

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04N 5/92
G11B 20/10

[12] 发明专利申请公开说明书

[21] 申请号 01801571.9

[43] 公开日 2002 年 11 月 20 日

[11] 公开号 CN 1381137A

[22] 申请日 2001.4.20 [21] 申请号 01801571.9

[30] 优先权

[32]2000.4.21 [33]JP [31]183771/00

[32]2000.9.7 [33]JP [31]271552/00

[86] 国际申请 PCT/JP01/03415 2001.4.20

[87] 国际公布 WO01/82606 日 2001.11.1

[85] 进入国家阶段日期 2002.2.1

[71] 申请人 索尼公司

地址 日本东京都

[72] 发明人 加藤元树 浜田俊也

[74] 专利代理机构 北京市柳沈律师事务所

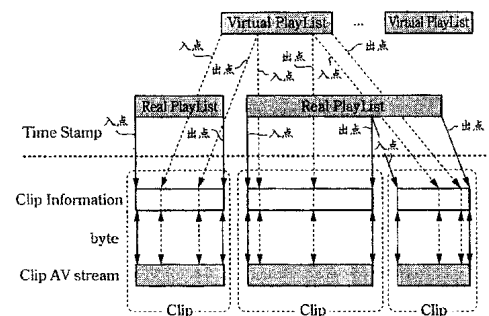
代理人 马莹 邵亚丽

权利要求书 10 页 说明书 58 页 附图 100 页

[54] 发明名称 信息处理方法和装置、程序和记录介质

[57] 摘要

在 Playlist() 中描述 CPI_type。CPI_type 包括 EP_map 和 TU_map 类型。如果能分析 I 图像的位置,则使用 EP_map,否则使用 TU_map。从而能共同管理分析 I 图像位置之后记录的 AV 流数据和不分析 I 图像位置记录的 AV 流数据。



ISSN 1008-4274

1. 一种用于在记录介质上记录 AV 流数据的信息处理装置，包括：
第一生成装置，用于生成描述呈现时间戳和对应访问单元的所述 AV 流
5 数据中地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时
间戳和对应传输包的所述 AV 流数据中地址之间对应关系的第二表；
选择装置，用于根据记录方法选择所述第一表和所述第二表中的一个；和
第一记录装置，用于在所述记录介质上与所述 AV 流数据一起记录所选
表。
- 10 2. 如权利要求 1 所述的信息处理装置，其中，所述第一表为 EP_map；
并且所述第二表为 TU_map。
3. 如权利要求 1 所述的信息处理装置，其中，所述选择装置在非认知
记录的情况下选择所述第二表。
4. 如权利要求 1 所述的信息处理装置，其中，所述选择装置在自编码记
15 录的情况下选择所述第一表。
5. 如权利要求 1 所述的信息处理装置，其中，所述选择装置在认知记
录的情况下选择所述第一表。
6. 如权利要求 1 所述的信息处理装置进一步包括：
第二生成装置，用于生成指定再现所述 AV 流数据的再现指定信息；和
20 第二记录装置，用于在所述记录介质上记录指定由所述第二生成装置生
成的所述再现指定信息；
所述再现指定信息包括用于指定是根据呈现时间还是根据到达时间表示
所述 AV 流数据再现域的时间信息的标识信息。
7. 如权利要求 1 所述的信息处理装置，其中，如果所述第一表与所述
25 AV 流数据一起进行记录，所述再现指定信息根据呈现时间表示所述 AV 流
数据再现范围的时间信息；并且其中
如果所述第二表与所述 AV 流数据一起进行记录，所述再现指定信息根
据到达时间表示所述 AV 流数据再现范围的时间信息。
8. 一种用于在记录介质上记录 AV 流数据的信息处理方法，包括：
30 生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的
地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和

对应传输包的所述 AV 流数据中地址之间对应关系的第二表；

选择步骤，根据记录方法选择所述第一表和所述第二表中的一个；和
记录步骤，在所述记录介质上与所述 AV 流数据一起记录所选表。

9. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质
5 上记录 AV 流数据的信息处理装置，所述程序包括：

生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的
地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和
对应传输包的所述 AV 流数据中地址之间对应关系的第二表；

- 10 选择步骤，根据记录方法选择所述第一表和所述第二表中的一个；和
记录步骤，在所述记录介质上与所述 AV 流数据一起记录所选表。

10. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算
机执行如下步骤的程序：

- 15 生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中地
址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和对
应传输包的所述 AV 流数据中地址之间对应关系的第二表；

选择步骤，根据记录方法选择所述第一表和所述第二表中的一个；和
记录步骤，在所述记录介质上与所述 AV 流数据一起记录所选表。

11. 一种用于从记录介质再现 AV 流数据的信息处理装置，包括：

- 20 再现装置，用于从其中根据记录方法记录有第一表或第二表的记录介质
再现所述第一表和所述第二表中的一个，所述第一表描述呈现时间戳和对应访问
单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自传输包
到达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关
系；和

控制装置，用于根据所再现的表控制所述 AV 流数据的输出。

- 25 12. 一种用于从记录介质再现 AV 流数据的信息处理方法，包括：

再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现
所述第一表和所述第二表中的一个，所述第一表描述呈现时间戳和对应访问单
元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自传输包到
达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系；

- 30 和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

13.一种其中记录有一个计算机可读程序的记录介质，用于在记录介质上记录 AV 流数据的信息处理装置，所述程序包括：

再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自传输包到达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系；
5 和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

14. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：

再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自传输包到达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系；
15 和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

15. 一种其中根据记录方法记录有第一表和第二个表的记录介质，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系。
20

16. 一种用于在记录介质上记录 AV 流数据的信息处理装置，包括：

生成装置，用于生成由表示主再现路径的第一信息和表示次再现路径的第二信息构成的再现指定信息；和
记录装置，用于在所述记录介质上记录所述 AV 流数据和再现指定信息。
25 息。

17. 如权利要求 16 所述的信息处理装置，其中，所述主再现路径为用于音频数据后期记录的路径。

18. 如权利要求 16 所述的信息处理装置，其中，所述第一信息为 main_path 并且所述第二信息为 Sub-path。

30 19. 如权利要求 16 所述的信息处理装置，其中，所述第二信息包括表示所述次再现信息类型的类型信息；

- 用所述次再现路径引用的所述 AV 流文件名；
所述次再现路径的所述 AV 流的入点和出点；和
在所述主路径的时间轴上所述再现路径的入点开始同步的所述主路径时间。
- 5 20.一种信息处理方法，用于在记录介质上记录 AV 流数据的信息处理装置，包括：
 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
- 10 21.一种其中记录有一个计算机可读程序的记录介质，用于在记录介质上记录 AV 流数据的信息处理装置，所述程序包括：
 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
15 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
- 22.一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：
 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
20 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
- 23.一种用于从记录介质再现 AV 流数据的信息处理装置，包括：
 再现装置，用于从所述记录介质再现由表示主再现路径的第一信息和表示与
25 所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
 控制装置，用于根据被再现的再现指定信息控制所述 AV 流数据的输出。
- 24.一种用于从记录介质再现 AV 流数据的信息处理方法，包括：
30 再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与
 所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和

控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。

25. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质上记录 AV 流数据的信息处理装置，所述程序包括：

再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与
5 所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。

26. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机
执行如下步骤的程序：

再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与
10 所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。

27. 一种其中记录有 AV 流数据的记录介质，其中，所述记录介质其中记
录有由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再
现路径的第二信息构成的再现指定信息。

信息处理方法和装置、程序和记录介质

5

技术领域

本发明涉及一种信息处理方法和装置、程序和记录介质，特别涉及一种用于记录包括如下信息的文件的信息处理方法和装置、程序和记录介质：在 GUI 上示出用于说明的信息，主再现路径信息，次再现路径信息，构成主再现路径的各个再现域之间的连接信息，或用户用来设置所需场景的书签或恢复点信息。

10

背景技术

近来，各种各样类型的光盘已被提出用作可从记录装置移走的记录介质。这些可记录光盘已被提出用作几个 GB 的大容量介质，并且被认为有前途用作记录 AV(audio visual, 音频视频)信号的介质。在数字 AV 信号(该信号记录在可记录光盘上)源(供应源)中，存在，例如，CS 数字卫星广播和 BS 数字广播。数字系统的地面波电视广播也已被提出以作将来使用。

15

需要说明的是从这些源提供的数字视频信号常规上根据 MPEG(Motion Picture Experts Group, 运动图像专家组)2 系统进行压缩。对于记录装置，设置适合该装置的记录率。如果来自数字广播的数字视频信号用家用的传统视频存储介质根据模拟记录系统进行记录，数字视频信号首先经过解码，然后进行频带限制，以作记录。作为替换，使用数字记录系统，首先作为例子的是 MPEG1 视频，MPEG2 视频或 DV 系统，数字视频信号根据适合装置的记录率和编码进行一次性解码，随后进行重新编码，以作记录。

20

然而，使用这种在记录之前对所提供的位流进行一次性解码，然后带宽限制或重新编码的记录方法，图像质量必定遭到破坏。如果记录经过压缩的数字信号时，输入数字信号的传输率不高于记录和/或再现装置的记录率，对所提供的位流进行直接记录而不解码或重新编码的方法，能使图像质量的损害降到最小程度。然而，如果所压缩数字信号的传输率超过作为记录介质的盘的记录率，就一定需要首先在记录和/或重现装置中对数字信号进行解码，并且对数字信号进行重新编码，以作记录，从而传输率将不高于盘的记

25

30

录率上限。

如果信号根据随时间提高或降低输入数字信号的位率的可变率系统进行传输，在可以将数据首先存储在缓冲区中，并以猝发方式(burst-like)进行记录的盘记录系统中，比在旋转头具有固定 rpm，因此记录率为固定记录率的记录系统中，可以更少浪费使用记录介质的容量。

在数字广播成为主流的不久将来，可以预见肯定需要对于广播信号在数字信号的状态下进行记录，而不解码或重新编码，如数据流的情况，并且使用盘作为记录介质的记录和/或再现装置。

同时，用上述记录装置在记录介质上记录 AV 流数据时，可以分析 AV 流数据，以允许快速回放来检测 I-图像的位置，以实现允许访问 I-图像的记录。作为替换，可以不分析而直接记录 AV 流。

在这种情况下，传统做法是提供各自专用应用程序，通过该程序，在记录介质上将 AV 流记录为不同格式的 AV 流。结果是，应用程序的开发趋向于昂贵且费时。记录在各自应用程序中的 AV 流中，AV 流的格式相互不同，结果是各个 AV 流由于缺乏兼容性不能在同一装置上进行再现。

另外，传统记录装置有例如，音频数据难于后期记录的缺点。

发明公开

因此，本发明的一个目的是提供一种方案，可以共同管理能够执行高速记录的 AV 流和不能执行高速记录的 AV 流。

本发明的另一目的是提供一种能够后期记录的方案。

一种用于在记录介质上记录 AV 流数据的信息处理装置包括：第一生成装置，用于生成描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系的第二表；选择装置，用于根据记录方法选择第一表和第二表中的一个；和第一记录装置，用于在记录介质上与 AV 流数据一起记录所选表。

第一表可以是 EP_map 并且第二表可以是 TU_map。

选择装置可以在非认知(non-cognizant)记录的情况下选择第二表。

选择装置可以在自编码记录的情况下选择第一表。

选择装置可以在认知记录的情况下选择第一表。

该信息处理装置可以进一步包括：第二生成装置，用于生成指定再现 AV 流数据的再现指定信息；和第二记录装置，用于在记录介质上记录指定由第二生成装置生成的再现指定信息。再现指定信息可以包括用于指定是根据呈现时间还是根据到达时间表示 AV 流数据再现域的时间信息的标识信息。

- 5 如果第一表与 AV 流数据一起进行记录，再现指定信息根据呈现时间表示 AV 流数据再现域的时间信息。如果第二表与 AV 流数据一起进行记录，再现指定信息根据到达时间表示 AV 流数据再现域的时间信息。

10 本发明还提供一种用于在记录介质上记录 AV 流数据的信息处理方法包括如下步骤：生成步骤，生成描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系的第二表；选择步骤，根据记录方法选择第一表和第二表中的一个；和记录步骤，在记录介质上与 AV 流数据一起记录所选表。

15 本发明还为在记录介质上记录 AV 流数据的信息处理装置提供一种其中记录有一个计算机可读程序的记录介质，其中，该程序包括如下步骤：生成步骤，生成描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系的第二表；选择步骤，根据记录方法选择第一表和第二表中的一个；和记录步骤，在记录介质上与 AV 流数据一起记录
20 所选表。

本发明还提供一种程序，用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤：生成步骤，生成描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系的第一表，或描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系的第二
25 表；选择步骤，根据记录方法选择第一表和第二表中的一个；和记录步骤，在记录介质上与 AV 流数据一起记录所选表。

30 本发明还提供一种用于从记录介质再现 AV 流数据的信息处理装置包括：再现装置，用于从其中根据记录方法记录有第一表或第二表的记录介质再现第一表和第二表中的一个，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系；和控制装置，用于根

据所再现的表控制 AV 流数据的输出。

本发明还提供一种用于从记录介质再现 AV 流数据的信息处理方法包括如下步骤：再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现第一表和第二表中的一个，第一表描述呈现时间戳和对应访问单元的
5 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系；和控制步骤，根据所再现的表控制 AV 流数据的输出。

本发明还为在记录介质上记录 AV 流数据的信息处理装置提供一种其中记录有一个计算机可读程序的记录介质，其中，该程序包括如下步骤：再现
10 步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现第一表和第二表中的一个，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系；和控制步骤，根据所再现的表控制 AV 流数据的输出。

本发明还提供一种程序，用于让控制信息处理装置在记录介质上记录 AV
15 流数据的计算机执行如下步骤：再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现第一表和第二表中的一个，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关
20 系；和控制步骤，根据所再现的表控制 AV 流数据的输出。

本发明还提供一种其中根据记录方法记录有第一表和第二表中的一个的记录介质，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系。

本发明还提供一种用于在记录介质上记录 AV 流数据的信息处理装置包括：生成装置，用于生成由表示主再现路径的第一信息和表示次再现路径的第二信息构成的再现指定信息；和记录装置，用于在记录介质上记录 AV 流
25 数据和再现指定信息。

主再现路径可以是用于音频数据后期记录的路径。

30 第一信息可以是子路径(main_path)并且第二信息可以是次路径(Sub-path)。

第二信息可以包括表示次再现信息类型的类型信息、用次再现路径引用的 AV 流文件名、次再现路径 AV 流的入点(in-point)和出点(out-point)和在主路径的时间轴上再现路径的入点开始同步的主路径时间。

5 本发明还为在记录介质上记录 AV 流数据的信息处理装置提供一种信息处理方法，包括如下步骤：生成步骤，生成由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和记录步骤，在记录介质上记录 AV 流数据和再现指定信息。

10 本发明还为在记录介质上记录 AV 流数据的信息处理装置提供一种其中记录有一个计算机可读程序的记录介质，其中，该程序包括如下步骤：生成步骤，生成由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和记录步骤，在记录介质上记录 AV 流数据和再现指定信息。

15 本发明还提供一种程序，用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤：生成步骤，生成由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和记录步骤，在记录介质上记录 AV 流数据和再现指定信息。

20 本发明还提供一种用于从记录介质再现 AV 流数据的信息处理装置包括：再现装置，用于从记录介质再现由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制装置，用于根据所再现的再现指定信息控制 AV 流数据的输出。

本发明还包括一种用于从记录介质再现 AV 流数据的信息处理方法，包括如下步骤：再现步骤，从记录介质再现由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制步骤，根据所再现的再现指定信息控制 AV 流数据的输出。

25 本发明还为在记录介质上记录 AV 流数据的信息处理装置提供一种其中记录有一个计算机可读程序的记录介质，其中，该程序包括如下步骤：再现步骤，从记录介质再现由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制步骤，根据所再现的再现指定信息控制 AV 流数据的输出。

30 本发明还提供一种程序，用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤：再现步骤，从记录介质再现由表示主再现路

径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制步骤，根据所再现的再现指定信息控制 AV 流数据的输出。

5 本发明的记录介质其中记录有由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息。

10 在本发明的信息记录和/或再现方法和装置、用于记录介质的程序、程序，和记录介质中，根据记录方法记录第一表和第二表中的一个，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系。

15 在本发明的信息处理方法和装置、用于记录介质的程序和程序中，从其中根据记录方法记录有第一表或第二表的记录介质再现第一表和第二个中的一个，第一表描述呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系，从而控制输出。

在本发明的信息处理方法和装置、用于记录介质的程序、程序和第二记录介质中，再现指定信息包括表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息。

20 在本发明的信息处理方法和装置、用于记录介质的程序和程序中，再现由表示主再现路径的第一信息和表示与主再现路径同步再现的次再现路径的第二信息构成的再现指定信息，从而控制输出。

通过阅读如图所示的本发明实施例，本发明的其它目的、特点和优点将变得更加清楚。

25 附图简述

- 图 1 示出本发明记录和/或再现装置的实施例结构；
- 图 2 示出由记录和/或再现装置 1 在记录介质上所记录数据的数据格式；
- 图 3 示出 Real PlayList(实播放列表)和 Virtual PlayList(虚播放列表)；
- 图 4A、4B 和 4C 示出 Real PlayList 的创建；
- 30 图 5A、5B 和 5C 示出 Real PlayList 的删除；
- 图 6A 和 6B 示出组合编辑；

- 图 7 示出在 Virtual PlayList 中提供次路径。
- 图 8 示出改变 PlayList 回放顺序；
- 图 9 示出 PlayList 上的标记和 Clip(片断)上的标记；
- 图 10 示出菜单缩略图；
- 5 图 11 示出加到 PlayList 的标记；
- 图 12 示出加到 Clip 的标记；
- 图 13 示出 PlayList、Clip 和缩略图文件之间的关系；
- 图 14 示出目录结构；
- 图 15 示出 infr.dvr 的语法；
- 10 图 16 示出 DVRVolume 的语法；
- 图 17 示出 ResumeVolume 的语法；
- 图 18 示出 UIAppInfoVolume 的语法；
- 图 19 示出字符集值表；
- 图 20 示出 TableOfPlayList 的语法；
- 15 图 21 示出 TableOfPlayList 的另一语法；
- 图 22 示出 MakersPrivateData 的语法；
- 图 23 示出 xxxx.rpls 和 yyyy.vpls 的语法；
- 图 24A 到 24C 示出 PlayList；
- 图 25 示出 PlayList 的语法；
- 20 图 26 示出 PlayList_type 表；
- 图 27 示出 UIAppInfoPlayList 的语法；
- 图 28A 到 28C 示出图 27 所示 UIAppInfoPlayList 语法中的标志；
- 图 29 示出一个 PlayItem(播放项)；
- 图 30 示出一个 PlayItem；
- 25 图 31 示出一个 PlayItem；
- 图 32 示出 PlayItem 语法；
- 图 33 示出 IN-time；
- 图 34 示出 OUT-time；
- 图 35 示出 Connection_Condition(连接条件)表；
- 30 图 36A 到 36D 示出 Connection_Condition；
- 图 37 示出 BridgeSequenceInfo；

- 图 38 示出 BridgeSequenceInfo 的语法；
图 39 示出 SubPlayItem；
图 40 示出 SubPlayItem 语法；
图 41 示出 Mark_type 表；
5 图 42 示出 PlayListMark 语法；
图 43 示出 Mark_type 表；
图 44 示出 Mark_time_stamp；
图 45 示出 zzzzz.clip 语法；
图 46 示出 ClipInfo 语法；
10 图 47 示出 Clip_stream_type 表；
图 48 示出 offset_SPN；
图 49 示出 offset_SPN；
图 50A、50B 示出 STC 域；
图 51 示出 STC_Info；
15 图 52 示出 STC_Info 语法；
图 53 示出 ProgramInfo；
图 54 示出 ProgramInfo 语法；
图 55 示出 VideoCondngInfo 语法；
图 56 示出 Video_format 表；
20 图 57 示出 frame_rate 表；
图 58 示出 display_aspect_ratio 表；
图 59 示出 AudioCondngInfo 语法；
图 60 示出 audio_coding 表；
图 61 示出 audio_component_type 表；
25 图 62 示出 sampling_frequency 表；
图 63 示出 CPI；
图 64 示出 CPI；
图 65 示出 CPI 语法；
图 66 示出 CPI_type 表；
30 图 67 示出视频 EP_map；
图 68 示出 EP_map；

- 图 69 示出 EP_map ;
- 图 70 示出 EP_map 语法 ;
- 图 71 示出 EP_typevalues 表 ;
- 图 72 示出 EP_map_for_one_stream_PID 语法 ;
- 5 图 73 示出 TU_map ;
- 图 74 示出 TU_map 语法 ;
- 图 75 示出 ClipMark 语法 ;
- 图 76 示出 Mark_type 表 ;
- 图 77 示出 Mark_type_stamp 表 ;
- 10 图 78 示出 menu.thmb 和 mark.thmb 语法 ;
- 图 79 示出缩略图语法 ;
- 图 80 示出 thumbnail_picture_format 表 ;
- 图 81A 和 81B 示出 tn_block ;
- 图 82 示出 DVR MPEG2 传输流的结构 ;
- 15 图 83 示出 DVR MPEG2 传输流的记录器模型 ;
- 图 84 示出 DVR MPEG2 传输流的播放器模型 ;
- 图 85 示出源包语法 ;
- 图 86 示出 TP_extra_header 语法 ;
- 图 87 示出拷贝准许指示器表 ;
- 20 图 88 示出无缝连接 ;
- 图 89 示出无缝连接 ;
- 图 90 示出无缝连接 ;
- 图 91 示出无缝连接 ;
- 图 92 示出无缝连接 ;
- 25 图 93 示出音频重叠 ;
- 图 94 示出采用 BridgeSequence 的无缝连接 ;
- 图 95 示出未采用 BridgeSequence 的无缝连接 ;
- 图 96 示出 DVR STD 模型 ;
- 图 97 示出用于解码和显示的时间图 ;
- 30 图 98 示出 PlayList 文件语法 ;
- 图 99 示出图 98 PlayList 文件中 UIAppInfoPlayList 的语法 ;

- 图 100 示出图 98 PlayList 文件中 PlayList()的语法；
图 101 示出 SubPlayItem 的语法；
图 102 是示出形成 RealPlayList 方法的流程图；
图 103 是示出形成 VirtualPlayList 方法的流程图；
5 图 104 是示出再现 PlayList 方法的流程图；
图 105 是示出再现 PlayList 次路径方法的流程图；
图 106 是示出形成 PlayListMark 方法的流程图；
图 107 是示出采用 PlayListMark 定位再现方法的流程图；
图 108 示出介质。

10

实现本发明的最佳方式

将参照附图对本发明的实施例进行详细的描述。图 1 示出实施本发明的记录和/或再现装置 1 的典型内部结构。首先，说明为记录从外部输入的信号配置的记录单元 2 的结构。记录和/或再现装置 1 被配置用于输入和记录
15 模拟或数字数据。

模拟视频信号和模拟音频信号分别输入到终端 11、12。输入到终端 11 的视频信号，输出到分析单元 14 和 AV 编码器 15。输入到终端 12 的音频信号输出到分析单元 14 和 AV 编码器 15。分析单元 14 从输入视频和音频信号提取特征点，如场景改变。

20 AV 编码器 15 对输入视频和音频信号进行编码，以将系统信息(S)，如编码视频流(V)、编码音频流(A)和 AV 同步输出到多路复用器 16。

编码视频流是使用 MPEG(Motion Picture Experts Group,运动图像专家组)2 系统编码的视频流，而编码音频流是根据 MPEG1 系统编码的音频流(编码音频流例如是在 MPEG1 系统中编码的音频流)，或根据 Dolby AC3(商标)
25 系统编码的音频流。多路复用器 16 根据输入系统信息多路复用输入视频和音频流，以通过开关 17 将多路复用流输出到多路复用流分析单元 18 和源分包器 19。

多路复用流，例如是 MPEG-2 传输流或 MPEG2 节目流。源分包器 19 根据记录介质 100 的应用格式将输入多路复用流编码为由源包组成的 AV
30 流，以在该记录介质上记录流。在输出到写入单元 22 之前，AV 流在 ECC(Error Correction Code, 纠错码)单元 20 和调制单元 21 中进行处理，以添加 ECC

码和进行调制，然后写入单元 22 根据由控制器 23 输出的控制信号写入 AV 流文件。

从数字接口或数字电视调谐器输入的传输流，如数字电视广播输入到终端 13。有两个记录系统用于记录输入到终端 13 的传输流，一个为透明记录系统，另一个为在记录之前重新编码以降低，例如，记录位率的系统。记录系统命令信息从作为用户界面的终端 24 输入到控制器 23。

在透明记录输入传输流中，输入到终端 13 的传输流通过开关 17 输出到多路复用流分析单元 18 和源分包器 19。在记录介质上记录 AV 流的随后处理如同编码和记录模拟输入音频和视频信号，如上所述，因此为简单起见，在此不作说明。

如果输入传输流先经过重新编码，然后进行记录，输入到终端 13 的传输流输入到多路分解器 26，它对输入传输流进行多路分解，以提取视频流(V)，音频流(A)和系统信息(S)。

由多路分解器 26 提取的流(信息)中，视频流输出到音频解码器 27，而音频流和系统信息输出到多路复用器 16。音频解码器 27 对输入传输流进行解码，以将编码视频流(V)输出到多路复用器 16。

从多路分解器 26 输出并输入到多路复用器 16 的音频流和系统信息，与由 AV 编码器 15 输出的视频流，根据输入系统信息进行多路复用，并且作为多路复用流，通过开关 17 输出到多路复用流分析单元 18 和源分包器 19。在记录介质上记录 AV 流的随后处理如同编码和记录模拟输入音频和视频信号，如上所述，因此为简单起见，在此不作说明。

本实施例的记录和/或再现装置 1 在记录介质 100 上记录 AV 流文件，同时还记录解释文件的应用数据库信息。输入到控制器 23 的信息是来自分析单元 14 的活动图像的特征信息，来自多路复用流分析单元 18 的 AV 流特征信息和在终端 24 输入的用户命令信息。

当 AV 编码器 15 对视频信号进行编码时，由分析单元 14 生成从分析单元 14 提供的活动图像的特征信息。分析单元 14 分析输入视频和音频信号的内容，以生成与输入活动图像信号的图像特征有关的信息(片断标记)。该信息是指示特征片断标记点的图像的信息，如节目开始点、场景变化点、CM 商业开始和结束点、输入视频信号中的标题或放映，并且还包含图像的缩略图和与音频信号的立体/单声切换点和静音部分相关的信息。

上述图像指示信息通过控制器 23 输入到多路复用器 16。当对由控制器 23 指定为片断标记的编码图像进行多路复用时，多路复用器 16 将指定 AV 流上编码图像的信息返回到控制器 23。具体地说，该信息是图像的 PTS(presentation time stamp, 呈现时间戳)或图像编码版本 AV 流的地址信息。

5 控制器 23 将这种特征图像和用于指定 AV 流上编码图像的信息相互关联进行存储。

来自多路复用流分析单元 18 的 AV 流特征信息是与要进行记录的 AV 流编码信息相关的信息，并且由分析单元 18 进行记录。例如，该特征信息包括 AV 流中 I-图像的时间戳和地址信息，系统时钟的不连续点信息，AV 10 流的编码参数和 AV 流中编码参数的变化点信息。当透明地记录从终端 13 输入的传输流时，多路复用流分析单元 18 从输入传输流检测上述片断标记的图像，并且生成用于指定由片断标记及其类型指定的图像的信息。

来自终端 24 的用户指定信息是由用户指定的，用于指定回放域、说明回放域内容的字符字母的信息，或如用户为他或她喜爱的场景设置的书签或 15 恢复点的信息。

根据上述输入信息，控制器 23 创建 AV 流(Clip)数据库，AV 流的重放域(PlayItem)组(PlayList)的数据库，记录介质 100 所记录内容的管理信息 (info.dvr)和缩略图信息。类似于 AV 流，从上述信息构造的应用数据库信息在 ECC 单元 20 和调制单元 21 中进行处理，并且输入到写入单元 22，然后 20 写入单元 22 在记录介质 100 上记录数据库文件。

随后将对上述应用数据库信息进行详细描述。

当记录在记录介质 100 上的 AV 流文件(图像数据和语音数据文件)和如此记录在记录介质 100 上的应用数据库信息由再现单元 3 进行再现，控制器 23 首先命令读出单元 28 从记录介质 100 读出应用数据库信息。读出单元 28 25 从记录介质 100 读出应用数据库信息，然后它从记录介质 100 读出应用数据库信息，以将通过解调单元 29 和 ECC 解码器 30 的解调和纠错处理的应用数据库信息发送到控制器 23。

根据应用数据库信息，控制器 23 将记录在记录介质 100 上的 PlayList 列表输出到终端 24 的用户界面。用户从 PlayList 列表选择要进行再现的 30 PlayList。与指定要再现的 PlayList 相关的信息输入到控制器 23。控制器 23 命令读出单元 28，读出再现 PlayList 时必要的 AV 流文件。根据该命令，读

出单元 28 从记录介质 100 读出对应 AV 流，以将所读出 AV 流输出到解调单元 29。如此输入到解调单元 29 的 AV 流通过预设处理进行解调，并且通过 ECC 解码器 30 的处理输出到源解包器 31。

源解包器 31 将从记录介质 100 读出并以预设方式处理的应用格式的 AV 流转换为可由多路分解器 26 处理的流。多路分解器 26 将形成由控制器 23 指定的 AV 流回放域(PlayItem)的系统信息(S)，如视频流(V)，音频流(A)或 AV 同步，输出到音频解码器 27，AV 解码器对视频流和音频流进行解码，以将回放视频信号和回放音频信号分别输出到相关终端 32、33。

如果从作为用户界面的终端 24 输入指示随机访问回放或特殊回放的信息，控制器 23 根据 AV 流的数据库内容(Clip)，确定从记录介质 100 读出 AV 流的位置，以命令读出单元 28 读出 AV 流。如果由用户选择的 PlayList 从预设时间点进行再现，控制器 23 命令读出单元 28 从具有离指定时间点最近的时间戳的 I-图像读出数据。

当用户已从存储在 Clip Information(片断信息)的 ClipMark 中的节目索引点或场景变化点，选择特定片断标记，如当用户从存储在 ClipMark 中的索引点或场景变化点的缩略图列表(显示在用户界面上)选择特定图像时，控制器 23 确定从记录介质 100 AV 流读出位置，以命令读出单元 28 读出 AV 流。也就是，控制器 23 命令读出单元 28 从具有离存储用户所选图像的 AV 流地址最近的地址的 I-图像读出数据。读出单元 28 从指定地址读出数据。读出数据由解调单元 29，ECC 解码器 30 和源分包器 19 进行处理，以提供给多路分解器 26，并由音频解码器 27 进行解码，以再现由标记点图像地址指示的 AV 数据。

如果用户已命令快进回放，控制器 23 命令读出单元 28 根据 AV 流的数据库(Clip)顺序读出连续 AV 流中的 I-图像数据。

读出单元 28 从指定随机访问点读出 AV 流数据。如此读出的数据通过下游方的各种组件的处理进行再现。

现在说明用户编辑记录在记录介质 100 上的 AV 流的情况。如果想要指定记录在记录介质 100 上的 AV 流的回放域，例如，如果想要创建从歌曲节目 A 再现歌手 A 所唱部分，并且随后从另一歌曲节目 B 再现歌手 A 所唱部分的回放路径，与回放域的开始点(IN-point)和结束点(OUT-point)相关的信息从作为用户界面的终端输入到控制器 23。控制器 23 创建 AV 流回放域

(PlayItem)组(PlayList)的数据库。

当用户想要删除记录在记录介质 100 上的 AV 流部分，与删除域的 IN-point 和 OUT-point 相关的信息输入到控制器 23，然后控制器 23 修改 PlayList 数据库，以仅引用所需 AV 流。控制器 23 还命令写入单元 22 删除 AV 流的不需要流部分。

现在说明用户想要指定记录在记录介质上的 AV 流的回放域，以创建新的回放路径，并且以无缝方式相互连接各个回放域。在这种情况下，控制器 23 创建 AV 流的回放域(PlayItem)组(PlayList)数据库，并且承担对回放域连接点附近的视频流部分进行重新编码和重新多路复用。

10 回放域的 IN-point 图像信息和 OUT-point 图像信息从终端 24 输入到控制器 23。控制器 23 命令读出单元 28 读出再现 IN-point 和 OUT_point 图像所需的数据。读出单元 28 从记录介质 100 读出数据。如此读出的数据通过解调单元 29、ECC 解码器 30 和源解包器 31 输出到多路分解器 26。

15 控制器 23 分析输入到多路分解器 26 的数据，以确定视频流的重新编码方法(改变 picture_coding_type 和分配重新编码的编码位数量)和重新多路复用系统，以将该系统发送到 AV 编码器 15 和多路复用器 16。

20 多路分解器 26 然后将输入流分解为视频流(V)、音频流(A)和系统信息(S)。视频流可以分类为输入到音频解码器 27 的数据和输入到多路复用器 16 的数据。前者是重新编码所需的数据，并且由音频解码器 27 进行解码，解码图像然后由 AV 编码器 15 进行重新编码，并且因此导致变成视频流。后者是不重新编码而从原始流复制的数据。音频流和系统信息直接输入到多路复用器 16。

25 多路复用器 16 根据从控制器 23 输入的信息对输入流进行多路复用，以输出多路复用流，该流由 ECC 单元 20 和调制单元 21 进行处理，以将其发送到写入单元 22。写入单元 22 根据从控制器 23 提供的控制信号在记录介质 100 上记录 AV 流。

下面对基于该信息的应用数据库信息和操作，如回放和编辑进行说明。图 2 示出用于 AV 流管理的具有两层，即 PlayList 和 Clip 的应用格式结构。Volume Information(卷信息)管理盘中所有 Clip 和 PlayList。在此，一起成对的一个 AV 流及其附属信息被认为是一个对象，并且称作 Clip。AV 流文件被称作 Clip AV 流文件，附属信息被称作 Clip Information(片断信息)文件。

一个 Clip AV 流文件存储在应用格式所规定的结构中安排的 MPEG-2 传输流的数据。大体上，文件视作字节串。Clip AV 流文件的内容在时间轴上进行扩展，Clip 中的进入点(I-图像)主要根据时间进行指定。当给定预设 Clip 访问点的时间戳时，Clip Information 文件在查找开始在 Clip AV 流文件中读出数据的地址信息时有用。

参照图 3，现在说明 PlayList，它为用户进行提供，以从 Clip 选择想要观看的回放域，并且容易地编辑回放域。一个 PlayList 是 Clip 中的一组回放域。预设 Clip 中的一个回放域被称作 PlayItem，并且由时间轴上的 IN-point 和 OUT-point 对进行表示。因此，PlayList 由一组多个 PlayItem 形成。

10 PlayList 分为两种类型：一种为 Real PlayList(实播放列表)，另外一种为 Virtual PlayList(虚播放列表)。Real PlayList 共有所引用 Clip 流部分。也就是，Real PlayList 在盘中占有与所引用 Clip 流部分对应的数据量，并且当删除 Real PlayList 时，也删除所引用 Clip 流部分的数据。

15 Virtual PlayList 不是共有 Clip 数据。因此，如果改变或删除 Virtual PlayList，Clip 内容绝不作改变。

对 Real PlayList 的编辑进行说明。图 4A 示出创建 Real PlayList，并且，如果 AV 流被记录为新 Clip，引用整个 Clip 的 Real PlayList 为最新创建的操作。

20 图 4B 示出 Real PlayList 的切分，即在所需点上切分 Real PlayList，以将 Real PlayList 裂分为两个 Real PlayList。当两个节目由单个 PlayList 在一个 Clip 中进行管理，并且用户意图将节目重新登记或重新记录为分开的单个节目时，执行切分操作。这个操作不导致 Clip 内容的改变，即对 Clip 自身进行切分。

25 图 4C 示出 Real PlayList 的组合操作，将两个 Real PlayList 组合为一个新 Real PlayList 的操作。如当用户想要将两个节目重新登记为单个节目时，执行该组合操作。这个操作不导致 Clip 内容的改变，即将 Clip 自身组合为一个。

图 5A 示出整个 Real PlayList 的删除。如果删除整个预设 Real PlayList，也删除由所删除 Real PlayList 引用的相关 Clip 流部分。

30 图 5B 示出 Real PlayList 的部分删除。如果删除所需的 Real PlayList 部分，改变相关联的 PlayItem，仅引用所需 Clip 流部分。删除对应的 Clip 流

部分。

图 5C 示出 Real PlayList 的最小化。它是导致与 Real PlayList 相关联的 PlayItem 只引用 Virtual PlayList 所需的 Clip 流部分的操作。删除 Virtual PlayList 不需要的对应 Clip 流部分。

