(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101002268 B
(45)授权公告日 2010.09.08

(21)申请号 200580027196.2
(22)申请日 2005.06.08
(30)优先权数据
174571/2004 2004.06.11 JP
(85)PCT申请进入国家阶段日
2007.02.12
(86)PCT申请的申请数据
PCT/JP2005/010905 2005.06.08
(87)PCT申请的公布数据
WO2005/122175 JA 2005.12.22
(73)专利权人 索尼株式会社
地址 日本东京都
专利权人 索尼计算机娱乐公司
(72)发明人 藤波靖 浜田俊也 各务辰哉
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 李芳华 邸万奎
(51)Int.Cl.
G11B 27/00(2006.01)
G11B 20/10(2006.01)
G11B 20/12(2006.01)
G11B 27/10(2006.01)
H04N 5/85(2006.01)
H04N 5/92(2006.01)
(56)对比文件
CN 1111873 A,1995.11.15, 说明书第27页第2、4段,第29页第7段,第35页,第36页第1段,附图19-22,25.
JP 特开平11-341440 A,1999.12.10, 全文.
JP 特开2001-176246 A,2001.06.29, 全文.
JP 特开2003-23604 A,2003.01.24, 全文.
JP 特开2002-57990 A,2002.02.22, 全文.
审查员 陈涛

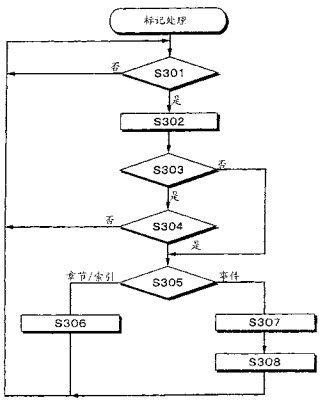
权利要求书 1 页 说明书 73 页 附图 50 页

(54)发明名称

数据处理设备和数据处理方法

(57)摘要

根据数据再现时间,执行与参数相对应的处理。当根据包括PlayListMark()(该PlayListMark()具有包括mark_time_stamp、mark_type和mark_data的Mark(),其中mark_time_stamp指示在该播放列表时间轴上的一个再现时间,mark_type指示Mark()的类型,而且mark_data在当mark_type是生成事件的类型时作为该事件的参数)的播放列表而再现的数据的再现时间与mark_time_stamp一致时,在步骤S302,识别出具有该mark_time_stamp的Mark()。当所识别的Mark()所具有的mark_type指示生成事件的类型时,在步骤S307,报告Mark()所具有的mark_data和事件的生成。在步骤S308,执行与mark_data相对应的处理。这个发明可以应用于使用DVD等的游戏设备。



CN 101002268 B

1. 一种数据处理设备，处理记录在数据记录介质上的记录数据，

该记录数据包括：

编码数据，其中编码了数据，以及

播放列表，其表示数据的再现过程，

该播放列表包括播放列表标记信息，其具有作为在该播放列表的时间轴上的标记的标记信息，

该标记信息包括：

时间信息，其表示在该播放列表的时间轴上的一个再现时间，

类型信息，其表示该标记信息的类型，以及

参数信息，其在所述类型信息表示生成事件的类型时、成为事件的参数，

该数据处理设备包括：

确定装置，用于确定与该播放列表相对应地再现的数据的再现时间是否匹配该时间信息；

识别装置，用于当所述确定装置已经确定该数据的再现时间已经匹配了时间信息时，识别出具有所述时间信息的标记信息；

通知装置，用于当所述标记信息具有的、并且由所述识别装置识别的类型信息表示生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现；以及

执行装置，用于执行与通知装置所通知的参数信息相对应的处理。

2. 如权利要求 1 所述的数据处理设备，

其中所述执行装置通过执行中断处理来执行与该参数信息相对应的处理。

3. 一种数据处理方法，处理记录在数据记录介质上的记录数据，

该记录数据包括：

编码数据，其中编码了数据，以及

播放列表，其表示数据的再现过程，

该播放列表包括播放列表标记信息，其具有作为在该播放列表的时间轴上的标记的标记信息，

该标记信息包括：

时间信息，其表示在该播放列表的时间轴上的一个再现时间，

类型信息，其表示该标记信息的类型，以及

参数信息，其在所述类型信息表示生成事件的类型时、成为事件的参数，

该数据处理方法包括步骤：

确定与该播放列表相对应地再现的数据的再现时间是否匹配该时间信息；

当在所述确定步骤已经确定该数据的再现时间已经匹配了时间信息时，识别出具有所述时间信息的标记信息；

当该标记信息所具有的、并且在所述识别步骤处识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现；以及

执行与在所述通知步骤通知的参数信息相对应的处理。

数据处理设备和数据处理方法

技术领域

[0001] 本发明涉及数据处理设备和数据处理方法。本发明尤其涉及允许非常便利地处理数据的上述领域。

背景技术

[0002] 近年来，具有大存储容量并可以随机存取的记录介质如数字通用盘 (DVD) 已传播开来。此外，用 DVD 执行各种处理的 DVD 设备也已传播开来。

[0003] 作为 DVD 设备，存在有向 / 从 DVD 记录 / 再现电视广播节目等的数据的 DVD 记录机、使用其上记录有地图信息等的 DVD 并从 DVD 再现地图信息和进行显示的车载导航系统、使用其上记录有游戏程序等的 DVD 并从 DVD 读出程序和执行程序的游戏设备、以及其它设备。

[0004] DVD 的细节例如在非专利文献 1“DVD Specifications for Read-Only DiscPart 3 ;Version 1.1 December 1997”中有所描述。

[0005] 如 DVD 这样的其上记录大量数据的记录介质以及使用 DVD 的 DVD 设备需要允许非常便利地处理如此大量的数据。

发明内容

[0006] 鉴于上述观点而作出了本发明。本发明的目的是允许高度便利地执行数据处理等。

[0007] 本发明是一种数据处理设备，其处理记录在数据记录介质上的记录数据，该记录数据包括其中编码了数据的编码数据和代表数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴中的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为事件的参数的参数信息，该数据处理设备包括 :确定装置，用于确定与该播放列表相对应地再现的数据的再现时间是否匹配该时间信息 ;识别装置，用于当确定装置已经确定该数据的再现时间已经匹配了该时间信息时，识别出具有时间信息的标记信息 ;通知装置，用于当该标记信息具有并且由识别装置所识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现 ;以及执行装置，用于执行与通知装置所通知的参数信息相对应的处理。

[0008] 本发明是一种数据处理方法，其处理记录在数据记录介质上的记录数据，该记录数据包括其中编码了数据的编码数据和代表数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴上的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为事件的参数的参数信息，该数据处理方法包括步骤 :确定与该播放列表相对应地再现的数据的再现时

间是否匹配该时间信息；当在所述确定步骤已经确定该数据的再现时间已经匹配了该时间信息时，识别出具有时间信息的标记信息；用于当该标记信息具有并且在所述识别步骤所识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现；以及执行与在所述通知步骤通知的参数信息相对应的处理。

[0009] 本发明是一种程序，其导致计算机处理记录在数据记录介质上的记录数据，该记录数据包括其中编码了数据的编码数据和代表数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴上的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为事件的参数的参数信息，该程序包括步骤：确定与该播放列表相对应地再现的数据的再现时间是否匹配该时间信息；当在所述确定步骤已经确定该数据的再现时间已经匹配了该时间信息时，识别出具有时间信息的标记信息；用于当该标记信息具有并且在所述识别步骤识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现；以及执行与在所述通知步骤通知的参数信息相对应的处理。

[0010] 本发明是一种程序记录介质，其上已经记录了使计算机处理记录在数据记录介质上的记录数据的程序，该记录数据包括其中编码了数据的编码数据和代表数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴上的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为该事件的参数的参数信息，该程序包括步骤：确定与该播放列表相对应地再现的数据的再现时间是否匹配该时间信息；当在所述确定步骤已经确定该数据的再现时间已经匹配了该时间信息时，识别出具有时间信息的标记信息；用于当该标记信息具有并且在所述识别步骤识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现；以及执行与在所述通知步骤通知的参数信息相对应的处理。

[0011] 本发明是一种其上已经记录了记录数据的数据记录介质。该记录数据包括其中编码了数据的编码数据和代表该数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴上的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为事件的参数的参数信息。

[0012] 本发明是一种数据结构，其包括其中编码了数据的编码数据和代表该数据的再现过程的播放列表，该播放列表包括播放列表标记信息，该播放列表标记信息具有作为在该播放列表时间轴上的标记的标记信息，该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示标记信息的类型的类型信息，以及当该类型信息代表其中生成事件的类型时、变为事件的参数的参数信息。

[0013] 在本发明的数据处理设备、数据处理方法、程序、和程序记录介质中，确定与播放列表相对应地再现的数据的再现时间是否与时间信息相匹配。当已经确定该数据的再现时间已经与时间信息相匹配时，识别出具有该时间信息的标记信息。该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示该标记信息类型的类型信息，以及当该类型信息表示生成事件的类型时、变为事件的参数的参数信息。当所识别出的标记

信息中的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现。执行与该参数信息相对应的处理。

[0014] 在本发明的数据记录介质中，已经记录了记录数据，该记录数据包括其中编码了数据的编码数据和表示数据的再现过程的播放列表。该播放列表包括播放列表标记信息，其具有作为在该播放列表的时间轴上的标记的标记信息。该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示该标记信息类型的类型信息，以及当该类型信息表示生成事件的类型时、变为事件的参数的参数信息。

[0015] 在本发明的数据结构中，该数据结构包括其中编码了数据的编码数据和表示数据的再现过程的播放列表。该播放列表包括播放列表标记信息，其具有作为在该播放列表的时间轴上的标记的标记信息。该标记信息包括表示在该播放列表的时间轴上的一个再现时间的时间信息，表示该标记信息类型的类型信息，以及当该类型信息表示生成事件的类型时、变为事件的参数的参数信息。

[0016] 根据本发明，可以高度便利地处理数据。尤其是，可以执行与用于数据的再现时间和参数信息相对应的处理。

附图说明

[0017] 图 1 是示出根据本发明的盘设备的硬件结构示例的方框图；

[0018] 图 2A 和图 2B 是示出 CPU 112 执行的软件模块组的结构示例的方框图；

[0019] 图 3 是示出缓冲器控制模块 215 的结构示例的框图；

[0020] 图 4 是示出盘 101 中的目录的结构示例示意图；

[0021] 图 5 是示出文件“PLAYLIST.DAT”的语法的示意图；

[0022] 图 6 是示出 PlayItem() 的语法的示意图；

[0023] 图 7 是示出 PlayListMark() 的语法的示意图；

[0024] 图 8 是示出 mark_type 的值和 Mark() 的类型之间的关系的示意图；

[0025] 图 9 是示出 PlayList()、PlayItem()、剪辑、以及存储在剪辑流文件中的节目流之间的关系的示意图；

[0026] 图 10 是示出剪辑信息文件 Clip() 的语法的示意图；

[0027] 图 11 是示出 stream_id、private_stream_id、和由此识别的基本流之间的关系的示意图；

[0028] 图 12 是示出 StaticInfo() 的语法的示意图；

[0029] 图 13 是示出 DynamicInfo() 的语法的示意图；

[0030] 图 14 是示出 EP_map() 的语法的示意图；

[0031] 图 15A 和图 15B 是示出 MPEG-2 系统中的节目流、节目流包 (pack)、和节目流包头的语法的示意图；

[0032] 图 16A 和图 16B 是示出 MPEG-2 系统中的 PES 分组的语法的示意图；

[0033] 图 17A、图 17B 和图 17C 是示出 MPEG-2 系统中的 PES 分组的语法的示意图；

[0034] 图 18A 和图 18B 是示出 MPEG-2 系统中的 PES 分组的语法的示意图；

[0035] 图 19A 和图 19B 是示出 MPEG-2 系统中的 PES_packet() 中的 stream_id 值和基本流的属性（类型）的关系的示意图；

[0036] 图 20 是示出盘设备使用的 stream_id 的示意图；

[0037] 图 21 是示出 private_stream1_PES_payload() 的语法的示意图；

[0038] 图 22 是示出 private_stream_id 值和存储在 private_payload() 中的基本流的属性的关系的示意图；

[0039] 图 23 是示出 private_stream2_PES_payload() 的语法的示意图；

[0040] 图 24 是示出 au_information() 的语法的示意图；

[0041] 图 25 是示出文件“PLAYLIST.DAT”的特定示例的示意图；

[0042] 图 26A 和图 26B 是示出剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”的特定示例的示意图；

[0043] 图 27 是示出剪辑信息文件“00001.CLP”中的 EP_map() 的特定示例的示意图；

[0044] 图 28 是示出 PlayList#0 和 PlayList#1 中的 PlayListMark() 的特定示例的示意图；

[0045] 图 29 是描述预再现处理的流程图；

[0046] 图 30 是描述再现处理的流程图；

[0047] 图 31 是描述 PlayItem 改变处理的流程图；

[0048] 图 32 是描述时间码显示处理的流程图；

[0049] 图 33 描述流改变处理的流程图；

[0050] 图 34 是描述缓冲器控制模块 215 的处理的流程图；

[0051] 图 35 是描述缓冲器控制模块 215 的处理的流程图；

[0052] 图 36 是描述视频流读取处理的流程图；

[0053] 图 37 是描述音频流读取处理的流程图；

[0054] 图 38 是描述子标题流读取处理的流程图；

[0055] 图 39 是描述再同步处理的流程图；

[0056] 图 40 是描述标记处理的流程图；

[0057] 图 41 是描述输出属性控制处理的流程图；

[0058] 图 42 是示出在剪辑信息文件“00003.CLP”中描述的一组 pts_change_point 和 DynamicInfo() 的特定示例的示意图；

[0059] 图 43 是描述子标题显示控制处理的流程图；

[0060] 图 44 是描述获取控制处理和背景 / 屏幕保护处理的流程图；

[0061] 图 45 是示出 private_stream2_PES_payload() 的其它语法的示意图；以及

[0062] 图 46 是示出 au_information() 的其它语法的示意图。

具体实施方式

[0063] 以下，将描述本发明的实施方式。在权利要求中的元素和实施例之间的关系如下。该关系表示在实施例中描述了支持在权利要求中阐述的本发明的特定示例。因此，即使未在权利要求的元素中描述一些特定示例，也不意味着这些特定示例不对应于权利要求中的元素。相反，即使在这个部分中将特定示例描述为权利要求中的元素的相应物，这也不意味着这些特定示例不对应于除了权利要求中的元素之外的其它元素。

[0064] 在这个部分中，不意指在权利要求中阐述了与在本发明的实施例中描述的特定示

例相对应的本发明的全部方面。换句话说，在这个部分中的描述对应于在本发明实施例中描述的特定示例。因此，在这个部分中的描述没有否认存在未在当前专利申请的权利要求中阐述的本发明的方面，而且可以提出分案专利申请和 / 或可以作为修改而添加本发明的另外方面。

[0065] 如权利要求 1 所述的数据处理设备是一种处理记录在数据记录介质上的记录数据的数据处理设备（例如，图 1 所示的盘设备），

[0066] 该记录数据包括：

[0067] 其中编码了数据的编码数据；以及

[0068] 表示数据的再现过程的播放列表（例如，图 5 所示的 PlayList()）；

[0069] 该播放列表包括播放列表标记信息（例如，图 5 所示的 PlayListMark()），该播放列表标记信息具有作为在播放列表时间轴上的标记的标记信息（例如，图 7 所示的 Mark()），

[0070] 该标记信息包括：

[0071] 时间信息（例如，图 7 所示的 mark_time_stamp），表示在该播放列表时间轴上的一个再现时间；

[0072] 类型信息（例如，图 7 所示的 mark_type），表示该标记信息的类型；以及

[0073] 参数信息（例如，图 7 所示的 mark_data），其当该类型信息表示生成事件的类型时，变为该事件的参数；

[0074] 该数据处理设备包括：

[0075] 确定装置（例如，如图 2A 和图 2B 所示并且执行在图 40 所示的步骤 S301 处的处理的解码控制模块 214），用于确定与该播放列表相对应地再现的数据的再现时间是否与该时间信息相匹配；

[0076] 识别装置（例如，如图 2A 和图 2B 所示并且执行在图 40 所示的步骤 S302 处的处理的播放器控制模块 212），用于当确定装置已经确定该数据的再现时间已经与该时间信息相匹配时，识别具有时间信息的标记信息；

[0077] 通知装置（例如，图 2A 和图 2B 所示并且执行图 40 所示的步骤 S307 处的处理的播放器控制模块 212），用于当该标记信息具有的并且由识别装置所识别的类型信息表示生成事件的类型时，通知该标记信息所具有的参数信息和该事件的出现；以及

[0078] 执行装置（例如，如图 2A 和图 2B 所示并且执行图 40 所示的步骤 S308 处的处理的脚本控制模块 211），用于执行与由通知装置通知的参数信息相对应的处理。

[0079] 如权利要求 3 所述的数据处理方法是一种处理记录在数据记录介质上的记录数据的数据处理方法，

[0080] 该记录数据包括：

[0081] 其中编码了数据的编码数据；以及

[0082] 表示数据的再现过程的播放列表（例如，图 5 所示的 PlayList()）；

[0083] 该播放列表包括播放列表标记信息（例如，图 5 所示的 PlayListMark()），该播放列表标记信息具有作为在播放列表时间轴上的标记的标记信息（例如，图 7 所示的 Mark()），

[0084] 该标记信息包括：

[0085] 时间信息（例如，图7所示的mark_time_stamp），表示在该播放列表时间轴上的一个再现时间；

[0086] 类型信息（例如，图7所示的mark_type），表示该标记信息的类型；以及

[0087] 参数信息（例如，图7所示的mark_data），其当该类型信息表示生成事件的类型时，变为该事件的参数；

[0088] 该数据处理方法包括步骤：

[0089] 确定与该播放列表相对应地再现的数据的再现时间是否与该时间信息相匹配（例如，在图40所示的步骤S301处）；

[0090] 当在确定步骤已经确定该数据的再现时间已经与该时间信息相匹配时，识别出具有时间信息的标记信息（例如，在图40所示的步骤S302处）；

[0091] 当该标记信息具有的并且在识别步骤处识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现（例如，在图40所示的步骤S307处）；以及

[0092] 执行与在通知步骤处通知的参数信息相对应的处理（例如，在图40所示的步骤S308处）。

[0093] 如权利要求4所述的程序和记录在如权利要求5所述的程序记录介质上的程序中的每一个都是导致计算机处理记录在数据记录介质上的记录数据的程序，

[0094] 该记录数据包括：

[0095] 其中编码了数据的编码数据；以及

[0096] 表示数据的再现过程的播放列表（例如，图5所示的PlayList()）；

[0097] 该播放列表包括播放列表标记信息（例如，图5所示的PlayListMark()），该播放列表标记信息具有作为在播放列表时间轴上的标记的标记信息（例如，图7所示的Mark()），

[0098] 该标记信息包括：

[0099] 时间信息（例如，图7所示的mark_time_stamp），表示在该播放列表时间轴上的一个再现时间；

[0100] 类型信息（例如，图7所示的mark_type），表示该标记信息的类型；以及

[0101] 参数信息（例如，图7所示的mark_data），其当该类型信息表示生成事件的类型时，变为该事件的参数；

[0102] 该程序包括步骤：

[0103] 确定与该播放列表相对应地再现的数据的再现时间是否与该时间信息相匹配（例如，在图40所示的步骤S301处）；

[0104] 当在确定步骤已经确定该数据的再现时间已经与该时间信息相匹配时，识别出具有时间信息的标记信息（例如，在图40所示的步骤S302处）；

[0105] 当该标记信息具有的并且在识别步骤处识别的类型信息代表生成事件的类型时，通知该标记信息所具有的参数信息和事件的出现（例如，在图40所示的步骤S307处）；以及

[0106] 执行与在通知步骤处通知的参数信息相对应的处理（例如，在图40所示的步骤S308处）。

[0107] 如权利要求 6 所述的数据记录介质是其上已经记录了记录数据的数据记录介质，

[0108] 该记录数据包括：

[0109] 其中编码了数据的编码数据；以及

[0110] 表示数据的再现过程的播放列表（例如，图 5 所示的 PlayList()）；

[0111] 该播放列表包括播放列表标记信息（例如，图 5 所示的 PlayListMark()），该播放列表标记信息具有作为在播放列表时间轴上的标记的标记信息（例如，图 7 所示的 Mark()），

[0112] 该标记信息包括：

[0113] 时间信息（例如，图 7 所示的 mark_time_stamp），表示在该播放列表时间轴上的一个再现时间；

[0114] 类型信息（例如，图 7 所示的 mark_type），表示该标记信息的类型；以及

[0115] 参数信息（例如，图 7 所示的 mark_data），其当该类型信息表示生成事件的类型时，变为该事件的参数；

[0116] 如权利要求 7 所述的数据结构是这样的数据结构，其包括：

[0117] 其中编码了数据的编码数据；以及

[0118] 表示数据的再现过程的播放列表（例如，图 5 所示的 PlayList()）；

[0119] 该播放列表包括播放列表标记信息（例如，图 5 所示的 PlayListMark()），该播放列表标记信息具有作为在播放列表时间轴上的标记的标记信息（例如，图 7 所示的 Mark()），

[0120] 该标记信息包括：

[0121] 时间信息（例如，图 7 所示的 mark_time_stamp），表示在该播放列表时间轴上的一个再现时间；

[0122] 类型信息（例如，图 7 所示的 mark_type），表示该标记信息的类型；以及

[0123] 参数信息（例如，图 7 所示的 mark_data），其当该类型信息表示生成事件的类型时，变为该事件的参数；

[0124] 以下，将参考附图对本发明的实施例进行描述。

[0125] 【硬件结构】

[0126] 图 1 示出根据本发明实施例的盘设备的硬件结构的方框图。

[0127] 图 1 所示的盘设备可以应用于例如盘播放器、游戏设备、汽车导航系统，等等。

[0128] 在图 1 所示的盘设备 101 中，盘 101 是诸如 DVD 的光盘、磁光盘、磁盘等。诸如视频数据、音频数据以及子标题数据的内容数据以及再现这些数据所需的附加数据记录在盘 101 上。

[0129] 需要时，记录在盘 101 上的数据（称为记录数据）包括可以由计算机执行的程序。根据本发明的实施例，作为记录介质，而使用作为盘形记录介质的盘 101。作为替换，该记录介质还可以是半导体存储器或带状记录介质。可以将从在远程位置处的盘 101 读取的数据传输和输入到图 1 所示的盘设备。换言之，可以由连接到盘设备的另一设备从盘 101 读取数据。盘设备可以接收并处理由所述另一设备读取的数据。此外，盘设备还可以通过如因特网的网络从存储类似于记录在盘 101 上的那些数据的数据的服务器等接收数据，并对接收的数据进行处理。此外，盘设备还可以从如服务器等的另一设备接收数据，将所接收的数

据记录到盘 101 中，并然后对记录到盘 101 的数据进行处理。

[0130] 可以向和从盘驱动器 102 加载和卸载盘 101。盘驱动器 102 具有内置接口（未示出）。盘驱动器 102 通过该内置接口连接到驱动器接口 114。盘驱动器 102 驱动盘 101，按照例如读命令从盘 101 读数据，和向驱动器接口 114 提供数据。

[0131] 中央处理单元 (CPU)112、存储器 113、驱动器接口 114、输入接口 115、视频解码器 116、音频解码器 117、视频输出接口 118、以及音频输出接口 119 连接到总线 111。

[0132] CPU 112 和存储器 113 构成计算机系统。换言之，CPU 112 执行软件模块组，该软件模块组是存储在存储器 113 中用于控制整个盘设备并执行以下将描述的各种处理的程序。存储器 113 还存储 CPU 112 执行的软件模块组。此外，存储器 113 暂时存储操作 CPU 112 所需的数据。存储器 113 可以仅由非易失性存储器构成，或者可以由易失性存储器和非易失性存储器组合而成。当图 1 所示的盘设备具有其上记录（安装）有 CPU 112 执行的软件模块组的硬盘时，存储器 113 可以仅由非易失性存储器构成。

[0133] CPU 112 执行的程序（软件模块组）可以预先记录（存储）在作为内置于盘设备中的记录介质的存储器 113 中。

[0134] 可替换地，还可以将程序暂时地或永久地存储（记录）到盘 101 或可移动记录介质（如软盘、紧密盘只读存储器 (CD-ROM)、磁光 (MO) 盘、磁盘、或存储卡）中。该可移动记录介质可以作为所谓的封装软件而提供。

[0135] 程序可以预先存储在存储器 113 中，或从上述可移动记录介质安装到盘设备中。可替换地，可以通过用于数字卫星广播的卫星将程序从下载站点无线地传输到盘设备，或者通过局域网 (LAN) 或如因特网的网络非无线地传输到盘设备。盘设备通过输入接口 115 接收程序，并将程序安装到内置存储器 113 中。

[0136] 程序可由一个 CPU 执行，或者由多个 CPU 分布地执行。

[0137] 驱动器接口 114 在 CPU 112 的控制下控制盘驱动器 102。盘驱动器 102 通过总线 111 将从盘 101 读取的数据提供给 CPU 112、存储器 113、视频解码器 116、以及音频解码器 117。

[0138] 输入接口 115 接收与用户对键（按钮）和遥控器（未示出）的操作相对应的信号，并通过总线 111 将这些信号提供给 CPU 112。输入接口 115 还起用于调制解调器（包括非对称数字用户线 (ADSL) 调制解调器）、网络接口卡 (NIC) 等的通信接口的作用。

[0139] 视频解码器 116 对由盘驱动器 102 从盘 101 读取、并通过驱动器接口 114 和总线 111 提供给视频解码器 116 的编码的视频数据（编码的音频数据）进行解码，并通过总线 111 将解码的视频数据提供给 CPU 112 和视频输出接口 118。

[0140] 音频解码器 117 对由盘驱动器 102 从盘 101 读取、并通过驱动器接口 114 和总线 111 提供给音频解码器 117 的编码的音频数据进行解码，并通过总线 111 将解码的音频数据提供给 CPU 112 和音频输出接口 119。

[0141] 视频输出接口 118 对通过总线 111 提供的视频数据执行预定的处理，并从视频输出端 120 输出处理后的视频数据。音频输出接口 119 对通过总线 111 提供的音频数据执行预定的处理，并从音频输出端 121 输出处理后的音频数据。

[0142] 视频输出端 120 连接到如阴极射线管 (CRT) 或液晶板（未示出）的视频输出设备。由此，从视频输出端 120 输出的视频数据被提供给视频输出设备并由其显示。音频输出端

121 连接到如扬声器和放大器（未示出）的音频输出设备。由此，将从音频输出终端 121 输出的音频数据提供给音频输出设备并由此输出该音频数据。

[0143] 视频数据和音频数据可以有线或者无线地从盘设备提供到视频输出设备和音频输出设备。

[0144] 【软件模块组的结构】

[0145] 图 2A 和图 2B 示出了图 1 中的 CPU 112 执行的软件模块组的结构示例。

[0146] CPU 112 执行的软件模块组主要分为操作系统 (OS)201 和作为应用程序的视频内容再现程序 210。

[0147] 【操作系统 201】

[0148] 当盘设备被接通时，操作系统 201 启动 (CPU 112 执行操作系统 201)，执行如初始设置的预定处理，并调用作为应用程序的视频内容再现程序 210。

[0149] 操作系统 201 向视频内容再现程序 210 提供诸如文件读取服务之类的基础服务。换句话说，操作系统 201 提供了这样的服务，其相对于从视频内容再现程序 210 接收的文件读取请求、通过驱动器接口 114 操作盘驱动器 102，从盘 101 中读出数据，并且将数据提供给视频内容再现程序 210。此外，操作系统 201 还解释该文件系统。

[0150] 操作系统 201 具有多任务处理功能。换言之，操作系统 201 可以在分时的基础上同时（表面上 (apparently)）操作多个软件模块。换言之，尽管视频内容再现程序 210 由若干软件模块组成，但它们可以并行地运行。

[0151] 【视频内容再现程序 210】

[0152] 视频内容再现程序 210 包括脚本控制模块 211、播放器控制模块 212、内容数据提供模块 213、解码控制模块 214、缓冲器控制模块 215、视频解码器控制模块 216、音频解码器控制模块 217、子标题解码器控制模块 218、图形处理模块 219、视频输出模块 220、以及音频输出模块 221。

[0153] 视频内容再现程序 210 是扮演从盘 101 再现数据的关键角色的软件。当将盘 101 装入（插入）盘设备 102 时，视频内容再现程序 210 检查盘 101 是否是已经以预定格式（将在稍后描述）在其上记录了内容的盘。视频内容再现程序 210 从盘 101 读取脚本文件（将在以下进行描述），执行该脚本，从盘 101 读取再现内容所需的元数据（数据库信息）文件，并根据元数据来控制内容的再现。

[0154] 以下，将描述图 2A 和 2B 中示出的构成视频内容再现程序 210 的软件模块。在图 2A 和 2B 中，一般地，实线箭头标记代表内容数据，而虚线箭头标记代表控制数据。

[0155] 【脚本控制模块 211】

[0156] 脚本控制模块 211 解释并执行记录在盘 101 上的脚本程序（脚本）。脚本程序可以描述诸如“操作图形处理模块 219 以创建诸如菜单之类的图像并显示该图像”、“与从诸如遥控器之类的用户接口 (UI) 提供的信号相对应地改变菜单显示（例如，在菜单上移动光标）”、以及“控制该播放器控制模块 212”之类的操作。

[0157] 【播放器控制模块 212】

[0158] 播放器控制模块 212 参考盘 101 上记录的元数据（数据库信息）等来控制对记录在盘 101 上的内容的再现。换言之，播放器控制模块 212 对记录在盘 101 上的 PlayList() 和 Clip() 进行分析，并根据分析结果控制内容数据提供模块 213、解码控制模块 214 和缓冲

器控制模块 215。此外，如稍后将要描述的那样，播放器控制模块 212 根据从脚本控制模块 211 和输入接口 115 接收的命令执行改变要再现的流的流改变控制。此外，播放器控制模块 212 从解码控制模块 214 获得时间，显示该时间，并执行用于标记 (Mark()) 的处理（以下将描述）。

[0159] 【内容数据提供模块 213】

[0160] 内容数据提供模块 213 在播放器控制模块 212 的控制下、或者根据存储在缓冲器控制模块 215 中的数据量，请求操作系统 201 从盘 101 中读取内容数据、元数据等。

[0161] 将操作系统 201 根据从内容数据提供模块 213 接收的请求而已经从盘 101 读取的元数据等提供给预定模块。另一方面，将操作系统 201 根据从内容数据提供模块 213 接收的请求而已经从盘 101 读取的内容数据提供给缓冲器控制模块 215。

[0162] 【解码控制模块 214】

[0163] 解码控制模块 214 在播放器控制模块 212 的控制下控制视频解码器控制模块 216、音频解码器控制模块 217、和子标题解码器控制模块 218 的操作。解码控制模块 214 具有对时间进行计数的时间计数部分 214A。解码控制模块 214 管理在视频解码器控制模块 216 控制下输出的视频数据输出与视频数据同步的数据输出的同步。在这种情况下，要与视频数据的输出同步的数据输出是在音频解码器控制模块 217 控制下输出的音频数据。

[0164] 【缓冲器控制模块 215】

[0165] 缓冲器控制模块 215 具有作为图 1 所示的存储器 113 的存储区域一部分的缓冲器 215A。内容数据提供模块 213 根据从操作系统 201 接收的请求，将从盘 101 读取的内容数据暂时存储在缓冲器 215A 中。

[0166] 此外，缓冲器控制模块 215 还分别根据从视频解码器控制模块 216、音频解码器控制模块 217、或子标题解码器控制模块 218 接收的请求，将存储在缓冲器 215A 中的数据提供给视频解码器控制模块 216、音频解码器控制模块 217、或子标题解码器控制模块 218。

[0167] 换言之，缓冲器控制模块 215 具有视频读取功能部分 233、音频读取功能部分 234、和子标题读取功能部分 235，这些将在以下在图 3 中描述。缓冲器控制模块 215 中的视频读取功能部分 233 处理从视频解码器控制模块 216 接收到的数据请求，以将存储在缓冲器 215A 中的数据提供给视频解码器控制模块 216。同样地，缓冲器控制模块 215 中的音频读取功能部分 234 处理从音频解码器控制模块 217 接收到的请求，以将存储在缓冲器 215A 中的数据提供给音频解码器控制模块 217。缓冲器控制模块 215 中的视频读取功能部分 233 处理从子标题解码器控制模块 218 接收的请求，并将存储在缓冲器 215A 中的数据提供给子标题解码器控制模块 218。

[0168] 【视频解码器控制模块 216】

[0169] 视频解码器控制模块 216 对缓冲器控制模块 215 中的视频读取功能部分 233（图 3）进行操作，以一次一个视频存取单位地从缓冲器控制模块 215 的缓冲器 215A 中读取编码的视频数据，并且将该视频数据提供给图 1 所示的视频解码器 116。此外，视频解码器控制模块 216 控制该视频解码器 116，以便一次一个视频存取单位地解码数据。此外，视频解码器控制模块 216 将由视频解码器 116 解码的视频数据提供给图形处理模块 219。

[0170] 一个视频存取单位例如是视频数据中的一个画面（一帧或者一场）。

[0171] 【音频解码器控制模块 217】

[0172] 音频解码器控制模块 217 对缓冲器控制模块 215 中的音频读取功能部分 234(图 3)进行操作,以一次一个音频存取单位地从缓冲器控制模块 215 的缓冲器 215A 中读取编码的音频数据,并且将编码的音频数据提供给图 1 所示的音频解码器 117。音频解码器控制模块 217 控制该音频解码器 117,以便一次一个音频存取单位地对编码的音频数据进行解码。此外,音频解码器控制模块 217 将由音频解码器 117 解码的音频数据提供给音频输出模块 221。

[0173] 一个音频存取单位是预定量的音频数据(例如,与一个画面同步地输出的数据量)。根据这个实施例,假定一个音频存取单位是预定的固定长度。

[0174] 【子标题解码器控制模块 218】

[0175] 子标题解码器控制模块 218 操作缓冲器控制模块 215 中的子标题读取功能部分 235(图 3),以便一次一个子标题存取单位地从缓冲器控制模块 215 的缓冲器 215A 中读取编码的子标题数据。此外,子标题解码器控制模块 218 具有子标题解码软件(未示出)。子标题解码软件解码从缓冲器 215A 读取的数据。子标题解码器控制模块 218 将解码的子标题数据(子标题的图像数据)提供给图形处理模块 219。

[0176] 一个子标题存取单位是预定量的子标题数据(例如,与一个画面同步地输出的数据量)。根据这个实施例,假定在其开始处描述了一个子标题存取单位的大小。

[0177] 【图形处理模块 219】

[0178] 图形处理模块 219 在播放器控制模块 212 的控制下(根据命令)放大或缩小从子标题解码器控制模块 218 接收的子标题数据,并将放大或缩小的子标题数据添加(重叠)到从视频解码器控制模块 216 接收的视频数据上。图形处理模块 219 放大或缩小已添加了子标题数据的视频数据的大小(图像帧),使得添加(重叠)的视频数据的帧的大小与图 1 所示的、连接到视频输出端 120 的视频输出设备的屏幕相匹配。添加(重叠)的视频数据输出到视频输出模块 220。

[0179] 此外,图形处理模块 219 根据脚本控制模块 211 和播放器控制模块 212 的命令(在其控制下)产生菜单、消息等,并将菜单、消息等与输出视频数据相重叠。

[0180] 此外,图形处理模块 219 根据图 1 所示的连接到视频输出端 120 的视频输出设备的纵横比和表示记录在盘 101 上的视频数据的纵横比的信息来转换输出到视频输出模块 220 的视频数据的纵横比。

[0181] 换言之,当视频输出设备的纵横比为 16:9 时,如果表示视频数据纵横比的信息为 4:3,则图形处理模块 219 在横向(水平方向)上对输出到视频输出模块 220 的视频数据执行挤压(缩小)处理,使得视频数据的左端和右端为黑,并输出得到的视频数据。当视频输出设备的纵横比为 4:3 时,如果表示视频数据纵横比的信息为 16:9,则图形处理模块 219 在纵向(垂直方向)上对输出到视频输出模块 220 的视频数据执行挤压(缩小)处理,使得视频数据的上端和下端为黑,并输出得到的视频数据。

[0182] 当视频输出设备的纵横比与表示视频数据纵横比的信息相同时,如为 4:3 或 16:9,则图形处理模块 219 向视频输出模块 220 输出未经挤压的视频数据。

[0183] 此外,图形处理模块 219 根据从例如播放器控制模块 212 接收的请求而捕获正被处理的视频数据。此外,图形处理模块 219 存储捕获的视频数据或将该视频数据提供给播放器控制模块 212。

[0184] 【视频输出模块 220】

[0185] 视频输出模块 220 排他地占据图 1 所示存储器 113 的一部分，作为先进先出(FIFO)220A(缓冲器)，并暂时存储从图形处理模块 219 接收的视频数据。此外，视频输出模块 220 频繁地从 FIFO 220A 读取视频数据，并将视频数据输出到视频输出端 120(图 1)。

[0186] 【音频输出模块 221】

[0187] 音频输出模块 221 排他地占据图 1 所示存储器 113 的一部分，作为 FIFO221A(缓冲器)，并暂时存储从音频解码器控制模块 217(音频解码器 117) 接收的音频数据。此外，音频输出模块 221 频繁地从 FIFO 221A 读取音频数据，并将音频数据输出到音频输出端 121(图 1)。

[0188] 此外，当从音频解码器控制模块 217 接收的音频数据是具有“主音频”数据和“子音频”数据的左、右声道的双(双语)模式音频数据时，音频输出模块 221 根据预先指定的音频输出模式，而将从音频解码器控制模块 217 接收的音频数据输出到音频输出端 121。

[0189] 换言之，如果已经将“主声音”指定为音频输出模式，音频输出模块 221 就将从音频解码器控制模块 217 接收的音频数据的左声道复制为音频数据的右声道，并将音频数据(“主音频”数据)的左和右声道输出到音频输出端 121。如果已经将“子音频”指定为音频输出模式，音频输出模块 221 就将从音频解码器控制模块 217 接收的音频数据的右声道复制为左声道，并将左和右声道(“子音频”数据)输出到音频输出端 121。如果已经将“主和子音频”二者指定为音频输出模式，则音频输出模块 221 就将从音频解码器控制模块 217 接收的音频数据直接输出到音频输出端 121。

[0190] 如果从音频解码器控制模块 217 接收的音频数据为立体声模式音频数据，则音频输出模块 221 将从音频解码器控制模块 217 接收的音频数据直接输出到音频输出端 121，而不考虑已经指定了什么音频输出模式。

[0191] 用户可以利用遥控器在用于由视频内容再现程序 210 生成的菜单的屏幕上交互地指定音频输出模式。

[0192] 【缓冲器控制模块 215 的结构】

[0193] 图 3 示出了图 2A 和图 2B 示出的缓冲器控制模块 215 的结构示例。

[0194] 缓冲器控制模块 215 排他地使用图 1 所示的存储器 113 的一部分作为缓冲器 215A，并将从盘 101 读取的数据暂存在缓冲器 215A 中。此外，缓冲器控制模块 215 还从缓冲器 215A 读取数据并将数据提供给图 2A 和 2B 中示出的视频解码器控制模块 216、音频解码器控制模块 217、和子标题解码器控制模块 218。

[0195] 除了缓冲器 215A，缓冲器控制模块 215 还具有作为存储器 113 一部分的数据开始指针存储部分 231 和数据写指针存储部分 232。此外，缓冲器控制模块 215 具有作为内部模块的视频读取功能部分 233、音频读取功能部分 234、和子标题读取功能部分 235。

[0196] 缓冲器 215A 例如是连续地存储从盘 101 读取的数据的环形缓冲器。在缓冲器 215A 读取了其存储量的数据之后，缓冲器 215A 以所谓的无限循环方式储存数据，以便该缓冲器 215A 将最新的数据叠加在该最老的数据上。

[0197] 数据开始指针存储部分 231 存储了数据开始指针，其代表在存储在缓冲器 215A 的数据中的、未从缓冲器 215A 读取的最老数据的位置(地址)。

[0198] 数据写指针存储部分 232 存储了这样的指针，其代表缓冲器 215A 中的、从盘 101

读取的最新数据的位置。

[0199] 只要将从盘 101 读取的数据存储到缓冲器 215A 中，就沿图 3 所示的顺时针方向来更新数据写指针所代表的位置。只要从缓冲器 215A 读取数据，就沿图 3 所示的顺时针方向来更新数据开始指针所代表的位置。因此，存储在缓冲器 215A 中的有效数据沿图 3 所示的顺时针方向、从数据开始指针所代表的位置到数据写指针所代表的位置。

[0200] 视频读取功能部分 233 对应于从图 2A 和 2B 示出的视频解码器控制模块 216 接收的请求，而从缓冲器 215A 读取视频流（视频数据的基本流），并将该视频流提供给视频解码器控制模块 216。音频读取功能部分 234 对应于从音频解码器控制模块 217 接收的请求，而从缓冲器 215A 读取音频流（音频数据的基本流），并将该音频流提供给音频解码器控制模块 217。类似地，子标题读取功能部分 235 对应于从子标题解码器控制模块 218 接收的请求，而从缓冲器 215A 读取子标题（子标题数据的基本流），并将该子标题流提供给子标题解码器控制模块 218。

[0201] 换言之，已经在盘 101 上记录了与例如活动图像专家组 (MPEG)2 标准相对应的节目流，该节目流被称为 MPEG2- 系统节目流。在该节目流中，已经在时分基础上多路复用了视频流、音频流、和子标题流中的至少一个基本流。视频读取功能部分 233 具有用于该节目流的解多路复用功能。视频读取功能部分 233 从存储在缓冲器 215A 中的节目流中解多路复用出视频流，并且读取该视频流。

[0202] 同样地，音频读取功能部分 234 具有用于节目流的解多路复用功能。音频读取功能部分 234 从存储在缓冲器 215A 中的节目流中解多路复用出音频流，并且读取该音频流。同样地，子标题读取功能部分 235 具有用于节目流的解多路复用功能。子标题读取功能部分 235 从存储在缓冲器 215A 中的节目流中解多路复用出子标题流，并且读取该子标题流。

[0203] 视频读取功能部分 233 具有作为如图 1 所示的存储器 113 的一部分的视频读指针存储部分 241、stream_id 寄存器 242、和 au_information() 寄存器 243。

[0204] 视频读指针存储部分 241 存储表示视频流在缓冲器 215A 中的位置（地址）的视频读指针。视频读取功能部分 233 从缓冲器 215A 中的视频读指针的位置读取作为视频流的数据。stream_id 寄存器 242 存储 stream_id，该 stream_id 用于分析存储在缓冲器 215A 中的节目流并用于标识从节目流读取的视频流。au_information() 寄存器 243 存储 au_information()，其中 au_information() 是从缓冲器 215A 中读取视频流所必需的数据（也就是说用于读取该视频流的数据）。

[0205] 音频读取功能部分 234 具有作为图 1 所示存储器 113 一部分的音频读指针存储部分 251、stream_id 寄存器 252、和 private_stream_id 寄存器 253。

[0206] 音频读指针存储部分 251 存储代表存储在缓冲器 215A 中的音频流的位置（地址）的音频读指针。音频读取功能部分 234 从缓冲器 215A 中的音频读指针的位置读取作为音频流的数据。stream_id 寄存器 252 和 private_stream_id 寄存器 253 分别存储 stream_id 和 private_stream_id（将在稍后描述），其中 stream_id 和 private_stream_id 用于分析存储在缓冲器 215A 中的节目流以及标识从节目流中读取的音频流。

[0207] 子标题读取功能部分 235 具有作为图 1 所示存储器 113 一部分的子标题读取功能标志存储部分 261、子标题读指针存储部分 262、stream_id 寄存器 263、和 private_stream_id 寄存器 264。

[0208] 子标题读取功能标志存储部分 261 存储子标题读取功能标志。当存储在子标题读取功能标志存储部分 261 中的子标题读取功能标志是例如“0”时，子标题读取功能部分 235 不进行操作。当存储在子标题读取功能标志存储部分 261 中的子标题读取功能标志是例如“1”时，子标题读取功能部分 235 进行操作。

