

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷
G11B 27/031
G11B 20/12



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 00104859.7

[45] 授权公告日 2004 年 8 月 25 日

[11] 授权公告号 CN 1163899C

[22] 申请日 2000.2.12 [21] 申请号 00104859.7
[30] 优先权
[32] 1999.2.12 [33] JP [31] 35000/1999
[71] 专利权人 索尼公司
地址 日本东京都
[72] 发明人 饭田健一
审查员 翁晓君

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 王忠忠

权利要求书 3 页 说明书 34 页 附图 29 页

[54] 发明名称 编辑装置和编辑方法
[57] 摘要

一种编辑装置将一个数据文件分割成分割位置之前和之后的第一文件和第二文件。首先确定所述分割位置。然后,包含所述分割位置的记录数据块被拷贝到所述第一文件的一个空白区域中。第一文件的管理信息被编辑。然后,建立和编辑用于第二文件的管理信息。

应用处理
文件管理处理
逻辑地址管理
物理地址管理
闪速存储器访问

I S S N 1 0 0 8 - 4 2 7 4

- 1.一种用于分割包括主数据和被添加到所述主数据上的第一管理数据的数据文件的编辑装置，所述主数据是通过互联固定长度的记录数据块形成的，所述
- 5 述第一管理数据是为管理所述管理用的主数据的记录位置和将不被重现的无效数据的位置而提供的，所述编辑装置包括：

操作装置，用于规定一个分割位置，在该位置处，一个预定的固定长度记录数据块包括所述主数据，并用于根据所述分割位置将所述主数据分割成第一数据文件和第二数据文件；

- 10 编辑装置，用于对所述第一管理数据进行编辑，以便使包括所述主数据的预定的固定长度记录数据块的一部分无效；和

产生装置，用于产生第二管理数据，以便使包括所述主数据的预定的固定长度记录数据块的另一部分无效，并用于将所述第二管理数据加到所述第二数据文件上；

- 15 其中，所述预定的固定长度的记录数据块的所述一部分包括在第一数据文件中，所述预定的固定长度的记录数据块的所述另一部分包括在第二数据文件中。

2.根据权利要求1所述的编辑装置，其特征是所述操作装置规定一个记录数据块，在该记录数据块中，所述主数据将被分割。

- 20 3.根据权利要求2所述的编辑装置，其特征是由所述第一管理数据管理的无效数据包括至少在由所述操作装置规定的分割位置处分割的所述记录数据块的第二部分的主数据，和由所述第二管理数据管理的无效数据包括至少在所述分割位置处分割的记录数据块的第一部分的主数据。

4.根据权利要求1所述的编辑装置，其特征是还包括：

- 25 拷贝装置，用于将包含由所述操作装置规定的分割位置的记录数据块拷贝到所述数据文件的空白区域中；和

组合编辑装置，用于将所述记录数据块的拷贝逻辑地组合成第二数据文件，

其中，所拷贝的记录数据块包括在所述第一数据文件和所述第二数据文件中。

- 30 5.根据权利要求4所述的编辑装置，其特征是消隐数据被记录在其中记录了

无效数据的所拷贝的记录数据块的第一和第二部分中。

6.根据权利要求1所述的编辑装置,其特征是还包括一个管理数据拷贝装置,用于拷贝所述第一管理数据,以便形成管理数据,其中,所形成的管理数据被编辑以便形成第二管理数据。

5 7.根据权利要求1所述的编辑装置,其特征是在所述数据文件被分割后彼此相邻的任意多项无效数据被组合成单项无效数据。

8.根据权利要求1所述的编辑装置,其特征是还包括一个加载/卸载装置,用于可拆卸地加载用于存储所述数据文件的存储介质,其中,在所述存储介质中建立所述第一数据文件和第二数据文件。

10 9.根据权利要求8所述的编辑装置,其特征是所述存储介质包括非易失存储器。

10.根据权利要求1所述的编辑装置,其特征是所述主数据包括音频数据。

11.根据权利要求1所述的编辑装置,其特征是所述主数据包括运动图象数据。

15 12.一种用于分割包括主数据和与所述主数据相关的第一管理数据的数据文件的编辑方法,所述主数据是通过互联一个或多个固定长度的记录数据块形成的,所述第一管理数据用于管理所述主数据的记录位置和将不被重现的无效数据的位置,所述编辑方法包括:

规定步骤,用于规定一个分割位置,在该位置处,一个预定的固定长度记录数据块包括所述主数据,并用于根据所述分割位置将所述主数据分割成第一和第二数据文件;

编辑步骤,用于对所述第一管理数据进行编辑,以便使包括所述主数据的预定的固定长度记录数据块的一部分无效;和

25 产生步骤,用于产生第二管理数据,以便使包括所述主数据的预定的固定长度记录数据块的另一部分无效,并用于将所述第二管理数据加到所述第二数据文件上。

13.根据权利要求12所述的编辑方法,其特征是在所述规定步骤中规定包含所述分割位置的一个记录数据块,在该记录数据块中记录了所述主数据。

14.根据权利要求13所述的编辑方法,其特征是还包括:

30 拷贝步骤,用于将包含由所述规定步骤规定的分割位置的记录数据块拷贝

到所述数据文件的空白区域中；和

组合编辑步骤，用于将记录数据块的拷贝组合成第二数据文件，

其中，所拷贝的记录数据块包括在所述第一数据文件和所述第二数据文件中。

- 5 15.根据权利要求 14 所述的编辑方法，其特征是消隐数据被记录在所述记录数据块的第一和第二部分中，在所述记录数据块中记录了无效数据。

16.根据权利要求 14 所述的编辑方法，其特征是还包括一个管理数据拷贝步骤，用于在编辑第一管理数据之前将所述第一管理数据拷贝到空白区域中，其中，所拷贝的第一管理数据被编辑。

- 10 17.根据权利要求 12 所述的编辑方法，其特征是在所述数据文件被分割之后彼此相邻的任意多项无效数据被组合成单项无效数据。

18.根据权利要求 12 所述的编辑方法，其特征是分割存储在可拆卸存储介质中的所述数据文件。

19.根据权利要求 12 所述的编辑方法，其特征是所述主数据包括音频数据。

- 15 20.根据权利要求 12 所述的编辑方法，其特征是所述主数据包括运动图象数据。

编辑装置和编辑方法

5 技术领域

本发明一般涉及编辑装置和编辑方法。特别是本发明涉及一种用于分割包括通过互联一个或多个固定长度记录数据块所形成的主数据和用于管理记录位置以及将不被重现的无效数据位置的管理数据的数据文件的编辑装置。本发明还涉及在上述类型编辑装置中使用的一种编辑方法和记录介质。

10 背景技术

已经开发了下述类型的存储系统。诸如闪速存储器之类的具有内置固态存储装置的小存储介质被形成并装载到特别用于上述类型存储介质的驱动系统中。另外，上述存储介质被装载到被集成为一个音频/视频机构或信息装置的一个驱动系统中。由此，计算机数据、图象数据和声音数据可以被存储在所述存储介
15 质中。

在前述使用固态存储装置的存储系统中，对存储在所述存储系统中的文件需要执行一个编辑操作。在所述存储介质中编辑一个文件还需要使诸如数据的运动、拷贝和重写等有效编辑操作最少，并使编辑操作所需的处理时间和功率损耗减少到最小值。

20 发明内容

因此，本发明的一个目的是提供一种能够更加有效地执行编辑操作的编辑装置。

为了实现上述目的，根据本发明的一个方面，提供了一种用于分割包括主数据和被添加到所述主数据上的第一管理数据的数据文件的编辑装置，所述主数据是
25 通过互联固定长度的记录数据块形成的，所述第一管理数据是为管理所述管理用的主数据的记录位置和将不被重现的无效数据的位置而提供的，所述编辑装置包括：操作装置，用于规定一个分割位置，在该位置处，一个预定的固定长度记录数据块包括所述主数据，并用于根据所述分割位置将所述主数据分割成第一数据文件和第二数据文件；编辑装置，用于对所述第一管理数据进行编辑，以便使包
30 括所述主数据的预定的固定长度记录数据块的一部分无效；和产生装置，用于产生第

二管理数据，以便使包括所述主数据的预定的固定长度记录数据块的另一部分无效，并用于将所述第二管理数据加到所述第二数据文件上；其中，所述预定的固定长度的记录数据块的所述一部分包括在第一数据文件中，所述预定的固定长度的记录数据块的所述另一部分包括在第二数据文件中。

5 根据本发明的另一个方面，提供了一种用于分割包括主数据和与所述主数据相关的第一管理数据的一个数据文件的编辑方法，所述主数据是通过互联一个或多个固定长度记录数据块形成的，所述第一管理数据用于管理所述主数据的记录位置和不被重现的无效数据的位置。所述编辑方法包括：规定步骤，用于规定在其位置处一个预定的固定长度记录数据块包括所述主数据的分割位置并用于根据所述分割位置将所述主数据分割为第一和第二数据文件；编辑步骤，用于对所述一
10 管理数据进行编辑以便使包括所述主数据的所述预定的固定长度记录数据块的第一部分无效；产生步骤，用于产生第二管理数据以便使包括所述主数据的所述预定的固定长度记录数据块的另一部分无效，并用于将所述第二管理数据加到所述第二数据文件上。

15 根据本发明的再一个方面，提供了一种记录介质，用于管理至少一个文件和用于以数据长度为单位读出或写入所述文件。所述记录介质包括一个数据记录区域，用于记录至少一个由至少一项主数据、至少一项用于保留一个记录区域的空间数据和用于在一个存储介质上管理所述主数据和空间数据的记录位置的第一管理数据形成的一个数据文件。管理数据记录区域记录用于管理记录在所述数据记录区域中的所述数据文件的第二管理数据。
20

附图说明

图 1A 示出了根据本发明一个实施例的条形存储器外部结构的前视图；

图 1B 示出了根据本发明一个实施例的条形存储器外部结构的侧视图；

图 1C 示出了根据本发明一个实施例的条形存储器外部结构的平面视图；

25 图 1D 示出了根据本发明一个实施例的条形存储器外部结构的仰视图；

图 2 示出了在本发明的一个实施例中使用的文件系统处理分级结构；

图 3 示出了根据本发明的一个实施例所述条形存储器的物理数据结构；

图 4 示出了根据本发明的一个实施例所述条形存储器的管理标记的内容；

图 5 示出了根据本发明的一个实施例在所述条形存储器的数据更新之前和之后所述物理地址和逻辑地址的概念；
30

图 6 示出了在本发明一个实施例中使用的的一个逻辑-物理(逻辑/物理)地址变换表的管理格式;

图 7 示出了在本发明一个实施例中使用的的所述逻辑/物理地址变换表的结构;

5 图 8 示出了本发明一个实施例中在闪速存储器容量、数据块的数量、每个数据块的容量、每页的容量、所述条形存储器逻辑/物理地址变换表的尺寸之间的关系;

图 9 的框图示出了根据本发明一个实施例的驱动系统;

图 10 示出了一个文件分配表(FAT)的结构;

10 图 11 示出了所述 FAT 上多个群集的连接结构;

图 12 示出了目录结构;

图 13 示出了子目录和文件存储内容;

图 14 示出了根据本发明一个实施例所述条形存储器的目录结构;

图 15 示出了根据本发明的一个实施例的消息列表文件;

15 图 16 示出了根据本发明一个实施例的消息列表文件的标题;

图 17 示出了根据本发明一个实施例所述消息列表文件的文件夹入口;

图 18 示出了根据本发明一个实施例所述消息列表文件的消息入口;

图 19 示出了根据本发明一个实施例所述消息列表文件的数据结构;

图 20 示出了根据本发明一个实施例所述消息列表文件的格式帧;

20 图 21A 示出了根据本发明一个实施例所述消息列表文件的帧结构的一个例子;

图 21B 示出了图 21A 所示消息数据文件的 TOC 帧的内容;

图 22A 示出了根据本发明一个实施例所述消息数据文件的帧结构的另一个例子;

25 图 22B 示出了图 22A 所示消息数据文件的 TOC 帧的内容;

图 23 示出了当一个消息数据文件是新建立时的空白帧;

图 24 示出了利用一个空白帧替换一个重现保护区的情况;

图 25 的流程示出了根据本发明一个实施例的分割操作;

图 26A 示出了在执行分割操作之前的一个文件 F1;

30 图 26B 示出了在执行分割操作之后的一个第一文件 F1-1;

图 26C 示出了在执行分割操作之后的一个第二文件 F1-2;

图 27 示出了在执行分割操作之前的 FAT;

图 28 示出了在执行分割操作之后的 FAT;

图 29 示出了在执行分割操作之前的 FAT 结构;

5 图 30 示出了在执行分割操作之后的 FAT 结构;

图 31 示出了根据本发明一个实施例所述 TOC 帧的数据结构;

图 32 示出了根据本发明一个实施例所述数据存储单元和数据存储器内容的一个例子;

10 图 33 示出了根据本发明一个实施例所述数据存储单元和数据存储器内容的另一个例子;

图 34 示出了根据本发明的一个实施例所述数据存储单元和数据存储器内容的再一个例子。

具体实施方式

15 下面参考附图描述本发明的一个实施例。在这个实施例中,一个具有条形外部结构的条形存储器被用做本发明的记录介质。

从用于向/从所述条形存储器记录/重现一个文件的角度来描述本发明的编辑装置,和通过在上述驱动系统中使用的编辑方法实施本发明的编辑方法。

1 条形存储器的外部结构

20 首先参看图 1A 到 1D 来描述用做本发明记录介质的条形存储器 1 的外部结构。

所述条形存储器 1 包括例如在诸如图 1A 到 1D 中的一个所示条形壳体內的具有预定容量的一个存储装置。在这个实施例中,一个闪速存储器被用做所述存储装置。

25 图 1A 到 1D 所示的壳体例如由塑料膜形成,并且尺寸分别为 60mm(W11)、20mm(W12)、和 2.8mm(W13),如图 1A 和 1C 所示。

从前侧下面部分到所述壳体的底表面形成了具有 9 个电极的终端 2,如图 1A 和 1D 所示。经过终端 2 在内置存储器装置上执行读出/写入操作。

30 如从图 1C 看到的,在所述壳体的左上部分形成了一个凹槽 3。由于提供了凹槽 3,条形存储器 1 能够避免被错误地插入到例如所述驱动系统的装载/非装载机构中。在所述壳体的底表面上形成了一个防止滑动的缺口 4,以改善操作的方便

性。在所述底表面上还提供了—个用于防止数据被意外删除的滑动开关5。

2 条形存储器的格式

2.1 存储器文件系统处理分级结构

用做所述记录介质的条形存储器1的系统格式如下。

- 5 图2示出了条形存储器1的文件系统处理分级结构。如图2所示,所述文件系统处理分级结构是由一个应用处理层、一个文件管理处理层、一个逻辑地址管理层、一个物理地址管理层和一个闪速存储器存取按照降序形成的。在这个分级结构中,所述文件管理处理层被用做所谓的“文件分配表(FAT)”。图2示出了在这个实施例中的所述文件系统引入了所述逻辑地址和物理地址的概念,这将在
- 10 后面详细描述。

2.2 物理数据结构

图3示出了被用做所述条形存储器1中的存储器装置的闪速存储器的物理数据结构。

- 在数据单元中主要确定具有固定长度并被称之为“段”的闪速存储器的存储
- 15 区域。规定每个段以使其具有4兆字节(MB)或8兆字节的尺寸,在单个闪速存储器内所述段的数量是根据所述闪速存储器的容量而变化的。

- 每个段被分成具有8千字节(KB)或16KB长度并被称之为“块”的多个固定长度数据单元,如图3所示。每个段基本上被分成512个块,由此,图3的A中所表示的块的数量是511。但是,由于预定数量的块被允许用于闪速存储器中的
- 20 写禁止故障区域,所以,写允许块的实际数量可能少于511。

在如图3的A所示安排的块0到n中,两个引导块、即块0和块1被称之为“引导块”。但是实际上,从有效块开始的两个块被规定为是引导块,块0和块1不总是引导块。剩下的块被用做存储用户数据的用户块。

- 如图3的D指出的,每个块被分成页0到m,如图3的E指出的,每个页由
- 25 512个字节数据区域和16个字节冗余区域形成,其总容量为固定长度的528个字节。下面将参照附图3的F详细描述所述冗余单元的结构。每个块内页的数量当所述块容量是8KB时是16,当所述块容量是16KB时是32。

- 与图3的D和E指出的相同页结构被用于引导块和用户块。在闪速存储器中,以页为单位执行数据的读出/写入,而以块为单位执行数据的擦除。但是,由于数
- 30 据仅仅被写入数据被擦除的空白页,所以,实际上是以块为单位执行数据的重写

和写入。

在引导块(块0)中,如图3的B所指出的,标题被存储在页0中,指出最初故障数据位置的地址信息被存储在页1中,和CIS/IDI信息被存储在页2中。第二引导块(块1)被用做备份引导块,如图3的C所示。

5 在图3的E中指出的冗余区域的结构如图3的F所示。在所述冗余区域内,前3个字节、即字节号0到字节号2被用做重写区域,根据所述数据区域数据内容的更新,所述重写区域能够被重写。在所述重写区域内,块状态被存储在字节号0中,块标记数据被存储在字节号1内作为所述数据的状态,所述页数据状态被存储在字节号2的预定上部位内作为一个转换表标记。

10 原则上,字节号3到字节号15根据当前页的内容存储固定数据,即它们存储不能被重写的信息。管理标记(Block info)被存储在字节号3中,逻辑地址(Logic Address)(将在后面讨论)被存储在随后的两个字节、即字节号4和5中。再后面的5个字节、即字节号6到字节号10被用做格式保留区域,和再后面的两个字节、即字节号11和字节号12存储用于校正所述格式保留区域误差的分布信息误差校正码(ECC)。剩下的3个字节、即字节号13到字节号15存储用于校正存储在图3
15 的E所指出的数据区域中的数据的误差的数据ECC。

存储在图3的F指出的所述冗余区域的字节号3中的管理标记的内容在位号7到位号0中被规定,如图4所示。

20 位号7、6、1和0是保留(未规定)区域。位号5存储指出对当前块的访问有效(‘1’,空白)或无效(‘0’,读保护)的一个标记。位号4存储指出拷贝保护有效(‘1’,OK)或无效(‘0’,不好(NG))的一个标记。

位号3存储是用于指出所述当前块是否存储将在后面详细描述的逻辑-物理(逻辑/物理)地址转换表的一个识别符的一个转换标记。如果位号3的值是‘0’,不能所述当前块被用于存储一个逻辑/物理地址转换表。如果位号3的值是‘1’,
25 那么,当前块不存储逻辑/物理地址转换表。

