

(12) 发明专利

(10) 授权公告号 CN 101258473 B

(45) 授权公告日 2012.05.30

(21) 申请号 200680028404.5

(56) 对比文件

(22) 申请日 2006.08.01

US 2002/0184436 A1, 2002.12.05, 全文.

(30) 优先权数据

WO 2005/066793 A2, 2005.07.21, 说明书

60/705,388 2005.08.03 US

第0002,0003,0017,0018,0138,0141,0159,0177
段,附图12A.

(85) PCT申请进入国家阶段日

US 2003/0229753 A1, 2003.12.11, 说明书第
0012段.

2008.02.01

(86) PCT申请的申请数据

审查员 李芳

PCT/US2006/030242 2006.08.01

(87) PCT申请的公布数据

W02007/019220 EN 2007.02.15

(73) 专利权人 桑迪士克股份有限公司

地址 美国加利福尼亚州

(72) 发明人 艾伦·W·辛克莱 巴里·赖特

(74) 专利代理机构 北京律盟知识产权代理有限

责任公司 11287

代理人 刘国伟

(51) Int. Cl.

G06F 12/02(2006.01)

权利要求书 3 页 说明书 41 页 附图 31 页

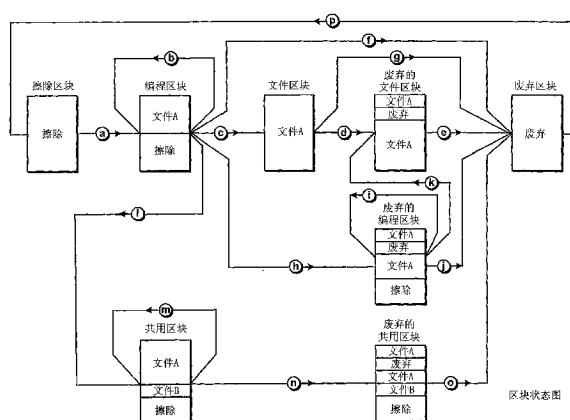
(54) 发明名称

直接数据文件存储存储器中的数据合并和垃圾收集

(57) 摘要

在不使用存储器的任何中间逻辑地址或虚拟地址空间的情况下,用每一文件的唯一识别和数据在所述文件内的偏移量将主机系统数据文件直接写入到大型擦除区块快闪存储器系统。通过所述存储器系统的控制器而不是通过主机将所述文件存储在所述存储器中哪个位置的目录信息维持在所述存储器系统内。

CN 101258473 B



1. 一种用于可重新编程的非易失性存储器系统的操作方法,所述存储器系统具有多个在将数据写入其中之前被分别擦除的存储器单元区块,且其以最小数目的准备好在其中存储数据的擦除区块的库存进行操作,所述方法包括:

接收通过唯一文件识别符和文件内的偏移量来逻辑寻址的数据;

将第一文件的所接收数据存储为所述擦除区块中的一者或一者以上内的页,所述页只部分填充所述擦除区块中的一者,进而在所述部分填充区块内留下擦除数据存储容量;以及

推迟将来自所述部分填充区块的有效数据与第二文件的有效数据合并到所述擦除区块的另一者中,直到至少认定所述数目的擦除区块的库存不足以维持所述最小数目为止。

2. 根据权利要求1所述的方法,其另外包括:响应于对提供另一擦除区块的需要,将来自所述第一文件和第二文件的有效数据合并到所述擦除区块的另一者中,且此后擦除至少所述部分填充区块,借此以将另一擦除区块添加到所述库存。

3. 根据权利要求1所述的方法,其另外包括:响应于接收到用以删除所述第一文件的命令,将所述第一文件在至少所述部分填充区块中的所有所述数据标记为废弃的,进而消除来自所述第一文件的在至少所述部分填充区块中的任何有效数据,借此不必对所述第一文件的在所述部分填充区块中的数据进行合并。

4. 根据权利要求1所述的方法,其另外包括:维持多个识别组成所述第一文件的可变量数据群组的记录,其中所述各个群组具有数据在所述群组内的连续逻辑偏移地址和连续物理地址两者。

5. 一种可重新编程的非易失性存储器系统的操作方法,所述存储器系统具有多个在将数据写入其中之前被分别擦除的存储器单元区块,且其接收具有唯一文件识别符的逻辑地址和各个文件内的偏移量的数据,其中:

不时地将来自第一群组的部分编程有两个或两个以上文件的数据的两个或两个以上区块的有效数据合并到另一区块中;

不时地对含有来自第二群组的还含有废弃数据的区块的有效数据的一个或一个以上区块进行垃圾收集;

一次只执行所述数据合并或垃圾收集中的一者;且

将优先级给予垃圾收集而不是数据合并。

6. 一种可重新编程的非易失性存储器系统的操作方法,所述存储器系统具有多个在将数据写入其中之前被分别擦除的存储器单元区块,且其接收具有唯一文件识别符的逻辑地址和各个文件内的偏移量的数据,其中:

以至少第一文件的数据可只部分填充第一区块且进而在所述第一区块中留下擦除存储容量的方式,将各个文件的所接收数据编程到一个或一个以上擦除区块中;

对所述存储器系统内的数据的后续操作导致第二文件的存储在第二区块中的至少一些数据变成废弃的;

响应于所述第二文件的在所述第二区块中的至少一些所述数据变成废弃的,将所述第二区块中的任何剩余有效数据复制到第三区块中;

响应于所述第一区块具有擦除存储容量,将有效数据从所述第一区块复制到第四区块中;且

将优先级给予上述将有效数据从所述第二区块复制到所述第三区块中,而不是上述将有效数据从所述第一区块复制到所述第四区块中。

7. 根据权利要求 6 所述的方法,其中所述第三或第四区块中的至少一者是向其中写入所述复制数据的擦除区块。

8. 根据权利要求 6 所述的方法,其中所述第三或第四区块中的至少一者在所述复制数据被写入其中的时候含有第三文件的数据。

9. 根据权利要求 6 所述的方法,另外其中维持多个识别组成所述各个文件的可变量数据群组的记录,其中所述群组分别具有数据在所述群组内的连续逻辑偏移地址和连续物理地址两者。

10. 一种可重新编程的非易失性存储器系统,其具有多个在将数据写入其中之前被分别擦除的存储器单元区块,其中:

维持最小数目的准备好在其中存储数据的擦除区块的库存;

通过将第一文件的所接收数据存储为所述擦除区块中的一者或一者以上内的页而将通过唯一文件识别符和文件内的偏移量在逻辑上寻址的文件的数据存储在所述存储器区块中,其中所述页只部分填充所述擦除区块中的一者,进而在所述部分填充区块内留下擦除数据存储容量;且

推迟将来自所述部分填充区块的有效数据与第二文件的有效数据合并到所述擦除区块的另一者中,直到至少认定所述数目的擦除区块的库存不足以维持所述最小数目为止。

11. 根据权利要求 10 所述的存储器系统,进一步其中响应于对提供另一擦除区块的需要,将来自所述第一文件和第二文件的有效数据合并到所述擦除区块的另一者中,且此后擦除至少所述部分填充区块,进而将另一擦除区块添加到所述库存。

12. 根据权利要求 10 所述的存储器系统,进一步其中响应于接收到用以删除所述第一文件的命令,将所述第一文件的在至少所述部分填充区块中的所有所述数据标记为废弃的,进而消除来自所述第一文件的在至少所述部分填充区块中的任何有效数据,借此不必对所述第一文件的在所述部分填充区块中的数据进行合并。

13. 一种可重新编程的非易失性存储器系统,其具有多个在将数据写入其中之前被分别擦除的存储器单元区块,其中:

接受具有唯一文件识别符的逻辑地址和各个文件内的偏移量的数据;

不时地将来自第一群组的部分编程有两个或两个以上文件的数据的两个或两个以上区块的有效数据合并到另一区块中;

不时地对含有来自第二群组的还含有废弃数据的区块的有效数据的一个或一个以上区块进行垃圾收集;

一次只执行所述数据合并或垃圾收集中的一者;且

将优先级给予垃圾收集而不是数据合并。

14. 一种可重新编程的非易失性存储器系统,其具有多个在将数据写入其中之前被分别擦除的存储器单元区块,其中:

接受具有唯一文件识别符的逻辑地址和各个文件内的偏移量的数据;

以至少第一文件的数据可只部分填充第一区块且进而在所述第一区块中留下擦除存储容量的方式,将各个文件的所接收数据编程到一个或一个以上擦除区块中;

对所述存储器系统内的数据的后续操作导致第二文件的存储在第二区块中的至少一些数据变成废弃的；

响应于所述第二文件的在所述第二区块中的至少一些所述数据变成废弃的，将所述第二区块中的任何剩余有效数据复制到第三区块中；

响应于所述第一区块具有擦除存储容量，将有效数据从所述第一区块复制到第四区块中；且

将优先级给予上述将有效数据从所述第二区块复制到所述第三区块中，而不是上述将有效数据从所述第一区块复制到所述第四区块中。

15. 根据权利要求 14 所述的存储器系统，其中在所述第三或第四区块中的至少一者是擦除区块时将所述复制数据写入到所述第三或第四区块中的所述至少一者中。

16. 根据权利要求 14 所述的存储器系统，其中在所述复制数据被写入其中的时候，第三文件的数据包含在所述第三或第四区块中的至少一者中。

直接数据文件存储器中的数据合并和垃圾收集

技术领域

[0001] 本申请案大体上涉及例如半导体快闪存储器等可重新编程的非易失性存储器系统的操作,其中包含管理主机装置与存储器系统之间的接口,且更具体地说,涉及有效地使用数据文件接口而不是常见的大容量存储器逻辑地址空间 (LBA) 接口。

背景技术

[0002] 本文中描述在待决的第 11/060,174 号、第 11/060,248 号和第 11/060,249 号美国专利申请案中描述的快闪存储器的各种操作方面的发展,所述申请案全部在 2005 年 2 月 16 日以 Alan W. Sinclair 一人或其与 Peter J. Smith 两人名义申请(下文中称为“现有申请案”)。

[0003] 进一步的发展在 Alan W. Sinclair 和 Barry Wright 的相关美国专利申请案中描述,即第 11/382,224 号和第 11/382,232 号非临时申请案以及第 60/746,740 号和第 60/746,742 号临时申请案,所述申请案全部在 2006 年 5 月 8 日申请;以及题为“Indexing Of File Data In Reprogrammable Non-Volatile Memories That Directly Store Data Files”、“Reprogrammable Non-Volatile Memory Systems With Indexing of Directly Stored Data Files”、“Methods of Managing Blocks in Nonvolatile Memory” 和“Nonvolatile Memory With Block Management”的非临时申请案,所述申请案全部在 2006 年 7 月 21 日申请。

[0004] 本文中引用的所有专利、专利申请案、文章及其它出版物、文献以及资料出于所有目的在此以引用的方式全文并入本文中。就任何并入的出版物、文献或资料中的任一者与本申请案之间在术语定义或使用方面的任何不一致或冲突而言,应以本申请案的为主。

发明内容

[0005] 本文中将数据合并与垃圾收集区别对待,且至少部分地通过不同的算法来实施所述两种处理。当文件或存储器区块中含有废弃数据时,利用垃圾收集操作来将所述文件或区块的有效数据移动到一个或一个以上其它区块。这种操作将有效数据聚集到数目较少的区块中,因而一旦擦除原始来源区块便释放了废弃数据所占据的容量。在数据合并中,将一个部分填充区块(例如通常因写入新文件而产生)的有效数据与另一部分填充区块的有效数据组合。作为数据来源的原始区块中的一者或两者现在含有废弃的重复数据,于是将其调度以进行垃圾收集。虽然提供多个队列以用于调度各个垃圾收集操作以恢复由废弃数据占据的存储器存储容量,但优选地在未调度任何垃圾收集且以其它方式满足了合并条件时发生数据合并。

[0006] 不是在关闭新近写入的文件之后立刻调度对所述文件的数据合并,而是将新近写入的文件的数据维持在其从主机接收之后被编程到的原始区块中。这最常见的是包含由新数据部分填充的区块。由于常见的是用会形成废弃数据的方式删除或更新数据文件,所以在关闭文件之后尽可能长期地拖延对部分填充区块的数据合并。可在不需要重新定位任何

数据的情况下将所述文件删除。因此,如果在此种合并变得必要之前删除或更新文件,那么会避免合并。当存储器变满时此种合并可能变得必要,以使得具有足够的擦除区块用于对新数据进行进一步编程。但是因为基于文件的存储器不保持已经被主机删除的数据文件,所以与使用逻辑接口的情况相反,存储器将通常具有充分数目的擦除区块,即使合并被延迟也是如此。因此节省了省略的合并所花费的时间,且因而改进了存储器的性能。

- [0007] 在直接文件存储中存在下述的若干其它发展,可概述如下:
- [0008] 1. 一旦关闭文件,不将其添加到文件垃圾收集队列,除非其含有废弃数据。
- [0009] 2. 对文件的垃圾收集不会产生含有来自另一文件的数据的共用区块。将文件的必须在垃圾收集期间复制的数据编程到所述文件的当前编程区块。编程区块在垃圾收集结束时保持部分编程。
- [0010] 3. 当因区块中的文件群组由于删除文件而变得废弃而必须从共用区块中重新定位数据时,将剩余的有效数据重新定位到编程区块的可用空间中。
- [0011] 4. 避免移动直接写入到编程区块的全部或一部分中的主机数据。
- [0012] 5. 在文件的垃圾收集期间,将数据重新定位到文件的编程区块。没有专门的中间复制区块。
- [0013] 6. 用完整的数据扇区在存储器系统控制器缓冲存储器与存储器单元阵列之间转移数据。这允许在编程期间产生 ECC 且在数据读取期间检验 ECC。
- [0014] 7. 将共用区块中的数据群组或文件群组的开头与元页的开头对准。可因此将芯片上复本用于区块合并。编程区块中的数据群组与物理结构之间没有特定对准关系。
- [0015] 8. 快闪存储器内的交换区块用来针对不活动的打开文件(也就是说,当最近写入命令涉及到不同文件时)对保持在非易失性控制器缓冲存储器中的数据进行安全复制。其还可用作虚拟缓冲器结构的一部分,以便通过在二者之间使用交换操作来允许可用的缓冲存储器容量支持较多打开文件。
- [0016] 9. 当因为 FIT 文件的当前范围溢出而将所述 FIT 文件移动到另一 FIT 范围时,更新目录中的文件数据指针以便反映新的 FIT 范围。
- [0017] 10. 当 FIT 更新区块中的一个 FIT 范围的数据量超过阈值时,将所述范围的 FIT 更新区块中的数据与所述范围的 FIT 区块中的数据合并。这允许将新文件的数据合并到 FIT 区块。
- [0018] 11. 在对 FIT 更新区块进行压缩期间,如果存在充分的已擦除空间,那么将关闭的文件的 FIT 文件重新定位到其范围的 FIT 区块。否则,将其重新定位到压缩的 FIT 更新区块。
- [0019] 12. 主机可使用写入 _ 指针和读取 _ 指针命令来控制一个组中的所有文件而使其具有相等大小(与元区块的大小相同),且可使用关闭和闲置命令来致使所述组中的文件在文件关闭之后立刻被合并到单个元区块中。
- [0020] 13. 所述主机命令集包含针对指定文件 ID 的读取和写入命令,其中包含针对写入 _ 指针和读取 _ 指针的值的伙伴命令,所述伙伴命令提供拟开始所命令的数据写入或读取的存储器地址。

附图说明

- [0021] 包含以下列举的图式作为本申请案的一部分,且在以下描述中参看所述图式:
- [0022] 图 1-1 :具有直接数据文件平台的存储卡;
- [0023] 图 1-2 :直接数据文件平台组成部分;
- [0024] 图 2-1 :文件命令;
- [0025] 图 2-2 :数据命令;
- [0026] 图 2-3 :信息命令;
- [0027] 图 2-4 :流命令;
- [0028] 图 2-5 :状态命令;
- [0029] 图 2-6 :装置命令;
- [0030] 图 3-1 :普通文件的格式;
- [0031] 图 3-2 :共用文件的格式;
- [0032] 图 3-3 :已编辑的普通文件的格式;
- [0033] 图 3-4 :已编辑的共用文件的格式;
- [0034] 图 4-1 :用于装置操作的流程图;
- [0035] 图 5-1 :用于编程文件数据的流程图;
- [0036] 图 6-1 :用于读取文件数据的流程图;
- [0037] 图 7-1 :用于删除文件的流程图;
- [0038] 图 8-1 :用于前台垃圾收集的交错操作;
- [0039] 图 8-2 :用于垃圾收集的自适应调度的操作原理;
- [0040] 图 8-3 :用于垃圾收集选择的流程图;
- [0041] 图 8-4(包括图 8-4A 到 8-4B) :用于文件垃圾收集的流程图;
- [0042] 图 8-5(包括图 8-5A 到 8-5B) :用于共用区块垃圾收集的流程图;
- [0043] 图 8-6(包括图 8-6A 到 8-6B) :用于区块合并的流程图;
- [0044] 图 8-7A 到 8-7D :共用区块垃圾收集实例,其展示四个时序阶段;
- [0045] 图 9-1 :连续的主机数据编程;
- [0046] 图 9-2 :中断的主机数据编程;
- [0047] 图 9-3 :缓冲器清洗编程;
- [0048] 图 9-4 :缓冲器换出编程;
- [0049] 图 9-5 :缓冲器清洗之后的主机数据编程;
- [0050] 图 9-6 :换入数据读取;
- [0051] 图 9-7 :缓冲器换入之后的主机数据编程;
- [0052] 图 9-8 :对缓冲器的对准的数据读取;
- [0053] 图 9-9 :从缓冲器的对准的数据编程;
- [0054] 图 9-10 :对缓冲器的非对准的数据读取;
- [0055] 图 9-11 :从缓冲器的非对准的数据编程;
- [0056] 图 9-12 :对缓冲器的非对准且非循序的数据读取;
- [0057] 图 9-13 :从缓冲器的非对准且非循序的数据编程;
- [0058] 图 10-1 :文件索引;
- [0059] 图 10-2 :文件索引结构;

- [0060] 图 10-3 :目录区块格式 ;
- [0061] 图 10-4 :文件索引表 (FIT) 逻辑结构 ;
- [0062] 图 10-5 :FIT 页格式 ;
- [0063] 图 10-6 :物理 FIT 区块 ;
- [0064] 图 10-7 :FIT 文件更新操作的实例 ;
- [0065] 图 11-1 :区块状态图 ;
- [0066] 图 12-1 :控制区块格式 ;
- [0067] 图 12-2 :常用区块日志格式 ;以及
- [0068] 图 13-1(包括图 13-1A 到 13-1B) :与静态文件一起使用的命令集 (在应当连在一起的部分 A 和 B 中)。
- [0069] 具体实施方式
- [0070] 1. 直接数据文件平台
- [0071] 1.1 概述
 - [0072] 在图 1-1 中说明具有直接数据文件平台的存储卡。直接数据文件平台是用文件形式组织的数据存储装置,其中通过文件名和文件偏移地址来识别数据。其充当存储卡中的存储平台,所述平台可并入除数据存储之外的功能。通过外部文件接口通道在平台中存取文件数据。
 - [0073] 存储装置没有逻辑地址。针对每一文件存在独立的地址空间,且存储器管理算法根据数据的文件结构来组织数据存储。与将常规的文件系统与常规的逻辑分区存储器管理集成的文件存储装置的操作特性相比,在直接数据文件平台中采用的数据存储组织会产生操作特性的显著改进。
- [0074] 1.2 平台组成部分
 - [0075] 直接数据文件平台具有以下组成部分,其如图 1-2 所示那样以功能性层构造 :
 - [0076] 直接数据文件接口 :文件 API,其提供来自所述卡中的其它功能区块对通过文件名和文件偏移地址识别的数据的存取。
 - [0077] 文件到快闪映射算法 :用于消除文件分裂并提供最大性能和耐久性的以文件形式组织的数据存储的方案。
 - [0078] 编程文件数据 :根据文件到快闪映射算法来编程文件数据。
 - [0079] 读取文件数据 :从快闪存储器中读取由文件偏移地址指定的数据。
 - [0080] 删除文件 :识别含有已删除文件的数据的区块,并将所述区块添加到垃圾收集队列。
 - [0081] 垃圾收集 :经执行以恢复废弃数据所占据的存储器容量的操作。这些可能需要将有效数据复制到另一位置,以便擦除区块。
 - [0082] 文件索引 :文件索引允许按偏移地址次序识别文件的有效数据群组的位置。
 - [0083] 数据缓冲与编程 :对于拟编程的数据使用缓冲存储器,以及对编程区块中的文件数据进行编程的序列。
 - [0084] 擦除区块管理 :对于装置中可供分配以用于存储文件数据或控制信息的擦除区块集
 - [0085] 区块状态管理 :用于存储文件数据的区块可被分类成的八种状态之间的转变。

- [0086] 控制数据结构 :存储在专用于所述用途的快闪区块中的控制数据结构。
- [0087] 2. 直接数据文件接口
- [0088] 直接数据文件接口是对直接数据文件平台的 API, 其形成用于并入有快闪大容量数据存储设备的装置内的快闪存储器管理的后端系统。
- [0089] 2.1 命令集
- [0090] 以下部分定义用以支持与多种来源的基于文件的介接的一般命令集。用六种类别定义命令。
- [0091] 1. 文件命令
- [0092] 2. 数据命令
- [0093] 3. 文件信息命令
- [0094] 4. 流命令 (只用于建模)
- [0095] 5. 状态命令
- [0096] 6. 装置命令
- [0097] 2.1.1 文件命令 (见图 2-1)
- [0098] 文件是装置内通过文件 ID 独立识别的对象。文件可包括由主机创建的一组数据, 或者可能没有任何数据, 在此情况下, 其代表目录或文件夹。
- [0099] 2.1.1.1 创建
- [0100] 创建命令在装置中的目录内创建通过 <文件 ID> 识别的条目。如果省略 <文件 ID> 参数, 那么装置向文件指派可用值并将所述值返回到主机。这是创建文件的正常方法。
- [0101] 主机可替代地向文件指派 <文件 ID> 值。如果文件 ID 的特定值指示主机接口协议内的特定类型的文件, 那么可使用这种方法。举例来说, 主机可向根目录指派特定的文件 ID。
- [0102] 2.1.1.2 打开
- [0103] 这个命令启用对 <文件 ID> 指定的文件执行后续的数据命令。如果文件不存在, 那么返回错误消息。将所述文件的写入 _ 指针设置到文件末尾, 且将所述文件的读取 _ 指针设置到文件开头。将文件 _ 信息的信息 _ 写入 _ 指针设置到文件 _ 信息末尾, 且将所述文件的信息 _ 读取 _ 指针设置到文件 _ 信息开头。存在可同时打开的文件的最大数目。如果超过这个数目, 那么不执行所述命令, 且返回错误消息。举例来说, 同时打开的文件的最大数目可以是 8。
- [0104] 只有在接收到后续写入、插入或移除命令之后, 才使得装置内用于写入到指定文件的资源可用。
- [0105] 2.1.1.3 关闭
- [0106] 这个命令停用对指定文件的后续数据命令的执行。所述文件的写入 _ 指针、读取 _ 指针、信息 _ 写入 _ 指针和信息 _ 读取 _ 指针值变得无效。
- [0107] 2.1.1.4 删除
- [0108] 删除命令指示应当删除 <文件 ID> 指定的文件的目录、文件索引表和信息表条目。可擦除所述文件的数据。随后可能不存取已删除的文件。
- [0109] 2.1.1.5 擦除
- [0110] 擦除命令指示应当删除 <文件 ID> 指定的文件的目录、文件索引表和信息表条目。

必须在可执行任何其它命令之前擦除文件数据。随后可能不存取已擦除的文件。

[0111] 2.1.1.6 列举_文件

[0112] 在接收到列举_文件命令之后,可按照数字次序将目录中的所有文件的文件 ID 值从装置中流出。当到达最后文件时终止文件 ID 流动,且可由主机借助于状态命令来识别这个条件。通过接收到其它任何命令来终止列举_文件命令。

[0113] 2.1.2 数据命令(见图 2-2)

[0114] 使用数据命令来起始对指定文件的数据输入和输出操作,并定义文件内的偏移地址值。指定的文件必须已经被主机打开。如果不是这样的话,那么返回错误。<文件 ID>是在文件最后被打开时返回到主机的文件句柄。

[0115] 2.1.2.1 写入

[0116] 在接收到写入命令之后流到装置的数据复写在由写入_指针的当前值定义的偏移地址处的指定文件中。使用写入命令来写入文件的新数据、将数据附加到文件和更新文件内的数据。通过接收到其它任何命令来终止写入命令。

[0117] 2.1.2.2 插入

[0118] 将在接收到插入命令之后流到装置的数据插入在写入_指针的当前值定义的偏移地址处的指定文件中。文件大小增加了插入数据的长度。通过接收到其它任何命令来终止插入命令。

[0119] 2.1.2.3 移除

[0120] 移除命令从写入_指针的当前值定义的偏移地址处的指定文件中删除<长度>定义的循序数据。文件大小减少了<长度>。

[0121] 2.1.2.4 读取

[0122] 可在接收到读取命令之后从装置中流出由读取_指针的当前值定义的偏移地址处的指定文件中的数据。

[0123] 当达到文件结尾时终止数据流动,且可由主机借助于状态命令来识别这个条件。通过接收到其它任何命令来终止读取命令。

[0124] 2.1.2.5 保存_缓冲器

[0125] 指定文件的包含在装置缓冲器中但尚未编程到快闪存储器的数据保存在快闪存储器中的临时位置处。

[0126] 当接收到后续写入或插入命令时将数据复原到缓冲器,并将其同与命令有关的数据一起编程到快闪。

[0127] 2.1.2.6 写入_指针

[0128] 写入_指针命令将指定文件的写入_指针设置到指定的偏移地址。当在写入或插入命令之后将数据流到装置时,装置使写入_指针递增。

[0129] 2.1.2.7 读取_指针

[0130] 读取_指针命令将指定文件的读取_指针设置到指定的偏移地址。当在读取命令之后从装置中流出数据时,装置使读取_指针递增。

[0131] 2.1.3 信息命令(见图 2-3)

[0132] 文件_信息是主机产生的与文件相关联的信息。文件_信息的性质和内容由主机确定,且其不由装置解译。使用信息命令来起始对指定文件的文件_信息输入和输出操作,

并定义文件_信息内的偏移地址值。

[0133] 2.1.3.1 写入_信息

[0134] 在接收到写入_信息命令之后流到装置的文件_信息复写由信息_写入_指针的当前值定义的偏移地址处的指定文件的文件_信息。指定文件的文件_信息的内容和长度由主机确定。通过接收到其它任何命令来终止写入_信息命令。

[0135] 2.1.3.2 读取_信息

[0136] 可在接收到读取_信息命令之后从装置中流出信息_读取_指针的当前值定义的偏移地址处的指定文件的文件_信息。当达到文件_信息的末尾时终止文件_信息流动，且可由主机借助于状态命令来识别这个条件。通过接收到其它任何命令来终止读取_信息命令。

[0137] 2.1.3.3 信息_写入_指针

[0138] 信息_写入_指针命令将指定文件的信息_写入_指针设置到指定偏移地址。当在写入_信息命令之后将文件_信息流到装置时，装置使信息_写入_指针递增。

[0139] 2.1.3.4 信息_读取_指针

[0140] 信息_读取_指针命令将指定文件的信息_读取_指针设置到指定的偏移地址。当在读取_信息命令之后从装置中流出文件_信息时，装置使信息_读取_指针递增。

[0141] 2.1.4 流命令（见图 2-4）

[0142] 流命令只与直接数据文件平台的行为建模一起使用。其用途是结合数据命令模拟去往和来自主机的流动数据。

[0143] 2.1.4.1 流

[0144] 流命令模拟由<长度>定义的应由主机传递到平台或从平台传递的不间断的数据流。当向缓冲存储器添加数据或从中移除数据时，通过平台的模型将表示流的剩余长度的变量递减。

[0145] 2.1.4.2 暂停

[0146] 暂停命令插入长度为<时间>的延迟，所述延迟在执行命令列表中的正在控制直接数据文件模型的操作的随后命令之前插入。<时间>以微秒定义。

[0147] 2.1.5 状态命令（见图 2-6）

[0148] 状态命令控制装置的状态。

[0149] 2.1.5.1 闲置

[0150] 闲置命令指示主机正在将直接数据文件装置置于闲置状态，在此期间，装置可执行内部内务处理操作。主机将不会在闲置状态下故意从装置中移除电力。不论装置是否忙于内部操作，均可通过由主机传输其它任何命令来结束闲置状态。一旦接收到此其它命令，装置中的任何正在进行的内部操作均必须暂停或在指定时间内终止。此时间的一个实例是 10 毫秒或更短。

[0151] 2.1.5.2 待命

[0152] 待命命令指示主机正在将直接数据文件装置置于待命状态，在此期间，装置可以不执行内部内务处理操作。主机将不会在待命状态下故意从装置中移除电力。可通过由主机传输其它任何命令来结束待命状态。

[0153] 2.1.5.3 关机

[0154] 关机命令指示当装置接下来不处于繁忙状态时,将通过主机从装置中移除电力。装置响应于关机命令而关闭所有打开的文件。

[0155] 2.1.6 装置命令(见图 2-6)

