(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 101002467 B
(45)授权公告日 2010.05.26

(21)申请号 200580027285.7

(22)申请日 2005.05.24

(30)优先权数据
174548/2004 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/JP2005/009435 2005.05.24

(87)PCT申请的公布数据
WO2005/122570 JA 2005.12.22

(73)专利权人 索尼株式会社
地址 日本东京都

(72)发明人 藤波靖 平野哲也

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 黄小临 王志森

(51)Int.Cl.
H04N 5/92(2006.01)
G11B 20/10(2006.01)
G11B 20/12(2006.01)
G11B 27/00(2006.01)
G11B 27/034(2006.01)

(56)对比文件
CN 1158540 A,1997.09.03, 说明书第2-6页,图1-图5.
JP 特开平9-200695 A,1997.07.31, 第3栏-第9栏,图1-5.
US 20020118744 A1,2002.08.29, 第11段-第41段,图1-图5.
JP 特开平9-139915 A,1997.05.27, 第50-53段,图1-图24.
CN 1149806 A,1997.05.14, 全文.
JP 3060919 B2,2000.04.28, 全文.

审查员 马永祥

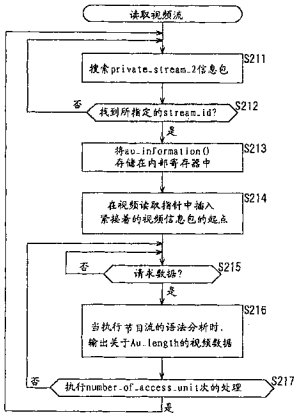
权利要求书 2 页 说明书 72 页 附图 44 页

(54)发明名称

数据处理设备和方法

(57)摘要

本发明涉及一种数据处理设备、数据处理方法、程序、程序记录介质、数据记录介质、和数据结构，其使得在不分析经编码的视频数据的情况下，在存取单元中的编码视频数据的读取成为可能。在步骤S211，搜索用于将编码视频数据解码，且位于紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前的private_stream_2信息包。另外，在步骤S216，基于private_stream_2信息包中所包括的且表示位于从private_stream_2信息包至下一个private_stream_2信息包的存取单元的每一个的大小的AU_length，以存取单元的形式读取编码视频数据。本发明能够适用于，例如，使用DVD的游戏机等。



CN 101002467 B

1. 一种用于处理经编码的数据的数据处理设备，其中：

所述经编码的数据包括：

通过以预定的单元编码视频数据而获得的、在存取单元中的编码视频数据；

紧邻在所述存取单元中的所述经编码的视频数据中的一个或多个解码可开始点的每一个之前的使用信息，所述使用信息在解码所述经编码的视频数据中使用，所述使用信息包括表示所述存取单元的每一个的大小的大小信息，所述存取单元的数目为一或更大，所述存取单元位于所述使用信息和下一个使用信息之间；和

识别视频流的第一识别信息以及识别音频流的第二识别信息；以及

所述数据处理设备包括：

搜索部件，用于从所述经编码的数据中搜索所述使用信息；和

提供部件，用于响应来自用于控制解码所述经编码的视频数据的视频解码器的视频解码器控制部件的请求，基于所述使用信息中所包括的所述大小信息，从所述经编码的数据中读取在所述存取单元中的所述经编码的视频数据，并将所读取的数据提供给所述视频解码器，其中

所述使用信息进一步包括表示所述存取单元的数目的数目信息，所述存取单元位于所述使用信息和所述下一个使用信息之间；以及

所述提供部件以与所述使用信息中所包括的所述数目信息所表示的所述数目一样多的次数、基于所述使用信息中所包括的所述大小信息从所述经编码的数据中重复读取在所述存取单元中的所述经编码的视频数据，并且将所读取的数据提供给视频解码器。

2. 如权利要求 1 所述的数据处理设备，其中通过以预定的单元将所述视频数据编码而获得的在所述存取单元中的所述经编码的视频数据为可变长度数据。

3. 一种用于处理经编码的数据的数据处理方法，其中：

所述经编码的数据包括：

通过以预定的单元编码视频数据而获得的、在存取单元中的编码视频数据；

紧邻在所述存取单元中的所述经编码的视频数据中的一个或多个解码可开始点的每一个之前的使用信息，所述使用信息在解码所述经编码的视频数据中使用，并且

所述使用信息包括表示所述存取单元的每一个的大小的大小信息，所述存取单元的数目为一或更大，所述存取单元位于所述使用信息和下一个使用信息之间；和

识别视频流的第一识别信息以及识别音频流的第二识别信息；以及

所述数据处理方法包括：

搜索步骤，用于从所述经编码的数据中搜索所述使用信息；和

提供步骤，用于响应来自用于控制解码所述经编码的视频数据的视频解码器的视频解码器控制部件的请求，基于所述使用信息中所包括的所述大小信息，从所述经编码的数据中读取在所述存取单元中的所述经编码的视频数据，并将所读取的数据提供给所述视频解码器，

其中所述使用信息进一步包括表示所述存取单元的数目的数目信息，所述存取单元位于所述使用信息和所述下一个使用信息之间；以及

所述提供步骤以与所述使用信息中所包括的所述数目信息所表示的所述数目一样多的次数、基于所述使用信息中所包括的所述大小信息从所述经编码的数据中重复读取在所

述存取单元中的所述经编码的视频数据，并且将所读取的数据提供给视频解码器。

数据处理设备和方法

技术领域

[0001] 本发明涉及数据处理设备、数据处理方法、程序、程序记录介质、数据记录介质、和数据结构，且特别地涉及，例如，使得具有高度方便性等等的数据处理成为可能的数据处理设备、数据处理方法、程序、程序记录介质、数据记录介质、和数据结构。

背景技术

[0002] 近年来，作为具有大容量存储器且能够随机存取的记录介质，例如 DVD（数字通用光盘）已经变得很普遍，并且，使用 DVD 来执行各种各样类型的处理的 DVD 设备也已经变得很普遍。

[0003] 例如，DVD 装置包括记录并重放 DVD 上的电视广播节目数据或诸如此类的 DVD 记录器，具有在 DVD 上所记录的地图信息或诸如此类的信息并显示该地图信息的汽车导航系统以及具有在 DVD 上所记录的游戏程序或诸如此类的程序并执行该程序的游戏机。

[0004] 例如，在非专利文献 1 中描述关于 DVD 的详细情况。

[0005] [非专利文献 1] 关于只读光盘的 DVD 规范说明第三部分；版本 1.1，1997 年 12 月。

发明内容

[0006] 本发明将要解决的问题

[0007] 采用能够记录大量的数据的记录介质（例如 DVD）和使用 DVD 执行各种各样类型的处理的 DVD 设备或诸如此类来执行具有高度方便性等等的数据处理是有必要的。

[0008] 本发明是鉴于这样的情况而作出的，且意图使得具有高度方便性等等的数据处理成为可能。

[0009] 用于解决问题的设备

[0010] 本发明的数据处理设备包括搜索部件，用于从包括在存取单元中的编码视频数据和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频数据解码中所使用的使用信息的编码数据中，搜索使用信息；和提供设备，其响应来自视频解码器控制设备的用于控制将编码视频数据解码的视频解码器的请求，基于使用信息中所包括的大小信息，从编码数据中读取在存取单元中的编码视频数据，并将所读取的数据提供视频解码器。

[0011] 本发明的数据处理方法包括搜索步骤，其从包括在存取单元中的编码视频数据和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频数据解码中所使用的使用信息的编码数据中，搜索使用信息；和提供步骤，其响应来自视频解码器控制设备的用于控制将编码视频数据解码的视频解码器的请求，基于使用信息中所包括的大小信息，从编码数据中读取在存取单元中的编码视频数据，并将所读取的数据提供视频解码器。

[0012] 本发明的程序包括搜索步骤，其从包括在存取单元中的编码视频数据和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频

数据解码中所使用的使用信息的编码数据中，搜索使用信息；和提供步骤，其响应来自视频解码器控制设备的用于控制将编码视频数据解码的视频解码器的请求，基于使用信息中所包括的大小信息，从编码数据中读取在存取单元中的编码视频数据，并将所读取的数据提供视频解码器。

[0013] 在本发明的程序记录介质上所记录的程序包括搜索步骤，其从包括在存取单元中的编码视频数据和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频数据解码中所使用的使用信息的编码数据中，搜索使用信息；和提供步骤，其响应来自视频解码器控制设备的用于控制将编码视频数据解码的视频解码器的请求，基于使用信息中所包括的大小信息，从编码数据中读取在存取单元中的编码视频数据，并将所读取的数据提供视频解码器。

[0014] 在本发明的数据记录介质和数据结构中，编码数据包括通过以预定的单元将视频数据编码所获得的在存取单元中的编码视频数据，和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频数据解码中所使用的使用信息，而使用信息包括表示存取单元的每一个的大小的大小信息，存取单元的数目为一或者更大，存取单元位于使用信息和下一个使用信息之间。

[0015] 在本发明的数据处理设备和数据处理方法、程序、和程序（程序记录介质上所记录的）中，从编码数据中搜索使用信息。另外，响应来自视频解码器控制设备的用于控制将编码视频数据解码的视频解码器的请求，基于使用信息中所包括的大小信息，从编码数据中以存取单元的形式读取编码视频数据，并提供视频解码器。

[0016] 在本发明的数据记录介质上所记录的编码数据中，以及在该编码数据的数据结构中，包括通过以预定的单元将视频数据编码所获得的在存取单元中的编码视频数据，和紧邻该在存取单元中的编码视频数据中的一个或多个解码可开始点的每一个之前且在将编码视频数据解码中所使用的使用信息，而使用信息包括表示存取单元的每一个的大小的大小信息，存取单元的数目为一或者更大，存取单元位于使用信息和下一个使用信息之间。

[0017] 优点

[0018] 根据本发明，能够执行具有高度方便性等等的数据处理。特别地，甚至在没有分析经编码的视频数据的情况下，就能够执行对在存取单元中的编码视频数据的读取。

附图说明

[0019] 图 1 是示出本发明应用于其上的盘装置的实施例的硬件配置的框图；

[0020] 图 2 是示出 CPU 112 所执行的软件模块的配置的实例的框图；

[0021] 图 3 是示出缓冲器控制模块 215 的配置的实例的框图；

[0022] 图 4 是盘 101 上的目录的配置的实例的解释说明；

[0023] 图 5 是“PLAYLIST.DAT”文件的语法的解释说明；

[0024] 图 6 是 PlayItem() 的语法的解释说明；

[0025] 图 7 是 PlayListMark() 的语法的解释说明；

[0026] 图 8 是 mark_type 的值和 Mark() 的类型之间的关系的解释说明；

[0027] 图 9 是 PlayList()、PlayItem()、片断、和片断流文件中所存储的节目流之间的关系的解释说明；

[0028] 图 10 是片断信息文件 Clip() 的语法的解释说明；

[0029] 图 11 是定义一个基本流的 stream_id 和 private_stream_id、以及基本流之间的关系的解释说明；

[0030] 图 12 是 StaticInfo() 的语法的解释说明；

[0031] 图 13 是 DynamicInfo() 的语法的解释说明；

[0032] 图 14 是 EP_map() 的语法的解释说明；

[0033] 图 15 是 MPEG-2 系统中的节目流、节目流包、和节目流包头的语法的解释说明；

[0034] 图 16 是 MPEG-2 系统中的 PES 信息包的语法的解释说明；

[0035] 图 17 是 PES 信息包的语法的解释说明；

[0036] 图 18 是 MPEG-2 系统中的 PES 信息包的语法的解释说明；

[0037] 图 19 是在 MPEG-2 系统中的 PES 信息包的 stream_id 中所描述的值和基本流的属性（类型）之间的关系的解释说明；

[0038] 图 20 是盘装置所采用的 stream_id 流的解释说明；

[0039] 图 21 是 private_stream1_PES_payload() 的语法的解释说明；

[0040] 图 22 是 private_stream_id 的值和在 private_payload() 中所存储的基本流的属性之间的关系的解释说明；

[0041] 图 23 是 private_stream2_PES_payload() 的语法的解释说明；

[0042] 图 24 是 au_information() 的语法的解释说明；

[0043] 图 25 是“PLAYLIST.DAT”文件的特定实例的解释说明；

[0044] 图 26 是片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”的特定实例的解释说明；

[0045] 图 27 是片断信息文件“00001.CLP”中的 EP_map() 的特定实例的解释说明；

[0046] 图 28 是 PlayList#0 和 PlayList#1 中的 PlayListMark() 项的特定实例的解释说明；

[0047] 图 29 是阐明重放之前的处理的流程图；

[0048] 图 30 是阐明重放处理的流程图；

[0049] 图 31 是阐明 PlayItem 改变处理的流程图；

[0050] 图 32 是阐明时间代码显示过程的流程图；

[0051] 图 33 是阐明流切换处理的流程图；

[0052] 图 34 是阐明缓冲器控制模块 215 的处理的流程图；

[0053] 图 35 是阐明缓冲器控制模块 215 的处理的流程图；

[0054] 图 36 是阐明视频流读取过程的流程图；

[0055] 图 37 是阐明音频流读取过程的流程图；

[0056] 图 38 是阐明字幕流读取过程的流程图；

[0057] 图 39 是阐明再同步处理的流程图；

[0058] 图 40 是阐明标记处理的流程图；

[0059] 图 41 是阐明输出属性控制处理的流程图；

[0060] 图 42 是片断信息文件“00003.CLP”中所描述的 pts_change_point 和 DynamicInfo() 项的组的特定实例的解释说明；

[0061] 图 43 是阐明字幕显示控制处理的流程图；

[0062] 图 44 由分别阐明捕获控制处理和背景 / 屏幕保护处理的流程图组成；

[0063] 图 45 是 private_stream2_PES_payload() 的其它语法的解释说明；

[0064] 图 46 是 au_information() 的其它语法的解释说明。

[0065] 附图标记

[0066] 101 盘

[0067] 102 盘驱动器

[0068] 111 总线

[0069] 112 CPU

[0070] 113 存储器

[0071] 114 驱动器接口

[0072] 115 输入接口

[0073] 116 视频解码器

[0074] 117 音频解码器

[0075] 118 视频输出接口

[0076] 119 音频输出接口

[0077] 120 视频输出终端

[0078] 121 音频输出终端

[0079] 201 操作系统

[0080] 210 视频内容重放程序

[0081] 211 脚本控制模块

[0082] 212 播放器控制模块

[0083] 213 内容数据提供模块

[0084] 214 解码控制模块

[0085] 214A 定时器单元

[0086] 215 缓冲器控制模块

[0087] 215A 缓冲器

[0088] 216 视频解码器控制模块

[0089] 217 音频解码器控制模块

[0090] 218 字幕解码器控制模块

[0091] 219 图形处理模块

[0092] 220 视频输出模块

[0093] 220A FIFO

[0094] 221 音频输出模块

[0095] 221A FIFO

[0096] 231 数据开始指针存储单元

[0097] 232 数据写入指针存储部分

[0098] 233 视频读取功能单元

[0099] 234 音频读取功能单元

[0100] 235 字幕读取功能单元

[0101] 241 视频读取指针存储部分

[0102] 242 stream_id 寄存器

[0103] 243 an_information() 寄存器

[0104] 251 音频读取指针存储部分

[0105] 252 stream_id 寄存器

[0106] 253 private_stream_id 寄存器

[0107] 261 字幕读取功能部分

[0108] 262 字幕读取指针存储部分

[0109] 263 stream_id 寄存器

[0110] 264 private_stream_id 寄存器

具体实施方式

[0111] 下文参考绘图描述本发明的实施例

[0112] 硬件配置

[0113] 图 1 是示出本发明应用于其上的盘装置的实施例的硬件的实例的框图。

[0114] 图 1 中的盘装置能够应用于，例如，盘播放器、游戏机、汽车导航系统等等。

[0115] 在图 1 中的盘装置中，例如，盘 101 为像 DVD 那样的光盘、磁光盘、磁盘或诸如此类，且包含像视频数据、音频数据、和字幕数据那样的内容数据，以及用于播放内容数据必需的标识数据。

[0116] 如果需要，盘 101 上所记录的数据（记录数据）包括计算机可执行程序。同样，尽管，在这个实施例中，采用作为盘记录介质的盘 101 作为记录介质，但是，记录介质可以为，例如，半导体存储器、或者磁带记录介质。另外，能够将在从位于远程位置的盘 101 中读取之后所传输的数据输入到图 1 中的盘装置。换句话说，能够通过连接到盘装置的不同的装置来执行从盘 101 的数据的读取，且能够通过该盘装置接收和处理由该不同的装置所读取的数据。另外，通过包括存储类似于在盘 101 上所记录的数据的数据的存储器的服务器或诸如此类，该盘装置能够处理在网络（例如因特网）上所分布的数据。而且，在从服务器或另一个装置接收到数据且暂时在盘 101 上记录了该数据之后，盘装置能够处理盘 101 上所记录的数据。

[0117] 在盘驱动器 102 中，能够加载和卸载盘 101。盘驱动器 102 具有一个内置接口（未示出），并通过该接口连接到驱动器接口 114。当将盘 101 加载到那里时，盘驱动器 102 驱动盘 101。根据来自驱动器接口 114 的命令（例如读取），盘驱动器 102 执行像从盘 101 读取数据和将该数据提供驱动器接口 114 那样的处理。

[0118] 总线 111 与 CPU（中央处理单元）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、和音频输出接口 119 连接。

[0119] CPU 112 和存储器 113 构成计算机系统。换句话说，CPU 112 通过执行稍后所描述的、在存储器 113 中所存储的程序的软件模块，而控制整个盘装置，并执行稍后所描述的各种各样类型的处理。存储器 113 暂时存储将要由 CPU112 执行的软件模块。存储器 113 暂时存储用于操作 CPU 112 所必需的数据。能够仅由非易失性存储器或由易失性存储器和非

易失性存储器的结合而构成存储器 113。另外，当图 1 中的盘装置提供有硬盘，且在该硬盘上记录（安装）了将要由 CPU 112 执行的软件模块时，能够仅由非易失性存储器而构成存储器 113。

[0120] 这里，能够在存储器 113 中预先记录（存储）将要由 CPU 112 执行的程序（软件模块），存储器 113 为盘装置的一个内嵌记录介质。

[0121] 可选择地，能够在盘 101 上，以及另外在像除盘 101 之外的软盘、CD-ROM（压缩光盘只读存储器）、MO（磁光）光盘、磁盘、和存储卡那样的可移动记录介质上暂时或永久地存储（记录）程序。这些可移动记录介质能够作为所谓的包组软件而提供。

[0122] 能够在存储器 113 中预先存储程序，或者如上文所描述的能够从可移动记录介质将该程序安装到盘装置。另外，能够通过用于数字卫星广播的卫星，从下载站点将该程序无线地传送至盘装置，以及能够通过像因特网那样的网络从下载站点将该程序有线地传送至盘装置。盘装置能够通过使用输入接口 115 接收程序，并能够在存储器 113 中安装该程序。

[0123] 而且，可以由单个 CPU 处理该程序，以及可以由多个 CPU 以分布式的形式处理该程序。

[0124] 在 CPU 112 的控制下，驱动器接口 114 控制盘驱动器 102，由此通过总线 111 将由盘驱动器 102 从盘 101 所读取的数据提供 CPU 112、存储器 113、视频解码器 116、音频解码器 117 等等。

[0125] 输入接口 115 接收因为用户操作未示出的键盘（按钮）和遥控器而提供的信号，并通过总线 111 将该信号提供 CPU 112。另外，输入接口 115 也具备，例如，调制解调器（包括 ADSL（非对程数字用户线路）调制解调器）和像 NIC（网络接口卡）那样的通信接口的功能。

[0126] 视频解码器 116 将由盘驱动器 102 从盘 101 所读取的，且通过驱动器接口 114 和总线 111 而提供的视频数据的编码数据（编码视频数据）解码，并通过总线 111 将作为结果的视频数据提供给 CPU 112 和视频输出接口 118。

[0127] 音频解码器 117 将由盘驱动器 102 从盘 101 所读取的，且通过驱动器接口 114 和总线 111 而提供的音频数据的编码数据（编码音频数据）解码，并通过总线 111 将作为结果的音频数据提供给 CPU 112 和音频输出接口 119。

[0128] 视频输出接口 118 对通过总线 111 而提供的视频数据执行必需的处理，并从视频输出终端 120 输出已处理过的数据。音频输出接口 119 对通过总线 111 而提供的音频数据执行必需的处理，并从音频输出终端 121 输出已处理过的数据。

[0129] 视频输出终端 120 连接到未示出的视频输出装置，例如 CRT（阴极射线管）或液晶面板。相应地，提供从视频输出终端 120 所输出的视频数据并将其显示在视频输出装置上。音频输出终端 121 连接到未示出的音频输出装置，例如扬声器或扩音器。相应地，提供从音频输出终端 121 所输出的音频数据并由音频输出装置输出。

[0130] 能够或者有线或者无线地执行对视频输出装置和音频输出装置的视频数据和音频数据的提供。

[0131] 软件模块的配置

[0132] 接下来，图 2 示出图 1 中的 CPU 112 执行的软件模块的配置的实例。

[0133] CPU 112 执行的软件模块广泛地分为操作系统 (OS)201 和为应用程序的视频内容

重放程序 210。

[0134] “操作系统 201”

[0135] 当给盘装置提供电力时，操作系统 201 最初开始 (CPU 112 执行操作系统 201) 执行必需的处理，例如初始设置，并调用为应用程序的视频内容重放程序 210。

[0136] 操作系统 201 给视频内容重放程序 210 提供像文件读取那样的基础架构服务。特殊地，例如关于文件读取，响应来自视频内容重放程序 210 的文件读取请求，操作系统 201 提供通过使用驱动器接口 114 操作盘驱动器 102 读取盘 101 上的数据，并将该数据传递至视频内容重放程序 210 的服务。操作系统 201 也执行文件系统的解释等等。

[0137] 操作系统 201 具有多任务处理功能，且能够使得多个软件模块以时分的方式（表面上看来）同时地操作。换句话说，视频内容重放程序 210 包括一些软件模块，且这些软件模块能够并行地操作。

[0138] “视频内容重放程序 210”

[0139] 视频内容重放程序 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、字幕解码器控制模块 218、图形处理模块 219、视频输出模块 220、和音频输出模块 221。

[0140] 视频内容重放程序 210 是在播放盘 101 中起主要作用，且当盘 101 加载（插入）到盘驱动器 102 中时，确认盘 101 是否具有以稍后所描述的格式记录的内容的软件。在从盘 101 读取稍后所描述的脚本文件之后，视频内容重放程序 210 执行该文件。另外，视频内容重放程序 210 从盘 101 中读取用于播放盘 101 上所记录的内容必需的元数据（数据库信息）的文件，并在该元数据的基础上控制内容播放。

[0141] 在下文描述构成图 2 中的视频内容重放程序 210 的软件模块。在图 2 中，通常，实线箭头指示内容数据，而虚线箭头指示控制数据。

[0142] “脚本控制模块 211”

[0143] 脚本控制模块 211 解释并执行盘 101 上所记录的脚本文件中所描述的脚本程序（脚本）。在脚本程序中，例如，能够描述操作，如“通过操作图形处理模块 219，创建并显示图形，例如菜单”、“根据来自像遥控器那样的 UI（用户接口）的信号（例如，在菜单上移动光标等等），改变菜单显示”、和“控制播放器控制模块 212”。

[0144] “播放器控制模块 212”

[0145] 播放器控制模块 212 通过参考盘 101 上所记录的元数据（数据库信息）等等，执行关于内容重放的控制。换句话说，播放器控制模块 212 分析，例如，稍后所描述的，盘 101 上所记录的 PlayList() 和 Clip()，并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。根据来自脚本控制模块 211 和输入接口 115 的指令，播放器控制模块 212 也执行像稍后所描述的在其中切换将要重放的流的流切换那样的控制。另外，播放器控制模块 212 从解码控制模块 214 获取时间，并执行显示时间、标记（稍后所描述的 Mark()）处理等等。

[0146] “内容数据提供模块 213”

[0147] 在播放器控制模块 212 的控制下，或者基于缓冲器控制模块 215 中所存储的大量数据，内容数据提供模块 213 请求操作系统 201 从盘 101 读取内容数据、元数据等等。

[0148] 将由操作系统 201 响应来自内容数据提供模块 213 的请求、从盘 101 所读取的元数据等等提供给必需的模块。另外，将由操作系统 201 响应来自内容数据提供模块 213 的请求、从盘 101 所读取的内容数据提供给缓冲器控制模块 215。

[0149] “解码控制模块 214”

[0150] 在播放器控制模块 212 的控制下，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、和字幕解码器控制模块 218 的操作。另外，解码控制模块 214 具有用于时间度量的内嵌定时器单元 214A。解码控制模块 214 管理将要在视频解码器控制模块 216 的控制下而输出的输出数据和将要与视频数据同步而输出的输出数据（所输出的数据），即将要在音频解码器控制模块 217 的控制下而输出输出音频数据，之间的同步。

[0151] “缓冲器控制模块 215”

[0152] 缓冲器控制模块 215 具有一个内嵌缓冲器 215A，其为图 1 中存储器 113 的存储区域的一部分。通过由内容数据提供模块 213 请求操作系统 201，将从盘 101 所读取的内容数据暂时地存储在缓冲器 215A 中。

[0153] 另外，根据来自视频解码器控制模块 216、音频解码器控制模块 217、或字幕解码器控制模块 218 的请求，缓冲器控制模块 215 将缓冲器 215A 中所存储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、或字幕解码器控制模块 218。

[0154] 特殊地，缓冲器控制模块 215 具有内嵌视频读取功能单元 233、音频读取功能单元 234、和字幕读取功能单元 235，其稍后将参考图 3 描述。通过使用视频读取功能单元 233 处理来自视频解码器控制模块 216 的数据请求，缓冲器控制模块 215 将缓冲器 215A 中所存储的数据提供视频解码器控制模块 216。类似地，缓冲器控制模块 215 通过使用音频读取功能单元 234 处理来自音频解码器控制模块 217 的数据请求，将缓冲器 215A 中所存储的数据提供给音频解码器控制模块 217，并通过使用字幕读取功能单元 235 处理来自字幕解码器控制模块 218 的数据请求，将存储在缓冲器 215A 中的数据提供给字幕解码器控制模块 218。

[0155] “视频解码器控制模块 216”

[0156] 通过操作缓冲器控制模块 215 中的视频读取功能单元 233（图 3），视频解码器控制模块 216 从缓冲器控制模块 215 中的缓冲器 215A 中读取在视频存取单元中的、其中将视频数据编码的数据（编码视频数据），并将该数据提供给图 1 中的视频解码器 116。视频解码器控制模块 216 也控制视频解码器 116 将在视频存取单元中的数据解码。另外，视频解码器控制模块 216 将通过在视频解码器 116 中解码而获得的视频数据提供给图形处理模块 219。

[0157] 这里，视频存取单元相应于，例如，视频数据的一画面（一帧或一个场）。

[0158] “音频解码器控制模块 217”

[0159] 通过操作缓冲器控制模块 215 中的音频读取功能单元 234（图 3），音频解码器控制模块 217 从缓冲器控制模块 215 中的缓冲器 215A 中读取在其中已将音频数据编码的数据（编码音频数据），并将该数据提供给音频解码器 117。音频解码器控制模块 217 也控制音频解码器 117 将在音频存取单元中存在的数据解码。另外，音频解码器控制模块 217 将通过在音频解码器 117 中解码而获得的音频数据提供给音频输出模块 221。

[0160] 这里，音频存取单元相应于预定数量（例如，与一个画面同步而输出的量）的音频数据。在这个实施例中，音频存取单元为，例如，一个已知固定的长度。

[0161] “字幕解码器控制模块 218”

[0162] 通过操作缓冲器控制模块 215 中的字幕读取功能单元 235(图 3),字幕解码器控制模块 218 从缓冲器控制模块 215 中的缓冲器 215A 中,读取在字幕存取单元中的、通过将字幕数据编码而获得的数据(编码字幕数据)。字幕解码器控制模块 218 也包括未示出的字幕解码软件,并将从缓冲器 215A 中所读取的数据解码。另外,字幕解码器控制模块 218 将通过解码而获得的字幕数据(字幕的图像数据)提供给图形处理模块 219。

[0163] 这里,一个字幕存取单元相应于预定数量(例如,与一个画面同步而输出的量)的字幕数据。在这个实施例中,例如,在字幕存取单元的开始描述字幕存取单元的大小。

[0164] “图形处理模块 219”

[0165] 在图形处理模块 219 中,根据播放器控制模块 212 的控制(指令),放大并缩小来自字幕解码器控制模块 218 的字幕数据,并将所获得的数据添加到(覆盖在)来自视频解码器控制模块 216 的视频数据上。另外,为了在添加了字幕数据之后所获得的视频数据的大小(图像帧)可以与视频输出装置上的显示屏相匹配,图形处理模块 219 执行放大、缩小、或诸如此类,并将作为结果的视频数据输出到视频输出模块 220。

[0166] 在图形处理模块 219 中,根据脚本控制模块 211 和播放器控制模块 212 的控制(指令),产生菜单、消息等等,并覆盖在输出视频数据上。

[0167] 而且,基于连接到图 1 中的视频输出终端 120 的视频输出装置的屏幕高宽比、盘 101 上所记录的且指定视频数据的屏幕高宽比的信息,图形处理模块 219 转换将要输出至视频输出模块 220 的视频数据的屏幕高宽比。

[0168] 特殊地,例如,当视频输出装置的屏幕高宽比为 16 ∶ 9,且指定视频数据的屏幕高宽比的信息表示 4 ∶ 3 的屏幕高宽比时,在图形处理模块 219 中,在横向方向(水平方向)挤压(缩小)将要输出至视频输出模块 220 的视频数据,并输出所获得的在其中右边和左边部分已变暗的数据。另外,例如,当视频输出装置的屏幕高宽比为 4 ∶ 3,且指定视频数据的屏幕高宽比的信息表示 16 ∶ 9 的屏幕高宽比时,在图形处理模块 219 中,在纵向方向(垂直方向)挤压将要输出至视频输出模块 220 的视频数据,且顶部和底部部分变暗,并输出已处理过的数据。

[0169] 当视频输出装置的屏幕高宽比和由表示视频数据的屏幕高宽比的信息所表示的屏幕高宽比两者都为 4 ∶ 3 或 16 ∶ 9 的相同的屏幕高宽比时,图形处理模块 219 直接输出将要输出至视频输出模块 220 的视频数据,而不执行挤压处理。

[0170] 另外,响应来自,例如,播放器控制模块 212 的请求,图形处理模块 219 捕获正被处理的视频数据。此外,图形处理模块 219 将所捕获的视频数据存储或提供给播放器控制模块 212。

[0171] “视频输出模块 220”

[0172] 视频输出模块 220 专有地占据图 1 中的存储器 113 的一部分,以用作 FIFO(先进先出)220A(缓冲器);暂时地存储来自图形处理模块 219 的视频数据;且如果需要,读取 FIFO 220A 中所存储的视频数据;并将所读取的数据输出至视频输出终端 120(图 1)。

[0173] “音频输出模块 221”

[0174] 音频输出模块 221 专有地占据图 1 中的存储器 113 的一部分,以用作 FIFO221A(缓冲器);暂时地存储来自音频解码器控制模块 217(音频解码器 117)的音频数据;且如果需

要，读取 FIFO 221A 中所存储的音频数据；并将所读取的数据输出至音频输出终端 121（图 1）。

[0175] 另外，当来自音频解码器控制模块 217 的音频数据为在其中由左声道表示“主音频”的音频数据，以及右声道表示“次音频”的音频数据的双声道（两声道）模式的音频数据时，根据预先指定的音频输出模式，音频输出模块 221 将来自音频解码器控制模块 217 的音频数据输出至音频输出终端 121。

[0176] 特殊地，当，例如，指定“主音频”为音频输出模式时，音频输出模块 221 复制来自音频解码器控制模块 217 的音频数据中的左声道音频数据作为右声道音频数据，并将左声道和右声道的音频数据（“主音频”的音频数据）输出至音频输出终端 121。另外，当指定“次音频”为音频输出模式时，音频输出模块 221 复制来自音频解码器控制模块 217 的音频数据中的右声道音频数据作为左声道音频数据，并将左声道和右声道的音频数据（“次音频”的音频数据）输出至音频输出终端 121。此外，当指定“主和次音频”为音频输出模式时，音频输出模块 221 直接将来自音频解码器控制模块 217 的音频数据输出至音频输出终端 121。

[0177] 当来自音频解码器控制模块 217 的音频数据为立体声模式的音频数据时，音频输出模块 221 直接将来自音频解码器控制模块 217 的音频数据输出至音频输出终端 121，而不管音频输出模式的指定。

[0178] 这里，例如，能够通过用户在显示由视频内容重放程序 210 所产生的菜单的屏幕上操作遥控器或诸如此类，而交互式地执行音频输出模式的指定。

[0179] “缓冲器控制模块 215 的配置”

[0180] 接下来，图 3 示出图 2 中的缓冲器控制模块 215 的配置的实例。

[0181] 缓冲器控制模块 215 专有地使用图 1 中的存储器 113 的一部分，以暂时地将从盘 101 所读取的数据存储在缓冲器 215A 中。缓冲器控制模块 215 读取缓冲器 215A 中所存储的数据，并将所读取的数据供给图 2 中的视频解码器控制模块 216、音频解码器控制模块 217、或字幕解码器控制模块 218。

[0182] 换句话说，除缓冲器 215A 之外，缓冲器控制模块 215 还包括为存储器 113 的一部分的数据开始指针存储单元 231 和数据写入指针存储部分 232，并具有视频读取功能单元 233、音频读取功能单元 234、和字幕读取功能单元 235 作为内部模块。

[0183] 缓冲器 215A 为，例如，循环缓冲器，并顺序地存储从盘 101 所读取的数据。在其存储数据到其存储容量之后，其存储最新的数据，以便以所谓的无限循环的形式覆盖最旧的数据。

[0184] 在缓冲器 215A 中所存储的数据中，将指示一位置（地址）的数据开始指针存储在数据开始指针存储单元 231 中，从缓冲器 215A 中，还未在该位置（地址）处读取最旧的数据。

[0185] 数据写入指针存储部分 232 存储指示缓冲器 215A 中一位置（地址）的写入指针，在该位置（地址）处写入从盘 101 所读取的最新的数据。

[0186] 这里，无论何时在缓冲器 215A 中存储从盘 101 所读取的数据，在图 3 中，都右旋（顺时针方向）地更新由数据写入指针所指示的位置，且响应从缓冲器 215A 的数据的读取，在图 3 中，右旋地更新由数据开始指针所指示的位置。因此，在缓冲器 215A 中所存储的数

据中，所谓有效的数据为从由数据开始指针所指示的位置至由数据写入指针所指示的位置右旋地存储的数据。

[0187] 视频读取功能单元 233 响应来自图 2 中的视频解码器控制模块 216 的请求从缓冲器 215A 中读取视频流（关于视频数据的基本流），并将所读取的流供给视频解码器控制模块 216。同样音频读取功能单元 234 响应来自音频解码器控制模块 217 的请求，从缓冲器 215A 中读取音频流（关于音频数据的基本流），并将所读取的流供给音频解码器控制模块 217。同样字幕读取功能单元 235 从缓冲器 215A 中读取字幕流（关于字幕数据的基本流），并将所读取的流供给字幕解码器控制模块 218。

[0188] 特殊地，在盘 101 上，例如，记录根据 MPEG（移动画面专家组）-2 的标准的节目流，并且缓冲器 215A 存储从盘 101 所读取的节目流。在这个节目流中，将一个或多个基本流，例如视频流、音频流、和字幕流，时分地多路复用。视频读取功能单元 233 具有将节目流去多路复用的功能，并从缓冲器 215A 中所存储的节目流中分离地读取视频流。

[0189] 类似地，同样音频读取功能单元 234 具有将节目流去多路复用的功能，并从缓冲器 215A 中所存储的节目流中分离地读取音频流。同样字幕读取功能单元 235 具有将节目流去多路复用的功能，并从缓冲器 215A 中所存储的节目流中分离地读取字幕流。

[0190] 这里，视频读取功能单元 233 包括视频读取指针存储部分 241、stream_id 寄存器 242、和 an_information() 寄存器 243，其为图 1 中存储器 113 的一部分。

[0191] 视频读取指针存储部分 241 存储指示缓冲器 215A 中在其存储视频流的位置（地址）的视频读取指针，以及视频读取功能单元 233 读取在由视频读取指针所指示的位置所存储的数据作为视频流。stream_id 寄存器 242 分析寄存器 215A 中所存储的节目流，并存储稍后所描述的用于识别将要从节目流读取的视频流的 stream_id。an_information() 寄存器 243 存储稍后所描述的用于从缓冲器 215A 读取视频流所必需的（用于读取视频流所利用的）au_information()。

[0192] 音频读取功能单元 234 包括音频读取指针存储部分 251、stream_id 寄存器 252、和 private_stream_id 寄存器 253，其为图 1 中存储器 113 的一部分。

[0193] 缓冲器控制模块 215 存储指示缓冲器 215A 中在其存储音频流的位置（地址）的音频读取指针，并且音频读取功能单元 234 读取在由音频读取指针所指示的位置所存储的数据作为音频流。stream_id 寄存器 252 和 private_stream_id 寄存器 253 分析缓冲器 215A 中所存储的节目流，并分别存储稍后所描述的用于识别将要从节目流读取的音频流的 stream_id 和 private_id。

[0194] 字幕读取功能单元 235 包括字幕读取功能部分 261、字幕读取指针存储部分 262、stream_id 寄存器 263、和 private_stream_id 寄存器 264、其为图 1 中的存储器 113 的一部分。

[0195] 字幕读取功能部分 261 存储字幕读取功能标志。当字幕读取功能部分 261 中所存储的字幕读取功能标志为，例如，0 时，字幕读取功能单元 235 不起作用，以及当字幕读取功能部分 261 中所存储的字幕读取功能标志为，例如，1 时，字幕读取功能单元 235 起作用。

[0196] 字幕读取功能部分 261 存储指示缓冲器 215A 中在其存储字幕流的位置（地址）的字幕读取指针，并且字幕读取功能单元 235 读取在由字幕读取指针所指示的位置所存储的数据作为字幕流。stream_id 寄存器 263 和 private_stream_id 寄存器 264 分析缓冲器

215A 中所存储的节目流，并分别存储稍后所描述的用于识别将要从节目流读取的字幕流的 stream_id 和 private_stream_id。

[0197] 盘 101 上所记录的数据的数据格式的描述

[0198] 其次，描述盘 101 上所记录的数据的数据格式。

[0199] 图 4 示出盘 101 的目录结构

[0200] 例如，将 ISO（国际标准化组织）-9660、UDF（通用磁盘格式 :http://www.osta.org/specs/）、或诸如此类中所供给的文件系统用作盘 101 的文件系统，且以目录结构分层级地管理盘 101 上所记录的数据的文件。这里，文件系统并不局限于上述文件系统。

[0201] 在图 4 中，在表示文件系统源的根目录中，设置有“VIDEO”目录，且，在“VIDEO”目录中，设置有两个目录，“CLIP”目录和“STREAM”目录。

[0202] 在“VIDEO”目录中，除两个目录，“CLIP”目录和“STREAM”目录之外，还设置有两个数据文件，“SCRIPT.DAT”文件和“PLAYLIST.DAT”文件。

[0203] “SCRIPT.DAT”文件是一个脚本文件，在其中描述脚本程序。特殊地，在“SCRIPT.DAT”文件中，描述用来使盘 101 的播放形式交互作用的脚本程序。“SCRIPT.DAT”文件中所描述的脚本程序由图 2 中的脚本控制模块 211 解释并执行。

[0204] “PLAYLIST.DAT”文件存储至少一个播放列表（稍后描述的图 5 中的 PlayList()），在播放列表中描述播放像盘 101 上所记录的视频数据那样的内容的过程。

[0205] 在“CLIP”目录中，设置有一个或多个片断信息文件，并且，在“STREAM”目录中，设置有一个或多个片断流文件。换句话说，在图 4 中，在“CLIP”目录中，设置有三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”，且，在“STREAM”目录中，设置有三个片断流文件“00001.PS”、“00002.PS”、和“00003.PS”。

[0206] 片断流文件存储一个节目流，在该节目流中将通过将一种或多种类型的数据（流），例如视频数据、音频数据、和字幕数据，压缩或编码而获得的一个或多个基本流时分地多路复用。

[0207] 在片断信息文件中，描述关于片断流的（文件）元数据，例如相应片断流文件的性质。

[0208] 换句话说，片断流和片断信息文件具有一对一的对应。在图 4 中，根据五个数字 + 句点 + “PS”的命名规则给片断流文件分配文件名，并根据与相应片断流文件的那些相同的五个数字 + 句点 + “CLP”的命名规则给片断信息文件分配文件名。

[0209] 相应地，关于一个文件是片断流文件和片断信息文件的哪一个，能够通过文件名的扩展名（从句点起的右边部分）来识别该文件。另外，关于相应的片断流文件和片断信息文件，能够依赖于除文件名的扩展名之外的部分（从句点起的左边部分）是否相匹配而识别它们。

[0210] 下文描述盘 101 上所记录的文件的详细情况。

[0211] “PLAYLIST.DAT”

[0212] 图 5 示出“VIDEO”目录中“PLAYLIST.DAT”文件的内部结构（语法）。

[0213] 这里，在图 5 中，“语法”域中的描述表示数据结构。“位数”域中的描述表示“语法”域的相应行中的数据的位长。另外，“记忆符号”域中的“bslbf”（位串左位优先）指示从最左边的位开始提供“语法”域中相应行中的数据，以及“uimsbf”（无符号整数最高有效位