位号2存储一个系统标记。如果所述系统标记是‘1’,则当前块被确定为用户块。如果所述系统标记是‘0’,那么,所述当前块是引导块。

下面结合图8描述所述段和块与闪速存储器容量之间的关系。条形存储器1
30 的闪速存储器容量被规定为是4MB、8MB、16MB、32MB、64MB和128MB中的任何一个。

当闪速存储器具有最小容量 4MB 时, 一个块的容量被规定为是 8KB, 块的数量是 512, 即所述闪速存储器的容量、即 4MB 等于一个段的容量。当所述闪存存储器的容量是 8MB 时, 一个块的容量被规定为是 8KB, 两个段中的块的数量是 1024(每个段是 512)。如上所述, 当一个段具有 8KB 时, 一个块中的页数

5 数量是 16。

关于具有 16MB 容量的闪速存储器, 一个块的容量可以是 8KB 或 16KB。因此, 所述闪速存储器被划分为两种类型, 即一种类型具有 4 个段和 2048 个块(一个块是 8KB), 另一种类型具有 2 个段和 1024 个块(一个块是 16KB)。当一个块是 16KB 时, 一个块中页的数量是 32。

10 至于容量为 32MB、64MB 或 128MB 的闪速存储器, 一个块的容量仅仅被规定为是 16KB。因此, 具有 32MB、64MB 和 128MB 容量的闪速存储器的段/块的数量分别是 4/2048、8/4096 和 16/8192。

2.3 物理地址和逻辑地址的概念

下面结合图 5 的 A 和 B 所示的数据重写操作描述从上述闪速存储器的物理数据

15 结构来看, 在本实施例文件系统中所述物理地址和所述逻辑地址的概念。

每个块都被提供有一个物理地址。所述物理地址是根据在闪速存储器中所述块物理配置的顺序确定的, 一个块和相应物理地址之间的关系是固定的。在图 5 的 A 所示的例子中, 从上到下有 4 个块被提供有诸如 105、106、107 和 108 的物理地址。实际物理地址中的每一个都是由两个字节表示的。

20 现在假设如图 5 的 A 所示, 被提供有物理地址 105 和 106 的块是已经记录的区域, 在这些区域中存储有数据, 被提供有物理地址 107 和 108 的块是未记录区域, 数据已经从这些区域中被擦除。

相反, 所述逻辑地址是一个指定给已经被写入到一个块中的数据的地址, 并且该地址是由所述 FAT 文件系统使用的, 这将在后面讨论。

25 在图 5 的 A 所示的例子中, 从上到下有 4 个块被提供有诸如 102、103、104 和 105 的逻辑地址。所述逻辑地址以及所述物理地址都是由两个字节表示的。

现在假设如图 5 的 A 所示, 存储在地址 105 中的数据是将被重写或部分擦除的数据。在执行这种操作中, 在闪速存储器的文件系统中, 经过更新的数据已经被写入未被使用的块而不是正在被写入同一个块(在这种情况下, 所述块具有物理

30 地址 105)。即如图 5 的 B 所示在物理地址 105 处存储的数据被擦除, 利用物理地

址 107 将经过更新的数据写入到一个未被使用的块中，如处理 1 所示。

然后，如处理 2 所指出的，如图 5 的 A 所示与数据被更新之前的物理地址 105 对应的逻辑地址 102 被重新分配，从而使其与物理地址 107 对应，并被提供给已经被写入了被更新数据的块。因此，与数据更新之前的物理地址 105 相关的逻辑地址 104 被重新分配，从而使它与物理地址 105 相关。

即，所述物理地址是唯一指定给相应块的一个地址，而所述逻辑地址是唯一指定给已经写入到一个块中的数据的地址。

通过如上所述地交换所述块，可以避免集中对相同的存储区域(块)进行访问，借此可以延长具有有限访问(重写)次数的闪速存储器的寿命。

10 通过如处理 2 所示的重新分配所述逻辑地址，即使是所述块在数据更新之后被交换或重新定位，所述 FAT 也能够识别相同的地址，借此可以正确地访问之后的数据。

为了简化在逻辑/物理地址转换表(将在后面详述)上数据更新的管理，所述块交换被限制在一个段内，换言之，在段之间进行所述交换是被禁止的。

15 2.4 逻辑/物理地址转换表

根据参考图 5 所做的描述，所述物理地址和逻辑地址之间的关系是通过执行块交换改变的。因此，为了访问所述闪速存储器以便从/向所述闪速存储器读出/写入数据，需要参考指出所述物理地址和逻辑地址之间的关系的逻辑/物理地址转换表。特别是，所述 FAT 参考所述逻辑/物理地址转换表以规定与由所述 FAT 指定的逻辑地址对应的物理地址，借此使得访问由所规定的物理地址表示的块成为可能。换言之，不用所述逻辑/物理地址转换表就不可能利用所述 FAT 访问所述闪速存储器。

25 通常，当条形存储器 1 被装载到例如一个驱动系统中时，所述驱动系统的微机检查所述条形存储器 1 的存储内容以构成所述逻辑/物理地址转换表和在所述驱动系统的 RAM 中存储所结构的表。即，所述逻辑/物理地址转换表不被存储在所述条形存储器 1 中。

相反，根据这个实施例，所述逻辑/物理地址地址表转换表被存储在所述条形存储器 1 中。

30 图 6 示出了将被存储在这个实施例的条形存储器 1 中的所构成的逻辑/物理地址转换表的内容。在这个实施例中，其中根据以升序排列的相关逻辑地址存储了

两字节物理地址的表信息被结构成所述逻辑/物理地址转换表。

如上所述,所述物理地址和逻辑地址都是由两个字节表示的。这以下述事实为基础,即需要与能够覆盖用于具有 128MB 的最大容量的闪速存储器的所有 8192 个块的位相同数量的位。

- 5 这样,如图 6 借助于举例所示的物理地址和逻辑地址是由两个字节表示的。在这个例子中,所述物理地址和逻辑地址是以十六进制数指出的,即 0x 之后的值是十六进制的。相同的说明用于本说明书的下面描述。但是,为了简单表示起见,在某些附图中,所述 0x 被省略。

- 10 图 7 借助于举例示出了以图 6 所示概念为基础的所述逻辑/物理地址转换表的结构。如图 7 的 A 所指出的,所述逻辑/物理地址转换表被存储在所述闪速存储器最后一个段内的一个块中。

- 15 如图 7 的 A 所指出的,在已经结合图 3 的 A 和 B 讨论过的用于划分一个块的页当中,用于两个页、即页 0 和页 1 的区域被指定给与段 0 相关的逻辑/物理地址转换表。例如,由于如参考图 8 所述具有容量为 4MB 的闪速存储器仅具有一个段,所以,只有页 0 和页 1 被用做用于存储所述逻辑/物理地址转换表的区域。

具有容量 8MB 的闪速存储器具有两个段。因此,页 0 和页 1 被指定给与段 0 相关的逻辑/物理地址转换表,随后的页 2 和页 3 被分配给与段 1 相关的逻辑/物理地址转换表。

- 20 此后,随着闪速存储器容量的增加,每两个页的区域被指定给与每个段相关的逻辑/物理地址转换表。由于用于具有 128MB 的最大容量闪速存储器的段共有 16 个,所以,共有 32 个页被指定给与段 0 到段 15 相关的逻辑/物理地址转换表。因此,在图 7 的 A 中指出的最大页数量是 31。如从前面的描述可以看出的,所述逻辑/物理地址转换表是以段为单位管理的。

- 25 如图 7 的 B 所示出的,两页的数据区域指出用于一个段的逻辑/物理地址转换表的结构。即,由于如图 3 的 E 所示用于一个页的数据区域是 512 个字节,所以,如图 7 的 B 所示,1024(=512 x 2)个字节被扩展。

- 30 如图 7 的 B 所示,具有 1024 个字节的两页数据区域被分成多个两个字节的字段,其中,被划分的两个字节的字段被指定给逻辑地址 0、逻辑地址 1 等,最后两个字节、即从头开始的第 991 个字节和第 992 个字节被规定为用于逻辑地址

495 的一个字段。然后，相关的物理地址被写入到相应的两字节字段中。然后，在在这个实施例所使用的逻辑/物理地址转换表中，当由于与数据更新相关的块交换而更新所述物理地址和逻辑地址之间的关系时，在所述逻辑地址的基础上更新所述物理地址的存储配置。

- 5 剩下的 32 个字节、即第 993 个字节到最后第 1024 个字节被指定给用于存储与附加块相关的物理地址的一个字段。即、管理与 16 个附加块相关的物理地址。所述附加块被称之为“工作块”，设置该附加块用于例如暂时以块为单位保持将被更新的数据。

虽然一个段被分成 512 个块，但是，在图 7 的 B 所示的表结构中，可管理块
10 的数量被设置为与逻辑地址 0 到逻辑地址 495 对应的 496 个块。这是因为设置了上述附加块和一定数量的块被允许用于所述闪速存储器中的无效块(不能使用的区域)。因此，实际上，可以认可数量的无效块可以被包含在所述闪速存储器中。由此，可以以仅仅能够管理 496 个块(被使能写入/擦除的块)的方式设置所述逻辑/物理地址转换表。

- 15 在存储了所述逻辑/物理地址转换表的一个块中，如图 4 所示，在所述管理标记的位号 3 中设置“0”。由此，可以识别对应块存储了所述逻辑/物理地址转换表。

在存储所述逻辑/物理地址转换表的块中，当重写所述逻辑/物理地址转换表的内容时，总是要执行一个交换处理，如图 5 的 A 和 B 所示。因此，存储所述逻辑
20 /物理地址转换表是不固定的，即不能将所述逻辑/物理地址转换表存储在一个专用块中。

然后，所述 FAT 访问所述闪速存储器以搜索在管理标记的位号 3 中设置了‘0’的块，借此识别存储了所述逻辑/物理地址转换表的块。为了便于搜索存储了所述逻辑/物理地址转换表的块，在这个实施例中规定存储所述逻辑/物理地址转换表的
25 块位于最后的一个段中。这使得所述 FAT 能够仅仅搜索与存储所述逻辑物理地址转换表的块相关的最后一个段。换言之，不需要为了所述逻辑/物理地址转换表而搜索所述闪速存储器的所有的段。

例如可以在制造所述条形存储器 1 时存储图 7 所示的逻辑/物理地址转换表。

- 30 再参看图 8，该图示出了所述闪速存储器的容量和所述逻辑/物理地址转换表的

尺寸之间的关系。

参看图 7, 用于管理一个段的逻辑物理地址转换表的尺寸等于与两个页相关的 1024 个字节, 即 1KB。因此, 当闪速存储器具有的容量为 4MB(只有一个段)时, 所述逻辑/物理地址转换表的尺寸是 1KB, 如图 8 的 E 所示。当所述闪速存储器的
5 容量是 8MB(2 个段)时, 所述逻辑/物理地址转换表的尺寸是 2KB(4 页)。

当所述闪速存储器的容量是 16MB 时, 所述逻辑/物理地址转换表的尺寸是用于具有 2048 个块(4 个段)的闪速存储器的 4KB(8 页)和用于具有 1024 个块(2 个段)的闪速存储器的 2KB(4 页)。。

至于容量为 32MB(4 个段)、64MB(8 个段)和 128MB(16 个段)的闪速存储器,
10 所述逻辑/物理地址转换表的尺寸分别是 4KB(8 页)、8KB(16 页)和 16KB(32 页)。

3 驱动系统的结构

这个实施例驱动系统的结构如下。图 9 示出了针对上述条形存储器 1 执行写入、读出和编辑数据的驱动系统的主单元构成。所述驱动系统主单元 100 和条形存储器 1 形成了一个文件存储系统。

15 利用所述驱动主单元 100 可以从/向所述条形存储器 1 读出/写入诸如运动图象数据、静止图象数据、经过话筒记录的消息、从诸如高密度盘(CD, 商标)和小盘(MD, 商标)的记录介质上记录的高质量音频数据(此后称之为“音乐数据”)和开控制数据的各种主数据。

为了便于表示, 在这个实施例中假设所述驱动系统主单元 100 是一个用于记录
20 和重现作为注释机的管理数据。但是, 用于诸如运动图象、静止图象或音乐的数据的输入/输出系统页可以被提供给驱动系统主单元 100, 从而使所述驱动系统主单元 100 被用做与相应数据文件相关的存储系统。

提供了一个用于将所述条形存储器 1 可拆卸地安装到所述驱动系统主单元 100 上的装载/卸载结构 120。在装载到所述装载/卸载机构 120 上的所述条形存储器 1
25 和微机 109 之间的数据通信是经过主接口(I/F)IC101 执行的。

话筒 103 页被提供给所述驱动系统主单元 100, 由话筒 103 收集的声音经过话筒放大器 104 作为声音信号被提供给数字信号处理器(DSP)102。DSP102 将所输入的声音信号转换为数字音频信号并对所述数字音频信号执行诸如编码的预定信号处理, 然后将其作为记录数据提供给控制微机 109。微机 109 通过主接口 IC101
30 将所述数据记录到条形存储器 1 中。

微机109还经过主接口IC101读出记录在条形存储器1上的音频数据或消息数据文件，并向DSP102输出所读出的音频/消息数据。DSP102然后对所提供的数
据执行诸如译码的预定信号处理并向扬声器放大器105输出其最后形式为模拟声
音信号的数据。扬声器放大器105放大所输入的音频数据并将它输出给所述扬声
器106，借此播放音频声音。

微机109还控制一个显示驱动器107以在显示单元108上显示例如用于引导
用户操作的一个菜单或存储在条形存储器1上的文件内容的预定图象。存储在
条形存储器1中的诸如运动图象或静止图象的图象数据可以被读出和显示在所述
显示单元108上。

操作单元112被提供有使用户能够在驱动系统主单元100上执行操作的各种
键。微机109响应在操作单元112上执行的操作接收一个命令并根据所述命令执
行预定的控制处理。操作内容可以是用于存储文件、选择文件、重现文件、分割
文件(后述)或编辑文件的一个指令。

图9所示驱动系统主单元100的结构仅仅是一个例子，该例子并不用于限制本
发明的结构。即，只要能够执行与条形存储器1的数据通信，任何类型的电子装
置都可以被用做驱动系统主单元100。

4 FAT 结构

如在图2所示文件系统处理分级结构中所描述的，利用所述FAT执行文件
管理处理。具体地说，为了利用图9所述驱动系统从/向条形存储器1读出/写
入(重现/记录)数据，所述FAT响应来自文件系统处理分级的应用处理请求参考
文件存储位置和并执行上述逻辑/物理地址转换，借此使它能够访问条形存储器
1。

下面参看图10描述所述FAT的管理结构。在这个实施例中，所述FAT和逻辑/物理地址转换表被存储在条形存储器1中。因此，图10所示的FAT结构是在
所述条形存储器1中被管理的。

如图10所示，所述FAT管理结构是由分割表、自由空间、引导扇区、FAT、
FAT拷贝、根辞典和数据区域形成的。

在所述数据区域中，使用了诸如段(Cluster)2、段3等的单元。所述段
是一个由所述FAT处理的数据单元，例如是一个管理单元。通常，在所述FAT
中，标准尺寸的段是4KB，并且可以是512个字节和32个KB之间中两个的任何

一个。

在这个实施例中，如上所述，一个块是 8KB 或 16KB。在每个块具有 8KB 的条形存储器 1 中，所述段的尺寸是 8KB。在每个块具有 16KB 的条形存储器 1 中，所述 FAT 段的尺寸是 16KB。即，8KB 或 16KB 被用于由所述 FAT 管理使用的数据单元尺寸或用于与所述条形存储器 1 的一个块相同的数据单元的尺寸，即，由
5 所述 FAT 处理的段的尺寸等于相应条形存储器 1 的块的尺寸。因此，为简便起见，现在假设一个块等于一个段。

在图 10 的左侧，所述块的号是由 x 到 $(x+m-1)$ 、 $(x+m)$ 、 $(x+m+1)$ 、 $(x+m+2)$ 指出的，和以这种方式将形成所述 FAT 结构的各种数据存储在与相应的块中。但是实
10 际上，如图 10 的配置所示，数据并不必须存储在物理连续的块中。

参看所述 FAT 结构，在所述分割表中存储了具有 2GB 的最大容量的 FAT 分割的开头地址和结束地址。所述引导扇区指出所述 FAT 是 12 位 FAT 还是 16 位 FAT，并指出所述 FAT 结构。所述 FAT 结构包括该 FAT 的尺寸、所述段的尺寸和每个区域的尺寸。

15 所述 FAT 是一个表示由一个相应文件形成的段的链接结构的表，这将在后面描述。在紧跟在所述 FAT 后面的区域中存储一个 FAT 拷贝。所述引导字典存储文件名、引导段号和各种属性。32 个字节被用于描述每个文件。

在所述 FAT 中，所述 FAT 的入口和所述段之间的关系是一一对应的。在每个段的入口中，描述了被链接的段，即后续的段号。即由于涉及由多个段(块)形成的
20 一个文件，所以，所述引导段号是在所述字典中指出的，第二段号是在所述 FAT 引导段的入口中表示的。第三段号是在第二段号的入口中表示的。利用类似的方式，在所述 FAR 中描述所述的段链接。

在图 11 中简单地示出了所述段链接结构的概念，图 11 所示的号是以分级结构表示的。如果存储了诸如 MAIN.C 和 FUNC.C 的两个文件，那么，在所述字典中
25 描述所述两个文件的例如 002 和 004 的引导段号。

关于文件 MAIN.C，在引导段号 002 的入口中指出后继段号 003，下一个后继段号 006 在段号 003 的入口中描述。如果段号 006 是这个文件 MAIN.C 的最后一个段，则在段号 006 的入口中描述表示所述最后段的 FFF。

因此，文件 MAIN.C 被按照段 002、003 和 006 的顺序存储。即，假设所述段
30 号与所述条形存储器 1 的块号相匹配，则文件 MAIN.C 被存储在所述条形存储器

1 的块 002、003 和 006 中。但是如上所述，由于由所述 FAT 处理的段等于所述逻辑地址，所以，所以不必须与所述物理地址相匹配。

类似的，所述 FAT 示出了文件 FUNC.C 按照段 004 和 005 的顺序被存储，并由图 11 的 B 和 C 指出。

5 在与未使用的块对应的段入口中指出了 000。

在存储在引导辞典中的每个文件的辞典中，不仅描述了在图 11 的 A 和 B 指出的引导段号，还描述了诸如图 12 所示的各种数据。即，利用在括号内指出的一定数量的字节存储文件名、扩展、属性、备用区域、最后更新的时间信息和最后更新的数据信息、引导段号和文件尺寸。

10 等于低于一个辞典的一个层的子辞典被存储在数据区域内而不是图 10 所示的引导辞典中，即，所述子辞典是被作为具有一个辞典结构的文件进行处理的。所述子辞典的尺寸不受限制，在所述子辞典中的入口和在引导辞典中的入口是需要的。

