



(12) 发明专利

(10) 授权公告号 CN 101278267 B

(45) 授权公告日 2012. 08. 22

(21) 申请号 200680036085. 2

(51) Int. Cl.

(22) 申请日 2006. 08. 01

G06F 12/02 (2006. 01)

(30) 优先权数据

(56) 对比文件

60/705, 388 2005. 08. 03 US

CN 1469391 A, 2004. 01. 21,

11/259, 423 2005. 10. 25 US

US 20050149686 A1, 2005. 07. 07,

11/259, 439 2005. 10. 25 US

审查员 孟宪超

(85) PCT申请进入国家阶段日

2008. 03. 28

(86) PCT申请的申请数据

PCT/US2006/030166 2006. 08. 01

(87) PCT申请的公布数据

W02007/019198 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·韦尔什·辛克莱

(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

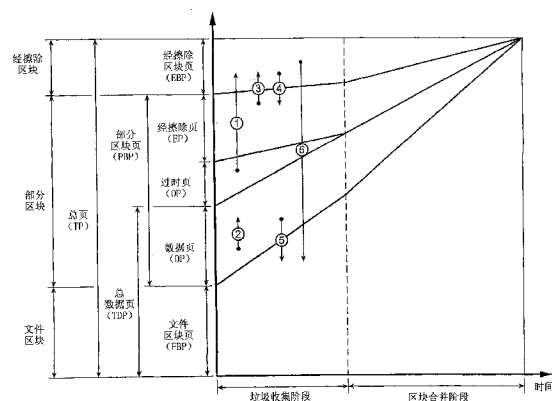
权利要求书 2 页 说明书 19 页 附图 14 页

(54) 发明名称

具有经调度回收操作的非易失性存储器

(57) 摘要

在非易失性存储器阵列中, 在出现经擦除区块的短缺之前发生的回收操作的调度避免了可能超过时间限制的延长的回收周期。存储器控制器使用关于存储在所述存储器阵列中的数据的信息来估计可被编程的额外主机数据和将执行的回收操作, 并调度所述回收操作以使其均匀分布在写入操作之间, 直到所述存储器为满为止。



1. 一种管理具有区块擦除单位的非易失性存储器阵列中的空间的方法,所述非易失性存储器阵列具有部分填充有有效数据的区块,所述方法包括:

根据交错比率交错 (a) 从主机接收的主机数据到所述非易失性存储器阵列的编程与 (b) 先前存储的有效数据从一个区块到另一区块的复制,所述交错比率是从对将填充所述非易失性存储器阵列的额外数据的估计及对将被复制以回收所述存储器阵列中的空间的有效数据量的估计中导出的,且所述交错比率被周期性重新计算以提供自适应调度。

2. 根据权利要求 1 所述的方法,其中所述交错提供从所述主机接收的主机数据到所述非易失性存储器的恒定编程速率。

3. 根据权利要求 2 所述的方法,其中计算所述交错比率以持续以所述恒定速率的所述编程,直到所述非易失性存储器充满有效数据为止。

4. 根据权利要求 1 所述的方法,其中响应于主机命令重新计算所述交错比率。

5. 根据权利要求 1 所述的方法,其中所述交错比率是从该对将填充所述非易失性存储器阵列的额外数据的估计与对将被复制以回收所述存储器阵列中的空间的有效数据量的估计的比率中导出。

6. 根据权利要求 1 所述的方法,其中所述对可存储在所述存储器阵列中的额外数据量的估计是从自主机接收的指示主机文件过时的信息中导出。

7. 根据权利要求 1 所述的方法,其中所述交错响应于从多个模式中选择交错模式的主机命令而发生,所述多个模式包含禁止复制先前存储的数据的回收关闭模式以及允许复制先前存储的数据而不对主机数据进行编程的回收打开模式。

8. 一种管理具有区块擦除单位的非易失性存储器中的空间的方法,其包括:

根据交错比率,在重复循环中交错 (a) 从主机接收的主机数据到所述非易失性存储器的编程与 (b) 先前存储的有效数据从一个区块到另一区块的复制,使得在所述存储器填充有主机数据时,在多个循环中以恒定速率对主机数据进行编程,以提供自适应调度。

9. 根据权利要求 8 所述的方法,其中所述交错比率是从对可编程到所述存储器阵列的额外主机数据总量的估计以及对必须从一个区块复制到另一区块以回收所述存储器阵列中的所有可回收空间的先前存储的有效数据总量的估计中算出。

10. 根据权利要求 9 所述的方法,其中通过假定有效数据均等地分布在部分充满有效数据的区块中来获得所述对必须复制的先前存储的有效数据的估计。

11. 一种使用存储器控制器来管理包含多个区块的非易失性存储器阵列的方法,其中区块是最小擦除单位,所述方法包括:

维持存储在所述非易失性存储器阵列中的主机数据的逻辑到物理映射的记录,所述记录具有多个条目,个别条目通过唯一文件识别符和偏移量指示逻辑地址,

根据主机提供的信息确定所存储数据是否有效或过时,

根据将被复制以回收剩余可回收空间的数据量和可在所述存储器阵列变满之前写入的额外主机数据总量的交错比率来编程由所述非易失性存储器阵列从所述主机接收的主机数据并复制有效数据。

12. 根据权利要求 11 所述的方法,其中响应于主机命令重新计算所述交错比率。

13. 根据权利要求 11 所述的方法,其中计算所述交错比率以提供主机数据的恒定编程速率,直到所述存储器填充有有效主机数据为止。

14. 根据权利要求 11 所述的方法,其中所述存储器阵列实施在通过标准接口与所述主机通信的可移除存储器卡中。

15. 根据权利要求 11 所述的方法,其中所述主机提供使用唯一文件识别符来指主机文件的关于主机文件的信息。

具有经调度回收操作的非易失性存储器

技术领域

[0001] 本申请涉及一种管理具有区块擦除单位的非易失性存储器中的空间的方法及该存储器系统。

背景技术

[0002] 本申请案涉及例如半导体快闪存储器的可重新编程的非易失性存储器系统的操作,且更具体地说,涉及所述存储器内可用空间的管理。本文引用的所有专利、专利申请案、文章和其它出版物、文档和事物出于所有目的在此以全文引用的方式并入本文中。

[0003] 在早一代商业快闪存储器系统中,将矩形存储器单元阵列划分为大量单元群组,每一者存储标准磁盘驱动器扇区的数据量,即 512 个字节。每个群组中还通常包含例如 16 个字节的额外数据量,以存储错误校正码 (ECC) 和可能的与用户数据和 / 或其中存储用户数据的存储器单元群组相关的其它开销数据。每个此群组中的存储器单元是可共同擦除的最小数目的存储器单元。也就是说,在所述存储器系统中,擦除单位有效地是存储一个数据扇区和所包含的任何开销数据的大量存储器单元。此类型的存储器系统的实例描述于第 5,602,987 号和第 6,426,893 号美国专利中。需要在用数据对其再编程之前擦除存储器单元是快闪存储器的特性。

[0004] 最常见以存储器卡或快闪驱动器的形式提供快闪存储器系统,所述存储器卡或快闪驱动器与例如个人计算机、相机等多种主机可移除地连接,但也可嵌入在这些主机系统内。当向存储器写入数据时,主机通常向存储器系统的连续虚拟地址空间内的扇区、群集或其它数据单位指派唯一的逻辑地址。类似于磁盘操作系统 (DOS),主机向存储器系统的逻辑地址空间内的地址写入数据以及从其读取数据。存储器系统内的控制器将从主机接收的逻辑地址转译为实际存储数据的存储器阵列内的物理地址,且接着跟踪这些地址转译。存储器系统的数据存储容量至少与在为存储器系统界定的整个逻辑地址空间上可寻址的数据量一样大。

[0005] 在较晚的几代快闪存储器系统中,擦除单位的大小增加到足够的存储器单元的区块以存储多个数据扇区。尽管与存储器系统连接的主机系统可以例如扇区的较小的最小单位来编程和读取数据,但大量扇区存储在快闪存储器的单个擦除单位中。常见的是区块内的一些数据扇区在主机更新或替换逻辑数据扇区时变过时。由于必须在可对区块中存储的任何数据进行覆写之前擦除整个区块,因此新的或更新的数据通常存储在已经擦除且具有用于数据的剩余容量的另一区块中。此过程清空了带有占据存储器内有价值的空间的过时数据的原始区块。但如果其中保留有任何有效数据,那么无法擦除所述区块。

[0006] 因此,为了更好地利用存储器的存储容量,常常合并或收集有效部分区块数据量,这通过以下方式完成:将数据复制到经擦除的区块中使得这些复制的数据所来自的区块接着可经擦除且其整个存储容量可再使用。以此方式,可回收存储器内不含有有效数据的空间,使得其可用于存储数据。还期望对数据进行复制以便按区块内数据扇区的逻辑地址的次序对其进行分组,因为这增加读取数据并将读取的数据传送到主机的速度。如果这种数

据复制的发生过于频繁,那么存储器系统的操作性能可能降级。在存储器的存储容量几乎与主机通过系统的逻辑地址空间可寻址的数据量没有差别的情况(典型情况)下,这尤其影响存储器系统的操作。在此情况下,在可执行主机编程命令之前可能需要进行数据合并或收集。因此编程时间增加。

[0007] 在连续几代存储器系统中,区块的大小逐渐增加,以便增加给定半导体面积中可存储的数据位的数目。存储 256 个数据扇区和更多数据扇区的区块变得常见。另外,常常将不同阵列或子阵列的两个、四个或更多区块逻辑上链接在一起成为元区块,以便增加数据编程和读取时的并行性程度。与这种大容量操作单位一起出现的是有效地对其进行操作挑战。

[0008] 因此,需要对存储在非易失性存储器中的数据的管理改进。还需要一种有效地回收不含有效数据但当前不可用于存储新的有效数据的存储器空间的系统。还需要一种以例如对主机数据进行编程的其它存储器操作具有极少不利影响或没有不利影响的方式实施回收操作的系统。

发明内容

[0009] 在存储器用完经擦除区块之前回收存储器阵列中的空间可避免在编程主机数据时可能超过时间限制的严重延迟的风险。在存储器填充有主机数据时,以确保在存储器变满之前存储器不会用完经擦除区块的方式回收存储器阵列中的空间。可根据调度来执行回收,所述调度在存在经擦除区块的短缺之前较长时间开始回收。回收操作根据交错比率与主机数据的写入交错,使得回收操作散布在延长的周期中而不是在较长连续突发中完成。将交错比率计算为所有剩余主机写入与所有剩余回收写入的比率。

[0010] 在基于文件的主机接口将存储器连接到主机的情况下,存储器控制器可具有关于存储在存储器阵列中的数据的准确且最新的信息,并可从可在存储器变满之前写入到存储器的额外主机数据量以及在存储器变满之前需要的回收量中估计适当的交错比率。通过将回收操作均匀散布在整个剩余时间中,实现恒定的主机数据编程速率。在具有基于扇区或其它接口的存储器中,主机可提供信息以允许进行所述估计,使得可类似地计算交错比率。

[0011] 可在间隔处或当存在触发事件(例如,主机对一些存储数据的删除)时计算交错比率。因此,在适当时更新交错比率,使得所述比率适应于变化的环境。

[0012] 可在不同模式中不同地管理回收。除了基于对将写入的主机数据和将完成的回收操作的估计的自适应回收以外,可存在最小回收模式,其中以某个低(或零)速率回收空间。通常,最小回收模式适用于与可回收空间相比存在足够的经擦除区块的情况。还可存在最大回收模式,其中以某个最大速率以交错方式回收空间。通常,最大回收模式适用于存在很少经擦除区块的情况(当存储器近似为满时通常发生的情形)。另外,可响应于主机命令以连续方式(非交错)完成回收。也可响应于主机命令禁止回收。

附图说明

[0013] 图 1 示意性说明目前实施的主机和连接的非易失性存储器系统;

[0014] 图 2 是用作图 1 的非易失性存储器的示范性快闪存储器系统的框图;

[0015] 图 3 是可在图 2 的系统中使用的存储器单元阵列的代表电路图;

- [0016] 图 4 说明图 2 的系统的示范性物理存储器组织；
- [0017] 图 5 展示图 4 的物理存储器的一部分的扩展图；
- [0018] 图 6 展示图 4 和 5 的物理存储器的一部分的进一步扩展图；
- [0019] 图 7 说明主机与可重新编程的存储器系统之间的常见逻辑地址接口；
- [0020] 图 8A 和 8B 展示非易失性存储器中的垃圾收集的实例。
- [0021] 图 9A 和 9B 展示非易失性存储器中的垃圾收集的另一实例。
- [0022] 图 10A 和 10B 展示非易失性存储器中的压缩的实例。
- [0023] 图 11A 和 11B 展示非易失性存储器中的合并的实例。
- [0024] 图 12 说明主机应用程序对文件的删除，其中文件存储在使用图 7 的接口的非易失性存储器中。
- [0025] 图 13 展示可如何管理非易失性存储器中的空间的实例。
- [0026] 图 14 展示可如何管理非易失性存储器中的空间的另一实例。
- [0027] 图 15 展示交错的主机写入操作与垃圾收集操作。
- [0028] 图 16 展示具有到主机的基于文件的接口的非易失性存储器。
- [0029] 图 17 说明主机应用程序对文件的删除，其中文件存储在使用图 16 的接口的非易失性存储器中；
- [0030] 图 18 说明具有基于文件的接口的非易失性存储器中的空间管理的实例。
- [0031] 图 19 说明具有基于文件的接口的非易失性存储器中的空间管理的另一实例，此实例提供恒定的主机数据编程速率，直到存储器阵列为满为止。
- [0032] 图 20 展示具有基于文件的接口的非易失性存储器中的空间管理的另一实例，其具有适合响应于文件删除的自适应回收调度。
- [0033] 图 21 展示在三种模式中具有基于文件的接口的非易失性存储器中的空间管理的另一实例，所述三种模式为具有最小回收速率的初始模式、具有自适应调度回收速率的自适应模式以及具有最大回收速率的最终模式。
- [0034] 图 22 展示具有基于文件的接口的非易失性存储器中的空间管理的详细视图，其包含存储器阵列的处于各种条件的部分以及那些条件之间数据的可能转变。
- [0035] 图 23A-23D 展示存储器阵列的含有经历对应于图 22 那些转变的转变的数据的区块。