[0209] 子标题读指针存储部分 262 存储代表存储在缓冲器 215A 中的子标题流的位置（地址）的子标题读指针。子标题读取功能部分 235 从缓冲器 215A 中的子标题读指针的位置读取作为子标题流的数据。stream_id 寄存器 263 和 private_stream_id 寄存器 264 分别存储 stream_id 和 private_stream_id（将在稍后描述），其中 stream_id 和 private_stream_id 用于分析存储在缓冲器 215A 中的节目流以及标识从节目流中读取的子标题流。

[0210] 【对记录在盘 101 上的数据的数据格式的描述】

[0211] 接下来，将描述记录在盘 101 上的数据的数据格式。

[0212] 图 4 示意性地示出了盘 101 的目录结构。

[0213] 用于盘 101 的文件系统例如是国际标准化组织 (ISO)-9660 和通用盘格式 (UDF) (http://www.osta.org/specs/) 定义的文件系统中的一种。以目录结构分级地管理记录在盘 101 上的数据文件。可用于盘 101 的文件系统不限于这些文件系统。

[0214] 在图 4 中，在代表文件系统的基础的根目录下有一个“视频”目录。在该“视频”目录下，有“剪辑”目录和“流”目录两个目录。

[0215] 除“剪辑”目录和“流”目录这两个目录之外，在“视频”目录下还有两个数据文件，即文件“SCRIPT.DAT”和文件“PLAYLIST.DAT”。

[0216] “SCRIPT.DAT”文件是描述脚本程序的脚本文件。换言之，“SCRIPT.DAT”文件描述允许交互地再现盘 101 上的数据的脚本程序。存储在“SCRIPT.DAT”文件中的脚本程序由图 2A 和 2B 所示的脚本控制模块 211 解释和执行。

[0217] “PLAYLIST.DAT”文件存储至少一个播放列表（将在以下参考图 5 描述的 PlayList()）。播放列表描述了诸如视频数据之类、记录在盘 101 上的内容的再现过程。

[0218] 在“剪辑”目录下至少有一个剪辑信息文件。在“流”目录下至少有一个剪辑流文件。换言之，在“剪辑”目录下有三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”。在“流”目录下有三个剪辑流文件“00001.PS”、“00002.PS”、和“00003.PS”。

[0219] 剪辑流文件存储节目流，在该节目流中，视频数据、音频数据以及子标题数据中的至少一个被压缩并编码，并在时分的基础上被多路复用。

[0220] 剪辑信息文件存储例如剪辑流的特征之类的、有关剪辑流的（文件）元数据。

[0221] 换言之，剪辑流文件和剪辑信息文件以一一对应的关系相关。在图 4 中，与五个数字 + 句点 +″ PS″ 的命名规则相对应地命名剪辑流文件，而与和对应的剪辑流相同的五个数字 + 句点 + “CLP”的命名规则相对应地来命名剪辑信息文件。

[0222] 因此，可以通过文件名的扩展名（句点右侧）标识出剪辑流文件和剪辑信息文件。此外，可以利用除了扩展名之外的文件名（句点的左侧部分）确定剪辑流文件和剪辑信息文件是否相关。

[0223] 以下，将详细描述记录在盘 101 上的文件。

[0224] 【PLAYLIST.DAT】

[0225] 图 5 示出图 4 中“视频”目录下的“PLAYLIST.DAT”文件的内部结构（语法）。

[0226] 在图 5 中,“PLAYLIST.DAT”文件具有“语法”字段,用于描述 PLAYLIST.DAT 文件的数据结构 ;“位数”字段,用于描述“语法”字段中每个数据项的位长度 ;“助记符”字段,其中“bslbf”(左边位开头的位串)和“uimsbf”(最高有效位开头的无符号整数)代表在“语法”字段中的数据项从左边的位开始移位、以及字段“语法”中的数据项是无符号整数并且从最高有效位开始移位。这些条件适用于与图 5 类似的其它列表。

[0227] 文件“PLAYLIST.DAT”从描述诸如名称(文件名)之类的信息的 name_length(8 位)和 name_string(255 字节)开始。

[0228] 换句话说,name_length 以字节为单位表示紧跟其后的 name_string 的大小。name_string 代表文件“PLAYLIST.DAT”的名称(文件名)。

[0229] 从 name_string 的开头开始的 name_length 个字节用作有效名。当 name_length 的值为 10 时,则从 name_string 开头开始的 10 个字节被解释为有效名称。

[0230] name_string 后跟随的是 number_of_PlayLists(16 位)。number_of_PlayLists 代表跟在 name_string 后的 PlayList() 的数量。number_of_PlayLists 后跟随由 number_of_PlayLists 代表的 PlayList()。

[0231] PlayList() 是描述对记录在盘 101 上的剪辑流文件的再现过程的播放列表。PlayList() 的内部结构如下 :

[0232] PlayList() 从 PlayList_data_length(32 位)开始。PlayList_data_length 代表 PlayList() 的大小。

[0233] PlayList_data_length 后顺序地跟随 reserved_for_word_alignment(15 位)和 capture_enable_flag_PlayList(1 位)。15 位的 reserved_for_word_alignment 之后跟随 1 位的 capture_enable_flag_PlayList,以用于在 capture_enable_flag_PlayList 的位置处的字定位,以便将其置于 16 位的位置处。capture_enable_flag_PlayList 是 1 位的标志,代表是否允许与 PlayList() 相对应地再现的视频流的视频数据(属于 PlayList() 的视频数据)在从盘 101 中再现数据的盘设备中被二次使用。当 capture_enable_flag_PlayList 的值例如为 1(0 或 1) 时,这代表允许属于 PlayList() 的视频数据被二次使用。当 capture_enable_flag_PlayList 的值例如为 0(0 或 1) 时,这代表不允许属于 PlayList() 的视频数据被二次使用(即禁止被二次使用)。

[0234] 在图 5 中,capture_enable_flag_PlayList 由 1 位构成。作为选择,capture_enable_flag_PlayList 还可以由多位构成。在这种情况下,可以逐渐地允许二次使用属于 PlayList() 的视频数据。换言之,capture_enable_flag_PlayList 可以由 2 位组成。当 capture_enable_flag_PlayList 的值为 00B(其中 B 代表前面的数字为二进制数)时,禁止二次使用视频数据。当 capture_enable_flag_PlayList 的值为 01B 时,允许二次使用缩小为 64×64 像素大小或者更小的视频数据。当 capture_enable_flag_PlayList 的值是 10B 时,允许二次使用视频数据而没有任何尺寸缩小。

[0235] 可以相对于应用而不是大小来限制对视频数据的二次使用。换言之,当 capture_enable_flag_PlayList 的值为 01B 时,仅仅可以允许视频内容再现程序 210(图 2A 和图 2B)二次使用该视频数据。当 capture_enable_flag_PlayList 的值是 10B 时,可以允许图 1 所示的盘设备中的包括视频内容再现程序 210 在内的任何应用二次使用该视频数据。在这个示例中,在图 1 所示的盘设备中的除了视频内容再现程序 210 之外的应用显示壁纸(背景)

或者屏幕保护程序。

[0236] 当 capture_enable_flag_PlayList 由 2 位组成时，跟随其后的 reserved_for_word_alignment 由 14 位组成，以便进行字对准。

[0237] 作为选择，利用 capture_enable_flag_PlayList，可以允许在在盘设备的外面二次使用视频数据。在利用 capture_enable_flag_PlayList 的情况下，允许在盘设备的外面二次使用视频数据，该视频数据被记录到例如可以装载到盘设备中或者可以连接到盘设备的记录介质中，或者通过诸如国际互联网之类的网络传输（分配）到另一个设备。在这种情况下，可以将代表可将视频数据记录到记录介质中的次数或者可以传输（分配）该视频数据的次数的信息添加到视频数据中。

[0238] capture_enable_flag_PlayList 之后连续地跟随 PlayList_name_length(8 位）和 PlayList_name_string(255 字节）。PlayList_name_length 以字节为单位表示 PlayList_name_string 的大小。PlayList_name_string 表示 PlayList() 的名称。

[0239] PlayList_name_string 之后跟随 number_of_PlayItems(16 位）。number_of_PlayItems 表示 PlayItem() 的数量。

[0240] number_of_PlayItems 之后跟随由 number_of_PlayItems 所表示的数量的 PlayItem()。

[0241] 一个 PlayList() 可以描述以 PlayItem() 为单位的内容的再现过程。

[0242] PlayList() 中唯一的标识 (ID) 码被添加到由 number_of_PlayItems 表示的数量的 PlayItem() 中。换言之，PlayList() 的第一个 PlayItem() 由数字 0 标识。其它 PlayItem() 顺序地由数字 1、2、... 等标识。

[0243] 在由 number_of_PlayItems 表示的数量的 PlayItem() 后面跟随一个 PlayListMark()。PlayListMark() 是作为与 PlayList() 对应的再现时间轴上的标记的一组 Mark()。以下将参考图 7 对 PlayListMark() 进行详细描述。

[0244] 【对 PlayItem() 的描述】

[0245] 图 6 示出了包括在图 5 所示的 PlayList() 中的 PlayItem() 的内部结构。

[0246] PlayItem() 以 length(16 位）开始。length 表示包括 length 的大小在内、PlayItem() 的大小。

[0247] 在 length 后面依次跟随有 Clip_Information_file_name_length(16 位）和 Clip_Information_file_name（可变长度）。Clip_Information_file_name_length 以字节为单位表示 Clip_Information_file_name 的大小。Clip_Information_file_name 表示与由 PlayItem() 再现的剪辑流文件（图 4 所示的具有扩展名 PS 的文件）相对应的剪辑信息文件（图 4 所示的具有扩展名 CLP 的文件）的文件名。与上述剪辑流文件和剪辑信息文件的命名规则相对应，可以利用 Clip_Information_file_name 识别出由 PlayItem() 再现的剪辑信息文件的文件名，并且可以标识出剪辑流文件。

[0248] 在 Clip_Information_file_name 后依次跟随有 IN_time(32 位）和 OUT_time(32 位）。

[0249] IN_time 和 OUT_time 是表示由 Clip_Information_file_name 标识的剪辑流文件的再现开始位置和再现结束位置的时间信息。

[0250] IN_time 可以指定剪辑流文件的中间位置（包括开头）作为再现开始位置。OUT_

time 可以指定剪辑流文件的中间位置（包括结尾）作为再现结束位置。

[0251] PlayItem() 再现由 Clip_Information_file_name 标识的剪辑流文件的从 IN_time 到 OUT_time 的内容。由 PlayItem() 再现的内容有时被称为剪辑。

[0252] 【对 PlayListMark() 的描述】

[0253] 图 7 示出了包括在图 5 所示的 PlayList() 中的 PlayListMark() 的内部结构。

[0254] 如上所述，PlayListMark() 是一组 Mark()，这些 Mark() 是在与图 5 所示的 PlayList() 相对应的再现的时间轴上的标记。Mark() 的数目为 0 或者更大。一个 Mark() 至少具有 :时间信息，其表示在与 PlayList() 相对应地执行的再现的时间轴上的一个时间（位置）;类型信息，表示 Mark() 的类型 ;以及当类型信息表示发生的事件类型时、该事件的参数 (argument) 的参数信息。

[0255] 换言之，PlayListMark() 以 length(32 位）开始。length 表示包括 length 的大小在内、PlayListMark() 的大小。

[0256] 在 length 后面跟随 number_of_PlayList_marks(16 位）。number_of_PlayList_marks 表示跟在 number_of_PlayList_marks 后的 Mark() 的数目。number_of_PlayList_marks 后跟随着由 number_of_PlayList_marks 表示的 Mark()。

[0257] Mark() 以 mark_type(8 位）开始。mark_type 是上述类型信息并且表示 mark_type 所属的 Mark() 的类型。

[0258] 根据该实施方式，Mark() 例如具有章节 (chapter)、索引和事件三种类型。

[0259] 当 Mark() 的类型为章节（有时称为章节标记）时，其是作为划分 (division) PlayList() 的一部分的搜索单位的章节的开始位置的标记。当 Mark() 的类型为索引（有时称为索引标记）时，则其是作为章节的子划分单位的索引的开始位置的标记。当 Mark() 的类型为事件（有时称为事件标记）时，Mark() 是在与 PlayList() 相对应地再现内容时、发生该事件的位置的标记。向脚本控制模块 211 通知已经发生了与事件标记相对应的事件。

[0260] 图 8 示出 mark_type 的值和 Mark() 类型之间的关系。图 8 中，章节标记的 mark_type 为 1 ;索引标记的 mark_type 为 2 ;而事件标记的 mark_type 为 3。图 8 中，由 8 位 mark_type 表示的其它值（即 0 和 4 ～ 255) 被保留以用于将来扩展。

[0261] 在图 7 中，mark_type 后跟随 mark_name_length(8 位）。Mark() 以 mark_name_string(24 字节）结束。mark_name_length 和 mark_name_string 用于描述 Mark() 的名称。mark_name_length 表示 mark_name_string 的有效大小。mark_name_string 表示 Mark() 的名称。因此，从 mark_name_string 的开头开始的 mark_name_length 的字节表示 Mark() 的有效名称。

[0262] mark_name_length 后面跟随有四个元素 ref_to_PlayItem_id(16 位）、mark_time_stamp(32 位）、entry_ES_stream_id(8 位）、以及 entry_ES_private_stream_id(8 位），这些元素将在 PlayList() 中定义的 Mark() 和剪辑流文件相关联。

[0263] ref_to_PlayItem_id 描述作为分配给 Mark() 所属的 PlayItem() 的顺序号的 ID。ref_to_PlayItem_id 标识 Mark() 所属的 PlayItem()（图 6)。由此，如在图 6 所描述的，标识了剪辑信息文件和剪辑流文件。

[0264] mark_time_stamp 表示在由 ref_to_PlayItem_id 标识的剪辑流文件中、Mark() 所

表示的位置（时间）。

[0265] 图 9 示出了 PlayList()、PlayItem()、剪辑以及存储在剪辑流文件中的节目流之间的关系。

[0266] 图 9 中，PlayList() 由三个顺序地编号为 ID#1、ID#2 和 ID#3 的 PlayItem() 组成。在以下的描述中，编号为 ID#i 的 PlayItem() 被表示为 PlayItem#i。

[0267] 图 9 中，作为由 PlayItem#0、PlayItem#1 和 PlayItem#2 再现的内容的剪辑被分别标示为 clip A、clip B 和 clip C。

[0268] 剪辑的实体是在由图 6 所示的 PlayItem() 中的 Clip_Information_file_name 所标识的剪辑流文件（也由剪辑信息文件所标识）中存储的节目流从 IN_time 到 OUT_time 的剪辑。图 9 中，作为剪辑 A、剪辑 B 和剪辑 C 的实体的节目流被分别表示为节目流 A、节目流 B、和节目流 C。

[0269] 在图 9 中，在作为在与 PlayList() 相对应的再现的时间轴上的位置（时间）t0 的标记的 Mark() 中，ref_to_PlayItem_id 和 mark_time_stamp 如下所述。

[0270] 在图 9 中，因为时间 t0 是再现 PlayItem#1 的时间，所以 ref_to_PlayItem_id 描述了作为 PlayItem#1 的 ID 的 1。因为在时间 t0 处再现了作为剪辑 B 实体的节目流 B，所以 mark_time_stamp 描述了与时间 t0 相对应地存储节目流 B 的剪辑流文件的时间。

[0271] 在图 7 中，当 Mark() 与特定基本流相关时，entry_ES_stream_id 和 entry_ES_private_stream_id 用于标识该基本流。换言之，entry_ES_stream_id 描述与 Mark() 相关的基本流的 stream_id（包括基本流的 PES_packet()，将在以下参考图 16A 和 16B 以及 18A 和 18B 描述 PES_packet()）。另一方面，entry_ES_private_stream_id 描述与 Mark() 相关的基本流的 private_stream_id（包括基本流的 private_stream1_PES_payload() 的 private_header()，将在以下参考图 21 描述 private_header()）。

[0272] 当再现其中已经多路复用了视频流 #1 和视频流 #2 的剪辑中的视频流 #1 时以及当在再现视频流 #2 的同时需要改变章节时间时，在正在再现视频流 #2 的同时，在章节标记时间处，在 Mark() 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 中描述视频流 #1 的 stream_id 和 private_stream_id。

[0273] 不与特定基本流相关的 Mark() 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 例如为 0。

[0274] entry_ES_private_stream_id 后跟随 mark_data(32 位)。当 Mark() 是事件标记时，mark_data 是作为利用该事件标记发生的事件的参数的参数信息。当 Mark() 是章节标记或者索引标记时，mark_data 可被用作由章节标记或者索引标记所代表的章节号或者索引号。

[0275] 【对 Clip() 的描述】

[0276] 以下，将描述存储在图 4 所示的“剪辑”目录中的、具有 CLP 扩展名的剪辑信息文件的内部结构。

[0277] 图 4 中，在“剪辑”目录下有三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”。这些剪辑信息文件包括表示存储在“流”目录下的剪辑流文件“00001.PS”、“00002.PS”、和“00003.PS”的特征的元数据。

[0278] 图 10 示出了剪辑信息文件 Clip() 的内部结构。

[0279] 剪辑信息文件 Clip() 以 presentation_start_time 和 presentation_end_time(各为 32 位)。presentation_start_time 和 presentation_end_time 表示与剪辑信息文件 Clip() 相对应的剪辑流文件(存储在其中的节目流)的开始时间和结束时间。剪辑流文件的时间被描述为用作 MPEG2- 系统的时间的 90 千赫的倍数。

[0280] 在 presentation_end_time 后跟随 reserved_for_word_alignment(7 位)和 capture_enable_flag_Clip(1 位)。7 位的 reserved_for_word_alignment 用于字对准。capture_enable_flag_Clip 是与图 5 所示的 capture_enable_flag_PlayList 相似的用于表示是否允许二次使用视频数据的标志。

[0281] 然而,图 5 所示的 capture_enable_flag_PlayList 表示是否允许对应于与 PlayList() 相对应地再现的视频流的视频数据(属于 PlayList())被二次使用。相反,图 10 所示的 capture_enable_flag_Clip 表示是否允许对应于存储在与剪辑信息文件 Clip() 相对应的剪辑流文件中的视频流(视频数据的基本流)的视频数据被二次使用。因此,图 5 所示 capture_enable_flag_PlayList 不同于图 10 所示的 capture_enable_flag_Clip 之处在于,允许二次使用的视频数据的单位(范围)。

[0282] 与图 5 描述的 capture_enable_flag_PlayList 相似,图 10 中描述的 capture_enable_flag_Clip 也可以由多位而非一位组成。

[0283] 在 capture_enable_flag_Clip 后跟随 number_of_streams(8 位)。number_of_streams 描述 StreamInfo() 的数量。因此,在 number_of_streams 后跟随由 number_of_streams 表示的 StreamInfo()。

[0284] StreamInfo() 以 length(16 位)开始。length 表示包括 length 的大小在内、StreamInfo() 的大小。在 length 后面跟随 stream_id(8 位)和 private_stream_id(8 位)。stream_id 和 private_stream_id 标识与 StreamInfo() 相关的基本流。

[0285] 图 11 示出 stream_id、private_stream_id、以及由此标识的基本流之间的关系。

[0286] stream_id 与 MPEG2- 系统标准中定义的相同。MPEG2- 系统标准为基本流(数据)的每个属性(类型)定义了 stream_id 的值。因此,MPEG2- 系统标准中定义的基本流的属性只能由 stream_id 来标识。

[0287] 本实施例能够处理 MPEG2- 系统标准中未定义的基本流的属性。private_stream_id 是标识 MPEG2- 系统标准中未定义的基本流的属性的信息。

[0288] 图 11 示出具有四个属性的基本流的 stream_id 和 private_stream_id 的关系,这些基本流是与在 MPEG 中定义的编码(解码)系统相对应地编码的视频基本流、与自适应变换声学编码 (ATARC) 系统相对应地编码的音频基本流(有时在下文中称为 ATRAC 音频流)、与线性脉码调制 (LPCM) 系统相对应地编码的音频基本流(有时在下文中称为 LPCM 音频流)、以及子标题基本流(有时在下文中称为子标题流)。

[0289] MPEG2- 系统标准定义了与在 MPEG 中定义的编码系统相对应地编码的视频基本流与在从 0xE0 到 0xEF(其中 0x 表示其后的字符串以十六进制表示)范围中的值多路复用。因此,可以用节目流来多路传输与在 MPEG 中定义的编码系统相对应地编码、并且由从 0xE0 到 0xEF 的范围中的 stream_id 所标识的 16 个视频基本流。

[0290] 由于与在 MPEG 中定义的编码系统相对应地编码的视频基本流可以由在从 0xE0 到 0xEF 的范围内的 stream_id 所标识,所以不需要 private_stream_id(可以被忽略)。

[0291] 另一方面，在 MPEG2- 系统中，没有为 ATRAC 音频流、LPCM 音频流、和子标题流定义 stream_id。

[0292] 因此，根据本实施例，对于未在 MPEG2- 系统中定义其 stream_id 的基本流，使用了作为代表 MPEG2- 系统中的属性 private_stream_1 的值的 0xBD。此外，如图 11 所示，由 private_stream_id 标识这些基本流。

[0293] 换句话说，ATRAC 音频流由在从 0x00 到 0x0F 的范围内的 private_stream_id 所标识。因此，可以用节目流来多路传输 16 个 ATRAC 音频流。LPCM 音频流由在从 0x10 到 0x1F 范围中的 private_stream_id 所标识。因此，可以用节目流来多路传输 16 个 LPCM 音频流。子标题流由在从 0x80 到 0x9F 范围内的 private_stream_id 所标识。因此，可以用节目流来多路传输 32 个子标题流。

[0294] 以下将详细描述 stream_id 和 private_stream_id。

[0295] 图 10 中，在 private_stream_id 后顺序地跟随 StaticInfo() 和 reserved_for_word_alignment(8 位)。StaticInfo() 描述在再现由 stream_id 和 private_stream_id 标识的基本流（在包括 StaticInfo() 的 StreamInfo() 中描述的）时发生变化的信息。以下将参考图 12 对 StaticInfo() 进行描述。

[0296] reserved_for_word_alignment 用于字对准。

[0297] reserved_for_word_alignment 后跟随 number_of_DynamicInfo(8 位)。number_of_DynamicInfo 表示跟在 number_of_DynamicInfo 后的 pts_change_point（各为 32 位）和 DynamicInfo() 的集合的数量。

[0298] 因此，number_of_DynamicInfo 后跟随由 number_of_DynamicInfo 表示的 pts_change_point 和 DynamicInfo() 的集合。

[0299] pts_change_point 表示与 pts_change_point 配对的 DynamicInfo() 中的信息变为有效的时间。表示基本流开始时间的 pts_change_point 等于在与存储该基本流的剪辑流文件相对应的剪辑信息文件 Clip() 的开头处描述的 presentation_start_time。。

[0300] DynamicInfo() 描述在再现由 stream_id 和 private_stream_id 标识的基本流时变化的所谓动态信息。在 DynamicInfo() 中描述的信息在由与 DynamicInfo() 配对的 pts_change_point 所代表的再现时间处变为有效。稍后将参考图 13 描述 DynamicInfo()

[0301] 在由 number_of_DynamicInfo 表示的 pts_change_point 和 DynamicInfo() 的集合后跟随 EP_map()。以下将参考图 14 对 EP_map() 进行描述。

[0302] 【对 StaticInfo() 的描述】

[0303] 以下，将参考图 12 对图 10 示出的 StaticInfo() 进行详细描述。

[0304] 图 12 示出了 StaticInfo() 的语法。

[0305] StaticInfo() 的内容取决于对应基本流的属性（类型）而改变。由图 10 所示包括 StaticInfo() 在内的 StreamInfo() 中包括的 stream_id 和 private_stream_id 来确定与 StaticInfo() 相对应的基本流的属性。

[0306] 当对应于 StaticInfo() 的基本流是视频流（流＝＝ VIDEO) 时，StaticInfo() 包括 picture_size(4 位)、frame_rate(4 位)、cc_flag(1 位)、以及用于字对准的 reserved_for_word_alignment。

[0307] picture_size 代表与视频流相对应的视频数据（利用该视频数据显示的图像）的

大小。frame_rate 代表与视频流相对应的视频数据的帧频。cc_flag 代表视频流是否包括封闭的 (closed) 字幕数据。当视频流包括封闭的字幕数据时，cc_flag 为 1。当视频流不包括封闭的字幕数据时，cc_flag 为 0。

[0308] 当对应于 StaticInfo() 的基本流为音频流（流＝＝ AUDIO) 时，StaticInfo() 包括 audio_language_code(16 位）、channel_configuration(8 位）、lfe_existence(1 位）、sampling_frequency(4 位）、以及用于字对准的 reserved_for_word_alignment。

[0309] audio_language_code 描述了代表包括在音频流中的音频数据的语言的代码。channel_configuration 代表包括在音频流中的音频数据的诸如非立体声（单声道）、立体声、多声道之类的属性。lfe_existence 代表音频流是否包括低频效应声道。当音频流包括低频效应声道时，lfe_existence 为 1。当音频流不包括低频效应声道时，lfe_existence 为 0。sampling_frequency 是代表包括在音频流中的音频数据的采样频率的信息。

[0310] 当对应于 StaticInfo() 的基本流是子标题流（流＝＝ SUBTITLE) 时，StaticInfo() 包括 subtitle_language_code(16 位）、configurable_flag(1 位）、以及用于字对准的 reserved_for_word_alignment。

[0311] subtitle_language_code 描述了表示包括在子标题流中的子标题数据的语言的代码。configurable_flag 是代表是否允许从缺省显示模式改变该子标题数据显示模式的信息。当允许显示模式改变时，configurable_flag 为 1。当不允许显示模式改变时，configurable_flag 为 0。子标题数据的显示模式包括子标题数据的显示大小、显示位置、显示颜色、显示模式（例如，闪烁）、显示方向（垂直或者水平）等。

[0312] 【对 DynamicInfo() 的描述】

[0313] 以下，将参考图 13 对图 10 所示 DynamicInfo() 进行详细描述。

[0314] 图 13 示出 DynamicInfo() 的语法。

[0315] DynamicInfo() 以用于字对准的 reserved_for_word_alignment(8 位）开始。跟随在 reserved_for_word_alignment 后面的元素取决于与 DynamicInfo() 对应的基本流的属性。由图 12 所示、包括 DynamicInfo() 的 StreamInfo() 中包括的 stream_id 和 private_stream_id 确定与 DynamicInfo() 相对应的基本流的属性。

[0316] 如图 10 所述，DynamicInfo() 描述了在再现基本流时改变的动态信息。动态信息不是特定的。然而，在图 13 所示的实施例中，在 DynamicInfo() 中描述了与 DynamicInfo() 相对应的基本流的数据，即作为对基本流的处理的输出的数据的输出属性（从基本流获得的数据的输出属性）。

[0317] 具体而言，当与 DynamicInfo() 相对应的基本流是视频流（流＝＝ VIDEO) 时，DynamicInfo() 包括 display_aspect_ratio(4 位）和用于字对准的 reserved_for_word_alignment。display_aspect_ratio 描述了视频流中的视频数据的输出属性（显示模式），例如视频数据的纵横比。换句话说，display_aspect_ratio 描述了代表作为纵横比的 16 ： 9 或者 4 ： 3 的信息。视频流的 DynamicInfo() 可以描述如视频数据中的图像的大小 (X 像素 ×Y 像素）以及纵横比。

[0318] 当与 DynamicInfo() 相对应的基本流是音频流（流＝＝ AUDIO) 时，DynamicInfo() 包括 channel_assignment(4 位）和用于字对准的 reserved_for_word_alignment。当音频流包括两个声道音频数据时，channel_assignment 描述两个声道的输出属性（输出模式）。

换句话说，channel_assignment 描述代表立体声或者双声道（双语）的声道分配的信息。

[0319] 当与 DynamicInfo() 相对应的基本流是子标题流（流＝＝SUBTITLE）时，DynamicInfo() 包括用于字对准的 reserved_for_word_alignment。换句话说，根据图 13 所示的实施例，没有为子标题流定义作为动态信息的输出属性。

[0320] 【对 EP_map() 的描述】

[0321] 以下将参考图 14 对图 10 所示 EP_map() 进行详细描述。

[0322] 图 14 示出 EP_map() 的语法。

[0323] EP_map() 描述了可解码起始（startable）点（入口点）的信息，从该起始点开始可以解码用存储在对应于包括 EP_map() 的剪辑信息文件 Clip() 的剪辑流文件中的节目流多路复用的每个基本流。

[0324] 可以通过计算获得具有固定速率的流中的可解码起始点。然而，对于诸如与 MPEG 标准相对应地编码的视频流之类、其大小在每个视频存取单元中发生改变的流，不能通过计算获得可解码起始点。除非对流进行分析，否则不能获得可解码起始点。为了随机存取数据，必须快速地识别出可解码起始点。利用 EP_map()，可以快速地识别出可解码起始点。

[0325] 在 MPEG2- 视频中，包括 Sequence_header() 等的帧内画面（intra_picture）的开头是可解码起始点。

[0326] EP_map() 从用于字对准的 reserved_for_word_alignment(8 位）开始。reserved_for_word_alignment 之后跟随 number_of_stream_id_entries(8 位）。number_of_stream_id_entries 代表在 EP_map() 中描述可解码起始点信息的基本流的数目。

[0327] number_of_stream_id_entries 之后跟随由 number_of_stream_id_entries 所代表的、标识基本流的信息和基本流的可解码起始点的信息的集合。

[0328] 换句话说，number_of_stream_id_entries 之后跟随作为标识基本流的信息的 stream_id(8 位）和 private_stream_id(8 位）。private_stream_id 之后跟随 number_of_EP_entries(32 位）。number_of_EP_entries 代表由 number_of_EP_entries 之前的 stream_id 和 private_stream_id 所标识的可解码起始点的数目。

[0329] number_of_EP_entries 之后跟随由 number_of_EP_entries 所代表的、作为由 stream_id 和 private_stream_id 所标识的基本流中的可解码起始点的信息 PTS_EP_start（每个 32 位）和 RPN_EP_start（每个 32 位）的集合。

[0330] 作为可解码起始点的一个信息元素的 PTS_EP_start 代表在剪辑流文件中的可解码起始点的时间（再现时间），其中该剪辑流文件存储了用由 stream_id 和 private_stream_id 所标识的基本流多路复用的节目流。

[0331] 作为可解码起始点的另一信息元素的 RPN_EP_start 将可解码起始点在剪辑流文件中的位置描述为节目流中的 pack() 数目的值，其中该剪辑流文件存储用由 stream_id 和 private_stream_id 所标识的基本流多路复用的节目流。根据这个实施例，pack() 的大小是固定的 2048 字节。此外，根据这个实施例，盘 101（图 1）中的一个扇区是 2048 字节。

[0332] 视频流中的可解码起始点（入口点）紧靠在 private_stream_2 分组（作为 private_stream_2 的属性的 PES_packet()）之后。private_stream_2 分组存储用于解码在两个相邻 private_stream_2 分组之间存储的视频流的信息。因此，对于视频流，作为可解码起始点信息的 RPN_EP_start 描述紧靠实际可解码起始点之前的 private_stream_2 分

组的开始位置。

[0333] 对于由 EP_map() 中的 stream_id 和 private_stream_id 所标识的每个基本流，预先以升序次序来对作为可解码起始点信息的 PTS_EP_start 和 RPN_EP_start 集合进行排序。因此，作为可解码起始点信息的 PTS_EP_start 和 RPN_EP_start 集合可被对半搜索 (binary-searched)。

[0334] 例如在日本专利待审公开 2000-341640（日本专利申请 HEI 11-317738）中描述了用于可变速率的流和其大小在视频存取单元方面不同的流的随机存取方法。

[0335] 【对剪辑流文件的描述】

[0336] 以下将描述存储在图 4 所示的“流”目录中的、具有 PS 扩展名的剪辑流文件的内部结构（图 4 中的“00001.PS”、“00002.PS”和“00003.PS”）。

[0337] 基于在 MPEG-2 系统 (ISO/IEC13818-1) 中定义的 MPEG2_Program_Stream() 构成剪辑流文件。

[0338] 图 15A 和图 15B 示出了在 MPEG-2 系统 (ISO/IEC 13818-1 :20000) 标准中描述的表格 2-31、表格 2-32、和表格 2-33。

[0339] 存储在剪辑流文件中的节目流是在 MPEG2- 系统标准的表格 2-31 中定义的 MPEG2_Program_Stream()。节目流包括至少一个 pack() 和一个 MPEG_program_end_code。日本专利第 2785220 号中描述了 MPEG2_Program_Stream()

[0340] 如 MPEG-2 系统标准的表格 2-32 中定义的那样，一个 pack() 包括一个 Pack_header() 和任意数目的 PES_packet()。在 MPEG2- 系统标准的表格 2-33 中详细地描述了 Pack_header()。

[0341] 在 MPEG2- 系统标准中，pack() 具有可变长度的大小。然而，如图 14 所述，假定 pack() 的大小是固定的 2048 字节。在这个示例中，一个 pack() 中的 PES_packet() 的数目是 1、2、或者 3。当 Pack() 从 private_stream_2 分组开始时，其后总是跟随对应视频流（在相同 Pack() 中的）的 PES_packet()。此外，作为第三 PES_packet()，可能有 padding_packet。 private_stream_2 分组总是出现在 Pack() 的开头处。

[0342] 当 Packet() 不是以 private_stream_2 分组开始时，Pack() 以包括视频数据、音频数据、子标题数据等中的内容数据的 PES_packet() 开始。第二 PES_packet() 可以是 padding_packet。

[0343] 图 16A 和图 16B 到图 18A 和 18B 示出了在 MPEG2- 系统标准的表格 2-17 中定义的 PES_packet()。

[0344] PES_packet() 主要包括 packet_start_code_prefix、stream_id、PES_packet_length（它们在图 16A 和图 16B 中示出）、对应于 stream_id 等而改变的头部部分（包括 stuffing_byte)（这些部分在图 16A 和图 16B 到图 18A 和图 18B 中示出）、PES_packet_data_byte（图 18A 和图 18B 所示）。当 PES_packet() 是 padding_packet(stream_id ＝＝ padding_stream) 时，重复所需数目的 padding_byte(0xFF)（图 18A 和图 18B) 来代替 PES_packet_data_byte。

[0345] 如图 16A、图 16B、图 17A、图 17B 和图 17C 所示，PES_packet() 的头部部分可以描述代表被称为呈现时间戳 (PTS) 的显示定时的信息、和代表被称为解码时间戳的解码定时的信息。根据这个实施例，将 PTS 添加到全部存取单元（构成在 MPEG2- 系统中定义的基本

流的解码单元）中的每一个中。当在 MPEG2- 系统中指定时，添加 DTS。

[0346] 用节目流多路传输的基本流被存储在 PES_packet() 的 PES_packet_data_byte（图 18A 和图 18B）中。PES_packet() 中的 stream_id 描述了与基本流的属性相对应的值，以便标识存储在 PES_packet_data_byte 中的基本流。

[0347] 在 MPEG2- 系统标准的表格 2-18 中定义了在 PES_packet() 的 stream_id 中描述的值和基本流的属性（类型）之间的关系。图 19A 和图 19B 示出了 MPEG-2 系统标准中的表格 2-18。

[0348] 根据本发明的实施例，图 20 所示的值用作在图 19A 和图 19B 所示的 MPEG2- 系统标准中定义的 stream_id。

[0349] 换句话说，根据这个实施例，使用了五个模式 10111101B、10111110B、10111111B、110xxxxxB、和 1110xxxxB 作为 stream_id 的值，其中“x”代表 0 和 1 中的任何一个。

[0350] 对应于图 20 所示的表格，具有属性 private_stream_1 的基本流中的 PES_packet() 的 stream_id 是 10111101B。对应于图 20 所示的表格，padding_packet 中的 PES_packet() 的 stream_id 是 10111110B。根据图 20 所示的表格，具有属性 private_stream_2 的基本流中的 PES_packet() 的 stream_id 是 10111111B。

[0351] 在 MPEG 中定义的音频流（音频基本流）的 PES_packet() 的 stream_id 是 110xxxxxB。110xxxxxB 中的低五位 xxxxxx 是标识音频流的音频流字节。可以用节目流对可由该音频流号码标识的 32（$=2^5$）个音频流 (MPEG 中定义的音频流）进行多路复用。

[0352] 在 MPEG 中定义的视频流（视频基本流）的 PES_packet() 的 stream_id 是 1110xxxxB。110xxxxB 中的低 4 位 xxxx 是标识视频流的视频流号码。可以用节目流对 16（$=2^4$）个视频流 (MPEG 中定义的视频流）进行多路复用。

[0353] stream_id 为 110xxxxB 的 PES_packet() 用于存储在 MPEG 中定义的视频流。stream_id 为 110xxxxxB 的 PES_packet() 用于存储在 MPEG 中定义的音频流。另一方面，没有在 MPEG 中定义用于与编码系统（例如，ATRAC 系统）相对应的基本流的 PES_packet() 中的 stream_id。因此，类似于在 MPEG 中定义的视频流和音频流，不能用 stream_id 在 PES_packet() 中存储与未在 MPEG 中定义的编码系统相对应的基本流。

[0354] 因此，根据这个实施例，扩展了 private_stream_1 的 PES_packet() 中的 PES_packet_data_byte，以便存储与未在 MPEG 中定义的编码系统相对应的基本流。

[0355] private_stream_1 的 PES_packet() 中的扩展 PES_packet_data_byte 被描述为 private_stream1_PES_payload()。

[0356] 【private_stream1_PES_payload() 的描述】

[0357] 图 21 示出了 private_stream1_PES_payload() 的语法。

[0358] private_stream1_PES_payload() 包括 private_header() 和 private_payload()。private_payload() 存储诸如 ATRAC 音频流、LPCM 音频流、子标题流等之类、与未在 MPEG 系统中定义的编码系统相对应地编码的基本流。

[0359] private_header() 从 private_stream_id(8 位）开始。private_stream_id 是标识存储在 private_payload() 中的基本流的标识信息。private_stream_id 具有与基本流的属性（类型）相对应的下列值。

[0360] 图 22 示出了 private_stream_id 值和存储在 private_payload() 中的基本流的

属性之间的关系。

[0361] 图 22 示出了作为 private_stream_id 值的三个模式 0000xxxxB、0001xxxxB、和 100xxxxxB，其中如果图 20 所示的情况那样，“x”是 0 和 1 的任何值。

[0362] 根据图 22 所示的表格，ATRAC 音频流的 private_payload() 的 private_stream1_PES_payload() 中的 private_stream_id 为 0000xxxxB。0000xxxxB 中的低 4 位是标识 ATRAC 音频流的音频流号码。可以用节目流 (MPEG2_Program_Stream()) 来多路复用可由该音频流号码标识的 16(= 2^4) 个 ATRAC 音频流。

[0363] 根据图 22 所示的表格，LPCM 音频流的 private_payload() 的 private_stream1_PES_payload() 中的 private_stream_id 为 0001xxxxB。0001xxxxB 中的低 4 位是标识 LPCM 音频流的音频流号码。可以用节目流多路复用可由该音频流号码标识的 16(= 2^4) 个 LPCM 音频流。

[0364] 根据图 22 所示的表格，子标题流的 private_payload() 的 private_stream1_PES_payload() 中的 private_stream_id 为 100xxxxxB。100xxxxxB 中的低五位是标识子标题流的子标题流号码。可以用节目流多路复用 32(= 2^5) 个子标题流。

[0365] 图 21 示出了图 20 和图 22 的关系。

[0366] 在图 21 中，在 private_stream1_PES_payload() 的 private_stream_id 之后的元素取决于存储在 private_payload() 中的基本流的属性而改变。存储在 private_payload() 中的基本流的属性由在 private_header() 开头处的 private_stream_id 所确定。

[0367] 当存储在 private_payload() 中的基本流是 ATRAC 音频流 (private_stream_id == ATRAC) 时，描述了用于将来扩展的 reserved_for_future_use(8 位)。reserved_for_future_use 之后跟随 AU_locator(16 位)。AU_locator 代表基于紧靠 AU_locator 之后的位置的、存储在 private_payload() 中的 ATRAC 音频流的音频存取单元 (ATRAC 音频存取单元)(音频帧) 的开始位置。当 private_payload() 没有存储音频存取单元时，例如在 AU_locator 中描述了 0xFFFF。

[0368] 当在 private_payload() 中存储的基本流是 LPCM 音频流 (private_stream_id == LPCM) 时，则连续地描述 fs_flag(1 位)、reserved_for_future_use(3 位)、ch_flag(4 位)、和 AU_locator(16 位)。

[0369] fs_flag 代表存储在 private_payload() 中的 LPCM 音频流的采样频率。当 LPCM 音频流的采样频率是 48kHz(千赫) 时，fs_flag 为 0。当 LPCM 音频流的采样频率是 44.1kHz 时，fs_flag 为 1。

[0370] ch_flag 代表存储在 private_payload() 中的 LPCM 音频流的声道数目。当 LPCM 音频流是单声道时，ch_flag 为 1。当 LPCM 音频流是立体声时，ch_flag 为 2。

[0371] AU_locator 代表基于紧靠 AU_locator 之后的位置的、存储在 private_payload() 中的 LPCM 音频流的音频存取单元 (LPCM 音频存取单元)(音频帧) 的开始位置。当 private_payload() 没有存储音频存取单元时，例如在 AU_locator 中描述了 0xFFFF。

[0372] 当存储在 private_payload() 中的基本流是子标题流 (private_stream_id == SUBTITLE) 时，描述了用于将来扩展的 reserved_for_future_use(8 位)。reserved_for_future_use 之后紧跟 AU_locator(16 位)。AU_locator 表示基于紧跟 AU_locator 之

后的位置的、存储在 private_payload() 中的子标题流的子标题存取单元的开始位置。当 private_payload() 没有存储子标题存取单元时，例如在 AU_locator 中描述了 0xFFFF。

[0373] 【private_stream2_PES_payload() 的描述】

[0374] 图 23 示出了 private_stream2_PES_payload() 的语法。

[0375] private_stream2_PES_payload() 是 private_stream_2 的 private_payload() 中的 PES_packet_data_byte（图 18A 和图 18B）的扩展，即 private_stream_2 的 PES_packet() 中的 PES_packet_data_byte 的扩展。private_stream2_PES_payload() 描述了用于解码视频流的信息。

[0376] 根据这个实施例，private_stream_2 的 PES_packet() 紧跟在视频流的可解码起始点之后。因此，根据这个实施例，当从节目流中检测到 private_stream_2PES_packet() 时，可以解码紧跟在 PES_packet() 之后的视频流。

[0377] 图 14 所示的 EP_map() 中的 RPN_EP_start 表示用于视频流的 private_stream_2 的 PES_packet() 的开始位置。

[0378] private_stream2_PES_payload() 从用于将来扩展的 reserved_for_future_use(8 位）开始。reserved_for_future_use 之后连续地跟随 video_stream_id(8 位）、1stRef_picture(16 位）、2ndRef_picture(16 位）、3rdRef_picture(16 位）、4thRef_picture(16 位）、au_information()、和 VBI()。

[0379] video_stream_id 描述了紧跟在 private_stream_2 的 PES_packet() 之后的视频流的 PES_packet() 中的 stream_id（与 stream_id 同值）。video_stream_id 标识了用存储在 private_stream_2 的 PES_packet() 中的信息 (PES_packet() 中的 private_stream2_PES_payload()) 解码的视频流（存储视频流的 PES_packet()）。

[0380] 1stRef_picture、2ndRef_picture、3rdRef_picture、和 4thRef_picture 分别表示包括第一、第二、第三、和第四参考图像的上一个 pack() 从由 video_stream_id 所标识的视频流中的 private_stream_2 中的 PES_packet() 到下一个 private_stream_2 的 PES_packet() 的位置相对值。在日本专利待审公开 HEI 09-46712（日本专利申请 HEI 07-211420）中将 1stRef_picture、2ndRef_picture、3rdRef_picture、和 4thRef_picture 的详细信息公开为 bytes_to_first_P_pic 和 bytes_to_second_P_pic。

[0381] au_information() 逐个 private_stream_2 中的 PES_packet() 地描述了有关视频流中的视频存取单元的信息。将参考图 24 详细描述 au_information()。