[0156] 装置命令允许主机询问装置。

[0157] 2.1.6.1 容量

[0158] 响应于容量命令,装置报告存储在装置中的文件数据的容量以及可用于新文件数据的容量。

[0159] 2.1.6.2 状态

[0160] 响应于状态命令,装置报告其当前状态。

[0161] 状态包含三种类型的繁忙状态:

[0162] 1. 装置忙于执行用于写入或读取数据的前台操作。

[0163] 2. 装置忙于执行在装置处于闲置状态时起始的后台操作。

[0164] 3. 缓冲存储器繁忙,且不可供主机用于写入或读取数据。

[0165] 2.1.7 命令参数

[0166] 以下参数与下文定义的命令一起使用。

[0167] 2.1.7.1 文件 ID

[0168] 这是用来识别装置的目录内的文件的文件识别符。

[0169] 2.1.7.2 偏移量

[0170] 偏移量是文件或文件_信息内的相对于所述文件或文件_信息开头的以字节计的逻辑地址。

[0171] 2.1.7.3 长度

[0172] 这是文件的一连串具有循序偏移地址的数据的以字节计的长度。

[0173] 2.1.7.4 时间

[0174] 这是以微秒计的时间。

[0175] 3. 文件到快闪映射算法

[0176] 由直接数据文件平台采用的文件到快闪映射算法是一种用于文件组织的数据存储设备的新机制,所述机制经定义以在主机经由基于文件的接口执行文件数据写入和文件删除操作时提供最大系统性能和最大存储器耐久性。所述映射算法已经设计以将快闪存储器中的区块之间的文件数据复制减到最少。通过以下方式来实现这点:以实现含有一个以上文件的数据的区块的最低发生几率的方式将文件数据映射到快闪区块。

[0177] 3.1 文件到快闪映射原理

[0178] 3.1.1 文件

[0179] 文件是主机创建和维持的一组数据。数据由主机通过文件名来识别,且可通过其偏移位置从文件开始处存取。文件偏移地址可通过主机设置,且可通过装置作为写入指针来递增。

[0180] 3.1.2 物理存储器结构

[0181] 直接数据文件平台在固定大小的元区块中存储文件的所有数据。组成元区块的快闪擦除区块的实际数目(即擦除区块并行度)可在产品之间有所不同。在整个本说明书中,术语“区块”始终用于指示“元区块”。

[0182] 使用术语“元页”来指示具有元区块的完整并行度的页。元页是最大的编程单位。
[0183] 使用术语“页”来指示存储器的平面内的页，也就是说，快闪擦除区块内的页。页是最小的编程单位。

[0184] 使用术语“扇区”来指示与 ECC 相关联的存储数据的单位。扇区是向快闪存储器传递数据或从中传递数据的最小单位。

[0185] 在文件的偏移地址与物理快闪存储器结构之间并未维持任何指定的对准。

[0186] 3.1.3 数据群组

[0187] 数据群组是在文件内具有连续偏移地址的一组文件数据，其在单个存储器区块中以连续的物理地址编程。文件通常将被编程为多个数据群组。数据群组可具有介于一个字节到一个区块之间的任何长度。每一数据群组编程有一个标头，所述标头中含有文件识别符信息以用于交叉参考用途。文件的数据根据其包括的数据群组来在物理存储器中索引。文件索引表提供文件的每一数据群组的文件偏移地址和物理地址信息。

[0188] 3.1.4 编程区块

[0189] 文件必须由主机打开以允许编程文件数据。每一打开的文件具有分派为编程区块的专用区块，且所述文件的数据在由编程区块内的编程指针定义的位置处编程。当文件由主机打开时，文件的编程区块被打开（如果不是已经存在一个编程区块的话）。将编程指针设置到编程区块的开始处。如果对于由主机打开的文件已经存在一个编程区块，那么其继续用于编程所述文件的数据。

[0190] 将文件数据以其从主机处接收的次序编程在编程区块中，不论其在文件内的偏移地址如何或所述偏移地址的数据先前是否已被编程。当编程区块变满时，其称为文件区块，且来自擦除区块集区的擦除区块作为新的编程区块打开。在存储文件的数据的区块之间没有任何物理地址关系。

[0191] 3.1.5 共用区块

[0192] 共用区块中含有一个以上文件的数据群组。如果同一文件的多个数据群组存在于共用区块中，那么其被连续定位且所述连续单位称为文件群组。只在区块合并操作或共用区块垃圾收集操作期间将数据编程到共用区块。

[0193] 共用区块内的各个数据群组或文件群组的开头必须与元页的开头对准。

[0194] 文件群组内的数据群组不具有插入空间。此类数据群组之间的边界可发生在页内。文件应当仅在单个共用区块中具有数据（见 8.3.4，其是这种情况的例外）。

[0195] 3.2 文件类型

[0196] 3.2.1 普通文件

[0197] 普通文件包括任何数目的完整文件区块和一个部分编程的编程区块。可通过通常以循序的偏移地址次序编程来自主机的文件的数据或通过对已编辑文件的垃圾收集来创建普通文件。图 3-1 中展示实例性普通文件。普通文件可以是打开的文件或关闭的文件。

[0198] 可在编程区块中的编程指针处编程文件的其它数据。如果主机将文件删除，那么可立刻将含有其数据的区块擦除，而无需将数据从此类区块复制到快闪存储器中的另一位置。因此，普通文件格式是非常高效的，且存在尽可能长久地以这种格式保持文件的优点。

[0199] 3.2.2 共用文件

[0200] 共用文件包括任何数目的完整文件区块和一个共用区块，所述共用区块含有所述

文件的数据连同其它无关文件的数据。图 3-2 中展示多个实例。共用文件可在垃圾收集操作期间从普通文件中创建或者通过合并编程区块来创建。

[0201] 共用文件一般是关闭的文件,且不具有相关联的写入指针。如果主机打开共用文件,那么编程区块被打开,且将编程指针设置到编程区块的开始处。如果主机将文件删除,那么其文件区块可立刻被擦除,但必须在擦除共用区块之前在垃圾收集操作中将无关文件的数据从共用区块复制到快闪存储器中的另一位置。

[0202] 3.2.3 已编辑的普通文件

[0203] 普通文件可在任何时间由主机编辑,主机写入文件的先前编程的偏移地址的更新数据。图 3-3 中给出多个实例。用正常方式在编程区块中的编程指针处编程此种更新数据,且所得的已编辑普通文件将在一个或一个以上废弃的文件区块或在编程区块本身中含有废弃数据。

[0204] 可在文件的垃圾收集操作中,将已编辑的普通文件还原成普通文件。在此垃圾收集期间,将任何有效文件数据从每一废弃的文件区块复制到文件的编程指针,并擦除所得的完全废弃的区块。如果可能的话,直到文件已被主机关闭之后才执行垃圾收集。

[0205] 3.2.4 已编辑的共用文件

[0206] 打开的共用文件可在任何时间由主机编辑,所述主机写入所述文件的先前编程的偏移地址的更新数据。图 3-4 中展示多个实例。用正常方式在编程区块中的编程指针处编程此种更新数据,且所得的已编辑共用文件将在一个或一个以上废弃的文件区块中、在共用区块中或在编程区块本身中含有废弃数据。

[0207] 可在文件的垃圾收集操作中将已编辑的共用文件还原成普通文件格式。在此垃圾收集期间,将任何有效的文件数据从每一废弃的文件区块和共用区块复制到所述文件的编程指针。擦除所得的完全废弃的文件区块,且记录废弃的共用区块以进行单独的后续垃圾收集操作。

[0208] 3.3 垃圾收集和区块合并

[0209] 3.3.1 垃圾收集

[0210] 执行垃圾收集操作,以便恢复废弃数据所占据的存储器容量。这些可能导致必须将有效数据复制到另一位置以便擦除区块。垃圾收集无需响应于废弃数据的创建而立刻执行。将待决的垃圾收集操作记录在垃圾收集队列中,且随后根据调度算法以最优速率执行。

[0211] 直接数据文件平台支持可由主机命令起始的后台垃圾收集操作。这允许主机向装置分派静止时间以进行内部内务处理操作,所述操作将在主机随后写入文件时实现更高性能。

[0212] 如果主机不使充分的后台时间可用,那么装置作为前台操作执行垃圾收集。垃圾收集操作的突发与编程来自主机的文件数据的突发交错。可自适应地控制交错工作循环,以便将垃圾收集率维持在最小,同时确保不积累积压工作。

[0213] 3.3.2 区块合并

[0214] 装置中的每一普通文件包含不完全填充的编程区块,且可将大量擦除容量锁定在此类编程区块中。共用区块还可含有擦除容量。因此实施合并关闭文件的编程区块和共用区块的持续进行的过程,以便控制锁定的擦除容量。将区块合并作为垃圾收集功能的一部分处理,且通过相同的调度算法对其进行管理。

[0215] 通过以下方式将编程区块或共用区块中的数据与一个或一个以上无关文件的数据合并：从另一共用区块或编程区块复制此种无关数据。如果原始区块是编程区块，那么其变成共用区块。优选的是将编程区块与废弃的共用区块合并，而不是与另一编程区块合并。废弃的共用区块中含有废弃数据，且因此不可避免地必须将有效数据从所述区块重新定位到另一位置。然而，编程区块不含有废弃数据，且将数据从所述区块复制到另一位置是一项不理想的额外开销。

[0216] 3.3.3 均衡状态

[0217] 当文件数据占据了较高百分比的装置容量时，主机必须对文件执行删除操作，以便创建用于写入新文件的容量。在此状态下，装置中的大多数文件将具有共用文件格式，因为在普通文件格式的文件的编程区块中将存在很少的容量可供擦除空间使用。

[0218] 删除共用文件需要在垃圾收集期间将无关文件的有效数据从其共用区块中重新定位。此类文件群组的数据最常见的是被重新定位到关闭文件的一个或一个以上编程区块中的可用容量。在用于最近由主机写入接着又关闭的文件的编程区块中的可用未使用容量与因文件被主机删除而从共用区块中重新定位文件数据所需的容量之间经常存在均衡。这种一般的均衡状态减少了从编程区块重新定位文件数据的需要，并且有助于文件到快闪映射算法的效率。

[0219] 4. 装置操作

[0220] 4.1 装置操作的执行

[0221] 通过主机供应的命令流来确定装置的操作序列。当接收到主机命令时，中断当前装置操作，且以正常方式解译命令。特定命令将导致执行如下四个主要装置操作之一：

[0222] 1. 数据读取；

[0223] 2. 数据编程；

[0224] 3. 文件删除；或

[0225] 4. 垃圾收集。

[0226] 装置操作继续进行，直到达到以下条件之一：

[0227] 1. 操作完成；

[0228] 2. 接收到另一主机命令；或

[0229] 3. 在前台垃圾收集模式下达到交错突发的末尾。

[0230] 如果将优先级垃圾收集操作排队以供执行，那么在解译任何新命令之前完成这些操作。

[0231] 展示装置操作的总体流程图见图 4-1。

[0232] 5. 编程文件数据

[0233] 5.1 编程文件数据的原理

[0234] 当文件的数据遵照来自主机的写入或插入命令而从主机流到装置时，将所述数据编程到快闪存储器。当已经在缓冲存储器中积累了充分数据以用于下一编程操作时，将其编程在文件的编程区块中。见第 9 章中对此操作的描述。

[0235] 当编程区块变满时，将其规定为文件区块，且将来自擦除区块集区的擦除区块分派为编程区块。此外，更新共用区块和废弃区块的文件索引表和垃圾收集队列。

[0236] 文件数据编程程序根据垃圾收集调度算法（见部分 8.4）所建立的交错参数 N1 起

始前台垃圾收集的突发。每当在快闪存储器中起始元页编程操作时,将交错编程计数器递增,且当这个计数器超过值 N1 时起始前台模式的垃圾收集操作。

[0237] 文件数据编程以一个元页为单位继续进行,直到主机传输另一命令为止。

[0238] 说明编程文件数据的流程图见图 5-1。

[0239] 6. 读取文件数据

[0240] 6.1 读取文件数据的原理

[0241] 响应于来自主机的读取命令,从快闪存储器读取从由读取_指针指定的地址处开始的文件偏移地址的数据,并将其循序返回到主机,直到达到文件末尾为止。读取文件索引表(FIT),且评估文件的 FIT 条目以识别对应于读取_指针的位置。随后的 FIT 条目指定文件的数据群组的位置。

[0242] 以一个元页为单位读取文件数据,直到达到文件末尾为止,或者直到主机传输另一命令为止。

[0243] 图 6-1 中给出读取文件数据的实例性过程。

[0244] 7. 删除文件

[0245] 7.1 删除文件的原理

[0246] 响应于来自主机的对文件的删除命令,识别含有所述文件的数据的区块并将其添加到垃圾收集队列,以用于随后的垃圾收集操作。删除文件的程序不会起始这些垃圾收集操作,且因此不会立刻擦除文件的数据。

[0247] 首先评估文件的 FIT 条目,以便识别可能含有所述文件的数据的共用区块。此后,以偏移地址次序评估文件的 FIT 条目,并将数据群组位于其中的数据区块添加到共用区块队列或废弃区块队列,以用于随后的垃圾收集。接着更新文件目录和文件索引表,以移除所述文件的条目。

[0248] 7.2 擦除文件

[0249] 响应于来自主机的对文件的擦除命令,应当遵循与用于删除命令相同的程序,但是 其中将含有文件的数据的区块添加到优先级共用区块队列和优先级废弃区块队列,以用于垃圾收集。

[0250] 接着,主要装置操作序列确保在执行其它任何主机命令之前执行这些区块的垃圾收集操作。这确保立刻擦除由擦除命令识别的文件数据。

[0251] 文件删除过程的流程图见图 7-1。

[0252] 8. 垃圾收集

[0253] 8.1 垃圾收集的原理

[0254] 垃圾收集是必须执行以恢复由废弃文件数据占据的快闪存储器容量的操作。垃圾收集可能因删除文件或因编辑文件数据而成为必要的。

[0255] 区块合并是一种形式的垃圾收集,执行所述形式的垃圾收集以恢复未由文件数据完全填充的区块中的擦除容量,以便允许将其用于存储无关文件。可对普通文件中的编程区块执行合并以将其转换成共用区块,或者对共用区块执行合并以减少其数目。

[0256] 垃圾收集和区块合并的过程按照文件到快闪映射算法的规定,将有效文件数据从来源快闪区块重新定位到一个或一个以上目的地区块,以便允许擦除来源区块。

[0257] 不立刻执行待决的垃圾收集操作,而是根据分阶段执行的调度算法执行。在装置

操作期间不时地将需要垃圾收集的对象的条目添加到三个垃圾收集队列。针对文件、废弃区块和共用区块存在单独队列。从所述队列中选择对象以用于以预定义的优先级次序进行下一垃圾收集操作。如果队列为空，那么可执行区块合并。

[0258] 可用两种方式调度垃圾收集操作。当主机不在对装置进行读取或写入存取时，主机可起始后台操作，且装置连续执行后台操作，直到主机进行另一存取为止。当装置正被主机存取时，装置可调度前台操作，且以与从主机接收的文件数据的编程操作突发交错的突发形式执行前台操作。可用自适应的方式控制交错突发的长度，以便将垃圾收集速率始终维持在所需的最小值。

[0259] 8.2 垃圾收集队列

[0260] 垃圾收集队列含有对其存在待决的垃圾收集操作的对象的条目。三个队列每一者分别含有用于废弃区块的条目、用于共用区块的条目和用于文件的条目。给予两个额外队列高于这三个队列的优先级，并且所述两个额外队列分别含有用于废弃区块的条目和用于共用区块的条目。将所述五个垃圾收集队列存储在快闪存储器中的控制区块中的控制日志中。

[0261] 8.2.1 优先级废弃区块队列

[0262] 这个队列含有已经因为来自主机的擦除命令而变得完全废弃的区块的条目。其是最高优先级垃圾收集队列。必须在从主机处接受其它任何命令之前或在对来自其它任何队列的对象起始垃圾收集操作之前完成对所述队列中识别的所有区块的垃圾收集操作。

[0263] 8.2.2 优先级共用区块队列

[0264] 这个队列含有已经因为来自主机的擦除命令而变得部分废弃的共用区块的条目。其是第二高优先级垃圾收集队列。必须在从主机处接受其它任何命令或在对来自较低优先级队列的对象起始垃圾收集操作之前完成对所述队列中识别的所有共用区块的垃圾收集操作。

[0265] 8.2.3 废弃区块队列

[0266] 这个队列含有因为来自主机的删除命令或因为对文件数据的编辑而变得完全废弃的区块的条目。其是第三高优先级垃圾收集队列。应当在对来自较低优先级队列的对象起始操作之前完成对所述队列中识别的所有区块的垃圾收集操作。

[0267] 8.2.4 共用区块队列

[0268] 这个队列含有已经因为来自主机的删除命令或因为对文件数据的编辑而变得部分废弃的共用区块的条目。其是第四高优先级垃圾收集队列。应当在对来自较低优先级队列的对象起始操作之前完成对所述队列中识别的所有区块的垃圾收集操作。

[0269] 8.2.5 文件队列

[0270] 这个队列含有因为对文件数据的编辑而具有废弃数据的文件的条目。其是最低优先级垃圾收集队列。当主机将文件关闭时，除非文件是普通文件，否则将其条目添加到文件队列。因此，在主机关闭文件时有必要对文件的 FIT 条目执行分析，以便确定文件是普通文件还是已编辑文件（普通的或共用的）。

[0271] 这个分析的程序如下：

[0272] 1. 以偏移地址次序评估相关 FIT 文件中的 FIT 条目。

[0273] 2. 确定数据群组和数据群组标头的累积容量。

[0274] 3. 确定由文件数据占据的物理容量。这是除了含有文件数据的编程区块之外的区块数目加上编程区块中的已使用容量。

[0275] 4. 如果数据群组容量超过物理容量的 X%，那么确定文件是普通文件。X 值的实例是 98%。X% 小于 100%，以允许因缓冲器清洗操作引起的未编程空间在普通文件中保持。

[0276] 8.3 垃圾收集操作

[0277] 8.3.1 废弃区块

[0278] 废弃区块只含有废弃数据且可被擦除，而无需将数据重新定位到另一区块。

[0279] 8.3.2 共用区块

[0280] 共用区块是来源区块，且含有一个或一个以上文件的一个或一个以上部分或完全废弃的数据群组。必须将有效数据从这个来源区块重新定位到一个或一个以上目的地区块。

[0281] 以完整文件群组为单位重新定位数据，其中文件群组包括共用区块内的同一文件的一个或一个以上数据群组。将每一文件群组完整地重新定位到目的地区块，但可将不同的文件群组重新定位到不同区块。

[0282] 共用区块是目的地区块的优选选择，如果没有合适的共用区块可用的话则后接编程区块，如果没有合适的编程区块可用的话则后接擦除区块。

[0283] 垃圾收集操作可继续进行，直到发生以下条件之一为止：操作完成；主机发送命令；或在前台模式下达到交错突发的末尾。

[0284] 图 8-5 中展示共用区块垃圾收集操作的流程图。

[0285] 8.3.3 文件垃圾收集

[0286] 执行文件垃圾收集，以便恢复所述文件的废弃数据所占据的容量。其将处于已编辑的普通文件状态或已编辑的共用文件状态下的文件还原成普通文件状态。第一步骤是对其 FIT 文件中的 FIT 条目执行分析，以识别其数据群组必须在垃圾收集期间被复制的废弃文件区块和共用区块。这个分析的程序如下：

[0287] 1. 以偏移地址次序评估相关 FIT 文件中的 FIT 条目。

[0288] 2. 构造数据群组列表以将数据群组与物理区块联系起来。从这个列表中排除编程区块中的数据群组。

[0289] 3. 确定列表中提及的每一区块中由数据群组和数据群组标头占据的物理容量。

[0290] 4. 如果数据群组容量超过区块容量的 X%，那么确定所述区块是文件区块。X 值的实例是 98%。X% 小于 100%，以允许因缓冲器清洗操作引起的未编程空间在文件区块中保持。

[0291] 5. 将被确定为文件区块的区块中的数据群组从以上构造的数据群组列表中移除。

[0292] 修订的数据群组列表中提及的数据群组包含在废弃文件区块或共用区块中，且在文件垃圾收集操作期间被复制到编程区块。可能因为文件垃圾收集操作而修改文件的数据群组结构，也就是说，可能通过区块边界将重新定位的数据群组一分为二，或者可将其与邻近的数据群组归并。将所述文件的编程区块用作目的地区块。当这个区块填满时，打开另一编程区块。

[0293] 垃圾收集操作可继续进行，直到发生以下条件之一为止：操作完成；主机发送命令；或在前台模式下达到交错突发的末尾。

[0294] 在图 8-4 中展示文件垃圾收集操作的流程图。

[0295] 8.3.4 区块合并

[0296] 执行区块合并以恢复已经被不完全编程的编程区块和共用区块中的擦除容量，并使得所述容量可用于存储其它文件的数据。

[0297] 将用于合并的来源区块选作编程区块日志或共用区块日志中具有最低编程容量的区块，以允许在尽可能最少的数据重新定位之后擦除所述区块。以完整文件群组为单位重新定位数据，其中文件群组包括编程区块或共用区块内的同一文件的一个或一个以上数据群组。将每一文件群组完整地重新定位到目的地区块，但可将不同的文件群组重新定位到不同区块。

[0298] 共用区块是目的地区块的优选选择，如果没有合适的共用区块可用的话则后接编程区块。在没有目的地区块可用的稀有情况下，可将文件群组进行分裂以将其重新定位到一个以上目的地区块。

[0299] 区块合并操作可继续进行，直到发生以下条件之一为止：操作完成；主机发送命令；或在前台模式下达到交错突发的末尾。

[0300] 图 8-6 中展示区块合并操作的流程图。

[0301] 8.4 垃圾收集操作的调度

[0302] 在主机装置正在存取卡的时期期间，优选地将垃圾收集作为后台任务执行。在直接数据文件平台中支持由主机起始的后台垃圾收集。

[0303] 然而，也可能有必要在主机正向装置写入数据时将垃圾收集作为前台任务来执行。在这个模式下，完整的垃圾收集操作不需要作为单个事件来完成。垃圾收集的突发可与编程来自主机的数据的突发交错，以使得可在多个单独阶段中完成垃圾收集操作，且对于主机对装置的可用性的中断有限。

[0304] 8.4.1 后台操作

[0305] 当主机向装置发送闲置命令时起始后台垃圾收集。这指示主机将不会故意从装置移除电力，且不会立刻意图存取装置。然而，主机可在任何时间通过传输另一命令来结束闲置状态。

[0306] 在所述闲置状态下，装置执行连续的垃圾收集操作，直到发生以下条件之一为止：主机传输另一命令；或所有垃圾收集队列为空且不可能有任何区块合并操作。

[0307] 8.4.2 交错操作

[0308] 通过用于编程文件数据的直接数据文件过程来起始交错的垃圾收集操作。图 8-1 中说明交错操作。在主机接口活动且对快闪存储器进行 N1 个编程操作的主机数据写入阶段之后，装置切换到垃圾收集阶段。在这个阶段中，执行一个或一个以上垃圾收集操作中的一部分，直到完成对快闪存储器的 N2 个编程操作为止。

[0309] 可在垃圾收集阶段末尾暂停正在进行的垃圾收集操作，且在下一个此种阶段中重新开始垃圾收集操作。通过自适应调度算法确定 N1 和 N2 的值。

[0310] 8.4.3 自适应调度

[0311] 使用自适应调度方法来控制主机数据编程和垃圾收集的交错突发的相对长度，以使得可将垃圾收集操作对主机数据写入操作的中断保持到最小值。在还确保不积累可能导致随后性能降低的待决垃圾收集的积压工作的同时实现这一点。

[0312] 8.4.3.1 操作原理

[0313] 在任何时间,装置状态均包括由先前写入的主机数据占据的容量、擦除区块集区中的区块中的擦除容量和可供垃圾收集操作写入其它主机数据使用的可恢复容量。此可恢复容量可能在编程区块、共用区块或与先前写入的主机数据共享的废弃文件区块中,或者在完全废弃的区块中。图 8-2 中展示这些类型的容量利用。

[0314] 垃圾收集的自适应调度控制编程递增主机数据和重新定位先前写入的主机数据的交错率,使得所述比率可在其间所有可恢复容量均可供主机数据使用的自适应时期中保持恒定。如果主机删除文件(这将先前写入的主机数据转换成可恢复容量),那么相应地改变交错率且开始新的自适应时期。

[0315] 可如下确定最佳交错率:

[0316] 如果

[0317] 擦除区块集区中的擦除区块的数目 = erased_blocks;

[0318] 编程区块和共用区块的组合数目 = data_blocks;

[0319] 编程区块和共用区块中的有效数据页的总数 = data_pages;

[0320] 废弃区块的数目 = obsolete_blocks;且

[0321] 区块中的页的数目 = pages_per_block,

[0322] 那么

[0323] 可通过垃圾收集创建的擦除区块的数目表达为 $data_blocks - (data_pages / pages_per_block)$;

[0324] 垃圾收集之后的擦除区块的总数表达为 $erased_blocks + obsolete_blocks + data_blocks - (data_pages / pages_per_block)$;

[0325] 可写入的递增数据元页的数目表达为 $pages_per_block * (erased_blocks + obsolete_blocks + data_blocks) - data_pages$ 。

[0326] 如果

[0327] 假设有效数据在所有编程区块和共用区块中均匀分布(这是个悲观的假设,因为将具有低数据页计数的区块选作用于区块合并操作的来源区块)

[0328] 那么

[0329] 垃圾收集期间重新定位的元页的数目表达为 $data_pages * (data_blocks - data_pages / pages_per_block) / data_blocks$ 。

[0330] 最佳交错率 N1 : N2 是可写入的递增数据元页的数目与在垃圾收集期间必须重新定位的元页的数目的比率。因此,

[0331] $N1 : N2 = (pages_per_block * (erased_blocks + obsolete_blocks + data_blocks) - data_pages) / (data_pages * (data_blocks - data_pages / pages_per_block) / data_blocks)$

[0332] 请注意,尚未在自适应调度算法中包含废弃文件区块中的废弃容量的恢复。此容量只是由编辑文件产生的,且不常发生。如果在废弃文件区块中存在相当大容量,那么以自适应方式确定的交错率可能不是最佳的,但是切换到具有最小比率的操作(在 8.4.3.2 中描述)将确保对此类区块的高效垃圾收集。

[0333] 8.4.3.2 交错控制

[0334] 在三个区段中定义交错率 N1 : N2, 如下。

[0335] 1) 最大值 : 设置交错率的最大限度, 以便确保永远不会完全抑制垃圾收集。此最大限度的实例是 10 : 1。

[0336] 2) 自适应 : 在自适应区段中, 将交错率控制为对于共用区块和废弃区块的待决垃圾收集以及编程区块和共用区块的合并来说为最佳的。其通过以下关系式定义 :

[0337] $N1 : N2 = (pages_per_block * (erased_blocks + obsolete_blocks + data_blocks) - data_pages) / (data_blocks * (data_blocks - data_pages / pages_per_block)) / data_blocks,$

[0338] 其中 N1 和 N2 定义为页编程操作的数目。

[0339] 定义 N2 的值, 其代表垃圾收集突发的优选持续时间。这个值的实例是 16。

[0340] 3) 最小值 : 如果擦除区块集区中的区块数目在所定义的最小值以下, 那么不是以自适应方式定义交错率, 而是将其设置成固定的小限度。这个小限度的实例是 1 : 10。

[0341] 8.4.3.3 控制参数

[0342] 以下参数维持在快闪存储器中的控制区块中的控制日志中, 以用于控制自适应调度 :

[0343] 擦除区块计数 : 维持擦除区块集区中的区块数目的计数。当向擦除区块集区添加区块或从中移除区块时更新这个计数。

[0344] 编程与共用区块计数 : 维持编程区块与共用区块的组合数目的计数。共用区块可含有废弃数据。当向编程区块日志和共用区块日志添加区块或从中移除区块时更新所述计数。

[0345] 编程与共用区块页计数 : 维持编程区块和共用区块中的有效数据页的数目的计数。当向编程区块日志和共用区块日志添加区块或从中移除区块时更新所述计数。

[0346] 废弃区块计数 : 维持等待垃圾收集的完全废弃区块的数目的计数。当向废弃区块垃圾收集队列添加区块或从中移除区块时更新所述计数。

[0347] 8.5 垃圾收集的流程图

[0348] 在图 8-3 中给出用于选择若干特定垃圾收集操作之一的特定算法的流程图。图 8-4 是用于图 8-3 的“文件垃圾收集”方框的流程图。用于图 8-3 的“共用区块垃圾收集”方框的流程图是图 8-5 的主题。通过图 8-6 的流程图展示图 8-3 的“区块合并”功能。

[0349] 图 8-7A 到图 8-7D 四个图展示可由图 8-5 的过程产生的共用区块的实例性垃圾收集。图 8-7A 展示初始条件, 而图 8-7B 到图 8-7C 说明垃圾收集过程中的三个步骤。箭头展示将有效数据从废弃区块传递到不满的文件区块中, 于是这些目的地文件区块变成共用区块。

[0350] 9. 数据缓冲与编程

[0351] 本部分中描述的数据缓冲和编程方法限于使用与当前产品中采用的相同的快闪接口和错误校正码 (ECC) 结构。如果将来引入新的快闪接口和 ECC 结构, 那么可采用替代的最优方法。