- 5 如果通过上述操作改变 Real PlayList，从而删除由 Real PlayList 引用的 Clip 流部分，可能存在使用被删除 Clip 的 Virtual PlayList，从而由于被删除的 Clip，Virtual PlayList 可能产生问题。

10 为了防止出现这种问题，为用户显示一条消息“如果存在 Virtual PlayList 引用 Real PlayList 所引用的 Clip 流部分，并且删除 Real PlayList，删除 Virtual PlayList 本身-可否？”，以通过确认或报警的方式响应用户的删除操作，随后根据用户命令执行或取消删除处理。作为替换，执行 Real PlayList 的最小化操作，而不是删除 Virtual PlayList。

15 现在说明 Virtual PlayList 的操作。如果执行一个 Virtual PlayList 的操作，不改变 Clip 的内容。图 6A 和 6B 示出组合和编辑(IN-OUT 编辑)。它是创建用户想要观看的回放域的 PlayItem，以创建 Virtual PlayList 的操作。PlayItem 之间的无缝连接由应用格式(后面说明)进行支持。

20 如果存在两个 Real PlayList 1、2 和与各个 Real PlayList 相关联的 Clip1、2，用户将 Real PlayList 1 的预设域(从 IN1 到 OUT1 的域：PlayItem 1)指定为回放域，并且还将 Real PlayList 2 的预设域(从 IN2 到 OUT2 的域：PlayItem 2)指定为下一步要进行回放的域，如图 6A 所示，单个 Virtual PlayList 由 PlayItem 1 和 PlayItem 2 构成，如图 6B 所示。

25 现在说明 Virtual PlayList 的重新编辑。重新编辑可以列举为如下操作：修改 Virtual PlayList 中的 IN-或 OUT-点，插入或添加新 PlayItem 到 Virtual PlayList，并且在 Virtual PlayList 中删除 PlayItem。也可以删除 Virtual PlayList 本身。

图 7 示出 Virtual PlayList 的配音(后期记录)。它是将音频后期记录作为次路径登记到 Virtual PlayList 的操作。该音频后期记录由应用软件进行支持。附加的音频流作为次路径加到 Virtual PlayList 主路径的 AV 流。

30 Real PlayList 和 Virtual PlayList 共同的操作是图 8 所示的修改(移动)PlayList 的回放顺序。该操作是对盘(卷)中 PlayList 的回放顺序进行改变，并且由应用格式中定义的 TableOfPlayList 进行支持，随后将参照例如图 20

进行说明。该操作不导致 Clip 内容的改变。

现在说明标记(Mark)。标记用来指定 Clip 和 PlayList 中的精彩场面或特征时间，如图 9 所示。加入到 Clip 的标记称作 ClipMark。ClipMark 是例如用于指定 AV 流中内容的特征场景的节目索引点或场景改变点。由例如图 1
5 的分析单元 14 生成 ClipMark。当再现 PlayList 时，可以引用并使用由 PlayList 引用的 Clip 标记。

添加到 PlayList 的标记称作 PlayListMark(播放列表标记)。PlayListMark 例如是由用户设置的书签点或恢复点。通过将表示标记时间点的时间戳加到标记列表，将标记设置到 Clip 和 PlayList。另一方面，标记删除是从标记列表
10 删除标记时间点。因此，标记设置或标记删除绝不改变 AV 流。

作为 ClipMark 的另一格式，由 ClipMark 引用的图像可以在 AV 流中根据地址进行指定。通过将指示标记点图像的地址基本信息加到标记列表，在 Clip 上设置标记。另一方面，标记删除是从标记列表删除指示标记点图像的地址基本信息。因此，标记设置或标记删除绝不改变 AV 流。

现在说明缩略图。缩略图是加到 Volume(卷)、PlayList 和 Clip 的静止图像。存在两种缩略图，其中一种是作为表示内容的代表性图像的缩略图。这主要用在主图像中，以为用户通过操作光标选择他或她想要观看的内容，没有显示。另一种缩略图是表示标记所指场景的图像。

Volume 和各个 PlayList 需要有代表性图像。Volume 的代表性图像预定
20 用作当盘置于记录和/或再现装置 1 中时，初始显示代表盘内容的静止图像。需要说明的是，盘是指假定为具有盘形状的记录介质 100。PlayList 的代表性图像预定用作表示 PlayList 内容的静止图像。

作为 PlayList 的代表性图像，可以考虑使用 PlayList 的初始图像作为缩略图(代表性图像)。然而，回放时间为 0 处的引导图像不一定是表示内容的
25 最佳图像。因此，用户允许将可选图像设置为 PlayList 的缩略图。两种缩略图，作为表示 Volume 的代表性图像的缩略图和作为表示 PlayList 的代表性图像的缩略图，称作菜单缩略图。由于顺序显示菜单缩略图，这些缩略图需要以上升的速度从盘进行读出。因此，在单个文件中存储全部菜单缩略图更有效率。菜单缩略图没有必要是从卷的活动图像提取的图像，它可以是从个人计算机或数字静止相机拍摄的图像，如图 10 所示。
30

另一方面，Clip 和 PlayList 需要用多个标记进行标记，而标记点的图像

需要进行容易的观看，以获得标记位置的内容。表示这种标记点的图像被称作标记缩略图。因此，作为标记缩略图原图的图像主要是所提取的标记点图像，而不是从外部拍摄的图像。

图 11 示出附于 PlayList 的标记和标记缩略图之间的关系，同时图 12 示出附于 Clip 的标记和标记缩略图之间的关系。与菜单缩略图不同，标记缩略图用在例如表示 PlayList 细节的子菜单中，同时它不要求以短访问时间进行读出。因此，无论什么时候需要缩略图，记录和/或再现装置 1 打开一个文件，并且读出该文件的一部分，同时即使记录和/或再现装置 1 的文件打开和读出该文件的一部分需要一些时间，也不会出现问题。

为减少卷中出现的文件数，全部标记缩略图最好存储在一个文件中。虽然 PlayList 可以拥有一个菜单缩略图和多个标记缩略图，但是用户不要求直接选择 Clip(通常，通过 PlayList 选择 Clip)，因此不是一定提供菜单缩略图。

图 13 示出菜单缩略图、标记缩略图、PlayList 和 Clip 之间的关系。在菜单中，缩略图文件是逐 PlayList 提供的文件菜单缩略图。在菜单中，表示记录在盘上数据内容的卷缩略图包含缩略图文件。在菜单中，缩略图文件是逐 PlayList 和逐 Clip 创建的文件缩略图。

下面说明 CPI(Character Point Information, 特征点信息)。CPI 是包含在 Clip 信息文件中的数据，并且主要用于当提供 Clip 访问点时间戳时，在 Clip AV 流文件中查找开始数据读出的数据地址。在本实施例中，使用两种 CPI，其中一个为 EP_map，另一个为 TU_map。

EP_map 是从基本流和传输流提取的入口点(entry point, EP)数据列表。它具有用来查找开始解码的 AV 流中入口点位置的地址信息。一个 EP 数据由呈现时间戳(presentation time stamp, PTS)和与 PTS 相关联的访问单元的 AV 流中的数据地址构成，该数据地址与 PTS 进行成对。

EP_map 主要用作两个目的。第一，它用来在 PlayList 中查找 PTS 所引用访问单元的 AV 流中的数据地址。第二，EP_map 用来快进回放或快退回放。如果通过记录和/或再现装置 1 记录输入 AV 流时，能分析流的语法，在盘上创建和记录 EP_map。

TU_map 具有来源于通过数字接口输入的传输包的到达时间点的时间单元(time unit, TU)列表。这提供到达时基的时间和 AV 流中数据地址之间的关系。当记录和/或再现装置 1 记录输入 AV 流，并且不能分析流的语法时，

在盘上创建并记录 TU_map。

STCInfo 在存储 MPEG-2 传输流的 AV 流文件中存储不连续点信息。

当 AV 流具有 STC 的不连续点，相同的 PTS 值可以出现在 AV 流文件中。因此，如果根据 PTS 指定 AV 流中的时间点，PTS pf 访问点不够指定
5 该点。而且，需要包含 PTS 的连续 STC 域的索引。采用这种格式，连续 STC 域及其索引分别称作 STC-sequence 和 STC-sequence-id。STC-sequence 信息由 Clip Information 文件的 STCInfo 进行定义。

STC-sequence-id 用在 AV 流文件中，并且在具有 TU_map 的 AV 流文件中是可选的。

10 节目是基本流的集合，并且共有单个系统时基，以对这些流进行同步再现。

对于再现装置(图 1 的记录和/或再现装置 1)，在编码之前知道 AV 流的内容是有用的。这些内容包括，例如传输音频或视频基本流的传输包的 PID 值，或者视频或音频组件类型，如 HDTV 视频或 MPEG-2 AAC 音频流。这
15 些信息对于创建向用户说明引用 AV 流的 PlayList 内容的菜单屏幕有用。类似地，对设置各个装置的 AV 解码器和多路分解器的初始状态有用。

由于这个原因，Clip Information 文件具有用于说明节目内容的 ProgramInfo。

可能出现节目内容在其中存储 MPEG-2 传输流的 AV 流文件中发生改变
20 的情况。例如，传输视频基本流的传输包 PID 可能发生改变，或者视频流的组件类型可能从 SDTV 改变为 HDTV。

ProgramInfo 存储 AV 流文件中节目内容改变点信息。其中节目内容保持不变的 AV 流文件域称作 program-sequence。

25 该 program-sequence 用在具有 EP_map 的 AV 流文件中，并且在具有 TU_map 的 AV 流文件中是可选的。

本实施例定义自编码流格式(self-encoding stream format, SESF)。SESF 用来对模拟输入信号进行编码，并且对数字输入信号进行解码，以随后将解码信号编码为 MPEG-2 传输流。

30 SESF 定义与 MPEG-2 传输流和 AV 流相关的基本流。当记录和/或再现装置 1 用 SESF 对输入信号进行编码和记录时，在盘上创建并记录 EP_map。

数字广播流使用下面系统中的一个，以在记录介质 100 上记录：首先，

数字广播流代码转换为 SESF 流。在这种情况下，所记录的流必须遵循 SESF，并且必须在盘上准备和记录 EP_map。

作为替换，形成数字广播流的基本流代码转换为一个新基本流，并且重新多路复用为一个遵循由标准化数字广播流的组织规定的流格式的新传输流。在这种情况下，必须在盘上创建和记录 EP_map。

例如，假定输入流为遵循 ISDB(standard appellation of digital BS of Japan，日本数字广播流标准命名)的 MPEG-2 传输流，传输流包含 HDTV 视频流和 MPEG AAC 音频流。HDTV 视频流代码转换为 SDTV 视频流，SDTV 视频流和原始 AAC 音频流重新多路复用到 TS。SDTV 流和传输流都需要遵循 ISDB 格式。

在记录介质 100 上记录数字广播流的另一系统是对输入传输流作透明记录，即不加修改地记录输入传输流，在这种情况下，在盘上列出和记录 EP_map。

作为替换，透明地记录输入传输流，即不加改变地记录输入传输流，在这种情况下，在盘上创建和记录 TU_map。

以下说明目录和文件。以下将记录和/或再现装置 1 作为 DVR(digital video recording，数字视频记录)进行描述。图 14 示出盘上的典型目录结构。DVR 盘的目录可以列举为包括“DVR”目录的根目录和包括“PLAYLIST”目录、“CLIPINF”目录、“M2TS”目录和“DATA”目录的“DVR”目录，如图 14 所示。虽然根目录下可以创建其它目录，但是在本实施例的应用格式中不考虑这些目录。

“DATA”目录下，存储有由 DVR 应用格式规定的所有文件和目录。

“DVR”目录包括四个目录。“PLAYLIST”目录下存放 Real PlayList 和 Virtual PlayList 数据库文件。后一目录可以在缺少 PlayList 的状态下存在。

“CLIPINF”下存放 Clip 数据库。该目录也可以在缺少 AV 流文件的状态下存在。“DATA”目录中，存储有数据广播文件，如数字 TV 广播。

“DVR”目录存储下面文件：即，在 DVR 目录下创建“info.dvr”，以存储应用层的综合信息。DVR 目录下，必须存在单个 info.dvr。该文件名假定被固定为 info.drv。“menu.thmb”存储与菜单缩略图相关的信息。DVR 目录下，必须存在 0 或 1 个标记缩略图。该文件名假定被固定为“menu.thmb”。如果不存在菜单缩略图，该文件可以不存在。

“mark.thmb”存储与标记缩略图相关的信息。DVR 目录下，必须存在 0 或 1 标记缩略图。该文件名假定被固定为“menu.thmb”。如果不存在菜单缩略图，该文件可以不存在。

“PLAYLIST”目录存储两种 PlayList 文件，一种为 Real PlayList，另
5 一种为 Virtual PlayList。“xxxxx.rpls”文件存储与一个 Real PlayList 相关的信息。为每个 Real PlayList 创建一个文件。文件名为“xxxxx.rpls”，其中“xxxxx”表示五个从 0 到 9 的数字。文件扩展名必须为“rpls”。

“yyyyy.vpls”存储与一个 Virtual PlayList 相关的信息。逐个 Virtual
10 PlayList 创建文件名为“yyyyy.vpls”的文件，其中“yyyyy”表示五个从 0 到 9 的数字。文件扩展名必须为“vpls”。

“CLIPINF”目录存储与每个 AV 流文件关联的一个文件。“zzzzz.clpi”是与一个 AV 流文件对应的 Clip 信息文件(Clip AV 流或 Bridge-Clip 流文件)。文件名为“zzzzz.clpi”，其中“zzzzz”表示五个从 0 到 9 的数字。文件扩展名必须为“clpi”。

15 “M2TS”目录存储 AV 流文件。“zzzzz.m2ts”是由 DVR 系统进行处理
的 AV 流文件。这是 Clip AV 流文件或 Bridge-Clip AV 流文件。文件名为“zzzzz.m2ts”，其中“zzzzz”表示五个从 0 到 9 的数字。文件扩展名必须为“m2ts”。

20 “DATA”目录存储从数据广播传输的数据。该数据可以例如是 XML
或 MPEG 文件。

现在说明每个目录(文件)的语法和语义。图 15 示出“info.drv”文件的语法。“info.drv”文件由三个对象构成，它们是 DVRVolume()、TableOfPlayLists()和 MakersPrivateData()。

25 对图 15 所示的 info.drv 语法进行说明。TableOfPlayLists_Start_address
表示根据相对于“info.dvr”文件起始字节的字节数，TableOfPlayLists()的起始地址。相对字节数从 0 开始计数。

MakersPrivateData_Start_address 表示根据相对于“info.dvr”文件起始字节的字节数，MakersPrivateData()的起始地址。相对字节数从 0 计数。根据“info.dvr”语法插入 padding_word。N1 和 N2 是可选正整数。每个填充字
30 可以采用一个可选值。

DVRVolume()存储表明卷(盘)内容的信息。图 16 示出 DVRVolume 语法。

现在说明图 16 所示的 DVRVolume() 语法。version_number 表示示出 DVRVolume() 版本号的四个字符字母。version_number 根据 ISO646 编码为“0045”。

length 用表示从紧接 length 域之后到 DVRVolume() 末端的字节数的 32 位无符号整数进行表示。

ResumeVolume() 存储卷中上次再现的 Real PlayList 或 Virtual PlayList 文件名。然而，用户中断回放 Real PlayList 或 Virtual PlayList 时的回放位置存储在 PlayListMark() 中定义的 resume_mark 中(参看图 42 和 43)。

图 17 示出 ResumeVolume() 语法。对图 17 所示的 ResumeVolume() 语法进行说明。valid_flag 表示当该 1 位标志分别设置为 1 或 0 时 resume_PlayList_name 域有效或无效。

resume_PlayList_name 的 10 字节域表示要进行恢复的 Real PlayList 或 Virtual PlayList 的文件名。

图 16 所示 DVRVolume() 语法中的 UIAppInfoVolume 存储有关卷的用户界面应用参数。图 18 示出 UIAppInfoVolume 的语法，现在对其语义进行说明。character_set 的 8 位域表示在 Volume_name 域中编码的字符字母编码方法。该编码方法对应于图 19 所示的值。

name_length 8 位域表示 Volume_name 域中所表示卷名的字节长度。Volume_name 域表示卷名称。从该域左边开始计数的 name_length 数目的字节数为合法字符数，并且表示卷名称。这些合法字符字母之后的值可以为任意值。

Volume_protect_flag 为表示卷中内容是否不加限制地显示给用户的标志。如果该标志设为 1，卷内容只在用户已成功输入 PIN 号码(密码)的情况下允许显示(再现)给用户。如果该标志设为 0，卷内容甚至在用户没有输入 PIN 号码的情况下也允许显示给用户。

当用户已插入一个盘到播放器时，如果该标志已设为 0，或该标志已设为 1，但是用户已成功输入 PIN 号码，记录和/或再现装置 1 显示盘中 PlayList 列表。对再现各个 PlayList 的限制与 Volume_protect_flag 无关，并且由 UIAppInfoVolume 中定义的 playback_control_flag 进行表示。

PIN 由四个从 0 到 9 的数字构成，每个数字根据 ISO/IEC 646 进行编码。ref_thumbnail_index 域表示加到 Volume 的缩略图信息。如果

ref_thumbnail_index 域为不同于 0xFFFF 的值，一个缩略图加到 Volume。缩略图存储在 menu.thmb 文件中。该图像使用 ref_thumbnail_index 值在 menu.thmb 文件中被引用。如果 ref_thumbnail_index 域为 0xFFFF，则表示缩略图已加到 Volume。

- 5 对图 15 所示 info.dvr 语法中的 TableOfPlayList() 进行说明。TableOfPlayList() 存储 PlayList(Real PlayList 和 Virtual PlayList) 文件名。记录在 Volume 中的所有 PlayList 文件包含在 TableOfPlayList() 中，TableOfPlayList() 表示 Volume 中 PlayList 缺省回放序列。

图 20 示出 TableOfPlayList() 语法，现在对其进行说明。TableOfPlayList() 的 version_number 表示示出 TableOfPlayList 版本号的四个字符字母。version_number 必须根据 ISO 646 编码为“0045”。

length 为无符号 32 位整数，表示从紧接 length 域之后到 TableOfPlayList() 末端的 TableOfPlayList() 字节数。number_of_PlayLists 16 位域表示包括 PlayList_file_name 的循环数。该数字必须等于记录在 Volume 中的 PlayList 数。PlayList_file_name 10 字节数字表示 PlayList 文件名。

图 21 示出 TableOfPlayList() 语法的另一结构。图 21 所示的语法包括图 20 所示的语法，并且其中包含 UIAppInfoPlayList。通过包括 UIAppInfoPlayList 这种结构，在读出 TableOfPlayList 时简单地创建菜单图像成为可能。下面说明以使用图 20 所示的语法为前提。

- 20 对图 15 所示 info.dvr 中的 MakersPrivateData 进行说明。MakersPrivateData 用来允许记录记录和/或再现装置 1 的制造者在 MakersPrivateData() 中为不同公司专有应用插入制造者私有数据。每个制造者的私有数据具有标准化 maker_ID，用于标识对其进行定义的制造者。MakersPrivateData() 可包含一个或多个 maker_ID。

- 25 如果预定制造者意图插入私有数据，并且不同制造者的私有数据已包含在 MakersPrivateData() 中，新私有数据加到 MakersPrivateData()，而不删除预先存在的旧私有数据。因此，在本实施例中，多个制造者的私有数据能包含在一个 MakersPrivateData() 中。

图 22 示出 MakersPrivateData 语法。对图 22 所示的 MakersPrivateData() 语法进行说明。TableOfPlayList() 的 version_number 表示示出 TableOfPlayList 版本号的四个字符字母。version_number 必须根据 ISO 646 编码为“0045”。

length 为无符号 32 位整数，表示从紧接 length 域之后到 MakersPrivateData () 末尾的 TableOfPlayList() 字节数。

mpd_blocks_start_address 表示根据相对于 MakersPrivateData() 起始字节的字节数，第一 mpd_block() 的起始端位置。number_of_maker_entries 为 16 5 位无编码整数，提供包括在 MakersPrivateData() 中制造者私有数据的条目数。在 MakersPrivateData() 中必须不能出现具有同一 maker_ID 值的两个或更多制造者私有数据。

mpd_blocks_size 为 16 位无符号整数，提供一个 mpd_block 大小(以 1024 字节为单位)。例如，如果 mpd_blocks_size=1，则表示一个 mpd_block 的大 10 小为 1024 字节。number_of_mpd_blocks 为 16 位无符号整数，提供包含在 MakersPrivateData() 中的 mpd_block 数。maker_ID 为 16 位无符号整数，表示已创建制造者私有数据的 DVR 系统的模型号代码。编码到 maker_ID 的值由授权者进行指定。

maker_model_code 为 16 位无符号整数，表示已创建制造者私有数据的 15 DVR 系统的模型号代码。编码到 maker_model_code 的值由接收到格式授权的制造者进行设置。start_mpd_block_number 为 16 位无符号整数，表示开始制造者私有数据的 mpd_block 号的号码。制造者私有数据的起始端必须与 mpd_block 的起始端对齐。start_mpd_block_number 对应于 mpd_block 循环 20 中的变量 j。

mpd_length 为 32 位无符号整数，表示制造者私有数据的大小。mpd_block 20 为存储私有数据的区域。MakersPrivateData() 中所有 mpd_block 必须具有相同大小。

对 Real PlayList 文件和 Virtual PlayList 文件，换句话说，xxxxx.rpls 和 25 yyyyy.vpls 进行说明。图 23 示出具有相同语法结构的 xxxxx.rpls(Real PlayList) 和 yyyyy.vpls(Virtual PlayList) 语法。每个 xxxxx.rpls 和 yyyyy.vpls 由三个对象，即 PlayList()、PlayListMark() 和 MakersPrivateData() 构成。

PlayListMark_Start_address 表示以相对于 PlayList 文件起始端的字节数为单位，PlayListMark () 的起始地址。相对字节数从 0 开始计数。

MarkersPrivateData_Start_address 表示以相对于 PlayList 文件起始端的字 30 节数为单位，MakersPrivateData() 的起始地址。相对字节数从 0 开始计数。

根据 PlayList 文件的语法插入 padding_word(填充字)，其中 N1 和 N2 为

可选正整数。每个填充字可以采用一个可选值。

虽然已对 `PlayList` 作过简短说明，下面对它进行进一步的说明。除 `Bridge-Clip` 外的所有 `Clip` 中的回放域必须由盘中所有 `PlayList` 进行引用。另外，两个或更多 `Real PlayList` 必须在相同 `Clip` 中不重叠由它们 `PlayItem` 示出的回放域。

参照图 24A、24B 和 24C。对于所有 `Clip`，存在对应的 `Real PlayList`，如图 24A 所示。甚至在完成编辑操作之后，仍遵循这个规则，如图 24B 所示。因此，所有 `Clip` 必须通过引用一个 `Real PlayList` 进行观看。

参照图 24C，`Virtual PlayList` 的回放域必须包含在回放域和 `Bridge-Clip` 回放域中。在盘中必须不出现没有由任何 `Virtual PlayList` 引用的 `Bridge-Clip`。

包含 `PlayItem` 列表的 `Real PlayList` 必须不包含 `SubPlayItem`。`Virtual PlayList` 包含 `PlayItem` 列表，并且，如果包含在 `PlayList()` 中的 `CPI_type` 为 `EP_map`，并且 `PlayList_type` 为 0 (包含视频和音频的 `PlayList`)，`Virtual PlayList` 可以包含一个 `SubPlayItem`。在本实施例的 `PlayList()` 中，`SubPlayItem` 仅用于音频后期记录。一个 `Virtual PlayList` 所拥有的 `SubPlayItem` 数必须为 0 或 1。

以下说明 `PlayList`。图 25 示出 `PlayList` 语法，现在对其进行说明。`version_number` 表示示出 `PlayList()` 版本号的四个字符字母。`version_number` 根据 ISO646 编码为 "0045"。`length` 为 32 位无符号整数，表示从紧接 `length` 域之后到 `PlayList()` 末端的 `PlayList()` 的总字节数。`PlayList_type`，它的一个例子如图 26 所示，为表示 `PlayList` 类型的 8 位域。

`CPI_type` 为 1 位标志，表示由 `PlayItem()` 和 `SubPlayItem()` 引用的 `Clip` 的 `CPI_type` 类型值。在一个 `PlayList` 所引用的所有 `Clip` 的 `CPI` 中定义的 `CPI_types` 必须具有相同的值。`number_of_PlayItems` 为 16 位域，表示出现在 `PlayList` 中的 `PlayItem` 数。

对应于预定 `PlayItem()` 的 `PlayItem_id` 通过 `PlayItem()` 出现在包含 `PlayItem()` 的循环中的顺序进行定义。`PlayItem_id` 从 0 开始。`number_of_SubPlayItems` 为 16 位域，表示 `PlayList` 中 `SubPlayItem` 数。该值为 0 或 1。附加音频流路径(音频流路径)是一种子路径。

对图 25 所示 `PlayList` 语法的 `UIAppInfoPlayList` 进行说明。`UIAppInfoPlayList` 存储有关 `PlayList` 的用户界面应用参数。图 27 示出 `UIAppInfoPlayList` 语法，现在对其进行说明。`character_set` 为 8 位域，表示

PlayList_name 域中编码的字符字母编码方法。该编码方法对应于遵循图 19 所示表的值。

name_length 为 8 位域，表示 PlayList_name 域中所表示 PlayList 名称的字节长度。PlayList_name 域表示 PlayList 名称。从该域左边开始计数的 name_length 数目的字节数为合法字符数，并且表示 PlayList 名称。这些合法字符字母之后的值可以为任意值。

record_time_and_date 为 56 位域，存储记录 PlayList 的日期和时间。该域为以二 - 十进制(binary coded decimal, BCD)编码的用于年/月/日/小时/分/秒的 14 个数字。例如，2001/12/23:01:02:03 编码为“0x20011223010203”。

10 duration 为 24 位域，表示以小时/分/秒为单位，PlayList 的总回放时间。该域为用 BCD 编码的六个数字。例如，01:45:30 编码为“0x014530”。

valid_period 为 32 位域，表示 PlayList 的有效时间范围。该域为用 4 位二 - 十进制(BCD)编码的 8 个数字。例如，valid_period 用在记录和/或再现装置 1 中，当有效范围已过，PlayList 将自动删除。例如，2001/05/07 编码为“0x20010507”。

15 maker_ID 为 16 位无符号整数，表示最后更新其 PlayList 的 DVR 播放器(记录和/或再现装置 1)的制造者。编码到 maker_ID 的值被分配给 DVR 格式的授权者。maker_code 为 16 位无符号整数，表示最后更新其 PlayList 的 DVR 播放器的型号代码。编码到 maker_code 的值由接收到 DVR 格式授权的制造者进行确定。

如果 playback_control_flag 标志设为 1，仅当用户成功输入 PIN 号码，才再现其 PlayList。如果该标志设为 0，用户可以观看 PlayList，而没有必要输入 PIN 号码。

25 如果 write_protect_flag 设为 1，不能删除或改变除 write_protect_falg 之外的 PlayList 内容。如果该标志设为 0，用户可以自由删除或修改 PlayList。如果该标志设为 1，记录和/或再现装置 1 在用户继续删除、编辑或覆写 PlayList 之前显示要求用户重新确认的消息。

30 可以存在其中 write_protect_flag 设为 0 的 Real PlayList，可以存在引用 Real PlayList 的 Clip 的 Virtual PlayList，并且 Virtual PlayList 的 write_protect_flag 可以设为 1。如果用户想要删除 Real PlayList，记录和/或再现装置 1 在删除 Real PlayList 之前向用户发出一个存在前述 Virtual

PlayList，或“最小化”Real PlayList 的警告。

如果 `is_played_flag` 设为 1，如图 28B 所示，则表示自从记录 PlayList 以来至少对它再现过一次，相反，如果它设为 0，则表示自从记录 PlayList 以来甚至没有对它再现过一次。

- 5 Archive 为 2 位域，表示 PlayList 是原始的或拷贝的，如图 28C 所示。
 `ref_thumbnail_index` 域表示代表 PlayList 的缩略图信息。如果
 `ref_thumbnail_index` 域为不同于 0xFFFF 的值，一个代表 PlayList 的缩略图加
 到该 PlayList，该 PlayList 存储在 `menu.thmb` 文件中。该图像使用
 `ref_thumbnail_index` 值在 `menu.thumb` 文件中进行引用。如果
 10 `ref_thumbnail_index` 域为 0xFFFF，则表示没有代表 PlayList 的缩略图加到该
 PlayList 中。

- 以下说明 PlayItem。一个 PlayItem() 基本上包含下列数据：
 `Clip_Information_file_name`，用于指定 Clip 文件名；`IN_time` 和 `OUT_time`，
 成对指定 Clip 的回放域；`STC_sequence_id`，在 PlayList() 中定义的 `CPI_type`
 15 为 `EP_map` 类型的情况下由 `IN_time` 和 `OUT_time` 进行引用；和
 `Connection_Condition`，表示前面 PlayItem 和当前 PlayItem 的连接条件。

- 如果 PlayList 由两个或更多 PlayItem 构成，这些 PlayItem 在 PlayList 的
 全局时间轴上排成一行，既没有时间间隔，也没有时间重叠。如果在 PlayList()
 中定义的 `CPI_type` 为 `EP_map` 类型，并且当前 PlayList 没有 `BridgeSequence()`，
 20 `IN_time` 和 `OUT_time` 对必须表示与 `STC_sequence_id` 所指定的相同的 STC
 连续域时间。这种例子如图 29 所示。

- 图 30 示出这样一种情况，在 PlayList() 中定义的 `CPI_type` 为 `EP_map` 类
 型的情况下，并且如果当前 PlayItem 具有 `BridgeSequence()`，应用如下说明
 的规则。当前 PlayItem 之前的 PlayItem 的 `IN_time`，如 `IN_time1` 所示，表示
 25 当前 PlayItem 的 `BridgeSequenceInfo()` 中指定的 Bridge-Clip 中的时间。该
 `OUT_time` 必须遵从随后将要说明的编码限制。

- 当前 PlayItem 的 `IN_time`，如 `IN_time2` 所示，表示当前 PlayItem 的
 `BridgeSequenceInfo()` 中指定的 Bridge-Clip 中的时间。该 `IN_time` 必须遵从后
 面将要说明的编码限制。当前 PlayItem 的 `OUT_time`，如 `OUT_time2`
 30 所示，表示由当前 PlayItem 的 `STC_sequence_id` 指定的 STC 连续域时间。

 如果 PlayList() 的 `CPI_type` 为 `TU_map` 类型，PlayItem 的 `IN_time` 和

OUT_time, 成对表示相同 Clip AV 流时间, 如图 31 所示。

PlayItem 语法如图 32 所示。对于 PlayItem 语法, 如图 32 所示, Clip_information_file_name 域表示 Clip Information 的文件名。由该 Clip Information 文件的 ClipInfo()定义的 Clip_stream_type 必须表示 Clip AV 流。

5 STC_sequence_id 为 8 位域, 并且表示由 PlayItem 引用的连续 STC 域的 STC_sequence_id。如果 PlayList()中指定的 CPI_type 为 TU_map 类型, 该 8 位域没有意义, 并且设为 0。IN_time 为 32 位域, 并且用来存储 PlayItem 的回放开始时间。IN_time 语义随在 PlayList()中定义的 CPI_type 不同, 如图 33 所示。

10 OUT_time 为 32 位域, 并且用来存储 PlayItem 的回放结束时间。OUT_time 语义随在 PlayList()中定义的 CPI_type 不同, 如图 34 所示。

Connection_condition 为 2 位域, 表示前面 PlayItem 和当前 PlayItem 之间的连接条件, 如图 35 所示。图 36A 到 36D 示出图 35 所示 Connection_condition 的各种状态。

15 参照图 37 对 BridgeSequenceInfo 进行描述。该 BridgeSequenceInfo 为当前 PlayItem 的辅助信息, 并且包括如下信息。即, BridgeSequenceInfo 包括用于指定 Bridge_Clip AV 文件的 Bridge_Clip_Information_file_name 和指定对应 Clip Information 文件的 Bridge_Clip_Information_file_name(图 45)。

20 它还是由前面 PlayItem 引用的 Clip AV 流的源包地址。Bridge_Clip AV 流的第一源包连接在该源包之后。该地址称作 RSPN_exit_from_previous_Clip。它还是由当前 PlayItem 引用的 Clip AV 流的源包地址。Bridge_Clip AV 流文件的最后源包连接在该源包之前。该地址称作 RSPN_enter_to_previous_Clip。

25 在图 37 中, RSPN_arrival_time_discontinuity 表示其中存在到达时基的不连续点的 Bridge_Clip AV 流的源包地址。该地址定义在 ClipInfo()中(图 46)。

30 图 38 示出 BridgeSequenceInfo 语法。参照图 38 所示的 BridgeSequenceInfo 语法, Bridge_Clip_Information_file_name 域表示与 Bridge_Clip_Information_file 对应的 Clip Information 文件的文件名。在该 Clip Information 文件的 ClipInfo()中定义的 Clip_stream_type 必须表示“Bridge_Clip AV 流”。

RSPN_exit_from_previous_Clip 32 位域为由前面 PlayItem 引用的 Clip AV 流的源包相对地址。Bridge_Clip AV 流文件的第一源包连接在该源包之后。RSPN_exit_from_previous_Clip 具有以源包数为单位的大小，并且用定义在 ClipInfo() 中的 offset_SPN 值从由前面 PlayItem 引用的 Clip AV 流文件第一源包进行计数。

RSPN_enter_to_current_Clip 32 位域为由当前 PlayItem 引用的 Clip AV 流的源包相对地址。Bridge_Clip AV 流文件的最后源包连接在该源包之前。RSPN_enter_to_current_Clip 具有以源包数为单位的大小。RSPN_enter_to_current_Clip 用定义在 ClipInfo() 中的 offset_SPN 值从由当前 PlayItem 引用的 Clip AV 流文件第一源包作为初始值进行计数。

参照图 39 对 SubPlayItem 进行说明。只有 PlayList() 的 CPI_type 为 EP_map 类型，才允许使用 SubPlayItem()。在本实施例中，SubPlayItem 仅用于音频后期记录。SubPlayItem() 包括下面数据。首先，它包括 Clip_Information_file_name, 用于指定由 PlayList 中的子路径引用的 Clip。

它还包括 SubPath_IN_time 和 SubPath_OUT_time, 用于指定 Clip 中子路径回放域。另外，它包括 sync_PlayItem_id 和 start_PTS_of_PlayItem, 用于在主路径时间轴上指定开始子路径再现的时间。由于子路径引用的 Clip AV 流必须不包含 STC 不连续点(系统时基的不连续点)。用在子路径中的音频样本时钟锁定于主路径的音频样本时钟。

图 40 示出 SubPlayItem 语法。参照图 40 所示的 SubPlayItem 语法，Clip_Information_file_name 域表示 Clip Information 文件的文件名，并且由 PlayList 中的子路径进行使用。定义在该 ClipInfo() 中的 Clip_stream_type 必须表示 Clip AV 流。

sync_PlayItem_id 8 位域表示子路径类型。在此，仅设置 '0x00'，如图 41 所示，而其它值被保留以备将来使用。

sync_PlayItem_id 8 位域表示在主路径的时间轴上包含子路径回放开始时间的 PlayItem 的 PlayItem_id。与预定 PlayItem 对应的 PlayItem_id 值定义在 PlayList() 中(图 25)。

sync_start_PTS_of_PlayItem 32 位域表示在主路径的时间轴上的子路径回放开始时间，并且表示由 sync_PlayItem_id 引用的 PlayItem 的 PTS(presentation time stamp, 呈现时间戳)高 32 位。SubPath_IN_time 的高 32

位域存储子路径回放开始时间。SubPath_IN_time 表示对应于子路径中第一呈现单元的 33 位 PTS 的高 32 位。

SubPath_OUT_time 的高 32 位域存储子路径回放结束时间。SubPath_OUT_time 表示由下面公式计算的 Presentation_end_TS 值的高 32 5 位：

$$\text{Presentation_end_TS} = \text{PTS_OUT} + \text{AU_duration}$$

其中，PTS_OUT 为对应于 SubPath 最后呈现单元的 33 位长度的 PTS，并且 AU_duration 为 SubPath 的最后呈现单元的基于 90kHz 的显示周期。

