

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/30 (2006.01)

G06F 12/00 (2006.01)

G06F 15/16 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510102690.2

[43] 公开日 2006 年 2 月 8 日

[11] 公开号 CN 1731402A

[22] 申请日 2005.9.13

[21] 申请号 200510102690.2

[71] 申请人 北京中星微电子有限公司

地址 100083 北京市海淀区学院路 35 号世宁大厦 15 层

[72] 发明人 王 军 李晓强 谭 帅 孙丰强
赵 原

[74] 专利代理机构 北京安信方达知识产权代理有限公司

代理人 龙 洪 霍育栋

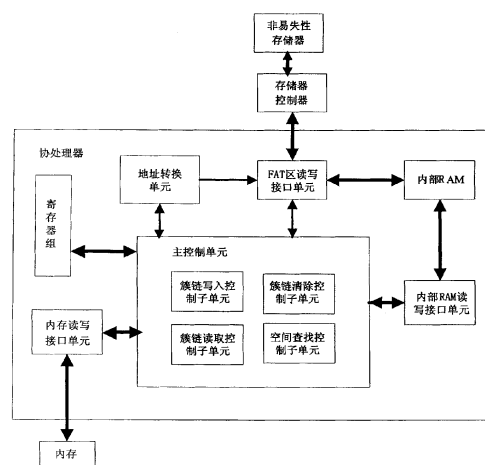
权利要求书 8 页 说明书 22 页 附图 6 页

[54] 发明名称

用协处理器加速文件系统操作的方法及装置

[57] 摘要

本发明公开一种用协处理器加速文件系统操作的方法及其装置，包括主处理器和具有文件分配表区的非易失性存储器，还包括一个通过线路与所述存储器控制器和主处理器连接的协处理器。所述协处理器收到对文件分配表进行操作的指令和参数后，执行对所述文件分配表区的操作，然后给出操作执行情况的指示信号，有操作结果数据的，存入到指定的存储位置。本发明用协处理器承担了对主处理器来说负担较重的对 FAT 表的操作，大大提高文件系统的操作速度，同时减少了占用处理器用于文件系统操作的时间及资源。



1、一种用协处理器加速文件系统操作的装置，包括主处理器和具有文件分配表区的非易失性存储器，其特征在于，还包括一个协处理器，该协处理器可与所述主处理器和非易失性存储器通信，用于根据指令和参数执行对文件分配表区的操作，给出操作执行情况的指示信号，有操作结果数据的，存入到指定的存储位置。

2、如权利要求1所述的装置，其特征在于，所述协处理器包括主控制单元、数据存储单元以及文件分配表区接口单元，其中：

所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据；

所述文件分配表区接口单元用于在主控制单元的控制下，读出所述非易失性存储器中给定扇区的数据，并暂存到所述数据存储单元；

所述主控制单元包括簇链读取控制单元，用于根据读取簇链的指令和给出的起始簇号，从所述文件分配表区中读出该起始簇号之后相链簇链的各个簇号，存放到指定的存储位置，然后发出执行完毕的信号。

3、如权利要求1所述的装置，其特征在于，所述协处理器包括主控制单元、数据存储单元以及文件分配表区接口单元，其中：

所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据，以及更新后待写入文件分配表区的数据；

所述文件分配表区接口单元用于在主控制单元的控制下，读出所述非易失性存储器中给定扇区的数据，并暂存到所述数据存储单元，并将更新后的数据写入到原扇区；

所述主控制单元包括：

空间查找控制单元，用于根据查找空间的指令和给定的空间需求，查找文件分配表区，如找到足够的剩余空间，将这些空间对应的簇号存入所述指定的存储位置，否则返回空间不足的结果；

簇链写入控制单元：用于根据写入簇链指令和给出的文件数据的存储位

置,将存有该文件数据的每个簇的簇号写入其上一簇簇号在文件分配表区中的对应区间,建立该文件的簇链,然后发出执行完毕的信号。

4、如权利要求2或3所述的装置,其特征在于,所述主控制单元还包括簇链清除控制单元,用于根据清除簇链的指令和给定的簇号,将FAT表中该簇号之后相链的簇链清除,然后发出执行完毕的信号。

5、如权利要求2或3所述的装置,其特征在于,还包括一个内存,所述指定的存储位置是指该内存中的给定位置,所述主控制单元还包括通过总线与该内存相连或者直接与该内存相连的内存接口单元,用于在主控制单元的控制下,完成对该内存的数据操作。

10 6、如权利要求2或3所述的装置,其特征在于,所述数据存储单元包括用于存储指令参数和返回参数的寄存器组,以及用于存储从文件分配表区读出的数据的内部RAM,且所述协处理器还包括内部RAM接口单元,用于在主控制单元的控制下,完成对内部RAM数据的操作。

15 7、如权利要求2或3所述的装置,其特征在于,所述主控制单元中还包括一个与所述主控制单元连接的地址转换单元,该地址转换单元可以具有以下功能中的一种或任意组合:

I,用于根据指令参数和/或操作过程中给出的簇号,计算出文件分配表中该簇号对应区间所在的逻辑扇区号以及在该扇区中的地址偏移量;

20 II,用于将获取的簇号和连续簇号数转换成扇区号和连续扇区数的形式存储到所述指定位置;

III,用于将指定位置存储的扇区号和连续扇区数转换为簇号和连续簇号数,供主控制单元使用。

8、如权利要求2或3所述的装置,其特征在于,所述主控制单元中还包括一个数据压缩子单元,用于将查找到的多个连续簇号采用起始簇号加上连续簇号长度的压缩格式表示。

25 9、如权利要求1所述的装置,其特征在于,还包括与所述非易失性存储器相连的存储器控制器,所述协处理器与主处理器均通过该存储器控制器访问所述非易失性存储器。

10、一种用协处理器加速文件系统操作的方法，应用于包括主处理器、协处理器及具有文件分配表区的非易失性存储器的装置，该方法包括以下步骤：

(a) 所述协处理器收到对文件分配表进行操作的指令和参数；

5 (b) 所述协处理器根据所述指令和参数，执行对所述文件分配表区的操作；

(c) 所述协处理器给出操作执行情况的指示信号，有操作结果数据的，存入到指定的存储位置。

11、如权利要求 10 所述的方法，其特征在于，所述步骤 (a) 中，协处理器收到的是读取簇链的指令和给出的起始簇号，所述步骤 (b) 进一步由以下步骤实现：

(b11) 所述协处理器以起始簇号为当前簇号，计算出其在文件分配表区对应区间所在的扇区号，然后读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区；

15 (b12) 所述协处理器由当前簇号得出其对应区间的数据在暂存区的位置，将该位置存储的数据读出，如果该数据是文件结束标志，执行步骤 (c)，否则该数据为相链簇链上的下一簇号，记录该簇号信息后将其作为新的当前簇号，执行步骤 (b13)；

(b13) 所述协处理器判断当前簇号对应区间的数据是否已读入暂存区，
20 如果没有读入，将包含其对应区间的扇区的数据读入暂存区，返回步骤 (b12)，如果已读入，直接返回步骤 (b12)；

所述步骤 (c) 中，所述协处理器将记录的簇号信息存入到指定的存储位置，并给出操作完成的指示信号。

12、如权利要求 11 所述的方法，其特征在于，所述步骤 (b12) 和 (b13) 在记录找到的簇号时，还统计找到的簇号的总数目，当该总数目达到设定的上限时，返回执行失败的指示信息，结束此次操作。

13、如权利要求 10 所述的方法，其特征在于，所述步骤 (a) 中，协处理器收到的是清除簇链的指令和给出的起始簇号，所述步骤 (b) 进一步由

以下步骤实现:

(b21) 所述协处理器以起始簇号为当前簇号, 计算出其在文件分配表区对应区间所在的扇区号, 然后读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区;

- 5 (b22) 所述协处理器由当前簇号得出其对应区间的数据在暂存区的位置, 将该位置存储的数据读出, 然后将暂存区中该位置的数据修改为空白簇标识, 如果读出数据是文件结束标志, 执行步骤(c), 否则该数据为相链簇链上的下一簇号, 将其作为新的当前簇号, 执行步骤(b23);

- 10 (b23) 所述协处理器判断当前簇号对应区间的数据是否已读入暂存区, 如果没有读入, 将更新后的暂存数据写回文件分配表区的原扇区, 再将包含其对应区间的一个或多个扇区的数据读出并存入暂存区, 返回步骤(b22), 如果已读入, 直接返回步骤(b22);

所述步骤(c)中, 所述协处理器将更新后的暂存数据写回文件分配表区的原扇区, 并给出操作完成的指示信号。

- 15 14、如权利要求10所述的方法, 其特征在于, 所述步骤(a)中, 协处理器收到的是查找剩余空间的指令和文件需要的簇数, 所述步骤(b)进一步由以下步骤实现:

- 20 (b31) 所述协处理器确定一个开始查找的扇区号, 读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区, 再确定一个当前偏移量, 从该偏移量在暂存区中指向的区间开始查找;

(b32) 所述协处理器判断当前偏移量指向区间存储的数据是否空闲簇的标识, 如果不是, 执行步骤(b34), 否则, 记录该区间对应的簇号信息, 将空闲簇的计数值加1, 执行步骤(b33);

- 25 (b33) 所述协处理器判断空闲簇的计数值是否已等于文件需要的簇数, 如果是, 执行步骤(c), 否则, 执行步骤(b34);

(b34) 将当前偏移量增加一个区间长度, 如未超过暂存区范围, 返回步骤(b32), 如已超过, 再判断文件分配表区所有扇区是否均已读出过, 如果是, 执行步骤(c), 否则, 将相邻的下一扇区或多个扇区的数据读入

暂存区，将当前偏移量置为 0，返回步骤 (b32)；

所述步骤 (c) 中，所述协处理器将记录的空闲簇号信息存入到指定的存储位置，并给出操作完成的指示信号。

15 15、如权利要求 14 所述的方法，其特征在于，所述步骤 (a) 协处理器收到的指令参数还包括要求的最小连续簇号数目，在找到空闲簇时还累计连续空闲簇的个数，只有查找到的连续空闲簇的个数大于等于该要求的数目时，才记录这些空闲簇的信息和计入总的空闲簇的数目。

10 16、权利要求 10 所述的方法，其特征在于，所述步骤 (a) 中，协处理器收到的是更新文件分配表区的指令和包括文件内容所在存储位置信息的一条或多条数据的存放位置，所述步骤 (b) 进一步由以下步骤实现：