[0352] 9.1 数据缓冲器

[0353] 缓冲存储器存在于控制器中的 SRAM(现有申请案的 RAM 31) 中, 以用于临时存储正在编程到快闪存储器或从中读取的数据。使用缓冲存储器中的分派区域来积累文件的充

分数据,以允许在单个操作中将完整元页编程到快闪存储器中。缓冲存储器中的文件的数据的偏移地址不重要。缓冲存储器可存储多个文件的数据。

[0354] 为了允许将主机的写入和读取操作两者管线化,具有两个元页的容量的缓冲存储器空间应当可用于正被缓冲的每一文件。缓冲存储器包括一组扇区缓冲器。可分派各个扇区缓冲器以用于临时存储单个文件的数据,且当已经将数据传递到其最终目的地时将所述扇区缓冲器解除指派。通过扇区缓冲器编号 0 到 N-1 来识别扇区缓冲器。扇区缓冲器的数目 (N) 的实例是 64。

[0355] 按扇区缓冲器的扇区缓冲器编号的次序循环分派可用的扇区缓冲器。每一扇区缓冲器具有文件标签以及定义其中所含有的数据开头和末尾的两个相关联指针。还记录扇区缓冲器中的数据内的文件偏移地址范围。扇区缓冲器以及与其相关联的控制信息两者均只存在于控制器中的易失性存储器内。

[0356] 9.2 数据编程

[0357] 9.2.1 元页

[0358] 元页是快闪存储器中的最大编程单位。应当在可能时以元页为单位编程数据,以便实现最大性能。

[0359] 9.2.2 页

[0360] 页是元页的子集,并且是快闪存储器中的最小编程单位。

[0361] 9.2.3 扇区

[0362] 扇区是页的子集,并且是控制器与快闪存储器之间的最小数据传递单位。扇区通常包括 512 个字节的文件数据。由控制器针对每一扇区产生 ECC(例如通过现有申请案的图 2 的控制器 ECC 电路 33),并且将所述 ECC 传递到附加于扇区末尾的快闪。当从快闪中读取数据时,必须将所述数据以多个完整扇区的形式传递到控制器,以便允许检验 ECC。

[0363] 9.3 缓冲器清洗

[0364] 文件的数据一般在扇区缓冲器中积累,直到有充分的数据可用于在快闪存储器中编程完整的元页为止。当主机停止流动文件的数据时,一个或一个以上扇区缓冲器保持有元页的一部分的数据。这个数据保持在缓冲存储器中,以便允许主机写入所述文件的其它数据。然而,在特定情形下,必须在一种称为缓冲器清洗的操作中将缓冲存储器中的数据提交给快闪存储器。缓冲器清洗操作导致将保持在扇区缓冲器中的文件的所有数据编程在元页内的一个或一个以上页中。

[0365] 在以下两种情况下执行缓冲器清洗操作:

[0366] 1) 主机关闭文件,或

[0367] 2) 主机接收到关机命令。

[0368] 如果已经将主机关闭的文件的数据换出到交换区块,那么应当将所述数据复原到缓冲存储器,且应当执行缓冲器清洗。应当将电源移除后在装置初始化期间处于交换区块中的数据复原到缓冲存储器,且应当执行缓冲器清洗。

[0369] 9.4 缓冲器交换

[0370] 缓冲器交换是这样一种操作:其中将一个或一个以上扇区缓冲器中的文件的数据编程在称为交换区块的临时位置中,以便随后在主机继续写入所述文件的数据时复原到缓冲存储器。

[0371] 9.4.1 交换区块的格式

[0372] 交换区块是存储已经从扇区缓冲器中换出的文件的数据的专用区块。将文件的数据连续存储在交换区块中专用于所述文件的一个或一个以上页中。当随后将数据换入回到缓冲存储器时,所述数据在交换区块中变成废弃的。

[0373] 当交换区块变满时,将交换区块内的有效数据以压缩形式写入到擦除区块,所述擦除区块于是变成交换区块。这种压缩形式允许不同文件的数据存在于同一页内。优选地只存在单个交换区块。

[0374] 9.4.2 对存储在交换区块中的数据进行索引

[0375] 将交换区块索引维持在快闪存储器中,其中针对交换区块中的每一文件含有先前针对缓冲存储器中的文件而记录的信息的复本(见9.1)。

[0376] 9.4.3 将数据移动到交换区块(换出)

[0377] 在没有充分的扇区缓冲器可用于分派到已经被主机打开的文件或者分派到因来自主机的对文件的写入命令而必须从交换区块中换入的文件时,发生换出操作。选定用于换出的文件应当是在缓冲存储器中存在相应缓冲器的文件中最早写入的文件。

[0378] 视情况而定,为了在未调度的电源移除的情况下改进数据安全性,可针对缓冲存储器中与来自主机的最近写入命令无关的任何文件执行换出。在此情况下,所述文件的数据可保留在缓冲存储器中,且如果尚未移除电源的话,则不需要进行后续的换入操作。

[0379] 9.4.4 从交换区块中复原数据(换入)

[0380] 将文件的完整数据从交换区块读取到一个或一个以上扇区缓冲器。文件数据无需具有与其换出之前完全相同的与扇区缓冲器的对准。由于压缩了缓冲区块,所以对准可能已经发生变化。交换区块中的文件的数据变成废弃的。

[0381] 9.5 编程来自主机的文件数据

[0382] 本部分中给出的实例涉及每个元页具有两个页且每个页具有两个扇区的快闪存储器配置。

[0383] 9.5.1 编程来自主机的连续数据

[0384] 文件的数据从主机中流出且在连续分派的扇区缓冲器中积累。当充分的扇区缓冲器已经被填满时,将其数据与每一扇区的ECC一起传递到快闪存储器,且将文件的编程区块中的目的地元页编程。图9-1中展示连续主机数据编程的实例。

[0385] 9.5.2 编程来自主机的中断数据

[0386] 文件的数据从主机中流出且在连续分派的扇区缓冲器中积累。图9-2展示已经被中断的主机数据编程的实例。在数据区段2A之后中断流,同时执行不同文件的写入操作。当接收到针对所述文件的另一写入命令时,在与以前一样的扇区缓冲器中积累从主机流出的数据,其中从数据区段2B处开始。当已经有充分的扇区缓冲器被填满时,将其数据连同每一扇区的ECC一起传递到快闪存储器,且编程文件的编程区块中的目的地元页。

[0387] 9.5.3 编程正从缓冲器中清洗的数据

[0388] 从主机流出文件数据,并将其积累在连续分派的扇区缓冲器中。然而,不存在充分的数据以编程在完整元页中。在图9-3中给出实例。将数据区段1和2A连同填补区段2B并连同每一扇区的ECC一起传递到快闪存储器,且对文件的编程区块中的目的地页进行编程。

[0389] 9.5.4 编程正从缓冲器换出的数据

[0390] 这个操作与用于缓冲器清洗编程的操作相同,区别只是目的地页是交换区块中的而不是文件的编程区块中的下一可用页。图 9-4 对此进行说明。

[0391] 9.5.5 编程在从缓冲器中清洗之后来自主机的数据

[0392] 与从缓冲存储器中清洗的数据分离地编程在文件的缓冲器清洗操作之后由主机供应的文件数据。编程因此必须从文件的编程区块中的下一可用页开始。积累足以完成当前元页的数据,且将其与 ECC 一起传递并编程,如针对图 9-5 中的扇区 3 和 4 所示那样。

[0393] 9.5.6 编程在换入到缓冲器之后来自主机的数据

[0394] 当接收到已经从缓冲存储器中换出的文件的其它数据时,将其积累在缓冲存储器中分派的扇区缓冲器中。还将换出数据从交换区块复原到缓冲存储器。当已经积累了充分的数据时,在单个操作中编程整个元页。

[0395] 如图 9-6 中所说明,从交换区块中读取数据区段 1 和 2A 连同填补区段 2B,并在两个扇区缓冲器中复原。在所述两个扇区上检验 ECC。

[0396] 如图 9-7 的实例中所示,在缓冲器扇区中积累来自主机的文件数据。将数据区段 1、2A/2B、3A/3B 和 4A/4B 连同每一扇区的 ECC 一起传递到快闪存储器,并将其编程为扇区 1、2、3 和 4。

[0397] 9.6 编程从快闪中复制的数据

[0398] 9.6.1 从对准的元页中复制数据

[0399] 当拟复制到完整目的地元页的数据占据了单个完整的来源元页时,将来源元页和目的地元页称为对准的,如图 9-8 中说明的。将数据扇区 1、2、3 和 4 从来源元页读取到四个扇区缓冲器,且在每一扇区上检验 ECC。

[0400] 9.6.1.1 从缓冲器中写回

[0401] 将数据从所述四个扇区缓冲器编程到目的地元页,如图 9-9 所示。针对扇区 1、2、3 和 4 产生和存储 ECC。

[0402] 9.6.1.2 芯片上复制

[0403] 当拟复制的数据的元页对准与在来源和目的地元页中相同时,可使用快闪芯片内的芯片上复制来提高复制操作的速度。如果 ECC 检验未显示任何错误,那么将数据编程到目的地元页。

[0404] 9.6.1.3 共用区块内的元页对准

[0405] 应当迫使共用区块内的每一文件群组的开头与元页的开头对准。编程区块中的数据群组也与区块中的第一元页的开头对准。因此,共用区块的所有数据复制操作(例如将编程区块合并到共用区块,并将文件群组从一个共用区块复制到编程区块或另一共用区块)将与对准的元页之间的数据复制一起操作。当将数据复制到共用区块或从共用区块中复制数据时应使用快闪芯片内的芯片上复制以便提高复制操作的速度。

[0406] 9.6.2 从未对准的循序元页中复制数据

[0407] 当拟复制到完整目的地元页的数据虽然连续但占据了两个循序来源元页时,将来源和目的地元页称为未对准。图 9-10 中展示读取来源元页的实例。将数据扇区 1A/1B、2 和 3 从第一来源元页读取到三个扇区缓冲器,且将数据扇区 4 和 5A/5B 从第二来源元页读取到另外两个扇区缓冲器。在每一扇区上检验 ECC。

[0408] 将数据部分 1A/1B、2A/2B、3A/3B 和 4A/4B 从扇区缓冲器编程到目的地元页中的扇区 1、2、3 和 4，如图 9-11 所示。针对扇区 1、2、3 和 4 产生和存储 ECC。

[0409] 当正被复制的元页数据是一长串连续数据中的一部分时，可将所述复本部分地管线化。将数据以完整元页的形式从来源位置读取到缓冲存储器。必须读取 N+1 个来源元页，以便编程 N 个目的地元页。

[0410] 9.6.3 从未对准的非循序元页中复制数据

[0411] 当拟复制到完整目的地元页的数据不连续并且占据两个或两个以上非循序的来源元页时，将来源和目的地元页称为未对准且非循序。这种情况代表将文件内的两个或两个以上非连续数据群组中的一部分复制到单个目的地元页。如图 9-12 所示，将数据扇区 1A/1B、2 和 3A/3B 从第一来源元页读取到三个扇区缓冲器，且将数据扇区 4A/4B 和 5A/5B 从第二来源元页读取到另外两个扇区缓冲器。在每一扇区上检验 ECC。

[0412] 接着将数据部分 1A/1B、2A/2B、3A/3B 和 4A/4B 从扇区缓冲器编程到目的地元页中的扇区 1、2、3 和 4，如图 9-13 所示。针对扇区 1、2、3 和 4 产生和存储 ECC。

[0413] 10. 文件索引

[0414] 10.1 文件索引的原理

[0415] 图 10-1 中大体上展示文件索引。将文件的数据存储为一组数据群组，其中每一数据群组在文件偏移地址空间和物理地址空间两者中跨越一连串连续地址。文件的所述组内的数据群组无需彼此具有任何特定的物理地址关系。文件索引表 (FIT) 允许以偏移地址次序识别文件的有效数据群组的位置。通过文件数据指针识别文件的一组 FIT 条目。

[0416] 将主机产生的与文件相关联的信息作为文件_信息存储在信息表 (IT) 中。文件_信息的性质和内容由主机确定，且其不由装置解译。文件_信息可包含文件的文件名、父目录、子目录、属性、权利信息和文件关联。IT 中的文件的文件_信息通过文件信息指针识别。

[0417] 目录含有装置中的每个有效文件的文件数据指针和文件信息指针。文件的这些目录条目通过文件 ID 来识别，所述文件 ID 是数字值。

[0418] 10.2 文件索引结构

[0419] 图 10-2 展示文件索引结构的实例。

[0420] 10.3 目录

[0421] 10.3.1 文件 ID

[0422] 文件 ID 是直接数据文件平台内的文件的数字识别符。其由直接数据文件平台响应于创建命令而分派，或者可通过创建命令指定为参数。

[0423] 当由装置分派文件 ID 值时，使用指向目录中的条目的循环指针来定位下一个可用文件 ID。当文件被删除或擦除时，将文件的文件 ID 识别的目录条目标记为可用的。

[0424] 文件 ID 值定义目录中的条目，所述条目含有文件的文件数据指针和文件信息指针的字段。

[0425] 通过针对文件 ID 分派的位的数目来确定可存储在装置中的文件的最大数目。

[0426] 10.3.2 文件数据指针

[0427] 文件数据指针是在控制日志内指向 FIT 区块列表中的文件条目的逻辑指针，且可能也是指向 FIT 更新区块列表中的文件条目的逻辑指针。

- [0428] 文件数据指针具有两个字段：
- [0429] 1) FIT 范围, 和
- [0430] 2) FIT 文件编号。
- [0431] 即使当文件具有零长度时也存在针对所述文件的文件数据指针。
- [0432] 10. 3. 2. 1 FIT 范围
- [0433] FIT 范围是 FIT 的子集。每一 FIT 范围映射到单独的物理 FIT 区块。FIT 范围可含有一个 FIT 文件到最大数目的 FIT 文件, 所述最大数目例如可为 512。
- [0434] 10. 3. 2. 2 FIT 文件编号
- [0435] FIT 文件编号是用来在 FIT 内识别 FIT 文件的逻辑编号。
- [0436] 10. 3. 3 文件信息指针
- [0437] 文件信息指针是在控制日志内指向信息区块列表中的文件条目的逻辑指针, 且可能也是指向信息更新区块列表中的文件条目的逻辑指针。
- [0438] 文件信息指针具有两个字段：
- [0439] 1) 信息范围 ; 和
- [0440] 2) 信息编号。
- [0441] 10. 3. 3. 1 信息范围
- [0442] 信息范围是信息表的子集。每一信息范围映射到单独的物理信息区块。信息范围可含有一组文件 _ 信息到最大数目的组的文件 _ 信息, 所述最大数目例如可为 512。
- [0443] 10. 3. 3. 2 信息编号
- [0444] 信息编号是用来在信息区块中识别一组文件 _ 信息的逻辑编号。
- [0445] 10. 3. 4 目录结构
- [0446] 目录存储在专用于所述用途的快闪区块中。图 10-3 展示实例性目录区块格式。所述目录被构造成一组页, 在每一页内针对具有连续文件 ID 值的文件存在一组条目。这组条目称为目录范围。
- [0447] 通过在由控制指针定义的下一擦除页位置处写入目录页的修订版本来更新目录。如果必要的话, 可通过将多个页编程到元页中的不同页来同时更新所述多个页。
- [0448] 通过目录区块中的最后写入页中的范围指针来识别目录范围的当前页位置。
- [0449] 10. 4 区块列表
- [0450] 文件索引表和信息表两者均包括一系列逻辑范围, 其中范围与物理快闪区块具有相关性。在控制日志中维持区块列表以便记录文件数据指针或文件信息指针中定义的范围与物理区块之间的相关性, 以及文件数据指针或文件信息指针中定义的逻辑编号与文件索引表和信息表内的物理区块中使用的逻辑编号之间的相关性。
- [0451] 10. 4. 1 FIT 区块列表
- [0452] FIT 区块列表是控制日志中的为文件的 FIT 中的条目分派 FIT 文件指针的列表。FIT 文件指针含有分派到文件数据指针中定义的范围的物理快闪区块的地址, 以及在文件数据指针中定义的同一 FIT 文件编号。FIT 区块列表中的条目含有单个字段——区块物理地址。
- [0453] FIT 更新区块列表是控制日志中的为 FIT 中正被更新的文件条目分派 FIT 文件指针的列表。FIT 文件指针含有当前分派为 FIT 更新区块条目的物理快闪区块的地址, 以及

FIT 更新区块中分派给正被更新的 FIT 文件的 FIT 更新文件编号。FIT 更新区块列表中的条目含有三个字段：

- [0454] 1) FIT 范围，
- [0455] 2) FIT 文件编号，和
- [0456] 3) FIT 更新文件编号。

[0457] 当应当从 FIT 区块列表中确定对应于文件数据指针的 FIT 文件指针时，搜索 FIT 更新区块以确定是否存在与文件数据指针有关的条目。如果不存在，那么 FIT 区块列表中的与文件数据指针有关的条目是有效的。

- [0458] 10.4.2 信息区块列表

[0459] 将主机写入的文件_信息直接存储在信息表中，所述信息表通过文件信息指针来识别。存在信息区块列表以向信息表中的文件_信息分派信息指针。用于这些信息区块列表的索引机制与针对 FIT 区块列表描述的机制完全相似。

- [0460] 信息区块列表中的条目含有单个字段——区块物理地址。

- [0461] 信息更新区块列表中的条目含有三个字段：

- [0462] 1) 信息范围，
- [0463] 2) 信息编号，和
- [0464] 3) 更新信息编号。

- [0465] 10.5 文件索引表

[0466] 文件索引表 (FIT) 包括一串 FIT 条目，其中每一 FIT 条目识别文件偏移地址和在数据群组的快闪存储器中的物理位置。FIT 含有存储在装置中的文件的所有有效数据群组的条目。不通过 FIT 来索引废弃的数据群组。图 10-4 中给出实例性 FIT 逻辑结构。

[0467] 将文件中的数据群组的一组 FIT 条目以文件偏移地址次序维持为连续条目。所述组条目称为 FIT 文件。将 FIT 维持为一系列 FIT 范围，且每一 FIT 范围与物理快闪区块具有相关性。FIT 范围的数目将根据装置中的数据群组的数目而变化。在装置操作期间将创建新的 FIT 范围和消除 FIT 范围。使用 FIT 区块列表来根据文件数据指针创建 FIT 文件指针，可通过所述 FIT 文件指针来识别 FIT 中的位置。

- [0468] 10.5.1 FIT 文件

[0469] FIT 文件是文件内的数据群组的一组连续 FIT 条目。所述组中的条目以文件偏移地址排序。FIT 文件中的 FIT 条目是连续的，且包含在单个 FIT 范围内或者从一个 FIT 范围溢出到下一个连续的 FIT 范围。

- [0470] 10.5.2 FIT 标头

- [0471] FIT 文件中的第一条目是 FIT 标头。其具有三个字段：

- [0472] 1) 文件 ID，
- [0473] 2) 编程区块，和
- [0474] 3) 编程指针。

- [0475] FIT 标头具有等于 FIT 条目的整数的固定长度。这个数目可能是 1。

- [0476] 10.5.2.1 文件 ID

- [0477] 文件 ID 识别目录中的文件的条目。

- [0478] 10.5.2.2 编程区块

[0479] 每当将 FIT 文件的更新版本写入到 FIT 中时, 将文件的编程区块的当前物理地址记录在 FIT 标头中。这用来在主机重新打开文件时定位文件的编程区块。其还可用来验证 FIT 文件与所述文件的编程区块之间的对应性, 所述文件已经被选定进行编程区块合并。

[0480] 10. 5. 2. 3 编程指针

[0481] 每当将 FIT 文件的更新版本写入在 FIT 中时, 将文件的编程区块内的编程指针的当前值记录在 FIT 标头中。这用来在主机重新打开文件时或者在编程区块已经被选定进行编程区块合并时定义文件的编程区块内用于编程数据的位置。

[0482] 10. 5. 3FIT 条目

[0483] FIT 条目指定数据群组。其具有四个字段 :

[0484] 1) 偏移地址,

[0485] 2) 长度,

[0486] 3) 指针, 和

[0487] 4) EOF 旗标。

[0488] 10. 5. 3. 1 偏移地址

[0489] 偏移地址是文件内与数据群组的第一字节有关的以字节计的偏移量。

[0490] 10. 5. 3. 2 长度

[0491] 其定义数据群组内的文件数据的以字节计的长度。完整数据群组的长度比这个值长出数据群组标头的长度。

[0492] 10. 5. 3. 3 指针

[0493] 其是指向数据群组的开头在快闪区块中的位置的指针。所述指针具有两个字段 :

[0494] 1) 区块地址, 其定义含有所述数据群组的物理区块, 和

[0495] 2) 字节地址, 其定义数据群组的开头在区块内的字节偏移量。这个地址含有数据群组标头。

[0496] 10. 5. 3. 4EOF 旗标

[0497] EOF 旗标是识别数据群组作为文件末尾的单个位。

[0498] 10. 5. 4FIT 区块格式

[0499] FIT 范围映射到单个物理区块, 其称为 FIT 区块。将这些区块中的更新版本的数据编程在共用更新区块中, 其称为 FIT 更新区块。以一个页为单位更新数据。如果必要的话, 可并行地更新元页内的多个页。

[0500] 10. 5. 4. 1 间接寻址

[0501] 通过 FIT 文件指针识别 FIT 文件。这个指针内的 FIT 文件编号字段是逻辑指针, 其在 FIT 文件的数据在用于索引的物理结构内移动时保持恒定。物理页结构内的指针字段提供逻辑到物理指针转译。

[0502] 10. 5. 4. 2 页格式

[0503] FIT 区块和 FIT 更新区块中采用的页格式相同。

[0504] 将页再分为两个区域, 第一区域用于 FIT 条目, 且第二区域用于文件指针。图 10-5 中给出实例。

[0505] 第一区域含有多个 FIT 条目, 其每一者指定数据群组或含有 FIT 文件的 FIT 标头。FIT 页中的 FIT 条目的数目的实例是 512。通过一组连续 FIT 条目来指定 FIT 文件, 所述组

FIT 条目在一个 FIT 页内或者与两个或两个以上 FIT 页重叠。通过第二区域中的文件指针来识别 FIT 文件的含有 FIT 标头的第一条目。

[0506] 第二区域只在最近编程的 FIT 页中含有有效文件指针。所有其它页中的第二区域是废弃的，且未使用。文件指针区域针对 FIT 区块中可能含有的每一 FIT 文件含有一个条目，也就是说，文件指针条目的数目等于 FIT 区块中可能存在的 FIT 文件的最大数目。根据 FIT 文件数目循序存储文件指针条目。第 N 个文件指针条目含有指向 FIT 区块内的 FIT 文件 N 的指针。其具有两个字段：

[0507] 1) 页编号，其指定 FIT 区块内的物理页，和

[0508] 2) 条目编号，其指定物理页内的 FIT 条目。

[0509] 文件指针条目提供用于将 FIT 区块内的逻辑 FIT 文件编号转译成区块内的物理位置的机制。当编程每个 FIT 页时更新整组文件指针，但是所述整组文件指针只在最近编程的页中有效。当在 FIT 更新区块中更新 FIT 文件时，其在 FIT 区块或 FIT 更新区块中的先前位置变成废弃的，且不再被文件指针参考。

[0510] 10.5.5 FIT 更新区块

[0511] 在所有 FIT 区块间共享的单个 FIT 更新区块中对 FIT 区块中的 FIT 文件作出改变。

图 10-6 展示物理 FIT 区块的实例。

[0512] 文件数据指针是指向 FIT 文件的逻辑指针。其 FIT 范围字段用来寻址 FIT 区块列表以识别映射到这个 FIT 范围的 FIT 区块的物理区块地址。接着，FIT 文件指针的 FIT 文件编号字段为 FIT 区块中的目标 FIT 文件选择正确的文件指针。

[0513] 文件数据指针的 FIT 范围字段和 FIT 文件编号字段两者均用来寻址 FIT 更新区块列表，以识别目标 FIT 文件是否已经被更新。如果在这个列表中找到条目，那么其提供 FIT 更新区块的物理区块地址以及 FIT 文件的更新版本的更新区块内的 FIT 文件编号。这可能与用于 FIT 区块中的 FIT 文件的 FIT 文件编号不同。FIT 更新区块含有 FIT 文件的有效版本，且 FIT 区块中的版本是废弃的。

[0514] 10.5.6 更新操作

[0515] 只在合并操作期间编程 FIT 区块。这导致 FIT 文件在区块内紧密压缩。当修改、添加或移除 FIT 条目时或者在压缩操作期间，更新 FIT 更新区块。图 10-7 展示对 FIT 文件的更新操作的实例。

[0516] 由于合并操作的缘故，FIT 文件在 FIT 区块中紧密压缩。FIT 区块可能不是被完全填满，因为其内可存在的 FIT 文件具有最大数目。FIT 文件可从一个页溢出到下一页。当更新 FIT 区块中的 FIT 文件或将其在 FIT 更新区块中复写时，所述 FIT 文件变成废弃的。

[0517] 当更新 FIT 文件时，将其完全复写在 FIT 更新区块中的下一可用页中。更新 FIT 文件可能由改变现有 FIT 条目的内容或改变 FIT 条目的数目组成。FIT 文件可从一个页溢出到下一页。FIT 更新区块内的 FIT 文件无需均与同一 FIT 范围有关。

[0518] 10.5.7 创建 FIT 范围

[0519] 当必须创建新的 FIT 范围以便适应 FIT 文件的额外存储空间时，不立刻创建 FIT 区块。首先将这个范围内的新数据写入到 FIT 更新区块。随后当针对所述范围执行合并操作时创建 FIT 区块。

[0520] 10.5.8 压缩与合并

[0521] 10.5.8.1 压缩目录更新区块或 FIT 更新区块

[0522] 当 FIT 更新区块变得填满时,可将其有效 FIT 文件数据以压缩形式编程到擦除区块,所述擦除区块于是变成更新区块。如果更新只涉及到几个文件,那么可能只有少到一页压缩有效数据要编程。

[0523] 如果拟在压缩操作中重新定位的 FIT 文件涉及关闭文件,且所述范围的 FIT 区块含有充分的未编程页,那么可将 FIT 文件重新定位到 FIT 区块,而不是重新定位到压缩的更新区块。

[0524] 10.5.8.2 合并目录区块或 FIT 区块

[0525] 当更新 FIT 条目时, FIT 区块中的原始 FIT 文件变成废弃的。此类 FIT 区块应当周期性地经受垃圾收集,以便恢复废弃空间。借助于合并操作来实现这一点。此外,可能已经在一范围内创建了新文件,且所述新文件可能在更新区块中具有条目,但是可能在 FIT 区块中不存在相应的废弃条目。应当周期性地将此类 FIT 文件重新定位到 FIT 区块。

[0526] 可对于相关范围将更新区块中的 FIT 文件合并到 FIT 区块中,且因此从更新区块中消除所述 FIT 文件,而其它 FIT 文件保留在更新区块中。

[0527] 如果 FIT 文件中的 FIT 条目的数目已经在更新过程期间增加,且无法将 FIT 范围的有效数据合并到单个擦除区块中,那么可将原先指派给所述 FIT 范围的一些 FIT 文件指派给另一 FIT 范围,且可在单独操作中在两个区块中执行合并。在 FIT 文件的此种重新指派的情况下,必须对目录中的文件数据指针进行更新以反映新 FIT 范围。

[0528] 当 FIT 更新区块中的一个范围的有效数据的容量达到所定义的阈值时,应当执行针对所述范围的合并操作。这个阈值的实例是 50%。

[0529] 对于涉及到仍然打开且主机可能继续存取的文件的活动 FIT 文件,应当优先于合并而执行压缩。

[0530] 10.6 信息表

[0531] 信息表使用与部分 10.5 中针对文件索引表所定义的相同的结构、索引机制和更新技术。然而,文件的文件 _ 信息包括不在直接数据文件平台内解译的单串信息。

[0532] 10.7 数据群组

[0533] 数据群组是一组具有文件的连续偏移地址的文件数据,其在单个存储器区块中的连续物理地址处编程。文件通常将被编程为多个数据群组。数据群组可具有介于一个字节与一个区块之间的任何长度。

[0534] 10.7.1 数据群组标头

[0535] 每一数据群组均编程有标头,所述标头含有文件识别符信息以用于交叉参考目的。标头含有包括数据群组的文件的 FIT 文件指针。

[0536] 11. 区块状态管理

[0537] 11.1 区块状态

[0538] 可将存储文件数据的区块分类成以下八种状态,如图 11-1 的状态图所示。

[0539] 11.1.1 擦除区块

[0540] 擦除区块在擦除区块集中具有擦除状态。从这个状态的可能转变如下:

[0541] (a) 擦除区块到编程区块

[0542] 当从主机供应单个文件的数据或者当在文件的垃圾收集期间复制所述文件的数

据时,将所述数据编程到擦除区块。

[0543] 11. 1. 2 编程区块

[0544] 编程区块中部分地编程有单个文件的有效数据,且含有一些擦除容量。所述文件可能是打开或关闭的。当主机供应文件的其它数据或者当在文件的垃圾收集期间复制所述文件的其它数据时,应当将所述数据编程到所述区块。

[0545] 从这个状态的可能转变如下:

