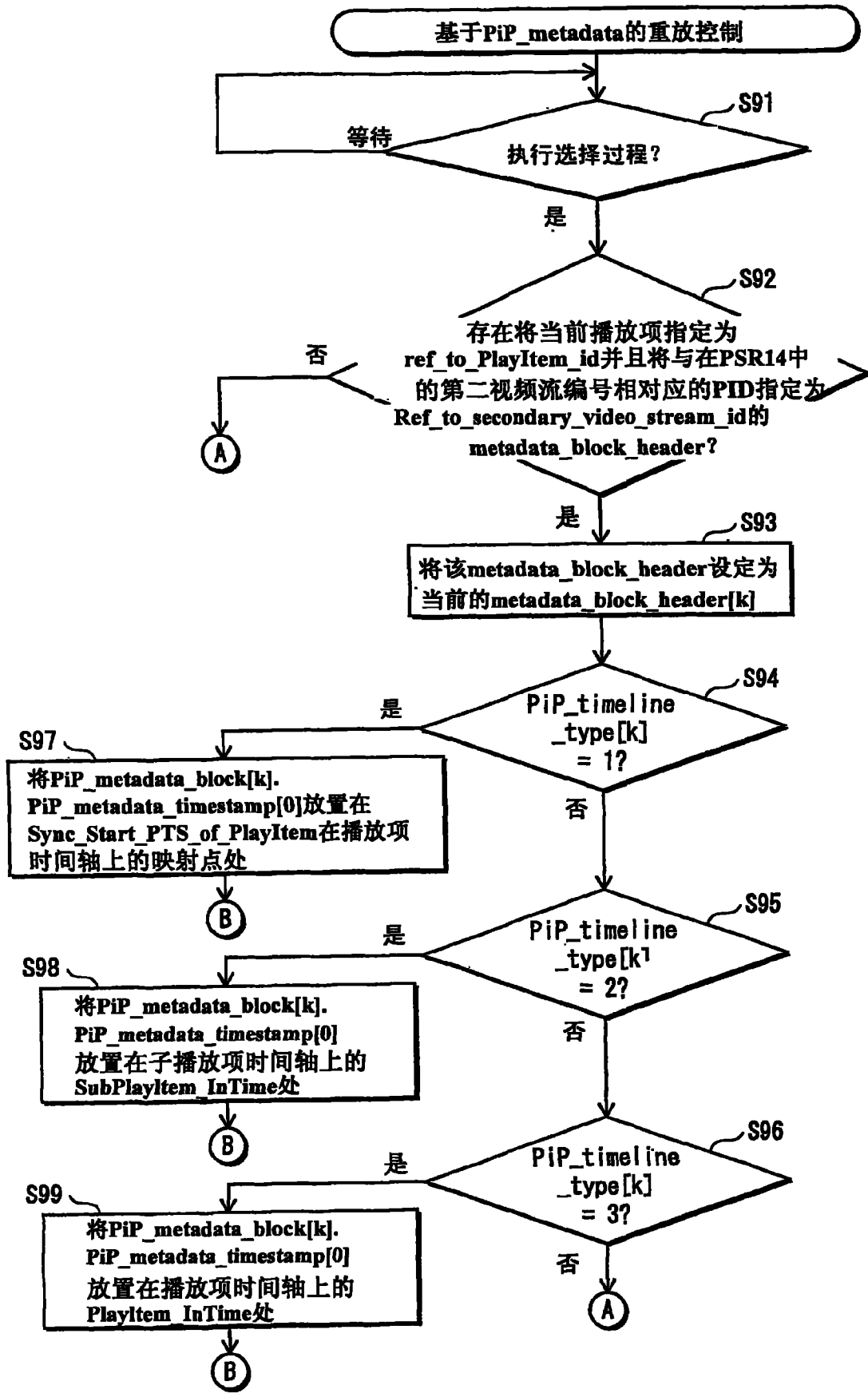


图 35

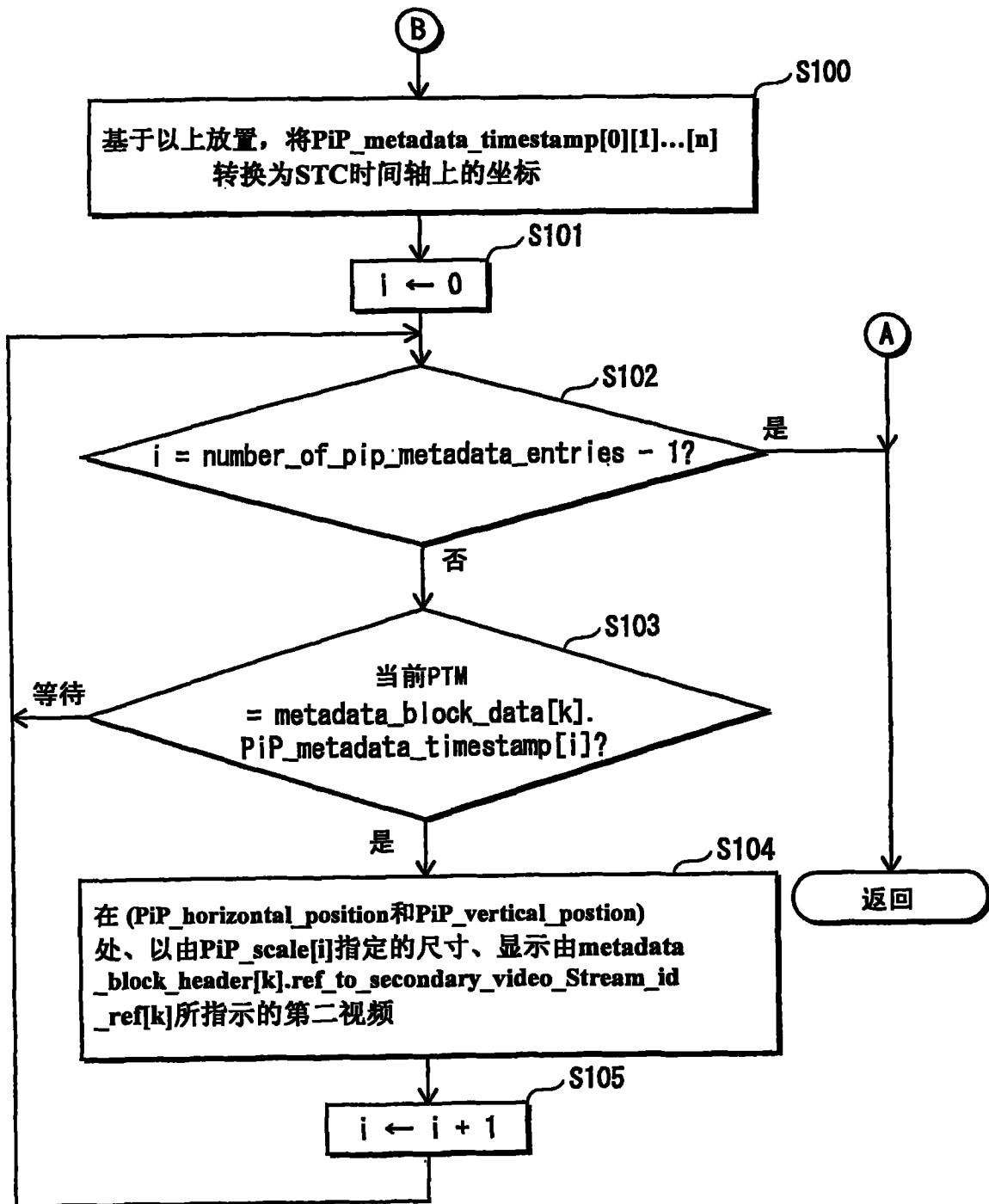
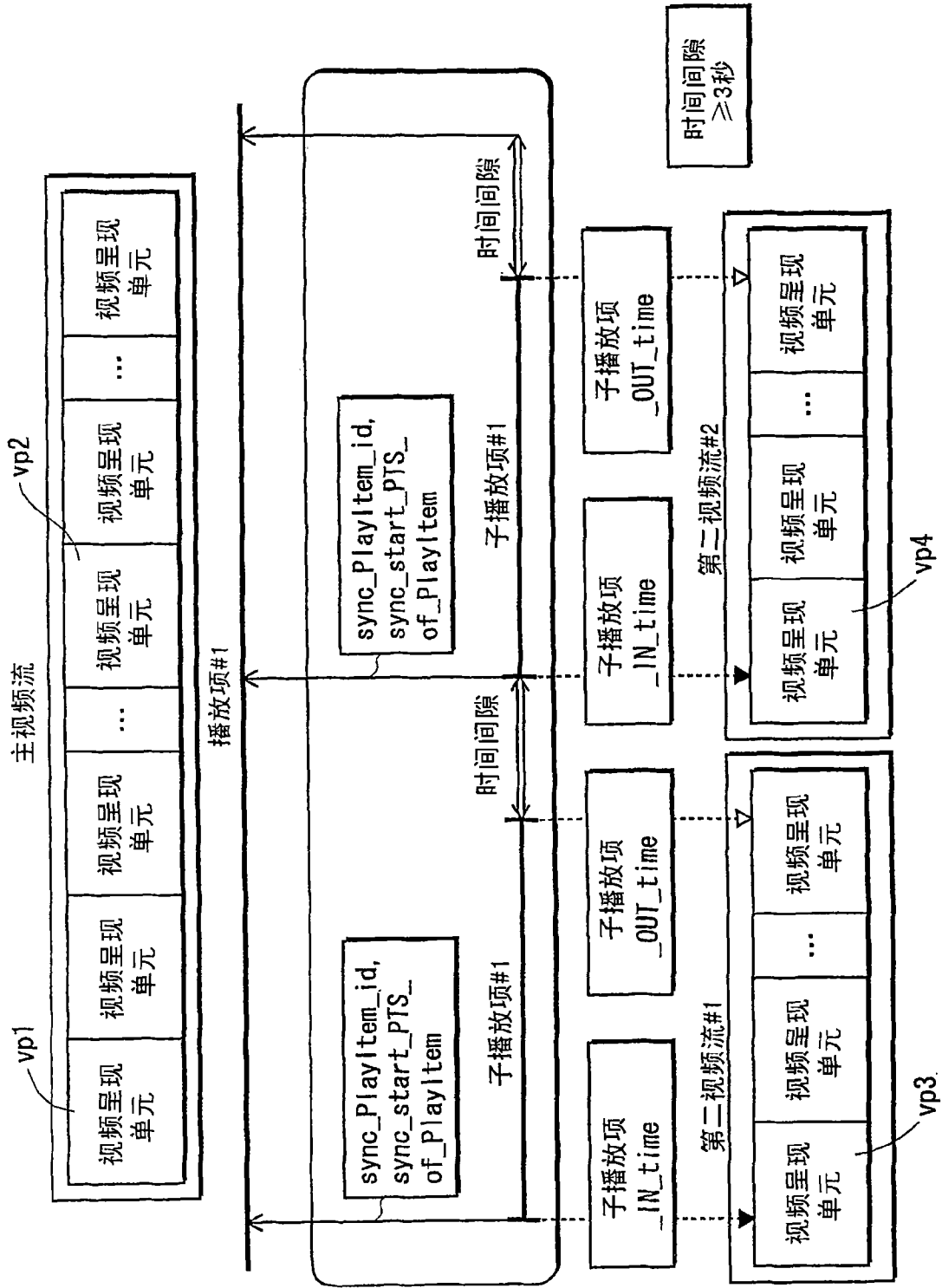


图 36



层次1:
主路径时间轴

层次2:
层次3:
子路径时间轴

层次4:

层次5:

