

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G11B 20/10 (2006.01)

H04N 5/91 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610094006.5

[45] 授权公告日 2009 年 9 月 16 日

[11] 授权公告号 CN 100541636C

[22] 申请日 1999.4.30

[21] 申请号 200610094006.5

分案原申请号 99105291.9

[30] 优先权

[32] 1998.4.30 [33] JP [31] 120393/1998

[73] 专利权人 索尼株式会社

地址 日本东京

[72] 发明人 藤波靖 浜田俊也

[56] 参考文献

WO9713365A1 1997.4.10

US6009433A 1999.12.28

CN1119804A 1998.2.11

审查员 刘羽楠

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

代理人 李镇江

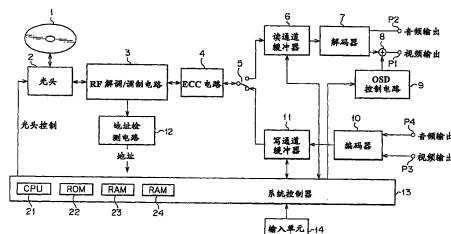
权利要求书 1 页 说明书 36 页 附图 30 页

[54] 发明名称

记录设备

[57] 摘要

为了保证设备的兼容性，当能够在与先前播放项连续的无缝再现中再现相继的播放项时，将无缝连接标记设置为 1，且当不能保证无缝再现时，将其设置为 0。



1. 一种记录设备，包括：

记录装置，用于记录管理数据和内容数据到记录介质；

其中所述管理数据具有：播放清单，其指示具有一个或多个播放项的播放项重放序列，所述一个或多个播放项指示到所述内容数据的入口点和出口点；以及关于相应于每一所述播放项的无缝重放的状态信息；

其中所述状态信息指示与前一播放项的无缝重放是不能得到保证、未知还是能得到保证，

确定装置，用于通过确定缓存了将要被再现的播放项的缓存器在直到相继的播放项被再现的时间段内是否出现空白，来确定所述播放项之间的所述内容数据的无缝重放是否可行，其中所述状态信息是基于所述确定装置的结果记录的。

2. 根据权利要求1的记录设备，包括：

部分记录装置，用于如果所述确定装置确定出无缝重放不可能的话重新记录所述内容数据流的一部分，以使所述内容数据的无缝重放可行。

记录设备

本申请是申请号为99105291.9、申请日为1999年4月30日、发明名称为“记录/再现设备和方法以及程序提供介质”的专利申请的方案申请。

技术领域

本发明涉及一种记录/再现设备和方法以及一种程序提供介质，特别是一种对用户错误地识别为与连续的数据再现有关的故障进行限制的记录/再现设备和方法以及一种程序提供介质。

背景技术

有一种情况，其中视频数据或音频数据不仅被连续地记录而且还间歇地记录在盘中。进一步地，有一种情况，其中已经被记录的数据被擦除且其他的数据被覆写。

有一种情况，其中当这种擦除或覆写处理被反复执行时，将要被连续再现的数据不一定被记录在盘上的相继位置，而是被记录在盘上彼此分离的位置。在这种情况下，当设备的再现中缓存器的容量不足时，有一种情况—其中数据不能被连续地再现且再现数据根据记录位置而变得暂时地不足。

另外，是否产生了不连续的部分，受到设备的缓存器容量的影响，因而设备之间的兼容性不能得到保证，且在最差的情况下，用户可能错误地认为设备发生了故障。

发明内容

本发明就是考虑到这种情况而作出的，且本发明的一个目的是保证设备的兼容性并限制用户错误地认为设备故障的情况。

根据本发明的一个方面，提供了一种记录设备，包括：用于把数据记录到记录介质上的记录装置；判定装置，用于判定数据是否能够与已经记录的数据连续地记录；以及，执行装置，用于根据判定装置

的判定结果就数据的连续再现进行处理。

根据本发明的另一个方面，提供了记录设备的一种记录方法，该记录方法包括：记录步骤，用于把数据记录到记录介质上；判定步骤，用于当数据被记录到记录介质上时判定数据是否能够与已经记录的数据连续地被再现；以及，执行步骤，用于根据判定步骤的判定结果就数据的连续再现执行处理。

根据本发明的另一个方面，提供了一种向记录设备提供用于执行一种处理的程序以把数据记录到记录介质上的程序提供介质，该处理包括：记录步骤，用于把数据记录到记录介质上；判定步骤，用于在数据被记录到记录介质上时判定数据是否能够与已经记录的数据连续地得到再现；以及，执行步骤，用于根据判定步骤的判定结果执行与数据的连续再现有关的处理。

根据本发明的另一个方面，提供了一种再现设备，包括：再现装置，用于再现记录在记录介质上的数据；提取装置，用于提取表示是否能够从再现装置再现的数据执行数据的连续再现的连续再现信息；以及，相加装置，用于根据提取装置提取的连续再现信息把一个间隙加到再现装置再现的数据上。

根据本发明的另一个方面，提供了一种再现方法，包括：再现步骤，用于再现记录在记录介质上的数据；提取步骤，用于提取表示是否能够从在再现步骤再现的数据提取数据的连续再现的连续再现信息；以及，相加步骤，用于根据在提取步骤提取的连续再现信息把一个间隙加到再现数据上。

根据本发明的另一个方面，提供了用于向一种再现设备提供执行一种处理的程序的程序提供介质，该处理包括：再现步骤，用于再现记录在记录介质上的数据；提取步骤，用于提取表示是否能够从在再现步骤再现的数据执行连续再现的连续再现信息；以及，相加步骤，用于根据在提取步骤提取的连续再现信息把一个间隙加到再现数据上。

根据本发明的特征所说明的记录设备、记录方法和程序提供介质，按照数据是否能够与已经记录的数据连续地得到再现的判定结果，执行有关连续再现的处理。

根据本发明的特征所限定的再现设备、再现方法和程序提供介质，按照从记录介质再现的连续再现信息，间隙被加到数据上。

附图说明

图1用于说明一种目录的格式；

图2用于说明VOLUM.TOC；

图3用于说明volume_information()(卷信息)；

图4用于说明volume_attribute()(卷属性)；

图5用于说明resume()(恢复)；

图6用于说明volume_rating()(卷评级)；

图7用于说明write_protect()(写保护)；

图8用于说明play_protect()(播放保护)；

图9用于说明recording_timer()(记录定时器)；

图10用于说明text_block()(文本块)；

图11用于说明language_set()(语言设定)；

图12用于说明text_item()(文本项)；

图13用于说明ALBUM.STR；

图14用于说明album()(专辑)；

图15用于说明TITLE_###.VDR；

图16用于说明title_info()(标题信息)；

图17用于说明PROGRAM_\$\$\$PGI；

图18用于说明program()(节目)；

图19用于说明play_list()(播放清单)；

图20用于说明play_item()(播放项)；

图21用于说明CHUNKGROUP_###.CGIT；

图22用于说明chunk_connection_info()(数据块连接信息)；

图23用于说明chunk_arrangement_info()(数据块排列信息)；

图24用于说明CHUNK_%%%.ABST；

图25是显示了本发明所应用于的光盘设备的结构例子的框图；

图26用于说明目录的结构；

图27用于说明目录的逻辑结构;

图28用于说明偏移;

图29用于说明目录的结构;

图30用于说明目录的结构;

图31用于说明目录的逻辑结构;

图32用于说明目录的结构;

图33用于说明目录的结构;

图34用于说明目录的逻辑结构;

图35用于说明目录的逻辑结构;

图36用于说明file_type_id(文件类型id);

图37用于说明mark_type(标志类型);

图38用于说明file_type_id(文件类型id);

图39用于说明chunk_sync_play_flag(数据块同步播放标记);

图40用于说明original_time_count_flag(原始时间计数标记);

图41用于说明file_type_id(文件类型id);

图42用于说明info_type(信息类型);

图43用于说明slot_unit_type(片单位类型);

图44用于说明file_type_id(文件类型id);

图45用于说明program_status(节目状态);

图46用于说明seamless_connection_flag(无缝连接标记);

图47用于说明seamless_connection_flag(无缝连接标记)的含意;

图48是用于说明标题的非连续点标记记录处理的流程图;

图49是用于说明节目形成中的非连续点标记的记录处理的流程图;

图50是说明再现标题中的非连续点标记的处理的流程图;

图51是说明再现节目中处理一个非连续点标记的流程图。

具体实施方式

下面描述根据本发明的实施例。为了澄清在以下的实施例中说明的本发明的各个装置之间的相应的关系, 以下将通过把相应的实施例

(只是例子)加在相应装置之后的括号中,来描述本发明的特征。然而,自然地,这种描述并不意味着相应的装置只限于所述的实施例。

本发明提供一种记录设备,包括:记录装置,用于把数据记录到记录介质上;判定装置,用于当数据被记录到记录介质上时判定数据是否能够连续地得到再现;以及执行装置,用于根据判定装置的判定结果执行有关数据的连续再现的处理。

其中执行装置在判定装置判定不能进行连续再现时改变数据的记录位置。

其中执行装置在判定装置判定不能执行连续再现时把表示不能执行连续再现的连续再现信息记录到记录介质上。

本发明提供一种记录设备的记录方法,所述记录方法包括:把数据记录到一种记录介质上的步骤;当数据被记录到记录介质上时判定该数据是否能够被连续再现的步骤;根据判定的结果执行有关数据的连续再现的处理的步骤。

本发明提供一种程序提供介质,用于提供执行一种处理的程序,所述处理包括:把数据记录到一种记录介质上的步骤;当数据被记录到记录介质上时判定该数据是否能够被连续再现的步骤;根据判定的结果执行有关数据的连续再现的处理的步骤。

本发明提供一种再现设备,包括:再现装置,用于再现记录在一种记录介质上的数据;提取装置,用于从再现装置再现的数据提取表示是否能够执行数据的连续再现的连续再现信息;以及添加装置,用于根据从提取装置提取的连续再现信息把一个间隙加到再现装置再现的数据上。

本发明提供一种再现方法,包括:再现记录在一种记录介质上的数据的步骤;用于从再现数据提取表示是否能够执行数据的连续再现的连续再现信息的步骤;以及相应于所提取的连续再现信息把一个间隙加到再现数据上的步骤。

本发明提供一种程序提供介质,用于提供执行一种处理的程序,

所述处理包括：再现记录在一种记录介质上的数据的步骤；用于从再现数据提取表示是否能够执行数据的连续再现的连续再现信息的步骤；以及相应于所提取的连续再现信息把一个间隙加到再现数据上的步骤。

根据本发明的一个方面的一种记录设备的特征，在于包括：记录装置，用于把数据记录到存储介质（例如，图25中的光头2）上；判定装置，用于当数据被记录到存储介质上时判定数据是否能够与已经记录的数据连续地得到再现（例如，图48的步骤S2）；以及，执行装置，用于根据判定装置的判定结果执行有关数据的连续再现的处理（例如，图48的步骤S5）。

根据本发明的另一方面的一种再现设备的特征，在于包括：再现装置（例如，图25中的光头2），用于再现记录在存储介质上的数据；提取装置，用于提取表示数据能够与再现装置再现的数据连续地得到再现的连续再现信息（例如，图50中的步骤41）；以及，相加装置，用于根据提取装置提取的连续再现信息，把一个间隙加到再现装置再现的数据上（例如，图50中的步骤46）。

首先，说明本发明中信息所记录到或再现自的存储介质上的文件分配。如图1所示，以下几种文件被记录在介质上：

VOLUM.TOC

ALBUM.STR

PROGRAM_\$\$\$PGI

TITLE_###.VDR

CHUNKGROUP_@@@CGIT

CHUNK_%%%%ABST

CHUNK_%%%%MPEG2

文件VOLUM.TOC和ALBUM.STR被放置在根目录中。进一步地，PROGRAM_\$\$\$PGI（在此情况下，\$\$\$表示一个节目号）被放置在刚好在根目录之下的目录PROGRAM中。类似地，TITLE_###.VDR（在此情况下，###表示一个标题号）被放置在目录

刚好在根目录之下的TITLE目录中，CHUNKGROUP_@@@.CGIT（在此情况下，@@@表示一个数据块组号）被放置在目录CHUNKGROUP中，且CHUNK_%%%%.ABST（在此情况下，%%%%表示一个数据块号）被放置在目录CHUNK中。

进一步地，在刚好在根目录之下的MPEGAV目录中产生一或多个子目录，且CHUNK_%%%%.MPEG2（在此情况下，%%%%表示一个数据块号）被放置在子目录下。

一个文件VOLUM.TOC通常被放置在介质上。然而，多个文件VOLUM.TOC也能够被放置在具有特殊结构的介质上，诸如具有ROM（只读存储器）和RAM（随机存取存储器）的混合结构的介质。该文件用于表示介质的整体性质。

图2中显示了VOLUM.TOC的结构。在一个开始部分放置了file_type_id(文件类型id)，它表示相关的文件是VOLUM.TOC。接着的是volume_information()(卷信息)，且最后是text_block()(文本块)。

图3显示了volume_information()(卷信息)的结构。该区域包括volume_attribute()(卷属性)、resume()(恢复)、volume_rating()(卷评级)、write_protect()(写保护)、play_protect()(播放保护)和recording_timer()(记录定时器)。

volume_attribute()(卷属性)是用于记录逻辑卷的属性的区，且其详细结构在图4显示。如该图所示，该区包括title_playback_mode_flag(标题重放模式标记)、program_playback_mode_flag(节目重放模式标记)等等。

resume()(恢复)是用于记录在介质被重新插入时用于恢复刚好在推出之前的状态的信息的区域，且其详细结构在图5中显示。

图3中的volume_rating()(卷评级)是用于记录用于根据年龄和种类就总的卷实现对观众的年龄的限制的信息的区，且其详细结构在图6中显示。

图3的write_protect()(写保护)是用于记录限制对记录在卷中的标题和节目进行改变和擦除操作的信息的区域，且其详细结构在图7中

显示。

图3中的play_protect()(播放保护)是用于记录设定再现确认或限制记录在卷中的标题和节目的次数的信息的区域，且其详细结构在图8中显示。

图3中的recording_timer()(记录定时器)是记录用于控制记录时间的信息的区域，且其详细结构在图9中显示。

图10显示了图2的文件VOLUM.TOC的区域text_block()。区域text_block()(文本块)包括区域language_set()(语言设定)和text_item()(文本项)，且其详细结构分别被显示在图11和图12中。