[0546] (b) 编程区块到编程区块

[0547] 当从主机供应单个文件的数据时或者当在文件的垃圾收集期间复制所述文件的数据时,将所述数据编程到所述文件的编程区块。

[0548] (c) 编程区块到文件区块

[0549] 编程来自主机的单个文件的数据,以便填充所述文件的编程区块。

[0550] (f) 编程区块到废弃区块

[0551] 由于在垃圾收集期间将有效数据复制到另一区块或者由于主机删除文件的全部或一部分,所以编程区块中的文件的所有数据变成废弃的。

[0552] (h) 编程区块到废弃编程区块

[0553] 由于主机将数据的更新版本写入到编程区块或者由于主机删除文件的一部分,所以同一编程区块中的数据的一部分变成废弃的。

[0554] (1) 编程区块到共用区块

[0555] 在文件或共用区块的垃圾收集期间或者在编程区块合并期间,将所述文件的剩余数据编程到不同关闭文件的编程区块。

[0556] 11. 1. 3 文件区块

[0557] 文件区块完全由单个文件的有效数据填充。

[0558] 从这个状态的可能转变如下:

[0559] (d) 文件区块到废弃的文件区块

[0560] 由于主机将数据的更新版本编程在文件的编程区块中,所以文件区块中的数据的一部分变成废弃的。

[0561] (g) 文件区块到废弃区块 (g)

[0562] 由于主机将区块中的数据的更新版本编程在文件的编程区块中或者由于主机删除文件的全部或一部分,所以文件区块中的所有数据变成废弃的。

[0563] 11. 1. 4 废弃的文件区块

[0564] 废弃的文件区块中填充有单个文件的有效数据与废弃数据的任意组合。

[0565] 从这个状态的可能转变如下:

[0566] (e) 废弃的文件区块到废弃区块 (e)

[0567] 由于主机将区块中的有效数据的更新版本编程在文件的编程区块中、由于在垃圾收集期间将有效数据复制到另一区块或者由于主机删除文件的全部或一部分,所以废弃的文件区块中的所有数据变成废弃的。

[0568] 11. 1. 5 废弃的编程区块

[0569] 废弃的编程区块中部分地编程有单个文件的有效数据与废弃数据的任意组合,且含有一些擦除容量。当主机供应文件的其它数据时,应当将所述数据编程到区块。然而,在

垃圾收集期间,不应当将文件的数据复制到区块,且应当打开新的编程区块。

[0570] 从这个状态的可能转变如下:

[0571] (i) 废弃的编程区块到废弃的编程区块

[0572] 当从主机供应单个文件的数据时,将所述数据编程到所述文件的废弃编程区块。

[0573] (j) 废弃的编程区块到废弃区块

[0574] 由于在垃圾收集期间将有效数据复制到另一区块或者由于主机删除文件的全部或一部分,所以废弃的编程区块中的文件的所有数据变成废弃的。

[0575] (k) 废弃的编程区块到废弃的文件区块

[0576] 当从主机供应单个文件的数据时,将所述数据编程以填充所述文件的废弃的编程区块。

[0577] 11.1.6 共用区块

[0578] 共用区块中编程有两个或两个以上文件的有效数据,且通常含有一些擦除容量。可在编程区块的垃圾收集或合并期间将任何文件的剩余数据编程到共用区块。

[0579] 从这个状态的可能转变如下:

[0580] (m) 共用区块到共用区块

[0581] 在文件或共用区块的垃圾收集期间或者在编程区块的合并期间,将文件的剩余数据编程到共用区块。

[0582] (n) 共用区块到废弃的共用区块

[0583] 由于主机将数据的更新版本编程在文件的编程区块中、由于在文件的垃圾收集期间将数据复制到另一区块或者由于主机删除文件的全部或一部分,所以共用区块中的一个文件的数据的一部分或全部变成废弃的。

[0584] 11.1.7 废弃的共用区块

[0585] 废弃的共用区块中编程有两个或两个以上文件的有效数据与废弃数据的任意组合,且通常含有一些擦除容量。不应将其它数据编程到区块。

[0586] 从这个状态的可能转变如下:

[0587] (o) 废弃的共用区块到废弃区块

[0588] 由于主机将一个文件的数据的更新版本编程在文件的编程区块中、由于在文件的垃圾收集期间将一个文件的数据复制到另一区块或者由于主机删除一个文件的全部或一部分,所以废弃的共用区块中的所有文件的数据变成废弃的。

[0589] 11.1.8 废弃区块

[0590] 废弃区块只含有废弃数据,但尚未被擦除。

[0591] 从这个状态的可能转变如下:

[0592] (p) 废弃区块到擦除区块 (p)

[0593] 在垃圾收集期间擦除废弃区块,并将其添加回到擦除区块集区。

[0594] 12. 擦除区块管理

[0595] 12.1 元区块链接

[0596] 将擦除区块链接到元区块中的方法与针对早先的第3代LBA系统定义的相比没有变化。

[0597] 12.2 擦除区块集区

[0598] 擦除区块集区是装置中的可供分派以用于存储文件数据或控制信息的擦除区块的集区。集区中的每一擦除区块是元区块，且所有元区块具有相同的固定并行度。

[0599] 将集区中的擦除区块记录为控制区块中的擦除区块日志中的条目。根据区块的擦除次序将条目在日志中排序。将用于分派的擦除区块选为日志顶部处的条目。当擦除区块时将条目添加到日志尾部。

[0600] 13. 控制数据结构

[0601] 将控制数据结构存储在专用于所述用途的快闪区块中。如下定义三类区块：

[0602] 1) 文件目录区块，

[0603] 2) 文件索引表区块，和

[0604] 3) 控制区块。

[0605] 13.1 文件目录区块

[0606] 先前已经描述了文件目录区块的结构。

[0607] 13.2 文件索引表区块

[0608] 先前已经描述了文件索引表的结构。

[0609] 13.3 控制区块

[0610] 控制区块将控制信息存储在四个独立的日志中。针对每一日志分派单独的页。如果有必要的话，可将此延伸为每一日志多个页。图 13-1 中展示控制日志的实例性格式。

[0611] 通过在控制指针定义的下一擦除页位置处写入完整日志的修订版本来更新日志。如果有必要的话，可通过将多个日志编程到元页中的不同页来同时更新所述多个日志。通过控制区块中的最后写入的页中的日志指针来识别所述四个日志中每一者的有效版本的页位置。

[0612] 13.3.1 共用区块日志

[0613] 共用区块日志记录关于装置中存在的每个共用区块的信息。将共用区块日志中的日志条目细分成两个区域，第一区域用于区块条目且第二条目用于数据群组条目，如图 13-2 中说明的。每一区块条目记录共用区块的物理位置。条目具有固定大小，且在共用区块日志中存在固定数目。每一条目具有以下字段：

[0614] 1) 区块物理地址；

[0615] 2) 指向共用区块中用于编程的下一可用页的指针，

[0616] 3) 指向区块的数据群组条目中的第一条目的指针，和

[0617] 4) 数据群组条目的数目。

[0618] 数据群组条目记录关于共用区块中的数据群组的信息。一组连续数据群组条目定义共用区块中的所有数据群组。在共用区块中存在可变数目的数据群组。每一条目优选地具有以下字段：

[0619] 1) 共用区块内的字节地址，和

[0620] 2) FIT 文件指针。

[0621] 13.3.2 编程区块日志

[0622] 编程区块日志记录关于装置中存在的用于关闭文件的每个编程区块的信息。针对每一编程区块存在一个条目，且条目具有以下字段：

[0623] 1) 区块物理地址，

[0624] 2) 指向编程区块中用于编程的下一可用页的指针, 和

[0625] 3) FIT 文件指针。

[0626] 13. 3. 3 擦除区块日志

[0627] 擦除区块日志记录装置中存在的每个擦除区块的身份。针对每一擦除区块存在一个条目。根据区块的擦除次序将条目在日志中排序。将用于分派的擦除区块选作日志顶部处的条目。当擦除区块时, 将条目添加到日志的尾部。条目具有单个字段 : 区块物理地址。

[0628] 13. 3. 4 控制日志

[0629] 控制日志在以下字段中记录各种控制信息 :

[0630] 13. 3. 4. 1 打开文件列表

[0631] 这个字段含有关于当前打开文件中每一者的信息, 如下 :

[0632] 1) 路径名,

[0633] 2) 文件名,

[0634] 3) FIT 文件指针, 和

[0635] 4) 编程区块物理地址。

[0636] 编程区块日志中不包含打开文件的编程区块。

[0637] 13. 3. 4. 2 共用区块计数

[0638] 这个字段含有共用区块日志中记录的共用区块的总数。

[0639] 13. 3. 4. 3 编程区块计数

[0640] 这个字段含有编程区块日志中记录的编程区块的总数。当向编程区块日志添加区块或从中移除区块时更新所述计数。

[0641] 13. 3. 4. 4 擦除区块计数

[0642] 这个字段含有擦除区块日志中记录的擦除区块的总数。当向擦除区块日志添加区块或从中移除区块时更新所述计数。

[0643] 13. 3. 4. 5 编程 / 共用区块页计数

[0644] 这个字段含有编程区块和共用区块中的有效数据页的数目的计数。当向编程区块日志和共用区块日志添加区块或从中移除区块时更新所述计数。

[0645] 13. 3. 4. 6 废弃区块计数

[0646] 这个字段含有等待垃圾收集的完全废弃区块的数目的计数。当向废弃区块垃圾收集队列添加区块或从中移除区块时更新所述计数。

[0647] 13. 3. 4. 7 FIT 区块列表

[0648] 这个字段含有用于将 FIT 范围映射到 FIT 区块的信息。其含有定义每一 FIT 范围的 FIT 区块物理地址的条目。

[0649] 13. 3. 4. 8 FIT 更新区块列表

[0650] 这个字段含有用于将 FIT 范围和 FIT 文件编号映射到 FIT 更新文件编号的信息。其含有针对存在于更新区块中的每一有效 FIT 文件的条目。条目具有以下三个字段 :

[0651] 1) FIT 范围,

[0652] 2) FIT 文件编号, 和

[0653] 3) FIT 更新文件编号。

[0654] 13. 3. 4. 9 目录区块列表

[0655] 这个字段含有用于将目录范围映射到目录区块的信息。其含有定义每一目录范围的目录区块物理地址的条目。

[0656] 13.3.4.10 目录更新区块列表

[0657] 这个字段含有用于将目录范围和子目录编号映射到更新子目录编号的信息。其含有针对于存在于更新区块中的每一有效子目录的条目。条目具有以下三个字段：

[0658] 1) 目录范围,

[0659] 2) 子目录编号, 和

[0660] 3) 更新子目录编号。

[0661] 13.3.4.11 缓冲器交换区块索引

[0662] 这个字段含有交换区块中的有效数据群组的索引。每一数据群组的索引含有以下字段：

[0663] 1) FIT 文件指针,

[0664] 2) 交换区块内的字节地址, 和

[0665] 3) 长度。

[0666] 13.3.4.12 优先级废弃区块队列

[0667] 这个字段含有用于垃圾收集的优先级废弃区块队列中所有区块的区块地址。

[0668] 13.3.4.13 优先级共用区块队列

[0669] 这个字段含有用于垃圾收集的优先级共用区块队列中所有区块的区块地址。

[0670] 13.3.4.14 废弃区块队列

[0671] 这个字段含有用于垃圾收集的废弃区块队列中所有区块的区块地址。

[0672] 13.3.4.15 共用区块队列

[0673] 这个字段含有用于垃圾收集的共用区块队列中所有区块的区块地址。

[0674] 13.3.4.16 文件队列

[0675] 这个字段含有用于垃圾收集的文件队列中所有文件的 FIT 文件指针。

[0676] 14. 静态文件

[0677] 14.1 静态文件

[0678] 有些主机可能通过创建一组具有相同大小的文件并在所述组中的文件内周期性地更新数据来在直接数据文件装置中存储数据。将作为此组的一部分的文件称为静态文件。主机可能在存储卡外部,或者可能是存储卡内的执行卡上应用程序的处理器。

[0679] 在与本申请案同时申请的 Sergey Anatolievich Gorobets 的题为“Interfacing systemsOperating Through A Logical Address Space and on a Direct Data File Basis”的专利申请案中描述了静态文件使用的实例性应用。在所述申请案中,通过存储器控制器将主机的逻辑地址空间划分成此类静态文件。

[0680] 直接数据文件装置以与针对其它任何文件的方式完全相同的方式管理静态文件的存储。然而,主机可以用静态文件优化装置的行为和性能的方式在直接数据文件命令集中 使用命令。

[0681] 14.1.1 静态文件分区

[0682] 将静态文件作为一个组存储在装置中的专用分区中。分区中的所有静态文件具有相同的文件大小。

[0683] 14. 1. 2 静态文件大小

[0684] 主机通过写入到文件的偏移地址范围来定义文件大小。静态文件的大小等于元区块的大小。

[0685] 主机管理写入 _ 指针和读取 _ 指针代表的文件偏移值,以便将其维持在始终对静态文件允许的值的范围内。

[0686] 14. 1. 3 删 除 静 态 文件

[0687] 与直接数据文件装置中的其它文件不同,主机不在正常操作期间删除静态文件。静态文件由主机创建,接着在装置中连续存在。在任何时间写入到文件的数据均复写现有的文件数据。

[0688] 然而,主机始终有能力(例如)在主机重新格式化装置或减少装置中的静态文件分区大小的操作期间删除静态文件。

[0689] 14. 2 与 静 态 文件 一 起 使用 的 命 令 集

[0690] 图 14-1 给出用于与静态文件一起使用的命令集,在图 2-1 到 2-6 中展示了其中的一个子集,所述命令集支持静态文件所需的所有操作。

[0691] 14. 3 创 建 静 态 文件

[0692] 通过使用来自主机的创建命令在装置中创建静态文件。主机通常将指定其希望用来识别文件的文件 ID。

[0693] 主机可追踪其已经在装置中创建了哪些文件,或者其可响应于在主机已经试图打开其文件 ID 在装置中尚不存在的文件之后来自装置的错误消息而创建文件。

[0694] 14. 4 打 开 静 态 文件

[0695] 主机通过使用文件的文件 ID 作为参数发送打开命令来打开静态文件。

[0696] 主机可用以下方式与装置中的所述组静态文件一起操作:其控制装置中同时打开的文件数目或在装置中同时打开的具有由主机定义的特定类型的文件的数目。主机可因此在打开另一静态文件之前关闭一个或一个以上静态文件。

[0697] 14. 5 对 静 态 文件 的 写 入

[0698] 当首次写入静态文件时,其占据装置中的单个完整的文件区块,因为主机将文件大小定义为恰好等于快闪存储器中的元区块的大小。因此,文件的偏移地址范围恰好等于快闪存储器中的元区块的大小。

[0699] 对静态文件的随后写入导致在这个偏移地址范围内更新数据。主机通过借助于写入 _ 指针命令控制文件的写入 _ 指针值来控制正在更新数据的文件偏移地址。主机不允许写入 _ 指针值超过与静态文件大小有关的偏移地址范围的末尾。类似地,主机将读取 _ 指针值限制在这个范围内。

[0700] 当在已经打开文件之后更新静态文件中的现有数据时,打开向其编程更新数据的编程区块。文件区块中具有相应偏移地址的数据变成废弃的。如果更新了完整的静态文件,那么编程区块中的所有数据有效,且编程区块变成所述文件的文件区块。文件的先前文件区块中的所有数据已经变成废弃的,且将区块添加到废弃区块垃圾收集队列。如果针对文件接收到其它更新数据,那么将擦除区块指派为编程区块。

[0701] 如果静态文件的编程区块变满,但其不含有文件的所有有效数据,那么所述编程区块中的一些数据废弃,因为已经对同一偏移地址进行了多次更新。在此情况下,编程区块

无法变成文件区块，且当接收到文件的其它数据时，不打开另一空编程区块。分派擦除区块，向其复制来自所述编程区块的有效数据（所述编程区块是压缩的），于是这个部分填充区块变成文件的编程区块。文件的先前编程区块中的所有数据现在是废弃的，且将所述区块添加到废弃区块垃圾收集队列。

[0702] 请注意，主机可通过如以下部分 14.6 中描述那样关闭文件来强制合并文件区块与编程区块，其中所述区块每一者均含有所述文件的一些有效数据。主机可选择在部分废弃的编程区块变满时临时关闭文件，而不是在接收到文件的其它数据时允许直接数据文件装置压缩编程区块。

[0703] 14.6 关闭静态文件

[0704] 主机通过使用静态文件的文件 ID 作为参数发送关闭命令来关闭所述文件。

[0705] 如果文件的数据中只有一部分已经被更新，那么静态文件的关闭导致将文件放置到文件垃圾收集队列中。这允许如以下部分 14.7 中描述那样对文件进行后续垃圾收集操作。然而，也可如部分 14.7 中描述那样，主机可强制立刻对文件进行垃圾收集操作。

[0706] 14.7 静态文件的垃圾收集

[0707] 已经在更新有条目处于文件垃圾收集队列中的静态文件中的一部分数据之后关闭了所述静态文件。所述文件的文件区块中含有一些有效数据和一些废弃数据，且编程区块含有一些有效数据，可能含有一些废弃数据，且可能含有一些擦除容量。

[0708] 文件垃圾收集操作将文件的所有有效数据合并到单个区块。如果编程区块不含有任何废弃数据，那么将有效数据从文件区块复制到编程区块，且擦除文件区块。如果编程区块含有废弃数据，那么将来自文件区块和编程区块两者的所有有效数据复制到擦除区块，且擦除文件区块和编程区块两者。

[0709] 当条目达到队列的顶部时，在垃圾收集调度算法确定的时间执行文件垃圾收集。然而，主机可在关闭文件时强制对文件立刻执行垃圾收集操作。主机通过在对文件的关闭命令之后立刻发送闲置命令来进行此操作，这导致装置连续执行垃圾收集或区块合并操作，直到接收到另一命令为止。主机在发送另一命令之前监视装置的内部繁忙状态，直到其检测到装置不再忙于执行内部操作为止。通过这种机制，主机可确保在关闭文件之后立刻对文件的文件区块和编程区块进行合并。

[0710] 根据以上描述的实例性存储器系统的概述

[0711] 直接数据文件平台

[0712] 直接数据文件平台充当用于管理快闪存储器中的数据存储的通用后端系统。

[0713] 直接数据文件接口是支持多个数据来源的内部文件存储接口。

[0714] 具有对没有预定义长度的文件数据的随机读取 / 写入存取的文件存取接口。

[0715] 具有对有预定义长度的完整文件对象的传递的对象接口。

[0716] 与并入有文件系统的常规主机的 LBA 接口。将逻辑区块存储为逻辑文件。

[0717] 具有对文件内的数据的随机存取的嵌入式应用程序。

[0718] 直接数据文件存储是在逐文件基础上组织数据存储的后端系统。

[0719] 对存储装置没有逻辑地址空间。

[0720] 没有文件系统。

[0721] 直接数据文件与现有系统

- [0722] 直接数据文件平台提供优于现有系统的益处：
- [0723] 高数据写入速度：
- [0724] 消除了由于文件分裂引起的逐渐性能降低；
- [0725] 当在后台操作中擦除主机所删除的文件时，可提高峰值数据写入速度。
- [0726] 数据写入速度的一致性：
- [0727] 当在后台或在与写入主机数据交错的突发中执行垃圾收集时，可改进用于流动数据的持续写入速度。
- [0728] 益处是由于直接数据文件平台中所使用的算法的特性引起的：
- [0729] 有限的文件分裂
- [0730] 有限的文件和区块合并
- [0731] 真实的文件删除
- [0732] 最佳的文件数据索引
- [0733] 高效的垃圾收集
- [0734] 直接数据文件接口 - 理想的特征
- [0735] 直接数据文件接口应当独立于主机中的操作系统：
- [0736] 在平坦的分级结构中管理具有数字识别符的文件；
- [0737] 可存储与文件相关联的数据，以便允许在高于接口的级别处构造和维持分级目录。
- [0738] 直接数据文件接口优先地支持各种格式的文件数据传递：
- [0739] 大小未定义且可向其流动数据的文件；
- [0740] 在被写入之前大小已被定义的文件；
- [0741] 大小固定且永久存在的文件。
- [0742] 直接数据文件接口 - 实施方案
- [0743] 文件内的数据具有随机写入和读取存取，其粒度为一个字节。
- [0744] 可将数据附加、复写或插入在文件的现有数据内。
- [0745] 将正被写入或读取的文件数据流动到装置或从中流出，其没有预定义长度。
- [0746] 通过接收到另一命令来终止当前操作。
- [0747] 打开文件以供写入数据，且在文件结束时或者当文件不活动时关闭文件。
- [0748] 装置为主机所指定的文件返回文件句柄。
- [0749] 支持但不维持分级目录。
- [0750] 可存储文件的相关联信息。
- [0751] 可通过主机起始其中装置可在后台执行内部操作的状态。
- [0752] 直接数据文件接口 - 命令集
- [0753] 文件命令：
- [0754] 用于控制文件对象的命令，
- [0755] 创建、打开、关闭、删除、擦除、列举 _ 文件。
- [0756] 数据命令：
- [0757] 用于写入和读取文件数据的命令，
- [0758] 写入、插入、移除、读取、保存 _ 缓冲器、写入 _ 指针、读取 _ 指针。

- [0759] 信息命令：
- [0760] 用于写入和读取与文件相关联的信息的命令，
- [0761] 写入_信息、读取_信息、信息_写入_指针、信息_读取_指针。
- [0762] 状态命令：
- [0763] 用于控制装置状态的命令，
- [0764] 闲置、待命、关机。
- [0765] 装置命令：
- [0766] 用于询问装置的命令，
- [0767] 容量、状态。
- [0768] 文件到快闪映射算法
- [0769] 数据结构：
- [0770] 文件
- [0771] 数据群组
- [0772] 区块类型：
- [0773] 编程区块
- [0774] 文件区块
- [0775] 共用区块
- [0776] 文件类型：
- [0777] 普通文件
- [0778] 共用文件
- [0779] 已编辑文件
- [0780] 存储器恢复：
- [0781] 垃圾收集
- [0782] 区块合并
- [0783] 文件到快闪映射算法 - 数据结构
- [0784] 文件：
- [0785] 文件是一组由主机创建和维持的数据；
- [0786] 主机可以是外部主机或者可以是存储卡内的应用程序；
- [0787] 通过主机创建的文件名或通过直接数据文件平台创建的文件句柄来识别文件；
- [0788] 通过文件偏移地址来识别文件内的数据；
- [0789] 不同文件的偏移地址组充当装置内的独立逻辑地址空间。对于装置本身没有逻辑地址空间。
- [0790] 数据群组：
- [0791] 数据群组是单个文件在文件内具有连续偏移地址的一组数据；
- [0792] 数据群组存储在单个区块中的连续物理地址处；
- [0793] 数据群组可具有介于一个字节与一个区块之间的任何长度；
- [0794] 数据群组是用于将逻辑文件地址映射到物理快闪地址的基本单位。
- [0795] 文件到快闪映射算法 - 区块类型
- [0796] 编程区块：

- [0797] 将主机写入的所有数据编程在编程区块中；
- [0798] 编程区块专用于单个文件的数据；
- [0799] 编程区块中的文件数据的文件偏移地址可具有任何次序，且编程区块可含有文件的多个数据群组；
- [0800] 针对每一打开文件和针对未指定数目的关闭文件存在单独的编程区块。
- [0801] 文件区块：
- [0802] 当已经编程了编程区块的最后位置时，所述编程区块变成文件区块。
- [0803] 共用区块：
- [0804] 共用区块含有一个以上文件的数据群组；
- [0805] 通过在共用区块的垃圾收集期间或在区块合并操作期间将无关文件的数据群组编程到编程区块来创建共用区块；
- [0806] 可在另一共用区块的垃圾收集期间或在区块合并操作期间将数据群组写入到共用区块。
- [0807] 文件到快闪映射算法 – 文件类型
- [0808] 普通文件（见图 3-1）：
- [0809] 普通文件包括任何数目的完整文件区块和一个部分写入的编程区块。
- [0810] 可将普通文件删除，而无需在其擦除之前从任何区块重新定位数据。
- [0811] 共用文件（见图 3-2）：
- [0812] 共用文件包括任何数目的完整文件区块和一个共用区块，所述共用区块含有所述文件的数据连同其它无关文件的数据。
- [0813] 在删除文件之后必须只对共用区块执行垃圾收集操作。
- [0814] 已编辑文件（见图 3-3 和图 3-4）
- [0815] 由于现有偏移地址处的数据已经被复写，所以已编辑文件在其一个或一个以上区块中含有废弃数据。
- [0816] 可通过文件垃圾收集操作来恢复废弃数据所占据的存储器容量。
- [0817] 文件垃圾收集操作将已编辑文件复原成普通文件格式。
- [0818] 文件到快闪映射算法 – 存储器恢复
- [0819] 垃圾收集：
- [0820] 执行垃圾收集操作以恢复废弃数据所占据的存储器容量。
- [0821] 将待决操作记录在垃圾收集队列中，并随后根据调度算法以最佳速率执行所述待决操作。
- [0822] 可通过主机命令来起始垃圾收集，并在主机接口静止时在后台执行垃圾收集。在接收到其它任何主机命令后，暂停操作。
- [0823] 也可作为前台操作在与主机数据写入操作交错的突发中执行垃圾收集。
- [0824] 区块合并：
- [0825] 可实施持续进行的区块合并过程，以便恢复编程区块和共用区块中锁定的擦除容量。
- [0826] 只有在编程区块中的文件数据容量与共用区块中的删除文件的废弃数据容量的分布不平衡时才有必要。

- [0827] 合并多个编程或共用区块中的数据以允许擦除一个或一个以上区块。
- [0828] 编程文件数据
- [0829] 当由文件句柄识别的文件的数据遵循写入或插入命令从主机中流出时, 将所述数据编程到快闪存储器。
- [0830] 通过写入指针来定义数据的初始文件偏移地址, 所述写入指针的值可由主机设置。
- [0831] 当已经在缓冲存储器中积累了充分的数据时, 在文件的编程区块中编程元页。
- [0832] 当编程区块变得填满时, 将其规定为文件区块, 且将擦除区块分派为文件的新编程区块。
- [0833] 每当编程区块变得填满时或者每当接收到另一主机命令时, 在快闪存储器中更新数据群组索引结构。
- [0834] 文件数据编程程序以自适应调度算法确定的主机数据流间隔来起始前台垃圾收集突发。
- [0835] 当接收到另一主机命令时, 退出文件数据编程程序。
- [0836] 读取文件数据
- [0837] 遵循读取命令从快闪存储器中读取由文件句柄识别的文件的数据, 并将其流动到主机。
- [0838] 通过读取指针来识别数据的初始文件偏移地址, 所述读取指针的值可由主机来设置。
- [0839] 以一个元页为单位读取文件数据, 直到达到文件末尾为止或直到接收到另一主机命令为止。
- [0840] 以文件偏移地址次序将数据传递到主机。
- [0841] 通过文件索引结构来定义拟针对文件读取的数据群组的位置。
- [0842] 当接收到另一主机命令时, 退出文件数据读取程序。
- [0843] 删除文件
- [0844] 响应于对文件的删除命令, 识别含有所述文件的数据的区块, 并将其添加到垃圾收集队列以进行后续的垃圾收集操作。
- [0845] 更新文件目录和文件索引表, 以移除文件的条目。
- [0846] 删除文件的程序并不起始垃圾收集操作, 且不立刻擦除文件的数据。
- [0847] 响应于对文件的擦除命令, 遵循与删除命令相同的程序, 但在执行其它任何主机命令之前起始并完成垃圾收集操作。
- [0848] 垃圾收集
- [0849] 垃圾收集是用以恢复废弃数据所占据的快闪容量的操作。
- [0850] 在装置操作期间不时地将对象添加到 3 个垃圾收集队列, 以定义后续的垃圾收集操作:
- [0851] 废弃区块队列 - 当区块由于更新文件数据或删除文件而变得完全废弃时, 将其添加到这个队列。
- [0852] 共用区块队列 - 当区块中的含有多个文件的数据的一部分中的数据由于文件数据更新、删除文件或对文件的垃圾收集而变成废弃的时, 将其添加到这个队列。

[0853] 文件队列 - 当文件被主机关闭时,将其添加到这个队列。可针对优先级垃圾收集规定对象。

[0854] 可用两种方式调度垃圾收集操作 :

[0855] 主机可在其不在对装置进行读取或写入存取时起始后台操作。

[0856] 直接数据文件平台可在其不正被主机存取时起始前台操作。

[0857] 垃圾收集 - 调度

[0858] 通过主机起始后台垃圾收集。主机经由直接数据文件接口处的特定命令来起始闲置状态,在所述闲置状态中准许装置执行内部操作。在闲置状态持续的同时,对来自垃圾收集队列的对象的垃圾收集继续进行。当从主机接收到任何命令时暂停垃圾收集。主机可视情况监视装置的繁忙状态,以允许垃圾收集操作在发送下一命令之前完成。

[0859] 当主机尚未起始后台操作时,由直接数据文件平台起始前台垃圾收集。根据自适应算法来调度垃圾收集。当前垃圾收集操作的编程和擦除操作的突发与从主机接收到的文件数据的编程操作的突发交错。可以自适应方式控制突发的长度,以便定义交错垃圾收集的工作循环。

[0860] 垃圾收集 - 自适应调度 (见图 8-2)