[0382] VBI() 用于描述有关封闭字幕的信息。

[0383] 为每个视频流的可解码起始点描述了具有 private_stream2_PES_payload() 的 private_stream_2 中的 PES_packet()。

[0384] 图 24 示出了图 23 所示的 au_information() 的语法。

[0385] au_information() 从 length(16 位）开始。length 表示包括 length 的大小在内的、au_information() 的大小。length 之后连续地跟随 reserved_for_word_alignment(8 位）和 number_of_access_unit(8 位）。reserved_for_word_alignment 用于字对准。

[0386] number_of_access_unit 代表从 private_stream2 的 PES_packet() 到下一个 private_stream2 的 PES_packet() 所存储的存取单元（画面）数目。

[0387] 换句话说，number_of_access_unit 代表在 private_stream2_PES_payload() 中

的 video_stream_id 相同的 private_stream_2 的 PES_packet() 中、从 au_information() 到下一个 au_information（或者当 au_information() 是其最后一个 au_information() 时，到剪辑流文件的结尾）、由 video_stream_id 所代表的视频流中包括的存取单元（画面）的数目。

[0388] number_of_access_unit 之后跟随与 number_of_access_unit 相对应的 for 循环的内容。换句话说，描述了有关从包括 number_of_access_unit 的 private_stream_2 的 PES_packet() 到下一个 private_stream_2 的 PES_packet() 中的至少一个视频存取单元的信息。

[0389] 在该 for 循环中描述的信息（有关视频存取单元的信息）如下。

[0390] 该 for 循环包括 pic_struct_copy(4 位)、au_ref_flag(位)、AU_length(21 位)、和 reserved(保留)。

[0391] pic_struct_copy 为与 MPEG4-AVC(ISO/IEC 14496-10) 相对应的视频存取单元描述了在 ISO/IEC 14496-10,D.2.2. 中定义的 pic_struct() 的拷贝。pic_struct() 是代表例如显示画面帧、或者在显示画面的上半帧之后显示下半帧的信息。

[0392] au_ref_flag 表示对应的存取单元是否为当解码另一个存取单元（的画面）时参考的参考画面。当对应的存取单元是参考画面时，au_ref_flag 为 1。当对应的存取单元不是参考画面时，au_ref_flag 为 0。

[0393] AU_length 以字节为单位表示对应存取单元的大小。

[0394] 【记录在盘 101 上的数据的特定示例】

[0395] 图 25 到图 28 示出了具有上述格式并且已经记录在图 1 所示的盘 101 上的数据的特定示例。

[0396] 在图 25 到图 28 中，使用了对应于 MPEG2- 视频的视频流和对应于 ATRAC 的音频流。然而，在本发明中使用的视频流和音频流不局限于这些流。换句话说，可以使用与 MPEG4-Visual 相对应的视频流、与 MPEG4-AVC 相对应的视频流等。另一方面，可以使用与 MPEG1/2/4 音频相对应的音频流、与 LPCM 音频相对应的音频流等。

[0397] 不同于视频流和音频流，子标题流可以不被连续地解码并且以相同间隔显示（输出）。换句话说，子标题流不时地从图 2A 和图 2B 所示的缓冲器控制模块 215 提供给子标题解码器控制模块 218。子标题解码器控制模块 218 解码该子标题流。

[0398] 图 25 到图 28 示出了在其中三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”存储在目录“剪辑”中、而且与三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”相对应的三个剪辑流文件“00001.PS”、“00002.PS”、和“00003.PS”存储在目录“流”中的情况下，文件“PLAYLIST.DAT”、三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”等的特定示例。然而，在图 25 到图 28 中，省略了诸如文件“PLAYLIST.DAT”等之类的部分数据。

[0399] 图 25 示出了图 5 所示的文件“PLAYLIST.DAT”的特定示例。

[0400] 在图 25 中，number_of_PlayLists 为 2。因此，存储在文件“PLAYLIST.DAT”中的 PlayList() 的数目为 2。在图 25 中，两个 PlayList() 是 PlayList#0 和 PlayList#1。

[0401] 第一 PlayList()，即 PlayList#0 中的 capture_enable_flag_PlayList 为 1。因此，允许二次使用根据 PlayList#0 再现的视频数据。另一方面，PlayList#0 中的 number_

of_PlayItems 为 2。因此，包括在 PlayList#0 中的 PlayItem() 的数目为 2。在图 25 所示的“PlayList#0”字段之下描述了作为两个 PlayItem() 的 PlayItem#0 和 PlayItem#1 的特定示例。

[0402] 在作为包括在 PlayList#0 中的第一 PlayItem() 的 PlayItem#0 中，图 6 中描述的 Clip_Information_file_name 是“00001.CLP”，IN_time 为 180,090，OUT_time 为 27,180,090。因此，由 PlayList#0 中的 PlayItem#0 再现的剪辑在与剪辑信息文件“00001.CLP”相对应的剪辑流文件“00001.PS”中的时间 180,090 到时间 27,180,090。

[0403] 在作为在 PlayList#0 中包括的第二 PlayItem() 的 PlayItem#1 中，图 6 中描述的 Clip_Information_file_name 是“00002.CLP”，IN_time 为 90,000，OUT_time 为 27,090,000。因此，由 PlayList#0 中的 PlayItem#1 再现的剪辑在与剪辑信息文件“00002.CLP”相对应的剪辑流文件“00002.PS”中的时间 90,000 到时间 27,090,000。

[0404] 在图 25 中，在作为第二 PlayList() 的 PlayList#1 中，capture_enable_flag_playlist 为 0。因此，不允许二次使用（禁止二次使用）根据 PlayList#1 再现的视频数据。在 PlayList#1 中，number_of_PlayItems 为 1。因此，包括在 PlayList#1 中的 PlayItem() 的数目为 1。在图 25 中，在字段“PlayList#1”之下描述了作为一个 PlayItem() 的 PlayItem#0 的特定示例。

[0405] 在作为在 PlayList#1 中包括的一个 PlayItem() 的 PlayItem#0 中，图 6 中描述的 Clip_Information_file_name 是“00001.CLP”，IN_time 为 90,000，OUT_time 为 81,090,000。因此，由 PlayList#1 中的 PlayItem#0 再现的剪辑在与剪辑信息文件“00003.CLP”相对应的剪辑流文件“00003.PS”中的时间 90,000 到时间 81,090,000。

[0406] 图 26A 和图 26B 示出了在图 10 中描述的剪辑信息文件 Clip() 的特定示例。因此，图 26A 和图 26B 示出了剪辑信息文件“00001.CLP”、“00002.CLP”和“00003.CLP”的特定示例。

[0407] 在剪辑信息文件“00001.CLP”中，presentation_start_time 是 90,000 而 presentation_end_time 是 27,990,00 因此，存储在与剪辑信息文件“00001.CLP”相对应的剪辑流文件“00001.PS”中的节目流可以使用内容达 310 秒 (27,990,000-90,000/90kHz)。

[0408] 在剪辑信息文件“00001.CLP”中，capture_enable_flag_Clip 为 1。因此，允许二次使用利用在与剪辑信息文件“00001.CLP”相对应的剪辑流文件“00001.PS”中存储的节目流所多路复用的视频流（与此对应的视频数据）。

[0409] 在图 26A 和图 26B 中，在剪辑信息文件“00001.CLP”中，number_of_streams 为 4。因此，用存储在剪辑流文件“00001.PS”中的节目流多路复用四个基本流。

[0410] 假定四个基本流由 stream#0、stream#1、stream#2、和 stream#3 表示，在图 26A 和图 26B 所示的字段“00001.CLP”下描述了 stream#0、stream#1、stream#2、和 stream#3 这四个基本流中的 StreamInfo()（图 10）的特定示例。

[0411] 在剪辑流文件“00001.PS”的第一基本流 stream#0 中，stream_id 为 0xE0。因此，如图 20 和图 22（或者图 11）所述，基本流 stream#0 是视频流。根据这个实施例，private_stream_id 没有与视频流相关。在图 26A 和图 26B 中，private_stream_id 为 0x00。

[0412] 在作为剪辑流文件“00001.PS”中的第一基本流的视频流 stream#0 中，包括在 StreamInfo() 中的 StaticInfo()（图 12）的 picture_size 为“720×480”，frame_rate 为

“29.97Hz”，而 cc_flag 为“是 (Yes)”。因此，视频流 stream#0 是具有 720×480 像素以及 29.97Hz 的帧周期的视频数据。此外，视频流 stream#0 包括封闭的字幕数据。

[0413] 在作为剪辑流文件“00001.PS”中的第一基本流的视频流 stream#0 中，StreamInfo() 中的 number_of_DynamicInfo（图 10）为 0。不存在 pts_change_point 和 DynamicInfo() 对。

[0414] 在剪辑流文件“00001.PS”的第二基本流 stream#1 中，stream_id 为 0xBD，private_stream_id 为 0x00。因此，如图 20 和图 22 所述，基本流 stream#1 为 ATRAC 音频流。

[0415] 在作为剪辑流文件“00001.PS”中的第二基本流的 ATRAC 音频流 stream#1 中，包括在 StreamInfo() 中的 StaticInfo()（图 12）的 audio_language_code 为“Japanese（日语）”，channel_configuration 为“STEREO（立体声）”，Ife_existence 为“NO（否）”，sampling_frequency 为“48kHz”。因此，ATRAC 音频流 stream#1 为日语立体声音频数据。此外，ATRAC 音频流 stream#1 不包括低频效应声道，而且采样频率为 48kHz。

[0416] 此外，在作为剪辑流文件“00001.PS”中的第二基本流的 ATRAC 音频流 stream#1 中，StreamInfo() 中的 number_of_DynamicInfo（图 10）为 0。不存在 pts_change_point 和 DynamicInfo() 对。

[0417] 在剪辑流文件“00001.PS”的第三基本流 stream#2 中，stream_id 为 0xBD，private_stream_id 为 0x80。因此，如图 20 和图 22 所述，基本流 stream#2 为子标题流。

[0418] 在作为剪辑流文件“00001.PS”中的第三基本流的子标题流 stream#2 中，包括在 StreamInfo() 中的 StaticInfo() 的 subtitle_language_code 为“Japanese（日语）”，configurable_flag 为 0。因此，子标题流 stream#2 是日语子标题数据。此外，不允许改变（禁止改变）这个显示模式。

[0419] 在作为剪辑流文件“00001.PS”中的第三基本流的子标题流 stream#2 中，StreamInfo() 中的 number_of_DynamicInfo（图 10）为 0。不存在 pts_change_point 和 DynamicInfo() 对。

[0420] 在剪辑流文件“00001.PS”的第四基本流 stream#3 中，stream_id 为 0xBD，private_stream_id 为 0x81。因此，如图 20 和 22 所述，基本流 stream#3 为子标题流。

[0421] 为了区分作为剪辑流文件“00001.PS”中的第三基本流的子标题流 stream#2 和作为第四基本流的子标题流 stream#3，它们的 private_stream_id 分别为 0x80 和 0x81。

[0422] 在作为剪辑流文件“00001.PS”中的第四基本流的子标题流 stream#3 中，包括在 StreamInfo() 中的 StaticInfo()（图 12）的 subtitle_language_code 为“Japanese（日语）”，configurable_flag 为 1。因此，子标题流 stream#3 是日语子标题数据。允许改变子标题流 stream#3 的显示模式。

[0423] 在作为剪辑流文件“00001.PS”中的第四基本流的子标题流 stream#3 中，StreamInfo() 中的 number_of_DynamicInfo（图 10）为 0。不存在 pts_change_point 和 DynamicInfo() 对。

[0424] 在图 26A 和图 26B 中，在剪辑信息文件“00002.CLP”中，presentation_start_time 为 90,000，presentation_end_time 为 27,090,000。因此，存储在与剪辑信息文件“00002.CLP”相对应的剪辑流文件“00002.PS”中的节目流可以使用内容达 300 秒

(27,090,000-90,000/90kHz)。

[0425] 在剪辑信息文件“00002.CLP”中，capture_enable_flag_Clip 为 0。因此，不允许二次使用（禁止二次使用）利用与剪辑信息文件“00002.CLP”相对应的剪辑流文件“00002.PS”中存储的节目流多路复用的视频流（与此对应的视频数据）。

[0426] 在图 26A 和图 26B 中，在剪辑信息文件“00002.CLP”中，number_of_streams 为 4。因此，如同上述剪辑流文件“00001.PS”一样，用存储在对应剪辑流文件“00002.PS”中的节目流多路复用了四个基本流。

[0427] 假定四个基本流用 stream#0、stream#1、stream#2、和 stream#3 表示，在图 26A 和图 26B 所示的字段“00002.CLP”下描述了 stream#0、stream#1、stream#2、和 stream#3 这四个基本流中的 StreamInfo()（图 10）的特定示例。

[0428] 在图 26A 和图 26B 中，剪辑流文件“00002.PS”中的第一到第四基本流的 StreamInfo() 的内容与剪辑流文件“00001.PS”中的第一到第四基本流的那些相同。因此，将省略对它们的描述。

[0429] 如上所述，剪辑流文件“00002.PS”中的第一到第四基本流 (stream#0 到 stream#3) 的 StreamInfo() 的内容与剪辑流文件“00001.PS”中的第一到第四基本流 (stream#0 到 stream#3) 的那些相同。因此，剪辑流文件“00002.PS”中的第一基本流 stream#0 是视频流。第二基本流 stream#1 是 ATRAC 音频流。第三和第四基本流 stream#2 和 stream#3 是子标题流。

[0430] 在图 26A 和图 26B 中，在剪辑信息文件“00003.CLP”中，presentation_start_time 为 90,000，presentation_end_time 为 81,090,000。因此，在与剪辑信息文件“00003.CLP”相对应的剪辑流文件“00003.PS”中存储的节目流可以使用内容达 900 秒 ((81,090,000-90,000)/90kHz)。

[0431] 在剪辑信息文件“00003.CLP”中，capture_enable_flag_Clip 为 1。因此，允许二次使用利用在与剪辑信息文件“00003.CLP”相对应的剪辑流文件“00003.PS”中存储的节目流多路复用的视频流。

[0432] 在图 26A 和图 26B 中，在剪辑信息文件“00003.CLP”中，number_of_streams 为 3。因此，用存储在剪辑流文件“00003.PS”中的节目流多路复用了三个基本流。

[0433] 假定三个基本流用 stream#0、stream#1、和 stream#2 表示，在图 26A 和图 26B 所示的字段“00003.CLP”下描述了 stream#0、stream#1、和 stream#2 这三个流中的 StreamInfo()（图 10）的特定示例。

[0434] 在剪辑流文件“00003.PS”的第一基本流 stream#0 中，stream_id 为 0xE0。因此，如图 20 和图 22（或者图 11）所述，基本流 stream#0 是视频流。类似于剪辑流文件“00001.PS”的第一基本流 stream#0，private_stream_id 为 0x00。

[0435] 在作为剪辑流文件“00003.PS”中的第一基本流的视频流 stream#0 中，包括在 StreamInfo() 中的 StaticInfo()（图 12）的 picture_size 为“720×480”，frame_rate 为“29.97Hz”，而 cc_flag 为“No（否）”。因此，视频流 stream#0 是具有 720×480 像素以及 29.97Hz 的帧周期的视频数据。视频流 stream#0 不包括封闭的字幕数据。

[0436] 在作为剪辑流文件“00003.PS”中的第一基本流的视频流 stream#0 中，StreamInfo() 中的 number_of_DynamicInfo（图 10）为 2。因此，在 StreamInfo() 中描述

了两组 pts_change_point 和 DynamicInfo()。

[0437] 在剪辑流文件“00003.PS”的第二基本流 stream#1 中，stream_id 为 0xE1。因此，如图 20 和图 21(或者图 11) 所述，基本流 stream#1 是视频流。为了区分作为剪辑流文件“00003.PS”中的第一基本流的视频流 stream#0 和作为第二基本流的视频流 stream#1，它们的 stream_id 分别为 0xE0 和 0xE1。类似于剪辑流文件“00001.PS”中的第一基本流 stream#0，private_stream_id 为 0x00。

[0438] 在作为剪辑流文件“00003.PS”中的第二基本流的视频流 stream#1 中，包括在 StreamInfo() 中的 StaticInfo()(图 12) 的 picture_size、frame_rate、和 cc_flag 和作为第一基本流的视频流 stream#0 中的那些相同。因此，作为剪辑流文件“00003.PS”中的第二基本流的视频流 stream#1 是具有 720×480 像素和 29.97Hz 帧周期的视频数据。视频流 stream#1 不包括封闭的字幕数据。

[0439] 在作为剪辑流文件“00003.PS”中的第二基本流的视频流 stream#1 中，StreamInfo() 中的 number_of_DynamicInfo(图 10) 为 0。因此，不存在 pts_change_point 和 DynamicInfo() 对。

[0440] 在剪辑流文件“00003.PS”的第三基本流 stream#2 中，stream_id 为 0xBD，private_stream_id 为 0x00。因此，如图 20 和图 22 中所述，基本流 stream#2 为 ATRAC 音频流。

[0441] 在作为剪辑流文件“00003.PS”中的第三基本流的 ATRAC 音频流 stream#2 中，包括在 StreamInfo() 中的 StaticInfo()(图 12) 的 audio_language_code、channel_configuration、lfe_existence 和 sampling_frequency 与作为剪辑流文件“00001.PS”中的第二基本流的 ATRAC 音频流 stream#1 中的那些相同。因此，作为剪辑流文件“00003.PS”中的第三基本流的 ATRAC 音频流 stream#2 是日语和立体声音频数据。此外，ATRAC 音频流 stream#2 不包括低频效应声道。ATRAC 音频流 stream#2 具有 48kHz 的采样频率。

[0442] 在作为剪辑流文件“00003.PS”中的第三基本流的 ATRAC 音频流 stream#2 中，StreamInfo() 中的 number_of_DynamicInfo(图 10) 为 3。因此，StreamInfo() 描述了三组 pts_change_point 和 DynamicInfo()。

[0443] 图 27 示出了在图 10 中描述的剪辑信息文件 Clip() 中的 EP_map() 的特定示例。图 27 示出了图 4 所示的剪辑信息文件“00001.CLP”中的如图 14 所示的 EP_map() 的特定示例。

[0444] 在图 27 中，在 EP_map() 中，number_of_stream_id_entries 为 1。因此，EP_map() 描述了一个基本流中的可解码起始点的信息。

[0445] 在图 27 所示的 EP_map() 中，stream_id 为 0xE0。因此，如图 20 和图 22 所述，EP_map() 描述了由 0xE0 的 stream_id 所标识的视频流中的可解码起始点的信息 (PTS_EP_start 和 RPN_EP_start(图 14))。在图 27 中，EP_map() 是剪辑信息文件“00001.CLP”。如图 26A 和图 26B 所述，在与剪辑信息文件“00001.CLP”相对应的剪辑流文件“00001.PS”中，stream_id 为 0xE0 的基本流是剪辑流文件“00001.PS”中的第一视频流。因此，在图 27 所示的 EP_map() 中描述的信息是视频流 stream#0 中的可解码起始点 PTS_EP_start 和 RPN_EP_start。

[0446] 在图 27 中，描述了剪辑流文件“00001.PS”中的第一视频流 stream#0 的可解码起

始点的前五个 PTS_EP_start 和 RPN_EP_start，但是省略了第六个和以后的 PTS_EP_start 和 RPN_EP_start。

[0447] 在图 27 所示的 EP_map() 中，private_stream_id 为 0x00。当 stream_id 代表视频流时，忽略 private_stream_id。

[0448] 图 28 示出了在图 25 中描述的 PlayList#0 和 PlayList#1（图 5 所示的 PlayList()）中的 PlayListMark() 的特定示例。

[0449] 图 28 所示的上部表格代表 PlayList#0 中的 PlayListMark()（图 7）。

[0450] 在图 28 所示的上部表格中，PlayList#0 中的 PlayListMark() 的 number_of_PlayList_marks 为 7。因此，包括在 PlayList#0 中的 Mark()（其中的 PlayListMark()）的数目为 7。

[0451] 在图 28 所示的上部表格中，作为包括在 PlayList#0 中的七个 Mark() 中的第一 Mark() 的 Mark#0 的 mark_type（图 7）为“Chapter（章节）”。因此，Mark#0 是章节标记。此外，因为 ref_to_PlayItem_id（图 7）为 0，所以 Mark#0 属于图 25 所示的两个 PlayItem#0 和 #1 中的 PlayItem#0。此外，Mark#0 的 mark_time_stamp 为 180,090。因此，Mark#0 是由包括在 PlayList#0 中的 PlayItem#0 再现的剪辑流文件的时间（再现时间）180,090 的标记。Mark#0 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 二者都是 0。因此，Mark#0 不与任何基本流相关。此外，Mark#0 中的 mark_data 为 1。因此，Mark#0 代表编号为 1 的章节。

[0452] 由包括在 PlayList#0 中的 PlayItem#0 再现的剪辑流文件是由在 PlayItem#0 的 Clip_Information_file_name（图 25）中描述的“00001.CLP”所标识的剪辑流文件“00001.PS”。因此，由 Mark#0 中的 mark_time_stamp 所表示的时间 180,090 是剪辑流文件“00001.PS”中的时间。

[0453] 在图 28 所示的上部表格中，作为包括在 PlayList#0 中的七个 Mark() 中的第五 Mark() 的 Mark#4 是与第一 Mark#0 相同的章节标记。

[0454] 换句话说，作为第五 Mark() 的 Mark#4 的 mark_type（图 7）是“Chapter（章节）”。因此，Mark#4 是章节标记。此外，Mark#4 中的 ref_to_PlayItem_id（图 7）为 1。因此，Mark#4 属于包括在 PlayList#0 中的两个 PlayItem#0 和 #1（如图 25 所示）中的 PlayItem#1。Mark#4 中的 mark_time_stamp 是 90,000。因此，Mark#4 是由包括在 PlayList#0 中的 PlayItem#1 再现的剪辑流文件的时间 90,000 的标记。此外，Mark#4 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 二者都是 0。因此，Mark#4 没有与任何基本流相关。此外，Mark#4 中的 mark_data 为 2。因此，Mark#4 代表编号为 2 的章节。

[0455] 在这个示例中，由包括在 PlayList#0 中的 PlayItem#1 再现的剪辑流文件是由在 PlayItem#1 的 Clip_Information_file_name 中描述的“00002.CLP”所标识的剪辑流文件“00002.PS”（图 25）。因此，由 Mark#4 中的 mark_time_stamp 所表示的时间 90,000 是剪辑流文件“00002.PS”中的时间。

[0456] 在图 28 所示的上部表格中，作为包括在 PlayList#0 中的七个 Mark() 中的第二 Mark() 的 Mark#1 的 mark_type（图 7）为“Index（索引）”。因此，Mark#1 是索引标记。此外，Mark#1 中的 ref_to_PlayItem_id（图 7）为 0。因此，Mark#1 属于包括在 PlayList#0 中的两个 PlayItem#0 和 #1（如图 25 所示）中的 PlayItem#0。此外，Mark#1 中的 mark_time_

stamp 为 5,580,090。因此，Mark#1 是由包括在 PlayList#0 中的 PlayItem#0 再现的剪辑流文件中的时间 5,580,090 的标记。此外，Mark#1 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 二者都是 0。因此，Mark#1 没有与任何基本流相关。此外，Mark#1 中的 mark_data 为 1。因此，Mark#1 代表编号为 1 的索引。

[0457] 在这个示例中，由包括在 PlayList#0 中的 PlayItem#0 再现的脚本流文件是如上所述的剪辑流文件“00001.PS”。因此，由 Mark#1 中的 mark_time_stamp 所表示的时间 5,580,090 是剪辑流文件“00001.PS”中的时间。

[0458] 在图 28 所示的上部表格中，作为包括在 PlayList#0 中的七个 Mark() 中的第三、第六、和第七 Mark() 的 Mark#2、Mark#5、和 Mark#6 是类似于第二 Mark#1 的索引标记。

[0459] 在图 28 所示的上部表格中，作为包括在 PlayList#0 中的七个 Mark() 中的第四个 Mark() 的 Mark#3 的 mark_type（图 7）为“Event（事件）”。因此，Mark#3 是事件标记。此外，Mark#3 中的 ref_to_PlayItem_id（图 7）为 0。因此，Mark#3 属于包括在 PlayList#0 中的两个 PlayItem#0 和 #1（如图 25 所示）中的 PlayItem#0。此外，Mark#3 中的 mark_time_stamp 为 16,380,090。因此，Mark#3 是由包括在 PlayList#0 中的 PlayItem#0 再现的剪辑流文件中的时间 16,380,090 的标记。Mark#3 中的 entry_ES_strearn_id 和 entry_ES_private_stream_id 为 0。因此，#3 没有与任何基本流相关。此外，Mark#3 中的 mark_data 为 0。因此，Mark#3 使得具有参数 0 的事件发生。

[0460] 如上所述，由包括在 PlayList#0 中的 PlayItem#0 再现的剪辑流文件是剪辑流文件“00001.PS”。由 Mark#3 中的 mark_time_stamp 所表示的时间 16,380,090 是剪辑流文件“00001.PS”中的时间。

[0461] 在图 28 所示的上部表格中，在 PlayList#0 中的 PlayListMark() 表格的右侧的左栏中描述了 Mark() 所属的 PlayItem() 的累积时间。在表格右侧的右栏中描述了 playlist#0 的累积时间。

[0462] 图 28 所示的下部表格表示 PlayList#1 中的 PlayListMark()（图 7）。

[0463] 在图 28 所示的下部表格中，PlayList#1 中的 PlayListMark() 的 number_of_PlayList_marks 为 3。因此，包括在 PlayList#1 中的 Mark()（其中的 PlayListMark()）的数目为 3。

[0464] 在图 28 所示的下部表格中，作为包括在 PlayList#1 中的三个 Mark() 中的第一个 Mark() 的 Mark#0 的 mark_type（图 7）为“Chapter（章节）”。因此，Mark#0 是章节标记。此外，Mark#0 中的 ref_to_PlayItem_id（图 7）为 0。因此，Mark#0 属于包括在 PlayList#1 中的一个 PlayItem#0（如图 25 所示）。Mark#0 中的 mark_time_stamp 是 90,000。因此，Mark#0 是由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件中的时间 90,0000 的标记。Mark#0 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 二者都是 0。因此，Mark#0 没有与任何基本流相关。此外，Mark#0 中的 mark_data 为 0。因此，Mark#0 代表其编号为 0 的章节。

[0465] 由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件是由在 PlayItem#0 中的 Clip_Information_file_name（图 25）描述的“00003.CLP”所标识的剪辑流文件“00003.PS”。因此，由 Mark#0 中的 mark_time_stamp 所表示的时间 90,000 是剪辑流文件“00003.PS”中的时间。

[0466] 在图 28 所示的下部表格中，作为包括在 PlayList#1 中的三个 Mark() 中的第二个 Mark() 的 Mark#1 的 mark_type（图 7）为“Event（事件）”。因此，Mark#1 是事件标记。此外，Mark#1 中的 ref_to_PlayItem_id（图 7）为 0。因此，Mark#1 属于包括在 PlayList#1 中的 PlayItem#0（如图 25 所示）。此外，Mark#1 中的 mark_time_stamp 为 27,090,000。因此，Mark#1 是由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件中的时间 27,090,000 的标记。此外，在 Mark#1 中，entry_ES_stream_id 为 0xE0 而 entry_ES_private_stream_id 为 0。因此，Mark#1 与其 stream_id 为 0xE0 的基本流相关，即 Mark#1 与图 20 和图 22 所述的视频流相关。此外，Mark#1 中的 mark_data 为 1。Mark#1 使得具有属性 1 的事件发生。

[0467] 如上所述，由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件为“00003.PS”。因此，由 Mark#1 中的 mark_time_stamp 所表示的时间 27,090,000 是剪辑流文件“00003.PS”中的时间。

[0468] 与 Mark#1 相关的其 stream_id 为 0xE0 的视频流是在用由图 26A 和图 26B 所示的剪辑信息文件“00003.CLP”所标识的剪辑流文件“00003.PS”多路复用的三个基本流 stream#0 到 #2 当中的、第一个基本流（视频流）stream#0。

[0469] 在图 28 所示的下部表格中，作为包括在 PlayList#1 中的三个 Mark() 中的第三个 Mark() 的 Mark#2 的 mark_type 为“Event（事件）”。因此，Mark#2 是事件标记。此外，Mark#2 中的 ref_to_PlayItem_id（图 7）为 0。因此，Mark#2 属于包括在 PlayList#1 中的 PlayItem 之一的 PlayItem#0（如图 25 所示）。此外，Mark#2 中的 mark_time_stamp 为 27,540,000。因此，Mark#2 是由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件中的时间 27,540,000 的标记。此外，在 Mark#2 中，entry_ES_stream_id 为 0xE1 而 entry_ES_private_stream_id 为 0。因此，Mark#2 是其 stream_id 为 0xE1 的基本流，即与图 20 和图 22 所述的视频流相关。此外，Mark#2 中的 mark_data 为 2。因此，Mark#2 使得具有参数 2 的事件发生。

[0470] 在这个示例中，如上所述，由包括在 PlayList#1 中的 PlayItem#0 再现的剪辑流文件是剪辑流文件“00003.PS”。因此，由 Mark#2 所代表的时间 27,540,000 是剪辑流文件“00003.PS”中的时间。

[0471] 与 Mark#2 相关的其 stream_id 为 0xE1 的视频流是在图 25 所示的 PlayList#1 中包括的 PlayItem#0 中的 Clip_Information_file_name 中描述的“00003.CLP”中描述的视频流，即用根据图 26A 和 26B 所示的剪辑信息文件“00003.CLP”所识别的剪辑流文件“00003.PS”多路复用的三个基本流 stream#0 到 #2 中的第二个基本流（视频流）stream#1。

[0472] 在图 28 所示的下部表格中，在 PlayList#1 中的 PlayListMark() 表格的右侧描述了 Mark() 所属的 PlayItem() 的累积时间。

[0473] 在图 28 中，虽然 mark_data 描述了章节和索引标记代表的章节和索引数目，但是它们可以不必在 mark_data 中描述。相反，通过对 PlayListMark() 中的章节和索引标记进行计数，可以识别出章节和索引数目。

[0474] 【对盘设备的操作的描述】

[0475] 接下来，对图 1 所示的盘设备的操作进行描述，在此假设图 25 到 28 所述的数据（文件）已记录在图 1 所示的盘 101 上。

[0476] 当将盘 101 装入盘设备 102 时，通过图 2A 和图 2B 所示的驱动器接口 114 和操作系统 201 将对应的消息发送到视频内容再现程序 210。当视频内容再现程序 210 已经从操作系统 201 收到表示盘 101 已装入盘设备 102 中的消息时，视频内容再现程序 210 启动图 29 所示的预再现处理。

[0477] 【预再现处理】

[0478] 图 29 是描述视频内容再现程序 210 执行的预再现处理的流程图。

[0479] 应当注意到，盘设备不必以流程图的时序执行操作或者处理。作为选择，盘设备可以并行地或分散地执行操作或者处理。然而，在本说明书中，为方便起见，将对应于该流程图来描述盘设备的操作或者处理。

[0480] 在预再现处理中，在步骤 S101 处，视频内容再现程序 210 利用操作系统 201 的文件系统功能来检查盘 101，并确定对于视频内容再现程序 210 盘 101 是否是正常的盘。

[0481] 如上所述，虽然利用操作系统 201 的文件系统功能对盘 101 进行存取（从中读取文件），但在此省略对其的描述。

[0482] 当在步骤 S101 的确定结果表示盘 101 不是正常的盘，即盘 101 中使用的文件系统不符合操作系统 201、或盘 101 的根目录不包括“视频”目录时，视频内容再现程序 210 确定视频内容再现程序 210 不符合盘 101，流程前进到步骤 S102。在步骤 S102，图形处理模块 219 执行错误处理并结束预再现处理。

[0483] 换言之，图形处理模块 219 作为错误处理而生成表示盘 101 不正常的错误消息（其视频数据），并使视频输出模块 220 输出错误消息以显示该错误消息。例如可以通过从音频输出模块 221 输出报警声音或将盘 101 从盘设备 102 中卸载来执行错误处理。

[0484] 当在步骤 S101 的确定结果表示盘 101 是正常的盘时，流程前进到步骤 S103。在步骤 S103，视频内容再现程序 210 使内容数据提供模块 213 请求操作系统 201 读取存储在盘 101 的“视频”目录（图 4）下的两个数据文件“SCRIPT.DAT”和“PLAYLIST.DAT”。然后，流程前进到步骤 S104。在步骤 S104，将文件“SCRIPT.DAT”提供给脚本控制模块 211。此外，将文件“PLAYLIST.DAT”提供给播放器控制模块 212。

[0485] 然后，流程从步骤 S104 前进到步骤 S105 至步骤 S107。在步骤 S105 到 S107，播放器控制模块 212 进行初始化处理。脚本控制模块 211 等待，直至播放器控制模块 212 完成初始化处理为止。

[0486] 【播放器控制模块 212 的初始化处理】

[0487] 在初始化处理中，在步骤 S105，播放器控制模块 212 对文件“PLAYLIST.DAT”进行分析，并检查文件“PLAYLIST.DAT”中描述的剪辑信息文件的数目以及它们的文件名。

[0488] 换言之，由于文件“PLAYLIST.DAT”如图 25 所示，并且如图 25 所示的文件“PLAYLIST.DAT”中的 number_of_PlayLists 为 2，所以播放器控制模块 212 识别出有两个 PlayList()，它们是 PlayList#0 和 PlayList#1。此外，由于图 25 所示的文件“PLAYLIST.DAT”的第一 PlayList#0 的 number_of_PlayItems 为 2，所以播放器控制模块 212 识别出 PlayList#0 包括两个 PlayItem()，它们是 PlayItem#0 和 PlayItem#1。此后，播放器控制模块 212 参考在图 25 所示的文件“PLAYLIST.DAT”的 PlayList#0 中包括的第一 PlayItem#0 和第二 PlayItem#1 的 Clip_Information_file_name，并识别出包括在 PlayList#0 中的第一 PlayItem#0 的剪辑信息文件（其文件名）是“00001.CLP”，而第二 PlayItem#1 的剪辑信

息文件是“00002.CLP”。

[0489] 类似地，由于number_of_PlayItems为1，播放器控制模块212还识别出第二PlayList#1包括一个PlayItem()(PlayItem#0)，并且由于PlayItem#0的Clip_Information_file_name而识别出PlayItem#0的剪辑信息文件是“00003.CLP”。

[0490] 然后，流程从步骤S105前进到步骤S106。在步骤S106中，播放器控制模块212从盘101的“视频”目录下的“剪辑”目录中读取在步骤S105识别出的剪辑信息文件，即三个剪辑信息文件“00001.CLP”、“00002.CLP”和“00003.CLP”。

[0491] 需要在步骤S106读取的PlayList()的PlayItem的剪辑信息文件仅为第一个被再现的PlayList()的PlayItem的剪辑信息。然而，根据该实施例，如上所述，预先读取PlayList()的PlayItem()的所有剪辑信息文件。

[0492] 在步骤S106之后，流程前进到步骤S107。在步骤S107，播放器控制模块212确定是否已经成功地读取了在步骤S105处识别出的剪辑信息文件。此外，播放器控制模块212确定（检查）在盘101上是否存在对应于剪辑信息文件的剪辑流文件。换言之，在步骤S107，播放器控制模块212确定是否已经成功读取了剪辑信息文件“00001.CLP”、“00002.CLP”和“00003.CLP”，以及对应于剪辑信息文件“00001.CLP”、“00002.CLP”和“00003.CLP”的剪辑流文件“00001.PS”、“00002.PS”和“00003.PS”是否存在于盘101的“视频”目录下的“流”目录中。

[0493] 当在步骤S107的确定结果表示还没有成功读取在步骤S105识别出的剪辑信息文件、或者在盘101上不存在对应于这些剪辑信息文件的剪辑流文件时，即对应于“PLAYLIST.DAT”文件的剪辑信息文件和剪辑流文件还没有记录在盘101上时，视频内容再现程序210确定盘101不正确。此后，流程返回到步骤S102。在步骤S102，视频内容再现程序210执行上述错误处理，并结束预再现处理。

[0494] 另一方面，当在步骤S107的确定结果表示已成功地读取了在步骤S105中识别出的剪辑信息文件、并且对应于这些剪辑信息文件的剪辑流文件存在于盘101上时，播放器控制模块212完成初始化处理。此后，流程前进到步骤S108。

[0495] 在步骤S108，脚本控制模块211解释并执行文件“SCRIPT.DAT”。

[0496] 在脚本控制模块211执行文件“SCRIPT.DAT”时，假设播放器控制模块212已促使第一PlayList()(PlayList#0)被再现，则执行如图30所示的再现处理。

[0497] 【再现处理】

[0498] 图30是视频内容再现程序210执行的再现处理的流程图。

[0499] 【再现准备处理】

[0500] 在步骤S121和S122，播放器控制模块212为PlayList()执行再现准备处理，该PlayList()是已经由脚本控制模块211促使其再现的PlayList()，即第一PlayList()(PlayList#0)。

[0501] 换言之，在步骤S121，播放器控制模块212检查包括在第一PlayList#0中的第一PlayItem#0的IN_time(图6)。此后，流程前进到步骤S122。在步骤S122，播放器控制模块212检查与PlayItem#0的IN_time相对应的、由包括在第一PlayList#0中的第一PlayItem#0再现的剪辑流文件“00001.CLP”的再现开始位置。

[0502] 当PlayItem()的IN_time(图6)表示剪辑流文件的开头时，从该剪辑流文件的开

头读取节目流。然而，当 IN_time 代表除了剪辑流文件的开头之外的其它位置时，播放器控制模块 212 需要搜索（检查）对应于 IN_Time 的位置，并从该位置读取剪辑流文件。

[0503] 具体而言，在图 25 中，包括在第一 PlayList#0 中的第一 PlayItem#0 的 IN_time 为 180,090。播放器控制模块 212 在由包括在第一 PlayList#0 中的第一 PlayItem#0 再现的剪辑流文件“00001.CLP”的 EP_map()（如图 27 所示）中搜索 PlayItem#0 的 IN_time 为 180,090 的再现开始位置。

[0504] 换言之，播放器控制模块 212 通过二分搜索方法等搜索满足 PTS_EP_start ≤ IN_time 的最大 PTS_EP_start，其中 PTS_EP_start 代表在 EP_map() 中描述的可解码开始点。这是因为由 IN_time 所代表的位置并不总是可解码开始点。

[0505] 在这种情况下，如上所述，IN_time 是 180,090。此外，在由包括在第一 PlayList#0 中的第一 PlayItem#0 再现的剪辑流文件“00001.CLP”的 EP_map()（如图 27 所示）中，满足 PTS_EP_start ≤ IN_time 的 PTS_EP_start 的最大值是 180,090。因此，播放器控制模块 212 在图 27 所示的 EP_map() 中搜索作为 180,090 的 PTS_EP_start。

[0506] 此外，播放器控制模块 212 读取 305（扇区）以搜索 RPN_EP_start，并将剪辑流文件“00001.CLP”中由 RPN_EP_start 所表示的位置确定为再现开始位置。

[0507] 在播放器控制模块 212 已经确定了再现开始位置后，流程从步骤 S122 前进到步骤 S123。在步骤 S123，播放器控制模块 212 控制图形处理模块 219 以显示时间码。图形处理模块 219 在播放器控制模块 212 的控制下生成时间码（其视频数据），并将时间码输出到视频输出模块 220。由此，显示时间码。

[0508] 在步骤 S123 显示的时间码例如是其中 PlayList() 的开头转换为 00:00:00（小时：分：秒）的值。除该时间码之外或者代替该时间码，还可以显示章节号和索引号。

[0509] 【对 PlayListMark() 的分析处理】

[0510] 在步骤 S123 已经显示了时间码后，流程前进到步骤 S124。在步骤 S124，播放器控制模块 212 执行分析处理，其对已经由脚本控制模块 211 促使其再现的 PlayList()，即在第一 PlayList()(PlayList#0) 中描述的 PlayListMark()（图 7），进行分析。

[0511] 具体而言，在图 28 的上部表格中，已被预先读取的文件“PLAYLIST.DAT”的第一 PlayList#0 中的 PlayListMark() 的 number_of_PlayList_marks 为 7。因此，播放器控制模块 212 识别出包括在 PlayList#0 中的 Mark() 的数量为 7。

[0512] 此外，播放器控制模块 212 分析图 28 所示上部表格中的 7 个 Mark()，并识别出 7 个 Mark() 中的第一至第四 Mark() 这 4 个 Mark() 属于 PlayList#0 中的第一个 PlayItem()(PlayItem#0)。

[0513] 此后，播放器控制模块 212 获得属于 PlayList#0 的第一 PlayItem#0 的 4 个 Mark() 中的 mark_time_stamp，并将它们作为 4 元素矩阵提供给解码控制模块 214。因此，作为图 28 所示的上部表格中的 7 个 Mark() 中的第一至第四 Mark() 这 4 个 Mark() 的 mark_time_stamp 的 4 个时间 {180,090}、{5,580,090}、{10,980,090} 和 {16,380,090} 从播放器控制模块 212 发送到解码控制模块 214。此时，将这些时间的属性“mark_process（标记处理）”从播放器控制模块 212 发送到解码控制模块 214。当由时间计数部分 214A 计数的时间与具有“markprocess（标记处理）”属性的时间相匹配时，解码控制模块 214 将对应的消息、与具有“mark_process”属性的时间相匹配的时间、以及“mark_process”属性发送到播放器

控制模块 212。

[0514] 【要再现的基本流的判断处理】

[0515] 此后，流程从步骤 S124 前进到步骤 S125。在步骤 S125，播放器控制模块 212 判断要再现的基本流。

[0516] 换言之，在作为已经由脚本控制模块 211 促使其再现的 PlayList() 的第一 PlayList#0 的第一 PlayItem#0（图 25）的 Clip_Information_file_name 中描述的、如图 26A 和 26B 所示的剪辑信息文件“00001.CLP”中，number_of_streams 为 4。因此，播放器控制模块 212 识别出已经用对应的剪辑流文件“00001.PS”多路复用了 4 个基本流。此外，播放器控制模块 212 检查 4 个基本流的剪辑信息文件“00001.CLP”（如图 26A 和 26B 所示）的 StaticInfo() 中的 stream_id 和 private_stream_id，并识别出这 4 个基本流是一个视频流、一个 ATRAC 音频流，以及两个子标题流。换言之，播放器控制模块 212 识别出用剪辑流文件“00001.PS”多路复用的、具有各自属性的基本流的数目。

[0517] 有关用剪辑流文件多路复用的、具有各自属性的基本流的数目的信息用于将一种基本流改变为要再现的另一基本流（从一种音频模式到另一种音频模式，或从一种子标题模式到另一种子标题模式）。当剪辑流文件不包括子标题流文件时，即内容不包括子标题数据时，利用有关具有属性“子标题流”的基本流的数目的信息来确定是否存在子标题流。

[0518] 与 StaticInfo() 的检查结果相对应，播放器控制模块 212 选择并确定要再现的基本流。在这种情况下，用剪辑流文件“00001.PS”多路复用的 4 个基本流包括一个具有属性“视频流”的基本流和一个具有属性“音频流”的基本流。因此，具有属性“视频流”的基本流和具有属性“音频流”的基本流（ATRAC 音频流）被单独确定为要再现的基本流。

[0519] 另一方面，用剪辑流文件“00001.PS”多路复用的 4 个基本流包括两个具有属性“子标题流”的基本流。因此，选择和确定这两个子标题流之一。在该示例中，选择在剪辑信息文件“00001.CLP”的两个子标题流中、最先出现的子标题流。

[0520] 在识别出用剪辑流文件“00001.PS”多路复用的基本流的属性和数目后，需要对四个基本流进行标识。播放器控制模块 212 用 stream_id 和 private_stream_id 来标识用剪辑流文件“00001.PS”多路复用的 4 个基本流。

[0521] 换言之，播放器控制模块 212 利用图 26A 和 26B 所示的剪辑信息文件“00001.CLP”中描述的、作为 0xE0 的 stream_id，来标识用剪辑流文件“00001.PS”多路复用的 4 个基本流中具有“视频流”属性的基本流。

[0522] 此外，播放器控制模块 212 还利用图 26A 和 26B 所示的剪辑信息文件“00001.CLP”中描述的、值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id，来标识用剪辑流文件“00001.PS”多路复用的 4 个基本流中具有“音频流”属性的基本流，即 ATRAC 音频流。