15 (b41) 所述协处理器依序从所述存放位置逐一读出各条数据，获取对应一段文件内容所在簇的一组簇号信息，如果读出的是第一条数据，以该组的起始簇号为当前簇号，执行步骤 (b42)，否则，将该组的起始簇号写入当前簇号对应区间的数据在暂存区的位置，并将该起始簇号作为新的当前簇号，执行步骤 (b42)；

(b42) 所述协处理器判断当前簇号对应区间的数据是否已读入暂存区，如果未读入，将更新后的暂存数据写回文件分配表区的原扇区，再将包含其对应区间的一个或多个扇区的数据读出并存入暂存区，执行步骤 (b43)；如果已读入，直接执行步骤 (b43)；

20 (b43) 所述协处理器算出当前簇号对应区间的数据在暂存区的位置，将相对该当前簇号的下一簇号写入该位置并将其作为新的当前簇号，然后判断该组簇号是否均已写入，如果是，执行步骤 (b44)，否则，返回步骤 (b42)；

25 (b44) 所述协处理器判断所述存放位置的数据是否已全部读出，如果是，将文件结束标志写入当前簇号对应区间的数据在暂存区的位置，并将更新后的暂存数据写回文件分配表区的原扇区，执行步骤 (c)，否则，保证当前簇号对应区间所在扇区的数据已读入暂存区，返回步骤 (b41)；

所述步骤 (c) 中，所述协处理器给出操作完成的指示信号。

17、如权利要求 16 所述的方法，其特征在于，所述一条或多条数据中

包括的文件内容所在存储位置信息包括起始簇号和连续簇号数目两个字段；或者包括起始扇区号和连续扇区数目两个字段，再由所述协处理器将其转换为起始簇号和连续簇号数目。

18、如权利要求 11 或 14 所述的方法，其特征在于，所述协处理器记录
5 找到的簇号时，是将连续簇号作为一组，用包括起始簇号和连续簇号数目两个字段的一条数据来表示该组簇号，作为操作结果的数据，或者，再将起始簇号和连续簇号数目转换为起始扇区号和连续扇区数目，再作为操作结果的数据。

19、如权利要求 11 或 14 所述的方法，其特征在于，所述装置还包含一个
10 与所述协处理器连接的内存，所述步骤（a）协处理器收到的指令参数中包括在该内存分配的缓冲区的起始地址和长度，作为所述指定的存储位置；或者所述指定的存储位置是指所述协处理上的寄存器组。

20、如权利要求 18 所述的方法，其特征在于，所述步骤（b）协处理器
15 每得到一条操作结果数据，先将其存入所述指定位置，并判断该位置是否已存满，如果缓冲区未滿，则继续其后续操作，如果已滿，结束此次操作并返回未完成标志、此次操作得到的最后一个簇号或最后操作的暂存区位置对应的簇号，由主处理器或其它模块重新分配缓存空间，向协处理器发起下一次操作的指令，并以该最后一个簇号为起始簇号。

21、如权利要求 10 所述的方法，其特征在于，所述文件分配表区采用
20 FAT12、FAT16 或者 FAT32 的格式。

22、如权利要求 11、13、14 或 16 所述的方法，其特征在于，所述根据簇号计算其在文件分配表对应区间所在的扇区号 LBA 的公式为：

$$LBA = FAT_{start} + \left[\frac{Index \times Len_{unit}}{Size_{sector}} \right]$$

其中，“ $[]$ ”表示向下取整运算， Len_{unit} 为每个簇号对应区间的大小，单
25 位为字节， $Size_{sector}$ 为每个扇区的字节数， FAT_{start} 为 FAT 表区的起始扇区号， $Index$ 为簇号值。

23、如权利要求 11、13 或 16 所述的方法，其特征在于，所述协处理器

由当前簇号计算其对应区间的数据在暂存区的位置时, 采用以下公式:

$$W = \frac{Index \times Len_{unit}}{Size_{sector}} \text{ 的余数加上 } (N-1) \times Size_{sector}$$

- 其中: W 表示该位置在暂存区的偏移量, Len_{unit} 为每个簇号对应区间的大小, 单位为字节, $Size_{sector}$ 为每个扇区的字节数, $Index$ 为簇号值, N 表示该簇号对应区间所在扇区是此次读入暂存区的第几个扇区。

24、一种可加速文件分配表操作的协处理器, 其特征在于, 包括主控制单元、数据存储单元以及文件分配表区接口单元, 其中:

所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据;

- 10 所述文件分配表区接口单元用于在主控制单元的控制下, 读出所述非易失性存储器中给定扇区的数据, 并暂存到所述数据存储单元;

所述主控制单元用于根据指令和参数执行对文件分配表区的操作, 给出操作执行情况的指示信号, 有操作结果数据的, 存入到指定的存储位置。

- 15 25、如权利要求 24 所述的协处理器, 其特征在于, 所述主控制单元包括: 簇链读取控制单元, 用于根据读取簇链的指令和给出的起始簇号, 从所述文件分配表区中读出该起始簇号之后相链簇链的各个簇号, 存放到指定的存储位置, 然后发出执行完毕的信号。

- 20 26、如权利要求 24 所述的协处理器, 其特征在于, 所述主控制单元包括: 空间查找控制单元, 用于根据查找空间的指令和给定的空间需求, 查找文件分配表区, 如找到足够的剩余空间, 将这些空间对应的簇号存入所述指定的存储位置, 否则返回空间不足的结果; 以及簇链写入控制单元: 用于根据写入簇链指令和给出的文件数据的存储位置, 将存有该文件数据的每个簇的簇号写入其上一簇簇号在文件分配表区中的对应区间, 建立该文件的簇链, 然后发出执行完毕的信号; 同时, 所述数据存储单元还用于存储更新后待写入文件分配表区的数据; 所述文件分配表区接口单元还用于在主控制单元的控制下, 将更新后的数据写入到原扇区。

- 25 27、如权利要求 25 或 26 所述的协处理器, 其特征在于, 所述主控制单

元还包括簇链清除控制单元,用于根据清除簇链的指令和给定的簇号,将FAT表中该簇号之后相链的簇链清除,然后发出执行完毕的信号。

28、如权利要求 24 所述的协处理器,其特征在于,所述数据存储单元包括用于存储指令参数和返回参数的寄存器组,以及用于存储从文件分配表
5 区读出的数据的内部 RAM,且所述协处理器还包括内部 RAM 接口单元,用于在主控制单元的控制下,完成对内部 RAM 数据的操作。

29、如权利要求 28 所述的协处理器,其特征在于,所述内部 RAM 的容量为一个或多个扇区的大小。

30、如权利要求 25 或 26 所述的协处理器,其特征在于,所述主控制单
10 元中还包括一个与所述主控制单元连接的地址转换单元,该地址转换单元可以具有以下功能中的一种或任意组合:

I,用于根据指令参数和/或操作过程中给出的簇号,计算出文件分配表中该簇号对应区间所在的逻辑扇区号以及在该扇区中的地址偏移量;

II,用于将获取的簇号和连续簇号数转换成扇区号和连续扇区数的形式
15 存储到所述指定位置;

III,用于将指定位置存储的扇区号和连续扇区数转换为簇号和连续簇号数,供主控制单元使用。

31、如权利要求 25 或 26 所述的协处理器,其特征在于,所述主控制单元中还包括一个数据压缩子单元,用于将查找到的多个连续簇号采用起始簇
20 号加上连续簇号长度的压缩格式表示。

32、如权利要求 25 或 26 所述的协处理器,其特征在于,所述主控制单元还包括与位于该协处理器外部的内存相连的内存接口单元,用于在主控制单元的控制下,完成对该内存的数据操作。

用协处理器加速文件系统操作的方法及装置

5 技术领域

本发明涉及计算机的文件系统,尤其涉及一种提高文件系统操作速度的方法及其装置。

背景技术

- 10 文件系统是为了存储和管理数据而在存储器上建立的一些结构的总和。文件系统主要完成三项功能:跟踪记录存储器中被耗用的空间和自由空间、维护目录名和文件名以及跟踪记录每一个文件的物理存储位置。

如图1所示,非易失性存储器上的文件系统一般具有BIOS参数块、文件分配表(FAT表)、根目录列表和数据区。其中:

- 15 BIOS参数块记录了每扇区字节数、每簇扇区数、每个FAT的扇区数、扇区总数、根目录项数等等参数。

FAT表是记录每个逻辑文件物理存储位置的链表,一般以簇为单位,一个簇可以根据需要包含两个、四个甚至更多的扇区。

- 20 根目录列表记录着每个文件(或目录)的文件名、文件物理和/或逻辑的起始位置,文件大小、文件创建的日期等描述逻辑文件数据结构的数据,也称文件项。

数据区用于存放文件内容的数据。

- 25 如图2所示,传统的实现文件系统操作的装置包括通过总线连接的处理
器、存储器控制器、易失性存储器、非易失性存储器及其它装置(如显示设备、网络设备、音频设备),处理器对非易失性存储器的读写操作通过存储器控制器执行。

以打开某一个存在于文件系统根目录下的文件为例,该过程首先由用户

发起,然后由处理器通过存储器控制器与非易失性存储器交互,完成以下几步操作:

第一步,枚举目录/文件项,读取根目录列表以取得该文件的文件项内容和位于文件系统数据区的首簇号;

- 5 第二步,搜索文件分配表,根据该文件的首簇号从FAT表查找该文件对应的簇链,并将得到的簇号转换成文件各项内容存储的物理地址(如,逻辑扇区号);

第三步,存储器控制器根据获得的所述物理地址,读出位于非易失性存储器数据区内的文件数据,并将其传输到目标设备。

- 10 同一台计算机各软件间的文件数据传输在处理器、非易失性存储器和易失性存储器之间来回传递。而文件数据在设备间传输时,处理器需要通过内存中转操作,同其它硬件设备进行数据传递。

- 在某些应用场合下,需要在芯片中内置处理器(CPU),如果使用性能强大的处理器,将会导致芯片成本过高。当出于成本考虑使用低性能的处理
15 器时,由于有大量的数据搬移工作要由处理器完成,将会导致运行速度太慢,影响系统的性能。

- 处理器在数据读写的时候,会持续等待数据的读、写等文件系统操作,这个时间相当漫长。一些统计数据表明,文件系统的操作占用了处理器很多的资源。因此,在使用低性能的处理器的同时要想达到好的性能,就需要想
20 办法加快文件系统操作的速度,同时减轻处理器的处理压力,以便留下更多的资源在执行其它任务时使用。