[0861] 快闪存储器通常具有用于写入其它主机数据所需的包含在编程区块、共用区块和废弃文件区块中的可恢复容量。

[0862] 自适应垃圾收集控制编程其它主机数据与重新定位先前写入的主机数据的交错率。通过将可恢复容量转换成擦除容量而使得所述容量可供新的主机数据使用。垃圾收集速率在自适应期间中保持恒定。

[0863] 垃圾收集 - 操作优先级

[0864] 从具有以下优先级次序的垃圾收集队列中选择经调度垃圾收集的操作 :

[0865] 1. 废弃区块优先级垃圾收集 :

[0866] 选择因文件擦除命令而创建的废弃区块的下一条目。

[0867] 2. 共用区块优先级垃圾收集 :

[0868] 选择因文件擦除命令而创建的部分废弃共用区块的下一条目。

[0869] 3. 废弃区块垃圾收集 :

[0870] 选择废弃区块的下一条目。

[0871] 4. 共用区块垃圾收集 :

[0872] 选择部分废弃共用区块的下一条目。

[0873] 5. 文件垃圾收集 :

[0874] 选择部分废弃文件的下一条目。

[0875] 6. 区块合并 :

[0876] 当垃圾收集队列中不存在条目时,针对区块合并操作选择来源区块和目的地区块。

[0877] 垃圾收集 - 共用区块垃圾收集

[0878] 有效文件在编程区块或共用区块中含有一些数据。

[0879] 当删除文件时,任何含有所述文件的废弃数据的共用区块均经历共用区块垃圾收集操作。

[0880] 将无关文件的数据群组重新定位到另一共用区块或编程区块（见图 8-7A 到图 8-7D）。

[0881] 在共用区块垃圾收集操作期间，将有效文件群组从来源共用区块重新定位到一个或一个以上选定的目的地区块。

[0882] 针对每一文件群组单独选择目的地区块。

[0883] 用于选择目的地区块的优先级如下：

[0884] 1. 具有作为拟重新定位的来源文件群组的最佳匹配的可用擦除容量的共用区块；

[0885] 2. 具有作为拟重新定位的来源文件群组的最佳匹配的可用擦除容量的编程区块；和

[0886] 3. 擦除区块，其接着被规定为编程区块。

[0887] **垃圾收集 - 文件垃圾收集**

[0888] 可在已经关闭文件之后执行文件垃圾收集，以恢复文件的废弃数据所占据的容量。这只有在所述文件的数据已经在编辑期间被复写的情况下才是必要的。

[0889] 将已编辑普通文件状态或已编辑共用文件状态下的文件还原成普通文件状态（含有单个编程区块且不含有共用区块）。

[0890] 通过将有效数据群组从含有废弃数据的区块复制到文件的编程区块来执行文件垃圾收集。

[0891] 遵循初始编程指针从偏移地址开始以循序次序复制数据群组，且在文件末尾处绕回。

[0892] **垃圾收集 - 区块合并**

[0893] 在区块合并操作期间，将有效文件群组从选定来源区块重新定位到一个或一个以上选定的目的地区块。

[0894] 将来源区块选作具有最低数据容量的共用区块或编程区块。

[0895] 针对每一文件群组单独选择目的地区块。

[0896] 用于选择目的地区块的优先级如下：

[0897] 1. 具有拟重新定位的来源文件群组的最佳匹配的可用擦除容量的共用区块。

[0898] 2. 具有拟重新定位的来源文件群组的最佳匹配的可用擦除容量的编程区块。

[0899] 3. 具有最高可用擦除容量的编程区块或共用区块，向其写入文件群组的一部分。在此情形下，可准许文件与其它无关文件共享两个区块。

[0900] 4. 具有来源文件群组的剩余部分的最佳匹配的可用擦除容量的第二编程区块或共用区块，向其写入文件群组的剩余部分。

[0901] 5. 随后规定为编程区块的擦除区块，向其写入文件群组的剩余部分。

[0902] **文件索引（见图 10-1）**

[0903] 通过在主机创建文件时由直接数据文件装置分派的文件 ID 来识别文件。

[0904] 平坦目录针对每一文件 ID 指定文件数据指针和文件信息指针。

[0905] 文件数据指针识别文件索引表中的一组条目，其中每一条目指定所述组所涉及的用于文件的数据群组。

[0906] 文件信息指针识别信息表中的一串文件信息：

- [0907] 文件_信息由主机写入且不由直接数据文件装置解译。
- [0908] 文件_信息可包含文件的文件名、父目录、子目录、属性、权利信息和文件关联。
- [0909] 文件索引 - 索引结构
- [0910] 见图 10-2
- [0911] 文件索引 - 文件索引表 (FIT) - 见图 10-4
- [0912] FIT 含有快闪存储器中的所有用于文件的有效数据群组的条目。不通过 FIT 来索引废弃的数据群组。
- [0913] 将 FIT 划分成多个逻辑范围，其中每一者映射到物理区块。
- [0914] FIT 文件是文件的一组具有文件偏移地址次序的连续条目。
- [0915] 通过 FIT 文件指针来识别 FIT 文件，所述 FIT 文件指针定义物理区块和逻辑文件编号。
- [0916] 文件索引 - 更新文件索引 (见图 10-6 和图 10-7)
- [0917] 针对文件索引表和信息表使用相同结构。
- [0918] 使用区块列表将逻辑文件数据指针与物理 FIT 区块或 FIT 更新区块内的 FTI 文件联系起来。
- [0919] FIT 文件以压缩格式存储在 FIT 区块中。
- [0920] FIT 文件的更新版本存储在共享 FIT 更新区块中，其中页中具有单个 FIT 文件。
- [0921] 不时地执行 FIT 更新区块的压缩和 FIT 区块中的 FIT 文件合并。
- [0922] 文件索引 - 索引页格式 (见图 10-5)
- [0923] 针对 FIT 区块、FIT 更新区块、信息区块和信息更新区块使用相同结构。以一个页为单位编程信息。
- [0924] 将页细分为两个区域，用于 FIT 条目和文件指针。
- [0925] 文件指针将一个范围内的逻辑文件编号转译成相应 FIT 文件的开头的页编号和条目 编号。
- [0926] FIT 文件包括物理上连续的 FIT 条目。
- [0927] 数据缓冲和编程
- [0928] 在一组扇区缓冲器中缓冲由主机写入或正在快闪存储器内重新定位的数据。
- [0929] 数据群组边界的分辨率是一个字节，但将数据以一个扇区的倍数传递到快闪和从快闪中传递出来，以用于 ECC 产生和检验。
- [0930] 在可能的情况下，以元页为单位在快闪中编程来自缓冲器的数据。
- [0931] 当关闭文件或关机正在逼近时，缓冲器清洗操作只编程页的一部分。文件索引技术允许页的未编程部分持续。
- [0932] 缓冲器换出操作允许将缓冲器中的文件数据临时存储在共用交换区块中，以用于管理缓冲器空间和缓冲器中的数据备份。
- [0933] 将编程区块或共用区块中的文件群组的开头与元页的开头对准。
- [0934] 可针对快闪中的大多数数据重新定位使用芯片上复制。
- [0935] 区块状态管理
- [0936] 直接数据文件系统维持与数据存储相关联的八种区块状态 (见图 11-1)。
- [0937] 擦除区块管理

[0938] 直接数据文件在固定大小的元区块中存储文件的所有数据和所有控制信息。(术语“区块”通常用来表示“元区块”。)

[0939] 将擦除区块链接到区块中的方法与以下待决美国专利申请案中描述的在具有逻辑地址空间 (LBA) 接口的系统中使用的方法相同:2003 年 12 月 30 日申请的题为“Management of Non-Volatile Memory Systems Having Large Erase Blocks”的第 10/749,831 号;2003 年 12 月 30 日申请的题为“Non-Volatile Memory and Method with Block Management System”的第 10/750,155 号;2004 年 8 月 13 日申请的题为“Non-Volatile Memory and Method with Memory Planes Alignment”的第 10/917,888 号;2004 年 8 月 13 日申请的题为“Non-Volatile Memory and Method with Phased Program Failure Handling”的第 10/917,889 号;2004 年 8 月 13 日申请的题为“Non-Volatile Memory and Method with Control Data Management”的第 10/917,725 号;2005 年 7 月 27 日申请的题为“Non-Volatile Memory and Method with Multi-Stream Update Tracking”的第 11/192,200 号;2005 年 7 月 27 日申请的题为“Non-Volatile Memory and Method with Improved Indexing for Scratch Pad and Update Blocks”的第 11/192,386 号;以及 2005 年 7 月 27 日申请的题为“Non-Volatile Memory and Method with Multi-Stream Updating”的第 11/191,686 号。

[0940] 可供分派以用于存储数据或控制信息的擦除区块保持在擦除区块集中。

[0941] 将擦除区块记录为擦除区块日志中的条目。

[0942] 将用于分派的擦除区块选为日志顶部处的条目。

[0943] 当擦除区块时,将条目添加在日志尾部处。

[0944] 控制数据结构

[0945] 将控制数据结构存储在专用控制区块中。

[0946] 将控制信息存储在四个独立日志中。每一日志占据控制区块中的一个或一个以上页。通过最后写入的页中的日志指针来追踪有效的日志页。

[0947] 共用区块日志含有快闪存储器中存在的所有共用区块的条目,按其含有的可用擦除容量排序。

[0948] 编程区块日志含有快闪存储器中存在的所有编程区块的条目,按其含有的可用擦除容量排序。

[0949] 擦除区块日志含有快闪存储器中存在的所有擦除区块的条目,按其擦除顺序排序。

[0950] 控制日志含有控制参数、计数和列表的预定义字段。

[0951] 通过在控制区块中的下一擦除页位置处写入完整日志的修订版本来更新日志。

具有直接数据文件平台的存储卡

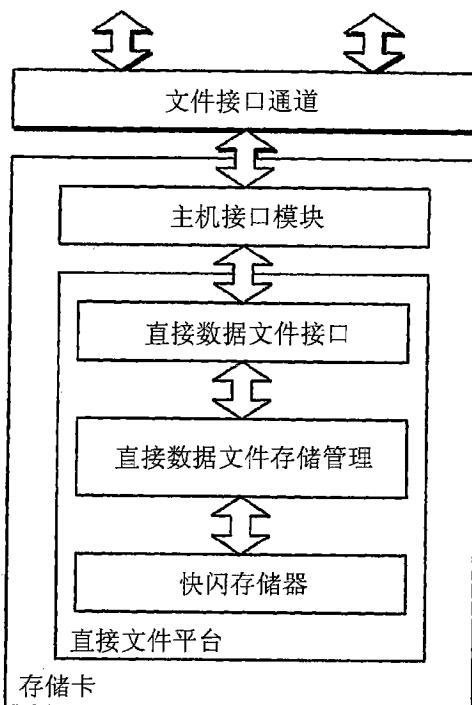


图1-1

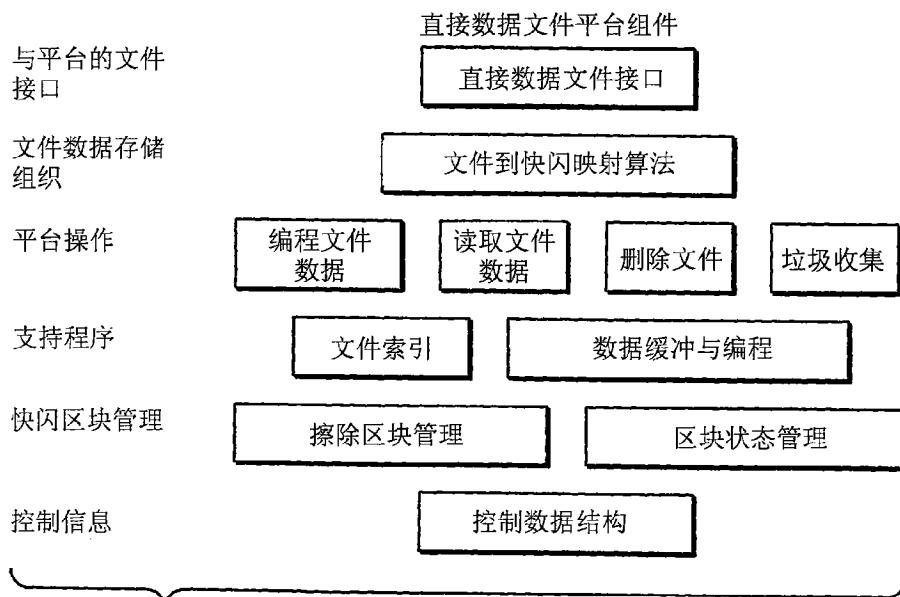


图1-2

文件命令

命令	参数	说明
创建	<文件ID>	在装置的目录中创建通过<文件ID>识别的条目。如果未用所述命令指定<文件ID>, 那么由装置指派<文件ID>并将其返回到主机。
打开	<文件ID>	启用对指定文件的后续数据命令的执行。
关闭	<文件ID>	停用对指定文件的后续数据命令的执行。
删除	<文件ID>	指示应当删除指定文件的目录、文件索引和文件信息条目。可擦除文件数据。
擦除	<文件ID>	指示应当删除指定文件的目录、文件索引和文件信息条目, 且应当立刻擦除文件数据。
列举_文件		以数字次序从装置中读取所有有效的<文件ID>值。

图2-1

数据命令

命令	参数	说明
写入	<文件ID>	在写入_指针的当前值定义的偏移地址处的指定文件中复写数据。
插入	<文件ID>	在写入_指针的当前值定义的偏移地址处的指定文件中插入数据。
移除	<文件ID><长度>	从写入_指针的当前值定义的偏移地址处的指定文件中删除大小为<长度>的数据。
读取	<文件ID>	从读取_指针的当前值定义的偏移地址处的指定文件中读取数据。
保存_缓冲器	<文件ID>	将指定文件的处于缓冲器中的数据保存到快闪存储器中的临时位置。
写入_指针	<文件ID><偏移量>	为指定文件定义写入_指针的新的当前值
读取_指针	<文件ID><偏移量>	为指定文件定义读取_指针的新的当前值。

图2-2

信息命令

命令	参数	说明
写入_信息	<文件ID>	写入信息_写入_指针的当前值定义的偏移地址处的指定文件的文件_信息。
读取_信息	<文件ID>	读取信息_读取_指针的当前值定义的偏移地址处的指定文件的文件_信息。
信息_写入_指针	<文件ID><偏移量>	为指定文件定义信息_写入_指针的新的当前值。
信息_读取_指针	<文件ID><偏移量>	为指定文件定义信息_读取_指针的新的当前值。

图2-3

流命令

命令	参数	说明
流动	<长度>	定义将传递到装置或从装置中传递的不中断数据流。
暂停	<时间>	定义在执行随后命令之前插入的延迟。

图2-4

状态命令

命令	参数	说明
闲置		装置应当进入闲置状态，在闲置状态中装置可执行内部操作。
待命		装置应当进入待命状态，在待命状态中装置可能不执行内部操作。
关机		装置将关机，且当其接下来不处于繁忙状态时将移除电力。