图 13 示出了所述结构的一个例子，在这种结构中，其属性是一个辞典的文件
15 DIR1 被存储在所述引导辞典中，其属性是一个辞典的文件 DIR2 被存储在围歼 DIR1 中，文件 FILE 存在于文件 DIR2 中。

用做一个子辞典的文件 DIR1 的引导段号是在所述引导辞典中指出的，即，利用上述 FAT 将段 X、Y 和 Z 链接到一起。图 13 示出了所述子辞典 DIR1 和 DIR2 被作为文件进行处理的和被集成为所述 FAT 链路。

20 参看图 29 综合上述 FAT 结构。下面参看图 29 的存储器映像图讨论 FAT 管理方法。

在所述存储器映像中，在图 29 示出的存储器映像中从上到下形成了分割表、自由空间、引导扇区、FAT、FAT 备用区域、根辞典、子辞典和数据区域。

图 29 所示的存储器映像是在所述逻辑/物理地址转换表的基础上执行逻辑/物理地址转换之后获得的一个映像。
25

前述引导扇区、FAT、FAT 备用区域、引导辞典、子辞典和数据区域被统称之为“FAT 分割区域”。在上述的分割表中，记录了所述 FAT 分割区域的开头和结束地址。通常，分割表不被提供给由软盘使用的 FAT。

由于在第一磁迹中除了所述分割表以外没有存储数据，所以建立了一个自由空间。
30

后继引导扇区指出所述 FAT 是 12 位 FAT 还是 16 位 FAT，和根据所述 FAT 的类型记录所述 FAT 结构的尺寸、段尺寸和每个区域的尺寸。

所述 FAT 管理存储在所述数据区域中的文件位置。所述文件备份区域是一个拷贝了所述 FAT 的区域。

- 5 引导辞典的一部分用于存储文件名、引导段地址和各种属性，每个文件使用 32 个字节。

子辞典的一部分被用做其属性是一个辞典的文件，在图 29 所示的例子中，诸如 PBLIST.MSV、CAT.MSV、DOG.MSV 和 MAN.MSV 的 4 个文件被存储在所述子辞典中。所述子辞典的一部分被用于管理所述 FAT 中的文件名和所述文件的存储位置。具体地说，在图 29 所示的子辞典中，所述 FAT 地址“5”被存储在
10 上面记录了文件名 CAT.MSV 的存储区中，FAT 地址“10”被存储在上面记录了文件名 DOG.MSV 的存储区中。FAT 地址“110”被存储在上面记录了文件名 MAN.MSV 的存储区中。

- 段 2 和后继段被用做实际上的数据区域，利用自适应差分式编码调制(ADPCM)
15 方法压缩的声音数据被记录在所述数据区域中。

在这个实施例中，文件名 CAT.MSV 下的 ADPCM 压缩的声音数据被记录在段 5 到段 8 中。是所述文件名 DOG.MSV 的第一部分的 ADPCM 压缩的音频数据、即 DOG-1 被记录在段 10 到段 12 中，同时，是所述文件名 DOG.MSV 的第二部分的 ADPCM 压缩的音频数据、即 DOG-2 被记录在段 100 和 101 中。在文件名
20 MAN.MSV 下的 ADPCM 压缩的音频数据被记录在段 110 和 111 中。

图 29 示出了一个例子，在这个例子中，一个文件被分割并不连续地记录在所述段中。由“空闲”指出的数据区域的段是用于记录数据的自由空间。

- 段 200 和后继段被用于管理所述文件名。所述文件 CAT.MSV 被记录在段 200 中，文件 DOG.MSV 被记录在段 201 上，和文件 MAN.MSV 被记录在段 202
25 中。

在段 200 和所述子段中可以重新排列所述文件的顺序。

- 当如上所述结构的条形存储器 1 被首次插入时，所述存储器的标题、即分割表被参考，借此以识别所述 FAT 分割区域的标记开头地址和结束地址。在从所述引导扇区读出部分数据后，存储在所述引导辞典中的数据和所述子辞典被重现。然
30 后，通过在所述子辞典上搜索上面记录了在所述子辞典上记录的重现管理信息、

即 BPLIST.MSV 的存储区, 检查上面记录了 BPLIST.MSV 的存储区的结束地址。在这个实施例中, 由于在上面记录了 BPLIST.MSV 的存储区的结束处指出了所述地址“200”, 所以, 参考被转移到段 200。

在段 200 和所述后继段中, 文件名被管理, 另外, 重现所述文件的顺序被管理。

- 5 在这个实施例中, 文件 CAT.MSV 是第一个将被重现的磁迹, 文件 DOG.MSV 是第二个磁迹, 文件 MAN.MSV 是第三个磁迹。

在完成参考段 200 之后的所有段的基础上, 参考被再次转移到所述子辞典上, 借此, 搜索与文件名 CAT.MSV、DOG.MSV 和 MAN.SMV 匹配的存储区。在图 29 所示的映像表中, 在上面记录了文件名 CAY.MSV 的存储区内指出了地址
10 “5”。在上面记录了文件名 DOG.MSV 的存储区的结束处存储了地址“10”。在上面记录了文件名 MAN.MSV 的存储区的结束处指出了地址“110”。

然后, 在搜索了以地址“5”为基础的入口地址之后, 作为一个入口发现了段地址“6”。参考地址入口“6”, 做为一个入口发现所述段地址“7”, 通过参考入口地址“7”, 发现作为一个入口的段地址“8”。通过进一步参考入口地址
15 “8”, 记录指出结束所述文件的代码“FFF”。

因此, 文件 CAT.MSV 占据所述段 5、6、7 和 8, 并通过参考数据区域的段 5、6、7 和 8 可以访问上面实际记录了名称为 CAT.MSV 的 ADPCM 压缩的数据的区域。

用于搜索不连续记录文件 DOG.MSV 的方法如下。

- 20 在记录了文件 DOG.MSV 的存储区域的结束处指出了地址“10”。在以地址“10”为基础对所述 FAT 上的入口地址进行搜索之后, 作为一个入口发现所述段地址“11”。通过参考所述入口地址“11”, 作为一个入口发现所述段地址“12”。通过参考所述入口地址“12”, 发现作为一个入口的段地址“100”。通过进一步参考所述入口地址“100”, 发现作为一个入口的所述段地址“101”。通过最后
25 参看所述入口地址“101”, 记录指出所述文件结束的代码 FFF。

因此, 文件 DOG.MSV 占据段 10、11、12、100 和 101, 通过参考所述数据区域的段 10、11 和 12, 可以访问上面记录了与文件 DOG.MSV 的第一部分对应的 ADPCM 数据的区域。通过进一步参考所述数据区域的段 100 和 101, 可以访问上面记录了与所述文件 DOG.MSV 的第二部分对应的 ADPCM 数据的
30 区域。

关于文件名 MAN.MSV, 通过在地址“110”的基础上在所述 FAT 上搜索所述入口地址, 做为一个入口发现段地址“111”。通过参考所述入口地址“111”, 记录表示所述文件结束的代码“FFF”。由此, 这个文件 MAN.MSV 占据段 110 和 111。

- 5 如上所述, 被不连续记录在一个闪速存储器上的一个文件的数据部分能够被互联并被陆续地重现。

现在描述分割图 29 所示三个文件中的一个文件 CAT.MSV 的情况。图 30 示出了当文件 CAT.MSV 被分割时的存储器映像。当执行分割操作时在所述 FAT 上的编辑处理如下。

- 10 现在假设用户在段 6 和段 7 之间的边界处执行分割操作, 从而建立两个文件、即 CAT1.MSV 和 CAT2.MSV。

- 预先分别记录在段 201 和 202 上的文件 DOG.MSV 和 MAN.MSV 首先被分别移位到段 202 和 203。然后, 段 200 的文件名被改变成由所述用户输入的 CAT1 和识别符 MSV 相结合的 CAT1.MSV 和由用户所输入的 CAT2 和识别符 MSV 相结合的 CAT2.MSV 并记录在段 201 中。
- 15

然后, 利用 CAT1.MSV 对记录在所述子辞典上的 CAT.MSV 进行重写, 和所述 CAT2.MSV 被记录在一个未使用的存储区上。

- 存储了 CAT2.MSV 的段号“7”被记录在上面记录了 CAT2.MSV 的存储区的结束处。此后, 在所述 FAT 的入口地址“6”被重写为“FFF”, 从而使由所述子辞典的 CAT1.MSV 存储区指出的文件末端可以是段 6。
- 20

5 条形存储器的文件结构

5.1 辞典结构

存储在所述条形存储器 1 中的文件结构如下。首先参考图 14 描述辞典结构。

- 如上所述, 能够由所述条形存储器处理的主数据包括运动图象数据、静止图象数据、从一个话筒记录的消息数据、从诸如 CD 和 MD 的记录介质记录的高质量音频数据、控制数据、电话辞典数据等。因此, 如图 14 所示, 在所述辞典结构中, 在一个根辞典下设置有一个消息辞典(COICE)、一个静止图象辞典(PCIM)、一个运动图象辞典(MOxxxxnn)、一个控制辞典(CONTROL)、一个电话辞典(TEL)。
- 25

- 30 在这个实施例中, 下面将结合消息数据文件详细讨论所述辞典的结构。作为辞

典 VOICE 的子辞典, 消息列表文件(MSGLIST.MSF)、消息列表文件 B、所述消息列表文件的拷贝(MSGLIST.MSF)、文件夹(FOLDER1 和 FOLDER2)等被形成, 如图 14 所述。例如文件名 98120100.MSV 的实际消息文件数据被形成在所述文件夹内。

- 5 上述的辞典结构仅仅是一个例子, 在诸如 FOLDER1 下也可以形成其它的文件夹。

所述 VOICE 辞典结构被登录在一个消息列表文件中, 并且利用相应的驱动系统被任意地建立。所述管理列表文件被用做用于所述辞典结构的管理文件, 并在所述条形存储器 1 内具有一个备份拷贝, 借此, 避免了所述数据被意外擦
10 除。

5.2 消息列表文件

下面将结合图 15 到 18 讨论所述消息列表文件的结构。在图的行和列中表示的数字是十六进制的字节数。

图 15 示出了所述消息列表文件的数据结构。所述消息列表文件的前 32 个字节
15 被用做一个标题, 后面跟随有 64 个字节的文件夹入口和多个 32 个字节的消息入口。利用这种方式, 预定数量的文件夹入口和消息入口的组合被设置在所述消息列表文件中。

在如上所述构成的消息列表文件中, 安排由所述文件夹入口指定的文件夹的顺序指出所述文件夹和显示所述文件夹的顺序的转换。由所述消息入口指定的消息
20 数据的安排顺序指出所述消息数据的转换和显示, 以及重现所述消息数据的顺序。

图 16 示出了所述消息列表文件 32 个字节的标题结构。在所述标题中, 如图 16 所示, 描述了 4 个字节的消息列表文件 ID(MSG-ID)、两个字节的格式版本号(FMT-VER)、两个字节的标记码(MCode)、8 个字节的编辑数据和时间
25 (YMDHMSV)、4 个字节的文件号(FILE-NO)、两个字节的文件夹入口尺寸(Psize)、两个字节的消息入口尺寸(MSize)、两个字节的文件夹入口偏移(OFFSET)、两个字节的字符码(CCODE)和两个字节的修正号(REV)。在所述标题中, (R)表示备用, 相同的简化被应用于图 17、18 和 20。

所述引导消息列表文件 ID(MSG-ID)是一个诸如 0x4D53474C=" MSGL 的用于
30 指出相应文件是一个消息列表文件的固定值。

格式化版本号(FMT-VER)指出在所述条形存储器 1 中用于控制声音数据文件的系统中格式的版本数,借此,可以识别当前所使用的版本数。所述数的较上字节表示所述主数,较下的两个字节指出次要的数,从而例如 0x0100 表示版本 1.0。

- 5 制造者代码(MCode)表示所述制造者的代码和最近曾对所述消息列表文件进行编辑的一个系统的模型。被编辑的日期和时间(YMDHMSW)指出年、月、日、小时、分、秒,和最近曾对所述消息列表文件进行编辑的某一天的某一天。使用二进制值,其中指定总数为 8 的字节:两个字节用于年,一个字节用于其它信息中的每一个。在所述信息中设置了用于指出某周某天(星期一到星期日)的特定
- 10 值。

所述文件号(FILE-NO)表示最新建立的文件号。所述文件号被做为用于建立消息文件名的系列号而指出且每当建立一个新文件名时被增加,借此以保留一个用于随后建立的文件名的值。当日期变化时所述文件号被复位。

- 所述文件夹入口尺寸(FSIZE)指出图 15 所示的文件夹入口(FOLDER
- 15 ENTRY),并且是例如 64(字节)的固定值。

消息入口尺寸(MSIZE)表示图 15 所示的一个消息入口的尺寸(MESSAGE ENTRY),并且例如是 32(字节)的固定值。

- 文件夹入口偏移(OFFSET)表示从所述文件开始所述第一文件夹入口偏移一个偏移值的位置。例如,对于图 15 所示位于从 0x0020 到 0x005F 的所述文件夹的开始位置,设置 32 作为从所述文件开始的偏移值。
- 20

所述文件夹入口的结构示于图 17。在所述文件夹入口中,描述了两个字节的文件夹 ID(FLD-ID)、两个字节的制造者代码(MCode)、12 个字节的文件夹名(FLD-NAME)、两个字节的字符代码(C-CODE)和 44 个字节的显示(DISPLAY-NAME)。

- 25 所述文件夹 ID 是一个例如 0x4644(=“FD”)的固定值,用于识别所述文件夹入口数据的标题。所述制造者代码(MCode)表示所述制造者的一个代码和用于建立相应文件夹的系统的模型。

- 关于所述文件夹名(FLD-NAME),在一个字符串中记录了诸如以由例如 JIS x 0201 规定的代码表示的一个辞典名的所述 FAT 上的文件夹名。所述文件夹名基本
- 30 上是由一个具有最大 8 个字节的辞典名、一个点“.”和一个具有最大三个字节

的扩展形成的。如果所述扩展是不需要的，可以仅仅指出一个辞典名而不需要所述扩展和所述一个点。在一个字符串的末尾指出了结束符，虽然如果所述字符串占据了所述文件夹名的整个记录区域而由于缺少间隔使这个结束符可以被省略。

- 5 所述字符代码(C-CODE)是一个用于识别描述后续显示名和指出例如与 JIS x 0208-1997(所谓被移位的 JIS 码)相关的 0x90 或与 ISO8859-1 相关的 0x03 的字符代码的代码。

所述显示名(DISP-NAME)指出在一个系统上显示并记录在一个字符串、即由上述 C-CODE 指出的字符代码中的文件夹名。至少有一个字节的结束符 0x00 被
10 记录在所述字符串的末尾，和在 0x00 后的值是任意的。如果所述文件夹名占据了所述显示名的整个记录区域，那么由于缺少间隔，所述结束符不是必须的。如果所述系统自动地建立所述文件夹名，那么所述 FAT 上的文件夹名被记录。

图 18 示出了所述消息入口的结构。所述消息入口是由一个字节的消息 ID(MID)、一个字节的优先级(PRI)、一个字节的报警模式(AL-M)、五个字节的报警日期(AL-DATE)、八个字节的文件名(FILE-NAME)和六个字节的记录时间和日期(REC-DATE)形成的。
15 警日期(AL-DATE)、八个字节的文件名(FILE-NAME)和六个字节的记录时间和日期(REC-DATE)形成的。

所述消息 ID 是一个用于识别所述消息入口数据的头的例如 0x4D (= “M”) 的固定值。所述优先级(PRI)表示所述消息的顺序(或优先级)，并且是由例如从 0x00 到 0x03 的 4 个级别中的一个较高优先级值指出的。当没有规定所述优先级(PRI)
20 时，设置为 0x00。

所述报警模式(AL-M)指出一个报警功能模式。各个位规定如下。

	位 1, 位 0...	设置时间和日期
	位 1=0, 位 0=0	在指定的时间处操作
	位 1=0, 位 0=1	在一周指定的一天处操作
25	位 1=1, 位 0=0	在指定的时间、日期和月处操作
	位 1=1, 位 0=1	备用
	位 2...	备用(设置为 0)
	位 3...	剩余标记
	0	报警操作后位 7 被设置为 0(清除)
30	1	报警操作后位 7 不被设置为 0(清除)

	位 4...	备用(设置为 0)
	位 5...	消息重现标记
	0	在报警时间不重现消息
	1	在报警时间重现消息
5	位 6...	报警声音标记
	0	在报警时间处不发出报警声音
	1	在报警时间处发出报警声音
	位 7...	报警设置标记
	0	不设置报警
10	1	设置报警

如果没有使用报警功能，所述报警模式(AL-M)将被设置为 0x00。当第 5 位和第 6 位是 1 时，按照发出报警声音、重现所述消息和擦除所述消息的顺序执行操作。

所述报警日期(AL-DATE)指出执行报警的时间和日期，并按照年、月、日、时、分、和某周的某天的顺序指出，每片信息具有一个字节。所述年是从 1980 年开始的一个偏移值。值 0 到 127 表示 1980 年到 2107 年。如果没有设置年，则设置 0xFF。禁止使用除 0 到 127 以外的值。

在所述文件名(FILE-NAME)中，在所述 FAT 上记录了消息文件名，和在由例如 JIS x 0201 规定的代码中记录了没有扩展的文件名。所述结束符 0x00 被记录在所述字符串的末尾，虽然如果所述字符串占据了所述文件名的整个记录间隔而使得由于缺少间隔可以省略所述结束符。

在所述记录日期(REC-DATE)中，在该处记录了一个消息的时间和日期以一个字节的二进制值按照年、月、日、时、分和秒的顺序被记录。所述年表示从 1980 年开始的偏移。值 0 到 127 表示 1980 年到 2107 年。如果没有设置年，则设置 0xFF。禁止使用除 0 到 127 和 0xFF 以外的值。

如上所述构成的消息列表文件直接位于所述 VOICE 词典之下，并由各独立系统共享。

当一个系统首先识别所述条形存储器 1 时，诸如当所述系统被复位或当所述条形存储器被插入所述系统时，执行下述(1)到 (7) 的操作。

(1) 首先检查所述 VOICE 词典中的子词典入口是否与所述消息列表文件的内

容相匹配。

(2) 将新建立的子辞典添加到所述消息列表文件上。

(3) 从所述消息列表文件中删除一个已经被擦除的子辞典。(由于在个人计算机(PC)上不需要执行所述消息列表文件的编辑操作就可以添加或删除一个子辞
5 典, 所以, 操作(1)和(2)是必须的)。