发明内容

- 本发明要解决的技术问题是提供一种用协处理器加速文件系统操作的方法。本发明还要提供一种可以实现该方法的装置及其协处理器。
25

为了解决上述技术问题,本发明提供了一种用协处理器加速文件系统操作的装置,包括主处理器和具有文件分配表区的非易失性存储器,其特征在于,还包括一个协处理器,该协处理器可与所述主处理器和非易失性存储器

通信,用于根据指令和参数执行对文件分配表区的操作,给出操作执行情况的指示信号,有操作结果数据的,存入到指定的存储位置。

进一步地,上述装置还可具有以下特点:所述协处理器包括主控制单元、数据存储单元以及文件分配表区接口单元,其中:

- 5 所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据;

所述文件分配表区接口单元用于在主控制单元的控制下,读出所述非易失性存储器中给定扇区的数据,并暂存到所述数据存储单元;

- 10 所述主控制单元包括簇链读取控制单元,用于根据读取簇链的指令和给出的起始簇号,从所述文件分配表区中读出该起始簇号之后相链簇链的各个簇号,存放到指定的存储位置,然后发出执行完毕的信号。

进一步地,上述装置还可具有以下特点:所述协处理器包括主控制单元、数据存储单元以及文件分配表区接口单元,其中:

- 15 所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据,以及更新后待写入文件分配表区的数据;

所述文件分配表区接口单元用于在主控制单元的控制下,读出所述非易失性存储器中给定扇区的数据,并暂存到所述数据存储单元,并将更新后的数据写入到原扇区;

- 20 所述主控制单元包括:

空间查找控制单元,用于根据查找空间的指令和给定的空间需求,查找文件分配表区,如找到足够的剩余空间,将这些空间对应的簇号存入所述指定的存储位置,否则返回空间不足的结果;

- 25 簇链写入控制单元:用于根据写入簇链指令和给出的文件数据的存储位置,将存有该文件数据的每个簇的簇号写入其上一簇簇号在文件分配表区中的对应区间,建立该文件的簇链,然后发出执行完毕的信号。

进一步地,上述装置还可具有以下特点:所述主控制单元还包括簇链清除控制单元,用于根据清除簇链的指令和给定的簇号,将FAT表中该簇号

之后相链的簇链清除，然后发出执行完毕的信号。

进一步地，上述装置还可具有以下特点：还包括一个内存，所述指定的存储位置是指该内存中的给定位置，所述主控制单元还包括通过总线与该内存相连或者直接与该内存相连的内存接口单元，用于在主控制单元的控制下，完成对该内存的数据操作。

进一步地，上述装置还可具有以下特点：所述数据存储单元包括用于存储指令参数和返回参数的寄存器组，以及用于存储从文件分配表区读出的数据的内部 RAM，且所述协处理器还包括内部 RAM 接口单元，用于在主控制单元的控制下，完成对内部 RAM 数据的操作。

10 进一步地，上述装置还可具有以下特点：所述内部 RAM 的容量为一个或多个扇区的大小。

进一步地，上述装置还可具有以下特点：所述主控制单元中还包括一个与所述主控制单元连接的地址转换单元，用于根据指令参数和/或操作过程中给出的簇号，计算出文件分配表中该簇号对应区间所在的逻辑扇区号以及在该扇区中的地址偏移量。

进一步地，上述装置还可具有以下特点：所述地址转换单元还用于将获取的簇号和连续簇号数转换成扇区号和连续扇区数的形式存储到所述指定位置，及/或将指定位置存储的扇区号和连续扇区数转换为簇号和连续簇号数，供主控制单元使用。

20 进一步地，上述装置还可具有以下特点：所述主控制单元中还包括一个数据压缩子单元，用于将查找到的多个连续簇号采用起始簇号加上连续簇号长度的压缩格式表示。

进一步地，上述装置还可具有以下特点：还包括与所述非易失性存储器相连的存储器控制器，所述协处理器与主处理器均通过该存储器控制器访问所述非易失性存储器。

25 本发明还提供了一种用协处理器加速文件系统操作的方法，应用于包括主处理器、协处理器及具有文件分配表区的非易失性存储器的装置，该方法

包括以下步骤:

(a) 所述协处理器收到对文件分配表进行操作的指令和参数;

(b) 所述协处理器根据所述指令和参数, 执行对所述文件分配表区的操作;

5 (c) 所述协处理器给出操作执行情况的指示信号, 有操作结果数据的, 存入到指定的存储位置。

进一步地, 上述方法还可具有以下特点: 所述步骤(a)中, 协处理器收到的是读取簇链的指令和给出的起始簇号, 所述步骤(b)进一步由以下步骤实现:

10 (b11) 所述协处理器以起始簇号为当前簇号, 计算出其在文件分配表区对应区间所在的扇区号, 然后读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区;

(b12) 所述协处理器由当前簇号得出其对应区间的数据在暂存区的位置, 将该位置存储的数据读出, 如果该数据是文件结束标志, 执行步骤(c),
15 否则该数据为相链簇链上的下一簇号, 记录该簇号信息后将其作为新的当前簇号, 执行步骤(b13);

(b13) 所述协处理器判断当前簇号对应区间的数据是否已读入暂存区, 如果没有读入, 将包含其对应区间的扇区的数据读入暂存区, 返回步骤(b12), 如果已读入, 直接返回步骤(b12);

20 所述步骤(c)中, 所述协处理器将记录的簇号信息存入到指定的存储位置, 并给出操作完成的指示信号。

进一步地, 上述方法还可具有以下特点: 所述步骤(b12)和(b13)在记录找到的簇号时, 还统计找到的簇号的总数目, 当该总数目达到设定的上
25 限时, 返回执行失败的指示信息, 结束此次操作。

进一步地, 上述方法还可具有以下特点: 所述步骤(a)中, 协处理器收到的是清除簇链的指令和给出的起始簇号, 所述步骤(b)进一步由以下步骤实现:

(b21) 所述协处理器以起始簇号为当前簇号, 计算出其在文件分配表区对应区间所在的扇区号, 然后读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区;

5 (b22) 所述协处理器由当前簇号得出其对应区间的数据在暂存区的位置, 将该位置存储的数据读出, 然后将暂存区中该位置的数据置为空白簇标识, 如果读出数据是文件结束标志, 执行步骤(c), 否则该数据为相链簇链上的下一簇号, 将其作为新的当前簇号, 执行步骤(b23);

10 (b23) 所述协处理器判断当前簇号对应区间的数据是否已读入暂存区, 如果没有读入, 将更新后的暂存数据写回文件分配表区的原扇区, 再将包含其对应区间的一个或多个扇区的数据读出并存入暂存区, 返回步骤(b22), 如果已读入, 直接返回步骤(b22);

所述步骤(c)中, 所述协处理器将更新后的暂存数据写回文件分配表区的原扇区, 并给出操作完成的指示信号。

15 进一步地, 上述方法还可具有以下特点: 所述步骤(a)中, 协处理器收到的是查找剩余空间的指令和文件需要的簇数, 所述步骤(b)进一步由以下步骤实现:

20 (b31) 所述协处理器确定一个开始查找的扇区号, 读出包含该扇区在内的一个或多个扇区的文件分配表数据并存入暂存区, 再确定一个当前偏移量, 从该偏移量在暂存区中指向的区间开始查找;

(b32) 所述协处理器判断当前偏移量指向区间存储的数据是否空闲簇的标识, 如果不是, 执行步骤(b34), 否则, 记录该区间对应的簇号信息, 将空闲簇的计数值加1, 执行步骤(b33);

25 (b33) 所述协处理器判断空闲簇的计数值是否已等于文件需要的簇数, 如果是, 执行步骤(c), 否则, 执行步骤(b34);

(b34) 将当前偏移量增加一个区间长度, 如未超过暂存区范围, 返回步骤(b32), 如已超过, 再判断文件分配表区所有扇区是否均已读出过, 如果是, 执行步骤(c), 否则, 将相邻的下一扇区或多个扇区的数据读入

暂存区，将当前偏移量置为 0，返回步骤（b32）；

所述步骤（c）中，所述协处理器将记录的空闲簇号信息存入到指定的存储位置，并给出操作完成的指示信号。

进一步地，上述方法还可具有以下特点：所述步骤（a）协处理器收到
5 的指令参数还包括要求的最小连续簇号数目，在找到空闲簇时还累计连续空闲簇的个数，只有查找到的连续空闲簇的个数大于等于该要求的数目时，才记录这些空闲簇的信息和计入总的空闲簇的数目。

进一步地，上述方法还可具有以下特点：所述步骤（a）中，协处理器收到的是更新文件分配表区的指令和包括文件内容所在存储位置信息的一
10 条或多条数据的存放位置，所述步骤（b）进一步由以下步骤实现：

（b41）所述协处理器依序从所述存放位置逐一读出各条数据，获取对应一段文件内容所在簇的一组簇号信息，如果读出的是第一条数据，以该组的起始簇号为当前簇号，执行步骤（b42），否则，将该组的起始簇号写入当前簇号对应区间的数据在暂存区的位置，并将该起始簇号作为新的当前簇
15 号，执行步骤（b42）；

（b42）所述协处理器判断当前簇号对应区间的数据是否已读入暂存区，如果未读入，将更新后的暂存数据写回文件分配表区的原扇区，再将包含其对应区间的一个或多个扇区的数据读出并存入暂存区，执行步骤（b43）；如果已读入，直接执行步骤（b43）；

20 （b43）所述协处理器算出当前簇号对应区间的数据在暂存区的位置，将相对该当前簇号的下一簇号写入该位置并将其作为新的当前簇号，然后判断该组簇号是否均已写入，如果是，执行步骤（b44），否则，返回步骤（b42）；

（b44）所述协处理器判断所述存放位置的数据是否已全部读出，如果是，将文件结束标志写入当前簇号对应区间的数据在暂存区的位置，并将更新后的暂存数据写回文件分配表区的原扇区，执行步骤（c），否则，保证
25 当前簇号对应区间所在扇区的数据已读入暂存区，返回步骤（b41）；

所述步骤（c）中，所述协处理器给出操作完成的指示信号。

进一步地,上述方法还可具有以下特点:所述一条或多条数据中包括的文件内容所在存储位置信息包括起始簇号和连续簇号数目两个字段;或者包括起始扇区号和连续扇区数目两个字段,再由所述协处理器将其转换为起始簇号和连续簇号数目。

- 5 进一步地,上述方法还可具有以下特点:所述协处理器记录找到的簇号时,是将连续簇号作为一组,用包括起始簇号和连续簇号数目两个字段的一条数据来表示该组簇号,作为操作结果的数据,或者,再将起始簇号和连续簇号数目转换为起始扇区号和连续扇区数目,再作为操作结果的数据。