10 下一步，对图 23 所示 xxxxx.rpls 和 yyyyy.vpls 中的 PlayListMark()进行说明。与 PlayList 相关的标记信息存储在该 PlayListMark 中。图 42 示出 PlayListMark 语法。参照图 42 所示的 PlayListMark 语法，version_number 为表示 PlayListMark()版本号的四个字符字母。version_number 根据 ISO646 编码为"0045"。

length 为 32 位无符号整数，表示从紧接 length 域之后到 PlayListMark() 15 末端的 PlayListMark()字节数。number_of_PlayListMarks 为 16 位无符号整数，表示存储在 PlayListMark 中的标记数。number_of_PlayListMarks 可以为 0。mark_type 为表示标记类型的 8 位域，并且用图 43 所示的表进行编码。

mark_time_stamp 32 位域存储表示标记所指定点的时间戳。mark_time_stamp 的语义随定义在 PlayList()中的 CPI_type 不同，如图 44 所 20 示。PlayItem_id 为 8 位域，指定其中置放标记的 PlayItem。对应于预定 PlayItem 的 PlayItem_id 值定义在 PlayList()中(参见图 25)。

character_set 8 位域表示在 mark_name 域中编码的字符字母编码方法。该编码方法对应于图 19 所示的值。name_length 8 位域表示 mark_name 域中 25 所示标记名的字节长度。mark_name 域表示 mark_name 域中所示的标记名称。对应于从该域左边开始计数的 name_length 数目的字节数为有效字符字母，并且表示标记名称。在 mark_name 域中，在这些有效字符字母之后的值可以为任意值。

ref_thumbnail_index 域表示加到标记的缩略图信息。如果 ref_thumbnail_index 域不是 0xFFFF，一个缩略图加到其标记，该缩略图存储 30 在 mark.thmb 文件中。该图像使用 ref_thumbnail_index 值在 mark.thmb 文件中进行引用，随后对此进行说明。如果 ref_thumbnail_index 域为 0xFFFF，

则表示没有缩略图加到该标记中。

现在说明 Clip Information 文件。zzzzz.clpi(Clip Information 文件)由六个对象构成,如图 45 所示。它们是 ClipInfo()、STC_Info()、Program()、CPI()、ClipMark()和 MakersPrivateData()。对于 AV 流(Clip AV 流或 Bridge-Clip AV 流)和对应 Clip 信息文件,使用相同的数字串“zzzzz”。

解释图 45 所示的 zzzzz.clpi(Clip Information 文件)语法。ClipInfo_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,ClipInfo()的起始端地址。相对字节数从 0 开始计数。

STC_Info_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,STC_Info 的起始端地址。ProgramInfo_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,ProgramInfo()的起始端地址。相对字节数从 0 开始计数。CPI_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,CPI()的起始端地址。相对字节数从 0 开始计数。

ClipMark_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,ClipMark()的起始端地址。相对字节数从 0 开始计数。MakersPrivateData_Start_address 表示以相对于 zzzzz.clpi 文件的起始端字节的相对字节数为单位,MakersPrivateData()的起始端地址。相对字节数从 0 开始计数。padding_word 根据 zzzzz.clpi 的语法进行插入。N1、N2、N3、N4 和 N5 必须为 0 或可选正整数。各个填充值也可以采用可选值。

现在说明 ClipInfo。图 46 示出 ClipInfo 语法。图 46 示出 ClipInfo 语法。对应 AV 流文件(Clip AV 流或 Bridge-Clip AV 流文件)的属性信息存储在 ClipInfo()中。

参照图 46 所示的 ClipInfo 语法,version_number 为表示 ClipInfo()版本号四个字符字母。version_number 必须根据 ISO 646 编码为“0045”。length 为 32 位无符号整数,表示从紧接 length 域之后到 ClipInfo()末端的 ClipInfo 字节数。Clip_stream_type 8 位域表示对应于 Clip Information 文件的 AV 流类型,如图 47 所示。随后将对各个 AV 流的流类型进行说明。

offset_SPN 32 位域给出 AV 流(Clip AV 流或 Bridge-Clip AV 流文件)的第一源包的第一源包号的源包号偏移值。当 AV 流文件首先记录在盘上时,该 offset_SPN 必须为 0。

参照图 48，当通过编辑删除 AV 流文件的开始部分时，off_SPN 可以采用不同于 0 的值。在本实施例中，引用 offset_SPN 的相对源包号(相对地址)经常以 RSPN_{xxx} 的形式进行描述，其中，修改 xxx，使 RSPN_{xxx} 为 RAPN_EP_start。相对源包号以源包数为单位改变大小，并且以 offset_SPN 5 值为初始值从 AV 流文件的第一源包号进行计数。

从 AV 流文件的第一源包到由相对源包号(SPN_{xxx})引用的源包的源包数用下面方程进行计算：

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}。$$

图 48 示出 offset_SPN 为 4 的例子。

10 TS_recording_rate 为 24 位无符号整数，提供 AV 流所需的到 DVR 驱动器(写入单元 22)或从 DVR 驱动器(读出单元 28)的输入/输出位率。record_time_and_date 为 56 位域，用于存储记录与 Clip 对应的 AV 流的日期和时间，并且是采用 4 位二-十进制(BCD)的年/月/日/小时/分/秒 14 位数字编码表示。例如，2001/12/23:01:02:03 编码为“0x20011223010203”。

15 duration 为 24 位域，表示用到达时基时钟的小时/分/秒表示的 Clip 总回放时间。该域为用 4 位二-十进制(BCD)编码的六位数字。例如，01:45:30 编码为“0x014530”。

time_controlled_flag 标志表示 AV 流文件的记录模式。如果该 time_controlled_flag 设为 1，则表示记录模式为文件大小与记录时间成比例 20 的模式，条件由下面方程给出：

$$\begin{aligned} & \text{Ts_average_rate} * 192 / 188 * (1 - \text{start_time}) - \alpha \leq \text{size_clip}(t) \\ & \leq \text{Ts_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha \end{aligned}$$

其中，Ts_average_rate 为用字节/秒表示的 AV 流文件传输流的平均位率。

在上述方程中，t 表示以秒为单位的时间，而 start_time 为记录 AV 流文 25 件的第一源包时的时间点。size_clip(t) 为 10*192 字节，并且 α 是依赖于 TS_average_rate 的常数。

如果 time_controlled_flag 设为 0，则表示不控制记录模式，从而记录时间与 AV 流的文件大小成正比。例如，输入传输流以透明方式进行记录。

如果 time_controlled_flag 设为 1，TS_average_rate 24 位域表示用在上述 30 方程中的 TS_average_rate 值。如果 time_controlled_flag 设为 0，该域没有意义，并且必须设为 0。例如，可变位率传输流采用下面顺序进行编码：首先，

传输率设为 TS_recording_rate 值。视频流用可变位率进行编码。不使用空包对传输包进行间歇性的编码。

RSPN_arrival_time_discontinuity 32 位域为 Bridge-Clip AV 流文件中所产生到达时间不连续点的相对地址。RSPN_arrival_time_discontinuity 以源包数为单位改变大小,并且用在 ClipInfo()中定义的 offset_SPN 值从 Bridge-Clip AV 流文件的第一源包进行计数。Bridge-Clip AV 流文件中的绝对地址根据前述方程进行计算:

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}。$$

reserved_for_system_use 144 位域为系统保留。如果 is_format_identifier_valid 标志为 1, 它表示 format_identifier 域有效。如果 is_format_identifier_valid 标志为 1, 它表示 format_identifier 域有效。如果 is_original_network_ID_valid 标志为 1, 它表示 is_transport_stream_ID_valid 域有效。如果 is_transport_stream_ID_valid 标志为 1, 它表示 transport_stream_ID 域有效。如果 is_service_ID_valid 标志为 1, 它表示 service_ID 域有效。

如果 is_country_code_valid 标志为 1, 它表示 country_code 域有效。format_identifier 32 位域表示传输流的登记描述符(在 ISO/IEC13818-1 中定义)所拥有的 format_identifier 值。original_network_ID 16 位域表示传输流中定义的 original_network_ID 值。

service_ID 16 位域表示传输流中定义的 service_ID 值。country_code 24 位域表示 ISO3166 所定义的国家代码。每个字符码采用 ISO8859-1 进行编码。例如, Japan 用“JPN”表示, 并且编码为“0x4A 0x50 0x4E”。stream_format_name 是 15 个 ISO-646 字符码, 表示提供传输流的流定义的格式组织的名称。该域中不合法字节的值为‘0XFF’。

format_identifier, original_network_ID, transport_stream_ID, service_ID, country_code 和 stream_format_name 表示传输流的服务提供者。这就允许认识到音频流或视频流的编码限制和不同于音频视频流或 SI(service information, 服务信息)的私有数据流的流定义。这些信息可以用来检查是否解码器能够对流进行解码。如果解码可能, 在开始解码之前, 可以使用该信息对解码器系统进行初始化。

现在说明 STC_info。不含 STC 不连续点(系统时间基的不连续点)的

MPEG-2 传输流中的时间域被称作 STC_sequence。在该 Clip 中，由 STC_sequence_id 值指定 STC_sequence。图 50A 和 50B 示出连续 STC 域。如后所述,虽然 Clip 最大时长受到限制,但在同一 STC_sequence 中禁止出现相同的 STC 值。如果 AV 流含有 N 个 STC 不连续点,其中 $N > 0$, Clip 系统
5 时间分化为 $(N+1)$ 个 STC_sequence。

STC_info 存储产生 STC 不连续(系统时间基的不连续)点的地址。如图 51 所示, RSPN_STC_start 表示该地址, 并且从第 $(k+1)$ RSPN_STC_start 所引用源包的到达时间点开始, 到最后一个源包的到达时间点结束。

图 52 示出 STC_info 的语法。参照图 52 所示的 STC_info 语法,
10 version_number 是表示 STC_info() 版本号的四个字符字母。version_number 必须根据 ISO 646 编码为“0045”。

length 为 32 位无符号整数, 表示从紧接该 length 域之后到 STC_info 末端的 STC_info() 的字节数。如果 CPI() 的 CPI_type 为 TU_map 类型, 该 length 域可以设为 0。如果 CPI() 的 CPI_type 为 EP_map 类型, num_of_STC_sequence
15 值必须不小于 1。

num_of_STC_sequence 为 8 位无符号整数, 表示 Clip 中的序列号。该值表示紧接该域之后的循环数。与预定 STC_sequence 对应的 STC_sequence_id 根据在含有 RSPN_STC_start 的循环中对应于 STC_sequence 的 RSPN_STC_start 的出现顺序进行定义。STC_sequence_id 从 0 开始。

20 RSPN_STC_start 32 位域表示 AV 流文件上 STC_sequence 的开始地址。RSPN_STC_start 表示 AV 流文件中产生系统时基不连续的地址。RSPN_STC_start 还可以是包含 AV 流中新系统时基的第一 PCR 的源包相对地址。RSPN_STC_start 具有基于源包数的大小, 并且以在 ClipInfo() 中定义的 offset_SPN 值为初始值从 AV 流文件的第一源包进行计数。在该 AV 流文
25 件中, 根据上述方程计算绝对地址, 也就是:

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

现在参照图 53 说明图 45 所示的 zzzz.clip 语法。Clip 中具有如下特征的时间域称作 program_sequence。这些特征是, PCR_PID 值不变, 视频基本流数目也不变, 各个视频流中的 PID 值不变, 由其 VideoCodingInfo 定义的编码信息不变, 音频基本流数目也不变, 各个音频流的 PID 值不变, 并且由其
30 AudioCodingInfo 定义的编码信息不变,

program_sequence 在同一时间点只有一个系统时基。program_sequence 在同一时间点有唯一的 PMT。ProgramInfo() 存储 program_sequence 开始点的地址。RSPN_program_sequence-start 表示该地址。

图 54 示出 ProgramInfo 的语法。参照图 54 所示的 ProgramInfo，
5 version_number 是表示 ProgramInfo () 版本号的四个字符字母。version_number 必须根据 ISO646 编码为“0045”。

length 为 32 位无符号整数，表示从紧接该 length 域之后到 ProgramInfo() 末端的 ProgramInfo() 字节数。如果 CPI() 的 CPI_type 为 TU_map 类型，该 length 域可以设为 0。如果 CPI() 的 CPI_type 为 EP_map 类型，num_of_programs 值
10 必须不小于 1。

num_of_program_sequences 为 8 位无符号整数，表示 Clip 中的 program_sequence 数目。该值表示紧接该域之后的循环数。如果 Clip 中的 program_sequence 不变，program_sequence 的数目必须设为 1。RSPN_program_sequence_start 32 位域表示 program_sequence 在 AV 流文件上
15 开始的相对地址。

RSPN_program_sequence_start 的大小以源包数为单位，并且用在 ClipInfo() 中定义的 offset_SPN 值从 AV 流文件的第一源包进行计数。AV 流文件中的绝对地址按照如下方式进行计算：

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

20 循环语法中的 RSPN_program_sequence_start 值必须以升序出现。

PCR_PID 16 位域表示含有对 program_sequence 有效的有效 PCR 域的传输包的 PID。number_of_audios 8 位域表示包含 audio_stream_PID 和 AudioCodingInfo() 的循环数。video_stream_PID 16 位域表示含有对其 program_sequence 有效的视频流的传输包的 PID。紧接在该域之后的
25 VideoCodingInfo() 必须说明其 video_stream_PID 所引用的视频流内容。

audio_stream_PID 16 位域表示含有对其 program_sequence 有效的音频流的传输包的 PID。紧接在该域之后的 AudioCodingInfo() 必须说明其 audio_stream_PID 所引用的视频流内容。

video_stream_PID 值在循环语法中的顺序必须等于视频流在对 program_sequence 有效的 PMT 中的 PID 编码顺序。另外，audio_stream_PID
30 值在循环语法中的顺序必须等于音频流在对 program_sequence 有效的 PMT

中的 PID 编码顺序。

图 55 示出图 54 所示的 ProgramInfo 语法中的 VideoCodingInfo 的语法。参照图 55 所示的 VideoCodingInfo 语法，video_format 8 位域表示与 ProgramInfo()的 video_stream_PID 对应的视频格式，如图 56 所示。

- 5 参照图 57，frame_rate 8 位域表示与 ProgramInfo()的 video_stream_PID 对应的视频帧率。display_aspect_ratio8 位域表示与 ProgramInfo()的 video_stream_PID 对应的视频显示纵横比。

- 图 59 示出图 54 所示的 ProgramInfo 语法中的 AudioCodingInfo 的语法。参照图 59 所示的 AudioCodingInfo 语法，Audio_format 8 位域表示与 ProgramInfo()的 audio_stream_PID 对应的音频编码方法，如图 60 所示。

audio_component_type 8 位域表示如图 61 所示的与 ProgramInfo()的 audio_stream_PID 对应的音频组件类型，而 sampling_frequency 8 位域表示如图 62 所示的与 ProgramInfo()的 audio_stream_PID 对应的音频取样频率。

- 对图 45 所示的 zzzzz.clip 语法中 CPI(Characteristics Point Information, 特征点信息)进行说明。CPI 用来将 AV 流中的时间信息与其文件中的地址相关联。CPI 有两种类型，即 EP_map 和 TU_map。在图 63 中，如果 CPI()中的 CPI_type 为 EP_map，其 CPI()包含 EP_map。在图 64 中，如果 CPI()中的 CPI_type 为 TU_map，其 CPI()包含 TU_map。一个 AV 流具有一个 EP_map 或一个 TU_map。如果 AV 流是 SESF 传输流，对应的 Clip 必须有 EP_map。

- 20 图 65 示出 CPI 语法。参照图 65 所示的 CPI 语法，version_number 是表示该 CPI()版本号的四个字符字母。version_number 必须根据 ISO646 编码为“0045”。length 为 32 位无符号整数，表示从紧接该 length 域之后到 CPI()末端的字节数。CPI_type 为 1 位标志，并且表示 Clip 的 CPI 类型，如图 66 所示。

- 25 对图 65 所示的 CPI 语法中的 EP_map 进行说明。有两种类型的 EP_map，即视频流 EP_map 和音频流 EP_map。EP_map 中的 EP_map_type 区分这两种 EP_map 类型。如果 Clip 包含一个或多个视频流，必须使用视频流 EP_map。如果 Clip 不包含视频流，但是包含一个或多个音频流，必须使用音频流 EP_map。

- 30 参照图 67 对视频流 EP_map 进行说明。视频流 EP_map 具有数据 stream_PID、PTS_EP_start 和 RSPN_EP_start。stream_PID 表示传输视频流

的传输包的 PID。PTS_EP_start 表示从视频流的序列头开始的访问单元的 PTS。RSPN_RP_start 表示 AV 流中包括 PTS_EP_start 所引用访问单元的第一字节的源包地址。

逐个字地从具有相同 PID 的传输包所传输的视频流创建称作 EP_map_for_one_stream_PID() 的子表。如果 Clip 中存在多个视频流，EP_map 可以包含多个 EP_map_for_one_stream_PID()。

音频流 EP_map 具有数据 stream_PID、PTS_EP_start 和 RSPN_EP_start。stream_PID 表示传输音频流的传输包的 PID。PTS_EP_start 表示音频流中访问单元的 PTS。RSPN_RP_start 表示 AV 流的包括 PTS_EP_start 所引用访问单元的第一字节的源包地址。

逐个字地从具有相同 PID 的传输包所传输的音频流创建称作 EP_map_for_one_stream_PID() 的子表。如果 Clip 中存在多个音频流，EP_map 可以包含多个 EP_map_for_one_stream_PID()。

参照 EP_map 和 STC_info 之间的关系，在一个与 STC 中不连续点无关的表中创建一个 EP_map_for_one_stream_PID()。比较 RSPN_EP_start 值和 STC_Info() 中定义的 RSPN_STC_start 值示出属于各个 STC_sequence 的 EP_map 数据的边界(参见图 68)。EP_map 必须有一个相同 PID 所传输连续流范围的 EP_map_for_one_stream_PID。在图 69 所示的情况下，节目 #1 和节目 #3 具有相同视频 PID，然而数据范围却不连续，从而必须为每个节目提供 EP_map_for_one_stream_PID。

图 70 示出 EP_map 的语法。作为对图 70 所示 EP_map 语法的说明，EP_type 为 4 位域，并且表示 EP_map 入口点类型，如图 71 所示。EP_type 表示紧接该域之后数据域的语义。如果 Clip 包括一个或多个视频流，EP_type 必须设为 0(‘视频’)。作为替换，如果 Clip 不包括视频流，而包括一个或多个音频流，EP_type 必须设为 1(‘音频’)。

Number_of_stream_PIDs 16 位域表示以 EP_map() 中 Number_of_stream_PIDs 为变量的 for-循环的循环次数。Stream_PID(k) 16 位域表示传输 EP_map_for_one_stream_PID 所引用第 k 个基本流(视频或音频流)(num_EP_entries(k))的传输包的 PID。如果 EP_type 为 0(‘视频’)，它的基本流必须是视频流。如果 EP_type 为 1(‘音频’)，它的基本流必须是音频流。

num_EP_entries(k) 16 位域表示 EP_map_entries(k) 所引用的 num_EP_entries(k)。EP_map_for_one_stream_PID_Start_address(k): 该 32 位域表示 EP_map_for_one_stream_PID(num_EP_entries(k))在 EP_map()中开始的相对地址位置。用来自 EP_map()第一字节的大小表示该值。

- 5 必须根据 EP_map()语法插入 Padding_word。X 和 Y 必须是可选正整数。各个填充字可以采用任何可选值。

图 72 示出 EP_map_for_one_stream_PID 语法。作为对图 72 所示 EP_map_for_one_stream_PID 语法的说明, PTS_EP_start 32 位域的语义与 EP_map()中定义的 EP_map 不同。如果 EP_type 为 0(‘视频’), 该域具有从视频流序列头开始的访问单元的 33 位精度 PTS 的高 32 位。如果 EP_type 为 1(‘音频’), 该域具有音频流访问单元的 33 位精度 PTS 的高 32 位。

RSPN_EP_start 32 位域的语义与 EP_map()中定义的 EP_type 不同。如果 EP_type 为 0(‘视频’), 该域表示 AV 流中包含 PTS_EP_start 所引用访问单元序列头第一字节的源包的相对地址。作为替换, 如果 EP_type 为 1(‘音频’), 该域表示 AV 流中包含 PTS_EP_start 所引用访问单元音频流中第一字节的源包的相对地址。

RSPN_EP_start 的大小以源包数为单位, 并且用在 ClipInfo()中定义的 offset_SPN 值作为初始值, 从 AV 流文件第一源包进行计数。AV 流文件中的绝对地址按照如下方式进行计算:

$$20 \quad \text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}.$$

注意, 语法中的 RSPN_EP_start 的值必须以升序出现。

现在参照图 73 对 TU_map 进行说明。TU_map 根据源包到达时钟(到达时基的时钟)形成时间轴。该时间轴称为 TU_map_time_axis。用 TU_map()中的 offset_time 表示 TU_map_time_axis 的原点。TU_map_time_axis 从 25 offset_time 以预定单元进行划分, 该单元称为 time_unit。

AV 流的每个 time_unit 中, 第一完整形式的源包在 AV 流文件的地址存储在 TU_map 中。该地址称为 RSPN_time_unit_start。在 TU_map_time_axis 上第 k(k ≥ 0)个 time_unit 的开始时间称为 TU_start_time(k)。该值根据如下方程进行计算:

$$30 \quad \text{TU_start_time}(k) = \text{offset_time} + k * \text{time_unit_size}.$$

注意, TU_start_time(k)的精度为 45kHz。

图 74 示出 TU_map 语法。对图 74 所示的 TU_map 语法进行说明，offset_time 32 位域给出相对于 TU_map_time_axis 的偏移时间。该值表示 Clip 中相对于第一 time_unit 的偏移时间。Offset_time 的大小以来自 27MHz 精度到达时钟的 45kHz 时钟为单位。如果作为新 Clip 记录 AV 流,offset_time 必须设为 0。

time_unit_size 32 位域提供 time_unit 的大小，并且基于来自 27MHz 精度到达时钟的 45kHz 时钟为单位。最好，time_unit_size 不大于 1 秒 (time_unit_size ≤ 45000)。number_of_time_unit_entries 32 位域表示存储在 TU_map() 中的条目数。

10 RSN_time_unit_start 32 位域表示 AV 流中每个 time_unit 开始点的相对地址。RSN_time_unit_start 的大小以源包数为单位，并且用在 ClipInfo() 中定义的 offset_SPN 值作为初始值，从 AV 流文件第一源包进行计数。AV 流文件中的绝对地址按照如下方式进行计算：

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}.$$

15 注意，在语法的 for-循环中 RSPN_time_unit_start 的值必须以升序出现。如果第 k+1 个 time_unit 中没有源包，第 k+1 个 RSN_time_unit_start 必须等于第 k 个 RSN_time_unit_start。

对图 45 所示 zzzzz.clip 语法中的 ClipMark 进行说明，ClipMark 是与 clip 有关的标记信息，并且存储在 ClipMark 中。该标记不是由用户进行设置，而是由记录器(记录和/或再现装置 1)进行设置。

图 75 示出 ClipMark 的语法。对图 75 所示的 ClipMark 语法进行说明。version_number 是表示该 ClipMark 版本号的四个字符字母。version_number 必须根据 ISO646 编码为“0045”。

length 是 32 位无符号整数，表示从紧接 length 域之后到 ClipMark () 末端的 ClipMark() 字节数。number_of_clip_marks 为 16 位无符号整数，表示存储在 ClipMark 中的标记数，并且可以等于 0。mark_type 为表示标记类型的 8 位域，并且根据图 76 所示的表进行编码。

mark_time_stamp 为 32 位域，并且存储表示带指定标记的指针的时间戳。mark_time_stamp 语义随在 PlayList() 中定义的 CPI_type 不同，如图 77 所示。

如果 CPI() 中的 CPI_type 表示 EP_map 类型，该 8 位域表示放置

mark_time_stamp 的连续 STC 域的 STC_sequence_id。如果 CPI() 中的 CPI_type 表示 TU_map 类型，该 8 位域没有意义，但是被设为 0。Character_set 8 位域表示在 mark_name 域中编码的字符字母的表示方法。编码方法对应于图 19 所示的值。

- 5 name_length 8 位域表示 mark_name 域所示标记名的字节长度。该 mark_name 域表示标记名。从该域左边对应于 name_length 数目的字节数为有效字符数，并且表示标记名。mark_name 域中，紧接这些有效字符字母之后的值可以是任意的。

- 10 ref_thumbnail_index 域表示附加到标记上的缩略图信息。如果 ref_thumbnail_index 域值不等于 0xFFFF，将缩略图添加到其标记，该缩略图存储在 mark.thmb 文件中。使用 ref_thumbnail_index 值引用 mark_thmb 文件中的该图。如果 ref_thumbnail_index 域值等于 0xFFFF，则不将缩略图添加到其标记上。

- 15 已经参照图 22 对 MakerPrivateData 进行过说明，因此在此不作具体说明。

下一步，对 thumbnail_information 进行说明。缩略图存储在 menu.thmb 或 mark.thmb 文件中。该文件具有相同的语法结构，并且拥有单个 Thumbnail()。Menu.thmb 存储表示各个 Playlist 的图。全部菜单缩略图存储在单个 menu.thmb 文件中。

- 20 mark.thmb 文件存储标记缩略图，即表示标记点的图。对应于全部 Playlist 和 Clip 的全部菜单缩略图存储在单个 mark.thmb 文件中。由于经常添加或删除缩略图，添加和部分删除的操作必须能够容易和快速地执行。为此，Thumbnail() 具有分块结构。图像数据划分为多个部分，每一个均存储在一个 tn_block 中。一个图像数据存储在多个连续 tn_block 中。在 tn_block 串中，
25 可以存在没有使用的 tn_block。单个缩略图的字节长度是可变的。

图 78 示出 menu.thmb 和 mark.thmb 的语法，并且图 79 示出图 78 所示 menu.thmb 和 mark.thmb 语法中 Thumbnail 语法。对图 79 所示 Thumbnail 语法进行说明，version_number 为表示该 Thumbnail() 版本号的四个字符字母。version_number 必须根据 ISO646 编码为“0045”。

- 30 length 为 32 位无符号整数，表示从紧接 length 域之后直到 Thumbnail() 末端的 MakerPrivateData() 字节数。Tu_block_start_address 为 32 位无符号整

数，表示以相对于 Thumbnail()起始端字节的字节数为单位，第一 tn_block 的起始端字节地址。相对字节数从 0 开始计数。number_of_thumbnails 为 16 位无符号整数，给出包含在 Thumbnail()中的缩略图的条目数。

Tu_block_size 为 16 位无符号整数,以 1024 字节为单位给出一个 tn_block 的大小。例如，如果 tn_block_size=1，它表示一个 tn_block 的大小为 1024 字节。Number_of_tn_blocks 为 16 位无符号整数,表示在该 Thumbnail()中 tn_block 的条目数。Thumbnail_index 为 16 位无符号整数，表示从 thumbnail_index 域开始的一个循环的由缩略图信息表示的缩略图的索引号。值 0xFFFF 禁止用作 Thumbnail_index。该 Thumbnail_index 由 UIAppInfoVolume(), UIAppInfoPlayList(), PlayListMark()和 ClipMark()中的 ref_thumbnail_index 进行引用。

Thumbnail_picture_format 为 8 位无符号整数，表示缩略图的图像格式，并且采用图 80 中所示的值。在表中，DCF 和 PNG 只允许在 menu.thumb 中。标记缩略图必须采用“0x00”值(MPEG-2 视频 1-图)。

Picture_data_size 为 32 位无符号整数，表示以字节为单位的缩略图字节长度。Start_tn_block_number 为 16 位无符号整数，表示缩略图数据开始的 tn_block 的 tn_block 号。缩略图数据起始端必须与 tn_block 的起始端一致。tn_block 号从 0 开始，并且与 tn_block 的循环中变量 k 的值相关。

X_picture_length 为 16 位无符号整数，表示一帧缩略图的水平方向象素数。Y_picture_length 为 16 位无符号整数，表示一帧缩略图的竖直方向象素数。tn_block 为存储缩略图的区域。Thumbnail()中的所有 tn_block 具有相同的大小(固定长度)并且由 tn_block_size 定义该大小。

图 81A 和 81B 示意性地示出如何在 tn_block 中存储缩略图。如图 81A 和 81B 所示，如果缩略图从 tn_block 起始端开始，并且大小超过 1 tn_block，它使用下一个 tn_block 进行存储。通过这样做，可变长度数据可以作为固定长度数据进行管理，从而删除编辑操作可以采用更简单的方式来处理。

现在说明 AV 流文件。AV 流文件存储在“M2TS”目录中(图 14)。有两种类型的 AV 流文件，即 Clip AV 流文件和 Bridge-Clip AV 流文件。两种 AV 流必须具有以下所定义的 DVR MPEG-2 传输流文件的结构。

首先，说明 DVR MPEG2 传输流。DVR MPEG2 传输流的结构如图 82 所示。AV 流文件具有 DVR MPEG 2 传输流的结构。DVR MPEG 2 传输流由

整数个对齐单元(Aligned unit)构成。对齐单元的大小为 6144 字节(2048*3 字节)。对齐单元从源包的第一个字节开始。源包长为 192 字节。一个源包由 TP_extra_header 和传输包组成。TP_extra_header 长为 4 字节,传输包长为 188 字节。

- 5 一个对齐单元由 32 个源包组成。DVR MPEG 2 传输流中的最后一个对齐单元也是由 32 个源包组成。因此, DVR MPEG 2 在对齐单元的边界结束。如果记录在盘上的输入传输流的传输包数目不是 32 的倍数,具有空包的源包(PID=0x1FFFF 的传输包)必须用作最后一个对齐单元。文件系统禁止在 DVR MPEG 2 传输流中使用多余信息。

- 10 图 83 示出 DVR MPEG 2 传输流的记录器模型。图 83 所示的记录器是用于规定记录过程的概念模型。DVR MPEG 2 传输流遵从该模型。

现在说明 MPEG 2 传输流的输入时刻。输入 MPEG 2 传输流是一个完整传输流或部分传输流。输入 MPEG 2 传输流必须遵循 ISO/IEC 13818-1 或 ISO/IEC 13818-9。MPEG 2 传输流的第 i 字节在时间 t(i)同时输入到 T-STD

- 15 (ISO/IEC 13818-1 中所提供的传输流系统目标解码器)和源分包器。R_{pk} 是传输包输入速度的瞬时最大值。

27MHz PLL 52 生成一个 27MHz 时钟的频率,该 27MHz 时钟频率被锁定在 MPEG 2 传输流的节目时钟引用(program clock reference, PCR)值。到达时钟计数器 53 计算 27MHz 频率的脉冲。Arrival_time_clock(i)是在时间 t(i)

20 的到达时钟计数器的计数值。

源分包器 54 将 TP_extra_header 附加到全部传输包,以创建一个源包。Arrival_time_stamp 表示传输包第一个字节到达 T-STD 和源分包器的时间。Arrival_time_stamp(k)是 Arrival_time_clock(k)的取样值,如下面等式所示:

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

- 25 其中, k 表示传输包的第一个字节。

如果两个相邻传输包的时间间隔为 230/27000000 秒(大约 40 秒),或更长,两个传输包的 arrival_time_stamp 之差需要设为 230/27000000 秒。记录器在这种情况下是这样提供的。

- 平滑缓冲区 55 平滑输入传输流的位率。平滑缓冲区禁止溢出。当平滑缓冲区不为空时, R_{max} 为从平滑缓冲区输出源包的位率。如果平滑缓冲区为
- 30 空,平滑缓冲区的输出位率为 0。

下一步,说明 DVR MPEG 2 传输流的记录器模型参数。Rmax 值由与 AV 流文件相关的 ClipInfo()中所定义的 TS_recording_rate 指定。该值可以根据如下方程来计算:

$$R_{\max} = \text{TS_recording_rate} * 192/188$$

5 其中 TS_recording_rate 值的大小以字节/秒为单位。

如果输入传输流是 SESF 传输流, Rpk 必须等于与 AV 流文件相关的 ClipInfo()中所定义的 TS_recording_rate。如果输入传输流不是 SESF 传输流,该值可以引用例如 MPEG 2 传输流描述符中所定义的值,如 maximum_bitrate_descriptor 或 partial_stream_descriptor。

10 如果输入传输流是 SESF 传输流,平滑缓冲区的大小为 0。如果输入传输流不是 SESF 传输流,可以引用 MPEG 2 传输流描述符中所定义的值,例如 smoothing_buffer_descriptor、short_smoothing_buffer_descriptor 或 partial_transport_stream_descriptor 中定义的值。

对于记录器和播放器(再现装置),需要提供足够大小的缓冲区。缺省缓冲区大小为 1536 字节。

下一步,说明 DVR MPEG 2 的传输流播放器模型。图 84 示出 DVR MPEG 2 传输流播放器模型。这是一个用于规定再现过程的概念模型。DVR MPEG 2 传输流遵从该模型。

20 27MHz X-tal 61 生成 27MHz 频率。27MHz 频率的误差范围必须为 $\pm 30\text{ppm}(27000000 \pm 810\text{Hz})$ 。到达时钟计数器 62 是用于对 27MHz 频率的脉冲进行计数的二进制计数器。Arrival_time_clock(i)是在时间 t(i)到达时钟计数器的计数值。

在平滑缓冲区 64 中,当平滑缓冲区不满时, Rmax 是向平滑缓冲区输入源包的位率。如果平滑缓冲区满,向平滑缓冲区的输入位率为 0。

25 对 MPEG 2 传输流的输出时刻进行说明,如果当前源包的 arrival_time_stamp 等于 arrival_time_clock(i)的 LSB 端的 30 位,源包的传输包从平滑缓冲区移走。Rpk 是传输包速率的瞬时最大值。平滑缓冲区不允许溢出。

30 DVR MPEG 2 传输流播放器模型的参数和上面所描述的 DVR MPEG 2 传输流记录器模型的参数相同。

图 85 示出源包的语法。Transport_packet()是在 ISO/IEC 13818-1 中提供

的 MPEG 2 传输流。如图 85 所示的源包语法中的 TP_Extra_header 语法如图 86 所示。对图 86 所示的 TP_Extra-header 语法进行说明，
copy_permission_indicator 为表示传输包有效荷载的拷贝限制。拷贝限制可以是自由拷贝，不再拷贝，一次拷贝或禁止拷贝。图 87 示出

5 copy_permission_indicator 指及其所指定模式之间的关系。

copy_permission_indicator 添加到全部传输包中。如果使用 IEEE 1394 数字接口记录输入传输流，copy_permission_indicator 值可以与 EMI(encryption mode indicator, 加密模式指示器)值相关联。如果没有使用 IEEE 1394 数字接口记录输入传输流，copy_permission_indicator 值可以与嵌入在传输包中的
10 CCI 值相关联。如果模拟信号输入是自编码，copy_permission_indicator 值可以与模拟信号的 CGMS-A 值相关联。

Arrival_time_stamp 为一个整数，具有由下面方程中的 arrival_time_stamp 指定的值。

$$\text{Arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230.$$

15 通过对 Clip AV 流的定义，Clip AV 流必须具有如上定义的 DVR MPEG 2 传输流的结构。Arrival_time_clock(i)必须在 Clip AV 流中连续增大。如果 Clip AV 流中存在系统时基(STC 基)的不连续点，Clip AV 流中的
Arrival_time_clock(i)必须连续增大。

Clip AV 流的开始和结束之间 arrival_time_clock(i)相差的最大值必须是
20 26 小时。这个限定保证，如果 MPEG2 传输流中不存在系统时基(STC 基)的不连续点，Clip AV 流中不会出现相同值的 PTS(presentation time stamp, 呈现时间戳)。MPEG 2 系统标准提供 PTS 具有 233/90000 秒(约 26.5 小时)的环绕周期。

通过对 Bridge-Clip AV 流的定义，Bridge-Clip AV 流必须具有如上定义的
25 的 DVR MPEG 2 传输流结构。Bridge-Clip AV 流必须包括一个到达时基的不连续点。到达时基的不连续点前面和后面的传输流必须遵从后面将要说明的编码限制和 DVR-STD。

本实施例支持正被编辑 PlayItem 之间的视频-声频无缝连接。PlayItem 之间的无缝连接保证向播放器/解码器的“连续数据供应”和“无缝解码处
30 理”。“连续数据供应”是保证以防止缓冲区下溢所需的位率向解码器供应数据的能力。为了使数据从盘中读出，以保证数据实时属性，以足够大的连续

块为单位存储数据。

“无缝解码处理”表示播放器在解码器回放输出时，显示记录在盘上的视频音频数据而不产生停顿或间隔的能力。

对由无缝连接 PlayItem 引用的 AV 流进行说明。是否保证前一 PlayItem
5 和当前 PlayItem 的无缝显示可以从当前 PlayItem 中定义的
connection_condition 域中得以证实。有两种用于 PlayItem 无缝连接的方法，
即采用 Bridge-Clip 的方法和不采用 Bridge-Clip 的方法。

图 88 示出采用 Bridge-Clip 情况下，前一 PlayItem 和当前 PlayItem 之间的
10 关系。在图 88 中，由播放器读出的流数据用阴影显示。在图 88 中，TS1
由 Clip1(Clip AV 流)阴影流数据和 RSPN_arrival_time_discontinuity 之前的阴
影流数据组成。