[0523] 此外，播放器控制模块 212 还利用值为 0xBD 的 stream_id 和值为 0x80 的 private_stream_id、以及利用值为 0xBD 的 stream_id 和值为 0x81 的 private_stream_id，来标识用剪辑流文件“00001.PS”多路复用的 4 个基本流中的具有“子标题流”属性的基本流，即两个子标题流。

[0524] 如上所述，可以用被描述为与剪辑流文件相对应的剪辑信息文件的元数据的 stream_id 和 private_stream_id，来标识用剪辑流文件多路复用的基本流。

[0525] stream_id 和 private_stream_id 的组合是一种被提供来扩展 MPEG2- 系统的多路

复用的机制。当将 stream_id 和 private_stream_id 的组合用作作为数据库的元数据时，可以可靠地标识基本流。此外，当扩展 private_stream_id 以便增加对应的基本流的数量和类型时，可以采用当前的机制而无需任何改变。因此，stream_id 和 private_stream_id 的组合具有高扩展性。

[0526] 换言之，例如，蓝光盘 (BD) 标准采用 MPEG2 标准的传输流的分组 ID(PID) 来标识数据。因此，BD 标准受到 MPEG2 标准的限制。另一方面，DVD- 视频标准定义了类似于 private_stream_id 的 sub_stream_id。然而，不能在数据库中描述 sub_stream_id 以标识流。在固定的区域内为仅仅 8 ～ 32 个流的信息描述了 sub_stream_id(见 VI4-49，表 4.2.1-2(VTS_AST_ATRT) 和 VI4-52，表 4.2.1-3(VTS_SPST_ATRT))。因此，sub_stream_id 不具有高可扩展性。

[0527] 另一方面，可以用元数据来描述 stream_id 和 private_stream_id 的组合。例如，图 10 所示的剪辑信息文件 Clip() 可以描述与由 number_of_streams 表示的值相对应的 stream_id 和 private_stream_id 的组合。因此，用剪辑流文件多路复用的基本流可以由作为剪辑信息文件 Clip() 中描述的元数据的 stream_id 和 private_stream_id 的组合来标识，而与基本流的数量无关 (在 number_of_streams 所表示的范围内)。

[0528] 根据这个实施例，将 stream_id 和 private_stream_id 的组合用于标识利用与图 10 所示的剪辑信息文件相对应的剪辑流文件多路复用的基本流。此外，这个组合可以用于标识这样的基本流，该基本流将 Mark() 相关为图 7 所示的 PlayListMark() 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 的组合。此外，stream_id 和 private_stream_id 的组合还用于标识这样的基本流，该基本流描述了图 14 所示的 Ep_map() 中的可解码开始点的信息。

[0529] 【输出属性控制处理】

[0530] 此后，流程从步骤 S125 前进到步骤 S126。在步骤 S126，播放器控制模块 212 对在步骤 S125 处确定为要在步骤 S125 再现的基本流执行输出属性控制处理。

[0531] 具体而言，播放器控制模块 212 检查代表 DynamicInfo()(图 13) 的数量的 number_of_DynamicInfo(图 10)，DynamicInfo() 描述了在步骤 S125 处确定为要再现的视频流、ATRAC 音频流以及子标题流的输出属性。

[0532] 在这种情况下，要再现的视频流、ATRAC 音频流以及子标题流是用剪辑流文件“00001.PS”多路复用的基本流。在图 26A 和 26B 所示的剪辑信息文件“00001.CLP”中，number_of_DynamicInfo 都为 0。当 number_of_DynamicInfo 都为 0 时，播放器控制模块 212 不对要再现的基本流的输出属性执行输出属性控制处理。

[0533] 当要再现的基本流的 number_of_DynamicInfo 不为 0 时，为该基本流执行输出属性控制处理。该输出属性控制处理将在稍后描述。

[0534] 【再现开始准备处理】

[0535] 在步骤 S126 之后，流程前进到步骤 S127。在步骤 S127，播放器控制模块 212 对要再现的基本流执行再现开始准备处理。

[0536] 换言之，播放器控制模块 212 向内容数据提供模块 213 提供已经用于多路复用要再现的基本流的剪辑流文件“00001.PS”的文件名，并提供在步骤 S122 确定的作为再现开始位置的、在 EP_map() 中描述的 RPN_EP_start(＝ 305)。

[0537] 此外，播放器控制模块 212 还在将存储在已经用于多路复用要再现的基本流的剪辑流文件“00001.PS”中的节目流提供给缓冲器控制模块 215 之前，初始化该缓冲器控制模块 215。

[0538] 具体而言，缓冲器控制模块 215（图 3）为存储在数据起始指针存储部分 231 中的数据起始指针、存储在数据写指针存储部分 232 中的数据写指针、存储在视频读指针存储部分 241 中的视频读指针、存储在音频读指针存储部分 251 中的音频读指针，以及存储在子标题读指针存储部分 262 中的子标题读指针设置相同的值。

[0539] 因此，存储在数据起始指针存储部分 231 中的数据起始指针和存储在数据写指针存储部分 232 中的数据写指针在缓冲器控制模块 215 的缓冲器 215A 中保持相同的位置。这表示还没有有效数据存储在缓冲器 215A 中。

[0540] 此外，播放器控制模块 212 向缓冲器控制模块 215 提供作为用于要再现的基本流的标识信息的 stream_id 以及需要时的 private_stream_id。

[0541] 换言之，如上所述，要再现的基本流中具有属性“视频流”的视频流由值为 0xE0 的 stream_id 标识。具有属性“音频流”的 ATRAC 音频流由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 来标识。具有属性“子标题流”的子标题流由值为 0xBD 的 stream_id 和值为 0x80 的 private_stream_id 来标识。播放器控制模块 212 将这些 stream_id 和 private_stream_id 提供给缓冲器控制模块 215。

[0542] 在缓冲器控制模块 215（图 3）中，视频读取功能部分 233 将从播放器控制模块 212 接收的、值为 0xE0 的用于视频流的 stream_id 存储到 stream_id 寄存器 242 中。此外，音频读取功能部分 234 将从播放器控制模块 212 接收的、值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 分别存储到 stream_id 寄存器 252 和 private_stream_id 寄存器 253 中。此外，子标题读取功能部分 235 将从播放器控制模块 212 接收的、值为 0xBD 的 stream_id 和值为 0x80 的 private_stream_id 分别存储到 stream_id 寄存器 263 和 private_stream_id 寄存器 264 中。

[0543] 播放器控制模块 212 存储提供给缓冲器控制模块 215 的、要再现的基本流的 stream_id 和 private_stream_id，以用于以后的处理。当出现流改变请求消息、或标识在标记处理（将在以下描述）中正被再现的流时，播放器控制模块 212 使用 stream_id 和 private_stream_id。

[0544] 为了初始化缓冲器控制模块 215（图 3），播放器控制模块 212 在子标题读取功能标志存储部分 261 中设置具有与用于多路复用要再现的基本流的剪辑流文件对应的值的子标题读取功能标志。

[0545] 换言之，在这种情况下，由于已经用于多路复用要再现的基本流的剪辑流文件“00001.PS”包括子标题流，所以将值为 1 的子标题读取功能标志设置到子标题读取功能标志存储部分 261 中，以激活子标题读取功能部分 235。当已用于多路复用要再现的基本流的剪辑流文件不包括子标题流时，将值为 0 的子标题读取功能标志设置到子标题读取功能标志存储部分 261 中。在这种情况下，不激活子标题读取功能部分 235（子标题读取功能部分 235 不执行任何处理）。

[0546] 此外，播放器控制模块 212 向解码控制模块 214 提供包括在脚本控制模块 211 已经促使播放器控制模块 212 再现的第一 PlayList#0 中的、第一 PlayItem#0（图 25）的值为

180,090 的 IN_time 和值为 27,180,090 的 OUT_time。解码控制模块 214 使用 IN_time 启动对由 PlayItem() 再现的剪辑的解码，并使用 OUT_time 停止对剪辑的解码，并控制将在稍后描述的 PlayItem 改变处理。

[0547] 播放器控制模块 212 初始化其中图形处理模块 219 显示子标题流的子标题流显示模式。换言之，播放器控制模块 212 控制图形处理模块 219 以默认的显示模式显示子标题流。

[0548] 【开始读取数据】

[0549] 此后，流程从步骤 S127 前进到步骤 S128。播放器控制模块 212 使用操作系统 201 的功能控制内容数据提供模块 213，以读取已经用于多路复用要再现的基本流的包括节目流的剪辑流文件。换言之，内容数据提供模块 213 指定盘 101 的“视频”目录下的“流”目录中的剪辑流文件“00001.PS”（图 4），指定已经在步骤 S122 判定的作为再现开始位置的扇区 305，并使操作系统 201 读取该文件。内容数据提供模块 213 使操作系统 201 将已经从盘 101 读取的数据提供给缓冲器控制模块 215。

[0550] 这样，从盘 101 中读取了剪辑流文件“00001.PS”的节目流。将该节目流提供给缓冲器控制模块 215。

[0551] 缓冲器控制模块 215（图 3）将已经从盘 101 读取的节目流写入到缓冲器 215A 中的由数据写指针存储部分 232 中的数据写指针所代表的位置，并将该数据写指针递增所写入数据的大小。

[0552] 除非另外指明，否则当缓冲器控制模块 215 的缓冲器 215A 具有自由空间时，内容数据提供模块 213 就从盘 101 读取数据，并将数据提供并存储到缓冲器控制模块 215 的缓冲器 215A 中。因此，缓冲器 215A 总是存储足够量的数据。

[0553] 【开始控制解码器】

[0554] 当从盘 101 读取数据并将数据存储到缓冲器控制模块 215 的缓冲器 215A 中时，流程从步骤 S128 前进到步骤 S129。在步骤 S129，解码控制模块 214 控制视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218，以便作为预解码操作而开始从缓冲器 215A 读取数据。

[0555] 因此，视频解码器控制模块 216 请求缓冲器控制模块 215 中的视频读取功能部分 233（图 3）发送数据。对应于该请求，视频解码器控制模块 216 从缓冲器控制模块 215 获得存储在缓冲器 215A 中的一个视频存取单元、添加到该视频存取单元中的 PTS 和 DTS（有时也称为时间戳）、作为紧跟在可解码开始点之后的 private_stream_2 的 PES_packet() 中描述的信息的 pic_struct_copy、au_ref_flag 和 AU_length（有时也称为附加信息）。每当视频解码器控制模块 216 获得视频存取单元时，将时间戳从视频解码器控制模块 216 提供给解码控制模块 214。

[0556] 另一方面，音频解码器控制模块 217 请求缓冲器控制模块 215 中的音频读取功能部分 234（图 3）发送数据。对应于该请求，音频解码器控制模块 217 从缓冲器控制模块 215 获得存储在缓冲器 215A 中的一个 (ATRAC) 音频存取单元、以及添加到该音频存取单元的时间戳 (PTS 和 DTS)。每当音频解码器控制模块 217 获得音频存取单元时，就将时间戳从音频解码器控制模块 217 提供给解码控制模块 214。

[0557] 此外，子标题解码器控制模块 218 请求缓冲器控制模块 215 中的子标题读取功能

部分 235(图 3)发送数据。对应于该请求,子标题解码器控制模块 218 从缓冲器控制模块 215 获得存储在缓冲器 215A 中的一个子标题存取单元、以及添加到该子标题存取单元的时间戳。每当子标题解码器控制模块 218 获得音频存取单元时,就将时间戳从子标题解码器控制模块 218 提供给解码控制模块 214。当要再现的基本流不包括子标题流、或缓冲器 215A 中没有存储子标题存取单元时,不将数据从缓冲器控制模块 215 提供给子标题解码器控制模块 218。

[0558] 每当视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 请求缓冲器控制模块 215 发送数据时,它们都将结果发送给解码控制模块 214。

[0559] 当缓冲器控制模块 215 向视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 发送数据时从缓冲器 215A 读取的数据的细节将在以下描述。

[0560] 【开始解码数据】

[0561] 当视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 开始从缓冲器控制模块 215 的缓冲器 215A 读取数据时,流程从步骤 S129 前进到 S130。在步骤 S130,这些模块开始解码已经读取的数据。

[0562] 换言之,解码控制模块 214 使视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 与在步骤 S127 从播放器控制模块 212 提供的包括在 PlayList#0 中的第一 PlayItem#0 的值为 180,090 的 IN_time 相对应地、以及与在步骤 S129 从视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 提供的时间戳相对应地、或必要时在改变的定时处开始解码,以便可靠地同步被解码的数据。

[0563] 在改变的定时处开始解码以便可靠地同步数据的方法例如在日本专利 No.3496725 中描述了。简言之,将从视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 提供的时间戳的最小值设置为时间计数部分 214A 的初始值。时间计数部分 214A 从该设置的时间开始计时。当由时间计数部分 214A 计数的时间与时间戳相匹配时,解码控制模块 214 使这些模块开始解码数据。

[0564] 视频解码器控制模块 216 从解码控制模块 214 接收解码开始命令,将从缓冲器控制模块 215(图 3)的视频读取功能部分 233 获得的一个视频存取单元提供给视频解码器 116(图 1),并使视频解码器 116 解码该视频存取单元。此外,视频解码器控制模块 216 将由视频解码器 116 解码的视频数据提供给图形处理模块 219。

[0565] 此后,视频解码器控制模块 216 使视频解码器 116 每次一个地连续解码从缓冲器控制模块 215 的视频读取功能部分 233 获得的视频存取单元,并将解码后的视频存取单元作为视频数据提供给图形处理模块 219。

[0566] 音频解码器控制模块 217 从解码控制模块 214 接收解码开始命令,将从缓冲器控制模块 215(图 3)的音频读取功能部分 234 获得的一个音频存取单元提供给音频解码器 117(图 1),并使音频解码器 117 解码该音频存取单元。音频解码器控制模块 217 将由音频解码器 117 解码的音频数据提供给音频输出模块 221。

[0567] 此后,音频解码器控制模块 217 使音频解码器 117 每次一个地连续解码从缓冲器控制模块 215 的音频读取功能部分 234 获得的音频存取单元,并将解码后的音频存取单元作为音频数据提供给音频输出模块 221。

[0568] 此外,子标题解码器控制模块 218 从解码控制模块 214 接收解码开始命令,并根据

该命令使内部子标题解码软件对从缓冲器控制模块 215（图 3）的子标题读取功能部分 235 获得的一个子标题存取单元进行解码，并将解码后的子标题存取单元作为子标题数据（子标题的图像数据）提供给图形处理模块 219。

[0569] 此后，子标题解码器控制模块 218 使内部解码软件每次一个地连续解码从缓冲器控制模块 215 的子标题读取功能部分 235 获得的子标题存取单元，并将解码后的子标题存取单元作为子标题数据提供给图形处理模块 219。

[0570] 【图形处理】

[0571] 此后，流程从步骤 S130 前进到步骤 S131。在步骤 S131，图形处理模块 219 对从视频解码器控制模块 216 提供的视频数据、以及需要时对从子标题解码器控制模块 218 提供的子标题数据进行图形处理。

[0572] 换言之，图形处理模块 219 执行子标题处理，该处理例如对应于从播放器控制模块 212 接收的显示模式命令而对由子标题解码器控制模块 218 提供的子标题数据进行放大或缩小。当图形处理模块 219 还没有从播放器控制模块 212 接收到显示模式命令、或存在默认的显示模式时，图形处理模块 219 存储从子标题解码器控制模块 218 接收的子标题数据。

[0573] 此外，图形处理模块 219 将从视频解码器控制模块 216 接收到的视频数据和从子标题解码器控制模块 218 接收到的子标题数据或已处理的子标题数据相加，获得已经叠加了子标题数据的输出视频数据，并将该叠加的视频数据提供给视频输出模块 220。

[0574] 当图形处理模块 219 从脚本控制模块 211 或播放器控制模块 212 接收到用于菜单、消息、时间码、章节号、或索引号的信息显示命令时，图形处理模块 219 生成信息，将其与输出视频数据叠加，并将该叠加的数据提供给视频输出模块 220。

[0575] 【输出处理】

[0576] 在步骤 S131 之后，流程前进到步骤 S132。在步骤 S132，视频输出模块 220 将从图形处理模块 219 提供的输出视频数据连续地存储到 FIFO 220A 中，并以预定的输出速率输出存储在 FIFO 220A 中的视频数据。

[0577] 只要 FIFO 220A 具有足够的存储容量（自由空间），视频输出模块 220 就接收来自图形处理模块 219 的输出视频数据。当 FIFO 220A 没有足够的存储容量时，视频输出模块 220 使图形处理模块 219 停止接收输出视频数据。因此，图形处理模块停止接收输出数据。此外，视频输出模块 220 使视频解码器控制模块 216 和子标题解码器控制模块 218 停止它们的处理。因此，视频解码器控制模块 216 和子标题解码器控制模块 218 停止它们的处理。

[0578] 在视频输出模块 220 已经使图形处理模块 219 停止接收输出视频数据且 FIFO 220A 已经输出了视频数据之后，当 FIFO 220A 具有了足够的存储容量时，视频输出模块 220 使图形处理模块 219 接收输出视频数据。与输出视频数据类似，图形处理模块 219 使得视频解码器控制模块 216 和图形处理模块 219 停止接收数据。因此，图形处理模块 219、视频解码器控制模块 216 和子标题解码器控制模块 218 恢复停止的处理。

[0579] 另一方面，音频输出模块 221 也使 FIFO 221A 连续地存储在步骤 S130 从音频解码器控制模块 217 提供的音频数据，并以预定的输出速率（采样频率）输出音频数据。

[0580] 只要 FIFO 221A 具有足够的存储容量（空余空间），音频输出模块 221 就接收来自音频解码器控制模块 217 的音频数据。然而，当 FIFO 221A 没有足够的存储容量时，音频

输出模块 221 使音频解码器控制模块 217 停止接收音频数据。因此，音频解码器控制模块 217 停止其处理。

[0581] 在音频输出模块 221 已经使音频解码器控制模块 217 停止接收音频数据且 FIFO 221A 已经输出了音频数据之后，当FIFO 221A 具有足够的存储容量时，音频输出模块 221 使音频解码器控制模块 217 接收音频数据。因此，音频解码器控制模块 217 恢复停止的处理。

[0582] 按照以上所述的方式，在视频输出模块 220 和音频输出模块 221 输出数据时，对基本流进行解码。

[0583] 如图 1 所示的盘设备与图 29 和图 30 所示的流程图相对应地再现来自盘 101 的数据。接下来，将描述在再现来自盘 101 的数据时对盘设备执行的其它处理或操作。

[0584] 【改变 PlayItem】

[0585] 如图 29 和图 30 所述，再现图 25 所示的第一 PlayList#0 中的第一 PlayItem#0。对应于 PlayList#0，在再现第一 PlayItem#0 之后，再现第二 PlayItem#1。换言之，执行将 PlayItem 从 PlayItem#0 改变到 PlayItem#1 的 PlayItem 改变处理。

[0586] 接下来，将参考图 31 所示的流程图对 PlayItem 改变处理进行描述。

[0587] 如图 29 和图 30 所述，在再现了图 25 所示的 PlayList#0 的第一 PlayItem#0（其剪辑）之后，在再现第一 PlayItem#0 时，解码控制模块 214（图 2A 和图 2B）检查由时间计数部分 214A 计数的时间。

[0588] 【结束 PlayItem#0 的再现】

[0589] 当时间计数部分 214A 已经计数的时间变为 27,180,090（图 25）、即与在图 30 所示的步骤 S127 从播放器控制模块 212 提供的第一 PlayItem#0 的 OUT_time 相同时，解码控制模块 214 在步骤 S151 执行解码停止控制，以停止再现 PlayItem#0。

[0590] 换言之，解码控制模块 214 操作视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 以停止它们的解码操作。解码控制模块 214 控制视频输出模块 220 连续地输出视频数据。

[0591] 此外，解码控制模块 214 向播放器控制模块 212 发送表示第一 PlayItem#0 已被再现的消息。

[0592] 【开始再现 PlayItem#1】

[0593] 如上所述，播放器控制模块 212 已经在图 29 所示的步骤 S105 处识别出第一 PlayList#0 包括第一 PlayItem#0 和第二 PlayItem#1。当解码控制模块 214 已经向播放器控制模块 212 通知了第一 PlayItem#0 已被再现时，流程从步骤 S151 前进到步骤 S152。在步骤 S152，以与第一 PlayItem#0 相同的方式，播放器控制模块 212 开始再现第二 PlayItem#1。

[0594] 类似于图 30 所示的步骤 S122，播放器控制模块 212 将在 EP_map() 中描述的 RPN_EP_start 之一确定为第二 PlayItem#1 的再现开始位置。

[0595] 此外，播放器控制模块 212 如图 30 所示的步骤 S124 所述、识别出属于第二 PlayItem#1 的 Mark()，以及如图 30 所示的步骤 S125 所述、识别出利用由 PlayItem#1 再现的剪辑流文件“00002.PS”多路复用的具有属性的基本流的数量，并确定要再现的基本流。

[0596] 播放器控制模块 212 执行与图 30 所示的步骤 S127 相同的处理。

[0597] 换言之，播放器控制模块 212 向内容数据提供模块 213 提供被确定为再现开始位

置的 EP_map() 中的 RPN_EP_start、以及用要再现的基本流多路复用的剪辑流文件的文件名，即与在第二 PlayItem#1（图 25）的 Clip_Information_file_name 中描述的“00002.CLP”相对应的剪辑流文件“00002.PS”的文件名。

[0598] 此外，在将存储在利用要再现的基本流多路复用的剪辑流文件“00002.PS”中的节目流提供给缓冲器控制模块 215 之前，播放器控制模块 212 初始化缓冲器控制模块 215。

[0599] 换言之，缓冲器控制模块 215（图 3）对以下指针设置相同的值，即存储在数据起始指针存储部分 231 中的数据起始指针、存储在数据写指针存储部分 232 中的数据写指针、存储在视频读指针存储部分 241 中的视频读指针、存储在音频读指针存储部分 251 中的音频读指针，以及存储在子标题读指针存储部分 262 中的子标题读指针。

[0600] 此外，播放器控制模块 212 向缓冲器控制模块 215 提供作为标识要再现的基本流的标识信息的 stream_id、以及需要时的 private_stream_id。

[0601] 缓冲器控制模块 215 中的视频读取功能部分 233（图 3）从播放器控制模块 212 接收要再现的基本流中的视频流的 stream_id，并将其存储到 stream_id 寄存器 242 中。此外，音频读取功能部分 234 从播放器控制模块 212 接收要再现的基本流中的音频流的 stream_id 和 private_stream_id，并将它们存储到 stream_id 寄存器 252 和 private_stream_id 寄存器 253 中。

[0602] 因为利用要再现的基本流多路复用的剪辑流文件“00002.PS”包括子标题流，所以从播放器控制模块 212 向子标题读取功能部分 235 提供要再现的基本流中的子标题流的 stream_id 和 private_stream_id。子标题读取功能部分 235 将 stream_id 和 private_stream_id 分别存储到 stream_id 寄存器 263 和 private_stream_id 寄存器 264 中。

[0603] 播放器控制模块 212 在子标题读取功能标志存储部分 261 中设置具有与用要再现的基本流多路复用的剪辑流文件相对应的值的子标题读取功能标志，以便初始化缓冲器控制模块 215（图 3），。

[0604] 在这种情况下，由于用要再现的基本流多路复用的剪辑流文件“00002.PS”包括子标题流，所以将值为 1 的子标题读取功能标志设置到子标题读取功能标志存储部分 261 中，以激活子标题读取功能部分 235。

[0605] 播放器控制模块 212 向解码控制模块 214 提供作为要再现的第二 PlayItem#1（图 25）的 IN_time 的 90,000 以及作为其 OUT_time 的 27,090,000。

[0606] 此外，播放器控制模块 212 初始化用于图形处理模块 219 的子标题流显示模式命令。换言之，播放器控制模块 212 控制图形处理模块 219 以默认的显示模式显示子标题数据。

[0607] 当要再现的子标题流的 configurable_flag（图 12）为 1 时（其表示允许改变显示模式），播放器控制模块 212 向图形处理模块 219 发送的子标题流显示模式命令可能是当前显示模式命令。

[0608] 以与第一 PlayItem#0 同样的方式再现第二 PlayItem#1。在再现第二 PlayItem#1 时，解码控制模块 214 监控由时间计数部分 214A 计数的时间。当时间计数部分 214A 已计数的时间变为 27,090,000（图 25）、即与在步骤 S152（图 31）从播放器控制模块 212 提供的第二 PlayItem#1 的 OUT_time 相同时，解码控制模块 214 执行与在步骤 S151 相同的解码停止控制，以停止再现 PlayItem#1。

[0609] 【显示时间码】

[0610] 接下来，如上所述，在图 30 所示的步骤 S123，显示时间码。时间码被连续更新。

[0611] 接下来，将参考图 32 所示的流程图描述时间码显示处理。

[0612] 当解码控制模块 214 的时间计数部分 214A（图 2A 和图 2B）已经计数了一秒时，流程前进到步骤 S171。在步骤 S171，解码控制模块 214 向播放器控制模块 212 提供表示已经过一秒的消息以及时间计数部分 214A 已计数的当前时间。然后流程前进到步骤 S172，在步骤 S172，播放器控制模块 212 接收来自解码控制模块 214 的消息和当前时间，并将当前时间转换为时间码。然后，流程前进到步骤 S173。

[0613] 在步骤 S173，播放器控制模块 212 控制图形处理模块 219 以显示在步骤 S172 获得的时间码。此后，流程返回到步骤 S171。

[0614] 因此，时间码以一秒的间隔被更新。时间码的更新间隔不限于一秒。

[0615] 【改变流】

[0616] 用图 26A 和 26B 所述的两个子标题流对由构成图 25 所述的第一 Playlist#0 的第一 PlayItem#0 再现的剪辑流文件“00001.PS”和由第二 PlayItem#1 再现的剪辑流文件“00002.PS”进行多路复用。

[0617] 当用剪辑流文件多路复用了具有相同属性的多个基本流时，要再现的基本流可以从一个基本流改变到另一个基本流。

[0618] 接下来，将参考图 33 所示的流程图对流改变处理进行描述。

[0619] 当流改变命令被描述为在例如文件“SCRIPT.DAT”文件（图 4）中的脚本程序，并且脚本控制模块 211 执行该脚本程序、或者用户操作遥控器来改变流时，将流改变命令提供给播放器控制模块 212。

[0620] 当脚本控制模块 211 执行描述该流改变请求的脚本程序时，脚本控制模块 211 向播放器控制模块 212 提供流改变请求消息。当用户利用遥控器输入流改变命令时，输入接口 115 从遥控器接收该流改变命令信号，并向播放器控制模块 212 提供该流改变请求消息。

[0621] 当将促使播放器控制模块 212 改变子标题流的子标题流改变请求消息提供给播放器控制模块 212 时，播放器控制模块 212 检查在图 30 的步骤 S125 中已识别出的、要再现的基本流的子标题流的数目。

[0622] 当播放器控制模块 212 已经检查的子标题流的数目为 1 或更小时，播放器控制模块 212 忽略该子标题流改变请求消息。因此，播放器控制模块 212 不执行从步骤 S192 到步骤 S194 的处理。

[0623] 相反，而当子标题流的数目为 2 或更多时，流程前进到步骤 S192 至步骤 S194。在这些步骤中，播放器控制模块 212 将正再现的子标题流改变为另一子标题流。

[0624] 换言之，在步骤 S192，播放器控制模块 212 标识在剪辑信息文件上正在被再现的子标题流。具体而言，假定正在再现与组成图 25 所述的第一 PlayList#0 的第二 PlayItem#1 相对应的、其 stream_id 为 0xBD 且 private_stream_id 为 0x80、并且是用剪辑流文件“00002.PS”多路复用的子标题流，则播放器控制模块 212 在步骤 S192 将利用剪辑流文件“00002.PS”多路复用的两个子标题流中的、正被再现的子标题流标识为 stream#2，其是图 26A 和图 26B 所示剪辑信息文件“00002.CLP”中的第三子标题流。

[0625] 此后，流程前进到步骤 S193。在步骤 S193，播放器控制模块 212 将在步骤 S192 标

识的剪辑信息文件中的下一个子标题流识别（标识）为下一个要再现的子标题流。在图26A和图26B中，第三子标题流stream#2的下一个子标题流是第四子标题流stream#3。因此，在步骤S193，播放器控制模块212将第四子标题流stream#3识别为下一个要再现的子标题流。

[0626] 当正被再现的子标题流被标识为利用剪辑流文件“00002.PS”多路复用的两个子标题流中的stream#3（其是在图26A和26B所示的剪辑信息文件“00002.CLP”中的第四子标题流）时，播放器控制模块212例如将第三子标题流stream#2识别为下一个要再现的子标题流。

[0627] 此后，流程前进到步骤S194。在步骤S194，播放器控制模块212向缓冲器控制模块215（图3）中的子标题读取功能部分235提供在步骤S193被识别为下一个要再现的子标题流的子标题流的stream_id和private_stream_id，以便子标题读取功能部分235从缓冲器215A中读取由stream_id和private_stream_id标识的下一个子标题存取单元。

[0628] 缓冲器控制模块215（图3）中的子标题读取功能部分235将在步骤S194从播放器控制模块212提供的stream_id和private_stream_id分别新设置到stream_id寄存器263和private_stream_id寄存器264中。子标题读取功能部分235读取由分别新设置到stream_id寄存器263和private_stream_id寄存器264中的stream_id和private_stream_id所标识的下一个子标题存取单元。

[0629] 在上述方式中，将正被再现的子标题流改变为下一个要再现的另一子标题流。

[0630] 【缓冲器控制模块215的处理】

[0631] 接下来，将参考图34到图38描述缓冲器控制模块215（图3）的处理、向缓冲器215A写入数据的处理和从缓冲器215A读取数据的处理。

[0632] 如图3所示，缓冲器控制模块215具有用于向缓冲器215A写入数据和从缓冲器215A读取数据的5个指针。

[0633] 换言之，如图34和图35所示，缓冲器控制模块215具有存储在数据起始指针存储部分231中的数据起始指针、存储在数据写指针存储部分232中的数据写指针、存储在视频读指针存储部分241中的视频读指针、存储在音频读指针存储部分251中的音频读指针、以及存储在子标题读指针存储部分262中的子标题读指针。

[0634] 在图34和35中，省略了图3所示的视频读取功能部分233中的stream_id寄存器242和au_information()寄存器243、音频读取功能部分234中的stream_id寄存器252和private_stream_id寄存器253、以及子标题读取功能部分235的子标题读取功能标志存储部分261、stream_id寄存器263和private_stream_id寄存器264。

[0635] 存储在数据起始指针存储部分231中的数据起始指针代表存储在缓冲器215A中的最老数据（需要被读取但尚未读取）的位置。存储在数据写指针存储部分232中的数据写指针表示数据在缓冲器215A中的写入位置。该位置是写入最新数据的位置。

[0636] 存储在视频读指针存储部分241中的视频读指针代表从缓冲器215A读取的视频流的位置。存储在音频读指针存储部分251中的音频读指针代表从缓冲器215A读取的音频流的位置。存储在子标题读指针存储部分262中的子标题读指针表示从缓冲器215A读取的子标题流的位置。

[0637] 如图3所述，数据起始指针、数据写指针、视频读指针、音频读指针、以及子标题读

指针在缓冲器 215A 中沿顺时针方向移动。

[0638] 如图 35 所示，根据该实施例，总是更新数据起始指针，以便其代表与视频读指针、音频读指针、和子标题读指针的最老数据位置相同的位置。在图 35 中，音频读指针代表在视频读指针、音频读指针、或子标题读指针中的最老数据的位置。数据起始指针与音频读指针相匹配。

[0639] 在具有数据起始指针、数据写指针、视频读指针、音频读指针、和子标题读指针的缓冲器控制模块 215 中，当从盘 101 读取新数据并写入到缓冲器 215A 中时，沿顺时针方向更新数据写指针，从而数据写指针代表紧跟在新写入的数据之后的位置。

[0640] 当从缓冲器 215A 读取视频流、音频流、或子标题流时，视频读指针、音频读指针、或子标题读指针沿顺时针方向被更新所读取的数据量。所读取的数据量是实际读取的视频数据、音频数据、或子标题数据和介于所读取的数据之间并在读取数据时被忽略的另一流的数据部分的总和。

[0641] 当更新视频读指针、音频读指针、或子标题读指针时，数据起始指针也被更新，以便其代表由视频读指针、音频读指针、或子标题读指针所代表的最老数据的位置。

[0642] 缓冲器控制模块 215 控制缓冲器 215A 的数据写入操作，从而使数据写指针不会超过数据起始指针。

[0643] 除非数据写指针超过了数据起始指针，否则缓冲器控制模块 215 就将从盘 101 读取的数据写到缓冲器 215A 中、由数据写指针所代表的位置处，并更新数据写指针。另一方面，如果数据写指针超过了数据起始指针，则缓冲器控制模块 215 使内容数据提供模块 213 停止从盘 101 读取数据并停止向缓冲器 215A 写入数据。因此，可以防止缓冲器 215A 的溢出。

[0644] 如前所述，与两个指针（即数据起始指针和数据写指针）的位置关系相对应地将从盘 101 读取的数据写到缓冲器 215A 中。

[0645] 另一方面，缓冲器控制模块 215 控制缓冲器 215A 的数据读取操作，以便视频读指针、音频读指针、子标题读指针、以及数据起始指针不会超过数据写指针。

[0646] 换言之，除非视频读指针、音频读指针、或子标题读指针超过了数据写指针，否则缓冲器控制模块 215 对应于从视频解码器控制模块 216、音频解码器控制模块 217、或子标题解码器控制模块 218 接收的请求，而从缓冲器 215A 中由视频读指针、音频读指针、或子标题读指针所代表的位置读取数据，并更新视频读指针、音频读指针、或子标题读指针，需要的话还更新数据起始指针。另一方面，如果视频读指针、音频读指针、或子标题读指针超过了数据写指针，则缓冲器控制模块 215 促使视频解码器控制模块 216、音频解码器控制模块 217、或子标题解码器控制模块 218 停止发送请求，或等待直到缓冲器 215A 存储了足够的数据为止。结果，可以防止缓冲器 215A 下溢。

[0647] 因此，缓冲器 215A 沿顺时针方向、在从数据起始指针表示的位置到由数据写指针表示的位置的区域内（图 34 和图 35 中的阴影部分），存储要提供给视频解码器控制模块 216、音频解码器控制模块 217、和子标题解码器控制模块 218 的数据。此外，视频读指针、音频读指针、和子标题读指针存在于该区域中。

[0648] 在以上情况下，更新数据起始指针，以便其表示由视频读指针、音频读指针、以及子标题读指针所表示的最老数据的位置。作为选择，还可以更新数据起始指针，以便其表示

比最老数据的位置早预定时间（如 1 秒）的数据的位置。

[0649] 可以假定在视频读指针、音频读指针、和子标题读指针中的视频读指针和音频读指针表示最老数据的位置。

[0650] 因此，如图 34 所示，当更新数据起始指针、以便其表示比视频读指针或音频读指针所表示的最老数据的位置早例如 1 秒的数据的位置时，可将比视频读指针或音频读指针所表示的最老数据早 1 秒的数据存储在缓冲器 215A 中。在图 34 中，音频读指针表示最老数据的位置，而数据起始指针表示比最老数据早 1 秒的数据的位置。

[0651] 当更新数据起始指针、以便其表示比最老数据的位置早 1 秒的数据的位置时，可以改进盘设备的响应。

[0652] 换言之，如图 35 所示，当更新数据起始指针以便其表示音频读指针所表示的最老数据的位置时，如果发出了例如反向再现 (reverse reproduction) 命令的特定再现命令，则需要再从盘 101 重新读取已从缓冲器 215A 读取的数据。因此，从发出特定再现命令到执行该特定再现命令需要一定的时间。

[0653] 相反，如图 34 所示，当更新数据起始指针以便其表示比由音频读指针所表示的最老数据的位置早 1 秒的数据的位置时，如果发出了例如反向再现命令的特定再现命令，并且开始该特定再现操作所需的数据是比最老数据早 1 秒的数据且存储在缓冲器 215A 中，则可以快速开始该特定再现操作，而无需从盘 101 重读数据。

[0654] 当更新数据起始指针以便其表示比由音频读指针所表示的最老数据的位置早 1 秒的数据的位置时，开始特定再现操作所需的数据可能未存储在缓冲器 215A 中。在这种情况下，需要从盘 101 重新读取开始特定再现操作所需的数据。

[0655] 接下来，将详细描述从缓冲器 215A 中读取视频流、音频流、和子标题流的操作。

[0656] 如图 30 中的步骤 S127 所述，当开始剪辑流文件的再现操作时，缓冲器控制模块 215 初始化数据起始指针、数据写指针、视频读指针、音频读指针、以及子标题读指针，以便它们表示缓冲器 215A 中的相同位置。

[0657] 当从盘 101 读取存储在剪辑流文件中的节目流 (MPEG2- 系统节目流)、并将其提供给缓冲器控制模块 215 时，缓冲器控制模块 215 将节目流存储在缓冲器 215A 中的数据写指针所表示的位置。此外，沿顺时针方向更新数据写指针。

[0658] 此外，缓冲器控制模块 215 的视频读取功能部分 233（图 3）对存储在缓冲器 215A 中的节目流进行分析，对应于从视频解码器控制模块 216 接收的请求而从存储在缓冲器 215A 中的节目流中提取（分离）出视频流（视频存取单元），并将提取的视频流提供给视频解码器控制模块 216。

[0659] 类似地，音频读取功能部分 234 对存储在缓冲器 215A 中的节目流进行分析，对应于从音频解码器控制模块 217 接收的请求而从存储在缓冲器 215A 中的节目流中提取出音频流（音频存取单元），并将该音频流提供给音频解码器控制模块 217。子标题读功能部分 235 对存储在缓冲器 215A 中的节目流进行分析，对应于从子标题解码器控制模块 218 接收的请求而从存储在缓冲器 215A 中的节目流中提取出子标题流（子标题存取单元），并将该子标题流提供给子标题解码器控制模块 218。

[0660] 【读取视频流】

[0661] 接下来，将参考图 36 所示的流程图详细描述对于视频读功能部分 233（图 3）的缓

冲器 215A 的视频流读取处理。

[0662] 在步骤 S211，视频读取功能部分 233 在存储在缓冲器 215A 中的节目流中搜索 private_stream_2 的 PES_packet()。如图 20 所述，在 private_stream_2 的 PES_packet() 中，stream_id 为 10111111B(= 0xBF)。视频读取功能部分 233 寻找其 stream_id 为 10111111B 的 PES_packet()。

[0663] 假设用存储在剪辑流文件“00001.PS”中的节目流多路复用的基本流是要再现的基本流，则当从盘 101 读取节目流并将其存储到缓冲器 215A 中时，用剪辑流文件“00001.PS”的 EP_map()(图 27) 中描述的关于可解码开始点的信息来确定扇区 305 为再现开始位置。在图 30 中的步骤 S128，指定作为再现开始点的扇区 305。视频读取功能部分 233 促使操作系统 201 从剪辑流文件“00001.PS”中读取节目流。

[0664] 在视频流的 EP_map() 中描述的、有关可解码开始点的信息表示紧跟在实际可解码开始点之后的 private_stream_2 的 PES_packet() 的位置。

[0665] 因此，在从盘 101 读取存储在剪辑流文件“00001.PS”中的节目流并将其存储到缓冲器 215A 中之后，立即将 private_stream_2 的 PES_packet() 存储到缓冲器 215A 中由数据起始指针和视频读指针表示的位置。

[0666] 当视频读取功能部分 233 在步骤 S211 已找到 private_stream_2 的 PES_packet() 时，流程前进到步骤 S212。在步骤 S212，视频读取功能部分 233 从 private_stream2_PES_payload()(图 23) 中提取 video_stream_id，其是 private_stream_2 的 PES_packet() 的 PES_packet_data_byte。在图 30 所示的步骤 S127，视频读取功能部分 233 确定 video_stream_id 是否与存储在 stream_id 寄存器 242(图 3) 中的要再现的视频流的 stream_id 相匹配。

[0667] 当在步骤 S212 的确定结果表示 private_stream2_PES_payload() 中描述的 video_stream_id 与存储在 stream_id 寄存器 242 中的 stream_id 不匹配时，即在步骤 S211 找到的 private_stream_2 的 PES_packet() 不在要再现的视频流的可解码开始点时，则流程返回到步骤 S211。在步骤 S211，视频读取功能部分 233 在缓冲器 215A 中存储的节目流中搜索 private_stream_2 的 PES_packet() 并重复相同的处理。

[0668] 相反，当步骤 S212 处的确定结果表示 private_stream2_PES_payload() 中描述的 video_stream_id 与存储在 stream_id 寄存器 242 中的 stream_id 相匹配时，即在步骤 S211 找到的 private_stream_2 的 PES_packet() 在要再现的视频流的可解码开始点处时，则流程前进到步骤 S213。在步骤 S213，视频读取功能部分 233 从缓冲器 215A 中读取在 private_stream_2 的 PES_packet() 的 private_stream2_PES_payload() 中描述的 au_information()，并将 au_information() 存储到 au_information() 寄存器 243(图 3) 中。此后，流程前进到步骤 S214。

[0669] 在步骤 S214，视频读取功能部分 233 将存储在视频起始指针存储部分 231 中的视频读指针更新达到在步骤 S211 找到的 private_stream_2 的 PES_packet() 的大小 (即 video_stream_id(图 23) 与存储在 stream_id 寄存器 242(图 3) 中的 stream_id 相匹配)。

[0670] 换言之，在剪辑流文件中，在 private_stream_2 的 PES_packet() 之后紧跟其 stream_id 与 video_stream_id 相匹配的视频流 (PES_packet())。因此，在步骤 S214，视频读取功能部分 233 更新视频读指针，以便使其表示视频流的可解码开始点的位置。

[0671] 此后，流程从步骤 S214 前进到步骤 S215。视频读取功能部分 233 确定视频解码器控制模块 216 是否已发出了数据请求。当在步骤 S215 的确定结果表示视频解码器控制模块 216 未发出数据请求时，流程返回到步骤 S215。在步骤 S215，视频读取功能部分 233 重复相同的操作。

[0672] 相反，当在步骤 S215 的确定结果表示视频解码器控制模块 216 已经发出了数据请求，则流程前进到步骤 S216。在步骤 S216，视频读取功能部分 233 从缓冲器 215A 中的音频读指针所表示的位置开始对节目流进行分析，从缓冲器 215A 中读取由存储在 au_information() 寄存器 243 中的 au_information() 的 Au_length 所描述的字节数的数据（即一个视频存取单元），将数据提供给视频解码器控制模块 216，并将视频读指针更新达到已从缓冲器 215A 中读取的、一个视频存取单元的大小。

[0673] 换言之，如图 24 所述，au_information() 描述 number_of_access_unit，其表示从包括 au_information() 的 private_stream_2 的 PES_packet() 到下一个 private_stream_2 的 PES_packet() 所包括的视频存取单元（画面）的数目。

[0674] 此外，如图 24 所示，au_information() 描述作为关于由 number_of_access_unit 表示的每个视频存取单元的信息的 pic_struct_copy、au_ref_flag 和 AU_length。

[0675] 如图 24 所述，因为在对应于 number_of_access_unit 的 au_information() 中描述的每个 AU_length 表示从 private_stream_2 的 PES_packet() 到下一个 private_stream_2 的 PES_packet() 的、由 number_of_access_unit 表示的每个视频存取单元的大小，所以视频读取功能部分 233 可以用 AU_length 提取存取单元，而无需对视频流进行分析。

[0676] 换言之，当提取 MPEG2- 视频或 MPEG4-AVC 存取单元时，需要了解视频流的结构并对其进行分析。然而，在记录在盘 101 上的剪辑流文件中存储的节目流包括 private_stream_2 的 PES_packet()，其描述表示视频存取单元的大小的 AU_length，并且其后紧跟视频流的至少一个可解码开始点。因此，视频读取功能部分 233 可以与在 private_stream_2 的 PES_packet() 中描述的 AU_length 相对应、从缓冲器 215A 读取视频存取单元（作为视频存取单元的视频流），并将视频存取单元提供给视频解码器控制模块 216，而无需对视频流进行分析。