通常，一种介质只包括一个文件ALBUM.STR。然而，在具有特殊结构的介质—诸如具有ROM和RAM混合结构的介质—中，可以有多个文件。该文件被用来构造一种结构—其中通过结合多种媒体而形成一种介质。

图13显示了文件ALBUM.STR的结构。file_type_id(文件类型id)被放置在一个先导区，表示相关文件是ALBUM.STR。随后跟着的是文件album()(专辑)且最后是文件text_block()(文本块)。

文件album()(专辑)是记录用于处理作为一个总体的多个卷（多种媒体）的信息的区域且其详细结构在图14中显示。

图1的文件TITLE_###.VDR有几个标题。一种标题表示例如在CD盘上的一段音乐的标题或电视广播中的一个节目的标题。该信息的结构如图15所示。一个区域file_type_id(文件类型id)被放置在一个先导区，它表示相关的文件是TITLE_###.VDR。跟着的是title_info()(标题信息)，且最后跟着的是text_block()(文本块)。记号###表示表明标题号的字符串。

区域title_info()(标题信息)是记录数据块组上的标题的开始点、结束点和其他属性的区，且其详细结构在图16中显示。

几个节目中都有图1的文件PROGRAM_\$\$\$.PGI。该节目由指定标题的一个区域的一部分（或全部）的多个片断组成，且相应的片断被以指定的序列得到再现。图17显示了该信息的结构。一个区

`file_type_id`(文件类型id)被放置在一个先导区, 它表示相关的文件是 `PROGRAM_$$$PGI`。随后跟着的是是一个区 `program()`(节目)且最后跟着的是区 `text_block()`(文本块)。记号 `$$$`指定了表示标题号的字符串。

区 `program()`(节目)是记录收集和再现标题的必要部分而其材料未受到不可逆编辑处理所需的信息的区域, 且其详细结构在图18中显示。

图18的区域 `program()`(节目)表示一个 `play_list()`(播放清单)。`play_list()`(播放清单)的细节在图19中显示。

区域 `play_list`表示多个区域 `play_item()`(播放项)。区域 `play_item()`(播放项)的细节在图20中显示。

若干个数据块组中都有图1的文件 `CHUNKGROUP_###.CGIT`。该数据块组是用于排列一个位流的数据结构。在用户正常操作诸如 `VDR`(视盘记录机)的介质记录和再现设备的情况下, 该文件不被用户所识别。

图21显示了该信息结构。区域 `file_type_id`(文件类型id)被放置在一个先导区, 它表示相关文件是 `CHUNKGROUP_@@@.CGIT`。跟着它的有以下区域 `chunkgroup_time_base_flag`(数据块组时基标记)和 `chunkgroup_time_base_offset`(数据块组时基偏移), 随后跟着的是区域 `chunk_connection_info()`(数据块连接信息)且最后跟着的是区域 `text_block()`(文本块)。

区域 `chunkgroup_time_base_flag`(数据块组时基标记)表示关于数据块组的标准计数器的标记, 且区域 `chunkgroup_time_base_offset`(数据块组时基偏移)表示数据块组中的标准时间轴的开始时间。这是为每90kHz进行计数并具有32位大小的计数器设定的一个值。区域 `chunk_connection_info()`(数据块连接信息)是存储诸如视频信号的切换点或视频和音频信号的同步的奇点的信息的区域, 且其详细结构在图22中显示。

在区域 `chunk_connection_info()`(数据块连接信息)中, 区域

chunk_arrangement_info()(数据块排列信息)的循环由属于数据块组的若干个数据块放置。区域**chunk_arrangement_info()**(数据块排列信息)的细节在图23中显示。

若干个数据块具有图1的文件**CHUNK_%%.ABST**。一个数据块表示与一个流文件相应的信息文件。该信息的结构如图24所示。一个区域**file_type_id**(文件类型id)被放置在一个先导区,它表示相关的文件是**CHUNK_%%.ABST**。

图1中的文件**CHUNK_%%.MPEG2**是一个流文件。该文件存储不同于其他只记录信息的文件的MPEG的位流。

图25显示了用于在具有上述文件的作为介质的光盘上记录信息或从其再现信息的光盘设备的结构的一个例子。在这种光盘设备中,一种系统的一个光头2被安装在一片可编程光盘1上,且光头2被用于读取和写入数据。

光头2从光盘1读取的位流被RF和一个解调/调制电路3解调,并随后受到一个ECC电路4的错误校正并被发送到一个读取通道缓存器6以经过一个开关5吸收读取速率与解码速率之差。读取通道缓存器6能够从一个系统控制器13读取数据并把数据写入它。

从读取通道缓存器6输出的位流被一个解码器7解码,从解码器7输出视频信号和音频信号。从解码器7输出的视频信号被输入一个结合电路8,与从一个OSD(屏上显示器)控制电路9输出的视频信号相结合,随后从一个输出端P1输出到一个显示器(未显示)并在其上得到显示。从解码器7输出的音频信号从一个输出端P2被输出到一个未显示的扬声器,并在那里得到再现。

同时,从一个输入端P3输入的视频信号和从一个输入端P4输入的一个音频信号被一个编码器10编码,并随后被发送到一个写入通道缓存器11以吸收一个编码速率与一个写入速率之差。写入通道缓存器11也能够从系统控制器13读取数据和向其写入数据。

存储在写入通道缓存器11的数据被从写入通道缓存器11读取,经开关5被输入到ECC电路4以加入一种错误校正码并随后受到解调/调

制电路3的调制。从RF和解调/调制电路3输出的信号（RF信号）被光头2写入光盘1。

一个地址检测电路12检测道的地址信息，以记录或再现光盘1。系统控制器13控制光盘设备的相应部分的工作并包括：ROM22，用于存储将要由CPU21执行的处理程序；RAM23，用于暂时存储处理过程中产生的数据；以及，RAM24用于存储将要记录到光盘1上或从光盘1再现的各种信息文件。CPU21根据地址检测电路12的检测结果对光头2的位置进行细调。进一步地，CPU21对开关5进行切换控制。一个由各种开关和按钮构成的输入单元14在输入各种指令时受到用户的操作。

以下描述读取基本信息文件的操作。例如，在读取VOLUM.TOC信息文件时，系统控制器13的CPU21，利用事先装入处理程序的文件系统操纵指令，确定光盘1上VOLUM.TOC的记录物理地址以及该文件的长度。随后，CPU21根据VOLUM.TOC的地址信息移动光头2。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于读取模式，把开关5切换至读取通道缓存器6一侧，对光头2的位置进行细调并借助光头2开始读取操作。由此，VOLUM.TOC的内容被光头2读取，被RF和解调/调制电路3解调，受到ECC电路4的错误校正，并被累积在读取通道缓存器6中。

当累积在读取通道缓存器6中的数据量等于或大于VOLUM.TOC的大小时，CPU21停止读取操作。随后，CPU21从读取通道缓存器6读取相关数据并将该数据存储在RAM24中。

以下，借助写入VOLUM.TOC信息文件的例子，说明写入基本信息文件的操作。CPU21采用了事先装入处理程序的文件系统操纵指令来在文件系统（光盘1）中搜索具有等于或大于所要写入的VOLUM.TOC的大小的容量的空白区并确定地址。

随后，CPU21把RAM24中准备的以新写入的VOLUM.TOC传送到写入通道缓存器11。随后，CPU21根据空白区的地址信息把光头2移动到写入位置。进一步地，CPU21把光头2、RF和解调/调制电路3

以及ECC电路4置于写入模式，把开关5切换到写入通道缓存器11一侧，细调光头2的位置并随后启动光头2的写入操作。

由此，新准备的VOLUM.TOC的内容被从写入通道缓存器11读取，经过开关5被输入到ECC电路4从而加上一个错误校正码，并随后被RF和解调/调制电路3调制。从RF和解调/调制电路3输出的一个信号被光头2记录在光盘1上。当从写入通道缓存器11读取并记录在光盘1上的数据量等于VOLUM.TOC的大小时，CPU21停止写入操作。

最后，CPU21重写表示文件系统（光盘1）中VOLUM.TOC的一个指针以利用事先装入处理程序的文件系统操纵指令表示新写入的位置。

以下借助再现图1的CHUNK_0001.MPEG2的流的例子，描述基本流再现操作。CPU21利用事先装入处理程序的文件系统操纵指令，确定记录在光盘1上的CHUNK_0001.MPEG2的物理地址和该文件形成长度。随后，CPU21根据CHUNK_0001.MPEG2的地址信息把光头2移到读取位置。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于读取模式，把开关5切换至读取通道缓存器6一侧，细调光头2的位置，并随后借助光头2启动读取操作。

光头2读取的CHUNK_0001.MPEG2的内容经过RF和解调/调制电路3、ECC电路4和开关5被累积在读取通道缓存器6上。累积在读取通道缓存器6上的数据被输出到解码器7以受到解码处理，从而分别输出视频信号和音频信号。音频信号从输出端P2输出且视频信号从输出端P1经过结合电路8输出。

当从光盘1读取、解码和显示的数据量等于CHUNK_0001.MPEG2的大小时，或者当输入单元14指示停止读取操作时，CPU21停止读取和解码处理。

以下结合写入CHUNK_0001.MPEG2信息文件的例子，描述基本流记录操作。CPU21，利用事先装入处理程序的文件系统操纵指令，在文件系统（光盘1）中搜索具有等于或大于将要写入的CHUNK_0001.MPEG2的大小的空白区，并确定地址。

从输入端P3输入的视频信号和从输入端P4输入的音频信号被编码器10编码并随后累积在写入通道缓存器11。随后，CPU21根据空白区的地址信息把光头2移到写入位置。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于写入模式，把开关5切换到写入通道缓存器11一侧，细调光头2的位置并随后借助光头2开始写入操作。以此，新准备的CHUNK_0001.MPEG2的内容被从写入通道缓存器11读取，经过开关5、ECC电路4和RF和解调/调制电路3被输入到光头2，并被记录在光盘1上。

当从写入通道缓存器11读取并记录在光盘1上的数据量变得等于事先设定的值时，或者当输入单元14指示停止写入操作时，CPU21停止写入操作。最后，CPU21，利用事先装入处理程序的一个文件系统操纵指令，将表示文件系统（光盘1）中CHUNK_0001.MPEG2的一个指针重写，以表示新写入位置。

现在假定如图26所示的信息文件和流文件被记录在光盘1上。根据该例子，包括了被称为PROGRAM_001.PGI的一个节目文件。进一步地，光盘1包括具有三个标题TITLE_001.VDR、TITLE_002.VDR和TITLE_003.VDR的标题。进一步地，光盘1还包括被称为CHUNK_0001.MPEG2、CHUNK_0011.MPEG2和CHUNK_0012.MPEG2的三个流文件，并带有作为与其相应的信息的三个信息文件CHUNK_0001.ABST、CHUNK_0011.ABST和CHUNK_0012.ABST。

图27显示了具有图26所示的信息文件和流文件的光盘1的逻辑结构。根据该例子，数据块信息文件CHUNK_0001.ABST指定流文件CHUNK_0001.MPEG2，数据块信息文件CHUNK_0011.ABST指定流文件CHUNK_0011.MPEG2且数据块信息文件CHUNK_0012.ABST指定流文件CHUNK_0012.MPEG2。具体地，一个流的文件ID在图24的CHUNK_%%%.ABST中被称为chunk_file_id的一个域中指定。

进一步地，根据该例子，数据块组信息文件CHUNKGROUP_001.CGIT指定数据块信息文件

CHUNK_0001.ABST, 且数据块组信息文件CHUNK_002.CGIT指定数据块信息文件CHUNK_0011.ABST和CHUNK_0012.ABST。具体地, 数据块信息的文件ID在图23的chunk_arrangement_info()中被称为chunk_info_file_id(数据块信息文件id)的一个域中得到指定。chunk_arrangement_info()(数据块排列信息)具有将要在数据块组信息文件中提供的数据结构并存在于属于相关数据块组的若干个数据块中(图23的chunk_arrangement_info()在图2的chunk_connection_info()(数据块连接信息)中描述且chunk_connection_info()(数据块连接信息)在图21的CHUNKGROUP_@@@.CGIT中得到描述)。

CHUNKGROUP_001只包括一个chunk_arrangement_info()(数据块排列信息), 且在该域中的chunk_info_file_id(数据块信息文件id)指定CHUNK_0001。CHUNK_002包括两个chunk_arrangement_info()(数据块排列信息)域, 其中CHUNK_0011和CHUNK_0012分别得到指定。考虑到这种情况, 数据块组能够指定多个数据块的再现顺序。

具体地, 相关的数据块组的时钟的初始值由图21的CHUNKGROUP_###.CGIT的chunkgroup_time_base_offset(数据块组时基偏移)确定。随后, 在寄存相应的数据块时, 图23的chunk_arrangement_info()(数据块排列信息)的presentation_start_cg_count(表演开始cg计数)和presentation_end_cg_count(表演结束cg时间计数)得到指定。

例如, 如图28所示, CHUNK_0011的长度(时间)由记号A指定, 且CHUNK_0012的长度(时间)由记号B指定。CHUNK_0011的域presentation_start_cg_count(表演开始cg计数)等于chunkgroup_time_base_offset(数据块组时基偏移)且域presentation_end_cg_count等于chunkgroup_time_base_offset+A。进一步地, CHUNK_0012的域presentation_start_cg_count(表演开始cg计数)等于chunkgroup_time_base_offset+A且presentation_end_cg_count等于chunk_group_time_base_offset+A+B。通过进行如此

方式的设定，CHUNKGROUP_002在连续再现CHUNK_0011和CHUNK_0012时得到定义。

进一步地，当CHUNK_0011和CHUNK_0012的再现时间重叠时，域可通过以这样的一种方式移动时间而得到指定。进一步地，可以通过在图23的chunk_arrangement_info()(数据块排列信息)中的transition_info()(切换信息())中进行说明，而在两个流之间的转移中指定特殊效果（淡入、淡出、扫描等等）。

在图26（图27）的例子中，标题信息文件TITLE_001.VDR和TITLE_002.VDR指定了数据块组信息文件CHUNKGROUP_001.CGIT，且标题信息文件TITLE_003.VDR指定了数据块组信息文件CHUNK_002.CGIT。具体地，在图16的title_info()(标题信息)中被表示为cgit_file_id的一个域中，数据块组的一个文件ID得到指定，且在域title_start_chunk_group_time_stamp(标题_开始数据块组时间标志)和title_end_chunk_group_time_stamp(标题_结束数据块组时间标志)中，指定了其中相关的标题在数据块组中得到定义的时间范围。

