

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G11C 16/06 (2006.01)

G11C 16/10 (2006.01)

G11C 16/14 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510101856.9

[43] 公开日 2006年8月9日

[11] 公开号 CN 1815629A

[22] 申请日 2005.11.25

[21] 申请号 200510101856.9

[71] 申请人 康佳集团股份有限公司

地址 518053 广东省深圳市南山区华侨城康佳集团股份有限公司

[72] 发明人 张建文

[74] 专利代理机构 深圳创友专利商标代理有限公司

代理人 郭 燕

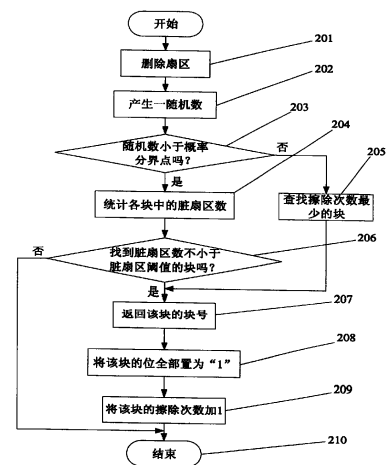
权利要求书 2 页 说明书 6 页 附图 3 页

[54] 发明名称

一种闪存装置的脏块回收方法

[57] 摘要

本发明公开了一种闪存装置的脏块回收方法，包括写扇区时进行的强制回收步骤和删除扇区后进行的条件回收步骤，其中强制回收是回收脏扇区最多的块，条件回收又分为回收脏扇区数达到脏扇区阈值的块和擦除次数最小的块。利用本发明既能够尽可能减少块的擦除次数，又能做到均匀擦除，以使所有物理块的使用寿命均衡。



1. 一种闪存装置的脏块回收方法，其特征在于：包括写扇区时进行的强制回收步骤和删除扇区后进行的条件回收步骤。
2. 如权利要求 1 所述的闪存装置的脏块回收方法，其特征在于：所述强制回收步骤包括以下步骤：
 - A1、闪存装置接收写操作指令；
 - B1、闪存装置中的闪存存储管理模块查找可写入的空闲扇区；
 - C1、如果没有足够的空闲扇区，则闪存存储管理模块查找出脏扇区最多的块进行回收。
3. 如权利要求 2 所述的闪存装置的脏块回收方法，其特征在于：所述条件回收步骤包括以下步骤：
 - A2、闪存装置接收删除指令，闪存存储管理模块对指定扇区进行删除；
 - B2、闪存存储管理模块随机产生一 0~99 内的随机数；
 - C2、将该随机数与预先设定的概率分界点进行比较，如果该随机数大于或等于概率分界点，则执行步骤 E2，如果该随机数小于概率分界点，则执行步骤 D2，其中概率分界点的取值范围为 1~100；
 - D2、将各块的脏扇区数与预先设定的脏扇区阈值进行比较，对于脏扇区数大于脏扇区阈值的块执行步骤 F2，其中脏扇区阈值为小于每个块所包含的扇区数的正整数；
 - E2、闪存存储管理模块查找出擦除次数最少的块并执行步骤 F2；
 - F2、对该块进行回收。
4. 如权利要求 3 所述的闪存装置的脏块回收方法，其特征在于：在步骤 A2 之后还包括将该被删除扇区加上“脏”标记的步骤。
5. 如权利要求 3 所述的闪存装置的脏块回收方法，其特征在于：所述随机数的产生方法是生成一任意正整数，然后用 100 去除，取其整数部分作为随机数。
6. 如权利要求 3 所述的闪存装置的脏块回收方法，其特征在于：所述概率分界点是 97。
7. 如权利要求 3 至 6 中任一项所述的闪存装置的脏块回收方法，其特征在于：在步骤 D2 或 E2 之后、步骤 F2 之前还包括返回该块的块号的步骤。