图 37

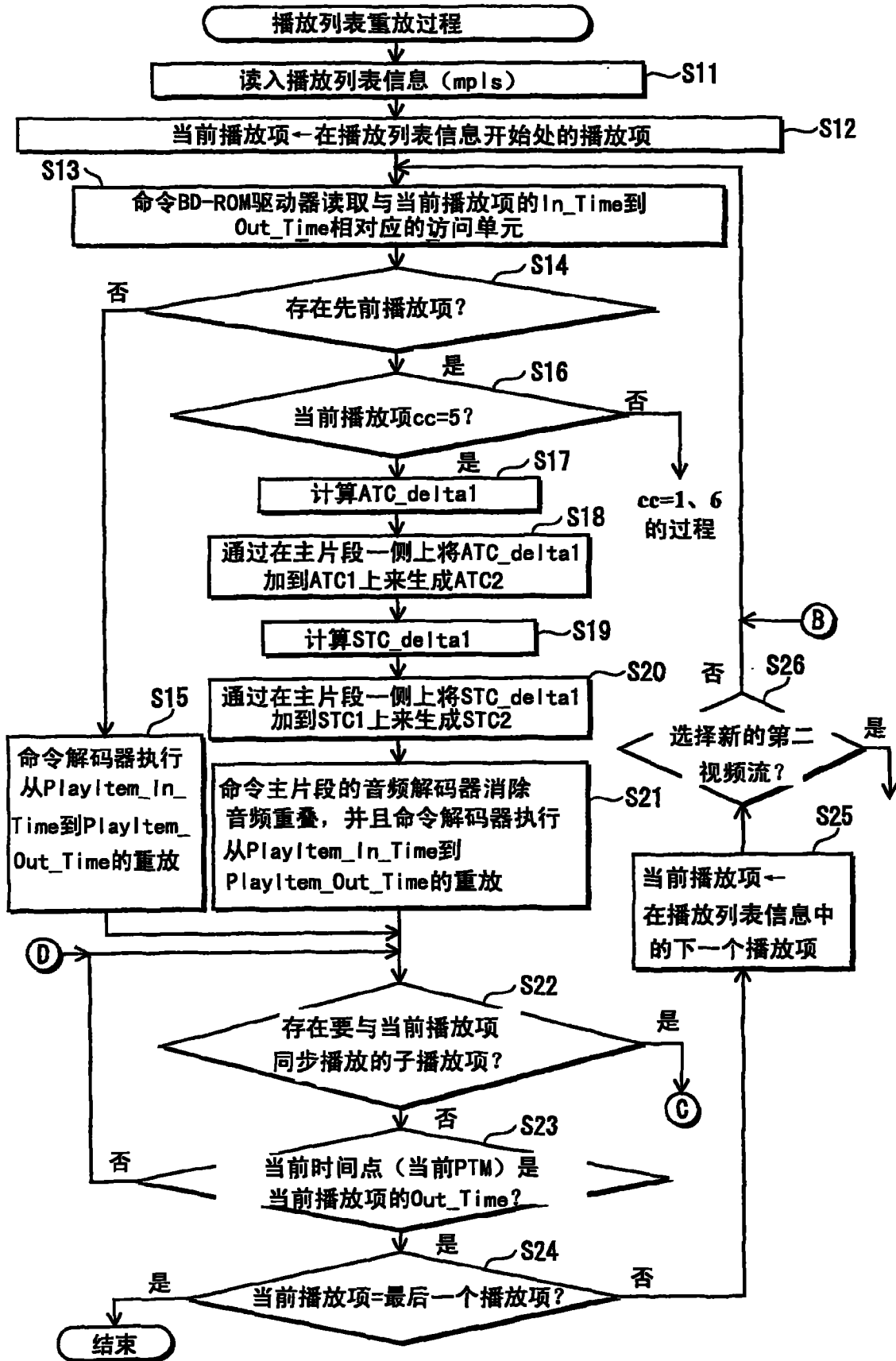


图 38

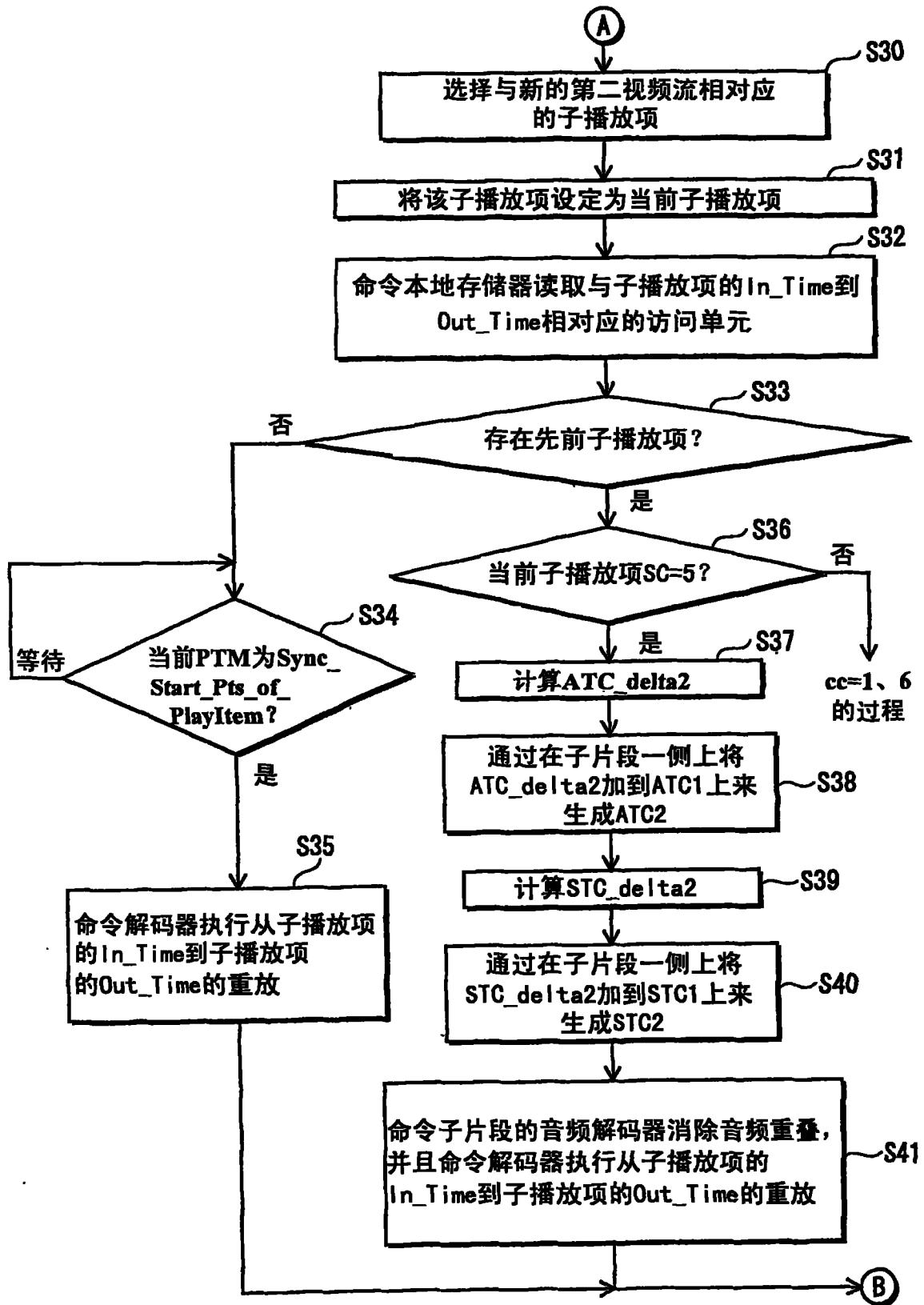


图 39

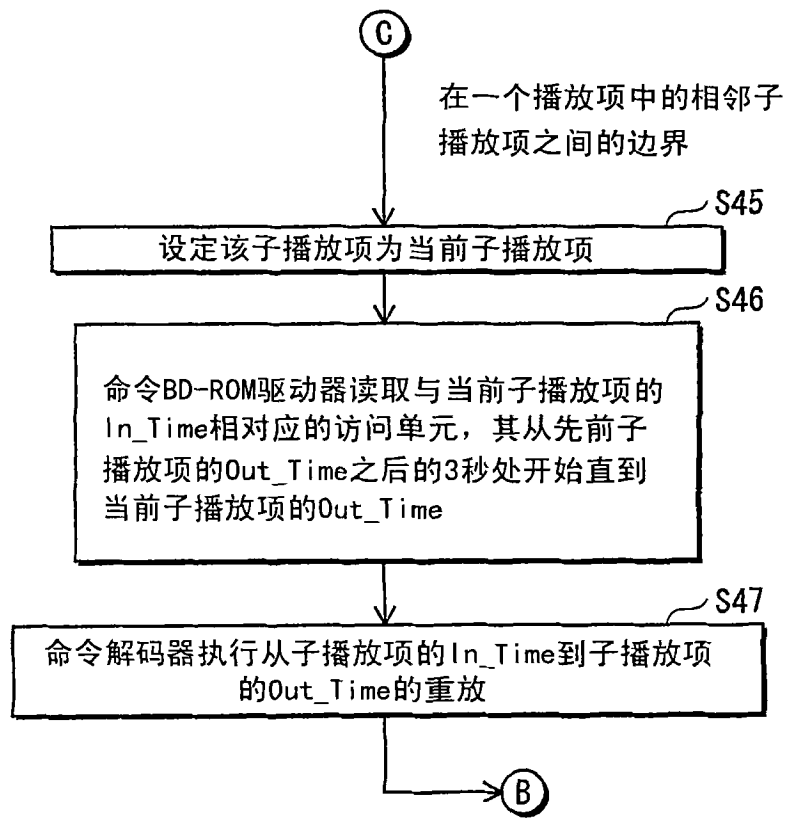
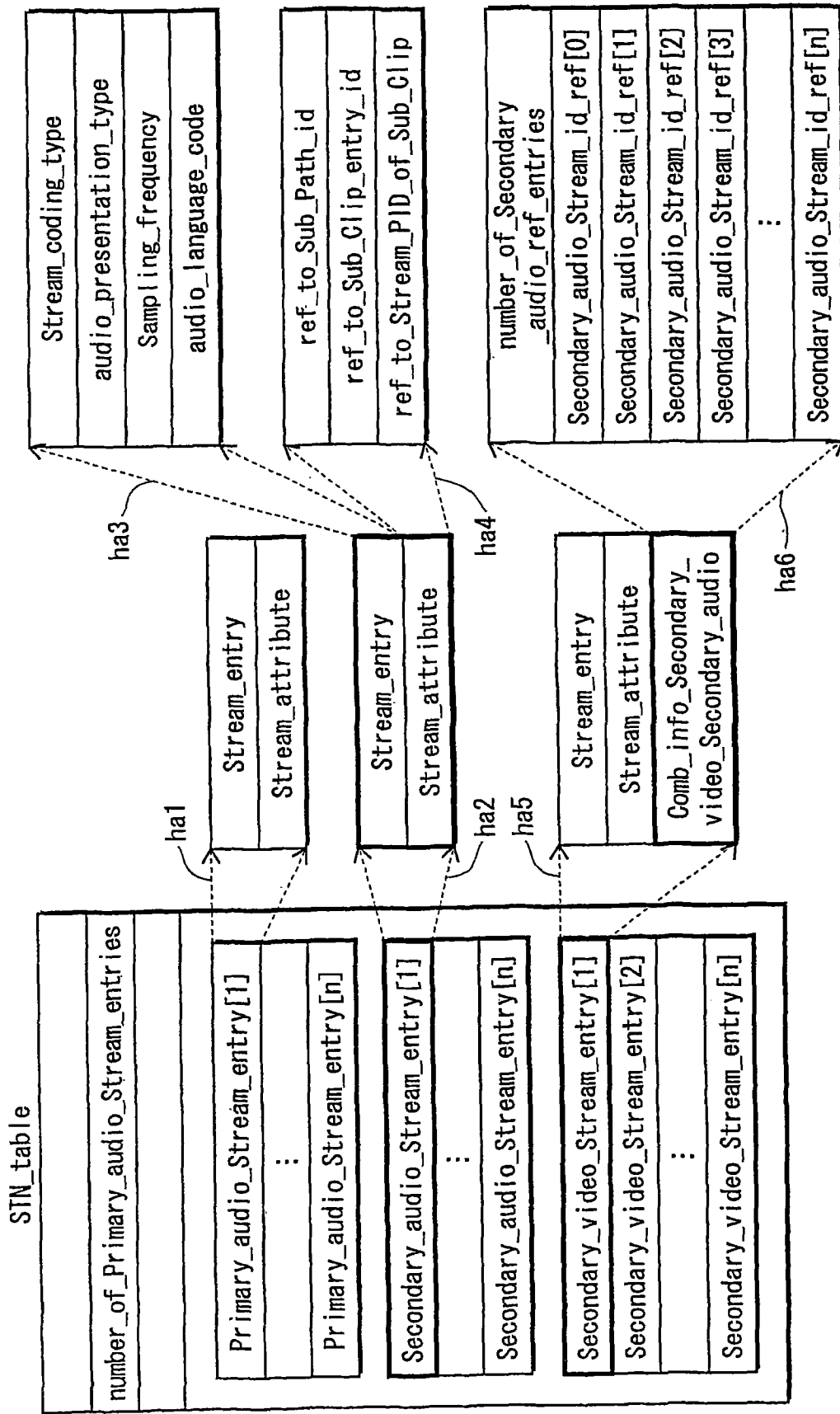
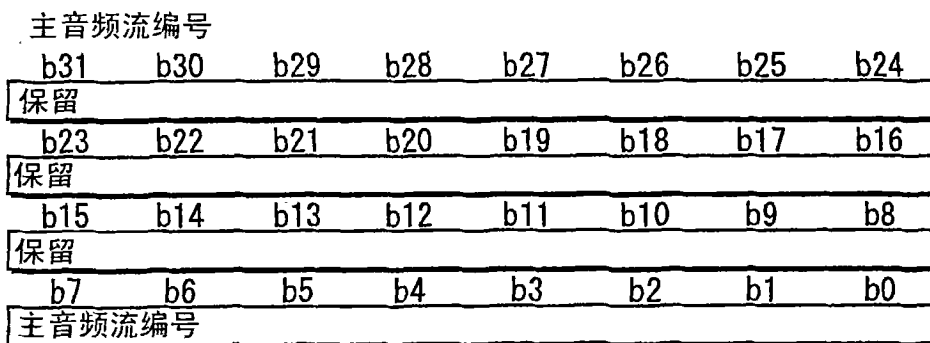


图 40



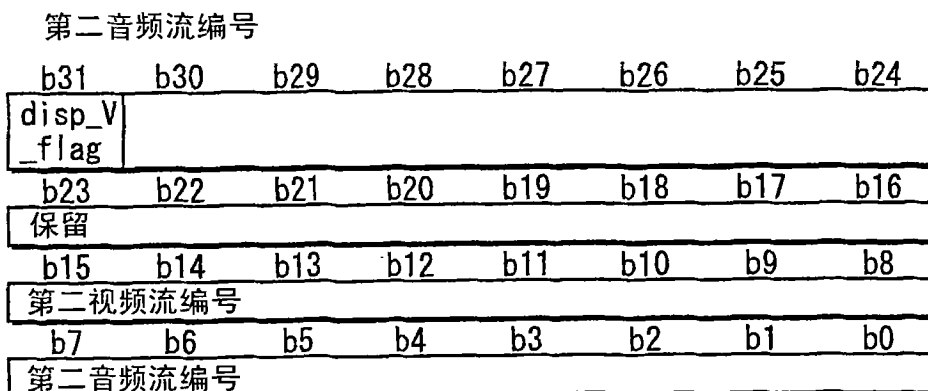
示出了可以用于与第二视频流进行组合的第二音频流

图 41



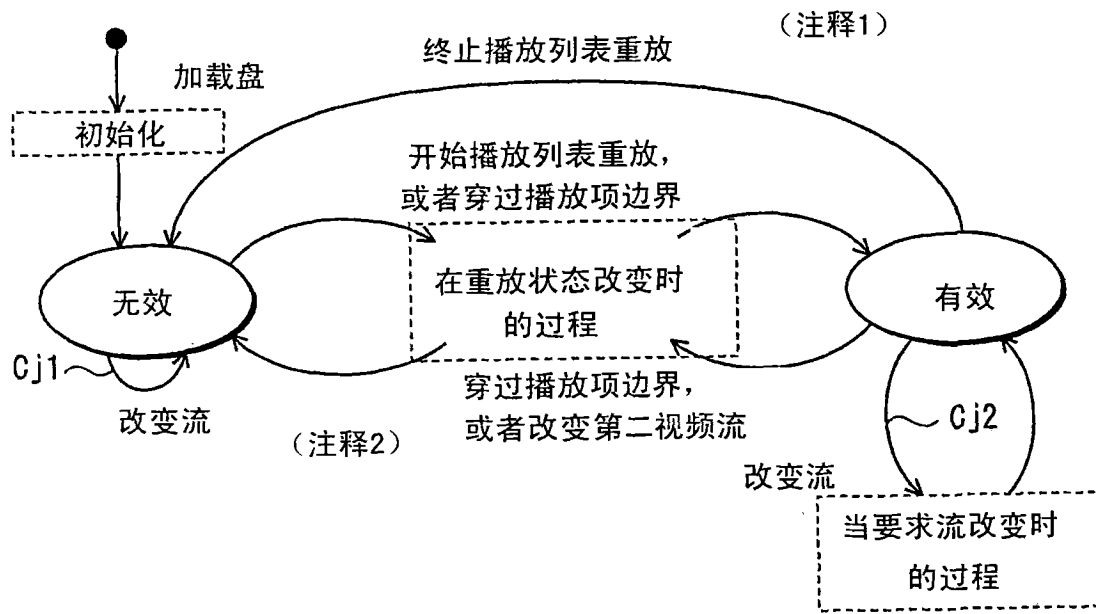
主音频流
 编号 ... 0: 保留
 1到32: 主音频流编号
 0xFF: 没有选择主音频流或者没有主音频流