优先）指示“语法”域中相应行中的数据为无符号整数值，且从最高有效位开始提供。这也适用于下文所描述的类似于图 5 的绘图。

[0214] 在“PLAYLIST.DAT”文件中，从其开始起，顺序地设置了用于写入像其名称（文件名）那样的信息的 name_length(8 位）和 name_string(255 字节）。

[0215] 换句话说，name_length 以字节数表示紧接着它的 name_string 的大小。name_string 表示“PLAYLIST.DAT”文件的名称（文件名）。

[0216] 关于 name_string，使用从其开始起的由 name_length 所表示的字节的数目的一部分作为有效名称。例如，当 name_length 具有为 10 的值时，从其开始起延伸 10 个字节的 name_string 的一部分被解释为有效名称。

[0217] 紧接着 name_string，设置了 number_of_PlayLists(16 位）。number_of_PlayLists 表示接下来的 PlayList() 项的数目。在 number_of_PlayLists 之后，设置了与 number_of_PlayLists 中的数目一样多的 PlayList() 项。

[0218] PlayList() 是一个播放列表，在其中描述播放盘 101 上所记录的片断流文件的过程，并具有下列内部结构。

[0219] 特殊地，在PlayList() 的开始，设置了PlayList_data_length(32位）。PlayList_data_length 表示 PlayList() 的大小。

[0220] 紧接着PlayList_data_length，顺序地设置了reserved_for_word_alignment(15 位）和 capture_enable_flag_PlayList(1 位）。为了建立文字对齐（为了在 16 位的位置执行对齐），在随后所设置的 1 位 capture_enable_flag_PlayList 的位置设置了 15 位的 reserved_for_word_alignment。Capture_enable_flag_PlayList 为 1 位标志，其指示是否允许在其中重放盘 101 的图 1 中的盘装置中，相应于基于该 PlayList() 而重放的视频流的视频数据（属于该 PlayList() 的视频数据）的二次使用 (secondary use)。当 capture_enable_flag_PlayList 为，例如，0 和 1 之间的 1 时，其指示允许属于该 PlayList() 的视频数据的二次使用，且当 capture_enable_flag_PlayList 为，例如，0 和 1 之间的 0 时，其指示不允许（禁止）属于该 PlayList() 的视频数据的二次使用。

[0221] 尽管，在图 5 中，capture_enable_flag_PlayList 由一位构成，通过使用多位构成 capture_enable_flag_PlayList，能够在所谓的阶段中允许属于该 PlayList() 的视频数据的二次使用。特殊地，capture_enable_flag_PlayList 能够由，例如，两位构成。当 capture_enable_flag_PlayList 的值为 00B(B 指示在其之前的数字为二进制数）时，能够禁止视频数据的二次使用。当 capture_enable_flag_PlayList 的值为 01B 时，仅能够在以减小到等于或小于 64×64 像素的大小的形式使用视频数据中允许二次使用。另外，当 capture_enable_flag_PlayList 的值为 10B 时，能够在不限制大小的情况下允许视频数据的二次使用。

[0222] 此外，如上文所描述的，在视频数据的二次使用中，能够在不限制大小的情况下限制该使用。特殊地，当 capture_enable_flag_PlayList 的值为 01B 时，仅视频内容重放程序 210(图 2) 能够允许二次使用。当 capture_enable_flag_PlayList 的值为 10B 时，能够允许通过包括图 1 中的盘装置中的视频内容重放程序 210 的任意应用程序的二次使用。这里，除图 1 中的盘装置中的视频内容重放程序 210 之外的应用程序包括，例如，用于显示墙纸（背景）和屏幕保护程序的应用程序。

[0223] 如上文所描述的，当 capture_enable_flag_PlayList 由两位构成时，位于其之前的 reserved_for_word_alignment 具有 14 位，用于建立文字对齐。

[0224] 除盘装置中视频数据的二次使用的允许之外，capture_enable_flag_PlayList 还能够允许盘装置外的二次使用。这里，当允许盘装置外的视频数据的二次使用时，视频数据记录在，例如，能够加载到该盘装置中的记录介质和能够加载到可连接到该盘装置的另一个装置中的记录介质中，或者通过像因特网那样的网络传输（分送）到另一个装置。在这种情形，能够将视频数据在记录介质上所记录的次数的数目和限制分送该视频数据的次数的数目的信息添加到该视频数据。

[0225] 紧接着 capture_enable_flag_PlayList，顺序地设置了 PlayList_name_length(8 位）和 PlayList_name_string(255 字节）。PlayList_name_length 使用字节数表示随后所设置的 PlayList_name_string 的大小，且 PlayList_name_string 表示 PlayList() 的名称。

[0226] 紧接着 PlayList_name_string，设置了 number_of_PlayItems(16 位）。像素的数目表示接下来的 PlayItem() 项的数目。

[0227] 紧接着 number_of_PlayItems，描述了与 number_of_PlayItems 中的数目一样多的 PlayItem() 项的结构。

[0228] 这里，在一个 PlayList() 项中，能够以 PlayItem() 单元描述播放内容的过程。

[0229] 另外，在 PlayList() 中，对于与 number_of_PlayItems 中的数目一样多的 PlayItem() 项的每一个，分配在 PlayList() 中的唯一 ID（识别号）。特殊地，0 分配给 PlayList() 中第一个 PlayItem() 作为 ID，且 ID 序列号，例如 1、2、...，以它们出现的顺序分配给接下来的 PlayItem() 项。

[0230] 紧接着与 number_of_PlayItems 中的数目一样多的 PlayItem() 项，设置了一个 PlayListMark() 项。PlayListMark() 为充当在根据 PlayList() 而执行的重放的时间基础上的标记的一组稍后描述的 Mark()，且稍后参考图 7 描述其详细情况。

[0231] “PlayItem() 的描述”

[0232] 接下来，图 6 示出图 5 中的 PlayList() 中所包括的 PlayItem() 的内部结构。

[0233] 在 PlayItem() 的开始，设置了长度 (16 位），且该长度表示包括它本身的 PlayItem() 的大小。

[0234] 紧接着该长度，顺序地设置了 Clip_Information_file_name_length(16 位）和 Clip_Information_file_name（可变长度）。

[0235] Clip_Information_file_name_length 使用字节数表示随后设置的 Clip_Information_file_name 的大小。Clip_Information_file_name 表示相应于将要由 PlayList() 重放的片断流文件（图 4 中其扩展名为 PS 的文件）的片断信息文件（图 4 中其扩展名为 CLP 的文件）的文件名。基于上文所描述的关于片断流文件和片断信息文件的文件名的命名规则，能够辨识将要由 PlayItem() 重放的片断信息文件的文件名，且能够指定片断流文件。

[0236] 紧接着 Clip_Information_file_name，顺序地设置了 IN_time(32 位）和 OUT_time(32 位）。

[0237] IN_time 和 OUT_time 分别为在由 Clip_Information_file_name 所指定的片断流

文件中，指明重放开始位置的时间信息和指明重放结束位置的时间信息。

[0238] 根据 IN_time，能够指明在片断流文件中间（包括其开始）的一个位置作为重放开始位置，且根据 OUT_time，能够指明在片断流文件中间（包括其结束）的一个位置作为重放结束位置。

[0239] 这里，根据 PlayItem()，重放由 Clip_Information_file_name 所指定的片断流文件的 IN_time 和 OUT_time 之间的内容。如果需要，由 PlayItem() 所重放的内容以下称为片断。

[0240] “PlayListMark() 的描述”

[0241] 接下来，图 7 示出图 5 中的 PlayList() 中所包括的 PlayListMark() 的内部结构。

[0242] 如上文所描述的，PlayListMark() 为每个都充当在根据包括该 PlayListMark() 的 PlayList()（图 5）而执行的重放的时间基础上的标记的一组零个或多个 Mark() 项。一个 Mark() 项至少包括表示在根据 PlayList() 而执行的重放的时间基础上的一个时间的时间信息、表示 Mark() 的类型的类型信息、和自变量信息，在其中，当类型信息表示创建事件的类型时，自变量信息充当关于该事件的自变量。

[0243] 特殊地，在 PlayListMark() 的开始，设置了长度（32 位）。该长度表示包括它本身的 PlayListMark() 的大小。

[0244] 紧接着该长度，设置了 number_of_PlayList_marks(16 位)，且 number_of_PlayList_marks 表示随后设置的标记（Mark()）的数目。紧接着 number_of_PlayList_marks，描述了与 number_of_PlayList_marks 中的数目一样多的标记结构。

[0245] 在 Mark() 的开始，设置了 mark_type(8 位)。mark_type 为上述类型信息并表示包括该 mark_type 的 Mark() 的类型。

[0246] 在这个实施例中，Mark() 的类型包括，例如，三种类型：章、索引、和事件。

[0247] 其类型为章的 Mark()（如果需要，以下称为章标记）为章中的开始位置的标记，章为分割 PlayList() 的提示单元。另外，其类型为索引的 Mark()（以下称为索引标记）为索引中的开始位置的标记，索引为进一步分割章的单元。其类型为事件的 Mark()（如果需要，以下称为事件标记）为在根据 PlayList() 重放内容期间在其创建事件的位置的标记。如稍后所描述的，将通过事件标记的事件的创建报告给脚本控制模块 211。

[0248] 这里，图 8 中示出 mark_type 的值和 Mark() 的类型之间的关系。根据图 8，在章标记中的 mark_type 中设定为 1。另外，在索引标记中的 mark_type 中设定为 2，以及在事件标记中的 mark_type 中设定为 3。在图 8 中，在由 mark_type 所表示的 8 位值之中，保留 0、和 4 至 255，以便将来扩展。

[0249] 返回参考图 7，紧接着 mark_type，设置了 mark_name_length(8 位)。另外，在 Mark() 的结尾，设置了 mark_name_string(24 字节)。使用 mark_name_length 和 mark_name_string 来描述 Mark() 的名称。mark_name_length 和 mark_name_string 分别表示 mark_name_string 的有效大小和 mark_name_string 的名称。相应地，从 mark_name_string 的开始起由 mark_name_length 所表示的字节数表示 Mark() 的有效名称。

[0250] 紧接着 mark_name_length，顺序地设置了将 PlayList() 上所定义的 Mark() 与片断流文件联系起来的四个元素：ref_to_PlayItem_id(16 位)、mark_time_stamp(32 位)、entry_ES_stream_id(8 位)、和 entry_ES_private_stream_id(8 位)。

[0251] 在ref_to_PlayItem_id中，关于Mark()所属的PlayItem()，描述了以序列号的形式的ID。ref_to_PlayItem_id指定Mark()所属的PlayItem()（图6），且另外，如图6中所描述的，指定片断信息文件和片断流文件。

[0252] mark_time_stamp表示由ref_to_PlayItem_id所指定的片断流文件中的Mark()所表示的位置（时间）

[0253] 这里，图9示出PlayList()、PlayItem()、片断和片断流文件中所存储的节目流之间的关系。

[0254] 在图9中，PlayList()由三个PlayItem()项的组成，并分别将序列号ID#0、#1、#2分配给这三个PlayItem()项。这里，如果需要，以下将分配ID#i的PlayItem()描述为PlayItem#i。

[0255] 这里，在图9中，为将要基于PlayItem#0、PlayItem#1、PlayItem#2而重放的内容片段的片断分别作为片断A、片断B、和片断C而示出。

[0256] 片断的实质是在由图6中中的PlayItem()中的Clip_Information_file_name所指定的（进一步由片断信息文件所指定的）片断流文件中所存储的节目流之中，从IN_time至OUT_time的节目流。在图9中，为片断A、片断B、和片断C的实质的节目流分别作为节目流A、节目流B、和节目流C而示出。

[0257] 例如，在图9中，在充当在根据PlayList()而执行的重放的时间基础上的位置（时间）的标记的Mark()中，以下列方式描述其ref_to_PlayItem_id和mark_time_stamp。

[0258] 换句话说，因为时间t0为在其执行PlayItem#1的重放的时间，所以1作为PlayItem#1的ID而描述。另外，因为在时间T0，基于PlayItem#1重放为片断B的实质的节目流B，所以描述相应于在其中存储节目流B的片断流文件的时间t0的时间。

[0259] 再次返回参考图7，当Mark()与特别的基本流相关联时，使用entry_ES_stream_id和entry_ES_private_stream_id来指定基本流。特殊地，在entry_ES_stream_id中，描述了（设置了）与Mark()相关联的基本流（稍后所描述的图16至18中所示的PES_packet()）的稍后描述的stream_id。另外，在entry_ES_private_stream_id中，描述（设置）了与Mark()相关联的基本流（稍后所描述的且在图21中所示的private_stream1_PES_payload()中的private_header()）中稍后描述的private_stream_id。

[0260] 例如，当在其中将视频流#1和视频流#2多路复用的片断中，需要在重放视频流#1的情形和重放视频流#2的情形之间改变产生章的时间时，在重放视频流#1的模式中产生时间的章标记处的Mark()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。另外，在重放视频流#2的模式中产生时间的章标记处的Mark()的entry_ES_stream_id和entry_ES_private_stream_id中，描述视频流#2的stream_id和private_stream_id。

[0261] 在不与特别的基本流相关联的Mark()的entry_ES_stream_id和entry_ES_private_stream_id的每一个中，例如，描述为0。

[0262] 紧接着entry_ES_private_stream_id，设置了mark_data(32位)。该mark_data为自变量信息，在其中，当Mark()为事件标记时，该自变量信息充当关于由该事件标记所创建的事件的自变量。当Mark()为章标记或索引标记时，能够将mark_data用作由章标记或索引标记所表示的章或索引号。

[0263] “Clip() 的描述”

[0264] 接下来，描述其扩展名为 CLP，且在“CLIP”目录中所设置的片断信息文件的内部结构。

[0265] 在图 4 中，在“CLIP”目录中设置了三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”。它们分别存储表示在“STREAM”目录中所设置的片断流文件“00001.PS”、“00002.PS”、和“00003.PS”的性质，或诸如此类的元数据。

[0266] 图 10 示出上述片断信息文件 Clip() 的内部结构。

[0267] 在片断信息文件 Clip() 的开始，顺序地设置了 presentation_start_time 和 presentation_end_time（每个 32 位）。Presentation_start_time 和 presentation_end_time 分别表示相应于该片断信息文件 Clip() 的片断流文件中的开始和结束时间。描述片断流文件中的时间为 MPEG2- 系统的时间中的 90kHz 的倍数。

[0268] 紧接着 presentation_end_time，顺序地设置了 reserved_for_word_alignment(7 位）和 capture_enable_flag_Clip(1 位）。使用 7 位 reserved_for_word_alignment 来建立文字对齐，且类似于上述图 5 中所示的 capture_enable_flag_PlayList，该 capture_enable_flag_Clip 为指示是否允许视频数据的二次使用的标志。

[0269] 然而，图 5 中的 capture_enable_flag_PlayList 指示是否允许相应于由 PlayList() 所重放的视频流的视频数据（属于 PlayList() 的视频数据）的二次使用，而图 10 中的 capture_enable_flag_Clip 指示是否允许相应于在相应于片断信息文件 Clip() 的片断流文件中所存储的视频流（视频的基本流）的视频数据的二次使用。因此，图 5 中的 capture_enable_flag_PlayList 和图 10 中的 capture_enable_flag_Clip 具有不同单元（范围）的其二次使用被允许的视频数据。

[0270] 如关于图 5 中的 capture_enable_flag_PlayList 所描述的，图 10 中的 capture_enable_flag_PlayList 不仅能够由一位而且能够由多位构成。

[0271] 紧接看 capture_enable_flag_Clip，设置了 number_of_streams(8 位），且在 number_of_streams 中，描述了接下来的 StreamInfo() 结构的数目。相应地，紧接着 number_of_streams，描述了与 number_of_streams 中的数目一样多的 StreamInfo() 结构。

[0272] 在 StreamInfo() 的开始，设置了长度 (16 位），且这个长度表示包括它本身的 StreamInfo() 的大小。紧接着该长度，设置了 stream_id(8 位）和 private_stream_id(8 位），且该 stream_id 和 private_stream_id 指定（识别）将要与 StreamInfo() 相关联的基本流。

[0273] 这里，图 11 示出指定基本流的 stream_id 和 private_stream_id 及基本流之间的关系。

[0274] stream_id 与 MPEG2- 系统规范说明中所供给的那个相同，且关于基本流（数据）的每个属性（类型）都预先确定它的值。因此，能够仅通过 stream_id 指定具有 MPEG2- 系统规范说明中所定义的属性的基本流。

[0275] 在这个实施例中，也能够使用在 MPEG2- 系统规范说明中不供给其属性的基本流。private_stream_id 为识别在 MPEG2- 系统规范说明中不供给其属性的基本流的信息。

[0276] 在图 11 中，关于四个属性的基本流，或者通过 MPEG 中所定义的编码（解码）而编码的视频的基本流、ATRAC（自适应变换声音编码）中所编码的音频的基本流（如果需要，以

下称为 ATRAC 音频流）、LPCM（线性脉冲编码调制）中所编码的音频的基本流（如果需要，以下称为 LPCM 音频流）、和字幕的基本流（如果需要，以下称为字幕流），描述了 stream_id 和 private_stream_id 之间的关系。

[0277] 在 MPEG2- 系统标准中，供给通过使用在 0xE0 至 0xEF(0x 指示接下来的字符串是 16 进制的）范围内的值作为识别基本流的 stream_id，而将通过 MPEG 中的编码而编码的视频的基本流多路复用。相应地，关于以 MPEG 中的编码而编码的视频的基本流，能够将能够由具有在 0xE0 至 0xEF 范围内的值的 stream_id 识别的视频的十六个基本流多路复用成一个节目流。

[0278] 因为能够通过具有 0xE0 至 0xEF 的值的 stream_id 执行以 MPEG 中所定义的编码而编码的视频的基本流的识别，就不需要（能够忽略）private_stream_id。

[0279] 另外，在 MPEG2- 系统中，关于 ATRAC 音频流、LPCM 音频流、和字幕音频流，未定义 stream_id。

[0280] 相应地，在这个实施例中，关于在其中在 MPEG2- 系统中未定义 stream_id 的基本流，采用为表示 MPEG2 系统中的 private_stream_1 的属性的值的 0xBD 作为 stream_id，且另外，如图 11 中所示，使用 private_stream_id 来执行识别（指定）。

[0281] 特殊地，在识别 ATRAC 音频流中，使用具有在 0x00 至 0x0F 范围内的值的 private_stream_id。因此，能够将 16 个 ATRAC 音频流多路复用成一个节目流。另外，在识别 LPCM 音频流中，使用具有在 0x10 至 0x1F 范围内的值的 private_stream_id。相应地，能够将 16 个音频流多路复用成一个节目流。另外，在识别字幕流中，能够将具有在 0x80 至 0x9F 范围内的值的 private_stream_id 多路复用成一个节目流。

[0282] 稍后进一步描述 stream_id 和 private_stream_id。

[0283] 返回参考图 10，紧接着 private_stream_id，顺序地设置了 StaticInfo()、和 reserved_for_word_alignment(8 位）。在 StaticInfo() 中，描述了在由（在包括该 StaticInfo() 的 streamInfo() 中所描述的）stream_id 和 private_stream_id 所指定的基本流的重放期间不改变的信息。稍后参考图 12 描述 StaticInfo() 的详细情况。

[0284] 使用 reserved_for_word_alignment 来建立文字对齐。

[0285] 紧接着 reserved_for_word_alignment，设置了 number_of_DynamicInfo(8 位），且该 number_of_DynamicInfo 表示随后所设置的 DynamicInfo() 和 pts_change_point(32 位）组的数目。

[0286] 因此，紧接着 number_of_DynamicInfo，描述了与 number_of_DynamicInfo 中的数目一样多的 pts_change_point 和 DynamicInfo() 的结构。

[0287] pts_change_point 表示 DyniamicInfo() 的信息变为有效的时间，该 DyniamicInfo() 与 pts_change_point 构成一组。这里，表示基本流头的时间的 pts_change_point，等于在相应于在其中存储该基本流的片断流文件的片断信息文件 Clip() 的开始所描述的 presentation_start_time。

[0288] 在 DynamicInfo() 中，描述了在由 stream_id 和 private_stream_id 所定义的基本流的重放期间而改变的所谓的动态信息。在 DynamicInfo() 中所描述的信息在由 pts_change_point 所表示的重放时间变为有效，该 pts_change_point 与 DynamicInfo() 构成一组。稍后参考图 13 描述 DynamicInfo() 的详细情况。

[0289] 紧接着与 number_of_DynamicInfo 中的数目一样多的 pts_change_point 和 DynamicInfo() 项的组，设置了 EP_map()。稍后参考图 14 描述 EP_map()。

[0290] 接下来，参考图 12 描述图 10 中的 StaticInfo() 的详细情况。

[0291] 图 12 示出 StaticInfo() 的语法。

[0292] 依赖于相应的基本流的属性（类型），StaticInfo() 在内容上不同。由图 10 中的包括 StaticInfo() 的 StreamInfo() 中所包括的 stream_id 和 private_stream_id 来识别相应于 StaticInfo() 的基本流的属性。

[0293] 当相应于 StaticInfo() 的基本流为视频流（流＝＝视频）时，该 StaticInfo() 包括 picture_size(4 位)、frame_rate(4 位)、和 cc_flag(1 位)、和用于建立文字对齐的 reserved_for_word_alignment。

[0294] picture_size 表示相应于视频流的视频数据（基于该视频数据所显示的画面）的大小。frame_rate 表示相应于视频流的视频数据的帧频。cc_flag 指示在视频流中是否包括隐藏字幕数据。换句话说，例如，当视频流中包括隐藏字幕数据时，cc_flag 设定为 1，且当不包括隐藏字幕数据时，cc_flag 设定为 0。

[0295] 当相应于 StaticInfo() 的基本流为音频流（流＝＝音频）时，该 StaticInfo() 包括 audio_language_code(16 位)、channel_configuration(8 位)、lfe_existence(1 位)、sampling_frequency(4 位)、和用于建立文字对齐的 reserved_for_word_alignment。

[0296] 在 audio_language_code 中，描述了表示音频流中所包括的音频数据的语言的代码。channel_configuration 表示音频流中所包括的音频数据的属性，例如非立体声的、立体声的、多声道的。lfe_existence 指示在音频流中是否包括低重音声道。当包括该声道时，lfe_existence 为 1，且当不包括该声道时，lfe_existence 为 0。sampling_frequency 为表示音频流中所包括的音频数据的采样频率的信息。

[0297] 当相应于 StaticInfo() 的基本流为字幕流（流＝＝字幕）时，该 StaticInfo() 包括 subtitle_language_code(16 位)、configurable_flag(1 位)、和用于建立文字对齐的 reserved_for_word_alignment。

[0298] 在 subtitle_language_code 中，描述了表示字幕流中所包括的字幕数据的语言的代码。configurable_flag 为指示是否允许从默认显示形式改变字幕流中所包括的字幕数据的显示。例如，当允许改变显示形式时，描述为 1，且当不允许时，描述为 0。字幕数据显示包括字幕数据的显示大小、显示位置、显示色彩、显示模式（例如，闪光或诸如此类）、和显示方向（例如，垂直方向和水平方向）。

[0299] “DynamicInfo() 的描述”

[0300] 接下来，参考图 13 描述图 10 中 DynamicInfo() 的详细情况。

[0301] 图 13 示出 DynamicInfo() 的语法。

[0302] 在 DynamicInfo() 的开始，设置了用于建立文字对齐的 reserved_for_word_alignment，且依赖于相应于 DynamicInfo() 的基本流的属性，以下的元素在内容上不同。类似于图 12 中所描述的 StaticInfo() 的情形，基于包括 DynamicInfo() 的图 10 中的 StreamInfo() 中所包括的 stream_id 和 private_stream_id 而识别相应于 DynamicInfo() 的基本流的属性。

[0303] 在 DynamicInfo() 中，如图 10 中所描述的，描述了在基本流的重放期间改变的动

态信息。尽管并不特别地限制这种动态信息，但在图 13 中的实施例中，在 DynamicInfo() 中描述了相应于对应于 DynaminInfo() 的基本流的数据，即，通过处理该基本流而输出的数据的输出属性（从该基本流而获得的数据的输出属性）。

[0304] 特殊地，当相应于 DynamicInfo() 的基本流为视频流（流＝＝视频）时，DynamicInfo() 包括 display_aspect_ratio(4 位）和关于文字对齐的 reserved_for_word_alignment。在 display_aspect_ratio 中，例如，视频数据的屏幕高宽比描述为视频数据的输出属性（显示形式），其相应于视频流。换句话说，在 display_aspect_ratio 中，描述了指示屏幕高宽比为，例如，16 ： 9 和 4 ： 3 中的一个的信息。在视频流的 DynamicInfo() 中，除屏幕高宽比之外，例如，能够描述基于视频数据，或诸如此类的画面的大小 (X 像素 ×Y 像素）。

[0305] 当相应于 DynamicInfo() 的基本流为音频流（流＝＝音频）时，DynamicInfo() 包括 channel_assignment(4 位）和用于建立文字对齐的 reserved_for_word_alignment。在 channel_assignment 中，当音频流包括两个声道的音频数据时，描述两个声道的输出属性（输出形式）。特殊地，在 channel_assignment 中，描述指示立体声和双重（两重）声道的哪个声道分配给该音频数据的信息。

[0306] 当相应于 DynamicInfo() 的基本流为字幕流（流＝＝字幕）时，DynamicInfo() 由用于建立文字对齐的 reserved_for_word_alignment 构成。换句话说，在图 13 中的实施例中，关于字幕流，未定义作为动态信息的输出属性。

[0307] “EP_map() 的描述”

[0308] 接下来，参考图 14 描述图 10 中的 EP_map() 的详细情况。

[0309] 图 14 示出 EP_map() 的语法。

[0310] 在 EP_map() 中，关于相应于图 10 中的包括 EP_map() 片断信息文件 Clip() 的片断流文件中所存储的基本流的每一个，描述了关于在其能够开始解码的解码可开始点（进入点）的每个基本流的信息。

[0311] 这里，关于固定比率的流，能够通过计算找到解码可开始点。然而，关于对于每个视频存取单元具有不同的大小的流，例如，可变比率流或根据 MPEG 标准而编码的视频流，就不能通过计算而找到解码可开始点，且除非分析该流，不能够找到解码可开始点。解码可开始点的迅速辨识对于执行随机存取是很重要的，而根据 EP_map()，能够迅速地辨识解码可开始点。

[0312] 在 MPEG2- 视频中，包括序列 header()（序列头）等等的帧内画面的开始为解码可开始点。

[0313] 在 EP_map() 的开始，设置了关于文字对齐的 reserved_for_word_alignment(8 位），且随后设置了 number_of_stream_id_entries。number_of_stream_id_entries 表示在其中 EP_map() 描述解码可开始点的信息的基本流的数目。紧接着 number_of_stream_id_entries，重复设置了与由 number_of_stream_id_entries 所表示的数目一样多次数的，识别基本流的信息和该基本流的解码可开始点的信息。

[0314] 特殊地，紧邻 number_of_stream_id_entries 之后，设置了 stream_id(8 位）和 private_stream_id(8 位），其为识别基本流的信息，且随后设置了 number_of_EP_entries(32 位）。number_of_EP_entries(32 位）表示由紧邻 number_of_EP_entries 之前

的 stream_id 和 private_stream_id 所识别（指定）的基本流中的解码可开始点的数目。

[0315] 紧接着 number_of_EP_entries，重复地设置了与 number_of_EP_entries 中的数目一样多次数的一组 PTS_EP_start(32 位）和 RPN_EP_start(32 位），其为由相邻的 stream_id 和 private_stream_id 所指定的基本流中的解码可开始点的信息。

[0316] 为一条解码可开始点的信息的 PTS_EP_start 表示在片断流文件中的解码可开始点的时间（重放时间），该片断流文件存储在其中将如上文所描述的由 stream_id 和 private_stream_id 所指定的基本流多路复用的节目流。

[0317] 在作为解码可开始点的信息的另一条的 RPN_EP_start 中，描述了一个值，在以节目流中的 pack() 项为单元计算片断流文件中的解码可开始点的位置的情形中，获得该值，其中该片断流文件存储在其中将如上文所描述的由 stream_id 和 private_stream_id 所指定的基本流多路复用的节目流。在这个实施例中，pack() 的大小为 2048 字节的固定值。同样在这个实施例中，盘 101（图 1）的一个扇区为 2048 字节。

[0318] 这里，关于视频流，紧邻其解码可开始点（进入点）之前，设置了稍后描述的 private_stream_2 信息包（具有 private_stream_2 的属性的 PES_packet()）。private_stream_2 信息包存储将从 private_stream_2 信息包至下一个 private_stream_2 信息包之间所设置的视频流解码所使用的信息。相应地，关于视频流，在为解码可开始点的信息的 RPN_EP_start 中，未描述实际的解码可开始点本身，而描述了紧邻实际的解码可开始点之前所设置的 private_stream_2 的开始的位置。

[0319] 另外，在 EP_map() 中，关于由 stream_id 和 private_stream_id 所指定的每个基本流，预先以升序存储了为解码可开始点的信息的 PTS_EP_start 和 RPN_EP_start 项的组。这使得作为解码可开始点的信息的 PTS_EP_start 和 RPN_EP_start 项的组的对分搜索成为可能。

[0320] 在例如，日本未经审核的专利申请公开 No.2000-341640（日本专利申请 No.11-317738）中，描述了关于对可变比率流和具有不同的大小的视频存取单元的流的随机存取的方法。

[0321] “片断流文件的描述”

[0322] 接下来，描述其扩展名为 PS 且在图 4 中的“STREAM”目录中所设置的片断流文件（图 4 中的“00001.PS”、“00002.PS”、和“00003.PS”）的每个内部结构。

[0323] 基于 MPEG-2 系统 (ISO/IEC 13818-1) 中所定义的 MPEG2_Program_Stream() 而构成片断流文件。

[0324] 换句话说，图 15 示出 MPEG-2 系统 (ISO/IEC 13818-1 :2000) 标准中所描述的表 2-31、表 2-32、和表 2-33。

[0325] 片断流文件中所存储的节目流为 MPEG-2 系统标准中的表 2-31 中所定义的 MPEG2_Program_Stream()，且包括至少一个 pack() 项和一个 MPEG_program_end_code。在日本专利 No.2785220 等等中描述了 MPEG2_Program_Stream() 的描述。

[0326] 如 MPEG-2 系统标准中的表 2-32 中所定义的，一个 pack() 项包括一个 PACKHEADER（包头）和任意数目的 PES 信息包 (PES_packet())。在 MPEG-2 系统标准中的表 2-33 中定义 Pack_header() 的详细情况。

[0327] 尽管，在 MPEG-2 系统标准中，pack() 的大小定义为可变长度，这里如图 14 中所描

述的，它固定为 2048 字节。此外，这里，一个 pack() 中 PES_packet() 项的数目为一、二或三。当 pack() 以稍后所描述的 private_stream_2 信息包开始时，紧接着该信息包（在相同的 pack() 中）必要地存在相应视频流的 PES_packet()。除此之外，能够将 padding_packet 作为第三 PES_packet() 而设置。在 pack() 的开始必要地设置 private_stream_2 信息包。

[0328] 当 pack() 不以 private_stream_2 信息包开始时，在 pack() 的开始，设置存储像视频、音频、和字幕那样的内容数据的 PES_packet()。除此之外，能够将 padding_packet 作为第二 PES_packet() 而设置。

[0329] 图 16 至 18 示出 MPEG-2 系统标准中的表 2-17 中所定义的 PES_packet()。

[0330] 能够将 PES_packet() 广泛地分割成为 packet_start_code_prefix、stream_id、和 PES_packet_length（图 16）、其结构依赖于 stream_id 或诸如此类而改变的头部分（包括 stuffing_byte）（图 16 至 18）、和 PES_packet_data_byte（图 18）。当 PES_packet() 为 padding_packet(stream_id == padding_stream) 时，重复地设置与所需要的数目（而非 PES_packet_data_byte）一样多的次数的 padding_byte(0xFF)（图 18）。

[0331] 在 PES_packet() 的头部分，如图 16 和 17 中所示，能够设置被称为 PTS（显示时间戳）且表示显示定时的信息，和被称为 DTS（解码时间戳）且表示解码定时的信息。在这个实施例中，将 PTS 项添加到所有的存取单元 (MPEG-2 系统中所定义的构成基本流的解码单元)，且当在 MPEG-2 系统中定义时，添加 DTS 项。

[0332] 在 PES_packet() 的 PES_packet_data_byte（图 18）中存储在节目流中多路复用的基本流。在 PES_packet() 中的 stream_id 中，为了识别 PES_packet_data_byte 中所存储的基本流，描述了根据该基本流的属性的值。

[0333] 在 MPEG-2 系统标准中的表 2-18 中定义了 PES_packet() 中所描述的值和基本流的属性（类型）之间的关系。这里，在图 19 中，示出 MPEG-2 系统标准中的表 2-18。

[0334] 在这个实施例中，在图 19 中所示的 MPEG-2 系统中所定义的 stream_id 值之中，例如，采用图 20 中所示的值。

[0335] 换句话说，在这个实施例中，采用五个模式：10111101B、10111110B、10111111B、110xxxxxB、和 1110xxxxB 作为 stream_id 值。“x”表示为 0 或 1 的任意值。

[0336] 根据图 20，具有被称为 private_stream_1 的属性的基本流的 PES_packet() 的 stream_id 设定为 10111101B。另外，根据图 20，padding_packet 的 pack() 的 stream_id 设定为 10111110B。此外，根据图 20，具有被称为 private_stream_2 的属性的基本流的 PES_packet() 的 stream_id 设定为 10111111B。

[0337] 另外，MPEG 中所定义的音频流的 PES_packet() 的 stream_id 设定为 110xxxxxB。110xxxxxB 的低五位表示识别音频流的音频流号。在节目流 (MPEG2_Program_Stream()) 中，可能将能够由音频流号而识别的 32（$=2^5$）个音频流 (MPEG 中所定义的音频流）多路复用。

[0338] 如 MPEG 中所定义的视频流的 PES_packet() 的 stream_id 设定为 1110xxxxB。1110xxxxB 的较低的四位 xxxx 表示识别视频流的视频流号。在节目流中，可能将能够由视频流号而识别的 16（$=2^4$）个视频流 (MPEG 中所定义的视频流）多路复用。

[0339] 将具有 1110xxxxB 的 stream_id 的 PES_packet() 用来存储如 MPEG 中所定义的视频流，且将具有 110xxxxxB 的 stream_id 的 PES_packet() 用来存储如 MPEG 中所定义的音

频流。另外，在 MPEG 中未定义用来存储根据 MPEG 中未定义的编码（例如，ATRAC 编码）的基本流的 PES_packet 的 stream_id。因此，不能够通过简单地指定类似于 MPEG 中所定义的视频流和音频流的 stream_id 而在 PES_packet() 中存储根据 MPEG 中未定义的编码的基本流。

[0340] 相应地，在这个实施例中，扩展了 private_stream_1 的 PES_packet() 中的 PES_packet_data_byte，以存储根据 MPEG 中未定义的编码的基本流。

[0341] 这里，将 private_stream_1 的 PES_packet() 中的扩展 PES_packet_data_byte 描述为 private_stream1_PES_payload()。

[0342] “private_stream1_PES_payload() 的描述”

[0343] 图 21 示出 private_stream1_PES_payload() 的语法。

[0344] private_stream1_PES_payload() 包括 Pack_header() 和 private_payload()。在 private_payload() 中，存储根据 MPEG 中未定义的编码的基本流，例如 ATRAC 音频流、LPCM 音频流、和字幕流。

[0345] 在 private_header() 的开始，设置了 private_stream_id(8 位)。private_stream_id 为识别 private_payload() 中所存储的基本流的识别信息，且根据它们的属性（类型）设定为，例如，以下的值。

[0346] 换句话说，图 22 示出 private_stream_id 值和 private_payload() 中所存储的基本流的属性之间的关系。

[0347] 在图 22 中，采用三种模式 :0000xxxxB、0001xxxxB、和 100xxxxxB 作为 private_stream_id 的值。“x”表示类似于图 20 中的情形的 1 和 1 之间的任意值。

[0348] 根据图 22，在其中 ATRAC 音频流位于 private_payload() 中的 private_stream1_PES_payload() 的 private_stream_id 设定为 0000xxxxB。0000xxxxB 的较低的四位 xxxx 表示识别 ATRAC 音频流的音频流号。在节目流 (MPEG2_Program_Stream()) 中，可能将能够由音频流号而识别的 16(= 2^4) 个 ATRAC 音频流多路复用。

[0349] 而且，根据图 22，在其中在 private_payload() 中存储 LPCM 音频流的 private_stream1_PES_payload() 的 private_stream_id 设定为 0001xxxxB。0001xxxxB 的较低四位表示识别 LPCM 音频流的音频流号。在节目流中，可能将能够由音频流号识别的 16(= 2^4) 个 LPCM 音频流多路复用。

[0350] 另外，根据图 22，在其中在 private_payload() 中存储字幕流的 private_stream1_PES_payload() 的 private_stream_id 设定为 100xxxxxB。100xxxxxB 的低五位 xxxxx 表示识别字幕流的字幕流号。在节目流中，可能将能够由字幕流号识别的 32(= 2^5) 个字幕流多路复用。

[0351] 这里，图 20 和图 22 之间的关系的汇集为上文所描述的图 11。

[0352] 返回参考图 21，在 private_stream1_PES_payload() 中，紧接着 private_stream_id 的元素依赖于 private_payload() 中所存储的基本流的属性而在内容上改变。private_payload() 中所存储的基本流的属性由在 private_header() 的开始的 private_stream_id 确定。

[0353] 当在 private_payload() 中所存储的基本流为 ATRAC 音频流 (private_stream_id == ATRAC) 时，设置了用于将来扩展的 reserved_for_future_use(8 位)，并随后设置

了 AU_locator(16 位)。通过使用紧接 AU_locator 之后的位置作为参考,AU_locator 表示 private_payload() 中所存储的 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() 中不存在音频存取单元时,例如,在 AU_locator 中描述为 0xFFFF。

[0354] 当在 private_payload() 中所存储的基本流为 LPCM 音频流 (private_stream_id == LPCM) 时,顺序地设置了 fs_flag(1 位)、reserved_for_future_use(3 位),ch_flag(4 位)、和 AU_locator(16 位)。

[0355] fs_flag 表示 private_payload() 中所存储的 LPCM 音频流的采样频率。特殊地,当采样频率为,例如,48kHz 时,fs_flag 设定为 0,而当采样频率为 44.1kHz 时,fs_flag 设定为 1。

[0356] ch_flag 表示 private_payload() 中所存储的 LPCM 音频流的声道的数目。当 LPCM 音频流为,例如,单声道时,ch_flag 设定为 1,而当 LPCM 音频流为立体声时,ch_flag 设定为 2。

[0357] 通过使用紧接 AU_locator 之后的位置作为参考,AU_locator 表示 private_payload() 中所存储的 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() 中不存在音频存取单元时,例如,在 AU_locator 中描述为 0xFFFF。

[0358] 当在 private_payload() 中所存储的基本流为字幕流 (private_stream_id == SUBTITLE) 时,设置了用于将来扩展的 reserved_for_future_use(8 位),且随后设置了 AU_locator(16 位)。通过使用紧接 AU_locator 之后的位置作为参考,AU_locator 表示 private_payload() 中所存储的字幕流的字幕存取单元的开始位置。当 private_payload() 中不存在字幕存取单元时,例如,在 AU_locator 中描述为 0xFFFF。

[0359] “private_stream2_PES_payload() 的描述”

[0360] 接下来,图 23 示出 private_stream2_PES_payload() 的语法。

[0361] private_stream2_PES_payload() 为 private_stream_2 中 PES_packet() 的 PES_packet_data_byte(图 18) 的扩展,即,它是 private_stream_2 的 PES_packet() 中的扩展 PES_packet_data_byte。在 private_stream2_PES_payload() 中,描述视频流解码中将要利用的信息。

[0362] 在这个实施例中,紧邻视频流中解码可开始位置之前,设置了 private_stream2_PES_payload()。相应地,在这个实施例中,通过从 presentation_start_time 中找到 private_stream_2 的 PES_packet(),能够从紧接着 PES_packet() 的视频流开始解码。

[0363] 这里,上述图 14 中 EP_map() 的 RPN_EP_start 表示 private_stream_2 中 PES_packet() 的开始位置。

[0364] 在 private_stream2_PES_payload() 的开始,设置了用于将来扩展的 reserved_for_future_use(8 位)。紧接着它,顺序地设置了 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()。

[0365] 在 vide0_stream_id 中,描述了紧接着 private_stream_2 中的 PES_packet() 而设置的 PES_packet()(视频流中)的 stream_id(等于该 stream_id 的值)。基于 video_

stream_id，通过使用 private_stream_2 中的 PES_packet()（的 private_stream2_PES_payload()）中所存储的信息，指定将要解码的视频流（存储该视频流的 PES_packet()）。

[0366] 通过使用相关的值，1stRef_picture、2ndRef_picture、3rdRef_picture、和 4thRef_picture 分别表示包括由 video_stream_id 所指定的视频流中的 private_stream_2 的 PES_packet() 与下一个 private_stream_2 的下一个 PES_packet() 之间的第一、第二、第三、和第四参考画面的上一个信息包 (pack()) 的位置。关于 1stRef_picture、2ndRef_picture、3rdRef_picture、和 4thRef_picture，在例如，日本未经审核的专利申请公开 No.09-46712(日本专利申请 No.07-211420) 中，它们的详细情况公布为 bytes_to_first_P_pic、bytes_to_second_P_pic。