具体实施方式

[0036] 相对于图 1 到图 8 描述当前快闪存储器系统和与主机装置的典型操作。在此系统中可实施本发明的各个方面。图 1 的主机系统 1 将数据存储到快闪存储器 2 中并从其中检索数据。尽管快闪存储器可嵌入在主机内，但将存储器 2 说明为较普遍形式的卡，其通过机械和电气连接器的配合部分 3 和 4 可移除地连接到主机。目前存在许多不同的市售的快闪存储器卡，实例是紧致快闪存储器 (CF)、多媒体卡 (MMC)、安全数字 (SD)、小型 SD、记忆棒、智能媒体和 TransFlash 卡。尽管这些卡中的每一者均具有根据其标准化规范的独特的机械和 / 或电气接口，但每一者中包含的快闪存储器是非常相似的。这些卡可从本申请案的受让人 SanDisk 公司购得。SanDisk 还提供以 Cruzer 为商标的快闪驱动器线，其为小封装的手持存储器系统，所述系统具有用于通过插入主机的通用串行总线 (USB) 插座而与主机

连接的 USB 插头。这些存储器卡和快闪驱动器中的每一者均包含与主机介接并控制其内快闪存储器的操作的控制器。

[0037] 使用这些存储器卡和快闪驱动器的主机系统有许多且各式各样。它们包含个人计算机 (PC)、膝上型和其它便携式计算机、蜂窝式电话、个人数字助理 (PDA)、数字静态相机、数字电影相机和便携式音频播放器。主机通常包含用于一种或一种以上类型的存储器卡或快闪驱动器的内置插座,但其中一些需要能插入存储器卡的适配器。

[0038] 就存储器 2 涉及到的范围而言,图 1 的主机系统 1 可被视为具有两个主要部分,由电路和软件的组合构成。它们是应用程序部分 5 以及存储器 2 介接的驱动程序部分 6。在例如个人计算机中,应用程序部分 5 可包含运行字处理、图形、控制或其它流行的应用程序软件的处理器。在相机、蜂窝式电话或其它主要专用于执行单一功能集合的主机系统中,应用程序部分 5 包含操作相机以拍摄和存储照片、操作蜂窝式电话以打电话和接电话等等的软件。

[0039] 图 1 的存储器系统 2 包含快闪存储器 7 和电路 8,电路 8 和与卡连接以用于来回传递数据的主机介接,且还控制存储器 7。控制器 8 通常在数据编程和读取期间在主机 1 使用的数据的逻辑地址与存储器 7 的物理地址之间进行转换。

[0040] 参看图 2,其描述可用作图 1 的非易失性存储器 2 的典型快闪存储器系统的电路。系统控制器通常实施在单个集成电路芯片 11 上,其通过系统总线 13 与一个或一个以上集成电路存储器芯片并联连接,图 2 中展示单个这种存储器芯片 15。所说明的特定总线 13 包含用于携带数据的单独一组导体 17、用于存储器地址的一组导体 19 以及用于控制和状态信号的一组导体 21。或者,单个一组导体可在这三个功能之间时间共享。此外,可采用系统总线的其它配置,例如 2004 年 8 月 9 日申请的标题为“Ring Bus Structure and Its Use in Flash Memory Systems”的第 10/915,039 号美国专利申请案中描述的环形总线。

[0041] 典型的控制器芯片 11 具有其自身的内部总线 23,内部总线 23 通过接口电路 25 与系统总线 13 介接。通常连接到总线的主要功能件是处理器 27(例如微处理器或微控制器)、含有用于初始化(“引导”)系统的代码的只读存储器 (ROM) 29、主要用于对存储器与主机之间传送的数据进行缓冲的只读存储器 (RAM) 31,以及对用于传递通过存储器与主机之间的控制器的数据的错误校正码 (ECC) 进行计算和检查的电路 33。控制器总线 23 通过电路 35 与主机系统介接,在图 2 的系统包含在存储器卡内的情况下,所述介接是通过卡的作为连接器 4 的一部分的外部接点 37 来完成的。时钟 39 与控制器 11 的每一其它组件连接并由其使用。

[0042] 存储器芯片 15 以及任何其它与系统总线 13 连接的芯片通常含有组织成多个子阵列或平面的存储器单元阵列,为了简明起见而说明两个这种平面 41 和 43,但可替代使用更多平面,例如四个或八个这种平面。或者,可不将芯片 15 的存储器单元阵列划分为平面。然而当如此划分时,每个平面具有其自身的可彼此独立地操作的列控制电路 45 和 47。电路 45 和 47 从系统总线 13 的地址部分 19 接收其各自存储器单元阵列的地址,并对其进行解码以对各自位线 49 和 51 中的特定一者或一者以上进行寻址。响应于在地址总线 19 上接收的地址通过行控制电路 55 来寻址字线 53。源电压控制电路 57 和 59 也与各自的平面连接,如同 p 阱电压控制电路 61 和 63。如果存储器芯片 15 具有单个存储器单元阵列,且如果系统中存在两个或两个以上这种芯片,那么可类似于上述多平面芯片内的平面或子阵列来操

作每个芯片的阵列。

[0043] 数据通过与系统总线 13 的数据部分 17 连接的各自数据输入 / 输出电路 65 和 67 而传送进和传送出平面 41 和 43。电路 65 和 67 用于通过经由各自的列控制电路 45 和 47 连接到平面的线 69 和 71, 来将数据编程到存储器单元中以及从其各自平面的存储器单元读取数据。

[0044] 尽管控制器 11 控制存储器芯片 15 的操作来编程数据、读取数据、擦除和照管各种内存管理事件, 但每个存储器芯片还含有某种控制电路, 其执行来自控制器 11 的命令以执行这些功能。接口电路 73 连接到系统总线 13 的控制和状态部分 21。将来自控制器的命令提供到状态机 75, 状态机 75 接着提供对其它电路的特定控制以便执行这些命令。控制线 77-81 连接状态机 75 与所述其它电路, 如图 2 所示。来自状态机 75 的状态信息通过线 83 传送到接口 73, 以用于通过总线部分 21 传输到控制器 11。

[0045] 存储器单元阵列 41 和 43 的 NAND 结构是目前优选的, 但也可替代使用例如 NOR 的其它结构。作为存储器系统的一部分的 NAND 快闪存储器及其操作的实例可参考第 5, 570, 315、5, 774, 397、6, 046, 935、6, 373, 746、6, 456, 528、6, 522, 580、6, 771, 536 和 6, 781, 877 号美国专利以及第 2003/0147278 号美国专利申请公开案。

[0046] 由图 3 的电路图说明示范性 NAND 阵列, 其为图 2 的存储器系统的存储器单元阵列 41 的一部分。提供大量全局位线, 但为了简明解释起见, 在图 2 中仅展示四个此类线 91-94。许多串联连接的存储器单元串 97-104 连接在这些位线之一与参考电位之间。使用存储器单元串 99 作为代表, 多个电荷存储存储器单元 107-110 在串的任一端处与选择晶体管 111 和 112 串联连接。当使一个串的选择晶体管导电时, 所述串连接在其位线与参考电位之间。接着对所述串内的一个存储器单元进行一次编程或读取。

[0047] 图 3 的字线 115-118 在许多存储器单元串中每一串中的一个存储器单元的电荷存储元件上个别地延伸, 且栅极 119 和 120 控制串的每一端处的选择晶体管的状态。使共享共用字线和控制栅极线 115-120 的存储器单元串形成可共同擦除的存储器单元区块 123。此单元区块含有可一次性物理擦除的最小数目的单元。一次对沿着字线 115-118 之一的一行存储器单元进行编程。通常, 以预定次序对 NAND 阵列的行进行编程, 在此情况下是以最靠近连接到接地或另一共用电位的串的末端的字线 118 的行开始。接着对沿着字线 117 的存储器单元行进行编程, 如此在整个区块 123 中进行。最后对沿着字线 115 的行进行编程。

[0048] 第二区块 125 是类似的, 其存储器单元串连接到与第一区块 123 中的串所连接的相同全局位线, 但具有不同的一组字线和控制栅极线。由行控制电路 55 将字线和控制栅极线驱动到其适当的操作电压。如果系统中存在一个以上平面或子阵列, 例如图 2 的平面 1 和 2, 那么一种存储器结构使用在其之间延伸的共用字线。替代地可存在共享共用字线的两个以上平面或子阵列。在其它存储器结构中, 单独驱动个别平面或子阵列的字线。

[0049] 如若干 NAND 专利以及上文引用的公开申请案中所述, 可操作存储器系统以在每个电荷存储元件或区域中存储两个以上可检测的电荷电平, 进而在每一者中存储一个以上数据位。存储器单元的电荷存储元件最通常是导电浮动栅极, 但替代地可以是非导电介电电荷捕集材料, 如第 2003/0109093 号美国专利申请公开案中所述。

[0050] 图 4 概念性说明用作下文进一步描述中的实例的快闪存储器单元阵列 7 (图 1) 的组织。存储器单元的四个平面或子阵列 131-134 可在单个集成存储器单元芯片上、在两个

芯片上（每个芯片上两个平面）或在四个单独芯片上。具体布置对下文的论述来说不重要。当然，在系统中可存在其它数目的平面，例如 1、2、8、16 或更多。将平面个别地划分为图 4 中用矩形展示的存储器单元区块，例如位于各自平面 131-134 中的区块 137、138、139 和 140。每个平面中可存在数十个或数百个区块。如上所述，存储器单元区块是擦除的单位，是物理上可共同擦除的最小数目的存储器单元。然而为了增加的并行性，以较大的元区块为单位来操作区块。将来自每个平面的一个区块逻辑上链接在一起以形成元区块。展示四个区块 137-140 形成一个元区块 141。元区块内的所有单元通常共同擦除。用于形成元区块的区块无需限于其各自平面内的相同的相对位置，如由区块 145-148 组成的第二元区块 143 中所示。尽管优选在所有平面上延伸元区块以获得高系统性能，但可以在不同平面中动态形成具有一个、两个或三个区块中的任一种或全部的元区块的能力来操作存储器系统。这允许元区块的大小较接近地与一次编程操作中可用于存储的数据量匹配。

[0051] 为了操作目的，又将个别区块划分为存储器单元页，如图 5 说明。例如区块 131-134 中每一者的存储器单元各划分为八个页 P0-P7。或者，每个区块内可存在 16 个、32 个或更多存储器单元页。页是在区块内数据编程和读取的单位，含有一次编程的最小数据量。在图 3 的 NAND 结构中，页由沿着区块内字线的存储器单元形成。然而为了增加存储器系统操作并行性，可将两个或两个以上区块内的所述页逻辑上链接成元页。图 5 中说明元页 151，其由来自四个区块 131-134 中每一者的一个物理页形成。元页 151 例如包含所述四个区块中每一者中的页 P2，但元页的页无需在区块的每一者内一定具有相同的相对位置。尽管优选在所有四个平面上并行编程和读取最大数据量以获得高系统性能，但也可操作存储器系统以在不同平面中的单独区块中形成具有一个、两个或三个页中的任一种或全部的元页。这允许编程和读取操作适应性地与可方便地并行处理的数据量匹配，且减少了元页的一部分保持未以数据编程的机会。

[0052] 如图 5 所说明，由多个平面的物理页形成的元页含有沿着那些多个平面的字线行的存储器单元。并非同时对一个字线行中的所有单元编程，而是较常见以两个或两个以上交错群组对其交替编程，每个群组存储一个数据页（在单个区块中）或一个数据元页（在多个区块上）。通过每次对交替的存储器单元进行编程，无需对每个位线均提供包含数据寄存器和读出放大器的外围电路单位，而是在邻近的位线之间时间共享。这节省了外围电路所需的衬底空间量，且允许以沿着行而增加的密度来封装存储器单元。另外，优选对沿着行的每个单元同时编程，以便最大化可从给定存储器系统得到的并行性。出于多数数据管理目的，可以与区块和页相同的方式处理元区块和元页。本申请案中在元区块和元页方面给出的实例通常也适用于使用区块和页作为擦除单位并分别编程的存储器。类似地，在区块和页方面给出的实例通常也适用于使用元区块和元页的存储器。