图 42A



第二音频流
 编号 ... 0: 保留
 1到32: 第二音频流编号
 0xFF: 没有选择第二音频流或者没有第二音频流
 其他: 保留

图 42B



注释1: 当状态从“有效”变为“无效”时保持PSR的最后的值

注释2: 当状态处于“无效”时, 简单地覆写PSR的值

图 43

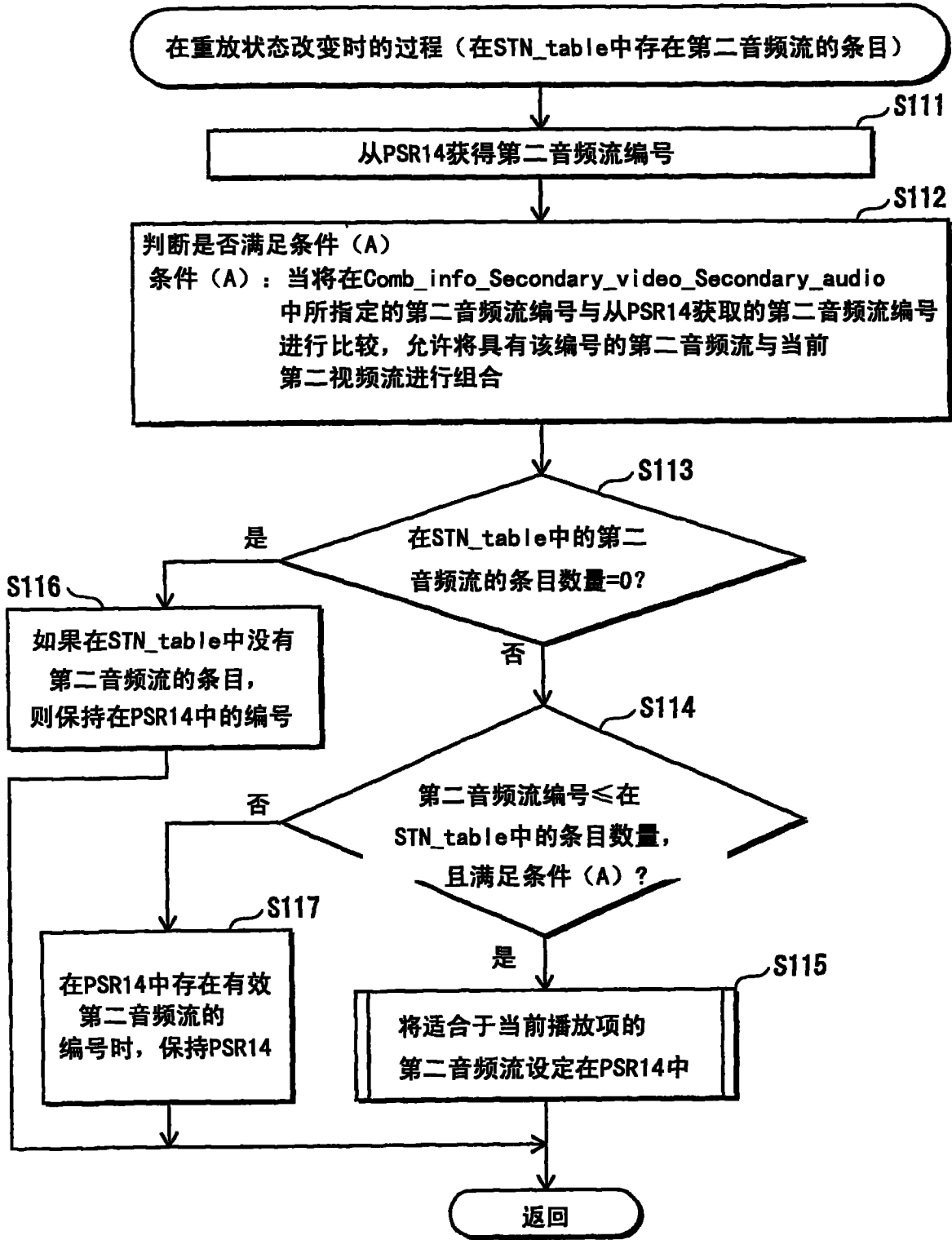


图 44

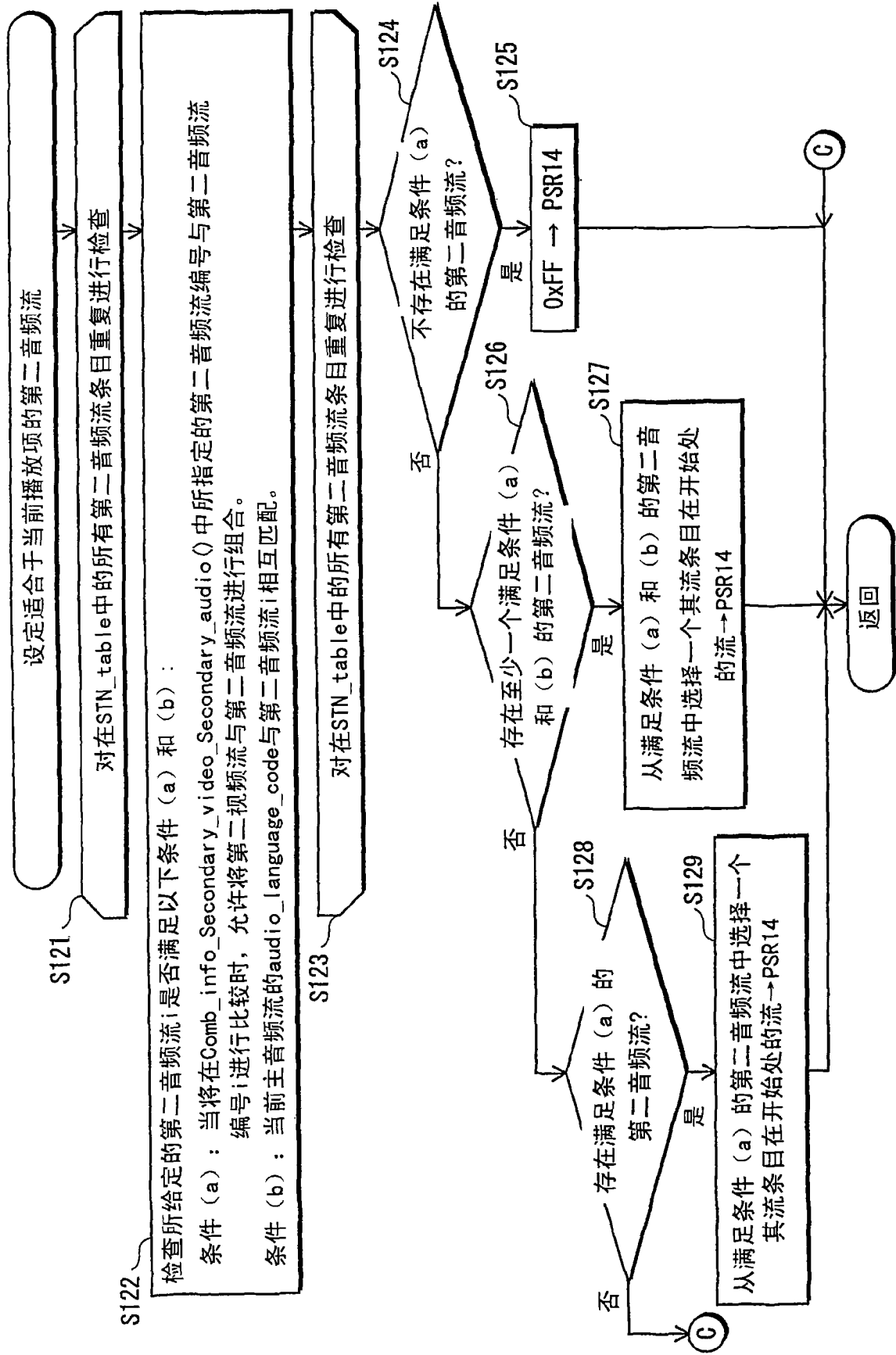


图 45

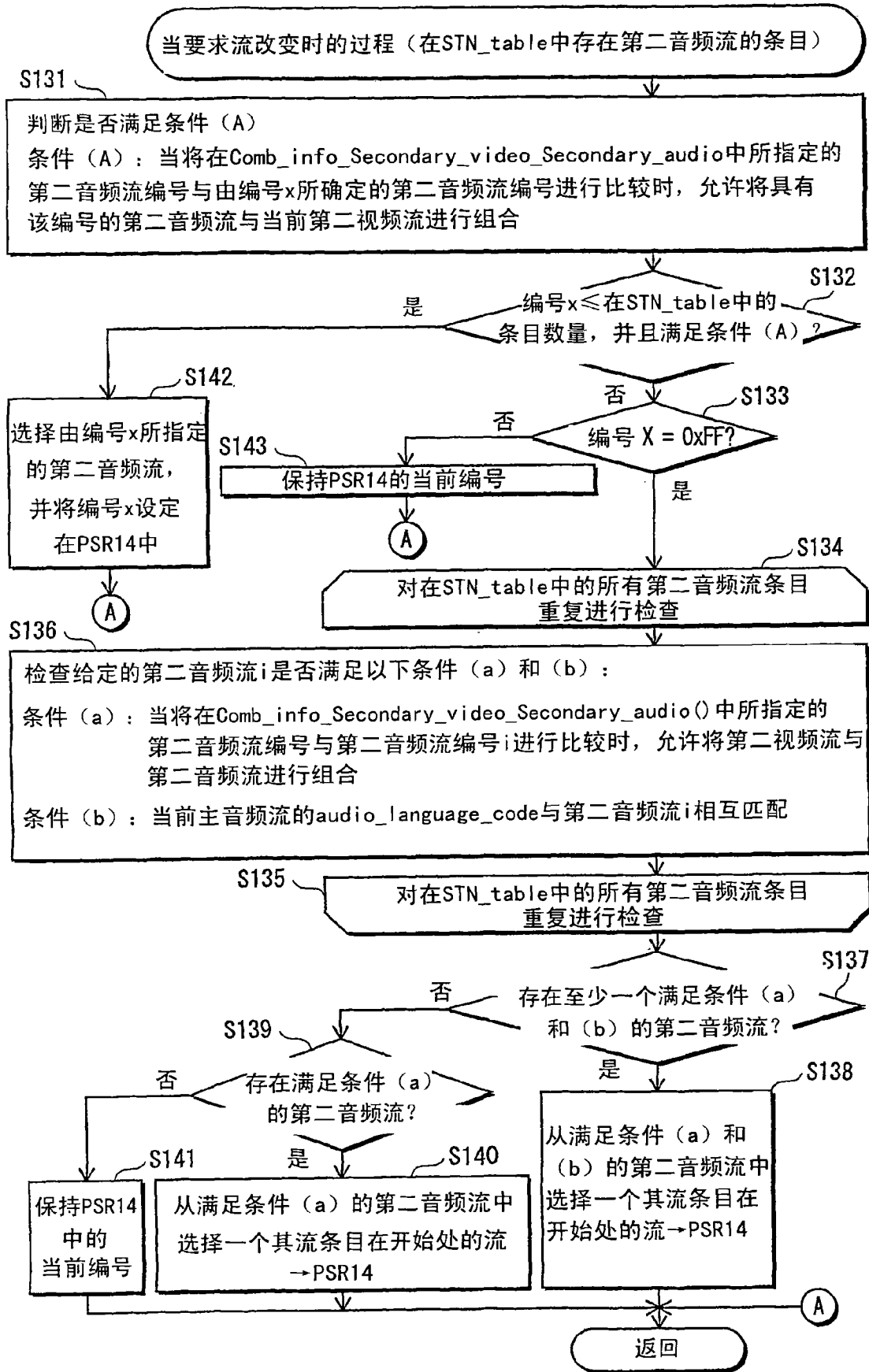
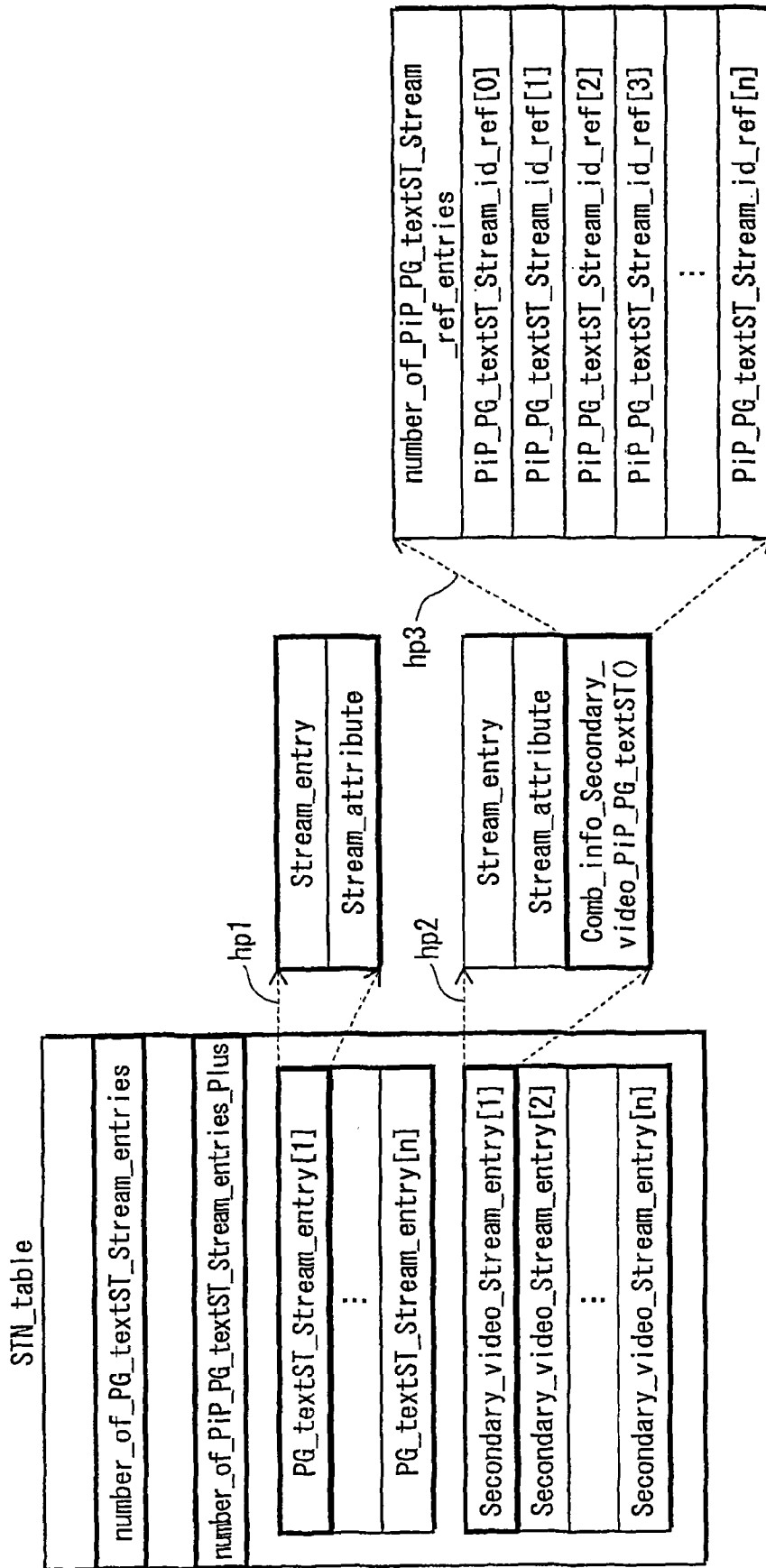