[0367] 在 au_information() 中，描述关于从 private_stream_2 的 PES_packet() 至下一个 private_stream_2 的 PES_packet() 的视频流中的视频存取单元的信息。稍后关于图 24 描述 au_information() 的详细情况。

[0368] 使用 VBI() 来描述隐藏字幕信息。

[0369] 在每个视频流中的每个解码可开始点都设置了具有上文所描述的 private_stream2_PES_payload() 的 private_stream_2 的 PES_packet()。

[0370] 接下来，图 24 示出图 23 中 au_information() 的语法。

[0371] 在 au_information() 的开始，设置了长度 (16 位)。长度表示包括该长度的 au_information() 的大小。紧接着该长度，顺序地设置了 reserved_for_word_alignment(8 位) 和 number_of_access_units(8 位)。使用 reserved_for_word_alignment 来建立文字对齐。

[0372] number_of_access_units 表示在包括该 number_of_access_units 的 private_stream_2 中的 PES_packet() 与下一个 private_stream_2 中的 PES_packet() 之间的视频存取单元（画面）的数目。

[0373] 特殊地，在其中图 23 中的 private_stream2_PES_payload() 具有同样的 video_stream_id 的 private_stream_2 的 PES_packet() 中，从其 au_information() 至与下一个 au_information() 相邻的位置（如果这个 au_information() 为片断流文件中的最后的 au_information()，则至片断流文件的结尾），number_of_access_units 表示由 video_stream_id 所表示的视频流中所包括的存取单元（画面）的数目。

[0374] 紧接着 number_of_access_units，设置了与 number_of_access_units 中的数目一样多的 for-loop 内容。特殊地，设置了关于包括该 number_of_access_units 的 private_stream_2 的 PES_packet() 与下一个 private_stream_2 的 PES_packet() 之间所包括的一个或多个视频存取单元的每一个的信息。

[0375] for-loop 中所设置的信息（视频存取单元信息）为如下文所描述。

[0376] 特殊地，在 for-loop 中，设置了 pic_struct_copy()(4 位)、au_ref_flag(1 位)、AU_length(21 位)、和保留。

[0377] 在 pic_struct_copy 中，描述了 pic_struct() 的范本，其为 ISO/IEC14496-10、D.2.2 中所定义的，且当该视频流基于 MPEG4-AVCv(ISO/IEC14496-10) 时，其关于相应的视频存取单元而设定。pic_struct() 为像作为帧而显示画面、或显示画面的顶部场且随后显示底部场那样的信息。

[0378] au_ref_flag指示当将其它存取单元(的画面)解码时,相应的存取单元是否表示将要参考的参考画面。当表示参考画面时,其设定为1,且当不表示参考画面时,设定为0。

[0379] AU_length以字节单元的形式表示相应的存取单元的大小。

[0380] 盘101上所记录的数据的特定实例。

[0381] 接下来,图25至28示出图1中盘101上以上文所描述的格式所记录的数据的特定实例。

[0382] 在图25至28中,关于视频流采用MPEG2-Video,且采用ATRAC音频流作为音频流。然而,视频流和音频流并不局限于上述格式。换句话说,例如,关于视频流,能够采用MPEG4-Visual、MPEG4-ACV等等,且另外,例如,能够采用MPEG1/2/4音频、LPCM音频流等等。

[0383] 与视频流和音频流不同,关于字幕流,并不总是执行在字幕流的等间隔的顺序解码和显示(输出)。换句话说,从缓冲器控制模块215不间断地供给字幕流,并在字幕解码器控制模块218中将其解码。

[0384] 图25至28示出,如图4中所示,在盘101上的“CLIP”目录中记录了三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”,并且当在“STREAM”目录中记录分别相应于三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”的三个片断流文件“00001.PS”、“00002.PS”、和“00003.PS”时,示出特定实例,例如片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”。然而,在图25至28中,省略了“PLAYLIST.DAT”文件等等中的部分数据。

[0385] 换句话说,图25示出图5中所描述的“PLAYLIST.DAT”文件的特定实例。

[0386] 在图25中,number_of_PlayLists为2。因而,“PLAYLIST.DAT”文件中所包括的PlayList()项的数目为2。在图25中,在两个PlayList()项之间,第一个和第二个分别描述为PlayList()#0和PlayList()#1。

[0387] 在作为第一PlayList()的PlayList()#0中,capture_enable_flag_PlayList为1。因而,允许根据PlayList()#0而重放的视频数据的二次使用。另外,在PlayList()#0中,number_of_PlayItems为2。因而,PlayList()#0中所包括的PlayItem()项的数目为2。在图25中,在“PlayList()#0”栏的较低部分描述了作为两个PlayItem()项的PlayItem()#0和PlayItem()#1的特定实例。

[0388] 在PlayList()#0中所包括的作为第一PlayItem()的PlayItem()#0中,图6中所描述的Clip_Information_file_name为“00001.CLP”,In_time为180,090,以及OUT_time为27,180,090。因而,基于PlayList()#0中的PlayItem()#0而重放的片断,在相应于片断信息文件“00001.CLP”的片断流文件“00001.PS”中从时间180,090至时间27,180,090范围内。

[0389] 在PlayList()#0中所包括的作为第二PlayItem()的PlayItem()#1中,图6中所描述的Clip_Information_file_name为“00002.CLP”,In_time为90,000,以及OUT_time为27,090,000。因而,基于PlayList()#0中的PlayItem()#1而重放的片断,在相应于片断信息文件“00002.CLP”的片断流文件“00002.PS”中从时间90,000至时间27,090,000范围内。

[0390] 另外,在图25中,在作为第二PlayList()的PlayList()#1中,capture_enable_flag_PlayList为0。因而,不允许(禁止)根据PlayList()#1而重放的视频数据的二次

使用。同样,在 PlayList()#1 中,number_of_PlayItems 为 1。因而,PlayList()#1 中所包括的 PlayItem() 项的数目为 1。在图 25 中,在“PlayList()#1”栏的较低部分描述了作为一个播放项 PlayItem() 的 PlayItem()#0 的特定实例。

[0391] 在 PlayList()#1 中所包括的作为一个 PlayItem() 的 PlayItem()#0 中,图 6 中所描述的 Clip_Information_file_name 为“00003.CLP”,OUT_time 为 81,090,000。因而,基于 PlayList()#1 中的 PlayItem()#0 而重放的片断,在相应于片断信息文件“00003.CLP”的片断流文件“00003.PS”中从时间 90,000 至时间 81,090,000 范围内。

[0392] 接下来,图 26 示出图 10 中所描述的片断信息文件 Clip() 的特定实例。换句话说,图 26 示出图 4 中的片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”的特定实例。

[0393] 在片断信息文件“00001.CLP”中,presentation_start_time 为 90,000,以及 presentation_end_time 为 27,990,000。因而,根据相应于片断信息文件“00001.CLP”的片断流文件“00001.PS”中所存储的节目流,能够使用有关 310((27,990,000-90,000)/90kHz) 秒的内容。

[0394] 另外,在片断信息文件“00001.CLP”中,capture_enable_flag_Clip 为 1。因而,允许在相应于片断信息文件“00001.CLP”的片断流文件“00001.PS”中所存储的节目流中多路复用的视频流(相应于该视频流的视频数据)的二次使用。

[0395] 此外,在图 26 中,在片断信息文件“00001.CLP”中的 number_of_streams 为 4。因而,在相应的片断流文件“00001.PS”中所存储的节目流中,将四个基本流多路复用。

[0396] 当将该四个基本流表示为 stream#0、stream#1、stream#2、和 stream#3 时,在图 26 中,在“00001.CLP”栏的较低部分描述了四个基本流 :stream#0、stream#1、stream#2、和 stream#3 的 StreamInfo() 的特定实例。

[0397] 在第一基本流 stream#0 中,在片断流文件“00001.PS”中,stream_id 为 0xE0。因而,这个基本流 stream#0 为图 20 和 22(或图 11) 中所描述的视频流。尽管,在这个实施例中,,如上文所描述的,private_stream_id 与视频流无关,它为图 26 中的 0x00。

[0398] 另外,关于作为片断流文件“00001.PS”中第一基本流的视频流 stream#0,StreamInfo() 中所包括的 StaticInfo()(图 12) 的 picture_size 为“720×480”,frame_rate 为“29.97Hz”,以及 cc_flag 为“Yes”。因而,这个视频流 stream#0 为具有 720×480 像素和 29.97Hz 的帧周期的视频数据,且另外,包括隐藏字幕数据。

[0399] 此外,关于作为片断流文件“00001.PS”中第一基本流的视频流 stream#0,StreamInfo()(图 10) 的 number_of_DynamicInfo() 为 0。因而,不存在一组 pts_change_point 和 DynanicInfo()。

[0400] 接下来,关于片断流文件“00001.PS”中的第二基本流的视频流 stream#1,stream_id 为 0xBD,且 private_stream_id 为 0x00。因此,如图 20 和 22 中所示,这个基本流 stream#1 为 ATRAC 音频流。

[0401] 另外,关于作为片断流文件“00001.PS”中第二基本流的 ATRAC 音频流 stream#1,在 StreamInfo() 中所包括的 StaticInfo()(图 12) 中,audio_language_code 为“JAPANESE”,channel_configuration 为“STEREO”,lfe_existence 为“NO”,以及 sampling_frequency 为“48kHz”。因此这个 ATRAC 音频流 stream#1 为日本及立体声音频数据。另外,不包括低重音声道,且采样频率为 48kHz。

[0402] 此外，关于作为片断流文件“00001.PS”中第二基本流的 ATRAC 音频流 stream#1，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 0。因而，不存在一组 pts_change_point 和 DynanicInfo()。

[0403] 接下来，关于片断流文件“00001.PS”中的第三基本流 stream#2，stream_id 为 0xBD，且 private_stream_id 为 0x80。因此，如图 20 和 22 中所示，这个基本流 stream#2 为字幕流。

[0404] 另外，关于作为片断流文件“00001.PS”中第三基本流的字幕流 stream#2，在 StreamInfo() 中所包括的 StaticInfo()（图 12）的 subtitle_language_code 为“JAPANESE”，且 configurable_flag 为 0。因此，这个字幕流 stream#2 为日本字幕数据，且不允许（禁止）改变其显示形式。

[0405] 此外，关于作为片断流文件“00001.PS”中第三基本流的字幕流 stream#2，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 0。因而，不存在一组 pts_change_point 和 DynanicInfo()。

[0406] 接下来，关于片断流文件“00001.PS”中的第四基本流 stream#3，stream_id 为 0xBD，且 private_stream_id 为 0x81。因此，如图 20 和 22 中所示，这个基本流 stream#3 为字幕流。

[0407] 为了在作为片断流文件“00001.PS”中第三基本流的字幕流 stream#2 与作为第四基本流的字幕流 stream#3 之间进行区别，它们的 private_stream_id 项分别为 0x80 和 0x81。

[0408] 另外，关于作为片断流文件“00001.PS”中第四基本流的字幕流 stream#2，在 StreamInfo() 中所包括的 StaticInfo()（图 12）的 subtitle_language_code 为“JAPANESE”，且 configurable_flag 为 1。因此，这个字幕流 stream#3 为日本字幕数据，且允许改变其显示形式。

[0409] 此外，关于作为片断流文件“00001.PS”中第四基本流的节目流 stream#3，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 0。因而，不存在一组 pts_change_point 和 DynanicInfo()。

[0410] 接下来，在图 26 中，关于片断信息文件“00002.CLP”，presentation_start_time 为 90,000，以及 presentation_end_time 为 27,090,000。因此，根据相应于片断信息文件“00002.CLP”的片断流文件“00002.PS”中所存储的节目流，能够使用有关 300 秒((27,090,000-90,000)/90kHz) 的内容。

[0411] 另外，在片断信息文件“00002.CLP”中，capture_enable_flag_Clip 为 0。因此，不允许（禁止）在相应于片断信息文件“00002.CLP”的片断流文件“00002.PS”中所存储的节目流中多路复用的视频流的二次使用。

[0412] 此外，在图 26 中，在片断信息文件“00002.CLP”中的 number_of_streams 为 4。因此，在相应的片断流文件“00002.PS”中所存储的节目流中，类似于上文所描述的片断流文件“00001.PS”中的情形，将四个基本流多路复用。

[0413] 当将该四个基本流表示为 stream#0、stream#1、stream#2、和 stream#3 时，在图 26 中，在“00002.CLP”栏的较低部分描述了作为 stream#0、stream#1、stream#2、和 stream#3 的四个基本流的 StreamInfo()（图 10）的特定实例。

[0414] 这里，在图 26 中，在片断流文件“00002.PS”中的第一至第四基本流 stream#0 至 #3 的每一个的 StreamInfo() 的内容与在上文所描述的片断流文件“00001.PS”中的第一至第四基本流 stream#0 至 #3 的每一个的 StreamInfo() 的那些相同。相应地，省略了它们的描述。

[0415] 如上文所描述的，因为在片断流文件“00002.PS”中的第一至第四基本流 stream#0 至 #3 的每一个的 StreamInfo() 的内容与在上文所描述的片断流文件“00001.PS”中的第一至第四基本流 stream#0 至 #3 的每一个的 StreamInfo() 的那些相同，在片断流文件“00002.PS”中的第一基本流 stream#0 为视频流，以及第二基本流 stream#1 为 ATRAC 音频流。另外，第三基本流 stream#2 和第四基本流 stream#3 两者都为字幕流。

[0416] 接下来，在图 26 中，关于片断信息文件“00003.CLP”，presentation_start_time 为 90,000，以及 presentation_end_time 为 81,090,000。因此，根据相应于片断信息文件“00003.CLP”的片断流文件“00003.PS”中所存储的节目流，能够使用有关 900 秒 ((81,090,000-90,000)/90kHz) 的内容。

[0417] 另外，在片断信息文件“00003.CLP”中，capture_enable_flag_Clip 为 1。因此，允许与在相应于片断信息文件“00003.CLP”的片断流文件“00003.PS”中所存储的节目流多路复用的视频流的二次使用。

[0418] 此外，在图 26 中，在片断信息文件“00003.CLP”中的 number_of_streams 为 3。因此，在相应的片断流文件“00003.PS”中所存储的节目流中，将三个基本流多路复用。

[0419] 当将该三个基本流表示为 stream#0、stream#1、和 stream#2 时，在图 26 中，在“00003.CLP”栏的较低部分描述了三个基本流 stream#0、stream#1、和 stream#2 的 StreamInfo()（图 10）的特定实例。

[0420] 关于片断流文件“00003.PS”中的第一基本流 stream#0，stream_id 为 0xE0。因而，如图 20 和 22（或图 11）中所描述的，这个基本流 stream#0 为视频流。类似于片断流文件“00001.PS”中的第一基本流 stream#0，private_stream_id 为 0x00。

[0421] 另外，关于作为片断流文件“00003.PS”中第一基本流的视频流 stream#0，在 StreamInfo() 中所包括的 StaticInfo()（图 12）中，picture_size 为“720×480”，frame_rate 为 29.97Hz，以及 cc_flag 为“NO”。因此，这个视频流 stream#0 为具有 720×480 像素和 29.97Hz 的帧周期的视频数据，且不包括任何隐藏字幕数据。

[0422] 此外，关于作为片断流文件“00003.PS”中第一基本流的视频流 stream#0，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 2。因而，在 StreamInfo() 中，描述了两组 pts_change_points 和 DynanicInfo() 项。

[0423] 接下来，关于片断流文件“00003.PS”中的第二基本流 stream#1，stream_id 为 0xE1。因而，如图 20 和 22（或图 11）中所描述的，这个基本流 stream#1 为视频流。为了在作为片断流文件“00003.PS”中第一基本流的视频流 stream#0 与作为第二基本流的视频信号 stream#1 之间进行区别，它们的 stream_id 项分别为 0xE0 和 0xE1。此外，与基本流类似，作为片断流文件“00001.PS”中第一基本流的 stream#0，private_stream_id 是 0x00。

[0424] 另外，关于作为片断流文件“00003.PS”中第二基本流的视频流 stream#1，StreamInfo() 中所包括的 StaticInfo()（图 12）的 picture_size、frame_rate、和 cc_flag 与关于作为第一基本流的视频流 stream#0 的那些相同。因此，作为片断流文件“00003.PS”

中第二基本流的视频流 stream#1 为具有 720×480 像素和 29.97Hz 的帧周期的视频数据，且不包括隐藏字幕数据。

[0425] 此外，关于作为片断流文件“00003.PS”中第二基本流的视频流 stream#1，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 0。因而，不存在一组 pts_change_points 和 DynanicInfo()。

[0426] 接下来，关于作为片断流文件“00003.PS”中第三基本流的 stream#2，stream_id 为 0xBD，且 private_stream_id 为 0x00。因此，如图 20 和 22 中所描述的，这个基本流 stream#2 为 ATRAC 音频流。

[0427] 另外，关于作为片断流文件“00003.PS”中第三基本流的 ATRAC 音频流 stream#2，StreamInfo() 中所包括的 StaticInfo()（图 12）的 audio_language_code、channel_configuration、lfe_existence、和 sampling_frequency 与作为片断流文件“00001.PS”中第二基本流的 ATRAC 音频流 stream#1 中的那些相同。因此，作为片断流文件“00003.PS”中第三基本流的 ATRAC 音频流 stream#2 为日本及立体声音频数据。另外，不包括低重音声道，且采样频率为 48kHz。

[0428] 此外，关于作为片断流文件“00003.PS”中第三基本流的 ATRAC 音频流 stream#2，StreamInfo()（图 10）的 number_of_DynamicInfo() 为 3。因而，在 StreamInfo() 中，描述了三组 pts_change_points 和 DynanicInfo()。

[0429] 接下来，图 27 示出图 10 中所描述的片断信息文件 Clip() 中的 EP_map() 的特定实例。换句话说，图 27 示出图 4 中的片断信息文件“00001.CLP”中的图 14 中的 EP_map() 的特定实例。

[0430] 在图 27 中，EP_map() 中的 number_of_stream_id_entries 为 1。因而，在这个 EP_map() 中，描述关于一个基本流的解码可开始点信息。

[0431] 另外，在图 27 中的 EP_map() 中，stream_id 为 0xE 0。因此，如图 20 和 22 中所描述的，描述了关于由为 0xE0 的 stream_id 所指定的视频流的解码可开始点信息 (PTS_EP_start 和 RPN_EP_start（图 14））。换句话说，图 27 示出关于片断信息文件“00001.CLP”的 EP_map()，且如图 26 中所描述的，在相应于片断信息文件“00001.CLP”的片断流文件“00001.PS”中，在其中 stream_id 为 0xE0 的基本流为片断流文件“00001.PS”的第一视频流 stream#0。因而，EP_map() 中所描述的信息为视频流 stream#0 的解码可开始点的 PTS_EP_start 和 RPN_EP_start。

[0432] 在图 27 中，关于片断流文件“00001.PS”的第一视频流 stream#0 的解码可开始点，描述了从开始起的五个 PTS_EP_start 和 RPN_EP_start 点，且省略了第六和其它 PTS_EP_start 和 RPN_EP_start 点的描述。

[0433] 尽管，在图 27 中的 EP_map() 中，private_stream_id 为 0x00，当 stream_id 表示视频流时，如上文所描述的，private_stream_id 无关（忽略）。

[0434] 接下来，图 28 示出图 25 中所描述 PlayList()#0 和 PlayList()#1（图 5 中的 PlayList()）中的 PlayListMark() 的特定实例。

[0435] 图 28 的上区示出 PlayList()#0 的 PlayListMark()（图 7）。

[0436] 在图 28 的上区，PlayList()#0 中的 PlayListMark() 中的 number_of_PlayList_Marks 为 7。因而，PlayList()#0（的 PlayListMark()）中所包括的 Mark() 项的数目为 7。

[0437] 另外，在图 28 的上区，在 PlayList()#0 中所包括的七个 Mark() 项之中，第一 Mark() 或 Mark()#0 为章标记，因为 mark_type（图 7）为“Chapter”。另外，因为 ref_to_PlayItem_id（图 7）为 0，Mark()#0 属于 PlayList()#0 中所包括的图 25 中的两个 PlayItem#0 和 #1 之间的 PlayItem()#0。因为 mark_time_stamp 为 180,090，Mark#0 为基于 PlayList()#0 中所包括的 PlayItem()#0 而重放的片断流文件中在时间（重放时间）180,090 上的标记。此外，因为 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0，Mark()#0 不与任何一个基本流相关联。另外，Mark()#0 表示其号码为 1 的章，因为 mark_data 为 1。

[0438] 这里，基于 PlayList#0 中所包括的 PlayItem#0 而重放的片断流文件为由在图 25 中所描述的且在 PlayItem#0 的 Clip_Information_file_name 中所描述的“00001.CLP”所指定的片断流文件“00001.PS”。因此，上述由 Mark()#0 的 mark_time_stamp 所表示的时间 180,090 为片断流文件“00001.PS”的时间。

[0439] 在图 28 的上区，在 PlayList()#0 中所包括的七个 Mark() 项之中，第五 Mark() 或 Mark()#4 为类似于第一 Mark#0 的章标记。

[0440] 换句话说，第五 Mark() 或 Mark#4 为章标记，因为 mark_type（图 7）为“Chapter”。另外，Mark#4 属于 PlayList#0 中所包括的图 25 中的两个 PlayItem#0 和 #1 之间的 PlayItem()#1。因为 mark_time_stamp 为 90,000，Mark#4 为基于 PlayList#0 中所包括的 PlayItem#1 而重放的片断流文件的时间 90,000 上的标记。此外，因为 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0，Mark#4 不与任何一个基本流相关联。Mark#4 表示其号码为 2 的章，因为 mark_data 为 2。

[0441] 这里，基于 PlayList#0 中所包括的 PlayItem#1 而重放的片断流文件为由在图 25 中所描述的且在 PlayItem#1 的 Clip_Information_file_name 中所描述的“00002.CLP”所指定的片断流文件“00002.PS”。因而，上述由 Mark#4 的 mark_time_stamp 所表示的时间 90,000 为片断流文件“00002.PS”的时间。

[0442] 另外，在图 28 的上区，在 PlayList#0 中所包括的七个 Mark() 项之中，第二 Mark() 或 Mark#1 为索引标记，因为 mark_type（图 7）为“Index”。另外，因为 ref_to_PlayItem_id（图 7）为 0，Mark#1 属于 PlayList#0 中所包括的图 25 中的两个 PlayItem#0 和 #1 之间的 PlayItem#0。因为 mark_time_stamp 为 5,580,090，Mark#1 为在基于 PlayList#0 中所包括的 PlayItem#0 而重放的片断流文件的时间 5,580,090 的标记。此外，因为 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0，Mark#1 不与任何一个基本流相关联。Mark#1 表示其号码为 1 的索引，因为 mark_data 为 1。

[0443] 这里，如上文所描述的，基于 PlayList#0 中所包括的 PlayItem#0 而重放的片断流文件为片断流文件“00001.PS”。因而，上述由 Mark#1 的 mark_time_stamp 所表示的时间 5,580,090 为片断流文件“00001.PS”的时间。

[0444] 在图 28 的上区，在 PlayList#0 中所包括的七个 Mark() 项之中，为第三 Mark()、第六 Mark()、和第七 Mark() 的 Mark#2、Mark#5、和 Mark#6 为类似于第二 Mark#1 的索引标记。

[0445] 另外，在图 28 的上区，在 PlayList#0 中所包括的七个 Mark() 项之中，第四 Mark() 或 Mark#3 为事件标记，因为 mark_type（图 7）为“Event”。因为 ref_to_PlayItem_id（图 7）为 0，Mark#3 属于 PlayList#0 中所包括的图 25 中的两个 PlayItem#0 和 #1 之间的

PlayItem#0。另外，因为 mark_time_stamp 为 16,380,090，Mark#3 为在基于 PlayList#0 中所包括的 PlayItem#0 而重放的片断流文件的时间 16,380,090 的标记。此外，因为 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0，Mark#3 不与任何一个基本流相关联。因为 mark_type 为 0，Mark#3 产生具有 0 作为自变量的事件。

[0446] 这里，如上文所描述的，基于 PlayList#0 中所包括的 PlayItem#0 而重放的片断流文件为片断流文件“00001.PS”。因而，上述由 Mark#3 的 mark_time_stamp 所表示的时间 16,380,090 为片断流文件“00001.PS”中的时间。

[0447] 在图 28 的上区，在 PlayList#0 的 PlayListMark() 列表外的右边，示出属于 Mark() 的从 PlayItem() 的开始起的每个时间，且在更右边，示出从 PlayList#0 的开始起的每个时间。

[0448] 接下来，在图 28 的下区，示出 PlayList#1 的 PlayListMark()（图 7）。

[0449] 在图 28 的下区，PlayList#1 的 PlayListMark() 中的 number_of_PlayList_Marks 为 3。因而，PlayList#1（的 PlayListMark()）中所包括的 Mark() 项的数目为 7。

[0450] 另外，在图 28 的下区，在 PlayList#1 中所包括的三个 Mark() 项之中，第一 Mark() 或 Mark#0 为章标记，因为 mark_type（图 7）为“Chapter”。另外，因为 ref_to_PlayItem_id（图 7）为 0，Mark#0 属于 PlayList#1 中所包括的图 25 中的一个 PlayItem#0。因为 mark_time_stamp 为 90,000，Mark#0 为在基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件中的时间 90,000 的标记。此外，因为 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0，Mark#0 不与任何一个基本流相关联。因为 mark_type 为 0，Mark#0 的数字表示章号。

[0451] 这里，基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件为由在 PlayItem#0 的 Clip_Information_file_name 中所描述的“00003.CLP”所指定的片断流文件“00003.PS”。因此，上述由 Mark#0 的 mark_time_stamp 所表示 90,000 为片断流文件“00003.PS”中的时间。

[0452] 在图 28 的下区，在 PlayList#1 中所包括的三个 Mark() 项之中，第二 Mark() 或 Mark#1 为事件标记，因为 mark_type（图 7）为“Event”。另外，因为 ref_to_PlayItem_id（图 7）为 0，Mark#1 属于 PlayList#1 中所包括的图 25 中的一个 PlayItem#0。同样，因为 mark_time_stamp 为 27,090,000，Mark#1 为在基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件中的时间 27,090,000 的标记。此外，因为 entry_ES_stream_id 为 0xE0，且 entry_ES_private_stream_id 为 0，Mark#1 与当 stream_id 为 0xE0 时而指定（识别）的基本流相关联，即，如图 20 和 22 中所示，它与视频流相关联。因为 mark_data 为 1，Mark#1 产生具有 1 作为自变量事件。

[0453] 这里，如上文所描述的，基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件为片断流文件“00003.PS”。因而，上述由 Mark#1 的 mark_time_stamp 所表示的时间 27,090,000 为片断流文件“00003.PS”中的时间。

[0454] 另外，在其中 stream_id 为 0xE0，且 Mark#1 与其相关联的视频流为 Mark#1 所属的视频流，且在该视频流中，在图 25 中的 PlayList#1 中所包括的 PlayItem#0 的 Clip_Information_file_name 中所描述的“00003.CLP”中所描述的 stream_id 为 0xE0。换句话说，它为片断流文件“00003.PS”中多路复用的三个基本流 stream#0 至 #2 之中的第一基本

流（视频流）stream#0。

[0455] 接下来，在图 28 的下区，在 PlayList#1 中所包括的三个 Mark() 项之中，第三 Mark() 或 Mark#2 为事件标记，因为 mark_type（图 7）为“Event”。另外，因为 ref_to_PlayItem_id（图 7）为 0，Mark#2 属于 PlayList#1 中所包括的图 25 中的一个 PlayItem#0。同样，因为 mark_time_stamp 为 27,540,000，Mark#1 为在基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件中的时间 27,540,000 的标记。因为 entry_ES_stream_id 为 0xE1，且 entry_ES_private_stream_id 为 0，Mark#2 与当 stream_id 为 0xE1 时所指定（识别）的基本流相关联，即，如图 20 和 22 中所示，它与视频流相关联。另外，因为 mark_data 为 2，Mark#2 产生具有 2 作为自变量的事件。

[0456] 这里，基于 PlayList#1 中所包括的 PlayItem#0 而重放的片断流文件为片断流文件“00003.PS”。因而，上述由 Mark#2 所表示的时间 27,540,000 为片断流文件“00003.PS”中的时间。

[0457] 另外，在其中 stream_id 为 0xE1，且 Mark#2 与其相关联的视频流为 Mark#2 所属的视频流，且在该视频流中，在图 25 中的 PlayList#1 中所包括的 PlayItem#0 的 Clip_Information_file_name 中所描述的“00003.CLP”中所描述的 stream_id 为 0xE1。换句话说，它为由片断信息文件“00003.CLP”所辨识的且在片断流文件“00003.PS”中多路复用的三个基本流 stream#0 至 #2 之中的第二基本流（视频流）stream#1。

[0458] 这里，在图 28 的下区，在 PlayList#1 中的 PlayListMark() 的列表外的右边，示出从 Mark() 所属的 PlayItem() 的开始起的每个时间。

[0459] 在图 28 中，在 mark_data 中描述由章和索引标记所表示的章和索引号。然而，能够通过，例如，计算 PlayListMark() 中章标记和索引标记的数目来辨识由章和索标记所表示的章和索引号，尽管未在 mark_data 中描述它们。

[0460] 盘装置操作的描述

[0461] 接下来，假设记录了如图 25 至 28 中所描述的数据（文件），描述图 1 中的盘装置的操作。

[0462] 在将盘 101 加载到盘驱动器 102 后，通过驱动器接口 114 将表示该加载的消息报告给视频内容重放程序 210，且另外，报告给图 2 中的操作系统 201。当从操作系统 201 接收到指示已经将盘 101 加载到盘驱动器 102 的消息时，视频内容重放程序 210 就开始图 29 中的重放之前的处理。

[0463] “重放之前的处理”

[0464] 特殊地，图 29 为阐明由视频内容重放程序 210 所执行的重放之前的处理的流程图。

[0465] 这里，由该流程图所描述的下列盘驱动器的操作或过程并不总是需要以在流程图表格中所描述的顺序而时序地执行，且可以并行地或分离地执行。然而，在这个规范说明中，为方便起见，根据流程图描述盘驱动器的操作或过程。

[0466] 在重放之前的处理中，在步骤 S101，视频内容重放程序 210 使用操作系统 201 的文件系统功能，关于对于视频内容重放程序 210 盘 101 是否正常，而检查盘 101。

[0467] 如上文所描述的，通过使用视频内容重放程序 210 的文件系统功能，执行盘 101 的访问（文件读取）。如果需要，下文省略其描述。

[0468] 如果在步骤S101确定盘101不正常，即，例如，如果盘101所采用的文件的系统为与操作系统201不相匹配的类型，且如果在根目录中未设置“VIDEO”目录，则视频内容重放程序210确定该文件系统与盘101不相匹配，处理前进到S102，且在完成重放之前的处理之前，图形处理模块219执行错误处理。

[0469] 换句话说，在错误处理中，图形处理模块219产生指示盘101不正常的错误消息（的视频数据），并从视频输出模块220输出该消息，由此显示错误消息。另外，在其它错误处理中，例如，能够从音频输出模块221输出警告声，能够从盘驱动器102卸载盘101等等。

[0470] 可选择地，如果在步骤S101确定盘101是正常的，则处理前进到步骤S103。视频内容重放程序210使用内容数据供给模块213请求并从操作系统201读取盘101（图4）中的“VIDEO”目录中所设置的两个数据文件：“SCRIPT.DAT”文件和“PLAYLIST.DAT”文件，并前进到步骤S104。在步骤S104，将“SCRIPT.DAT”文件供给脚本控制模块211，并将“PLAYLIST.DAT”文件供给播放器控制模块212。

[0471] 在那之后，过程从步骤S104前进到步骤S105直到S107，且播放器控制模块212执行初始化。脚本控制模块211等待脚本控制模块211完成初始化。

[0472] “播放器控制模块212的初始化”

[0473] 在初始化中，在步骤S105，通过分析“PLAYLIST.DAT”文件，播放器控制模块212找到“PLAYLIST.DAT”文件中所使用的片断信息文件的数目和它们的文件名。

[0474] 特殊地，因为在本情形中，“PLAYLIST.DAT”文件为如图25中所示的，且图25中的“PLAYLIST.DAT”文件中的number_of_PlayLists为2，播放器控制模块212辨识存在两个PlayList()项或者第一PlayList#0和第二PlayList#1。另外，因为关于图25中的“PLAYLIST.DAT”文件中的第一PlayList#0，number_of_PlayLists为2，播放器控制模块212辨识，在PlayList#0中，存在两个PlayItem()项或者第一PlayItem#0和第二PlayItem#1。通过参考图25中的“PLAYLIST.DAT”文件中的PlayList#0中所包括的第一PlayItem#0和第二PlayItem#1的每一个的Clip_Information_file_name，播放器控制模块212辨识PlayList#0中所包括的第一PlayItem#0的片断信息文件为“00001.CLP”，以及第二PlayItem#1的片断信息文件为“00002.CLP”。

[0475] 因为number_of_PlayLists为1，关于第二PlayList#1以类似的方式，播放器控制模块212辨识存在一个PlayItem()(PlayItem#0)，并从PlayItem#0中的Clip_Information_file_name辨识PlayItem#0的片断信息文件为“00003.CLP”。

[0476] 在那之后，从步骤S105前进到步骤S106，播放器控制模块212从盘101读取在步骤S105所辨识的片断信息文件。特殊地，它从盘101的“VIDEO”目录中的“CLIP”目录中读取三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”。

[0477] 尽管，在步骤S106的片断信息文件的读取中，仅读取最初所产生的PlayList()的PlayItem的片断信息文件就足够，但在这个实施例中，如上文所描述的，预先读取所有PlayList()项的PlayItem()项的片断信息文件。

[0478] 在步骤S106的处理之后，前进到步骤S107，播放器控制模块212确定在步骤S105中所辨识的片断信息文件的读取的过程是否已成功，并进一步（检查）确定在盘101上是否存在相应于所读取的片断信息文件的片断流文件。特殊地，在步骤S107，确定读取片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”的过程是否已成功，并进一步确

定盘 101 上的“VIDEO”目录下的“STREAM”目录中是否存在片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”，以及在其中仅文件名的扩展名不同的片断流文件“00001.PS”、“00002.PS”、和“00003.PS”。

[0479] 在步骤 S107，如果确定在步骤 S105 中所辨识的片断信息文件的读取已失败，或者确定在盘 101 上不存在相应于片断信息文件的片断流文件，即，例如，如果在盘 101 上未记录用于根据“PLAYLIST.DAT”文件的重放所必需的片断信息文件和片断流文件，则视频内容重放程序 210 确定其与盘 101 不匹配，并前进到步骤 S102。执行上文所描述的错误处理，且重放之前的处理结束。

[0480] 可选择地，如果在步骤 S107 确定，在步骤 S105 中所辨识的片断信息文件的读取过程已成功，并确定在盘 101 上存在相应于片断信息文件的片断流文件，则播放器控制模块 212 结束初始化，并前进到步骤 S108。

[0481] 在步骤 S108，脚本控制模块 211 启动解释和执行“SCRIPT.DAT”文件。

[0482] 作为例子，脚本控制模块 211 执行“SCRIPT.DAT”文件，由此，当命令播放器控制模块 212 重放第一 PlayList()(PlayList#0) 时，执行图 30 中的重放处理。

[0483] “重放处理”

[0484] 特殊地，图 30 为阐明由视频内容重放程序 210 所执行的重放处理的流程图。

[0485] “重放准备处理”

[0486] 在重放处理中，首先，在步骤 S121 和 S122，播放器控制模块 212 执行关于由脚本控制模块 211 请求其重放的 PlayList()，即，第一 PlayList()(PlayList#0) 的重放准备处理。

[0487] 换句话说，在步骤 S121，播放器控制模块 212 辨识第一 PlayList#0 中所包括的第一 PlayItem#0 的 IN_time（图 6），并前进到步骤 S122。播放器控制模块 212 在基于第一 PlayList#0 中所包括的第一 PlayItem#0 而重放的片断流文件“00001.PS”中，检查在其中将要开始重放且相应于 PlayItem#0 的 IN_time 的重放开始位置。

[0488] 这里，当 PlayItem() 的 IN_time（图 6）表示片断流文件的开始时，可以从片断流文件的开始读取节目流。当 IN_time 表示除片断流的开始之外的位置时，执行寻找（检查）相应于 IN_time 的位置并从那里读取节目流是必需的。

[0489] 特殊地，如图 25 中所示，第一 PlayList#0 中所包括的第一 PlayItem#0 的 IN_time 为 180,090。播放器控制模块 212 从基于第一 PlayList#0 中所包括的第一 PlayItem#0 而重放的片断流文件“00001.CLP”的图 27 中的 EP_map() 中，寻找为 PlayItem#0 的 IN_time 的，适合于 180,090 的重放开始位置。

[0490] 换句话说，通过使用对分搜索或诸如此类，播放器控制模块 212 从 EP_map() 中所描述的且表示解码可开始位置的 PTS_EP_start 之中，搜索满足表达式 PTS_EP_start ≤ IN_time 的最大的 PTS_EP_start。搜索不大于 IN_time 的 PTS_EP_start 的原因是由 IN_time 所表示的位置并不总是解码可开始位置。

[0491] 在这种情形，如上文所描述的，IN_time 为 180,090。另外，在基于第一 PlayList#0 中所包括的第一 PlayItem#0 而重放的片断流文件“00001.CLP”中的图 27 中所示的 EP_map() 中，180,090 描述为满足表达式 PTS_EP_start ≤ IN_time 的最大的 PTS_EP_start。因此，在播放器控制模块 212 中，从图 27 中所示的 EP_map() 中，搜索 180,090 的 PTS_EP_

start。

[0492] 另外，在播放器控制模块 212 中，读取为相应于所找到的 RPN_EP_start 的 RPN_EP_start 的（扇区）305，且确定由 305 的 RPN_EP_start 所表示的片断流文件“00001.PS”中的一个位置为重放开始位置。

[0493] 如上文所描述的，在播放器控制模块 212 确定重放开始位置之后，它从步骤 S122 前进到 S123，并控制图形处理模块 219 显示时间代码。根据播放器控制模块 212 的控制，图形处理模块 219 产生并将时间代码（的视频数据）输出至视频输出模块 220。这启动时间代码的显示。

[0494] 这里，在步骤 S123 启动其显示的时间代码，例如，为通过将 PlayList() 的开始转换为 00:00:00（小时：分钟：秒）而获得的一个值。另外，可以在具有时间代码或不具有时间代码的情况下显示章和索引号。

[0495] “分析 PlayListMark()”

[0496] 在步骤 S123 启动时间代码的显示之后，处理前进到步骤 S124。播放器控制模块 212 执行在其中分析由脚本控制模块 211 请求其重放的 PlayList()，即，第一 PlayList() (PlayList#0) 中所描述的 PlayListMark()（图 7）。

[0497] 特殊地，因为在已经读取的“PLAYLIST.DAT”文件中的第一 PlayList#0 的图 28 中上区上所示的 PlayListMark() 中，number_of_PlayList_marks 为 7，播放器控制模块 212 辨识 PlayList#0 中所包括的 Mark() 项为 7。

[0498] 另外，通过分析 PlayList#0 中所包括的图 28 中上区上的七个 Mark() 项，基于 ref_to_PlayItem_id，播放器控制模块 212 辨识在该七个 Mark() 项之中，从第一至第四的四个 Mark() 项属于 PlayList#0 中的第一 PlayItem() (PlayItem#0)。

[0499] 在那之后，播放器控制模块 212 提取属于 PlayList#0 的第一 PlayItem#0 的四个 Mark() 项的 mark_time_stamp 项，并将所提取的项作为在其中元素的数目为 4 的一个排列传递给解码控制模块 214。换句话说，以上述方式，将为在图 28 中上区的七个 Mark() 项之中的从第一至第四的四个 Mark() 项的 mark_time_stamp 项的四个时间点：{180,090}、{5,580,090}、{10,980,090}、和 {16,380,090} 从播放器控制模块 212 传递至解码控制模块 214。在这个时刻，播放器控制模块 212 也通知解码控制模块 214 这些时间的属性为“Mark Process”。当由定时器单元 214A 所度量的时间与“MarkProcess”属性的时间一致时，解码控制模块 214 通知播放器控制模块 212 表示该一致的消息、与“Mark Process”属性的时间一致的时间、和“MarkProcess”属性。

[0500] “确定将要重放的基本流”

[0501] 接下来，从步骤 S124 前进到 S125，播放器控制模块 212 确定将要重放的基本流。

[0502] 换句话说，因为在图 26 中的片断信息文件“00001.CLP”中 number_of_streams 为 4，其文件名在第一 PlayList#0 的第一 PlayItem#0（图 25）的 Clip_Information_file_name 中描述，该 PlayList#0 为由脚本控制模块 211 请求其重放的 PlayList() 的所以播放器控制模块 212 辨识在相应的片断流文件“00001.PS”中多路复用了四个基本流。另外，通过顺序地检查片断信息文件“00001.CLP”的 StaticInfo() 的 stream_id 和图 26 中必需的 private_stream_id，播放器控制模块 212 辨识四个基本流为一个视频流、一个 ATRAC 音频流、和两个字幕流。换句话说，辨识具有属性且在片断流文件“00001.PS”中多路复用的基