例如，根据图27的例子，TITLE_001指定了CHUNKGROUP_001的前一半且TITLE_002指定了其后一半。进一步地，根据来自用户的一个请求进行了分割，该位置对用户是任意的，且这种位置不能事先确定。在此情况下，假定TITLE_001和TITLE_002的分割位置被确定在远离于CHUNKGROUP_001先导区为A的位置。

TITLE_001把CHUNKGROUP_001指定为一个数据块组，把CHUNKGROUP_001的开始时间指定为标题的开始时间，并把用户指定的一个点指定为标题结束时间。

即，CHUNKGROUP_001的chunkgroup_time_base_offset(数据块组时基偏移)（先导位置）被设定为TITLE_001的title_start_chunk_group_time_stamp(标题_开始数据块组时间标志)，且加有长度A的CHUNKGROUP_001的chunkgroup_time_base_offset(数据块组时基偏移)被设定为

TITLE_001的title_end_chunk_group_time_stamp(标题_结束数据块组时间标志)。

进一步地, TITLE_002把CHUNKGROUP_001指定为一个数据块组, 把用户指定的一个时间点指定为标题的开始时间, 并把CHUNKGROUP_001的结束时间指定为标题的结束时间。

即, 作为TITLE_002的title_start_chunk_group_time_stamp(标题_开始数据块组时间标志), 加有长度A的CHUNKGROUP_001的chunkgroup_time_base_offset(数据块组时基偏移)(先导位置)被设定为TITLE_002的title_end_chunk_group_time_stamp(标题_结束数据块组时间标志), 加有CHUNKGROUP_001的长度的CHUNKGROUP_001的chunkgroup_time_base_offset(数据块组时基偏移)得到设定。

进一步地, TITLE_003把CHUNKGROUP_002指定为一个数据块组, 把CHUNKGROUP_002的开始时刻指定为标题的开始时刻, 并把CHUNKGROUP_002的结束时刻指定为标题的结束时刻。

即, 作为TITLE_003的title_start_chunk_group_time_stamp(标题_开始数据块组时间标志), CHUNKGROUP_002的chunkgroup_time_base_offset(数据块组时基偏移)得到设定, 且作为TITLE_003的title_end_chunk_group_time_stamp(标题_结束数据块组时间标志), 加有CHUNKGROUP_002的长度的CHUNKGROUP_002的chunkgroup_time_base_offset(数据块组时基偏移)得到设定。

进一步地, 根据该例子, 一个节目信息文件PROGRAM_001.PGI指定TITLE_001的一部分和TITLE_003的一部分, 以按照这种顺序进行再现。具体地, 一个片断可通过用图20的play_item()(播放项)中的title_number(标题号)指定标题并用各个标题定义的时间定义开始点和结束点, 而得到抽取。通过汇集多个这样的片断, 可构成一个节目。

以下描述把新的信息附加记录(附加记录)到光盘1上的操作。具体地, 这种记录是例如借助定时器记录或当用户通过操作输入单元

14命令至光盘的实时记录而进行的。在后一种情况下，虽然当记录按钮被按下时不能预测记录结束时刻，当单动作记录功能（在操作了该按钮之后执行恒定时间的记录的功能）的按钮被按下时，结束时刻是可以预测的。

在此情况下，将描述定时器记录的一个例子。在此情况下，假定光盘设备的用户已经事先指定了记录开始时刻、记录结束时刻、位流的位速率、用于执行记录的通道等等。进一步地，假定已经在预约记录时事先认识到在光盘1中剩有与位速率和记录时间符合的空白容量。

当在预约记录时与执行预约记录时之间在光盘1上进行进一步的记录时，会有一种情况—其中不能借助指定的位速率保证当前所预约的节目记录的容量。在此情况下，CPU21从指定值减小位速率并记录预约时间的信息，或在使位速率保持不变的同时记录可记录时间中的信息。在此情况下，当执行进一步记录且在预约记录中产生了不利情况时，CPU21自然发出一个消息，把这种情况通知用户。

现在，当接近预约记录的开始时刻时，CPU21，利用内装的定时器或时钟，自动从睡眠模式恢复到操作模式。进一步地，CPU21利用事先装入处理程序的文件系统操纵指令，并在光盘上保证一个能够记录预约节目的区域。即，预约记录的结束时间减去开始时间而获得的一个数字值（记录时间）乘以位速率，是记录预约节目所需的区域的大小，且CPU21首先保证该区域大小。除了上述操作之外，在记录操作中，当需要记录流文件以外的信息文件时，在其中例如寄存一个新标题需要一个标题信息等时，需要在光盘1上保证能够记录该信息文件的容量。当不能保证具有所需量的区时，借助上述方法（改变位速率、只在可记录时间内记录的方法等等）来处理。

进一步地，在此情况下，操作涉及新标题下的记录，因而用户把一个新流文件的文件名指定为新流目录的新流文件。在此情况下，该文件名被指定为 ¥ MPEGAV ¥ STREAMS_003 ¥ CHUNK_0031。即，如图29所示，它是根目录下目录MPEGAV下目录STREAM_003

下具有名称CHUNK_0031.MPEG2的一个文件。

CPU21命令相应部分执行记录模式。例如，从未显示的调谐器输入到输入端P3的视频信号和输入至输入端P4的音频信号被编码器10编码并随后累积在写入通道缓存器11中。随后，CPU21根据预先保证的区域的地址信息把光头2移到一个写入位置。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于写入模式，把开关5切换到写入通道缓存器11一侧，细调光头2的位置，并随后借助光头2开始写入操作。因此，新准备的CHUNK_0031.MPEG2被从写入通道缓存器11读出并经过开关5、ECC电路4、RF和解调/调制电路3以及光头2而被记录到光盘1上。

当上述写入操作继续进行且产生了以下条件之一时，CPU21停止写入操作。

- 1) 当达到了预约记录的结束时刻时；
- 2) 当至光盘1的记录由于容量不足或其他原因而不能进行时；
- 3) 当记录操作被命令停止时。

随后，CPU21，利用事先装入处理程序的一个文件系统操纵指令，把文件系统中指定CHUNK_0031.MPEG2的一个指针重写为表示新写入位置的一个值。进一步地，CPU21准备数据块信息、数据块组信息、以及标题信息的相应文件，并通过指定适当的名称记录数据。进一步地，在记录或预约操作中，需要在光盘1上保证能够记录这些文件的空白容量。

以此方式，如例如图30所示地形成了新的信息文件。在图30中，新形成的当前文件在文件名的右上部分加有星号(*)。

图31显示了新形成的信息文件之间的关系。TITLE_004指定CHUNKGROUP_003，CHUNKGROUP_003指定CHUNK_0031且CHUNK_0031指定STREAM_0031。

即，新的流作为TITLE_004而被寄存到信息文件。用户能够借助光盘设备确认标题的功能而被通知TITLE_004的属性等，进一步地，TITLE_004能够得到再现。

以下描述当在图26（图27）示例性显示的光盘1上进行覆写时的操作。覆写指的是在已经记录的节目上（通过擦除该节目）记录新节目的操作，这与在录象带上记录信号的情况类似。

在覆写中，重要的是定义覆写的开始位置。例如，假定用户指定从TITLE_001的先导区开始覆写。在此情况下，覆写通过按TITLE_001、TITLE_002和TITLE_002的顺序相继进行重写。当记录操作即使在重写进行到TITLE_003的最后区域之后也未完成时，记录操作通过在光盘1上的空白区中保证新区域而得以继续进行。例如，当TITLE_002限定了记录开始位置时，TITLE_001不被当前的记录操作所重写，因为它被设置在记录开始位置之前。

现在，假定覆写操作是借助从TITLE_003的先导区记录的定时器进行的。在此情况下，光盘设备的用户预先指定了记录的开始时间和结束时间、位流的位速率、用于进行记录的通道等等。进一步地，假定对于覆写是重要的记录开始位置被指定为TITLE_003的先导区。进一步地，假设在此情况下，当记录得到预约时，与位速率和记录时间兼容的容量已经事先被确认为是存在于光盘1上的。在覆写的情况下，能够从一个指定位置覆写的（多个）标题的总容量和光盘1的空白容量构成了一个可记录容量。即，在当前情况下，由TITLE_003控制的流STREAM_0011和STREAM_0012的总容量与在光盘1上的空白容量之和构成了一个可记录容量。

在覆写中，有相对于可记录容量来说按什么顺序进行实际记录的选择。首先，考虑按照标题指定的流顺序的第一种记录方法。即，在当前情况下，它是这样一种方法，即其中首先从STREAM_0011的先导区开始记录，且当记录已经进行到STREAM_0011的结束时，从STREAM_0012的先导区继续记录，且当记录进行到STREAM_0012的结束时，在一个空白区中进行记录。另一种方法是这样的，即其中首先在一个空白区进行记录，且当空白区满了时，在已有的流上进行记录。

前一种方法就仿真录象带来说是优秀的。即，该方法的特征，在

于在与录象带的操作类似的操作的意义上用户容易理解。后一种方法的特征，在于它在保护记录的数据方面是优秀的，因为稍后进行对已经记录的流的擦除。

进一步地，当在记录预约时与执行预约记录时之间在光盘1上进行进一步的记录时，有一种情况—其中以指定的位速率记录当前预约的节目的容量不能得到保证。在此情况下，与上述的情况类似地，当预约得到执行时，位速率被自动减小，且在位速率保持不变的情况下以预约的时间进行记录或者只以可记录的时间进行记录。

当接近预约记录的开始时间时，光盘设备从睡眠模式恢复到工作模式。CPU21保证光盘1上的所有空白容量。自然地，虽然有一种方法—其中空白容量在该时刻未得到保证，但在需要的时刻得到了保证，在此情况下，为了说明，假定为开始记录操作保证了必需的区域。

进一步地，当由于开始时间、结束时间和位速率都象在定时器记录中那样得到指定，因而所需的区域的大小已经知道时，能够以所需的量（或者一个量加上某些余量）保证容量。在记录操作中，当需要记录信息文件时，例如，当需要标题信息文件以作为新标题寄存文件时，需要保留能够记录该信息文件的容量。

在此情况下，假定一个文件名被给予一个新的流文件，以作为新流目录的新流文件。即，在此情况下，文件名被指定为¥MPEGAV¥STREAMS_002¥CHUNK_0031。即，如图32所示，在根目录下的MPEGAV目录下的STREAM_002目录下，形成了一个名为CHUNK_0031.MPEG2的文件。

输入到输入端P3的视频信号和输入到输入端P4的音频信号被编码器10编码，且随后在写入通道缓存器11中累积。随后，CPU21根据事先保证的区域的地址信息，将光头2移到写入位置。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于写入模式，把开关5切换到写入通道缓存器11一侧，细调光头2的位置，并随后使光头2开始写入操作。因而，新准备的CHUNK_0031.MPEG2的

内容被从光通道缓存器11读出并经过开关5、ECC电路4、RF和解调/调制电路3和光头2而被记录到光盘1上。

此时，流文件CHUNK_001.MPEG2首先被重写。进一步地，当记录操作已经进行到CHUNK_0011.MPEG2的最后区域时，对CHUNK_0012.MPEG2以及且进一步地CHUNK_0031.MPEG2继续进行记录操作。

当上述操作继续且发生了上述情况下的三个条件中的任何一个时，CPU21停止写入操作。

随后CPU21，通过利用事先装入处理程序的文件系统操纵指令，更新流文件、数据块信息、数据块组信息和标题信息。

另外，文件的结构根据写入操作完成的时序而得到改变。例如，在其中在进行了在两个流CHUNK_0011.MPEG2和CHUNK_0012.MPEG2之上的覆写之后的情况下，在CHUNK_0031.MPEG2上进行记录，光盘1的文件的结构如图33所示。当前新形成的文件的文件名的右上部分带有星号(*)。

图34显示了以此方式新形成的文件（图33的文件）之间的关系。如与图31比较可见，CHUNK_0031，作为包括在TITLE_003指定的CHUNKGROUP_002和STREAM_0031指定的CHUNK_0031中的数据块而得到了包括。

同时，当覆写操作在已有流上的覆写中间完成时，例如，当覆写操作是在CHUNK_0011上的记录中间完成时，为覆写保证的流CHUNK_0031未被覆写，因而它得到了释放。在此情况下，进行特别的标题处理。即，当覆写操作已经从TITLE_003的前区开始且记录操作在中间完成时，标题在那里被分开。即，如图35所示，从开始位置至覆写操作的结束位置的域由新的TITLE_003指定，且随后的部分（原来的TITLE_003的其余部分）被TITLE_004指定。

以下描述文件的再现操作。现在，假定具有图26显示的的文件的光盘1被插入光盘设备且标题得到再现。首先，当光盘1被装载时，CPU21从光盘1读取信息文件并将它们存储在RAM24中。该操作通过

重复上述读取基本信息文件的操作而进行。

首先，CPU21读取VOLUM.TOC和ALBUM.STR。随后CPU21搜索多少个具有扩展名VDR的文件出现在目录TITLE下的域中。具有该扩展名的文件是具有标题信息的文件且这种文件的数目等于标题的数目。根据图26的例子，这种标题的数目是3。随后CPU21读取这三个标题信息文件并将它们存储在RAM24中。

CPU21控制OSD控制电路9以产生表示记录在光盘1上的标题信息的字符信息，借助结合电路8将其与视频信号混合，并从输出端P1把该信号输出到一个显示器。在此情况下，三个标题和相应的大小和属性（名称、记录时间和日期等等）得到显示。

在此情况下，假定用户指定了例如TITLE_002的再现。TITLE_002的一个信息文件（图16中的title_info()的cgit_file_id）以指定CHUNKGROUP_001的文件ID得到记录，且CPU21记录该文件ID并存储它，并把CHUNKGROUP_001存储到RAM24中。

随后，CPU21进行搜索，以确定TITLE_002的哪一个数据块开始时间与结束时间（图16的title_info()中的title_start_chunk_group_time_stamp(标题开始数据块组时间标志)和title_end_chunk_group_time_stamp(标题结束数据块组时间标志)）对应。这是通过比较数据块组信息中相应的数据块所寄存到的信息（图23中的chunk_arrangement_info()(数据块排列信息)中的presentation_start_cg_time_count(表演开始cg时间计数)和presentation_end_cg_time_count(表演结束cg时间计数)）而进行的。在此情况下，如图27所示，已知TITLE_002的开始时间被设置在CHUNK_0001的中间。即，可以理解到TITLE_002能够通过从流文件CHUNK_0001.MPEG2的中间开始再现而从一个先导区得到再现。