[0053] 有效地控制具有非常大的擦除区块的存储器阵列的操作的一个挑战是使在给定写入操作期间存储的数据扇区数目与存储器区块的容量和边界匹配并对准。一种方法是配置元区块，所述元区块用于存储来自主机的具有少于最大数目区块的新数据，在必要时用于存储少于填充整个元区块的量的数据量。自适应元区块的使用参见 2003 年 12 月 30 日申请的标题为“Adaptive Metablocks”的第 10/749,189 号美国专利申请案。数据区块之间的边界与元区块之间的物理边界的配合描述于 2004 年 5 月 7 日申请的第 10/841,118 号美国专利申请案以及 2004 年 12 月 16 日申请的标题为“Data Run Programming”的第

11/016, 271 号美国专利申请案。

[0054] 参看图 3, 通过沿着 NAND 串的至少一端提供两行选择晶体管 (未图示) 而不是所示的单个行, 来最方便地完成将数据同时编程到沿着一行的每隔一个存储器单元中。一行的选择晶体管接着响应于一个控制信号将区块内的每隔一个串连接到其各自的位线, 且另一行的选择晶体管响应于另一控制信号将介入的每隔一个串连接到其各自的位线。因此将两个数据页写入到每行存储器单元中。

[0055] 每个逻辑页中的数据量通常是整数个的一个或一个以上数据扇区, 每个扇区惯例上含有 512 个字节的数据。图 6 展示页或元页数据的两个扇区 153 和 155 的逻辑数据页。每个扇区通常含有所存储的用户或系统数据的 512 个字节的部分 157 以及用于涉及部分 157 中的数据或涉及其中存储有数据的物理页或区块的开销数据的另一数目的字节 159。开销数据的字节数通常为 16 个字节, 从而对于扇区 153 和 155 中每一者为总共 528 个字节。开销部分 159 可含有在编程期间从数据部分 157 计算出的 ECC、其逻辑地址、区块已擦除和再编程次数的经历计数、一个或一个以上控制旗标、操作电压电平和 / 或类似物, 加上从此开销数据 159 计算出的 ECC。或者, 开销数据 159 或其一部分可存储在其它区块中的不同页中。

[0056] 随着存储器的并行性增加, 元区块的数据存储容量增加且数据页和元页的大小也因此增加。数据页接着可含有两个以上数据扇区。由于数据页中的两个扇区且每个元页两个数据页, 所以一个元页中存在四个扇区。每个元页因此存储 2048 个字节的数据。这是高度的并行性, 且可甚至在行中存储器单元数目增加时进一步增加。出于此原因, 延伸快闪存储器的宽度以便增加页和元页中的数据量。上文指出的物理上较小的可再编程非易失性存储器卡和快闪驱动器是市售的, 其具有 512 兆字节 (MB)、千兆字节 (GB)、2GB 和 4GB 的数据存储容量, 且可能变得更高。

[0057] 图 7 说明主机与此大容量存储器系统之间的常见的接口。主机处理由主机执行的应用程序软件或固件程序产生或使用的数据文件。字处理数据文件是一个实例, 且计算机辅助设计 (CAD) 软件的绘图文件是另一实例, 主要存在于例如 PC、膝上型计算机等一般计算机主机中。pdf 格式的文档也是这种文件。图片数字摄像机为存储在存储器卡上的每幅图产生一个数据文件。蜂窝式电话利用来自内部存储器卡上的文件的数据, 例如电话簿。PDA 存储并使用若干不同的文件, 例如地址文件、日历文件等。在任何这种应用中, 存储器卡还可含有操作主机的软件。

[0058] 在图 7 中, 连续的逻辑地址空间 161 足够大以便为可存储在存储器系统中的所有数据提供地址。通常, 逻辑地址空间稍微小于存储器阵列的物理地址空间, 使得存储器阵列中存在一些额外空间。通常将主机逻辑地址空间分成数据群集的增量。每个群集可在给定主机系统中设计成含有若干数据扇区, 大约在 4 与 64 个扇区是典型的。标准扇区含有 512 个字节的数据。

[0059] 图 7 的实例中展示三个文件 1、2 和 3。在主机系统上运行的应用程序产生每个文件作为一组有序数据, 并通过唯一名称或其它参考对其进行识别。尚未分配给其它文件的足够可用的逻辑地址空间由主机指派给文件 1。文件 1 被展示为已经被指派有连续的可用逻辑地址范围。通常还出于特定目的而分配地址范围, 例如针对主机操作软件的特定范围, 接着为存储数据而避开这些特定范围, 即使在主机向数据指派逻辑地址时仍未利用这些地

址。

[0060] 当主机稍后产生文件 2 时,主机类似地在逻辑地址空间 161 内指派两个不同范围的连续地址,如图 7 中所示。文件无需被指派有连续的逻辑地址,而是可为已经分配给其它文件的地址范围之间的地址片段。此实例接着展示主机所产生的又一文件 3 被分配有主机地址空间的先前未分配给文件 1 和 2 和其它数据的其它部分。

[0061] 主机通过维持文件分配表 (FAT) 来跟踪存储器逻辑地址空间,其中由主机指派给各个主机文件的逻辑地址得以维持。FAT 表通常存储在非易失性存储器中以及主机存储器中,且随着新文件被存储、其它文件被删除、文件被修改等等,FAT 表由主机频繁地更新。例如当主机文件被删除时,主机接着通过更新 FAT 表来对先前分配给被删除文件的逻辑地址解除分配,以展示它们现在可用于与其它数据文件一起使用。

[0062] 主机不关心其中存储器系统控制器选择用来存储文件的物理位置。典型的主机仅知道其逻辑地址空间和其已经分配给其各个文件的逻辑地址。另一方面,存储器系统通过典型的主机 / 卡接口,仅知道逻辑地址空间的数据已经被写入到的部分,但不知道分配给特定主机文件的逻辑地址,或甚至不知道主机文件的数目。存储器系统控制器将主机所提供的用于存储或检索数据的逻辑地址转换成其中存储有主机数据的快闪存储器单元阵列内的唯一物理地址。区块 163 表示这些逻辑到物理地址转换的工作表,其由存储器系统控制器维持。

[0063] 存储器系统控制器经编程以便以使系统的性能维持在高等级的方式将数据文件存储在存储器阵列 165 的区块和元区块内。在此说明中使用四个平面或子阵列。在由来自每个平面的区块形成的整个元区块上优选以系统允许的最大程度的并行性编程和读取数据。至少一个元区块 167 通常经分配作为用于存储由存储器控制器使用的操作固件和数据的保留区块。可分配另一元区块 169 或多个元区块以用于存储主机操作软件、主机 FAT 表等。保留物理存储空间的大部分以用于存储数据文件。然而存储器控制器不知道主机如何在其各个文件对象之间分配所接收的数据。存储器控制器从与主机交互作用中通常知道的全部内容就是由主机写入到特定逻辑地址的数据存储在由控制器的逻辑到物理地址表 163 维持的相应物理地址中。

[0064] 在典型的存储器系统中,提供比必需的多几个的额外区块的存储容量,以存储地址空间 161 内的数据的量。可提供这些额外区块中的一者或一者以上作为用于代替在存储器使用寿命期间可能变为有缺陷的其它区块的冗余区块。个别元区块内含有的区块的逻辑分组通常可因为各种原因而改变,包含冗余区块对原始指派给元区块的有缺陷区块的代替。例如元区块 171 的一个或一个以上额外区块通常维持在经擦除的区块池 (blockpool) 中。当主机将数据写入到存储器系统时,控制器将由主机指派的逻辑地址转换成擦除区块池中的元区块内的物理地址。未用于将数据存储在逻辑地址空间 161 内的其它元区块接着经擦除并指定作为经擦除的池区块以供在后续数据写入操作期间使用。

[0065] 当原始存储的数据变得过时,存储在特定主机逻辑地址处的数据由新数据频繁地替换。作为响应,存储器系统控制器将新数据写入经擦除的区块,并接着针对那些逻辑地址改变逻辑到物理地址表以识别存储这那些逻辑地址处数据的新的物理区块。接着擦除含有那些逻辑地址处原始数据的区块并使其可用于新数据的存储。如果在写入开始时在来自擦除区块池的预擦除区块中没有足够的存储容量,那么这种擦除常常必须在当前数据写入

操作可完成之前发生。这可能不利地影响系统数据编程速度。存储器控制器通常了解到仅主机向其相同逻辑地址写入新数据时由主机使给定逻辑地址处的数据变过时。因此许多存储器区块可能存储此类无效数据历时一定时间。

[0066] 区块和元区块的大小逐渐增加,以便有效地使用集成电路存储器芯片的面积。这导致较大比例的个别数据写入存储的数据量小于元区块的存储容量,且在许多情况下甚至小于区块的存储容量。由于存储器系统控制器通常将新数据引导到来自经擦除区块池的元区块,所以这可能导致元区块的多个部分变得未被填充。如果新数据是对存储在另一元区块中的某个数据的更新,那么还期望以逻辑地址次序将其余有效的数据元页从具有与新数据元页的逻辑地址连续的逻辑地址的所述另一元区块复制到新的元区块中。老的元区块可保持其它有效的数据元页。这随着时间过去而导致个别元区块的某些元页的数据变为过时且无效,并由写入到不同元区块的具有相同逻辑地址的新数据所替换。

[0067] 为了维持足够的物理存储器空间以在整个逻辑地址空间 161 上存储数据,可通过在垃圾收集操作中回收存储器的由过时数据占据的部分。还可通过其中经擦除空间在含有有效数据的区块中的合并来回收经擦除的空间,且将有效数据合并到较少的区块中允许将区块添加到经擦除的区块池。因此,区块经受垃圾收集或合并以回收存储器空间供再使用。尽可能地将数据扇区以与其逻辑地址相同的次序维持在元区块内也是合乎要求的,因为这使得以连续逻辑地址读取数据更高效。因此,通常为了此额外目标而执行数据合并和垃圾收集。第 6,763,424 号美国专利中描述当接收部分区块数据更新时管理存储器的一些方面以及元区块的使用。垃圾收集与合并在本申请案中统称为“回收”,且作为垃圾收集或合并的一部分执行的操作称为“回收操作”。

[0068] 回收实例

[0069] 在垃圾收集期间,从含有过时数据的一个或一个以上源区块搜集具有连续或几乎连续的逻辑地址范围的有效数据的页,并将其重写到目的地区块中。目的地区块可来自经擦除的区块池,或可含有一些有效数据。当已从所述一个或一个以上源区块复制所有有效数据页时,其可经擦除以供将来使用。图 8A 和 8B 展示示范性垃圾收集操作。图 8A 展示来自区块 1 的页 0-2 的有效数据 X、X+1 和 X+2 从区块 1 复制到区块 3,且有效数据 X+3 从区块 2 的页 1 复制到区块 3。在数据 X、X+1、X+2 和 Y 复制到区块 3 之后,区块 1 和 2 可添加到准备好用于立即擦除且通常稍后短时间内擦除的区块池。图 8B 展示在垃圾收集之后的情形,其中区块 1 和 2 经擦除且区块 3 填充有数据。由于此垃圾收集操作,经擦除的区块池增加了一个区块,因为区块 1 和 2 添加到经擦除区块池中,但区块 3 不再在经擦除区块池中。为了完成此情况,复制数据 X、X+1、X+2 和 X+3。此实例的数据单位等于一页的内容,且可含有一个或一个以上数据扇区。或者,数据可不在扇区逻辑单位中,使得页的内容可能不具有均匀大小的单独可寻址单位。类似地,回收操作的其它实例可应用于在逻辑可寻址的扇区单位中或某种其它格式的数据。在此实例中,复制数据 X、X+1、X+2 和 X+3,使得其依次存储在区块 3 中。在其它实例中,可将数据复制到其未依次存储的区块。

[0070] 图 9A 和 9B 展示垃圾收集的另一实例。这里有效数据 Y 是从在页 0 和 2 中含有过时数据的区块 2 复制,并复制到仅含有效数据的区块 1。在复制数据 Y 之后,仅过时数据保留在区块 2 中,因此区块 2 被擦除。图 9B 展示垃圾收集之后的情形,其中数据 Y 存储在区块 1 的先前擦除页 3 中且区块 2 经擦除。因此,在此实例中,不需要来自经擦除区块池的区

块。在数据 Y 为填充存储器阵列的一个页的数据量的情况下,在此操作中仅复制一个页和擦除一个区块,从而导致额外的擦除区块添加到经擦除区块池。在此实例中,数据 Y 与数据 X、X+1、X+2 逻辑上不相关。在其它实例中,数据可复制到含有逻辑相关的数据的区块。

[0071] 数据压缩是特定形式的垃圾收集,其通常涉及从区块读取所有有效数据页并将其写入新的区块,忽略在过程中具有无效数据的页。具有有效数据的页还优选以与存储在其中的数据的逻辑地址次序匹配的物理地址次序布置。可对具有以非顺序(无序)格式存储的数据的区块执行数据压缩,使得在压缩之后数据以顺序格式存储。新的区块中被占据的页的数目将小于旧的区块中被占据的那些页的数目,因为含有过时数据的页没有被复制到新的区块。接着擦除旧的区块,并使其可用于存储新的数据。通过合并获得的额外页容量可接着用于存储其它数据。