本流的数目。

[0503] 在切换正重放的基本流（音频交换、字幕交换等等）中使用具有属性且在片断流文件中多路复用的基本流的数目的信息。另外，在片断流文件中可能不存在字幕流（在内容中不包括字幕），且在确定字幕流是否存在中，使用具有“Subtitle Stream”属性的基本流的数目的信息。

[0504] 如上文所描述的，基于检查 StaticInfo() 的结果，播放器控制模块 212 选择并确定将要重放的基本流。然而，因为，在这种情形，片断流文件“00001.PS”中多路复用的四个基本流仅包含仅一个“Video Stream”属性基本流和仅一个“Audio Stream”属性基本流，“Video Stream”属性基本流和“Audio Stream”属性基本流是不可选择的，因此确定该一个视频流和一个音频流（ATRAC 音频流）为将要重放的基本流。

[0505] 另外，关于“Subtitle Stream”属性基本流，在片断流文件“00001.PS”中多路复用的四个基本流之中存在两个流。因而，选择并确定该两个字幕流的一个为将要重放的基本流。这里，在两个字幕流之间，例如，选择片断信息文件“00001.CLP”中以出现的顺序的第一个字幕流。

[0506] 如上文所描述的，在辨识片断流文件“00001.PS”中多路复用的四个基本流的属性和数目的情形中，需要分别制定该四个基本流。相应地，播放器控制模块 212 使用 stream_id 和必需的 private_stream_id 来指定片断流文件“00001.PS”中多路复用的四个基本流。

[0507] 特殊地，如参考图 26 关于片断信息文件“00001.CLP”所描述的，通过使用为 0xE0 的 stream_id，播放器控制模块 212 指定为在片断流文件“00001.PS”中多路复用的四个基本流之中的“Video Stream”属性基本流的视频流。

[0508] 另外，如参考图 26 关于片断信息文件“00001.CLP”所描述的，通过使用为 0xBD 的 stream_id 和为 0x00 的 private_stream_id，播放器控制模块 212 指定为在片断流文件“00001.PS”中多路复用的四个基本流之中的“AudioStream”属性基本流的 ATRAC 音频流。

[0509] 此外，如参考图 26 关于片断信息文件“00001.CLP”所描述的，通过使用为 0xBD 的 stream_id、为 0x80 的 private_stream_idstream_id，为 0xBD 的 stream_id、和为 0x81 的 privae_stream_id，播放器控制模块 212 指定两个字幕流，其为在片断流文件“00001.PS”中多路复用的四个基本流之中的“Subtitle Stream”属性基本流。

[0510] 如上文所描述的，基于描述为相应于片断流文件的片断信息文件的元数据的 stream_id 和 private_stream_id 的结合，能够指定在片断流文件中多路复用的基本流。

[0511] 这里，stream_id 和 private_stream_id 的结合为用于扩展 MPEG-2 系统的多路复用而供给的机制。通过在元数据中，使用 stream_id 和 private_stream_id 的结合来指定基本流，确保能够指定基本流。另外，即使在将来扩展了 private_stream_id 的含意，以增加相应的基本流的类型（属性）的数目，也能够使用本机制，而不必改变。因而，该结合优胜在扩展性上。

[0512] 换句话说，例如，在 BD（蓝光光盘）标准中，使用根据 MPEG-2 标准的传送流的 PID（信息包 ID）来指定数据。因而，规范说明受 MPEG-2 标准限制。同样，尽管，例如，在 DVD-Video 标准中，定义了类似于 private_stream_id 的 sub_stream_id，但未形成 sub_stream_id，以便在用于指定流的数据库中描述，而其仅形成以至在描述 8 或 32 个流的信息的固定区域中描述（参见，例如，VI4-49，表 4.2.1-2(VTS-AST_ATRT) 和 VI4-52，表

4.2.1-3(VTS_SPST_ATRT) 等等),所以其具有很低的扩展性。

[0513] 与那不同,能够将 stream_id 和 private_stream_id 的结合描述与能够由,例如,在其中描述元数据的图 10 中的片断信息文件 Clip() 中的 number_of_streams 所表示的数目一样多的次数。因而,不论基本流的数目(在能够由 number_of_streams 所表示的数目的范围内),基于 stream_id 和 private_stream_id 的结合,能够将在片断流文件中多路复用的基本流指定为片断信息文件 Clip() 中所描述的元数据。

[0514] 在这个实施例中,在图 10 中的片断信息文件中使用 stream_id 和 private_stream_id 的结合来指定在相应的片断流文件中多路复用的基本流,且另外,例如,将其用作图 7 中的 PlayListMark() 中的 entry_ES_stream_id 和 entry_ES_private_stream_id 的结合,以指定用于与 Mark() 相关联的基本流。此外,例如,在图 14 中的 EP_map() 中,使用 stream_id 和 private_stream_id 的结合来指定描述解码可开始点的信息的基本流。

[0515] “控制输出属性”

[0516] 在那之后,从步骤 S125 前进到步骤 S126,播放器控制模块 212 执行控制将要重放的基本流,即,在步骤 S125 所确定的将要重放的基本流的输出属性。

[0517] 特殊地,关于将要重放的基本流,即,在步骤 S125 所确定的将要重放的视频流、ATRAC 音频流、和字幕流的每一个,播放器控制模块 212 检查表示在其中描述输出属性的 DynamicInfo()(图 13) 的数目的 number_of_DynamicInfo(图 10)。

[0518] 这里,在这种情形,如关于参考图 26 的“00001.CLP”所描述的,将要重放的视频流、ATRAC 音频流、和字幕流为在片断流文件“00001.PS”中多路复用的基本流,且其 number_of_DynamicInfo 项为 0。在关于将要重放的所有的基本流,number_of_DynamicInfo 项都为 0 的这样的情形中,播放器控制模块 212 并不特别地执行关于控制将要重放的基本流的输出属性的处理的处理。

[0519] 稍后描述当关于将要重放的基本流,number_of_DynamicInfo 不为 0 时所执行的用于控制该基本流的输出属性的处理。

[0520] “用于开始重放的准备”

[0521] 在步骤 S126 的处理之后,前进到步骤 S127,播放器控制模块 212 执行用于开始将要重放的基本流的重放的准备。

[0522] 换句话说,播放器控制模块 212 将片断流文件“00001.PS”的文件名和在步骤 S122 确定为重放开始位置的 EP_map() 中所描述的 RPN_EP_start(= 305) 供给内容数据供给模块 213,在该片断流文件中多路复用将要重放的基本流。

[0523] 另外,在开始将片断流文件“00001.PS”中所存储的节目流供给缓冲器控制模块 215 之前,在该片断流文件中将要重放的基本流多路复用,播放器控制模块 212 将该缓冲器控制模块 215 初始化。

[0524] 特殊地,在缓冲器控制模块 215(图 3) 中,给在数据开始指针存储单元 231 中所存储的数据开始指针、在数据写入指针存储部分 232 中所存储的数据写入指针、在视频读取指针存储部分 241 中所存储的视频读取指针、在音频读取指针存储部分 251 中所存储的音频读取指针、和在字幕读取指针存储部分 262 中所存储的字幕读取指针分配相等的值。

[0525] 相应地,在数据开始指针存储单元 231 中所存储的数据开始指针和在数据写入指针存储部分 232 中所存储的数据写入指针表示缓冲器控制模块 215 的缓冲器 215A 中的相

同位置。这指示出在其中有效数据不存储在缓冲器 215A 中的状态。

[0526] 此外，播放器控制模块 212 将作为识别（指定）将要重放的基本流的识别信息的 stream_id 以及 private_stream_id（如果需要），供给缓冲器控制模块 215。

[0527] 换句话说，如上文所描述的，在将要重放的基本流之中，具有“VideoStream”属性的视频流由为 0xE0 的 stream_id 而指定，具有“Audio Stream”属性的 ATRAC 音频流由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 而指定，以及具有“Subtitle Stream”属性的字幕流由为 0xBD 的 stream_id 和为 0x80 的 private_stream_id 而指定。播放器控制模块 212 将 stream_id 和 private_stream_id 供给缓冲器控制模块 215。

[0528] 在缓冲器控制模块 215（图 3）中，视频读取功能单元 233 将关于来自播放器控制模块 212 的视频流的为 0xE0 的 stream_id 存储在 stream_id 寄存器 242 中。另外，音频读取功能单元 234 分别将来自播放器控制模块 212 的为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 存储在 stream_id 寄存器 252 和 private_stream_id 寄存器 253 中。此外，字幕读取功能单元 235 分别将来自播放器控制模块 212 的为 0xBD 的 stream_id 和为 0x80 的 private_stream_id 存储在 stream_id 寄存器 263 和 private_stream_id 寄存器 264 中。

[0529] 关于更进一步的处理，播放器控制模块 212 存储供给缓冲器控制模块 215 的将要重放的基本流的 stream_id 和 private_stream_id。在其中有消息请求稍后描述的流切换的情形和在稍后描述的标记处理中，播放器控制模块 212 使用 stream_id 和 private_stream_id 来指定目前正重放的流。

[0530] 关于初始化缓冲器控制模块 215（图 3），播放器控制模块 212 也在字幕读取功能部分 261 中设定字幕读取标志，该字幕读取标志具有根据在片断流文件的一个值，在该中片断流文件多路复用将要重放的基本流。

[0531] 换句话说，在这种情形，因为在其中将将要重放的基本流多路复用的片断流文件“00001.PS”中包括字幕流，为了使字幕读取功能单元 235 起作用，在字幕读取功能部分 261 中设定具有为 1 的值的字幕读取功能标志。当在其中将将要重放的基本流多路复用的片断流文件中不包括字幕流时，在字幕读取功能部分 261 中设定具有为 0 的值的字幕读取功能标志。在这种情形，字幕读取功能单元 235 不起作用（不特别地执行处理）。

[0532] 另外，播放器控制模块 212 将由脚本控制模块 211 请求其重放的第一 PlayList#0 中所包括的第一 PlayItem#0（图 25）的作为 IN_time 的 180,090 和作为 OUT_time 的 27,180,090 供给解码控制模块 214。在解码控制模块 214 中，使用 IN_time 来控制开始基于 PlayItem() 而重放的片断的解码，以及使用 OUT_time 来结束该片断的解码，且另外，控制稍后所描述的 PlayItem 切换。

[0533] 另外，播放器控制模块 212 初始化关于图形处理模块 219 的字幕显示形式的引导。换句话说，播放器控制模块 212 控制图形处理模块 219，以便将字幕流显示形式改变为，例如，默认显示形式。

[0534] “开始数据读取”

[0535] 在那之后，从步骤 S127 前进到 S128，播放器控制模块 212 控制内容数据供给模块 213，由此，通过操作系统 201 的功能，内容数据供给模块 213 读取存储节目流的片断流文件，其中多路复用将要重放的基本流。换句话说，通过指明位于盘 101（图 4）上的“VIDEO”

目录中的“STREAM”目录中的片断流文件“00001.PS”，并进一步指明扇区 305 作为步骤 S122 中所确定重放开始位置，内容数据供给模块 213 请求操作系统 201 执行文件读取。另外，内容数据供给模块 213 将从盘 101 所读取的数据的供给引导至缓冲器控制模块 215。

[0536] 以上述方式，开始从盘 101 的片断流文件“00001.PS”中所存储的节目流的读取，并将该节目流供给缓冲器控制模块 215。

[0537] 缓冲器控制模块 215（图 3）将在从盘 101 读取之后而供给的节目流写入由数据写入指针存储部分 232 的数据写入指针所表示的位置，且数据写入指针以已经写入的数据的大小而增加。

[0538] 这里，以下，除非另外声明，如果缓冲器控制模块 215 中的缓冲器 215A 具有空余空间，则内容数据供给模块 213 从盘 101 读取数据，并将该数据供给和存储在缓冲器控制模块 215 中的缓冲器 215A 中。因此，假设缓冲器 215A 不断地存储足够的数据。

[0539] “启动解码器控制”

[0540] 如上文所描述的，在启动从盘 101 的数据的读取，且开始在缓冲器控制模块 215 中的缓冲器 215A 中存储数据之后，过程从步骤 S128 前进到 S129。通过控制视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218，在解码操作之前的一个阶段中，解码控制模块 214 启动从缓冲器 215A 的数据的读取。

[0541] 换句话说，这允许视频解码器控制模块 216 获得缓冲器 215A 中所存储的一个视频存取单元、添加到该视频存取单元上的 PTS 和 DTS（如果需要，以下称为时间戳）、和充当在紧邻解码可开始点之前所设置的 private_stream_2 的 PES_packet() 中所描述的信息（如果需要，以下称为附加信息）的 pic_struct_copy、au_ref_flag、和 AU_length，其为在将数据请求发布到缓冲器控制模块 215（图 3）中的视频读取功能单元 233 之后，响应该数据请求而从缓冲器控制模块 215 传递。无论何时视频解码器控制模块 216 获得了视频存取单元，都将时间戳从视频解码器控制模块 216 传递至解码控制模块 214。

[0542] 另外，音频解码器控制模块 217 也获取缓冲器 215A 中所存储的一个 (ATRAC) 音频存取单元和添加到该音频存取单元上的，在将数据请求发布到缓冲器控制模块 215（图 3）中的音频读取功能单元 234 之后，响应该数据请求而从缓冲器控制模块 215 传递的时间戳 (PTS、DTS)。无论何时音频解码器控制模块 217 获得了音频存取单元，都将时间戳从音频解码器控制模块 217 传递至解码控制模块 214。

[0543] 此外，字幕解码器控制模块 218 获取缓冲器 215A 中所存储的一个字幕存取单元和添加到该字幕存取单元上的，在将数据请求发布到缓冲器控制模块 215（图 3）中的字幕读取功能单元 235 之后，响应该数据请求而从缓冲器控制模块 215 传递的时间戳。无论何时字幕解码器控制模块 218 获得了字幕存取单元，都将时间戳从字幕解码器控制模块 218 传递至解码控制模块 214。另外，当将要重放的基本流中不存在字幕流时，且当在缓冲器 215A 中未存储字幕存取单元时，不从缓冲器控制模块 215 传递数据至字幕解码器控制模块 218。

[0544] 这里，无论何时从缓冲器控制模块 215 请求数据，视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 每一个都将响应该数据请求的结果传递至解码控制模块 214。

[0545] 另外，稍后描述在从缓冲器控制模块 215 传递数据至视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 的情形中，从缓冲器 215A 的数据的读取

的详细情况。

[0546] “启动解码”

[0547] 如上文所描述的，当视频解码器控制模块 216、音频解码器控制模块 217、或字幕解码器控制模块 218 启动从缓冲器控制模块 215 中的缓冲器 215A 读取数据时，过程从步骤 S129 前进到 S130，并启动数据的解码。

[0548] 换句话说，基于为 PlayList#0 中所包括的第一 PlayItem#0 的 IN_time 且在步骤 S127 从播放器控制模块 212 而供给的 180,090，以及如在步骤 S129 中所描述的从视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 而传递的时间戳，以如果需要建立同步而移位的定时，命令视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 启动解码。

[0549] 例如，在日本专利 No.3496725 中描述了用于以用于建立同步而移位的定时命令解码的启动的方法。简单来说，存在一种方法，在其中，在从视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 所传递的时间戳之中，设定最小的值作为由定时器单元 214A 所度量的时间的初始值，启动时间度量，且在由定时器单元 214A 所度量的时间与时间标志一致的时刻，命令解码的启动。

[0550] 视频解码器控制模块 216 从解码控制模块 214 接收解码启动指令；响应该指令，将从缓冲器控制模块 215（图 3）中的视频读取功能单元 233 所获得的一个视频存取单元传递给视频解码器 116（图 1）；并控制视频解码器 116 执行解码。另外，视频解码器控制模块 216 将作为视频解码器 116 的解码的结果而获得的视频数据供给图形处理模块 219。

[0551] 随后，视频解码器控制模块 216 使用视频解码器 116 顺序地将从缓冲器控制模块 215 中的视频读取功能单元 233 所获得的每个视频存取单元解码，并将作为解码的结果而获得的视频数据供给图形处理模块 219。

[0552] 另外，同样音频解码器控制模块 217 从解码控制模块 214 接收解码启动指令。响应该指令，音频解码器控制模块 217 将从缓冲器控制模块 215（图 3）中的音频读取功能单元 234 所获得的一个音频存取单元传递给音频解码器 117（图 1），并控制音频解码器 117 执行解码。此外，音频解码器控制模块 217 将作为音频解码器 117 的解码的结果而获得的音频数据供给音频输出模块 221。

[0553] 随后，音频解码器控制模块 217 使用音频解码器 117 顺序地将从缓冲器控制模块 215 中的音频读取功能单元 234 所获得的每个音频存取单元解码，并将作为解码的结果而获得的音频数据供给音频输出模块 221。

[0554] 同样字幕解码器控制模块 218 从解码控制模块 214 接收解码启动指令。响应该指令，字幕解码器控制模块 218 使用其内部字幕解码软件将从缓冲器控制模块 215（图 3）中的字幕读取功能单元 235 所获得的一个字幕存取单元解码，并将作为解码的结果而获得的字幕数据（字幕图像数据）供给图形处理模块 219。

[0555] 随后，字幕解码器控制模块 218 使用其内部字幕解码软件顺序地将从缓冲器控制模块 215 中的字幕读取功能单元 235 所获得的每个字幕存取单元解码，并将作为解码的结果而获得的字幕数据供给图形处理模块 219。

[0556] “图形处理”

[0557] 在那之后，过程从步骤 S130 前进到 S131，且音频内容重放程序 210 对如上文所描

述的从视频解码器控制模块 216 而供给的视频数据，以及如果需要，从字幕解码器控制模块 218 而供给的字幕数据，执行图形处理。

[0558] 换句话说，根据来自播放器控制模块 212 的显示形式指令，首先图形处理模块 219 对来自字幕解码器控制模块 218 的字幕数据执行字幕处理，例如放大和缩小。当未从播放器控制模块 212 接收到显示形式指令时，或者当接收到默认显示形式指令时，图形处理模块 219 直接存储来自图形处理模块 219 字幕数据。

[0559] 另外，通过添加来自视频解码器控制模块 216 的视频数据，和来自字幕解码器控制模块 218 的字幕数据或在字幕处理之后所获得的字幕数据，图形处理模块 219 获得在其中字幕数据覆盖在来自视频解码器控制模块 216 的视频数据上的输出视频数据，并将该数据供给视频输出模块 220。

[0560] 当播放器控制模块 212 命令显示像例如，菜单、消息、时间代码、章、或索引那样的信息时，图形处理模块 219 以覆盖在输出视频数据上的形式产生并将该信息供给视频输出模块 220。

[0561] “输出处理”

[0562] 在步骤 S131 的处理之后，处理前进到步骤 S132。如步骤 S131 中所描述的，视频输出模块 220 顺序地将从图形处理模块 219 所供给的输出视频数据存储在 FIFO 220A 中，并以预定的输出速率顺序地输出在 FIFO 220A 中所存储的输出视频数据。

[0563] 尽管只要 FIFO 220A 的存储容量（剩余量）足够，视频输出模块 220 就从图形处理模块 219 接收输出视频数据，但如果存储容量不足，则其请求图形处理模块 219 停止接收输出视频数据。这促使图形处理模块 219 停止处理，并请求视频解码器控制模块 216 和字幕解码器控制模块 218 停止处理。这停止视频解码器控制模块 216 和字幕解码器控制模块 218 的处理。

[0564] 在请求图形处理模块 219 停止接收输出视频数据之后，视频输出模块 220 请求图形处理模块 219，在提前了来自 FIFO 220A 的输出视频数据的输出且 FIFO 220A 恢复具有足够的空间的时间接收输出视频数据。类似于停止接收输出视频数据的请求，从图形处理模块 219 报告这个请求给视频解码器控制模块 216 和字幕解码器控制模块 218。这促使图形处理模块 219，以及视频解码器控制模块 216 和字幕解码器控制模块 218 重新开始已停止的处理。

[0565] 另外，如上文所描述的，同样音频输出模块 221 顺序地将从音频解码器控制模块 217 所供给的音频数据存储在 FIFO 221A 中，并以预定的输出速率（采样频率）顺序地输出在 FIFO 221A 中所存储的音频数据。

[0566] 尽管只要 FIFO 221A 的存储容量（剩余量）足够，音频输出模块 221 就从音频解码器控制模块 217 接收音频数据，但如果存储容量不足，则音频输出模块 221 请求音频解码器控制模块 217 停止接收音频数据。这促使音频解码器控制模块 217 停止处理。

[0567] 在请求音频解码器控制模块 217 停止接收音频数据之后，音频输出模块 221 请求音频解码器控制模块 217，在提前了来自 FIFO 221A 的音频数据的输出且 FIFO 221A 恢复具有足够的空间的时间接收音频数据。这促使音频解码器控制模块 217 重新开始已停止的处理。

[0568] 如上文所描述的，随着从视频输出模块 220 和音频输出模块 221 输出数据，执行基

本流的解码。

[0569] 当图 1 中的盘装置重放盘 101 时，整个处理或操作的流程为如图 29 和 30 中所描述的。下文描述当盘装置重放盘 101 时的其它处理或操作。

[0570] “PlayItem 改变”

[0571] 如图 29 和 30 中所描述的，开始图 25 中第一 PlayList#0 中的第一 PlayItem#0。根据 PlayItem#0，在第一 PlayItem#0 的重放结束后，第二 PlayItem#1 开始。换句话说，执行在其中 PlayItem 从 PlayItem#0 改变为 PlayItem#1 的 PlayItem 改变。

[0572] 相应地，参考图 31 中的流程图描述 PlayItem 改变的过程。

[0573] 在如图 29 和 30 中所描述的开始了图 25 中 PlayList#0 中的第一 PlayItem#0（的片断）的重放之后，当重放该第一 PlayItem#0 时，解码控制模块 214（图 2）连续不断地确认由内嵌定时器单元 214A 所度量的时间。

[0574] “PlayItem#0 的重放的结束”

[0575] 当由定时器单元 214A 所度量的时间等于为第一 PlayItem#0 的 OUT_time 且为在图 30 中的步骤 S127 从播放器控制模块 212 所供给的 27,180,090（图 25）时，在步骤 S151，解码控制模块 214 执行解码中断控制，以结束 PlayItem#0 的重放。

[0576] 换句话说，解码控制模块 214 通过操作视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 而停止解码操作。另外，解码控制模块 214 控制视频输出模块 220 连续不断地输出目前正输出的视频数据。

[0577] 另外，解码控制模块 214 通知播放器控制模块 212 指示第一 PlayItem#0 的重放已经结束的消息。

[0578] “启动 PlayItem#1 的重放”

[0579] 如上文所描述的，在图 29 中的步骤 S105，播放器控制模块 212 辨识在第一 PlayList#0 中，存在第一 PlayItem#0 和第二 PlayItem#1。相应地，当解码控制模块 214 通知指示第一 PlayItem#0 的重放已经结束的消息时，播放器控制模块 212 从步骤 S151 前进到 S152，并启动类似于上文所描述的关于第一 PlayItem#0 的情形的第二 PlayItem#1 的重放。

[0580] 换句话说，纵观用于重放 PlayItem#1 的过程，类似于图 30 中的步骤 S122 中的情形，首先，播放器控制模块 212 确定 EP_map() 中所描述的 RPN_EP_start 项的一个为重放开始位置。

[0581] 另外，播放器控制模块 212 执行：属于第二 PlayItem#1 的 Mark() 的辨识，如图 30 中的步骤 S124 中所描述的；具有属性的且在基于 PlayItem#1 而重放的片断流文件“00002.PS”中多路复用的基本流的数目的辨识，如图 30 中的步骤 S125 中所描述的；以及另外，将要重放的基本流的确定。

[0582] 播放器控制模块 212 执行类似于图 30 中的步骤 S127 的情形中的那个的处理。

[0583] 换句话说，播放器控制模块 212 将确定为重放开始位置的 EP_map() 的 RPN_EP_start 和在其中将将要重放的基本流多路复用的片断流文件的文件名，即，在这种情形，在相应于第二 PlayItem#1（图 25）的 Clip_Information_file_name 中所描述的“00002.CLP”的片断流文件“00002.PS”的文件名，供给内容数据供给模块 213。

[0584] 此外，在启动将在其中将将要重放的基本流多路复用的片断流文件“00002.PS”中

所存储的节目流供给缓冲器控制模块215之前，播放器控制模块212将缓冲器控制模块215初始化。

[0585] 特殊地，这在缓冲器控制模块215(图3)中，将相等的值分配给在数据开始指针存储单元231中所存储的数据开始指针、在数据写入指针存储部分232中所存储的数据写入指针、在视频读取指针存储部分241中所存储的视频读取指针、在音频读取指针存储部分251中所存储的音频读取指针、和在字幕读取指针存储部分262中所存储的字幕读取指针。

[0586] 此外，播放器控制模块212将作为识别将要重放的基本流的识别信息的stream_id，以及private_stream_id(如果需要)，供给缓冲器控制模块215。

[0587] 在缓冲器控制模块215(图3)中，视频读取功能单元233将来自播放器控制模块212的关于将要重放的基本流之中的视频流的stream_id存储在stream_id寄存器242中。同样，音频读取功能单元234分别将来自播放器控制模块212的将要重放的基本流之中的音频流的stream_id和private_stream_id存储在stream_id寄存器252和private_stream_id寄存器253中。

[0588] 因为在其中将目前将要重放的基本流多路复用的片断流文件“00002.PS”中包括字幕流，将将要重放的基本流之中的字幕流的stream_id和private_stream_id从播放器控制模块212供给字幕读取功能单元235，且字幕读取功能单元235分别将该stream_id和private_stream_id存储在stream_id寄存器263和private_stream_id寄存器264中。

[0589] 在缓冲器控制模块215(图3)的初始化中，播放器控制模块212在字幕读取功能部分261中设定具有根据剪辑片断流文件的一个值的字幕读取功能标志，在该片断流文件中多路复用将要重放的基本流。

[0590] 换句话说，在这种情形，因为在其中将将要重放的基本流多路复用的片断流文件“00002.PS”中包括字幕流，为了使得字幕读取功能单元235起作用，在字幕读取功能部分261中设定其值为1的字幕读取功能标志。

[0591] 另外，播放器控制模块212将将要重放的第二PlayItem#1的IN_time 90,000和OUT_time 27,090,000(图25)供给解码控制模块214。

[0592] 此外，播放器控制模块212为图形处理模块219初始化字幕显示形式指令。换句话说，播放器控制模块212控制图形处理模块219，以便字幕流显示形式改变为默认显示形式。

[0593] 当关于将要重放的基本流，configurable_flag(图12)为1，其表示允许改变显示形式时，从播放器控制模块212至图形处理模块219的字幕流显示形式指令可以不改变当前的显示形式。

[0594] 随后，类似于第一PlayItem#0的重放，执行第二PlayItem#1的重放。当正在重放第二PlayItem#1时，解码控制模块214连续不断地确认由内嵌定时器单元214A所度量的时间，且当由定时器单元214A所度量的时间等于从播放器控制模块212所供给的第二PlayItem#1的OUT_time 27,090,000(图25)时，通过执行类似于步骤S151中的情形的解码中断控制，而结束PlayItem#1的重放。

[0595] “时间代码的显示”

[0596] 接下来，如上文所描述的，在图30中的步骤S123，启动时间代码的显示，且顺序地更新时间的显示。

[0597] 相应地，参考图 32 中的流程图描述用于显示时间代码的过程。

[0598] 当内嵌定时器单元 214A 度量一秒时，在步骤 S171，解码控制模块 214（图 2）将指示一秒已经过去的消息和由定时器单元 214A 所度量的当前时间供给播放器控制模块 212，并前进到步骤 S172。在步骤 S172，播放器控制模块 212 从解码控制模块 214 接收消息和当前时间，将当前时间转换为时间代码，并前进到步骤 S173。

[0599] 在步骤 S173，播放器控制模块 212 控制图形处理模块 219 显示在步骤 S172 所获得的时间代码，并返回到步骤 S171。

[0600] 这在每秒都更新时间代码。更新时间代码的间隔并不局限于一秒。

[0601] “流切换”

[0602] 接下来，如图 26 中所示，在基于图 1 中所描述的第一 PlayList#0 中所包括的第一 PlayItem#0 而重放的片断流文件“00001.PS”和基于第二 PlayItem#1 而重放的片断流文件“00002.PS”中，将两个字幕流多路复用。

[0603] 在片断流文件中将具有相同的属性的多个基本流多路复用的这样的情形中，关于将要重放的基本流，能够执行流切换，其中，在具有相同的属性的基本流之中，将一个切换至另一个。

[0604] 相应地，参考图 33 中的流程图描述关于流切换的过程。

[0605] 供给流切换请求，以便，例如，当流切换指令描述为“SCRIPT.DAT”文件（图 4）中的脚本程序时，脚本控制模块 211 执行脚本程序，或者用户操作遥控器。

[0606] 换句话说，当执行在其中描述流切换指令的脚本程序时，脚本控制模块 211 将请求流切换的消息供给播放器控制模块 212。另外，当通过用户操作遥控器从遥控器接收到流切换指令信号时，输入接口 115 将流切换请求消息供给播放器控制模块 212。

[0607] 作为例子，目前，如果将请求字幕流的切换的字幕切换消息供给播放器控制模块 212，在步骤 S191，播放器控制模块 212 检查在图 30 中的步骤 S125 中所执行的将要重放的基本流的确定中所辨识的字幕流的数目。

[0608] 当检查字幕流的数目的结果指示一或更少时，播放器控制模块 212 忽略字幕流切换消息。因而，不执行随后的步骤 S192 至 S194 中的处理。

[0609] 另外，当字幕流的数目为二或更大时，处理顺序地前进到步骤 S192 至 S194，将要重放的字幕流从目前正重放的字幕流切换至另一个字幕流。

[0610] 换句话说，在步骤 S192，播放器控制模块 212 在片断信息文件中指定目前正重放的字幕流。特殊地，例如，假设基于图 25 中所描述的第一 PlayList#0 中所包括的第二 PlayItem#1，重放具有为 0xBD 的 stream_id 和为 0x80 的 private_stream_id 且在片断流文件“00002.PS”中多路复用的字幕流，在步骤 S192，指定目前正重放的字幕流为在片断流文件“00002.PS”中多路复用的两个字幕流之间的图 26 中的片断信息文件“00002.CLP”中的第三字幕流或 stream#2。

[0611] 前进到步骤 S193，播放器控制模块 212 在片断信息文件中识别（指定）紧接着在步骤 S 192 中所指定的字幕流的字幕流为将要重放的下一个字幕流。因为，在图 26 中，在片断信息文件“00002.CLP”中，紧接着第三字幕流或 stream#2 的字幕流为第四字幕流或 stream#3，在步骤 S193，辨识第四字幕流或 stream#3 为将要重放的下一个字幕流。

[0612] 当指定目前正重放的字幕流为在片断流文件“00002.PS”中多路复用的两个字幕

流之间的图 26 中的片断信息文件“00002.CLP”中的第四字幕流或 Stream#3 时，例如，辨识第三字幕或 stream#2 为将要重放的下一个字幕流。

[0613] 在那之后，前进到步骤 S194，播放器控制模块 212 将在步骤 S193 中所辨识的将要重放的下一个字幕流的 stream_id 和 private_stream_id 供给缓冲器控制模块 215（图 3）中的字幕读取功能单元 235，并命令字幕读取功能单元 235 使用字幕存取单元中的来自缓冲器 215A 的下一个读取的 stream_id 和 private_stream_id。

[0614] 在缓冲器控制模块 215 中的字幕读取功能单元 235（图 3）中，分别在 stream_id 寄存器 263 和 private_stream_id 寄存器 264 中设定在步骤 S194 中从播放器控制模块 212 所供给的新的 stream_id 和 private_stream_id，并关于由在 stream_id 寄存器 263 和 private_stream_id 寄存器 264 中分别设定的新的 stream_id 和 private_stream_id 所指定的字幕存取单元而执行从缓冲器 215A 的下一个和随后的读取。

[0615] 如上文所描述的，将要重放的字幕流从目前正重放的字幕流切换到另一个字幕流。

[0616] 缓冲器控制模块 215 的处理

[0617] 接下来，参考图 34 至 38 描述缓冲器控制模块 215（图 3）的处理，即，写入数据至缓冲器 215A 及从缓冲器 215A 的数据的读取。

[0618] 如图 3 中所描述的，缓冲器控制模块 215 具有五个指针，用于在缓冲器 215A 中读取和写入数据。

[0619] 特殊地，如图 34 和 35 所示，缓冲器控制模块 215 具有存储在数据开始指针存储单元 231 中的数据开始指针、存储在数据写入指针存储部分 232 中的数据写入指针、存储在视频读取指针存储部分 241 中的视频读取指针、存储在音频读取指针存储部分 251 中的音频读取指针、和存储在字幕读取指针存储部分 262 中的字幕读取指针。

[0620] 图 34 和 35 省略了图 3 中视频读取功能单元 233 的 stream_id 寄存器 242 和 au_information() 寄存器 243，音频读取功能单元 234 的 stream_id 寄存器 252 和 private_stream_id 寄存器 253，字幕读取功能单元 235 的字幕读取功能部分 261、stream_id 寄存器 263、和 private_stream_id 寄存器 264 的解释说明。

[0621] 数据开始指针存储单元 231 中所存储的数据开始指针表示保留在缓冲器 215A 中的最旧的数据（需要被读取且为还未被读取的数据中最旧的数据的数据）的位置。数据写入指针存储部分 232 中所存储的数据写入指针表示在缓冲器 215A 中在其写入数据的位置，且这个位置为在缓冲器 215A 中写入最新的数据的位置。

[0622] 视频读取指针存储部分 241 中所存储的视频读取指针表示从缓冲器 215A 所读取的视频流的位置。另外，音频读取指针存储部分 251 中所存储的音频读取指针表示从缓冲器 215A 所读取的音频流的位置，以及字幕读取指针存储部分 262 中所存储的字幕读取指针表示从缓冲器 215A 所读取的字幕流的位置。

[0623] 如图 3 中所描述的，在缓冲器 215A 中，数据开始指针、数据写入指针、视频读取指针、音频读取指针、和字幕读取指针的任何一个都顺时针方向移动。

[0624] 此外，在这个实施例中，不断地更新数据开始指针，以便表示由表示视频读取指针、音频读取指针、和字幕读取指针之中的最旧的数据的指针所表示的相同位置。这里，在图 35 中，在视频读取指针、音频读取指针、和字幕读取指针之中，音频读取指针表示最旧的

数据，且数据开始指针与音频读取指针一致。

[0625] 在具有上述数据开始指针、视频读取指针、音频读取指针、和字幕读取指针的缓冲器控制模块 215 中，当从盘 101 读取之后在缓冲器 215A 中写入新的数据时，顺时针方向地更新数据写入指针，以便表示紧接所写入的新的数据之后的位置。

[0626] 此外，当从缓冲器 215A 读取了视频流、音频流、或字幕流时，以根据读取的数量的一个值顺时针方向地更新视频读取指针、音频读取指针、或字幕读取指针。这里，根据读取的数量的值包括在相应于实际所读取的视频、音频、和字幕数据的一部分与所读取的数据之间，且为包括在读取中所跳过的另一个流的数据部分的一个值。

[0627] 另外，当更新了与视频读取指针、音频读取指针、或字幕读取指针时，更新数据开始指针，以便表示由表示视频读取指针、音频读取指针、和字幕读取指针之中的最旧的数据的指针所表示的位置相同的位置。

[0628] 关于对缓冲器 215A 的数据的写入，缓冲器控制模块 215 控制缓冲器 215A 中数据的写入，以便防止数据写入指针超过数据开始指针。

[0629] 换句话说，只要不出现数据写入指针超过数据开始指针，在缓冲器控制模块 215 中，在由数据写入指针所表示的缓冲器 215A 中的一个位置写入从盘 101 所读取的数据，以便更新数据写入指针。另外，当将要出现数据写入指针超过数据开始指针时，在缓冲器控制模块 215 中，请求内容数据供给模块 213 停止（中断）从盘 101 的数据的读取，并停止缓冲器 215A 中的数据的写入。这能够防止缓冲器 215A 溢出。

[0630] 如上文所描述的，仅通过数据开始指针和数据写入指针之间的位置关系控制从盘 101 所读取的数据的写入缓冲器 215A。

[0631] 另外，关于从缓冲器 215A 的数据读取，缓冲器控制模块 215 控制从缓冲器 215A 的数据读取，以便防止视频读取指针、音频读取指针、和字幕读取指针、以及数据开始指针超过数据写入指针。

[0632] 换句话说，只要不出现视频读取指针、音频读取指针、或字幕读取指针超过数据写入指针，在缓冲器控制模块 215 中，响应来自视频解码器控制模块 216、音频解码器控制模块 217、或字幕解码器控制模块 218 的请求，从由视频读取指针、音频读取指针、或字幕读取指针所表示的缓冲器 215A 中的一个位置读取数据，并更新视频读取指针、音频读取指针、或字幕读取指针。另外，如果需要，更新数据开始指针。另外，当将要出现视频读取指针、音频读取指针、或字幕读取指针超过数据写入指针时，在缓冲器控制模块 215 中，例如，冻结来自视频解码器控制模块 216、音频解码器控制模块 217、或字幕解码器控制模块 218 的请求，并保持等待准备足够的数据。这能够防止缓冲器 215A 溢出。

[0633] 相应地，在缓冲器 215A 中，在从由数据开始指针所表示的位置至由数据写入指针所表示的位置的范围（图 34 和 35 中的阴影部分）中，存储将要供给视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 的数据。此外，在该范围中，视频读取指针、音频读取指针、和字幕读取指针退出。

[0634] 在上述情形中，更新数据开始指针，以便表示由视频读取指针、音频读取指针、和字幕读取指针所表示位置之中最旧的数据的位置。除此之外，能够执行数据开始指针的更新，以便表示从最旧的数据起过去一段预定的时间（例如，一秒）的数据的位置。

[0635] 换句话说，通常，估计认为在许多情形，在视频读取指针、音频读取指针、和字幕读

取指针之中，视频读取指针和音频读取指针表示最旧的数据的位置。

[0636] 因此，如图 34 中所示，在更新数据开始指针以便表示关于从由视频读取指针或音频读取指针所表示的最旧的数据的位置起的，例如，过去一秒的数据的位置的情形中，能够将从由视频读取指针或音频读取指针所表示的最旧的数据的位置起的过去一秒内的数据遗留在缓冲器 215A 中。这里，在图 34 中，音频读取指针表示最旧的数据的位置，且数据开始指针表示关于从该最旧的数据位置起的过去一秒的数据的位置。

[0637] 如上文所描述的，通过更新数据开始指针以便表示关于仅过去一秒的位置，能够提高盘装置的响应。

[0638] 换句话说，如图 35 中所示，在更新数据开始指针以便表示由音频读取指针所表示的最旧的数据的位置的情形中，例如，当请求至相反的方向的特殊重放时，从盘 101 重新读取从缓冲器 215A 的读取已经结束的数据是必需的。因而，在请求该特殊的重放之后需要一段时间，直到该特殊的重放成为可能。

[0639] 与此不同，如图 34 中所示，在更新数据开始指针以便表示从由音频读取指针所表示的最旧的数据的位置起的仅过去一秒的数据的位置的情形中，如果，在相反方向的特殊重放的请求下，用于启动该特殊重放所需要的数据为缓冲器 215A 中所存储的关于过去一秒的数据，则能够立即启动该特殊重放，而不必如上文所描述的从盘 101 重新读取数据。

[0640] 即使在更新数据开始指针以便表示从由音频读取指针所表示的最旧的数据的位置起的过去一秒的数据的位置的情形中，用于启动该特殊重放所需要的数据不存储在缓冲器 215A 中是很可能的。在这种情形，从盘 101 重新读取用于启动该特殊重放所需要的数据。

[0641] 接下来，描述从缓冲器 215A 的视频流、音频流、和字幕流的每一个的读取的详细情况。

[0642] 如图 30 中的步骤 S127 中所描述，当启动片断流文件的重放时，在缓冲器控制模块 215 中，数据开始指针、数据写入指针、视频读取指针、音频读取指针、和字幕读取指针都初始化，以便表示缓冲器 215A 中相同的位置。

[0643] 当从盘 101 读取片断流文件中所存储的节目流 (MPEG2- 系统节目流)，并供给缓冲器控制模块 215 时，在缓冲器控制模块 215 中，在由数据写入指针所表示的缓冲器 215A 中的一个位置存储该节目流，且顺时针方向地更新该数据写入指针。

[0644] 此外，在缓冲器控制模块 215(图 3) 中，视频读取功能单元 233 执行缓冲器 215A 中所存储的节目流的语法分析。响应来自视频解码控制模块 216 的请求，它从缓冲器 215A 中所存储的节目流中提取 (分离) 并读取视频流 (视频存取单元)，并将该视频流供给视频解码控制模块 216。

[0645] 类似地，同样音频读取功能单元 234 执行缓冲器 215A 中所存储的节目流的语法分析。响应来自音频解码控制模块 217 的请求，它从缓冲器 215A 中所存储的节目流中提取并读取音频流 (音频存取单元)，并将该音频流供给音频解码控制模块 217。同样字幕读取功能单元 235 执行缓冲器 215A 中所存储的节目流的语法分析。响应来自字幕解码控制模块 218 的请求，它从缓冲器 215A 中所存储的节目流中提取并读取字幕流 (字幕存取单元)，并将该字幕流供给字幕解码控制模块 218。

[0646] “视频流的读取”