(4) 检查在所述 VOICE 辞典下的所述子辞典中的文件入口是否与所述消息列表文件的内容匹配。

(5) 将新建立的文件添加到所述消息列表文件上。

(6) 从所述消息列表文件中删除一个已经被擦除的文件。(由于在 PC 上不
10 要执行所述消息列表文件的编辑操作就可以添加或删除一个文件, 所以, 操作(5)和(6)是需要的)。

(7) 如果在所述系统中没有使用辞典, 新建立一个辞典。

所述消息列表文件的尺寸将被固定为 32736 个字节(32 x1024 - 32)。在最后消息入口之后的 32 个字节中设置 0x00。当在所述 VOICE 辞典下的子辞典的数量是
15 1 时, 可以进入的消息(FAT 文件)的最大数量是 1020。在所述消息列表文件中, 一个文件夹入口(子辞典)需要等于两个消息入口区域的区域, 由此, 每当文件夹入口的数量增加 1 时, 能够进入的消息的数量减少 2。

5.3 消息数据文件

用于存储实际消息数据的消息数据文件如下。所述消息数据文件的文件名被登
20 录在所述消息列表文件中, 并由所述系统任意建立。

图 19 示出了一个消息数据文件的结构。所述消息数据文件是由一个格式帧(FORMAT FRAME)、一个 TOC 帧(TOC FRAME)、一个题目帧(TITLE FRAME)、一个制造者帧(MAKER FRAME)、一个作者帧(AUTHOR FRAME)、一个信息帧(INFORMATION FRAME)、一个空白帧(SPACE FRAME)和一个数据帧(DATA
25 FRAME)形成的。但是, 所述 TITLE FRAME、MAKER FRAME、AUTHOR FEAME 和 INFORMATION FRAME 的提供是任选的, 所述消息数据文件基本上是由围绕图 19 所示实线的所述 FORMAT FRAME、TOC FRAME、SPACE FRAME 和 DATA FRAME 形成的。

所述格式帧被用做与相应消息数据文件相关的基本管理信息, 并用于指出所述
30 编码译码器类型等, 这将在后面详细讨论。

所述 TOC 帧被用做指出所述消息数据文件各帧排列的管理信息,换言之,所述消息数据文件的结构可以由所述 TOC 帧来识别。

在所述数据帧中存储了实际的消息数据。将在下面详细讨论的空白帧是一个重现禁止(不使用)区域,该区域被用做用于扩展所述 TOC 帧的备用区域或用于在一个文件内设置重现禁止区域。

尽管在同一个文件内只能单一地提供其它帧,但是,在同一个文件内可以提供多个数据帧和多个空白帧。所述格式帧总是被设置在所述文件的头部,后面紧跟着所述 TOC 帧。

可以任选的所述题目帧、制造者帧、作者帧和信息帧按照帧 ID 的升序被集中设置在所述 TOC 帧之后(下面讨论)。所述空白帧总是位于这些任选帧的后面。

图 20 示出了所述格式帧的结构。所述格式帧是一个基本帧,在该帧中,描述了编码译码器的类型等,它必须被置于所述帧的头部。

如图 20 所示,所述格式帧是由一个 8 个字节的文件 ID(FILE-ID)、4 个字节的帧格式化尺寸(SIZE-FMT)、两个字节的格式化版本数(FMT-VER)、16 个字节的公司名(C-NAME)、16 个字节的集名(S-NAME)、两个字节的集固件版本数(SET-VER)、8 个字节的记录时间和日期(DATE-TIME)、两个字节的格式化 ID(FMT-ID)、两个字节的信道号(CHAN)、4 个字节的取样频率(每秒取样)(SAMP)、4 个字节的每秒平均字节(BYTE)、两个字节的块对准(ALIGN)、两个字节的每次取样的位(BIT)和两个字节的附加区域尺寸(EXT)形成。可以提供附加区域。

所述文件 ID(FILM-ID)使用例如 ISO8859-1 字符码指出相应的文件是一个声音格式文件(在这个实施例中是一个声音数据文件格式)。所述文件 ID 是一个例如 MS_VOICE 的固定值。

所述帧格式化尺寸(SIZE-FMT)表示以字节为单位的所述格式帧的尺寸。如在下文将要详细描述,由于所述格式帧尺寸的这个描述,可以访问所述 TCO 帧。

所述格式化版本数(FMT-VER)指出所述声音格式化的版本。较上面的一个字节指出主版本数,较下面的一个字节表示次要的版本数。例如,0x0100 是版本 1.0,0x0203 是版本 2.3。

所述公司名(C-NAME)以字符串的形式使用例如 ISO8859-1 字符码指出建立所述文件的公司名称。在所述字符串的末尾处设置结束码 0x00, 虽然如果所述字符串占据了所述公司名的整个区域而使得由于缺少间隔可以省略所述结束码。

- 5 所述集名(S-NAME)以字符串的形式使用例如 ISO8859-1 字符码指出建立所述文件的系统的名称, 在所述字符串的末尾设置结束码 0x00, 虽然如果所述字符串占据所述集名的整个区域而使得由于缺少间隔可以省略所述结束码。

所述集固件版本数(SET-VER)指出所述系统固件的版本。较上面的一个字节指出主版本数, 较下面的一个字节指出次要的版本数。例如, 0x0100 是版本 1.0,

- 10 0x0203 是版本 2.3。

所述记录时间和日期(DATE-TIME)是由一个二进制值表示的, 在该值中, 两个字节被指定给年, 一个字节被指定给诸如月、日、时、分、秒和某周的某天的其它信息中的每一个。

在所述格式 ID(FMT-ID)中设置的所述编码译码器类型如下。

- 15 0x0002 G726 ADPCM 22kHz/3 位
 0x0005 G726 ADPCM 11kHz/3 位
 0x0007 G726 ADPCM 8kHz/4 位
 0x0009 G726 ADPCM 8kHz/2 位

所述信道号(CHAN)指出多个信道的号如下。

- 20 0x0001 单声道
 0x0002 立体声

每秒取样(SAMP)指定取样频率如下。

- 0x00001F40 8 kHz
 0x00002B11 11.025kHz

- 25 所述每秒平均字节(BYTE)表示每秒的字节数, 它被用于根据数据尺寸计算重现时间。

所述块对准(ALIGN)以字节为单位指出一组不能被分离的数据, 它被用于定位数据的开始。例如, 0x0030 是 46 个字节, 0x0010 是 16 个字节。

所述每次取样的位(BIT)指出每次取样的位如下。

- 30 0x0004 4 位

0x0003 3 位

0x0003 2 位

在所述附加区域尺寸(EXT)中,用于描述所述编码译码器特有信息的一个区域(附加区域)的尺寸是以字节位单位指出的。即,如果建立了一个附加数据区域,则
5 指出所述区域的尺寸。所述附加数据区域是以 16 字节为单位建立的。如果不需要所述附加数据区域,则在所述附加区域尺寸(EXT)中设置 0x0000。

在所述格式帧之后,配置了图 31 所示的 TOC 帧。在所述 TOC 帧中,描述了在一个文件中各帧的排列。由于 8 个字节被用于每个帧,所以,所述 TOC 帧的尺寸等于 8 个字节与帧总数的乘积。因此,所述 TOC 帧的尺寸根据帧的增加或删除而变化。如果帧的总数是奇数,则在所述 TOC 帧结束处的未使用的 8 字节区域
10 中设置 0x00。在图 31 中,在从 0x0028 到 0x002F 的地址中设置了 0x00。

用于一帧的 8 个字节的结构如下。

帧 ID(FRAME ID) 1 字节

备用(RESERVED) 3 字节(固定值 0x00)

15 帧尺寸(FRAME SIZE) 4 字节

由所述帧 ID 规定的帧类型如下。

格式帧 0x01

TOC 帧 0x02

空白帧 0x03

20 数据帧 0x04

题目帧 0x05

制造者帧 0x06

作者帧 0x07

信息帧 0x08

25 所述帧尺寸表示由所述帧 ID 规定的以字节为单位的帧的尺寸。

在所述 TOC 帧中,按照所述帧的排列顺序描述了一组上述 8 个字节的数据。下面将描述所述 TOC 帧的一个特定例子。

根据所述 TOC 帧的内容,可以识别相应声音数据文件的帧结构。在访问所述声音数据文件的过程中,首先识别所述格式帧,然后检查所述 TOC 帧的内容。如
30 上所述,由于在所述格式帧尺寸(SIZE-FMT)中对所述格式帧的描述,使得能够对

所述 TOC 帧进行访问。即, 所述 TOC 帧的开头地址是一个从所述文件的头开始的偏移所述格式帧的尺寸的值的地址。

如上所述, 在所述 TOC 帧之后, 可以提供所述题目帧、制造者帧、作者帧和信息帧。

- 5 所述题目帧是一个用于记录所述题目的帧, 两个引导字节指出例如所记录题目的代码类型如下。

0x0000 JIS x 0208-1997(SJIS)

0x0001 ISO8859-1

- 10 所述题目帧的尺寸是 16 个字节的整数倍, 在所述题目字符串的结束处设置了结束码 0x00。

所述制造者帧表示公司名、机器名和用于建立一个文件的的机器的固件版本号。在所述制造者帧中, 以及在所述题目帧中, 两个引导字节指定后面跟随有一个字符串的字符码的类型。所述制造者帧的尺寸是 16 个字节的整数倍, 在所述字符串的结束处设置了结束码 0x00。

- 15 所述作者帧表示一个字符串作者(版权所有者)的名字。两个引导字节指出后面跟随有一个字符串的所述字符码的类型。所述作者帧的尺寸是 16 个字节的整数倍, 在所述字符串的结束处设置了结束码 0x00。

- 20 所述信息帧是一个用于记录附加信息的帧。所述附加信息包括歌曲名、艺术家名和指挥家名。所述附加信息还可以包括当执行平均音量学习时播放的数量。第一字节指定信息种类 ID, 第三和第四字节表示字符码类型, 其中描述了一个信息字符串。所述字符串被存储在所述帧标题从第 9 开始的字节中。所述信息帧的尺寸是 16 个字节的整数倍, 在未使用的区中设置了 0x00。

- 25 在前述任选帧或 TOC 帧之后是所述空白帧。用做重现禁止区域的空白帧的提供是必要的。由此当重现所述数据时跳过所述空白帧也是必要的。所述空白帧被用做一个备用区域以扩展所述 TOC 帧或当由于执行分割操作(后述)而在一个段中建立重现禁止区域时使用。任一类型的数据都可以存储在所述空白帧中, 特殊的空数据可以被记录。所述空白帧的尺寸是 16 字节的整数倍。下面将描述所述空白帧的处理。

- 30 在所述数据帧中存储了实际的消息数据。所述消息数据作为从所述数据帧的标题字节的最高有效位(MSB)开始的规定数量的位被无间隔地存储。对于数据的存

储边界，使用了所述字节边界。例如，分别对于诸如图 32 和 33 所示的每次取和两位数据，数据的存储边界是一个字节。对于每次样 3 位数据，例如如图 4 中所示的数据，数据的存储边界是三个字节。当分割所述声音数据时，上述存储单元被用做数据单元。

- 5 所述消息数据文件是由上述各个帧构成的。该消息数据文件结构的一个例子示于图 21A 到 22B。在下面的描述中，假设没有提供可以任选的题目名、制造者名、作者名和信息名。

当声音数据被正常记录时构成的帧结构如图 21A 所示。如图 21A 所示，从所述文件的头开始形成了所述格式帧、所述 TOC 帧、所述空白帧和所述数据帧。如
10 上所述，形成所述单个文件的多个块(段)是由所述 FAT 链接的多个块，并且不必是物理连续的块。

在所述 TOC 帧中指出诸如图 21B 所示信息的上述帧结构的信息。

如上所述，其中每个都具有 8 个字节的各项信息被按照帧的排列顺序设置。在第一个 8 字节的字段中，帧 ID、即 0x01 被设置，后面跟随有所述格式帧的尺寸。
15 在第二个 8 字节的字段中，在所述帧 ID 中设置 0x02，后面跟随所述 TOC 帧的尺寸。由于帧的数量为 4，所以，所述 TOC 帧的尺寸为 4 x 8 字节 = 32 字节 (0x00000020)。在第三个 8 字节的字段中，在所述帧 ID 中设置 0x03，后面跟随所述空白帧的尺寸。在最后一个 8 字节的字段中，在所述帧 ID 中设置 0x04，后面跟随有所述数据帧的尺寸。

20 在所述消息数据文件中，可以使用所述空白帧擦除部分数据并设置成一个重现保护区域。例如，现在假设如图 22A 所示在所述数据帧内具有一个将被擦除并不被重现的消息部分，并且这个消息部分将被所述空白帧取代。由于当正在重现所述数据时所述空白帧将被跳过，所以在数据重现期间所述消息部分将被擦除。

25 当所述帧结构从图 21A 所示的结构变成图 22A 所示的结构时，所述 TOC 帧的内容被更新，如图 22B 所示。

第一个 8 个字节的格式帧的尺寸保持相同。第二个 8 个字节的字段的帧 ID 是 0x02。由于帧的数量从 2 增加到 6，所以所述 TOC 帧的尺寸变化到 6 x 8 字节 = 48 字节 (=0x00000030)。在第三个 8 个字节的字段中，在所述帧 ID 中设置 0x03，后
30 面跟随有所述空白帧。当所述 TOC 帧被扩展 16 字节时，所述空白帧的尺寸减少

16 字节, 变成 0x00000180。

- 在随后的 8 个字节的字段中, 指出一个数据帧的第一部分的尺寸, 所述数据帧包含将不被重现的消息部分并且已经被一个新空白帧分割。在第五个 8 个字节的字段中, 指定了一个包含上述消息部分的新空白帧的尺寸。在最后一个 8 个字节的
- 5 的字段中, 示出了由所述新空白帧分割的数据帧的第二部分的尺寸。

如上所述, 所述消息数据文件的帧结构是由所述 TOC 帧管理的。由于附加一个空白帧或一个数据帧或由于执行了诸如文件分割操作或文件组合操作的编辑操作, 所以, 所述 TOC 帧的内容是根据所述帧结构的变化更新的, 这些将在下面讨论。

- 10 所述空白帧的处理如下。当新建立一个消息数据文件时, 如图 23 所示, 在所述 TOC 帧的后面配置了其尺寸为 128 字节到 640(=128 + 512)字节的一个空白帧。在这种情况下, 所述空白帧的尺寸被确定, 从而使一个数据帧的开始位置位于所述扇区的边界。

- 下面将进行详细描述当一个消息数据文件被分割时, 在包括一个分割点的段中
- 15 一个重现保护区被设置成一个空白帧。

- 具体地说, 如图 24 所示, 如果一个分割点被设置在一个段内, 那么, 所述消息数据的头、即所述文件的第二部分位于所述分割点处。所述文件的最小单元是一个段。由此, 在所述段内分割点之前所述文件的第一部分包含将不被重现的数据部分(图 24 所示的阴影部分), 并且这个数据部分被一个空白帧替换, 从而可以
- 20 避免被重现。

在执行所述消息数据文件的分割操作或组合操作后, 两个空白帧可能被彼此相邻地设置。在这种情况下, 两个空白帧组合成一个空白帧, 换言之, 它们被所述 TOC 帧管理为一个单一的空白帧。

- 如上所述, 一个消息数据文件不仅仅被提供有一个数据帧, 而且还被提供有一个格式帧、一个 TOC 帧、一个题目帧、一个制造者帧、一个作者帧、一个信息帧
- 25 和一个空白帧。综上, 由于下述原因, 所述格式帧、所述 TOC 帧和所述空白帧是主要的。

所述条形存储器 1 不仅可以被用在诸如图 9 所示的一个驱动系统中, 而且可以被用在诸如在一个 PC 中使用 FAT 系统的装置中。

- 30 具体地说, 包含由诸如图 9 所示系统的一个驱动系统记录和管理的消息数据

文件的条形存储器 1 被装载到一个 PC 中,从而使用所述 FAT 系统能够使所述文件被移动或所述文件名被改变。但是,这扰乱了生成文件信息与由所述消息列表文件管理的消息数据文件内容的匹配。为了克服这个缺点,当利用诸如图 9 所示的一个驱动系统首先识别所述条形存储器 1 时,在所述 VOICE 词典中的子词典入口与所述消息列表文件的内容相匹配。这使得即使是例如一个用户改变了所述 PC 上的文件名也可以在所述驱动系统中使用所述条形存储器 1。在这种情况下,如果在所述数据帧中直接提供并在其它帧中间接提供所述消息数据文件,重现所述数据帧需要的信息也能够被传送给图 9 所示的驱动系统。由此,即使是存储在所述条形存储器 1 中的文件在 PC 上被移动,在所述驱动系统中也能够重现所述数据而不会引起甚麽问题。

在使用上述信息重现所述消息的过程中,可期望显示记录时间和剩余时间。在这种情况下,它们可以被显示在图 9 所示驱动系统的显示单元 108 上。

通常,在所述 FAT 文件系统中管理的文件尺寸是以字节为单位显示的。但是,在所述消息数据文件中,所述文件尺寸是按照时间以下述方式而不是使用由所述 FAT 文件系统管理的文件尺寸计算的。

首先使用 TOC 帧计算所述消息文件内数据帧的总的尺寸,然后,使用记录在格式帧中的每秒平均字节将其转换为总的重现时间。剩余时间可以通过从总的重现时间中减去被重现消息的重现时间计算。根据这个计算技术,可以抑制由空白帧和其它帧的数据量引起的计算误差。

20 6.分割编辑操作

下面讨论作为该实施例系统特性的分割编辑操作。

根据所述分割编辑操作,通过一个用户指令规定某个分割点,以便在所述分割点处将一个单一的声音数据文件分割成两个声音数据文件。下面参考图 9, 25 到 28 描述这个操作。

25 图 25 的流程示出了在微机 109 控制下执行的分割编辑操作。图 26A、26B 和 26C 示出了在所述条形存储器 1 上分割消息数据的概念。图 27 和 28 示出了在分割操作之前和之后所述 FAT 的概念。

当条形存储器 1 被装载到所述驱动系统中时,用户能够在显示单元 108 上识别存储在所述条形存储器 1 中的消息数据文件。如果用户希望将某个消息数据文件分割成两个文件,那么,该用户通过操作所述操作单元 112 选择将被分割的消息

数据文件，同时检查在显示单元 108 上显示的文件名。

在用户选择了文件名之后，在图 25 所示的步骤 F101 中，微机 109 设置将被分割的声音数据文件。然后，用户指定被选择文件的一个分割点。各种操作都可以被认为是指定所述分割点，例如，微机 109 可以重现所选择的消息数据文件，

5 用户可以在听重现消息的同时在所述文件将被分割的点处执行全部操作。但是，这个操作可能从所希望的分割点产生轻微的位移，由此，可以更好地提供精确调节所述分割点的功能。