[0072] 图 10A 和 10B 展示数据压缩的实例。图 10A 展示填充有数据的区块 1,其中一些数据是过时的且一些数据是有效的。存储在区块 1 中的数据没有顺序次序。区块 1 通常是用于某些存储器设计的无序更新区块,所述存储器设计例如为第 10/750,155 号美国专利申请案中描述的那些设计。区块 1 的页 2 和 3 含有数据 Z+1 和 Z+2 的有效副本,而页 0 和 1 含有这些数据的过时副本。在无序更新区块映射到有限的逻辑地址范围的情况下,只要区块变满就对区块进行压缩,使得额外的更新在区块的逻辑地址范围内是可能的。图 10B 展示压缩之后情形。区块 1 的页 2 和 3 的有效数据 Z+1 和 Z+2 已复制到区块 2 且经布置以按顺序存储。按顺序存储数据的一个优点在于,可能不必维持不同扇区的位置的索引,因此减少与维持此索引相关联的开销。区块 2 含有页 2 和 3 中可用于存储额外数据的经擦除空间。在所有有效数据已复制到区块 2 且区块 1 已擦除之后展示区块 1。因此,由于压缩,经擦除区块池仍具有相同数目的经擦除区块,但存在先前不可用的可用于在区块 2 中写入数据的空间。复制来自两个页的数据以实现此压缩。

[0073] 数据压缩可用于形成可用于存储数据的空间。在一些存储器系统中,存储器中的经擦除空间可能不是全部可使用,因为其处于分布在还含有有效数据的区块之间的较小部分中。当接收到与存储在具有足够空间对数据进行编程的任何区块中的数据逻辑上不相关的新数据时,需要将其编程到经擦除区块而不是部分充满的区块。这种与已经存储的数据逻辑上不相关的新数据通常存储在来自经擦除区块池的经擦除区块中。在某个时间之后,可能存在具有不可用于逻辑上不相关的新数据的经擦除空间的多个区块。这导致空间浪费。可组合所述部分写入的区块的有效数据。举例来说,来自含有经擦除空间的两个区块的有效数据可经组合以使得组合的经擦除空间形成经擦除区块。

[0074] 图 11A 展示区块 1 在页 0-2 中含有数据 X、X+1 和 X+2,而区块 2 在页 0 中含有数据 Y。数据 Y 与数据 X、X+1 和 X+2 逻辑上不相关。区块 1 和 2 可保持在具有经擦除空间的条件下历时某个时间以查看是否接收到接续到区块 1 的数据 X+2 或接续到区块 2 的数据 Y 的额外数据。如果在某个阈值时间之后或在满足某个其它条件之后没有接收到额外的连续数据,那么可标记区块 1 和 2 以进行合并。可对准备好合并的区块维持一个列表,且可根据区块含有的有效数据量从列表中选择区块,使得当组合时其填充一个区块或近似于填充一个区块。图 11B 展示合并之后的区块 1 和 2。数据 Y 已复制到区块 1,且区块 2 已擦除。此合并仅需要从一个页复制数据(来自区块 2 的页 0 的数据 Y)以便向经擦除区块池添加一个区块。通常需要以需要较少复制的方式进行合并,因此将数据 Y 复制到区块 1 而不是将

数据 X、X+1 和 X+2 复制到区块 2。

[0075] 数据合并和垃圾收集花费时间且可影响存储器系统的性能,尤其是在数据合并或垃圾收集需要在来自主机的命令可执行之前进行的情况下。所述回收操作通常由存储器系统控制器调度以尽可能在后台进行,但这并不总是可能的。主机命令的执行可延迟的实例是擦除区块池中没有足够的预擦除元区块用来存储主机想要写入到存储器中的所有数据,且首先需要数据合并或垃圾收集来清除一个或一个以上有效数据的元区块,所述元区块接着可被擦除。因此,注意力已经被引导到管理对存储器的控制,以便使此类中断减到最小。在以下美国专利申请案中描述许多此类技术:2003 年 12 月 30 日申请的标题为“Management of Non-Volatile Memory Systems Having Large Erase Blocks”的第 10/749,831 号美国专利申请案;2003 年 12 月 30 日申请的标题为“Non-Volatile Memory and Method with Block Management System”的第 10/750,155 号美国专利申请案;2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Memory Planes Alignment”的第 10/917,888 号美国专利申请案;2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Phased Program Failure Handling”的第 10/917,889 号美国专利申请案;以及 2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Control Data Management”的第 10/917,725 号美国专利申请案。

[0076] 存储器控制器还可使用来自 FAT 表(其由主机存储在非易失性存储器中)的数据,以更有效地操作存储器系统。一种此类使用是通过对其逻辑地址解除分配来了解数据何时已经由主机识别为过时的。知道此情况将允许存储器控制器在其会正常地通过主机将新数据写入到那些逻辑地址而了解它之前调度含有此类无效数据的区块的擦除。2004 年 7 月 21 日申请的标题为“Method and Apparatus for Maintaining Data on Non-Volatile Memory Systems”的第 10/897,049 号美国专利申请案中描述这种情况。其它技术包含监视将新数据写入到存储器的主机模式,以便推断给定写入操作是单个文件还是(如果有多个文件)文件之间的边界所在之处。2004 年 12 月 23 日申请的标题为“FAT Analysis for Optimized Sequential Cluster Management”的第 11/022,369 号美国专利申请案描述这种类型的技术的使用。

[0077] 为了有效地操作存储器系统,对控制器来说合乎需要的是尽可能多地知道由主机指派给其个别文件的数据的逻辑地址。数据文件接着可由控制器存储在单个元区块或元区块群组内,而不是当不知道文件边界时分散在较大数目的元区块之间。结果是数据合并和垃圾收集操作的数目和复杂性减小。因此,存储器系统的性能改进。但如上文所述,当主机/存储器接口包含逻辑地址空间 161(图 7)时,存储器控制器难以知道关于主机数据文件结构的较多信息。

[0078] 用基于逻辑地址的接口进行回收

[0079] 图 12 展示当删除文件时图 7 的主机/存储器接口的操作。在主机系统上运行的应用程序 201 确定应在存储器中删除文件 2。举例来说,在 PC 上运行的应用程序 201 可基于用户输入或出于其它原因确定不再需要文件 2。因此,应用程序向主机的驱动程序部分发送删除文件 2 的指令。在此情况下,使用 FAT 管理对数据的存储以指示用于存储在存储器中的每个文件的逻辑地址。举例来说,文件 2 展示为具有两个单独的逻辑地址范围 203 和

205。每个逻辑地址范围 203 和 205 可包含多个连续的群集。尽管文件 2 展示为仅具有两个单独的逻辑地址范围 203 和 205,但可将文件频繁地分段为许多逻辑地址范围,其中其它文件映射到介入的逻辑地址。删除文件 2 的指令导致 FAT 被修改,使得分配给文件 2 的逻辑地址范围 203 和 205 的群集被解除分配并变为可用于主机的后续分配。然而,主机的 FAT 中这些群集的解除分配通常不引起对存储器控制器使用的存储器管理结构的任何修改。逻辑到物理地址转译 163 包含由存储器控制器维持的记录存储数据群集的物理地址的表。地址范围 203 和 205 被分别映射到物理地址范围 209 和 211。此记录不会由于来自应用程序的“删除文件 2”命令而改变。因此,表维持用于逻辑地址范围 203 和 205 的条目。而且,相应的物理位置 209 和 207 保持填充有数据,即使应用程序不再需要此数据。通常,存储器控制器仅在主机以这些地址发送新数据时才改变用于逻辑地址范围 203 和 205 的逻辑到物理地址转译。因为存储器控制器不知道何时群集被解除分配,所以其通常每个逻辑地址维持至少一个条目,且因此将整个逻辑地址范围视为已被占据。通常存在比存储器的逻辑地址范围多的物理空间,使得即使整个逻辑地址范围填充有明显有效的数据,也存在可为经擦除空间或可由过时数据占据的额外空间。

[0080] 图 13 展示在将主机数据写入存储器时可如何管理存储器阵列中的物理空间的实例。在具有对应于整个逻辑地址范围的有效数据的情况下,将物理空间视为最满。存储器中的剩余空间由经擦除空间和由过时数据占据的空间组成。在将主机数据编程到存储器阵列时,由过时数据占据的空间量如图示增加。这是因为当接收到具有特定逻辑地址的新扇区时,其存储在在逻辑到物理转译 163 中记录的新的物理位置处。这替换了先前存储的具有相同逻辑地址的扇区。先前存储的扇区接着变得过时,且将先前存储扇区的物理位置记录为含有过时数据。经擦除空间随着过时数据量增加而减小。在某个点时,没有足够的经擦除空间来继续对主机数据编程。图 13 展示在时间 t_1 处在存储器中没有经擦除空间。因此在时间 t_1 处,没有另外的主机数据可编程到存储器阵列。在其它实例中,当某个最少量的经擦除空间保留在存储器阵列中时,编程可停止。在时间 t_1 处,垃圾收集操作开始以回收由过时数据占据的空间。当所有过时空间被回收并作为经擦除空间时在时间 t_2 处此操作结束。在其它实例中,垃圾收集操作可在已回收所有可能的空间之前结束,例如当已回收恰好足够的空间以允许继续主机数据的编程时。在时间 t_2 处,主机数据的编程再次开始,且在时间 t_3 处,主机数据的写入停止,且另一垃圾收集操作开始。图 13 所示的视图是存储器控制器的存储器中数据的条件的视图,且并不总是与主机的视图相同。在存储器控制器将逻辑空间视为充满时,主机可将同一空间视为较大程度上自由。

[0081] 以图 13 所示的方式管理存储器具有的缺点在于,在时间 t_1 与 t_2 之间,存储器对主机不可用。因此,在从 t_1 到 t_2 的时间段中不写入任何主机数据。由于可能必须复制的大量的页,从 t_1 到 t_2 的时间可具有相当大的长度。在一些情况下,此时间很大而使得其导致主机超时。也就是说,主机具有用于写入一部分数据的最大时间,且在存在不足够的擦除空间时允许写入新数据所必需的垃圾收集可能超过此最大值。

[0082] 图 14 展示管理存储器的替代方法,其中在仍存在足够的经擦除空间时执行垃圾收集以允许写入主机数据。在此实例中,在垃圾收集绝对必要之前实施垃圾收集。可以第 11/040,325 号美国专利申请案中描述的交错方式在主机数据的写入之间完成垃圾收集。用于开始此类交错垃圾收集的触发可以是经擦除区块的数目达到某个阈值。交错的垃圾收集

操作减慢了主机数据到存储器的编程。然而,垃圾收集操作可防止经擦除区块的数目减少到主机数据不再能写入的点。因此,主机超时或取消编程的风险减小或被消除。交错的垃圾收集可在足够的经擦除区块可用时停止,使得不会不必要地影响编程速度。因此,经擦除空间的量随着交错垃圾收集的打开和关闭或者交错垃圾收集的速率改变而变化。

[0083] 图 15 展示与主机数据的写入交错的垃圾收集的时序图。写入 N3 个主机数据页,接着写入 N4 个数据页或 x 个区块作为垃圾收集操作的一部分而擦除。此循环重复。以此方式,与垃圾收集相关联的开销随着时间散布,而不是集中于一个时间,集中于一个时间将导致超时。合并、其它回收操作或其它内务操作也可以此方式交错。在此上下文中内务操作指由存储器控制器起始以维持存储器阵列中的数据的操作。这些操作可包含损耗均衡 (wear leveling) 和数据清洗 (data scrub) 操作。

[0084] 用基于文件的接口进行回收

[0085] 通过一些存储器接口,额外的信息可用于存储器控制器以允许更有效的回收。这些存储器接口的实例在以下美国专利申请案中描述:标题为“Direct Data File Storage in Flash Memories”的第 11/060,249 号美国专利申请案;标题为“Direct File Data Programming and Deletion in Flash Memories”的第 11/060,174 号美国专利申请案;标题为“Direct Data File Storage Implementation Techniques in Flash Memories”的第 11/060,248 号美国专利申请案(以上全部在 2005 年 2 月 16 日申请);以及 2005 年 8 月 3 日申请的标题为“Direct Data File Storage in Flash Memories”的第 60/705,388 号临时专利申请案。文件从主机发送到存储器,而不映射到为存储器界定的逻辑地址范围的逻辑地址。此存储器可视为具有基于文件的接口。在存储器中,文件主要存储在专用于所述文件的元区块中。使用文件识别符和偏移量来记录文件存储的位置。图 16 展示使用这种直接数据文件存储的存储器接口,其中文件发送到存储器且文件 / 偏移量到物理地址转译 173 在存储器中进行。这些存储器可以减少分段的方式维持文件,使得多数元区块含有来自仅一个文件的数据。在所述存储器中,存储器控制器具有较多的关于所存储数据的可用信息。特定来说,因为存储器控制器通过文件识别数据,所以其能够以基于文件的方式存储文件。

[0086] 图 17 展示当应用程序删除存储在使用图 16 的接口的存储器系统中的文件时发生的事。应用程序 225 发送指示应删除文件 2 的命令。此命令从主机发送到存储器而没有任何通过逻辑扇区地址识别文件的需要。文件识别符可由主机用于识别文件。文件 2 映射到区块 a、b 和 c,且映射由文件 / 偏移量到物理转译 173 记录。命令致使文件 / 偏移量到物理转译 173 被更新以反映文件 2 被删除。因此,区块 a、b 和 c 经调度以用于垃圾收集,使得其可再用于新数据。因此,可通过由应用程序删除文件而不是等待替换数据存储在存储器阵列中为止(如使用逻辑扇区地址的存储器的情况)来起始垃圾收集。在此情况下存储器控制器并不总是看见充满的逻辑地址空间。替代地,存储器控制器具有关于存储在存储器中的数据的有效性的准确信息。在有些时候,存储器控制器可意识到存储器阵列含有极少有效数据或不含有有效数据。在其它时候,存储器控制器可将存储器阵列视为充满或几乎充满有效数据。通过关于存储器阵列中的有效和过时数据的所述信息,可以比之前更有效的方式管理回收。