- 10 进一步地,上述方法还可具有以下特点:所述装置还包含一个与所述协处理器连接的内存,所述步骤(a)协处理器收到的指令参数中包括在该内存分配的缓冲区的起始地址和长度,作为所述指定的存储位置;或者所述指定的存储位置是指所述协处理上的寄存器组。

- 15 进一步地,上述方法还可具有以下特点:所述步骤(b)协处理器每得到一条操作结果数据,先将其存入所述指定位置,并判断该位置是否已存满,如果缓冲区未滿,则继续其后续操作,如果已滿,结束此次操作并返回未完成标志、此次操作得到的最后一个簇号或最后操作的暂存区位置对应的簇号,由主处理器或其它模块重新分配缓存空间,向协处理器发起下一次操作的指令,并以该最后一个簇号为起始簇号。

- 20 进一步地,上述方法还可具有以下特点:所述文件分配表区采用 FAT12、FAT16 或者 FAT32 的格式。

进一步地,上述方法还可具有以下特点:所述根据簇号计算其在文件分配表对应区间所在的扇区号 LBA 的公式为:

$$LBA = FAT_{start} + \left\lfloor \frac{Index \times Len_{unit}}{Size_{sector}} \right\rfloor$$

- 25 其中,“ $\lfloor \rfloor$ ”表示向下取整运算, Len_{unit} 为每个簇号对应区间的大小,单位为字节, $Size_{sector}$ 为每个扇区的字节数, FAT_{start} 为 FAT 表区的起始扇区号, $Index$ 为簇号值。

进一步地,上述方法还可具有以下特点:所述协处理器由当前簇号计算

其对应区间的数据在暂存区的位置时, 采用以下公式:

$$W = \frac{Index \times Len_{unit}}{Size_{sector}} \text{ 的余数加上 } (N-1) \times Size_{sector}$$

- 其中: W 表示该位置在暂存区的偏移量, Len_{unit} 为每个簇号对应区间的大小, 单位为字节, $Size_{sector}$ 为每个扇区的字节数, $Index$ 为簇号值, N 表示该簇号对应区间所在扇区是此次读入暂存区的第几个扇区。

本发明还提供了一种可加速文件分配表操作的协处理器, 其特征在于, 包括主控制单元、数据存储单元以及文件分配表区接口单元, 其中:

- 所述数据存储单元用于存储发给所述协处理器的指令参数和从文件分配表区读取的数据;

所述文件分配表区接口单元用于在主控制单元的控制下, 读出所述非易失性存储器中给定扇区的数据, 并暂存到所述数据存储单元;

所述主控制单元用于根据指令和参数执行对文件分配表区的操作, 给出操作执行情况的指示信号, 有操作结果数据的, 存入到指定的存储位置。

- 进一步地, 上述协处理器还可以具有以下特点: 所述主控制单元包括: 簇链读取控制单元, 用于根据读取簇链的指令和给出的起始簇号, 从所述文件分配表区中读出该起始簇号之后相链簇链的各个簇号, 存放到指定的存储位置, 然后发出执行完毕的信号。

- 进一步地, 上述协处理器还可以具有以下特点: 所述主控制单元包括: 空间查找控制单元, 用于根据查找空间的指令和给定的空间需求, 查找文件分配表区, 如找到足够的剩余空间, 将这些空间对应的簇号存入所述指定的存储位置, 否则返回空间不足的结果; 以及簇链写入控制单元: 用于根据写入簇链指令和给出的文件数据的存储位置, 将存有该文件数据的每个簇的簇号写入其上一簇簇号在文件分配表区中的对应区间, 建立该文件的簇链, 然后发出执行完毕的信号, 同时, 所述数据存储单元还用于存储更新后待写入文件分配表区的数据; 所述文件分配表区接口单元还用于在主控制单元的控制下, 将更新后的数据写入到原扇区。

进一步地，上述协处理器还可以具有以下特点：所述主控制单元还包括簇链清除控制单元，用于根据清除簇链的指令和给定的簇号，将 FAT 表中该簇号之后相链的簇链清除，然后发出执行完毕的信号。

进一步地，上述协处理器还可以具有寄存器组、内部 RAM、地址转换单元，数据压缩子单元和内存接口单元，其功能与前述装置中描述的一样，不再重复。

由上可知，本发明通过提供一个协处理器，承担了对主处理器来说负担较重的对 FAT 表的操作，使处理器可以忽略数据读写的等待时间，直接进入下一步指令操作，从而大大提高文件系统的操作速度，同时减少了占用处理器用于文件系统操作的时间及资源。

附图说明

- 图 1 是文件系统结构的示意。
- 图 2 是传统的实现文件系统操作的装置的结构图。
- 图 3 是本发明实施例实现文件系统操作的装置的结构图。
- 图 4 是本发明实施例协处理器的功能框图。
- 图 5 是本发明实施例写入内存的数据的格式示意图。
- 图 6 是本发明实施例读取 FAT 表中簇链的流程图。
- 图 7 是本发明实施例清除 FAT 表中簇链的流程图。
- 图 8 是本发明实施例在 FAT 表中查找剩余空间的流程图。
- 图 9 是本发明实施例更新 FAT 表建立簇链的流程图。

具体实施方式

- 如图 3 所示，本实施例实现文件系统操作的装置包括通过总线连接的主处理器、存储器控制器、内存、协处理器，以及与存储器控制器连接的非易失性存储器。主处理器和协处理器都通过存储器控制器来执行对非易失性存储器的操作，协处理器还直接连接到内存，两者之间可以进行数据的双向传递。当然，协处理器也可以按常规方式通过总线与内存间进行通信。

在本发明中,新增的协处理器(包括硬件和在其中运行的软件)用于根据主处理器的指令和给定的参数,完成文件系统操作中对文件分配表(FAT表)内容的读取、空间查找、更新和删除等操作。该协处理器在硬件上可以是独立于主处理器的一个单独的逻辑计算模块,也可以是一个特定用途集成电路(ASIC)或其内部模块、计算设备或另一个处理器。

为了有助于理解本发明的具体实施例,这里先介绍一下FAT表的结构。FAT表区可以按1个半字节(FAT12, 12位)、2个字节(FAT16, 16位)、4个字节(FAT32, 32位)划分为多个区间(也可称之为单元、元素),每一区间对应于数据区的一个簇,按顺序为这些区间编号,称为簇号,该簇号除了是一个数据外,还对应于FAT表区的某一区间及数据区的一个簇。

在文件系统中,文件数据是以簇为最小单位存储的,而在FAT表中,每个簇号对应的区间都存放有该文件数据所在的下一个簇的簇号,从而构成一个链表,称为簇链,这样,只要知道首簇号,即可找到一个簇链,进而得到该文件所有内容的存储位置。本实施例中,当某一簇号对应区间的内容为0时,表示该簇为空白簇,当某一簇号对应区间的内容为0xFF8, 0xFF8 或 0x0FFFFFF8时,表示该簇是该文件的最后一个簇,另外,FAT表起始位置的若干区间可能并不用于存储下一簇号,而作其它用途。

如图4所示,本实施例的协处理器进一步包括地址转换单元、FAT区读写接口单元、内部RAM、内部RAM读写接口单元、内存读写接口单元、寄存器组 and 主控制单元。其中:

所述地址转换单元用于根据参数中或操作过程中给出的簇号,计算出FAT表中该簇所在的逻辑扇区号LBA以及在该扇区中的地址偏移量。

假定每个扇区的字节数为 $Size_{sector}$,FAT表区的起始扇区号为 FAT_{start} (该参数以及数据区的起始扇区号等参数均存储于BIOS参数块中), Len_{unit} 为每个簇号的长度(也即其在FAT表中对应区间的长度),Index为簇号值,有:

$$LBA = FAT_{start} + \left\lceil \frac{Index \times Len_{unit}}{Size_{sector}} \right\rceil \quad Len_{unit} = \begin{cases} 12bit, fat12 \\ 16bit, fat16 \\ 32bit, fat32 \end{cases} \quad (1)$$

上式中“ $\lfloor \rfloor$ ”表示向下取整运算, 计算时, Len_{unit} 的单位需要从 bit 转换为字节。在一实例中, 假定每个扇区为 512 个字节, 每簇包含 2 个扇区, 采用 FAT16 格式, 即每个簇对应的区间占 2 个字节, 待转换的簇号为 48, 则其对应的逻辑扇区号为: $FAT_{start}+0$ 。

- 5 而 $\frac{Index \times Len_{unit}}{Size_{sector}}$ 的余数即为该簇号对应区间在其扇区的偏移量, 对于上面的实例, 其在扇区中的地址偏移量为: 96。

- 10 该地址转换单元还用于将获取的簇号和连续簇号数转换成扇区号和连续扇区数的形式存储到所述指定位置, 及/或将指定位置存储的扇区号和连续扇区数转换为簇号和连续簇号数, 供主控制单元使用。具体的转换方法在后述的步骤中再介绍。

所述 FAT 区读写接口单元用于在主控制单元的控制下, 根据地址转换单元提供的扇区号, 通过存储器控制器读出该扇区上的 FAT 表数据, 并暂存到内部 RAM 中, 或者将内部 RAM 中的数据写入该扇区。

- 15 所述内部 RAM 用于暂存从相应扇区读出的数据或者需写入相应扇区的数据, 其大小在本实施例等于一个扇区, 即 512bytes, 其目的在于只需采用 512 字节的缓冲区大小, 从而节省缓冲区空间; 当然, 也可以根据具体应用, 一次读取多个扇区记录, 这种方式的优点是可以进一步加速访问, 缺点是需要大容量的缓冲区, 当一次读取多个扇区记录时, 可以从当前扇区开始, 依次读取指定的多个扇区。

- 20 所述内部 RAM 读写接口单元用于在主控制单元的控制下, 完成对内部 RAM 数据的读写操作。

所述内存读写接口单元用于在主控制单元的控制下, 完成对与协处理器相连的内存中数据的读写。当协处理器通过总线与内存通信时, 该内存读写接口单元即为一个外部总线读写接口单元。

- 25 所述寄存器组用于保存传送给协处理器的控制指令和参数, 并保存协处理器执行操作后返回的参数, 供其它外部模块读取。

上述的各种接口单元主要用于将主控制单元对其它模块的操作指令转换为实际操作的信号, 即用于与其它不同类型的模块的时序通信信号的产生

和接收。

所述主控制单元进一步包括：

- 簇链读取控制单元：用于根据读取簇链的指令和给定的起始簇号，从
- 5 FAT 表中读出该起始簇号之后相链簇链的各个簇号，并转换成逻辑扇区号，按规定的压缩格式存放到指定的存储位置，然后发出执行完毕的信号。