在由用户指定分割点后，微机 109 在步骤 F102 确定所述分割点，并进入到下面的分割操作。在步骤 F103，如果所确定的分割点位于同一个段(块)内，存储在

10 所述段内的数据被拷贝到一个未使用的段内，借此，复制包含所述分割点在内的所述段的数据。但是，如果所设置的分割点位于一个段(块)的边界处，就不需要复制所述段的数据，这将在下面详细描述。另外，存储在包括一个管理标题的文件的开头处所述段中的数据被拷贝到一个未使用的段中，借此，复制包括所述管理标题在内的所述段的数据。所述管理标题是一个数据部分，它包括一个格式帧、

15 一个 TOC 帧和一个空白帧。

在步骤 F104，改变所述段链接，在步骤 F105，改变所述 FAT 的词典入口。因此，形成一个新的入口，从而能够通过使用包括在步骤 F103 新拷贝的分割点的段和包括所述管理标题的段将原始单一文件分割成两个文件。这将在下面详细讨论，

20 然后，在步骤 F106，位于原始文件开头处的管理标题的 TOC 帧被更新，借此，形成与所述分割点前的原始文件的第一部分对应的第一文件。

在步骤 F107，被拷贝和链接到所述文件开头的所述管理标题被更新，借此形成与所述分割点后原始文件的第二部分对应的第二文件。

根据上述处理，完成所述分割操作。与该分割操作处理相关的例子示于图 26A

25 到 28。

现在假设图 26A 所示的文件 F1 是一个由用户选择的消息数据文件，并且被存储在段 CL(2)到 CL(9)中。

指出文件 F1 段链接的 FAT 示于图 27。根据所述词典入口(未示出)，段文件 F1 的标题段是段 CL(2)。在所述 FAT 的段 CL(2)中存储了 003，它指出所述段 CL(2)

30 被链接到段 CL(3)。在段 CL(3)的入口中存储了 004，它表示段 CL(3)被链接到段

CL(4)。此后，链接状态以类似的方式记录，并且在最后段 CL(9)中设置值 FFF，它指出段 CL(9)是最后一个段。

根据上述构成的 FAT，在段 CL(2)到 CL(9)中示于图 26A 的文件存储被管理。在段 CL(A)和后继段中，设置值 000，即这些段未被使用。

- 5 现在假设用户指令在图 26A 所示的分割点 DP 处分割所述文件 F1。所述分割点 DP 位于段 CL(5)内。因此，执行在图 25 所述步骤 F103 中的处理，即，所述段 CL(5)被拷贝到一个未使用的段 CL(A)中，和包括所述管理标题的段 CL(2)被拷贝到一个未使用的段 CL(B)中。

- 10 然后，在步骤 F104 和 F105 中，更新所述 FAT 的段链接和词典链接，从而形成图 26B 和 26C 所示的段链接。如果分割后的两个文件由文件 F1-1 和 F1-2 指出，那么首先形成文件 F1-1 的词典入口，并且所述标题段被设置成所述段 CL(2)。

- 15 如图 28 所示，在所述 FAT 的入口中，在段 CL(2)中设置了 003，在段 CL(3)中设置了 004，在段 CL(4)中存储了 00A，和在段 CL(A)中存储了 FFF。由此，所述 FAT 管理文件 F1-1，从而使得文件 F1-1 被按照段 CL(2)、CL(3)、CL(4)和 CL(A)的顺序存储，如图 26B 所示。

- 20 同时，形成所述文件 F1-2 的词典入口，和所述标题段被设置为段 CL(B)。如图 28 所示，在所述 FAT 的入口中，在段 CL(B)中设置了 005，在段 CL(5)中存储了 006，在段 CL(6)中设置了 007，在段 CL(7)中设置了 008，在段 CL(8)中存储了 009，和在段 CL(9)中存储了 FFF。因此，所述 FAT 管理文件 F1-2，从而使文件 F1-2 按照段 CL(B)、CL(5)、CL(6)、CL(7)、CL(8)和 CL(9)的顺序被存储，如图 26C 所示。

此后，在图 25 的步骤 F106 中，文件 F1-1 的管理标题、即段 CL(2)的内容被更新。具体地说，在所述 TOCA 帧中指出的数据帧的尺寸被更新。

- 25 图 26B 示出了最后段 CL(A)是一个包括所述分割点 DP 的一个段，在所述分割点 DP 后的消息数据的第二部分是不需要的数据。由此，所述数据帧的尺寸被改变为由图 26B 中的一点链线所规定的部分，即、从所述数据帧中删除了阴影部分。这使得所述段 CL(A)的阴影部分无效，由此，在文件 F1-1 中，在分割点 DP 之前原始文件 F1 的数据部分将不被重现。

- 30 在图 25 所示的步骤 F107 中，文件 F1-2 的管理标题、即段 CL(B)的内容被更

新。具体地说,在所述 TOC 帧中指定的所述数据帧的尺寸被更新,并改变所述空白帧的设置。

图 26C 示出了第二段 CL(5)是一个包括所述分割点 DP 的段,在所述分割点 DP 之前的段 CL(5)的消息数据是不需要的数据。另外,段 CL(B)是段 CL(2)的拷贝,对于文件 F1-2 来讲,位于段 CL(2)中所述空白帧之后的数据帧的消息数据也是不需要的数据。

由此,TOC 帧中的所述空白帧的尺寸被改变,以便使位于原来段 CL(2)中并被拷贝到段 CL(B)中的所述空白帧被延伸到所述段 CL(5)的分割点 DP。在 TOC 帧中的所述数据帧的尺寸也被改变。

所述空白帧在文件 F1-2 中经过段 CL(B)和 CL(5)延伸,并作为重现保护区。结果是在文件 F1-2 中,分割点 DP 之后的原始文件的数据部分将不被重现。

下面参照附图 29 和 30 进一步描述所述分割操作。如从前述看出的,在这个实施例中,在执行分割操作过程中,只需要下面简单的处理。包括一个分割点的段和包括一个管理标题的段被拷贝。然后,分割后文件的管理标题(TOC 帧)被部分地更新,数据帧和空白帧的尺寸被改变。结果是原始文件被分割成在所述分割点之前和之后的两个声音数据文件。由于这个处理,可以使执行诸如消息数据文件的分割操作的编辑操作所需的数据拷贝和重写的量最小。这使得可以增加所述编辑操作的效率和使编辑操作所需的访问量最小,借此,减少了处理时间和使功率损耗降到最低级。

如果所述分割点 DP 位于一个段的边界处,不需要拷贝包括所述分割点的段。例如,如果文件 F1 的段 CL(5)和段 CL(6)之间的边界是所述分割点,那么,分割后的文件 F1-1 是由段 CL(2)到段 CL(5)形成的,分割后的文件 F1-2 是由段 CL(2)的复制品段 CL(X)和段 CL(6)到 CL(9)形成的。在这种情况下,对于执行所述分割操作来讲只需拷贝包括所述管理标题的段,借此,使得操作更简单、时间更短。

在图 26A、26B 和 26C 所示的例子中,段 CL(5)被集中到文件 F1-1 中,而段 CL(A)被集中到文件 F1-2 中。但是,段 CL(A)也可以被集中到文件 F1-1 中,而段 CL(2)也可以被集中到文件 F1-2 中。同样的处理可以被应用到包含所述管理标题的段中。

在最初曾经存储原来段 CL(5)和现在被用做部分空白帧的区域中,即在文件

F1-2 的段 CL(5)的一部分当中，保留了所述声音数据，或可以填充诸如 0 数据的空数据。另外，它可以被用做用于存储任意类型附加数据的存储区域。相同的处理可以被用于已经被无效而不是正在用做空白帧的一部分的区域，即图 26B 所示在文件 F1-1 的段 CL(A)的第二部分处的阴影部分。

5 各种类型的数据都可以存储在上述的空白帧中。所述消息数据可以保留在同样的帧中而不必使用一个空白帧来替换它。在这种情况下，，当被分割的两个帧被组合成所述原始帧时，全部段可如原来那样被使用，借此，使得处理更加有效。如果在执行所述分割操作之后两个空白帧位置彼此相邻，它们可以被组合成一个单一的空白帧。

10 所述处理和分割操作的内容不局限于上述的例子，可以进行各种修改。

在前面的实施例，在复制了包括所述分割点的段之后，通过更新所述 TOC 帧，被分割文件的一部分形成一个空白帧或一个无效部分。另外，一个空数据也可以被添加到在所述分割点之前的消息数据中从而形成整个段数据，然后将该段数据存储在一个未使用的段中。空数据也可以被添加到所述分割点之后的消息数据中，从而形成整个段数据，该数据可以被集中到形成所述分割文件的段链接中。在这种情况下，各段的空数据可以作为一个空白帧或无效部分进行管理，如在这个实施例所示。

虽然通过这个实施例就所述系统结构和分割编辑操作对本发明进行了描述，但本发明并不局限于上述结构和操作。

20 特别是，在这个实施例中，已经讨论了使用所述条形存储器在一个系统中对消息数据文件进行的分割操作。但是，诸如音乐数据文件和运动图象数据文件的其它类型文件也可以在系统中使用所述条形存储器进行处理。对所述音乐数据文件或运动图象数据文件执行类似的分割操作，借此使得编辑操作更加有效。

本发明的记录介质并不局限于诸如图 1A 到 1D 所示的条形存储。也可以使用
25 诸如存储器芯片、存储器卡、存储器模块等具有其它结构的固态存储器介质。

另外，也可以对这个实施例中使用的文件系统的格式部分进行修改。

所述闪速存储器的容量变化不局限于图 8 所示。在本发明记录介质中的存储器装置可以是除闪速存储器以外的存储器装置。

如能够从前面的描述中看出的，本发明提供如下优点。用于存储在记录介质中
30 每个文件的管理信息也被存储在所述记录介质中。利用这种管理信息，可以管理

- 主数据和重现保护区的位置。在对文件执行分割编辑操作的过程中，包括一个分割点的例如上述段或块的单元数据被复制，将要被分割文件的管理信息被更新，借此，形成与原始文件第一部分对应的第一文件。用于所述文件第二部分的管理信息被建立并添加到第二部分文件的开头处，在第二文件内的所述分割点之
- 5 前的原始文件的第一部分被形成为一个重现保护区，借此，形成所述第二文件。利用这种安排，分割一个文件所需的数据的运动、拷贝和重写可以最少，和所述编辑操作所需的处理时间和功率损耗也可以下降到最低级。

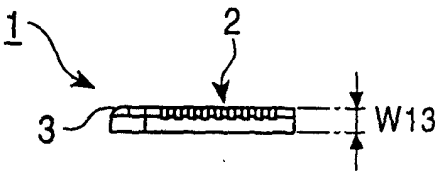


图 1A

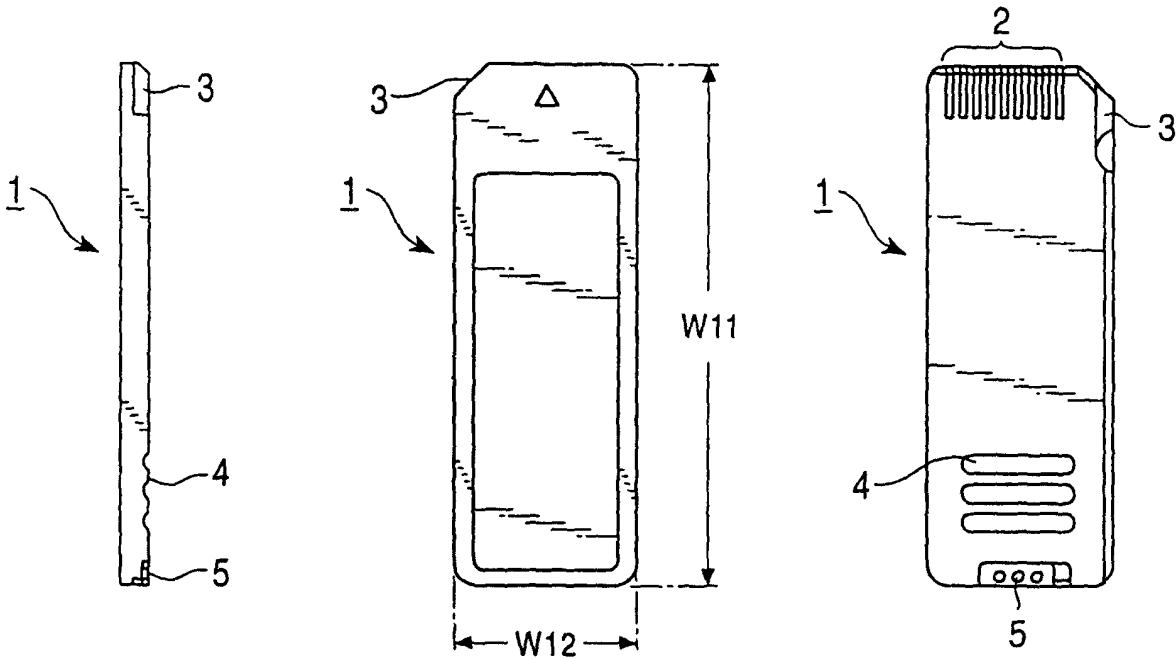


图 1B

图 1C

图 1D

应用处理
文件管理处理
逻辑地址管理
物理地址管理
闪速存储器访问

图 2

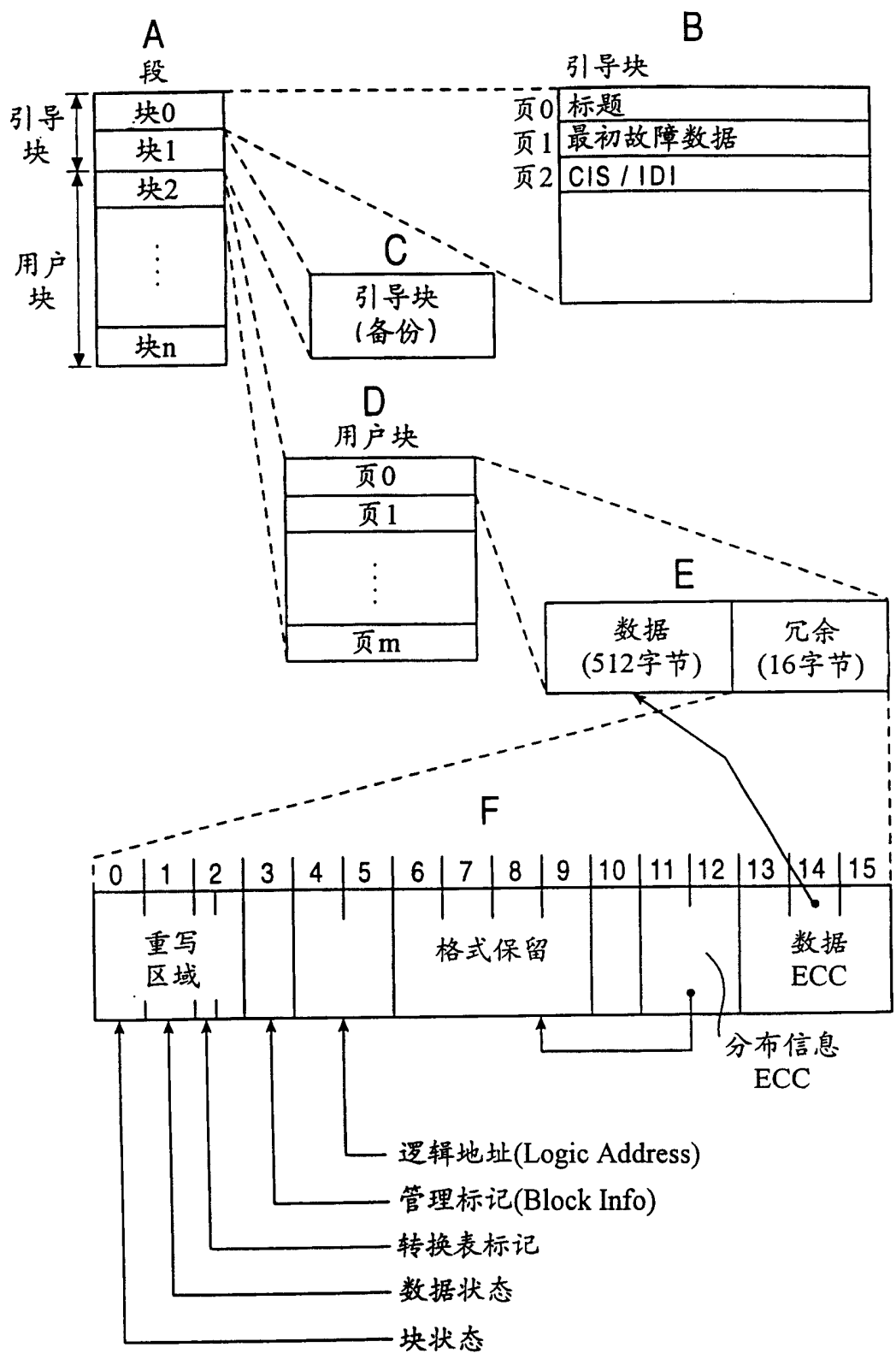


图 3

比特	定义
7	备用
6	备用
5	访问有效(1:空白, 0: 读保护)
4	拷贝 (1: OK, 0: NG)
3	转换表标记 (1: 无效, 0: 表块) *只有最后段有效
2	系统标记 (1: 用户块, 0: 引导块)
1	备用
0	备用