[0677] 在步骤 S216，当视频读取功能部分 233 向视频解码器控制模块 216 提供视频存取单元时，视频读取功能部分 233 还向视频解码器控制模块 216 提供作为关于视频存取单元的信息的在 au_information() 中描述的 pic_struct_copy、au_ref_flag 和 AU_length，以及为每个视频存取单元添加的时间戳 (PTS 和 DTS)。

[0678] 在视频读取功能部分 233 已经从缓冲器 215A 读取了一个视频存取单元、并将其提供给视频解码器控制模块 216 之后，流程前进到步骤 S217。在步骤 S217，视频读取功能部分 233 确定其是否已处理了由存储在 au_information() 寄存器 243 中的 au_information() (图 24) 的 number_of_access_unit 所表示的存取单元。

[0679] 当在步骤 S217 的确定结果表示视频读取功能部分 233 尚未处理完由 number_of_access_unit 表示的存取单元时，即视频读取功能部分 233 尚未从缓冲器 215A 读取由 number_of_access_unit 表示的存取单元并将它们提供给视频解码器控制模块 216 时，流程返回到步骤 S215。在步骤 S215，视频读取功能部分 233 重复相同的处理。

[0680] 相反，当在步骤 S217 的确定结果表示视频读取功能部分 233 已经处理了由 number_of_access_unit 所表示的存取单元时，即视频读取功能部分 233 已经从缓冲器 215A 读取了由 number_of_access_unit 表示的存取单元并将它们提供给了视频解码器控制模块 216 时，流程返回到步骤 S211。在步骤 S211，视频读取功能部分 233 搜索 private_stream_2 的 PES_packet()，并重复相同的处理。

[0681] 【读取音频流】

[0682] 接下来，将参考图 37 所示的流程图描述音频读取功能部分 234（图 3）对缓冲器 215A 的音频流读取处理。

[0683] 在步骤 S230，音频读取功能部分 234 确定在图 30 所示的步骤 S127 处存储到 stream_id 寄存器 252 中的、要再现的音频流的 stream_id 是否表示 private_stream_1 的 PES_packet()。

[0684] 当在步骤 S230 的确定结果表示存储在 stream_id 寄存器 252 中的 stream_id 不表示 private_stream_1 的 PES_packet() 时，即如图 20 所示，当存储在 stream_id 寄存器 252 中的 stream_id 是分配给已与 MPEG 标准相对应地编码的音频流的 110xxxxxB 时，流程前进到步骤 S231。在步骤 S231，音频读取功能部分 234 在存储在缓冲器 215A 中的节目流中搜索表示 MPEG 音频中定义的音频帧的开头的同步码。因为该同步码的位置在音频帧的开头处，所以音频读取功能部分 234 更新音频读指针，以使其表示音频帧开头的位置。此后，流程从步骤 S231 前进到步骤 S232。在步骤 S232，音频读取功能部分 234 与音频读指针表示的位置相对应地、在存储在缓冲器 215A 中的节目流中搜索与存储在 stream_id 寄存器 252 中的 stream_id 相匹配的 PES_packet()，并获得该 PES_packet()。此后，流程前进到步骤 S233。

[0685] 在步骤 S233，音频读取功能部分 234 更新存储在音频读指针存储部分 251 中的音频读指针，从而音频读指针表示已经在步骤 S232 找到的、PES_packet() 的 PES_packet_data_byte（图 16A 和 16B 到图 18A 和 18B）的开头。此后，流程前进到步骤 S237。

[0686] 在步骤 S237，音频读取功能部分 234 确定音频解码器控制模块 217 是否已发出了数据请求。当步骤 S237 的确定结果表示音频解码器控制模块 217 还未发出数据请求时，流程返回到步骤 S237。在步骤 S237，音频读取功能部分 234 重复同样的处理。

[0687] 相反，当步骤 S237 的确定结果表示音频解码器控制模块 217 已经发出了数据请求时，流程前进到步骤 S238。在步骤 S238，音频读取功能部分 234 从缓冲器 215A 中的音频读指针所表示的位置开始对节目流进行分析，从缓冲器 215A 读取一个具有预定固定长度的音频存取单元，并将该音频存取单元与添加到音频存取单元的时间戳（PTS 和 DTS）一起提供给音频解码器控制模块 217。

[0688] 音频读取功能部分 234 更新音频读指针达到从缓冲器 215A 读取的一个音频存取单元的大小。然后，流程返回到步骤 S237。在步骤 S237，音频读取功能部分 234 重复同样的处理。

[0689] 相反，当步骤 S230 的确定结果表示存储在 stream_id 寄存器 252 中的 stream_id 表示 private_stream_1 的 PES_packet() 时，即如图 20 所述，存储在 stream_id 寄存器 252 中的 stream_id 为 10111101B（＝ 0xBD）并表示 private_stream_1 的 PES_packet() 时，流程前进到步骤 S234。在步骤 S234，音频读取功能部分 234 在存储在缓冲器 215A 中的节目

流中搜索 private_stream_1 的 PES_packet() 并获得该 PES_packet()。换言之，音频读取功能部分 234 搜索其 stream_id 为 10111101B 的 PES_packet() 并获得该 PES_packet()。

[0690] 当音频读取功能部分 234 在步骤 S234 已经找到了 private_stream_1 的 PES_packet() 时，流程前进到步骤 S235。在步骤 S235，音频读取功能部分 234 从 private_stream1_PES_payload()（图 21）中提取作为 private_stream_1 的 PES_packet() 的 PES_packet_data_byte 的 private_stream_id，并确定 private_stream_id 是否与已经在图 30 所示的步骤 S127 中存储到 private_stream_id 寄存器 253 中的、要再现的音频流的 private_stream_id 相匹配。

[0691] 当在步骤 S235 的确定结果表示 private_stream1_PES_payload() 中描述的 private_stream_id 与存储在 private_stream_id 寄存器 253 中的 private_stream_id 不匹配时，即在步骤 S234 找到的 private_stream_1 的 PES_packet() 不是要再现的音频流时，流程返回到步骤 S234。在步骤 S234，音频读取功能部分 234 在存储在缓冲器 215A 中的节目流中搜索 private_stream_1 的 PES_packet()。此后，音频读取功能部分 234 重复同样的处理。

[0692] 另一方面，当在步骤 S235 的确定结果表示 private_stream1_PES_payload() 中描述的 private_stream_id 与存储在 private_stream_id 寄存器 253 中的 private_stream_id 相匹配时，即在步骤 S234 找到的 private_stream_1 的 PES_packet() 是要再现的音频流时，流程前进到步骤 S236。在步骤 S236，音频读取功能部分 234 从缓冲器 215A 中读取 private_stream_1 的 PES_packet() 的 private_stream1_PES_payload()（图 21）中描述的 AU_locator，将紧跟在 AU_locator 后的位置与 AU_locator 所表示的值相加，并获得音频存取单元的起始位置。

[0693] 换言之，如图 21 所述，AU_locator 表示基于紧跟在 AU_locator 之后的位置存储在 private_stream1_PES_payload() 的 private_payload() 中的音频存取单元（或子标题存取单元）的起始位置。因此，通过将 AU_locator 的值与紧跟在 AU_locator 之后的位置相加，可以获得音频存取单元的（绝对）起始位置。

[0694] 在步骤 S236，音频读取功能部分 234 更新存储在音频读指针存储部分 251 中的音频读指针，从而使音频读指针表示已经获得的音频存取单元的起始位置。此后，流程前进到步骤 S237。

[0695] 在步骤 S237，音频读取功能部分 234 确定音频解码器控制模块 217 是否已发出了数据请求。当步骤 S237 的确定结果表示音频解码器控制模块 217 还未发出数据请求时，流程返回到步骤 S237。在步骤 S237，音频读取功能部分 234 重复同样的处理。

[0696] 相反，当步骤 S237 的确定结果表示音频解码器控制模块 217 已经发出了数据请求时，流程前进到步骤 S238。在步骤 S238，音频读取功能部分 234 从缓冲器 215A 中由音频读指针表示的位置开始对节目流进行分析，从缓冲器 215A 读取一个具有预定长度的音频存取单元，并将该音频存取单元与添加到该音频存取单元的时间戳一起提供给音频解码器控制模块 217。

[0697] 音频读取功能部分 234 更新音频读指针达到从缓冲器 215A 读取的一个音频存取单元的大小。然后，流程返回到步骤 S237。在步骤 S237，音频读取功能部分 234 重复相同的处理。

[0698] 【读取子标题流】

[0699] 接下来，将参考图 38 所示的流程图对子标题读取功能部分 235（图 3）对缓冲器 215A 的子标题流读取处理进行描述。

[0700] 在步骤 S251，子标题读取功能部分 235 确定已在图 30 所示的步骤 S127 中存储到视频解码器控制模块 216 中的子标题读取功能标志。当在步骤 S251 的确定结果表示子标题读取功能标志为 0 时，即用要再现的基本流多路复用的剪辑流文件不包括子标题流、并且在图 30 所示的步骤 S127 处已将 0 设置到子标题读取功能标志存储部分 261 中时，子标题读取功能部分 235 不执行任何处理。

[0701] 相反，当在步骤 S251 中的确定结果表示子标题读取功能标志为 1 时，即用要再现的基本流多路复用的剪辑流文件包括子标题流、并且在图 30 所示的步骤 S127 中已将 1 设置到子标题读取功能标志存储部分 261 中时，流程前进到步骤 S252。在步骤 S252，子标题读取功能部分 235 在存储在缓冲器 215A 中的节目流中搜索与已存储在 stream_id 寄存器 263（图 3）中的、要再现的子标题流的 stream_id 相匹配的 PES_packet()。

[0702] 如图 30 所示的步骤 S127 所述，要再现的子标题流的 stream_id 被存储在 stream_id 寄存器 263（图 3）中。另一方面，如图 20 所述，子标题流的 stream_id 为 10111101B（＝0xBD），其表示 private_stream_1 的 PES_packet()。

[0703] 因此，在步骤 S252，子标题读取功能部分 235 在存储在缓冲器 215A 中的节目流中搜索 private_stream_1 的 PES_packet()。

[0704] 当子标题读取功能部分 235 已经搜索了 private_stream_1 的 PES_packet() 并获得它之后，流程前进到步骤 S253。子标题读取功能部分 235 从 private_stream_PES_payload()（图 21）中提取 private_stream_id，其是 private_stream_1 的 PES_packet() 的 PES_packet_data_byte，并确定 private_stream_id 是否与在图 30 所示步骤 S127 中已经存储到 private_stream_id 寄存器 2264 中的要再现的子标题流的 private_stream_id 相匹配。

[0705] 当在步骤 S253 的确定结果表示 private_stream_PES_payload() 中描述的 private_stream_id 与存储在 private_stream_id 寄存器 264 中的 private_stream_id 不匹配时，即已经在步骤 S252 获得的 private_stream_1 的 PES_packet() 不是要再现的子标题流时，流程返回到步骤 S252。在步骤 S252，子标题读取功能部分 235 在存储在缓冲器 215A 中的节目流中搜索另一个 private_stream_1 的 PES_packet()。此后，子标题读取功能部分 235 重复同样的处理。

[0706] 相反，当在步骤 S253 的确定结果表示 private_stream1_PES_payload() 中描述的 private_stream_id 与存储在 private_stream_id 寄存器 264 中的 private_stream_id 相匹配时，即已经在步骤 S252 获得的 private_stream_1 的 PES_packet() 是要再现的子标题流时，流程前进到步骤 S254。在步骤 S254，子标题读取功能部分 235 从缓冲器 215A 中读取在 private_stream_1 的 PES_packet() 的 private_stream1_PES_payload()（图 21）中描述的 AU_locator，将紧跟在 AU_locator 之后的位置与 AU_locator 所表示的值相加，并获得子标题存取单元的起始位置。

[0707] 如图 21 所述，AU_locator 表示基于紧跟在 AU_locator 之后的位置存储在 private_stream1_PES_payload() 的 private_payload() 中的子标题存取单元（或音频存

取单元）的起始位置。因此，通过将 AU_locator 所表示的值与紧跟在 AU_locator 之后的位置相加，可以获得子标题存取单元的（绝对）起始位置。

[0708] 此外，在步骤 S254，子标题读取功能部分 235 更新存储在子标题读指针存储部分 262 中的子标题读指针，以便子标题读指针表示所获得的子标题存取单元的起始位置。此后，流程前进到步骤 S255。

[0709] 在步骤 S255，子标题读取功能部分 235 确定子标题解码器控制模块 218 是否已经发出了数据请求。当步骤 S255 的确定结果表示子标题读取功能部分 235 还未发出数据请求时，流程返回到步骤 S255。在步骤 S255，子标题读取功能部分 235 重复同样的处理。

[0710] 相反，当步骤 S255 的确定结果表示子标题解码器控制模块 218 已经发出了数据请求时，流程前进到步骤 S256。在步骤 S256，子标题读取功能部分 235 从缓冲器 215A 中子标题读指针表示的位置开始对节目流进行分析，从缓冲器 215A 读取一个其长度在该子标题存取单元的开头描述的子标题存取单元，并将子标题存取单元与添加到该子标题存取单元的时间戳一起提供给子标题解码器控制模块 218。如图 2A 和图 2B 中所述，子标题存取单元的大小在该子标题存取单元的开头处描述。子标题读取功能部分 235 从缓冲器 215A 中由子标题读指针表示的位置开始读取该大小的子标题存取单元，并将子标题存取单元与添加到该子标题存取单元的时间戳一起提供给子标题解码器控制模块 218。

[0711] 子标题读取功能部分 235 将子标题读指针更新从缓冲器 215A 读取的一个子标题存取单元的大小。然后，流程返回到步骤 S255。在步骤 S255，子标题读取功能部分 235 重复相同的处理。

[0712] 【再同步处理】

[0713] 接下来，将对图 2A 和图 2B 所示的解码控制模块 214 对视频数据和音频数据的同步控制进行描述。

[0714] 如图 30 所示的步骤 S130 所述，解码控制模块 214 使视频解码器控制模块 216、音频解码器控制模块 217 和子标题解码器控制模块 218 开始对它们的数据进行解码。如果需要，解码控制模块 214 促使这些模块以不同的定时开始对它们的数据进行解码，以使它们同步。例如，当视频解码器 116 和音频解码器 117 执行其解码处理时，根据它们的进展程度，它们可能以不同的定时输出视频数据和音频数据。

[0715] 因此，解码控制模块 214 执行再同步处理，以对视频数据和音频数据输出定时的差别进行补偿，并使视频解码器 116 和音频解码器 117 同步地输出视频数据和音频数据。

[0716] 接下来，将参考图 39 所示的流程图描述再同步处理。

[0717] 在再同步处理中，在步骤 S271，解码控制模块 214 确定出从视频解码器控制模块 216 输出的视频存取单元的时间戳与从音频解码器控制模块 217 输出的音频存取单元的时间戳之间的差别是否是大的。

[0718] 如图 30 所示的步骤 S129 所述，每当视频解码器控制模块 216 从缓冲器控制模块 215 接收到视频存取单元时，视频解码器控制模块 216 向解码控制模块 214 提供该视频存取单元的时间戳。类似地，每当音频解码器控制模块 217 从缓冲器控制模块 215 接收音频存取单元时，音频解码器控制模块 217 向解码控制模块 214 提供该音频存取单元的时间戳。

[0719] 在步骤 S271，解码控制模块 214 比较在相同定时（在被认为是相同定时的预定时间段内）处、从视频解码器控制模块 216 和音频解码器控制模块 217 接收的时间戳，并确定

所述时间戳的差别是否是大的。

[0720] 当在步骤 S271 确定的结果表示在从视频解码器控制模块 216 接收的视频存取单元的时间戳和从音频解码器控制模块 217 接收的音频存取单元的时间戳之间的差别不大时，即在视频存取单元的时间戳和音频存取单元的时间戳之间的差别处于可认为存取单元是同步的预定范围内（例如两个视频帧（约 66 毫秒））时，流程返回到 S271。在步骤 S271，解码控制模块 214 确定（监视）时间戳之间的差别。

[0721] 反之，当在步骤 S271 的确定结果表示从视频解码器控制模块 216 接收的视频存取单元的时间戳和从音频解码器控制模块 217 接收的音频存取单元的时间戳之间的差别大时，即该差别不处于存取单元可被认为是同步的预定范围之内时，流程前进到 S272。在步骤 S272，解码控制模块 214 比较从视频解码器控制模块 216 接收的视频存取单元的时间戳和从音频解码器控制模块 217 接收的音频存取单元的时间戳，以确定视频数据的输出和音频数据的输出中的哪一个更晚。

[0722] 当在步骤 S272 的确定结果表示视频数据的输出晚于音频数据的输出时，流程前进到 S273。在步骤 S273，解码控制模块 214 使视频解码器控制模块 216 停止解码和输出（显示）视频存取单元，即跳过对视频存取单元的处理，以使该处理前进一个视频存取单元。此后流程前进到步骤 S274。

[0723] 在步骤 S274，视频解码器控制模块 216 从解码控制模块 214 接收跳过请求，并检查与视频存取单元一起从缓冲器控制模块 215 提供的 au_ref_flag（图 24）。

[0724] 换言之，存储在 private_stream_2 的 PES_packet() 的 private_stream2_PES_payload()（图 23）中的 au_information()（图 24）包括作为关于存取单元的信息的 au_ref_flag。如图 30 所示的步骤 S129 和如图 36 所示的步骤 S216 所述，与视频存取单元一起，缓冲器控制模块 215 还向视频解码器控制模块 216 提供其 au_ref_flag。

[0725] 在步骤 S274，视频解码器控制模块 216 检查与存取单元一起提供的、该存取单元的 au_ref_flag。

[0726] 此后，流程从步骤 S274 前进到步骤 S275。在步骤 S275，与对已经从缓冲器控制模块 215 提供的该视频存取单元的 au_ref_flag 的检查结果相对应，视频解码器控制模块 216 确定该视频存取单元是否是在对其它画面进行解码时未被参考的非参考图像。

[0727] 如图 24 所述，视频存取单元中的 au_ref_flag 表示该存取单元是否是参考图像。当存取单元是参考图像时，au_ref_flag 为 1。相反，当存取单元不是参考图像时，au_ref_flag 为 0。

[0728] 当步骤 S275 的确定结果表示从缓冲器控制模块 215 提供的视频存取单元不是非参考图像（其视频存取单元）时，即从缓冲器控制模块 215 提供的视频存取单元是参考图像时，流程前进到步骤 S276。在步骤 S276，视频解码器控制模块 216 使视频解码器 116 正常地处理视频存取单元。在视频解码器控制模块 216 已经从缓冲器控制模块 215 收到下一个视频存取单元之后，流程返回到步骤 S274。

[0729] 相反，当步骤 S275 的确定结果表示从缓冲器控制模块 215 提供的视频存取单元是非参考图像时，流程前进到步骤 S277。在步骤 S277，视频解码器控制模块 216 使视频解码器 116 跳过对该视频存取单元的处理。在缓冲器控制模块 215 已经提供了下一个视频存取单元之后，流程返回到步骤 S271。

[0730] 因为跳过了用于视频存取单元的处理，所以该处理前进了几乎一个视频存取单元（减小了处理时间）。因此，晚于音频数据的输出的视频数据的输出前进了。

[0731] 相反，当步骤 S272 的确定结果表示视频数据的输出不晚于音频数据的输出，即音频数据的输出晚于视频数据的输出时，流程前进到步骤 S278。在步骤 S278，解码控制模块 214 向视频解码器控制模块 216 输出连续输出命令，以连续地输出与被解码的视频存取单元相对应的视频数据，以保持该视频解码器控制模块 216 等待对下一个视频存取单元的处理。此后，流程前进到步骤 S279。

[0732] 在步骤 S279，视频解码器控制模块 216 从解码控制模块 214 接收连续输出请求，并且与该连续输出请求相对应地、将正被解码的视频存取单元的视频数据连续地输出到图形处理模块 219。在缓冲器控制模块 215 已经提供了下一个视频存取单元之后，流程前进到步骤 S271。

[0733] 如上所述，解码控制模块 214 确定视频数据的输出是否晚于音频数据的输出。当视频数据的输出晚于音频数据的输出时，解码控制模块 214 使视频解码器控制模块 216 跳过对一个存取单元的处理。视频解码器控制模块 216 与存取单元的 au_ref_flag 相对应地确定要跳过的存取单元是参考图像还是非参考图像。当存取单元是非参考图像时，解码控制模块 214 使视频解码器 116 跳过对存取单元的处理。因此，可以容易地同步视频数据的输出和音频数据的输出。

[0734] 换句话说，当要跳过的存取单元是参考图像时，需要解码该存取单元中的视频数据，以便当解码其它存取单元时参考该视频数据。因此，当同步视频数据的输出和音频数据的输出时，如果跳过了对参考图像存取单元的处理，则不能解码参考该参考图像的其它存取单元。因此，当显示与音频数据同步的视频数据时，出现了噪声。

[0735] 因此，优选为跳过不是非参考图像的存取单元。

[0736] 另一方面，为了在传统的基本流中搜索非参考图像的存取单元，需要分析该基本流。与例如 MPEG4-AVC 系统相对应地编码的基本流是非常复杂的。因此，当分析该基本流时，需要非常高的成本。

[0737] 相反，除包括视频存取单元的、具有 PES_packet_data_byte 的 PES_packet()（图 16A 和图 16B 到图 18A 和图 18B）之外，还用包括 private_stream2_PES_payload()（图 23）（其是 PES_packet_data_byte 的扩展）的 private_stream_2 的 PES_packet() 来多路复用存储在记录在盘 101 上的剪辑流文件中的节目流。private_stream2_PES_payload() 中的 au_information()（图 24）描述了 au_ref_flag，其表示该视频存取单元是参考图像还是非参考图像。au_ref_flag 和对应的视频存取单元从缓冲器控制模块 215 提供给视频解码器控制模块 216。因此，视频解码器控制模块 216 可以通过检查视频存取单元的 au_ref_flag 来确定视频存取单元是参考图像还是非参考图像，而几乎没有额外的成本。

[0738] 【标记处理】

[0739] 接下来，将参考图 40 所示的流程图描述基于在 PlayListMark()（图 7）中描述的 Mark() 的标记处理。

[0740] 解码控制模块 214 一直检查由内置时间计数部分 214A 计数的当前时间。在步骤 S301，解码控制模块 214 确定当前时间是否与在 PlayListMark()（图 7）中描述的任何 Mark() 的 mark_time_stamp 相匹配。

[0741] 如图 30 所示步骤 S124 所述，当播放器控制模块 212 再现图 25 所示的第一个 PlayList#0 中的第一个 PlayItem#0 时，播放器控制模块 212 识别出在图 28 所示的上部表格的 PlayListMark() 中包括的七个 Mark() 中的四个 Mark()（第一到第四个 Mark()）属于 PlayList() 中的第一个 PlayItem#0，并且将作为这四个 Mark() 的 mark_time_stamp 的 {180,090}、{5,580,090}、{10,980,090} 和 {16,380,090} 连同表示 mark_time_stamp 所表示的时间属性是“标记处理”的信息一起提供给解码控制模块 214。

[0742] 在步骤 S301，解码控制模块 214 确定已经从播放器控制模块 212 提供的、具有属性“标记处理”的四个时间 (mark_time_stamp) 中的哪个与当前时间相匹配。

[0743] 当步骤 S301 的确定结果表示当前时间不与任何具有属性“标记处理”的时间相匹配时，该流程返回到步骤 S301。在步骤 S301，解码控制模块 214 重复相同的处理。

[0744] 相反，当步骤 S301 的确定结果表示当前时间与具有属性“标记处理”的四个时间之一相匹配时，解码控制模块 214 将表示当前时间成为具有属性“标记处理”的时间的消息连同具有属性“标记处理”的匹配时间一起提供给播放器控制模块 212。此后，流程前进到步骤 S302。

[0745] 在步骤 S302，播放器控制模块 212 从解码控制模块 214 接收表示当前时间成为具有属性“标记处理”的时间的消息、连同具有属性“标记处理”的匹配时间，并识别其 mark_time_stamp 与当前时间相匹配的 Mark()，将其作为在标记处理中将被处理的 Mark()（以下有时也将该 Mark() 称为目标标记）。

[0746] 换句话说，播放器控制模块 212 已经识别出正再现的 PlayList() 中的 PlayItem()。通过参考具有 PlayList()、PlayItem()、和具有属性″标记处理″的已匹配时间 (mark_time_stamp)（在下文中有时将这个时间称为标记时间）的“PLAYLIST.DAT”文件（图 5）的 PlayListMark()（图 7），播放器控制模块 212 识别出目标标记。

[0747] 具体而言，假定正在再现图 25 所示的第一个 PlayList#0 中的第一个 PlayItem#0 时，播放器控制模块 212 识别出标记时间是包括在图 28 所示的上部表格的 PlayListMark() 中的七个 Mark() 中的四个 Mark()（其是第一到第四个 Mark()）中任何一个的 mark_time_stamp。

[0748] 当从解码控制模块 214 提供给播放器控制模块 212 的标记时间例如是 16,380,090 时，播放器控制模块 212 将在图 28 所示的上部表格的 PlayListMark() 中包括的四个 Mark()（其是第一到第四个 Mark()）中的、mark_time_stamp 与标记时间 16,380,090 相匹配的第四个 Mark() 识别为目标标记。

[0749] 当播放器控制模块 212 已经识别出目标标记时，流程从步骤 S302 前进到步骤 S303。在步骤 S303，播放器控制模块 212 确定该目标标记是否描述了标识基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id（图 7）。

[0750] 当步骤 S303 的确定结果表示目标标记未描述标识基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id 时，即 entry_ES_stream_id 和 entry_ES_private_stream_id 二者都是 0x00 时，流程前进到步骤 S305，跳过步骤 S304。在步骤 S305，解码控制模块 214 执行对目标标记的处理。

[0751] 相反，当步骤 S303 的确定结果表示目标标记描述了标识基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id（图 7）时，流程前进到步骤 S304。在步骤 S304，

播放器控制模块 212 确定正被再现的基本流是否包括由 entry_ES_stream_id 以及必要时的 entry_ES_private_stream_id 所标识的基本流。

[0752] 当步骤 S304 的确定结果表示正被再现的基本流不包括由目标标记中的 entry_ES_stream_id 和 entry_ES_private_stream_id 所标识的基本流时，流程返回到步骤 S301。换句话说，当不正在再现由目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 所标识的基本流时，忽略该目标标记

[0753] 相反，当步骤 S304 的结果表示正被再现的基本流包括由目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 标识的基本流时，即正在再现由目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 标识的基本流时，确定该目标标记是有效的。此后，流程前进到步骤 S305。在步骤 S305，播放器控制模块 212 执行对目标标记的处理。

[0754] 换句话说，在步骤 S305，通过参考目标标记的 mark_type（图 7），播放器控制模块 212 确定目标标记。

[0755] 当步骤 S305 的确定结果表示目标标记是章节标记或者索引标记时，即目标标记中的 mark_type 是“Chapte（章节）”或者“Index（索引）”时，流程前进到步骤 S306。在步骤 S306，播放器控制模块 212 使图形处理模块 219 用目标标记的章节号或者索引号来更新章节号或者索引号。此后，流程返回到步骤 S301。

[0756] 当步骤 S305 的确定结果表示目标标记是事件标记时，即目标标记的 mark_type 是“Event（事件）”时，流程前进到步骤 S307。在步骤 S307，播放器控制模块 212 向脚本控制模块 211 通知表示已经发生了事件的事件消息和目标标记的 mark_data。此后，流程前进到步骤 S308。

[0757] 在步骤 S308，脚本控制模块 211 从播放器控制模块 212 接收事件消息和 mark_data，并且与该事件消息的中断请求相对应地、利用 mark_data 的参数执行文件“SCRIPT.DAT”中描述的处理序列。此后，流程返回到步骤 S301。

[0758] 换句话说，脚本控制模块 211 执行与 mark_data 相对应的处理。

[0759] 具体而言，在图 28 所示的下部表格中的 PlayList#1 的 PlayListMark() 中，第二 Mark()(Mark#1) 和第三 Mark()(Mark#2) 中的每一个的 mark_type 是“Event（事件）”，而且 Mark#1 的 mark_data 不同于 Mark#2 的 mark_data。

[0760] 当脚本控制模块 211 接收到与第二 Mark() 相对应的事件消息以及与第三 Mark() 相对应的事件消息时，脚本控制模块 211 用相同的事件处理器（中断处理例程）执行与所接收的事件消息相对应的处理。脚本控制模块 211 检查连同事件消息一起提供的 mark_data，并且执行与 mark_data 相对应的处理。

[0761] 具体而言，当 mark_data 例如为 1 时，脚本控制模块 211 控制图形处理模块 219 以显示第一类型图标。当 mark_data 例如为 2 时，脚本控制模块 222 控制图形处理模块 229 以显示第二类型图标。

[0762] mark_data 不局限于 1 和 2。此外，与 mark_data 相对应的处理不限于简单图标的显示。

[0763] 换句话说，当 mark_data 处于从 3 到 18 的范围中时，脚本控制模块 211 控制图形处理模块 219，以便用与从 mark_data 中减去 2 的值（在从 1 到 16 的范围中的值）相对应

的强度显示第一类型图标。另一方面，当 mark_data 处于从 19 到 34 的范围中时，脚本控制模块 1811 控制图形处理模块 219，以便用与从 mark_data 中减去 18 的值（在从 1 到 16 的范围中的值）相对应的强度显示第二类型图标。

[0764] 当用户操作的控制器连接到输入接口 115（图 1）而且控制器具有振动电机（其是具有安装在电机轴上的偏心陀 (eccentric_weight) 的直流 (DC) 电机并且当驱动该电机时发生振动），如果 mark_data 的值处于从 35 到 42 的范围内，则可以驱动振动电机达到与从 mark_data 中减去 34 的值（从 1 到 8 范围中的值）相对应的操作时间段。

[0765] mark_data 是数字值。可以用脚本控制模块 211 执行的脚本程序描述 mark_data 的使用和算法。因此，可以与由盘 101 的制造商或者提供记录在盘 101 上的数据的内容供应商指定的预定规则或者原始规则相对应地使用 mark_data。

[0766] 当当前时间匹配具有属性“标记处理”的时间时，从作为具有属性“标记处理”的时间的标记时间中识别出目标标记。当目标标记未描述标识基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id 时，执行与目标标记的 mark_type 相对应的处理。即使目标标记描述了标识基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id 的话，在正再现由 entry_ES_stream_id 和 entry_ES_private_stream_id 所标识的基本流的同时，执行与目标标记的 mark_type 相对应的处理。

[0767] 当正再现图 25 所示的第二 PlayList#1 时，执行以下的标记处理。

[0768] 换句话说，如图 28 中的下部表格所示，第二 PlayList#1 中的 PlayListMark() 描述了其 mark_time_stamp 分别为 90,000、27,090,000、和 27,540,000 的第一 Mark() (Mark#0)、第二 Mark() (Mark#1)、和第三 Mark() (Mark#2)。

[0769] 此外，因为在图 28 所示的下部表格中的 PlayListMark() 的第二 Mark() 和第三 Mark() 的 entry_ES_stream_id 描述了 0xE0 和 0xE1，所以第二 Mark() 和第三 Mark() 与分别为 0xE0 和 0xE1 的 stream_id 所标识的基本流相关。

[0770] 如图 25 所述，第二 PlayList#1 仅仅描述了一个 PlayItem() (PlayItem#0)。利用 PlayItem#0，再现剪辑流文件“00003.PS”。如与剪辑流文件“00003.PS”相对应的图 26A 和图 26B 所示的剪辑信息文件“00003.CLP”所述，用三个基本流多路复用该剪辑流文件“00003.PS”，这三个基本流是用值为 0xE0 的 stream_id 标识的视频流 stream#0、用值为 0xE1 的 stream_id 标识的视频流 stream#1、和用值为 0x00 的 private_stream_id 标识的音频 stream#2。

[0771] 因此，在图 28 所示的下部表格中的 PlayListMark() 的第二个 Mark() 与用剪辑流文件“00003.PS”多路复用的其 stream_id 为 0xE0 的视频流文件 stream#0 相关。第三个 Mark() 与用剪辑流文件“00003.PS”多路复用的其 stream_id 为 0xE1 的视频流 stream#1 相关。

[0772] 当再现图 25 所示的第二个 PlayItem#1 的 PlayItem#0 时，如图 30 所示的步骤 S124 所述，播放器控制模块 212 识别出在图 28 所示的下部表格中的 PlayListMark() 包括的三个 Mark() 属于 PlayList#1 的 PlayItem#0，并且将作为三个 Mark() 的 mark_time_stamp 的 {90,000}、{27,090,000}、和 {27,540,000} 连同表示时间具有属性“标记处理”的信息一起提供给解码控制模块 214。

[0773] 在标记处理中，当正再现 PlayList#1 中的 PlayItem#0 时，解码控制模块 214 一直

确定时间 {90,000}、{27,090,000}、和 {27,540,000} 中的哪个与由时间计数部分 214A 所计数的当前时间相匹配（步骤 S301）。当当前时间与具有属性“标记处理”的时间相匹配时，解码控制模块 214 将作为与当前时间相匹配的具有属性″ 标记处理″ 的时间的标记时间连同表示当前时间变为具有属性“标记处理”的时间的消息一起提供给播放器控制模块 212。

[0774] 当当前时间与具有属性“标记处理”的时间 {90,000}、{27,090,000}、和 {27,540,000} 中的 27,090,000 相匹配时，解码控制模块 214 将具有属性“标记处理”的标记时间 27,090,000 连同表示当前时间变为具有属性“标记处理”的时间的消息一起提供给播放器控制模块 212。

[0775] 播放器控制模块 212 已识别出正在再现 PlayList#1 中的 PlayItem#0。播放器控制模块 212 将 90,000、27,090,000、和 27,540,000（它们是属于在图 28 所示的下部表格中的 PlayListMark() 中描述的 Mark() 的 PlayItem#0 的三个 Mark() 的 mark_time_stamp）与 27,090,000（其是从解码控制模块 214 提供的标记时间）相比较，并且识别出其 mark_time_stamp 与作为标记时间的 27,090,000 相匹配的 Mark()（即在图 28 所示的下部表格的 PlayListMark() 中描述的第二个 Mark()(Mark#1)）是目标标记（步骤 S302）。

[0776] 在作为目标标记的、在图 28 所示的下部表格的 PlayListMark() 中描述的第二个 Mark() 中，entry_ES_stream_id 为 0xE0。如上所述，值为 0xE0 的 entry_ES_stream_id 表示用剪辑流文件“00003.PS”多路复用的、其 stream_id 为 0xE0 的视频流 stream#0（图 26A 和图 26B）。播放器控制模块 212 确定正被再现的基本流是否包括视频流 stream#0（步骤 S303 和 S304）。

[0777] 当正被再现的基本流不包括视频流 stream#0 时，播放器控制模块 212 忽略该目标标记（步骤 S304）。

[0778] 相反，当正被再现的基本流包括视频流 stream#0 时，播放器控制模块 212 认为目标标记是有效的，并且执行与该目标标记相对应的处理（步骤 S305 到 S308）。

[0779] 在这种情况下，在作为目标标记的、在图 28 所示的下部表格的 PlayListMark() 中描述的第二个 Mark() 中，mark_type 是“Event（事件）”。因此，第二个 Mark() 是事件标记。播放器控制模块 212 将表示发生了事件的事件消息和该目标标记的 mark_data 一起提供给脚本控制模块 211（步骤 S305 和 S307）。与从播放器控制模块 212 接收的事件消息的中断请求相对应地，脚本控制模块 211 利用 mark_data 的参数执行在“SCRIPT.DAT”中描述的处理序列（步骤 S308）。

[0780] 如上所述，在标记处理中，播放器控制模块确定当前时间是否匹配 mark_time_stamp，其中当前时间是与表示在 PlayList() 的时间轴上的一个再现时间的 mark_time_stamp、表示 Mark() 类型的 mark_type、以及 PlayList() 相对应地再现的剪辑流文件的再现时间，而 PlayList() 包括不具有 Mark() 或具有超过一个 Mark() 的 PlayListMark()（图 7），且 Mark() 包括作为事件标记的参数的 mark_data。当当前时间匹配 mark_time_stamp 时，播放器控制模块 212 将具有 mark_time_stamp 等于作为当前时间的标记时间的 Mark() 识别为目标标记。当具有目标标记的 mark_type 表示事件发生的类型时，即目标标记是事件标记时，提供目标标记所具有的 mark_type 和事件消息。播放器控制模块 212 执行与 mart_data 相对应的处理。因此，播放器控制模块 212 可以在剪辑流文件的再现时间中执行

与 mark_data 相对应的处理。

[0781] 【输出属性控制处理】

[0782] 接下来，将参考图 41 所示的流程图详细描述图 30 所示的步骤 S126 执行的输出属性控制处理。

[0783] 如图 30 所示的步骤 S126 所述，播放器控制模块 212 检查表示 DynamicInfo()（图 13）的数目的 number_of_DynamicInfo（图 10），其中 DynamicInfo() 描述了至少一个要再现的基本流（即已经在步骤 S125 判定的、要再现的至少一个基本流）的输出属性。

[0784] 当至少一个要再现的基本流中的每一个的 number_of_DynamicInfo 是 0 时，播放器控制模块 212 不执行任何处理。

[0785] 相反，当要再现的基本流的 number_of_DynamicInfo 不是 0 时，播放器控制模块 212 执行与图 41 所示的流程图相对应的输出属性控制处理。

[0786] 因此，当记录在盘 101 上的三个剪辑信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”如图 26A 和图 26B 所示、而且再现与剪辑信息文件“00001.CLP”相对应的剪辑流文件“00001.PS”（再现剪辑流文件“00001.PS”的第一 PlayList#0 中的第一 PlayItem#0）时，因为全部四个基本流（它们是 stream#0 到 stream#3）的 number_of_DynamicInfo 是 0 时，所以播放器控制模块 212 不执行输出属性控制处理。

[0787] 同样地，当再现与剪辑信息文件“00002.CLP”相对应的剪辑流文件“00002.PS”（再现剪辑流文件“00002.PS”的第一 PlayList#0 中的第二 PlayItem#1）时，因为用剪辑流文件“00002.PS”多路复用的、作为 stream#0 到 stream#3 的四个基本流的全部 number_of_DynamicInfo 都是 0，所以播放器控制模块 212 不执行输出属性控制处理。

[0788] 相反，当再现与剪辑信息文件“00003.CLP”相对应的剪辑流“00003.PS”时（再现剪辑流文件“00003.PS”的第二 PlayList#1 中的 PlayItem#0），因为作为第一基本流的视频流 stream#0 和作为第三基本流的音频流 stream#2 的 number_of_DynamicInfo 分别为 2 和 3，所以播放器控制模块 212 执行输出属性控制处理。

[0789] 换句话说，在输出属性控制处理中，在步骤 S320，播放器控制模块 212 将在与要再现的剪辑流文件相对应的剪辑信息文件 Clip()（图 10）中描述的 pts_change_point 连同表示具有属性“DynamicInfo() 处理”的时间的信息一起提供给解码控制模块 214。解码控制模块 214 从播放器控制模块 212 接收 pts_change_point，其是具有属性“DynamicInfo() 处理”的时间。此后，流程前进到步骤 S321。

[0790] 在步骤 S321，解码控制模块 214 确定由时间计数部分 214A 计数的当前时间是否与作为具有属性“DynamicInfo() 处理”的时间的 pts_change_point 之一相匹配。当步骤 S321 的确定结果表示当前时间不与任何一个 pts_change_point 相匹配时，流程返回到步骤 S321。

[0791] 相反，当步骤 S321 的确定结果表示当前时间与具有属性“DynamicInfo() 处理”的任何一个时间相匹配时，解码控制模块 214 将表示当前时间变为具有属性“DynamicInfo() 处理”的时间的消息、以及具有属性“DynamicInfo() 处理”的匹配时间（在下文中有时称为 DynamicInfo 时间）提供给播放器控制模块 212。此后，流程前进到步骤 S322。

[0792] 在步骤 S332，播放器控制模块 212 从解码控制模块 214 接收表示当前时间变为具有属性“DynamicInfo() 处理”的时间的消息以及 DynamicInfo 时间，并且识别出与

DynamicInfo 时间相匹配的、与 pts_change_point（图 10）配对的 DynamicInfo() 作为目标 DynamicInfo()。此后，流程前进到步骤 S323。

[0793] 在步骤 S323，播放器控制模块 212 将作为目标 DynamicInfo() 的 DynamicInfo()（图 13）中描述的输出属性提供给图形处理模块 219 或者音频输出模块 221。此后，流程前进到步骤 S324。

[0794] 在步骤 S324，图形处理模块 219 或者音频输出模块 221 与已经在步骤 S323 从播放器控制模块 212 提供的输出属性相对应地、开始控制视频数据或者音频数据的输出。此后，流程返回到步骤 S321。

[0795] 因此，与例如描述为输出属性（显示模式）的纵横比相对应地输出视频数据。作为选择，与例如被描述为输出属性的立体声模式或者双声道（双语）模式相对应地输出音频数据。

[0796] 接下来，将参考图 42 详细描述输出属性控制处理。

[0797] 图 42 示出了在图 26A 和图 26B 所示的剪辑信息文件“00003.CLP”中描述的一对 pts_change_point 和 DynamicInfo()（图 10）。

[0798] 如上所述，在图 26A 和图 26B 所示的剪辑信息文件“00003.CLP”中，视频流 stream#0 和音频流 stream#2（它们是用剪辑流文件“00003.PS”多路复用的三个基本流 stream#0 到 stream#2 中的第一基本流和第三基本流）的 number_of_DynamicInfo 分别为 2 和 3。因此，剪辑信息文件“00003.CLP”描述了用于剪辑流文件“00003.PS”中的第一视频流 stream#0 的两组 pts_change_point 和 DynamicInfo()、以及用于剪辑流文件“00003.PS”中的第三音频流 stream#2 的三组 pts_change_point 和 DynamicInfo()。

[0799] 图 42 所示的上部表格示出了剪辑流文件“00003.PS”中的第一视频流 stream#0 的两组 pts_change_point 和 DynamicInfo()。图 42 所示的下部表格示出了剪辑流文件“00003.PS”中的第三音频流 stream#2 的三组 pts_change_point 和 DynamicInfo()。

[0800] 除第一视频流 stream#0 的两组 pts_change_point 和 DynamicInfo() 之外，图 42 所示的上部表格还表示出第一视频 stream#0 的 stream_id（＝0xE0）、private_stream_id（＝0x00）、以及剪辑信息文件“00003.CLP”的 number_of_DynamicInfo（＝2）（如图 26A 和图 26B 所示）。同样地，除第三音频流 stream#2 的三组 pts_change_point 和 DynamicInfo() 之外，图 42 的下部表格还示出音频流 stream#2 的 stream_id（＝0xBD）、private_stream_id（＝0x00）、和剪辑信息文件“00003.CLP”中的 number_of_DynamicInfo（＝3）（如图 26A 和图 26B 所示）。

[0801] 在图 42 所示的上部表格中，视频流 stream#0 的两组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 是 90,000 且其 DynamicInfo() 的 display_aspect_ratio（图 13）是“4 ： 3”，第二组的 pts_change_point 是 54,090,000 且其 DynamicInfo() 的 display_aspect_ratio 是“16 ： 9”。

[0802] 在图 42 所示的下部表格中，音频流 stream#2 的三组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 是 90,000 且其 DynamicInfo() 的 channel_assignment（图 13）是“Dual（双声）”、第二组的 pts_change_point 是 27,090,000 且其 DynamicInfo() 的 channel_assignment 是“Stereo（立体声）”、第三组的 pts_change_point 是 32,490,000 且其 DynamicInfo() 的 channel_assignment 是“Dual（双声）”。

[0803] 现在，假定在图 30 所示的步骤 S125 处，已经将由值为 0xE0 的 stream_id 所标识的第一视频流 stream#0 以及由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 所标识的第三音频流 stream#2 确定为要从剪辑流文件“00003.PS”的流。

[0804] 在这种情况下，播放器控制模块 212 为由值为 0xE0 的 stream_id 所标识的视频流 stream#0 检查在图 42 所示的上部表格中的两组 pts_change_point 和 DynamicInfo()，并且为由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 所标识的音频流 stream#2 检查图 42 所示的下部表格中的三组 pts_change_point 和 DynamicInfo()，并识别处初始值。