[0087] 图 18 展示对使用直接数据文件接口的存储器操作的存储器回收系统的实例。图

18 展示在主机数据写入时存储器阵列随着时间变化的条件。所示视图呈现控制器的视图,其在此情况下与主机的视图一致。这里,仅存储器的相对小的一部分视为含有有效数据,存储器的其余部分由经擦除区块和可回收的空间组成。可回收空间由被过时数据占据的空间以及经擦除但不在经擦除区块中的空间组成。尽管在一些情形中经擦除空间的所述分散部分可用于存储额外数据,但在其它系统中,将这些经擦除空间合并为经擦除区块,其接着用于存储数据。图 18 展示直到时间 t_5 的有效数据量的稳定增加。在此时间期间,可回收空间的量保持恒定,因为没有执行回收操作。经擦除区块中的空间随着在此时期期间有效数据写入到经擦除区块而减小。在时间 t_5 处,一个或一个以上文件被删除,且来自存储在存储器阵列中的那些文件的数据变得过时。因此,有效数据占据的空间量减少且可回收空间的量相应地增加。接着,从时间 t_5 到 t_6 ,更多数据编程到存储器阵列,再次增加有效数据的量并减少经擦除区块中的空间。在时间 t_6 处,存储器达到回收操作被视为必要的以允许另外数据的编程的点。可当经擦除区块中的空间达到某个阈值时或可基于某种其它标准而达到此点。从时间 t_6 到 t_7 ,执行回收操作以将可回收空间中的一些转换为经擦除区块。在这些回收操作期间,没有新数据写入到存储器阵列,使得有效数据占据的空间量保持恒定。在时间 t_7 处,回收操作停止且新数据的编程再次开始。此系统类似于之前描述的系统,其中仅在必要时执行回收操作。此系统的一个缺点在于,从 t_6 到 t_7 的时间可能超过时间限制,使得主机取消写入操作。替代于仅在必要时执行回收操作,可以对主机写入操作没有显著影响的方式在回收操作成为必要之前执行回收操作。尤其在具有关于存储在存储器阵列中的过时数据量的更完整信息的存储器中,例如使用直接数据文件存储系统的那些存储器,可能以智能方式调度回收操作,使得其对主机写入操作具有极小的影响。

[0088] 图 19 展示可如何管理存储器阵列中的空间的模型。有效数据以恒定速率增加,直到存储器阵列填满有效数据为止。在同时,可回收空间随着回收操作将可回收空间转换为经擦除区块而减少。用于以此方式管理存储器阵列中的空间的系统根据调度来实施回收操作,使得个别回收操作分布在个别主机写入操作之间以提供恒定的主机数据写入速度。为了进行此操作,控制器可估计将用于填充存储器阵列的额外有效数据量并估计在存储器充满之前需要的回收操作的次数。接着以在剩余时间上均匀散布回收操作的速率调度回收操作。这确保了存储器不会过早地用完经擦除区块。回收操作可交错在如之前的将新数据编程到存储器阵列的操作之间。回收操作包含将数据部分从一个区块复制到另一区块,并擦除不含有效数据的区块。回收空间的速率可由交错比率确定,所述比率为回收操作的次数与主机写入操作的次数的比率。在一些情况下,为计算此比率而忽略区块擦除操作,因为存在比擦除操作多得多的复制操作。接着,所述比率变为用于回收的写入操作与用于新数据的写入操作的比率。

[0089] 图 20 说明当在时间 t_{10} 处删除一个或一个以上文件时在此系统中发生的事。在 t_{10} 之前,有效数据占据的空间正在增加,且可回收空间和经擦除区块中的空间两者随着有效数据量的增加而减少。以确保当存储器如虚线指示变满时将回收所有可回收空间的速率回收可回收空间。在时间 t_{10} 处,删除一个或一个以上文件,使得先前由有效数据占据的空间由过时数据占据,且因此变为可回收空间。在 t_{10} 之后,存储器阵列能够存储更多有效数据,但需要额外的回收操作来完成此存储。因此,控制器重新计算回收操作应执行的速率。这提供了响应于存储在存储器阵列中的数据的状态改变的自适应调度系统。因此修改了回

收的速率,且对有效数据进行编程的速率也改变,因为回收速率影响到对有效数据进行编程的速率。自适应回收调度系统可周期性或在由主机命令触发或由某个其它事件触发时重新计算交错比率。

[0090] 图 21 展示存储器管理的另一实例,其中存在三种不同的回收模式。在第一模式中,在 t_{12} 之前,在主机数据的写入期间不执行回收操作。有效数据量以经擦除区块中空间量减少的相同速率增加。可回收空间的量保持近似相同,因为在此时间期间没有数据变得过时。一些区块可能具有以保持区块的一部分经擦除且未使用的方式存储的有效数据。可通过合并有效数据以使得经擦除空间合并在一个或一个以上经擦除区块中来回收这些经擦除部分。因此,在此模式中,即使没有数据变得过时,也可能存在可回收空间的一些增加。在一些实例中,在此第一模式中可维持最小回收速率,使得回收速率从不为零。因为回收速率较低或为零,因此在此第一模式中主机数据编程速率较高。

[0091] 在第二模式中,从 t_{12} 到 t_{13} ,根据图 19 和 20 所示的自适应调度执行回收操作。因此,计算回收速率,且在新数据的编程之间交错回收操作,使得可回收空间随着存储器填满而减少。由于交错的回收操作,新数据编程速率在第二模式中比在第一模式中稍慢。当存储器中的有效数据超过阈值时,当经擦除区块中的空间下降到阈值以下时或基于某种其它标准,可开始第二模式。

[0092] 在第三模式中,在 t_{13} 之后,以最大速率执行回收操作,使得存在可用于写入新数据的经擦除区块。回收操作速率的增加致使主机数据编程速率的减小。当有效数据量超过阈值时,或当经擦除区块中的空间量下降到阈值以下时或基于某种其它标准,可开始第三模式。

[0093] 自适应调度——详细实例

[0094] 现将使用其中存储器具有直接数据文件存储接口的实例详细描述对非易失性存储器进行管理以使得其根据自适应调度系统实施回收操作,同时维持主机数据编程的恒定速率。此实例展示可如何从由存储器控制器监视的参数计算适当的交错比率。图 22 中详细说明针对此存储器的随着时间变化的存储器空间页的不同条件。还展示对于存储器阵列中的页或区块来说可能的许多转变。存储器阵列包含若干区块且每个区块含有多个页。术语“区块”和“页”将用于此实例,但实例还可用于具有元区块和元页作为擦除单位并分别编程的存储器。将区块视为在任意时间处于三种条件之一,且将这些区块内的页视为处于五个条件之一。图 22 中呈现随着时间过去,在这五个条件中的每一者中页的数目。

[0095] 用于数据存储的区块的分类

[0096] 文件区块:文件区块充满主机数据且不含过时数据。实际上,可允许文件区块含有最小数目的未编程页,例如 2 个。

[0097] 部分区块:部分区块含有一些主机数据且还含有一些经擦除页和 / 或过时页。

[0098] 经擦除区块:经擦除区块池中的完全擦除的区块。

[0099] 除了这三个类别以外,一些区块可能充满过时数据。举例来说,紧接在指示由主机删除或擦除一个文件之后,含有所述文件的区块经调度以进行垃圾收集。这些区块中的一些区块充满数据。然而,因为快速擦除这些区块而不需要许多资源,所以出于当前计算的目而不考虑这些区块。

[0100] 用于数据存储的页的分类

[0101] 以下是用于根据页的条件对存储器阵列的页和同一区块中的其它页进行分类的术语列表。图 22 中展示每个条件中页的数目。

[0102] 文件区块页 (FBP) :这是包含在文件区块中的有效主机数据的页的数目。

[0103] 数据页 (DP) :这是包含在部分区块中的有效主机数据的页的数目。

[0104] 过时页 (OP) :这是包含在部分区块中的过时主机数据的页的数目。

[0105] 经擦除页 (EP) :这是包含在部分区块中的经擦除页的数目。过时页 (OP) 和经擦除页 (EP) 一起可视为形成存储器阵列中的可回收空间。

[0106] 经擦除区块页 (EBP) :这是包含在经擦除区块中的经擦除页的数目。

[0107] 在此列表中忽略了可能存在于仅含过时数据的区块中的页,因为这些页不会长期存在,且在对其回收所必要的垃圾收集方面不会呈现显著的负担。

[0108] 除了上文列出的五个互斥类别以外,与存储器阵列中的页相关的另外两个数目是总页和总数据页。这些术语描述在以上类别中的页。存储器控制器跟踪这些数目。

[0109] 总页 (TP) :这是装置中可用于数据存储的总页数,且表示装置的数据容量。

[0110] 总数据页 (TDP) :这是在任意时间含有有效主机数据的页的数目。数据页可在文件区块中或部分区块中。

[0111] 数据复制操作阶段 :存在两个数据复制操作阶段,但其在下文的自适应调度分析中无需单独考虑。

[0112] (1) 垃圾收集阶段 :过时的页存在于装置中,且执行垃圾收集操作以将其消除。

[0113] (2) 区块合并阶段 :执行区块合并操作以恢复部分区块中占用的经擦除容量。

[0114] 通常,较好的是首先进行垃圾收集并推迟区块合并,因为如果主机将数据写入到同一文件作为有效数据,那么仅含有效和经擦除空间的区块在没有合并的情况下可能仍可使用。即使在文件关闭的情况下,也可重新打开文件并将数据写入到程序区块。具有过时数据的区块必须在其可再次使用之前进行垃圾收集,使得推迟此垃圾收集并没有优点。

[0115] 由装置操作引起的页转变

[0116] 图 22 中展示以下页转变 :

[0117] 过时页到经擦除区块页 (1) :当含有过时页的区块在有效数据页已从其复制之后被擦除时发生。

[0118] 数据页到数据页 (2) :涉及从源区块复制到目的地区块的数据页。

[0119] 经擦除页到经擦除区块页 (3) :在所有有效数据页已从源区块复制到目的地区块之后,在目的地区块中编程的经擦除页有效地变为当源区块经擦除时源区块中的经擦除区块页。

[0120] 经擦除区块页到经擦除页 (4) :在有效数据页从源区块复制到经擦除区块之后,经擦除区块中的剩余经擦除区块页变为经擦除页。

[0121] 数据页到文件区块页 (5) :部分区块中的有效数据页在区块变满时变为文件区块页。

[0122] 经擦除区块页到文件区块页 (6) :经擦除区块页在区块已被主机写入的数据填满时变为文件区块页。

[0123] 图 23 展示这些转变在存储器阵列的区块中如何发生的实例。图 23A 展示存储器阵列中的四个区块 A-D,其中不同区块在不同条件下具有不同数据量。区块 A 填充有有效数

据,因为页 0-3 每一者含有有效数据(由阴影指示)。因此,区块 A 视为文件区块。区块 B 包含含有有效数据(数据 X)的页 0、含有过时数据(由影线指示)的页 1 和 2 以及处于擦除状态的页 3。因为区块 B 含有一些有效数据,但未充满有效数据,所以区块 B 视为部分区块。区块 C 具有填充有有效数据的页 0 和 1 以及处于擦除状态的页 2 和 3。区块 C 也被视为部分区块,因为其含有一些有效数据,但未充满有效数据。区块 D 完全擦除,不含数据,且因此视为经擦除区块。

[0124] 因为图 23A 的区块 A 是仅文件区块,所以文件区块页(FBP)的数目等于区块 A 中页的数目,即 4。数据页的数目是区块 B 和 C 中有效页的数目,且因此等于 3。过时页(OP)的数目是区块 B 和 C 中填充有过时数据的页的数目,且等于 2。经擦除页(EP)的数目是区块 B 和 C 中处于擦除状态的页的数目,且等于 3。因为区块 D 是仅经擦除区块,所以经擦除区块页(EBP)的数目是区块 D 中的页的数目,且等于 4。在此实例中总页(TP)仅是所有区块中的总页数,这里为 16 页。总数据页(TDP)是含有有效数据的页的数目,这里为 7 页。

[0125] 图 23A 展示来自区块 B 的页 0 的有效数据 X 复制到区块 C 的页 2。这是类型 (2) DP 到 DP 的转变的实例。随后,在图 23B 中,仅过时数据保留在区块 B 中,使得区块 B 可在此点被放入擦除队列中。在图 23B 中,新的有效数据 Y 写入到区块 D,使得区块 D 的页 0 变为数据页。区块 D 的页 1-3 在类型 (4) 转变中从经擦除区块页改变为经擦除页。图 23C 展示在区块 B 经擦除之后的区块 A-D。页 0-2 在类型 (1) 的转变中从过时页转换为经擦除区块页。在图 23C 中,区块 B 的页 3 从部分区块中的经擦除页改变为经擦除区块中的经擦除区块页。这是类型 (3) 转变的实例。同样在图 23C 中,有效数据 Y 从区块 D 的页 0 复制到区块 C 的页 3,导致区块 C 变为充满有效数据,如图 23D 所示,因此,图 23D 的区块 C 视为文件区块。区块 C 的先前是数据页的页 0-2 在区块 C 变为文件区块时在类型 (5) 转变中变为文件区块页。同样在图 23D 中,区块 B 以填充区块 B 的有效数据编程。区块 B 的页在类型 (6) 转变中从经擦除区块页改变为文件区块页。