图 4

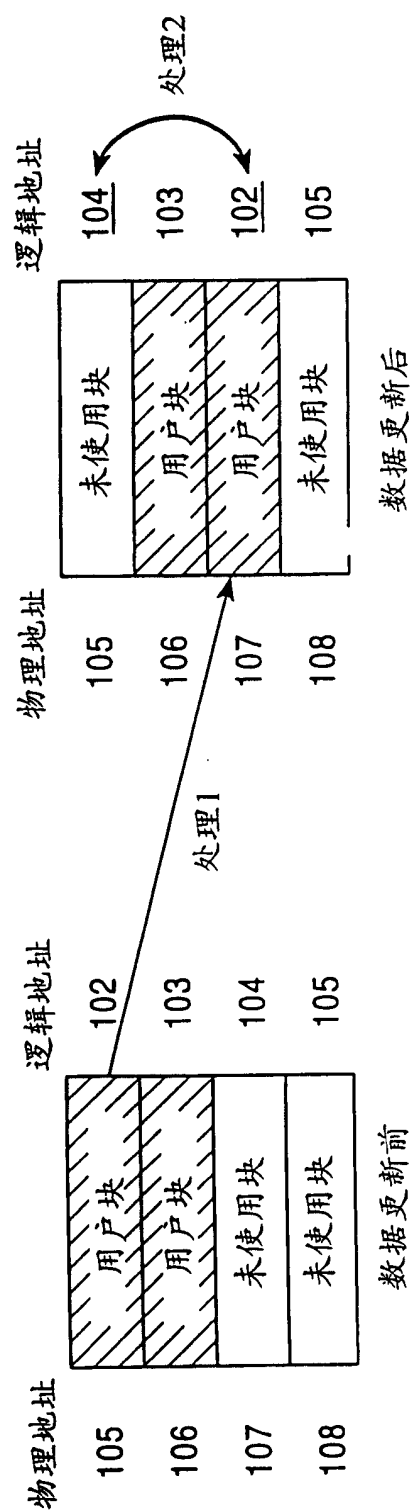


图 5A

图 5B

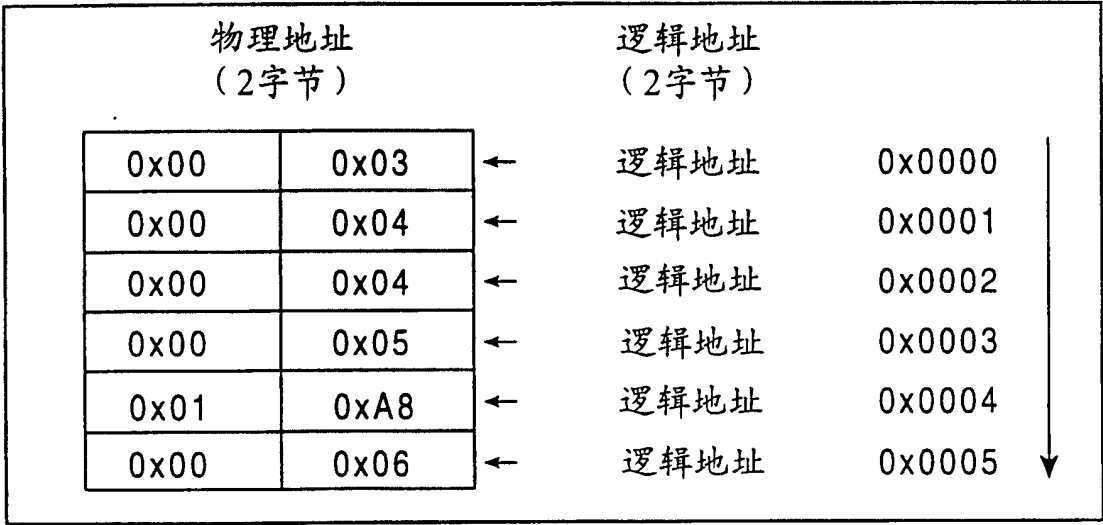


图 6

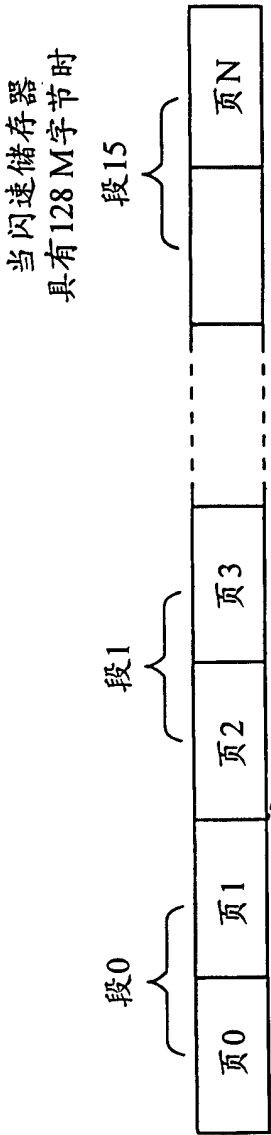


图 7A

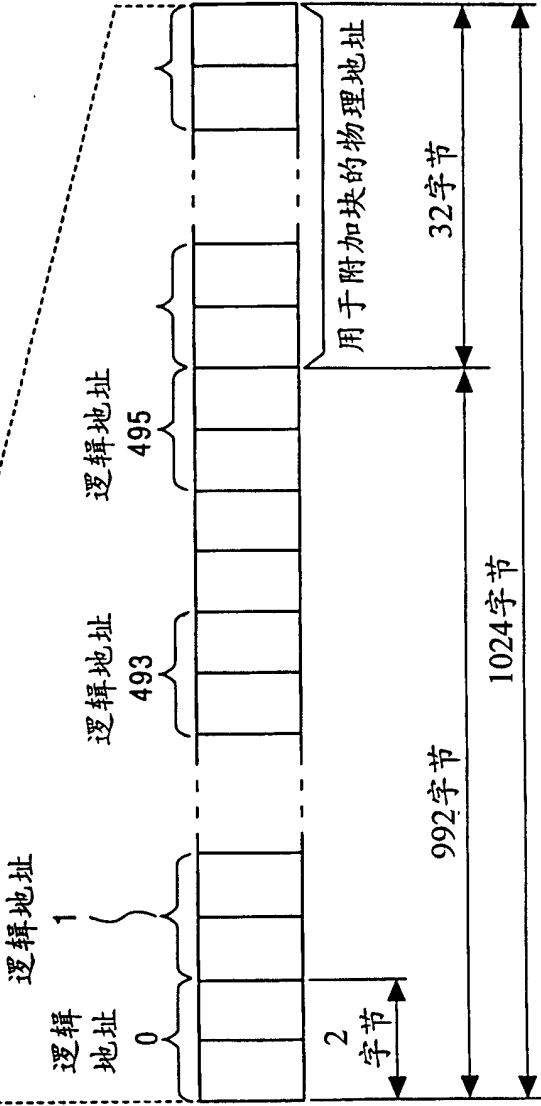


图 7B

A	B	C	D	E
闪存存储器 容量	块数	每个块的容量	每页的 容量	逻辑/物理 地址转换表尺寸
4MB	512 (1段)	8KB (16页)	(512 + 16)B	1KB (2页)
8MB	1024 (2段)	8KB (16页)	(512 + 16)B	2KB (4页)
16MB	2048 (4段)	8KB (16页)	(512 + 16)B	4KB (8页)
	1024 (2段)	16KB (32页)	(512 + 16)B	2KB (4页)
32MB	2048 (4段)	16KB (32页)	(512 + 16)B	4KB (8页)
64MB	4096 (8段)	16KB (32页)	(512 + 16)B	8KB (16页)
128MB	8192 (16段)	16KB (32页)	(512 + 16)B	16KB (32页)

图 8

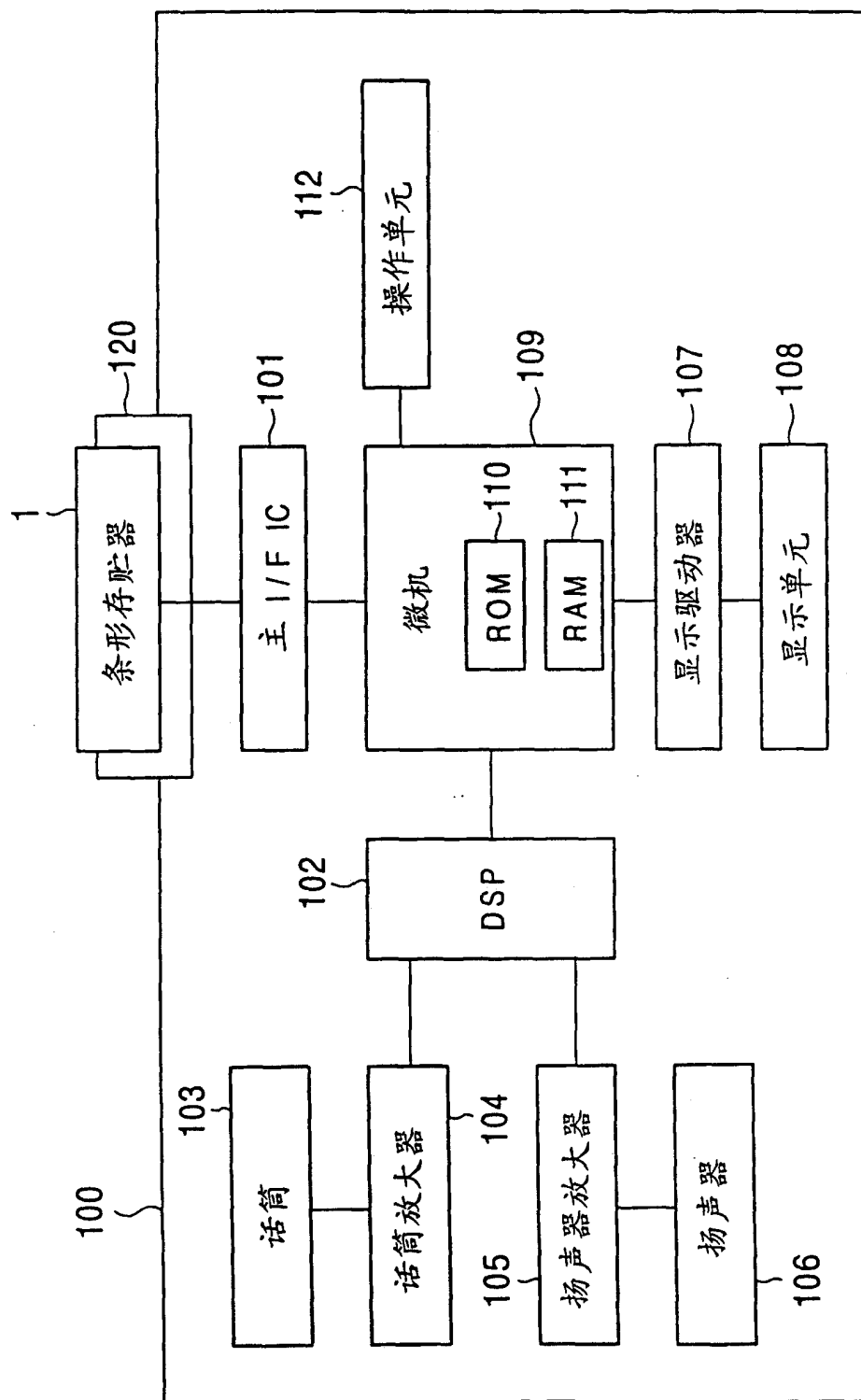


图 9

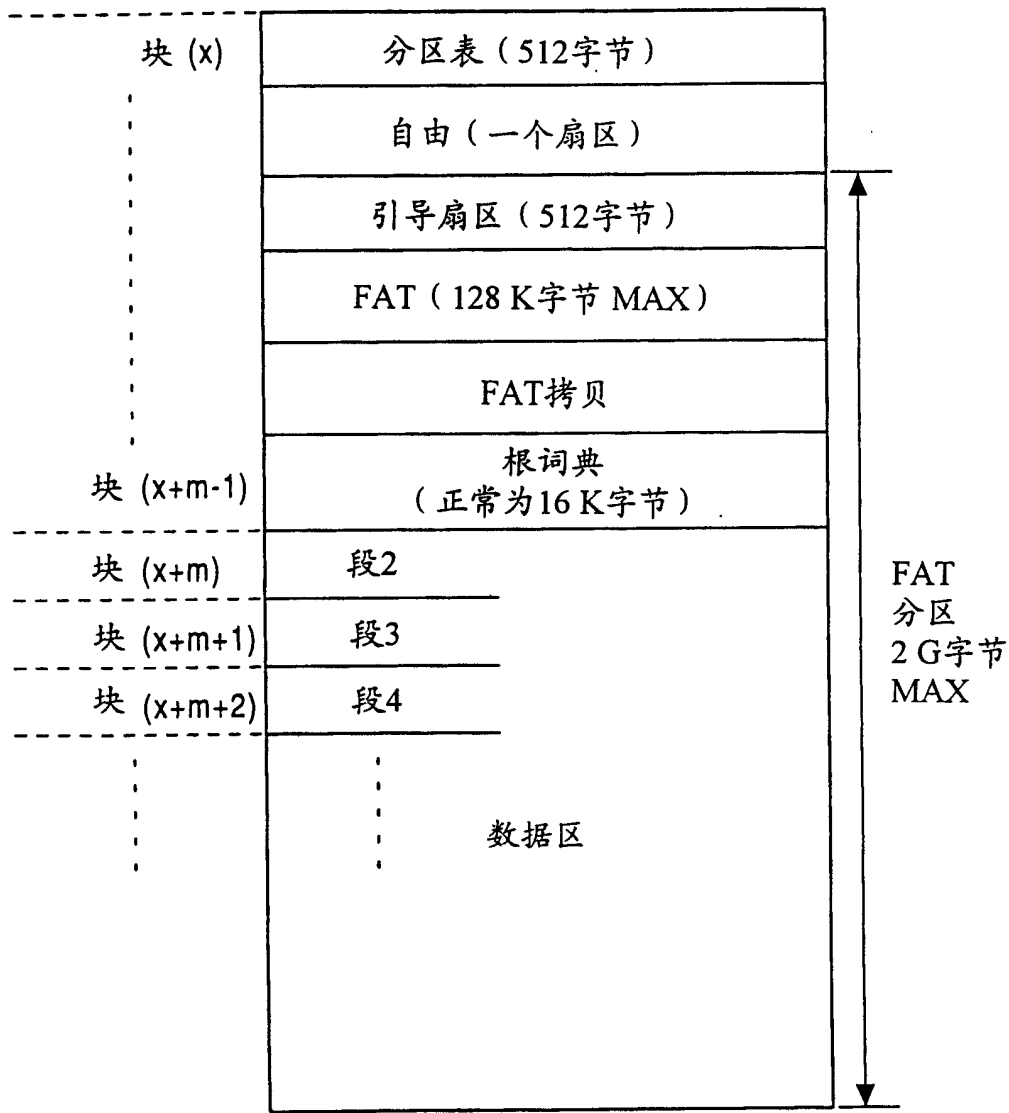


图 10

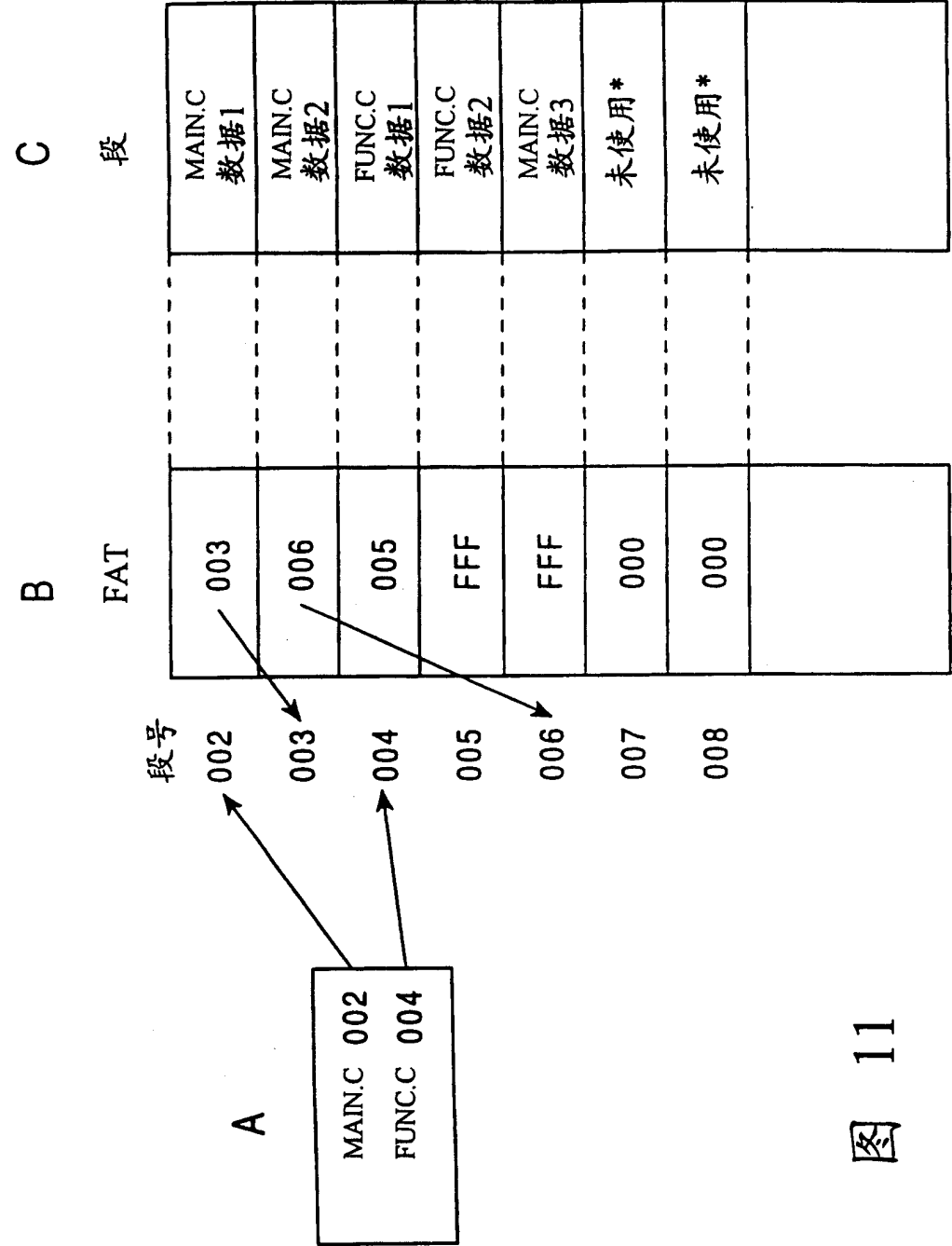


图 11

文件名 (8字节)	扩展 (3字节)	属性 (1字节)	备用 (10字节)	时间 (2字节)	日期 (2字节)	引导段 (2字节)	尺寸 (4字节)
--------------	-------------	-------------	--------------	-------------	-------------	--------------	-------------