[0805] 换句话说，在图 42 所示的上部表格中，由值为 0xE0 的 stream_id 所标识的视频流 stream#0 的两组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 是 90,000。90,000 与在与已经用其多路复用了视频流 stream#0 的剪辑流文件″ 00003.PS″ 相对应的、图 26A 和图 26B 所示的剪辑信息文件″ 00003.CLP″ 中的 presentation_start_time 描述的 90,000 相匹配，其中 presentation_start_time 表示剪辑流文件“00003.PS”的开始时间。

[0806] 同样地，在图 42 所示的下部表格中，由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 所标识的音频流 stream#2 的三组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 是 90,000。90,000 与在与已经用其多路复用了音频流 stream#2 的剪辑流文件″ 00003.PS″ 相对应的、图 26A 和图 26B 所示的剪辑信息文件″ 00003.CLP″ 中的 presentation_start_time 描述的 90,000 相匹配，其中 presentation_start_time 表示剪辑流文件“00003.PS”的开始时间。

[0807] 播放器控制模块 212 将与 presentation_start_time（其表示剪辑流文件″ 00003.PS″ 的开始时间）中描述的 90,000 相匹配的 pts_change_point 识别为初始值。因此，播放器控制模块 212 将图 42 所示上部表格中的两组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 和图 42 所示的下部表格中的三组 pts_change_point 和 DynamicInfo() 中的第一组的 pts_change_point 识别为初始值。

[0808] 在再现剪辑流文件“00003.PS”之前，播放器控制模块 212 指定与被识别为初始值的 pts_change_point 成对的 DynamicInfo() 相对应的基本流的输出属性（图 30 所示的步骤 S126）。

[0809] 对于图 42 所示的上部表格中、由值为 0xE0 的 stream_id 所标识的视频流 stream#0，与作为初始值的值为 90,000 的 pts_change_point 配对的 DynamicInfo() 的 display_aspect_ratio 为“4 ： 3”。在这种情况下，播放器控制模块 212 用表示 display_aspect_ratio 为“4 ： 3”的信息（即有关表示视频流 stream#0 是纵横比为 4 ： 3 的视频数据的输出属性的信息），来控制图形处理模块 219。

[0810] 对于图 42 所示的下部表格中、由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 所标识的音频流 stream#2，与作为初始值的值为 90,000 的 pts_change_point 配对的 DynamicInfo() 的 channel_assignment 为“双声 (Dual)”。在这种情况下，播放器控制模块 212 将表示 channel_assignment 是“Dual（双声）”的信息（即有关表示音频流 stream#2 是双声音频数据的输出属性的信息）提供给音频输出模块 221。

[0811] 在图 30 所示的步骤 S126 处，播放器控制模块 212 为作为初始值的 pts_change_

point 执行输出属性控制处理。

[0812] 此后，播放器控制模块 212 向解码控制模块 214 提供在图 42 所示的上部表格中、用于视频流 stream#0 的 90,000 和 54,090,000（它们是两个 pts_change_point），以及在图 42 所示的下部表格中、用于音频流 stream#2 的 90,000、27,090,000、和 32,490,000（它们是三个 pts_change_point 的时间）中除了作为初始值的 90,000 之外的 {27,090,000}、和 {32,490,000}，连同表示这些时间具有属性“DynamicInfo() 处理”的信息。（步骤 S320）

[0813] 解码控制模块 214 从播放器控制模块 212 接收具有属性“DynamicInfo() 处理”的时间 {27,090,000}、{32,490,000}、和 {54,090,000}。在开始再现视频流 stream#0 和音频流 stream#2（再现剪辑流文件“00003.PS”的第二 PlayList#1 中的 PlayItem#0）之后，解码控制模块开始监控由时间计数部分 214A 计数的当前时间。

[0814] 当当前时间与具有属性“DynamicInfo() 处理”的时间 {27,090,000}、{32,490,000}、和 {54,090,000} 之一相匹配时，解码控制模块 214 向播放器控制模块 212 提供 DynamicInfo 时间，其是具有属性“DynamicInfo() 处理”并且与当前时间相匹配的时间（步骤 S321）。

[0815] 当当前时间变为例如 27,090,000 时，解码控制模块 214 向播放器控制模块 212 提供作为 DynamicInfo 时间的 27,090,000，其与当前时间相匹配并且是具有属性“DynamicInfo() 处理”的时间之一。

[0816] 播放器控制模块 212 从解码控制模块 214 接收作为 DynamicInfo 时间的 27,090,000，并且从在图 42 所示的上部表格中用于视频 stream#0 的两个 pts_change_point 和图 42 所示的下部表格中用于音频 stream#2 的三个 pts_change_point 中检查与作为 DynamicInfo 时间的 27,090,000 相匹配的 pts_change_point，并且识别出与匹配 27,090,000 的 pts_change_point 配对的 DynamicInfo()（即图 42 所示的下部表格中用于音频流 stream#2 的第二 DynamicInfo()）作为目标 DynamicInfo()（步骤 S322）。

[0817] 当目标 DynamicInfo() 是视频流的 DynamicInfo() 时，播放器控制模块 212 向图形处理模块 219 提供在目标 DynamicInfo() 中描述的输出属性（步骤 S323）。当目标 DynamicInfo() 是音频流的 DynamicInfo() 时，播放器控制模块 212 向音频输出模块 221 提供在目标 DynamicInfo() 中描述的输出属性（步骤 S323）。

[0818] 当图形处理模块 219 已经从播放器控制模块 212 接收了输出属性时，图形处理模块 219 与该输出属性相对应地开始控制视频数据的输出（步骤 S324）。

[0819] 换句话说，图形处理模块 219 与由从例如播放器控制模块 212 收到的输出属性所表示的视频数据的纵横比 (display_aspect_ratio（图 13）) 以及连接到图 1 所示的视频输出端 120 的视频输出设备的纵横比相对应，转换输出到视频输出模块 220 的视频数据的纵横比。

[0820] 具体而言，当视频输出设备的纵横比例如为 16 ： 9 而且由输出属性所代表的视频数据的纵横比是 4 ： 3 时，图形处理模块 219 在水平方向为输出到视频输出模块 220 的视频数据执行挤压处理并且使视频数据的左右端变黑。当视频输出设备的纵横比例如为 4 ： 3 而且由输出属性所代表的视频数据的纵横比是 16 ： 9 时，图形处理模块 219 在垂直方向为输出到视频输出模块 220 的视频数据执行挤压处理并且使视频数据的上下端变黑。当视频输出设备的纵横比和作为输出属性的视频数据的纵横比相同时，例如为 4 ： 3 或者 16 ： 9

时，图形处理模块 219 将视频数据输出到视频输出模块 220，而不为该视频数据执行挤压处理。

[0821] 利用在图 42 所示的上部表格中、由值为 0xE0 的 stream_id 所标识的视频流 stream#0 的两组 pts_change_point 和 DynamicInfo()，在时间 54,090,000 之前，在作为视频流 stream#0 的再现开始时间的时间 90,000 之后，获得具有纵横比 4 ： 3 的视频数据。在时间 54,090,000 之后，获得具有纵横比 16 ： 9 的视频数据。

[0822] 因此，假定连接到图 1 所示的视频输出端 120 的视频输出设备的纵横比为 4 ： 3，则在时间 90,000 之后且在时间 54,090,000 之前，图形处理模块 219 将从视频流 stream#0 获得的具有纵横比 4 ： 3 的视频数据提供给纵横比为 4 ： 3 的视频输出设备。视频输出设备显示所收到的视频数据。

[0823] 在时间 54,090,000 之后，图形处理模块 219 在垂直方向为具有纵横比 16 ： 9 的视频数据执行挤压处理并使该视频数据的上下端变黑，以便将纵横比 16 ： 9 的视频数据转换为纵横比 4 ： 3 的视频信号。将转换后的视频信号提供给视频输出设备。视频输出设备显示转换后的视频数据。

[0824] 当音频输出模块 221 从播放器控制模块 212 接收到输出属性时，音频输出模块 221 与该输出属性相对应地开始控制音频数据的输出（步骤 S324）。

[0825] 换句话说，与由从播放器控制模块 212 收到的输出属性所表示的音频数据的声道分配 (channel_assignment（图 13)) 相对应地、并且与通过用户利用遥控器操作的输入接口 115（图 1）从播放器控制模块 212 提供的音频输出模式相对应地，音频输出模块 221 处理从音频解码器控制模块 217 收到的音频数据，并且将处理后的音频数据输出到音频输出端 121（图 1）。

[0826] 具体而言，当由输出属性所代表的用于音频数据的声道分配是双声（双语）模式（其中，左声道是“主音频”数据而右声道是“子音频”数据）时，音频输出模块 221 与从播放器控制模块 212 提供的音频输出模式相对应地处理从音频解码器控制模块 217 提供的音频数据，并且将处理后的音频数据输出到音频输出端 121。

[0827] 换句话说，如果已经将“主声音”指定为音频输出模式的话，则音频输出模块 221 将从音频解码器控制模块 217 收到的音频数据的左声道复制为音频数据的右声道，并且将该音频数据（“主音频”数据）的左右声道输出到音频输出端 121。如果已经将“子音频”指定为音频输出模式的话，则音频输出模块 221 将从音频解码器控制模块 217 收到的音频数据的右声道复制为左声道，并且将该左右声道（″ 子音频 ″ 数据）输出到音频输出端 121。如果已经将“主和子音频”二者指定为音频输出模式，则音频输出模块 221 直接将从音频解码器控制模块 217 收到的音频数据输出到音频输出端 121。

[0828] 如果由输出属性所表示的音频数据的声道分配例如是立体声模式，则音频输出模块 221 直接将从音频解码器控制模块 217 收到的音频数据输出到音频输出端 121 而不考虑已经指定了什么音频输出模式。

[0829] 利用图 42 所示的下部表格中、由值为 0xBD 的 stream_id 和值为 0x00 的 private_stream_id 标识的音频流 stream#2 的三组 pts_change_point 和 DynamicInfo()，在作为再现开始时间的时间 90,000 之后且在时间 27,090,000 之前，从音频流 stream#2 中获得双声道音频数据。此外，在时间 27,090,000 之后且在时间 32,490,000 之前，从音频流 stream#2

获得立体声音频数据。此外，在时间 32,490,000 之后，从音频流 stream#2 获得双声道音频数据。

[0830] 因此，当已经将“主音频”指定为音频输出模式时，音频输出模块 221 将在时间 90,000 之后且在时间 27,090,000 之前、从音频流 stream#2 获得的双声音频数据的左声道的音频数据复制为该音频数据的右声道。将音频数据的左声道和右声道输出到音频输出端 121。

[0831] 将在时间 27,090,000 之后且在时间 32,490,000 之前、从音频流 stream#2 获得的立体声音频数据输出到音频输出端 121。

[0832] 将时间 32,490,000 之后、从音频流 stream#2 获得的双声道音频数据的左声道复制为音频数据的右声道。将音频数据的左声道和右声道输出到音频输出端 121。

[0833] 如上所述，在输出属性控制处理中，确定正被再现的基本流的再现时间是否与对应于剪辑信息文件 Clip()（图 10）的 pts_change_point 相匹配，其中该剪辑信息文件 Clip() 包括 n 组 pts_change_point 和 DynamicInfo()，其中 pts_change_point 表示用剪辑流文件多路复用的每个基本流的再现时间，而 DynamicInfo() 表示基本流的输出属性（其中 n 是 0 或者更大的任何整数）。当正被再现的基本流的再现时间与 pts_change_point 相匹配时，识别出与 pts_change_point 配对的 DynamicInfo()。与在 DynamicInfo() 中描述的输出属性相对应地控制正被再现的基本流的输出。因此，可以与基本流的再现时间和输出属性相对应地控制基本流的输出。

[0834] 【子标题显示控制处理】

[0835] 接下来，将参考图 43 所示的流程图，描述控制与子标题流相对应的子标题数据的显示的子标题显示控制处理。

[0836] 当开始 PlayList()（图 5）（其 PlayList()）的再现时，在步骤 S341，播放器控制模块 212 初始化用于图形处理模块 219 的子标题数据显示模式。换句话说，播放器控制模块 212 控制该图形处理模块 219，以将子标题数据显示模式改变为缺省显示模式。在步骤 S341 执行的显示模式的初始化对应于在图 30 所示的步骤 S127 执行的显示模式的初始化。

[0837] 在步骤 S341 之后，流程前进到步骤 S342。在步骤 S342，播放器控制模块 212 确定用户是否已经通过遥控器向输入接口 115 输入了新子标题数据显示模式命令。

[0838] 当步骤 S342 的确定结果表示已经输入了新显示模式命令时，流程前进到步骤 S343。在步骤 S343，播放器控制模块 212 确定是否正在再现子标题流（与此对应的子标题数据）。

[0839] 当步骤 S343 的确定结果表示没有在再现子标题流时，流程返回到步骤 S342。

[0840] 相反，当步骤 S343 的确定结果表示正在再现子标题流时，流程前进到步骤 S345。在步骤 5345，播放器控制模块 212 确定新显示模式命令是否为缺省显示模式命令。当步骤 S343 的确定结果表示新显示模式命令是缺省显示模式命令时，流程返回到步骤 S341。在步骤 S341，如上所述，播放器控制模块 212 控制该图形处理模块 219，以便将子标题数据显示模式改变为缺省显示模式。

[0841] 相反，当步骤 S345 的确定结果表示新显示模式命令不是缺省显示模式命令时，即新显示模式命令是例如子标题数据放大命令、子标题数据缩小命令、或者亮度增加命令的非缺省显示模式命令时，流程前进到步骤 S346。在步骤 S346，播放器控制模块 212 从与用

其多路复用了正被再现的子标题流的剪辑流文件相对应的剪辑信息文件Clip()(图10)的StaticInfo()(图12)中、获得正被再现的子标题流StaticInfo()。此后,流程前进到步骤S347。

[0842] 在步骤S347,播放器控制模块212确定步骤S346获得的StaticInfo()中的configurable_flag。

[0843] 当步骤S347的确定结果表示configurable_flag为0(表示不允许改变子标题数据显示模式)时,流程前进到步骤S348。在步骤S348,播放器控制模块212控制图形处理模块219,以便将表示不能改变子标题数据显示模式的消息叠加到输出的视频数据上。此后,流程返回到步骤S342。在步骤S342,显示错误信息。

[0844] 相反,当步骤S347的确定结果表示configurable_flag为1(其表示允许改变子标题数据显示模式)时,流程前进到步骤S349。在步骤S349,播放器控制模块212将已经由用户通过输入接口115从遥控器输入的新显示模式命令提供给图形处理模块219。此后,流程前进到步骤S350。

[0845] 在步骤S350,图形处理模块219与已经在步骤S349从播放器控制模块212提供的显示模式命令相对应地,为从子标题解码器控制模块218提供的子标题数据执行放大处理、缩小处理、或者亮度改变处理。此后,流程返回到步骤S342。因此,以与已经由用户通过遥控器输入的显示模式命令相对应的显示大小、显示位置、或者显示颜色来显示该子标题数据。

[0846] 相反,当步骤S342的确定结果表示还没有输入新显示模式命令时,流程前进到步骤S351。在步骤S351,播放器控制模块212确定是否已经如图31所述改变了PlayItem()。当步骤S342的确定结果表示还没有改变PlayItem()时,流程返回到步骤S342。

[0847] 相反,当步骤S351的确定结果表示已经改变了PlayItem()时,流程返回到步骤S341。在步骤S341,如上所述,播放器控制模块212控制该图形处理模块219,以便将子标题数据显示模式改变为缺省显示模式。换句话说,当已经改变了PlayItem()时,子标题数据显示模式恢复为缺省显示模式。

[0848] 如上所述,只有当子标题流的configurable_flag为1(其表示允许改变显示模式)时,可以与用户通过遥控器输入的显示模式命令相对应地改变子标题流的子标题数据显示模式。

[0849] 因此,例如在图26A和图26B所示的剪辑信息文件“00001.CLP”中,因为子标题流stream#2(其是用剪辑流文件“00001.PS”多路复用的四个基本流中的第三个基本流)的configurable_flag是表示不允许改变显示模式的0,所以当正在显示子标题流stream#2时,即使用户操作遥控器以改变子标题显示模式,显示模式也不被改变。

[0850] 相反,因为子标题流stream#3(其是用剪辑流文件“00001.PS”多路复用的四个基本流中的第四个基本流)的configurable_flag是表示允许改变显示模式的1,所以当正在显示子标题流stream#3时,当用户操作遥控器以改变子标题显示模式时,子标题的显示大小被改变。

[0851] 现在,假定与图25所示的第一PlayList#1的第一PlayItem#1相对应地再现剪辑流文件“00001.PS”。此外,在图26A和图26B所示的剪辑信息文件“00001.CLP”中,假定用剪辑流文件“00001.PS”多路复用的四个基本流中的第三和第四基本流是子标题流,而且正

在再现第三和第四子标题流 stream#2 和 stream#3 中的第三子标题流 stream#2。

[0852] 当用户操作遥控器以输入子标题显示模式命令（步骤 S342）时，显示模式命令从输入接口 115（图 1）提供到播放器控制模块 212。当播放器控制模块 212 收到显示模式命令时，播放器控制模块 212 在剪辑信息文件中搜索与正被再现的子标题流相对应的 StaticInfo()（图 10）（步骤 S346）。

[0853] 换句话说，正被再现的子标题流是用剪辑流文件“00001.PS”多路复用的第三子标题流 stream#2。播放器控制模块 212 在对应的剪辑信息文件“00001.CLP”中搜索第三子标题流 stream#2 的 StaticInfo()。

[0854] 此外，播放器控制模块 212 确定在图 26A 和图 26B 所示的第三子标题流 stream#2 的 StaticInfo() 中描述的 configurable_flag（其为 0）（步骤 S347）。因此，播放器控制模块 212 识别出不允许改变第三子标题流 stream#2 的显示模式。

[0855] 在这种情况下，播放器控制模块 212 确定正被再现的子标题流（与此对应的子标题数据）不对应于放大和缩小模式，并且控制该图形处理模块 219 来生成对应的错误信息（步骤 S348），并且将视频数据叠加在该错误信息上，并输出叠加后的视频数据。

[0856] 在正在再现用剪辑流文件“00001.PS”多路复用的四个基本流中的第三和第四子标题流 stream#2 和 stream#3 当中的第四子标题流 stream#3 期间，当播放器控制模块 212 收到已经由用户通过遥控器输入的显示模式命令时，播放器控制模块 212 在对应的剪辑信息文件“00001.CLP”中搜索第四子标题流 stream#3 的 StaticInfo()。

[0857] 播放器控制模块 212 确定在图 26A 和图 26B 所示的第四子标题流 stream#3 的 StaticInfo() 中描述的 configurable_flag（其为 1）（步骤 S347）。因此，播放器控制模块 212 识别出已经允许改变第四子标题流 stream#3 的显示模式。

[0858] 在这种情况下，播放器控制模块 212 确定正被再现的子标题流（与此对应的子标题数据）对应于放大模式或者缩小模式，并且将已经由用户通过遥控器输入的显示模式命令提供给图形处理模块 219（步骤 S349）。

[0859] 因此，图形处理模块 219 与从播放器控制模块 212 收到的显示模式命令相对应地例如放大或者缩小从子标题解码器控制模块 218 收到的子标题数据，用从视频解码器控制模块 212 提供的视频数据叠加得到的子标题数据，并且输出叠加后的数据。

[0860] 当播放器控制模块 212 开始 PlayList() 中的第一个 PlayItem() 的再现时，播放器控制模块 212 初始化用于图形处理模块 219 的子标题数据显示模式（步骤 S341）。换句话说，播放器控制模块 212 控制该图形处理模块 219，以将子标题数据显示模式改变为缺省显示模式。

[0861] 当改变了 PlayItem() 时，播放器控制模块 212 初始化用于图形处理模块 219 的子标题数据显示模式（步骤 S341 和 S351）。

[0862] 当改变 PlayItem() 时，播放器控制模块 212 检查与新再现的 PlayItem() 相对应的、要再现的新子标题流的 configurable_flag。当 configurable_flag 为 0 时，播放器控制模块 212 初始化用于图形处理模块 219 的子标题数据显示模式。当 configurable_flag 为 1 时，播放器控制模块 212 使图形处理模块 219 保持用于 PlayItem() 的显示模式。

[0863] 在图 43 所示的子标题显示控制处理中，当由用户通过遥控器输入了新显示模式命令时，将新显示模式命令提供给图形处理模块 219（步骤 S349）。显示模式命令可以存储

在例如组成存储器 113(图 1)的非易失性存储器中。可以将存储在非易失性存储器中的显示模式命令提供给图形处理模块 219。

[0864] 假定用户已经设置的显示模式命令作为图 1 所示的盘设备的初始设置被存储在非易失性存储器中,则当用户利用遥控器输入新显示模式命令时,存储在非易失性存储器中的显示模式命令被替换为该新显示模式命令,并且将存储在非易失性存储器中的新显示模式命令提供给图形处理模块 219。在这种情况下,因为非易失性存储器存储了在上次再现完成时已经设置的显示模式命令,所以当再现下一个 PlayList() 时,用该显示模式命令来显示该子标题数据,而不需要通过遥控器输入显示命令。

[0865] 在这种情况下,假定在非易失性存储器中存储的显示模式命令包括例如放大或者缩小子标题流的放大率或者缩小率。

[0866] 如上所述,在子标题显示控制处理中,与用于子标题数据(其在当正在再现包括在剪辑信息文件 Clip()(图 10)中的基本流期间未发生改变)的 StaticInfo() 中包括的 configurable_flag 相对应地,确定是否允许从缺省显示模式开始改变子标题数据显示模式。当允许改变正被再现的子标题数据的缺省显示模式时,为子标题数据执行例如放大处理、缩小处理、或者颜色改变处理的显示处理。因此,可以控制子标题数据显示模式。

[0867] 【捕获控制处理】

[0868] 接下来,将参考图 44 所示的流程图描述控制对与视频流相对应的视频数据的捕获的捕获控制处理。图 44 还示出了这样的流程图,其描述了二次使用已经在捕获控制处理中捕获的视频数据的背景 / 屏幕保护程序处理。

[0869] 当由用户通过输入接口 115(图 1)从遥控器将视频数据捕获命令输入到播放器控制模块 212 时,开始捕获控制处理。

[0870] 换句话说,在捕获控制处理中,在步骤 S371 处,播放器控制模块 212 确定是否正在再现视频流。当步骤 S371 的确定结果表示没有正在再现视频流时,播放器控制模块 212 完成该捕获控制处理。

[0871] 相反,当步骤 S371 的确定结果表示正在再现视频流时,流程前进到步骤 S372。播放器控制模块 212 从与正被再现的视频流相对应的 PlayList()(图 5)获得 capture_enable_flag_PlayList,并且从与正被再现的视频流相对应的剪辑信息文件 Clip()(图 10)获得 capture_enable_flag_Clip。

[0872] 如图 5 所示,PlayList() 中的 capture_enable_flag_PlayList 表示是否允许二次使用对应于与 PlayList() 相对应地再现的视频流的视频数据(包括在 PlayList() 中的视频数据)。另一方面,如图 10 所示,剪辑信息文件 Clip() 中的 capture_enable_flag_Clip 表示是否允许二次使用对应于存储在与剪辑信息文件 Clip() 相对应的剪辑流文件中的视频流的视频数据。

[0873] 在步骤 S372 之后,流程前进到步骤 S373。与已经在步骤 S373 获得的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 相对应地,播放器控制模块 212 确定是否允许捕获当从输入接口 115(图 1)输入捕获命令时正在再现的视频数据中的画面。

[0874] 当步骤 S373 的确定结果表示不允许捕获当从输入接口 115 输入捕获命令时正被再现的视频数据的画面时,即在步骤 S373 获得的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 中的至少一个为 0(其表示不允许二次使用视频数据)时,流程

前进到步骤 S374。在步骤 S374，播放器控制模块 212 控制图形处理模块 219，以便将视频数据叠加在表示不允许捕获视频数据的错误消息上，并且完成该捕获控制处理。结果，显示该错误消息。

[0875] 相反，当步骤 S373 的确定结果表示允许捕获当从输入接口 115 输入捕获命令时正被再现的视频数据画面时，即已经在步骤 S373 获得的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 二者均为 1（其表示允许二次使用视频数据）时，流程前进到步骤 S375。在步骤 S375，播放器控制模块 212 向图形处理模块 219 提供用于当从输入接口 115 输入捕获命令时正被再现的视频数据的捕获命令。此后，流程前进到步骤 S376。

[0876] 在步骤 S376，与从播放器控制模块 212 收到的捕获命令相对应、图形处理模块 219 从视频解码器控制模块 216 捕获视频数据画面，将该画面存储在存储器 113（图 1）中，并完成该捕获控制处理。当 capture_enable_flag 包括多个位并且指定了它们的使用条件时，此时，执行对应的操作。换句话说，当捕获画面的大小受限时，捕获其大小被缩小的画面。当所使用的应用受限时，还记录表示该限制的标志。

[0877] 如上所述，在捕获控制处理中，与当用户输入捕获命令时正被再现的视频流相对应的 PlayList()（图 5）和剪辑信息文件 Clip()（图 10）的 capture_enable_fag_PlayList 和 capture_enable_flag_Clip 被进行与 (AND) 操作。当与操作的结果是 1，即 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 二者均为 1 时（其表示允许二次使用视频数据），确定可以二次使用视频数据。因此，捕获该视频数据。

[0878] 当与图 25 所示的第一 PlayList#0 的第一 PlayItem#0 相对应地再现视频流时，即正在再现用剪辑流文件“00001.PS”多路复用的视频流时，如果用户输入了捕获命令，则因为第一 PlayList#0 的 capture_enable_flag_PlayList 为 1，且如图 26A 和图 26B 所示、与由第一 PlayItem#0 再现的剪辑流文件“00001.PS”相对应的剪辑信息文件“00001.CLP”的 capture_enable_flag_Clip 为 1，所以确定可以二次使用正被再现的视频数据（与用剪辑流文件“00001.PS”多路复用的视频流相对应的视频数据），并且捕获该视频数据。

[0879] 在与图 25 所示的第一 PlayList#0 的第二 PlayItem#1 相对应地再现视频流时，即正在再现用剪辑流文件“00002.PS”多路复用的视频流时，当用户输入了捕获命令时，因为第一 PlayList#0 的 capture_enable_flag_PlayList 为 1，且如图 26A 和图 26B 所示、与对应于第二 PlayItem#1 表示的剪辑流文件“00002.PS”相对应的剪辑信息文件“00002.CLP”的 capture_enable_flag_Clip 为 0，所以确定不能二次使用正被再现的视频数据（与用剪辑流文件“00002.PS”多路复用的视频流相对应的视频数据），并且不捕获该视频数据。

[0880] 在与图 25 所示的第二 PlayList#1 中的 PlayItem#0 相对应地再现视频流时，即正在再现用剪辑流文件“00003.PS”多路复用的视频流时，当用户输入了捕获命令时，因为第二 PlayList#1 的 capture_enable_flag_PlayList 为 0，且如图 26A 和图 26B 所示、与对应于第二 PlayList#1 中的 PlayItem#0 再现的剪辑流文件“00003.PS”相对应的剪辑信息文件“00003.CLP”的 capture_enable_flag_Clip 为 1，所以确定不允许二次使用正被再现的视频数据（与用剪辑流文件“00003.PS”多路复用的视频流相对应的视频数据）。因此，不捕获该视频数据。

[0881] 在这种情况下，当已经检查出第二 PlayList#1 的 capture_enable_flag_PlayList 为 0 时，可以确定不允许二次使用视频数据。因此，可以省略对如图 26A 和图 26B

所示、与对应于第二 PlayList#1 中的 PlayItem#0 再现的剪辑流文件“00003.PS”相对应的剪辑信息文件“00003.CLP”中的 capture_enable_flag_Clip。

[0882] 在捕获控制过处理中捕获并存储在存储器 113 中的画面可以在背景 / 屏幕保护程序处理中被二次使用。

[0883] 例如在播放器控制模块 212 正在操作、但是没有正在再现基本流、即盘 101 还未被插入到盘驱动器 102(图 1) 中或者早已再现了基本流期间，执行该背景 / 屏幕保护程序处理。

[0884] 在背景 / 屏幕保护程序处理中，在步骤 S381，播放器控制模块 212 控制图形处理模块 219，以便显示在捕获控制处理中已经存储在存储器 113 中的画面。图形处理模块 219 在播放器控制模块 212 的控制下显示在捕获控制处理中已经存储在存储器 113 中的画面。

[0885] 当图形处理模块 219 将存储在存储器 113 中的画面显示为静止画面时，实现了所谓的壁纸 (背景)。当正在放大、缩小、并移动画面的同时显示画面时，实现了屏幕保护程序。可以由另一个独立的应用而不是播放器控制模块 212 来执行显示在捕获控制处理中存储在存储器 113 中的画面的背景 / 屏幕保护程序处理。

[0886] 当将表示限制的标志添加到存储在存储器 133 中的画面时，与该标志相对应地限制所显示的画面。

[0887] 如上所述，与例如大于视频存取单元的 PlayList() 或者 PlayItem() 相对应地获得表示是否允许二次使用正被再现的视频数据的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip。 与 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 相对应，确定是否允许二次使用正被再现的视频数据。当确定结果表示允许二次使用正被再现的视频数据时，捕获正被再现的视频数据，并且执行使用所捕获的视频数据的背景 / 屏幕保护程序处理。因此，可以控制视频数据的二次使用。

[0888] 在图 44 所示的捕获控制处理中，PlayList()(图 5) 包括 capture_enable_flag_PlayList，而且与由 PlayItem() 再现的剪辑流文件相对应的剪辑信息文件 (图 10) 包括 capture_enable_flag_Clip。利用 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 二者，确定是否允许二次使用视频数据。作为选择，当 PlayList()(图 5) 包括 capture_enable_flag_PlayList，而且与由 PlayItem() 再现的剪辑流文件相对应的剪辑信息文件 Clip()(图 10) 包括 capture_enable_flag_Clip 时，利用 capture_enable_flag_PlayList 或者 capture_enable_flag_Clip，可以确定是否允许二次使用视频数据。

[0889] 在图 44 所示的捕获控制处理中，在步骤 S376，与从播放器控制模块 212 收到的捕获命令相对应，图形处理模块 219 仅仅从视频解码器控制模块 216 捕获一个视频数据画面。作为选择，图形处理模块 219 可以从视频解码器控制模块 216 捕获多个画面。换句话说，可以捕获视频解码器控制模块 216 输出的多个画面 (作为活动画面的一系列多个画面)。在这种情况下，可以预先指定一次捕获的画面的数目。作为选择，可以为表示一次可以捕获的画面数目的信息扩展 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 中的位。

[0890] 在上述情况下，在 PlayList() 和剪辑信息文件 Clip() 中描述了表示是否允许二次使用视频数据的使用许可信息，即 capture_enable_flag_PlayList 和 capture_enable_flag_Clip。利用该使用许可信息，确定是否允许二次使用与 PlayList() 相对应地再现的

整个视频数据以及与用与剪辑信息文件 Clip() 相对应的剪辑流文件多路复用的视频流相对应的整个视频数据。使用许可信息可以描述任何单元的视频数据。利用该使用许可信息，可以确定是否允许二次使用在任何单元中的视频数据。

[0891] 图 45 示出了包括使用许可信息的 private_stream2_PES_payload() 的语法。图 46 示出了包括使用许可信息的 au_information() 的语法。

[0892] 除了作为使用许可信息的 capture_enable_flag_ps2 紧靠 video_stream_id 之前之外，图 45 所示的 private_stream2_PES_payload() 与图 23 所示的相同。同样地，除了作为使用许可信息的 capture_enable_flag_AU 紧靠 pic_struct_copy 之前之外，图 46 所示的 au_information() 与图 24 所示的相同。

[0893] 在图 45 所示的 private_stream2_PES_payload() 中包括的 capture_enable_flag_ps2 表示是否允许二次使用在包括该 private_stream2_PES_payload() 的 private_stream_2 的 PES_packet() 之后、并在下一个 private_straem_2 的 PES_packet() 之前的视频流的视频数据。因此，利用包括在图 45 所示的 private_stream2_PES_payload() 中的 capture_enable_flag_ps2，可以确定是否允许二次使用在特定可解码起始点之后并在下一个可解码起始点之前的视频数据。

[0894] 此外，在图 46 所示的 au_information() 中包括的 capture_enable_flag_AU 表示是否允许二次使用在与 capture_enable_flag_AU 相对应的每个视频存取单元中的视频数据。因此，利用在图 46 所示的 au_information() 中包括的 capture_enable_flag_AU，可以确定是否允许二次使用在每个视频存取单元（即每个画面）中的视频数据。

[0895] 可以同时使用作为 PlayList() 的使用许可信息的 capture_enable_flag_PlayList（图 5）、作为剪辑信息文件 Clip() 的使用许可信息的 capture_enable_flag_Clip（图 10）、作为 private_stream2_PES_payload() 的使用许可信息的 capture_enable_flag_ps2（图 45）、和作为 au_information() 的使用许可信息的 capture_enable_flag_AU（图 46）中的至少两个。在这种情况下，利用作为使用许可信息的这些值中的至少两个的与操作的结果，可以确定是否允许二次使用视频数据的画面。

[0896] 如图 36 所示的步骤 S211 所述，视频读取功能部分 233 在存储在缓冲器 215A 中的节目流中搜索包括图 23 或者图 34 所示的 private_stream2_PES_payload 的 private_stream_2 的 PES_packet()，其中该 private_stream2_PES_payload 包括图 46 所示的 au_information()。因此，当使用图 45 所示的 private_stream2_PES_payload()（其包括 capture_enable_flag_ps2）和图 45 所示的 au_information()（其包括 capture_enable_flag_AU）时，播放器控制模块 212 需要向视频读取功能部分 233 询问 capture_enable_flag_ps2 和 capture_enable_flag_AU，以确定是否允许二次使用视频数据。

[0897] 根据上述实施例，由软件执行处理序列。作为选择，这些处理可以由专用硬件执行。

[0898] 此外，根据该实施例，视频解码器 116（图 1）是硬件解码器。作为选择，视频解码器 116 可以是软件解码器。这个关系适用于音频解码器 117（图 1）。

[0899] 此外，根据该实施例，子标题解码器是软件解码器。作为选择，子标题解码器 116 可以是硬件解码器。

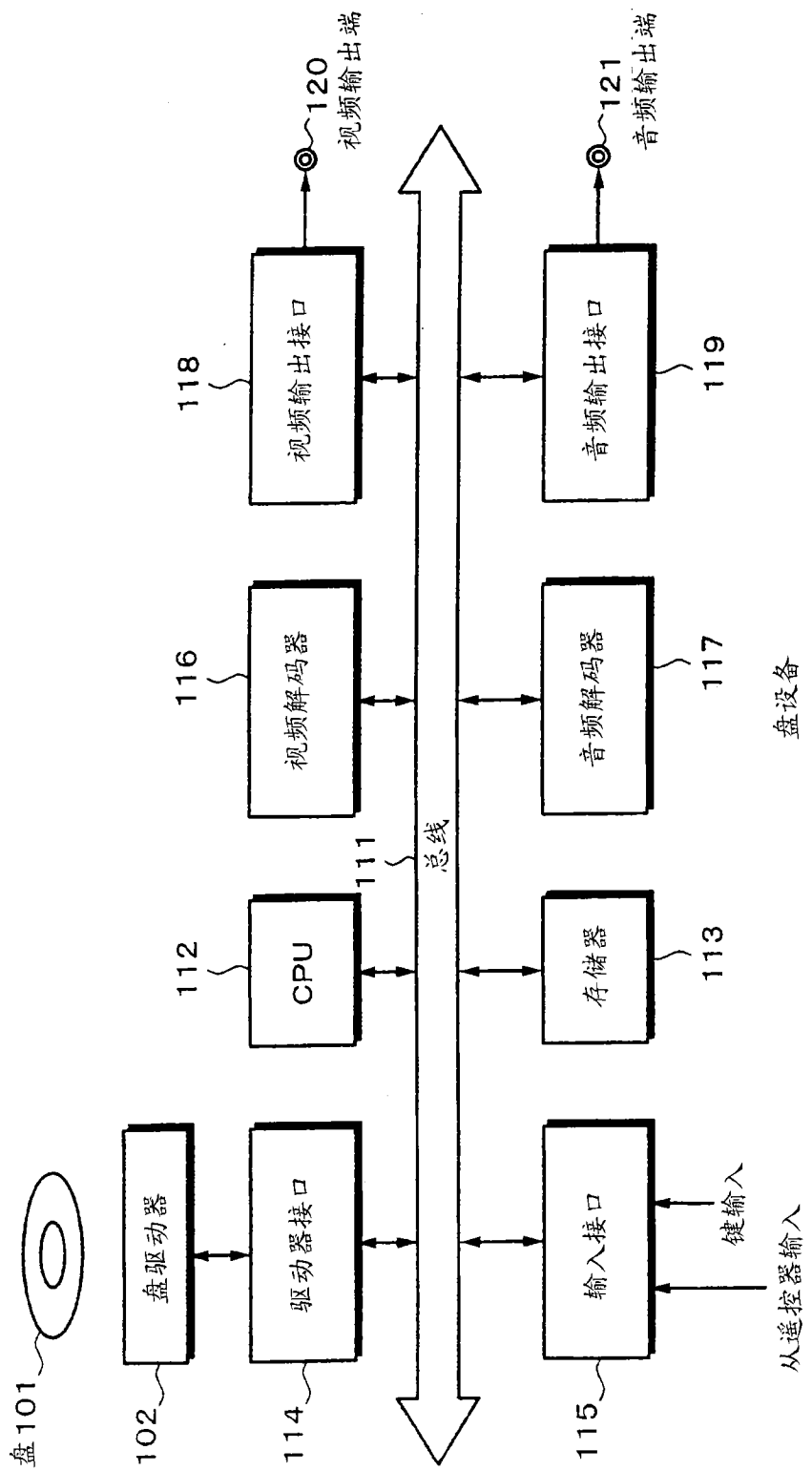

盘101
盘驱动器
102
驱动器接口
114
CPU
112
视频解码器
116
视频输出接口
118
120
视频输出端
总线
111
输入接口
115
键输入
从遥控器输入
存储器
113
音频解码器
117
音频输出接口
119
121
音频输出端
盘设备

图 1

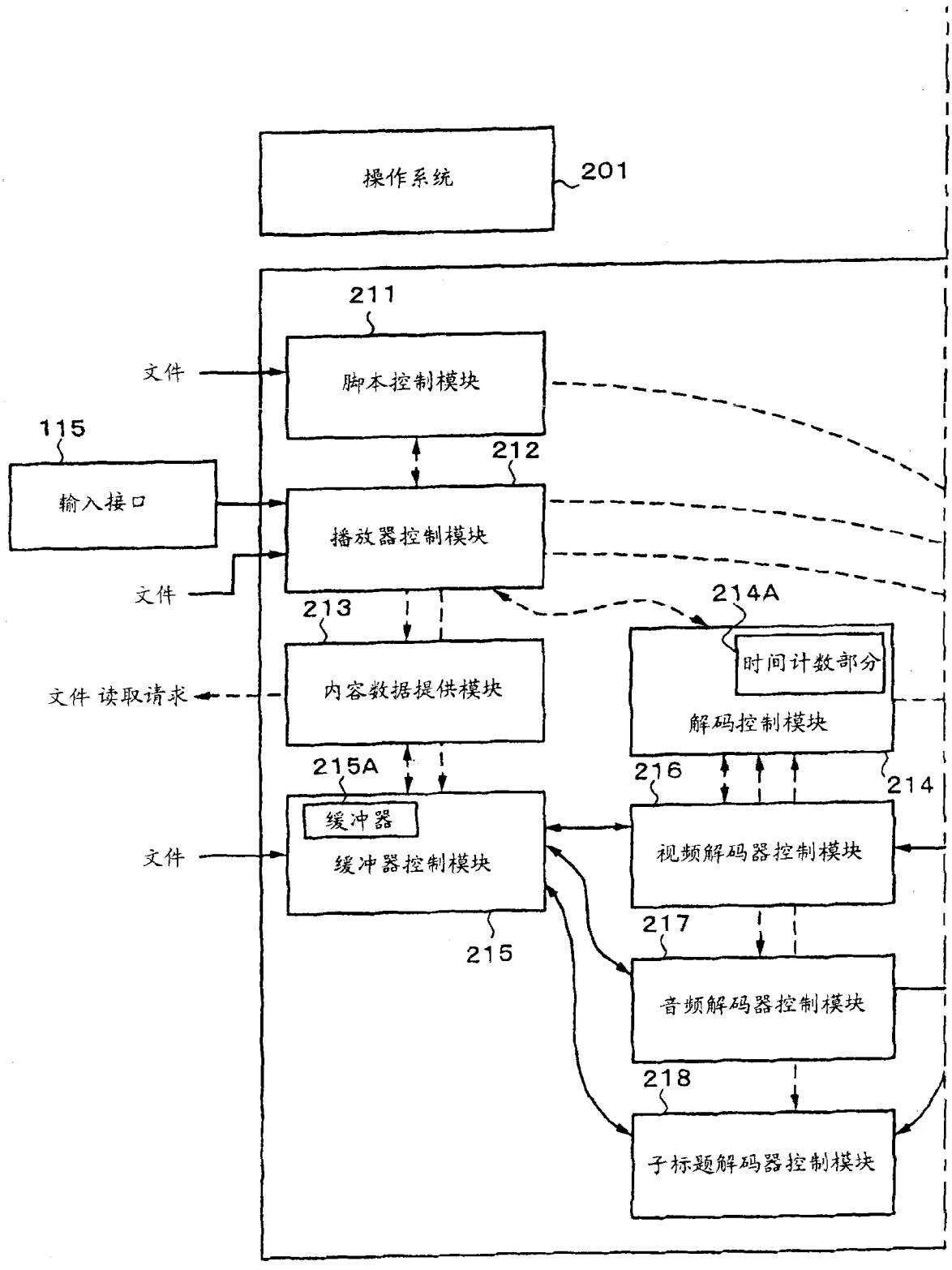

操作系统
201
211
脚本控制模块
文件
115
输入接口
212
播放器控制模块
文件
213
内容数据提供模块
文件 读取请求
214A
时间计数部分
解码控制模块
214
215A
缓冲器
缓冲器控制模块
文件
215
216
视频解码器控制模块
217
音频解码器控制模块
218
子标题解码器控制模块