图 46



示出了可以用于与第二视频流进行组合的PIP_PG_textST_Stream

图 47

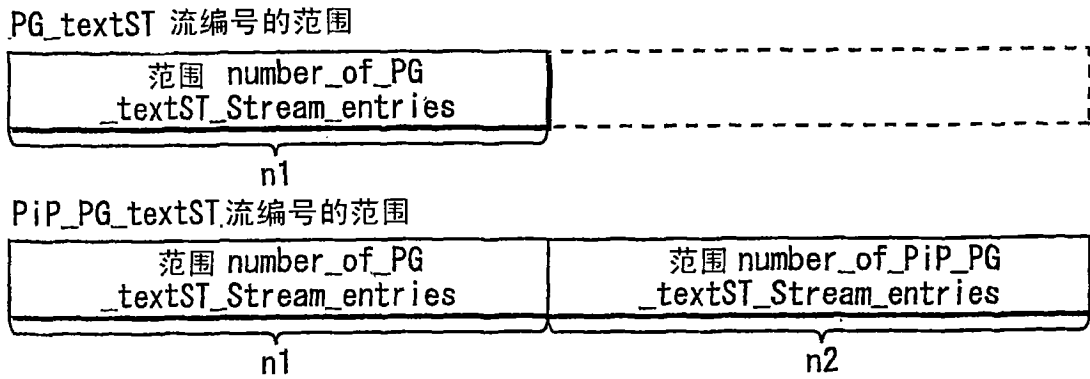
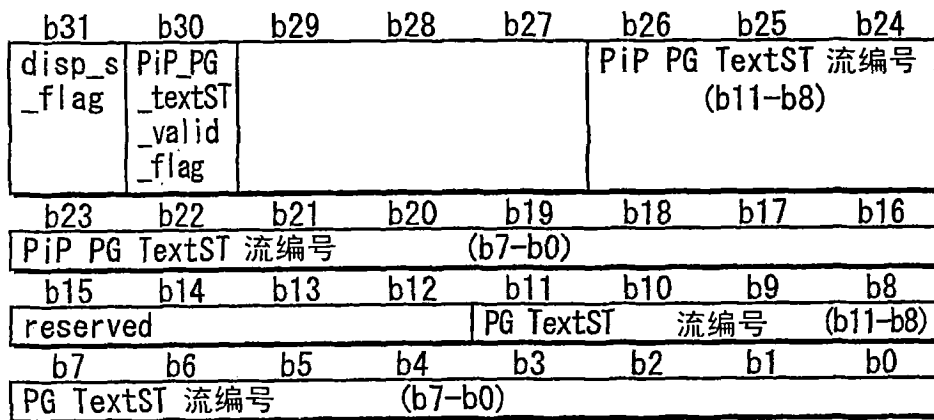


图 48A



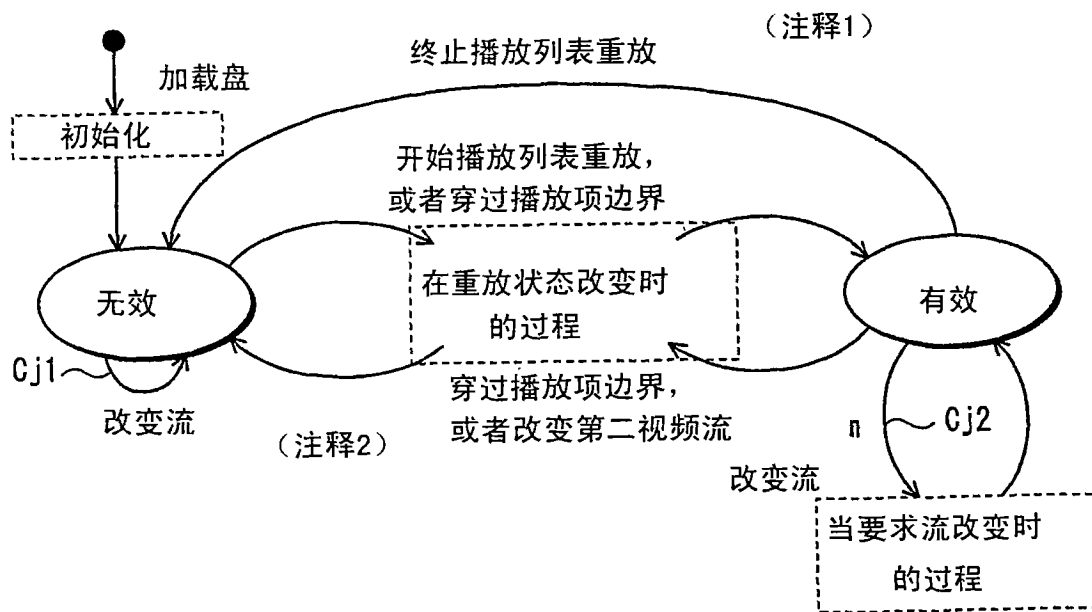
disp_s_flag ... 0b: 禁止“PG textST流”和“PiP PG textST流”的显示; 以及
 1b: 允许“PG textST流”和“PiP PG textST流”的显示。

PiP_PG_textST_valid_flag ... 0b: 在显示第二视频流时, 应使用在PSR2中所定义的 PG TextST流编号; 以及
 1b: 当显示第二视频流时, 应使用在PSR2中定义的 PiP PG TextST流编号。

PG TextST 流编号 ... 0: 保留
 1到255: PG TextST流编号
 0xFFF: 未选择PG TextST流编号或者没有PG TextST流

PiP PG TextST 流编号 ... 0: 保留
 1到255: PiP PG TextST流编号
 0xFFF: 未选择PiP PG TextST流编号或者没有PiP PG TextST流