[0126] 本自适应调度方案的一个目的是交错主机写入和回收操作以提供恒定的主机数据编程速率。因此总数据页(TDP)的加速率恒定,直到总数据页(TDP)等于总页(TP)为止,即直到存储器变满为止。在本实例中在计算各种参数时进行某些近似。然而,其它实例可基于其它假定或可以其它方式实施计算。在本计算中,忽略所复制数据的数据群组结构。因此,数据群组可在复制时分裂。数据页可从源区块复制以填充目的地区块中的可用经擦除页,接着剩余部分可复制到单独的目的地区块。

[0127] 控制器维持用于计算回收操作与新数据写入之间的适当交错比率的某些参数值。这里呈现的计算使用页作为用于计算的数据的基本单位,然而也可使用其它单位,例如区块或元区块。可仅使用由控制器维持的参数值、总页(TP)、总数据页(TDP)、部分区块页(PBP)和经擦除区块页(EBP)来计算交错比率,如下文所述。

[0128] (主机数据):(复制数据)交错比率的推导

[0129] 主机数据的可在装置变满之前写入到装置的额外页的数目由以下表达式给出

[0130] 待写入的主机数据 = $TP - TDP$

[0131] 还可使用控制器监视的参数如下写出经擦除页(EP)和数据页(DP)的数目。

[0132] 经擦除页(EP) = $TP - TDP - OP - EBP$

[0133] 数据页(DP) = $PBP - EP - OP$

$$[0134] \quad = \text{PBP} - (\text{TP} - \text{TDP} - \text{OP} - \text{EBP}) - \text{OP}$$

$$[0135] \quad = \text{TDP} + \text{PBP} + \text{EBP} - \text{TP}$$

[0136] 基于所有部分区块含有相同数目的有效数据页的近似来确定必须复制的有效数据量。如果 N 是区块中的总页数,那么部分区块中的平均有效页数为 $N * \text{DP} / \text{PBP}$ 。

[0137] 存在于部分区块中的数据页的仅一小部分需要复制。其余部分存在于变为数据复制到的目的地区块的部分区块中。假定部分区块作用于全部复制的目的地区块,且不必使用经擦除区块。必须复制的数据所来自的区块的数目等于将产生的经擦除区块的数目,因为每个经擦除的部分区块具有首先从其复制的有效数据。所产生的经擦除区块的数目 = $(\text{PBP} - \text{DP}) / N$ 。

[0138] 因此,将待擦除的区块数目乘以每个区块的有效页的平均数目给出了将被复制以回收存储器阵列中的所有可回收空间的数据量的近似。

$$[0139] \quad \text{将复制的数据} \quad = \{N * \text{DP} / \text{PBP}\} * \{(\text{PBP} - \text{DP}) / N\}$$

$$[0140] \quad = \text{DP} * (\text{PBP} - \text{DP}) / \text{PBP}$$

$$[0141] \quad = (\text{TDP} + \text{PBP} + \text{EBP} - \text{TP}) * (\text{PBP} - \text{TDP} - \text{PBP} - \text{EBP} + \text{TP}) / \text{PBP}$$

$$[0142] \quad = (\text{TDP} + \text{PBP} + \text{EBP} - \text{TP}) * (\text{TP} - \text{TDP} - \text{EBP}) / \text{PBP}$$

[0143] 应注意,这过度估计了将复制的数据量,因为实际选择用于复制操作的源区块作为具有最低的将复制的有效数据量的区块。基于所有部分区块含有相同比例的数据页的简化的数据复制速率因此将比必要的速率稍高。

[0144] (主机数据):(复制数据)交错比率

$$[0145] \quad = (\text{将写入的主机数据}) / (\text{将复制的数据})$$

$$[0146] \quad = (\text{TP} - \text{TDP}) * \text{PBP} / (\text{TDP} + \text{PBP} + \text{EBP} - \text{TP}) / (\text{TP} - \text{TDP} - \text{EBP})$$

[0147] 这给出了一个交错比率,其允许主机数据编程以恒定速率继续,直到存储器变满为止。可周期性或响应于某个触发事件来更新比率。此比率是可用于调度回收操作以允许恒定的主机数据编程速率的公式的一个实例。也可使用其它公式。用于计算交错比率的公式可基于例如上文计算的计算,或可基于实际存储器中的经验。可通过做出关于在计算中忽略哪些因数的额外假定来简化公式。或者,更复杂的公式可考虑额外因数,例如擦除区块花费的时间或可能需要经擦除区块作为用于垃圾收集的目的地区块的可能性。

[0148] 尽管以上实例的存储器具有基于文件的主机接口,但本发明的方面可应用于使用基于扇区的主机接口或某种其它主机接口的存储器。尽管具有基于扇区的接口的一些存储器可能缺乏足够的信息来显著得益于应用所描述的技术,但可实现一些性能改进。另外,具有基于扇区的接口的一些存储器可分析 FAT,或另外获得关于所存储数据的情况的额外信息。一些主机可使用额外命令来向控制器提供关于所存储数据的信息。这些信息可用于在比原本可能的时间更早的时间调度回收。

[0149] 主机操作

[0150] 在一些实例中,可在不同模式中不同地管理非易失性存储器中的回收操作。如先前描述,回收操作可在第一模式中以某个最小速率(在一些实例中包含零)操作,在第二模式中以自适应方式操作,以及在第三模式中以最大速率操作。可由存储器控制器根据预定标准选择回收模式。在一些实例中也可由主机控制回收模式。主机可确定选择三种所述模式中的哪一种。另外,主机可具有基于当前主机活动或预期主机活动选择适当回收模式的

命令。主机系统可与图 1 所示的存储器系统物理上分离。或者,存储器卡内执行卡上应用程序的处理器可视为主机。此配置描述于第 60/705,388 号美国临时申请案。

[0151] 关于存储器回收模式的第一主机命令是“回收_打开”命令,其允许持续的回收操作而不是交错的操作。以此方式完成的回收操作视为后台操作,因为其不导致对主机命令执行的任何延迟,且对主机是透明的。“回收_打开”等效于“闲置”命令,“闲置”命令告诉存储器主机在一段时间中将不发送额外命令。在一些系统中,这些可能是相同的命令。只要接收到另一主机命令,所得的回收_打开模式就结束。

[0152] “回收_正常”命令允许存储器在默认回收模式中操作。这可能意味着根据自适应调度的回收,或可能意味着向存储器控制器给出回收模式选择的控制,存储器控制器接着基于某个预定标准选择回收模式。当接收到导致回收_打开模式终止的主机命令时,存储器可能默认到此模式。

[0153] “回收_关闭”命令导致禁止回收操作并仅执行主机操作。可选择此模式以提供最大的主机数据写入性能。通过回收_打开或回收_正常命令来终止此模式。

[0154] 按下降的回收速率次序的可能回收模式的分级为:

[0155] 回收_打开:装置执行持续的回收操作,直到接收到另一命令为止。

[0156] 最大交错:回收操作以固定的最大交错比率与主机数据写入操作交错。这是自适应交错比率的上限。

[0157] 自适应交错:回收操作根据自适应交错比率与主机数据写入操作交错。

[0158] 最小交错:回收操作以固定的最小交错比率与主机数据写入操作交错。这是自适应交错比率的下限。

[0159] 回收_关闭:禁止回收操作,且仅执行主机数据写入操作。

[0160] 应注意,以上描述涉及以恒定速率写入主机数据。在多个循环中观察由交错回收提供的恒定速率。当在个别循环级观察时,主机数据写入在散布有回收操作突发的周期性主机写入操作突发中发生。然而,每个循环或许多循环中的主机数据写入速率可以估计的速率保持恒定,使得其可维持,直到存储器阵列充满为止。

[0161] 在以上描述中,可回收空间转换为存储器阵列中的经擦除区块的速率展示为恒定。然而,即使每个循环或每单位时间的回收操作的数目可能恒定,但通过回收操作产生经擦除区块的速率可能不恒定。这是因为可选择将回收的区块,使得较容易回收的区块首先被回收。因此,当自适应回收开始时,可针对复制的每 R 个数据页回收区块,因为正在回收的区块具有 R 个有效数据页。稍后,可回收具有 $2R$ 个有效数据页的区块,使得针对所产生的每个经擦除区块花费 $2R$ 次复制操作。因此,通过回收产生经擦除区块的速率减小到早先速率的一半。在其它实例中,产生经擦除区块的速率可根据区块回收的次序以其它方式变化。如果不考虑区块中包含的有效数据量而回收区块,那么通过回收产生经擦除区块的速率将相当恒定。

[0162] 尽管已相对于本发明的示范性实施例描述了本发明的各个方面,但应了解本发明受到所附权利要求书的完整范围内的保护。

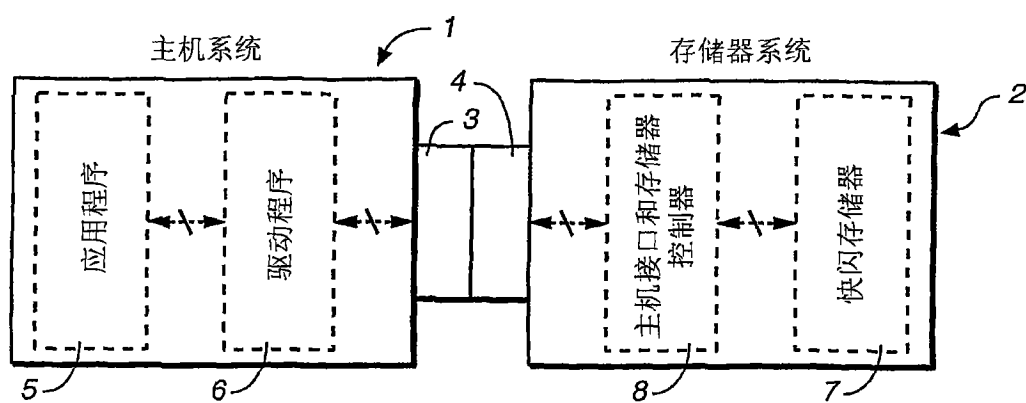
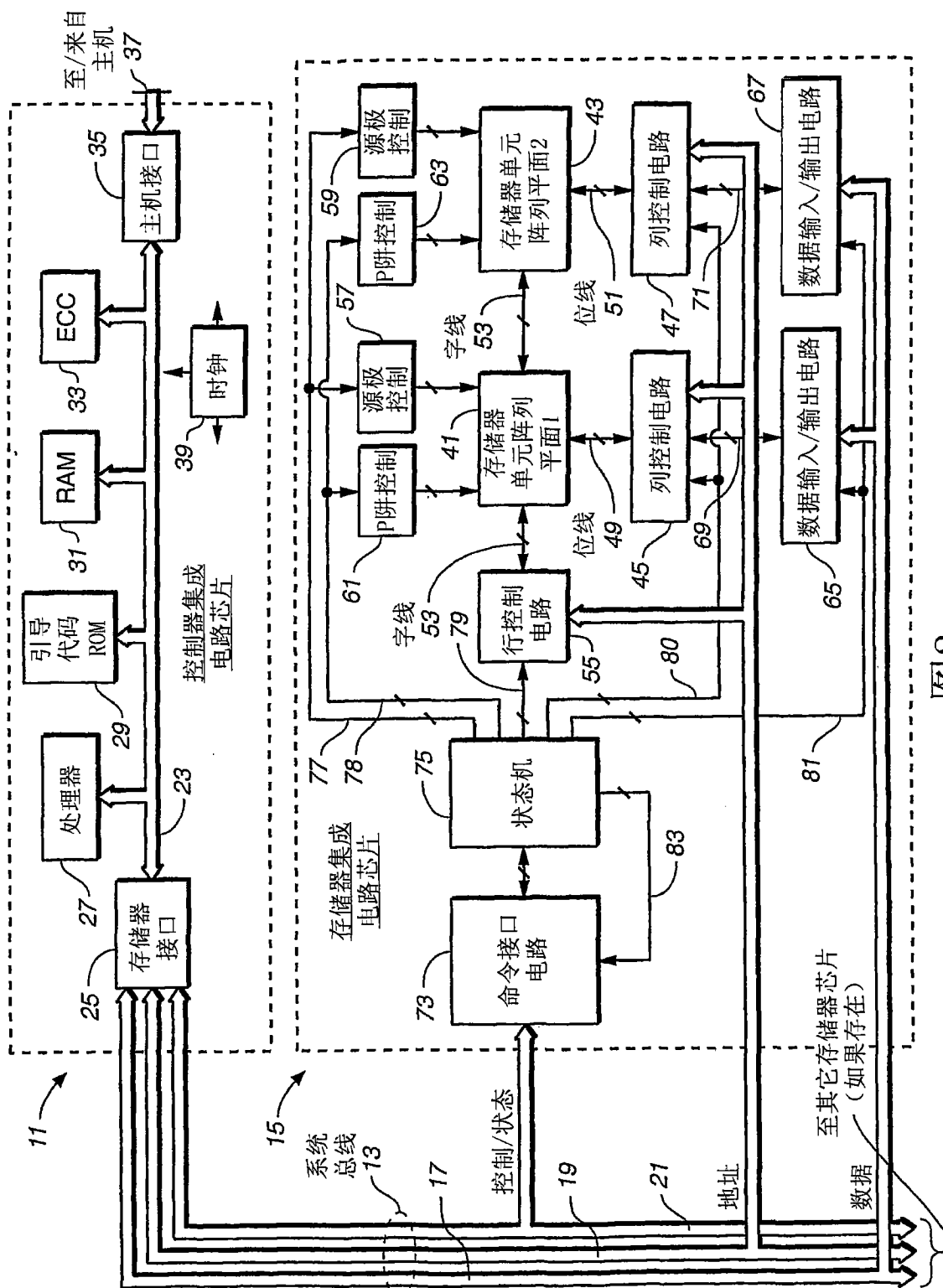