图2-5

装置命令

命令	参数	说明
容量		来自主机的让装置报告文件数据占据的容量和可供新文件数据使用的容量的请求。
状态		来自主机的让装置报告其当前状态的请求。

图2-6

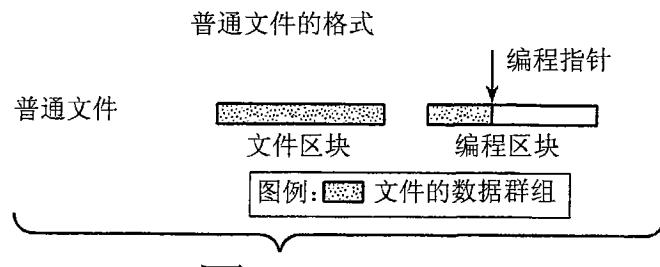


图3-1

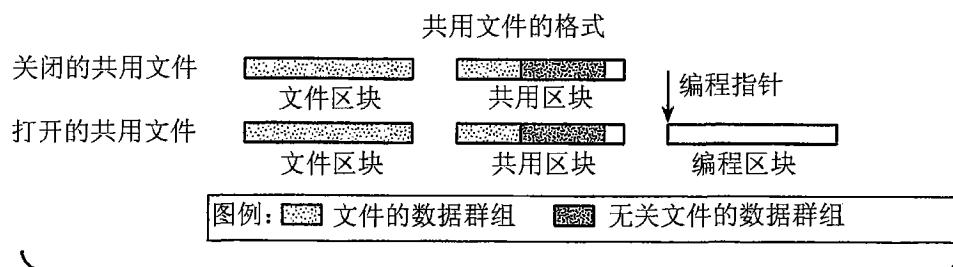


图3-2

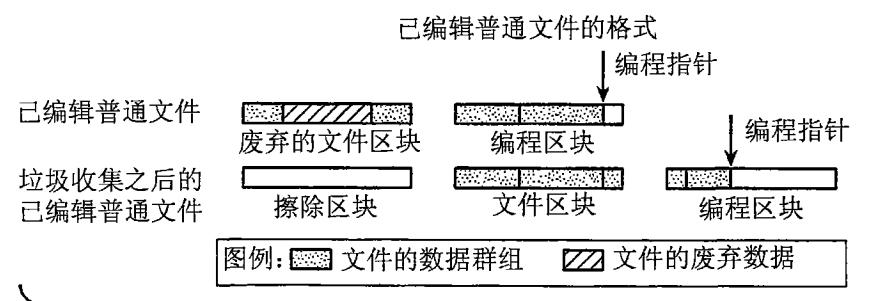


图3-3

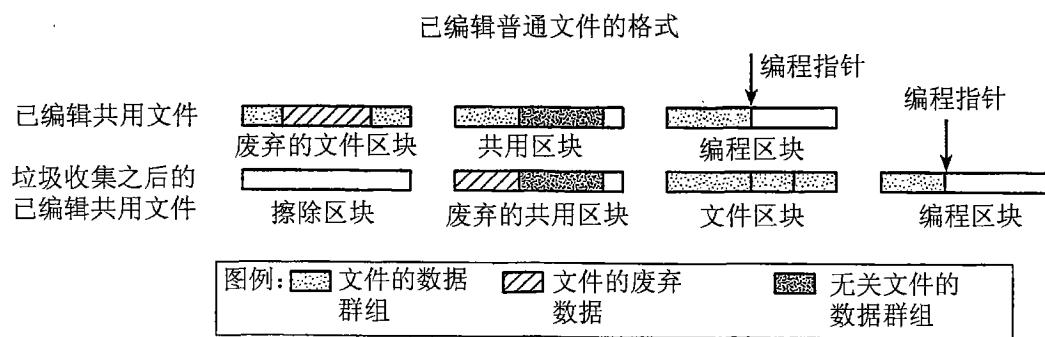


图3-4

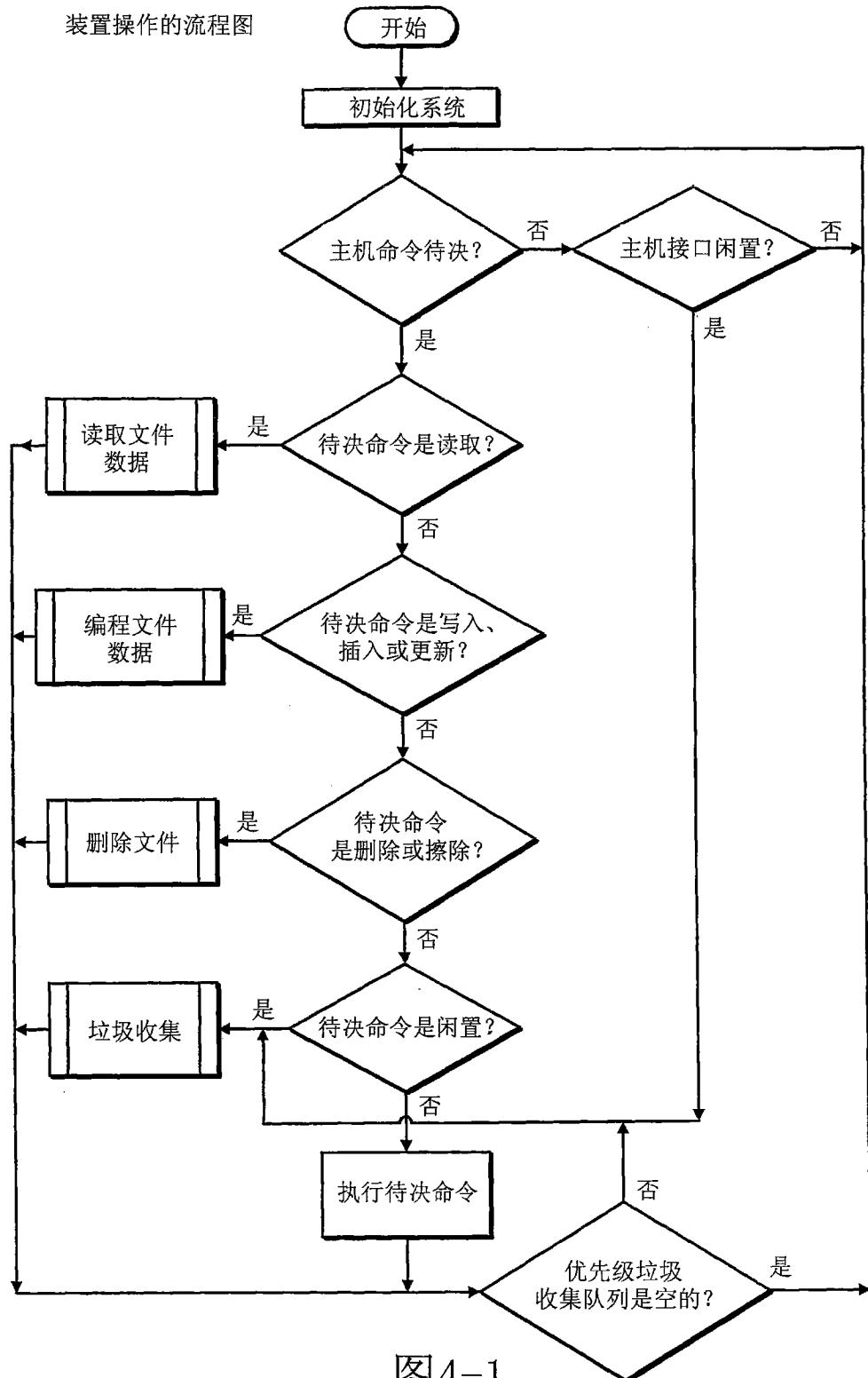


图 4-1

用于编程文件数据的流程图

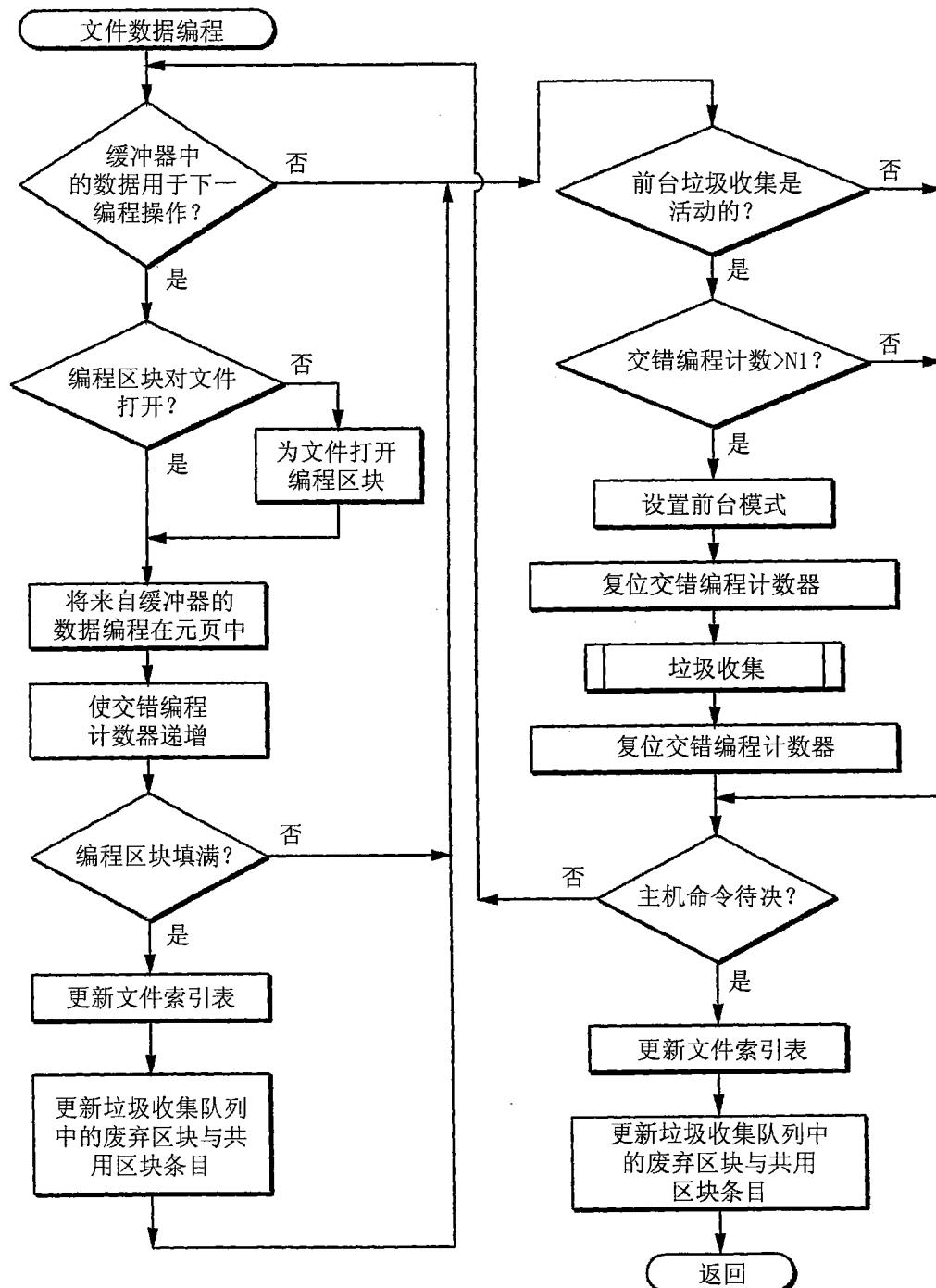


图5-1

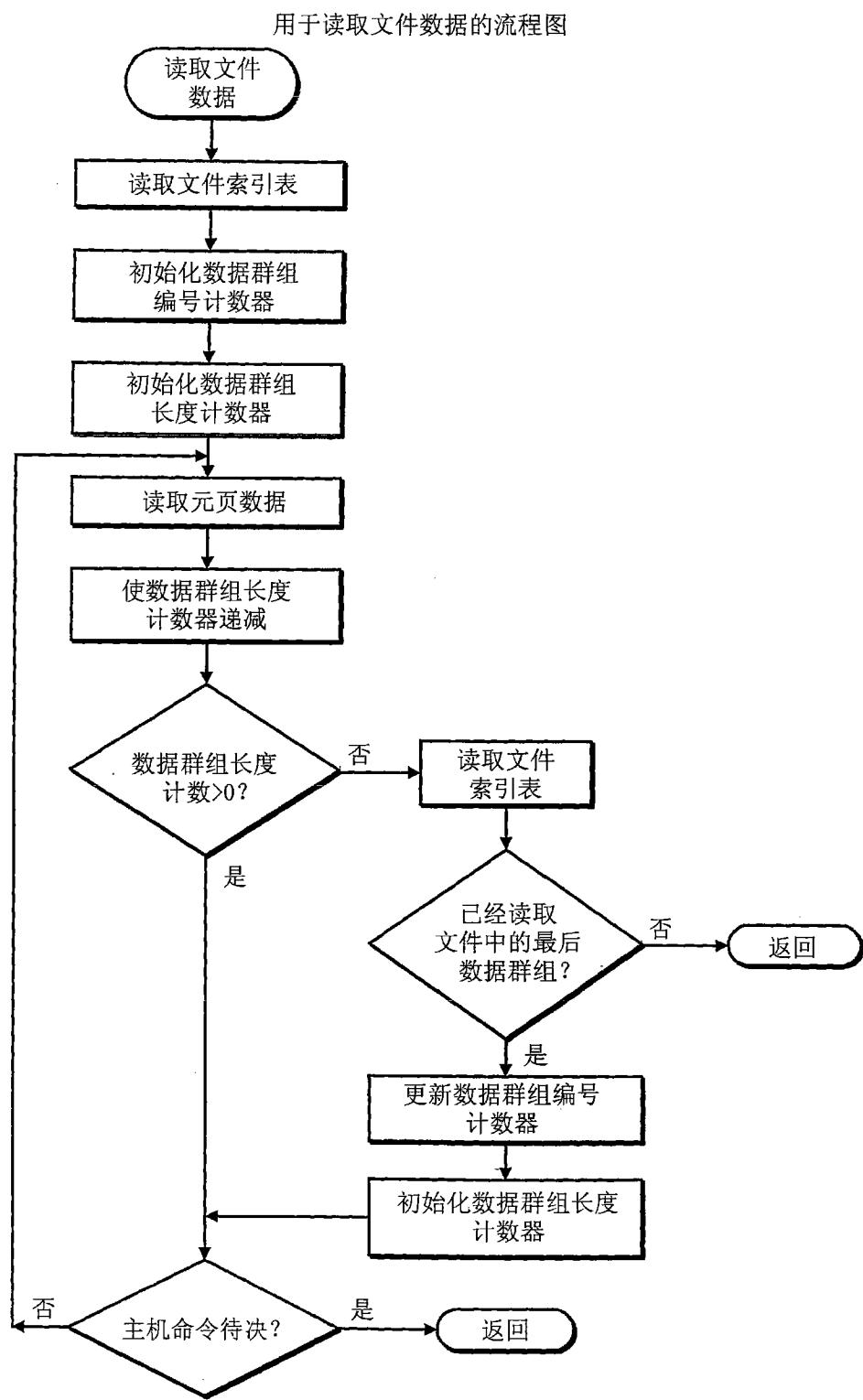


图6-1

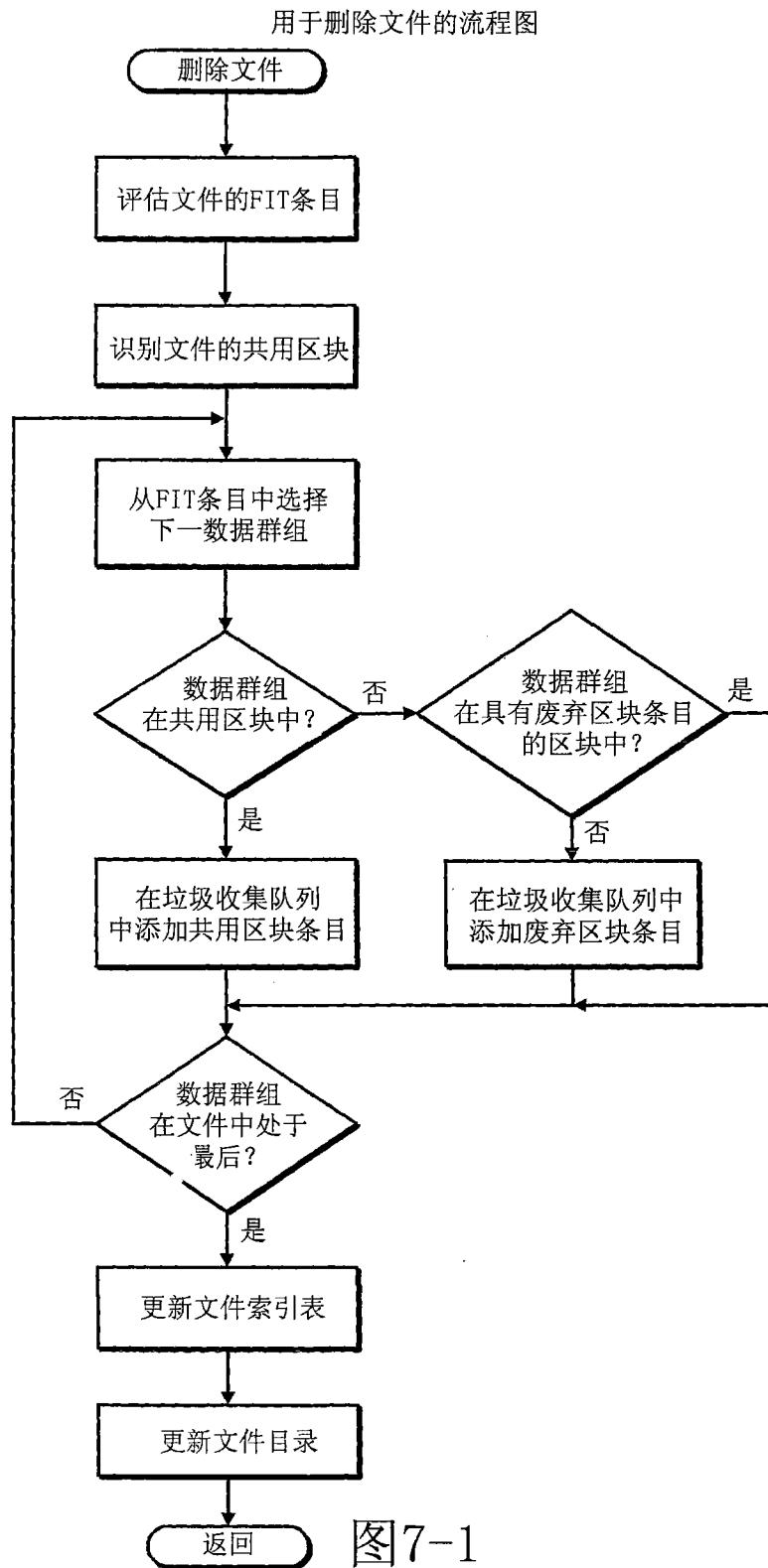
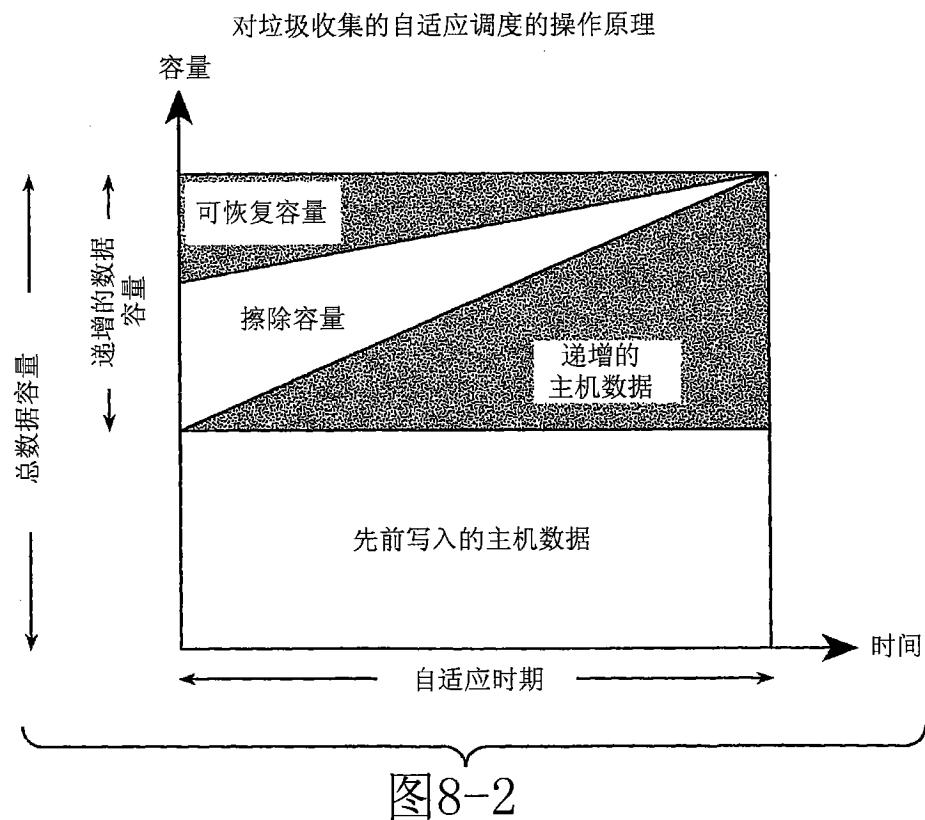
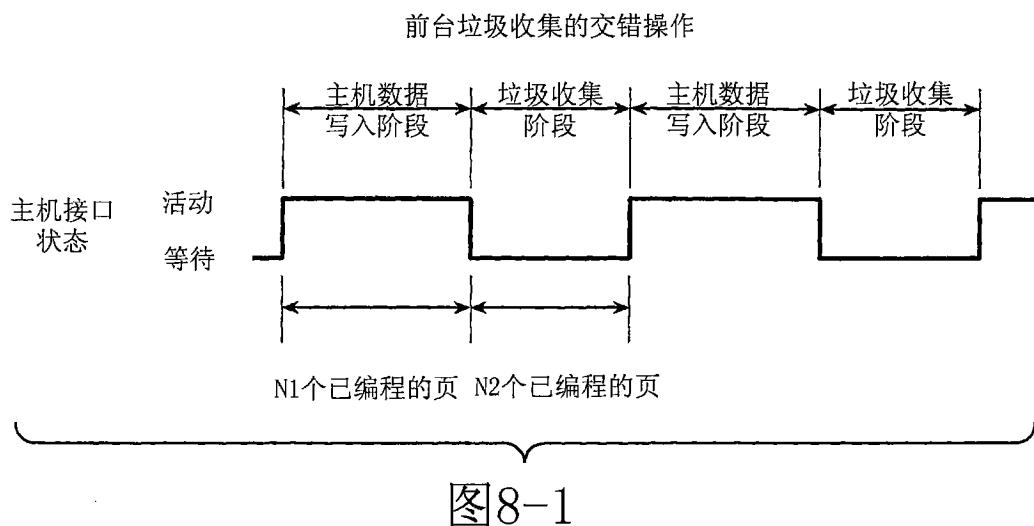