[0647] 接下来，参考图 36 中的流程图描述具有视频读取功能单元 233(图 3) 从缓冲器

215A 中读取视频流的处理的详细情况。

[0648] 在步骤 S211，首先，视频读取功能单元 233 搜索并找到缓冲器 215A 中所存储的节目流中的 private_stream_2 的 PES_packet()。换句话说，如图 20 中所描述的，private_stream_2 中的 PES_packet() 的 stream_id 为 10111111B(= 0xBF)。相应地，视频读取功能单元 233 搜索并找到具有 10111111B 的 stream_id 的 PES_packet()。

[0649] 这里，例如，假设如上所描述的将要重放在片断流文件“00001.PS”中所存储的节目流中多路复用的基本流，当从盘 101 中读取了节目流并存储在缓冲器 215A 中时，基于在图 30 中的步骤 S122 中的片断流文件“00001.PS”中的 EP_map()(图 27) 中所描述的解码可开始点的信息，确定扇区 305 为重放开始位置，且另外，在图 30 中的步骤 S128，指定重放开始位置的扇区 305，并请求操作系统 201 读取片断流文件“00001.PS”中所存储的节目流。

[0650] 另外，关于视频流，EP_map() 中所描述的解码可开始点的信息表示 private_stream_2 中的 PES_packet() 的位置。

[0651] 相应地，紧接片断流文件“00001.PS”中所存储的节目流在从盘 101 中读取之后存储在缓冲器 215A 中之后，在由数据开始指针或视频读取指针所表示的缓冲器 215A 中的一个位置存储 private_stream_2 中的 PES_packet()。

[0652] 当在步骤 S211 找到 private_stream_2 中的 PES_packet() 时前进到步骤 S212，视频读取功能单元 233 提取在作为 private_stream_2 中的 PES_packet() 的 PES_packet_data_byte 的 private_stream2_PES_payload()(图 23) 中所描述的 video_stream_id，并确定该 video_stream_id 是否与在图 30 中的步骤 S127 中的 stream_id 寄存器 242(图 3) 中所存储的将要重放的视频流的 stream_id 一致。

[0653] 如果在步骤 S212 确定 private_stream2_PES_payload() 中所描述的 video_stream_id 与 stream_id 寄存器 242 中所存储的 stream_id 不一致，即，如果在上个步骤 S211 中所找到的 private_stream_2 中的 PES_packet() 不是位于将要重放的视频流中的解码开始点的一个，则处理返回到步骤 S211，并搜索缓冲器 215A 中所存储的节目流中的另一个 private_stream_2 中的 PES_packet()。随后，重复地执行类似的处理。

[0654] 可选择地，如果在步骤 S212 确定 private_stream2_PES_payload() 中所描述的 video_stream_id 与 stream_id 寄存器 242 中所存储的 stream_id 一致，即，如果在上个步骤 S211 中所找到的 private_stream_2 中的 PES_packet() 是位于将要重放的视频流中的解码开始位置的一个，则视频读取功能单元 233 前进到步骤 S213，从缓冲器 215A 中读取 private_stream_2 中的 PES_packet() 的 private_stream2_PES_payload() 中所描述的 au_information()，并在前进到步骤 S214 之前将该 au_information() 存储在缓冲器 215A 中。

[0655] 在步骤 S214，视频读取功能单元 233 通过在上个步骤 S211 中所找到的 private_Stream_2 中的 PES_packet()(private_stream_2 中的 PES_packet()，在 private_stream_2 中 video_stream_id(图 23) 与 stream_id 寄存器 242(图 3) 中所存储的 stream_id 一致) 的大小，更新视频读取指针存储部分 231 中所存储的视频读取指针。

[0656] 换句话说，在片断流文件中，紧接着 private_stream_2 中的 PES_packet()，设置了具有与其 video_stream_id 一致的 stream_id 的视频流 (PES_packet())。因而，在步骤 S214，更新视频读取指针，以便表示视频流中真实的解码可开始点的位置。

[0657] 在那之后，从步骤 S214 前进到 S215，视频读取功能单元 233 确定视频解码器控制模块 216 是否请求数据。如果确定还未请求数据，则处理返回到步骤 S215，并重复地执行类似的处理。

[0658] 可选择地，如果在步骤 S215 确定视频解码器控制模块 216 已请求数据，则处理前进到步骤 S216。在执行来自由视频读取指针所表示的位置的节目流的语法分析的时候，视频读取功能单元 233 从缓冲器 215A 中读取具有 an_information() 寄存器 243 中所存储的 au_information() 的 AU_length 中所描述的字节数的数据（即，一个视频存取单元），将该数据供给视频解码器控制模块 216，并以从缓冲器 215A 中所读取的一个视频存取单元的大小，更新视频读取指针。

[0659] 换句话说，如图 24 中所描述的，在 au_information() 中，描述了表示从包括该 au_information() 的 private_stream_2 中的 PES_packet() 至下一个 private_stream_2 中的 PES_packet() 中所包括的视频存取单元（画面）的数目的 number_of_access_units。

[0660] 另外，如图 24 中所描述的，在 au_information() 中，描述了作为关于与 number_of_access_units 中的数目一样多的视频存取单元的每一个的信息的 pic_struct_copy、au_ref_flag、和 AU_length。

[0661] 如图 24 中所描述的，在与 number_of_access_units 中的数目一样多的 au_information() 项中所描述的 AU_length 项的每一个都表示从包括该 au_information() 的 private_stream_2 中的 PES_packet() 至下一个 private_stream_2 中的 PES_packet() 中所包括的与 number_of_access_units 中的数目一样多的视频存取单元的每一个的大小。因而，通过使用 AU_length，视频读取功能单元 233 能够在未执行视频流语法分析的情况下执行存取单元片断。

[0662] 换句话说，照常规，在片断 MPEG2-Video 或 MPEG4-AVC 中的存取单元的情形中，在已知视频流的语法之后，需要执行视频流的语法分析。然而，因为紧邻视频存取单元的视频流中的一个或多个解码可开始点的每一个之前，盘 101 上所记录的片断流文件中所存储的节目流包括在其中描述表示该视频存取单元的大小的 AU_length 的 private_stream_2 中的 PES_packet()，基于 private_stream_2 中的 PES_packet() 中所描述的 AU_length，视频读取功能单元 233 在未执行视频流语法分析的情况下，执行从缓冲器 215A 中读取视频存取单元（单元视频流）以及将该单元供给视频解码器控制模块 216。

[0663] 当在步骤 S 216 视频读取功能单元 233 将视频存取单元供给视频解码器控制模块 216 时，视频读取功能单元 233 也将 au_information() 中所描述的 pic_struct_copy、au_ref_flag、和 AU_length，以及添加到该视频存取单元上的时间戳 (PTS、DTS) 作为关于该视频存取单元的信息而供给视频解码器控制模块 216。

[0664] 在步骤 S216 之后，即在从缓冲器 215A 读取之后将一个视频存取单元供给视频解码器控制模块 216 之后，处理前进到步骤 S217，且视频读取功能单元 233 确定是否具有与由 an_information() 寄存器 243 中所存储的 au_information() 的 number_of_access_units 所表示的数目一样多的处理过的存取单元。

[0665] 如果在步骤 S217 确定还未处理与由 number_of_access_units 所表示的数目一样多的存取单元，即，在从缓冲器 215A 中读取之后，还未将与由 number_of_access_units 所表示的数目一样多的存取单元供给视频解码器控制模块 216，则处理返回到步骤 S215，且

随后重复类似的处理。

[0666] 可选择地，如果步骤 S217 确定已经处理了与由 number_of_access_units 所表示的数目一样多的存取单元，即，在从缓冲器 215A 中读取之后，已将与由 number_of_access_units 所表示的数目一样多的存取单元供给视频解码器控制模块 216，则处理返回到步骤 S211，并搜索下一个 private_stream_2 中的 PES_packet()。随后，重复类似的处理。

[0667] “音频流的读取”

[0668] 接下来，参考图 37 中的流程图描述从缓冲器 215A 中读取音频流的音频读取功能单元 234（图 3）的过程的详细情况。

[0669] 首先，在步骤 S230，音频读取功能单元 234 确定 stream_id 寄存器 252 中所存储的将要重放的音频流的 stream_id 是否表示 private_stream_1 中的 PES_packet()。

[0670] 如果在步骤 230 确定 stream_id 寄存器 252 中所存储的 stream_id 不表示 private_stream_1 中的 PES_packet()，即，如果，如图 20 中所描述的，stream_id 寄存器 252 中所存储的 stream_id 被分配给根据 MPEG 标准而编码的音频流的 110xxxxxB，则音频读取功能单元 234 前进到步骤 S231，并在缓冲器 215A 中所存储的节目流中搜索表示音频帧起点并在 MPEG 音频中所定义的同步代码。因为同步代码的位置为音频帧起点，音频读取功能单元 234 更新音频读取指针，以便表示音频帧起点的位置，并从步骤 S231 前进到 S232。在步骤 S232，音频读取功能单元 234 从由音频读取指针所表示的位置搜索并找到缓冲器 215A 中所存储的节目流中的与 stream_id 寄存器 252 中所存储的 stream_id 一致的 PES_packet()，并前进到步骤 S233。

[0671] 在步骤 S233，音频读取功能单元 234 更新音频读取指针存储部分 251 中所存储的音频读取指针，以便表示在上一个步骤 S232 中所找到的 PES_packet() 的 PES_packet_data_byte（图 16 至 18）的起点，并前进到步骤 S237。

[0672] 在步骤 S237，音频读取功能单元 234 确定音频解码器控制模块 217 是否已经请求数据。如果确定为否定的，则处理返回到步骤 S237，并重复类似的处理。

[0673] 可选择地，如果在步骤 S237 确定音频解码器控制模块 217 已经请求了数据，则处理前进到步骤 S238。当执行来自由音频读取指针所表示的位置的节目流的语法分析时，音频读取功能单元 234 从缓冲器 215A 中读取具有已知固定长度的一个音频存取单元，并将该音频存取单元和添加到该音频存取单元上的时间戳（PTS、DTS）一起供给音频解码器控制模块 217。

[0674] 音频读取功能单元 234 通过从缓冲器 215A 中所读取的一个音频存取单元的大小来更新音频读取指针，并返回到步骤 S237。随后，重复类似的处理。

[0675] 可选择地，如果在步骤 S230 确定 stream_id 寄存器 252 中所存储的 stream_id 表示 private_stream_1 中的 PES_packet()，即，如果，如图 20 中所描述的，stream_id 寄存器 252 中所存储的 stream_id 为 10111101B（＝ 0xBD），并表示 private_stream_1 中的 PES_packet()，则步骤前进到步骤 S234。音频读取功能单元 234 搜索并找到缓冲器 215A 中所存储的节目流中的 private_stream_1 中的 PES_packet()。换句话说，音频读取功能单元 234 搜索并找到具有为 10111101B 的 stream_id 的 PES_packet()。

[0676] 在步骤 S234 找到 private_stream_1 中的 PES_packet() 之后，音频读取功能单元 234 前进到步骤 S235。音频读取功能单元 234 提取作为 private_stream_1 中的 PES_

packet() 的 PES_packet_data_byte 的 private_stream1_PES_payload()(图 21) 中所描述的 private_stream_id，并确定该 private_stream_id 是否与 private_stream_id 寄存器 253(图 3) 中所存储的将要重放的音频流的 private_stream_id 一致。

[0677] 如果在步骤 S235 确定 private_stream1_PES_payload() 中所描述的 private_stream_id 与 private_stream_id 寄存器 253 中所存储的 private_stream_id 不一致，即，如果在上一个步骤 S234 中所找到的 private_stream_1 中的 PES_packet() 不是将要重放的音频流，则处理返回到步骤 S234，并搜索缓冲器 215A 中所存储的节目流中的另一个 private_stream_1 中的 PES_packet()。随后，重复类似的处理。

[0678] 可选择地，如果在步骤 S235 确定 private_stream1_PES_payload() 中所描述的 private_stream_id 与 private_stream_id 寄存器 253 中所存储的 private_stream_id 一致，即，如果在上一个步骤 S234 中所找到的 private_stream_1 中的 PES_packet() 为将要重放的音频流，则处理前进到步骤 S236，且音频读取功能单元 234 执行从缓冲器 215A 中读取 private_stream_1 中的 private_stream1_PES_payload()(图 21) 中所描述的 AU_locator，并通过相加紧接着该 AU_locator 的位置和由该 AU_locator 所表示的值而找到音频存取单元的开始位置。

[0679] 换句话说，如图 21 中所描述的，通过使用紧接着 AU_locator 的位置作为参考，该 AU_locator 表示 private_stream1_PES_payload() 中所存储的音频存取单元(或字幕存取单元)的开始位置。因而，通过将由 AU_locator 所表示的值加到紧接着该 AU_locator 的位置上，能够找到音频存取单元的(绝对)开始位置。

[0680] 此外，在步骤 S236，音频读取功能单元 234 更新音频读取指针存储部分 251 中所存储的音频读取指针，以便表示如上文所描述而找到的音频存取单元的开始位置，并前进到步骤 S237。

[0681] 在步骤 S 237，音频读取功能单元 234 确定音频解码器控制模块 217 是否已经请求数据。如果确定为否定的，则处理返回到步骤 S237，并重复类似的处理。

[0682] 可选择地，如果在步骤 S237 确定音频解码器控制模块 217 已经请求了数据，则处理前进到步骤 S238。当执行来自由音频读取指针所表示的缓冲器 215A 中的位置的节目流的语法分析时，音频读取功能单元 234 从缓冲器 215A 中读取具有已知固定长度的一个音频存取单元，并将该单元和加到该音频存取单元上的时间戳一起供给音频解码器控制模块 217。

[0683] 音频读取功能单元 234 通过从缓冲器 215A 中所读取的一个音频存取单元的大小来更新音频读取指针。处理返回到到步骤 S237，且随后重复类似的处理。

[0684] “字幕流的读取”

[0685] 接下来，参考图 38 中的流程图描述从缓冲器 215A 中读取字幕流的字幕读取功能单元 235(图 3) 的处理的详细情况。

[0686] 首先，在步骤 S251，字幕读取功能单元 235 识别在图 3 中的步骤 S127 中在字幕读取功能部分 261 中所存储的字幕读取功能标志。如果在步骤 S251 确定静止图像文件为 0，即，例如，如果在其中将将要重放的基本流多路复用的片断流文件中不包括字幕流，且在图 30 中的步骤 S127 中在字幕读取功能部分 261 中设定为 0，则字幕读取功能单元 235 并不具体地执行处理。

[0687] 可选择地，如果在步骤 S251 确定字幕读取功能标志为 1，即，例如，如果在其中将将要重放的基本流多路复用的片断流文件中包括字幕流，且在图 30 中的步骤 S127 中在字幕读取功能部分 261 中设定为一，则字幕读取功能单元 235 前进到步骤 S252，并在缓冲器 215A 中所存储的节目流中搜索与 stream_id 寄存器 263（图 3）中所存储的将要重放的字幕流中的 stream_id 一致的 PES_paeket()。

[0688] 如图 30 中的步骤 S127 中所描述的，在 stream_id 寄存器 263（图 3）中存储将要重放的字幕流的 stream_id。如图 20 中所描述的，字幕流的 stream_id 为 10111101B（＝0xBD），表示 private_stream_1 中的 PES_packet()。

[0689] 因此，在步骤 S252，搜索缓冲器 215A 中所存储的节目流中的 private_stream_1 中的 PES_packet()。

[0690] 当在步骤 S252 在搜索 private_stream_1 中的 PES_packet() 之后找到 private_stream_1 中的 PES_packet() 时，字幕读取功能单元 235 前进到步骤 S253，提取作为 private_stream_1 中 的 PES_packet() 的 PES_packet_data_byte 的 private_stream1_PES_payload()（图 21）中所描述的 private_stream_id，并确定该 private_stream_id 是否与在图 30 中的步骤 S127 在 private_stream_id 寄存器 264（图 3）中所存储的将要重放的字幕流的 private_stream_id 一致。

[0691] 如果在步骤 S253 确定 private_stream1_PES_payload() 中所描述的 private_stream_id 与 private_stream_id 寄存器 264 中所存储的 private_stream_id 不一致，即，如果在上一个步骤 S252 中所找到的 private_stream_1 中的 PES_packet() 不是将要重放的字幕流，则处理返回到步骤 S252，并搜索缓冲器 215A 中所存储的节目流中的另一个 private_stream_1 中的 PES_packet()。随后，重复类似的处理。

[0692] 可选择地，如果在步骤 S253 确定 private_stream1_PES_payload() 中所描述的 private_stream_id 与 private_stream_id 寄存器 264 中所存储的 private_stream_id 一致，即，如果在上一个步骤 S252 中所找到的 private_stream_1 中的 PES_packet() 为将要重放的字幕流，则字幕读取功能单元 235 前进到步骤 S254，并执行从缓冲器 215A 中读取 private_stream_1 中的 PES_packet() 的 private_stream1_PES_payload()（图 21）中所描述的 AU_locator，并通过相加紧接着该 AU_locator 的位置和由该 AU_locator 所表示的值而找到字幕存取单元的开始位置。

[0693] 换句话说，如图 21 中所描述的，通过使用紧接着 AU_locator 的位置作为参考，该 AU_locator 表示 private_stream1_PES_payload() 中所存储的字幕存取单元（或音频存取单元）的开始位置。因而，通过将由 AU_locator 所表示的值加到紧接着该 AU_locator 的位置上，能够找到字幕存取单元的（绝对）开始位置。

[0694] 此外，在步骤 S254，字幕读取功能单元 235 更新字幕读取指针存储部分 262 中所存储的字幕读取指针，以便表示如上文所描述而找到的字幕存取单元的开始位置，并前进到步骤 S255。

[0695] 在步骤 S255，字幕读取功能单元 235 确定字幕解码器控制模块 218 是否已经请求数据。如果确定为否定的，则处理返回到步骤 S255，并重复类似的处理。

[0696] 可选择地，如果在步骤 S255 确定字幕解码器控制模块 218 已经请求了数据，则处理前进到步骤 S256。当执行来自由字幕读取指针所表示的缓冲器 215A 中的位置的节目流

的语法分析时，字幕读取功能单元 235 从缓冲器 215A 中以在字幕存取单元的开始处所描述的大小读取一个字幕存取单元，并将该单元和添加到该字幕存取单元上的时间戳一起供给字幕解码器控制模块 218。换句话说，如图 2 中所描述的，在字幕存取单元的开始，描述了该字幕存取单元的大小。相应地，字幕读取功能单元 235 从由字幕读取指针所表示的缓冲器 215A 中的位置起以该大小读取数据，并将该字幕存取单元作为所读取的数据和添加到该字幕存取单元上的时间戳一起供给字幕解码器控制模块 218。

[0697] 另外，字幕读取功能单元 235 通过从缓冲器 215A 中所读取的一个字幕存取单元的大小来更新字幕读取指针，并返回到到步骤 S255。随后，重复类似的处理。

[0698] “再同步处理”

[0699] 接下来，描述通过图 2 中的解码控制模块 214 的视频数据和音频数据之间的同步控制。

[0700] 如在图 30 中的步骤 S130 中所描述的，如果建立同步必需，解码控制模块 214 通过转换时间，命令视频解码器控制模块 216、音频解码器控制模块 217、和字幕解码器控制模块 218 启动解码。例如，依赖于视频解码器 116 和音频解码器 117 随后的解码进行，视频数据和作为与视频数据同步输出的输出数据的音频数据的输出的数据可以彼此移动。

[0701] 相应地，解码控制模块 214 执行再同步处理，以纠正视频数据的输出和作为将与视频数据同步而输出的输出数据的音频数据的输出之间出现的移位，并彼此同步地输出视频数据和音频数据。

[0702] 参考图 39 中的流程图描述再同步处理。

[0703] 在再同步处理中，首先，在步骤 S271，解码控制模块 214 确定是否在来自视频解码器控制模块 216 的视频存取单元的时间戳和来自音频解码器控制模块 217 的音频存取单元的时间戳之间找到一个大的移位。

[0704] 换句话说，如在图 30 中的步骤 S129 中所描述的，无论何时视频解码器控制模块 216 从缓冲器控制模块 215 获得视频存取单元，其将该视频存取单元的时间戳供给解码控制模块 214。类似地，无论何时音频解码器控制模块 217 从缓冲器控制模块 215 获得音频存取单元，其将该音频存取单元的时间戳供给解码控制模块 214。

[0705] 在步骤 S271，通过比较从视频解码器控制模块 216 和音频解码器控制模块 217 以相同的定时（在能够被看作相同的定时的一段确定时间内）所供给的时间戳，解码控制模块 214 确定在这些时间戳之间是否有大的移位。

[0706] 如果在步骤 S271 确定在来自视频解码器控制模块 216 的视频存取单元的时间戳和来自音频解码器控制模块 217 的音频存取单元的时间戳之间无大的移位，即，如果视频存取单元的时间戳和音频存取单元的时间戳之间的移位为，例如，两个视频帧（大约 66 毫秒）或在能够被看作具有预定的同步的范围之内，则处理返回到步骤 S271，并连续不断地执行两个时间戳的确定（监控）。

[0707] 可选择地，如果在步骤 S271 确定在来自视频解码器控制模块 216 的视频存取单元的时间戳和来自音频解码器控制模块 217 的音频存取单元的时间戳之间有大的移位，即，如果视频存取单元的时间戳和音频存取单元的时间戳之间的移位在能够被看作具有预定的同步的范围之外，则处理前进到步骤 S272。解码控制模块 214 通过比较来自视频解码器控制模块 216 的视频存取单元的时间戳和来自音频解码器控制模块 217 的音频存取单元的

时间戳，确定在视频数据的输出和音频数据的输出之间哪个落后。

[0708] 如果在步骤 S272 确定视频数据的输出落后于音频数据的输出，则处理前进到步骤 S273。为了将对视频存取单元的处理提前一个视频存取单元，解码控制模块 214 命令视频解码器控制模块 216 不执行解码和视频存取单元的输出（显示），即，解码控制模块 214 输出指令，以跳过对视频存取单元的处理，并前进到步骤 S274。

[0709] 在步骤S274，视频解码器控制模块216从解码控制模块214接收跳过指令，并响应该跳过指令，检查从缓冲器控制模块215与视频存取单元一起供给的au_ref_flag（图24）。

[0710] 换句话说，在位于 private_stream_2 中的 PES_packet() 的 private_stream2_PES_payload() 中的 au_information()（图 24）中，包括作为关于该存取单元的信息的 au_ref_flag。如在图 30 中的步骤 S129 和在图 36 中的步骤 S216 中所描述的，缓冲器控制模块 215 将视频存取单元的 au_ref_flag 与该视频存取单元一起供给视频解码器控制模块 216。

[0711] 在步骤 S274，检查以上述方式所供给的存取单元的 au_ref_flag。

[0712] 从步骤S274前进到步骤S275，基于检查从缓冲器控制模块215所供给的视频存取单元的au_ref_flag的结果，视频解码器控制模块216确定当将另一个画面解码时，该视频存取单元是否为不被参考的非参考图像。

[0713] 这里，如图 24 中所描述的，视频存取单元的 au_ref_flag 指示该存取单元是否为参考图像。如果该存取单元为参考图像，其设定为 1。如果该存取单元不是参考图像，其设定为 0。

[0714] 如果在步骤 S275 确定从缓冲器控制模块 215 所供给的视频存取单元不是非参考图像（的视频存取单元），即，如果从缓冲器控制模块 215 所供给的视频存取单元为参考图像，则处理前进到步骤 S276。视频解码器控制模块 216 控制视频解码器 116 正常地处理视频存取单元，并在返回到步骤 S274 之前等待将要从缓冲器控制模块 215 供给的下一个视频存取单元。

[0715] 可选择地，如果在步骤 S275 确定从缓冲器控制模块 215 所供给的视频存取单元是非参考图像，则处理前进到步骤 S277。视频解码器控制模块 216 跳过视频解码器 116 对视频存取单元的处理，并在返回到步骤 S271 之前等待将要从缓冲器控制模块 215 供给的下一个视频存取单元。

[0716] 以这样的方式，跳过对视频存取单元的处理，由此对视频存取单元的处理提前（处理时间缩短）大约一个视频存取单元。这导致了将落后于音频数据的输出的视频数据的输出加速。

[0717] 可选择地，如果在步骤 S272 确定视频数据的输出不落后于音频数据的输出，即，如果音频数据的输出落后于视频数据的输出，则处理前进到步骤 S278。为了使对视频存取单元的处理保持等待，解码控制模块 214 将重复地输出相应于目前正被解码的视频存取单元的视频数据的重复输出指令输出至视频解码器控制模块 216，并前进到步骤 S279。

[0718] 在步骤S279，视频解码器控制模块216从解码控制模块214接收重复输出指令，响应该重复输出指令，将相应于正由视频解码器 116 解码的视频存取单元的视频数据重复地输出至图形处理模块 219，并在返回到步骤 S271 之前等待将要从缓冲器控制模块 215 供给的下一个视频存取单元。

[0719] 如上文所描述的，解码控制模块 214 确定视频数据的输出是否落后于音频数据的

输出。如果视频数据的输出落后于音频数据的输出，则解码控制模块 214 命令视频解码器控制模块 216 跳过对存取单元的处理。另外，基于命令跳过的存取单元的 au_ref_flag，视频解码器控制模块 216 确定该存取单元是参考图像和非参考图像中的哪一个。当该存取单元是非参考图像时，视频解码器控制模块 216 控制视频解码器 116 跳过处理命令跳过的存取单元。因此，能够很容易地建立视频数据的输出和音频数据的输出之间的同步。

[0720] 换句话说，当将要跳过被处理的存取单元是参考图像时，要求将相应于该存取单元的视频数据解码，以便当将随后要解码的另一个存取单元解码时将其作参考。因此，在用于建立视频数据的输出和音频数据的输出之间的同步的同步控制中，对参考图像的存取单元的处理的跳过使得不可能将参考该参考图像的另一个存取单元解码，所以，在视频数据的显示中，同步控制呈现为噪声。

[0721] 相应地，关于处理的跳过，使用不作为参考图像的存取单元，即，非参考图像的存取单元，是更可取的。

[0722] 另外，关于相关技术的基本流，为了搜索非参考图像的存取单元，要求执行对该基本流的语法分析。通过根据 MPEG4-AVC 或诸如此类而编码所获得的基本流具有非常复杂的语法。因而，关于语法分析需要相当大的成本。

[0723] 与此不同，在盘 101 上所记录的片断流文件中所存储的节目流中，与在 PES_packet_data_byte 中设置视频存取单元的 PES_packet()（图 16 至 18）分离地将设置通过扩展 PES_packet_data_byte 而获得的 private_stream2_PES_payload()（图 23）的 private_stream_2 中的 PES_packet() 项多路复用，且在 private_stream2_PES_payload() 的 au_information()（图 24）中，关于每个视频存取单元，描述了指示该视频存取单元作为参考图像和非参考图像中的哪一个的 au_ref_flag。另外，从缓冲器控制模块 215 将 au_ref_flag 与相应的视频存取单元一起供给视频解码器控制模块 216。因此，通过检查与视频存取单元一起而供给的视频存取单元的 au_ref_flag，视频解码器控制模块 216 能够辨识该视频存取单元为参考图像还是非参考图像，而几乎不需要任何成本。

[0724] “标记处理”

[0725] 接下来，参考图 4 中的流程图描述基于 PlayListMark() 中所描述的 Mark()（图 7）而执行的标记处理。

[0726] 解码控制模块 214 不断地确认由内嵌定时器单元 214A 所度量的当前时间。在步骤 S301，确定当前时间是否与 PlayListMark() 中所描述的任何 Mark()（图 7）的 mark_time_stamp 一致。

[0727] 换句话说，如图 30 中的步骤 S124 中所描述的，当重放图 25 中所示的第一 PlayList#0 中的第一 PlayItem#0 时，播放器控制模块 212 辨识在图 28 的上区所示的 PlayListMark() 中所包括的七个 Mark() 项之中，从第一至第四的四个 Mark() 项属于第一 PlayItem#0，且播放器控制模块 212 将作为该四个 Mark() 项的 mark_time_stamp 的 {180,090}、{5,580,090}、{10,980,090}，和 {16,380,090} 与由该 mark_time_stamp 所表示的属性为“MarkProcess”的事实一起传递给解码控制模块 214。

[0728] 在步骤 S 301，解码控制模块 214 确定当前时间是否与从播放器控制模块 212 中如上文所描述而供给的“Mark Process”属性的时间 (mark_time_stamp) 的任何一个一致。

[0729] 如果在步骤 S301 确定当前时间不与“Mark Process”属性的时间的任何一个一

致，则处理返回到步骤 S301，并重复类似的处理。

[0730] 可选择地，如果在步骤 S301 确定当前时间与“Mark Process”属性的时间的任何一个一致，则解码控制模块 214 将指示当前时间为“Mark Process”属性的时间的一个并为与当前时间一致的“Mark Process”属性的时间的消息供给播放器控制模块 212，并前进到步骤 S302。

[0731] 在步骤S302，播放器控制模块212从解码控制模块214接收指示当前时间为“Mark Process”属性的时间的一个和作为与当前时间一致的“MarkProcess”属性的时间的消息，并辨识 mark_time_stamp 与当前时间一致的 Mark() 为经历标记处理的 Mark()（如果需要，以下称为处理目标标记）。

[0732] 换句话说，播放器控制模块 212 辨识目前正重放的 PlayList() 中的 PlayItem()。相应地，基于PlayList() 的PlayItem() 和与当前时间一致的“Mark Process”属性的时间，播放器控制模块 212 通过参考“PLAYLIST.DAT”（图 5）中的 PlayListMark()（图 5），辨识处理目标标记。

[0733] 特殊地，例如，如果正重放图 25 中所示的第一 PlayList#0 中的第一 PlayItem#0，则该事实促使播放器控制模块 212 辨识标记时间为在图 28 中的上区中所示的 PlayListMark() 中所包括的七个Mark() 项之中从第一至第四的四个Mark() 项中的一个的 mark_time_stamp。

[0734] 另外，假设从解码控制模块 214 供给播放器控制模块 212 的标记时间为，例如，16,380,090，播放器控制模块 212 辨识在图 28 中的上区中所示的 PlayListMark() 中所包括的从第一至第四的四个 Mark() 项之中，与为 16,380,090 的标记时间一致的第四个 Mark() 为处理目标标记。

[0735] 如所描述的，在辨识处理目标标记之后从步骤 S302 前进到步骤 S303，播放器控制模块 212 确定在处理目标标记中，是否描述了用于指定基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id（图 7）。

[0736] 如果在步骤 S303 确定在处理目标标记中，未描述用于指定基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id（图 7），即，如果 entry_ES_stream_id 和 entry_ES_private_stream_id 两者都为 0x00，则过程跳过步骤 S304 并前进到步骤 S305，且随后执行与处理目标标记相匹配的处理。

[0737] 可选择地，如果在步骤 S303 确定在处理目标标记中，描述了用于指定基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id（图 7），则播放器控制模块 212 前进到步骤 S304，并确定在正重放的基本流中是否包括由 entry_ES_stream_id 以及 entry_ES_private_stream_id（如果需要）所指定的基本流。

[0738] 如果在步骤 S304 确定在正重放的基本流中，不包括由处理目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 所指定的基本流，则处理前进到步骤 S301。换句话说，当未重放由处理目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 所指定的基本流时，忽略该处理目标标记。

[0739] 可选择地，如果在步骤 S304 确定在正重放的基本流中，包括由处理目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 所指定的基本流，即，如果重放了由处理目标标记的 entry_ES_stream_id 和 entry_ES_private_stream_id 所指定的基本流，

则处理目标标记被看作为有效的，且处理前进到步骤S305。随后，执行与该处理目标标记相匹配的处理。

[0740] 换句话说，在步骤S305，播放器控制模块212通过参考处理目标标记的mark_type（图7），识别该处理目标标记。

[0741] 如果在步骤S305确定处理目标标记为章标记或索引标记，即，当处理目标标记的mark_type为“Chapter”或“Index”时，播放器控制模块212前进到步骤S306并命令图形处理模块219将章或索引的显示号码更新成作为由为处理目标标记的章标记或索引标记所表示的章或索引号。处理返回到步骤S301。

[0742] 如果在步骤S305确定处理目标标记为事件标记，即，当处理目标标记的mark_type为“Event”时，播放器控制模块212前进到步骤S307并将表示事件的发生的事件消息和该处理目标标记的mark_data报告（供给）给脚本控制模块211。处理前进到步骤S308。

[0743] 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，通过将该事件消息用作中断请求，执行当将mark_data用作自变量时在“SCRIPT.DAT”文件中预先描述的连续处理，并返回到步骤S301。

[0744] 换句话说，在脚本控制模块211中，执行与mark_data相匹配的处理。

[0745] 特殊地，例如，在图28中的下区中所示的PlayList#1中的PlayListMark()中，第二Mark()(Mark#1)和第三Mark()(Mark#2)每个都具有为“Event”的mark_type。然而，它们由于它们的mark_data项为1(Mark#1)和2(Mark#2)而不同。

[0746] 在接收相应于第二Mark()的事件消息的情形和接收相应于第三Mark()的事件消息的情形的每一个中，脚本控制模块211都通过使用相同的事件处理器（中断处理程序）执行响应事件消息的处理。在事件处理器中，通过检查与事件消息一起而供给的mark_data，脚本控制模块211关于每个mark_data，执行不同的处理。

[0747] 特殊地，例如，当mark_data为1时，脚本控制模块211控制图形处理模块219显示第一类型的图标。另外，例如，当mark_data为2时，脚本控制模块211控制图形处理模块219显示第二类型的图标。

[0748] mark_data并不局限于1和2。另外，如上文所描述的，相应于mark_data的处理并不局限于简单的图标显示。

[0749] 换句话说，例如，当mark_data为在3至18范围内的值时，脚本控制模块211控制图形处理模块219以相应于通过从mark_data减去2而获得的值（1至16的数字值）的亮度显示第一类型的图标。另外，例如，当mark_data为在19至34范围内的值时，脚本控制模块211控制图形处理模块219以相应于通过从mark_data减去18而获得的值（1至16的数字值）的亮度显示第二类型的图标。

[0750] 而且，例如，在其中由用户所操作的控制器连接到输入接口115（图1），且该控制器具有在其中在DC（直流）电动机轴上安装了偏心重量且当开动该DC电动机时产生振动的内嵌振动电动机的情形中，当mark_data为在35至42范围内的值时，能够在根据通过从该mark_data减去34而获得的值（1至8的数字值）的操作时间期间开动该振动电动机。

[0751] mark_data为数字值，且能够通过由脚本控制模块211所执行的脚本程序而描述其用法和算法。因此，依预先确定的规则而使用mark_data，且另外能够依由盘101的制造商，供给盘101上所记录的数据的内容供给者或诸如此类独立地设定的规则而使用它。

[0752] 如上文所描述的，在标记处理中，当当前时间与“Mark Process”属性的时间一致时，从作为“Mark Process”属性的时间的标记时间辨识处理目标标记。此外，当在处理目标标记中未描述用于指定基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id 时，执行与处理标记的 mark_type 相匹配的处理。另外，即使在处理目标标记中描述了用于指定基本流的 entry_ES_stream_id 和 entry_ES_private_stream_id，如果正在重放由 entry_ES_stream_id 和 entry_ES_private_stream_id 所指定的基本流，则执行与该 mark_type 相匹配的处理。

[0753] 因此，例如，假设正在重放图 25 中所示的第二 PlayList#1，执行下列标记处理。

[0754] 换句话说，如在图 28 中的下区中所示，在第二 PlayList#1 中的 PlayListMark() 中，描述了分别指定其 mark_time_stamp 项为 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)。

[0755] 此外，因为在图 28 中的下区中的 PlayListMark() 中，在第二 Mark() 和第三 Mark() 的 entry_ES_stream_id 项中分别描述为 0xE0 和 0xE1，分别由 0xE0 和 0xE1 所指定的基本流分别与第二 Mark() 和第三 Mark() 相关联。

[0756] 这里，如图 25 中所描述的，在第二 PlayList#1 中，仅描述了一个 PlayItem() (PlayItem#0)，且根据 PlayItem#0，重放片断流文件“00003.PS”。在片断流文件“00003.PS”中，如关于相应于该片断流文件“00003.PS”的图 26 中的片断信息文件“00003.CLP”所描述的，将三个基本流，即，由为 0xE0 的 stream_id 所指定的视频流 stream#0、由为 0xE1 的 stream_id 所指定的视频流 stream#1、和由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的音频流 stream#3 多路复用。

[0757] 相应地，具有为 0xE0 的 stream_id 且在片断流文件“00003.PS”中多路复用的视频流 stream#0 与在图 28 中的下区中的 PlayListMark() 中的第二 Mark() 相关联，以及具有为 0xE1 的 stream_id 且在片断流文件“00003.PS”中多路复用的视频流 stream#1 与第三 Mark() 相关联。

[0758] 当启动第二 PlayList#1 中 PlayItem#0 的重放时，如在图 30 中的步骤 S124 中所描述的，播放器控制模块 212 辨识在图 28 中的下区中所示的 PlayListMark() 中所包括的三个 Mark() 项属于 PlayList#1 中的 PlayItem#0，并将该三个 Mark() 项的为 {90,000}、{27,090,000}、和 {27,540,000} 的 mark_time_stamp_items 与由它们所表示的时间的属性为“MarkProcess”的事实一起传递给解码控制模块 214。

[0759] 在标记处理中，当正重放 PlayList#1 中的 PlayItem#0 时，解码控制模块 214 不断地确认由定时器单元 214A 所度量的当前时间是否与其属性为“Mark Process”的时间 {90,000}、{27,090,000}、和 {27,540,000} 中的一个一致（步骤 S301）。当当前时间与其属性作为“Mark Process”的一个时间一致时，解码控制模块 214 将为具有属性“Mark Process”的时间的标记时间和指示当前时间作为具有属性“Mark Process”的时间的消息供给播放器控制模块 212。

[0760] 换句话说，例如，如果当前时间与具有“Mark Process”属性的时间 {90,000}、{27,090,000}、和 {27,540,000} 之中的 27,090,000 一致，则解码控制模块 214 将为具有“Mark Process”属性的时间的标记时间 27,090,000 和指示当前时间为具有“Mark Process”属性的时间的消息供给播放器控制模块 212。

[0761] 播放器控制模块 212 辨识目前正重放 PlayList#1 中的 PlayItem#0。相应地，通过将作为在图 28 中的下区中所示的 PlayList#1 的 PlayListMark() 中所描述的 Mark() 项之中属于 PlayItem#0 的三个 Mark() 项的 mark_time_stamp_items 的 {90,000}、{27,090,000}、和 {27,540,000} 中的每一个和来自解码控制模块 214 的标记时间 27,090,000 作比较，播放器控制模块 212 辨识 mark_time_stamp 与标记时间 27,090,000 一致的 Mark()，即，在图 28 中的下区中的 PlayListMark() 中所描述的第二 Mark()(Mark#1) 为处理目标标记（步骤 S302)。

[0762] 在作为处理目标标记的，图 28 中的下区中的 PlayListMark() 中所描述的第二 Mark() 中，设定 0xE0 为 entry_ES_stream_id。如上文所描述的，为 0xE0 的 entry_ES_stream_id 表示具有为 0xE0 的 stream_id 且在片断流文件“00003.PS”中多路复用的视频流 stream#0（图 26)。相应地，播放器控制模块 212 确定在正重放的基本流中是否包括视频流 stream#0（步骤 S303、S304)。

[0763] 另外，如果在正重放的基本流中，不包括视频流 stream#0，则忽略该处理目标标记（步骤 S304)。

[0764] 可选择地，如果在正重放的基本流中，包括视频流 stream#0，则该处理目标标记被看作为有效，且执行与该处理目标标记相匹配的处理（步骤 S305 至 S308)。

[0765] 换句话说，在这种情形中，在图 28 中的下区中的 PlayListMark() 中所描述的第二 Mark() 为事件标记，因为 mark_type 为“Event”。相应地，播放器控制模块 212 将表示事件的发生的事件消息和处理目标标记的 mark_data 供给脚本控制模块 211（步骤 S305、S307)。通过将来自播放器控制模块 212 的事件消息用作中断请求，当将与事件消息一起而供给的 mark_data 用作自变量时，脚本控制模块 211 执行在“SCRIPT.DAT”文件中预先描述的连续处理（步骤 S308)。

[0766] 如上文所描述的，根据标记处理，确定当前时间是否与 mark_time_stamp 一致，该当前时间为根据 PlayList()（图 5) 而重放的片断流文件的重放时间，该 PlayList() 包括具有包括表示在 PlayList() 的时间基础上的一个重放时间的 mark_time_stamp、表示 Mark() 的类型的 mark_type、和充当事件标记的自变量的 mark_data 的零个或多个 Mark() 项的 PlayListMark()（图 7)。当当前时间与 mark_time_stamp 一致时，辨识由于与 mark_time_stamp 一致的当前时间而具有等于标记时间的 mark_time_stamp 的 Mark() 作为处理目标标记。另外，当处理目标标记的 mark_type 表示产生事件的类型时，即，当处理目标标记为事件标记时，报告处理目标标记的 mark_data 和事件消息，并执行与 mark_data 相匹配的处理。因此，根据片断流文件的重放时间，能够执行与 mark_data 相匹配的处理。

[0767] “输出属性控制处理”

[0768] 接下来，参考图 41 中的流程图描述在步骤 S126，等等中所执行的输出属性控制处理的详细情况。

[0769] 如在图 30 中的步骤 S126 中所描述的，首先，关于将要重放的一个或多个基本流，即，将要重放的在图 30 中的步骤 S125 中所确定的一个或多个基本流的每一个，播放器控制模块 212 检查表示描述输出属性的 DynamicInfo() 项（图 13) 的数目的 number_of_DynamicInfo（图 10)。