图 2A

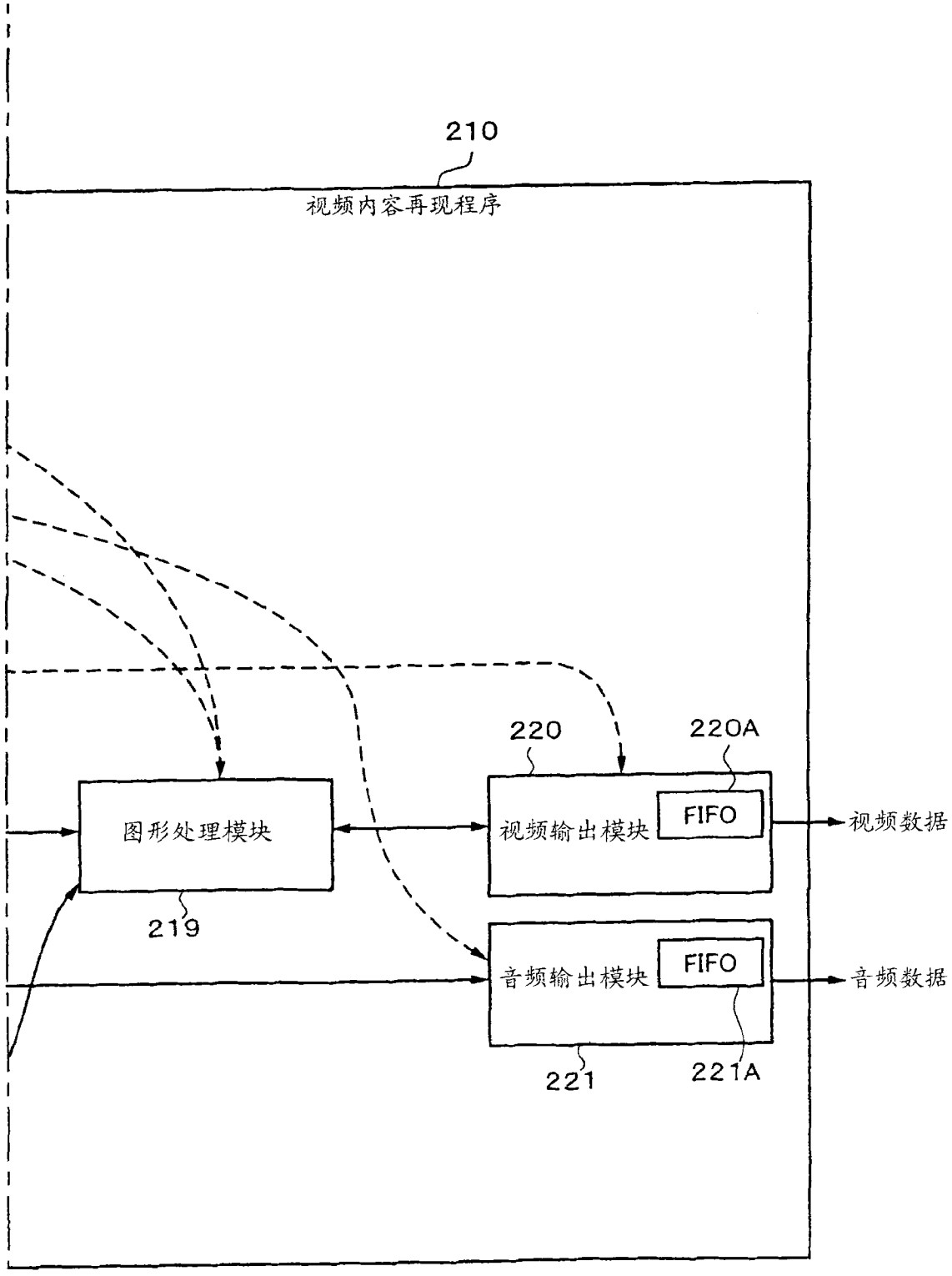

210
视频内容再现程序
220
220A
图形处理模块
视频输出模块
FIFO
视频数据
219
音频输出模块
FIFO
音频数据
221
221A

图 2B

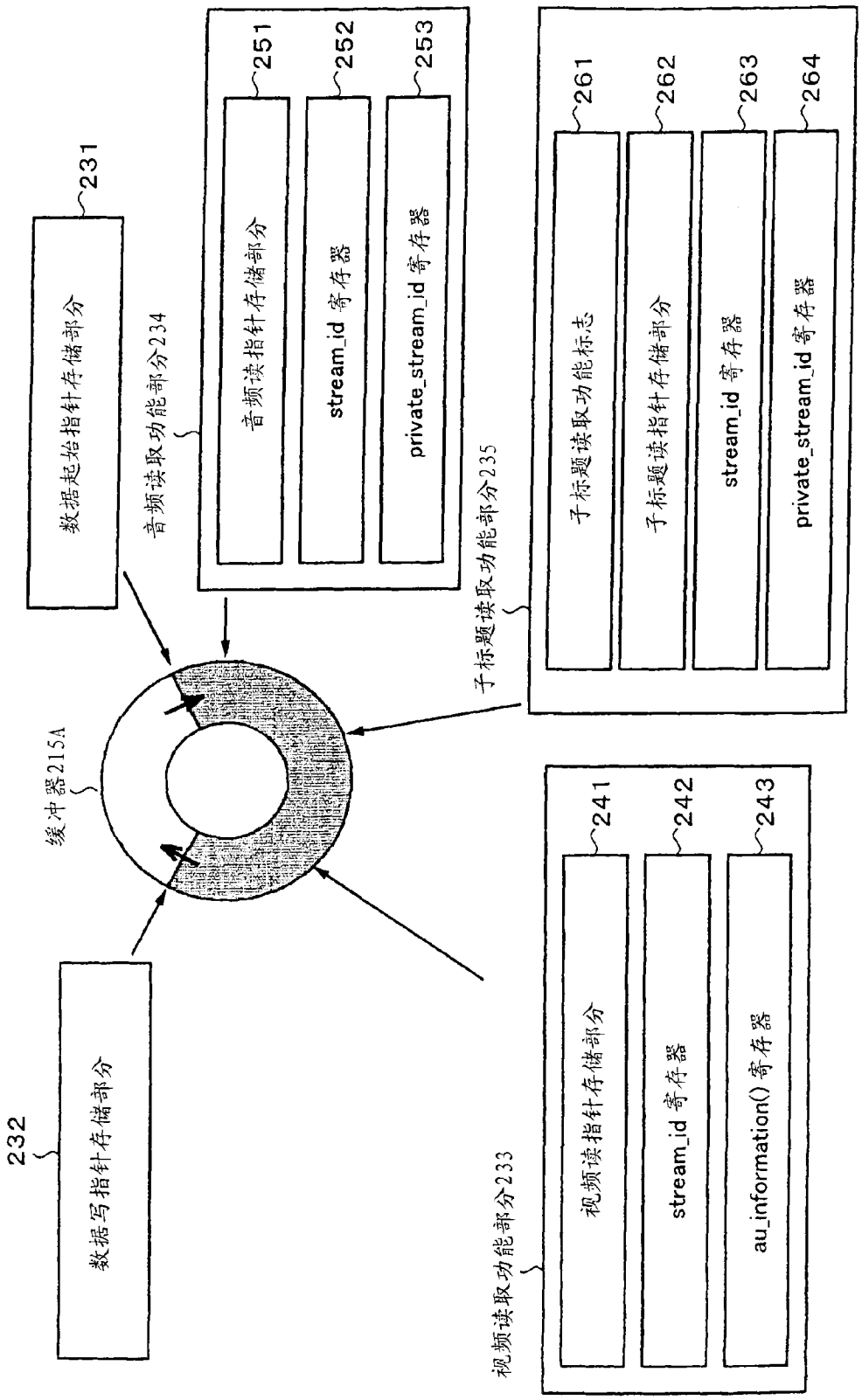

数据起始指针存储部分
231
音频读取功能部分234
音频读指针存储部分
251
stream_id 寄存器
252
private_stream_id 寄存器
253
子标题读取功能部分235
子标题读取功能标志
261
子标题读指针存储部分
262
stream_id 寄存器
263
private_stream_id 寄存器
264
缓冲器215A
数据写指针存储部分
232
视频读取功能部分233
视频读指针存储部分
241
stream_id 寄存器
242
au_information() 寄存器
243

图 3

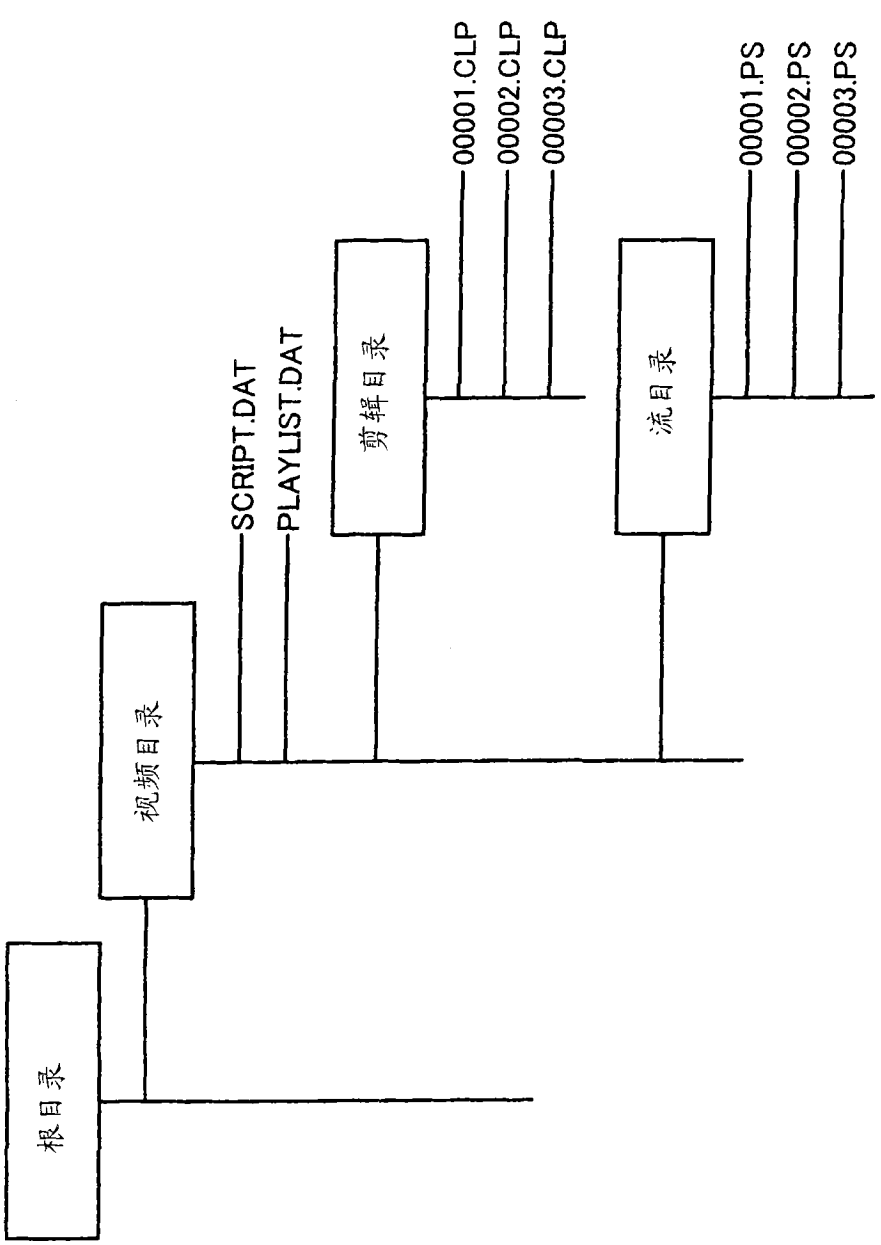

根目录
视频目录
SCRIPT.DAT
PLAYLIST.DAT
剪辑目录
00001.CLP
00002.CLP
00003.CLP
流目录
00001.PS
00002.PS
00003.PS

图 4

语法	位数	助记符
"PLAYLIST.DAT"{		
name_length	8	uimsbf
name_string	8x255	bslbf
number_of_PlayLists	16	uimsbf
for(i=0;i<number_of_PlayLists;i++){		
PlayList(){ // A PlayList()		
PlayList_data_length	32	uimsbf
// ATTRIBUTE INFORMATION		
reserved_for_word_alignment	15	bslbf
capture_enable_flag_PlayList	1	bslbf
PlayList_name_length	8	uimsbf
PlayList_name_string	8*255	bslbf
//		
number_of_PlayItems	16	uimsbf
for(i=0;i<number_of_PlayItems;i++){		
PlayItem()		
}		
PlayListMark()		
}		
}		
}		

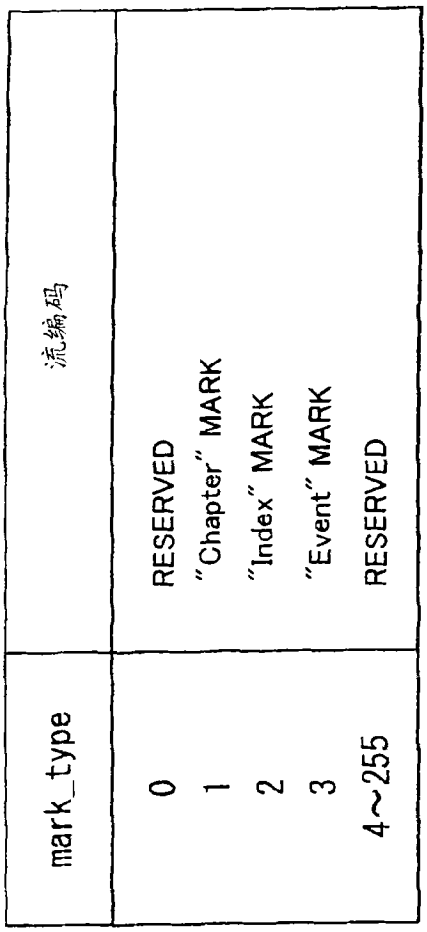
图 5

语法	位数	助记符
PlayItem(){		
length	16	uimsbf
Clip_Information_file_name_length	16	uimsbf
Clip_Information_file_name	8*Clip_Information_file_name_length	bslbf uimsbf uimsbf
IN_time	32	
OUT_time	32	
}		

图 6

语法	位数	助记符
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0;i < number_of_PlayList_marks;i++){		
Mark(){		
mark_type	8	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_stream_id	8	uimsbf
entry_ES_private_stream_id	8	uimsbf
mark_data	32	bslbf
mark_name_string	8*24	bslbf
}		
}		
}		

图 7



mark_type	流编码
0	RESERVED
1	"Chapter" MARK
2	"Index" MARK
3	"Event" MARK
4～255	RESERVED

图 8

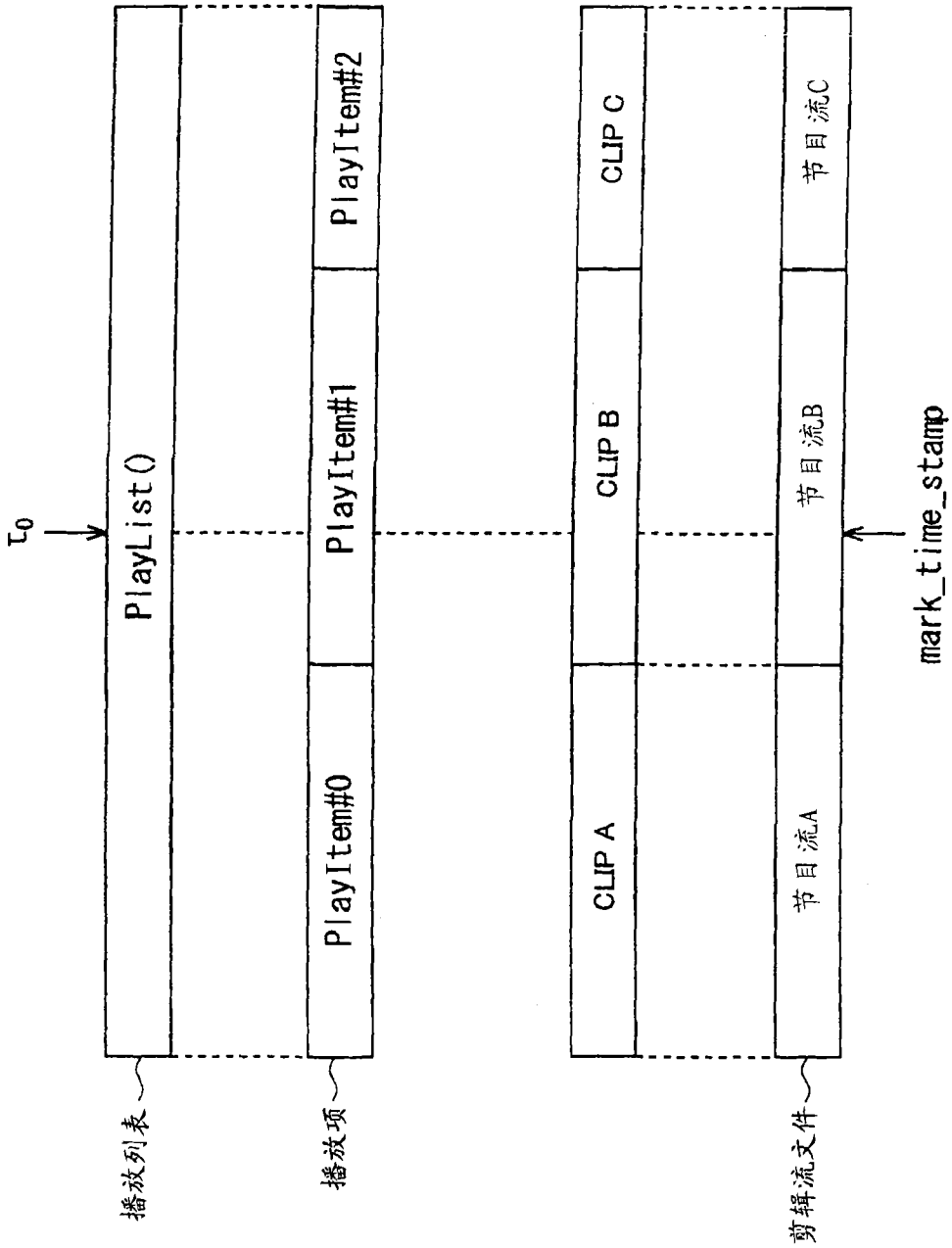

PlayList()
PlayItem#0
PlayItem#1
PlayItem#2
CLIP A
CLIP B
CLIP C
节目流A
节目流B
节目流C
播放列表
播放项
剪辑流文件
mark_time_stamp

图 9

语法	位数	助记符
XXXXX.CLP{		
presentation_start_time	32	uimsbf
presentation_end_time	32	uimsbf
reserved_for_word_alignment	7	bslbf
capture_enable_flag_Clip	1	bslbf
number_of_streams	8	uimsbf
for(i=0;i < number_of_streams;i++){		
StreamInfo(){		
length	16	uimsbf
stream_id	8	uimsbf
private_stream_id	8	uimsbf
StaticInfo()		
reserved_for_word_alignment	8	bslbf
number_of_DynamicInfo	8	uimsbf
for(j=0;j < number_of_DynamicInfo;j++){		
pts_change_point	32	uimsbf
DynamicInfo()		
}		
}		
}		
EP_map()		
}		

图 10

基本流类型	stream_id	private_stream_id
视频	0xE0～0xEF	（无）
ATRAC音频	0xBD	0x00～0x0F
LPCM音频	0xBD	0x10～0x1F
子标题	0xBD	0x80～0x9F

图 11

语法	位数	助记符
StaticInfo() {		
if(stream == VIDEO){		
reserved_for_word_alignment	16	bslbf
picture_size	4	uimsbf
frame_rate	4	uimsbf
reserved_for_word_alignment	7	bslbf
cc_flag	1	bslbf
}else if (stream == AUDIO) {		
audio_language_code	16	bslbf
channel_configuration	8	uimsbf
reserved_for_word_alignment	3	bslbf
lfe_existence	1	bslbf
sampling_frequency	4	uimsbf
}else if (stream == SUBTITLE) {		
subtitle_language_code	16	bslbf
reserved_for_word_alignment	15	bslbf
configurable_flag	1	uimsbf
}		
}		

图 12

语法	位数	助记符
DynamicInfo(i, j) {		
reserved_for_word_alignment	8	bslbf
if(stream == VIDEO) {		
reserved_for_word_alignment	4	bslbf
display_aspect_ratio	4	uimsbf
} else if(stream == AUDIO) {		
reserved_for_word_alignment	4	bslbf
channel_assignment	4	uimsbf
} else if(stream == SUBTITLE) {		
reserved_for_word_alignment	8	bslbf
}		
}		

图 13

语法	位数	助记符
EP_map(){		
reserved_for_word_alignment	8	bslbf
number_of_stream_id_entries	8	uimsbf
for(k=0;k<number_of_stream_id_entries;k++){		
stream_id	8	bslbf
private_stream_id	8	bslbf
number_of_EP_entries	32	uimsbf
for(i=0;i< number_of_EP_entries;i++){		
PTS_EP_start	32	uimsbf
RPN_EP_start	32	uimsbf
}		
}		
}		

图 14

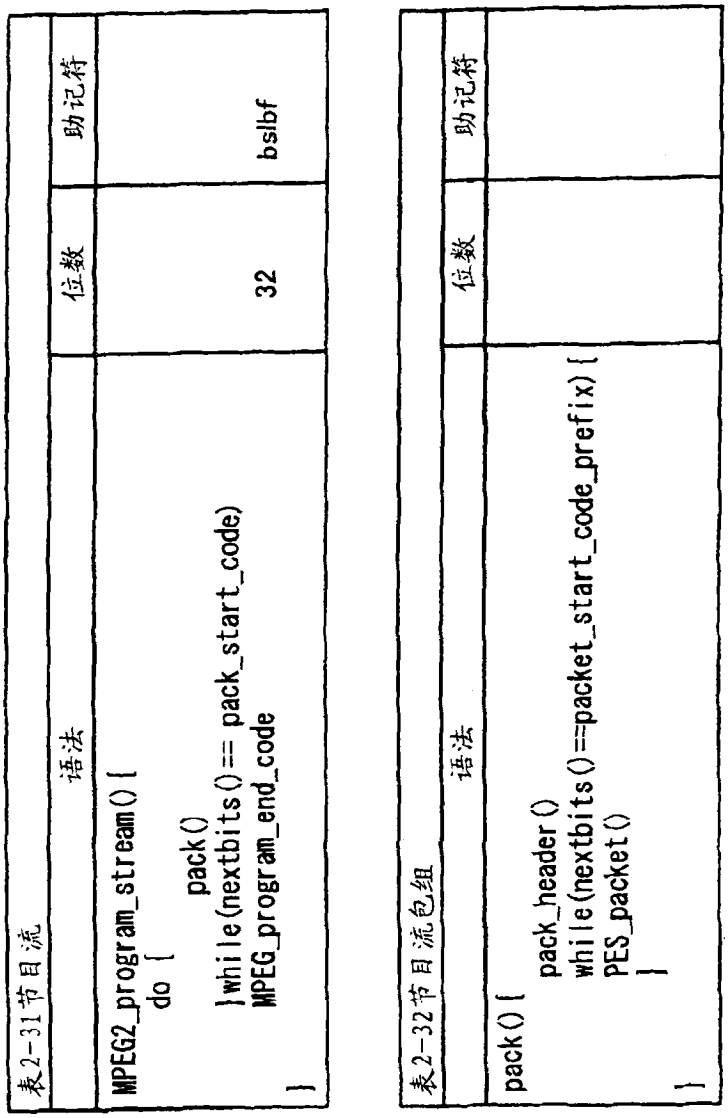


表2-31节目流

语法	位数	助记符
MPEG2_program_stream() {		
do {		
pack()		
} while (nextbits() == pack_start_code)		
MPEG_program_end_code	32	bslbf
}		

表2-32节目流包组

语法	位数	助记符
pack() {		
pack_header()		
while (nextbits()==packet_start_code_prefix) {		
PES_packet()		
}		
}		

图 15A

表2-33节目流包组头

语法	位数	助记符
pack_header() {		
pack_start_code	32	bslbf
'01'	2	bslbf
system_clock_reference_base[32..30]	3	bslbf
marker_bit	1	bslbf
system_clock_reference_base[29..15]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_base[14..0]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_extension	9	uimsbf
marker_bit	1	bslbf
program_mux_rate	22	uimsbf
marker_bit	1	bslbf
marker_bit	1	bslbf
reserved	5	bslbf
pack_stuffing_length	3	uimsbf
for(i=0;i<pack_stuffing_length;i++) {		
stuffing_byte	8	bslbf
}		
if(nextbits()==system_header_start_code) {		
system_header()		
}		
}		

图 15B

表2-17PES分组

语法	位数	助记符
PES_packet(){		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if(stream_id !=program_stream_map		
&& stream_id !=padding_stream		
&& stream_id !=private_stream_2		
&& stream_id !=ECM		
&& stream_id !=EMM		
&& stream_id !=program_stream_directory		
&& stream_id !=DSMCC_stream		

图 16A

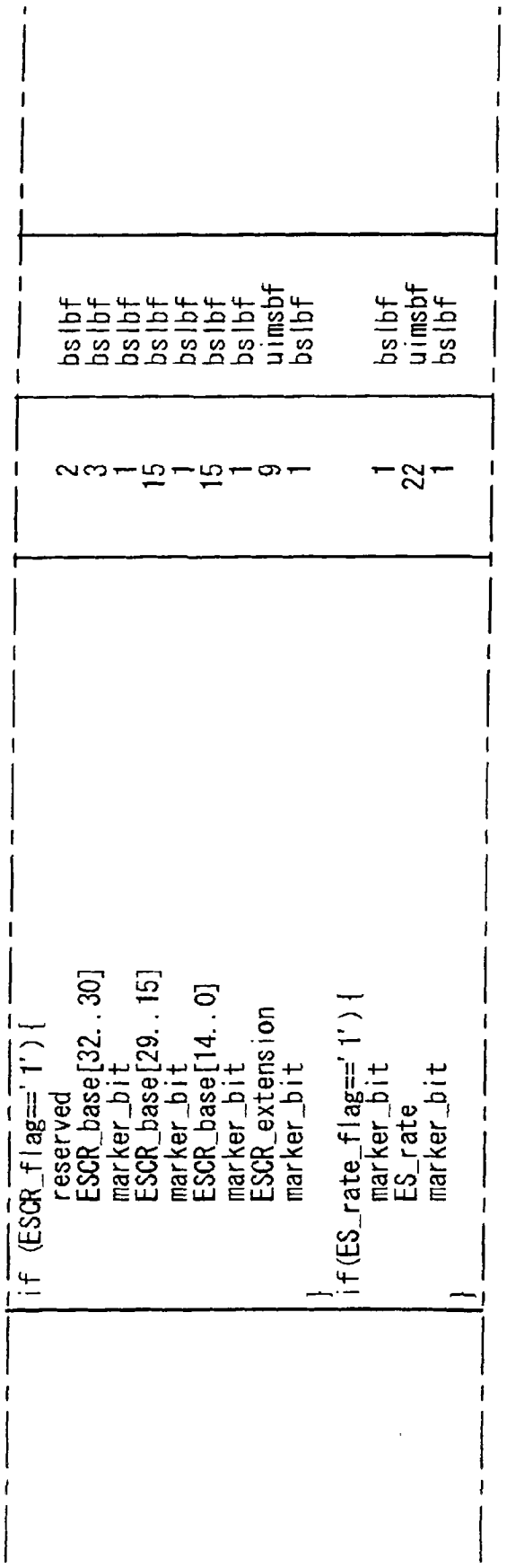
&& stream_id !=ITU-T Rec. H.222.1 type E stream){		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsbf
if (PTS_DTS_flags == '10') {		
'0010' 4 bslbf	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		

图 16B

表2-17PES分组(续)

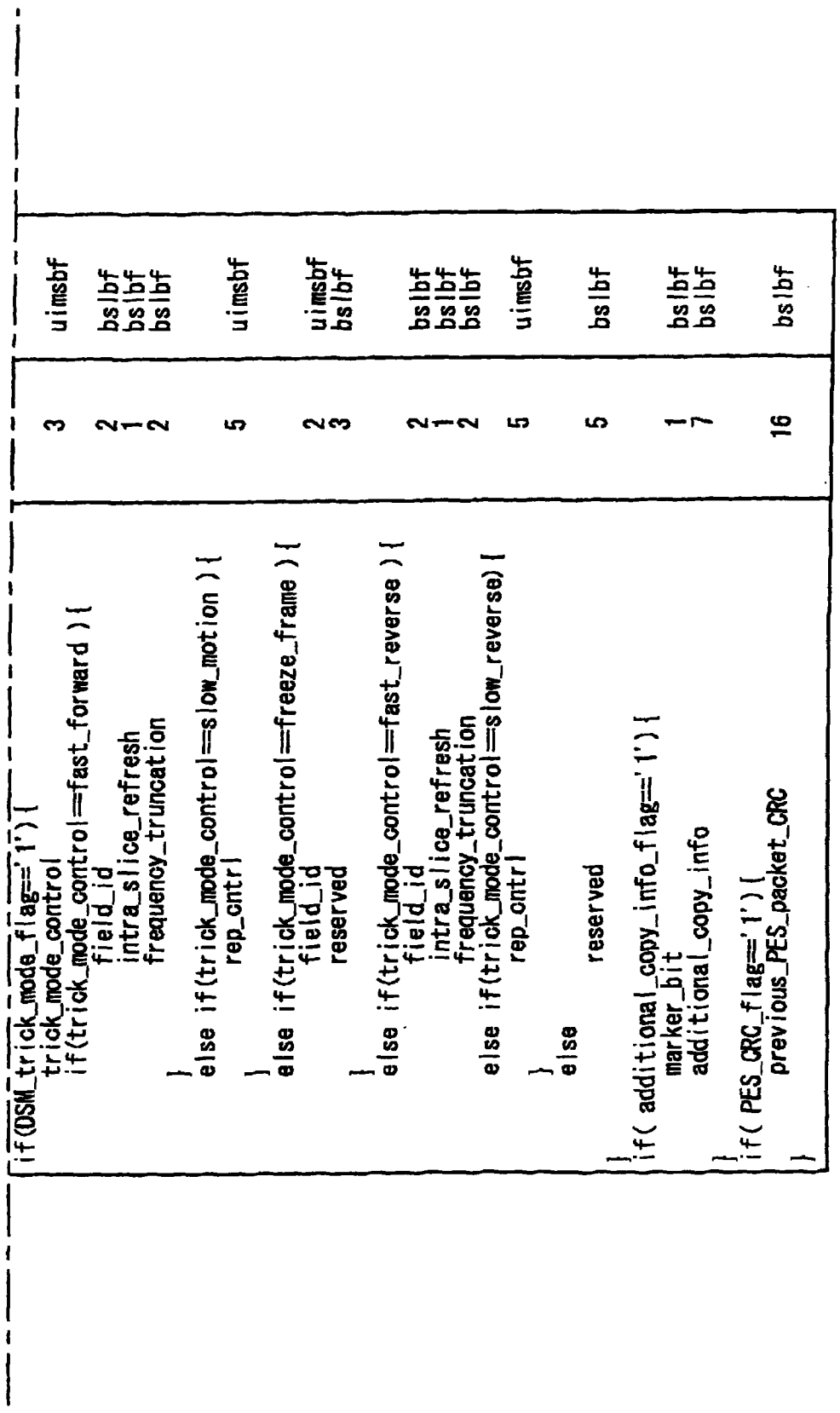
语法	位数	助记符
if(PTS_DTS_flags=='11'){		
'0011'	4	bslbf
PTS[32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		

图 17A



if (ESCR_flag=='1'){		
reserved	2	bslbf
ESCR_base[32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base[29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base[14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if(ES_rate_flag=='1'){		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		

图 17B



if(DSM_trick_mode_flag=='1'){		
trick_mode_control	3	uimsbf
if(trick_mode_control==fast_forward){		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if(trick_mode_control==slow_motion){		
rep_cntrl	5	uimsbf
}		
else if(trick_mode_control==freeze_frame){		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if(trick_mode_control==fast_reverse){		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if(trick_mode_control==slow_reverse){		
rep_cntrl	5	uimsbf
}		
else		
reserved	5	bslbf
}		
if(additional_copy_info_flag=='1'){		
marker_bit	1	bslbf
additional_copy_info	7	bslbf
}		
if(PES_CRC_flag=='1'){		
previous_PES_packet_CRC	16	bslbf
}		

图 17C

表2-17PES分组(结束)

语法	位数	助记符
if(PES_extension_flag=='1'){		
PES_private_data_flag	1	bslbf
pack_header_field_flag	1	bslbf
program_packet_sequence_counter_flag	1	bslbf
P-STD_buffer_flag	1	bslbf
reserved	3	bslbf
PES_extension_flag_2	1	bslbf
if(PES_private_data_flag=='1'){		
PES_private_data	128	bslbf
}		
if (pack_header_field_flag == '1') {		
pack_field_length	8	uimsbf
pack_header ()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	1	bslbf
program_packet_sequence_counter	7	uimsbf
marker_bit	1	bslbf
MPEG1_MPEG2_identifier	1	bslbf
original_stuff_length	6	uimsbf
}		
if(P-STD_buffer_flag == '1') {		
'01'	2	bslbf
P-STD_buffer_scale	1	bslbf
P-STD_buffer_size	13	uimsbf
}		

图 18A

if(PES_extension_flag_2 == '1') {		
marker_bit	1	bslbf
PES_extension_field_length	7	uimsbf
for (i=0; i<PES_extension_field_length; i++) {		
reserved	8	bslbf
}		
}		
}		
for (i=0; i<N1; i++) {		
stuffing_byte	8	bslbf
}		
for (i=0; i<N2; i++) {		
PES_packet_data_byte	8	bslbf
}		
}		
else if(stream_id == program_stream_map		
||stream_id == private_stream_2		
||stream_id == ECM		
||stream_id == EMM		
||stream_id == program_stream_directory		
||stream_id == DSMCC_stream		
||stream_id == ITU-T Rec. H. 222.1 type E stream) {		
for (i=0; i<PES_packet_length; i++) {		
PES_packet_data_byte	8	bslbf
}		
}		
else if(stream_id == padding_stream) {		
for (i=0; i<PES_packet_length; i++) {		
padding_byte	8	bslbf
}		
}		
}		

图 18B

TABLE 2-18, stream_id 分配		
stream_id	注释	流编码
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 … 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

图 19A

The notation x means that the values '0' or '1' are both permitted and results in the same stream type. The stream number is given by the values taken by the x's.
NOTE 1: PES packets of type program_stream_map have unique syntax specified in 2.5.4.1.
NOTE 2: PES packets of type private_stream_1 and ISO/IEC_13552_stream follow the same PES packet syntax as those for ITU-T Rec. H.262 | ISO/IEC 13818-2 video and ISO/IEC 13818-3 audio streams.
NOTE 3: PES packets of type private_stream_2, ECM_stream and EMM_stream are similar to private_stream_1 except no syntax is specified after PES_packet_length field.
NOTE 4: PES packets of type program_stream_directory have a unique syntax specified in 2.5.5.
NOTE 5: PES packets of type DSM-CC_stream have a unique syntax specified in ISO/IEC 13818- 6.
NOTE 6: This stream_id is associated with stream_type 0x09 in Table 2-29.
NOTE 7: This stream_id is only used in PES packets, which carry data from a Program Stream or an ISO/IEC 11172-1 System Stream, in a Transport Stream (refer to 2.4.3.7).

图 19B

stream_id	流编码 (流类型)
1011 1101	private_stream_1
1011 1110	padding_stream
1011 1111	private_stream_2
110x xxxx	音频流号 x xxxx
1110 xxxx	视频流号 xxxx

图 20

语法	位数	助记符
private_stream1_PES_payload() {		
private_header() {		
private_stream_id	8	uimsbf
if(stream == ATRAC) {		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
if(stream == LPCM) {		
fs_flag	1	uimsbf
reserved_for_future_use	3	bslbf
ch_flag	4	uimsbf
AU_locator	16	uimsbf
}else if (stream == SUBTITLE) {		
reserved_for_future_use	8	bslbf
AU_locator	16	uimsbf
}		
}		
private_payload()		
}		

图 21

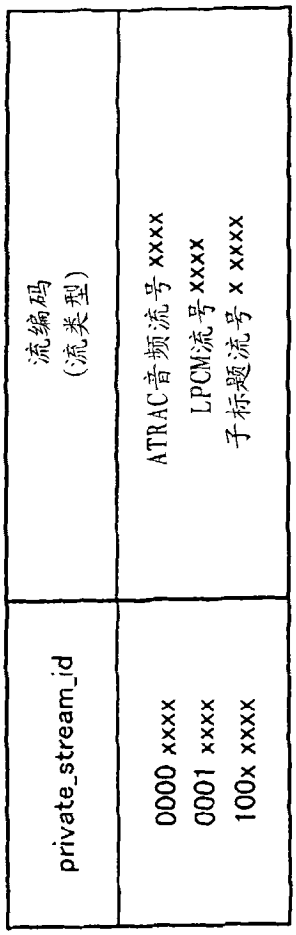

流编码
(流类型)
private_stream_id
ATRAC音频流号 xxxx
LPCM流号 xxxx
子标题流号 x xxxx
0000 xxxx
0001 xxxx
100x xxxx

图 22

语法	位数	助记符
private_stream2_PES_payload(){		
reserved_for_future_use	8	bslbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
VBI()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
}		

图 23

语法	位数	助记符
au_information(){		
length	16	uimsbf
reserved_for_word_alignment	8	bslbf
number_of_access_unit	8	uimsbf
for(i=0;i<number_of_access_unit;i++){		
reserved	4	bslbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

图 24

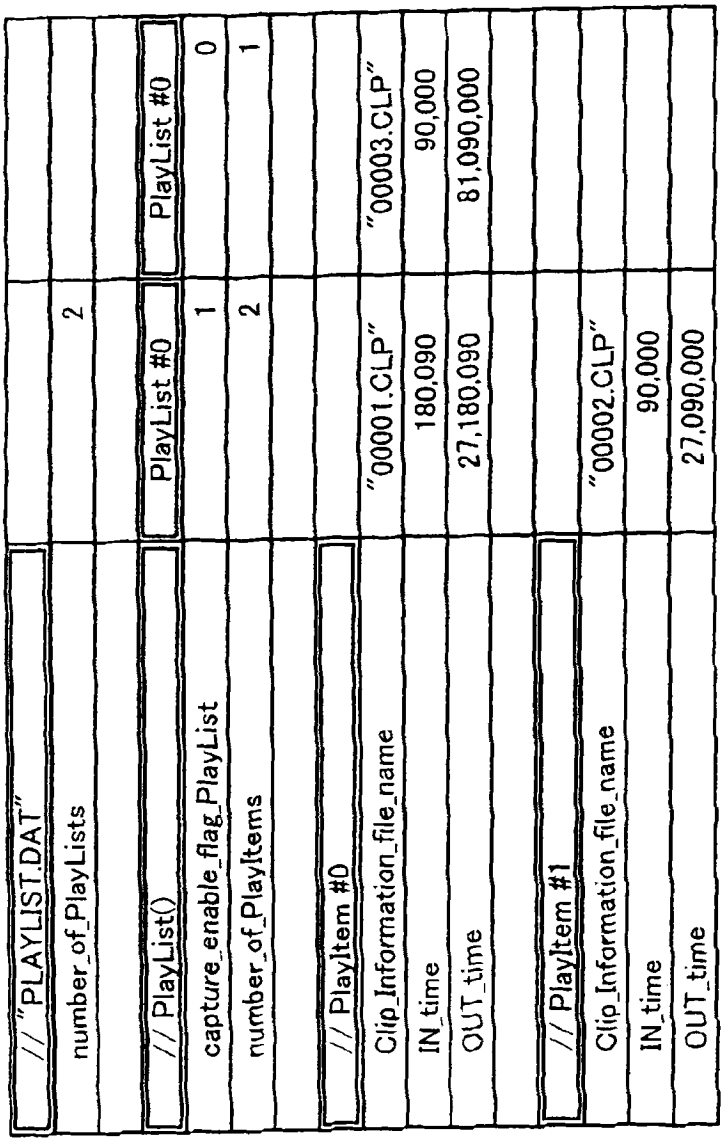
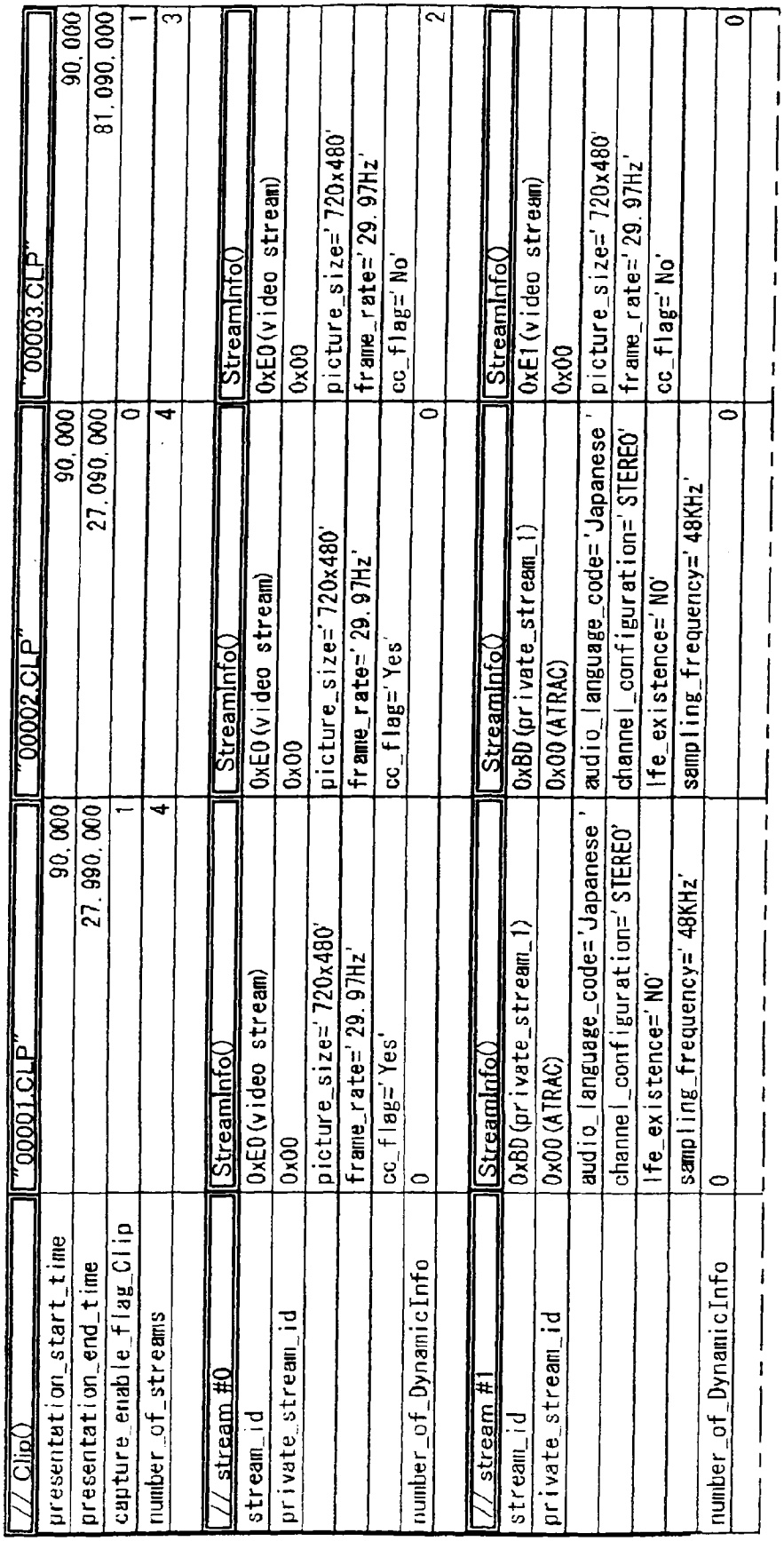
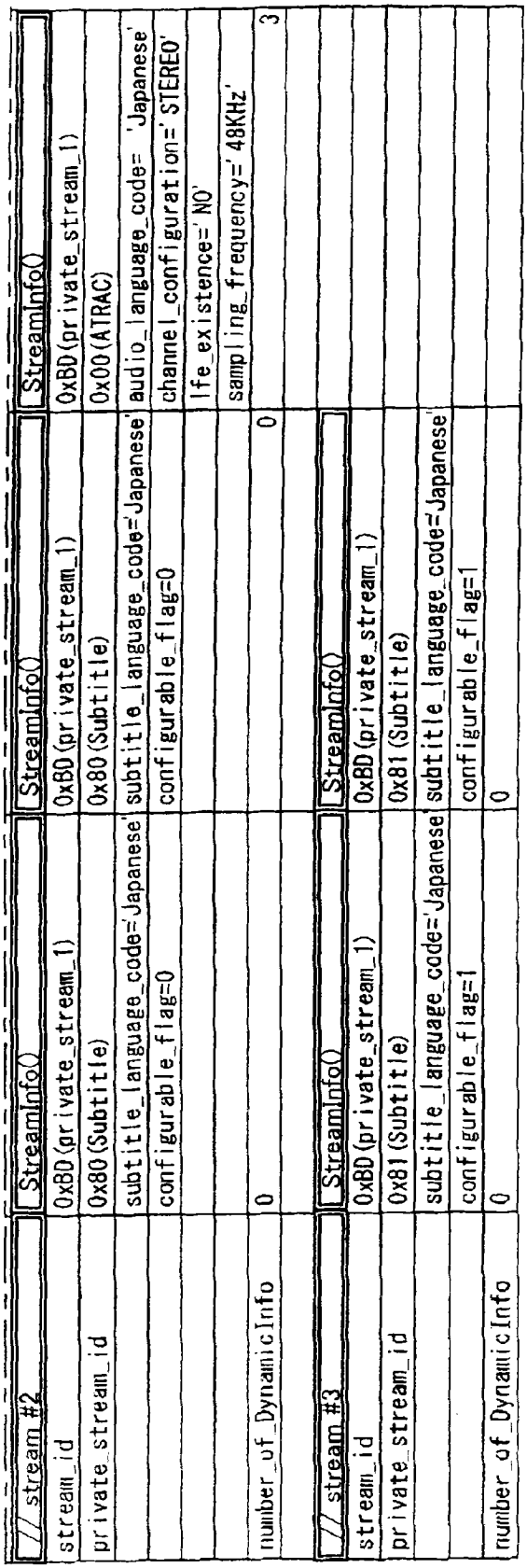