图 7-1



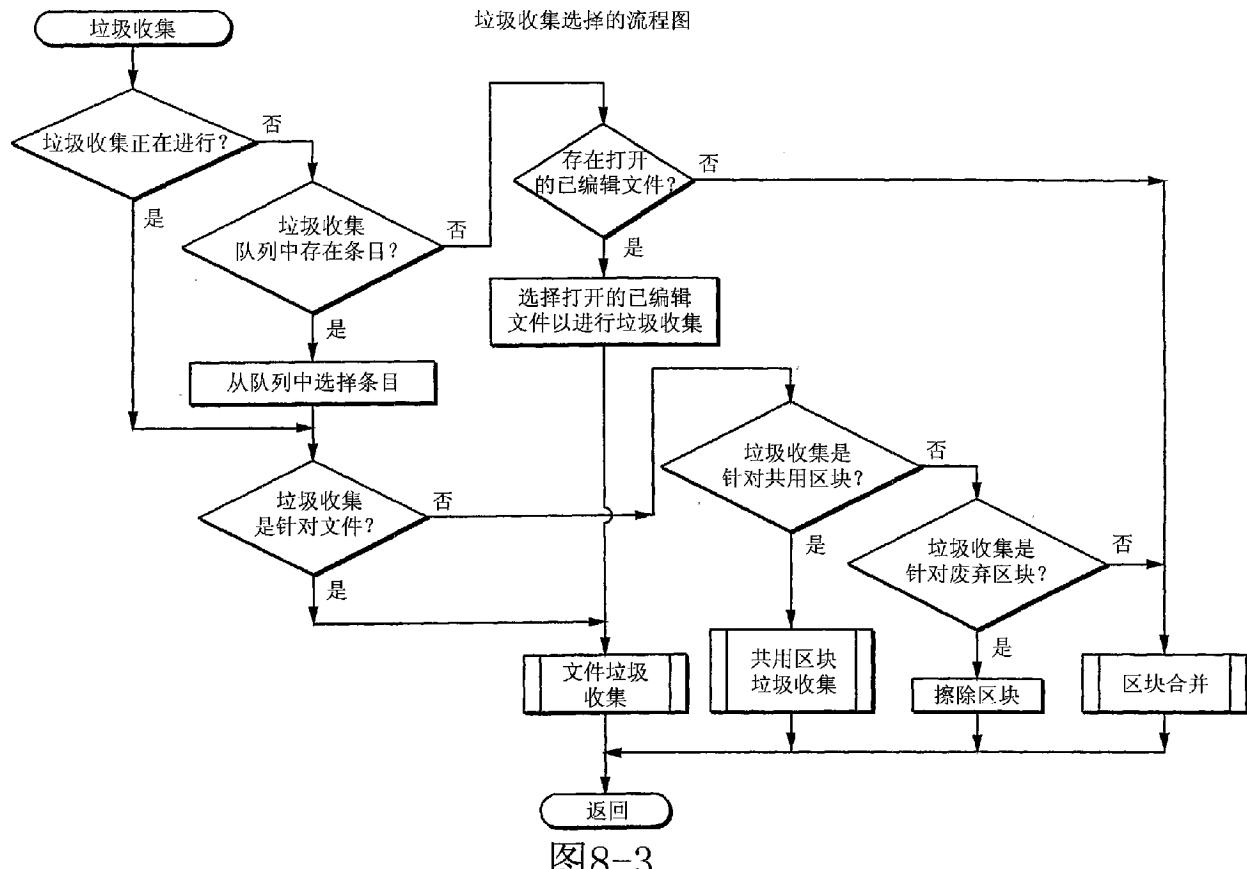


图8-3

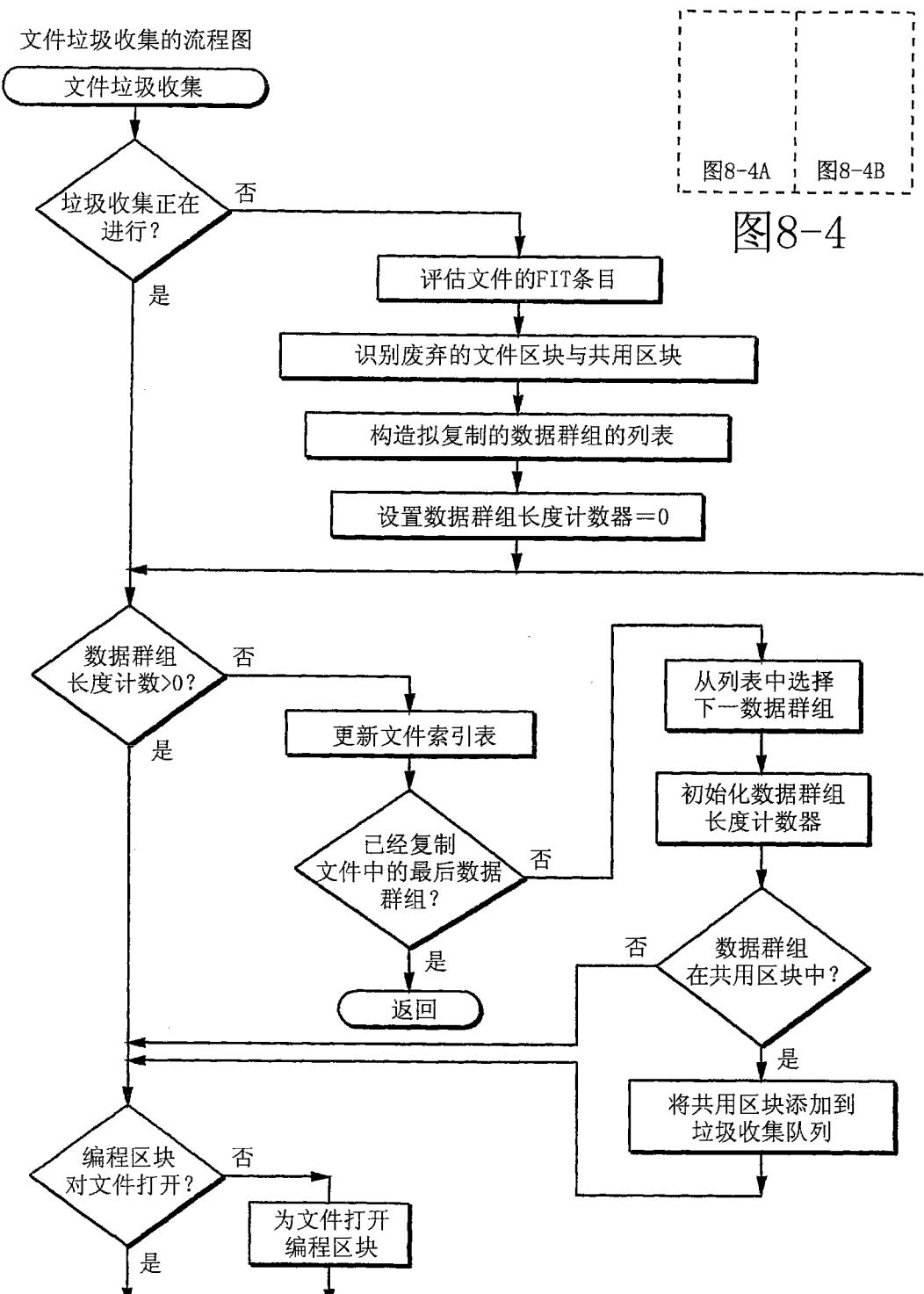


图8-4A

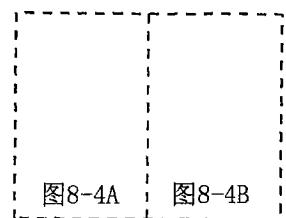


图8-4

图8-4A

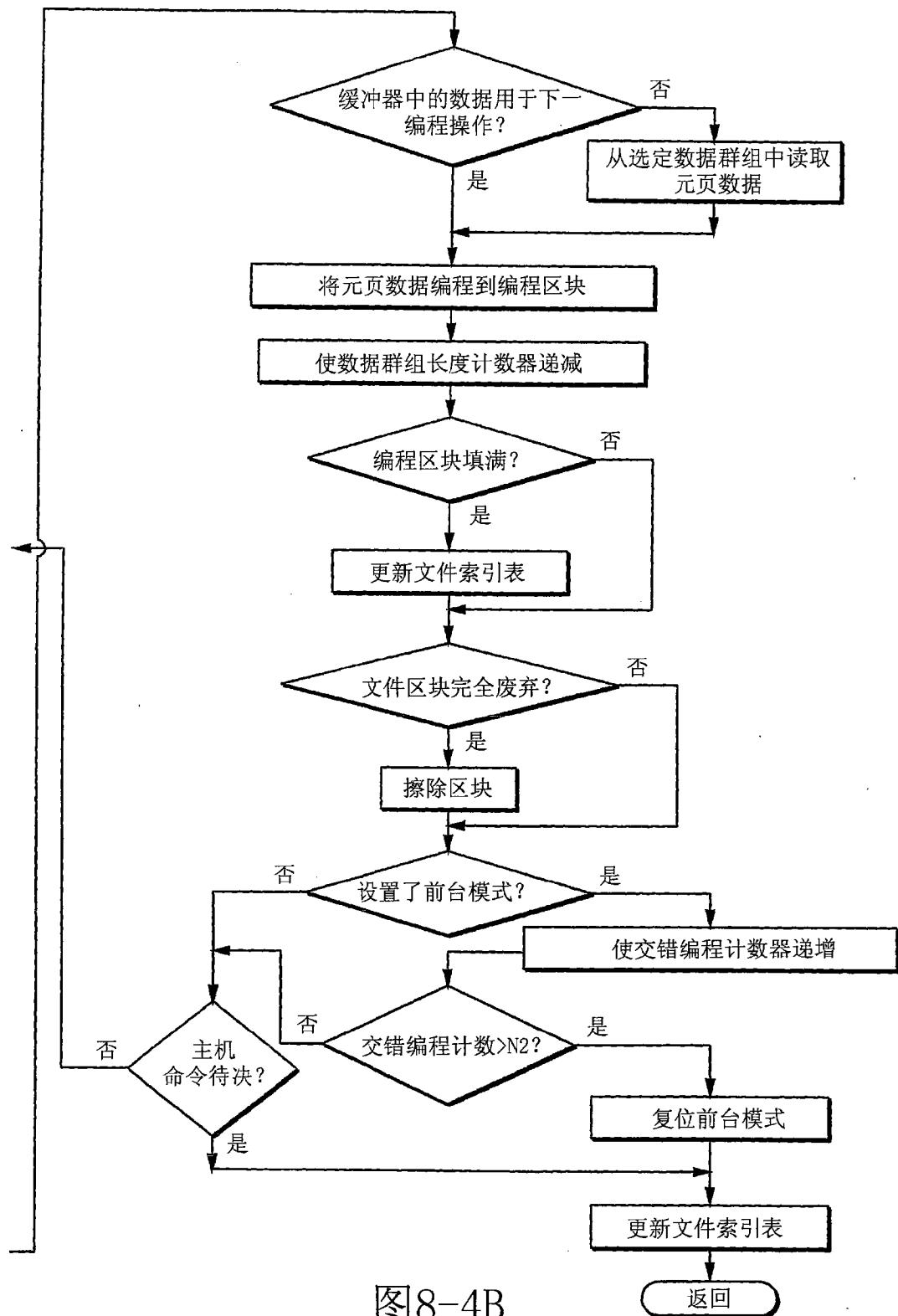


图8-4B

返回

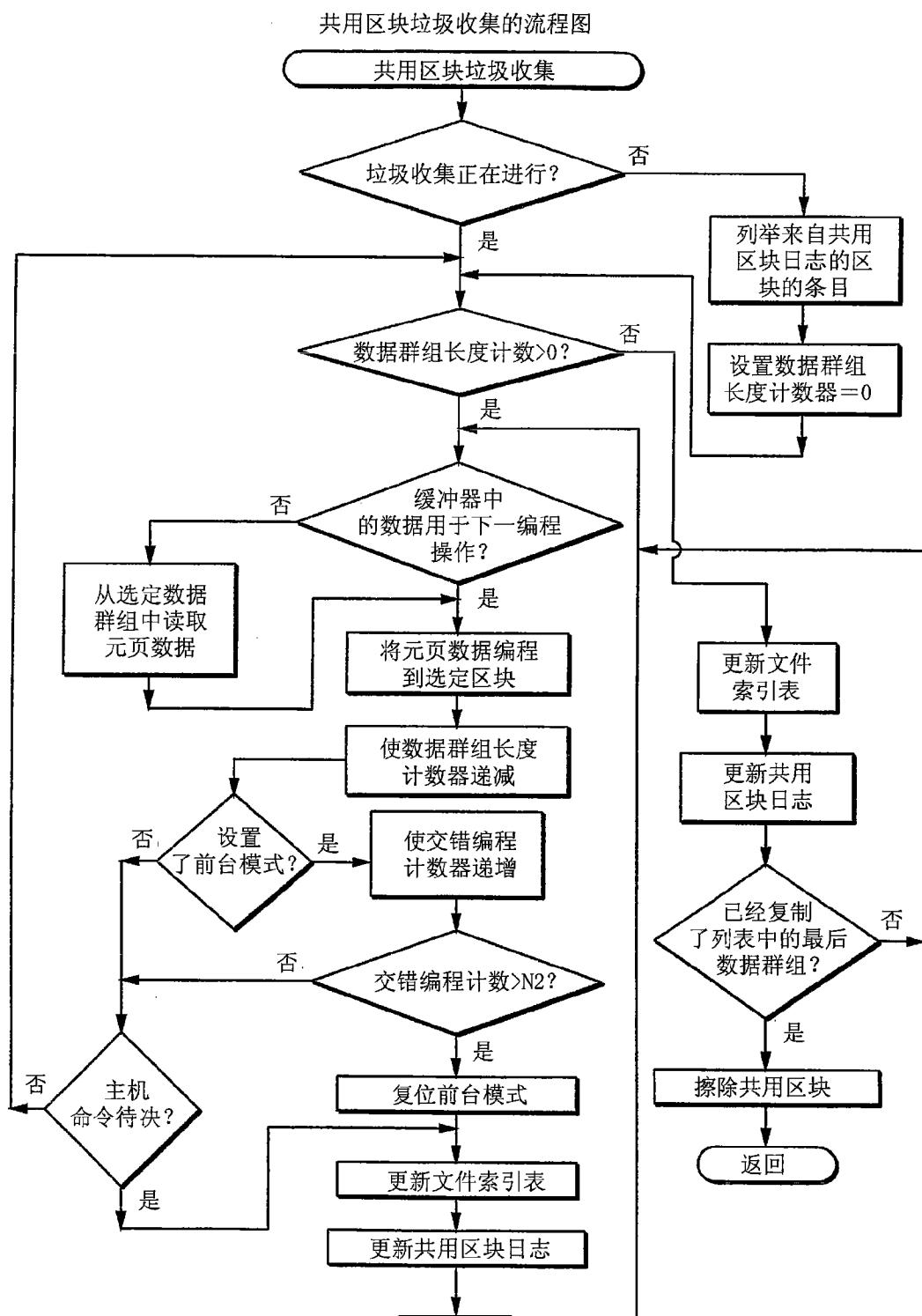


图8-5A

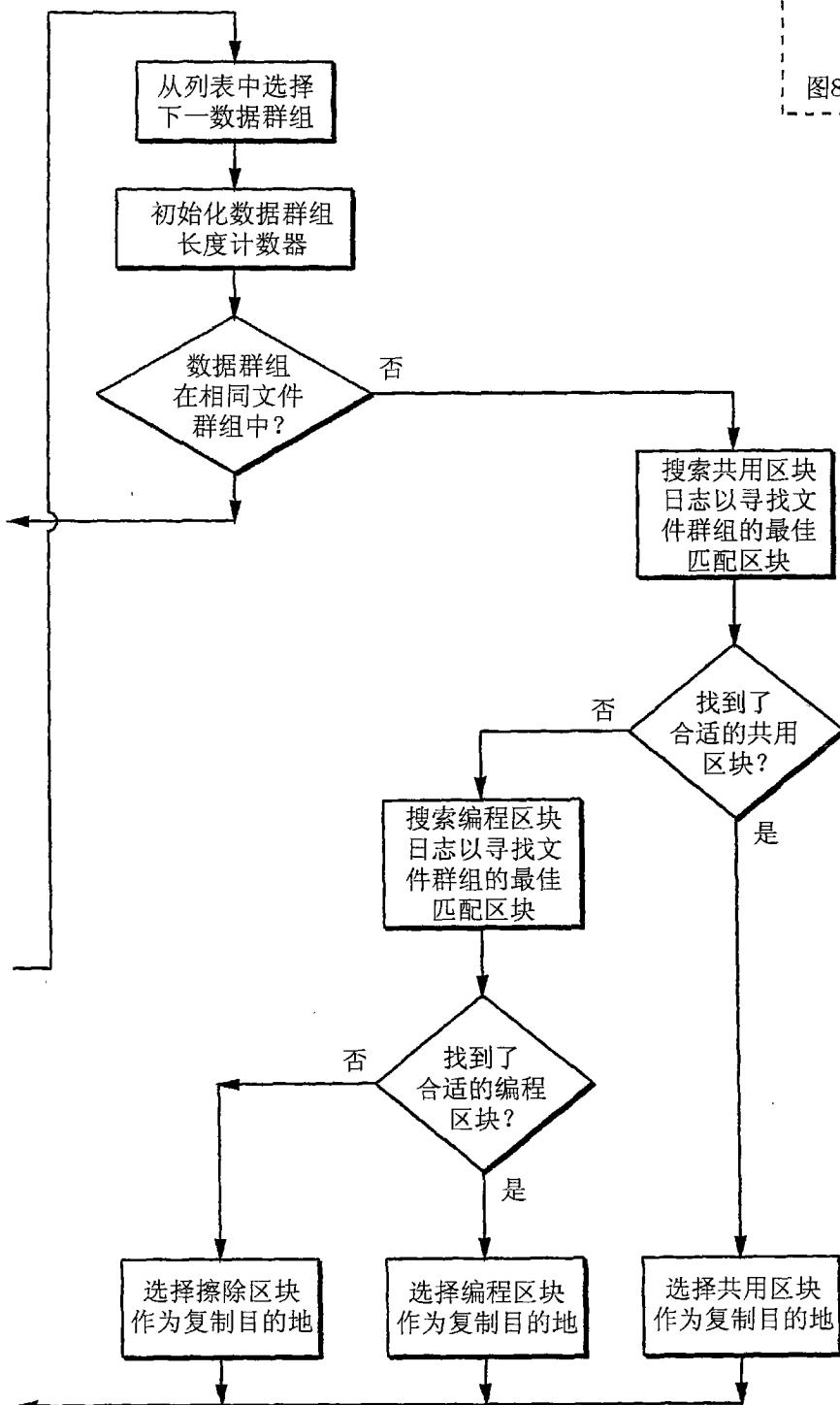


图8-5A 图8-5B

图8-5

图8-5B

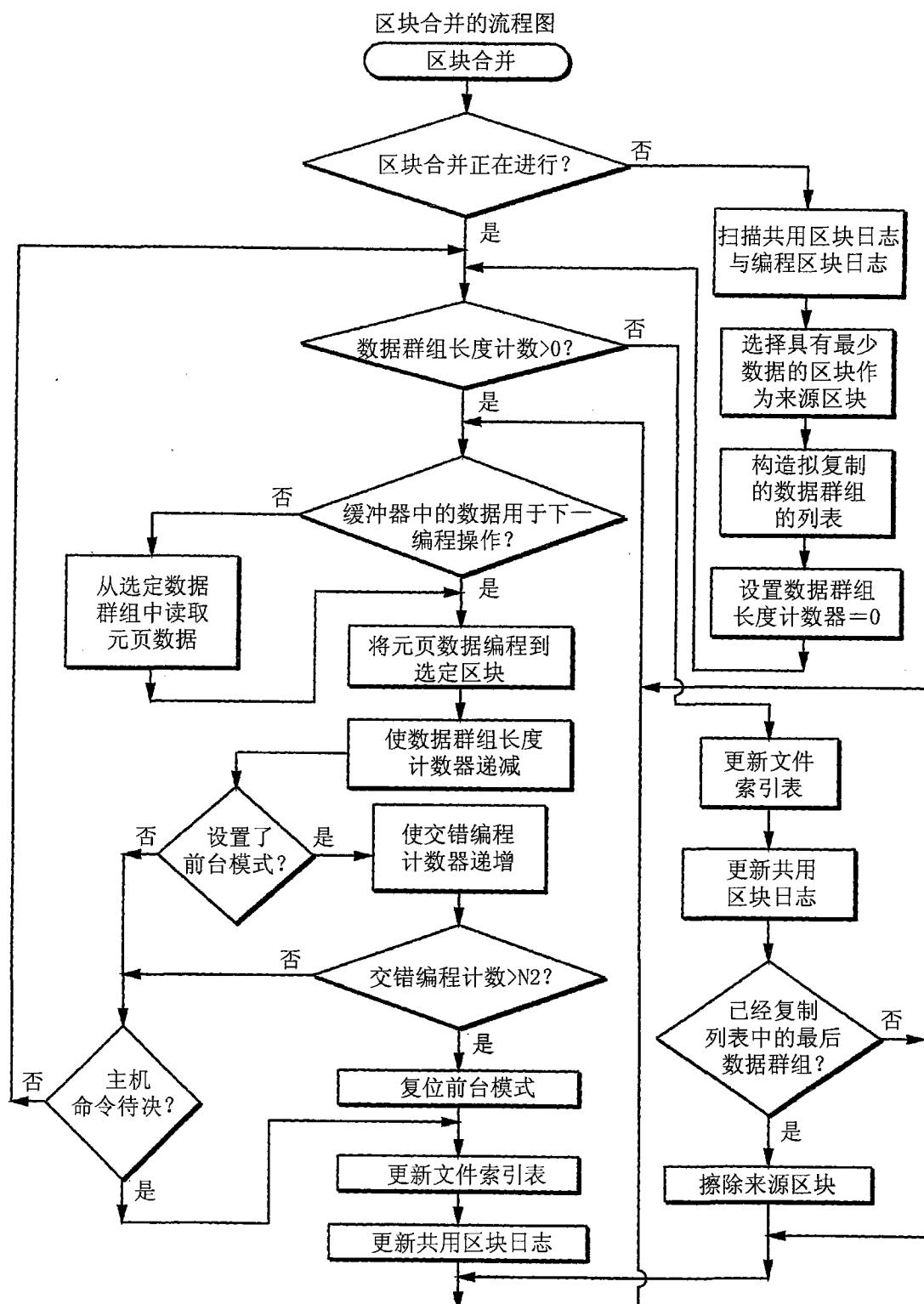


图8-6A

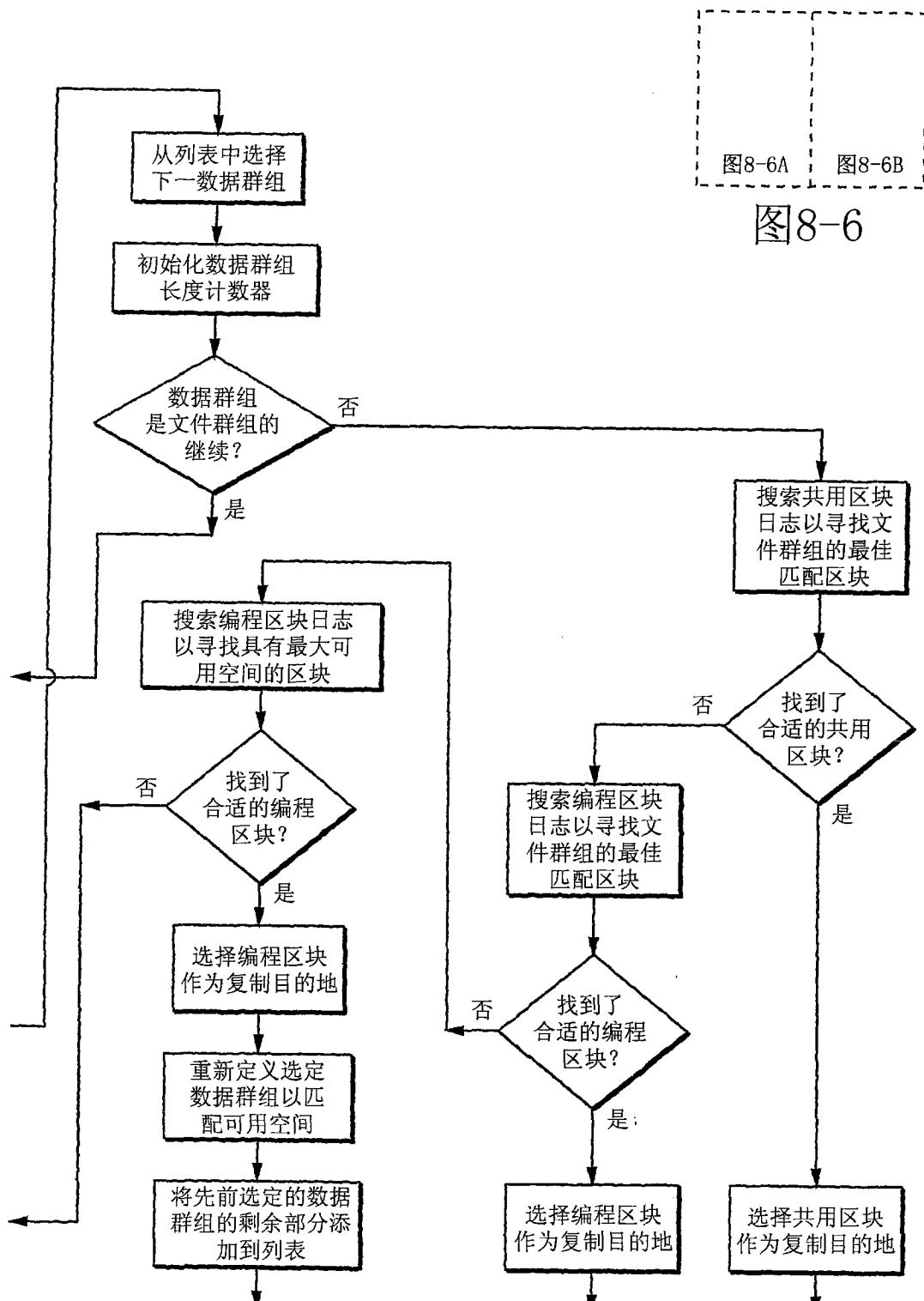


图8-6B

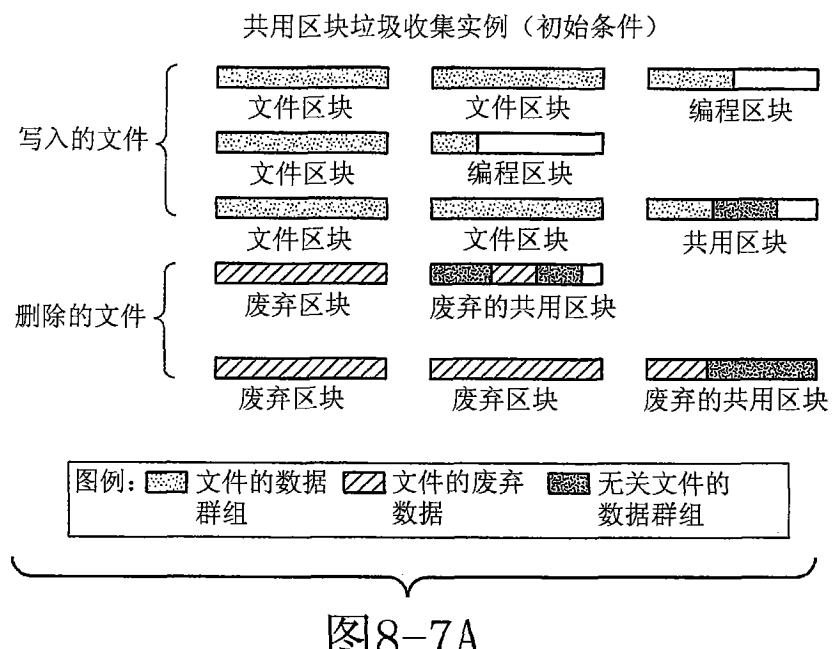


图8-7A

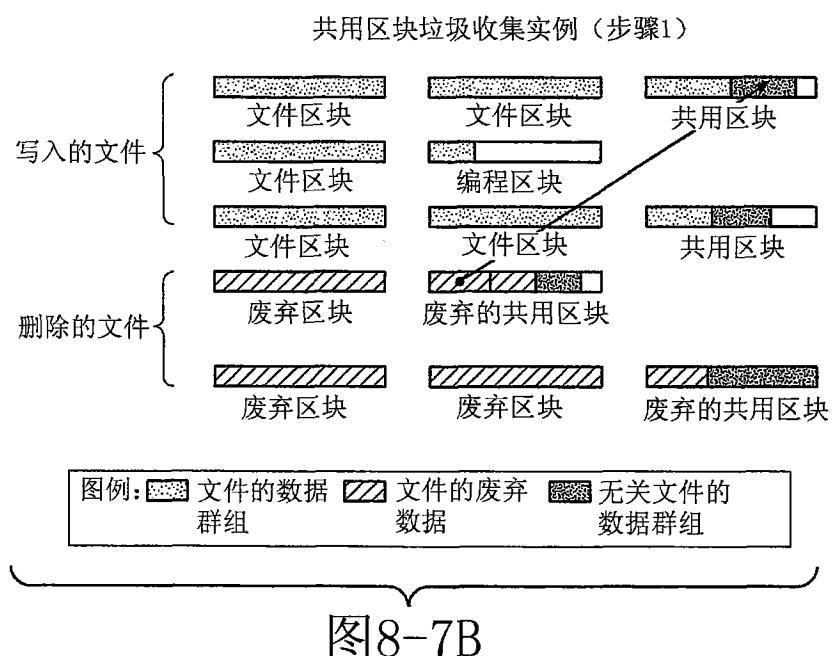
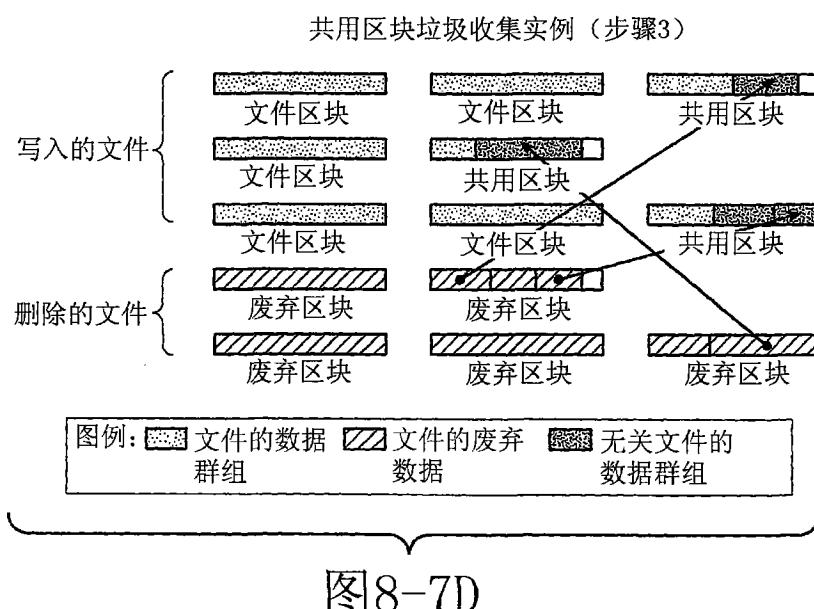
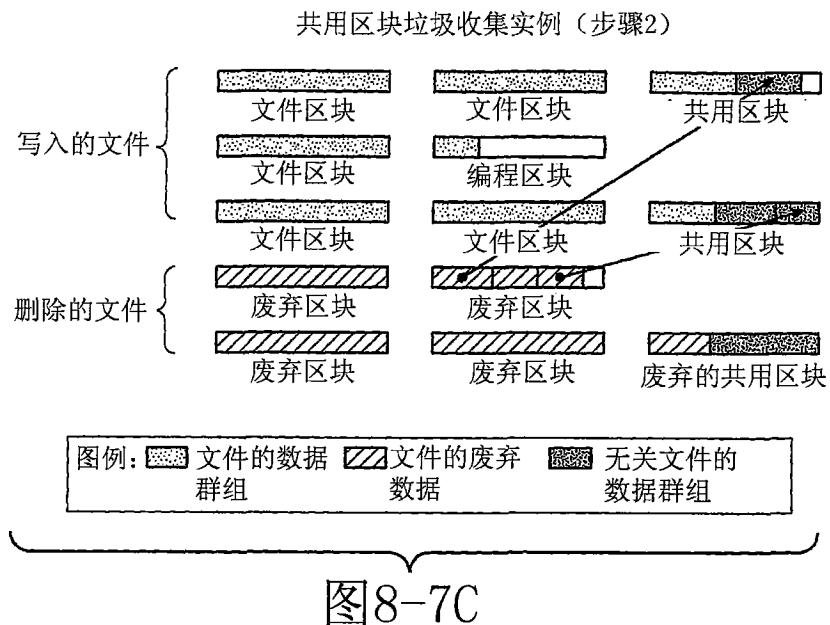
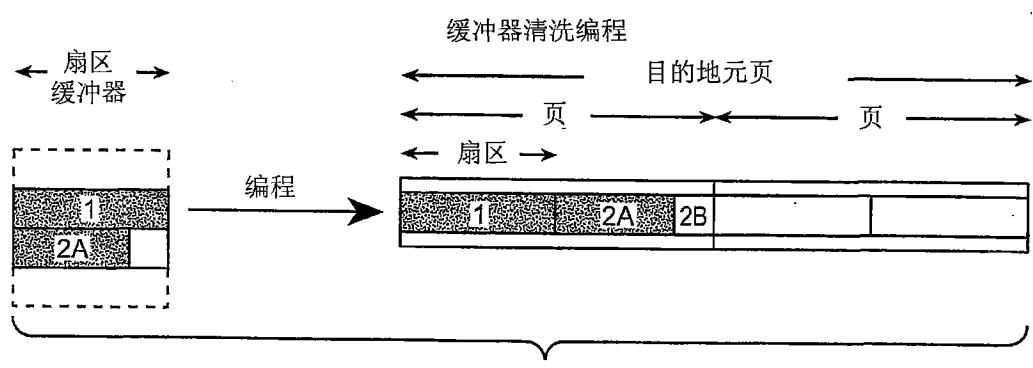
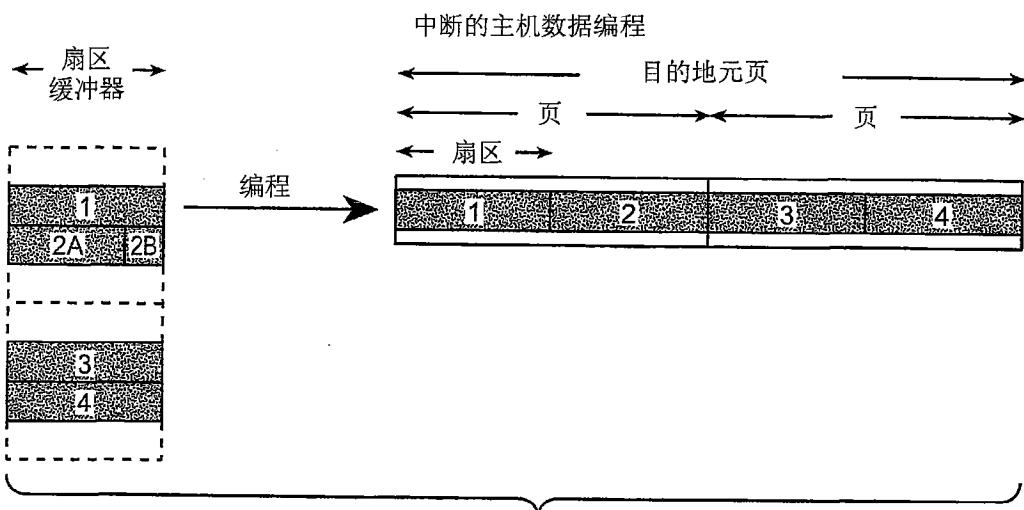
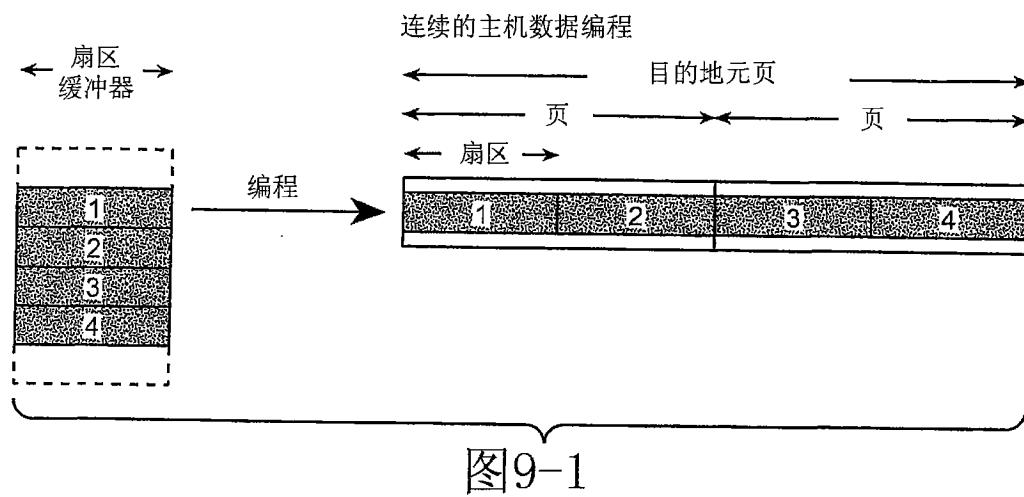
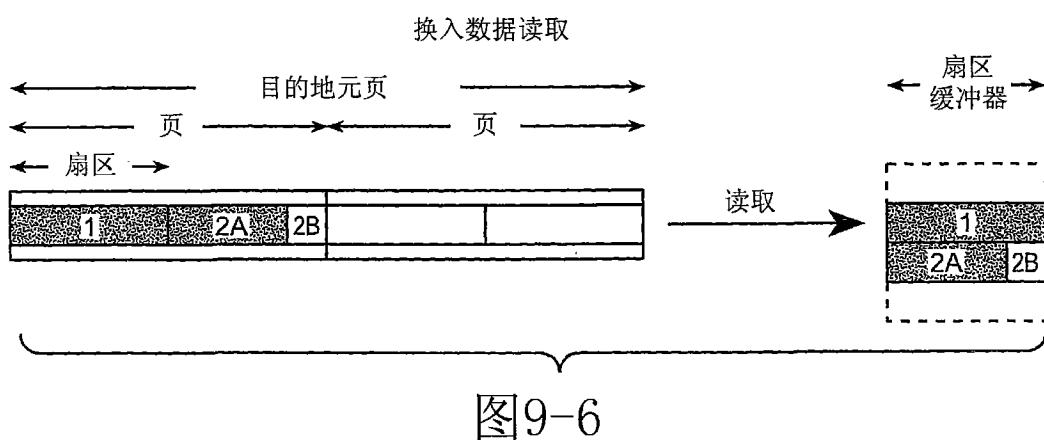
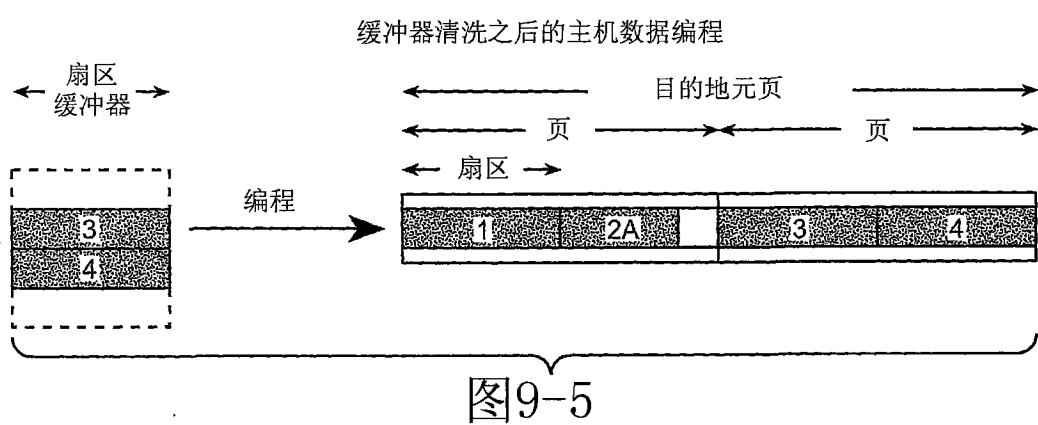
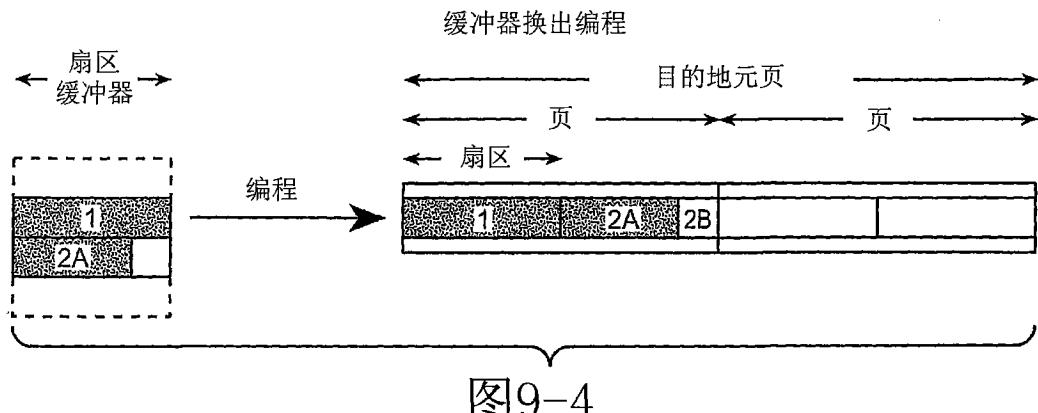


图8-7B







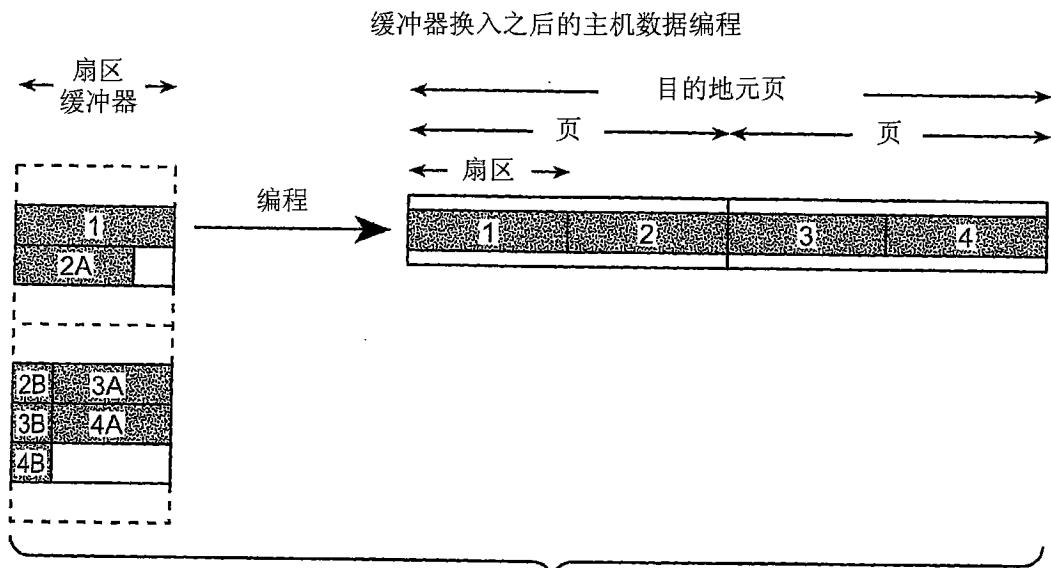


图9-7

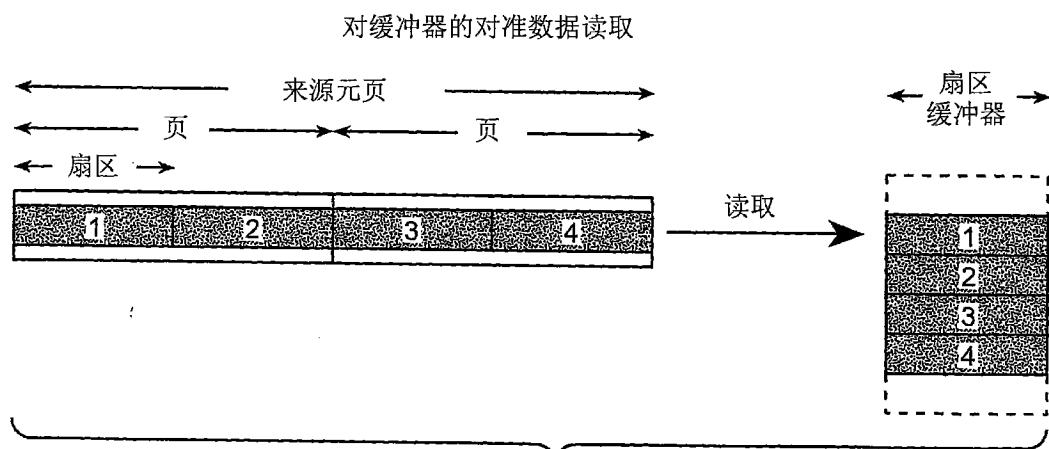


图9-8

从缓冲器的对准数据编程

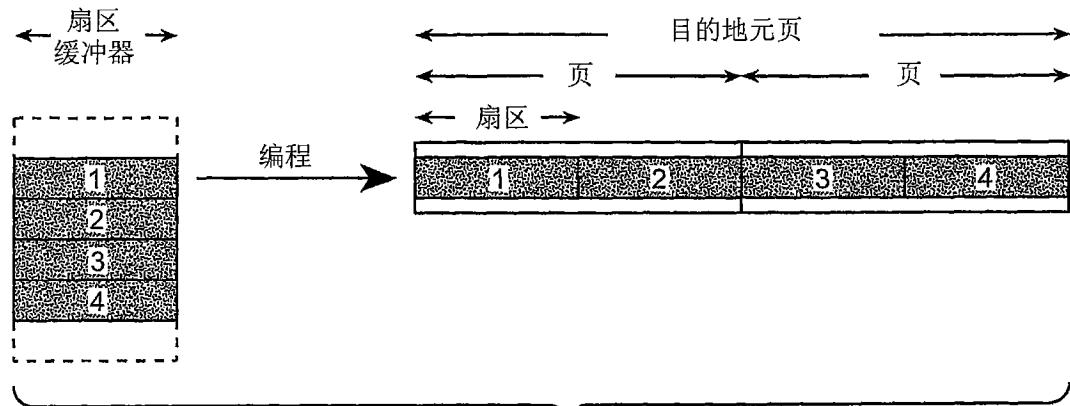


图9-9

对缓冲器的未对准数据读取

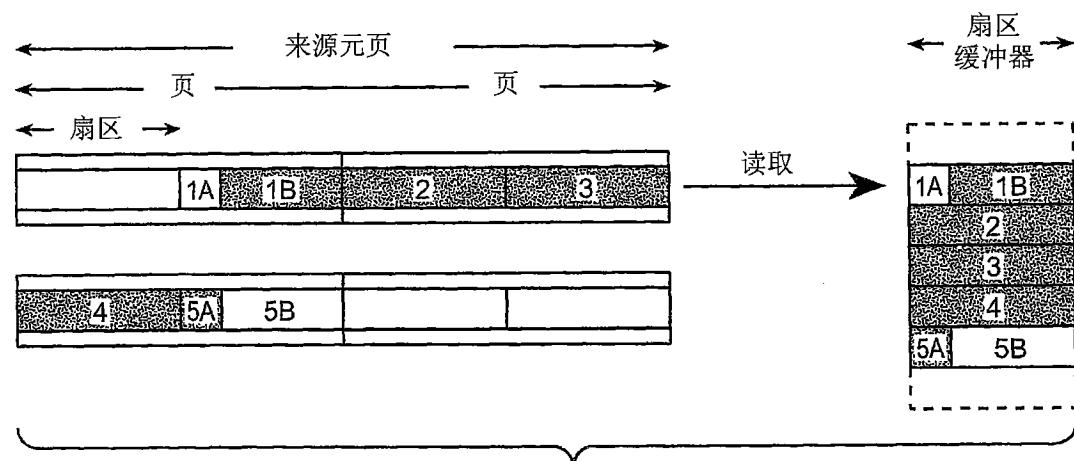
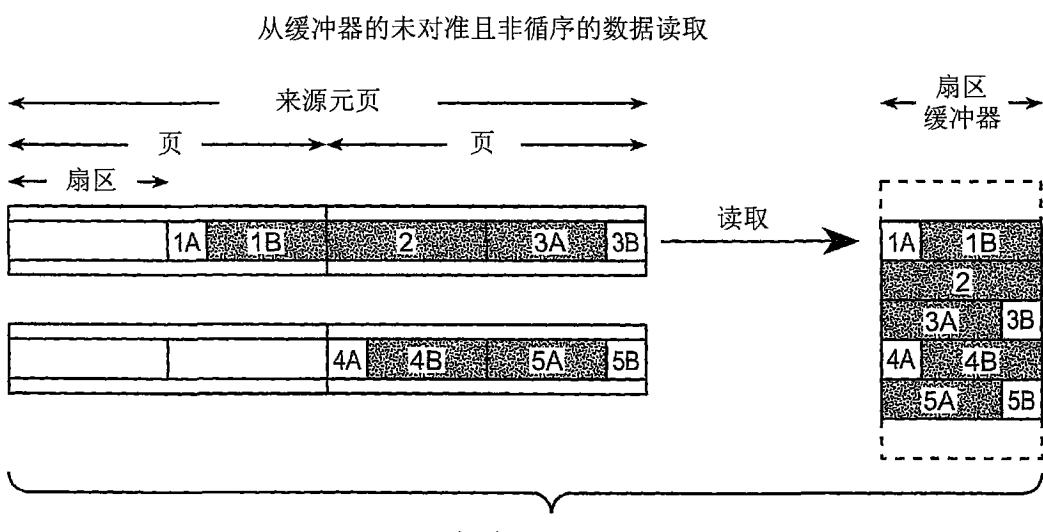
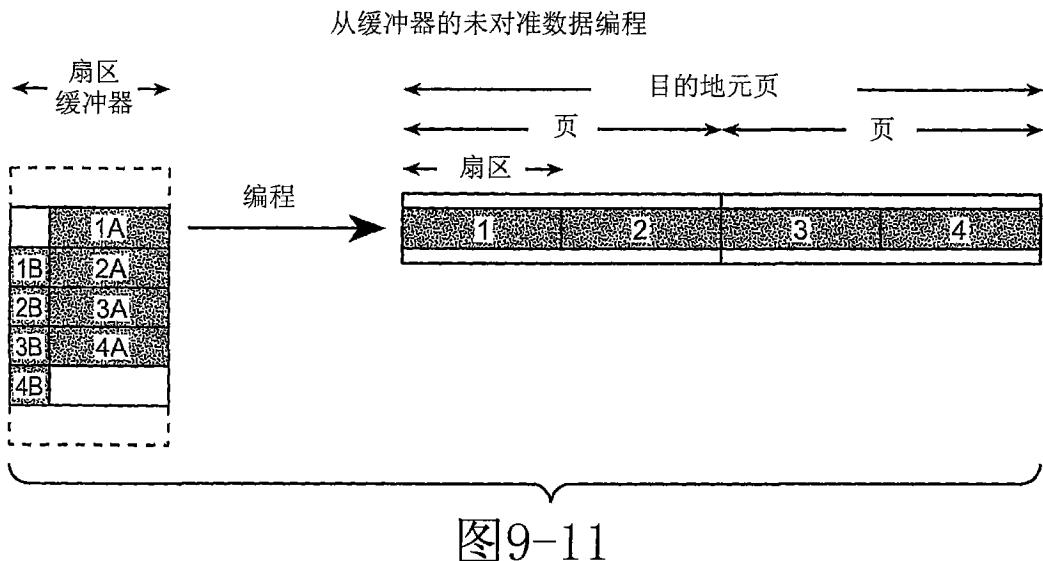