[0770] 当关于所有的将要重放的一个或多个基本流，number_of_DynamicInfo 都为 0 时，

播放器控制模块 212 并不具体地执行处理。

[0771] 可选择地，当关于将要重放的基本流，number_of_DynamicInfo 不为 0 时，根据图 41 的流程图，播放器控制模块 212 执行输出属性控制处理。

[0772] 相应地，在盘 101 上所记录的三个片断信息文件“00001.CLP”、“00002.CLP”、和“00003.CLP”为，例如，如图 26 中所示的情形中，当重放相应于片断信息文件“00001.CLP”的片断流文件“00001.PS”（用于重放的第一列表 PlayList#0 中的第一项 PlayItem#0）时，在片断信息文件“00001.CLP”（图 26）中，关于片断流文件“00001.PS”中多路复用的所有的四个基本流 stream#0 至 stream#3，number_of_DynamicInfo 为 0，以便不执行输出属性控制处理。

[0773] 类似地，同样在重放相应于片断信息文件“00002.CLP”的片断流文件“00002.PS”（用于重放的第一列表 PlayList#0 中的第二项 PlayItem#1）的情形中，在片断信息文件“00002.CLP”（图 26）中，关于片断流文件“00002.PS”中多路复用的所有的四个基本流 stream#0 至 stream#3，number_of_DynamicInfo 为 0，以便不执行输出属性控制处理。

[0774] 另外，在重放相应于片断信息文件“00003.CLP”的片断流文件“00003.PS”（用于重放的第二列表 PlayList#1 中的 PlayItem#0）的情形中，在片断信息文件“00003.CLP”（图 26）中，在片断流文件“00003.PS”中多路复用的三个基本流 stream#0 至 stream#2 之中，关于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，它们的 number_of_DynamicInfo 项分别为 2 和 3，以便执行输出属性控制处理。

[0775] 换句话说，在输出属性控制处理中，首先，在步骤 S320，播放器控制模块 212 将在相应于将要重放的片断流文件的片断信息文件 Clip() 中所描述的 pts_change_point 与该 pts_change_point 为具有“DynamicInfo()Process”的属性的时间的事实一起传递给解码控制模块 214。解码控制模块 214 从播放器控制模块 212 接收作为具有“DynamicInfo()Process”的属性的时间的 pts_change_point，并前进到步骤 S321。

[0776] 在步骤 S321，解码控制模块 214 确定由定时器单元 214A 所度量的当前时间是否与作为具有“DynamicInfo()Process”的属性的时间的（一个）pts_change_point 一致。如果确定当前时间与 pts_change_point 不一致，则处理返回到步骤 S321。

[0777] 可选择地，如果在步骤 S321 确定当前时间与具有“DynamicInfo()Process”的属性的（一个）时间一致，则解码控制模块 214 将指示当前时间为具有“DynamicInfo()Process”的属性的时间的消息和与当前时间一致的具有“DynamicInfo()Process”的属性的时间，供给播放器控制模块 212，并前进到步骤 S322。

[0778] 在步骤 S322，播放器控制模块 212 从解码控制模块 214 接收指示当前时间为具有“DynamicInfo()Process”的属性的时间的消息和 DynamicInfo 时间，并辨识 DynamicInfo() 作为将经历处理的处理目标 DynamicInfo()，该 DynamicInfo() 和与 DynamicInfo 时间一致的 pts_change_point（图 10）构成一组。处理前进到步骤 S323。

[0779] 在步骤 S323，播放器控制模块 212 将在视为处理目标 DynamicInfo() 的 DynamicInfo()（图 13）中所描述的输出属性供给图形处理模块 219 或音频输出模块 221，并前进到步骤 S324。

[0780] 在步骤 S324，根据在上一个步骤 S323 中从播放器控制模块 212 所供给的输出属性，图形处理模块 219 或音频输出模块 221 启动输出视频数据或音频数据的控制，并返回到

步骤 S321。

[0781] 根据例如描述为输出属性（显示形式）的屏幕高宽比，输出视频数据。可选择地，根据例如描述为输出属性的立体声或双声道（两声道）模式，输出音频数据。

[0782] 接下来，参考图 42 进一步描述输出属性控制处理的详细情况。

[0783] 换句话说，图 42 示出在片断信息文件“00003.CLP”中所描述的 pts_change_point 和 DynamicInfo() 项的组（图 10）。

[0784] 这里，如上文所描述的，在片断流文件“00003.PS”中多路复用的三个基本流 stream#0 至 stream#2 之中，在图 26 中的片断信息文件“00003.CLP”中，关于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，number_of_DynamicInfo 项分别为 2 和 3。因此，在片断信息文件“00003.CLP”中，关于片断流文件“00003.PS”中的第一视频流 stream#0，描述了两组 pts_change_point 和 DynamicInfo() 项，且关于第三音频流 stream#2，描述了三组 pts_change_point 和 DynamicInfo() 项。

[0785] 在图 42 的上区示出关于片断流文件“00003.PS”中的第一视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项，以及在图 42 的下区示出关于片断流文件“00003.PS”中的第三音频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项。

[0786] 在图 42 的上区，除关于第一视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项之外，也示出了在片断信息文件“00003.CLP”中所描述的关于视频流 stream#0 的 stream_id(＝0xE0)、private_stream_id(＝0x00)、和 number_of_DynamicInfo(＝2)。类似地，同样在图 42 的下区，除关于第三音频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项之外，也示出了在图 26 的片断信息文件“00003.CLP”中所描述的关于音频流 stream#2 的 stream_id(＝0xBD)、private_stream_id(＝0x00)、和 number_of_DynamicInfo(＝3)。

[0787] 在图 42 的上区，在关于视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项的第一组中，pts_change_point 为 90,000，以及 DynamicInfo() 中的 display_aspect_ratio(图 13) 为“4 ： 3”。另外，在第二组中，pts_change_point 为 54,090,000，DynamicInfo() 中的 display_aspect_ratio 为“16 ： 9”。

[0788] 另外，在图 42 的下区，在关于音频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项之中，在第一组中，pts_change_point 为 90,000，以及 DynamicInfo() 中的 channel_assignment 为“Dual”。在第二组中，pts_change_point 为 27,090,000，以及 DynamicInfo() 中的 channel_assignment 为“Stereo”。在第三组中，pts_change_point 为 32,490,000，以及 DynamicInfo() 中的 channel_assignment 为“Dual”。

[0789] 例如，假设在图 30 中的步骤 S125，在片断流文件“00003.PS”中，将由为 0xE0 的 stream_id 所指定的第一视频流 stream#0，和由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的第三音频流 stream#2 确定为将要重放的流。

[0790] 在这种情形，播放器控制模块 212 检查在图 42 的上区的关于由为 0xE0 的 stream_id 所指定的视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项、在图 42 的下区的关于由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的视频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项、以及 DynamicInfo() 中的

pts_change_point，并辨识初始值。

[0791] 换句话说，在图 42 的上区的关于由为 0xE0 的 stream_id 所指定的视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项的第一组中，pts_change_point 为 90,000。为 90,000 的这个时间与在图 26 的片断信息文件“00003.CLP”中、表示片断流文件“00003.PS”中的开始时间的 presentation_start_time 中所描述的时间 90,000 一致，其相应于其中将视频流 stream#0 多路复用的片断流文件“00003.PS”。

[0792] 类似地，在图 42 的下区的关于由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的视频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项之中，在第一组中，pts_change_point 为 90,000。为 90,000 的这个时间与在片断信息文件“00003.CLP”中、表示片断流文件“00003.PS”中的开始时间的 presentation_start_time 中所描述的时间 90,000 一致，片断流文件“00003.PS”相应于其中将音频流 stream#2 多路复用的剪辑流文件“00003.PS。

[0793] 播放器控制模块 212 辨识与在表示片断流文件“00003.PS”中的开始时间的 presentation_start_time 中所描述的时间 90,000 一致的 pts_change_point 作为初始值。相应地，辨识在图 42 的上区的两组 pts_change_point 和 DynamicInfo() 项的第一组中的 pts_change_point，和在图 42 的下区的三组 pts_change_point 和 DynamicInfo() 项之中的第一组中的 pts_change_point 为初始值。

[0794] 在启动片断流文件“00003.PS”的重放（在图 30 的步骤 S126）之前，根据以辨识为初始值的 pts_change_point 构成一个组的 DynamicInfo()，播放器控制模块 212 指定相应的基本流的输出属性。

[0795] 换句话说，关于由为 0xE0 的 stream_id 所指定的视频流 stream#0，在以作为初始值 90,000 的 pts_change_point 构成一个组的 DynamicInfo() 中，display_aspect_ratio 为“4 ： 3”。在这种情形，播放器控制模块 212 通过使用 display_aspect_ratio 为“4 ： 3”的事实，即，指示视频流 stream#0 为具有 4 ： 3 的屏幕高宽比的视频数据的输出属性信息，来控制图形处理模块 219。

[0796] 另外，关于由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的音频流 stream#2，在以作为初始值 90,000 的 pts_change_point 构成一个组的 DynamicInfo() 中，channel_assignment 为“Dual”。在这种情形，播放器控制模块 212 将 channel_assignment 为“Dual”的事实，即，指示音频流 stream#2 为双声道音频数据的输出属性信息供给音频输出模块 221。

[0797] 这里，在图 30 中的步骤 S126，执行如上文所描述的关于作为初始值的 pts_change_point 的输出属性控制。

[0798] 在此之后，播放器控制模块 212 将，在图 42 的上区的关于视频流 stream#0 的两个 pts_change_point 项 90,000 和 54,090,000，以及在图 42 的下区的关于音频流 stream#2 的三个 pts_change_point 项 90,000、27,090,000、和 32,490,000 之中，除初始值 90,000 之外的时间 {27,090,000}、{32,490,000}、和 {54,090,000}，与具有属性“DynamicInfo() Processing”的时间一起传递给解码控制模块 214。

[0799] 在解码控制模块 214 从播放器控制模块 212 接收了具有“DynamicInfo() Processing”的时间 {27,090,000}、{32,490,000}、和 {54,090,000}，并启动了重放（重

放用于重放片断流文件“00003.PS”的第二列表 PlayList#1 中的 PlayItem#0) 视频流 stream#0 和音频数据 stream#2 之后，解码控制模块 214 启动监控由定时器单元 214A 所度量的当前时间。

[0800] 另外，当当前时间与具有“DynamicInfo()Processing”的时间 {27,090,000}、{32,490,000}、和 {54,090,000} 的一个一致时，解码控制模块 214 将作为具有“DynamicInfo()Processing”属性并与当前时间一致的时间的 DynamicInfo 时间供给播放器控制模块 212。(步骤 S321)。

[0801] 换句话说，例如，如果当前时间为 27,090,000，则解码控制模块 214 将在具有“DynamicInfo()Processing”属性的时间之中与当前时间一致的 27,090,000 作为 DynamicInfo 时间供给播放器控制模块 212。

[0802] 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，从在图 42 的上区的关于视频流 stream#0 的两个 pts_change_point 项和在图 42 的下区的关于音频流 stream#2 的三个 pts_change_point 项之中检查与 DynamicInfo 时间 27,090,000 一致的 pts_change_point，并辨识以与 27,090,000 一致的 pts_change_point 构成一个组的 DynamicInfo()，即，在图 42 的下区的关于音频流 stream#2 的第二 DynamicInfo()，作为处理目标 DynamicInfo()(步骤 S322)。

[0803] 当处理目标 DynamicInfo() 为关于视频流时，播放器控制模块 212 将处理目标 DynamicInfo() 中所描述的输出属性供给图形处理模块 219(步骤 S323)。另外，处理目标 DynamicInfo() 为关于音频流时，播放器控制模块 212 将处理目标 DynamicInfo() 中所描述的输出属性供给音频输出模块 221(步骤 S323)。

[0804] 当已从播放器控制模块 212 供给了输出属性时，根据所供给的输出属性，图形处理模块 219 启动控制视频数据输出 (步骤 S324)。

[0805] 换句话说，例如，基于由来自播放器控制模块 212 的输出属性所表示的视频数据屏幕高宽比指令 (display_aspect_ratio(图 13))，以及连接到图 1 中的视频输出终端 120 的视频输出装置的屏幕高宽比，图形处理模块 219 转换输出到视频输出模块 220 的视频数据的屏幕高宽比。

[0806] 特殊地，例如，当视频输出装置的屏幕高宽比为 16 ： 9 时，如果作为输出属性的视频数据屏幕高宽比的指令表示 4 ： 3 的屏幕高宽比，则通过对输出到视频输出模块 220 的视频数据执行水平挤压，图形处理模块 219 输出右和左部分变暗的经处理过的视频数据。另外，例如，当视频输出装置的屏幕高宽比为 4 ： 3 时，如果作为输出属性的视频数据屏幕高宽比的指令表示 16 ： 9 的屏幕高宽比，则通过对输出到视频输出模块 220 的视频数据执行垂直挤压，图形处理模块 219 输出顶和底部分变暗的经处理过的视频数据。此外，例如，当视频输出装置的屏幕高宽比和由作为输出属性的视频数据的屏幕高宽比的指令所表示的屏幕高宽比具有 4 ： 3 或 16 ： 9 的相同的值时，图形处理模块 219 直接输出输出到视频输出模块 220 的视频数据，而不执行挤压。

[0807] 这里，根据关于由为 0xE0 的 stream_id 所指定的视频流 stream#0 而描述的两组 pts_change_point 和 DynamicInfo() 项，从作为视频流 stream#0 的重放开始时间的时间 90,000，直到紧邻时间 54,090,000 之前，从视频流 stream#0 获取具有 4 ： 3 的屏幕高宽比的视频数据。在时间 54,090,000 之后，从视频流 stream#0 获取具有 16 ： 9 的屏幕高宽比

的视频数据。

[0808] 因此，例如，如果连接到图 1 中的视频输出终端 120 的视频输出装置的屏幕高宽比为 4 ： 3，从时间 90,000 直到紧邻时间 54,090,000 之前，图形处理模块 219 直接将从视频流 stream#0 所获得的，具有 4 ： 3 的屏幕高宽比的视频数据供给具有 4 ： 3 的屏幕高宽比的视频输出装置。

[0809] 另外，在时间 54,090,000 之后，垂直地挤压具有 16 ： 9 的屏幕高宽比且从视频流 stream#0 所获得的视频数据，并将其转换为具有 4 ： 3 的屏幕高宽比且在顶和底部变暗的视频信号。该视频数据供给并显示在具有 4 ： 3 的屏幕高宽比的视频输出装置上。

[0810] 当已从播放器控制模块 212 供给了输出属性时，根据该输出属性，音频输出模块 221 启动控制音频数据的输出（步骤 S324）。

[0811] 换句话说，例如，基于由来自播放器控制模块 212 的输出属性所表示的音频数据声道分配指令 (channel_assignment（图 13))，以及由用户操作遥控器通过输入接口从播放器控制模块 212 所供给的音频输出模式，音频输出模块 221 处理来自音频解码器控制模块 217 的音频数据并将经处理的数据输出到音频输出终端 121（图 1）。

[0812] 特殊地，例如，当由输出属性所表示的音频数据声道分配指令表示其中左声道相应于“主音频”的音频数据以及右声道相应于“次音频”的音频数据的双声道（两声道）模式时，根据从播放器控制模块 212 所供给的音频输出模式，音频输出模块 221 处理来自音频解码器控制模块 217 的音频数据，并将经处理的数据输出至音频输出终端 121。

[0813] 换句话说，当例如指定“主音频”为音频输出模式时，音频输出模块 221 复制来自音频解码器控制模块 217 的音频数据中的左声道音频数据作为右声道音频数据，并将左声道和右声道音频数据（“主音频”的音频数据）输出至音频输出终端 121。另外，当指定“次音频”为音频输出模式时，音频输出模块 221 复制来自音频解码器控制模块 217 的音频数据中的右声道音频数据作为左声道音频数据，并将左声道和右声道音频数据（“次音频”的音频数据）输出至音频输出终端 121。当指定“主和次音频”为音频输出模式时，音频输出模块 221 直接将来自音频解码器控制模块 217 的音频数据输出至音频输出终端 121。

[0814] 另外，例如当音频数据声道分配指令表示立体声模式时，音频输出模块 221 直接将来自音频解码器控制模块 217 的音频数据输出至音频输出终端 121，而不管从播放器控制模块 212 所供给的音频输出模式。

[0815] 这里，根据在图 42 的下区的关于由为 0xBD 的 stream_id 和为 0x00 的 private_stream_id 所指定的音频流 stream#2 而描述的三组 pts_change_point 和 DynamicInfo() 项，从作为音频流 stream#2 的重放开始时间的 90,000，直到紧邻时间 27,090,000 之前，从音频流 stream#2 获得双声道音频数据。另外，从时间 27,090,000 直到紧邻时间 32,490,000 之前，从音频流 stream#2 获得立体声音频数据。在时间 32,490,000 之后，从音频流 stream#2 获得双声道音频数据。

[0816] 因此，例如，如果指定“主音频”为音频输出模式时，则从时间 90,000 直到紧邻时间 27,090,000 之前，音频输出模块 221 复制从音频流 stream#2 所获得的双声道音频数据中的左声道音频数据作为右声道音频数据，并将左声道和右声道音频数据输出至音频输出终端 121。

[0817] 另外，从时间 27,090,000 直到紧邻时间 32,490,000 之前，直接将从音频流

stream#2 所获得立体声音频数据输出至音频输出终端 121。

[0818] 在时间 32,490,000 之后，复制从音频流 stream#2 所获得的双声道音频数据中的左声道音频数据作为右声道音频数据，并将左声道和右声道音频数据输出至音频输出终端 121。

[0819] 如上文所描述的，在输出属性控制处理中，关于在片断流文件中多路复用的基本流的每一个，基于包括表示基本流的重放时间的零或多组 pts_change_point 项和包括基本流的输出属性的 DynamicInfo() 项的片断信息文件 Clip()（图 10），确定正重放的一个基本流的重放时间是否与 pts_change_point 一致。如果正重放的一个基本流的重放时间与 pts_change_point 一致，则辨识以该 pts_change_point 构成一个组的 DynamicInfo()，并根据在所辨识的 DynamicInfo() 中所包括的输出属性，控制正重放的基本流的输出。因此，能够根据基本流的重放时间和输出属性控制正重放的基本流的输出。

[0820] 字幕显示控制处理

[0821] 接下来，参考图 43 中的流程图描述控制相应于字幕流的字幕数据的显示的字幕显示控制处理。

[0822] 当启动 PlayList()（图 5）（中的 PlayItem()）的重放时，在步骤 S341，播放器控制模块 212 为图形处理模块 219 初始化字幕数据显示形式指令。换句话说，播放器控制模块 212 控制图形处理模块 219，以便设定字幕数据显示形式为默认的。在步骤 S341 所执行的显示形式的初始化相应于在图 30 中的步骤 S127 中所描述的显示形式指令的初始化。

[0823] 在步骤 S341 的处理之后前进到步骤 S342，播放器控制模块 212 确定是否已经接收到由用户操作遥控器从输入接口 115 而发送的关于字幕数据显示的新的显示形式指令。

[0824] 如果，在步骤 S342 确定接收到新的显示形式的指令，则播放器控制模块 212 前进到步骤 S343，并确定目前是否正重放字幕流（相应于该字幕流的字幕数据）。

[0825] 如果在步骤 S343 确定未重放字幕流，则播放器控制模块 212 返回到步骤 S342。

[0826] 可选择地，如果在步骤 S343 确定正在重放字幕流，则播放器控制模块 212 前进到步骤 S345，并确定新的显示形式指令是否为默认显示形式指令。如果在步骤 S345 确定新的显示形式指令为默认显示形式指令，则播放器控制模块 212 返回到步骤 S341，并控制图形处理模块 219，以便设定字幕数据显示形式为默认的，如上文所描述。

[0827] 如果在步骤 S345 确定新的显示形式指令不为默认显示形式指令，即，当新的显示形式指令为像以放大或缩小的形式显示字幕数据、或为了清楚的画面而改变亮度那样的非默认显示形式指令时，处理前进到步骤 S346。播放器控制模块 212 从在相应于将目前正重放的字幕流多路复用的片断流文件的片断信息文件 Clip()（图 10）中的 StaticInfo() 项（图 12）之中，获得关于目前正重放的字幕流的 StaticInfo()，并前进到步骤 S347。

[0828] 在步骤 S347，播放器控制模块 212 识别在步骤 S346 中所获得的 StaticInfo() 的 configurable_flag。

[0829] 如果在步骤 S347 确定 configurable_flag 为 0，其指示不允许字幕数据显示形式的改变，则播放器控制模块 212 前进到步骤 S348，并控制图形处理模块 219，由此指示不能改变字幕数据显示形式的错误消息覆盖在输出视频数据上。处理返回到步骤 S342。这显示错误消息。

[0830] 可选择地，如果在步骤 S347 确定 configurable_flag 为 1，其指示允许字幕数据显

示形式的改变，则播放器控制模块 212 前进到步骤 S349，并将由用户操作遥控器从输入接口 115 所供给的新的显示形式指令供给图形处理模块 219。处理前进到步骤 S350。

[0831] 在步骤 S350，图形处理模块 219 启动根据在上一个步骤 S349 从播放器控制模块 212 所供给的显示形式指令，放大或缩小从字幕解码器控制模块 218 所供给的字幕数据，改变其亮度，或诸如此类的处理，并返回到步骤 S342。以根据由用户操作遥控器所指定的显示形式的显示大小、显示位置、显示色彩、等等显示字幕数据。

[0832] 可选择地，如果在步骤 S342 确定未接收到新的显示形式的指令，则播放器控制模块 212 前进到步骤 S351，并确定是否已经执行图 31 中所描述的 PlayItem() 改变。如果确定还未执行 PlayItem() 改变，则处理返回到步骤 S342。

[0833] 另外，如果在步骤 S351 确定已经执行 PlayItem() 改变，则播放器控制模块 212 返回到步骤 S341，并控制图形处理模块 219，以便如上文所描述的，设定字幕数据显示形式为默认的。换句话说，在这种情形，如果已经执行 PlayItem() 改变，则字幕数据显示形式返回到默认显示形式。

[0834] 如上文所描述的，在字幕显示控制处理中，仅字幕流中的 configurable_flag 为 1，其指示允许显示形式的改变的情形中，例如响应由用户操作遥控器而输入的显示形式指令，改变关于相应于字幕流的字幕数据的显示形式。

[0835] 相应地，根据，例如，图 26 中所示的片断信息文件“00001.CLP”，在相应的片断流文件“00001.PS”中多路复用的四个基本流之中，关于作为第三基本流的字幕流 stream#2，configurable_flag 为 0，其指示不允许显示形式的改变。因而，即使当显示字幕流 stream#2 时用户操作遥控器改变字幕，它们的显示也不改变。

[0836] 例如，在相应的片断流文件“00001.PS”中多路复用的四个基本流之中，关于作为第四基本流的字幕流 stream#3，configurable_flag 为 1，其指示允许显示形式的改变。因而，当显示字幕流 stream#3 时用户操作遥控器改变字幕时，响应该操作，改变字幕显示大小，等等。

[0837] 换句话说，例如，假设根据图 25 中的第一 PlayList#0 中的第一 PlayItem#0，正重放片断流文件“00001.PS”。另外，如关于图 26 中的片断信息文件“00001.CLP”所描述的，在相应的片断流文件“00001.PS”中多路复用的四个基本流之中，第三和第四基本流为字幕流。假设在第三字幕流 stream#2 和第四字幕流 stream#3 之间，例如，目前正重放第三字幕流 stream#2。

[0838] 通过由用户操作遥控器输入字幕显示形式指令（步骤 S342），从输入接口 115（图 1）将显示形式指令供给播放器控制模块 212。当供给了显示形式指令时，播放器控制模块 212 在片断信息文件中搜索相应于目前正重放的字幕流的 StaticInfo()（图 10）（步骤 S346）。

[0839] 换句话说，在这种情形，目前正重放的字幕流为第三字幕流 stream#2，其在片断流文件“00001.PS”中多路复用。相应地，播放器控制模块 212 在相应的片断信息文件“00001.CLP”中搜索关于第三字幕流 stream#2 的 StaticInfo()。

[0840] 此外，播放器控制模块 212 识别为 0 的，在图 26 中关于第三字幕流 stream#2 的 StaticInfo() 中所描述的 configurable_flag（步骤 S347），由此辨识关于第三字幕流 stream#2，不允许显示形式的改变。

[0841] 在这种情形，通过确定目前正重放的字幕流（相应于该字幕流的字幕数据）不适合于放大、缩小，等等，以及控制图形处理模块 219，播放器控制模块 212 产生表示所述确定的错误消息（步骤 S348），并以覆盖在视频数据上的形式输出该消息。

[0842] 另外，在相应的片断流文件“00001.PS”中多路复用的四个基本流之中，在第三字幕流 stream#2 和第四字幕流 stream#3 之间，目前重放的不是第三字幕流 stream#2，而是第四字幕流 stream#3。响应遥控器的用户的操作而供给有显示形式指令的播放器控制模块 212 在相应的片断信息文件“00001.CLP”中搜索关于第四字幕流 stream#3 的 StaticInfo()。

[0843] 此外，播放器控制模块 212 识别为 1 且在关于图 26 中的第四字幕流 stream#3 的 StaticInfo() 中所描述的 configurable_flag（步骤 S347），由此辨识关于第四字幕流 stream#3，允许显示形式的改变。

[0844] 在这种情形，通过确定目前正重放的字幕流（相应于该字幕流的字幕数据）适合于放大、缩小，等等，播放器控制模块 212 将由用户操作遥控器而供给的显示形式指令供给图形处理模块 219（步骤 S349）。

[0845] 相应地，在那之后，响应来自播放器控制模块 212 的显示形式指令，图形处理模块 219 对来自字幕解码器控制模块 218 的字幕数据执行放大、缩小，或诸如此类，并以覆盖在来自视频解码器控制模块 216 的视频数据的形式输出所获得的数据。

[0846] 在 PlayList() 中的第一 PlayItem() 的重放的开始，播放器控制模块 212 为图形处理模块 219 初始化字幕数据显示形式指令（步骤 S341）。换句话说，播放器控制模块 212 控制图形处理模块 219，以便设定字幕数据显示形式为默认的。

[0847] 此外，同样在 PlayItem() 改变时间，播放器控制模块 212 为图形处理模块 219 初始化字幕数据显示形式指令（步骤 S341、S351）。

[0848] 然而，在 PlayItem() 改变时间，检查关于根据随后重放的新的 PlayItem() 而重放的新的字幕流的 configurable_flag。当 configurable_flag 为 0 时，能够为图形处理模块 219 初始化字幕数据显示形式指令。当 configurable_flag 为 1 时，能够为图形处理模块 219 将字幕数据显示形式指令维持为在 PlayItem() 改变之前的状态。

[0849] 在图 43 中的字幕显示控制处理中，当由用户操作遥控器而输入新的显示形式指令时，将该新的显示形式指令供给图形处理模块 219（步骤 S349）。然而，能够将该显示形式指令存储在例如构成存储器 113（图 1）的非易失性存储器中，且能够将该非易失性存储器中所存储的显示形式指令供给图形处理模块 219。

[0850] 换句话说，在关于图 1 中的盘装置的初始设置中，通过在非易失性存储器中存储用户设定的显示形式指令，当由用户操作遥控器而输入新的显示形式指令时，更新在非易失性存储器中所存储的显示形式指令。另外，能够将非易失性存储器中所存储的显示形式指令供给图形处理模块 219。在这种情形，在非易失性存储器中存储在上一个重放结束时间的显示形式指令。因而，通过由用户在下一个 PlayList() 重放时间操作遥控器，以该显示形式启动字幕数据的显示，而不必在上一个重放结束时间输入显示形式指令。

[0851] 在这种情形，在非易失性存储器中所存储的显示形式指令包括，例如，放大率或缩小率，以在放大或缩小字幕数据中使用。

[0852] 如上文所描述的，根据字幕显示控制处理，对于片断信息文件 Clip()（图 10）中

所包括的基本流，在基本流的重放期间不改变的 StaticInfo() 项之中，获取字幕数据的 StaticInfo()，并基于 StaticInfo() 中所包括的且指示是否允许从默认显示形式的字幕数据显示的改变的 configurable_flag，确定是否允许从默认显示形式的字幕数据显示的改变。另外，如果允许从默认显示形式的字幕数据显示的改变，根据改变字幕数据显示形式的指令，执行字幕数据显示过程，即，例如，放大或缩小字幕数据或改变显示色彩的显示处理。因此，能够控制字幕数据显示形式的改变。

[0853] 捕获控制处理

[0854] 接下来，参考图 44 中的流程图描述控制相应于视频流的视频数据的捕获的捕获控制处理。除阐明捕获控制处理的流程图之外，图 44 也包括阐明作为执行由捕获控制处理所捕获的视频数据的二次使用的处理的例子的背景 / 屏幕保护处理的流程图。

[0855] 开始捕获控制处理，以便例如通过由用户操作遥控器，将作为视频数据捕获指令的捕获指令通过输入接口 115（图 1）供给播放器控制模块 212。

[0856] 换句话说，在捕获控制处理中，首先，在步骤 S371，确定播放器控制模块 212 是否正重放视频流。如果确定播放器控制模块 212 未重放视频流，则捕获控制处理结束。

[0857] 可选择地，在步骤 S371，如果确定播放器控制模块 212 正重放视频流，则播放器控制模块 212 前进到步骤 S372，从相应于正重放的视频流的 PlayList()（图 5）中获取 capture_enable_flag_PlayList，并从相应于正重放的视频流的片断信息文件 Clip()（图 10）中获取 capture_enable_flag_Clip。

[0858] 如图 5 中所描述的，PlayList() 中的 capture_enable_flag_PlayList 指示是否允许相应于基于该 PlayList() 而重放的视频流的视频数据（属于 PlayList() 的视频数据）的二次使用。另外，如图 10 中所描述的，片断信息文件 Clip() 中的 capture_enable_flag_Clip 指示是否允许相应于在相应于该片断信息文件 Clip() 的片断流文件中所存储的视频流的视频数据的二次使用。

[0859] 在步骤 S372 的处理之后前进到步骤 S373，基于在上一个步骤 S373 所获取的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip，播放器控制模块 212 确定当从输入接口 115 输入捕获指令时是否能够执行重放的视频数据的画面的捕获。

[0860] 如果在步骤 S373 确定当从输入接口 115 输入捕获指令时不能执行重放的视频数据的画面的捕获，即，当在上一个步骤 S373 所获取的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 的至少一个为 0，其指示不允许视频数据的二次使用时，处理前进到步骤 S374。通过控制图形处理模块 219，播放器控制模块 212 覆盖一个指示不能执行视频数据捕获的错误消息，且捕获控制处理结束。这显示错误消息。

[0861] 可选择地，如果在步骤 S373 确定当从输入接口 115 输入捕获指令时能够执行重放的视频数据的画面的捕获，即，当在上一个步骤 S373 所获取的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 两者都为 1，其指示允许视频数据的二次使用时，处理前进到步骤 S375。播放器控制模块 212 将捕获当从输入接口 115 输入捕获指令时而重放的视频数据的画面的指令供给控制图形处理模块 219，并前进到步骤 S376。

[0862] 在步骤 S376，根据来自播放器控制模块 212 的捕获指令，在图形处理模块 219 中，捕获来自视频解码器控制模块 216 的视频数据的画面并存储在存储器 113（图 1）中。捕获控制处理结束。capture_enable_flag 由多个位组成，因此，当使用条件受限制时，在这个时

间执行响应。换句话说，当所捕获的图像的大小有限制时，捕获在这个时间缩小的图像。另外，当使用的应用程序有约束时，同步记录报告该约束的标志。

[0863] 如上文所描述的，捕获控制处理找到相应于当捕获指令时的视频流重放数据的 PlayList()（图 5）和片断信息文件 Clip()（图 10）的每一个中的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 的逻辑积。当逻辑积为 1，即，仅当 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 两者都为一其允许二次使用时，过程确定能够执行视频数据的二次使用，以便执行捕获。

[0864] 因此，例如，当执行根据图 25 中的第一 PlayList()#0 中的第一 PlayItem#0 的视频流重放，即，在执行片断流文件“00001.PS”中多路复用的视频流的重放时，如果接收到来自用户的捕获指令，则第一 PlayList#0 中的 capture_enable_flag_PlayList 为 1，且相应于基于该第一 PlayItem#0 而重放的片断流文件“00001.PS”的片断信息文件“00001.CLP”中的 capture_enable_flag_Clip 为 1。因而，确定能够执行正重放的视频数据（相应于在片断流文件“00001.PS”中多路复用的视频流的视频数据）的二次使用，以便执行捕获。

[0865] 另外，例如，当执行根据图 25 中的第一 PlayList#0 中的第二 PlayItem#1 的视频流重放，即，在执行片断流文件“00002.PS”中多路复用的视频流的重放时，如果接收到来自用户的捕获指令，则第一 PlayList()#0 中的 capture_enable_flag_PlayList 为 1，且相应于基于该第二 PlayItem#1 而重放的片断流文件“00002.PS”的图 26 中的片断信息文件“00002.CLP”中的 capture_enable_flag_Clip 为 0。因而，确定不能执行正重放的视频数据（相应于在片断流文件“00002.PS”中多路复用的视频流的视频数据）的二次使用，以便不执行捕获。

[0866] 此外，例如，当执行根据图 25 中的第二 PlayList#1 中的 PlayItem#0 的视频流重放，即，在执行片断流文件“00003.PS”中多路复用的视频流的重放时，如果接收到来自用户的捕获指令，则第二 PlayList#1 中的 capture_enable_flag_PlayList 为 0，相应于基于该第二 PlayList#1 中的 PlayItem#0 而重放的片断流文件“00003.PS”的图 26 中的片断信息文件“00003.CLP”中的 capture_enable_flag_Clip 为 1。因而，确定不能执行正重放的视频数据（相应于在片断流文件“00003.PS”中多路复用的视频流的视频数据）的二次使用，以便不执行捕获。

[0867] 在这种情形，在确认第二 PlayList#1 中的 capture_enable_flag_PlayList 为 0 的时间，能够确定不能执行视频数据的二次使用。因而，能够忽略相应于基于该第二 PlayList#1 中的 PlayItem#0 而重放的片断流文件“00003.PS”的图 26 中的片断信息文件“00003.CLP”中的 capture_enable_flag_Clip 的确认。

[0868] 能够将在捕获控制处理中捕获之后在存储器 113 中所存储的画面在背景 / 屏幕保护处理中二次地使用。

[0869] 在如下状态中，执行背景 / 屏幕保护处理 :例如，尽管播放器控制模块 212 操作，但不执行基本流重放的状态，即，盘 101 未加载到盘驱动器 102（图 1）中的状态，或基本流重放已经结束的状态。

[0870] 换句话说，在背景 / 屏幕保护处理中，在步骤 S381，播放器控制模块 212 控制图形处理模块 219 显示在捕获控制处理中在存储器 113 中所存储的画面。在播放器控制模块 212 的控制下，图形处理模块 219 显示在捕获控制处理中在存储器 113 中所存储的画面。

[0871] 这里，在图形处理模块 219 中，如果在存储器 113 中所存储的画面显示为静止画面，则实现所谓的墙纸（背景），且如果其在放大、缩小、以及以固定的频率移动时显示，则实现屏幕保护。另外，不仅能够通过播放器控制模块 212，而且能够通过另一个独立的应用程序，执行显示通过捕获控制处理在存储器 113 中所存储的画面的背景 / 屏幕保护处理。

[0872] 当在这个时间将表示用户约束的标志添加到存储器 113 中所存储的画面时，该画面遵从该约束。

[0873] 如上文所描述的，以大于视频存取单元的单元获取关于正重放的视频数据的 capture_enable_flag_PlayList 和 capture_enable_flag_Clip，其每个都指示例如允许相应于 PlayList() 或 PlayItem() 的视频数据的二次使用，且基于 capture_enable_flag_PlayList 和 capture_enable_flag_Clip，确定是否允许正重放的视频数据的二次使用。如果确定允许正重放的视频数据的二次使用，则捕获正重放的视频数据，并执行使用所捕获的视频数据的背景 / 屏幕保护处理。相应地，视频数据的二次使用的控制成为可能。

[0874] 在图 44 中的捕获控制处理中，通过供给 PlayList 中的 capture_enable_flag_PlayList（图 5），供给相应于基于 PlayItem() 而重放的片断流文件的片断信息文件 Clip() 中的 capture_enable_flag_Clip（图 10），并使用 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 两者，确定是否允许二次使用。然而，通过仅供给仅只 PlayList()（图 5）中的 capture_enable_flag_PlayList，或仅供给相应于基于 PlayItem() 而重放的片断流文件的片断信息文件 Clip()（图 10）中的 capture_enable_flag_Clip，并使用任何一个，能够确定是否允许二次使用。

[0875] 另外，在图 44 中的捕获控制处理中，在步骤 S376，根据来自播放器控制模块 212 的捕获指令，图形处理模块 219 从视频解码器控制模块 216 捕获视频数据的画面，即，仅一个画面。然而，除此之外，能够捕获多个画面。换句话说，能够捕获由视频解码器控制模块 216 所输出的多个时间序列画面（作为活动画面的多个画面序列）。在这种情形，能够例如预先确定每次捕获的画面的数目。可选择地，通过扩展 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 的位，可以在 capture_enable_flag_PlayList 和 capture_enable_flag_Clip 中包括每次能够捕获的画面的数目的信息。

[0876] 此外，在上述情形中，在 PlayList() 和片断信息文件 Clip() 中描述指示是否允许视频数据的二次使用的用户许可信息 (capture_enable_flag_PlayList、capture_enable_flag_Clip)，并基于该用户许可信息，确定是否允许基于 PlayList() 而重放的视频数据的全体和相应于在相应于片断信息文件 Clip() 的片断流文件中多路复用的视频流的视频数据的全体的二次使用。用户许可信息能够以任意的单元描述其它视频数据，并基于这个用户许可信息，能够确定是否允许任意单元的视频数据的二次使用。

[0877] 换句话说，图 45 示出 private_stream2_PES_payload() 的语法，且图 46 示出在其中设置用户许可信息的 au_information() 的语法。

[0878] 除紧邻 video_stream_id 之前设置了作为用户许可信息的 capture_enable_flag_ps2 之外，图 45 中的 private_stream2_PES_payload() 在配置上与图 23 的情形类似。同样，除紧邻 pic_struct_copy 之前设置了作为用户许可信息的 capture_enable_flag_AU 之外，图 46 中的 au_information() 在配置上与图 24 的情形类似。

[0879] 图 45 中的 private_stream2_PES_payload() 中所设置的 capture_enable_

flag_ps2 指示是否允许相应于位于从包括 private_stream2_PES_payload() 的 private_stream_2 的 PES_packet() 至紧邻下一个 private_stream_2 的 PES_packet() 之前的位置的视频流的视频数据的二次使用。因此，根据图 45 中的 private_stream2_PES_payload() 中所设置的 capture_enable_flag_ps2，能够确定是否允许从解码可开始点至下一个解码可开始点的视频数据的二次使用。

[0880] 图 46 中的 au_information() 中所设置的 capture_enable_flag_AU 指示是否允许相应于该 capture_enable_flag_AU 的视频存取单元中的视频数据的二次使用。因此，根据图 46 中的 au_information() 中所设置的 capture_enable_flag_AU，能够确定是否允许一个视频存取单元或一个画面单元中的视频数据的二次使用。

[0881] 这里，关于作为 PlayList()（图 5）中的用户许可信息的 capture_enable_flag_PlayList、作为片断信息文件 Clip()（图 10）中的用户许可信息的 capture_enable_flag_Clip、作为 private_stream2_PES_payload() 中的用户许可信息的 capture_enable_flag_ps2、和作为 au_information()（图 46）中的用户许可信息的 capture_enable_flag_AU，在它们之中，能够重复地采用两个或多个项。在这种情形，基于重复采用的两或多个用户许可信息的逻辑积，或诸如此类，能够确定是否允许某些视频数据的二次使用。

[0882] 另外，关于包括设置 au_information() 的图 23 或图 45 中的 private_stream2_PES_payload() 的 private_stream_2 的 PES_packet()，如图 36 中的步骤 S211 中所描述的，缓冲器控制模块 215（图 3）中的视频读取功能单元 233 从缓冲器 215A 中所存储的节目流中搜索信息包。因此，在采用在其中设置 capture_enable_flag_ps2 的图 45 中的 private_stream2_PES_payload() 和在其中设置 capture_enable_flag_AU 的图 46 中的 au_information() 的情形中，在确定是否允许视频数据的二次使用中，播放器控制模块 212 需要询问视频读取功能单元 233 有关 capture_enable_flag_ps2 和 capture_enable_flag_AU 的情况。

[0883] 尽管，在这个实施例中，通过软件执行上文所描述的连续处理，但能够通过专用硬件执行上文所描述的连续处理。

[0884] 另外，尽管，在这个实施例中，采用硬件解码器作为视频解码器 116（图 1），但能够采用软件解码器作为视频解码器 116。这同样适用于音频解码器 117（图 1）。