随后，CPU21进行搜索流中的TITLE_002的先导区的位置。即，计算作为流的偏移时间（时间标志）的TITLE_002的开始时间在何时。随后，利用数据块文件中的特征点信息指定刚好在该开始时间之前的一个再现开始点。因此，再现开始时间从文件的先导区的偏移距

离能够得到确定。

随后，CPU21，利用事先装入处理程序的文件系统操纵指令，确定光盘1上CHUNK_0001.MPEG2得到记录的物理地址和大小。进一步地，前面计算出的记录开始位置的偏移地址被加到该地址上，且TITLE_002的再现开始位置的地址得到最后确定。

随后，CPU21根据CHUNK_0001.MPEG2的地址信息，把光头2移到先导位置。进一步地，CPU21把光头2、RF和解调/调制电路3以及ECC电路4置于读取模式，把开关5切换到读取通道缓存器6一侧，对光头2的位置进行细调，并使光头2开始读取操作。因此，CHUNK_0001.MPEG2的内容被累积在读取通道缓存器6上。

累积在读取通道缓存器6上的数据被输出到解码器7并受到解码处理，且一个视频信号和一个音频信号得到输出。当显示的数据量等于CHUNK_0001.MPEG2的大小时，CPU21进行TITLE_003的再现。TITLE_003的再现操作与TITLE_002的再现操作类似。

当完成再现寄存的文件时，或者当命令停止读取操作时，停止读取和解码处理。

进一步地，当一个新的盘作为光盘1被装载到光盘设备上，或者当不同格式的盘得到装载时，CPU21要在盘装载时读取VOLUM.TOC和ALBUM.STR，然而，在这些盘上没有这些文件。在此情况下，即，当不能读取VOLUM.TOC和ALBUM.STR时，CPU21输出一个消息以从用户获得指令。用户向CPU21发出一个指令，以推出光盘1（例如当装载了不同格式的盘时）或者初始化该盘（例如当装载了相同格式的新盘时）或者用一定的方法恢复数据（例如当装载了相同格式的盘但数据被破坏时）。

以下进一步描述标题。如图15所示，文件TITLE_###.VDR是用于存储标题信息的文件。有关一个标题的信息被记录在一个title_info()(标题信息)中。出现在TITLE_###.VDR中的title_info()(标题信息)的数目是一。因此，TITLE_###.VDR在该卷中以若干个标题存在。

标题号不在图16的title_info()(标题信息)中被定义,而是由文件名或文件ID确定。因此, TITLE_###.VDR中的正整数###表示标题号。标题不表示结构而是表示从一个表示开始点的标题索引至表示一个附属于一个数据块组的相继的标题的先导区标题索引之间的范围或直到数据块组结束点的一部分范围。

图15的TITLE_###.VDR的file_type_id(文件类型id)是一个ID,它表示了如图36所示的利用title_info()(标题信息)记录并用大小为16的字符串表示的文件。文件text_block()(文本块)是用于存储各种文本的区域,且只有text_block()(文本块)中允许使用的文本项得到了说明。

如图16所示, title_info()(标题信息)是这样一个区域,即其中标题的开始点和结束点以及与其他标题有关的其他属性被写入到数据块组上。进一步地,文件title_info()(标题信息)能够具有一个标记,该标记表示当数据以标题号的顺序被再现时是否能够在标题之间保证无接缝再现。借助该标记,能够预先检测是否能够借助光盘设备实现标题之间的无接缝再现,且还能够知道在组后数据中是否需要重新排列。

虽然保证了在标题内和数据块组内的无接缝再现,有一种情况——其中由于标题的边界可以是文件的边界而未保证标题之间的无接缝再现。然而,可以建立一种状态,其中通过作为光盘设备的一种功能而进行重新排列等,而通常进行了无接缝再现。

图16的title_info()中的一个域title_info_length表示以字节为单位的title_info()(标题信息)的长度。一个域flags_for_title中记录有一个相应的标题的写入属性(改变确认)、再现次数的限制、速率的量级等。一个域cgit_file_id中记录有作为相应的标题的基础的数据块组(CHUNKGROUP_@@@.CGIT)的信息文件的file_id。

域title_start_chunk_group_time_stamp(标题开始数据块组时间标志)记录有标题在数据块组所定义的本地时间轴上的再现开始点时间。该值表示了标题的标题指标所表示的图象的显示时间。域title_end_chunk_group_time_stamp(标题结束数据块组时间标志)记录

有标题在数据块组所定义的本地时间轴上的再现结束点时间。该值等于再现数据块组的结束点或由表示时间轴上刚好在其后的一个标题的开始点的标题指标表示的值。

一个域title_playback_time()记录有标题的再现时间（时间编码值或帧或场数）。一个域number_of_marks记录有标题中的标志组的总数（除了标题指标）。如图37所示，一个域mark_type(标志类型)记录有附属在标题中的任意位置的标志种类。标志还被用作标题中的随机存取点。一个域mark_chunk_group_time_stamp记录有在这样的位置的时间标志，即在该位置标志以从具有小值的标志开始的顺序被设定在数据块组的时间轴上。可能存在具有与标题的开始点和结束点的相同的时间标记的指标。一个域stuffing_bytes记录有填充字节，且其长度是8n位（ n^30 ）。

以下进一步说明图21至24显示的数据块和数据块组。CHUNKGROUP_####.CGIT是描述标题的时间轴、数据块的结构和包括在标题中的非连续点处理的定义的文件。

标题由各种位流构成，诸如没有视频信号的位流和DV（数字视频）信号构成的位流。在DV格式下，时间轴由帧单位指定，且当基准是由MPEG2视频信号的STC（系统时间时钟）构成时，格式不同且DV信号的位流不能得到控制。

因此，在标题中设定了本地时间轴。这种时间轴不取决于构成标题的流。标题的边界得到设定，而不论数据块的边界如何。因此，相对于包括多个（任意数目的）标题的数据块的集合来设定本地时间轴，而不是为各个数据块（与位流成一一对应关系地）或者为各个标题设定本地时间轴，是更为适当的。

在一个数据块组中，一个定义了单个的时间轴，且一个数据块被贴在时间轴上，借助它确定了数据块的显示时间。即，在其中位流文件（字节串）的内容在时间轴发展的状态下将数据块排列在一个数据块组中。包括在排列在时间轴上的单个的位流文件中的所有数据块都被称为路径。多个路径可被排列在一个数据块组中。指定再现一个数

据块组的开始时间和结束时间的路径被称为主路径且其他的路径被称为子路径。子路径主要代表了随后附加记录的音频信号的数据块。

数据块的一个连接点不一定与标题的边界一致，因而该连接点不构成标题的属性。然而，当数据块之间的关系被包括在各个数据块的属性中时，在层级结构中就会产生矛盾。非连续点的这种信息被设置在数据块与标题之间，且将该信息置于数据块组的层级结构中似乎是适当的。

总之，提供给数据块组的信息是在时间轴排列数据块的一种方式，再现数据块的一种序列，在用于连接数据块的结束和将要连续地再现的数据块的开始的点处出现的一个非连续点等等。

如图38所示，CHUNKGROUP_###.CGIT的file_type_id(文件类型id)是一个标识符，表示该文件是CHUNKGROUP_###.CGIT的，并由符合ISO646的16字符的字符串表示。在chunkgroup_time_base_flag(数据块组时基标记)中，记录一个与数据块组的基准计数器有关的标志。在chunkgroup_time_base_offset(数据块组时基偏移)中，记录数据块组中的基准时间轴的开始时间。该值是被置于一个用于以90kHz的时钟计数的计数器的值并用64位表示。一个区域text_block()(文本块)是用于存储各种文本的区域，其中只有在text_block()(文本块)中允许使用的文本项得到了描述。

如图22所示，chunk_connection_info()(数据块连接信息)是用于记录奇点（视频信号的切换点、视频信号和音频信号的同步等等）的信息的文件，它指定了数据块之间的连接情况。在诸如编辑操作产生的数据块之间的连接点的奇点，数据块必须在GOP中间得到改变。在编辑点附近的这种信息在此得到描述。数据块不属于两个或更多的数据块组。

一个区域chunk_connection_info_length记录有以字节为单位表示的chunk_connection_info()(数据块连接信息)的长度。一个区域number_of_chunk记录有数据块组中所用的数据块的总数。如图39所示，一个区域chunk_sync_play_flag(数据块同步播放标记)是表示是否

需要同时再现两个或更多的数据块的标记，值0表示再现单个数据块且值1表示同时再现多个数据块。

在图 23 的 `chunk_arrangement_info()`(数据块排列信息)中，`chunk_arrangement_info_length`记录有以字节为单位表示的每个数据块的信息的长度（包括从`chunk_arrangement_info_length`的开始字节至 `transition_info()`(切换信息)的结束字节的长度）。区域 `chunk_info_file_id`(数据块信息文件id)记录有构成一个对象的 `chunk_info_file`。

一个区域`chunk_switch_stream_id`记录有一个流程的`stream_id`，当两个数据块连接在一起时连续再现它们。作为这种ID，例如，可采用用于识别记录在MPEG2的包标头中的视频信号和音频信号的ID。一个区域`presentation_start_cg_time_count`(表演开始cg时间计数)记录有用在一个数据块组中的时间表示相关数据块的显示开始时间的时间计数值。数据块的该显示开始时间用数据块组中定义的全局时间标志表示。相关数据块从数据块组中的时间开始受到显示。区域`presentation_end_cg_time_count`(表演结束cg时间计数)记录有以一个数据块组中的时间表示相关数据块的显示结束时间的时间计数值。数据块的该显示结束时间用数据块组中定义的全局时间标志表示。

如图40所示，一个区域`original_time_count_type`记录有流中所用的时间计数的种类。例如，在MPEG2视频信号的情况下，`original_time_count_type`被置于0000。一个区域`number_of_start_original_time_count_extension`记录有当需要多个时间计数时代表新要求的开始时间的时间计数数目。一个区域`number_of_end_original_time_count_extension`记录有当需要多个时间计数时代表新要求的结束时间的时间计数数目。区域`presentation_start_original_time_count`记录有同`presentation_start_cg_time_count`(表演开始cg时间计数)相应的流中的计数器值或时间。区域`present_end_original_time_count`记录有同`present_end_cg_time_count`相应的流中的计数器值或时间。

一个区域to_ext_attributes记录有用于time_count_extension的属性。time_count_extension可以输入有例如它被加到哪一个流上的信息。一个区域start_original_time_count_extension记录有切换数据块所需的开始计数器值或开始时间。这是当需要记录多个时间或计数器值时采用的一种选择。一个区域end_original_time_count_extension记录有切换数据块所需的结束时间或结束计数器值。这也是当需要记录多个时间或计数器值时采用的一种选择。一个区域transition_info()(变换信息)记录有提供切换数据块时的特殊效果所需的信息。例如,数据块的指定、切换时间和特殊效果的种类在此得到描述。

如图24所示,一个文件CHUNK.%%%%.ABST是这样文件,即它记录有从与sub_file(子文件)号%%%%构成一个数据块的位流提取的特征点。该文件以开始字节位置、构成GOP的各个位流的各个单元处的长度和属性、音频帧等等描述。GOP信息或音频帧信息分别在各个数据块的CHUNK.%%%%.ABST上得到概述(子文件)。

如图41所示,CHUNK.%%%%.ABST的file_type_id(文件类型id)记录有一个标识符,用符合ISO646的16个字符表示它是与stream_info()一起记录的文件。

如图42所示,info_type(信息类型)记录有图24中显示的跟随的stream_info的类型。在此,指定一种流。区域number_of_programs(节目数)记录有包括在MPEG2的TS(输送流)中的节目数目。为了检测该数目,需要读取PSI(节目专用信息)。对于TS以外的项,值变为1。一个区域number_of_streams记录有用在节目中的流的数目。在TS的情况下,该值变为等于不同的PID(包标识符)的数目。在TS以外的MPEG流的情况下,具有不同的流ID的流的数目被记录在这里。

一个区域stream_identifier记录有流ID或流ID的扩展。在TS的情况下,PID被用作流ID。

如图43所示，一个区域slot_unit_type(片单位类型)，当流以恒定的间隔得到分割时，记录有流的分割方式。当分割的指标是诸如帧、域等等的时间时，采用了时间标志值。一个区域slot_time_length记录有对应于1个片的时间。该值用采用90kHz的时钟计数的计数器的时间标志的值表示。一个区域number_of_slot记录有写入CHUNK.%%%%.ABST的slot_info()的数目。一个区域number_of_I_pictures_in_a_slot记录有一个片中的I图象的数目。该值是等于或大于1并小于等于15的整数值。然而，包括在设置在刚好在以GOP标头开始的一个片之前的一个片中的I图象的数目可以小于该值。当以没有设置在刚好在GOP标头之后的I图象的图象标头开始的片得到设定时，该值得到采用。

以下进一步说明图17和18中显示的节目。在PROGRAM_\$\$\$PGI中只有一个program()(节目)。一个卷中有若干个节目PROGRAM_\$\$\$PGI。一个节目号在program()(节目)中未被定义，但借助文件名或文件ID而得到指定。

如图44所示，表示记录有program()(节目)的一个文件的ID由PROGRAM.\$\$\$PGI的file_type_id(文件类型id)中长度为16的一个字符串记录。一个区域text_block()(文本块)带有用于存储各种文本的区域。在此情况下，只描述允许在text_block()(文本块)中使用的文本项。

图18的program()(节目)的区域flag_for_program记录有关于节目的各种标记。例如，记录有节目的记录写入属性(改变确认)，对再现次数的限制和速率级。

如图45所示，一个区域program_status(节目状态)记录有节目的属性。虽然域的设定是可选的，当设定未进行时，必须设定“无”。

一个区域program_playback_time()记录有节目的再现时间。一个区域number_of_play_sequence记录有用在该节目中的play_sequence的数目。然而，根据这种格式的例子，该值被固定在1。即，根据该格式的例子，1一个节目与1个ch(通道)的再现相对应，

因而为了实现2个通道的的同时再现，可以指定2个节目的同时再现。没有了与1个通道的再现相应的限制，可以借助1个节目进行2个通道的同时再现。当利用多通道I/O同时再现两个播放序列时，光盘设备判定播放序列被分配了哪一个输出通道。

一个区域number_of_play_lists记录有用在该播放序列中的play_list。在此例子中，该值被置于1。一个区域play_list_start_time_stamp_offset记录有由一个定时器计数的播放序列—该定时器从播放序列的开始时间启动。该值构成播放清单的开始时间。在该节目中，在播放序列中必须有一个播放清单。时间系统的单位是90kHz（1/90000秒构成了最小的时间单位）。一个区域stuffing_bytes（填充字节）记录有填充的字节。该长度被设定在8n位（ n^30 ）。