TS1 的 Clip1 的阴影流数据是从对与前一 PlayItem 的 IN_time(图 88 所
示的 IN_time1)对应的呈现单元进行解码所需的流地址直到由
RSPN_exit_from_previous_clip 引用的源包的流数据。包含在 TS1 中的
15 Bridge-Clip 的 RSPN_arrival_time_discontinuity 之前的阴影流数据是从
Bridge-Clip 的第一源包直到紧接在由 RSPN_arrival_time_discontinuity 引用的
源包之前的源包的流数据。

图 88 中，TS2 由 Clip2(Clip AV 流)阴影流数据和 Bridge-Clip 的
RSPN_arrival_time_discontinuity 之后的阴影流数据组成。包含在 TS2 流数据
20 中从 Bridge-Clip 的 RSPN_arrival_time_discontinuity 开始的阴影流数据是从
由 RSPN_arrival_time_discontinuity 引用的源包到 Bridge-Clip 最后一个源包
的流数据。TS2 的 Clip2 阴影流数据是从由 RSPN_enter_to_current_Clip 引用的
源包到对与当前 PlayItem 的 OUT_time(图 88 所示的 OUT_time2)对应的呈
现单元进行解码所需的流地址之间的流数据。

图 89 示出不采用 Bridge-Clip 的情况下，前一 PlayItem 和当前 PlayItem
25 之间的关系。在这种情况下，由播放器读出的流数据用阴影显示。图 89 中，
TS1 由 Clip1(Clip AV 流)的阴影流数据构成。TS1 的 Clip1 的阴影流数据是
从对与前一 PlayItem 的 IN_time(图 89 中 IN_time1 所示)对应的呈现单元进行
解码所需的流地址开始直到 Clip1 最后一个源包的数据。

30 图 89 中，TS2 为 Clip2(Clip AV 流)的阴影流数据。

TS2 的 Clip2 阴影流数据是从 Clip2 的第一源包开始直到对与当前

PlayItem 的 OUT_time(图 89 中 OUT_time2 所示)对应的呈现单元进行解码所需的流地址的流数据。

在图 88 和 89 中, TS1 和 TS2 为源包的连续流。下一步, 对 TS1 和 TS2 的流提供和两者之间的连接条件进行详细描述。首先, 对无缝连接的编码限制进行详细描述。作为对传输流的编码结构的限制, 包含在 TS1 和 TS2 中的节目数必须为 1。包含在 TS1 和 TS2 中的音频流数必须为 2 或更少。包含在 TS1 和 TS2 中的音频流数必须相等。不同于上述流的基本流或私有流包含在 TS1 和/或 TS2 中, 也是可能的。

现在说明对视频位流的限制。图 90 示出由图像显示序列表示的典型无缝连接。为了使视频流在连接点附近无缝地显示, 必须通过在连接点的附近对 Clip 的部分流进行重新编码的过程, 删除在 OUT_time1(Clip1 的 OUT_time)之后和 IN_time2(Clip2 的 IN_time)之前的不需要图像。

图 91 示出使用 BridgeSequence 实现无缝连接的实施例。RSPN_arrival_time_discontinuity 之前的 Bridge-Clip 视频流包括直到与图 90 的 Clip1 的 OUT_time1 对应的图像的编码视频流。该视频流连接到前面 Clip1 的视频流, 并且进行重新编码, 以形成遵循 MPEG 2 标准的基本流。

RSPN_arrival_time_discontinuity 之后的 Bridge-Clip 视频流包括与图 90 的 Clip2 的 IN_time2 对应的图像之后的编码视频流。为将该视频流连接到下一 Clip2 视频流, 可以适当地启动对该视频流的解码。重新解码是为了形成遵循 MPEG 2 标准的单一连续基本流。为创建 Bridge-Clip, 一般需要对几个图像进行重新编码, 同时可以从原始 Clip 拷贝其它图像。

图 92 示出不使用图 90 所示实施例中的 BridgeSequence 实现无缝连接的实施例。Clip1 视频流包括直到与图 90 的 OUT_time1 对应的图像的编码视频流, 并且进行重新编码, 以提供遵循 MPEG 2 标准的基本流。采用类似方式, Clip2 视频流包括与图 90 的 Clip2 的 IN_time2 相关的图像之后的编码位流。这些编码位流已经经过重新编码, 以提供遵循 MPEG 2 标准的单个连续基本流。

对视频流的编码限制进行说明, TS1 和 TS2 的视频流的帧率必须相等。TS1 的视频流必须在 sequence_end_code 终止。TS2 的视频流必须在序列头、GOP 头并且使用 I-图像开始。TS2 的视频流必须在结束 GOP 开始。

在位流中定义的视频呈现单元(帧或域)必须用其间的连接点来连续。在

连接点处不允许存在域或帧的间隔。在采用 3-2 下拉编码的情况下，可能有必要重写“top_field_first”和“repeat_first_field”标志。作为替换，可以进行局部重新编码，以防止产生域间隔。

对音频位流的限制进行说明，TS1 和 TS2 的音频采样频率必须相等。TS1 和 TS2 的音频编码方法(例如，MPEG1 层 2、AC-3、SESF LPCM 和 AAC)必须相同。

对 MPEG-2 传输流的编码限制进行说明，TS1 的音频流的最后一个音频帧必须包含显示时间等于 TS1 最后一个显示图像的显示结束时间的音频样本。TS2 的音频流的第一音频帧必须包含显示时间等于 TS2 第一个显示图像的显示开始时间的音频样本。

在连接点，音频呈现单元序列中不允许存在间隔。如图 93 所示，可以存在小于两个音频帧域的音频呈现单元的长度所限定的重叠。传输 TS2 基本流的第一包必须为视频包。在连接点的传输流必须遵从后面将要说明的 DVR-STD。

对 Clip 和 Bridge-Clip 的限制进行说明，在 TS1 或 TS2 中不允许存在关于到达时基的不连续。

下面限制仅适用于采用 Bridge-Clip 的情况。Bridge-Clip AV 流只在 TS1 的最后源包和 TS2 的第一源包的连接点，有单个关于到达时基的不连续点。在 ClipInfo()中定义的 SPN_arrival_time_discontinuity 表示不连续点的地址，它必须表示引用 TS2 第一源包的地址。

在 BridgeSequenceInfo()中定义的 RSPN_exit_from_previous_Clip 所引用的源包可以是 Clip1 中的任何源包。该源包没有必要是对齐单元的边界。在 BridgeSequenceInfo()中定义的 RSPN_enter_to_current_Clip 所引用的源包可以是 Clip2 中的任何源包。该源包没有必要是对齐单元的边界。

对 PlayItem 的限制进行说明，前一 PlayItem 的 OUT_time(图 89 所示的 OUT_time1)必须表示 TS1 的最后视频呈现单元的显示结束时间。当前 PlayItem 的 IN_time(图 89 和 89 所示的 IN_time2)必须表示 TS2 的第一呈现单元的显示开始时间。

参照图 94，对采用 Bridge-Clip 情况下的数据分配的限制进行说明，为保证文件系统的连续数据提供，必须进行无缝连接。这必须通过安排连接到 Clip1(Clip AV 流文件)和 Clip2(Clip AV 流文件)的 Bridge-Clip AV 流来实现，

以满足数据分配规定。

必须选择 RSPN_exit_from_previous_Clip，从而将 RSPN_exit_from_previous_Clip 之前的 Clip1(Clip AV 流文件)流部分安排在不小于半段(half fragment)的连续区域中。必须选择 Bridge-Clip AV 流的数据长度，从而将该数据安排在不小于半段的连续区域中。必须选择 RSPN_enter_to_current_Clip，从而将 RSPN_enter_to_current_Clip 之后的 Clip2(Clip AV 流文件)流部分安排在不小于半段的连续区域中。

参照图 95，对不采用 Bridge-Clip 的无缝连接的情况下数据分配的限制进行说明，为保证文件系统的连续数据提供，必须进行无缝连接。这必须通过安排 Clip1(Clip AV 流文件)的最后部分和 Clip2(Clip AV 流文件)的第一部分来实现，以满足数据分配规定。

Clip1(Clip AV 流文件)的最后流部分必须安排在不小于半段的连续区域中。Clip2(Clip AV 流文件)的第一流部分必须排列在不小于一个半段的连续区域中。

下一步，对 DVR-STD 进行说明。该 DVR-STD 是一个概念模型，用于在 DVR MPEG 2 传输流的生成和验证中对解码处理进行建模。DVR-STD 还是一个概念模型，用于在如上所述的相互无缝连接的两个 PlayItem 所引用的 AV 流的生成和验证中对解码处理进行建模。

图 96 示出一个 DVR-STD 模型。图 96 所示的该模型包括，作为组成单元的 DVR MPEG 2 传输流播放器模型。 n , Tbn , Mbn , Ebn , $Tbsys$, $Bsys$, Rxn , $Rbxn$, $Rxsys$, Dn , $Dsys$, On 和 $P9(k)$ 的注释与 ISO/IEC 13818-1 的 T-STD 中所定义的相同。其中 n 是基本流的索引号，并且 TBn 是基本流 n 的传输缓冲区。

MBn 是基本流 n 的多路复用缓冲区，并且只为视频流存在。 EBn 是基本流 n 的基本流缓冲区，并且只为视频流提供。 $TBsys$ 为用于正被解码节目的系统信息的系统目标编码器中的主缓冲区。 Rxn 是从 TBn 移走数据的传输率。 $Rbxn$ 是从 MBn 移走 PES 包有效荷载的传输率，并且只为视频流提供。

$Rxsys$ 是从 $Tbsys$ 移走数据的传输率。 Dn 是基本流 n 的解码器。 $Dsys$ 是与正被解码的节目系统信息相关的解码器。 On 是视频流 n 的重新排序缓冲区。 $Pn(k)$ 是基本流的第 k 个呈现单元。

对 DVR-STD 的解码过程进行说明。在单个 DVR MPEG 2 传输流正被

再现的时候，由源包的 `arrival_time_stamp` 确定将传输包输入到 TB1、TBn 或 TBsys 的时刻。对 TB1，MB1，EB1，TBn Bn，TBsys 和 Bsys 的缓冲操作的规定与 ISO/IEC 13818-1 所提供的 T-STD 的规定一致，而且对解码和显示操作的规定也与 ISO/IEC 13818-1 所提供的 T-STD 一致。

5 现在说明在无缝连接的 PlayLists 正被再现时候的解码过程。在此，说明对由无缝连接的 PlayItems 引用的两个 AV 流的再现。下面说明中，说明如图 88 中例子所示的 TS1 和 TS2 的再现。TS1 和 TS2 分别是前一流和当前流。

10 图 97 示出，当从给定 AV 流(TS1)传输到与其无缝连接的下一 AV 流(TS2)时，对传输包进行输入、解码和显示的时间图。在从预定 AV 流(TS1)传输到与其无缝连接的下一 AV 流(TS2)的期间，TS2 的到达时基的时间轴和 TS1 的到达时基的时间轴(图 97 中 ATC1 所示)不同。

15 而且，TS2 的系统时基的时间轴(图 97 中 ATC1 所示)和 TS1 的系统时基的时间轴(图 97 中 STC1 所示)不同。视频显示必须为连续无缝。然而，在呈现单元的显示时间上可以存在重叠。

对向 DVR-STD 的输入时序进行说明。在直到时间 T1 为止的时间内，即直到最后一个视频包输入到 DVR-STD 的 TB1 为止，由 TS1 的到达时基的 `arrival_time_stamp` 确定向 DVR-STD 的 TB1、TBn 或 TBsys 的输入时序。

20 TS1 的剩余包必须以 `TS_recording_rate(TS1)` 位率，输入到 DVR-STD 的 TBn 或 TBsys 的缓冲区。`TS_recording_rate(TS1)` 是对应于 Clip1 的 `ClipInfo()` 中所定义的 `TS_recording_rate` 值。将 TS1 的最后字节输入到缓冲区的时间是时间 T2。因此，在时间 T1 到时间 T2 的时间段内，不考虑源包的 `arrival_time_stamp`。

25 如果 N1 是紧接在 TS1 的最后视频包之后的 TS1 传输包的字节数，从时间 T1 直到时间 T2 的时间 DT1 是以 `TS_recording_rate(TS1)` 的位率完全输入 N1 个字节所需的时间，并且根据如下方程进行计算：

$$DT1 = T2 - T1 = N1 / TS_recording_rate$$

30 在从时间 T1 直到时间 T2(TS1)的时间内，RXn 和 RXsys 两者的值均改变为 `TS_recording_rate(TS1)` 的值。除该规则之外，缓冲操作与 T-STD 的相同。

在时间 T2，到达时钟计数器复位为 TS2 的第一源包的 `arrival_time_stamp`

的值。由 TB2 的源包的 arrival_time_stamp 确定向 DVR-STD 的 TB1, TBn 或者 TBsys 缓冲区的输入时序。RXn 和 RXsys 两者均改变为 T-STD 中所定义的值。

对附加的音频缓冲和系统数据缓冲进行说明, 除 T-STD 中所定义的缓冲区数量之外, 音频解码器和系统解码器还需要具有附加的缓冲区数量(相当于 1 秒的数据量), 以允许输入从时间 T1 到时间 T2 的域数据。

对视频呈现时间进行说明, 视频呈现单元的显示必须通过连接点保持连续, 即没有间隔。注意, STC1 是 TS1 的系统时基的时间轴(图 97 中 STC1 所示), 而 STC2 是 TS2 的系统时基的时间轴(图 97 中 STC2 所示; 准确地说, STC2 在 TS2 的第一 PCR 已被输入到 T-STD 的时间开始)。

STC1 和 ST2 之间的偏移按照如下方式进行确定: PTS1end 是对应于 TS2 最后一个视频呈现单元的 STC1 上的 PTS。PTS2start 是对应于 TS2 第一个视频呈现单元的 STC2 上的 PTS, 并且 Tpp 是 TS1 最后一个视频呈现单元的显示时间段, 两个系统时基之间的偏移 STC_delta 根据如下方程进行计算:

15 $STC_delta = PTS1end + Tpp - PTS2start$ 。

对音频呈现时间进行说明, 音频呈现单元的显示时刻可以存在重叠, 重叠小于 0 到 2 个音频帧(参见图 97 所示的“音频重叠”)。在播放器上设置, 对在连接点处之后要选择哪个音频样本并且将音频呈现单元的显示重新同步到经过修正的时基的指示。

20 对 DVR-STD 的系统时钟进行说明, TS1 的最后音频呈现单元在时间 T5 时显示。系统时钟可以在时间 T2 和时间 T5 之间重叠。在该时间域内, DVR-STD 将系统时钟在旧时基(STC1)值和新时基(STC2)值之间进行切换。STC2 的值可以根据如下方程进行计算:

$$STC2 = STC1 - STC_delta$$

25 对缓冲连续性进行说明。STC11video_end 是当第一视频包的第一字节到达 DVR-STD 的 TB1 时系统时基 STC2 上的 STC 值。STC22video_start 是当第一视频包的第一字节到达 DVR-STD 的 TB1 时系统时基 STC2 上的 STC 值。STC21video_end 是按照系统时基 STC2 的 STC2 上的值计算的 STC11video_end 值。STC2video_end 根据如下方程进行计算:

30 $STC21video_end = STC11video_end - STC_delta$

为了遵从 DVR-STD, 必须满足下面两个条件: 第一, 在 TB1 的 TS2

的第一视频包的到达时刻必须满足如下不等式：

$$\text{STC22video_start} > \text{STC21video_end} + \Delta T1。$$

如果为满足上述不等式，有必要重新对 Clip1 和/或 Clip2 的部分流进行重新编码和/或多路复用，执行该重新编码或多路复用是适当的。

- 5 第二，在从 STC1 和 STC2 映射在相同时间轴上的系统时基的时间轴上，从 TS1 输入视频包，随后从 TS2 输入视频包禁止上溢或下溢视频缓冲区。

如果作为基础使用上面语法，数据结构和规则，可以适当地管理记录在记录介质中的数据内容或再现信息，使得用户能够非常容易地在再现时确认记录在记录介质中的数据内容，或再现所需的数据。

- 10 在上述实施例中，采用 MPEG 2 传输流作为多路复用流的例子。然而，这仅是示例性的，因此 MPEG 2 节目流 DSS 或 USA 的 DirecTV 服务(商标)中所使用的传输流也可以用作多路复用流。

- 图 98 示出 PlayList 文件的变型。图 98 和图 99 的语法之间的显著区别是存储 UIAppInfoPlayList()的位置。在图 98 的实施例中，UIAppInfoPlayList()在 PlayList()的外部，这样可以相当容易地获得 UIAppInfoPlayList()的将来信息扩展。

Version_number 是表示缩略图标题信息文件版本号的四个数字。

PlayList_start_address 以相对于 PlayList 文件起始端的字节数为单位，表示 PlayList()的起始地址。相对字节数从 0 开始计数。

- 20 PlayListMark_start_address 以相对于 PlayList 文件起始端的字节数为单位，表示 PlayListMark()的起始地址。相对字节数从 0 开始计数。

MarkersPrivateData_start_address 以相对于 PlayList 文件起始端的字节数为单位，表示 MarkersPrivateData 的起始地址。相对字节数从 0 开始计数。

图 99 示出图 98 的 PlayList 文件中的 UIAppInfoPlayList 语法。

- 25 PlayList_service_type 表示 PlayList 文件的类型，其示例如图 26 所示。

PlayList_service_type 可以与由数字 TV 广播表示的服务类型具有相同的含义。例如，在日本的 BS 广播中，有三种服务类型，即 TV 服务，音频服务和数据广播服务。在 PlayList_service_type 中设置表示包含在 Clip AV 流中由 PlayList 使用的节目服务类型的值。

- 30 PlayList_character_set 表示在 channel_name, PlayList_name 和 PlayList_detail 域中编码的字符字母编码方法，同时还表示在 PlayListMark

的 mark_name 域中编码的字符字母编码方法。

Channel_number 表示当对 Playlist 进行记录时，用户所选择的广播频道号或服务号。如果多个 Playlist 组合成一个 Playlist，channel_number 表示 Playlist 的代表值。如果该域设为 0xFFFF，该域无意义。

- 5 Channel_name_length 表示 channel_name 域所表示的频道名称的字节长度。该域的值不大于 20。

Channel_name 表示当对 Playlist 进行记录时，用户所选择的服务或广播频道名称。从该域的左边开始的 channel_name_length 指定数的字节数为有效字符字母，并且表示上述名称。在这些有效字符字母之后的剩余字节可以设为任意值。如果多个 Playlist 组合成一个 Playlist，该域表示 Playlist 的代表性名称。

Playlist_name_length 表示 Playlist_name 域指定的 Playlist 名称的字节长度。

- 15 Playlist_name_length 指定数目的字节数为有效字符字母，并且表示上述名称。该域中这些有效字符字母之后的剩余字节可以设为任何可选值。

Playlist_detail_length 表示 Playlist_detail 域指定的 Playlist 详细信息的字节长度。该域的值不超过 1200。

- 20 Playlist_detail 表示用于说明 Playlist 详细信息的文本。从该域的左边开始的 Playlist_detail_length 指定数的字节数为有效字符字母。该域中这些有效字符字母之后的剩余字节可以设为任何可选值。

在其它方面该语法域的含义与图 27 所示的同名域相同。

图 100 示出图 98 的 Playlist 文件中 Playlist() 的语法。除本语法缺少 UIAppInfoPlaylist() 之外，该语法基本上与图 25 的实施例相同。

- 25 图 101 示出 SubPlayItem 语法的变型。本语法明显与图 40 的实施例不同，因为在此增加了 STC_sequence_id。

- 30 STC_sequence_id 表示由用来标识对应于 Clip_Information_file_name 的 AV 流文件中的再现域的 SubPath_IN_time 和 SubPath_OUT_time 引用的 STC 的 STC_sequence_id。SubPath_IN_time 和 SubPath_OUT_time 表示由 STC_sequence_id 指定的相同 STC 连续域时间。

通过将 STC_sequence_id 添加到 SubPlayItem，允许由 SubPlayItem 引用

的 AV 流文件具有 STC 不连续点。

在其它方面，该语法域具有与图 40 所示的同名域相同的含义。

图 102 示出一个说明形成 Real PlayList 方法的流程图。参照图 1 所示的记录和/或再现装置的方框图。

5 在步骤 S11，控制器 43 记录 Clip AV 流。

在步骤 S12，控制器 23 检查是否可以准备 AV 流的 EP_map。如果步骤 S12 的检查结果是 YES，控制器 23 继续到步骤 S13。否则，控制器 23 继续到步骤 S14，以形成 TU_Map。

在步骤 S15，控制器 23 然后设置 PlayList 的 CPI_type。

10 在步骤 S16，控制器 23 形成由覆盖整个 Clip 可能再现范围的 PlayItem 组成的 PlayList()。如果 CPI_type 为 EP_map 类型，根据 PTS 设置时间信息。如果 Clip 中存在 STC 不连续点并且 PlayList() 由两个或更多 PlayItem 组成，还确定 PlayItem 之间的 connection_condition。如果 CPI_type 为 TU_map 类型，根据到达时间设置时间信息。

15 在步骤 S17，控制器 23 形成 UIAppInfoPlayList()。

在步骤 S18，控制器 23 形成 PlayListMark。

在步骤 S19，控制器 23 形成 MakerPrivateData。

在步骤 S20，控制器 23 形成 RealPlayList 文件。

这样，无论何时新记录一个 Clip AV 流，都形成一个 Real PlayList 文件。

20 图 103 是示出形成 Virtual PlayList 方法的流程图。

在步骤 S31，通过用户界面指定一个记录在盘上的 Real PlayList。通过用户界面从 Real PlayList 的再现范围指定由 IN 和 OUT 点指定的再现范围。如果 CPI_type 为 EP_map 类型，根据 PTS 设置再现域。如果 CPI_type 为 TU_map 类型，根据到达时间设置再现域。

25 在步骤 S32，控制器检查是否完成用户指定再现范围的全部操作。如果用户选择要在所指定的再现域之后再再现的域，控制器返回到步骤 S31。如果用户指定再现范围的全部操作结束，控制器 23 继续到步骤 S33。

在步骤 S33，用户通过界面或通过控制器 23 确定两个连续被再现的再现域之间的连接条件(Connect_condition)。

30 在步骤 S34，如果 CPI_type 为 EP_map 类型，用户指定 sub-path 信息(后期记录音频信息)。如果没有由用户 0 形成 sub-path，忽略该步骤。

在步骤 S35，控制器 23 根据用户所指定的再现范围信息和连接条件形成 Playlist()。

在步骤 S36，控制器 23 形成 UIAppInfoPlaylist()。

在步骤 S37，控制器 23 形成 PlaylistMark。

5 在步骤 S38，控制器 23 形成 MakerPrivateData。

在步骤 S39，控制器 23 形成 VirtualPlaylist 文件。

采用这种方式，为从记录在盘上的 Real Playlist 的再现范围选择的并且用户想要观看的每一组再现域形成一个 virtual Playlist 文件。

图 104 示出说明 Playlist 再现方法的流程图。

10 在步骤 S51，控制器 23 获取 Info.dvr、Clip Information 文件、Playlist 文件和缩略图文件的信息，并且形成显示记录在盘上的 Playlist 列表的 GUI 图像，以通过用户界面显示在 GUI 上形成的 GUI 图像。

在步骤 S52，控制器 23 根据各个 Playlist 中的 UIAppInfoPlaylist 显示说明 GUI 图像上 Playlist 的信息。

15 在步骤 S53，用户通过用户界面命令从 GUI 图像再现一个 Playlist。

如果 CPI_type 为 EP_map 类型，控制器 23 在步骤 S54 从 STC_sequence_id 和 IN_time PTS 获取从时间上入口点在 IN_time 之前且离其最近的源包号。如果 CPI_type 为 TU_map 类型，控制器 23 从当前 PlayItem 的 IN_time 获取开始从时间上在 IN_time 之前且离其最近的时间单元的源包号。

20 在步骤 S55，控制器 23 从在上述步骤获取的源包号读出 AV 流数据，以将所读出的数据发送到 AV 解码器 27。

在步骤 S56，如果存在时间上位于当前 PlayItem 之前的 PlayItem，控制器 23 根据 connection_condition 执行前一 PlayItem 和当前 PlayItem 之间的显示连接处理。

25 在步骤 S57，如果 CPI_type 为 EP_map 类型，AV 解码器 27 命令从 IN_time PTS 的图像开始显示。如果 CPI_type 为 TU_map 类型，AV 解码器 27 命令从从 IN_time 之后的流图像开始显示。

在步骤 S58，控制器 23 命令 AV 解码器 27 继续对 AV 流进行解码。

30 如果 CPI_type 为 EP_map 类型，在步骤 S59，控制器 23 检查当前所显示的图像是否为 OUT_time PTS 的图像。另外，如果 CPI_type 为 TU_map 类型，控制器 23 检查当前所解码的图像是否超过 OUT_time。

如果步骤 S59 的检查结果是 NO, 控制器 23 继续到步骤 S60。在步骤 S60, 控制器 23 显示当前图像, 然后返回到步骤 S58。如果步骤 S59 的检查结果是 YES, 控制器 23 继续到步骤 S61。

在步骤 S61, 控制器 23 检查当前 PlayItem 是否为 PlayList 的最后一个 PlayItem。如果检查结果是 NO, 控制器 23 返回到步骤 S54, 否则 PlayList 再现结束。

图 105 示出说明 PlayList 的次路径再现方法的流程图。只有 PlayList 的 CPI_type 为 EP_map 类型, 才使用图 105 的次路径再现方法。该流程图的处理与图 104 的 PlayList 再现中步骤 S54 之后的处理进行同时执行。同时, 预先假定 AV 解码器 27 能够同时对两个音频流进行解码。

在步骤 S71, 控制器 23 获取 SubPlayItem 的信息。

在步骤 S72, 控制器 23 获取入口点从时间上在 SubPath_IN_time 之前且离其最近的源包号。

在步骤 S73, 控制器 23 从具有上述入口点的源包号读出次路径的 AV 流数据, 以将所读出的数据发送到 AV 解码器 27。

在步骤 S74, 当主路径的再现到达由 sync_PlayItem_id 和 sync_start_PTS_of_PlayItem 指定的图像, 控制器 23 命令 AV 解码器 27 开始显示次路径音频。

在步骤 S75, AV 解码器 27 继续对次路径的 AV 流进行解码。

在步骤 S76, 控制器 23 检查当前所显示的次路径的 PTS 是否为 SubPath_OUT_time。如果检查结果是 NO, 控制器 23 继续到步骤 S77, 在该步骤中, 控制器 23 继续显示次路径。然后控制器 23 返回到步骤 S75。

在步骤 S76, 如果当前所显示的次路径的 PTS 是 SubPath_OUT_time, 次路径的显示结束。

再现由用户命令要再现的一个 PlayList 文件的主路径和次路径, 如图 104 和 105 所示。

图 106 示出说明形成 PlayListMark 方法的流程图。参照图 1 的记录和/或再现装置的方框图。

在步骤 S91, 控制器 23 获取 Info.dvr, Clip Information 文件, PlayList 文件和缩略图的信息, 并且形成显示记录在盘上的 PlayList 列表的 GUI 图像, 以通过用户界面显示在 GUI 上形成的 GUI 图像。

在步骤 S92, 用户通过用户界面命令控制器 23 再现一个 PlayList。

在步骤 S93, 控制器 23 开始回放所命令的指定 PlayList (参见图 104)。

在步骤 S94, 用户通过用户界面命令控制器 23 在喜爱的场景设置标记。

5 在步骤 S95, 如果 CPI_type 是 EP_map, 控制器 23 获取标记 PTS 及其所属 PlayItem 的 PlayItem_id。另外, 如果 CPI_type 为 TU_map, 控制器 23 获取标记点的到达时间。

在步骤 S95, 控制器 23 在 PlayListMark() 中存储标记信息。

在步骤 S97, 控制器 23 在记录介质 100 上记录 PlayList 文件。

10 图 107 示出说明采用 PlayListMark 的定位再现方法的流程图。参照图 1 的记录和/或再现装置的方框图。

在步骤 S111, 控制器 23 获取 Info.dvr, Clip Information 文件, PlayList 文件和缩略图的信息, 并且形成显示记录在盘(记录介质 100)上的 PlayList 列表的 GUI 图像, 以通过用户界面显示在 GUI 上形成的 GUI 图像。

在步骤 S112, 用户通过用户界面命令控制器 23 再现一个 PlayList。

15 在步骤 S113, 控制器 23 通过用户界面在 GUI 上显示从 PlayListMark 所引用的图像生成的缩略图列表。

在步骤 S114, 用户通过用户界面指定再现开始点的标记点。

20 如果 CPI_type 为 EP_map 类型, 控制器 23 获取标记 PTS 及其所属 PlayItem_id。如果 CPI_type 为 TU_map, 控制器 23 获取标记 ATS(Arrival Time Stamp, 到达时间戳)。

如果 CPI_type 为 EP_map 类型, 在步骤 S116, 控制器 23 获取由 PlayItem_id 所指定 PlayItem 引用的 AV 流的 STC-sequence-id。

25 在步骤 S117, 如果 CPI_type 为 EP_map 类型, 控制器 23 根据标记 PTS 和 STC-sequence-id 将 AV 流输入到解码器。具体地说, 通过使用标记 PTS 和 STC-sequence_id, 控制器 23 执行与步骤 S55 相似的处理。如果 CPI_type 为 TU_map 类型, 控制器 23 根据标记 ATS 将 AV 流输入到解码器。具体地说, 通过使用 ATS, 控制器执行与图 104 的步骤 S54 和步骤 S55 相似的处理。

30 在步骤 S118, 如果 CPI_type 为 EP_map 类型, 控制器 23 从标记点 PTS 的图像开始显示。如果 CPI_type 为 TU_map 类型, 控制器 23 从标记点 ATS 之后的图像开始显示。

采用这种方式, 用户从 PlayList 选择例如喜爱场景的开始点。所选开始

点由记录器(记录和/或再现装置 1 的控制器 23)在 PlayListMark 中进行管理。而且, 用户从存储在 playListMark 中的标记点列表选择再现开始点, 从而播放器在开始点开始再现, 如图 107 所示。

5 如果作为基础使用上面语法, 数据结构和规则, 可以适当地管理记录在记录介质中的数据内容, 或再现信息, 使得用户能够非常容易地在再现时确认记录在记录介质中的数据内容, 或再现所需的数据。

如果可以分析 I-图像的位置, 在使用 TU_map 的情况下, 就可以通过使用公共应用程序(软件)记录、再现和管理不同格式的 AV 流。

10 如果在记录介质上记录 AV 流时, 对其内容(I-图像位置)进行分析(认知记录), 则使用 EP_map, 反之, 如果在记录介质上直接记录 AV 流而不对其内容(I-图像位置)进行分析(非认知记录), 则使用 TU_map。这样, 可以使用公共应用程序记录、再现和管理 AV 数据。

15 因此, 如果加密 AV 数据通过分析进行解密, 记录到记录介质上, 则使用 EP_map, 反之, 如果加密 AV 数据没有解密(没有分析)直接记录在记录介质上, 则使用 TU_map。这样, 可以使用公共应用程序记录、再现和管理 AV 数据。

20 而且, EP_map 类型和 TU_map 类型可以作为 CPI_type 在 PlayList() 中进行描述, 如果能够分析 I-图像位置, 则可以使用 EP_map, 反之, 如果不能分析 I-图像位置, 则可以使用 TU_map。这样, 采用 I-图像位置的分析记录的 AV 流数据和没有采用 I-图像位置的分析记录的 AV 流数据可以简单地设置相应标志, 通过公共程序以统一的方式进行管理。

而且, 分别记录 PlayList 文件和 Clip 信息文件, 从而, 如果给定 PlayList 或 Clip 的内容由于例如编辑发生改变时, 没有必要改变与被改变文件无关的文件。结果, 可以容易地改变文件内容, 减少这种改变或记录所需的时间。

25 另外, 只要首先读出 Info.dvr, 将盘记录内容显示在用户界面上, 以只从盘中读出用户所命令要再现的 PlayList 文件, 和相关 Clip Information 文件, 就可以缩短用户等候时间。

30 如果将全部 PlayList 文件或 Clip Information 文件收集在一个用于记录的文件中, 文件大小将变得极为庞大。这样, 改变记录文件内容的时间比分开记录各个文件的情况下明显更长。本发明的目标是克服这个缺陷。

不仅可以用硬件, 还可用软件执行上述操作序列。如果用软件执行上述

操作序列，将它从记录介质安装到具有形成软件程序的专用硬件的计算机或图 38 的能够根据安装在其中的各种软件执行各种功能的通用个人计算机。

除计算机之外，记录介质不仅包括用于将程序提供给用户而分发的包介质，如其中带有该程序的磁盘 221，包含软盘，光盘 222(包括 CD-ROM(压缩盘-只读存储器)或 DVD 数字化多用途盘))，光磁盘 223(包括微型盘)，或 5 半导体存储器 224，还包括内置于计算机中向用户提供的硬盘(包含带有程序的 ROM 202 和存储器 208)，如图 108 所示。

在本说明书中，由介质提供的程序步骤不仅包括按照所示顺序的时间顺序处理，还包括不按照时间顺序而用并行或分离方式执行的处理。

10 另外，在说明书中，系统是指由多个组件设备组成的整个装置。

工业应用性

根据本发明，在本发明的信息处理方法和装置、用于记录介质的程序、程序和记录介质中，根据记录方法记录第一表和第二表中的一个，第一表描述 15 呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系。

在本发明的信息处理方法和装置、用于记录介质的程序和程序中，从记录介质再现根据记录方法记录在记录介质中的第一表和第二表中的一个，第一表描述 20 呈现时间戳和对应访问单元的 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的 AV 流数据中地址之间对应关系，从而控制输出。

在本发明的信息处理方法和装置、用于记录介质的程序、程序和第二记录介质中，记录再现指定信息(包括指定主再现路径的第一信息和指定与主 25 再现路径同步再现的次再现路径的第二信息)。

在本发明的信息处理方法和装置、用于记录介质的程序和程序中，从记录介质再现再现指定信息(包括指定主再现路径的第一信息和指定与主再现路径同步再现的次再现路径的第二信息)，从而控制输出。