图 48B



注释1: 当状态从“有效”变为“无效”时保持PSR的最后的值

注释2: 当状态处于“无效”时, 简单地覆写PSR的值

图 49

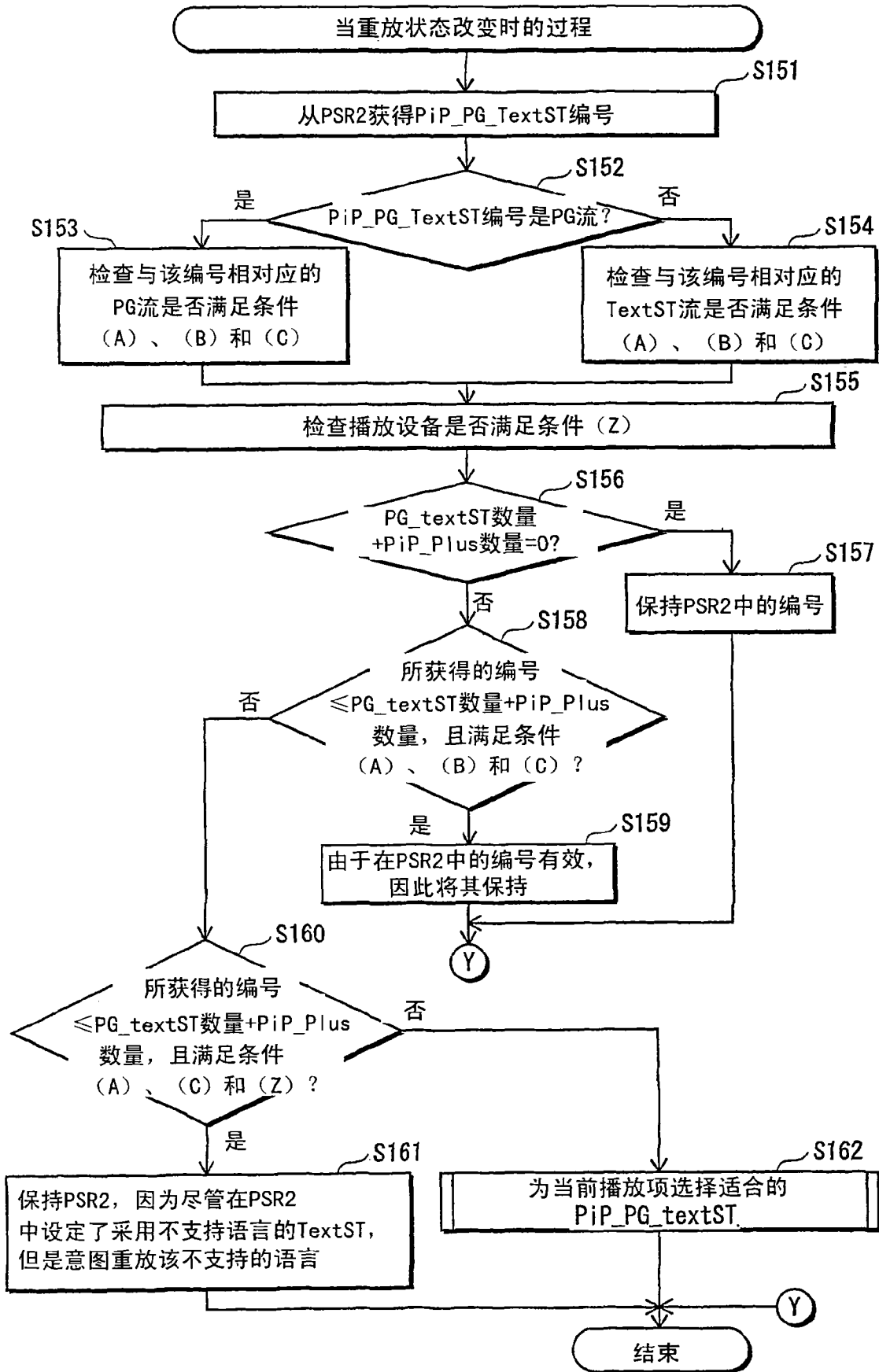


图 50

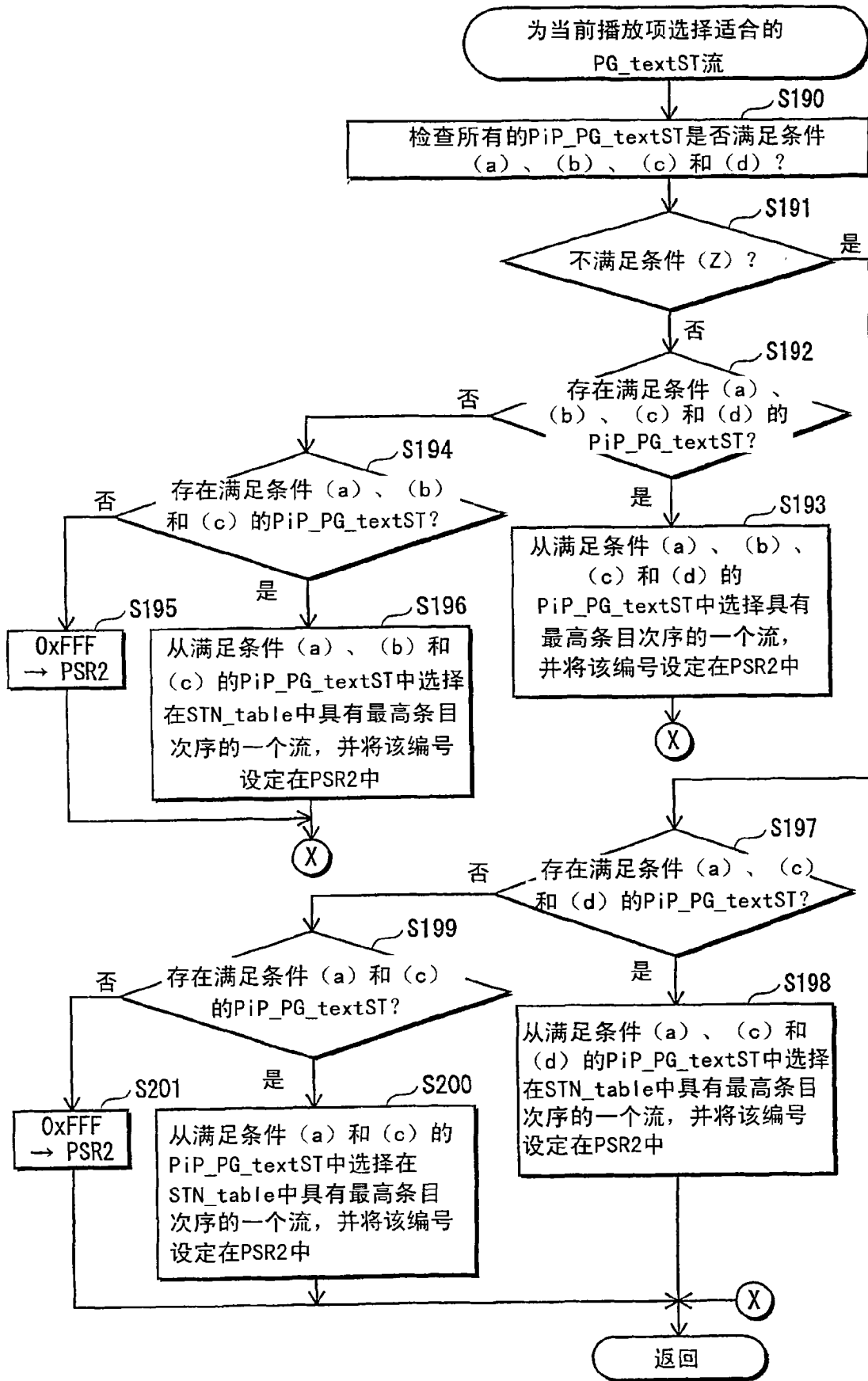


图 51

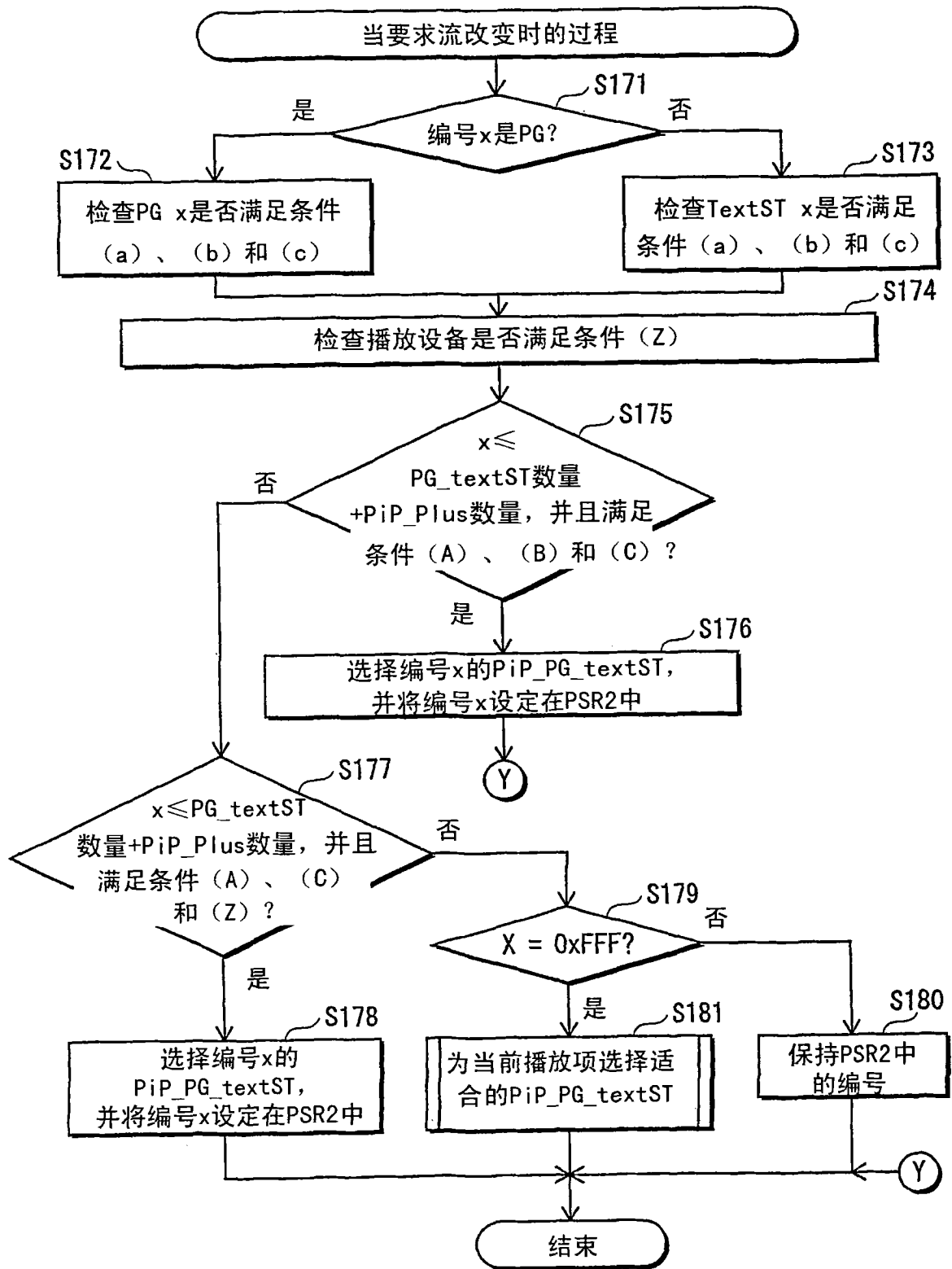


图 52

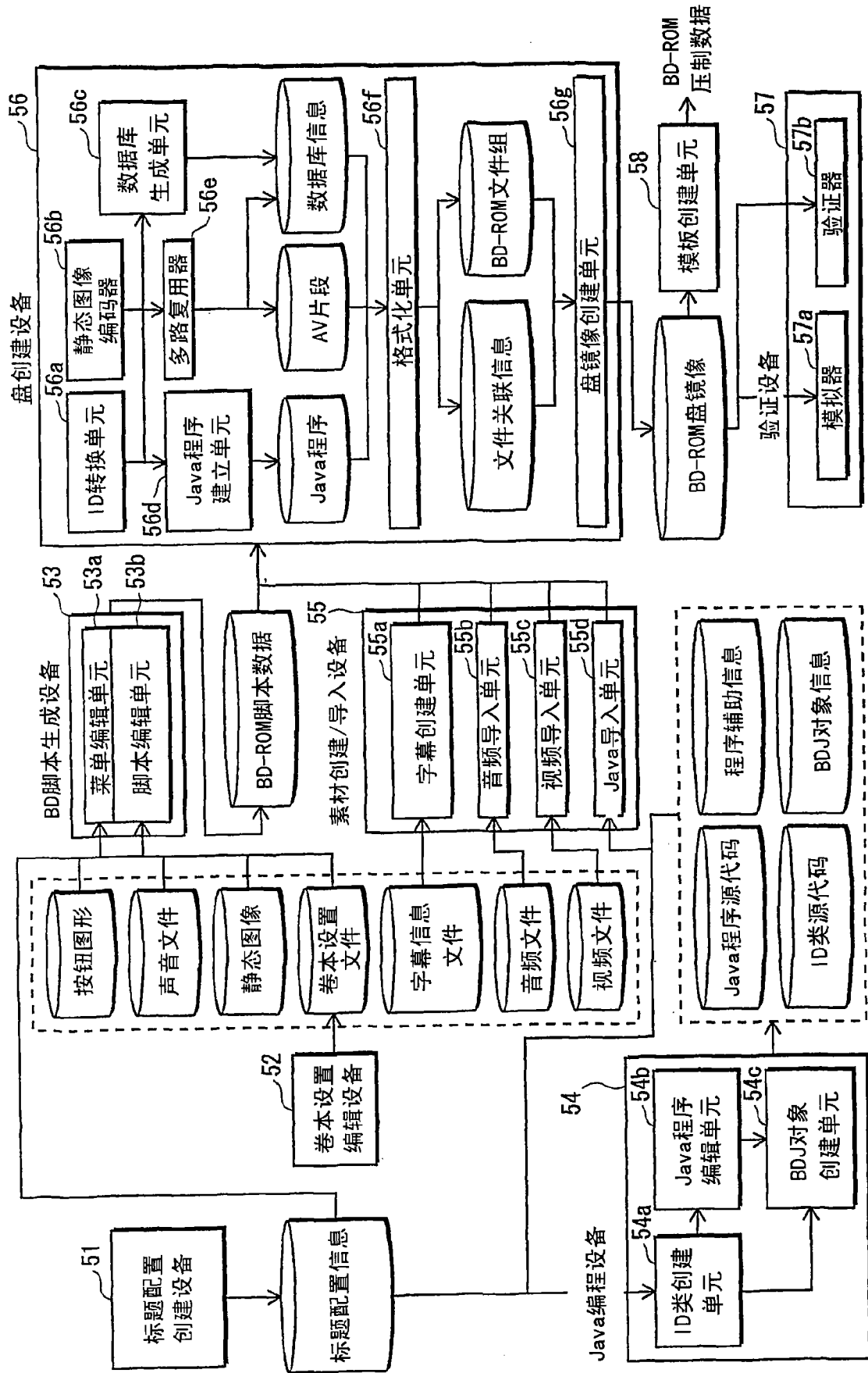


图 53