簇链清除控制单元：用于根据清除簇链的指令和给定的簇号，将 FAT 表中该起始簇号之后相链的簇链清除，释放其在 FAT 表区及相应数据区占用的空间，然后发出执行完毕的信号。

- 10 空间查找控制单元：用于根据查找空间的指令和给定的空间需求，查找 FAT 表区，如找到足够的剩余空间，将这些空间对应的簇号转换为逻辑扇区号，按规定的压缩格式存储到指定位置，返回执行成功的结果，否则返回空间不足的结果。

- 簇链写入控制单元：用于将文件数据写入查找到的剩余空间后，根据写
- 15 入簇链的指令，完成扇区号到簇号的转换后，将存有该文件数据的每个簇的簇号写入其上一簇簇号在 FAT 表中的对应区间，在 FAT 表区建立该新文件数据的簇链，然后发出执行完毕的信号。

- 上述将簇号转换为相应扇区号（还涉及到连续簇号数到连续扇区数的转换）由地址转换单元完成，将得到的簇号转换为压缩格式则由主控制单元中的
- 20 的数据压缩子单元（图 4 中没有示出）完成，这两个子单元供上述相关的控制单元调用，具体的转换方法将在下文中介绍。

协处理器中的上述控制单元可以通过装置中的中断线，或者其它信号线来发出反映操作执行情况的信号。

- 在本实施例中，上述的指令都是由主处理器发送给协处理器的，但是本
- 25 发明不局限于此，也可以由另一协处理器（逻辑模块，可以与本发明的协处理器在一个芯片上）来完成对文件系统的操作，向本发明的协处理器发送上述指令和参数。而本发明协处理器所返回的参数和数据，可以由主处理器、另一协处理器或者具有直接读写功能的存储器控制器使用，完成后续的文件

操作。

下面详细介绍本实施例完成对 FAT 表的读取、空间查找、更新和删除操作的流程。这些流程只是一种具体的实现方式，其中的各种循环判断步骤及各步骤先后顺序等，完全可以有多个不同的方案，只要能完成相应的功能即可。

协处理器读取 FAT 表中簇链的流程图如图 6 所示，用于读文件的操作中，包括以下步骤：

步骤 110，协处理器收到读取簇链的指令和参数，包括起始簇号、在内存中分配的缓冲区起始地址和长度，以该起始簇号为当前簇号，将总簇号的计数值置为 1，连续簇号的计数值置 1；

步骤 120，根据当前簇号计算出其对应区间所在的逻辑扇区号以及在该扇区中的偏移量（可以字节或半字节为单位），然后判断该逻辑扇区的数据是否已读出，如果未读出，先读出该扇区上的 FAT 表数据并暂存到内部 RAM 中，执行下一步；否则，直接执行下一步；

这里，因为文件簇链上的簇号不连续时，下一簇号也可能在同一扇区上，所以先判断一下是否已读出该扇区的数据，以免重复读取，加快速度。

步骤 130，根据计算出的所述偏移量，将指针指向当前簇号在内部 RAM 中对应的区间；

步骤 140，读出该区间的数，并判断该数据是否等于当前簇号加 1，如果等于，执行步骤 150，否则，执行步骤 160；

步骤 150，将当前簇号值加 1，连续簇号和总簇号的计数值加 1，然后判断当前簇号对应区间是否在下一扇区，如果是，返回步骤 120（在步骤 120 中，可以直接将要读取的扇区设为下一扇区，将偏移量设为 0，不必实际运算），否则将指针指向内部 RAM 中的下一区间，返回步骤 140；

步骤 160，将此时的起始簇号转换为逻辑扇区号，并根据此时连续簇号的计数值计算出相应的连续扇区数，组成规定格式的一条数据，然后将读出的数据作为新的起始簇号；

将簇号转换为逻辑扇区号的公式如下:

$$LBA = Position_{data} + (Cluster - 2) \times Sectors_{cluster}$$

其中, LBA 为逻辑扇区号, $Position_{data}$ 表示数据区的起始扇区号, Cluster 为簇号, $Sectors_{cluster}$ 表示每个簇包含的扇区数。

- 5 图 5 示出了该条数据的格式, 其由逻辑扇区号和长度字段组成, 其中的逻辑扇区号占用 4 个(或 6 个)字节, 可同时适用于 FAT12、FAT16 和 FAT32 格式, 而长度即为计算出的连续扇区数, 可占用 2 或 4 个字节。

- 该格式是一种压缩格式, 如果不采用这种方式, 协处理器虽然也可以代替主处理器完成一部分的 FAT 表操作, 但每得到一个簇号就需要写入内存一次, 就需要占用较大的空间, 或者需要主处理器或其它后续处理模块进行多次的读取。
- 10

- 步骤 170, 将组织好的该条数据写入到内存中的当前位置(该当前位置一开始为指定的起始地址, 每写入一条数据后, 向后偏移 6-8 个字节), 并对写入内存的数据条数进行累计;
- 15

步骤 180, 判断当前簇号对应区间的数据是否为文件结束标志(如 0xFF8, 0xffff8 或 0x0FFFFFFF8)或者总簇号的计数值已达到最大允许值(防止出现死循环), 如果是, 执行步骤 210, 否则, 执行步骤 190;

- 步骤 190, 判断分配的缓冲区是否用完, 如果没有用完, 执行步骤 200; 否则, 执行步骤 210;
- 20

步骤 200, 令当前簇号=当前簇号对应区间中的簇号, 总簇号的计数值加 1, 并将连续簇号的计数值置 1, 返回步骤 120;

- 步骤 210, 发出本次读取结束的指示信号(如中断信号), 将统计的总簇数、写入内存的数据条数、此次操作读出的最后一个簇号、文件是否读完的标志写入寄存器组, 作为返回的参数。
- 25

按上述流程执行一次操作, 可以根据起始簇号, 从 FAT 表中读出该文件簇链的各个簇号, 直到读完该文件的簇链、出错或者分配的缓存已填满。可以作为打开文件的一个环节。写入缓存的数据条数等于簇链上连续簇号的

组数,有可能只占用了部分分配的缓存空间,后续处理模块可根据该参数读出有用的数据。但是,也有可能出现一次操作缓存已满而簇链还没有读完的情况,此时主处理器或其它模块根据返回的结束标志判断是否读完(根据总簇数判断也可以),如未读完则以最后一个簇号为起始簇号,其内容为 0xFF8, 5 0xffff8 或 0x0FFFFFFF8h 时以最后一个簇号为起始簇号,重新分配缓存空间,然后指示协处理器再次运行上述流程。

在又一实施例中,协处理器也可不完成将起始簇号转换成逻辑扇区号,及根据连续簇号的计数值算出连续扇区号的工作,而是直接在每条数据中记录起始簇号和连续簇号的计数值,将这一转换工作交给后续处理模块。

10 由上述可见,本发明的 FAT 表簇链读取方法与传统的纯软件处理的方法相比较的优点在于:主处理器只需要发出一次读取簇链的指令和参数,协处理器就可以在极短的时间内获得该簇链的所有簇,并以压缩方式存入内存供主处理器使用;并且本发明所必需的缓冲区大小比纯软件方法要小很多;此外,传统的软件处理方法中,主处理器需要频繁的读取 FAT 表数据区, 15 或者需要占用相当大的缓冲区,而且由于速度限制不大可能一次将整个簇链读取出来。

协处理器清除 FAT 表中簇链的流程图如图 7 所示,包括以下步骤:

步骤 310,协处理器收到清除簇链的指令和给定的起始簇号,以起始簇 20 号为当前簇号;

步骤 320,根据当前簇号计算出其对应区间所在的逻辑扇区号及在该扇区中的偏移量;

步骤 330,判断该逻辑扇区的数据是否已读出,如果未读出,读出该扇区上的 FAT 表数据并暂存到内部 RAM 中,执行下一步,如果已读出,则直接 25 执行下一步;

步骤 340,根据计算出的所述偏移量,将指针指向当前簇号在内部 RAM 中所对应的区间;

步骤 350,将该区间的数据读出后,将其置为空白标识符,可用“0”表示;

步骤 360, 判断读出的该区间数据是否等于当前簇号加 1, 如果等于, 执行步骤 370, 否则, 执行步骤 390;

步骤 370, 将当前簇号值加 1, 然后判断当前簇号对应区间是否在下一扇区, 如果是, 执行步骤 380, 否则将指针指向内部 RAM 中的下一区间 (步骤 370a), 返回步骤 350;

步骤 380, 将更新后内部 RAM 的数据写入到原逻辑扇区, 返回步骤 320;

步骤 390, 判断当前簇号对应区间的数据是否为文件结束标志, 如果是, 执行步骤 410, 否则, 执行步骤 400;

步骤 400, 令当前簇号=当前簇号对应区间中的簇号, 返回步骤 380;

10 步骤 410, 将更新后内部 RAM 的数据写入到原逻辑扇区, 给出处理完成的指示信号, 结束。

按上述流程执行一次操作, 即可根据给出的起始簇号, 将 FAT 表中该簇链之后相链簇链上的原有数据全部清除。该流程可以作为删除文件的一个环节。

15

在向非易失性存储器的数据区写文件时, 首先需要查找 FAT 表, 看是否有足够的剩余空间供写入, 如果有, 将文件数据写入到找到的空间内。在文件数据写入完成后, 再在 FAT 表中建立该文件的簇链。即对于本发明的协处理器来说, 一个写入文件的过程先后要执行两个操作: 在 FAT 表中查找剩余空间, 以及更新 FAT 表区以建立新写入文件的簇链。

20

协处理器查找 FAT 表中空闲簇链的流程图如图 8 所示, 包括以下步骤:

步骤 510, 协处理器收到查找剩余空间的指令, 并获得起始簇号、要求的最小连续簇号数目、内存中分配的缓冲区起始地址及其长度、文件需要的簇的最大数目等参数, 将总空闲簇和连续簇号的计数值置 0, 以起始簇号为当前簇号;

25

该步中, 传送的参数也可以不包括起始簇号, 协处理器可以从默认的扇区开始查找。

步骤 520, 根据当前簇号计算出其对应区间所在的逻辑扇区号及在该扇

区中的偏移量;

步骤 530, 读出当前扇区上的 FAT 表数据并暂存到内部 RAM 中;