以下进一步描述一个非连续点标记。在此情况下，非连续点标记表示图19的play_list()(播放清单)的seamless_connection_flag(无缝连接标记)或者记录的一个标志，该标志代表了如图37中显示的指标类型8的非无缝再现点。

如图46所示，seamless_connection_flag(无缝连接标记)的值0表示与前面播放的项的连续再现（无缝重放）未得到保证或者是未知的，且该值1表示无缝重放是得到保证的。

即，如图47所示，当该标记是0时，从先前播放项再现一个预定播放项（不中断图象或语音）。与之相比，当标记是1时，有一种情况，其中在先前播放时间已经完成且直到相继播放项被再现之间有一个间隔，这造成了非连续的部分。

以下结合图48的流程图描述标题的非连续标记的记录处理。首先，在步骤S1，用户通过操作输入单元14而指定构成一个记录对象的标题。此时，在步骤S2，当步骤S1指定的标题被记录在光盘1上时，CPU21判定该标题是否能够与已经记录的先前标题连续地再现。这种判定是通过比较读取通道缓存器6中储存的数据量（解码器7解码数据容量所需的时间）和标题之间产生的处理而进行的。即，当读取通道

缓存器6在直到相继的标题被再现的时间段内出现空白时，连续再现被判定为不可能，且当读取通道缓存器6未出现空白时，连续再现被判定为可能。

当连续再现不可能时，操作进行到步骤S3，CPU21控制OSD控制电路9并发出一个消息，该消息表明被命令进行记录的标题当前不能与已经记录的标题连续地得到再现，即在两者之间产生了一个非连续点。该消息从结合电路8经过输出端P1而显示在显示器上。

用户看到该消息并通过操作输入单元14输入是否确认了非连续点的出现。当用户确认了非连续点的出现时，在步骤S5，CPU21在图37显示的标志中作为表示非连续点的指标的指标类型8设定到图16的title_info()(标题信息)中的mark_type(标志类型)并执行一个处理以把非连续点的产生位置置于图16的mark_chunk_group_time_stamp。进一步地，至mark_chunk_group_time_stamp的非连续点记录位置（非连续点的产生位置）可以是任意的的位置。因此，正常情况下，即使当产生了非连续点时（例如相继的标题的一个前导区或先前标题的结束部分）造成不显著影响的点在此得到记录。

进一步地，以此方式设定的title_info()(标题信息)被提供并存储在写入通道缓存器11中，随后，以预定的时序从其中读出并经过开关5、ECC电路4、RF和解调/调制电路3以及光头2被提供和记录在光盘1上。

随后操作进行到步骤S6，CPU21判定在标题中是否有不能连续再现的其他点，且当具有其他连续再现点时，操作返回到步骤S3，且随后重复执行一个处理。当判定在该标题中没有其他非连续再现点时，处理完成。

同时，当用户输入未确认产生非连续点时，操作进行到步骤S7，且CPU21判定在构成一个对象的两个标题之间的间隔是否能够通过改变标题在盘上的记录位置，而连续地得到再现。当判定所注意的部分可通过改变标题的记录位置而得到连续再现时，CPU21进行到步骤S8，改变标题的一部分在盘上的记录位置从而使所注意的部分能够连

续再现，随后操作进行到步骤S6。

在步骤S7，当判定标题的记录位置不能改变时，操作进行到步骤S5，如上所述，非连续点作为mark_type(标志类型)而得到记录，且非连续点的发生位置被记录在mark_chunk_group_time_stamp中。

在处理之后，执行在步骤S1指定的标题的记录操作。或者，这种处理可在记录了标题之后进行。

以下参见图49的流程图，描述在形成一个节目时记录非连续标记的情况下的处理。首先，在步骤S21，用户通过操作输入单元14指定包括在该节目中的标题。当进行指定时，在步骤S22，CPU21，借助一个再现开始点和一个再现结束点，指定步骤S21数据的标题中的一个所需部分。CPU21根据该指定形成一个播放项（图20）。

随后，在步骤S23，CPU21判定与前面的播放项的连续再现是否可以（这种判定也是根据播放项造成的处理时间与读取通道缓存器6的容量（容量的解码时间）之间的比较而进行的）。当不能进行连续再现时操作进行到步骤S24并控制OSD控制电路9并显示产生了非连续点的消息。对于这种消息，用户通过操作输入单元14输入是否确认非连续点的发生。此时，CPU21在步骤S25判定用户是否确认了非连续点的发生。当判定用户确认了非连续点的发生时，操作进行到步骤S26，并把播放项的seamless_connection_flag(无缝连接标记)置于0。如结合图46所描述的，该标记为0表明连续再现未得到保证。

随后，操作进行到步骤S27，且CPU21判定形成节目的处理是否已经完成。当判定该处理还没有完成时，操作返回到步骤S23并随后重复进行一个处理。当在步骤S27判定形成节目的处理已经完成时，处理完成。

同时，当在步骤S25判定用户确认了非连续点的发生时，操作进行到步骤S28且CPU21判定构成一个对象的流在光盘1上的记录位置（排列位置）是否需要改变或是否可以部分重新记录。当可以进行流的重新排列处理或部分重新记录时，操作进行到步骤S29且CPU21执行把记录位置改变到可以进行连续再现的位置的处理。进一

步地，在此情况下，在步骤S30，CPU21把播放项的seamless_connection_flag(无缝连接标记)置于1。如结合图46已经描述的，标记为1表示连续再现得到了保证。随后，操作进行到步骤S27，且随后执行一个处理。

当在步骤28判定不能进行重新排列处理或流的部分重新记录时，操作进行到步骤S26且执行与在其中用户确认了非连续点的发生的情况下类似的处理。

当在步骤S23判定可以进行与前面的播放项连续的再现时，操作进行到步骤S30，且播放项的seamless_connection_flag(无缝连接标记)立即被置于1。

以下描述当一个非连续标记如结合图48所示地被记录到标题上时进行的处理，其再现结合图50的流程图描述。首先，在步骤S41，CPU21读取指定的标题的一个非连续点，在步骤S42，CPU21开始再现标题的处理并在步骤S43判定其再现已经开始的标题是否完成。当该标题的再现还未完成时，操作进行到步骤S44，且CPU21判定非连续点标记(mark_chunk_group_time_stamp)代表的位置是否已经被再现，并当判定该位置还未被再现时返回到步骤S43。

当在步骤S44判定非连续点标记代表的位置已经再现时，操作进行到步骤S45，且CPU21判定一个播放机（在当前情况下，光盘设备）是否需要一个间隙以进行相继部分的连续再现。该判定是根据该时刻存储到读取通道缓存器6的数据量（解码器7解码该数据量所需的时间）来进行的。当该数据量足够时，判定不需要该间隙，且当该数据量不够时，判定需要该间隙。当光盘设备不需要间隙以进行相继部分的连续再现时（当数据被充分存储在读取通道缓存器6中时），操作返回到步骤S43，随后的一个处理被重复执行。

与此相对比，当在步骤S45判定光盘设备需要用于进行相继部分的再现的间隙时（当数据未充分地存储在读取通道缓存器6中时），操作进行到步骤S46且CPU21产生该间隙。即，虽然CPU21继续把数据不变地写入读取通道缓存器6，CPU21进行中断以从解码器7读取数

据并增大读取通道缓存器6的数据量。进一步地，操作进行到步骤S47，判定相继部分的连续再现是否可能，并在不可能时返回到系统46并继续进行产生间隙的处理。

以此方式，当预定的数据量被写入读取通道缓存器6且在步骤S47判定其中能够执行相继部分的连续再现的状态出现时，操作进行到步骤S48，且CPU21重新继续进行再现（重新启动解码器7的解码），且随后操作返回到步骤S43且随后的处理得到重复执行。

同时，当在步骤S43判定标题的连续再现已经完成时，操作进行到步骤S49且CPU21判定所要相继地再现的标题是否出现。当没有相继再现的标题时，处理完成。与此相对比，当判定有相继再现的标题时，操作进行到步骤S50，且CPU21判定在前一个标题的结束区中是否有一个非连续标记。当判定在前面再现的标题的结束部分中有一个非连续标记时，操作进行到步骤S51，且CPU21判定播放机（光盘设备）是否需要一个间隙以再现相继的标题。当判定需要该间隙时，操作进行到步骤S52，CPU21执行产生一个间隙的处理，在步骤S53判定是否可以再现相继的标题，当不可能时返回到步骤S52并重复执行间隙产生处理。

当在步骤S53判定可以进行相继的标题的连续再现时，操作返回到步骤S41，且随后的处理得到重复执行。

当在步骤S50判定在前一个标题的最后区域中没有非连续标记时，操作返回到步骤S41，且随后的处理得到执行。进一步地，即使当在前一个标题的结束区中有非连续标记时，在其中在步骤S51判定光盘设备不需要一个间隙以再现相继的标题时（例如，当光盘设备的读取通道缓存器6的容量足够大时等等），操作返回到步骤S41，且随后的处理得到执行。

以下，结合图51的流程图，描述如图49显示的当形成一个节目时命令对节目的再现的处理。

首先，在步骤S61，CPU21再现构成指定的节目的第一播放项。在步骤S62，CPU21判定播放项的再现是否已经完成，且当播放项还

未完成时，CPU21等候再现的完成。

当在步骤S62判定播放项的再现已经完成时，CPU21进行到步骤S63并判定是否有一个相继的播放项。当判定有相继播放项时，操作进行到步骤S64且CPU21判定相继的播放项的seamless_connection_flag(无缝连接标记)是否为0。当判定该标记是0时（当无缝再现无保证时），操作进行到步骤S65，且CPU21判定光盘设备是否需要一个间隙以进行相继的播放项的再现。当需要该间隙时，操作进行到步骤S66并执行产生该间隙的处理。进一步地，在步骤S67，该操作判定其中能够再现相继的播放项的一种状态是否出现，并在其中能够再现相继的播放项的状态还没有出现的情况下返回步骤S66并重复执行间隙产生处理。

当在步骤S67判定其中相继的播放项能够再现的状态出现时（当足够的数据量被写入读取通道缓存器6时），操作进行到步骤S68且CPU21开始再现相继的播放项的处理。随后，操作返回到步骤S62，且随后的处理得到重复执行。

当在步骤S64判定相继的播放项的seamless_connection_flag(无缝连接标记)不是0时（当它被判定为1）或者当在步骤S65判定光盘设备不需要用于再现相继的播放项的间隙时，在步骤S66和步骤S67的处理被跳过且操作进行到步骤S68。进一步地，开始再现相继的播放项，且随后操作返回到步骤S62。

虽然在上述描述中，对于其中本发明被用于一种光盘设备的情况的一个例子进行了描述，本发明也可应用于其中信息被记录到其他记录介质上或从该记录介质再现的情况。

进一步地，作为用于向用户提供执行上述处理的计算机程序的程序提供介质，采用了除了磁盘、CD-ROM、固态存储器等之外的诸如网络或卫星的通信介质。

如上所述，根据本发明的记录/再现设备、记录/再现方法和程序提供介质，判定数据是否能够与已经记录的数据相连续地得到再现，且根据这种判定的结果执行与连续再现有关的处理，因而能够实现能

够保证兼容性的记录介质。

根据本发明的记录/再现设备、记录/再现方法和程序提供介质，根据从一种记录介质提取的连续再现信息，添加了一个间隙，因而不论记录/再现设备的缓存器的容量如何改变，兼容性都能够得到保证。因此，可以限制用户对设备故障的错误识别。

图 1

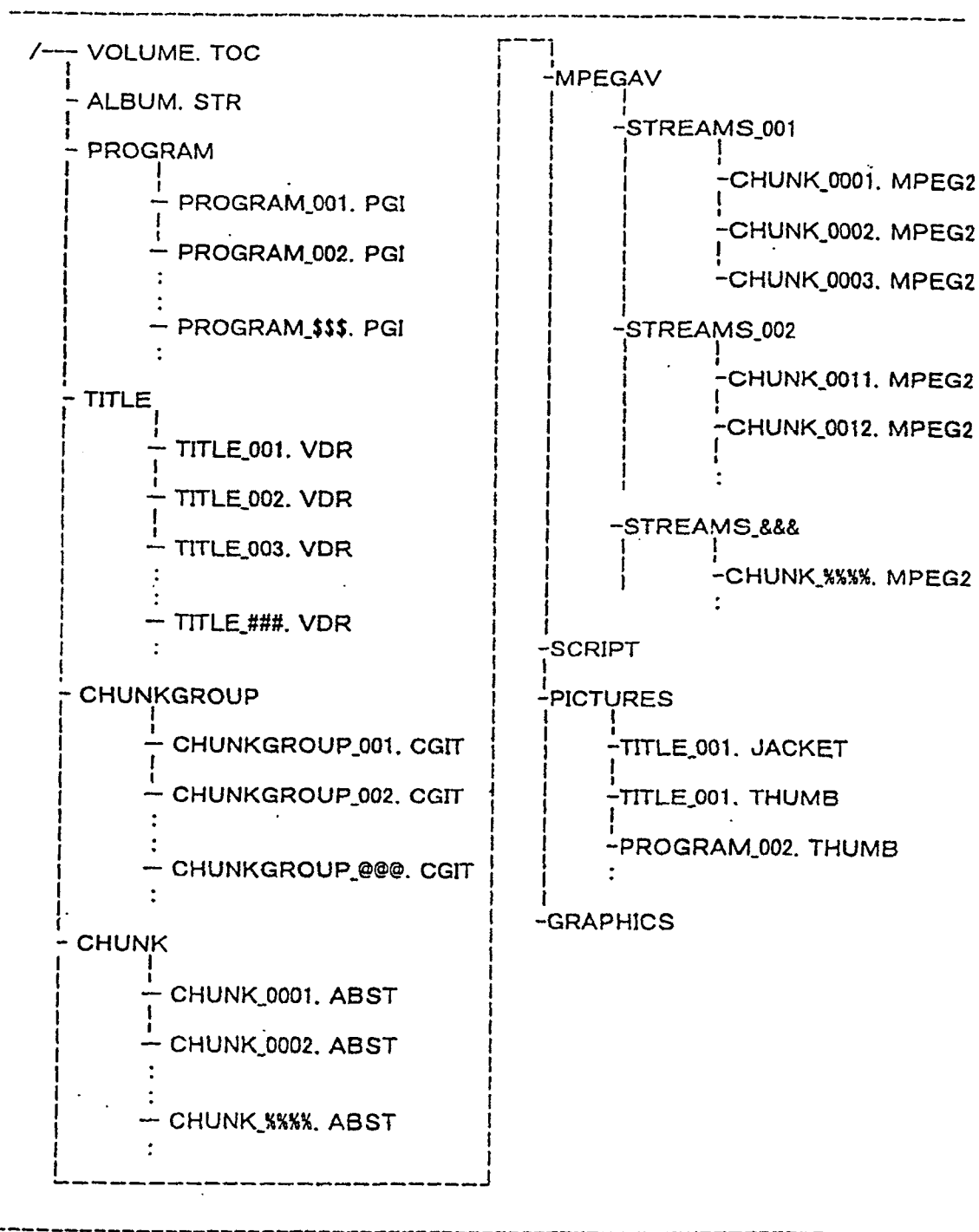


图 2

语法	位数	助记符
VOLUME.TOC{		
file_type_id	8*16	char[16]
volum_information()		
text_block()		
}		