因此，在任何情况下，都可以共同管理能够执行高速再现的 AV 流和不能 30 执行高速再现的 AV 流，同时后期记录也成为可能。

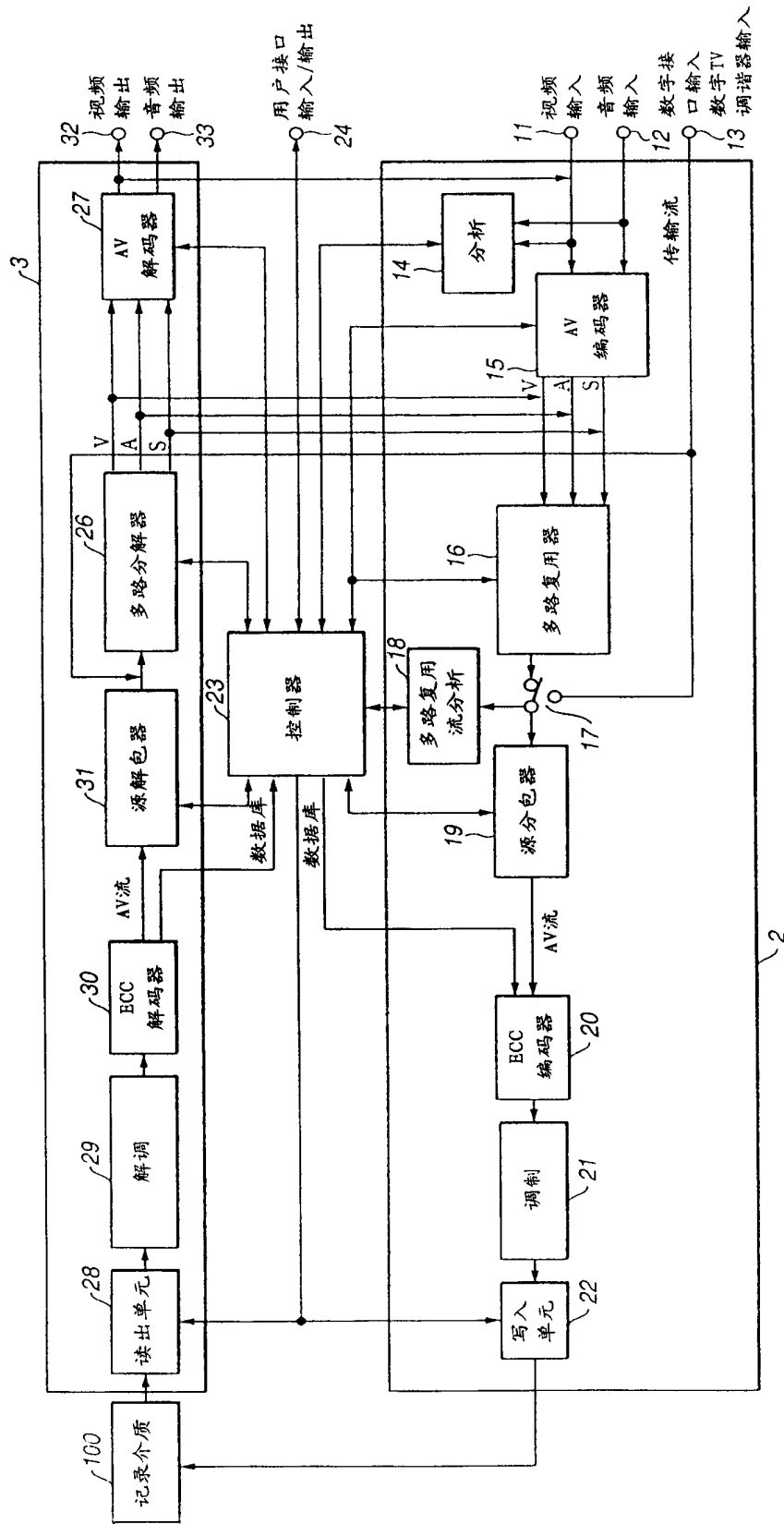


图 1

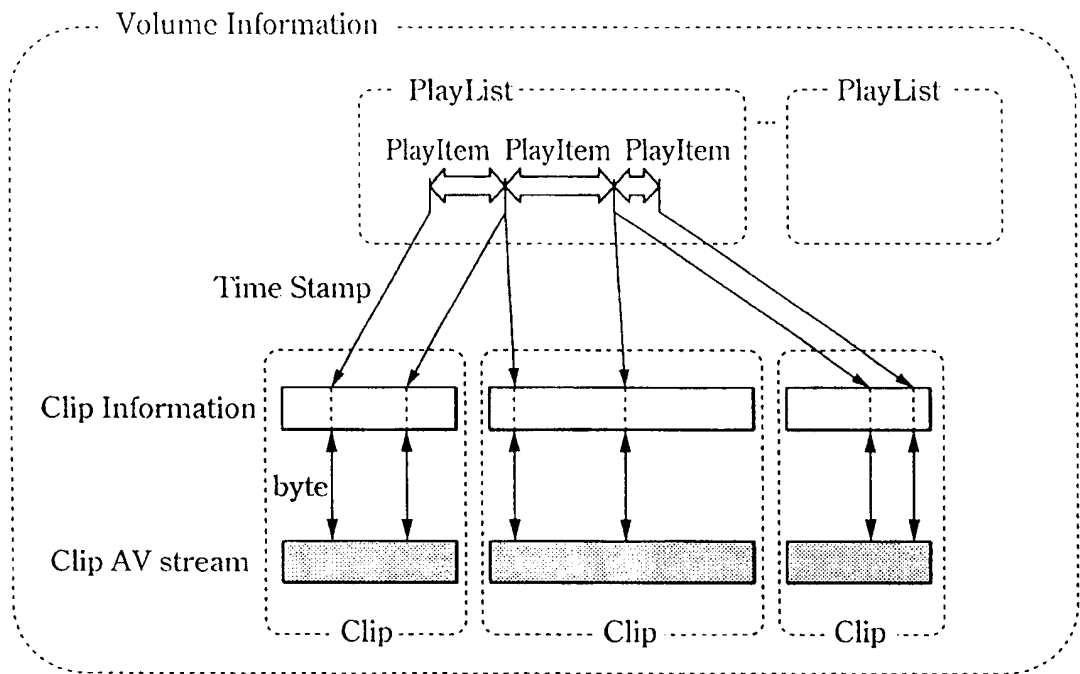


图 2

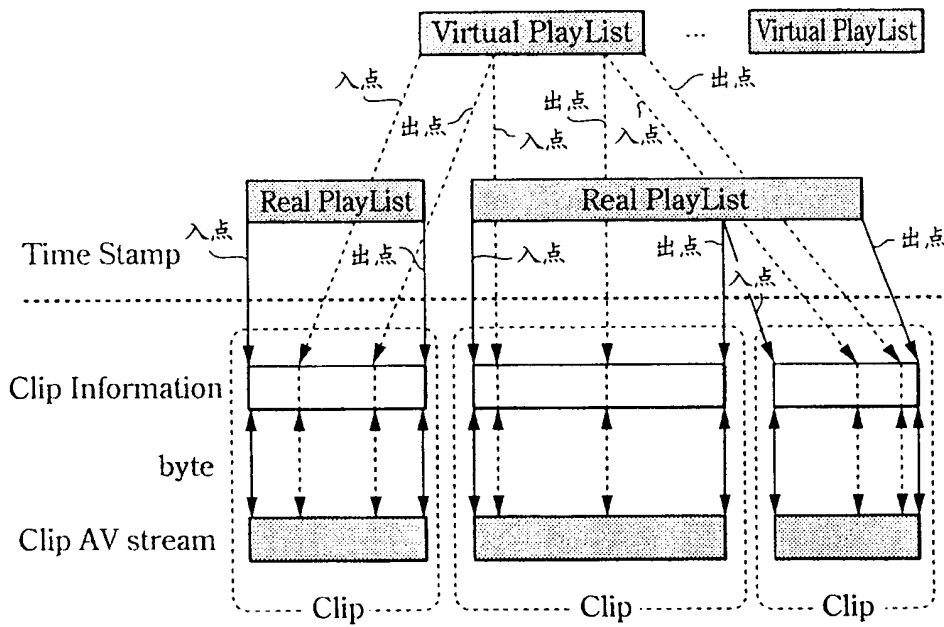


图 3

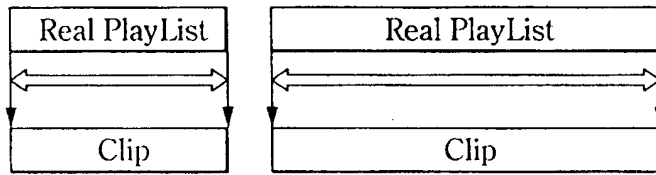


图 4A

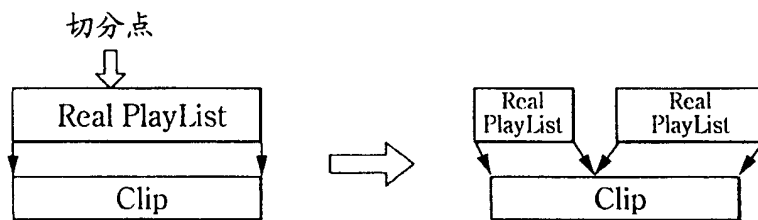


图 4B

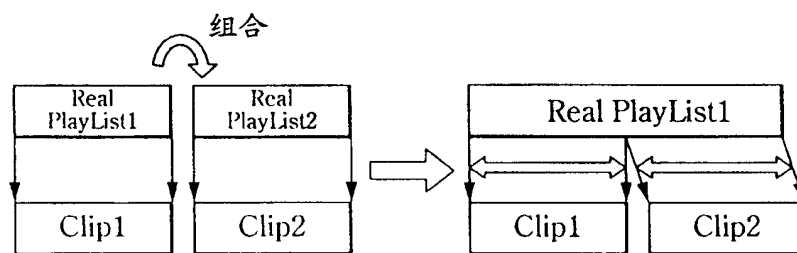


图 4C

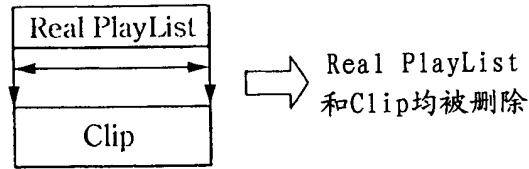


图 5A

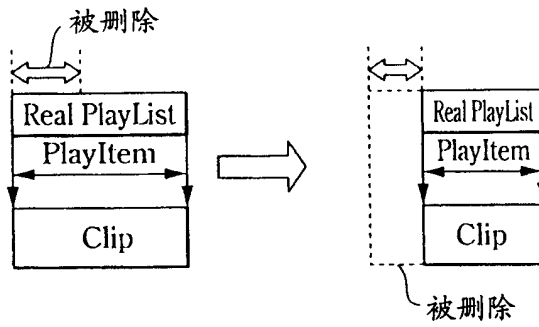


图 5B

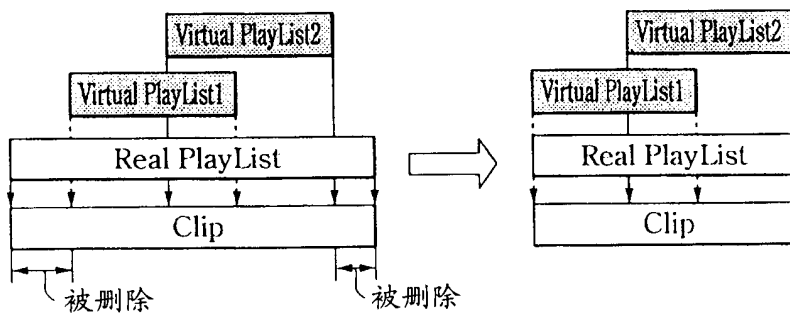
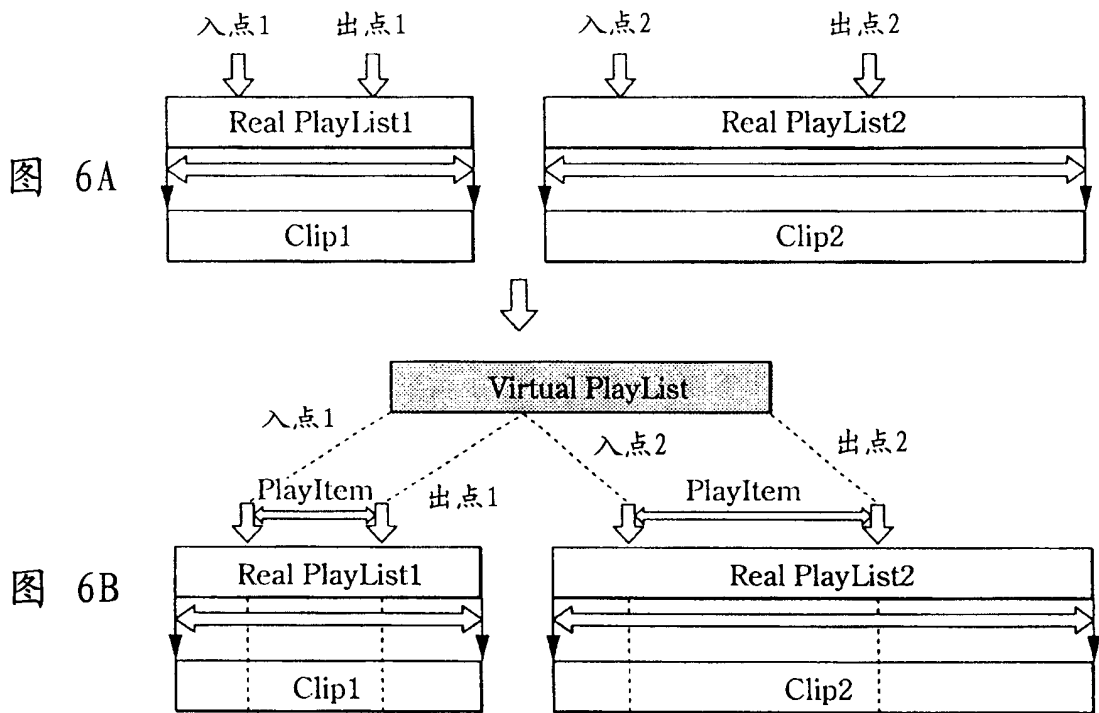


图 5C



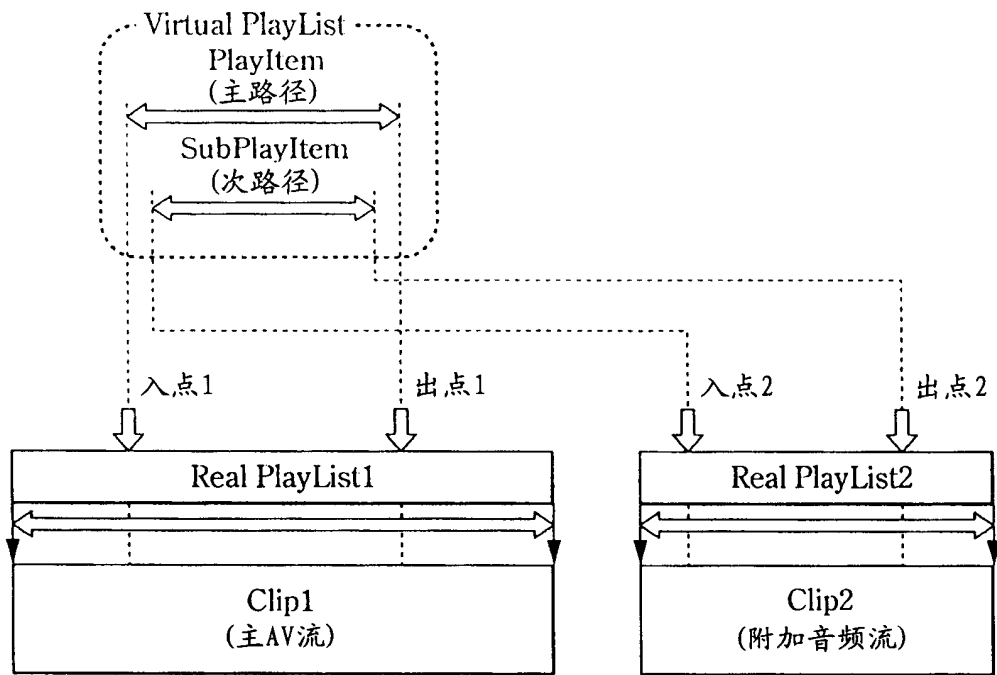


图 7

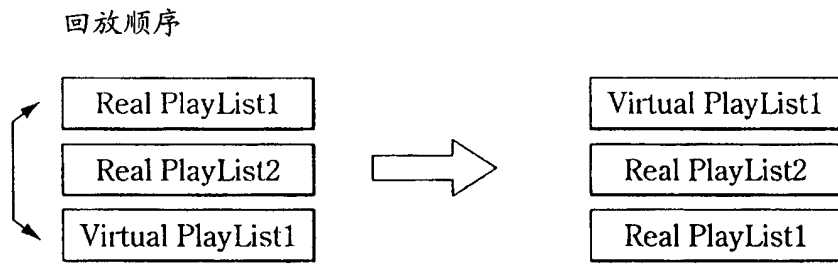


图 8

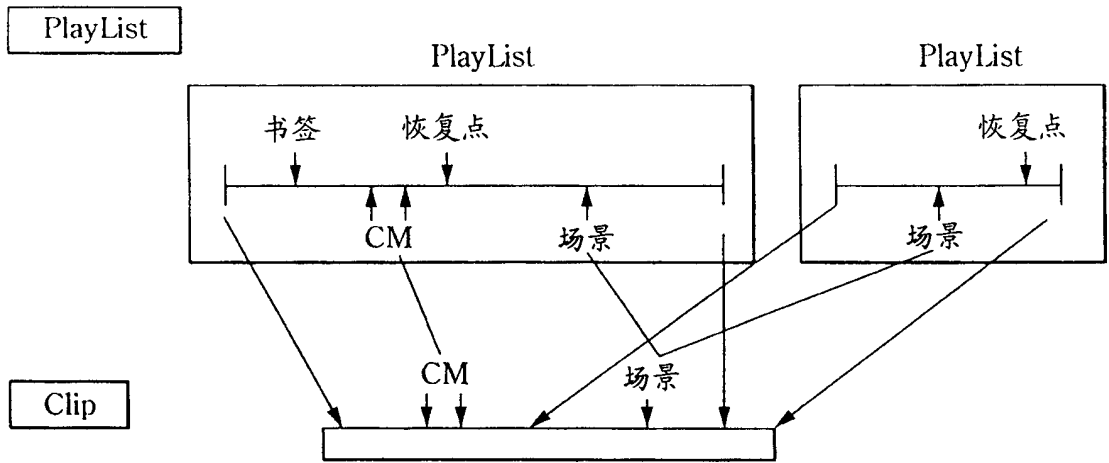


图 9

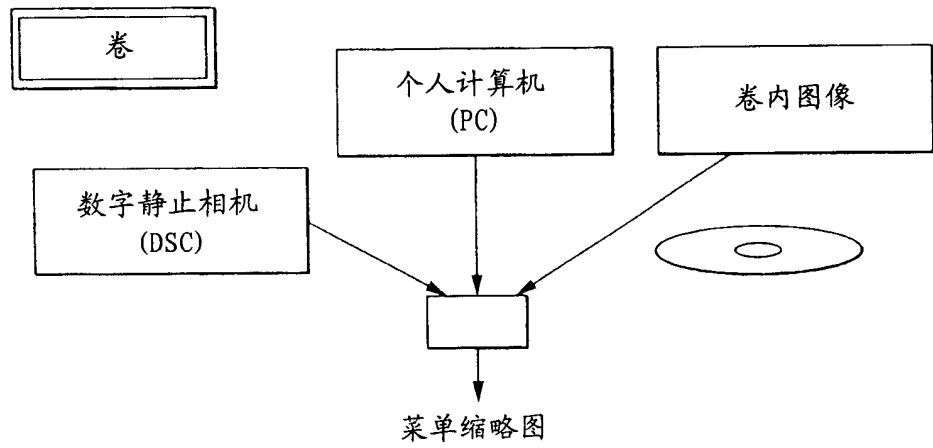


图 10

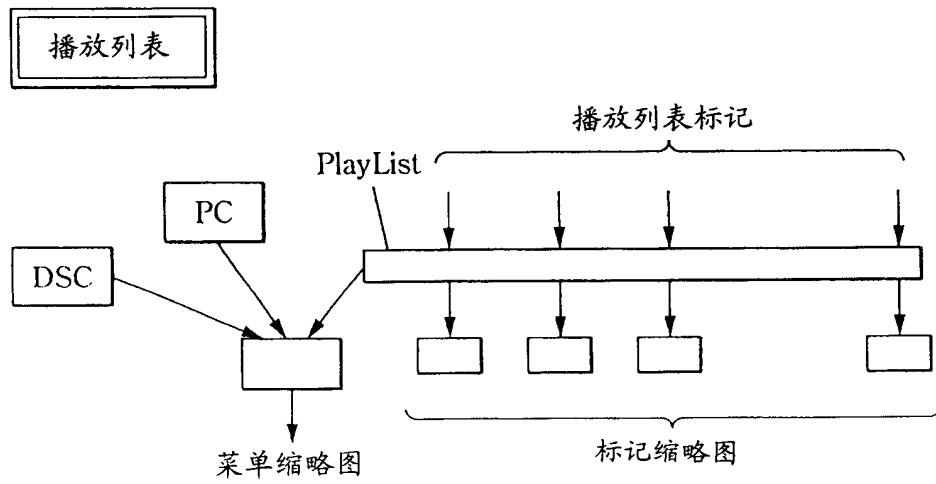


图 11

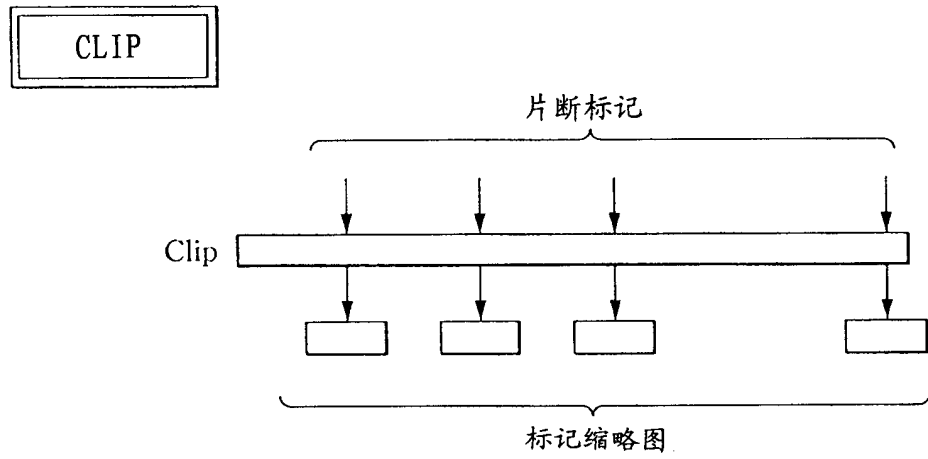


图 12

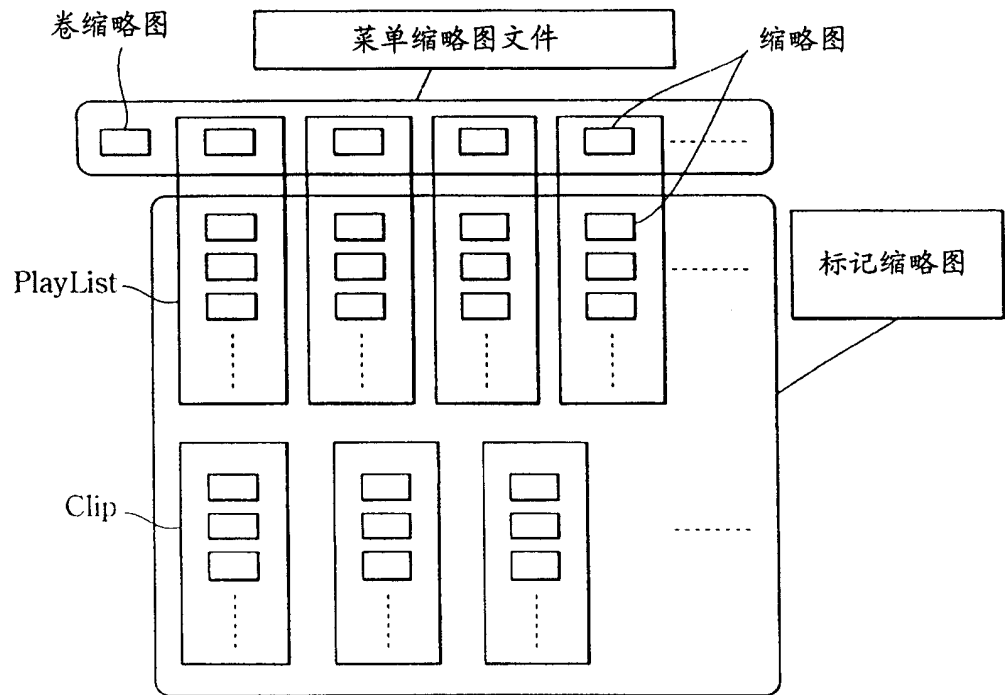


图 13

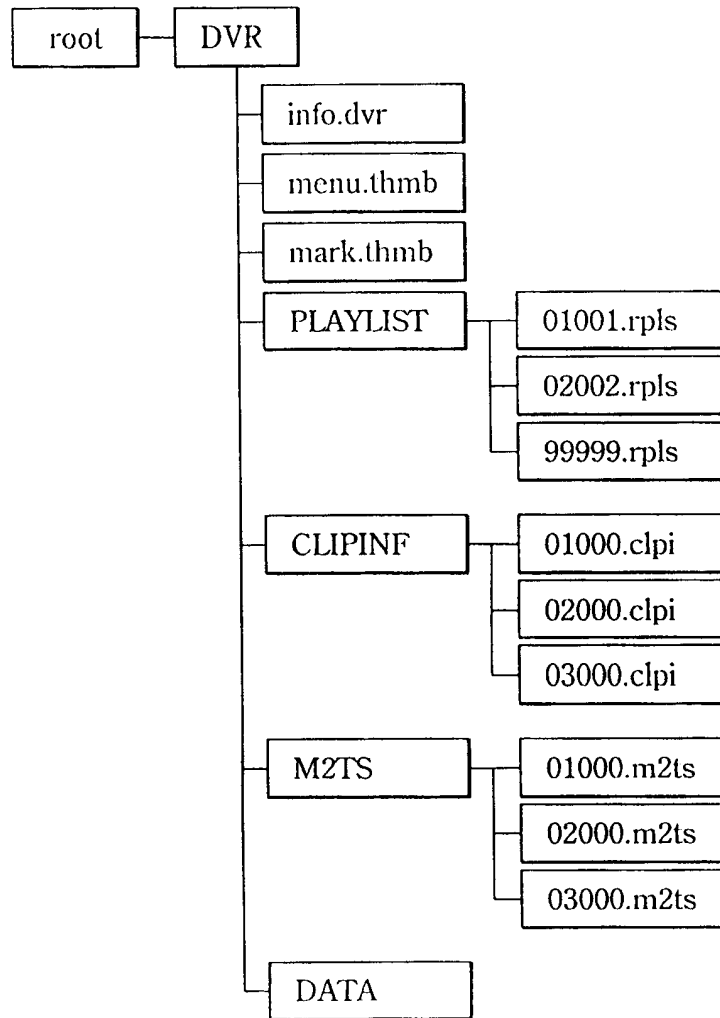


图 14

语法	字节数	缩写
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 15

语法	字节数	缩写
DVRVolume(){		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

图 16

语法	字节数	缩写
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

图 17

语法	字节数	缩写
UIAppInfoVolume(){		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

图 18

值	字符字母编码
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

图 19

语法	字节数	缩写
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

图 20

语法	字节数	缩写
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

图 21

语法	字节数	缩写
MakersPrivateData(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		

图 22

语法	字节数	缩写
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 23

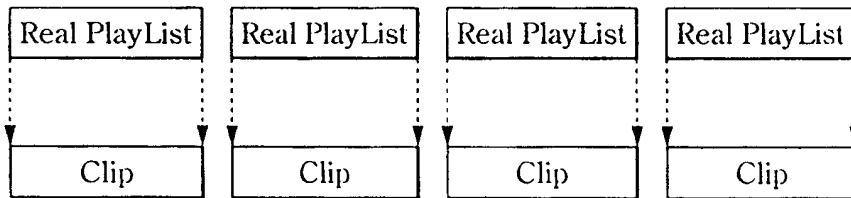


图 24A

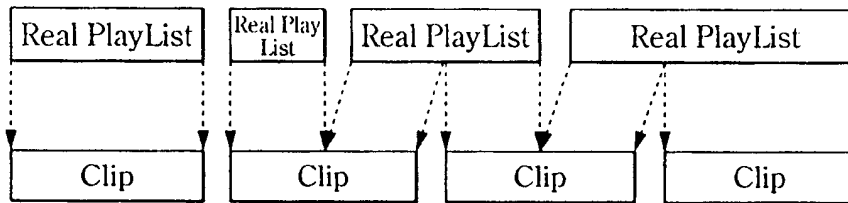


图 24B

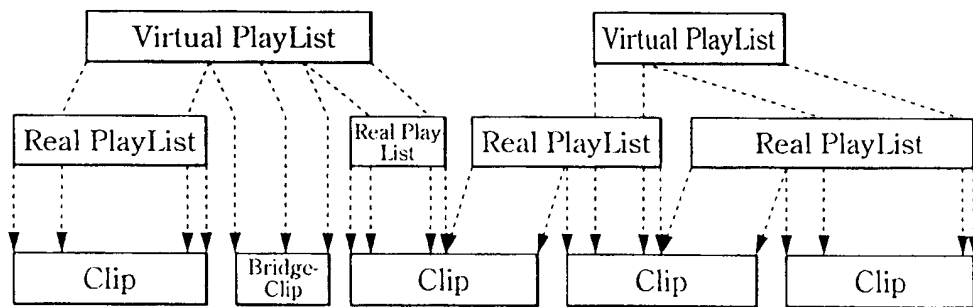


图 24C

语法	字节数	缩写
PlayList(){		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

图 25

PlayList_type	含义
0	AV记录的播放列表 该播放列表中所引用的所有片断必须包含一个或多个视频流
1	音频记录的播放表 该播放列表中所引用的所有片断必须包含一个或多个音频流，并且必须不包含视频流
2-255	保留

图 26

语法	字节数	缩写
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

图 27

write_protect_flag	含义
0b	能自由地删除PlayList
1b	除write_protect_flag之外不能删除或改变PlayList内容

图 28A

is_played_flag	含义
0b	PlayList自从记录以后尚未被再现
1b	PlayList自从记录以后已经被再现过一次

图 28B

archive	含义
00b	没有意义
01b	原始
10b	拷贝
11b	保留

图 28C

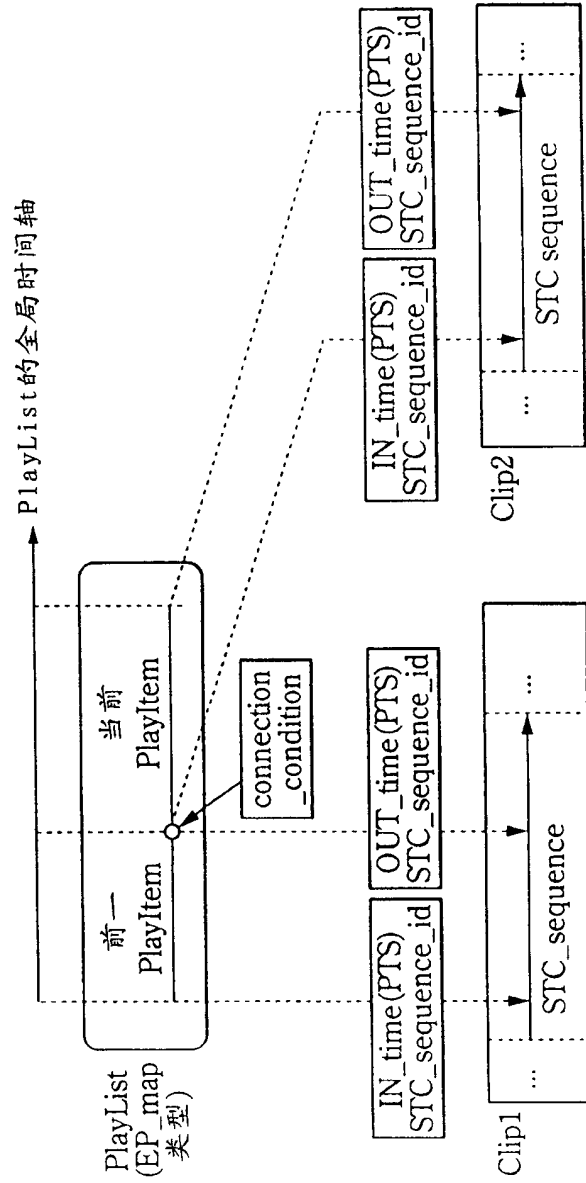


图 29

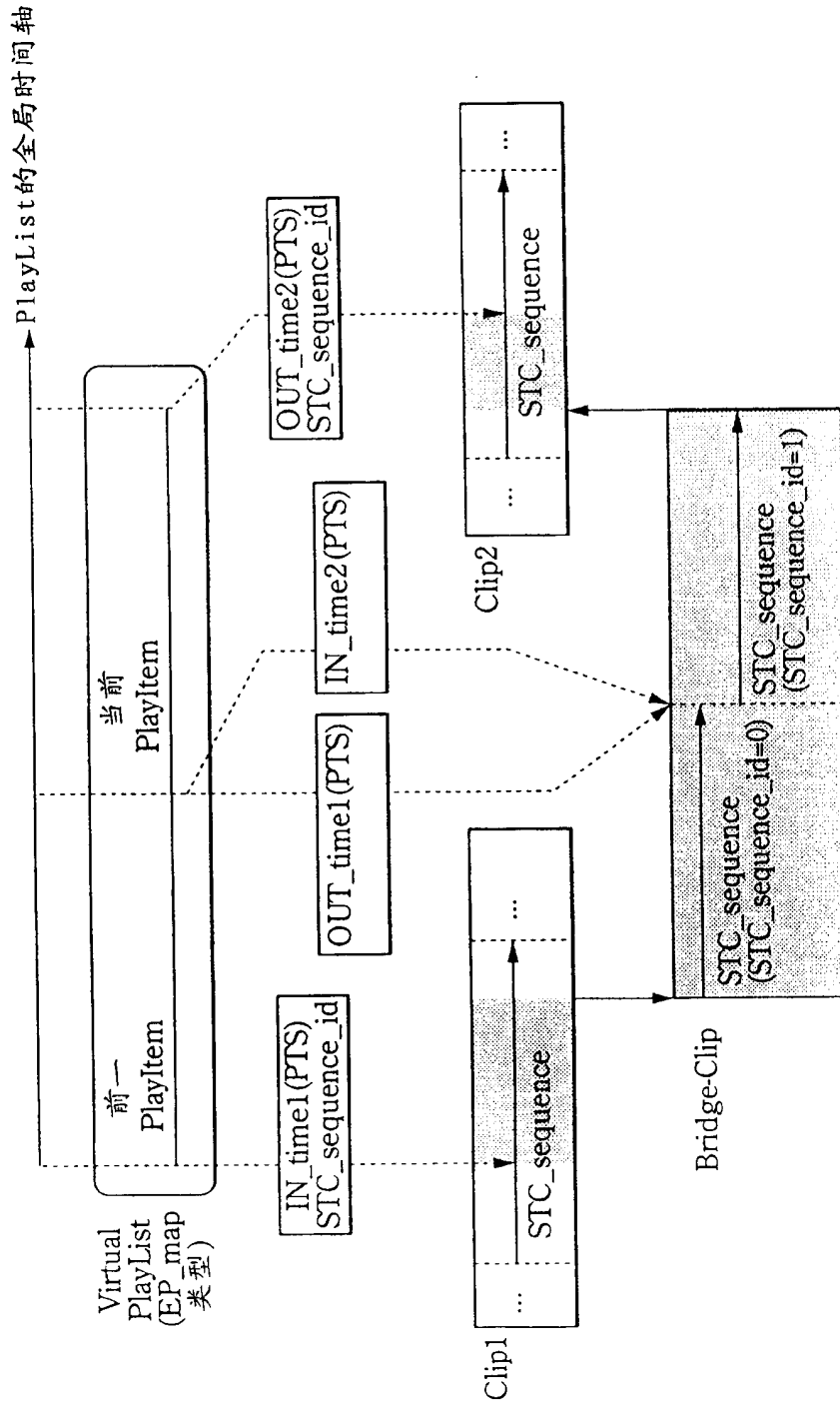


图 30

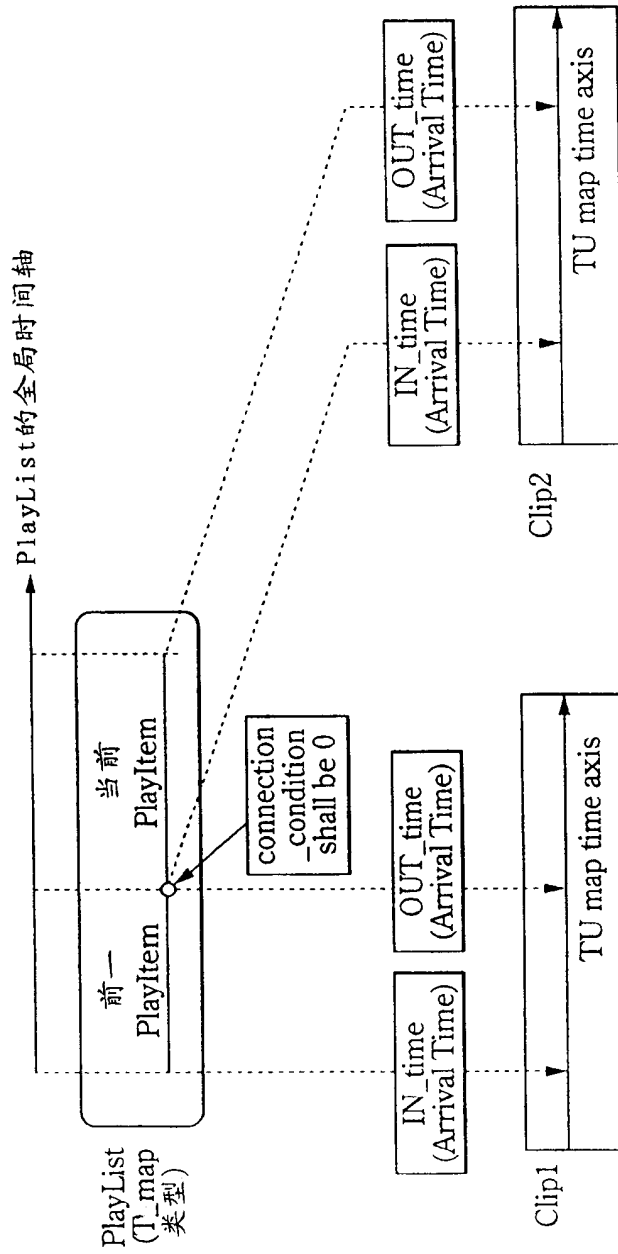


图 31

语法	字节数	缩写
PlayItem(){		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual Playlist>){		
if (<i>connection_condition</i> =='10'){		
BridgeSequenceInfo()		
}		
}		
}		

图 32

PlayList() 中的CPI-type	IN_time语义
EP_map类型	IN_time必须表示对应于PlayItem中第一呈现单元的33位长度的高32位
TU_map类型	IN_time必须是TU_map-time-axis上的时间，并且必须四舍五入到time-unit精度。IN_time用下面公式进行计算： $IN_time = TU_start_time \% 2^{32}$