图 12

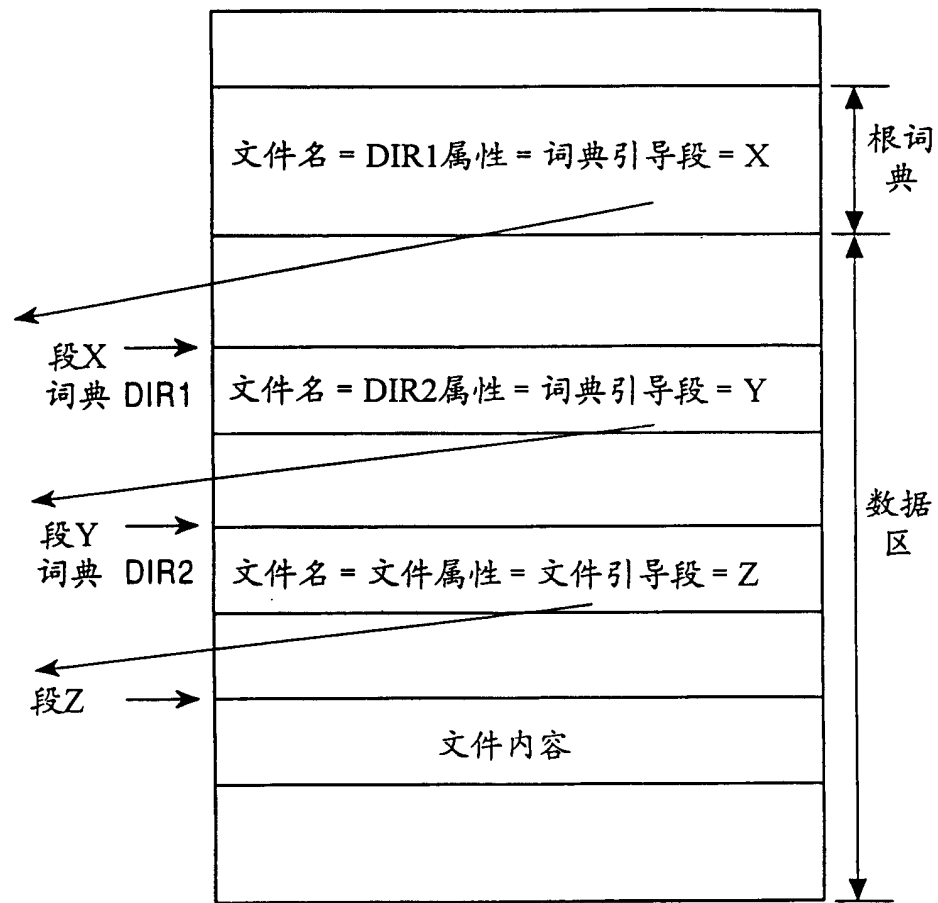


图 13

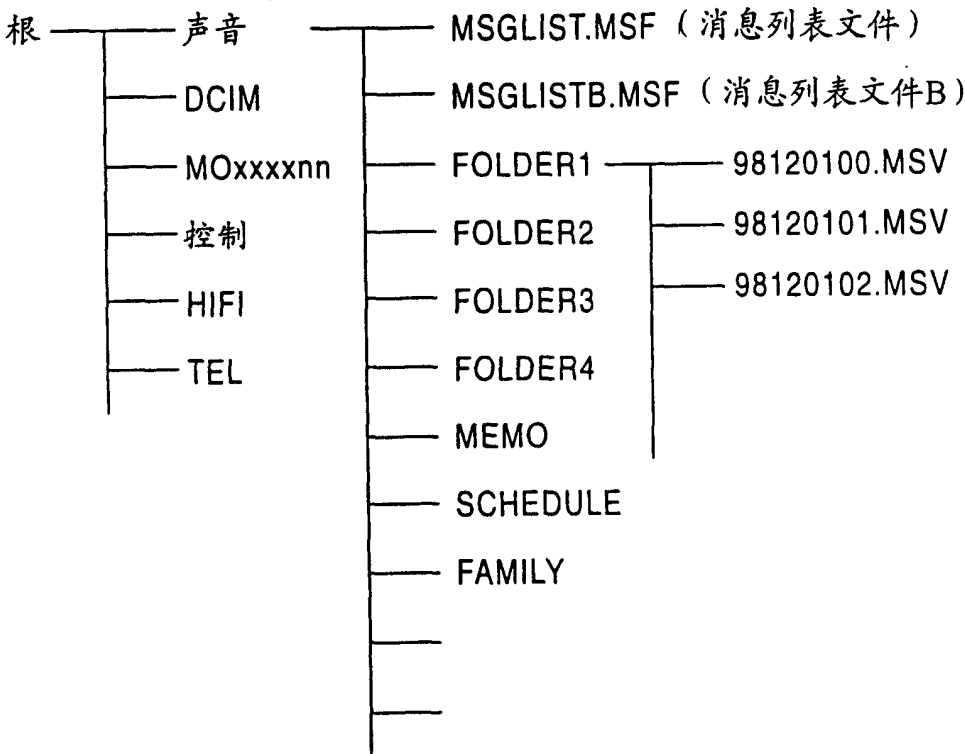


图 14

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0x0000	标题 (32字节)																
0x0010																	
0x0020																	
0x0030	文件夹入口 (64字节)																
0x0040																	
0x0050																	
0x0060	消息入口 (32字节)																
0x0070																	
0x0080	消息入口 (32字节)																
0x0090																	
0x00A0	消息入口 (32字节)																
0x00B0																	
:									:								
:																	
:																	
	消息入口 (32字节)																
	文件夹入口 (64字节)																
	消息入口 (32字节)																
	消息入口 (32字节)																
	消息入口 (32字节)																
									:								
	消息入口 (32字节)																
	文件夹入口 (64字节)																
	消息入口 (32字节)																
	消息入口 (32字节)																
	消息入口 (32字节)																
									:								

图 15

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	MSG-ID				FMT-VER		MCode		YMDHMSW							
0x0010	FILE-NO				FSIZE		MSIZE		OFFSET		CCODE		REV		(R)	

图 16

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F												
0x0000	FLD-ID		MCode		FLD-名																							
0x0010	C-CODE		(R)		DISP- 名																							
0x0020																												
0x0030																												

图 17

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	MID	PRI	AL-M	AL- 数据										(R)		
0x0010	文件名								记录数据						(R)	

图 18

格式帧
TOC帧
题目帧
制造者帧
作者帧
信息帧
空白帧
数据帧

图 19

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	FILE-ID								SIZE-FMT				FMT-VER		(R)	
0x0010	C-NAME															
0x0020	S-NAME															
0x0030	SET-VER		(R)		DATE-TIME								FMT-ID		CHAN	
0x0040	SAMP				BYTE				ALIGN		BIT		EXT		(R)	

图 20

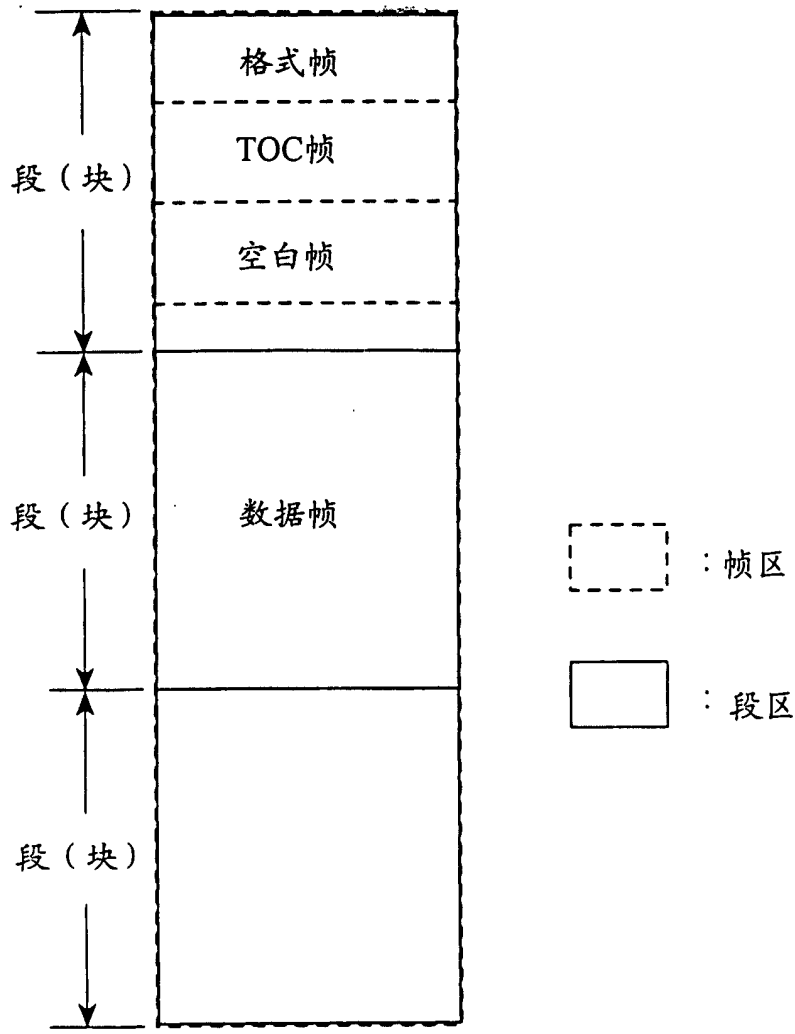


图 21A

ID	备用	尺寸
格式帧 (0x01)	0x000000	0x00000050
TOC帧 (0x02)	0x000000	0x00000020
空白帧 (0x03)	0x000000	0x00000190
数据帧 (0x04)	0x000000	0x00005E00

一个段
(一个块) = 8KB

图 21B

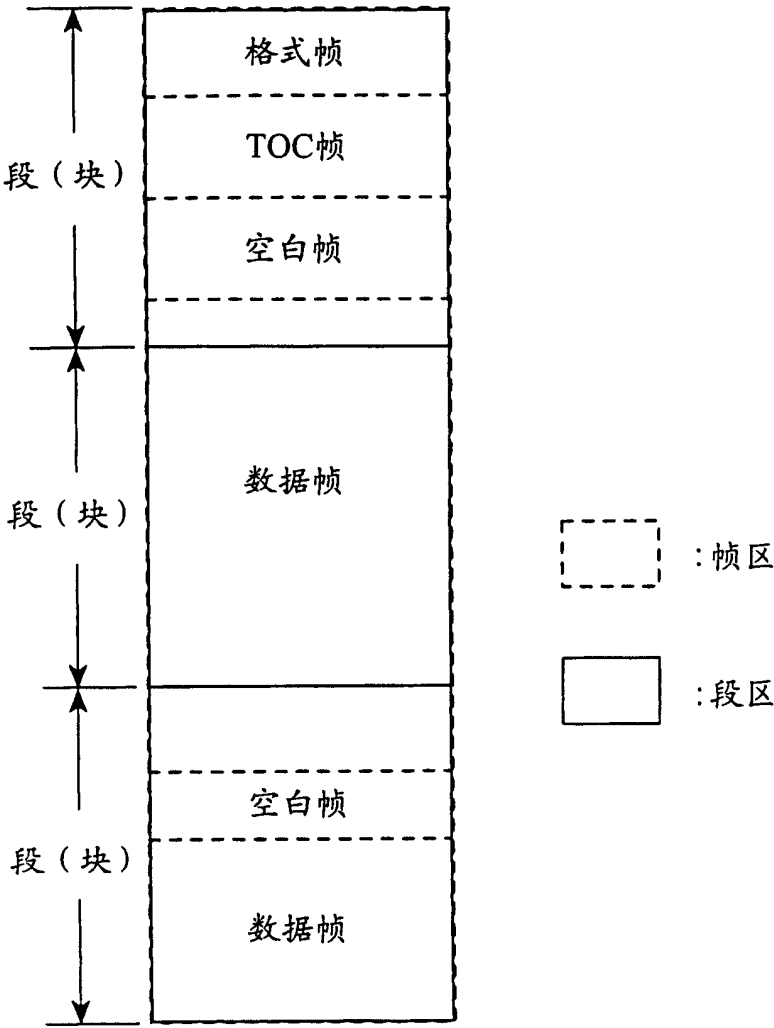


图 22A

ID	备用	尺寸
格式帧 (0x01)	0x000000	0x00000050
TOC帧 (0x02)	0x000000	0x00000030
空白帧 (0x03)	0x000000	0x00000180
数据帧 (0x04)	0x000000	0x00004E00
空白帧 (0x03)	0x000000	0x00000100
数据帧 (0x04)	0x000000	0x00000F00

图 22B

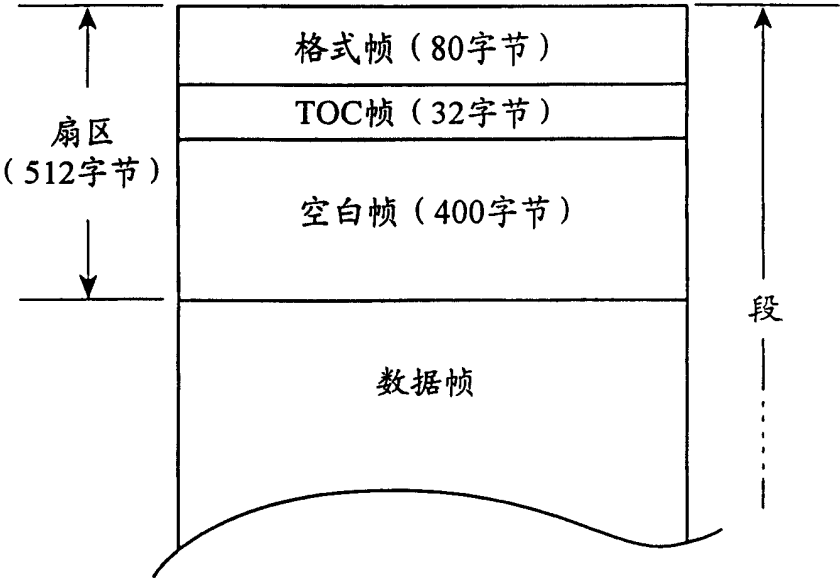


图 23

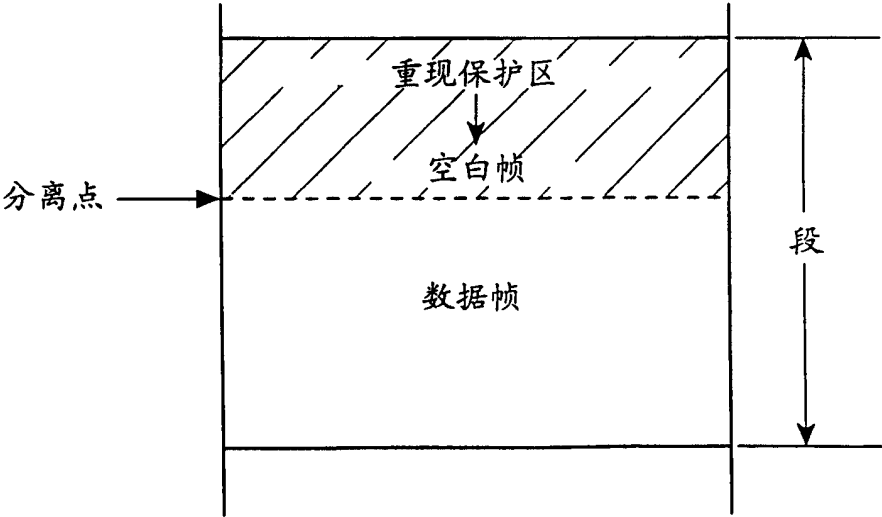


图 24

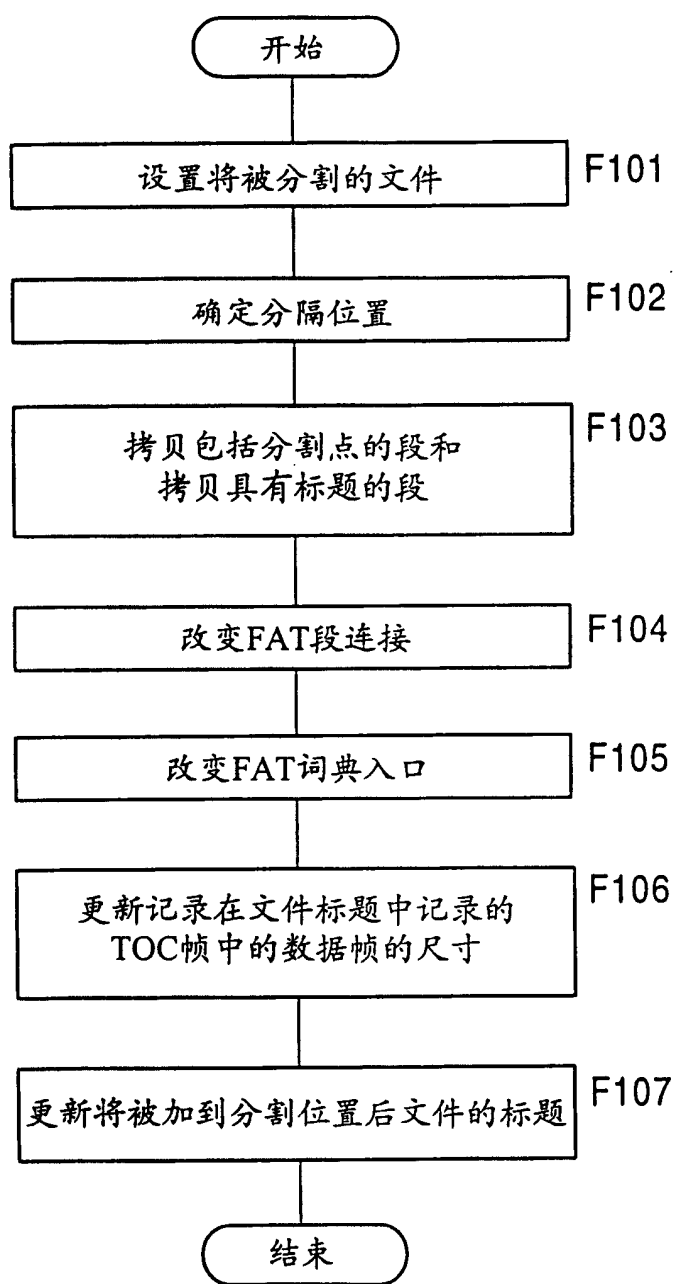
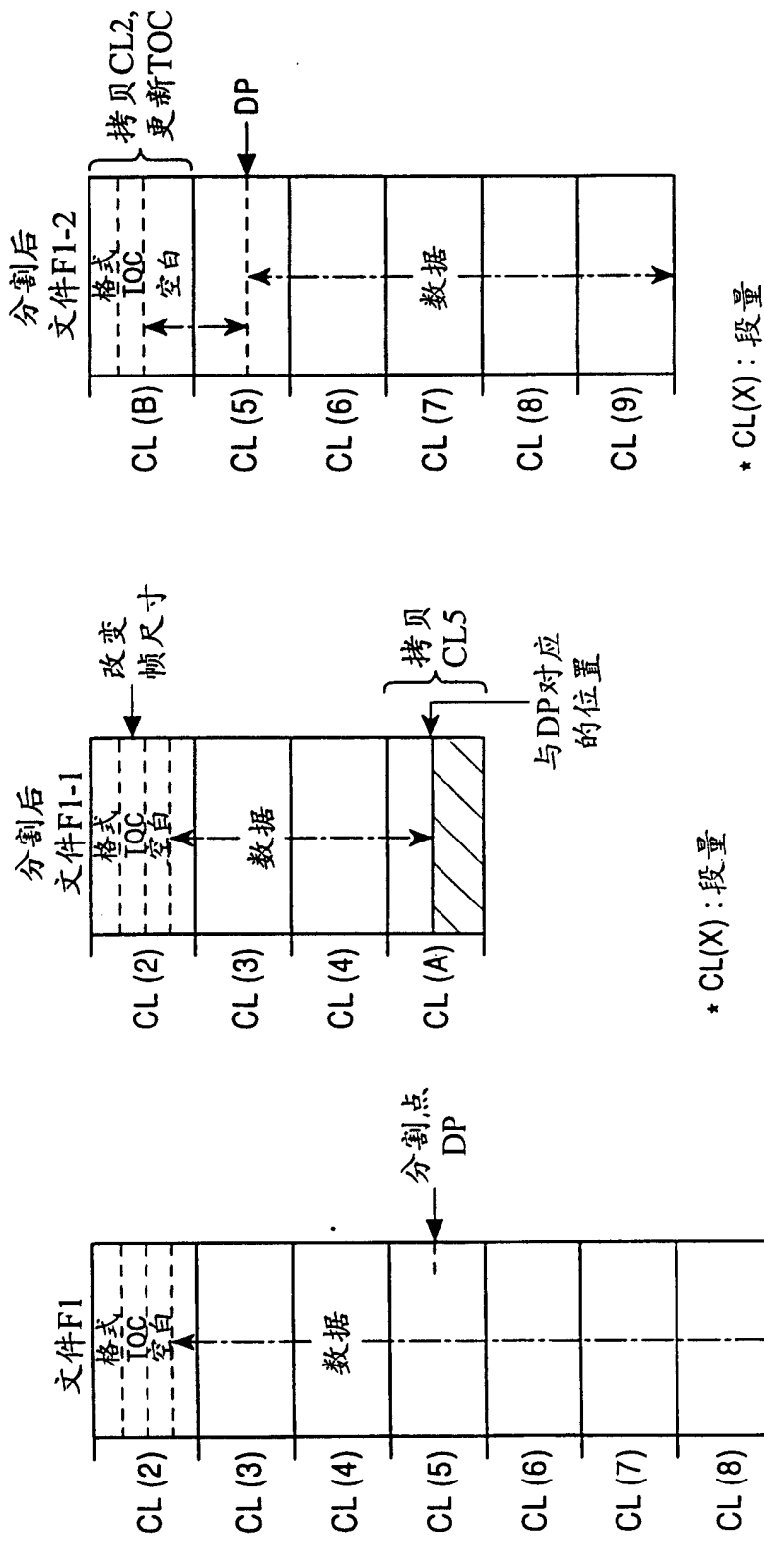