图 3

语法	位数	助记符
volume_information() {		
volume_attribute()		
resume()		
volume_rating()		
write_protect()		
play_protect()		
recording_timer()		
}		

图 4

语法	位数	助记符
volume_attribute() {		
volume_attribute_length	32	uimsdf
vdr_version	4*4	bcd
reserved	6	bslbf
title_playback_mode_flag	1	bslbf
program_playback_mode_flag	1	bslbf
volume_play_time()	4*8	bcd
update_time_count()	32	uimsbf
maker_id	8*16	char[16]
model_code	8*16	char[16]
POSID	32	bslbf
}		

图 5

语法	位数	助记符
resume ()[		
resume_length	32	uimbsf
reserved //用于字节对齐	3	bslbf
resume_switch	1	bit
reserved	4	bslbf
number_of_records	4	uimbsf
reserved //用于字节对齐	7	bslbf
resume_auto_execute_time_flag	1	bit
resume_auto_execute_time_()	4*14	bcd
reserved	4	bslbf
resume_auto_execute_record_number	4	uimbsf
for(i=0;i<number_of_records;i++){		
resume_mode_flag	4	bslbf
object_type	4	bslbf
linked_record_number	4	uimbsf
number_of_times	16	uimbsf
resume_updated_time_()	4*14	bcd
switch(object_type){		
case title:		
title_number	16	uimbsf
title_local_time_stamp	64	uimbsf
break;		
case program:		
program_number	16	uimbsf
program_local_time_stamp	64	uimbsf
break;		
case program_bind		
program_bind_number	16	uimbsf
program_ordre	16	uimbsf
program_number	16	uimbsf
program_local_time_stamp	64	uimbsf
break;		
case play_item:		
play_item_number	16	uimbsf
play_item_local_time_stamp	64	uimbsf
break		
}		
}		
}		

图 6

语法	位数	助记符
volume_rating() volume_rating_length	32	uimsbf
reserved	6	bslbf
volume_rating_type	2	bslbf
 volume_rating_password	128	bslbf
switch(rating_type) { case age_limited		
country_code_for_rating	32	bslbf
for(i=0;i<32;i++){ rating_bit_for_ago_limited	1	bslbf
break;		
case CARA:		
CARA_category	4	bslbf
reserved	4	bslbf
reserved	16	bslbf
break;		
case RSAC:		
RSAC_category	4	bslbf
level	4	bslbf
reserved	16	bslbf
break;		
}		
}		

图 7

语法	位数	助记符
write_protect() write_protect_length	32	uimsbf
volume_write_protect_level	4	uimsbf
password_enable_flag	1	bslbf
apped_only_flag	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
 password_for_volume_write_protect()	128	bslbf
reserved	8	bslbf
write_protect_set_time()	56	bcd
reserved	8	bslbf
write_protect_expiration_time()	56	bcd
number_of_times	16	uimsbf
}		

图 8

语法	位数	助记符
play_protect() {		
play_protect_length	32	uimsbf
volume_play_protect_flag	2	bslbf
reserved	2	bslbf
password_enable_flag	1	bslbf
reserved	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_play_protect	128	bslbf
rederved	8	bslbf
play_protect_set_time()	56	bcd
reserved	8	bslbf
play_protect_expiration_time()	56	bcd
number_of_times	16	uimsbf
}		

图 9

语法	位数	助记符
recording_timer() {		
recording_timer_length		
recording_timer_flag		
number_of_entry		
for(i=0;i<number_of_entry;i++){		
date_and_time		
channel		
program		
:		
}		
}		

图 10

语法	位数	助记符
text_block(){	32	uimsbf
text_block_length	8	uimsbf
number_of_language_sets	16	uimsbf
number_of_text_items		
for(i=0;i<number_of_language_sets;i++){		
language_set()		
}		
for(i=0;i<number_of_text_items;i++){		
text_item()		
}		
}		

图 11

语法	位数	助记符
language_set(){		
reserved	8	bslbf
language_code	24	bslbf
character_set_type	8	bslbf
number_of_language_set_names	8	uimsbf
for(i=0;i<number_of_language_set_names;i++){		
character_set_type	8	bslbf
language_set_name_length	8	uimsbf
language_set_name	8*language_set_name_length	bslbf
}		
}		

图12

语法	位数	助记符
text_item(){		
text_item_length	16	uimsbf
text_item_id	16	uimsbf
text_item_sub_id	16	uimsbf
flags	8	bslbf
number_of_used_language_sets	8	uimsbf
// 对每个语言设置的循环		
for(i=0;i<number_of_used_language_sets;i++){		
language_set_id	8	uimsbf
reserved	4	bslbf
text_string_length	16	uimsbf
text_string	8*text_string_length	bslbf
bitmap()		
}		
stuffing_bytes	8*n	bslbf
}		

图13

语法	位数	助记符
ALBUM. STR{		
file_type_id	8*16	char[16]
album()		
text_block()		
}		

图14

语法	位数	助记符
album () {		
album_length	32	uimbsf
reserved	6	bslbf
volume_status	1	bslbf
if(volume_status=="1b"){		
chief_volume_flag	1	bslbf
} else {		
reserved	1	"0"
}		
if(volume_status=="1b"){		
if(chief_volume_flag=="1b"){		
reserved	6	bslbf
album_type	2	bslbf
album_id	128	bslbf
number_of_discs_in_album	16	uimbsf
number_of_volumes_in_album	16	uimbsf
for(i=0;i<number_of_volumes_in_album;i++){		
disc_id_for_album_member	128	bslbf
volume_id_for_album_member	128	bslbf
title_offset_number	16	uimbsf
}		
reserved_for_program_bind	8	bslbf
number_of_program_binds	8	uimbsf
for(i=0;i<number_of_program_binds;i++){		
number_of_programs_in_this_program_bind	16	uimbsf
for(i=0;i<number_of_programs_in_this_program_bind;i++){		
disc_id_for_program_bind_member	128	uimbsf
volume_id_for_program_bind_member	128	uimbsf
program_number	16	uimbsf
}		
}		
} else { // chief_volume_flag=="0b"		
chief_disc_id	128	uimbsf
chief_volume_id	128	uimbsf
(album_id	128	bslbf
}		
}		
}		

图 15

语法	位数	助记符
TITLE_###. VDR{ file_type_id title_info() text_block() }	8*16	char[16]

图 16

语法	位数	助记符
title_info() { title_info_length flags_for_title cgit_file_id title_start_chunk_group_time_stamp title_end_chunk_group_time_stamp title_playback_time() reserved number_of_marks for(i=0;i<number_of_marks;i++){ reserved mark_type mark_chunk_group_time_stamp } stuffing_bytes }	 32 32 16 64 64 32 32 16 4 4 64 8*n	 uimbsf bslbf uimbsf uimbsf uimbsf bcd bslbf uimbsf bslbf bslbf uimbsf bslbf

图 17

语法	位数	助记符
PROGRAM_\$\$\$. PGI file_type_id program() text_block() }	8*16	char[16]

图 18

语法	位数	助记符
program() {		
program_length	32	uimsbf
flags_for_program	32	bslbf
program_status	4	bslbf
program_playback_time()	32	bcd
reserved	32	bslbf
number_of_play_sequences	16	uimsbf
for(j=0;j<number_of_play_sequence;j++){		
number_of_play_lists	16	uimsbf
for(k=0;k<number_of_play_lists;k++){		
play_list_start_time_stamp_offset	64	uimsbf
play_list(k)		
}		
}		
stuffing_bytes	8*n	bslbf
}		

图 19

语法	位数	助记符
play_list() {		
// 播放清单中播放项的重放序列		
number_of_play_items	16	uimsbf
for(k=0;k<number_of_play_items;k++){		
play_item_number	16	uimsbf
reserved	31	bslbf
seamless_connection_flag	1	bslbf
}		
//play_item_table		
for(PIN=t; PIN<=number_of_play_items_in_program; PIN++){		
play_item()		
}		
}		

图 20

语法	位数	助记符
play_item(){		
play_item_length	32	uimbsbf
play_item_type	8	bslbf
play_mode	8	bslbf
total_playback_time ()	32	bcd
menu_item_number	16	uimbsbf
return_item_number	16	uimbsbf
next_item_number	16	uimbsbf
prev_item_number	16	uimbsbf
if(play_item_type="0000b") {		
// 对于一个片断的播放项		
title_number	16	uimbsbf
// IN 点		
item_start_time_stamp	64	uimbsbf
// OUT 点		
item_end_time_stamp	64	uimbsbf
}		
}		

图 21

语法	位数	助记符
CHUNKGROUP_###.CGIT {		
file_type_id	8*16	char[16]
chunkgroup_time_base_flags	32	bslbf
chunkgroup_time_base_offset	64	uimbsbf
chunk_connection_info()		
text_block()		
}		

图 22

语法	位数	助记符
chunk_connection_info() {	32	uimsbf
chunk_connection_info_length	16	bslbf
reserved	16	uimsbf
number_of_chunks	8	bslbf
chunk_sync_play_flag		
// 数据块信息文件清单		
for(i=0;i<number_of_chunks;i++){		
chunk_arrangement_info()		
}		
}		

图 23

语法	位数	助记符
chunk_arrangement_info() {		
chunk_arrangement_info_length	32	uimsbf
chunk_info_file_id	16	bslbf
reserved	5	bslbf
chunk_switch_stream_id	16	bslbf
presentation_start_cg_time_count	64	uimsbf
presentation_end_cg_time_count	64	uimsbf
reserved	4	bslbf
chunk_time_count_type	4	bslbf
number_of_start_original_time_count_extension	8	uimsbf
number_of_end_original_time_count_extension	8	uimsbf
//表示开始位置和时间		
presentation_start_original_time_count	64	uimsbf
presentation_end_original_time_count	64	uimsbf
for(j=0;j<number_of_start_original_time_count_extension;j++){		
tc_ext_attributes	16	bslbf
start_original_time_count_extension	64	uimsbf
}		
//表示结束位置和时间		
for(k=0;k<number_of_end_original_time_count_extension;k++){		
tc_ext_attributes	16	bslbf
end_original_time_count_extension	64	uimsbf
}		
transition_info()		
}		

图 24

语法	位数	助记符
CHUNK_%%%. ABST{		
file_type_id	8*16	char[16]
reserved	4	bslbf
chunk_file_id	16	uimsbf
info_type	4	bslbf
// stream_info ()		
if(info_type=="MPEG2_System_TS"){		
number_of_programs	8	uimsbf
else {		
number_of_programs	8	"0000 0001"
}		
for(i=0;i<number_of_programs;i++){		
number_of_streams	8	uimsbf
for(i=0;i<number_of_streams;i++){		
stream_identifier	8	bslbf
// 片类型信息		
reserved	4	bslbf
slot_unit_type	4	bslbf
if(slot_unit_type=="time_stamp"){		
slot_time_length	32	uimsbf
} else {		
reserved	32	bslbf
}		
number_of_slots	32	uimsbf
reserved	4	bslbf
switch(info_type){		
case MPEG1_System:		
case MPEG2_System_PS:		
case MPEG2_System_TS:		
case video_elementary_stream		
number_of_i_pictures_in_a_slot	4	uimsbf
break		
default:		
reserved	4	bslbf
break;		
}		
// 流属性		
ES_attribute()		
}		
// 片信息循环		
for(i=0;i<number_of_streams;i++){		
for(i=0;i<number_of_slots;i++){		
slot_info ()		
}		
}		
}		