8. 如权利要求 7 所述的闪存装置的脏块回收方法, 其特征在于: 在步骤 F2 之后还包括将该块的擦除次数加 1 的步骤。

一种闪存装置的脏块回收方法

【技术领域】

本发明涉及半导体存储领域，尤其涉及一种闪存装置的脏块回收方法。

【背景技术】

闪存存储器(Flash Memory)具有高密度和良好的存取速度等优点，是嵌入式领域应用最为广泛的存储设备，属于EEPROM(电擦除可编程只读存储器)类型。常用的闪存介质有nand flash和nor flash两种，他们都将存储空间划分为特定大小的擦除块，如16KB、64KB等。它们在读取方面与普通的SRAM存储器类似，一般可以实现完全随机的读取，最大的不同在于写操作方面。Flash存储器的写操作需要经过“擦除——写入”两个操作过程。如果希望对某一个单元进行写入时，首先必须对这个存储单元所在的扇区(Sector)或块(Block)执行擦除操作，执行擦除操作时一次性将一个擦除块的位全部置为“1”，擦除操作成功完成后，整个扇区或块的数据内容都被清空，将被擦除的块称为脏块，擦除操作过程称为回收。然后对目的单元所在的扇区或块执行写入操作，执行写操作时可以将单个位的“1”写为“0”，但不可以将“0”写成“1”。所以通常的写闪存动作要先擦除整个擦除块原来的内容，再写入新的数据。

一般flash都提供了扇区擦除(Sector-Erase)、块擦除(Block-Erase)和芯片擦除(Chip-Erase)三种擦除方式。Chip-Erase是把整个Flash擦除，内容设置为1；Block-Erase和Sector-Erase分别以块和扇区为单位删除flash。flash的擦除次数是有限的，通常上限为每个块可擦除100,000~1,000,000次。当一个块或扇区提前达到擦除次数的上限时将导致整个flash无法使用。为了避免任意一个块在其它块之前达到这个极限，文件系统或FMM(Flash Media manager，即闪存存储管理，其作用主要是针对Flash自身的物理特性，利用一些特定的算法来提高Flash的使用效率，加快操作速度和管理Flash各单元的使用频率)必须保证擦除次数在各擦除块之间尽可能均匀分布，这一过程称为“均衡磨损(Wear Leveling)”。

现在，人们都采取搬块的方法，即先将原块中的有效内容搬到另一个块，然后再将原块擦除。其中，在向新块搬迁和写入数据的时候，都要通过加标注信息，以做到掉电保护和恢复数据。但是，到底该选择哪一个块

进行回收，不同厂家有不同的策略，但都只偏重某一方面，要么尽可能减少块的擦除次数，要么做到均匀擦除，以使所有物理块的寿命均衡。这样导致闪存设备的整体利用效率不高。

【发明内容】

本发明的主要目的就是为了解决现有技术的问题，提供一种闪存装置的脏块回收方法，既能尽可能减少块的擦除次数，又能做到均匀擦除，以使所有物理块的寿命均衡。

为实现上述目的，本发明公开了一种闪存装置的脏块回收方法，包括写扇区时进行的强制回收步骤和删除扇区后进行的条件回收步骤。

本发明的进一步改进是：所述强制回收步骤包括以下步骤：

A1、闪存装置接收写操作指令；

B1、闪存装置中的闪存存储管理模块查找可写入的空闲扇区；

C1、如果没有足够的空闲扇区，则闪存存储管理模块查找出脏扇区最多的块进行回收。

所述条件回收步骤包括以下步骤：

A2、闪存装置接收删除指令，闪存存储管理模块对指定扇区进行删除；

B2、闪存存储管理模块随机产生一 0~99 内的随机数；

C2、将该随机数与预先设定的概率分界点进行比较，如果该随机数大于或等于概率分界点，则执行步骤 E2，如果该随机数小于概率分界点，则执行步骤 D2，其中概率分界点的取值范围为 1~100；