图 1



2
例

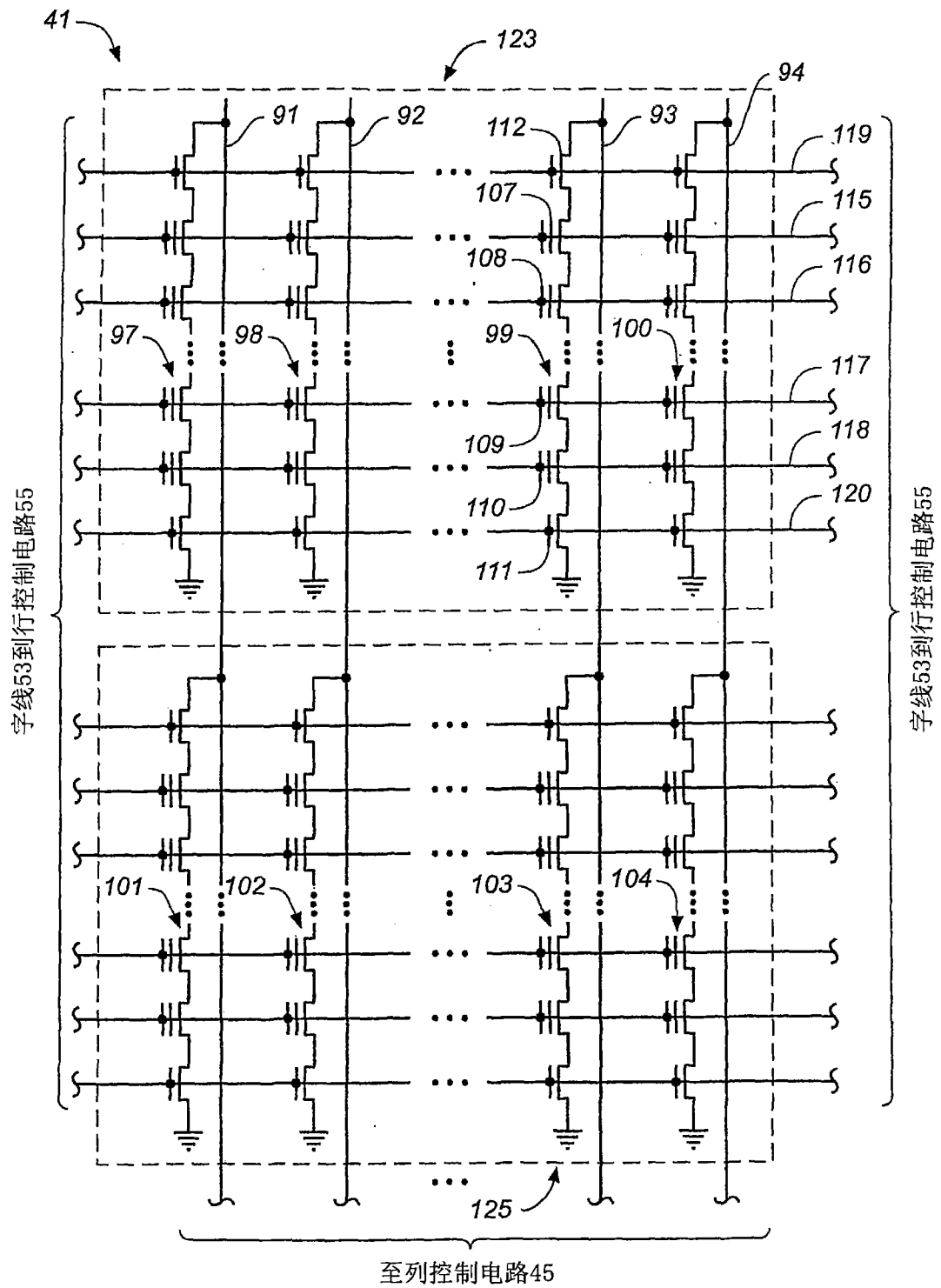


图 3

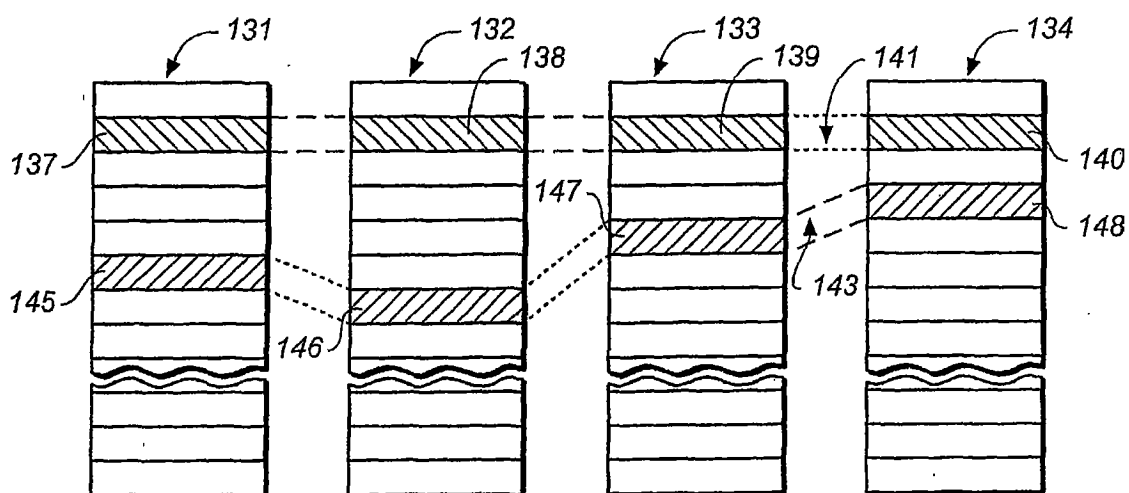


图 4

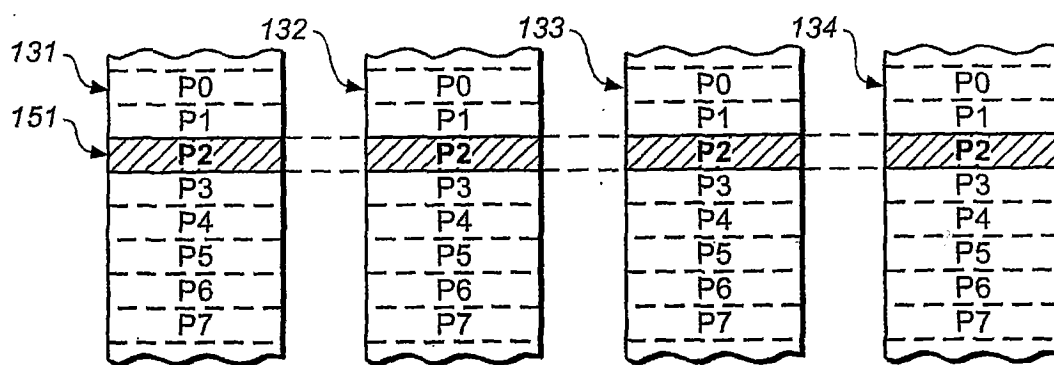


图 5

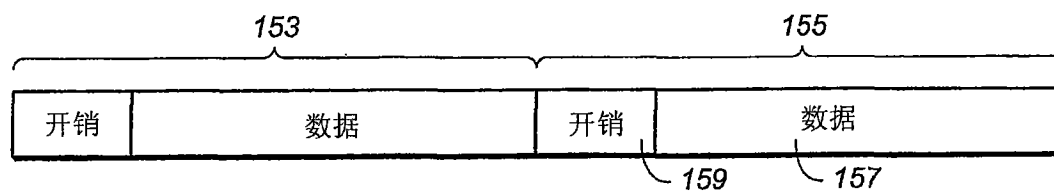


图 6

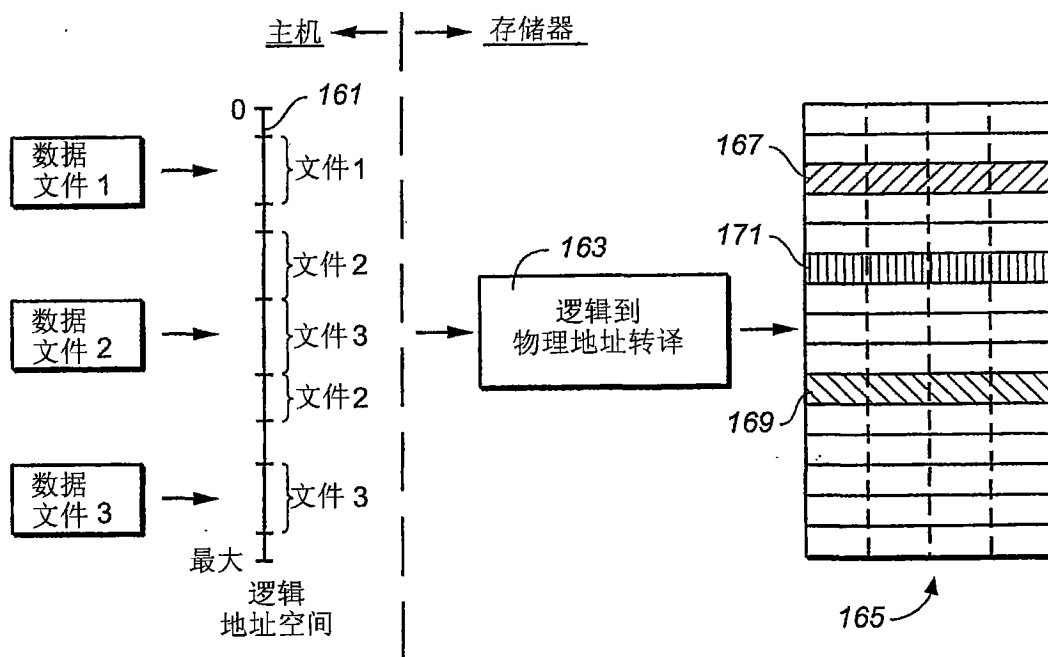


图 7

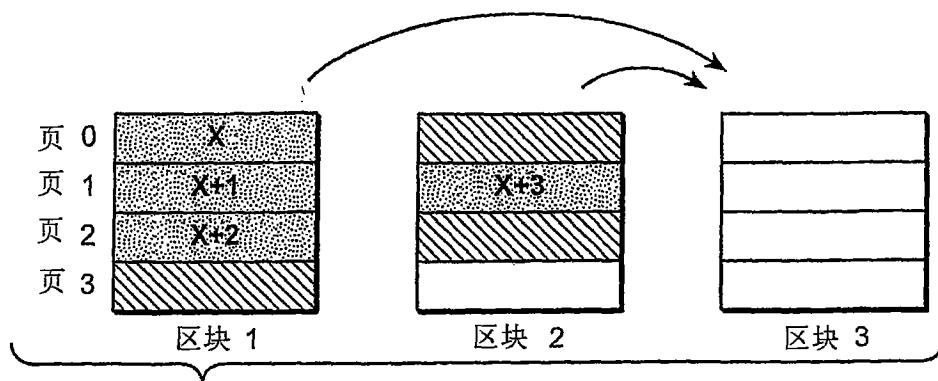


图 8A

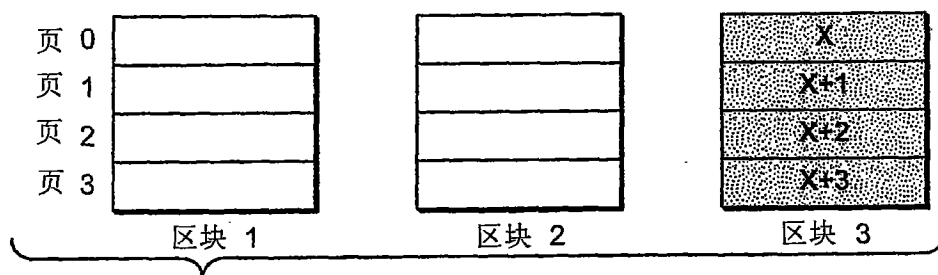


图 8B

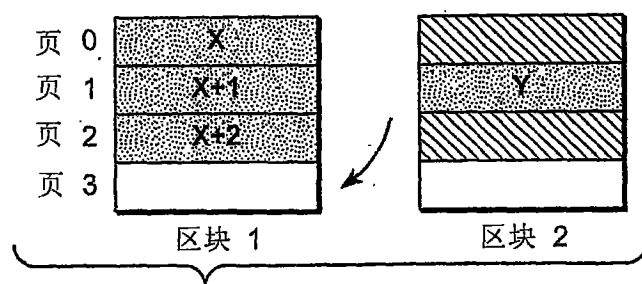


图 9A

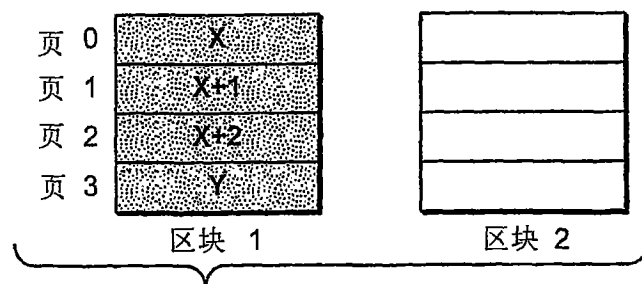


图 9B