图 25

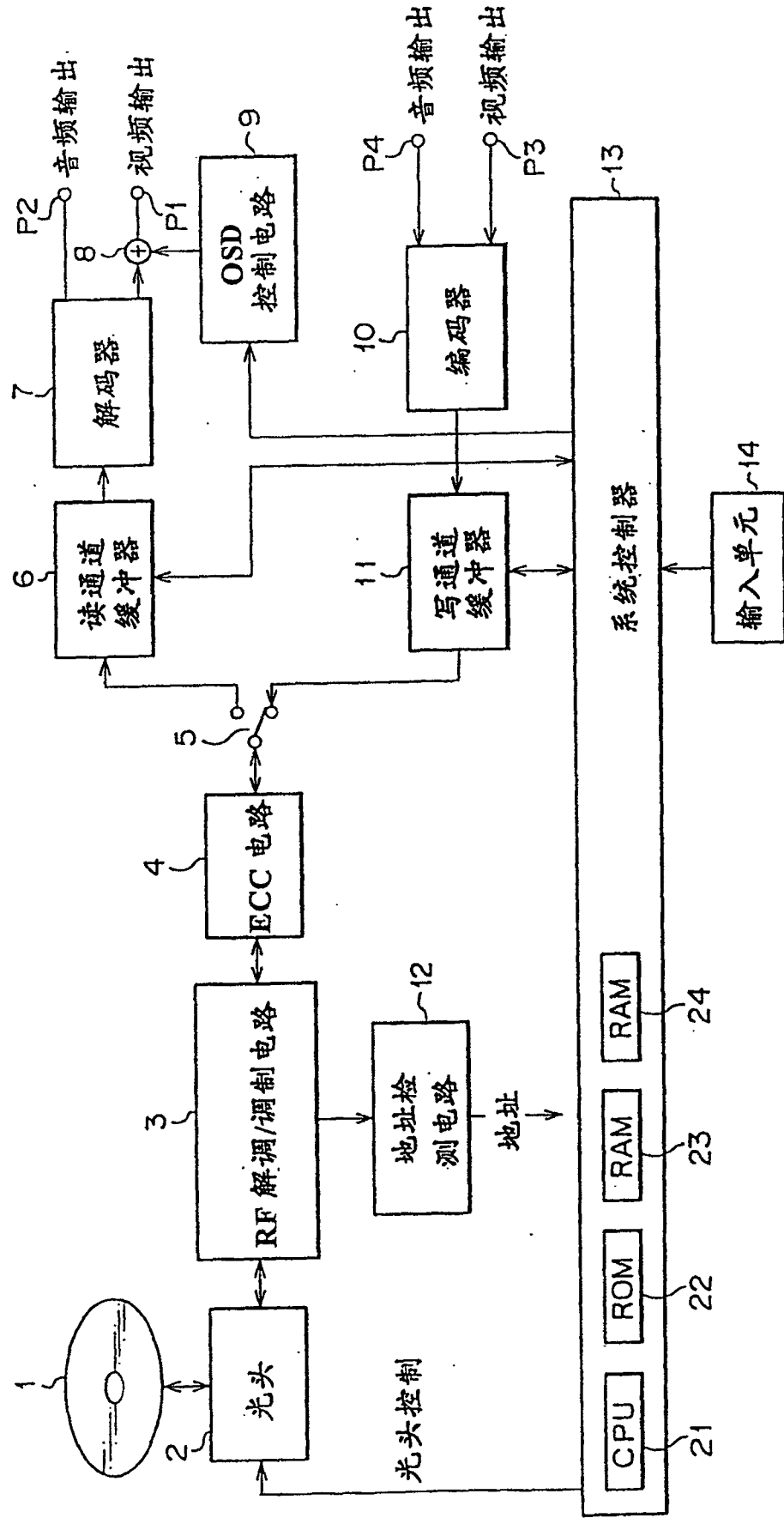


图 26

```

/--- VOLUME. TOC
  --- ALBUM. STR
  --- PROGRAM
    |   ---PROGRAM_001. PGI
  --- TITLE
    |   ---TITLE_001. VDR
    |   ---TITLE_002. VDR
    |   ---TITLE_003. VDR
  --- CHUNKGROUP
    |   ---CHUNKGROUP_001. CGIT
    |   ---CHUNKGROUP_002. CGIT
  --- CHUNK
    |   ---CHUNK_0001. ABST
    |   ---CHUNK_0011. ABST
    |   ---CHUNK_0012. ABST
  --- MPEGAV
    |   ---STREAMS_001
    |     |   ---CHUNK_0001. MPEG2
    |   ---STREAMS_002
    |     |   ---CHUNK_0011. MPEG2
    |     |   ---CHUNK_0012. MPEG2

```

图 27

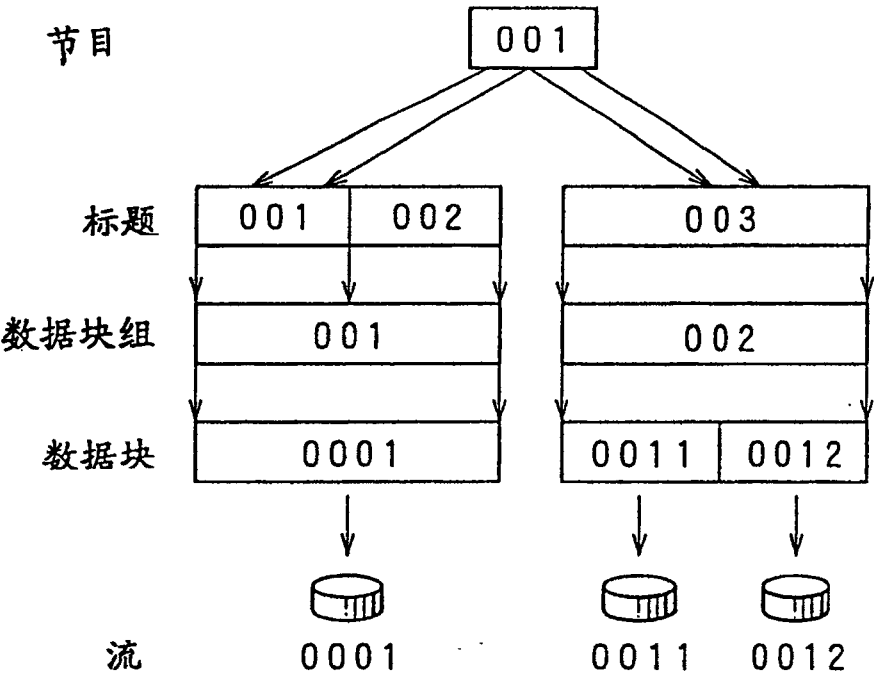


图 28

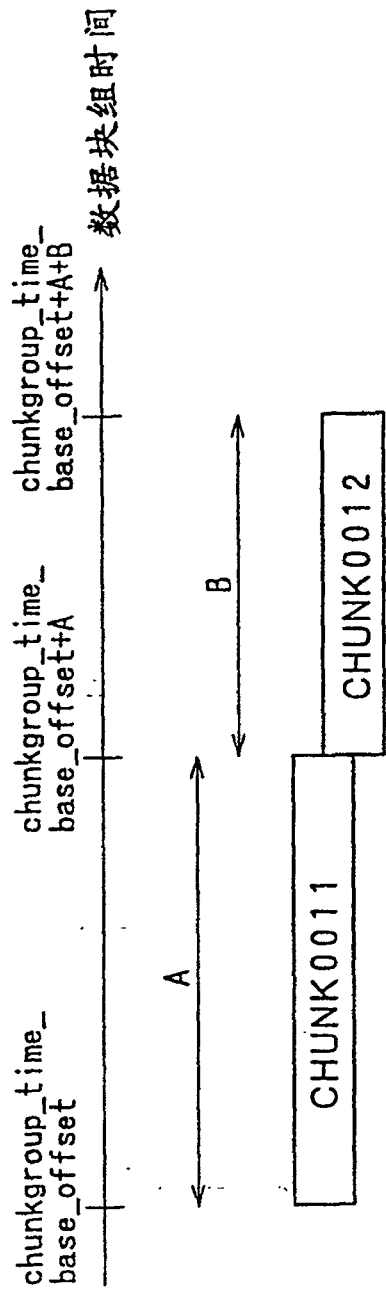


图 31

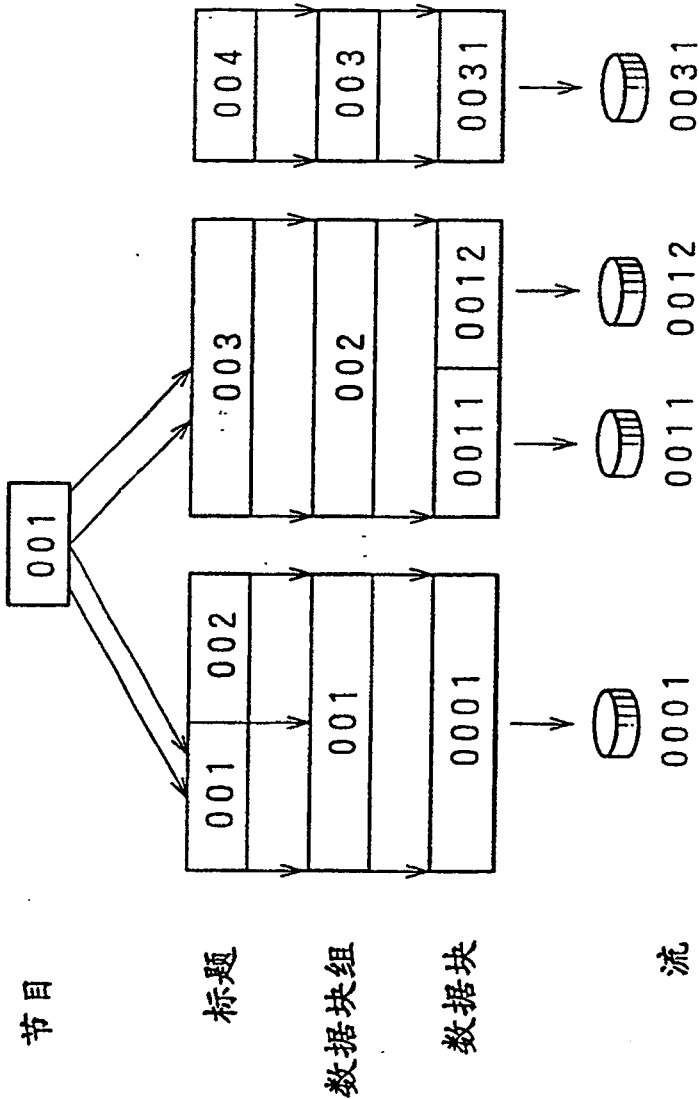


图 32

```

/ --- MPEGAV
    |
    |--- STREAMS_002
    |
    |--- CHUNK 0031.MPEG2
  
```

图 33

```

/ --- VOLUME. TOC
    |
    |--- ALBUM. STR
    |
    |--- PROGRAM
    |   |
    |   |--- PROGRAM_001. PGI
    |
    |--- TITLE
    |   |
    |   |--- TITLE_001. VDR
    |   |--- TITLE_002. VDR
    |   |--- TITLE_003. VDR
    |   |--- TITLE_004. VDR*
    |
    |--- CHUNKGROUP
    |   |
    |   |--- CHUNKGROUP_001. CGIT
    |   |--- CHUNKGROUP_002. CGIT
    |
    |--- CHUNK
    |   |
    |   |--- CHUNK_0001. ABST
    |   |--- CHUNK_0011. ABST
    |   |--- CHUNK_0012. ABST
    |   |--- CHUNK_0031. ABST*
    |
    |--- MPEGAV
    |   |
    |   |--- STREAMS_0001
    |   |   |
    |   |   |--- CHUNK_0001. MPEG2
    |   |
    |   |--- STREAMS_0002
    |   |   |
    |   |   |--- CHUNK_0011. MPEG2
    |   |   |--- CHUNK_0012. MPEG2
    |   |   |--- CHUNK_0031. MPEG2*
  
```

图 34

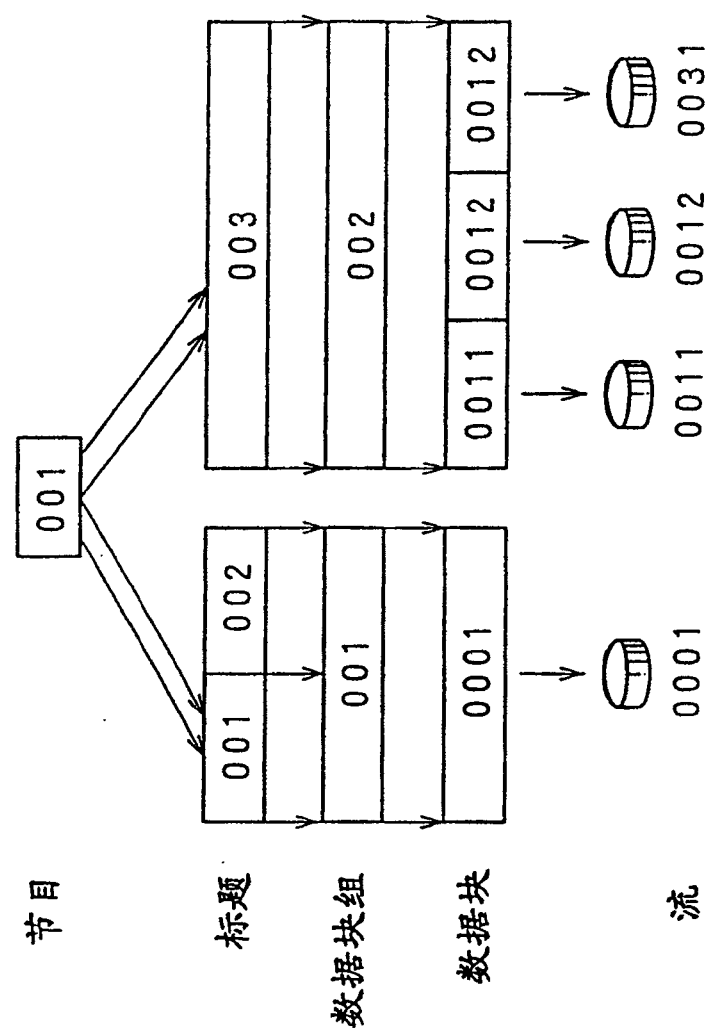


图35

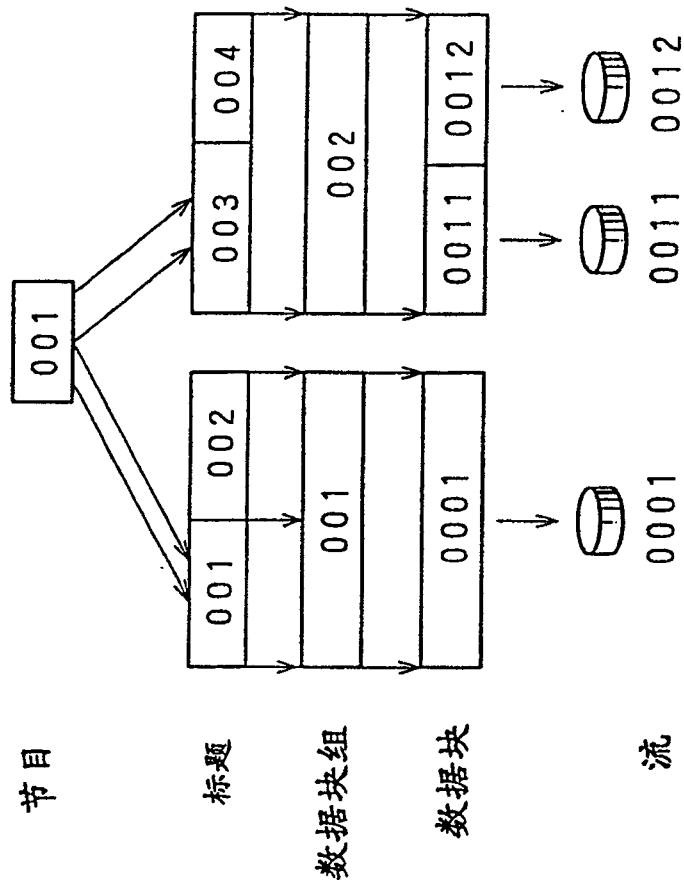


图 36

域名	值
file_type_id	"TITLE_INFO_00000"