D2、将各块的脏扇区数与预先设定的脏扇区阈值进行比较，对于脏扇区数大于脏扇区阈值的块执行步骤 F2，其中脏扇区阈值为小于每个块所包含的扇区数的正整数；

E2、闪存存储管理模块查找出擦除次数最少的块并执行步骤 F2；

F2、对该块进行回收。

在步骤 A2 之后还包括将该被删除扇区加上“脏”标记的步骤。

优选的，所述概率分界点是 97。

在步骤 D2 或 E2 之后、步骤 F2 之前还包括返回该块的块号的步骤。

在步骤 F2 之后还包括将该块的擦除次数加 1 的步骤。

本发明既能够尽可能减少块的擦除次数，又能做到均匀擦除，以使所有物理块的寿命均衡，从而延长了闪存装置的使用寿命，理由如下：

首先，本方案采用两种回收模式确定待回收的块，强制回收下找脏扇区最多的块进行回收。条件回收下，又细分了两种情况，即在绝大部分概

率下对脏扇区数达到脏扇区阈值的块进行回收，这考虑了尽可能少的块擦除次数；而在很小机会下回收擦除次数最小的块，这考虑了均匀磨损。

其次，以上两种回收模式又在不同的情况下使用。强制回收是在写扇区过程中无空闲扇区可用情况下采用的，而在删除扇区后才采用条件回收进行脏块回收。

最后，在条件回收模式下的概率分界点和脏扇区阈值，是经过大量实验得到的，都能代表闪存装置正常使用情况，准确、可靠。

本发明的特征及优点将通过实施例结合附图进行详细说明。

【附图说明】

图 1 是闪存装置的组成示意图；

图 2 是闪存装置存储块和扇区的组织结构示意图；

图 3 是本发明的条件回收流程图；

图 4 是本发明的强制回收流程图。

【具体实施方式】

闪存装置包括 MCU（微控制器）、存储阵列、底层驱动模块、闪存存储管理模块和文件系统，如图 1 所示，文件系统接收外部（例如外部应用程序）的读、写、擦除请求，通过闪存存储管理模块、底层驱动模块到达存储阵列，MCU 根据底层驱动模块的设定输出信号对存储阵列执行读、写、擦除操作。闪存存储管理模块负责完成各块之间的擦写次数均衡和坏块管理等工作，它通过底层驱动模块对存储阵列执行各种操作。存储阵列包括多个块，每个块由多个扇区构成，有时也被称为“页”，每个扇区分为存储区和冗余区，如图 2 所示。每个块中有一个特定区域存储该块的擦除次数、各扇区的状态和各扇区的逻辑号等信息，例如各扇区是否被删除而加上“脏”标记、是否被标记为空闲扇区等。

为了尽可能减少块的擦除次数，又能做到均匀擦除，本实施例通过两种回收方式来对块进行回收，即强制回收和条件回收。强制回收用于写操作时，当没有足够的空闲扇区写入数据的时候，闪存存储管理模块控制底层驱动模块对存储阵列中脏扇区数最多的块执行擦除。条件回收用于在执行扇区删除后，闪存存储管理模块根据设定的条件控制底层驱动模块对存储阵列中符合条件的块执行擦除。

其中，条件回收的具体流程如图 3 所示：

在步骤 201，闪存存储管理模块接收外部的删除指定扇区的信号，对指

定的扇区进行删除，并对该扇区加上“脏”标记，表示该扇区内的数据无效，具备了回收的必要条件；然后执行步骤 202；

在步骤 202，闪存存储管理模块随机产生一正的数值，将该数值用 100 去除，取其余数，则该余数是位于 0~99 内的一个随机数，然后执行步骤 203；

在步骤 203，闪存存储管理模块判断该随机数是否小于预先设定的概率分界点，概率分界点一般为大于 50 的值，如果该随机数小于概率分界点，则执行步骤 204；如果该随机数大于或等于概率分界点，则执行步骤 205；

在步骤 205，闪存存储管理模块读取各块中的擦除次数，将所有块的擦除次数进行比较，查找出擦除次数最少的块，将这个块作为待回收的脏块，然后执行步骤 207；

在步骤 204，闪存存储管理模块读取各块中的扇区是否有“脏”标记，统计各块中带有“脏”标记的扇区数量，然后执行步骤 206；

在步骤 206，判断每个块中的脏扇区数量是否大于或等于脏扇区阈值，如果脏扇区数量大于或等于脏扇区阈值，则将这个或这些块作为待回收的脏块，然后执行步骤 207；如果脏扇区数量小于脏扇区阈值，则执行步骤 210；