步骤 540, 根据当前偏移量, 读出内部 RAM 对应区间的数据, 判断该数据是否等于“0”, 如果等于, 执行步骤 550, 否则, 执行步骤 580;

- 5 步骤 550, 将当前簇号加 1, 连续簇号的计数值加 1, 判断总空闲簇的计数值是否等于文件需要的簇的最大数目, 如果是, 执行步骤 630, 否则, 执行步骤 560;

步骤 560, 判断当前簇号对应区间是否在下一扇区, 如果是, 执行步骤 570, 否则令当前偏移量增加一个区间的长度 (步骤 560a), 返回步骤 540;

- 10 步骤 570, 判断是否已查找了一个 FAT 表的所有扇区, 如果是, 执行步骤 620, 否则, 令当前扇区指向下一个要查找的扇区 (如果是从中间开始查找, 可能会返回前面部分的扇区), 当前偏移量置 0 (步骤 570a), 返回步骤 530;

- 15 步骤 580, 判断此时连续簇号的计数值是否大于等于要求的最小连续簇号数目, 如果是, 令新的总空闲簇的计数值等于其原计数值加上连续簇号的计数值 (步骤 580a), 执行步骤 600, 否则, 执行步骤 590;

步骤 590, 将当前簇号加 1 并作为新的起始簇号, 将连续簇号的计数值置 0, 返回步骤 560;

- 20 步骤 600, 将此时的起始簇号转换为逻辑扇区号, 并根据此时的连续簇号的计数值计算出连续扇区数, 组成规定压缩格式 (如图 5) 的一条数据, 写入到内存中的当前位置, 并对写入的数据条数进行累计;

步骤 610, 判断分配的缓冲区是否已满, 如果未满, 执行步骤 590, 否则, 执行步骤 640;

- 25 步骤 620, 判断此时连续簇号的计数值是否大于等于要求的最小连续簇号数目, 如果是, 令新的总空闲簇的计数值等于其原计数值加上连续簇号的计数值 (步骤 620a), 执行步骤 630, 否则, 执行步骤 640;

步骤 630, 将此时的起始簇号转换为逻辑扇区号, 并根据此时的连续簇

号的计数值计算出连续扇区数,组成规定格式的一条数据,写入到内存中的当前位置,并对写入的数据条数进行累计;

步骤 640,发出本次查找结束的指示信号,将统计的总空闲簇的计数值、写入内存的数据条数和结束时的当前簇号作为返回的参数。

- 5 按上述流程执行一次操作,可以从 FAT 表中查找空闲簇,直到找到足够数量的空闲簇、分配的缓冲已满或者查找完整个 FAT 表也没有找到足够的空闲簇。如果因分配的缓冲已满而中止(此时返回参数中的簇号与给定的起始簇号不同),主处理器或其它模块可以将此次操作读出的最后一个簇号为起始簇号,重新分配缓存空间,再次指示协处理器进行查找。
- 10 给定连续簇号数目是为了避免找到的空闲簇过于零散,而要求其具有一定的连续性,以方便后续的文件操作。该参数不是必须的。

- 15 当主处理器或其它模块将文件数据写入到找到的空闲簇后,需要在 FAT 表中建立该文件的簇链,此时,该模块会向本发明的协处理器发出更新 FAT 表区建立文件簇链的指示,并在内存的指定位置给出文件实际占用的簇,同样按照上述压缩格式的多条数据来表示,即每条数据由起始扇区号和从该扇区开始的连续扇区数组成的。

协处理器更新 FAT 表区建立文件簇链的流程图如图 9 所示,包括以下步骤:

- 20 步骤 710,协处理器收到更新 FAT 表区的指令和相关参数,包括内存中数据存放的缓冲区起始地址和各个数据条数加起来的总长度,总长度= $I \times 6$ 或 $I \times 8$, I 为存放的数据条数,令表示已读取数据条数的计算值 $i=0$;

- 25 步骤 720,从内存读出第 i 条数据,将其中的起始扇区号转换为当前簇号,根据连续扇区长度计算出连续簇号数 M_i (包括第 1 个簇号,且假定 $M_i > 1$),设连续簇号计数值 $m=1$;

步骤 730,判断 i 是否等于 0,如等于,执行步骤 750,否则,执行步骤 740;

步骤 740,将当前簇号作为数据写入当前偏移量在内部 RAM 所指向的

区间;

步骤 750, 根据当前簇号计算出其对应区间所在的逻辑扇区号及在该扇区中的偏移量, 将其作为当前偏移量, 判断该逻辑扇区的数据是否已读出, 如果已读出, 执行步骤 770, 否则, 执行步骤 760;

- 5 步骤 760, 将 RAM 中已更新的数据写入到非易失性存储器的原扇区(但 $i=0$ 时不必进行该操作), 然后读出当前簇号对应区间所在扇区的 FAT 表数据, 将其暂存到内部 RAM 中;

步骤 770, 将当前簇号加 1, 并将加 1 后得到的簇号作为数据写入当前偏移量在内部 RAM 所指向的区间;

- 10 步骤 780, 将当前偏移量增加一个区间的长度, 判断当前簇号对应区间是否在下一扇区, 如果是, 执行步骤 810, 否则, 执行步骤 790;

步骤 790, 令 $m=m+1$, 判断 m 是否等于 M_i , 如果等于, 执行步骤 800, 否则, 执行步骤 770;

- 15 步骤 800, 令 $i=i+1$, 判断 i 是否等于 I , 如果是, 执行步骤 820, 否则, 执行步骤 720;

步骤 810, 将 RAM 中已更新的数据写入到非易失性存储器原扇区, 读出下一扇区的 FAT 表数据并暂存到内部 RAM 中, 将当前偏移量置 0, 执行步骤 790;

- 20 步骤 820, 将结束标志数据写入当前偏移量在内部 RAM 所指向的区间, 然后将 RAM 中已更新的数据写入到非易失性存储器的原扇区, 给出操作完成的指示信号, 结束。

按上述流程执行一次操作, 可以根据给出的文件写入簇的信息号, 更新 FAT 表相应区域的数据, 在 FAT 中建立该文件的簇链。

- 25 当允许 $M_i=1$ 时, 需要在步骤 770 中先加一个判断, 即 $M_i=1$ 时, 直接返回步骤 720, $M_i>1$ 时, 再执行原步骤 770 的操作。

对于上述修改 FAT 表的操作, 本发明所采用的方法极大的加速了对 FAT 表的改写过程, 每次只修改相关的几个 FAT 扇区, 而不需要重写整个 FAT 表; 另外, 从开始修改操作到整个过程结束也只需要极短的时间。

可以看出,利用上述装置和方法,在执行读文件操作时,协处理器根据指示和相应参数,可以将 FAT 表区的簇链读出并放到指定位置。在执行删除文件操作时,协处理器可以清除 FAT 表该文件的簇链。在执行写入文件操作时,协处理器可以从 FAT 表中查找到空闲的簇并放到指定位置,而在文件数据写入完成后,可以更新 FAT 表数据建立该文件的簇链。

需要说明的是,本领域的技术人员基于本发明公开的技术方案,可以很容易地对上述实施例的具体流程进行变换,例如:

在又一实施例中,文件操作系统只需要进行读文件的操作,如只需要进行语音播放,这时,可以将文件写入、删除相关的模块和流程都删除掉,只保留读文件相关的控制模块即可。相反,如果文件操作系统只需要进行写文件的操作,如只需要进行语音录制,则可以将文件读出相关的模块和流程都删除掉,保留写入和删除文件相关的控制模块即可。

在又一实施例中,协处理器从 FAT 表中读取的簇链不是放在内存中,而是直接存入在内部的寄存器中供其它模块来读取,这会对一次操作存储的数据数量有一定的限制,但同样是可行的。此外,该实施例中,协处理器收到的指令参数已经采用了扇区号的形式,并给出了地址偏移量,不必再进行地址的转换。

在又一实施例中,主处理器只给协处理器一个规定格式数据长度(如6-8个字节)的缓存,协处理器在内存中写入一条数据后,就发出一次中断,通知主处理器(或其它模块)将存储的数据取出,并在得到读取成功的响应后再写入下一条数据。这时,主处理器传递给协处理器的参数中可以不包含缓冲区的长度,而协处理器返回的参数中可以不包括数据条数,也不必进行缓存是否已满的判断。但该方法中,需要主处理器在协处理器一次操作中做出多次响应。

在另一实施例中,协处理器内部 RAM 的容量等于多个扇区的大小,如2个、4个等等,这样,每次可以读取 FAT 表区多个扇区的数据进行处理,在实现流程上与上述实施例基本相同,只是在处理了 RAM 中多个扇区的数据后,再从 FAT 表区读取新的数据,另外在计算当前簇号对应区间的数据

在暂存区的位置时，除了算出的在其所在扇区的偏移量外，还要考虑加上内部 RAM 中在前扇区的字节数。可采用以下修正后的公式计算：

$$W = \frac{Index \times Len_{unit}}{Size_{sector}} \text{ 的余数加上 } (N-1) \times Size_{sector}$$

- 其中： W 表示该位置在暂存区的偏移量， Len_{unit} 为每个簇号对应区间的大小，单位为字节， $Size_{sector}$ 为每个扇区的字节数， $Index$ 为簇号值， N 表示该簇号对应区间所在扇区是此次读入暂存区的第几个扇区。该实施例在文件数据比较集中时，可以加快处理的效率。
- 5

在另一实施例中，也可以直接从片外存储设备读取数据，而不需要通过存储控制器来完成该操作。

- 10 因此，本发明的保护范围不局限于具体实施例的方案，而应以权利要求书为准。本发明同样适用于有主备两个 FAT 表的情况，在一个 FAT 数据修改完成后与另一个同步即可。

- 综上所述，本发明方法将文件系统中最为耗时的操作之一：对 FAT 表的操作，改为由独立于主处理器之外的协处理器来完成，从而提高了文件系
- 15 统操作的速度。同时，主处理器不必承担对 FAT 表区的大量访问工作，而后续操作所需的数据和参数可以从指定的位置直接读出，从而为主处理器节约了大量处理时间和资源。