// "PLAYLIST.DAT"
number_of_PlayLists
2
PlayList #0
PlayList #0
// PlayList()
capture_enable_flag_PlayList
1
0
number_of_PlayItems
2
1
// PlayItem #0
"00001.CLP"
"00003.CLP"
Clip_Information_file_name
IN_time
180,090
90,000
OUT_time
27,180,090
81,090,000
// PlayItem #1
"00002.CLP"
Clip_Information_file_name
IN_time
90,000
OUT_time
27,090,000

图 25



// Clip()	"00001.CLP"	"00002.CLP"	"00003.CLP"
presentation_start_time	90,000	90,000	90,000
presentation_end_time	27,990,000	27,090,000	81,090,000
capture_enable_flag_Clip	1	0	1
number_of_streams	4	4	3
// stream #0	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xE0(video stream)	0xE0(video stream)	0xE0(video stream)
private_stream_id	0x00	0x00	0x00
picture_size	'720x480'	'720x480'	'720x480'
frame_rate	'29.97Hz'	'29.97Hz'	'29.97Hz'
cc_flag	'Yes'	'Yes'	'No'
number_of_DynamicInfo	0	0	2
// stream #1	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD(private_stream_1)	0xBD(private_stream_1)	0xE1(video stream)
private_stream_id	0x00(ATRAC)	0x00(ATRAC)	0x00
audio_language_code	'Japanese'	'Japanese'	'Japanese'
channel_configuration	'STEREO'	'STEREO'	'STEREO'
lfe_existence	'NO'	'NO'	'NO'
sampling_frequency	'48KHz'	'48KHz'	'48KHz'
number_of_DynamicInfo	0	0	0

图 26A



// stream #2	StreamInfo()	StreamInfo()	StreamInfo()
stream_id	0xBD (private_stream_1)	0xBD (private_stream_1)	0xBD (private_stream_1)
private_stream_id	0x80 (Subtitle)	0x80 (Subtitle)	0x00 (ATRAC)
	subtitle_language_code='Japanese'	subtitle_language_code='Japanese'	audio_language_code= 'Japanese'
	configurable_flag=0	configurable_flag=0	channel_configuration='STEREO'
			lfe_existence='NO'
			sampling_frequency='48KHz'
number_of_DynamicInfo	0	0	3
// stream #3	StreamInfo()	StreamInfo()	
stream_id	0xBD (private_stream_1)	0xBD (private_stream_1)	
private_stream_id	0x81 (Subtitle)	0x81 (Subtitle)	
	subtitle_language_code='Japanese'	subtitle_language_code='Japanese'	
	configurable_flag=1	configurable_flag=1	
number_of_DynamicInfo	0	0	

图 26B

EP_map() for "00001.CLP"			
number_of_stream_id_entries	1		
stream_id	0xE0(video stream)		
private_stream_id	0x00		
	PTS_EP_start	RPN_EP_start	
	90000	0	
	135045	244	
	180090	305	
	225135	427	
	270180	701	

图 27

PlayListMark() OF PlayList #0
number_of_PlayList_marks==7

Mark()	mark_type	ref_to_PlayItem_id	mark_time_stamp	entry_ES_stream_id	entry_ES_private_stream_id	mark_data	
#0	章节	0	180,090	0	0	1	开始
#1	索引	0	5,580,090	0	0	1	1分钟
#2	索引	0	10,980,090	0	0	2	2分钟
#3	事件	0	16,380,090	0	0	0	3分钟
#4	章节	1	90,000	0	0	2	开始
#5	索引	1	5,490,000	0	0	1	1分钟
#6	索引	1	10,890,000	0	0	2	2分钟

PlayListMark() OF PlayList #1
number_of_PlayList_marks==3

Mark()	mark_type	ref_to_PlayItem_id	mark_time_stamp	entry_ES_stream_id	entry_ES_private_stream_id	mark_data	
#0	章节	0	90,000	0	0	0	开始
#1	事件	0	27,090,000	0xE0	0	1	5分钟
#2	事件	0	27,540,000	0xE1	0	2	5分5秒 (305秒)

图 28

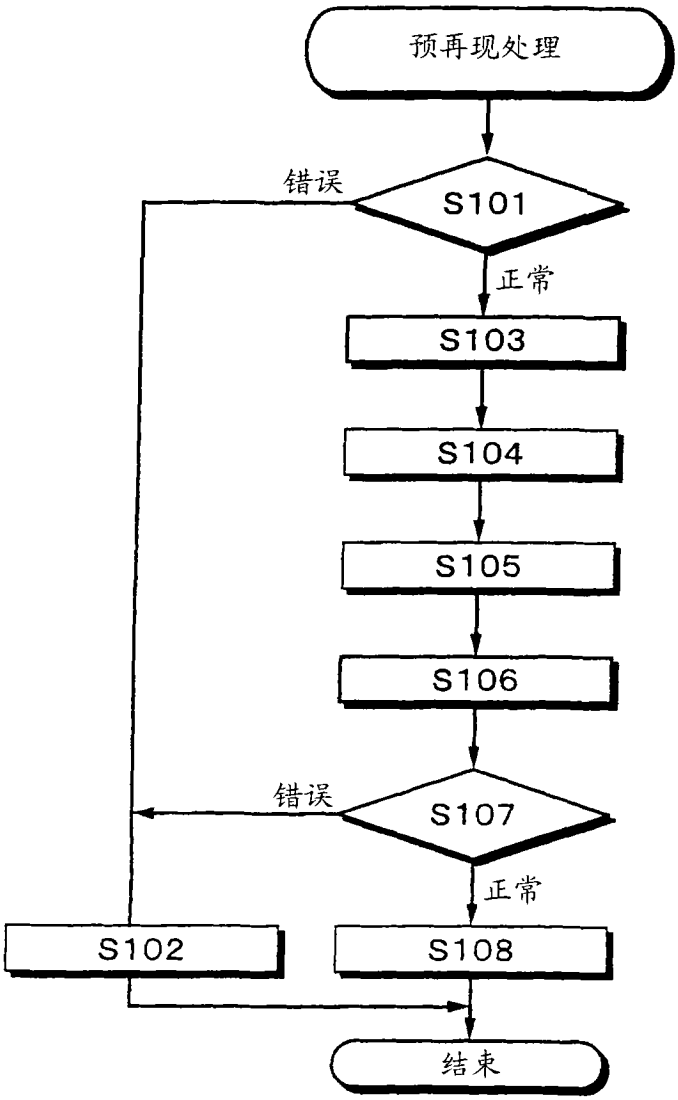

预再现处理
S101
错误
正常
S103
S104
S105
S106
S107
错误
正常
S102
S108
结束

图 29

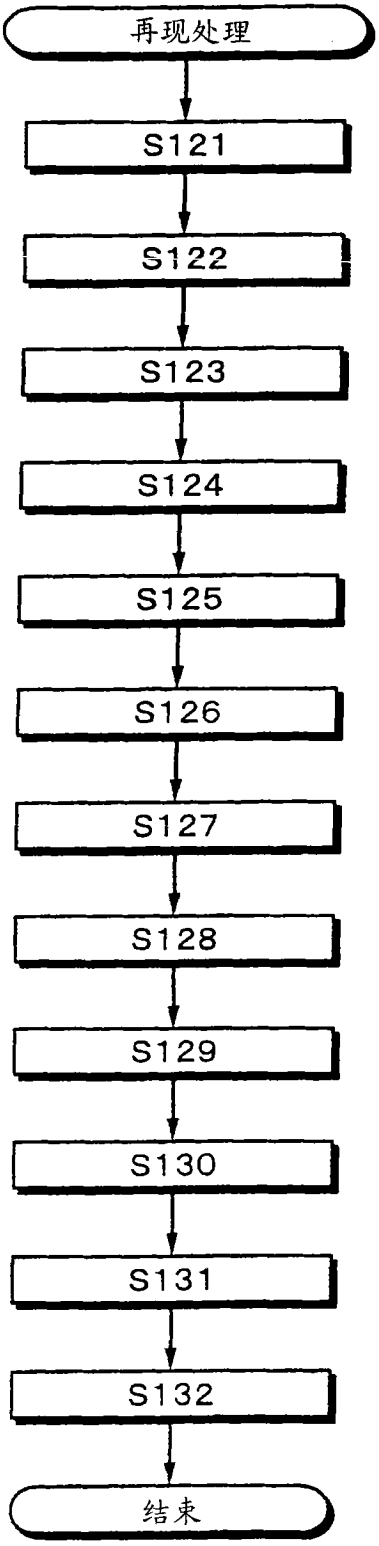

再现处理
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
结束

图 30

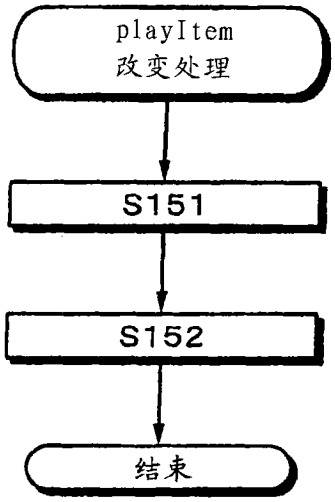

playItem
改变处理
S151
S152
结束

图 31

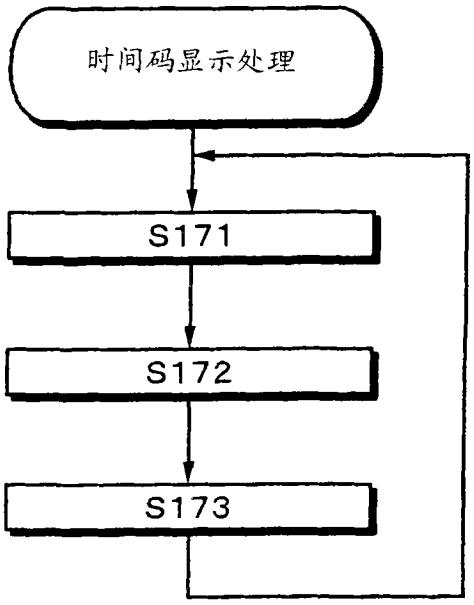

时间码显示处理
S171
S172
S173

图 32

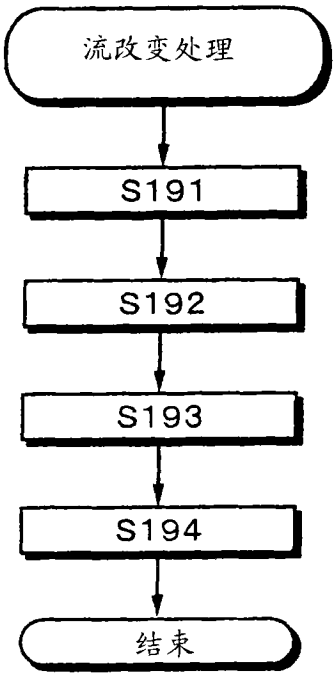

流改变处理
S191
S192
S193
S194
结束

图 33

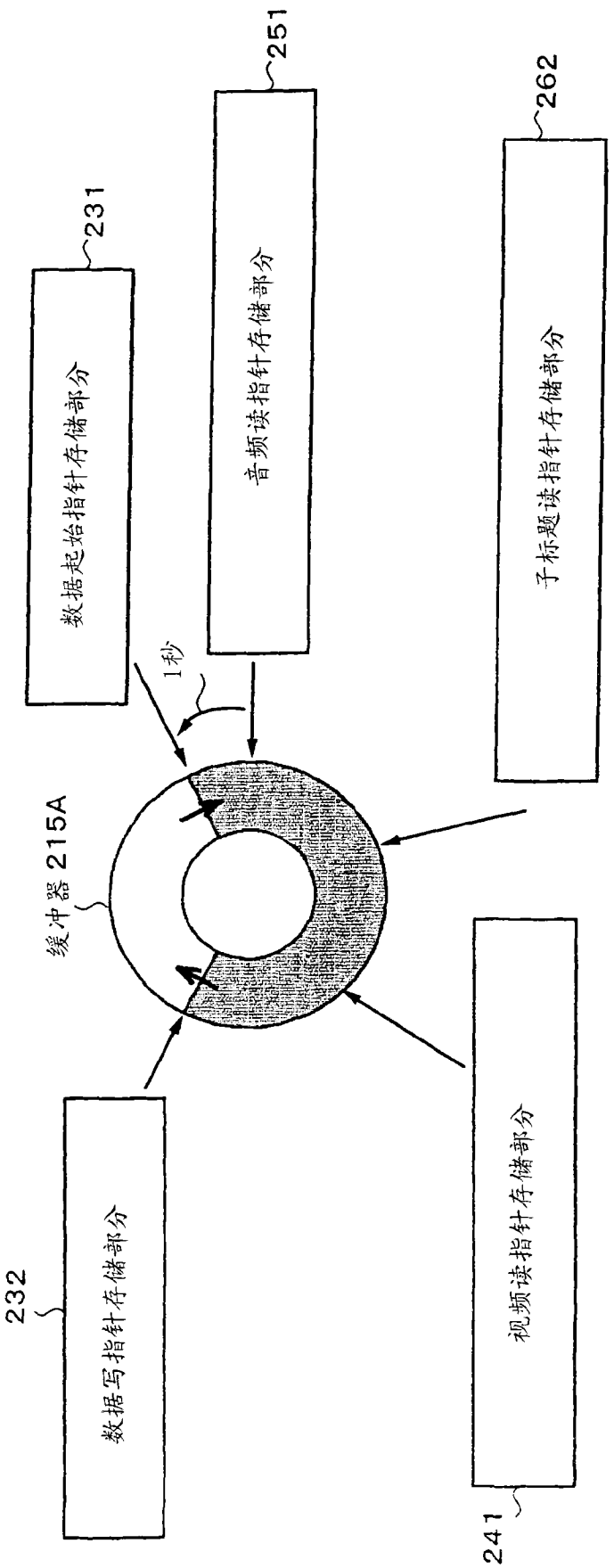

231
数据起始指针存储部分
251
音频读指针存储部分
262
子标题读指针存储部分
1秒
缓冲器 215A
232
数据写指针存储部分
241
视频读指针存储部分

图 34

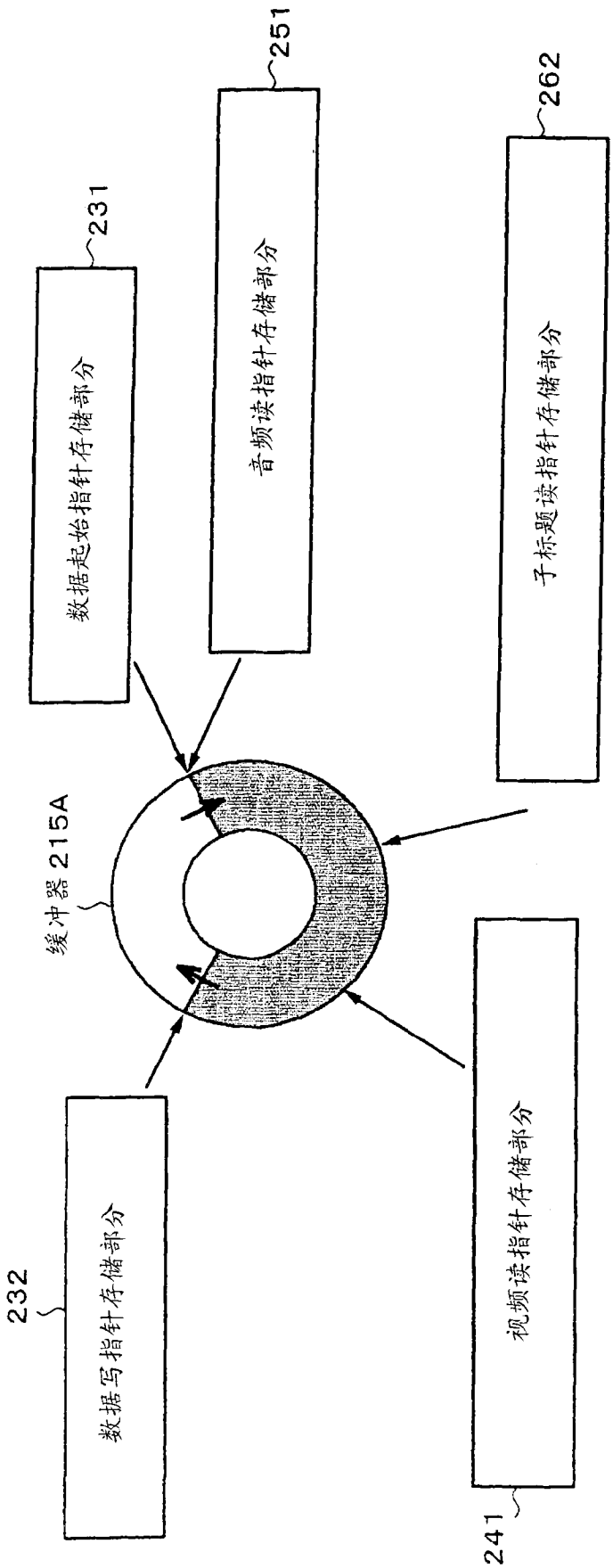

231
数据起始指针存储部分
251
音频读指针存储部分
262
子标题读指针存储部分
缓冲器 215A
232
数据写指针存储部分
241
视频读指针存储部分

图 35

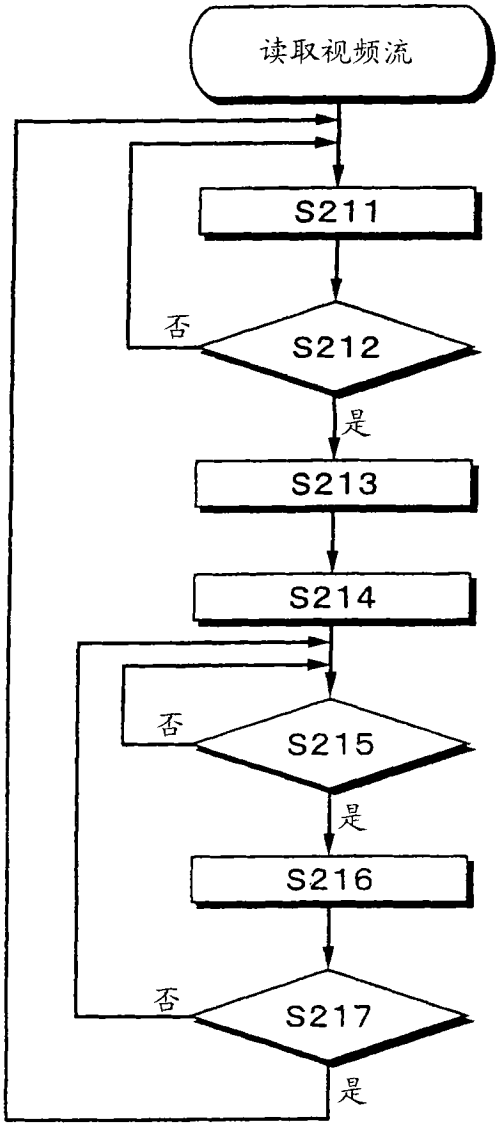


图 36

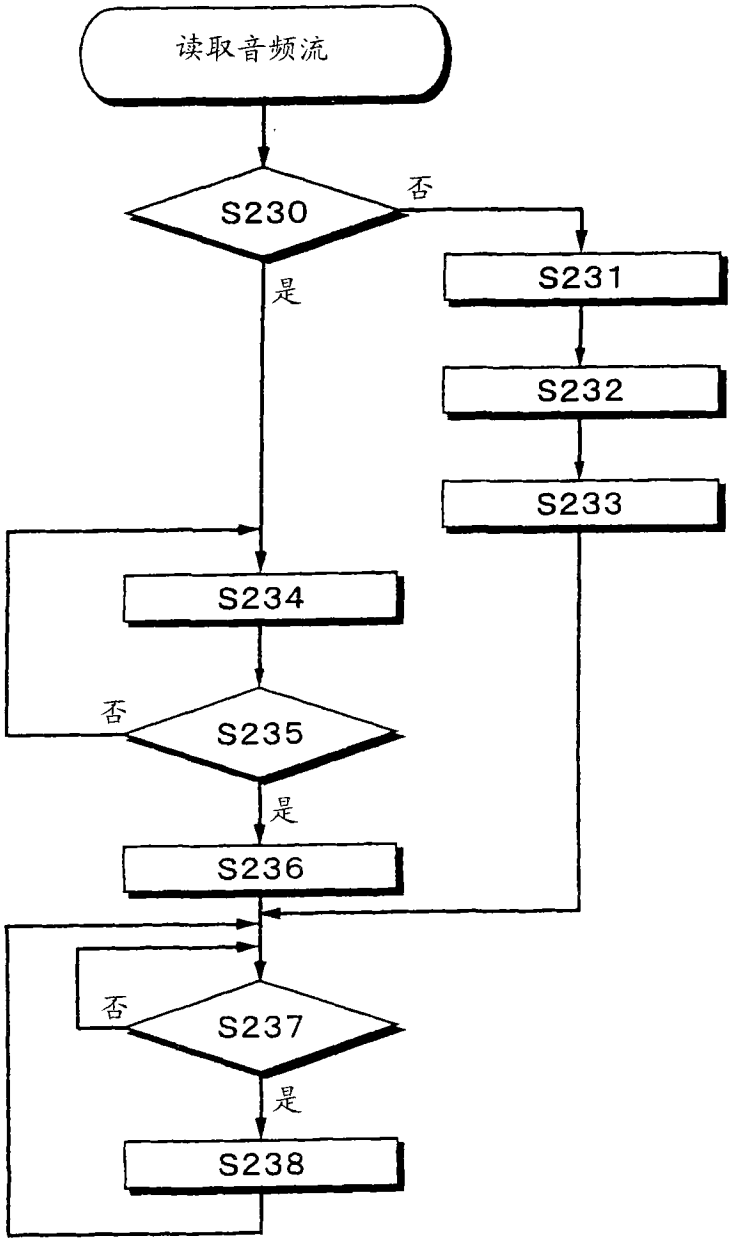


图 37

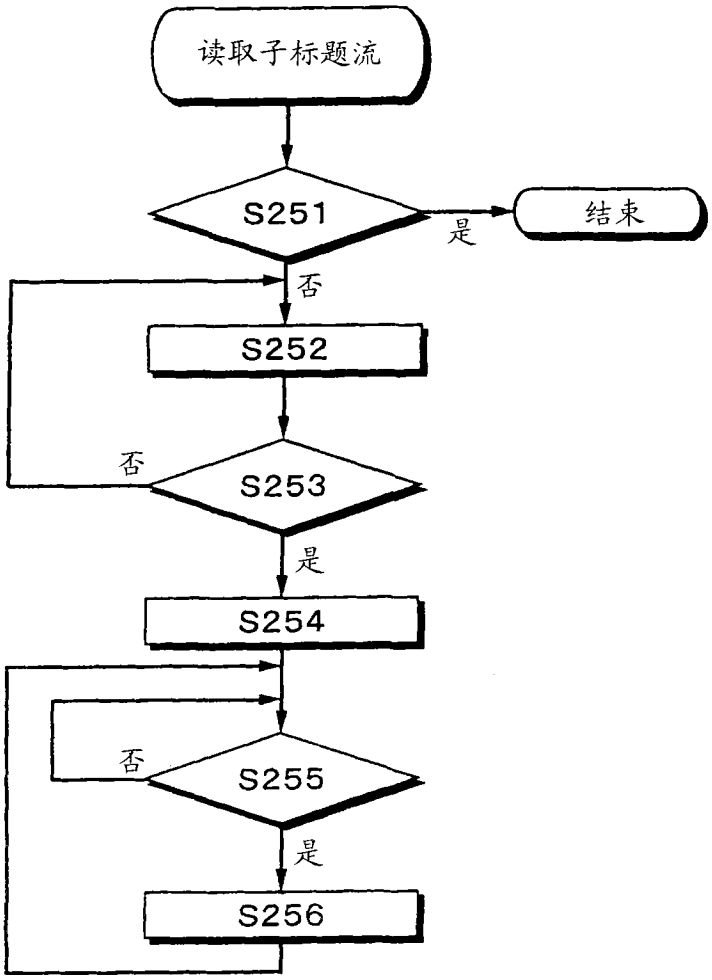


图 38

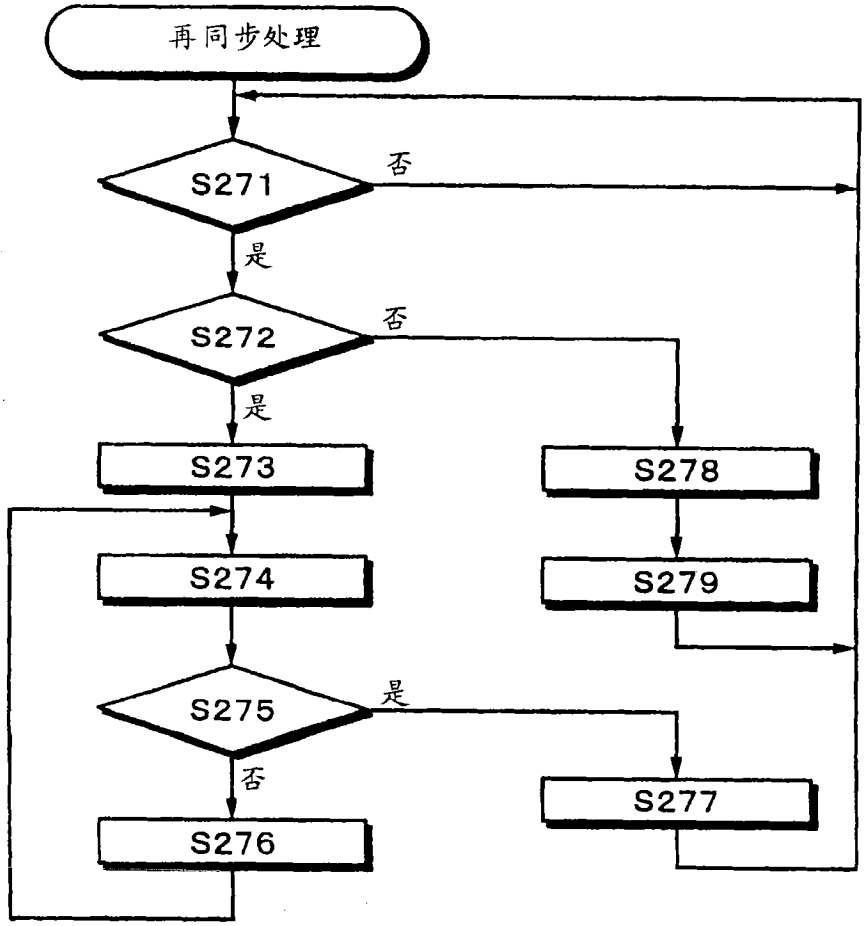


图 39

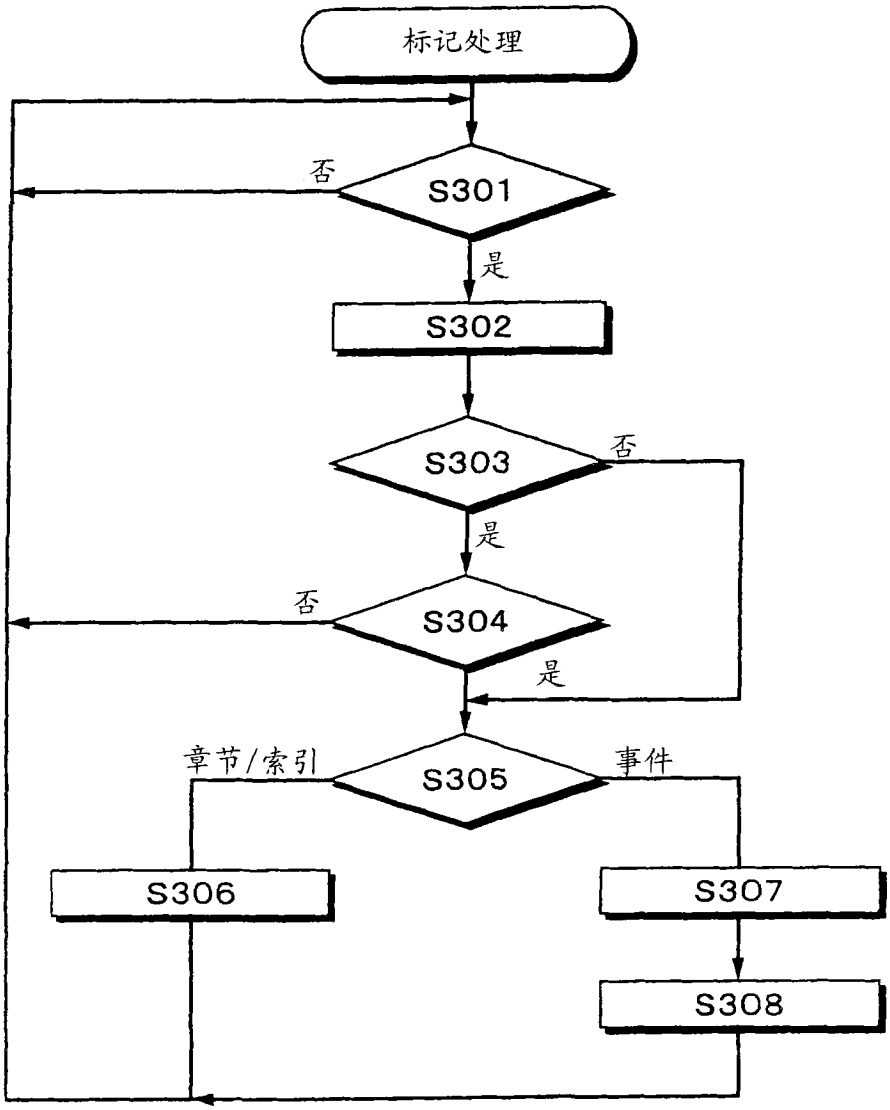


图 40

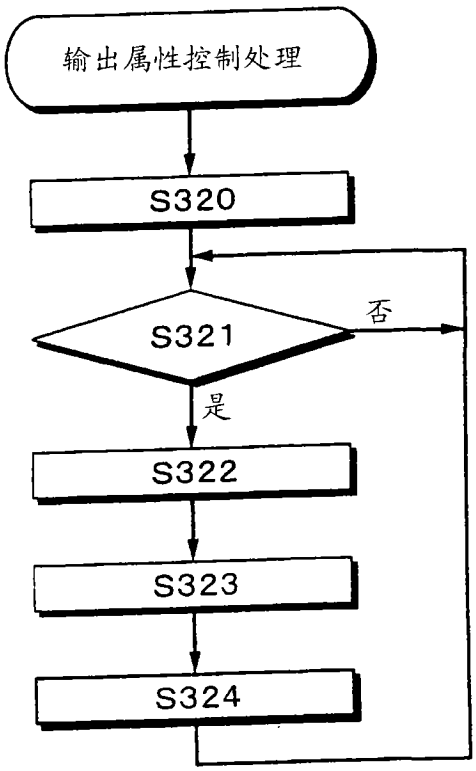


图 41

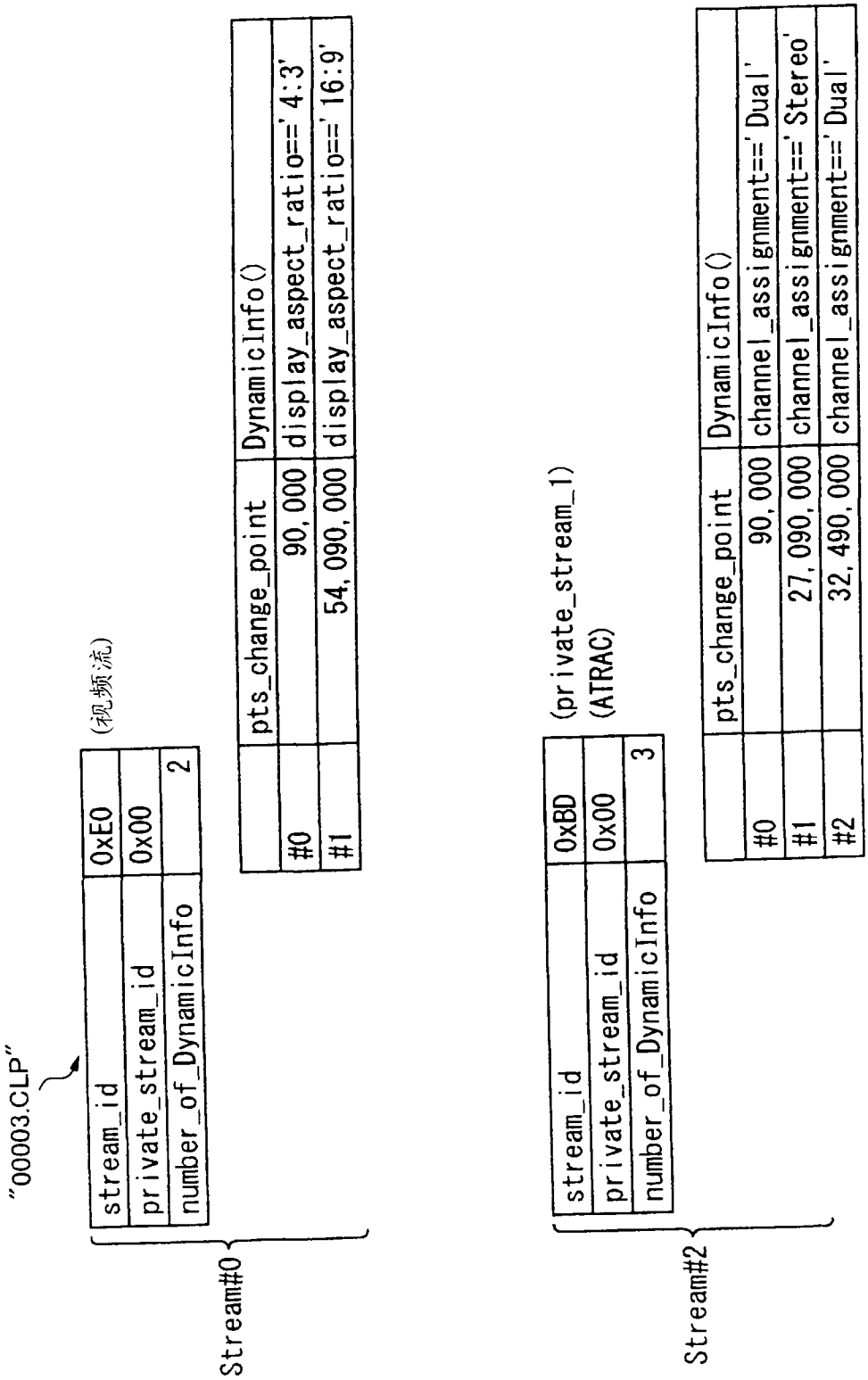

"00003.CLP"
stream_id
0xE0
(视频流)
private_stream_id
0x00
number_of_DynamicInfo
2
Stream#0
pts_change_point
DynamicInfo()
#0
90,000
display_aspect_ratio=='4:3'
#1
54,090,000
display_aspect_ratio=='16:9'
stream_id
0xBD
(private_stream_1)
(ATRAC)
private_stream_id
0x00
number_of_DynamicInfo
3
Stream#2
pts_change_point
DynamicInfo()
#0
90,000
channel_assignment=='Dual'
#1
27,090,000
channel_assignment=='Stereo'
#2
32,490,000
channel_assignment=='Dual'

图 42

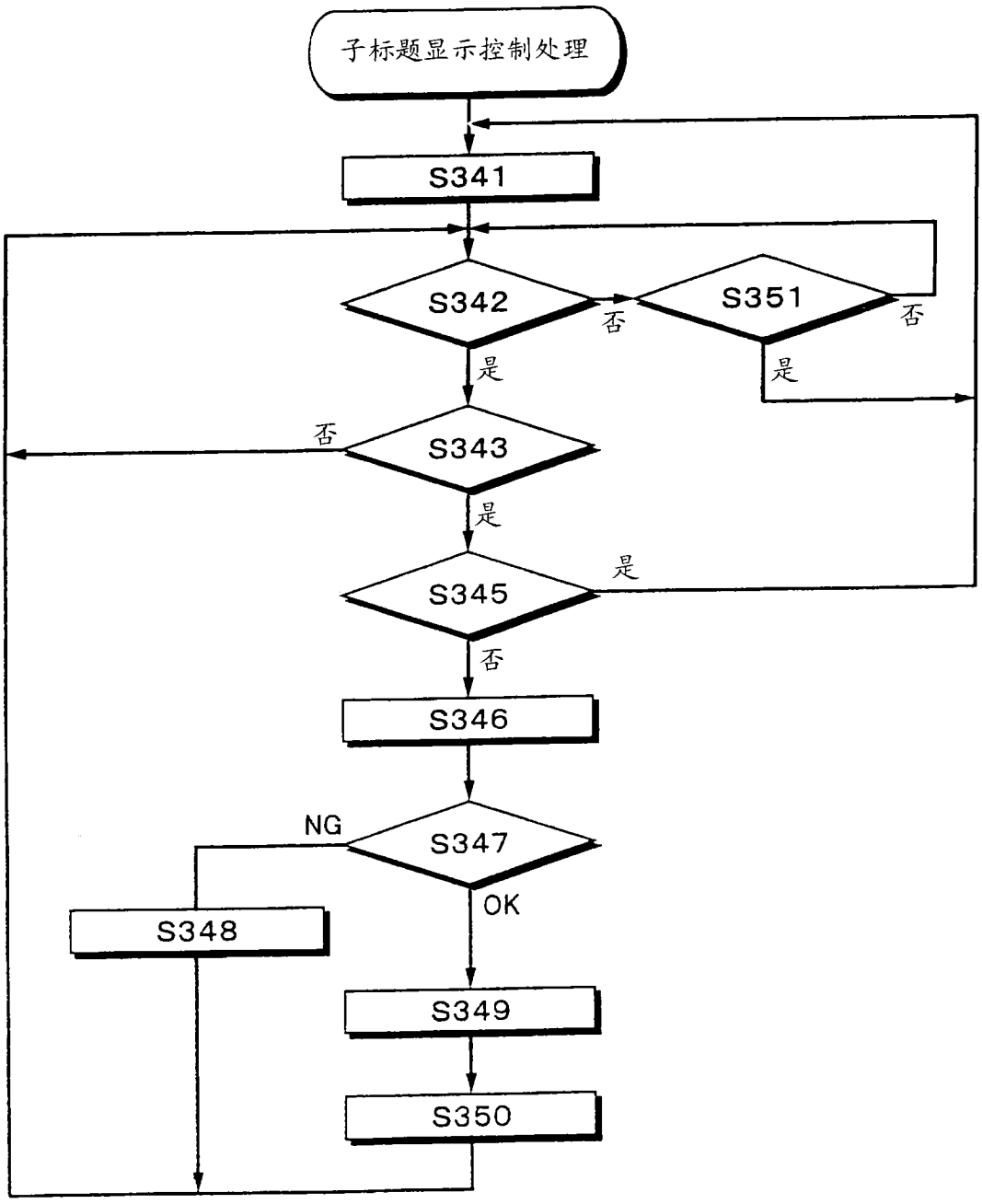
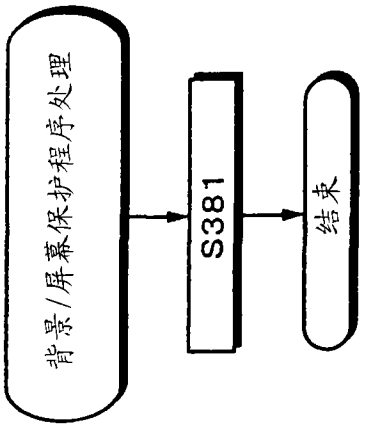

子标题显示控制处理
S341
S342
否
S351
否
是
是
否
S343
是
是
S345
否
S346
NG
S347
OK
S348
S349
S350

图 43


背景/屏幕保护程序处理
S381
结束

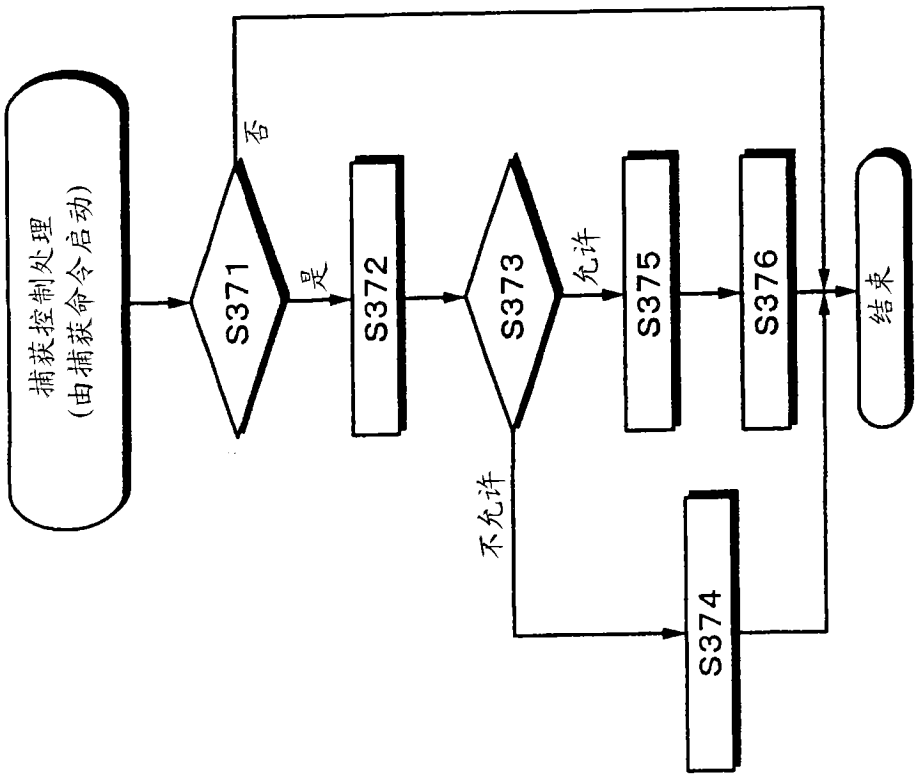

捕获控制处理
(由捕获命令启动)
S371
否
是
S372
S373
允许
S375
S376
结束
不允许
S374

图 44

语法	位数	助记符
private_stream2_PES_payload() {		
reserved_for_future_use	7	bslbf
capture_enable_ps2	1	uimsbf
video_stream_id	8	uimsbf
1stRef_picture	16	uimsbf
2ndRef_picture	16	uimsbf
3rdRef_picture	16	uimsbf
4thRef_picture	16	uimsbf
au_information()		
grain_variance_information()		
VBI()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
}		

图 45

语法	位数	助记符
au_information(){		
length	16	uimsbf
reserved_for_word_align	8	bslbf
number_of_access_unit	8	uimsbf
for(i=0;i<number_of_access_unit;i++){		
reserved	3	bslbf
capture_enable_flag_AU	1	uimsbf
pic_struct_copy	4	uimsbf
au_ref_flag	1	uimsbf
reserved	2	bslbf
AU_length	21	uimsbf
}		uimsbf
}		

图 46

图 46 对附图标记的描述 101 盘 102 盘驱动器 111 总线 112 CPU 113 存储器 114 驱动器接口 115 输入接口 116 视频解码器 117 音频解码器 118 视频输出接口 119 音频输出接口 120 视频输出端 121 音频输出端 201 操作系统 210 视频内容再现程序 211 脚本控制模块 212 播放器控制模块 213 内容数据提供模 214 解码控制模块 214A 时间计数部分 215 缓冲器控制模块 215A 缓冲器 216 视频解码器控制模块 217 音频解码器控制模块