在步骤 207，返回待回收脏块的块号，然后执行步骤 208；

在步骤 208，闪存存储管理模块将与返回的块号相对应的块的位全部置为“1”，对该块执行回收，然后执行步骤 209；

在步骤 209，闪存存储管理模块将执行回收的块的擦除次数加 1，并保存在该块的特定区域，然后执行步骤 210；

在步骤 210，结束条件回收操作。

上述步骤中的概率分界点和脏扇区阈值都是实验结论值，再结合数理统计分析方法和 Flash 实际运行效果确定，可通过以下方法取得：两个变化参数中首先暂定一个，比如先将概率分界点设为 97，让文件系统在不同的脏扇区阈值下（假定 1Block = 128 Sectors，那么脏扇区阈值可取 127、126、125、124 等等）运行一段时间，打印每个块的擦除次数，并统计最大擦除次数、最小擦除次数、平均擦除次数。如此反复，在平均擦除次数较小，并且最大擦除次数、最小擦除次数与平均擦除次数比较接近的情况下，确定脏扇区阈值。然后，改变概率分界点（可取 99、98、97、96 等等），重复上面的操作。最终得到一个实验最佳值。

上面得到的实验最佳值还可用数学方法进行分析验证。用上面的大量

测量值构造一个二元随机函数，以概率分界点和脏扇区阈值为两变量，以平均擦除次数为函数值。利用数据分析软件，如 SPSS 或 SAS，得到理论最佳值，使得平均擦除次数最小。然后，比较理论最佳值与实验最佳值，看两者的偏差大小。

概率分界点和脏扇区阈值确定后可输入到闪存存储管理模块中。

下面举例对概率分界点和脏扇区阈值的应用进行说明：经过实验，先设定概率分界点为 97。1 个块 (block) 是 64KB，1 个扇区 (sector) 按 512 字节计算的话，该块一共有 128 个扇区。在此，设定脏扇区阈值为 120，也就是说，只要一个 block 中脏扇区数等于或大于 120，则说明该块达到了条件回收的条件。随机生成一个数，用 100 去模，则得到 0~99 内的数。若模后的数小于 97，也即在 97% 的概率下，寻找块内脏扇区数等于或大于 120 的块，找到后返回块号进行回收，若没找到，不做处理。反之，若模后的数大于或等于 97，也即在 3% 的概率下，寻找擦除次数最小的块，找到后返回块号进行回收。回收一次该块，将该块的擦除次数加 1，此值保存在块的特定位置。

强制回收的具体流程如图 4 所示：

在步骤 301，闪存存储管理模块接收外部的写操作指令，然后执行步骤 302；

在步骤 302，闪存存储管理模块读取各个扇区的状态，查找出空闲的扇区，然后执行步骤 303；

在步骤 303，闪存存储管理模块判断是否有足够的空闲扇区供写入，如果有则执行步骤 308，如果找遍整个存储阵列都没有足够的空闲扇区供写入，则执行步骤 304；

在步骤 304，闪存存储管理模块闪存存储管理模块读取各块中的扇区是否有“脏”标记，统计各块中带有“脏”标记的扇区数量，查找出脏扇区最多的块，将该块作为待回收的脏块，然后执行步骤 305；

在步骤 305，返回待回收脏块的块号，然后执行步骤 306；

在步骤 306，闪存存储管理模块将与返回的块号相对应的块的位全部置为“1”，对这个块执行回收，然后执行步骤 307；

在步骤 307，闪存存储管理模块将执行回收的块的擦除次数加 1，并保存在该块的特定区域，然后执行步骤 308；

在步骤 308，结束条件回收操作。

在对块进行回收后，在回收的可用块内写入数据，同时进行逻辑地址到物理地址对应表的修改。在回收脏块和写数据的过程中，为防止意外掉电，都通过加标记的方法，进行掉电后的数据恢复。