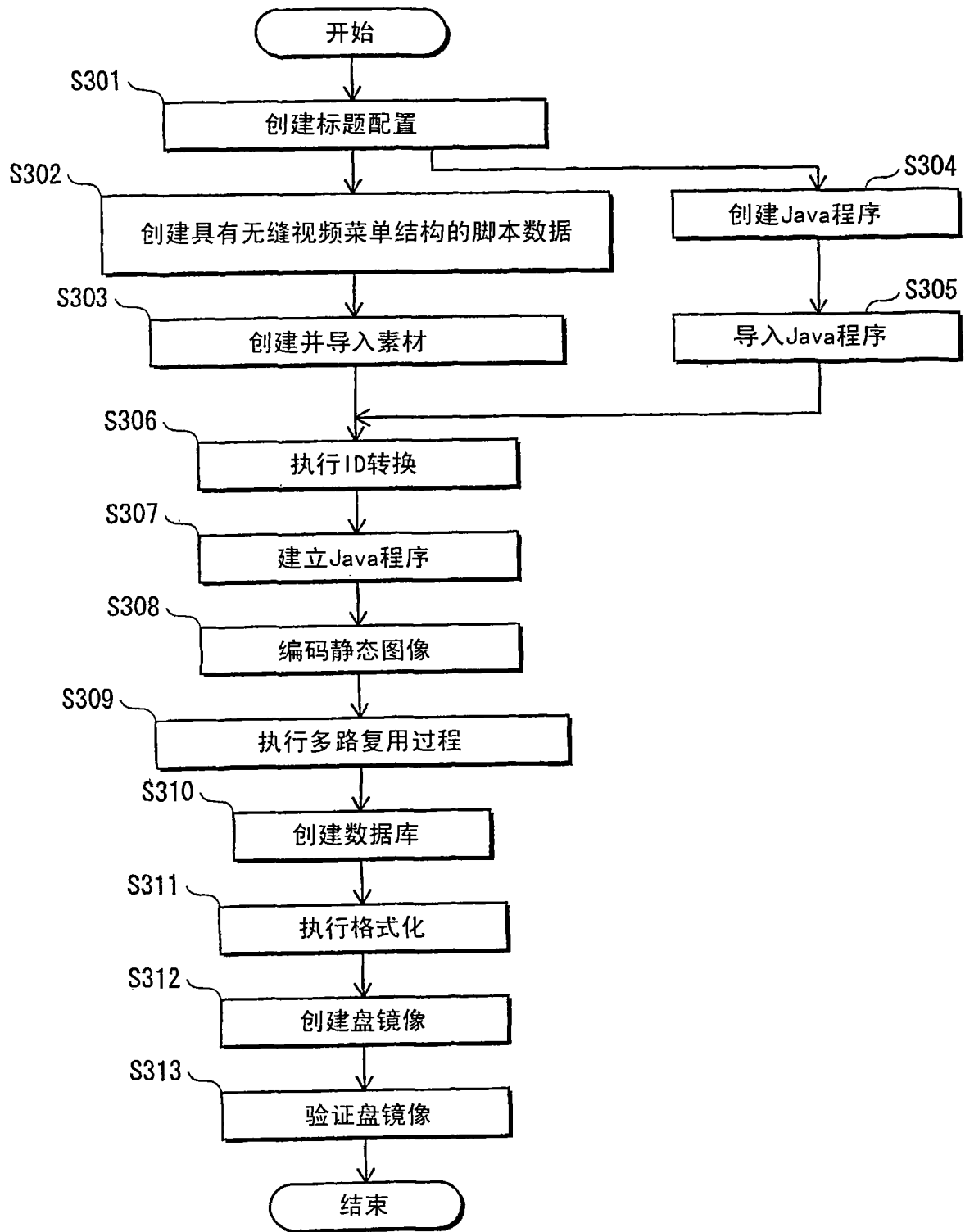


图 54

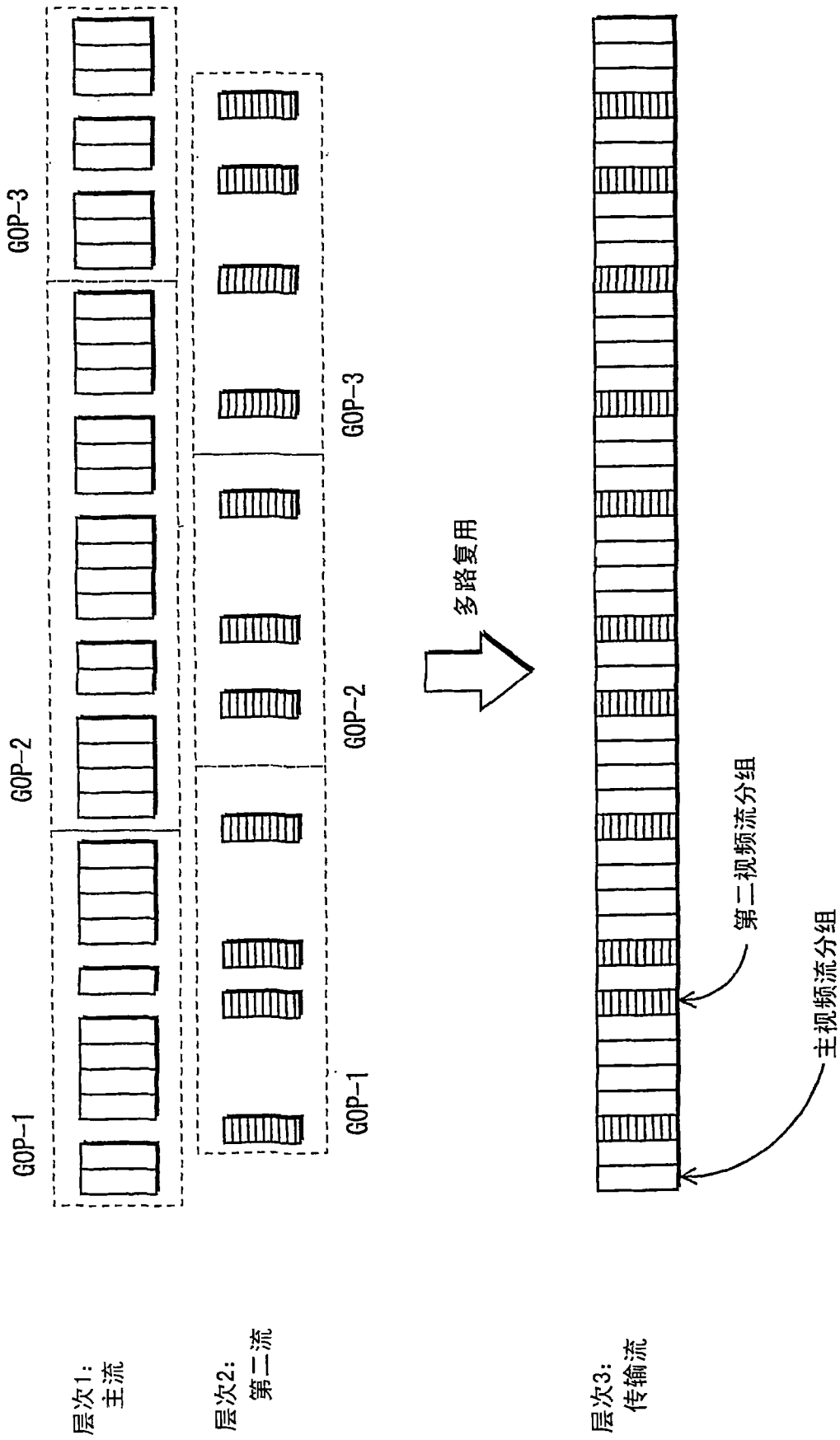


图 55

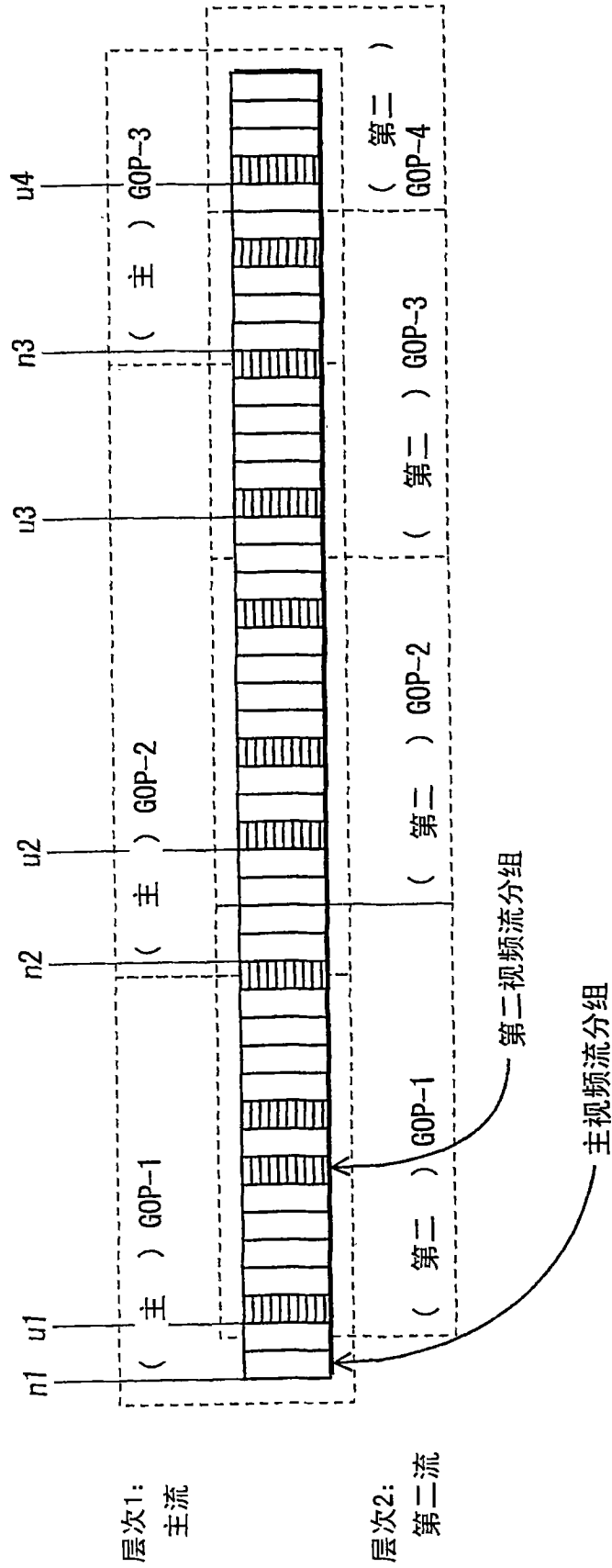


图 56

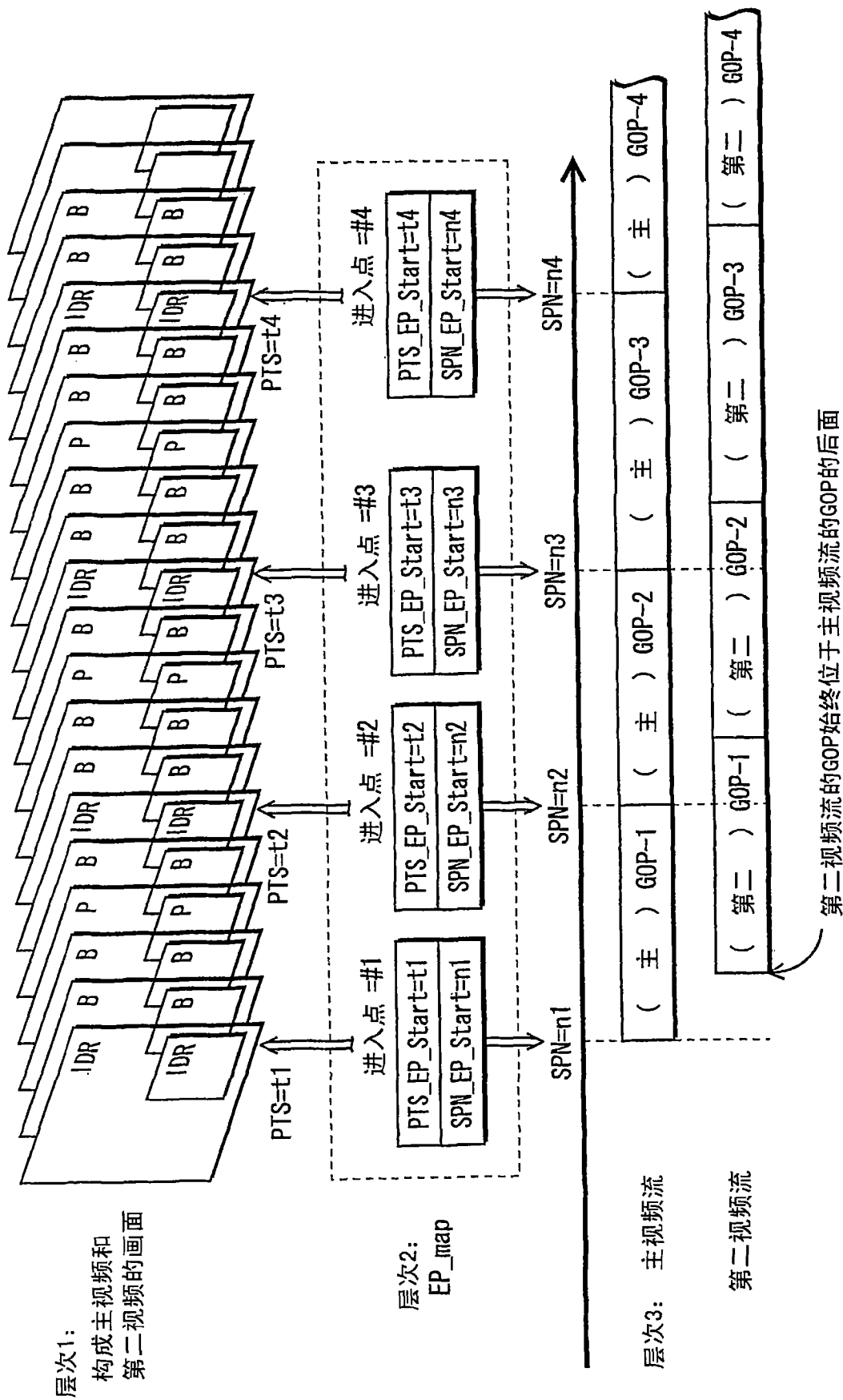


图 57

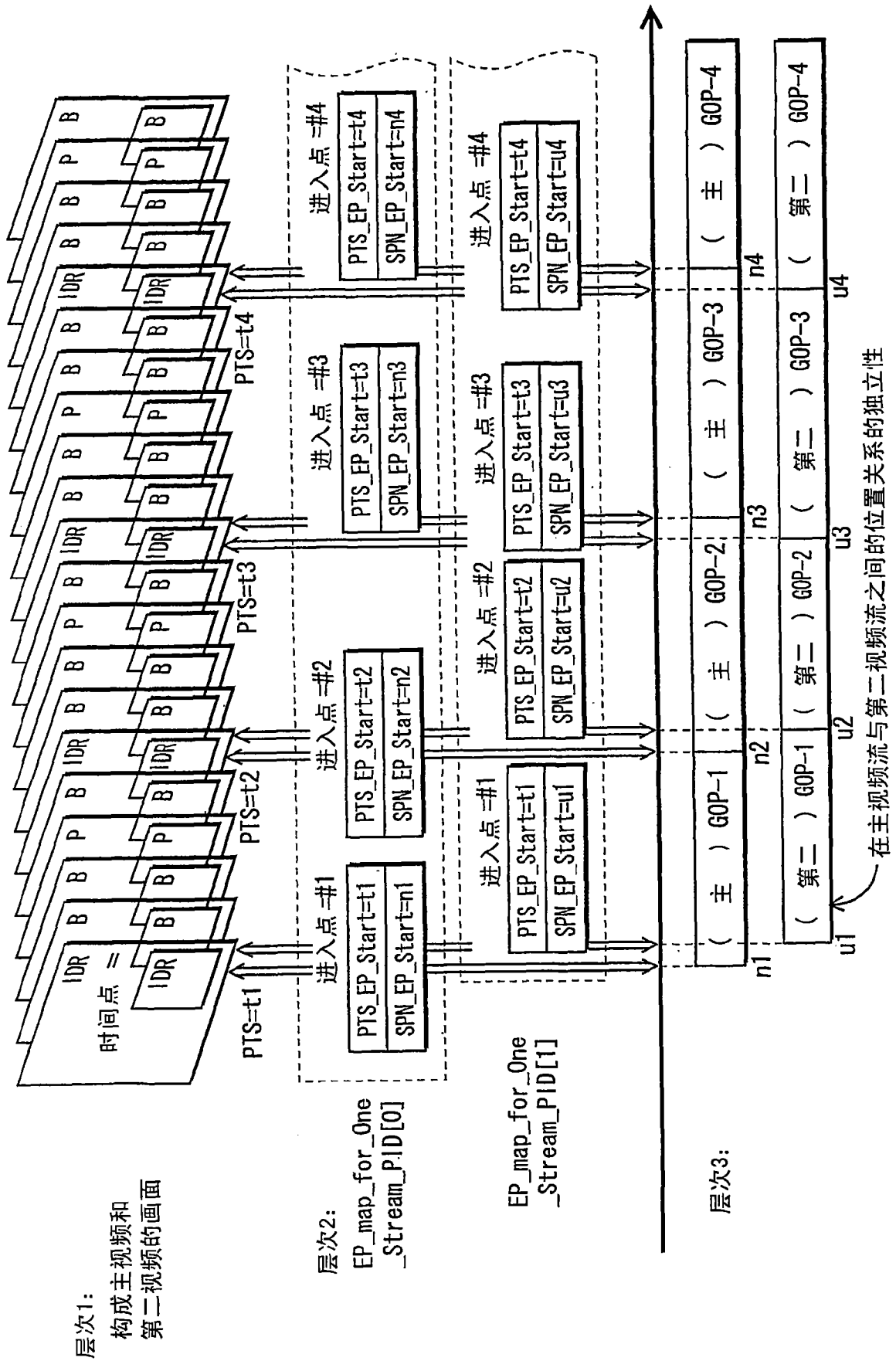


图 58

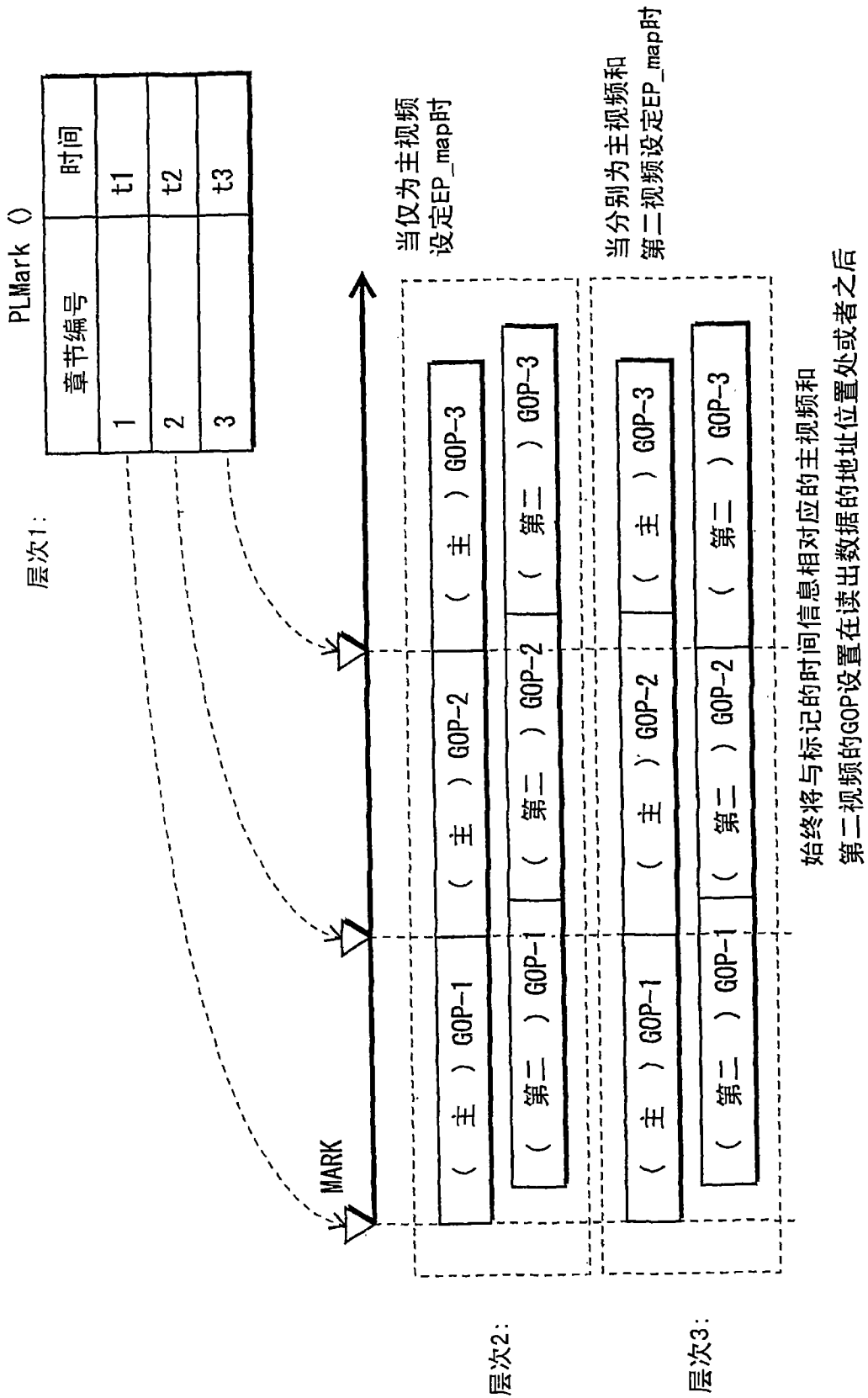