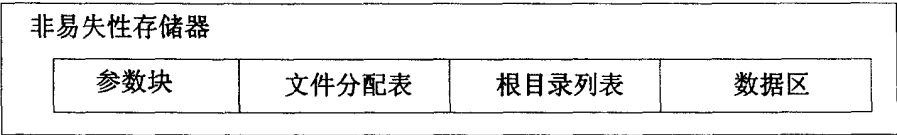


图 1

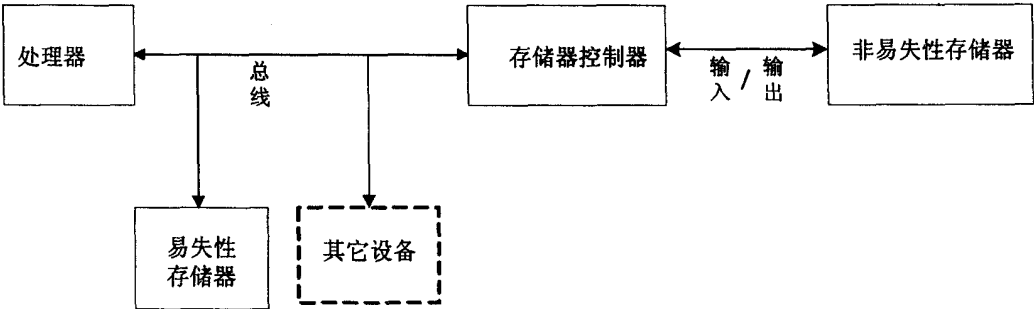


图 2

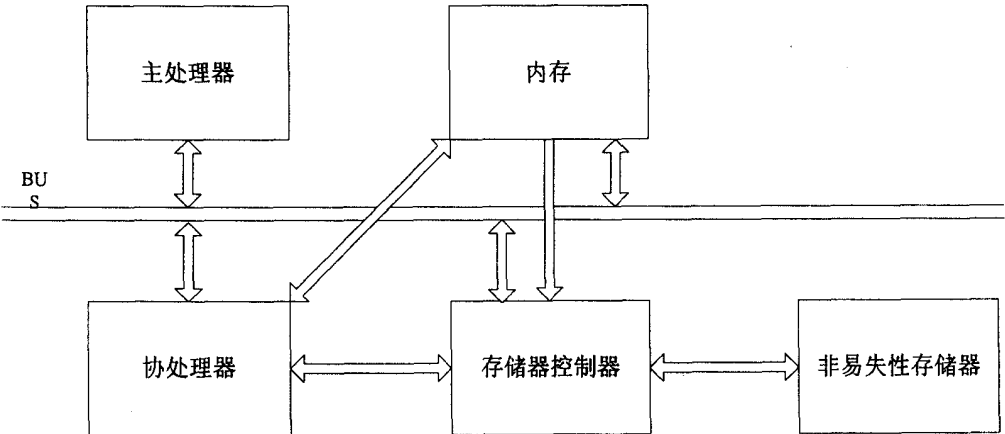


图 3

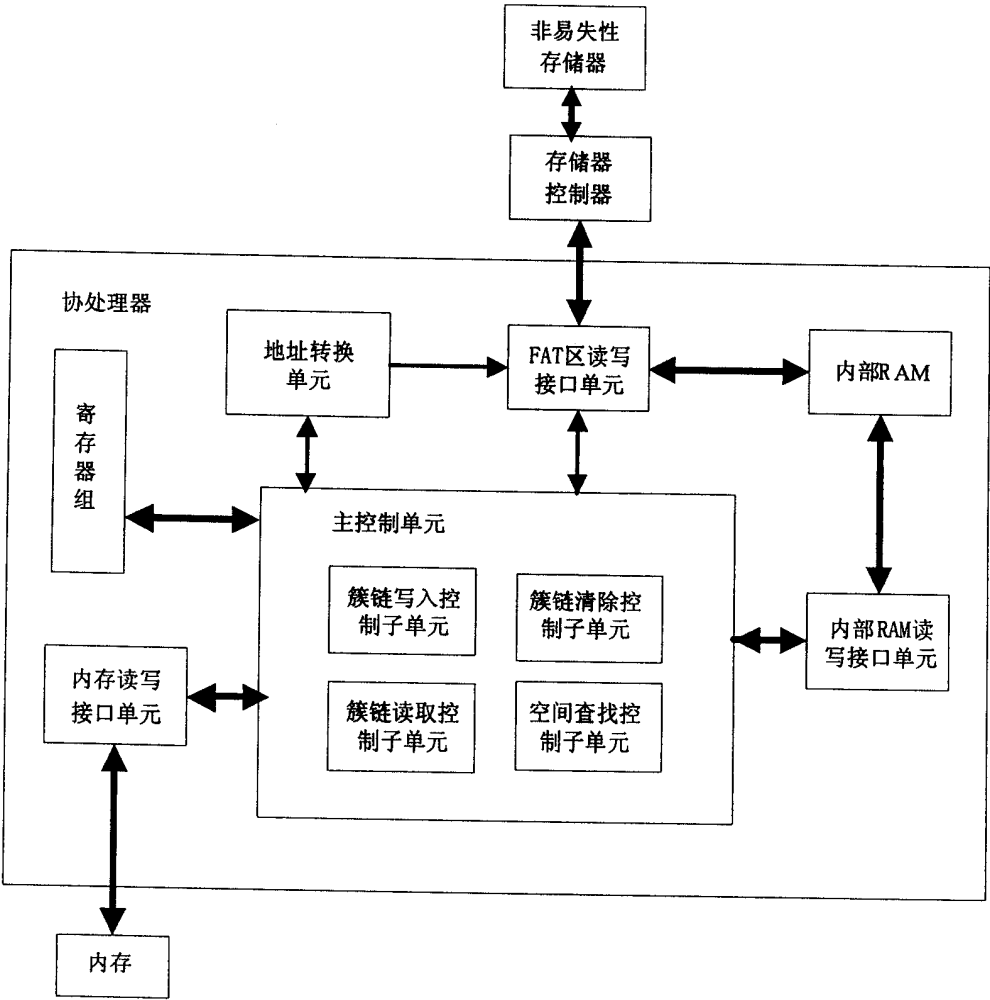


图 4

逻辑扇区号(4 Bytes)	长度 (2/4Bytes)