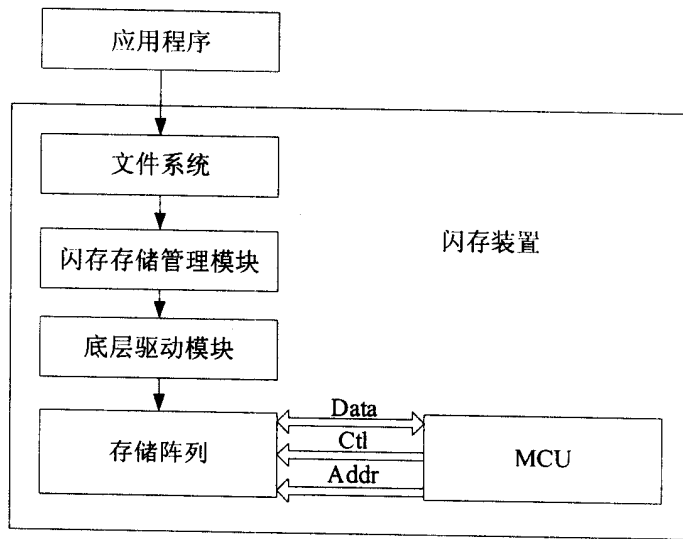


图 1

第 0 块	第0扇区	数据存储区	冗余区
	第1扇区		
	⋮		
	最后扇区		
第 1 块	第0扇区	数据存储区	冗余区
	第1扇区		
	⋮		
	最后扇区		
⋮	⋮	⋮	⋮
最后 一块	第0扇区	数据存储区	冗余区
	第1扇区		
	⋮		
	最后扇区		