图 37

标记类型	含意
0000b	指标类型 1
0001b	指标类型 2
0010b .. 00111b	指标类型 3...指标类型 8
1000b	跳过 IN
1001b	跳过 OUT
1010b	跳到标题结束或下一标题
1011b	场景改变
1100b	音频关闭
1101b	音频峰
1110b	图象静止
1111b	保留

指标：标题中的直接进入点

跳过 IN：标题跳过的开始点

跳过 OUT：标题跳过的结束点

图 38

域名	值
file_type_id	"CHUNKGROUP_0000"

图 39

数据块同步播放标志	含意
0b	同时只播放一个数据块
1b	需要同时播多个数据块

图 40

原始时间计数类型	含意
0000	MPEG2 系统时间标志 (90KHz)
0001	SMPTE 时间码
0010	场 (525/60)(59.94Hz)
0011	场 (625/50)
0100	帧 (525/60)(29.97Hz)
0101	帧 (625/50)
0110	下落帧时间码 (525/60)
0111	非下落帧时间码 (525/60)
1000	60Hz
1001	50Hz
1010	24Hz
1011	秒
1100 .. 1111	保留

图 41

域名	值
field_type_id	"STREAM_INFO_0000"

图 42

信息类型	含意
0000	MPEG2_System_PS
0001	MPEG2_System_TS
0010	MPEG2_System_PES
0011	video_elementary_stream
0100	elementary_stream_except_video
0101	MPEG1_System_stream
0110	保留
0111	保留
1000	Consumer_DVC
1001 .. 1111	保留

图 43

片单位类型	含意
0000b	'time_stamp' : 时间标志值
0001b	'GOP' : 一个 GOP (图象组)
0010b	'audio_frame' : 一个音频帧
0011b .. 1111b	保留

图 44

域名	值
file_type_id	"PROGRAM_INFO_000"

图 45

节目状态	含意
0000b	无
0001b	原始
0010b	复制
0011b	预览
0100b	重排
0101b	暂时
0110b	完成
0111b	打破
1000b	编辑
1001b	备份
1010b .. 1111b	保留

图 46

无缝连接标志	值
0b	与前面播放项的无缝重放连接无保证或未知
1b	与前面的播放项的无缝重放连接有保证

图 47

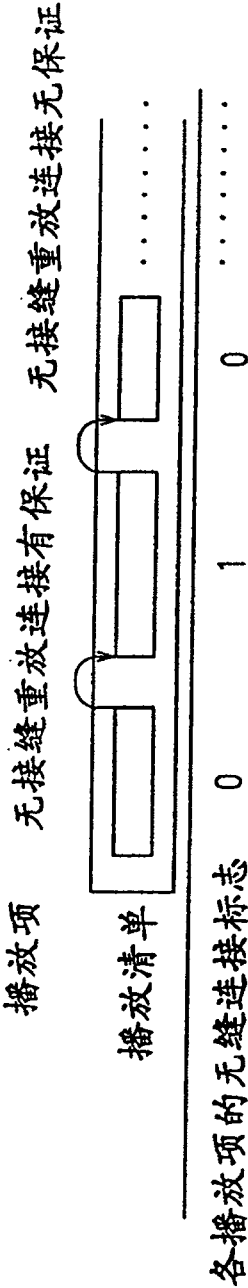


图 48

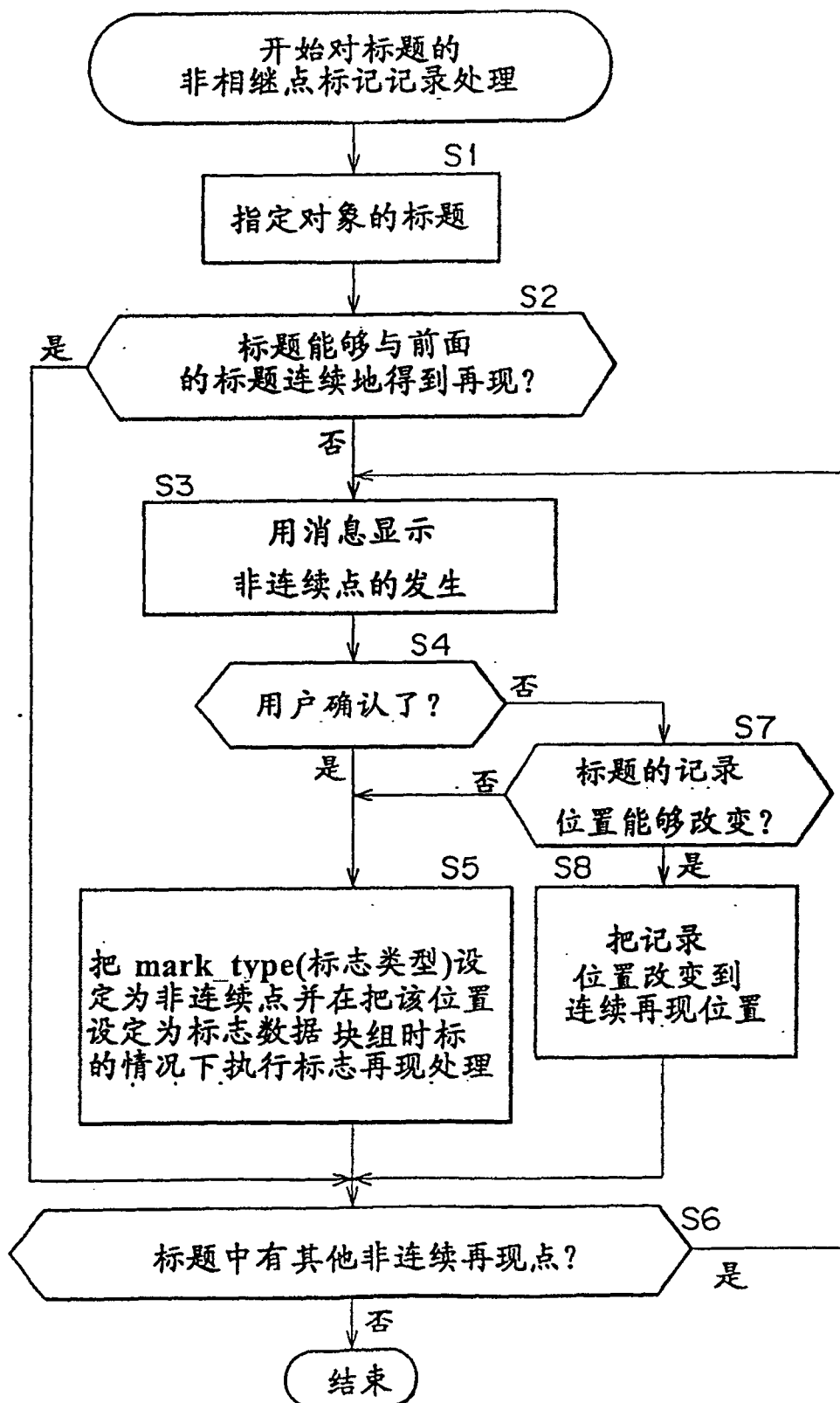


图 49

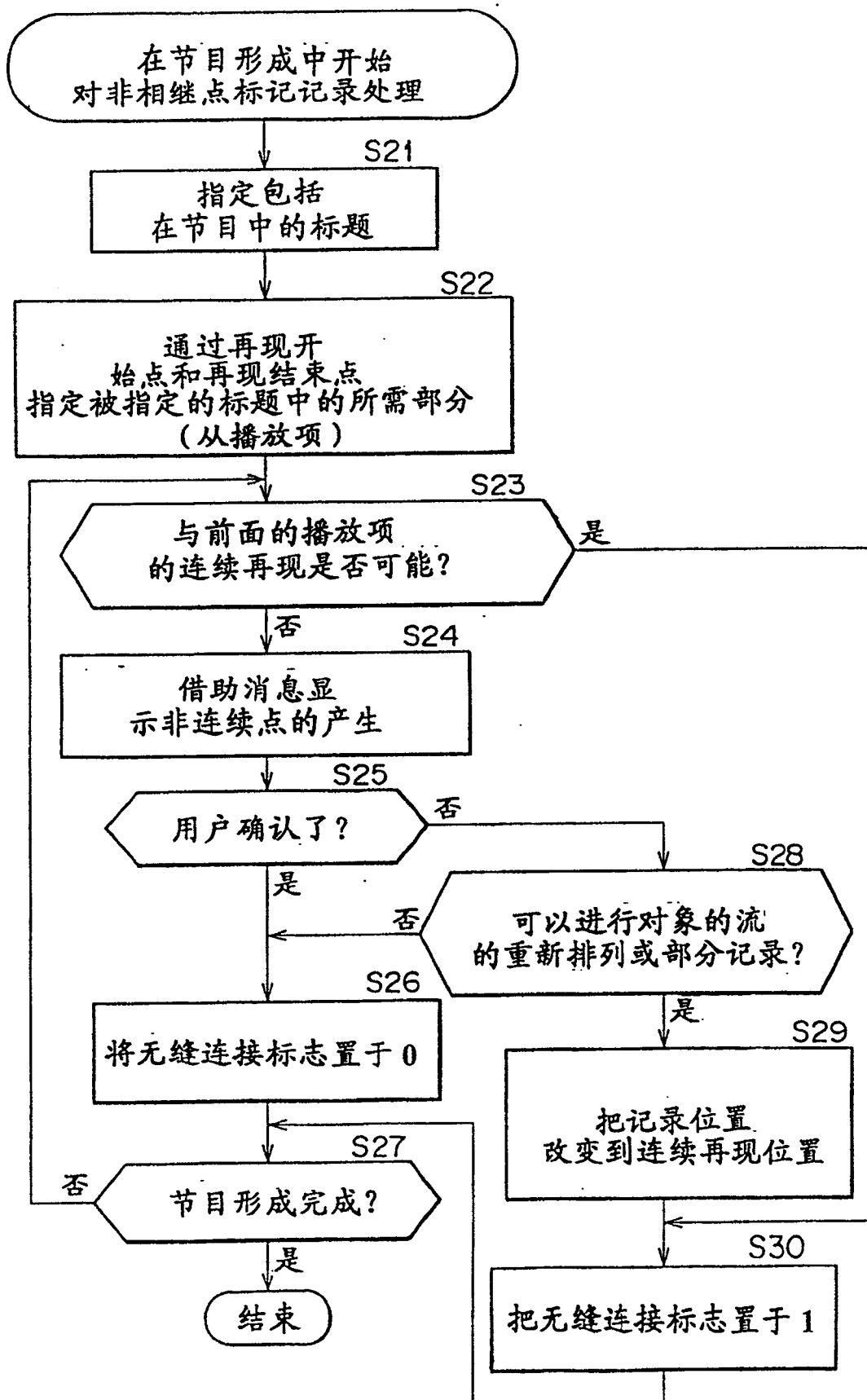


图 50

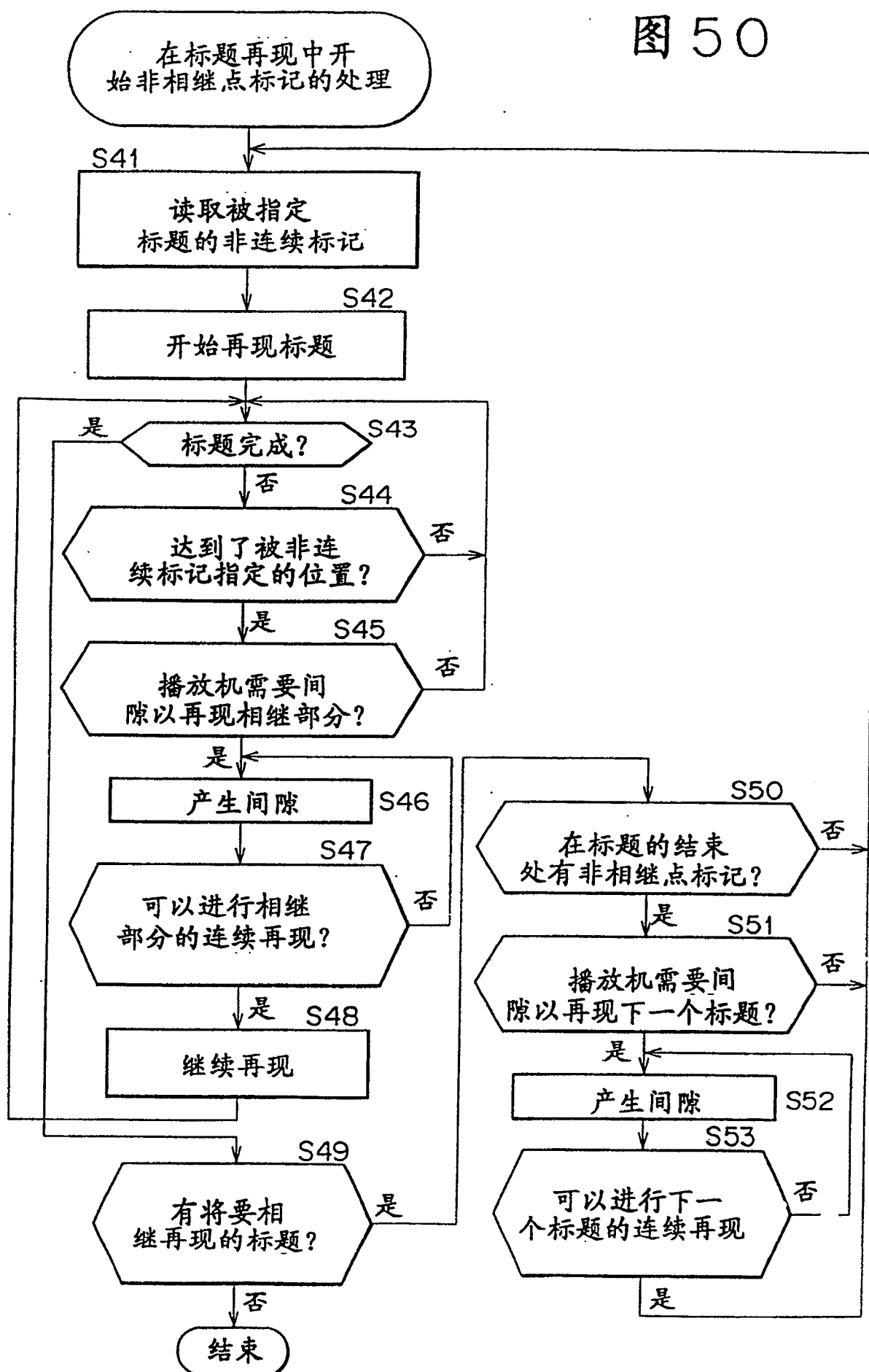


图 51