图 33

PlayList () 中的CPI_type	OUT_time语义
EP_map类型	<p>OUT_time必须由下面公式计算的 Presentation_end_TS值的高32位:</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>其中PTS_out为对应于PlayItem中最后呈现单元的33位长的PTS。AU_duration为最后呈现单元的90kHz显示时间。</p>
TU_map类型	<p>OUT_time必须是TU_map_time_axis上的时间，并且四舍五入到time_unit精度。OUT_time用下面公式进行计算:</p> $\text{OUT_time} = \text{TU_start_time} \% 2^{32}$

图 34

connection _condition	含义
00	<ul style="list-style-type: none"> . 前一PlayItem到当前PlayItem的连接不能确定为无缝连接。 . 如果PlayList的CPI_type为TU_map类型, 该值必须在connection_condition中设置。
01	<ul style="list-style-type: none"> . 只有当PlayList的CPI_type为EP_map类型时, 才允许该状态。 . 前一PlayItem和当前PlayItem由于系统时基(STC基)的非连续点表示切分。
10	<ul style="list-style-type: none"> . 只有当PlayList的CPI_type为EP_map类型时, 才允许该状态。 . 该状态仅对Virtual PlayList允许。 . 前一PlayItem到当前PlayItem的连接确定为无缝连接。 . 前一PlayItem使用BridgeSequence连接到当前PlayItem。DVR MPEG-2传输流必须遵从后面描述的DVR-STD。
11	<ul style="list-style-type: none"> . 只有当PlayList的CPI_type为EP_map类型时, 才允许该状态。 . 前一PlayItem到当前PlayItem的连接确定为无缝连接。 . 前一PlayItem不使用BridgeSequence连接到当前PlayItem。DVR MPEG-2传输流必须遵从后面描述的DVR-STD。

图 35

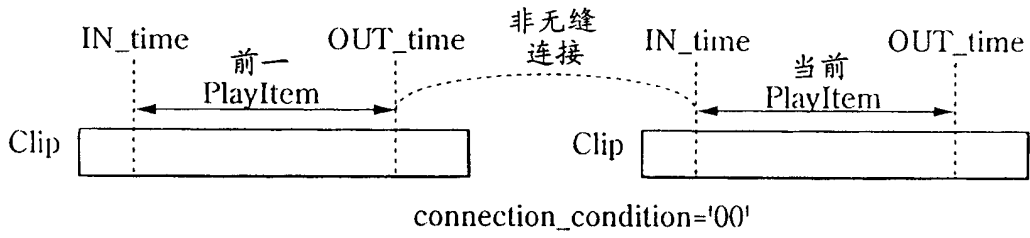


图 36A

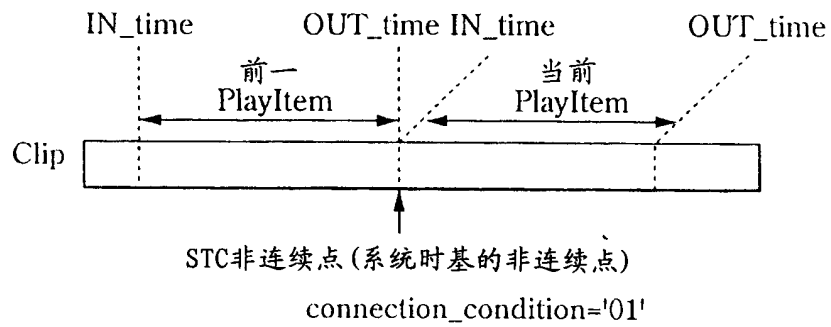


图 36B

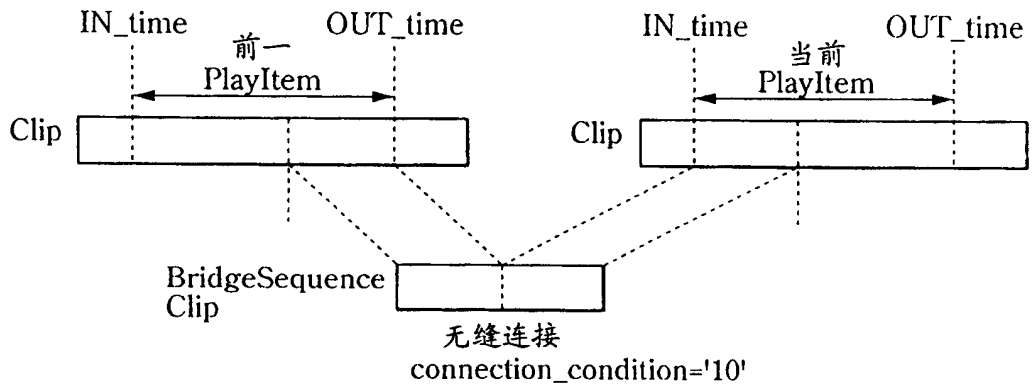


图 36C

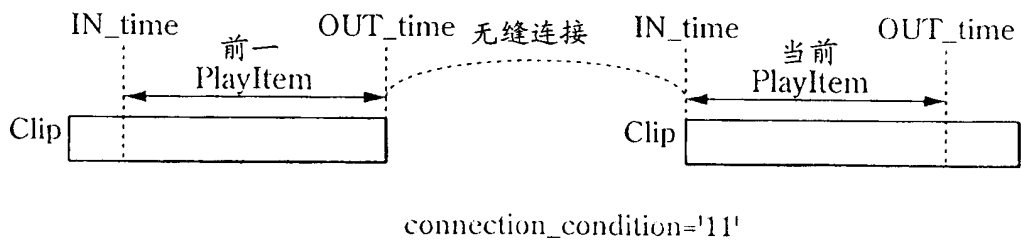


图 36D

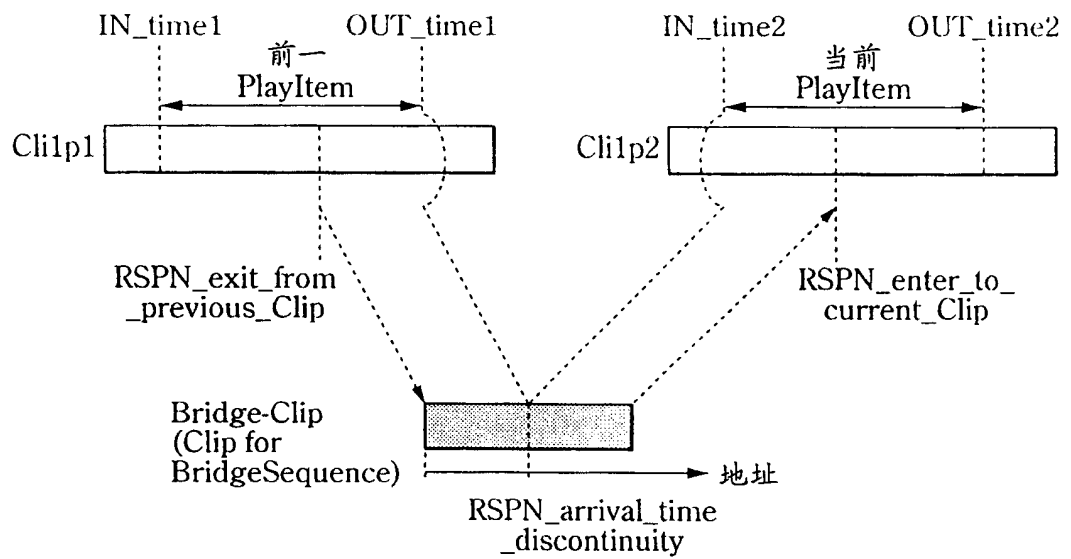


图 37

语法	字节数	缩写
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

图 38

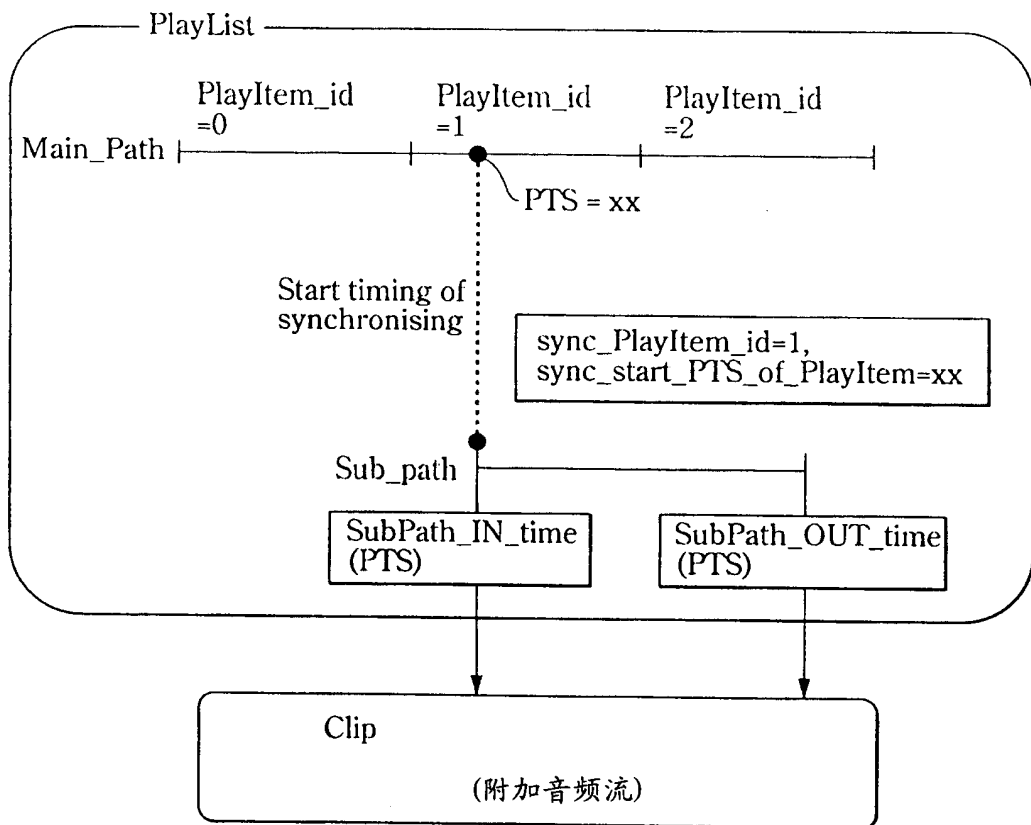


图 39

语法	字节数	缩写
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

图 40

SubPath_type	含义
0x00	辅助音频流路径
0x01-0xff	保留

图 41

语法	字节数	缩写
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 42

Mark_type	含义	注释
0x00	resume-mark	回放恢复点。在PlayListMark()中定义的回放恢复点数必须为0或1。
0x01	book-mark	PlayList的回放入口点。该标记能由用户进行设置，并且用作喜爱场景的标记指定开始点。
0x02	skip-mark	跳读标记点。播放器从该点跳读节目到节目的结束处。在PlayListMark()中定义的跳读标记点数必须为0或1。
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

图 43

PlayList O 中的CPI_type	mark_time_stamp语义
EP_map类型	mark_time_stamp必须是表示对应于标记 所引用呈现单元的33位长PTS的32位
TU_map类型	mark_time_stamp 必须是 TU_map.time-axis上的时间, 并且必须四舍五入到time-unit精度。 mark_time_stamp用下面公式进行计算: $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

图 44

语法	字节数	缩写
zzzzz.clpi {		
STC_Info_Start_address	32	uimbsf
ProgramInfo_Start_address	32	uimbsf
CPI_Start_address	32	uimbsf
ClipMark_Start_address	32	uimbsf
MakersPrivateData_Start_address	32	uimbsf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

图 45

语法	字节数	缩写
ClipInfo{		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
<i>if (Clip_stream_type==1) // Bridge-Clip AV stream</i>		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

图 46

Clip_stream_type	含义
0	Clip AV流
1	Bridge-Clip AV流
2-255	保留

图 47

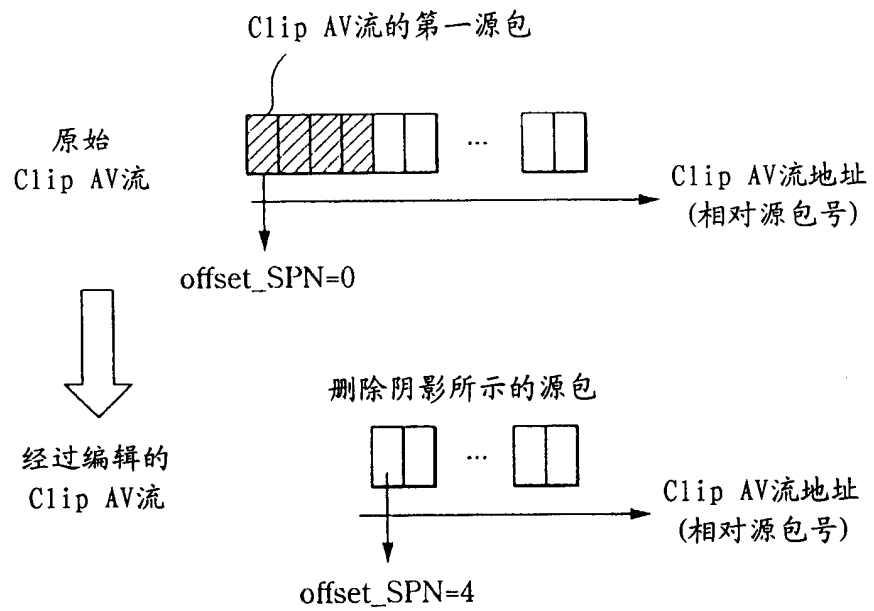


图 48

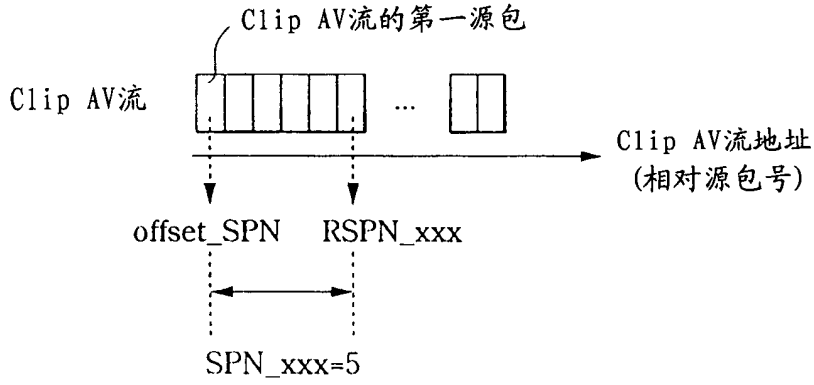


图 49

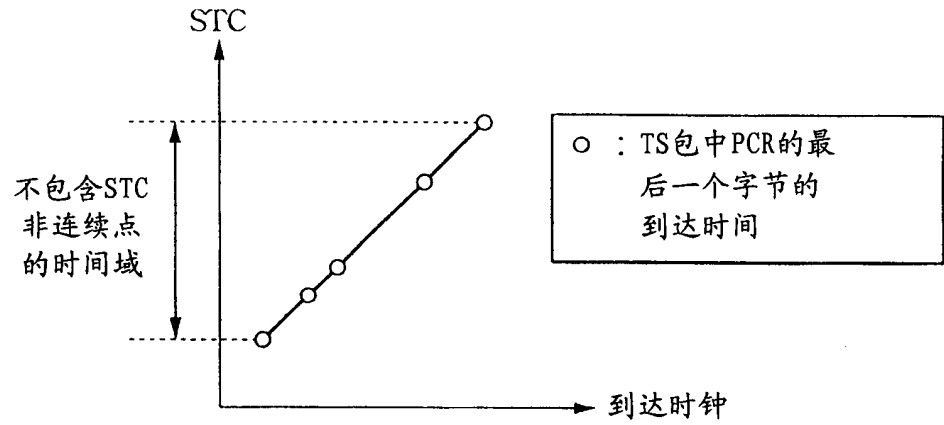


图 50A

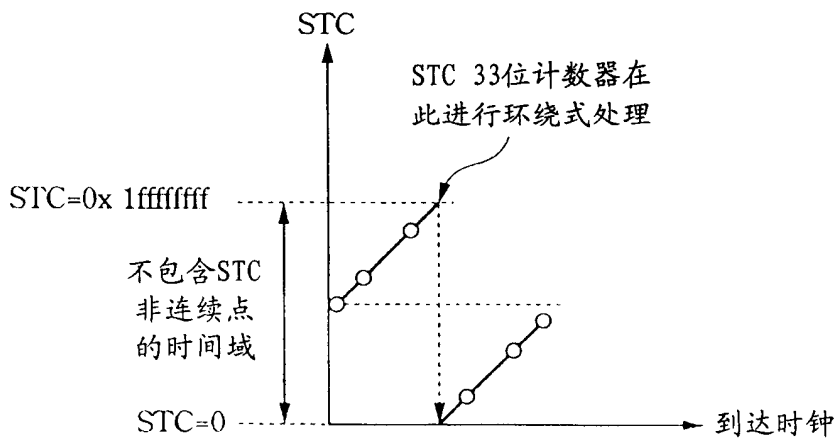


图 50B

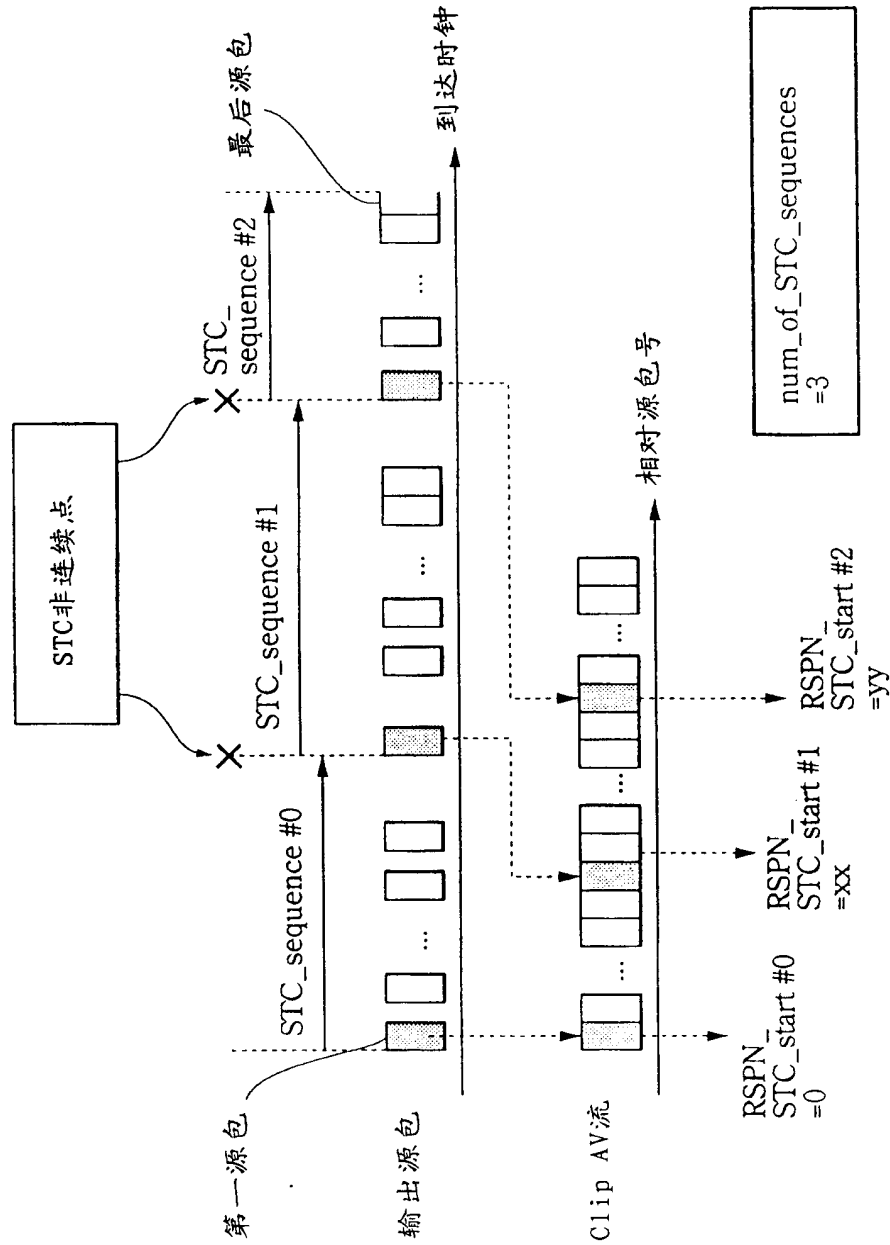


图 51

语法	字节数	缩写
STC_Info(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0; STC_sequence_id<num_of_STC_sequences; STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

图 52

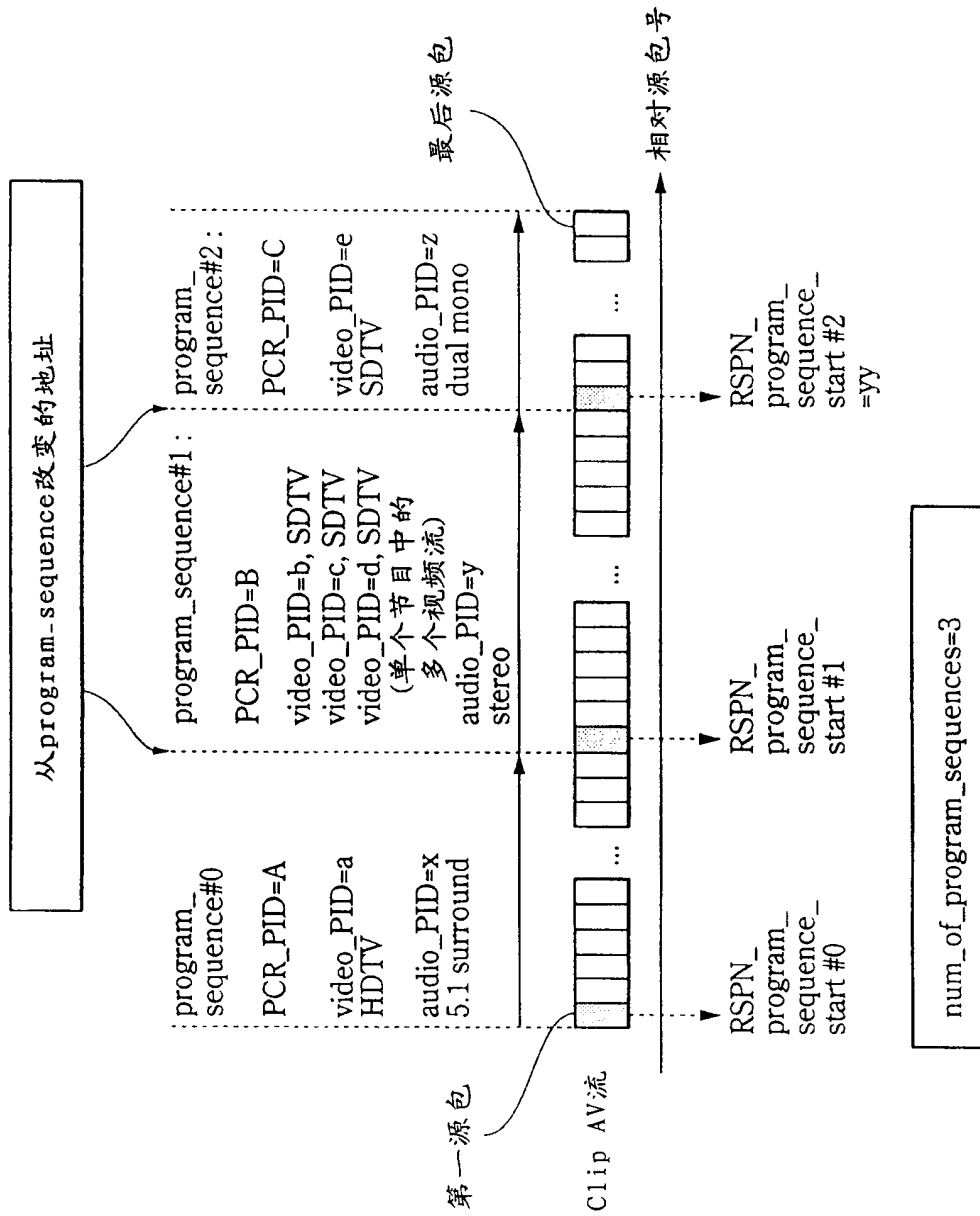


图 53

语法	字节数	缩写
ProgramInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

图 54

语法	字节数	缩写
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

图 55

video_format	含义
0	480i
1	576i
2	480p (包括640×480p格式)
3	1080i
4	720p
5	1080p
6-254	保留
255	没有信息

图 56

frame_rate	含义
0	禁止
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	保留
255	没有信息

图 57

display_aspect_ratio	含义
0	禁止
1	保留
2	4:3显示纵横比
3	16:9显示纵横比
4-254	保留
255	没有信息

图 58

语法	字节数	缩写
AudioCodingInfo() {		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

图 59

audio_coding	含义
0	MPEG-1音频层I或II
1	Dolby AC-3音频
2	MPEG-2 AAC
3	MPEG-2多声道音频, 向下兼容于MPEG-1
4	SESF LPCM音频
5-254	保留
255	没有信息

图 60

audio_component_type	含义
0	单个单声道
1	双单声道
2	立体声 (2声道)
3	多语言, 多声道
4	环绕声
5	用于视力损伤者的音频描述
6	用于听力困难的音频
7-254	保留
255	没有信息

图 61

sampling_frequency	含义
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	保留
255	没有信息

图 62

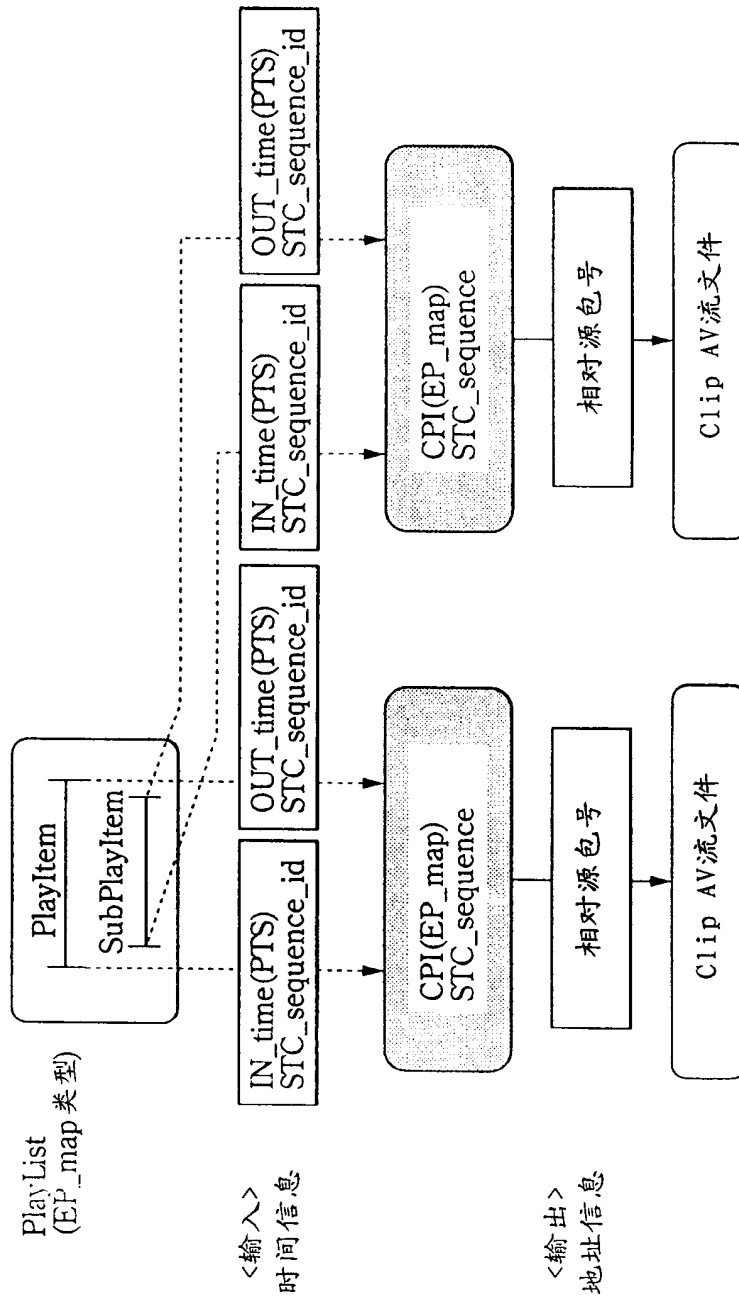


图 63

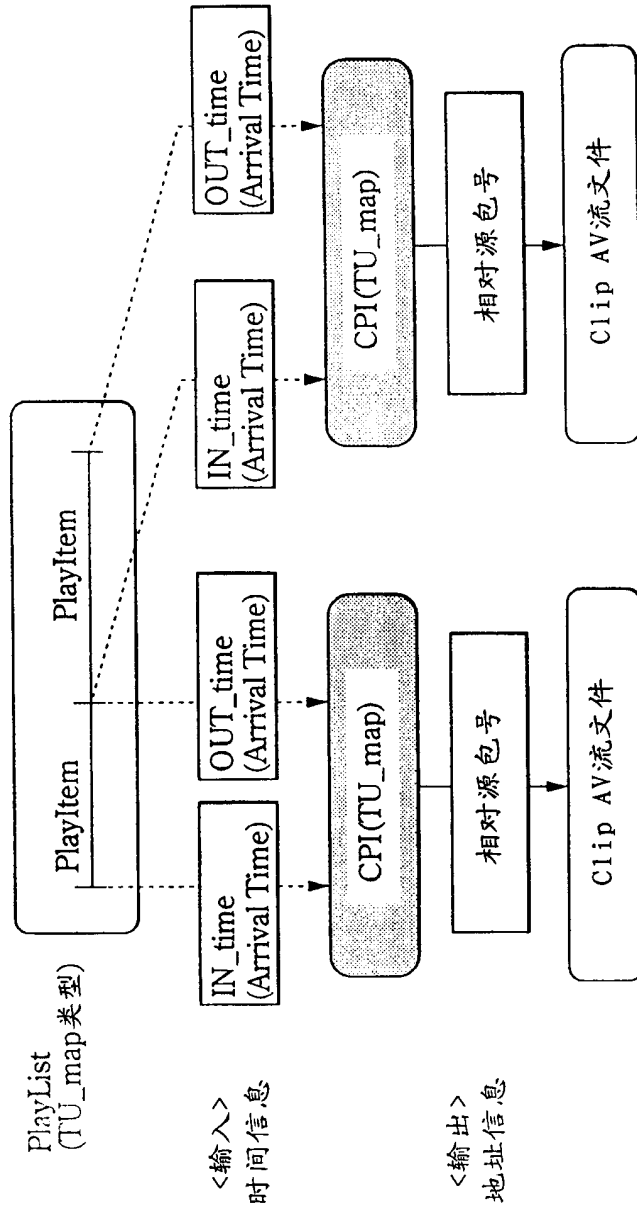


图 64

语法	字节数	缩写
CPI() {		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

图 65

CPI_type	含义
0	EP map类型
1	TU map类型

图 66

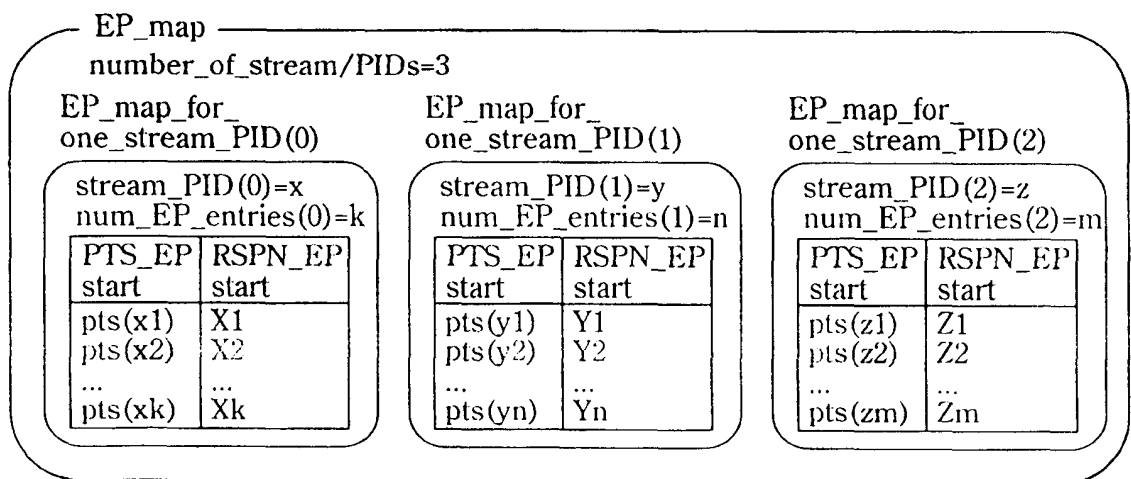
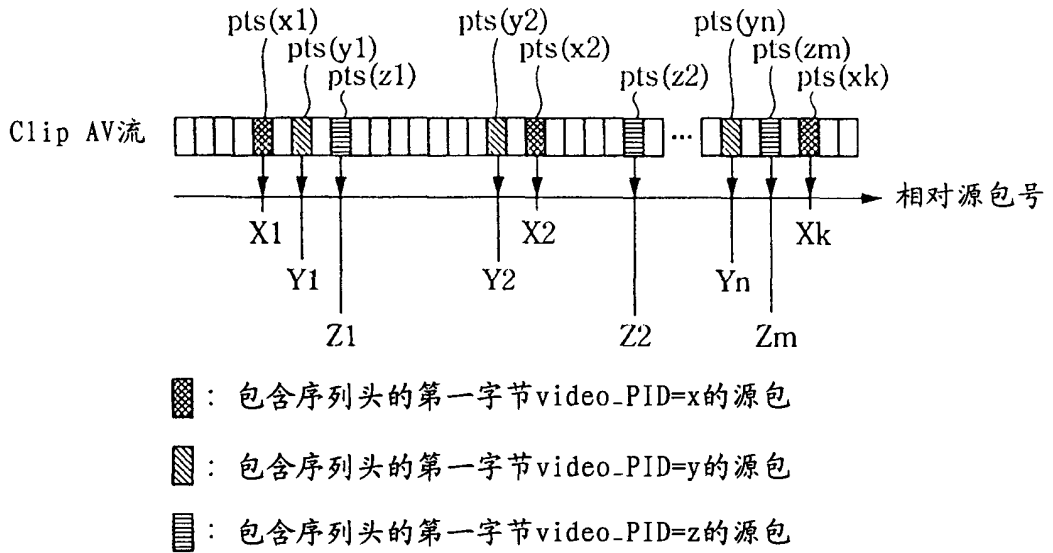
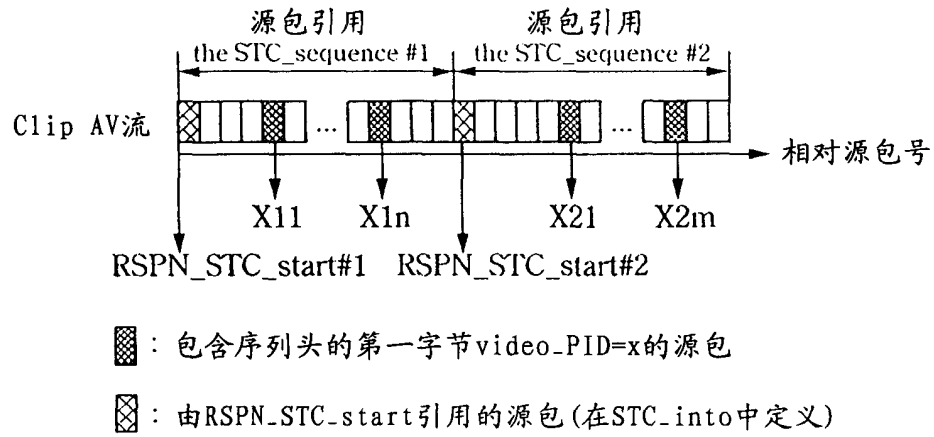


图 67



EP_map_for_one_stream_PID
video_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11) 属于STC_sequence#1的数据
...	...	
pts(x1n)	X1n) 属于STC_sequence#2的数据
...	...	
pts(x21)	X21	
...	...	
pts(x2m)	X2m	