图 2

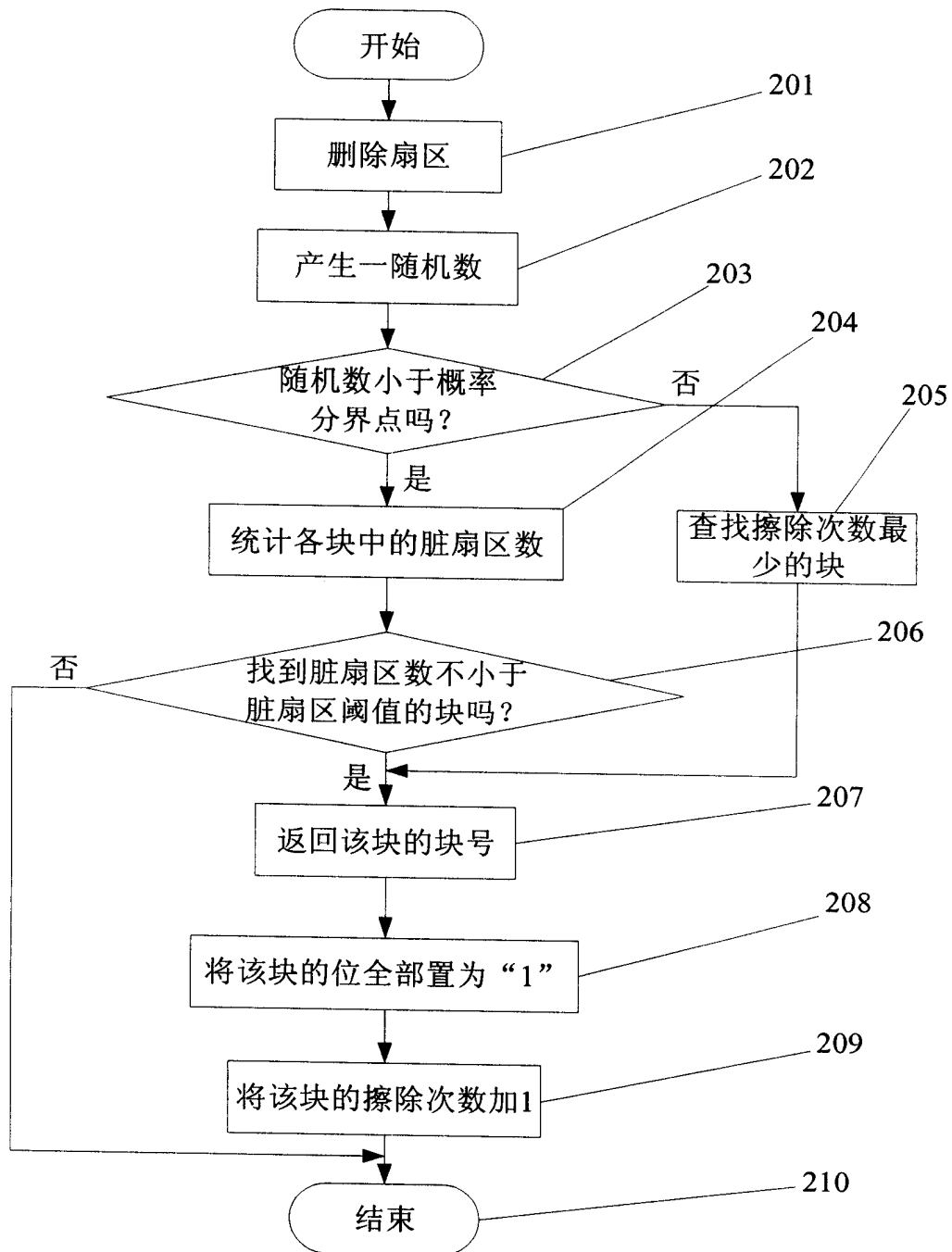


图3

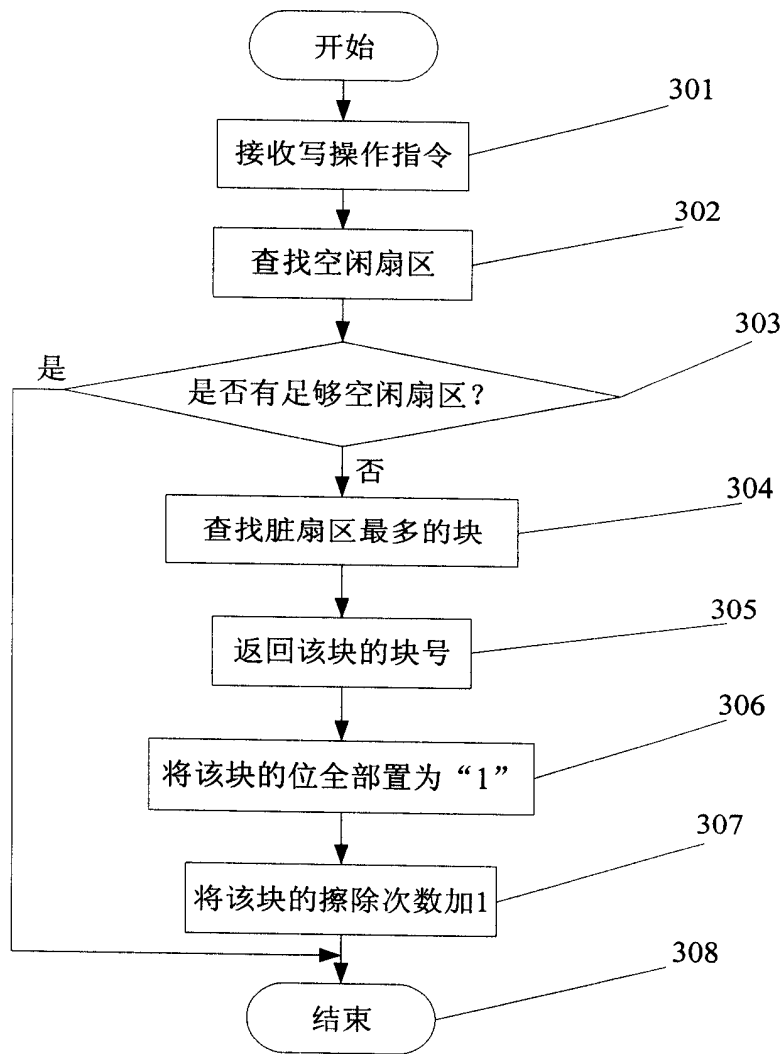


图 4