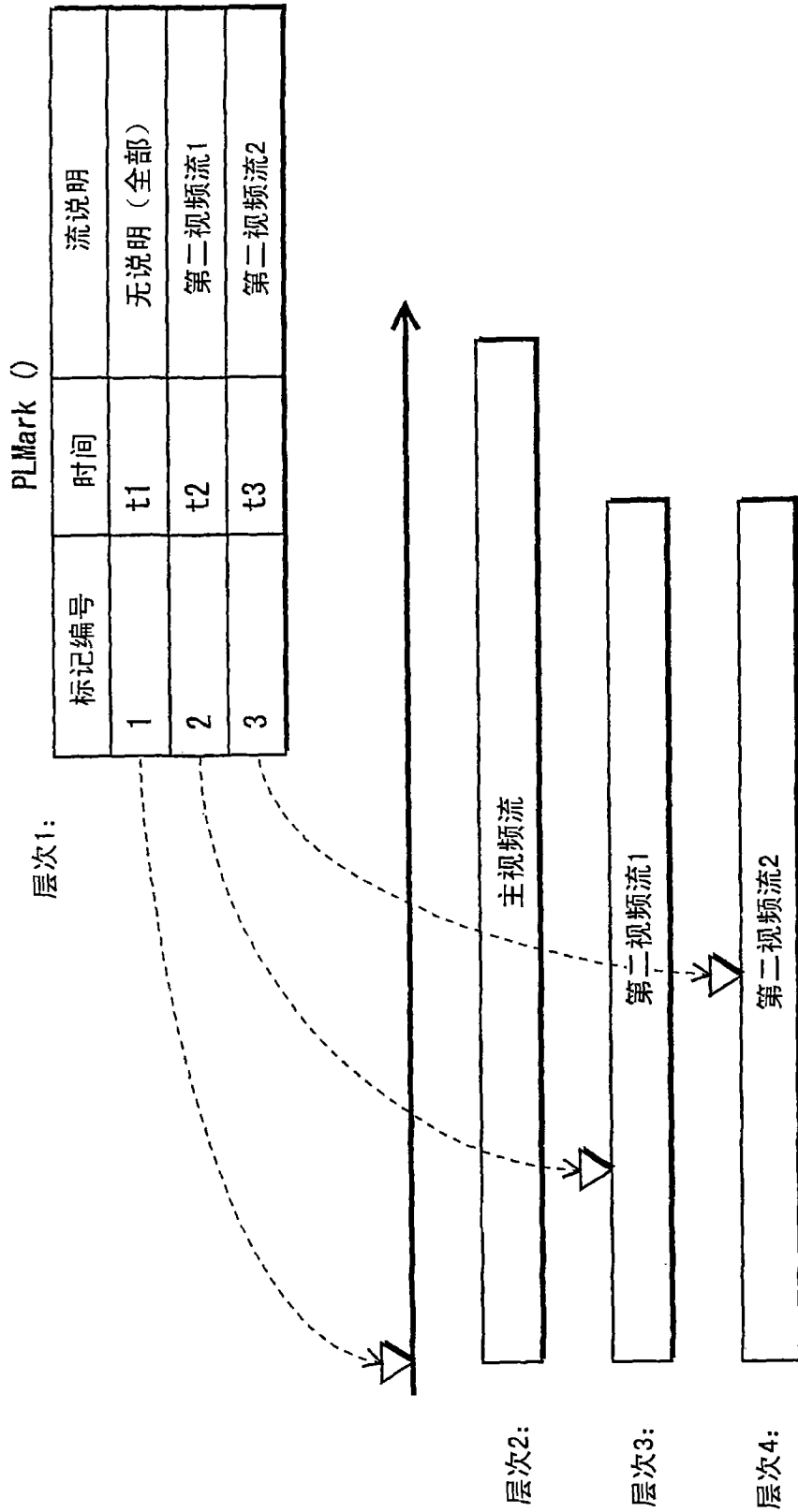


图 59



仅在正在显示适合的
第二视频流时，
用于第二视频流的标记变为有效并能够用作跳点

图 60

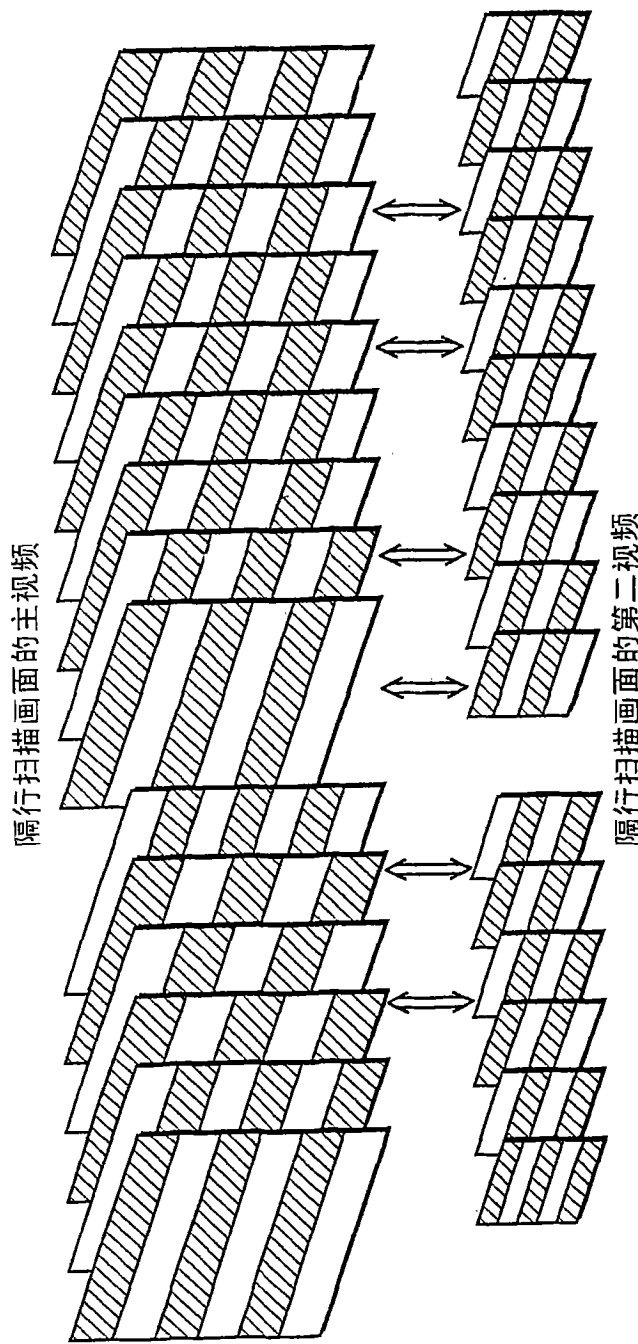


图 61A

用于同相的组合

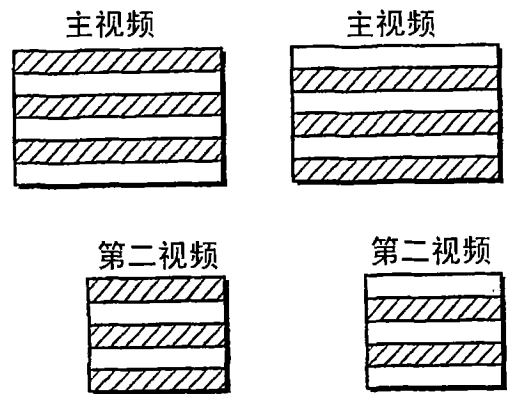


图 61B

用于反相的组合

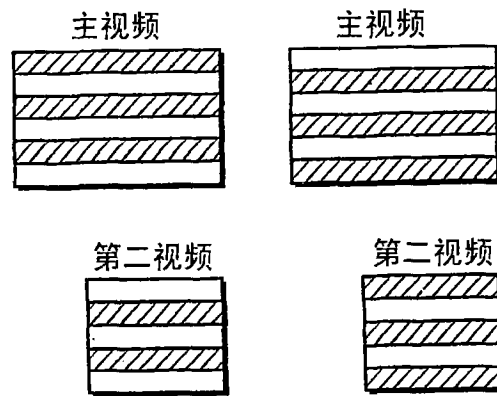
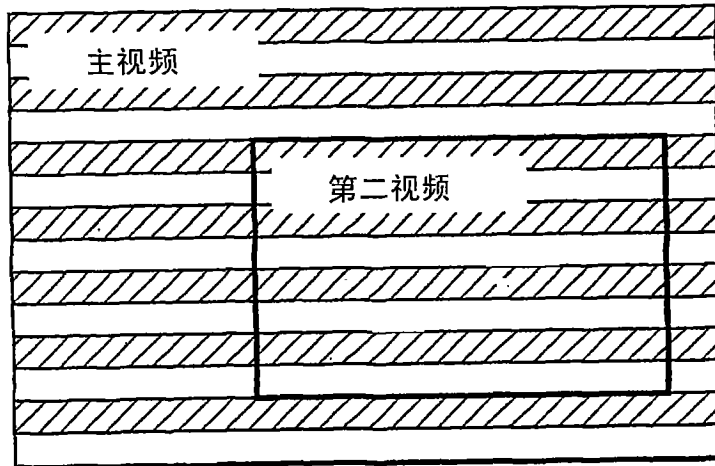