图9-10



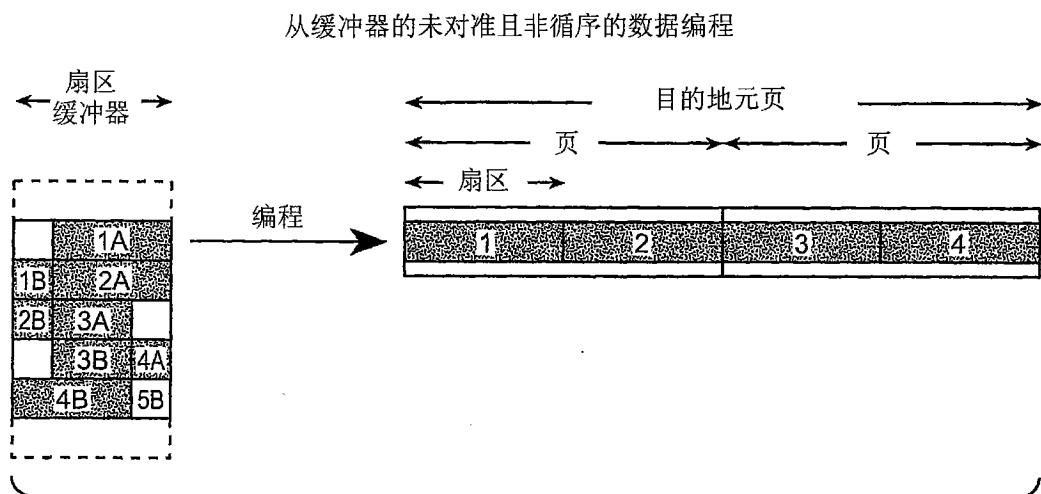


图9-13

文件索引

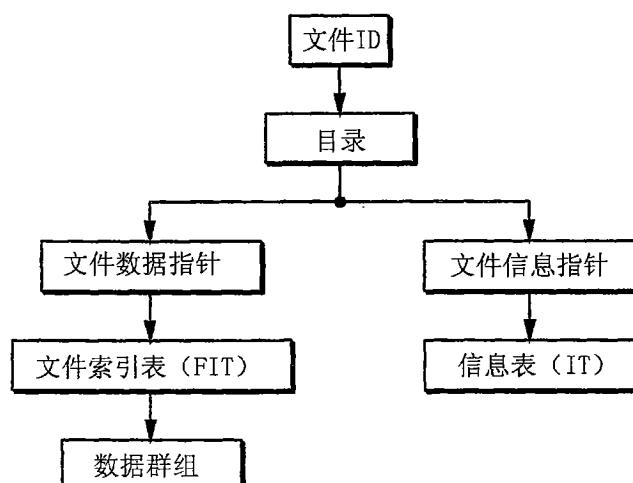


图10-1

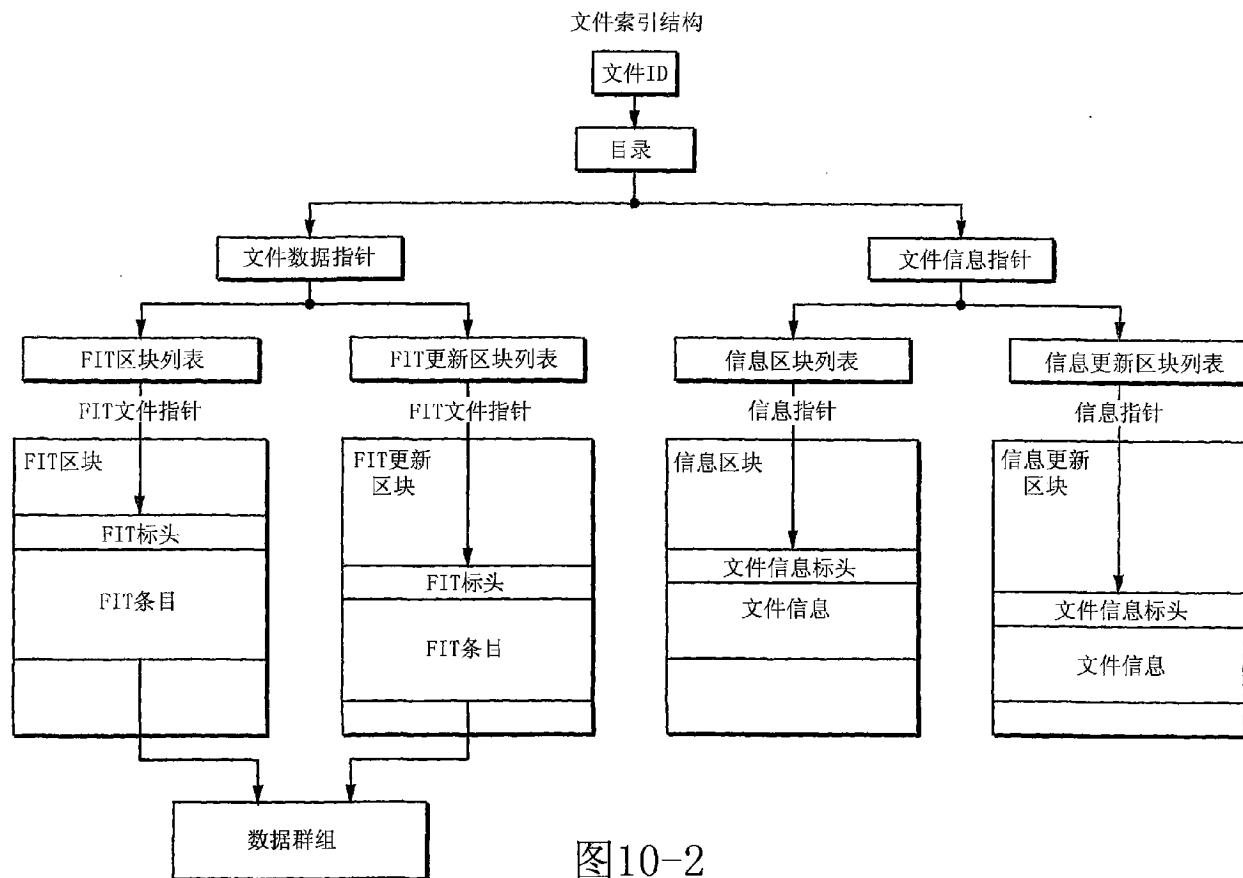


图 10-2

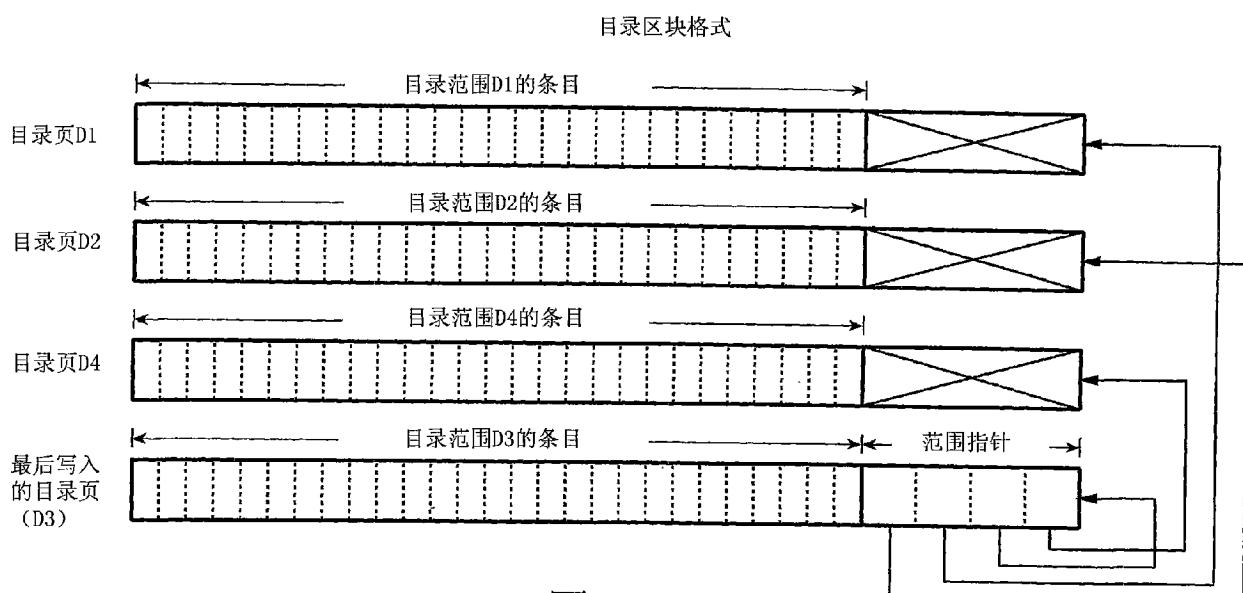


图 10-3

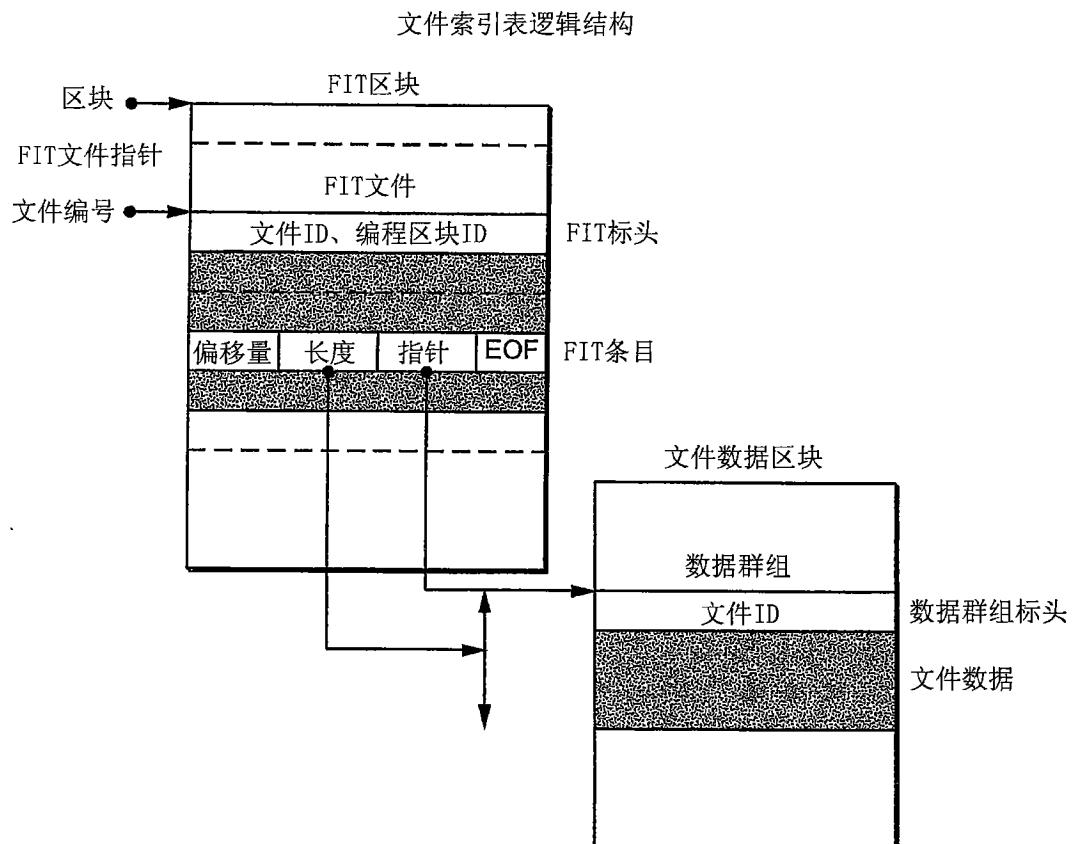


图10-4

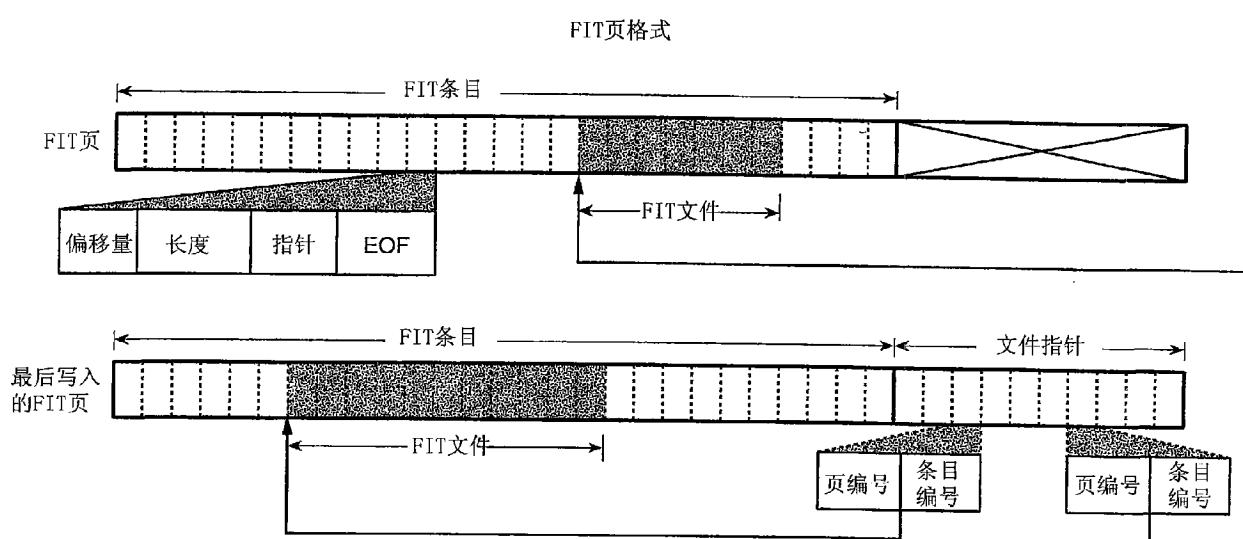


图10-5

物理FIT区块

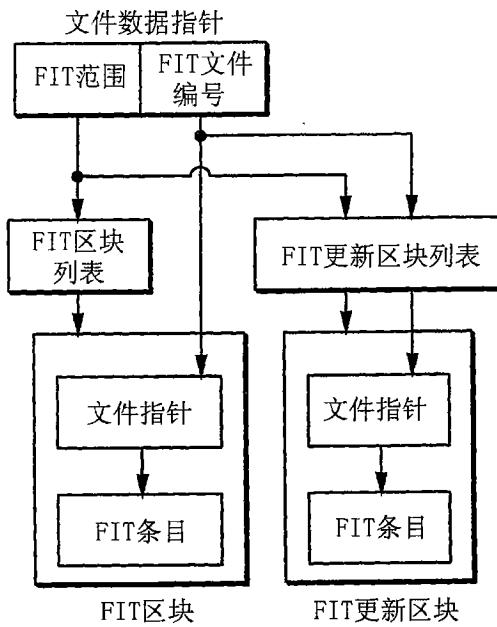


图 10-6

FIT文件更新操作的实例

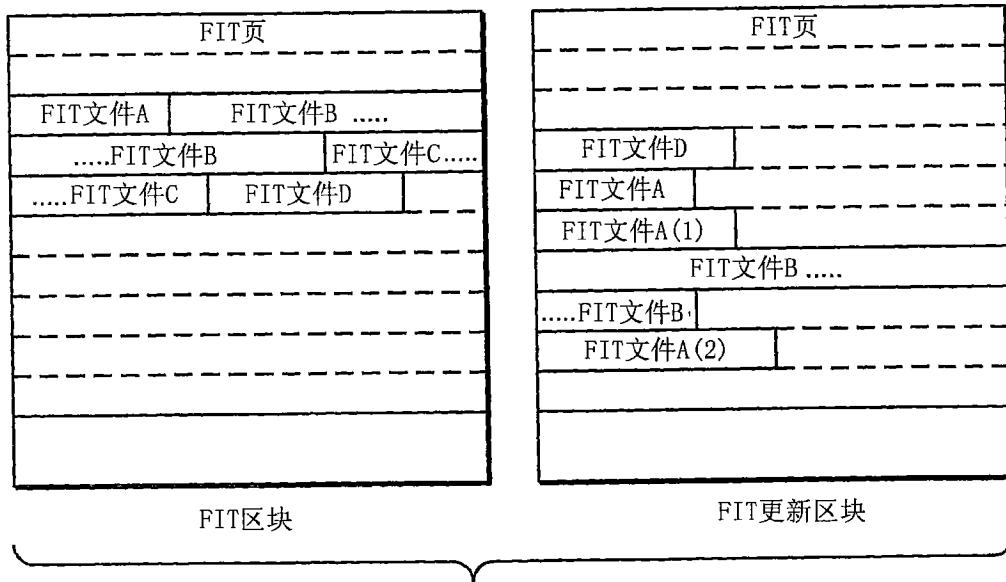
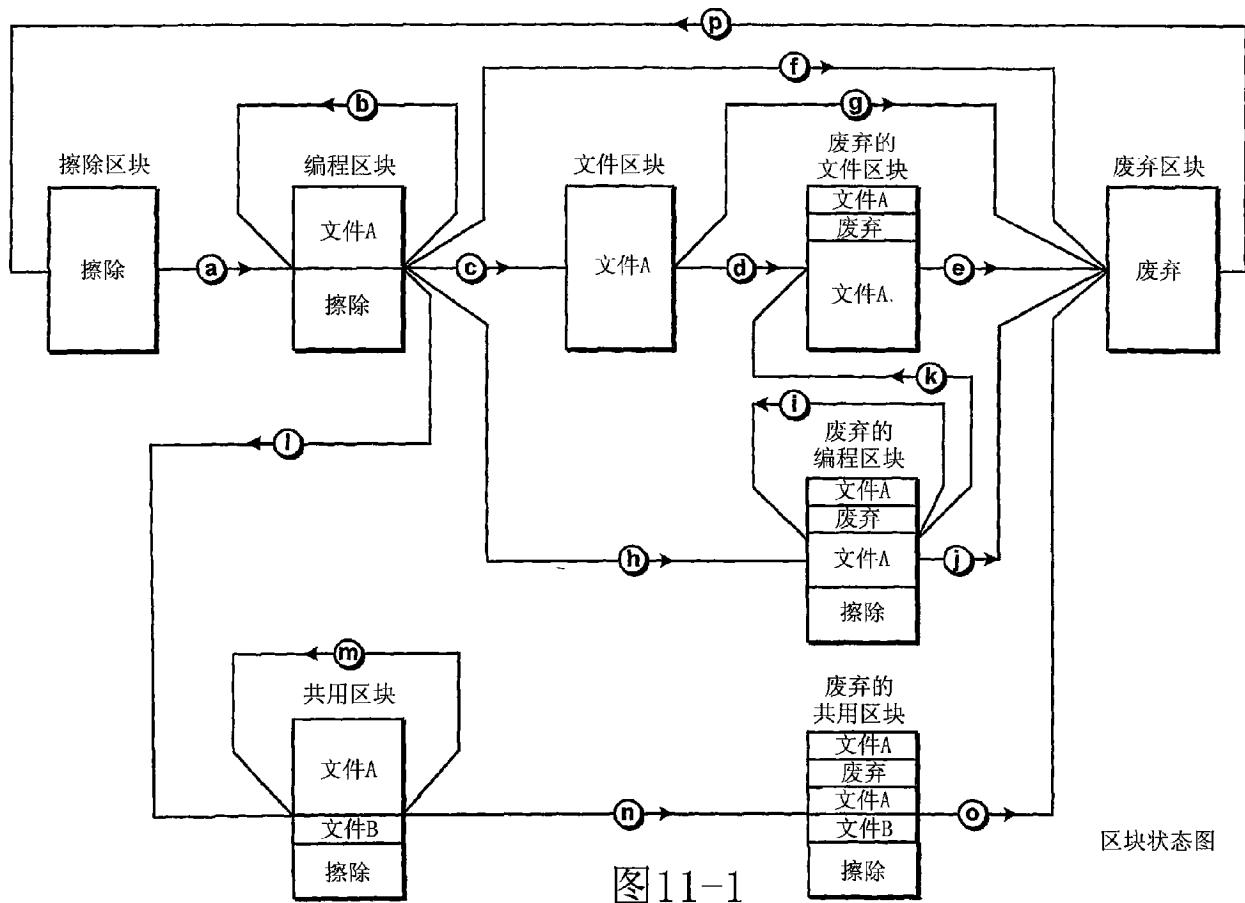
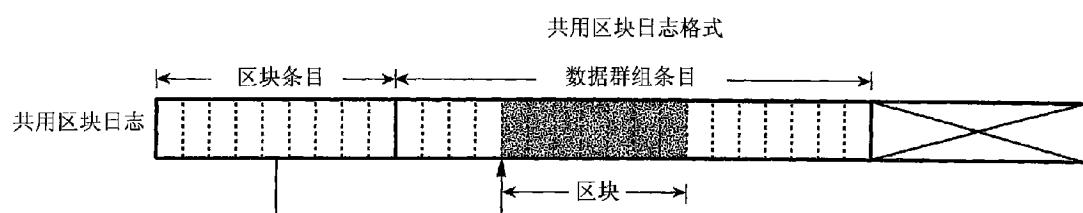
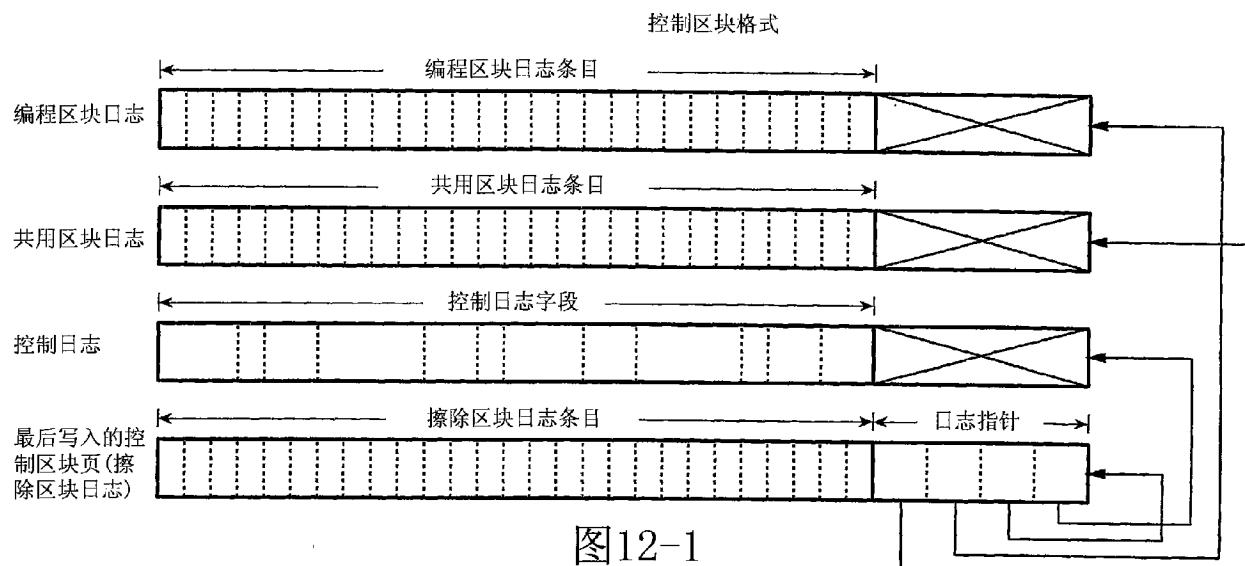


图 10-7





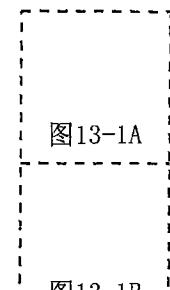


图13-1

与静态文件一起使用的命令集（A部分）

命令	参数	说明
创建	<文件ID>	在装置的目录中创建通过<文件ID>识别的条目。 应当通过命令来指定<文件ID>，以表示静态文件的适当数字识别符。 可在直接数据文件装置已经根据对不存在于直接数据文件装置的目录中的文件的打开命令而返回错误状态之后使用创建命令。
打开	<文件ID>	启用对指定文件的后续数据命令的执行。 将写入_指针设置到文件末尾，且将读取_指针设置到文件开始处。 如果文件不存在，那么返回错误状态。
关闭	<文件ID>	停用对指定文件的后续数据命令的执行。 对静态文件的关闭命令导致将条目插入到废弃区块垃圾收集队列或文件垃圾收集队列中。
删除	<文件ID>	删除命令通常不与静态文件一起使用。 然而，如果为静态文件指派的装置分区的大小减少，可使用删除命令。
列举_文件		以数字次序从装置读取所有有效的<文件ID>值。
写入	<文件ID>	复写由写入_指针的当前值定义的偏移地址处的指定静态文件中的数据。 偏移地址不应当在对应于主机对所有静态文件使用的文件大小的范围之外。
读取	<文件ID>	从由读取_指针的当前值定义的偏移地址处的指定文件中读取数据。 偏移地址不应当在对应于主机对所有静态文件使用的文件大小的范围以外。

图13-1A

与静态文件一起使用的命令集 (B部分)

命令	参数	说明
保存_缓冲器	<文件ID>	将指定静态文件的处于缓冲器中的数据保存到快闪存储器中的临时位置。 当用于静态文件的协议要求必须将已经为文件供应的数据提交到快闪存储器时使用保存_缓冲器。
写入_指针	<文件ID><偏移量>	为指定文件定义写入_指针的新的当前值。 对于文件打开之后的第一数据或对于与先前写入的数据不连续的数据，使用写入_指针来定义拟写入的数据的静态文件内的偏移量。 偏移量不应当在对应于主机对所有静态文件使用的文件大小的范围以外。
读取_指针	<文件ID><偏移量>	为指定文件定义读取_指针的新的当前值。 对于文件打开之后的第一数据或对于与先前读取的数据不连续的数据，使用读取_指针来定义拟读取的数据的静态文件内的偏移量。 偏移量不应当在对应于主机对所有静态文件使用的文件大小的范围以外。
流动	<长度>	定义将传递到装置或从装置传递出的不中断数据流。 只用于建模。
暂停	<时间>	定义在执行随后命令之前插入的延迟。 只用于建模。
闲置		装置应当进入闲置状态，在闲置状态中装置可执行内部操作。 在对静态文件的关闭命令之后立刻使用闲置命令，以强制执行任何待决的垃圾收集操作。 不应当将其它命令发送到装置，直到装置展示后台操作的不繁忙状态为止。
待命		装置应当进入待命状态，在待命状态中装置可以不执行内部操作。
关机		当装置接下来不处于繁忙状态时，装置将关机且电力将被移除。
容量		来自主机的让装置报告由文件数据占据的容量和可供新文件数据使用的容量的请求。
状态		来自主机的让装置报告其当前状态的请求。 状态命令不终止正在执行的命令。

图13-1B