图 25



FAT (分割前)

(0)	(1)	(2) 003	(3) 004	(4) 005	(5) 006	(6) 007	(7) 008
(8) 009	(9) FFF	(A) 000	(B) 000	(C) 000	(D) 000	(E) 000	(F) 000
(10) 000	(11) 000	(12) 000	(13) 000	(14) 000	(15) 000	(16) 000	(17) 000
(18) 000	(19) 000	(1A) 000	(1B) 000	(1C) 000	(1D) 000	(1E) 000	(1F) 000
(CLmax-7) 000	(CLmax-6) 000	(CLmax-5) 000	(CLmax-4) 000	(CLmax-3) 000	(CLmax-2) 000	(CLmax-1) 000	(CLmax) 000

★ 16进制的对应的段号 ()

图 27

FAT (分割后)

(0)	(1)	(2) 003	(3) 004	(4) 00A	(5) 006	(6) 007	(7) 008
(8) 009	(9) FFF	(A) FFF	(B) 005	(C) 000	(D) 000	(E) 000	(F) 000
(10) 000	(11) 000	(12) 000	(13) 000	(14) 000	(15) 000	(16) 000	(17) 000
(18) 000	(19) 000	(1A) 000	(1B) 000	(1C) 000	(1D) 000	(1E) 000	(1F) 000
(CLmax-7) 000	(CLmax-6) 000	(CLmax-5) 000	(CLmax-4) 000	(CLmax-3) 000	(CLmax-2) 000	(CLmax-1) 000	(CLmax) 000

* 16进制的对应的段号 ()

图 28

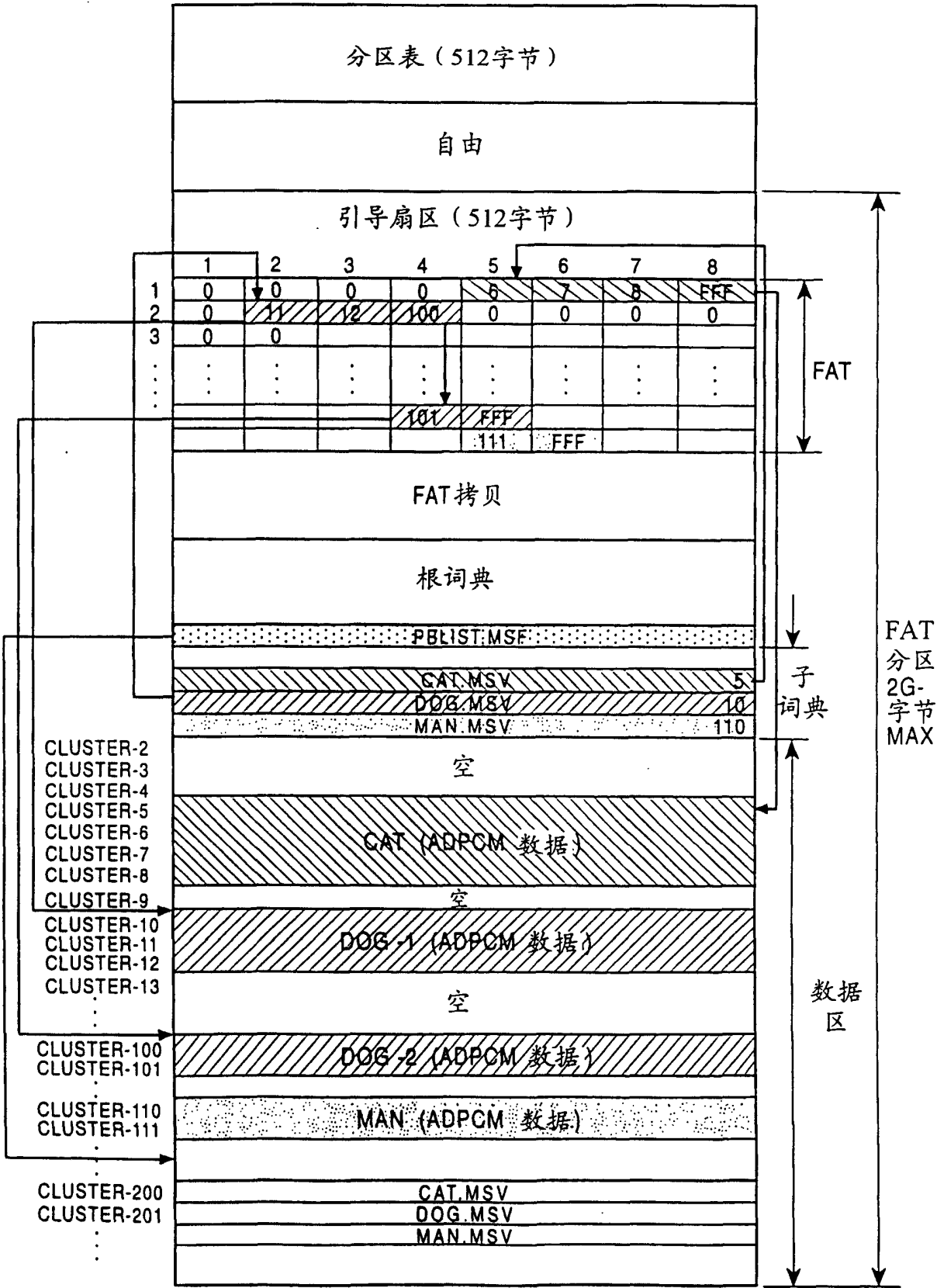


图 29

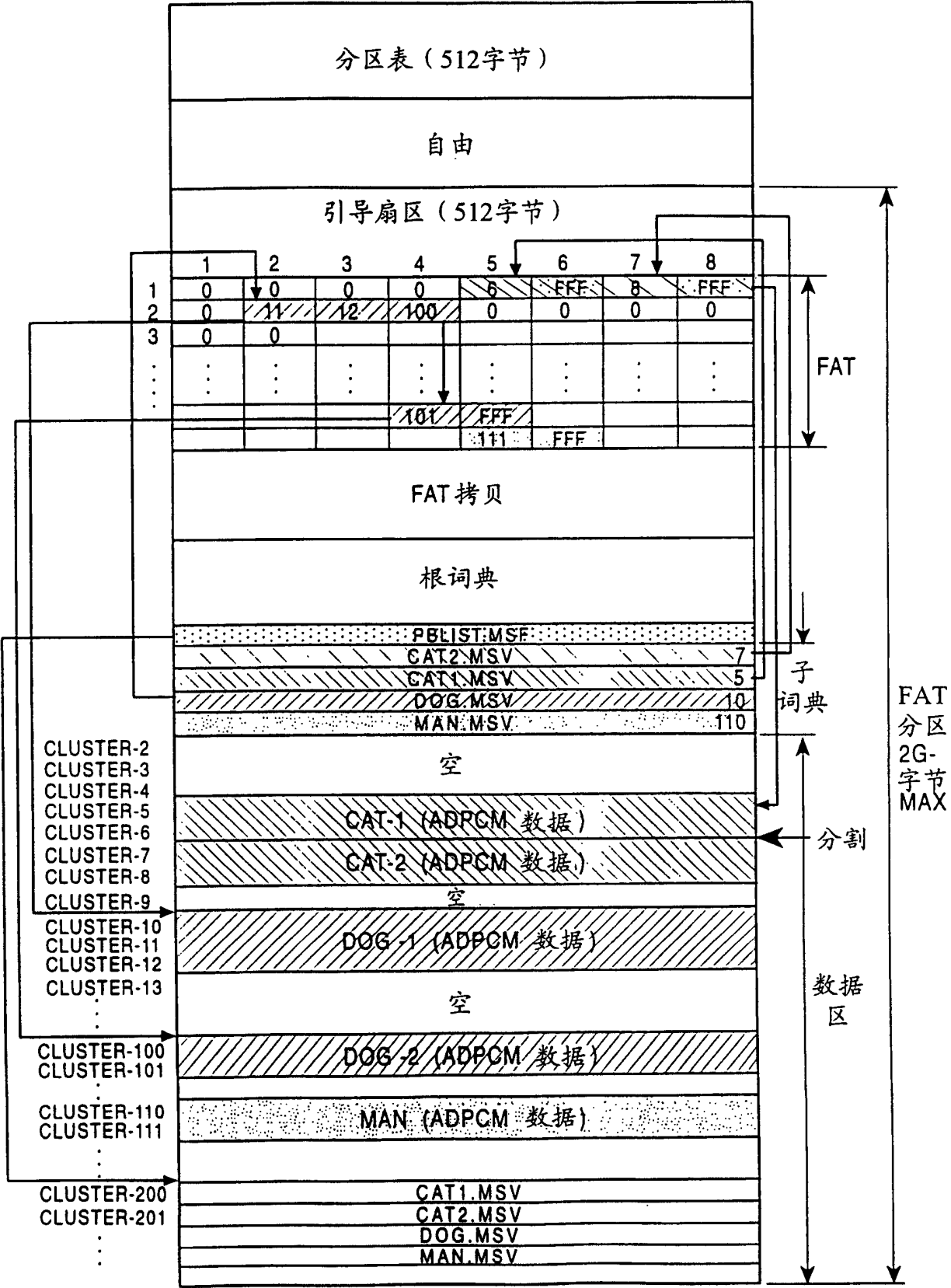


图 30

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	ID		(R)			尺寸			ID		(R)			尺寸		
0x0010	ID		(R)			尺寸			ID		(R)			尺寸		
0x0020	ID		(R)			尺寸			ALL 0 x 00							

图 31

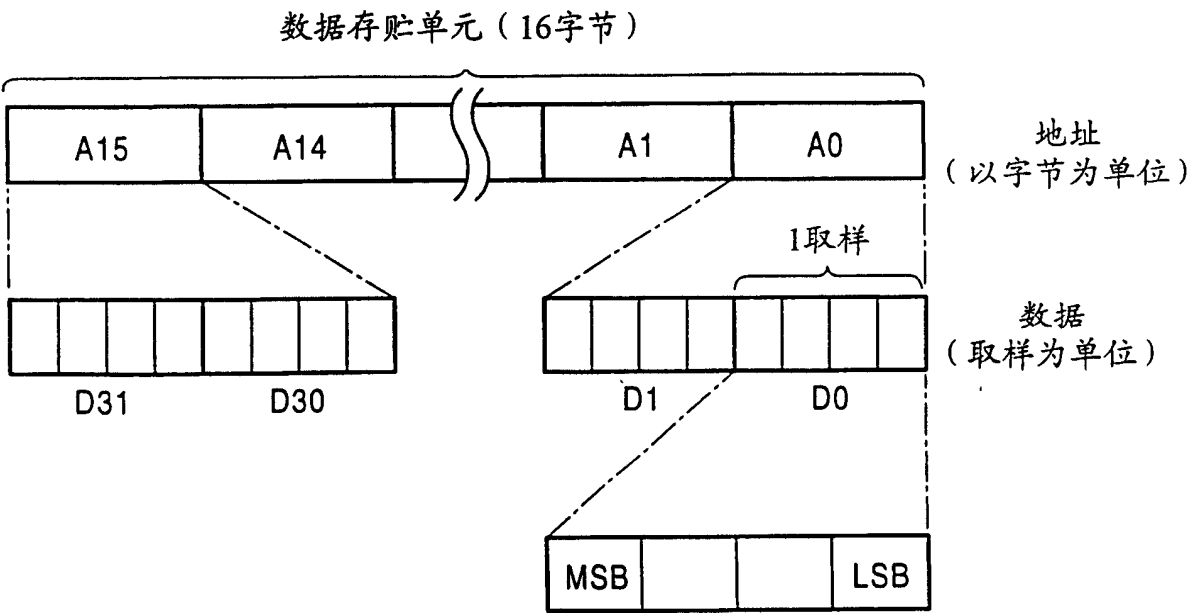


图 32

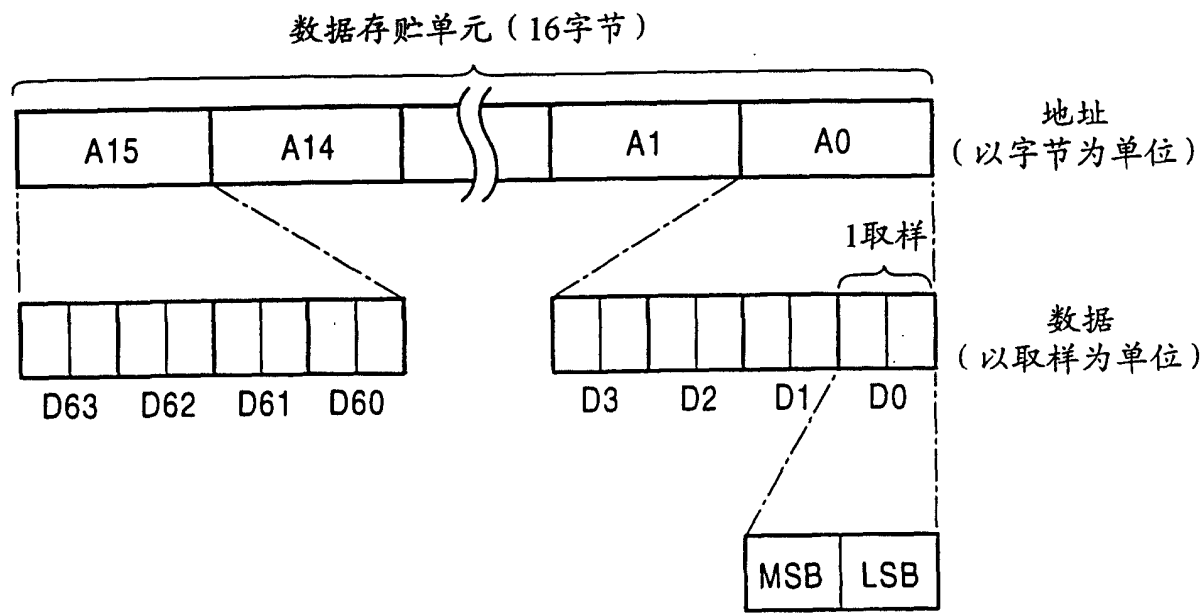


图 33

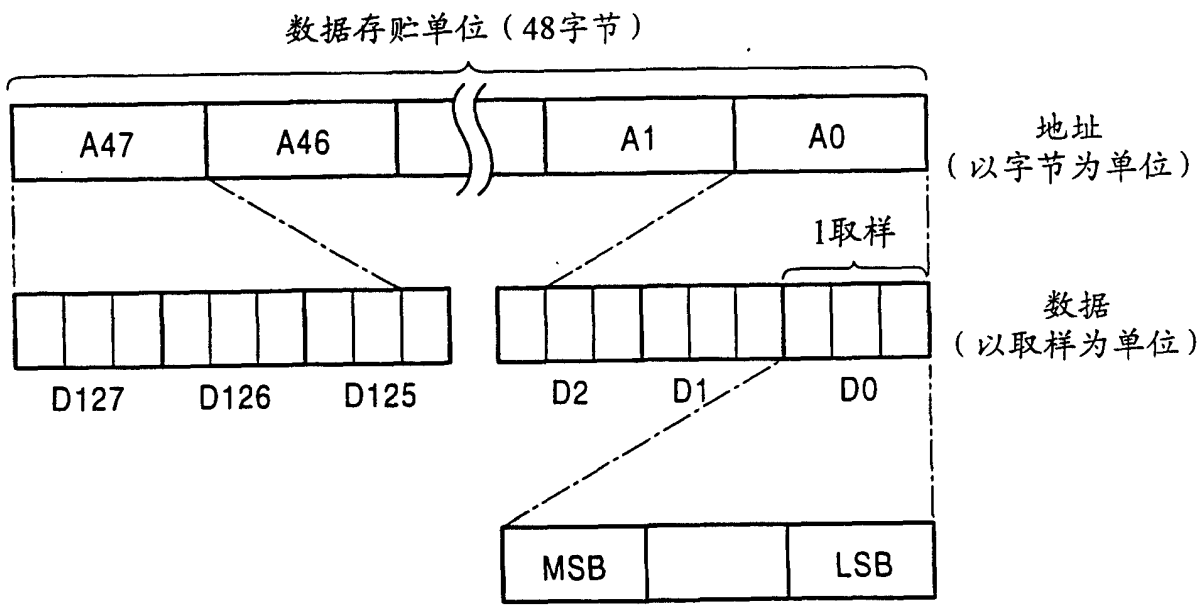


图 34