边界

RSPN_STC_start #2 < X21

图 68

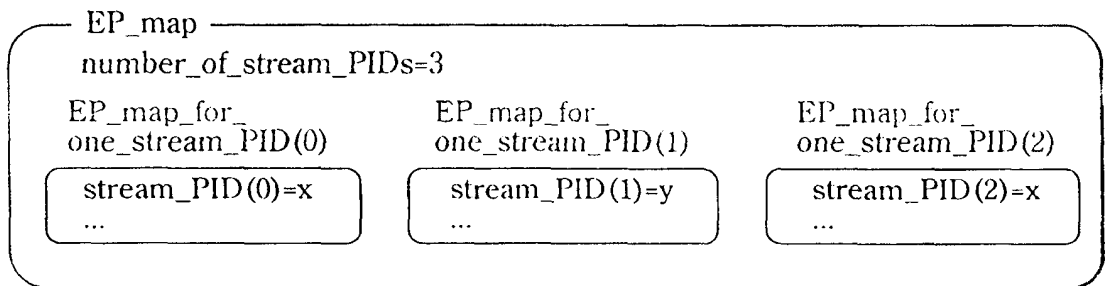
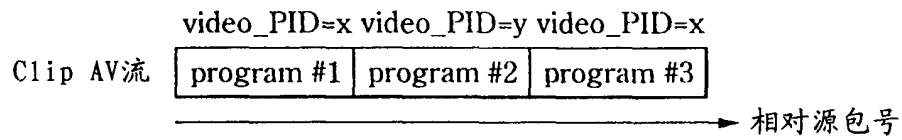


图 69

语法	字节数	缩写
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

图 70

EP_type	含义
0	视频
1	音频
2-15	保留

图 71

语法	字节数	缩写
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

图 72

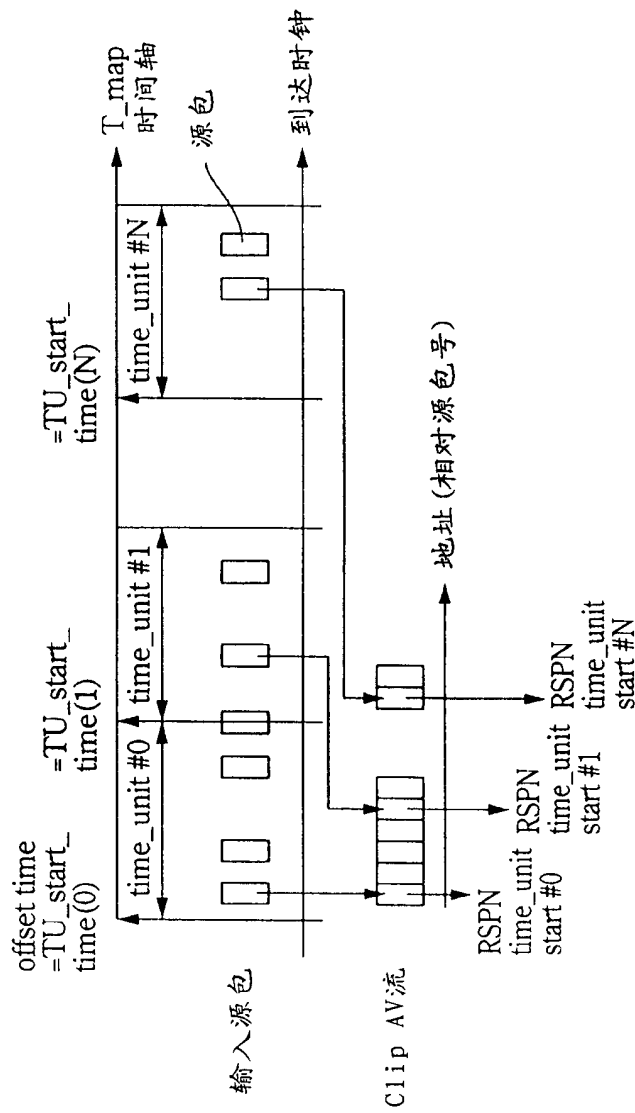


图 73

语法	字节数	缩写
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

图 74

语法	字节数	缩写
ClipMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_clip_marks; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

图 75

Mark_type	MEANING	注释
0x00-0x8F	reserved	为PlayListMark()保留
0x90	Event-start mark	表示节目开始点的标记点
0x91	Local event-start mark	表示节目中局部场景的标记点
0x92	Scene-start mark	示出场景变化点的标记点
0x93-0xFF	reserved	

图 76

PlayList () 中的CPI_type	mark-time-stamp语义
EP_map类型	mark-time-stamp必须表示对应于标记 所引用呈现单元的33位长PTS的高32位
TU_map类型	mark-time-stamp必须是TU_map-time-axis上的时间, 并且必须四舍五入到time-unit精度。 mark-time-stamp用下面公式进行计算: $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

图 77

语法	字节数	缩写
menu.thmb/mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for (i=0; i<N1; i++)		
padding_word	16	bslbf
}		

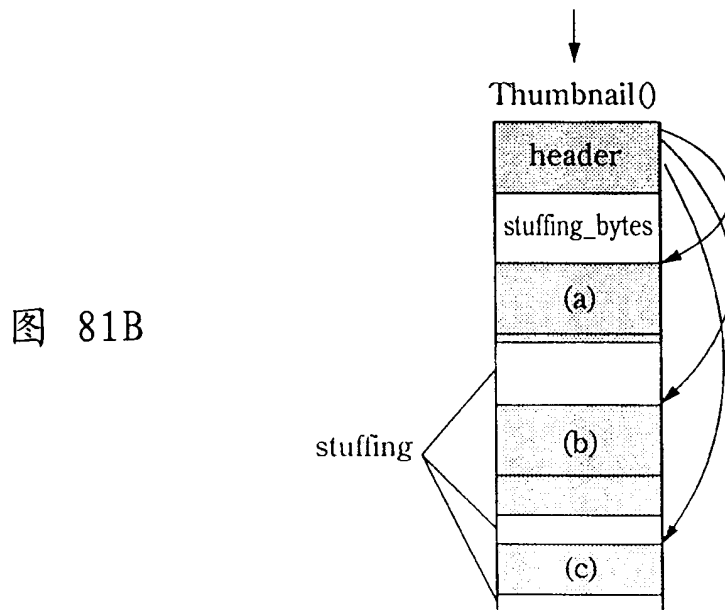
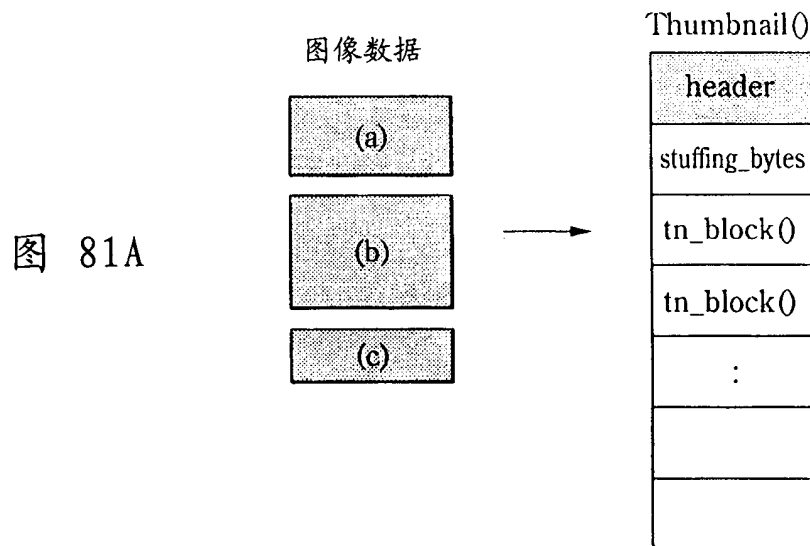
图 78

语法	字节数	缩写
Thumbnail(){		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_ size*1024*8	
}		
}		
}		

图 79

Thumbnail_picture_format	含义
0x00	MPEG-2视频I-图像
0x01	DCF (受限制的JPEG)
0x02	PNG
0x03-0xff	保留

图 80



DVR MPEG-2传输流

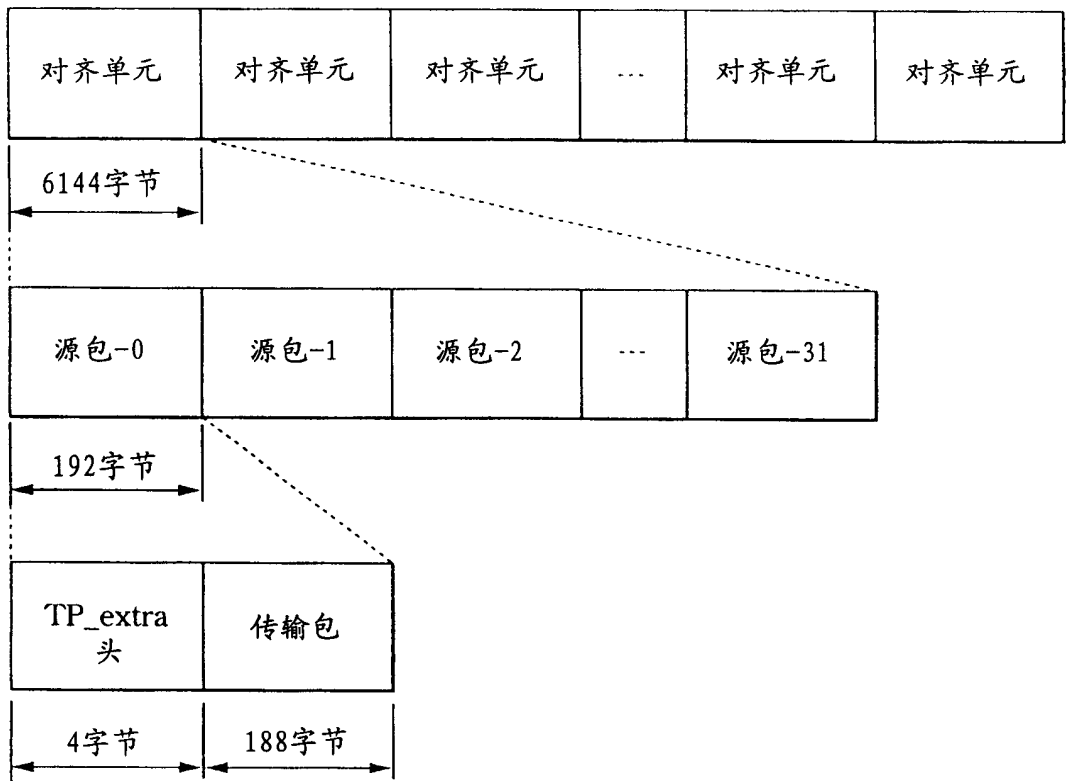


图 82

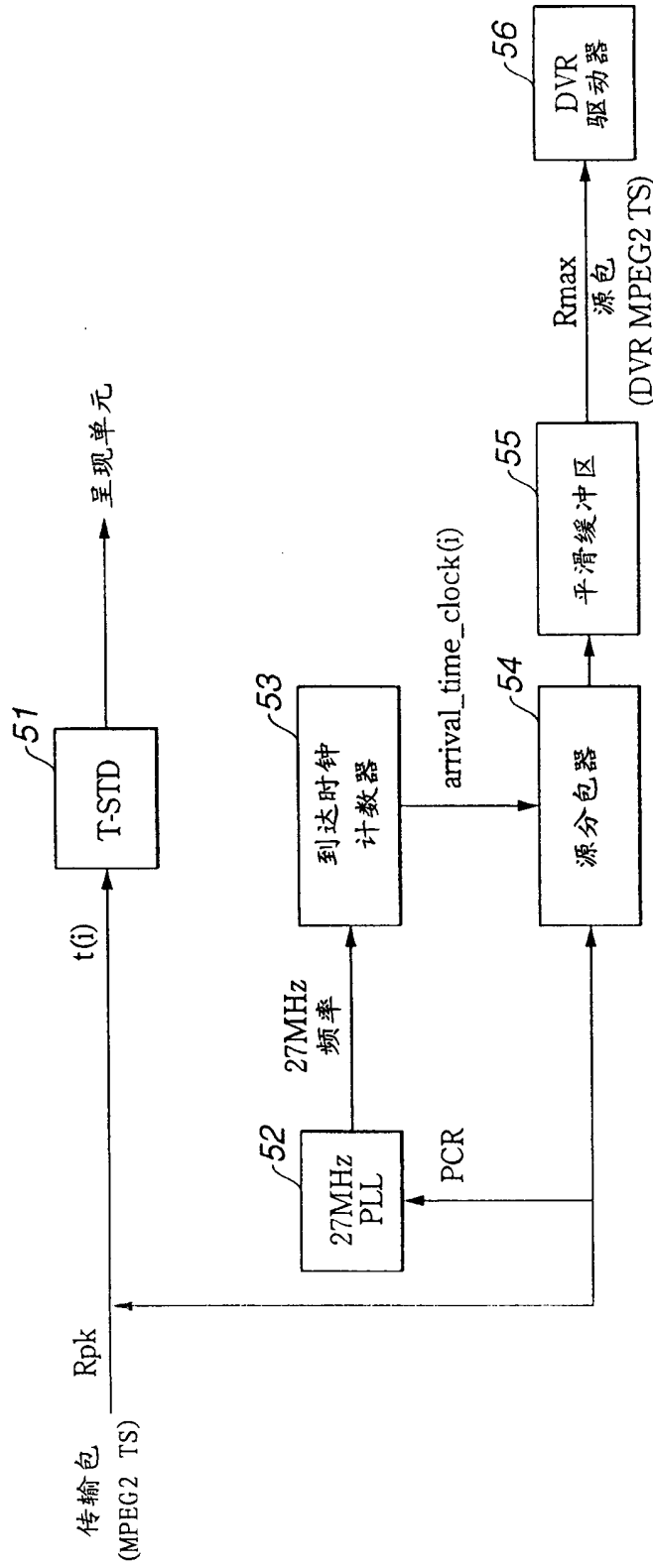


图 83

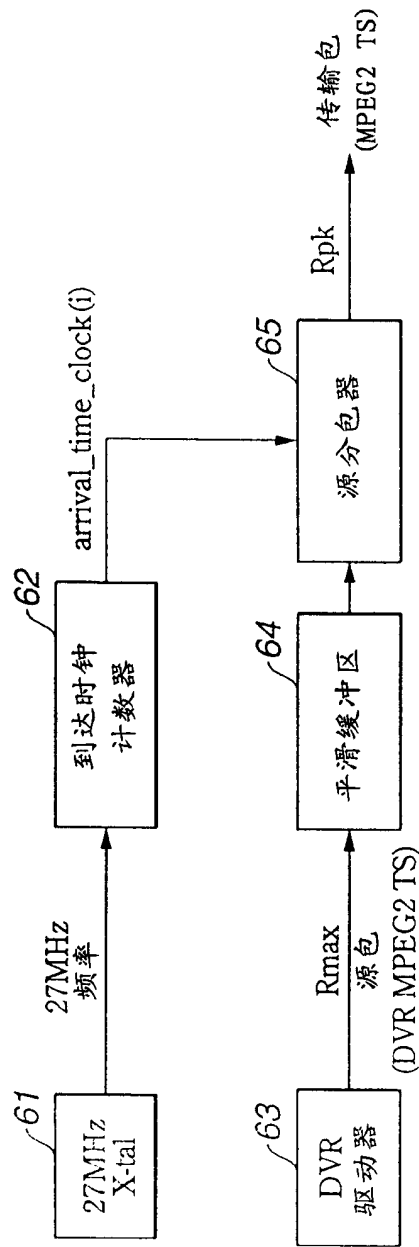


图 84

语法	字节数	缩写
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

图 85

语法	字节数	缩写
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

图 86

copy_permission _indicator	含义
00	自由拷贝
01	不再拷贝
10	拷贝一次
11	禁止拷贝

图 87

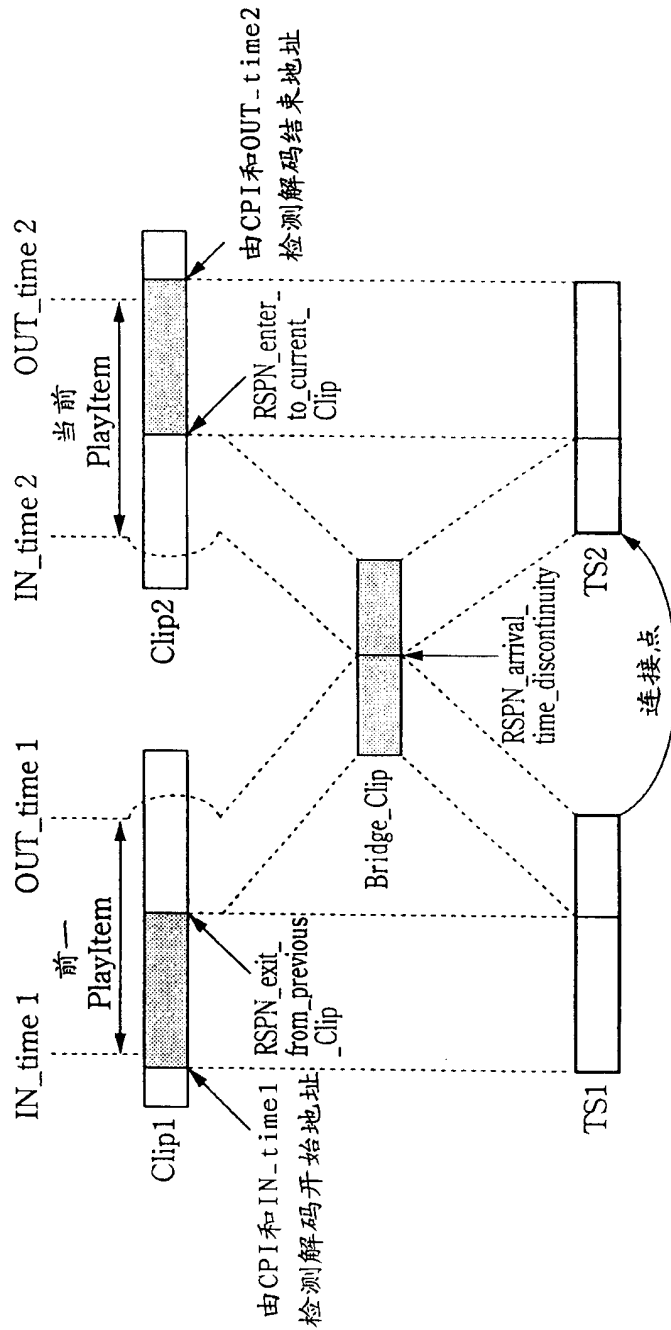


图 88

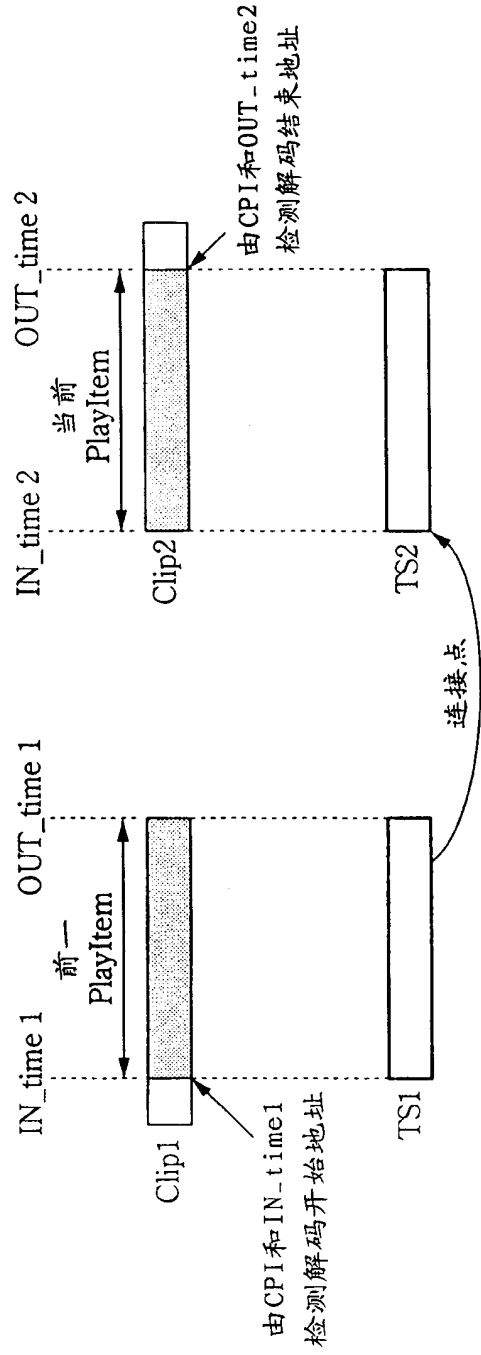


图 89

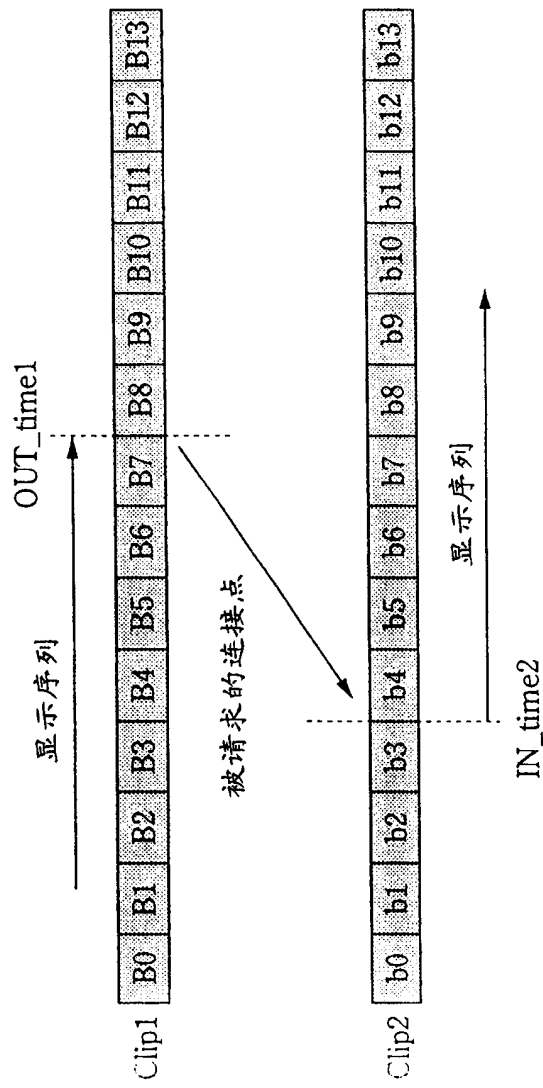


图 90

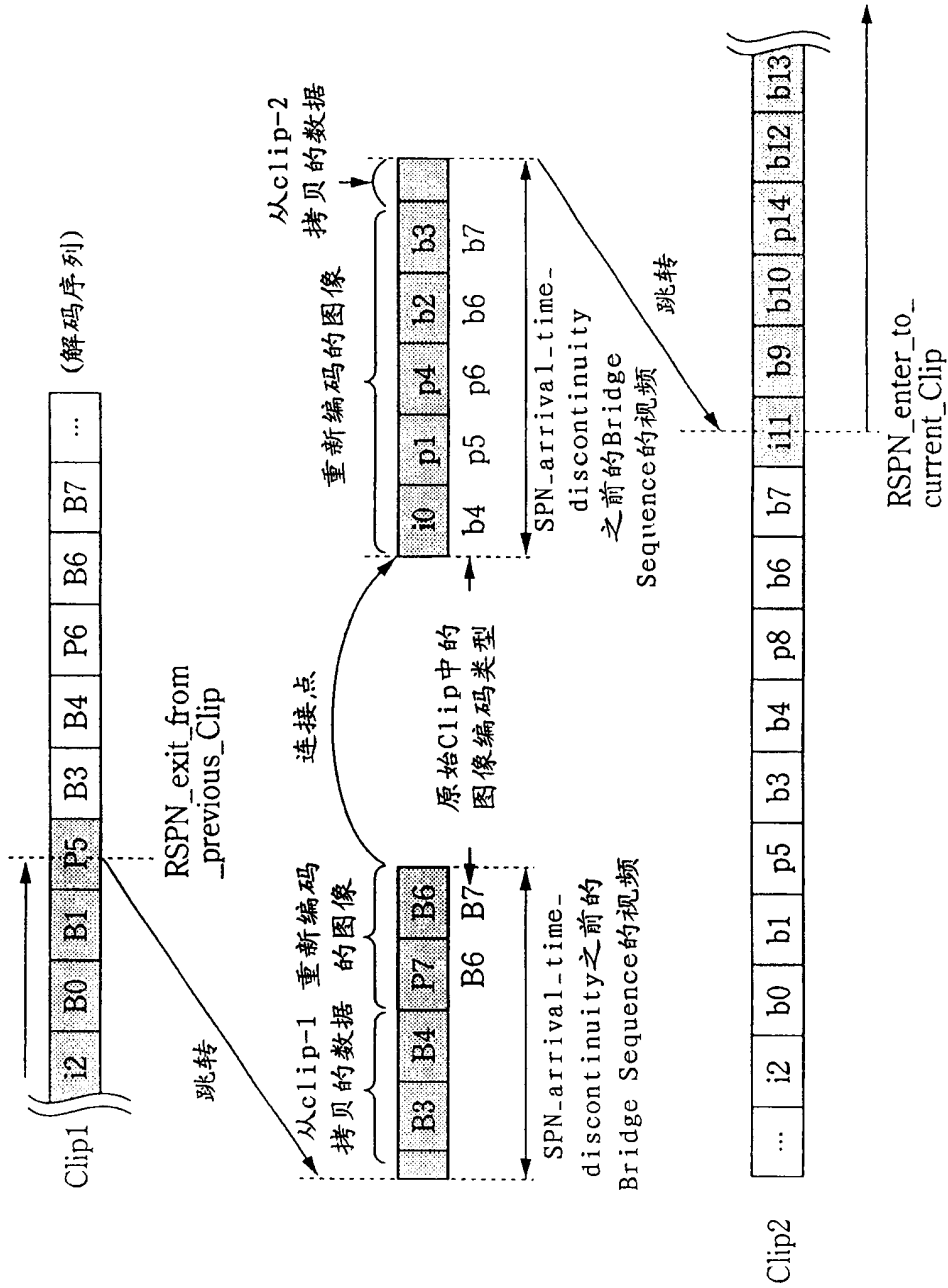


图 91

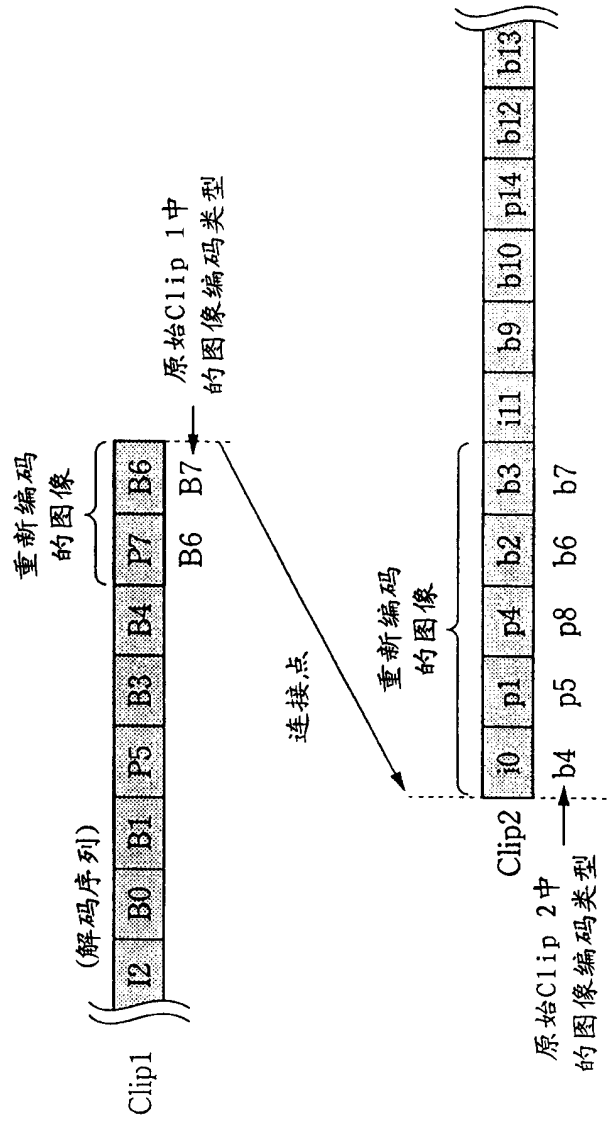


图 92

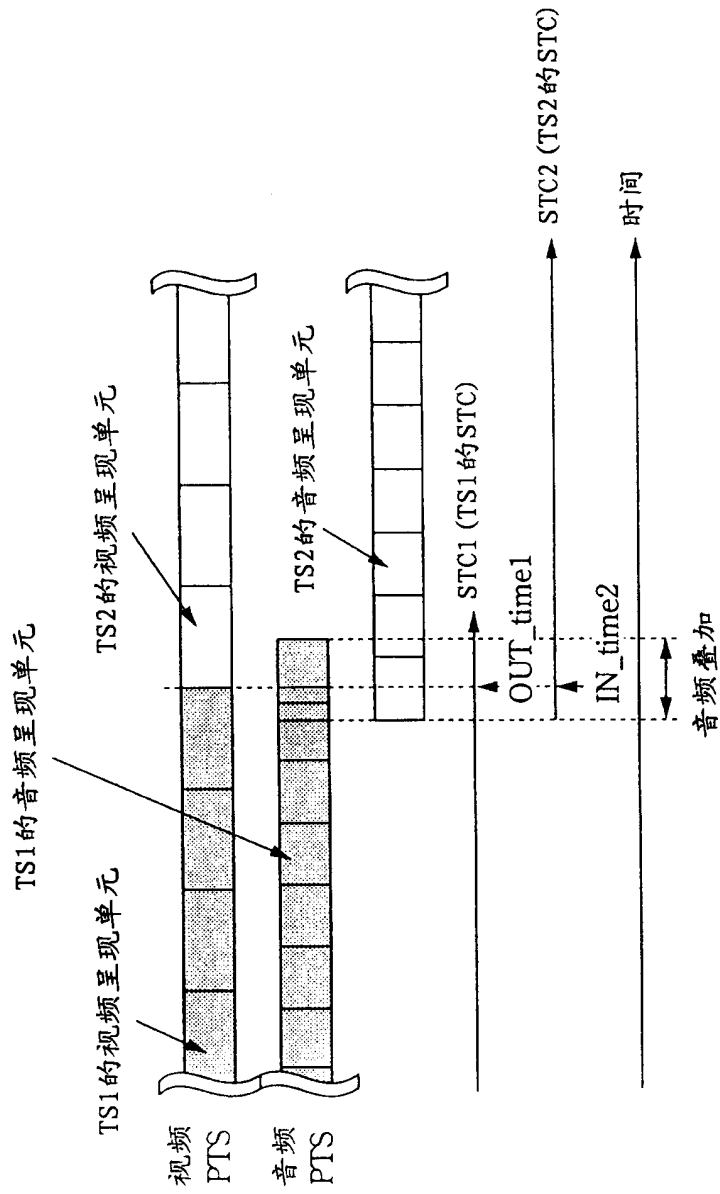


图 93

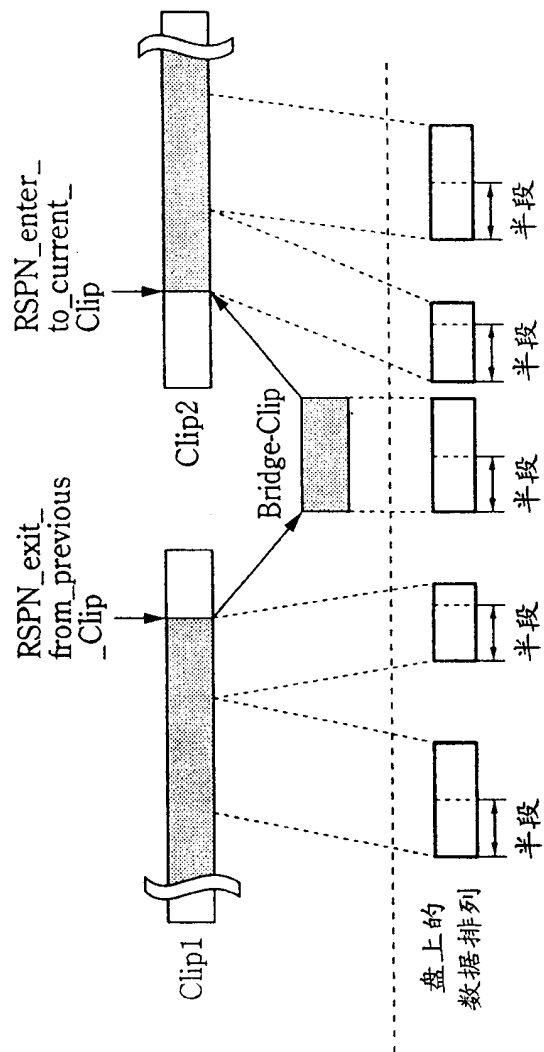


图 94

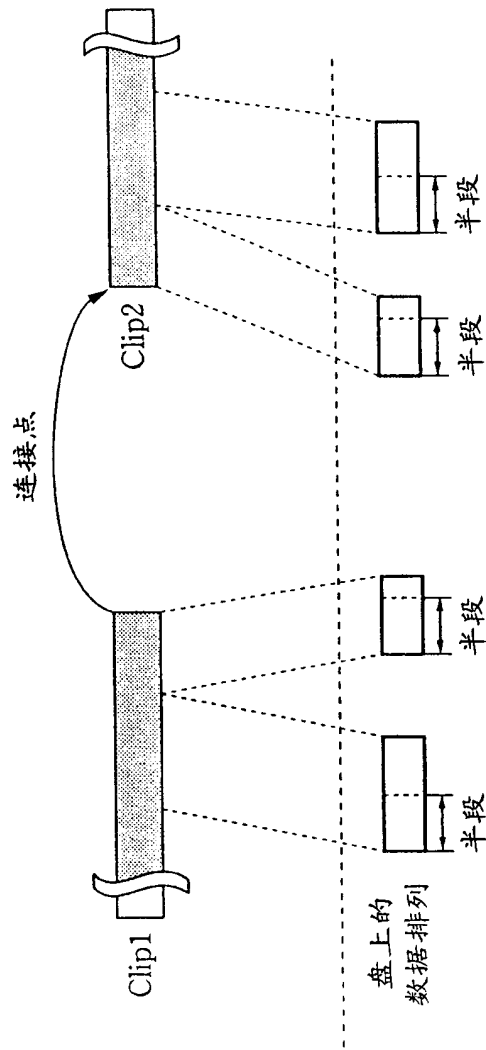


图 95

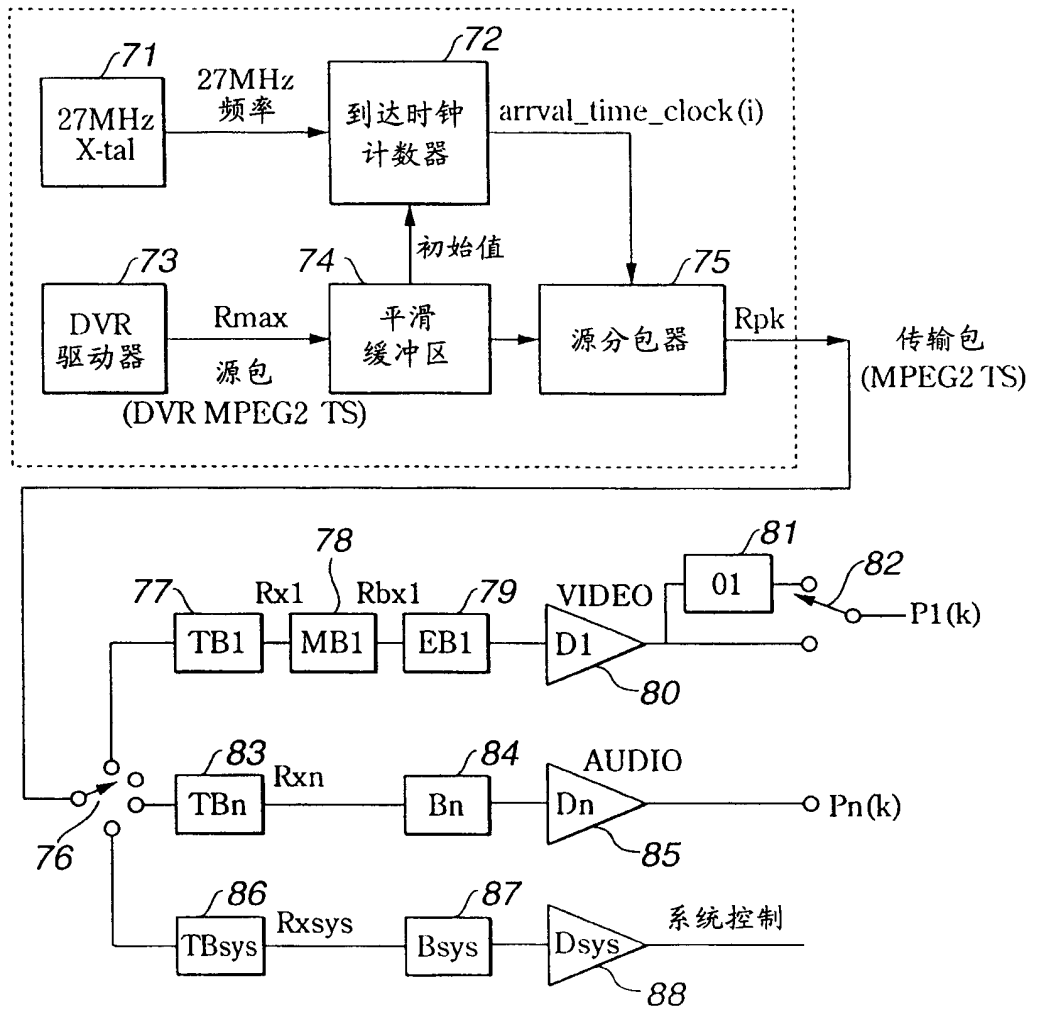


图 96

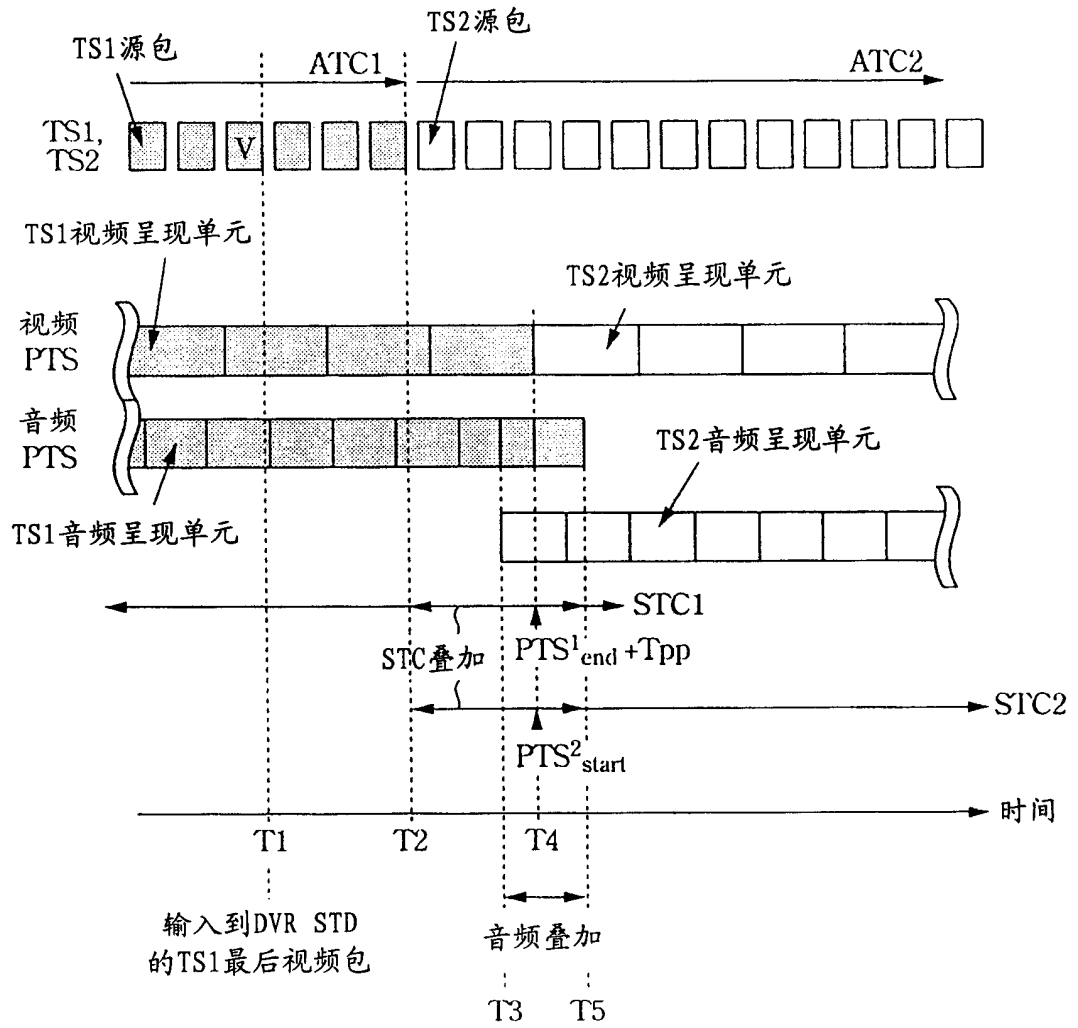


图 97

语法	字节数	缩写
xxxxx.rpls / yyyyy.vpls {		
version_number	8*4	bslbf
PlayList_start_address	32	uimsbf
PlayListMark_start_address	32	uimsbf
MakersPrivateData_start_address	32	uimsbf
reserved_for_future_use	160	bslbf
UIAppInfoPlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayList()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
}		