----------------	---------------

图 5

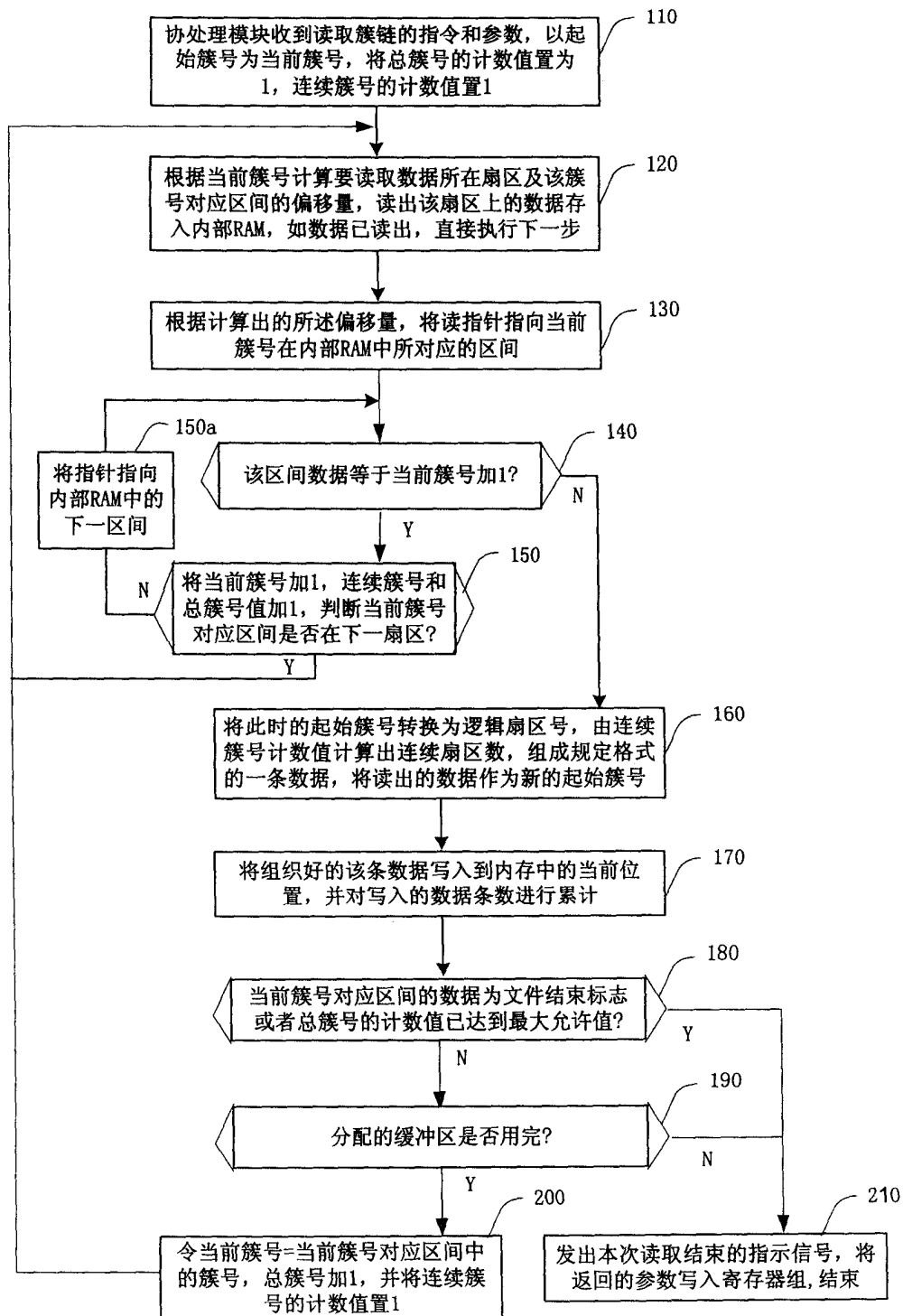


图 6

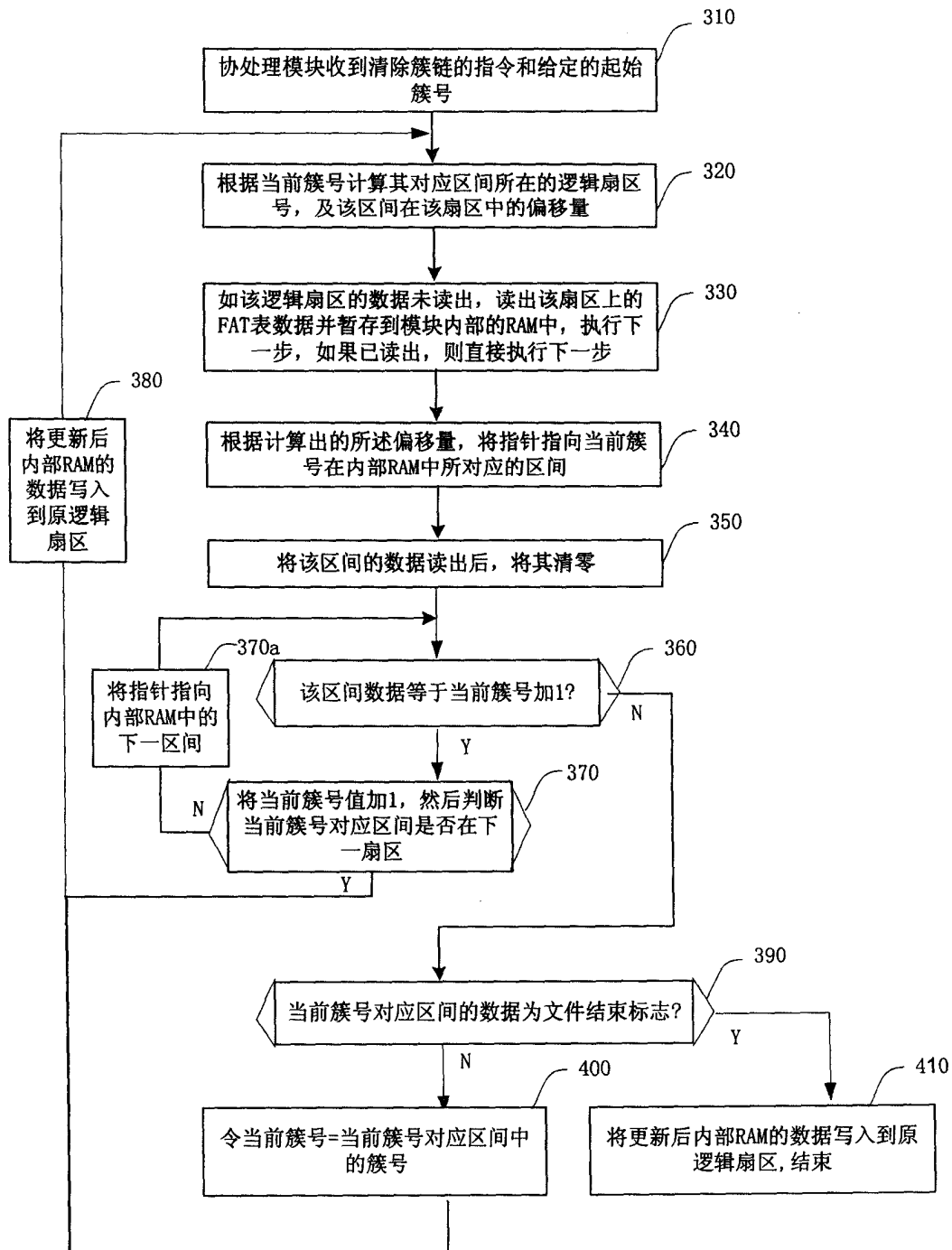


图 7

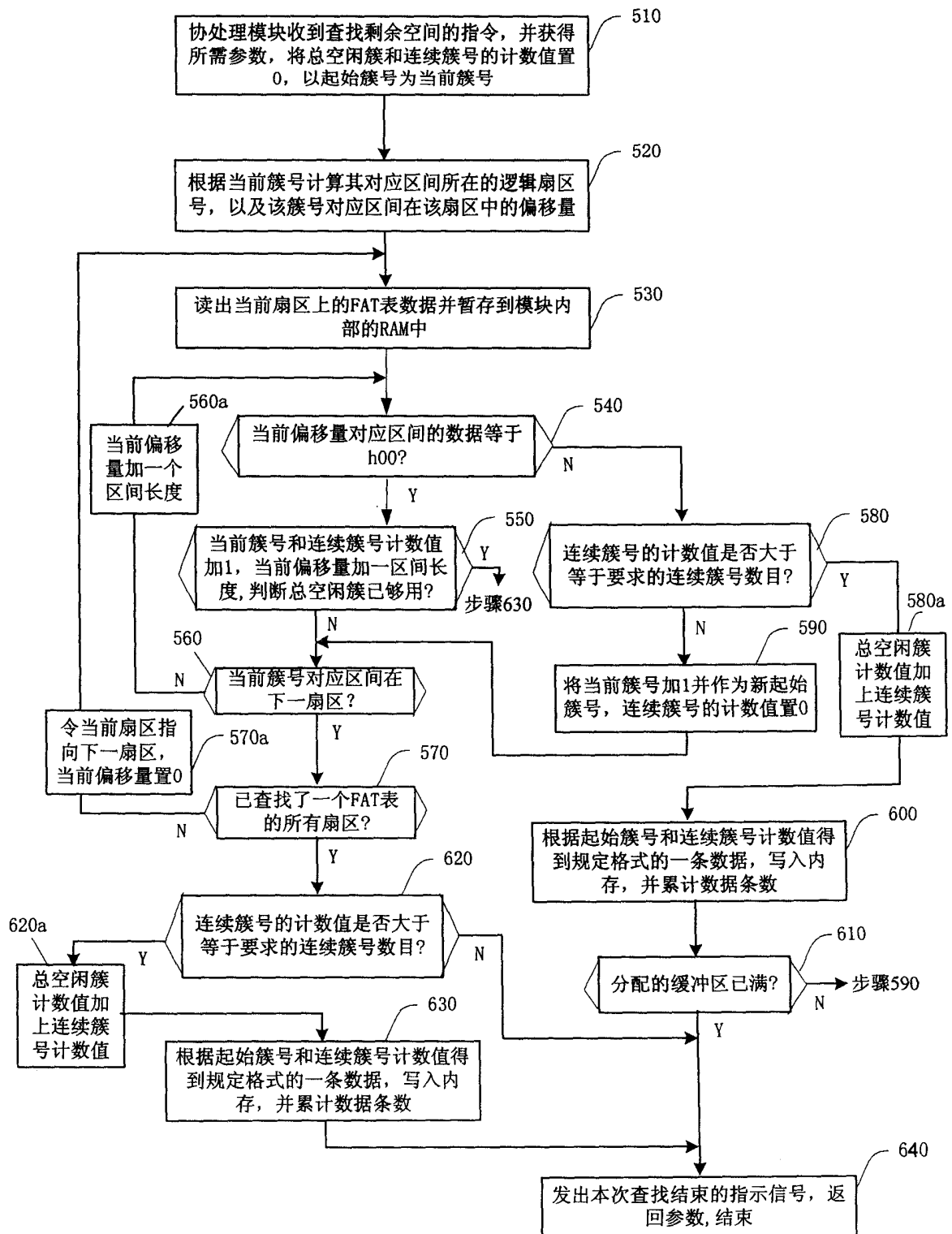


图 8

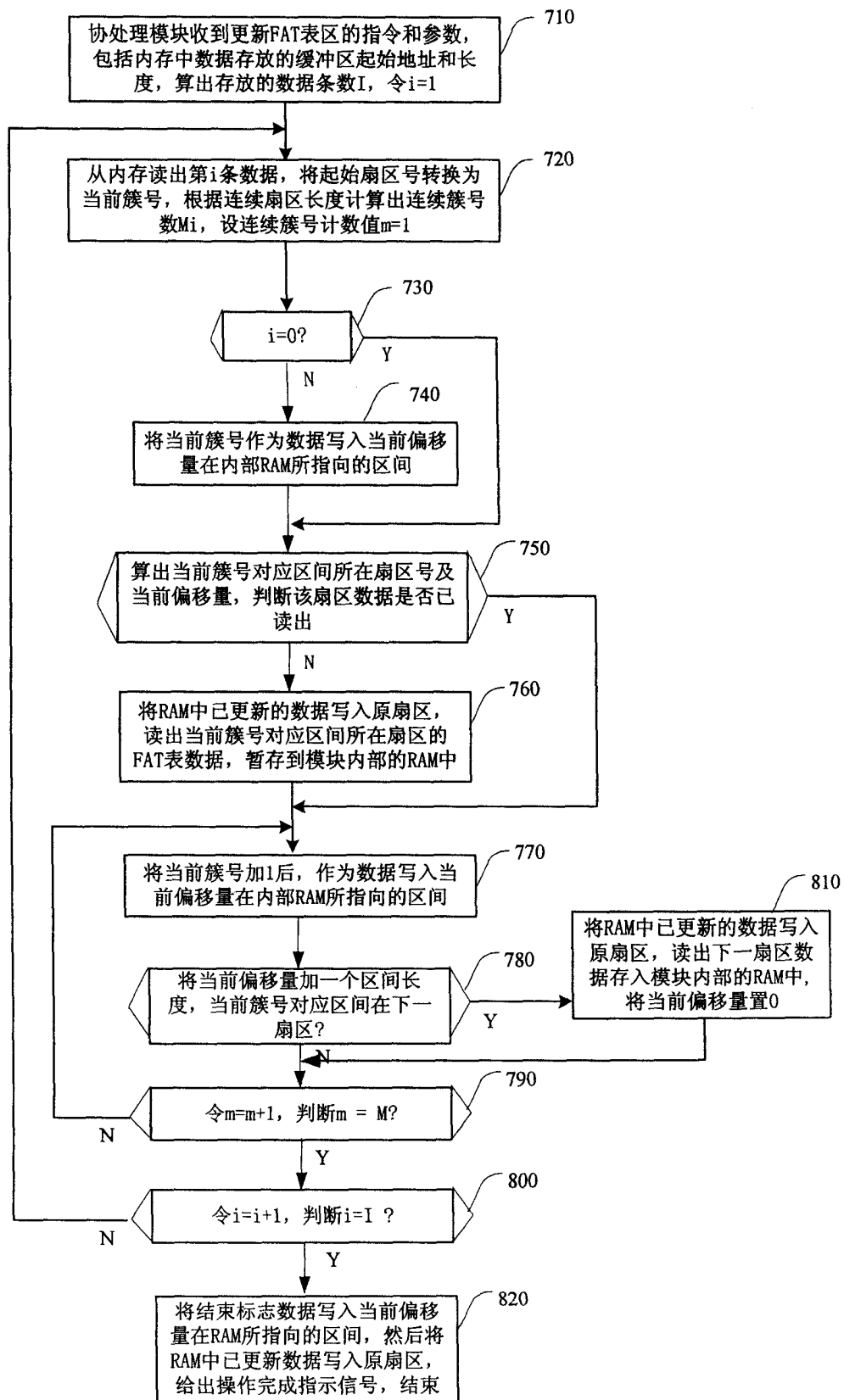


图 9