图 61C

可播放的合成模式



不可播放的合成模式

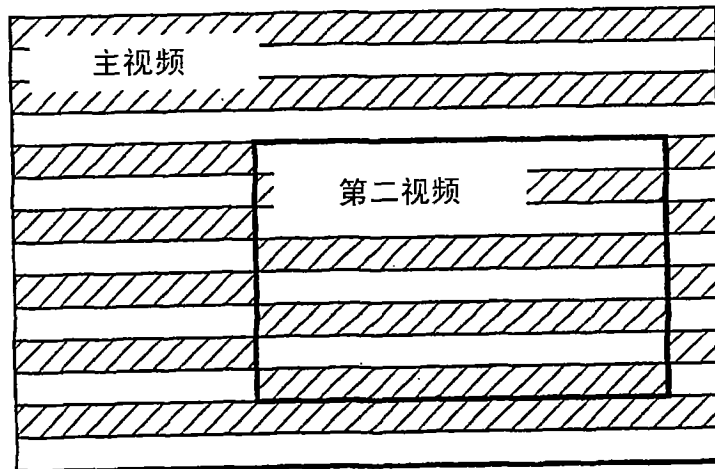


图 62

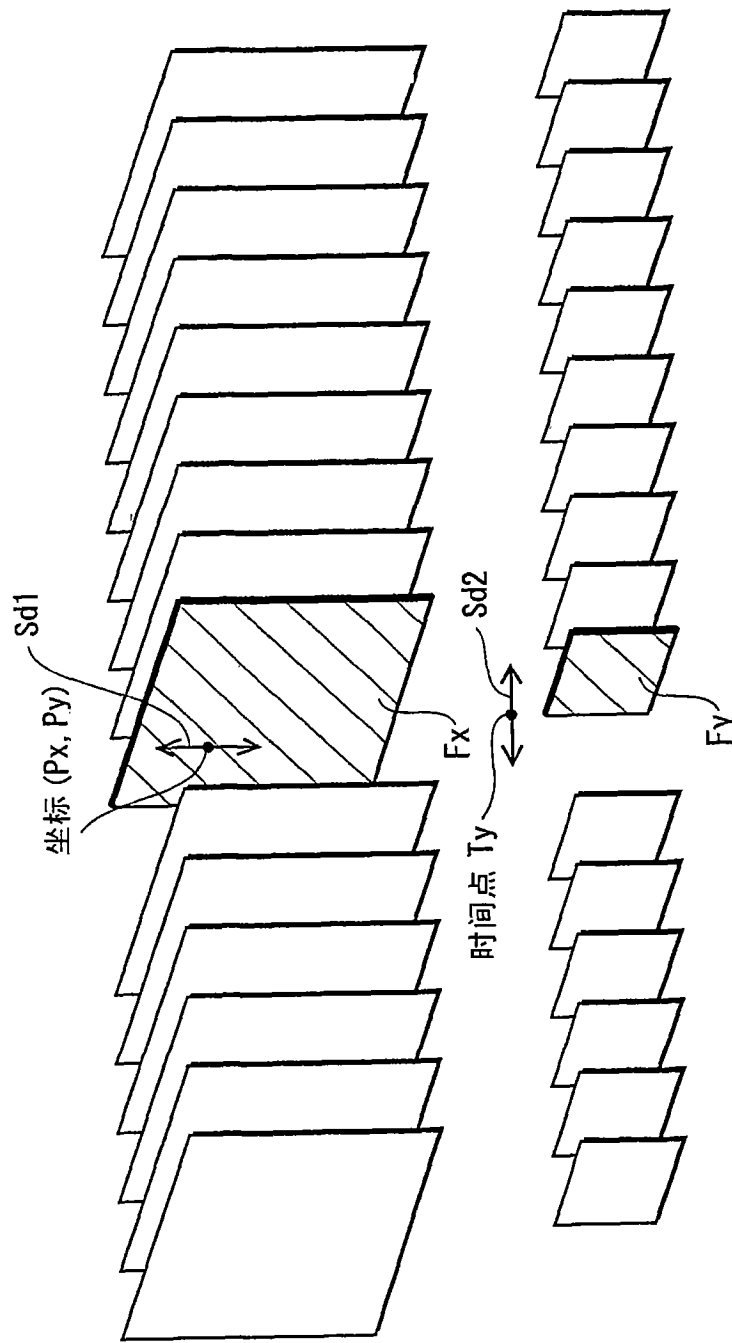


图 63

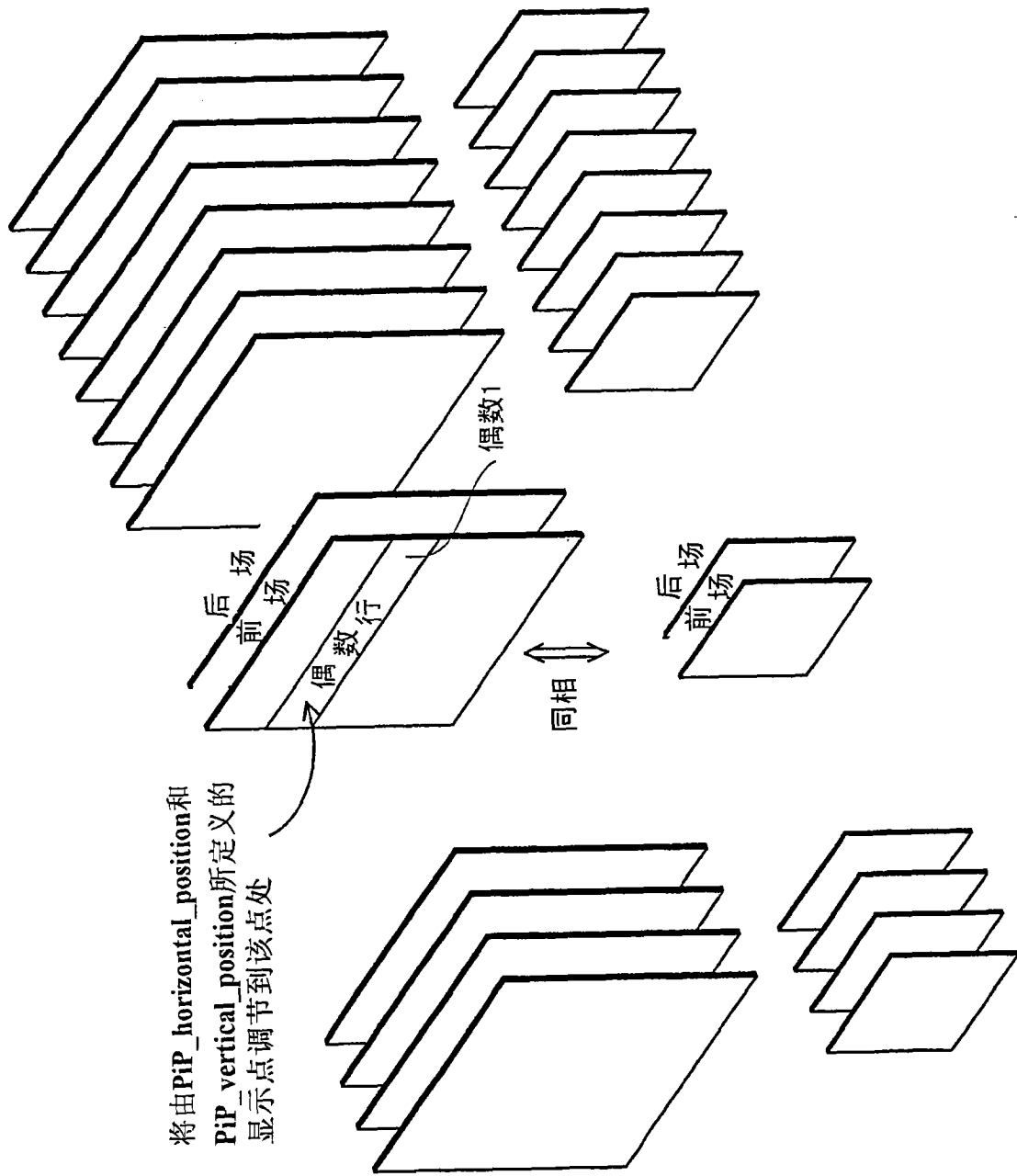


图 64

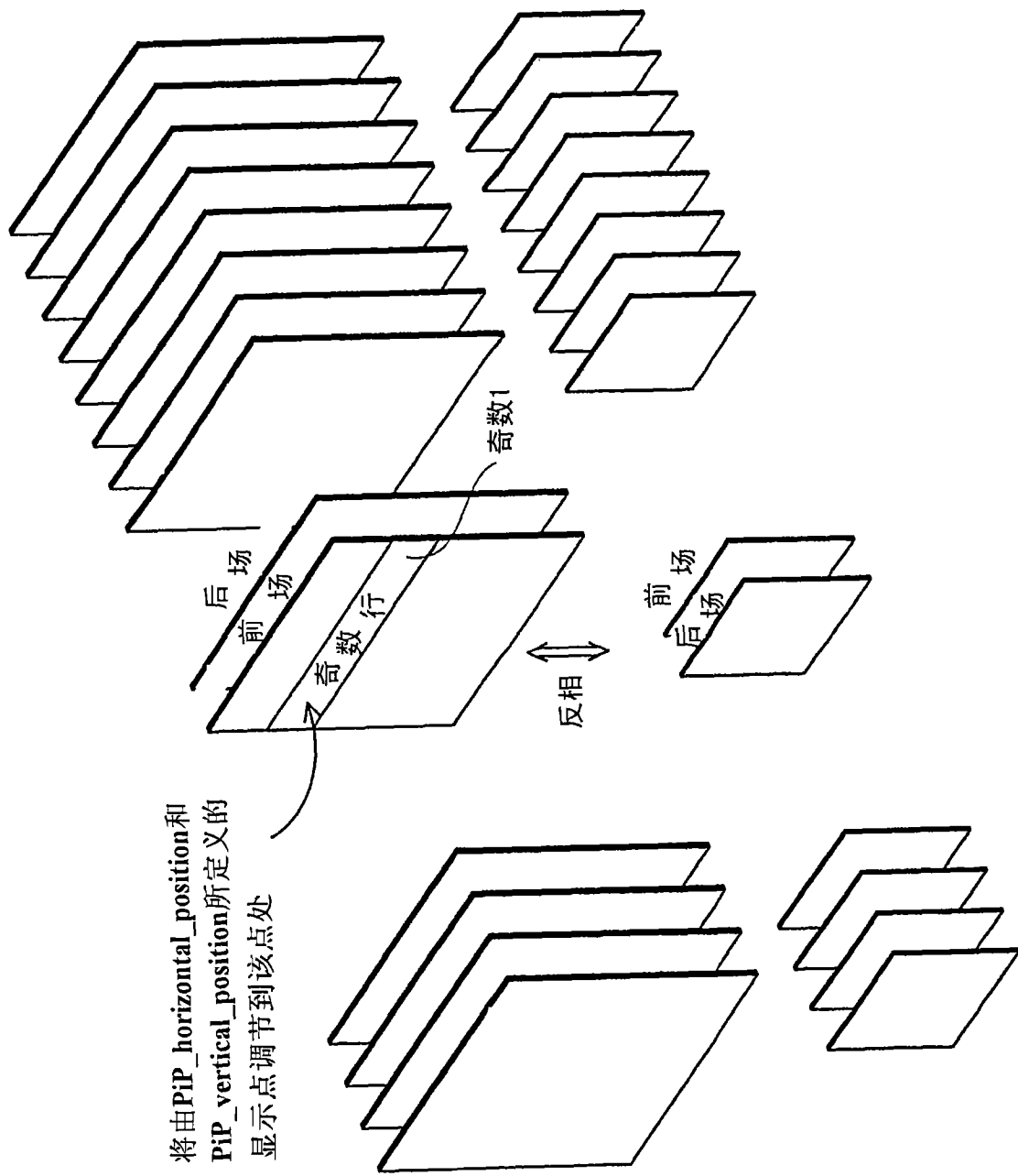


图 65

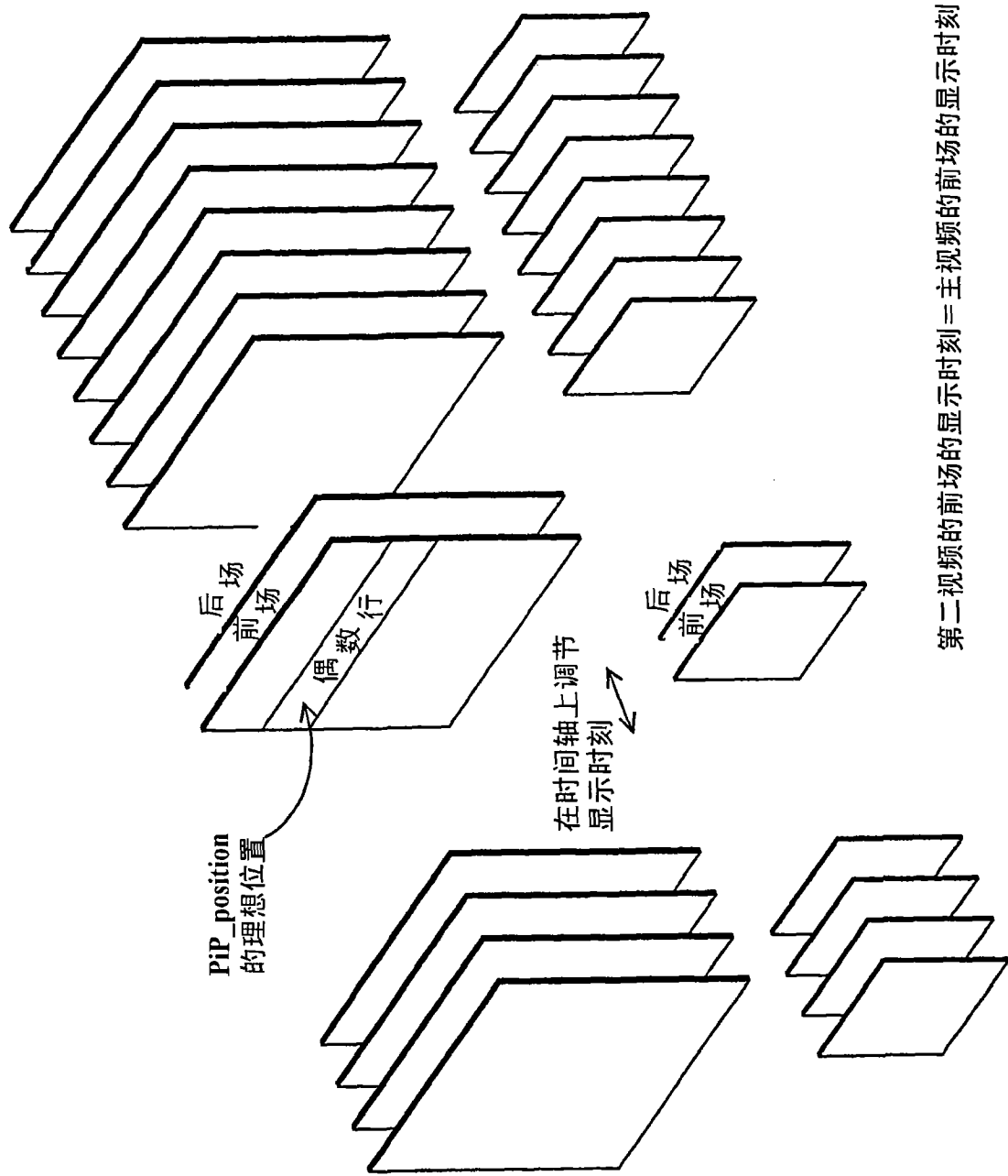


图 66

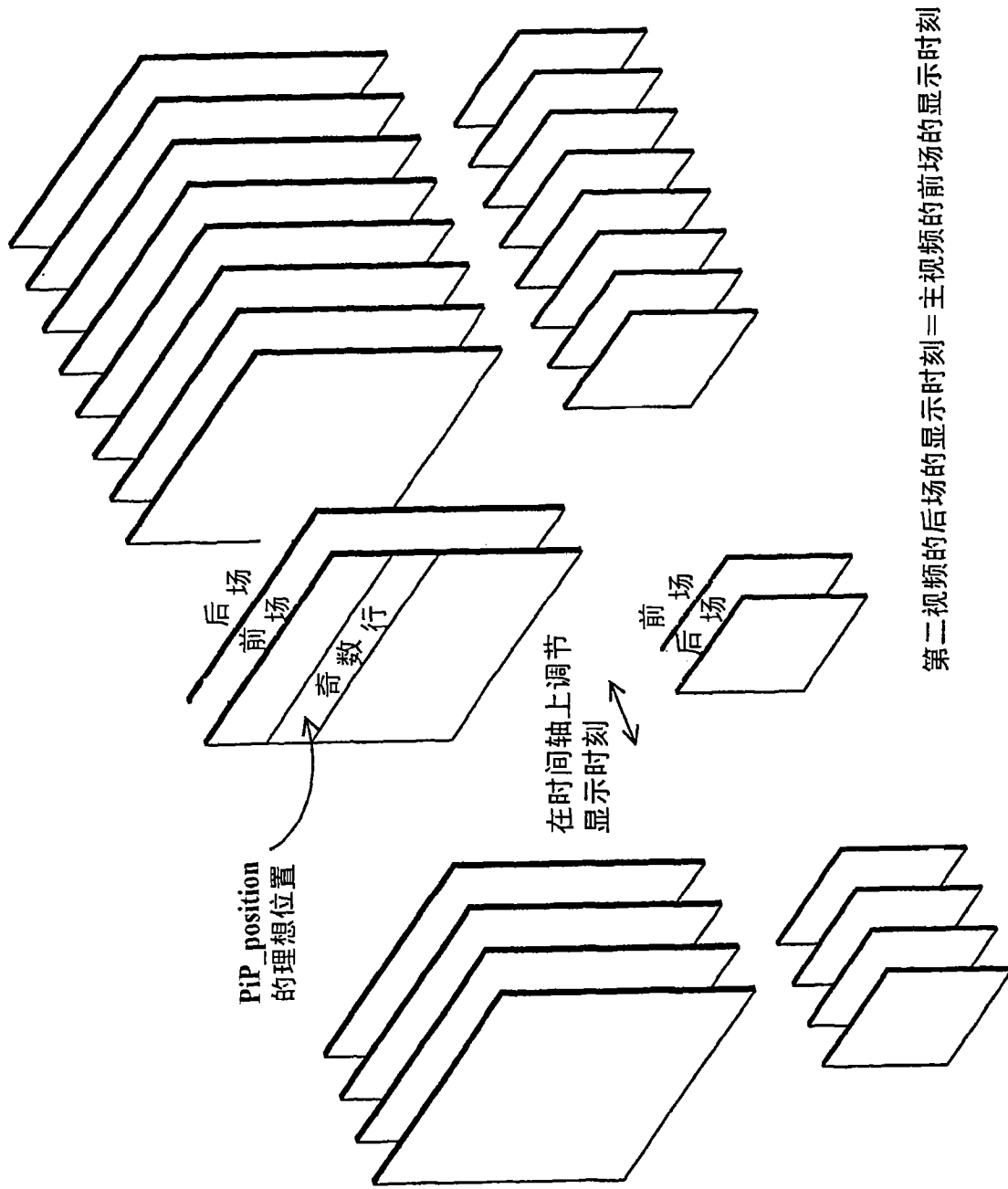


图 67