图 98

语法	字节数	缩写
UIAppInfoPlayList(){		
length	32	uimsbf
PlayList_service_type		
PlayList_character_set	8	uimsbf
reserved_for_word_align	3	bslbf
playback_control_flag	1	uimsbf
write_protect_flag	1	uimsbf
is_played_flag	1	uimsbf
archive	2	uimsbf
record_time_and_date	4*14	bslbf
duration	4*6	bslbf
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
channel_number	16	uimsbf
reserved_for_word_align	8	bslbf
channel_name_length	8	uimsbf
channel_name	8*20	bslbf
PlayList_name_length	8	uimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	uimsbf
PlayList_detail	8*1200	bslbf
}		

图 99

语法	字节数	缩写
PlayList(){		
length	32	uimsbf
reserved_for_word_align	15	bslbf
CPI_type	1	bslbf
number_of_PlayItems	16	uimsbf
if (<Virtual Playlist> && CPI_type==0){		
number_of_SubPlayItems	16	uimsbf
}else{		
reserved_for_word_align	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem()		
}		
if (<Virtual Playlist>&& CPI_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem()		
}		
}		
}		

图 100

语法	字节数	缩写
SubPlayItem(){		
length	16	uimsbf
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
STC_sequence_id	8	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
sync_PlayItem_id	16	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
}		

图 101

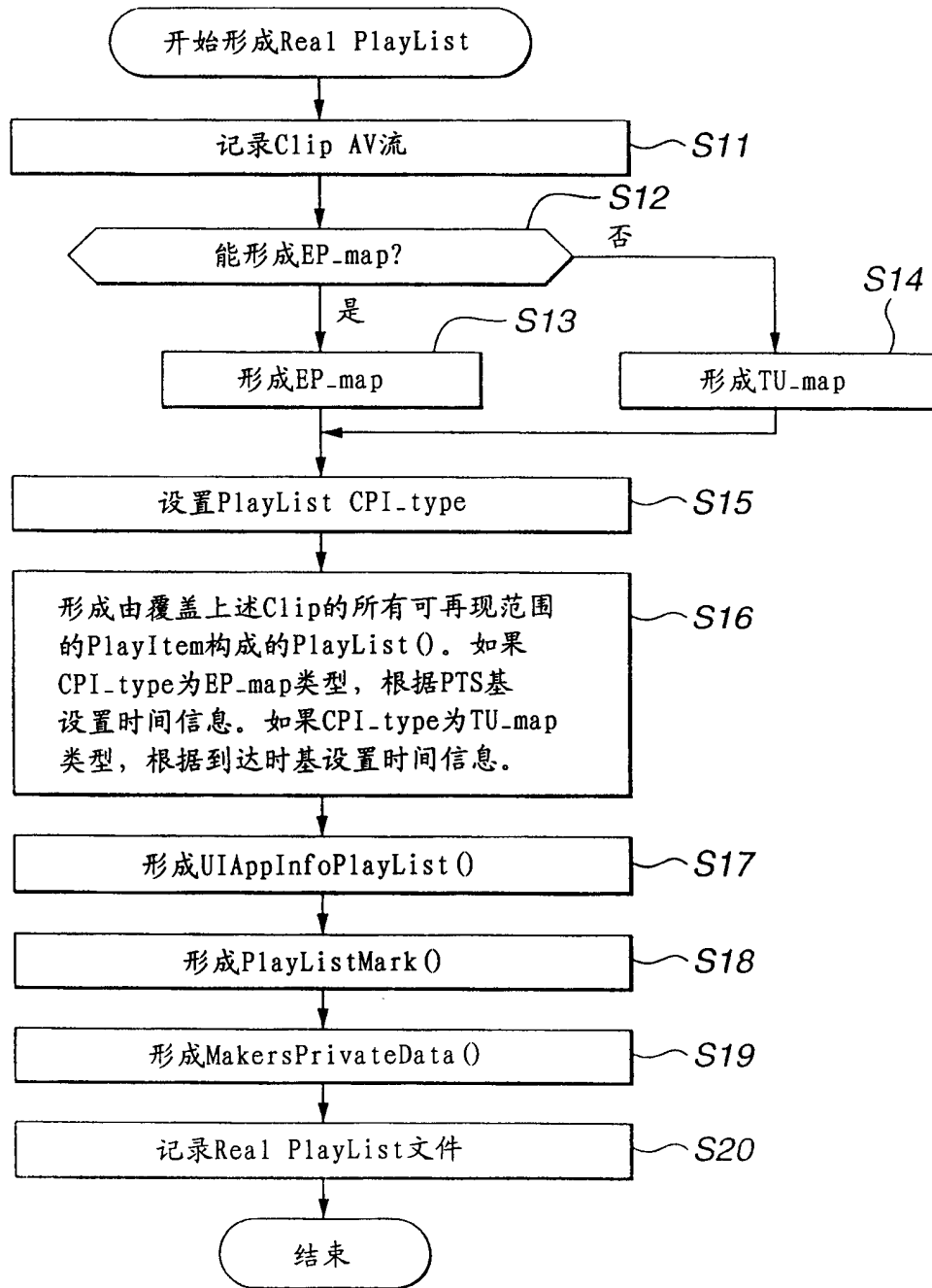


图 102

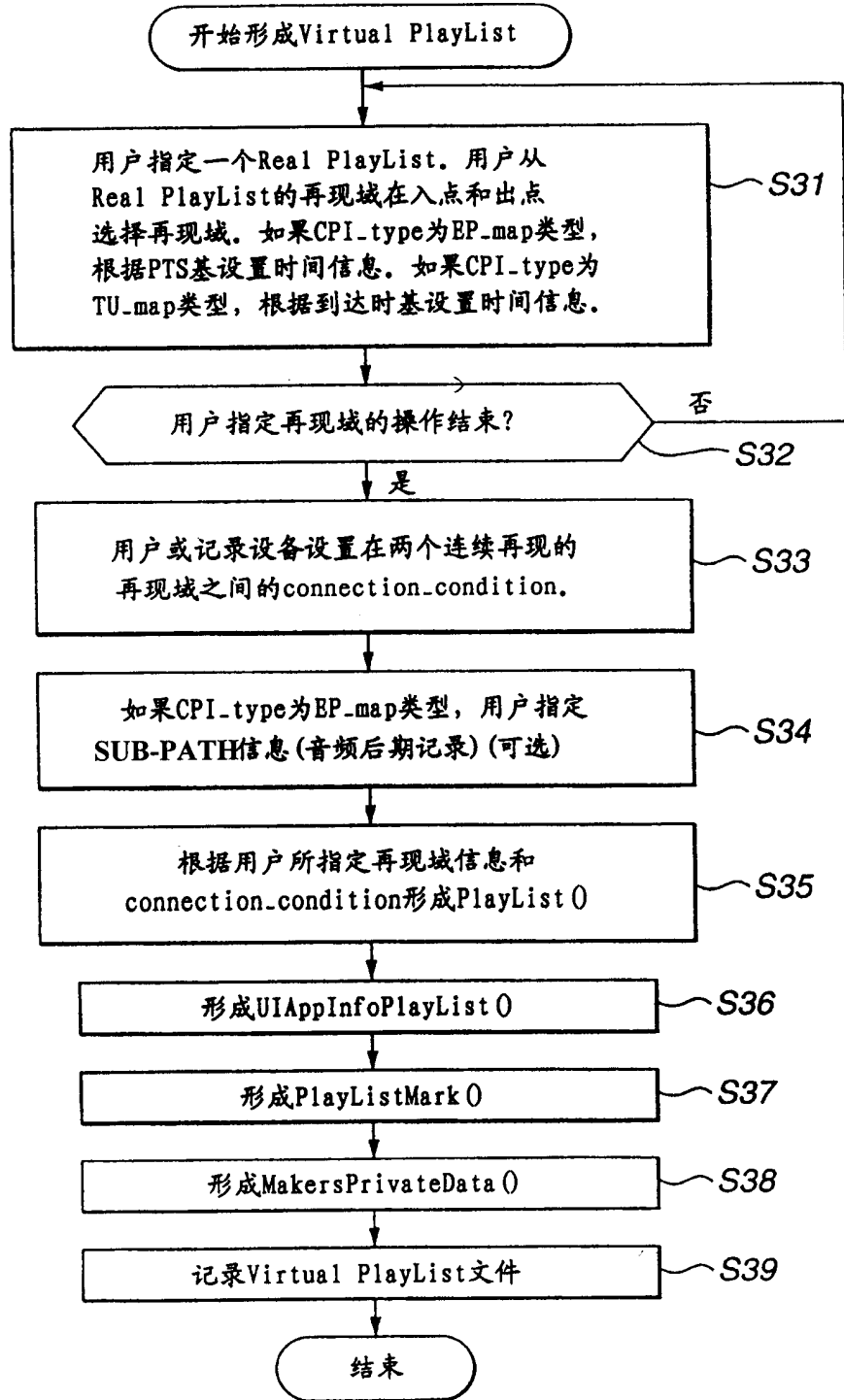


图 103

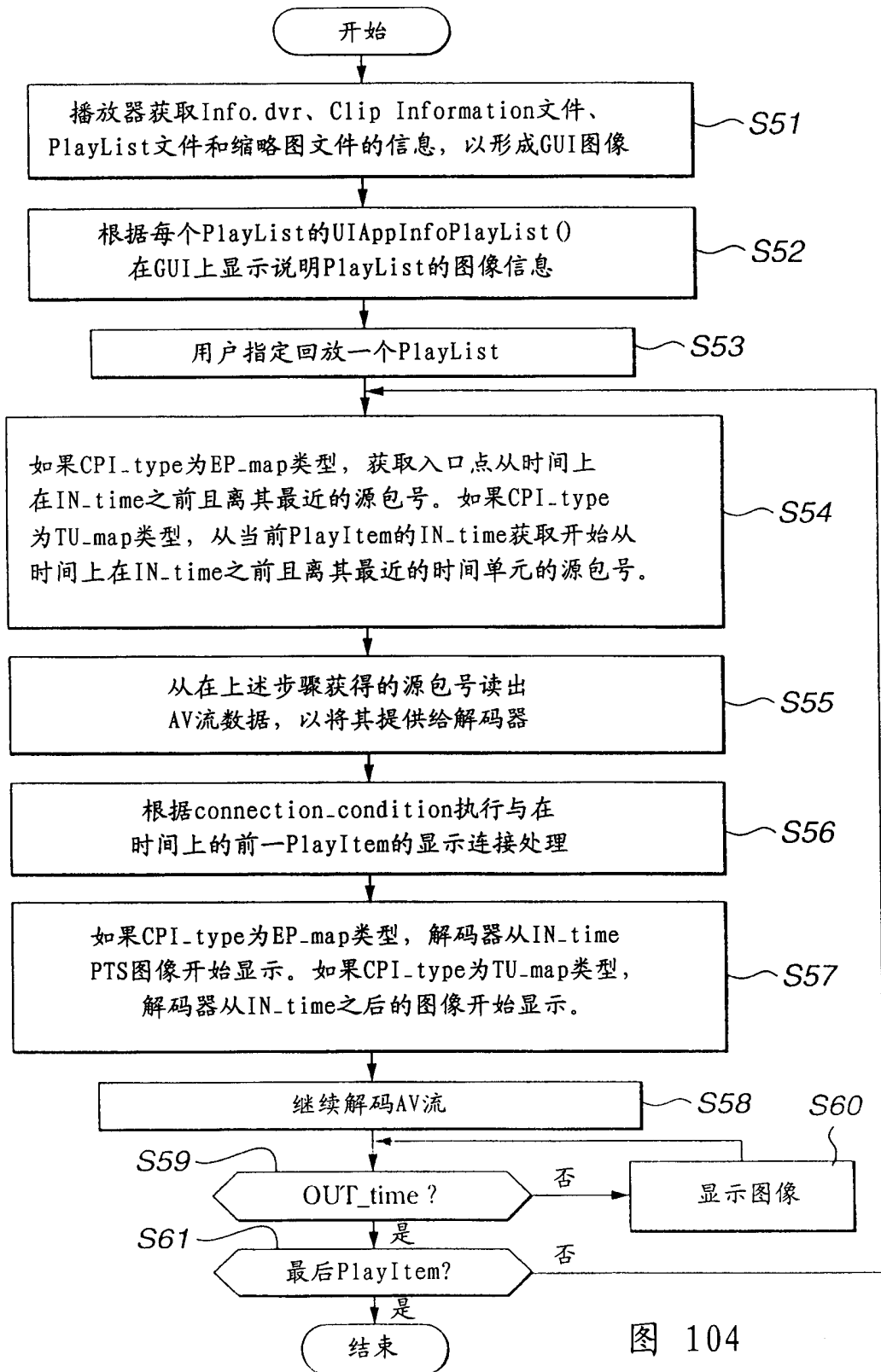


图 104

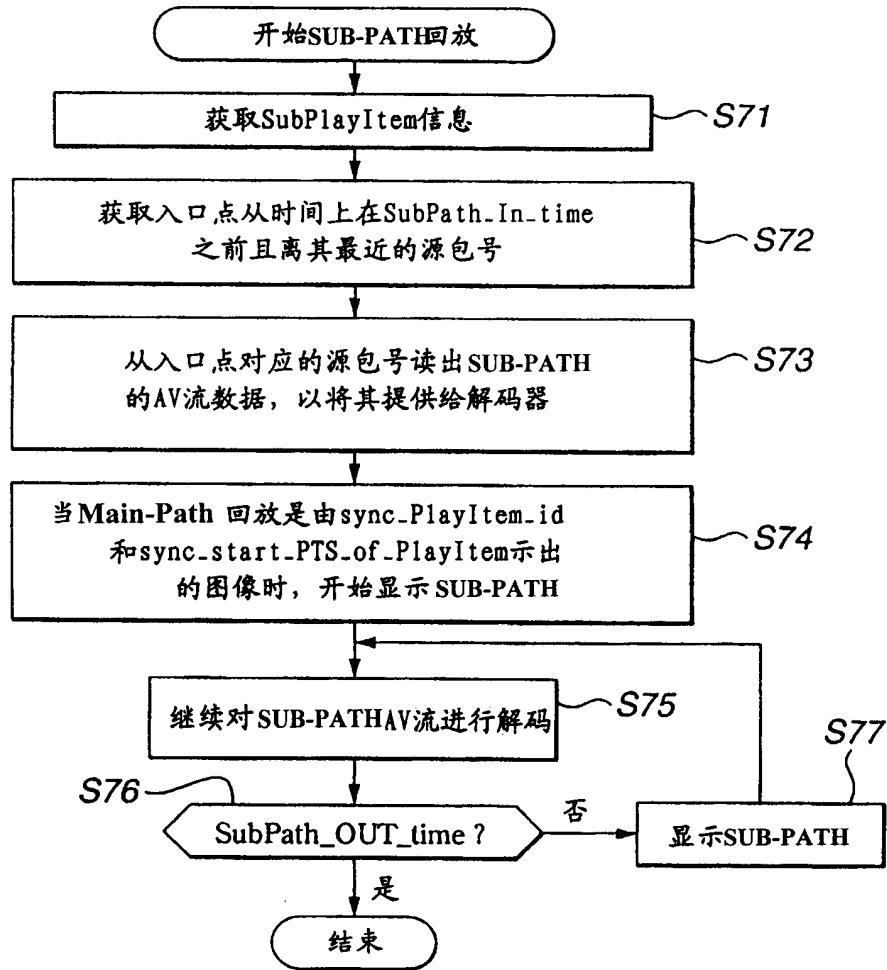


图 105

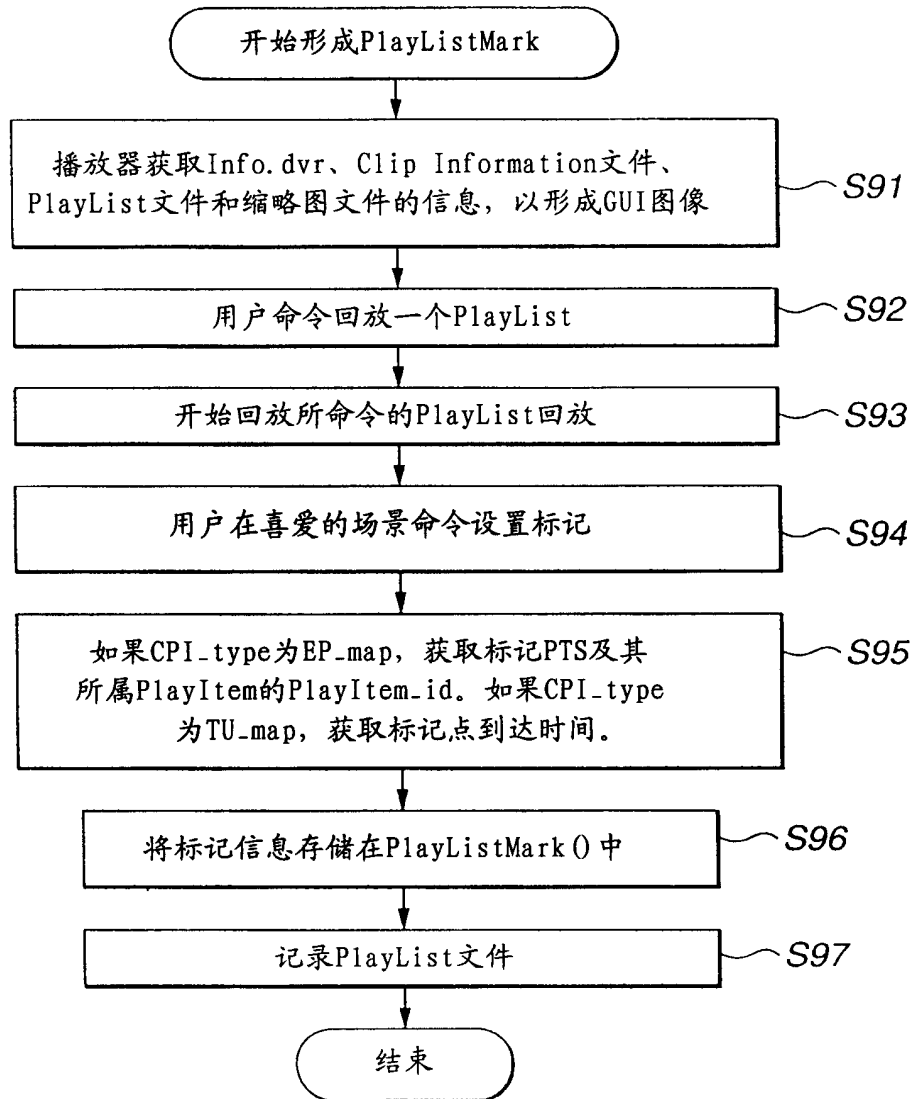


图 106

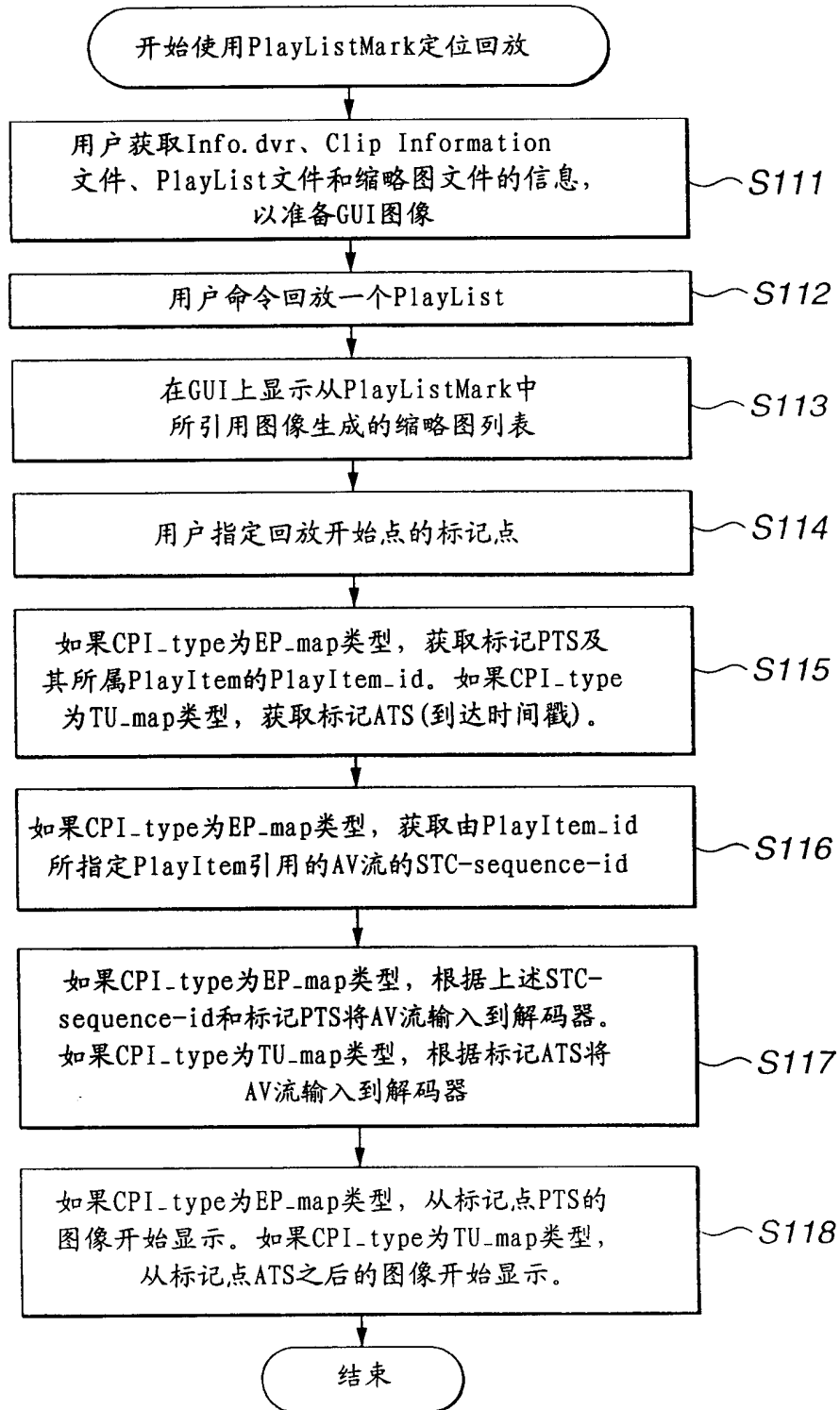


图 107

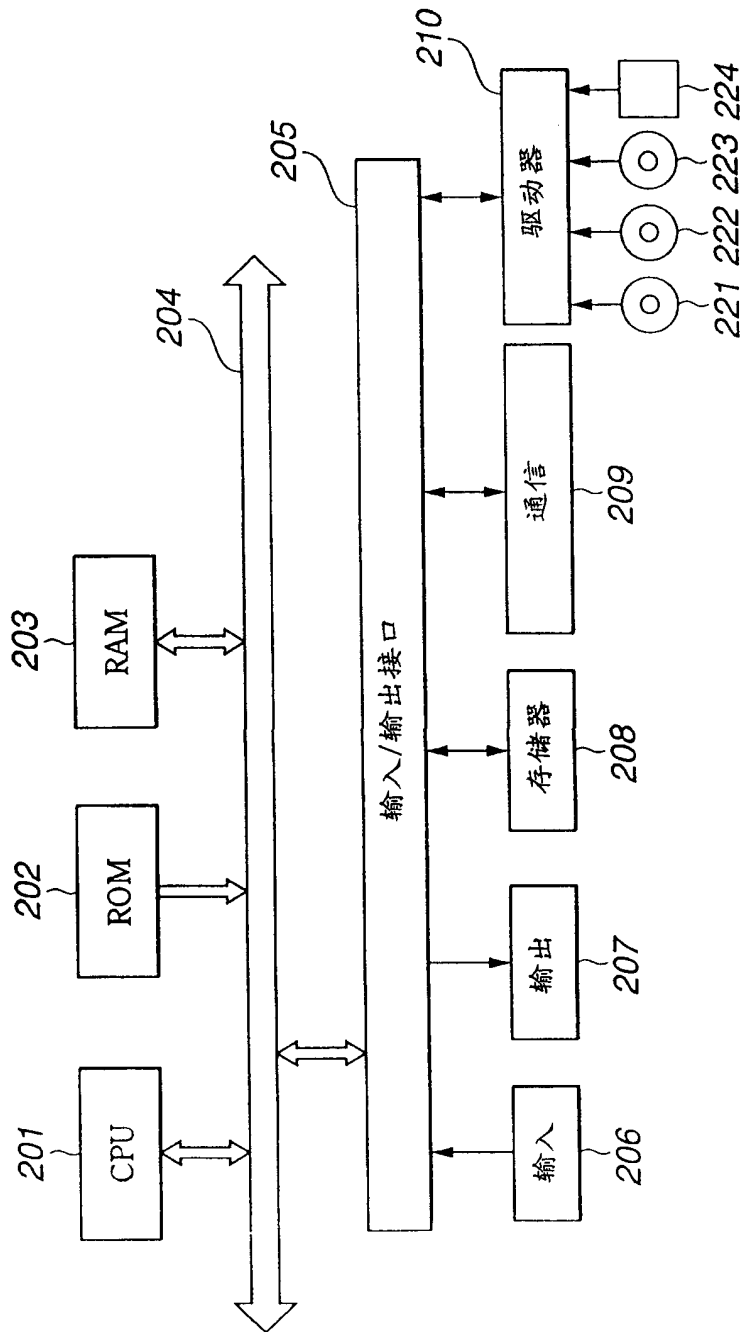
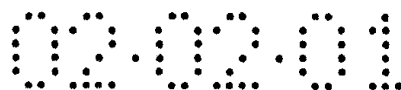


图 108

权 利 要 求 书
按照条约第 19 条的修改

1. 一种用于在记录介质上记录 AV 流数据的信息处理装置，包括：
控制器，用于生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的
5 的地址之间对应关系的第一表，或描述来自输送包到达时间点的到达时间戳
和对应输送包的所述 AV 流数据中地址之间对应关系的第二表；
记录器，用于在所述记录介质上与所述 AV 流数据一起记录根据记录方
法选择的第一表和第二表中的一个。
2. 如权利要求 1 所述的信息处理装置，其中，所述第一表为 EP_map；
10 并且其中所述第二表为 TU_map。
3. 如权利要求 1 所述的信息处理装置，其中，所述控制器在非认知记
录的情况下选择所述第二表。
4. 如权利要求 1 所述的信息处理装置，其中，所述控制器在自编码记录
的情况下选择所述第一表。
- 15 5. 如权利要求 1 所述的信息处理装置，其中，所述控制器在认知记录
的情况下选择所述第一表。
6. 如权利要求 1 所述的信息处理装置，其中，所述控制器生成表示所
述第一和第二表中的哪一个已被记录的标志信息；所述记录器存储所述标志
信息。
- 20 7. 如权利要求 6 所述的信息处理装置，其中，所述控制器管理控制，
从而如果所述第一表与所述 AV 流数据一起进行记录，所述再现指定信息根
据呈现时间表示所述 AV 流数据再现范围的时间信息；并且如果所述第二表
与所述 AV 流数据一起进行记录，所述再现指定信息根据到达时间表示所述
AV 流数据再现范围的时间信息。
- 25 8. 一种用于在记录介质上记录 AV 流数据的信息处理方法，包括：
生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的
地址之间对应关系的第一表，或描述来自输送包到达时间点的到达时间戳和
对应输送包的所述 AV 流数据中地址之间对应关系的第二表；
记录步骤，在所述记录介质上与所述 AV 流数据一起记录根据记录方法
30 选择的第一表和第二表中的一个。
9. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质



上记录 AV 流数据的信息处理装置，所述程序包括：

生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系的第一表，或描述来自输送包到达时间点的到达时间戳和对应输送包的所述 AV 流数据中地址之间对应关系的第二表；

- 5 记录步骤，在所述记录介质上与所述 AV 流数据一起记录根据记录方法选择的第一表和第二表中的一个。

10. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：

- 10 生成步骤，生成描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系的第一表，或描述来自输送包到达时间点的到达时间戳和对应输送包的所述 AV 流数据中地址之间对应关系的第二表；

记录步骤，在所述记录介质上与所述 AV 流数据一起记录根据记录方法选择的第一表和第二表中的一个。

11. 一种用于从记录介质再现 AV 流数据的信息处理装置，包括：

- 15 再现单元，用于从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自输送包到达时间点的到达时间戳和对应输送包的所述 AV 流数据中地址之间对应关系；和

- 20 控制单元，用于根据所再现的表控制所述 AV 流数据的输出。

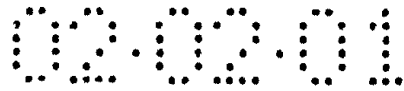
12. 一种用于从记录介质再现 AV 流数据的信息处理方法，包括：

- 25 再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自输送包到达时间点的到达时间戳和对应输送包的所述 AV 流数据中地址之间对应关系；和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

13. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质上记录 AV 流数据的信息处理装置，所述程序包括：

- 30 再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元



的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自输送包到达时间点的到达时间戳和对应输送包的所述 AV 流数据中地址之间对应关系；和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

5 14. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：

再现步骤，从其中根据记录方法记录有第一表或第二表的记录介质再现所述第一表和第二表中的一个，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，所述第二表描述来自传输包到达
10 时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系；和

控制步骤，根据所再现的表控制所述 AV 流数据的输出。

15 15. 一种其中根据记录方法记录有第一表和第二个的记录介质，所述第一表描述呈现时间戳和对应访问单元的所述 AV 流数据中的地址之间对应关系，第二表描述来自传输包到达时间点的到达时间戳和对应传输包的所述 AV 流数据中地址之间对应关系。

16. 一种用于在记录介质上记录 AV 流数据的信息处理装置，包括：

生成装置，用于生成由表示主再现路径的第一信息和表示次再现路径的第二信息构成的再现指定信息；和

20 记录装置，用于在所述记录介质上记录所述 AV 流数据和再现指定信息。

17. 如权利要求 16 所述的信息处理装置，其中，所述主再现路径为用于音频数据后期记录的路径。

25 18. 如权利要求 16 所述的信息处理装置，其中，所述第一信息为 main_path 并且所述第二信息为 Sub-path。

19. 如权利要求 16 所述的信息处理装置，其中，所述第二信息包括表示所述次再现信息类型的类型信息；

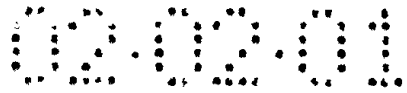
用所述次再现路径引用的所述 AV 流文件名；

所述次再现路径的所述 AV 流的入点和出点；和

30 在所述主路径的时间轴上所述再现路径的入点开始同步的所述主路径时间。



20. 一种信息处理方法，用于在记录介质上记录 AV 流数据的信息处理装置，包括：
- 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
- 5 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
21. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质上记录 AV 流数据的信息处理装置，所述程序包括：
- 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径
- 10 同步再现的次再现路径的第二信息构成的再现指定信息；和
- 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
22. 一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：
- 15 生成步骤，生成由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
- 记录步骤，在所述记录介质上记录所述 AV 流数据和所述再现指定信息。
23. 一种用于从记录介质再现 AV 流数据的信息处理装置，包括：
- 20 再现装置，用于从所述记录介质再现由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
- 控制装置，用于根据被再现的再现指定信息控制所述 AV 流数据的输出。
- 25 24. 一种用于从记录介质再现 AV 流数据的信息处理方法，包括：
- 再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和
- 控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。
25. 一种其中记录有一个计算机可读程序的记录介质，用于在记录介质
- 30 上记录 AV 流数据的信息处理装置，所述程序包括：
- 再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与



所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。

26.一种用于让控制信息处理装置在记录介质上记录 AV 流数据的计算机执行如下步骤的程序：

- 5 再现步骤，从所述记录介质再现由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再现路径的第二信息构成的再现指定信息；和控制步骤，根据被再现的再现指定信息控制所述 AV 流数据的输出。

27.一种其中记录有 AV 流数据的记录介质，其中，所述记录介质其中记录有由表示主再现路径的第一信息和表示与所述主再现路径同步再现的次再

- 10 再现路径的第二信息构成的再现指定信息。