[0885] 而且，尽管，在这个实施例中，采用软件解码器作为字幕解码器，但能够采用硬件解码器作为字幕解码器。

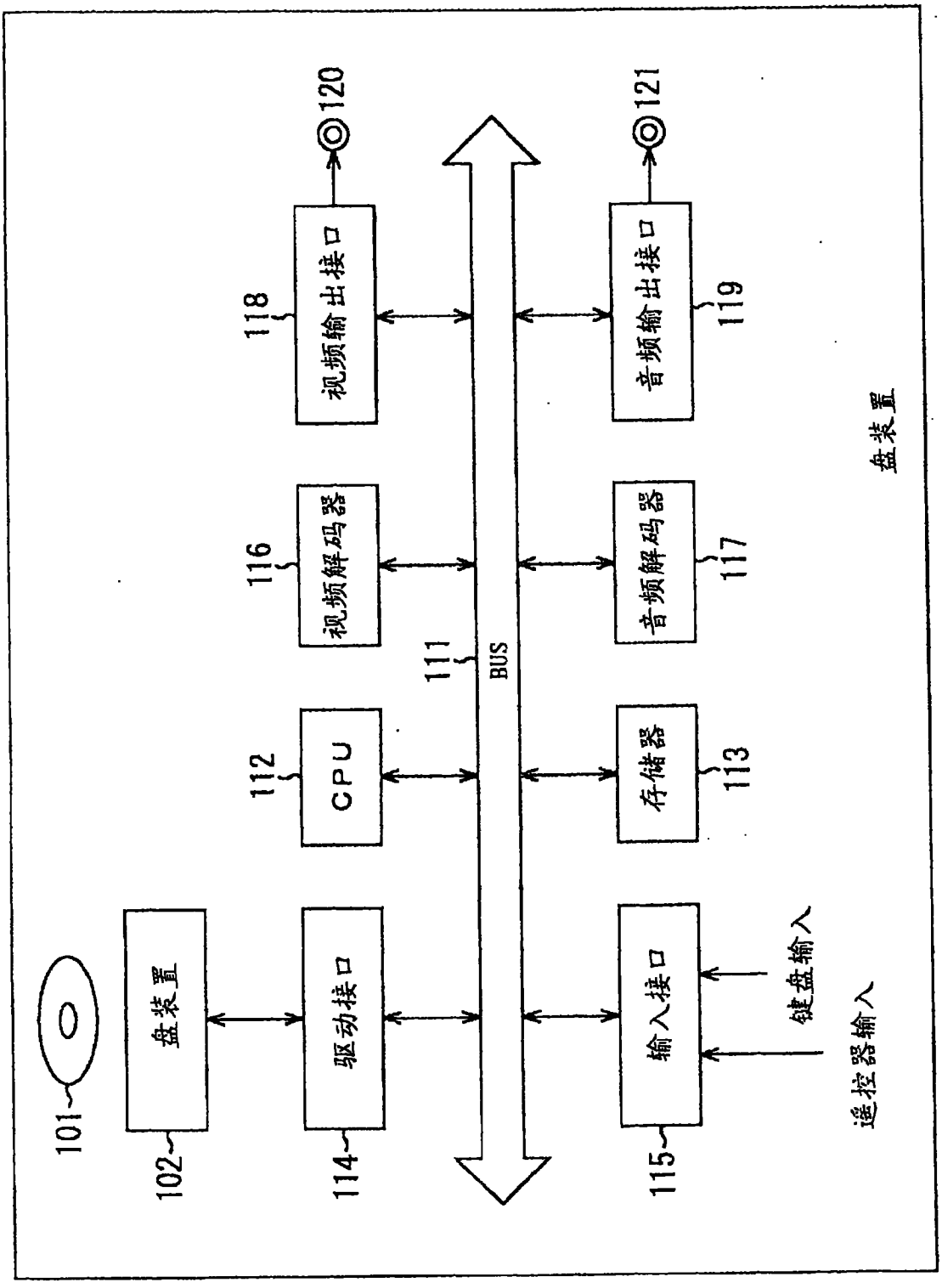

101
盘装置
102
驱动接口
114
CPU
112
视频解码器
116
视频输出接口
118
120
BUS
111
存储器
113
音频解码器
117
音频输出接口
119
121
输入接口
115
键盘输入
遥控器输入
盘装置

图 1

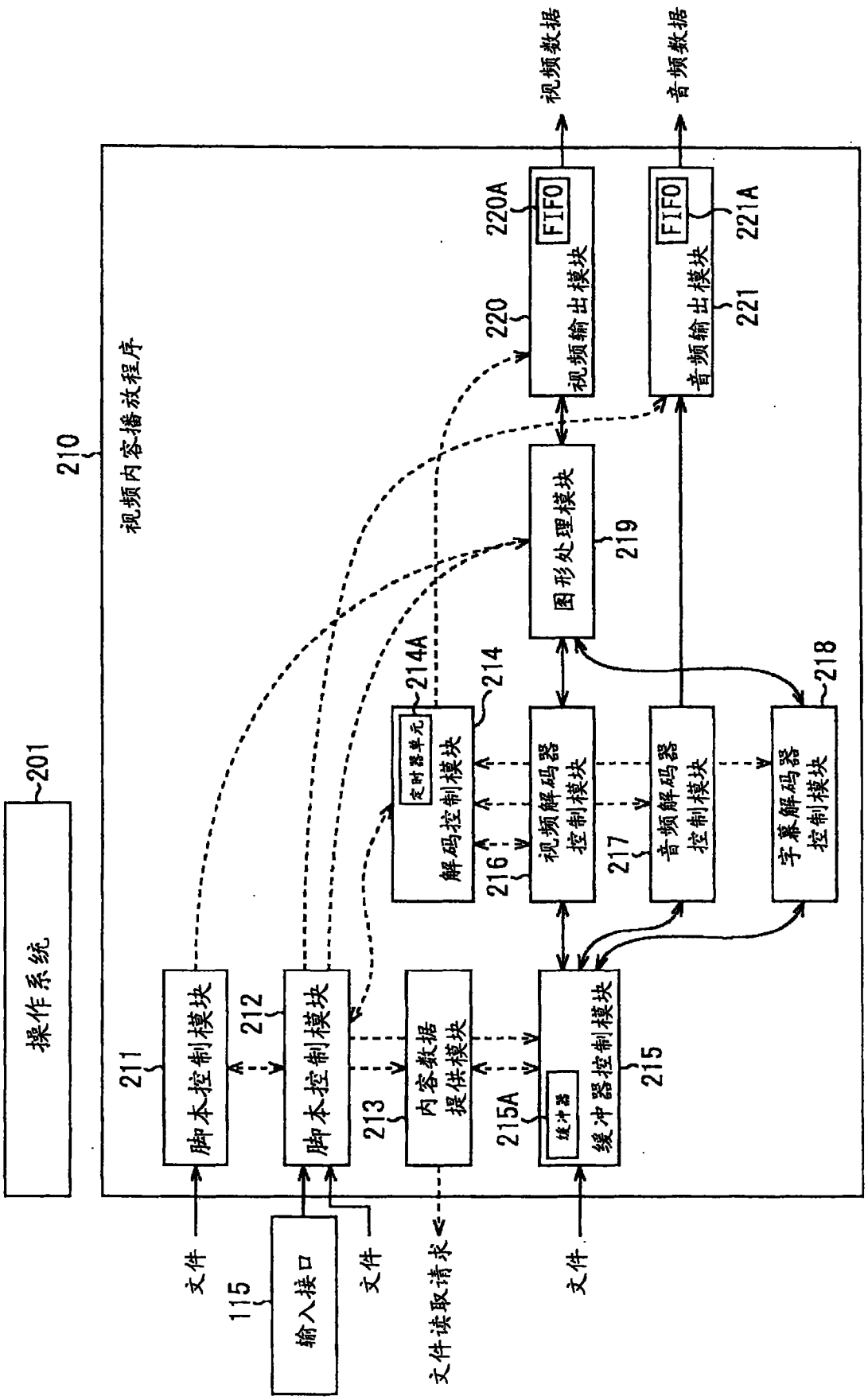

操作系统
201
视频内容播放程序
210
脚本控制模块
211
脚本控制模块
212
内容数据提供模块
213
缓冲器控制模块
215
缓冲器
215A
解码控制模块
216
定时器单元
214A
214
视频解码器控制模块
217
音频解码器控制模块
218
图形处理模块
219
视频输出模块
220
FIFO
220A
音频输出模块
221
FIFO
221A
视频数据
音频数据
输入接口
115
文件
文件
文件读取请求
文件

图 2

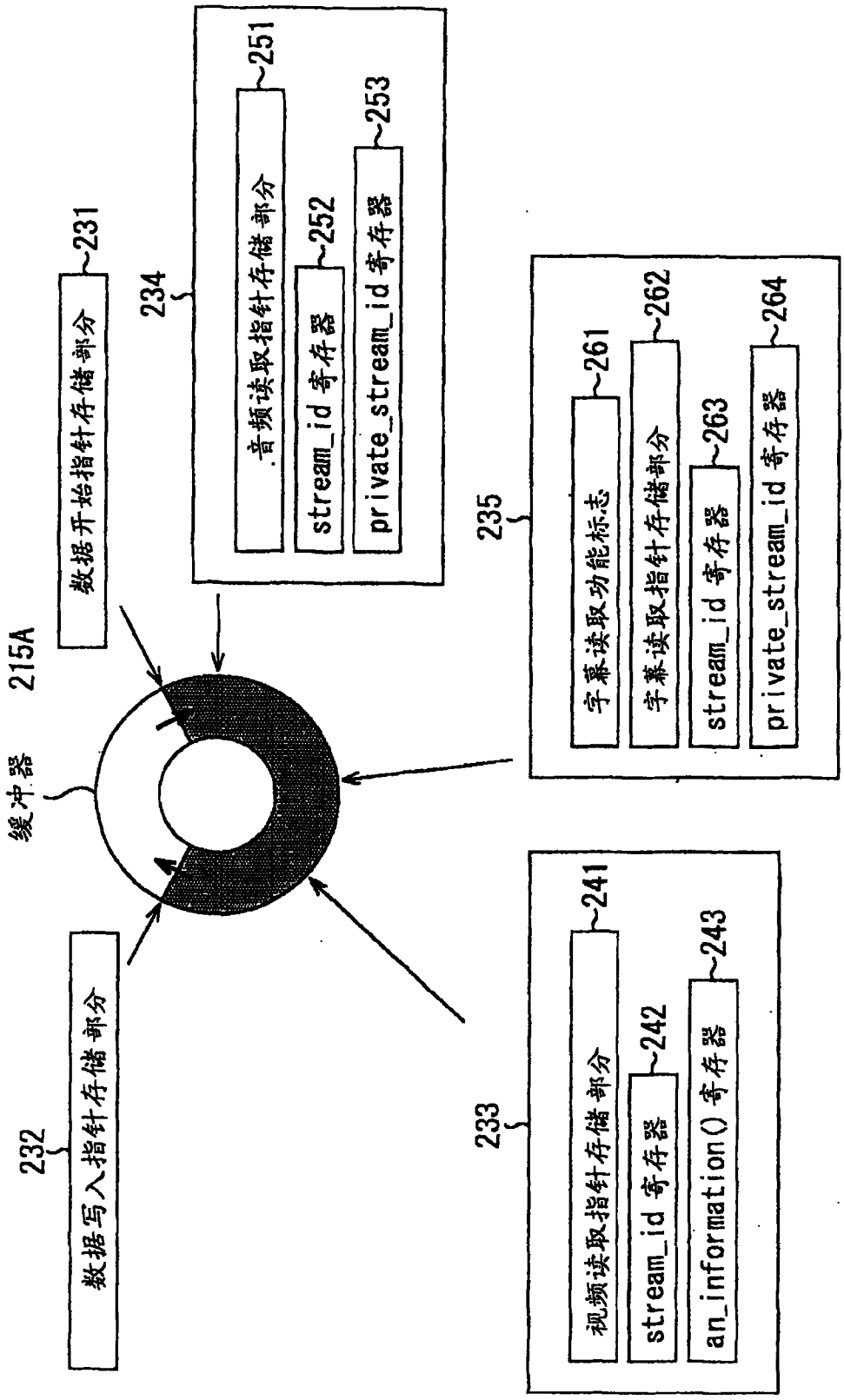

数据写入指针存储部分
232
数据开始指针存储部分
231
缓冲器
215A
234
音频读取指针存储部分
251
stream_id 寄存器
252
private_stream_id 寄存器
253
235
字幕读取功能标志
261
字幕读取指针存储部分
262
stream_id 寄存器
263
private_stream_id 寄存器
264
233
视频读取指针存储部分
241
stream_id 寄存器
242
an_information() 寄存器
243

图 3

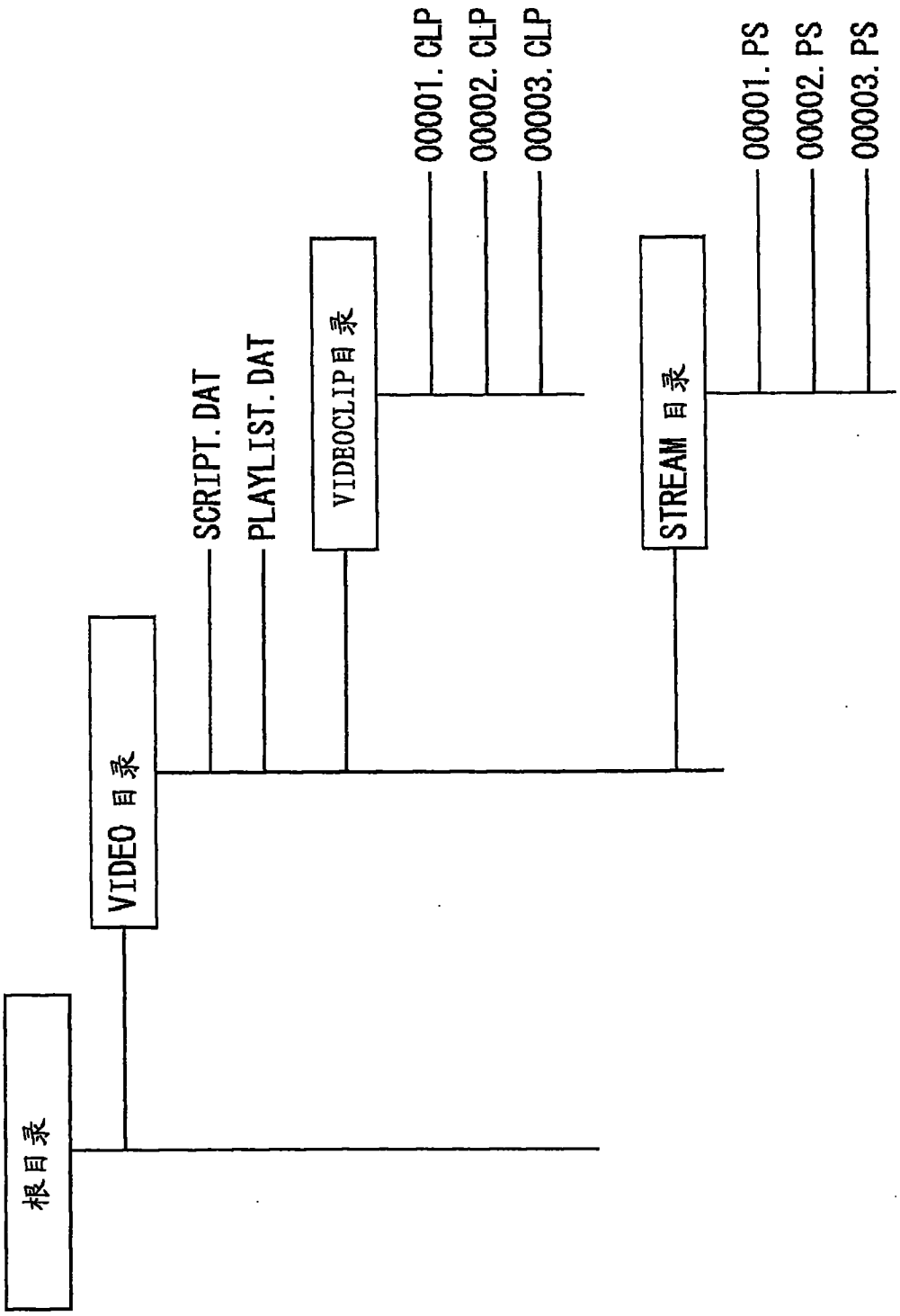

根目录
VIDEO 目录
SCRIPT.DAT
PLAYLIST.DAT
VIDEOCLIP目录
00001.CLP
00002.CLP
00003.CLP
STREAM 目录
00001.PS
00002.PS
00003.PS

图 4

PlayList.PAT

语法	位数	记忆符号
"PLAYLIST.文件"{		
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
//属性信息		
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

PlayItem()

语法	位数	记忆符号
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

PlayListMark()

语法	位数	记忆符号
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	保留
1	"Chapter" 标记
2	"Index" 标记
3	"Event" 标记
4～255	

图 8

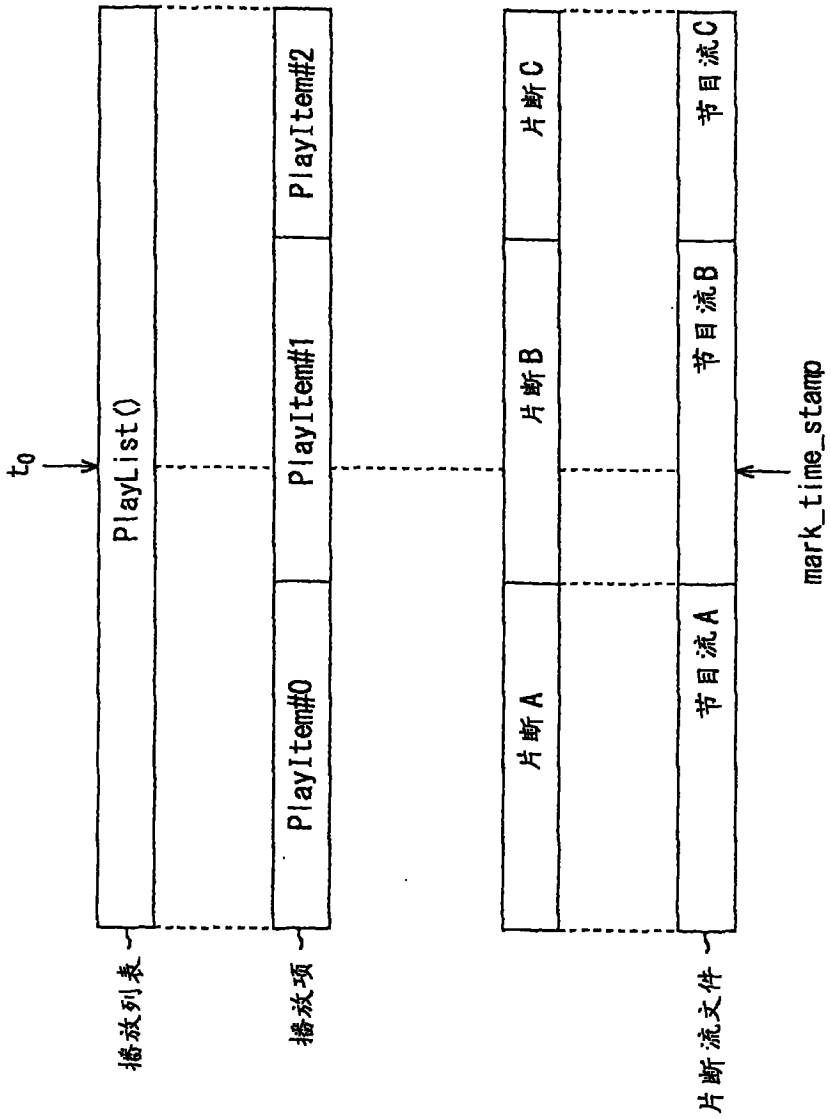

t0
PlayList()
PlayItem#0
PlayItem#1
PlayItem#2
片断A
片断B
片断C
节目流A
节目流B
节目流C
mark_time_stamp
播放列表
播放项
片断流文件

图 9

clip() "片断信息文件" (.CLP)

语法	位数	记忆符号
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

基本流的类型	stream_id	private_stream_id
视频	0xE0～0xEF	(なし)
ATRAC 音频	0xBD	0x00～0x0F
LPCM 音频	0xBD	0x10～0x1F
字幕	0xBD	0x80～0x9F

图 11

StaticInfo()

语法	位数	记忆符号
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

DynamicInfo

语法	位数	记忆符号
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

EP_map()

语法	位数	记忆符号
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	记忆符号

MPEG2_program_stream(){
do {
pack()
}while(nextbits()== pack_start_code)
MPEG_program_end_code
}
	32	bslbf

表 2-32—节目流包

语法	位数	记忆符号

pack(){
pack_header()
while(nextbits()==packet_start_code_prefix){
PES_packet()
}
}		

表 2-33—节目流包头

语法	位数	记忆符号
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 15

表 2-17—PES 信息包

语法	位数	记忆符号
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		
&& 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 16

表 2-17-PES 信息包（继续）

语法	位数	记忆符号
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
}		
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
}		
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 17

表 2-17—PES 信息包（完）

语法	位数	记忆符号
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
}		
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 18

表 2-18—Stream_id 分配

stream_id	记号	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 音频流号 x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 视频流号 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		保留的数据流
1111 1111	4	program_stream_directory

记号 X 意味着允许值 '0'和'1'两者并且号数相同的流类型。流号由所述 X 所取的值给出。

记号 1— program_stream_map类型的PES包具有 2.5.4.1 中定义的独特的语法

记号 2— private_stream_1和ISO/IEC_13552_stream 类型的PES 包依据和 ITU-T Rec. H.262 | ISO/IEC 13818-2视频和 ISO/IEC 13818-3音频流的语法那样的相同的PES包语法

记号 - 3 除在 PES_packet_length字段之后没有指定语法private_stream_2、ECM_stream和EMM_stream类型的PES包与private_stream_1类似

记号 4 - 类型program_stream directory的PES包具有2.5.5中定义的独特的语法

记号 5 - DSM-CC stream类型的PES包具有ISO/IEC 13818- 6.中定义的独特的语法

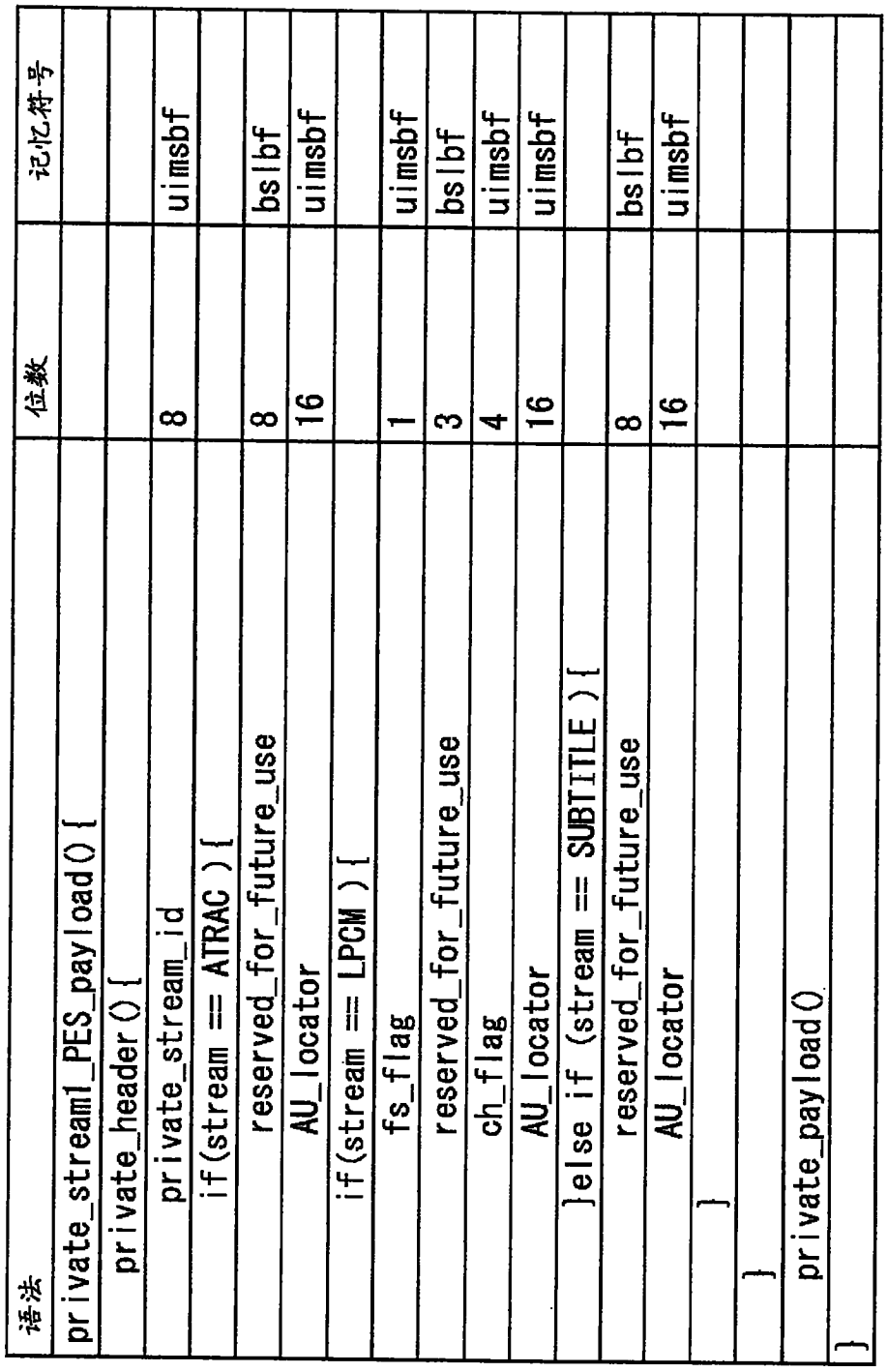
记号 6 - 在表 2-29 中，该 stream_id 与stream_type 0x09 相关联

记号 7 - 仅在 PES 包中使用该stream_id，其在传输流（参见2.4.3.7)中承载来自节目流或者 ISO/IEC 11172-1 系统流的数据

图 19

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	记忆符号
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

private_stream_id	流编码（流类型）
0000 xxxx 0001 xxxx 100x xxxx	ATRAC 音频流号 xxxx LPCM 视频流号 xxxx 字幕流号 x xxxx

图 22

语法	位数	记忆符号
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
}		
}		

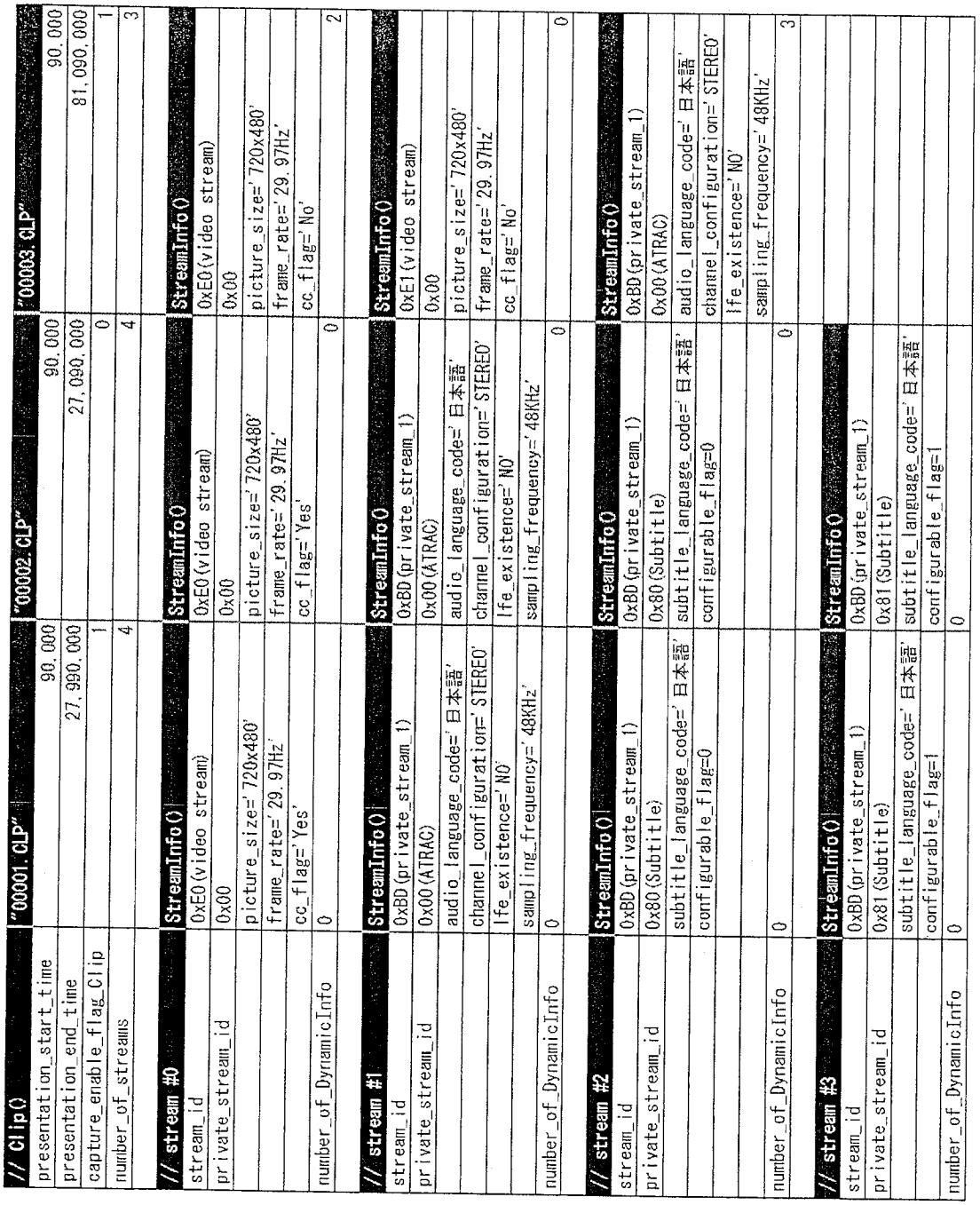
图 23

语法	位数	记忆符号
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()	PlayList #0	PlayList #1
capture_enable_flag_PlayList	1	0
number_of_PlayItems	2	1
// PlayItem #0		
Clip_Information_file_name	"00001.CLP"	"00003.CLP"
IN_time	180,090	90,000
OUT_time	27,180,090	81,090,000
// PlayItem #1		
Clip_Information_file_name	"00002.CLP"	
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'	picture_size='720x480'	picture_size='720x480'
	frame_rate='29.97Hz'	frame_rate='29.97Hz'	frame_rate='29.97Hz'
	cc_flag='Yes'	cc_flag='Yes'	cc_flag='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='日本語'	audio_language_code='日本語'	picture_size='720x480'
	channel_configuration='STEREO'	channel_configuration='STEREO'	frame_rate='29.97Hz'
	lfe_existence='NO'	lfe_existence='NO'	cc_flag='No'
	sampling_frequency='48KHz'	sampling_frequency='48KHz'	
number_of_DynamicInfo	0	0	0
// 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='日本語'	subtitle_language_code='日本語'	audio_language_code='日本語'
	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='日本語'	subtitle_language_code='日本語'	
	configurable_flag=1	configurable_flag=1	
number_of_DynamicInfo	0	0	

图 26

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

PlayList #0 PlayListMark()

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		0	180,090	0	0	1	开始 开始
#1		0	5,580,090	0	0	1	1 MIN 1 MIN
#2		0	10,980,090	0	0	2	2 MIN 2 MIN
#3		0	16,380,090	0	0	0	3 MIN 3 MIN
#4	Chpater	1	90,000	0	0	2	头 5 MIN
#5		1	5,490,000	0	0	1	1 MIN 6 MIN
#6		1	10,890,000	0	0	2	2 MIN 7 MIN

PlayList #1ØPlayListMark()

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		0	90,000	0	0	0	头
#1		0	27,090,000	0xE0	0	1	5 MIN
#2		0	27,540,000	0xE1	0	2	5 MIN 5 SEC (305 MIN)

图 28

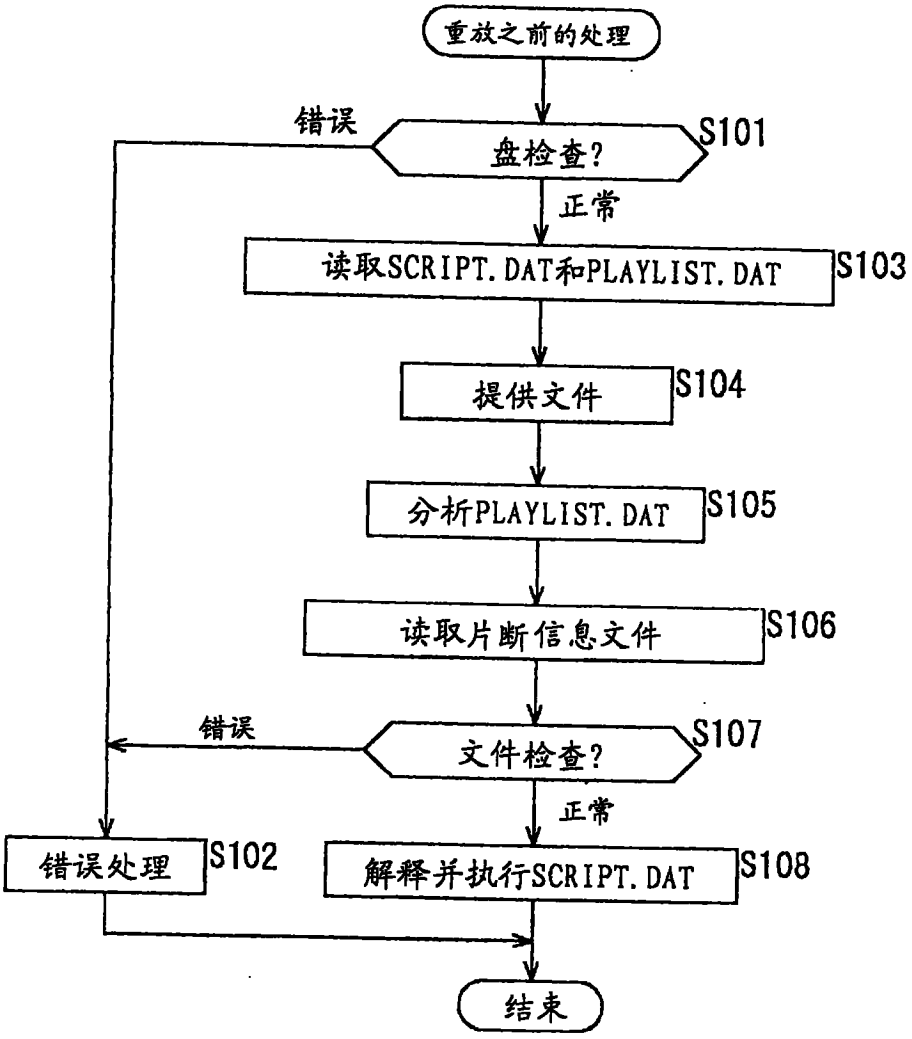

重放之前的处理
盘检查?
S101
错误
正常
读取SCRIPT.DAT和PLAYLIST.DAT
S103
提供文件
S104
分析PLAYLIST.DAT
S105
读取片断信息文件
S106
文件检查?
S107
错误
正常
错误处理
S102
解释并执行SCRIPT.DAT
S108
结束

图 29

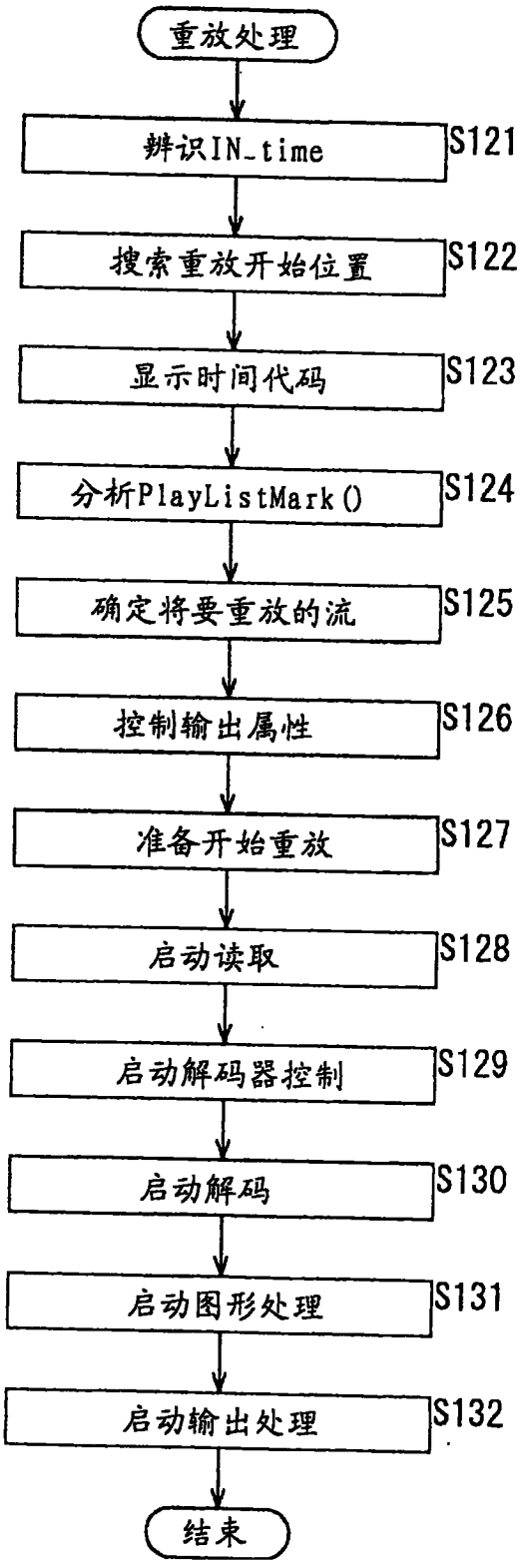

重放处理
辨识IN_time
S121
搜索重放开始位置
S122
显示时间代码
S123
分析PlayListMark()
S124
确定将要重放的流
S125
控制输出属性
S126
准备开始重放
S127
启动读取
S128
启动解码器控制
S129
启动解码
S130
启动图形处理
S131
启动输出处理
S132
结束

图 30

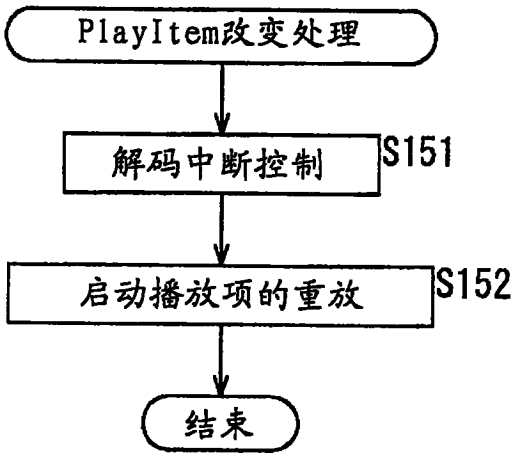

PlayItem改变处理
解码中断控制
S151
启动播放项的重放
S152
结束

图 31

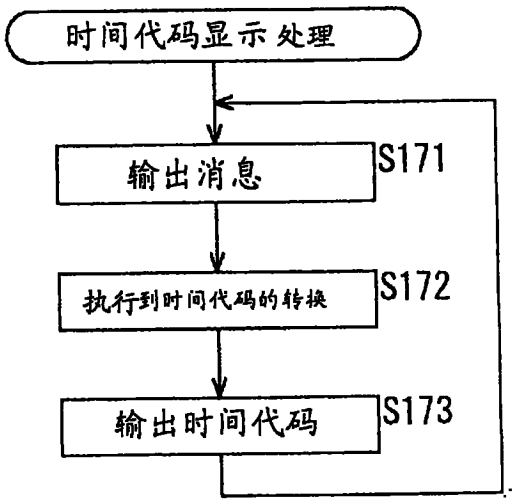

时间代码显示处理
输出消息
S171
执行到时间代码的转换
S172
输出时间代码
S173

图 32

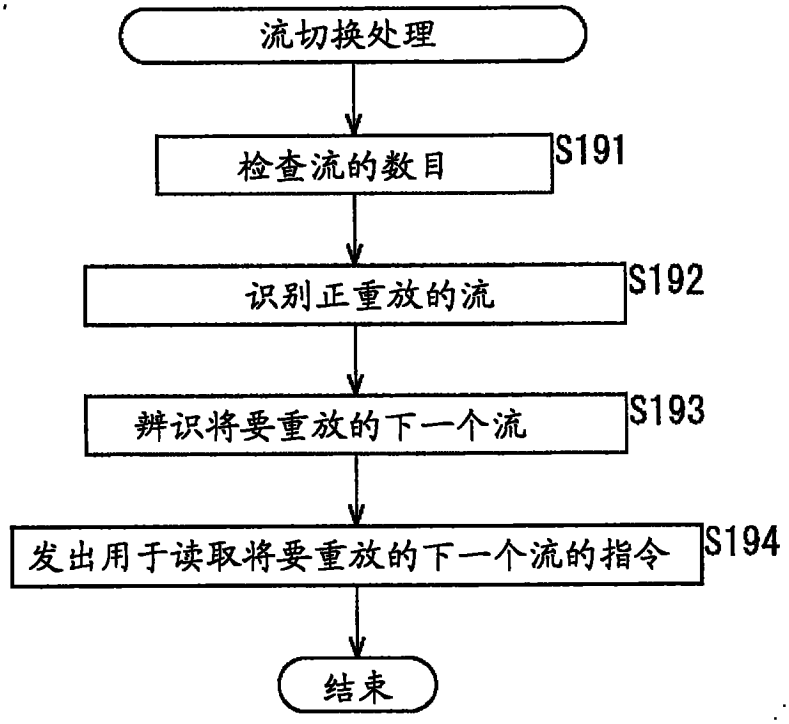

流切换处理
检查流的数目
S191
识别正重放的流
S192
辨识将要重放的下一个流
S193
发出用于读取将要重放的下一个流的指令
S194
结束

图 33

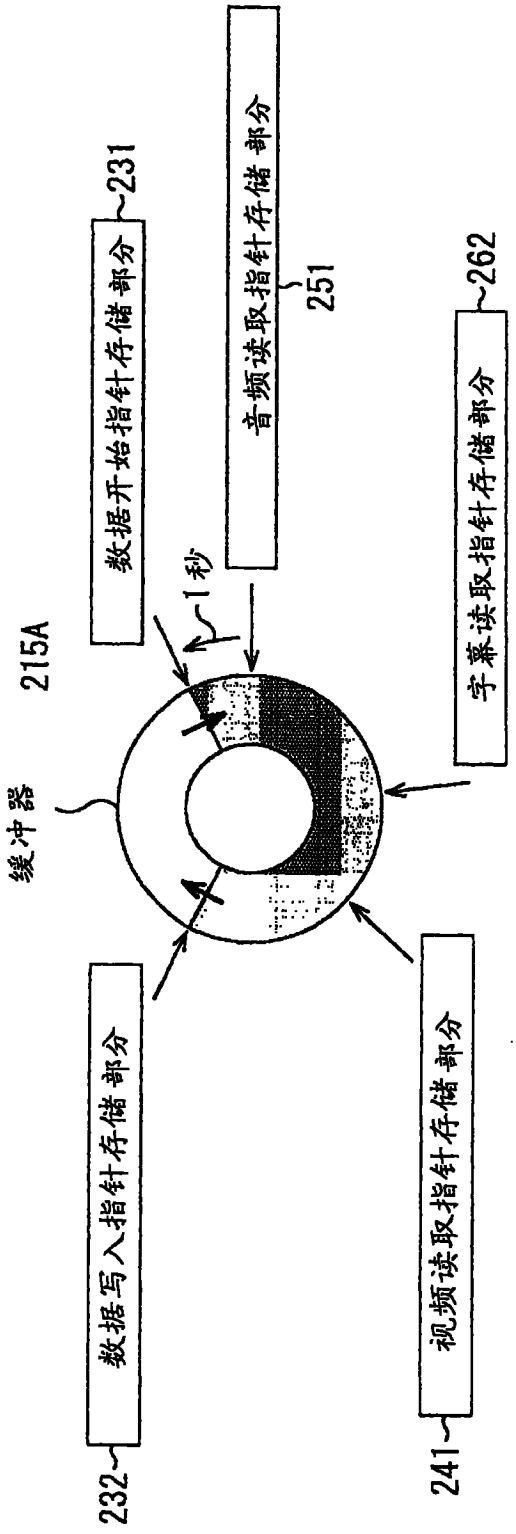

数据开始指针存储部分
231
音频读取指针存储部分
251
1秒
字幕读取指针存储部分
262
215A
缓冲器
数据写入指针存储部分
232
视频读取指针存储部分
241

图 34

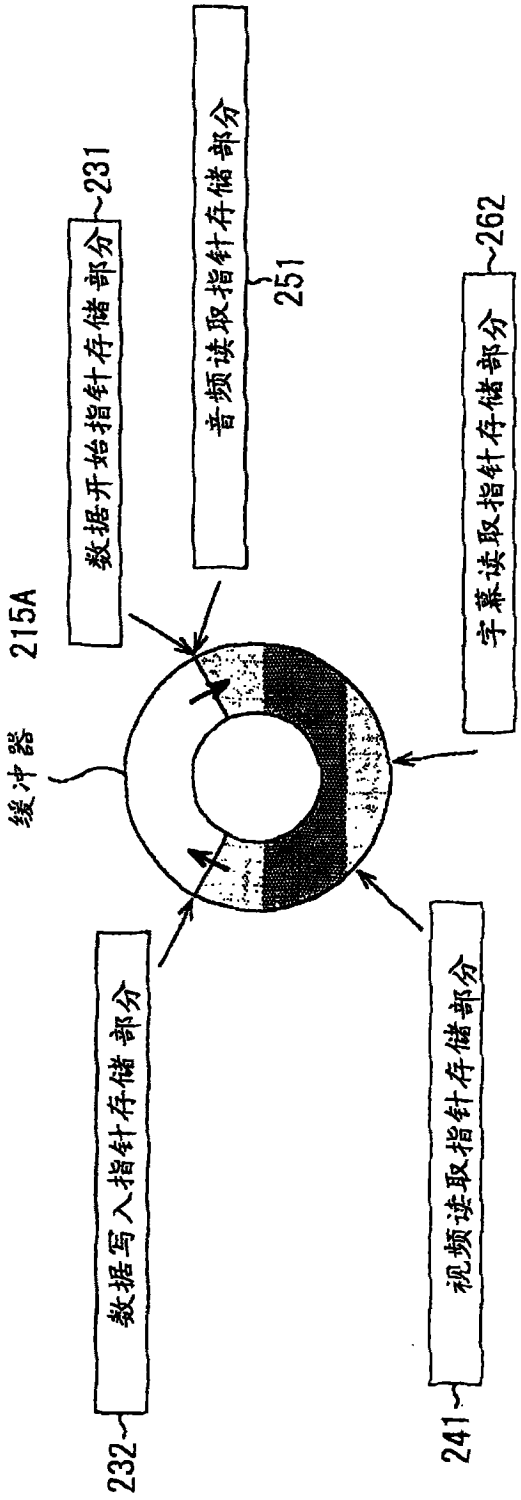

数据开始指针存储部分
231
音频读取指针存储部分
251
字幕读取指针存储部分
262
缓冲器
215A
数据写入指针存储部分
232
视频读取指针存储部分
241

图 35

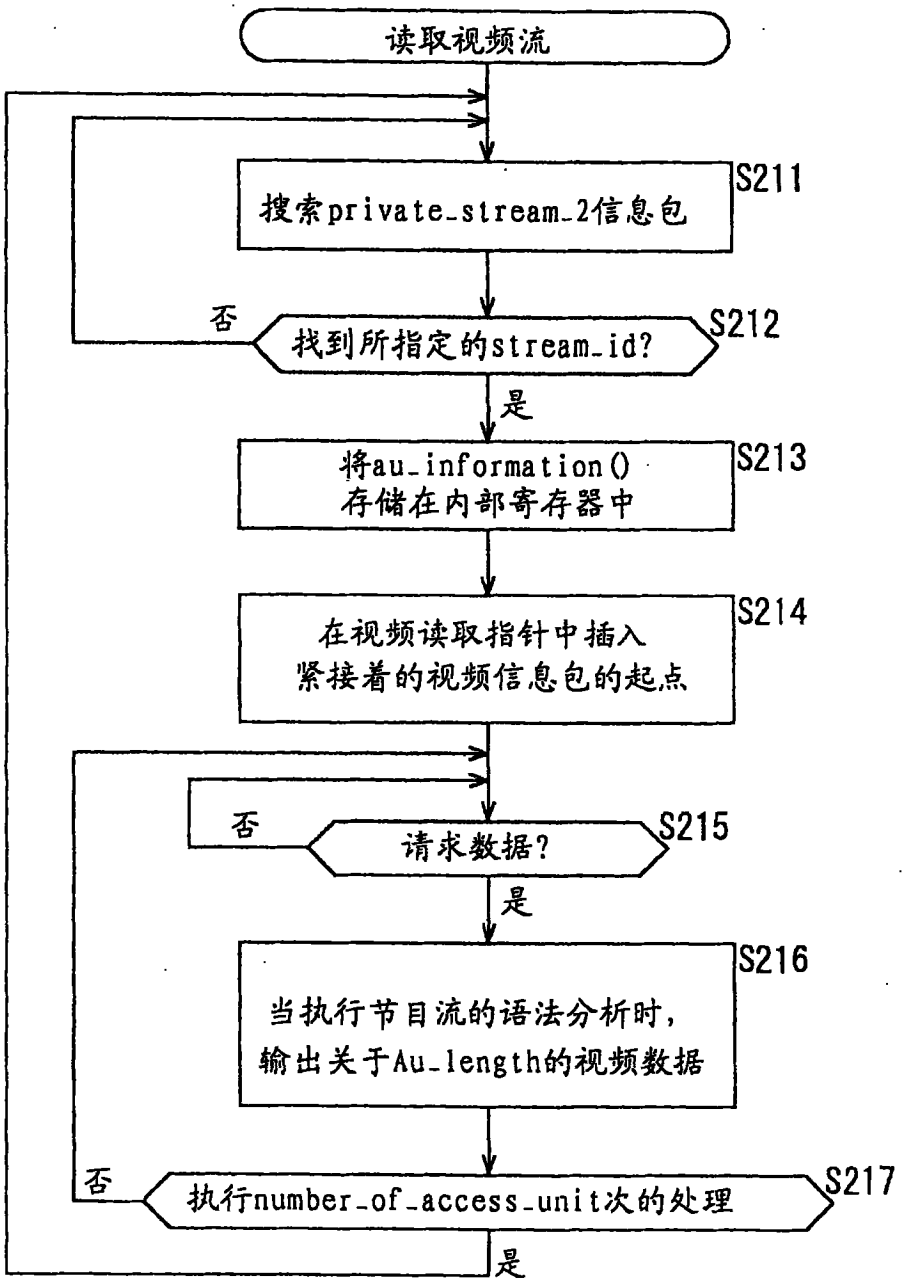

读取视频流
搜索private_stream_2信息包
S211
否
找到所指定的stream_id?
S212
是
将au_information()
存储在内部寄存器中
S213
在视频读取指针中插入
紧接着的视频信息包的起点
S214
否
请求数据?
S215
是
当执行节目流的语法分析时，
输出关于Au_length的视频数据
S216
否
执行number_of_access_unit次的处理
S217
是

图 36

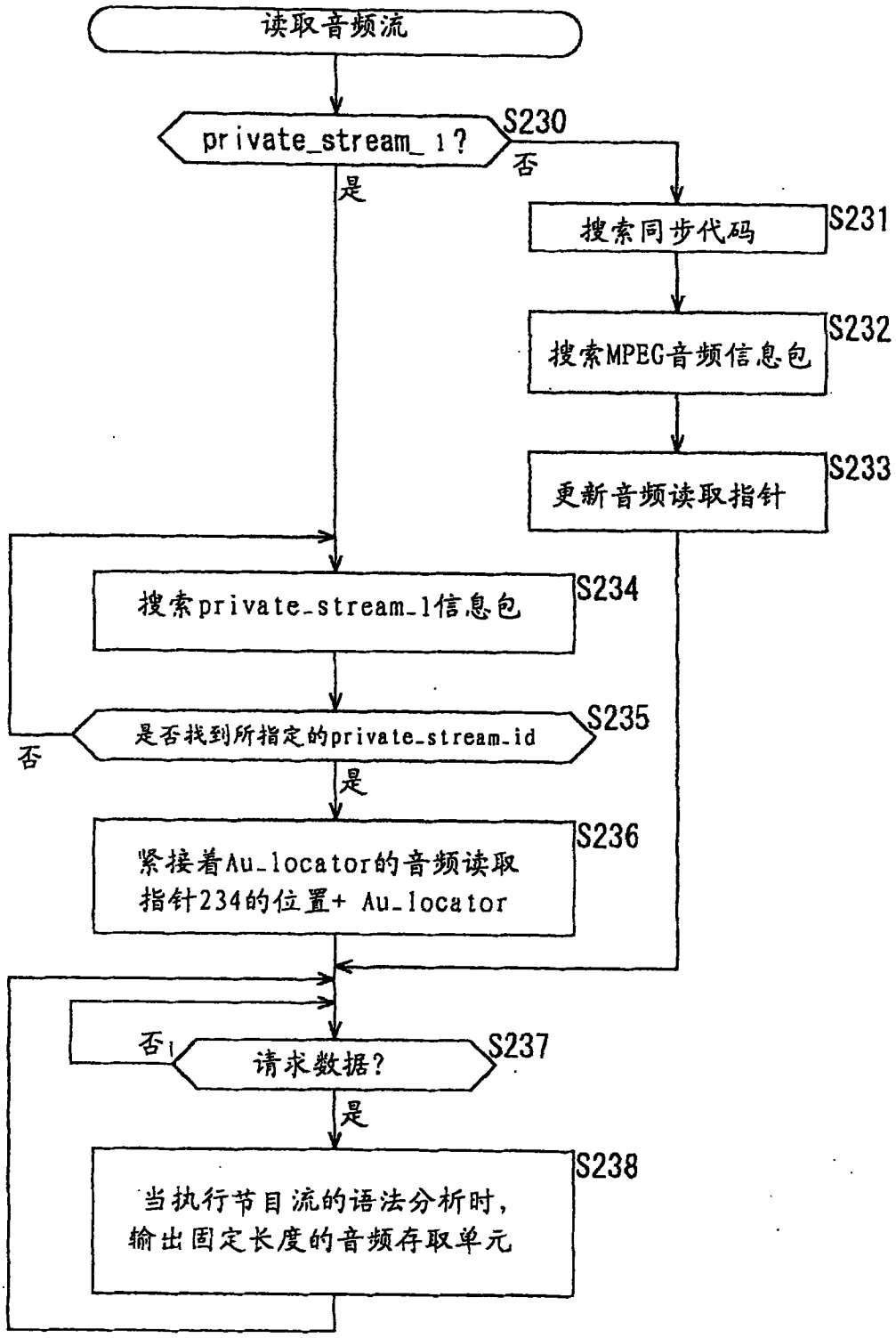

读取音频流
private_stream_1?
S230
否
是
搜索同步代码
S231
搜索MPEG音频信息包
S232
更新音频读取指针
S233
搜索private_stream_1信息包
S234
是否找到所指定的private_stream_id
S235
否
是
紧接着Au_locator的音频读取
指针234的位置+ Au_locator
S236
否
请求数据?
S237
是
当执行节目流的语法分析时，
输出固定长度的音频存取单元
S238

图 37

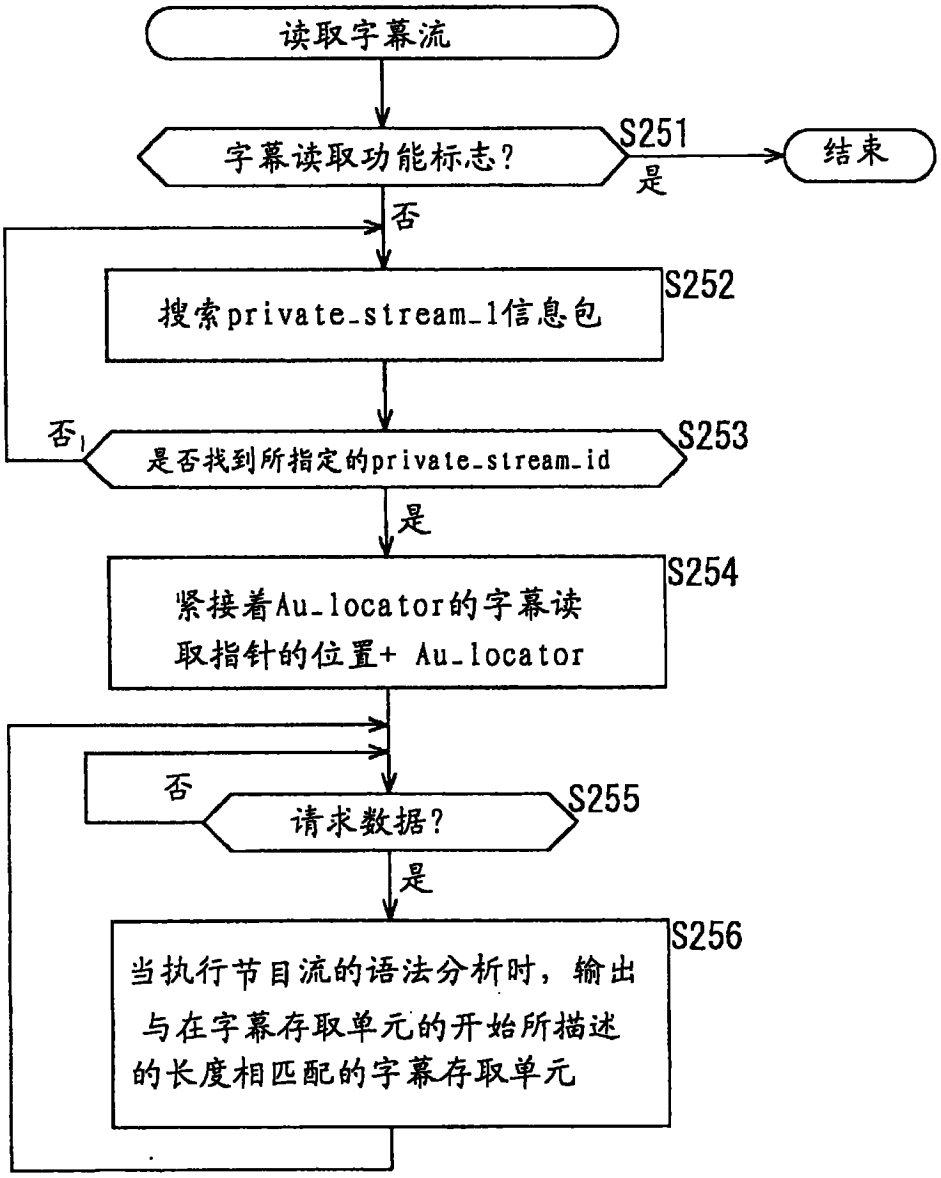

读取字幕流
字幕读取功能标志?
S251
是
结束
否
搜索private_stream_1信息包
S252
否
是否找到所指定的private_stream_id
S253
是
紧接着Au_locator的字幕读取指针的位置+ Au_locator
S254
否
请求数据?
S255
是
当执行节目流的语法分析时，输出与在字幕存取单元的开始所描述的长度相匹配的字幕存取单元
S256

图 38

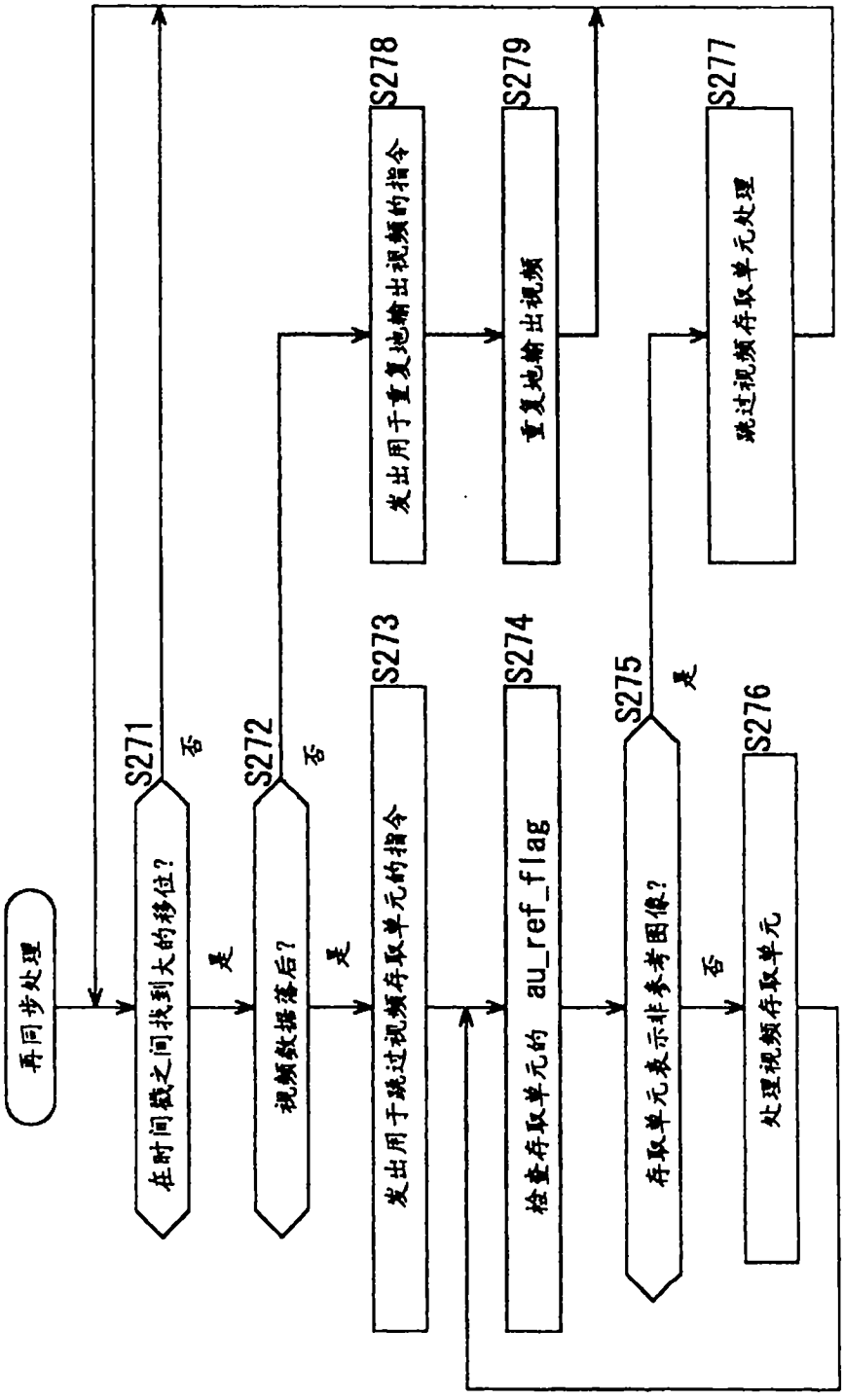

再同步处理
S271
在时间戳之间找到大的移位?
否
是
S272
视频数据落后?
否
是
S273
发出用于跳过视频存取单元的指令
S274
检查存取单元的 au_ref_flag
S275
存取单元表示非参考图像?
是
否
S276
处理视频存取单元
S277
跳过视频存取单元处理
S278
发出用于重复地输出视频的指令
S279
重复地输出视频

图 39

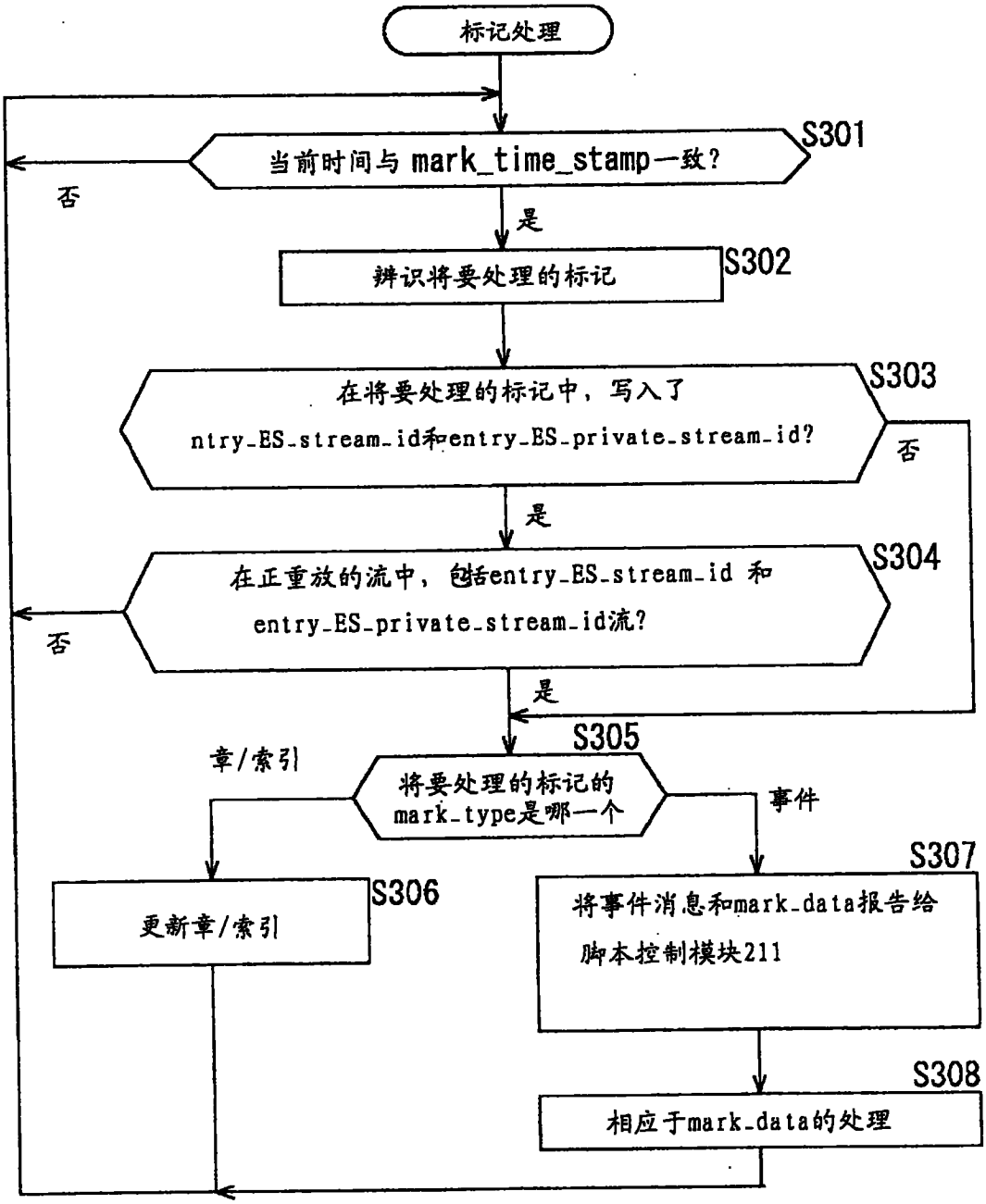

标记处理
S301
当前时间与 mark_time_stamp一致?
否
是
S302
辨识将要处理的标记
S303
在将要处理的标记中，写入了
ntry_ES_stream_id和entry_ES_private_stream_id?
否
是
S304
在正重放的流中，包括entry_ES_stream_id 和
entry_ES_private_stream_id流?
否
是
S305
章/索引
将要处理的标记的
mark_type是哪一个
事件
S306
更新章/索引
S307
将事件消息和mark_data报告给
脚本控制模块211
S308
相应于mark_data的处理

图 40

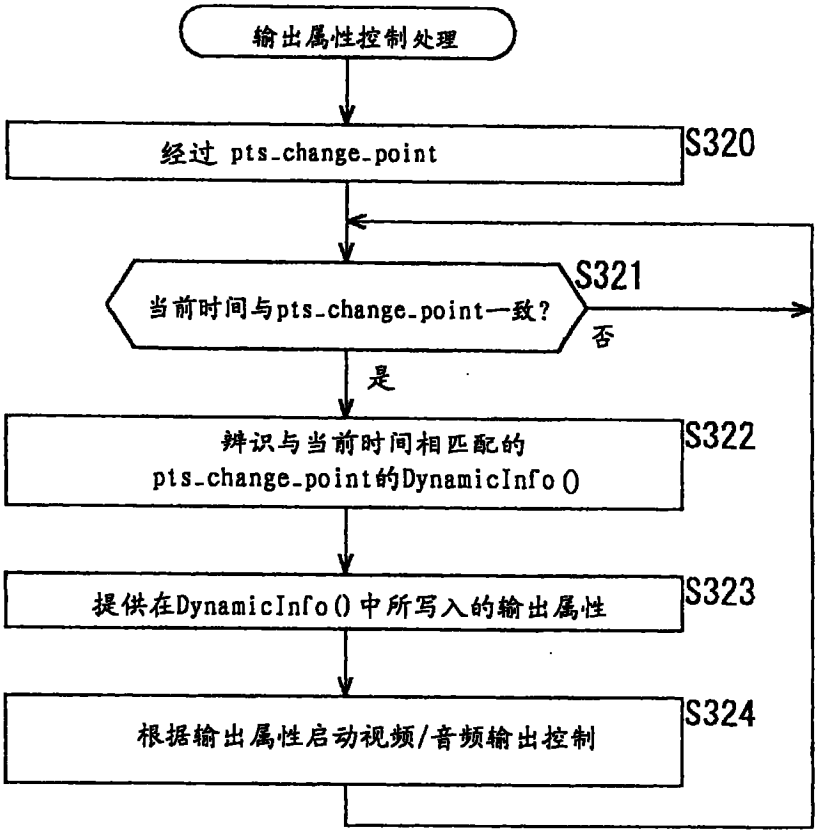

输出属性控制处理
经过 pts_change_point
S320
当前时间与pts_change_point一致?
S321
否
是
辨识与当前时间相匹配的
pts_change_point的DynamicInfo()
S322
提供在DynamicInfo()中所写入的输出属性
S323
根据输出属性启动视频/音频输出控制
S324

图 41

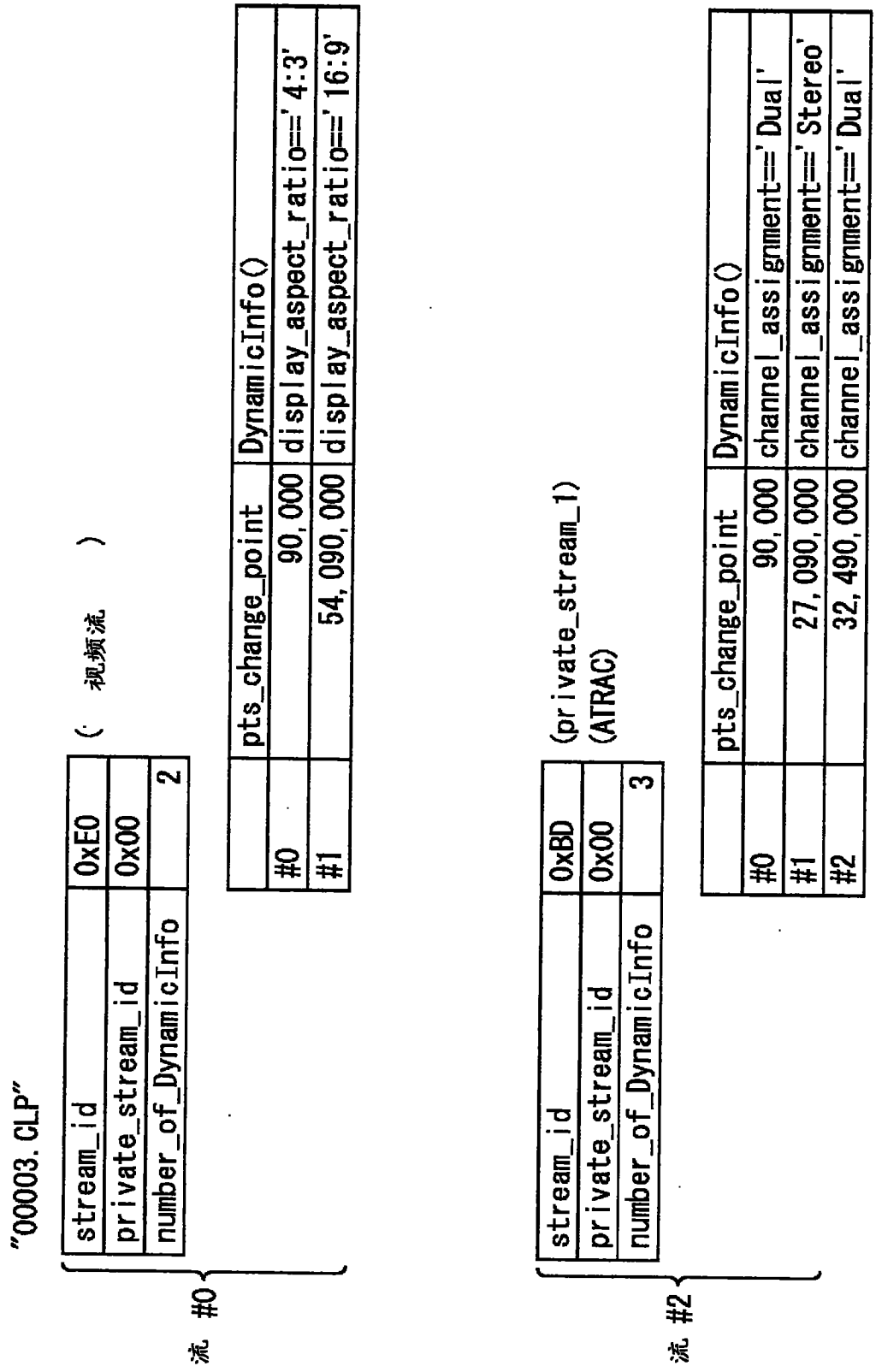

"00003.CLP"
流 #0
stream_id 0xE0 (视频流)
private_stream_id 0x00
number_of_DynamicInfo 2
pts_change_point DynamicInfo()
#0 90,000 display_aspect_ratio=='4:3'
#1 54,090,000 display_aspect_ratio=='16:9'
流 #2
stream_id 0xBD (private_stream_1)
private_stream_id 0x00 (ATRAC)
number_of_DynamicInfo 3
pts_change_point DynamicInfo()
#0 90,000 channel_assignment=='Dual'
#1 27,090,000 channel_assignment=='Stereo'
#2 32,490,000 channel_assignment=='Dual'

图 42

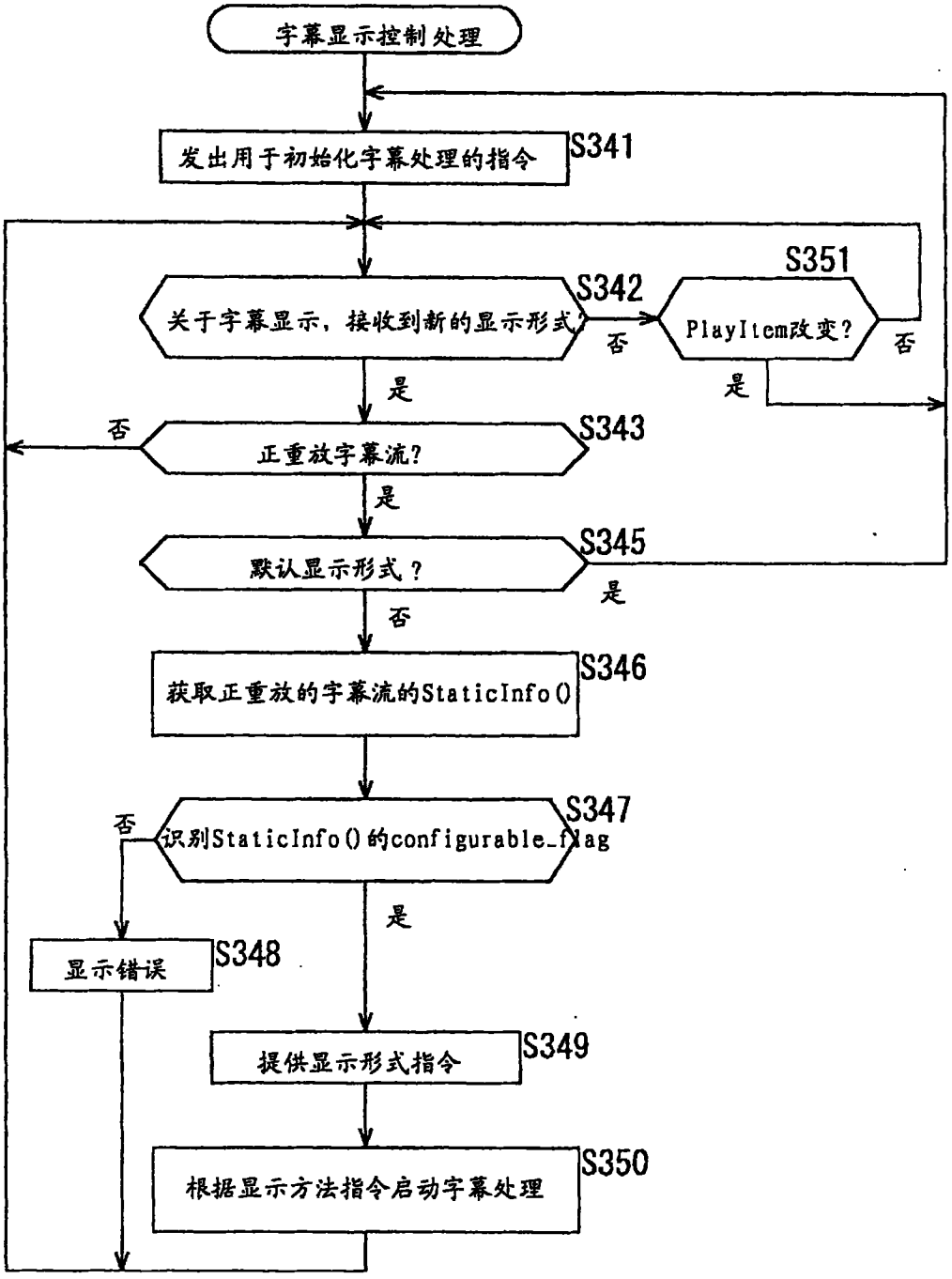

字幕显示控制处理
发出用于初始化字幕处理的指令
S341
关于字幕显示，接收到新的显示形式？
S342
否
PlayItem改变？
S351
否
是
是
否
正重放字幕流？
S343
是
默认显示形式？
S345
是
否
获取正重放的字幕流的StaticInfo()
S346
否
识别StaticInfo()的configurable_flag
S347
是
显示错误
S348
提供显示形式指令
S349
根据显示方法指令启动字幕处理
S350

图 43

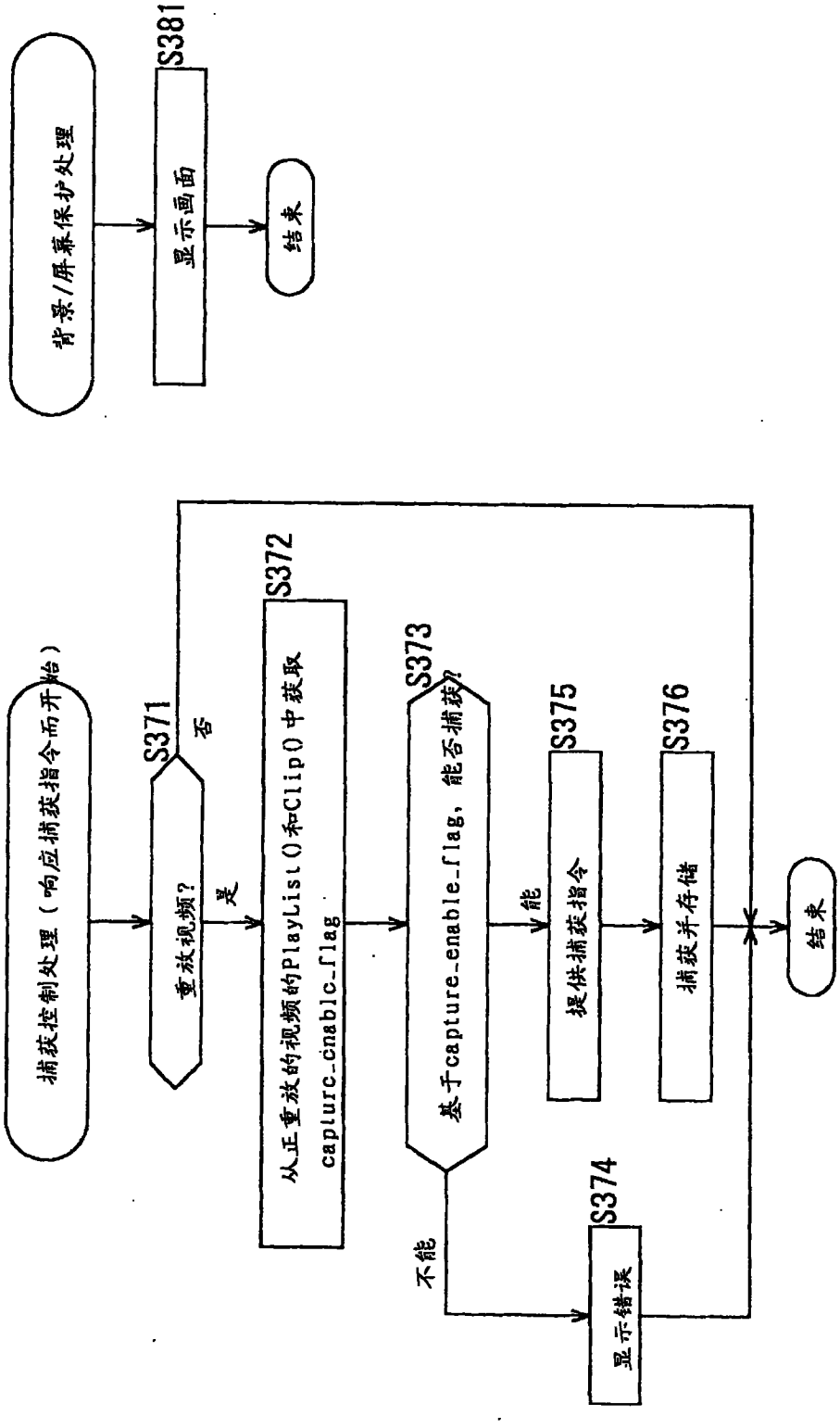

背景/屏幕保护处理
S381
显示画面
结束
捕获控制处理（响应捕获指令而开始）
S371
重放视频?
否
是
S372
从正重放的视频的PlayList()和Clip()中获取
capture_enable_flag
S373
基于capture_enable_flag，能否捕获?
能
S375
提供捕获指令
S376
捕获并存储
结束
不能
S374
显示错误

图 44

语法	位数	记忆符号
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	记忆符号
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