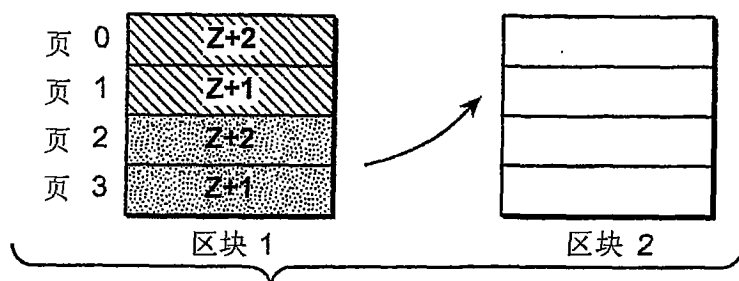


图 10A

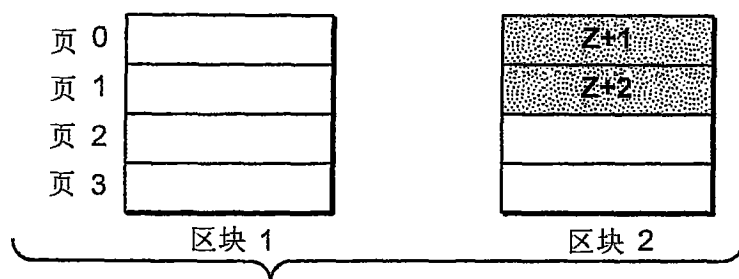


图 10B

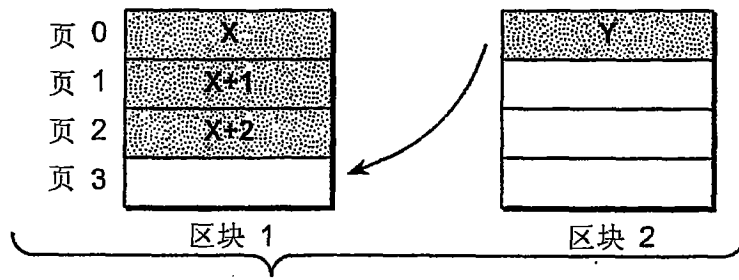


图 11A

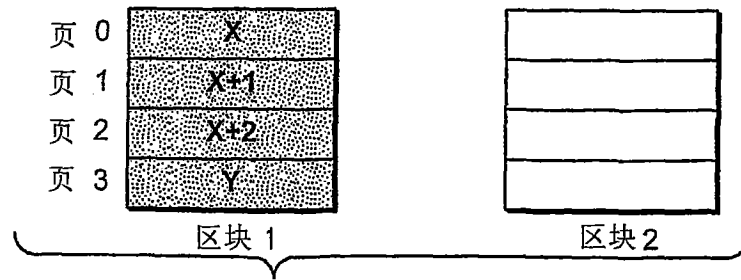


图 11B

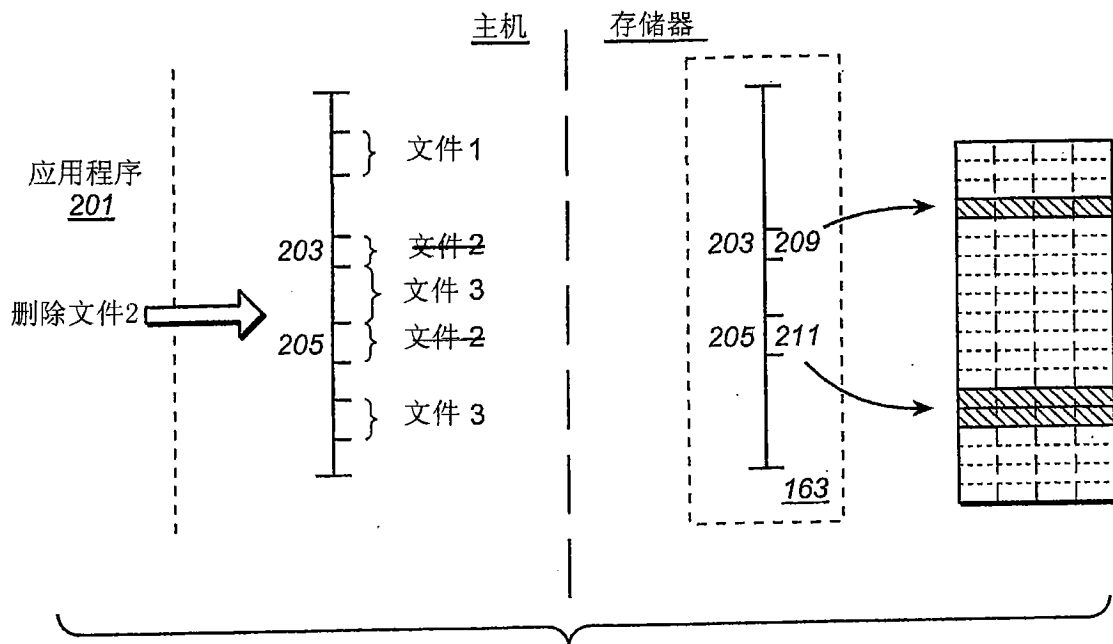


图 12

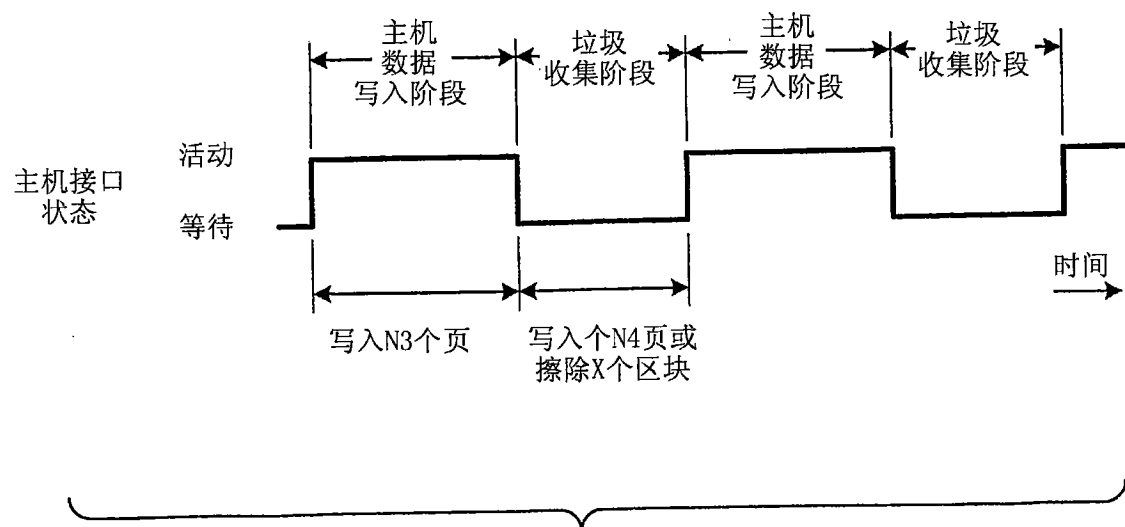


图 15

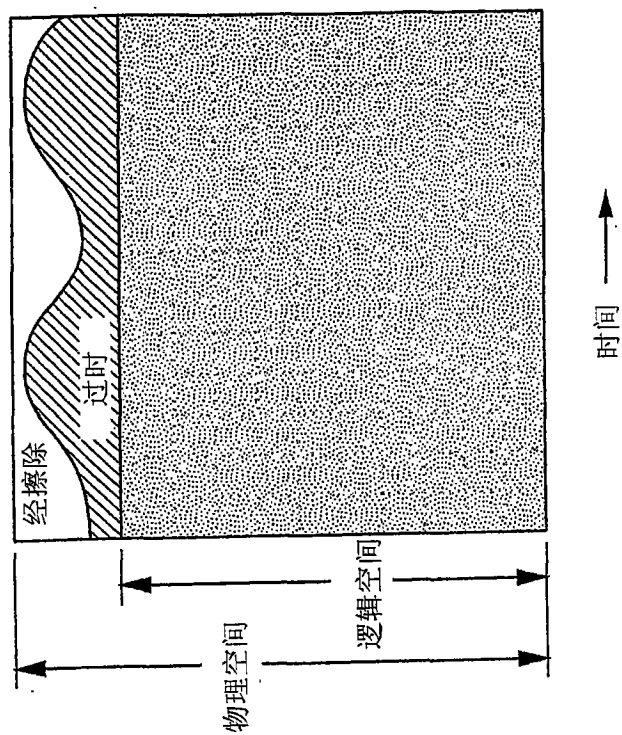


图14

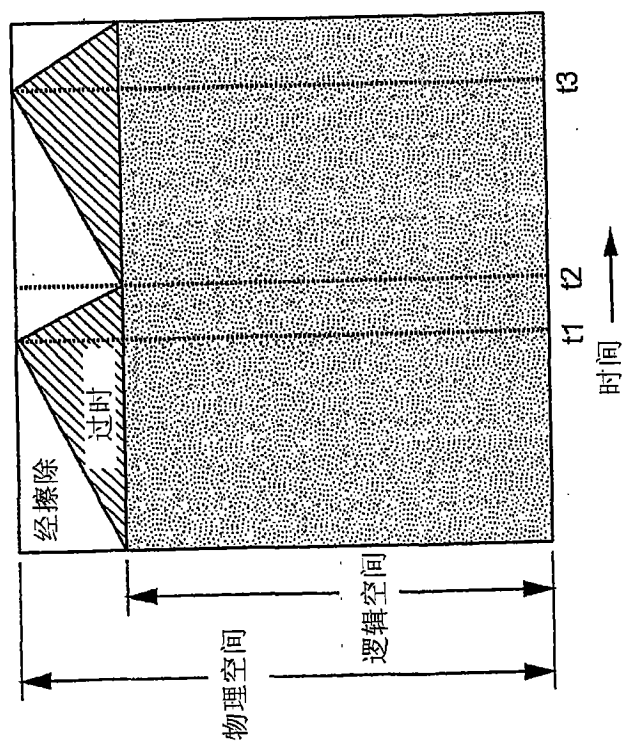


图13

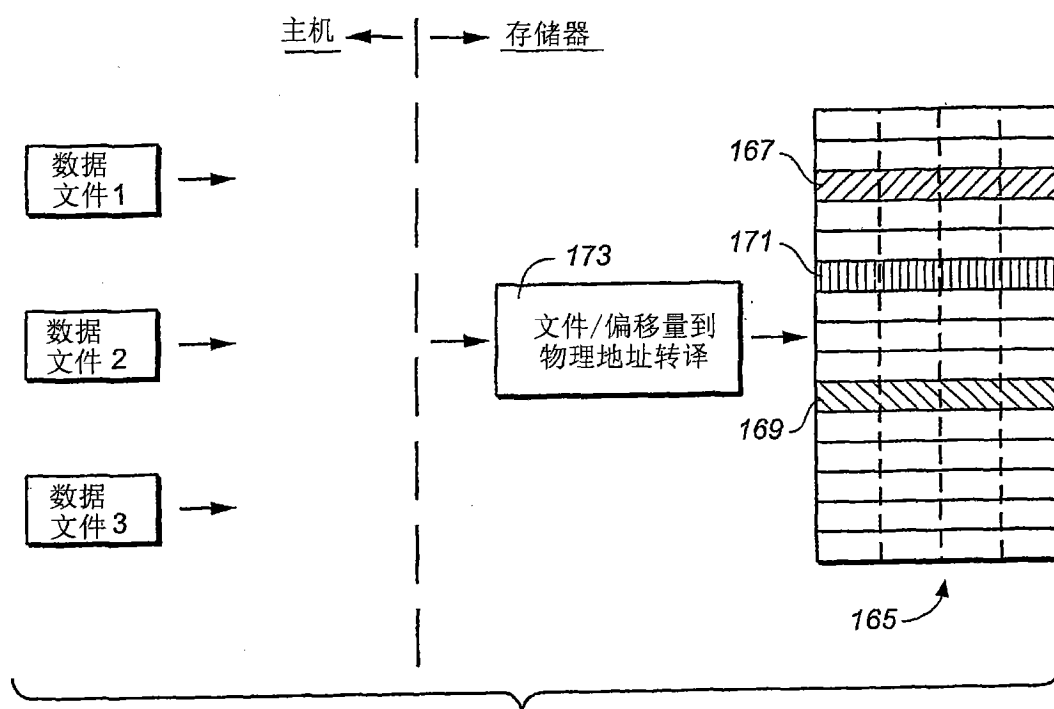


图 16

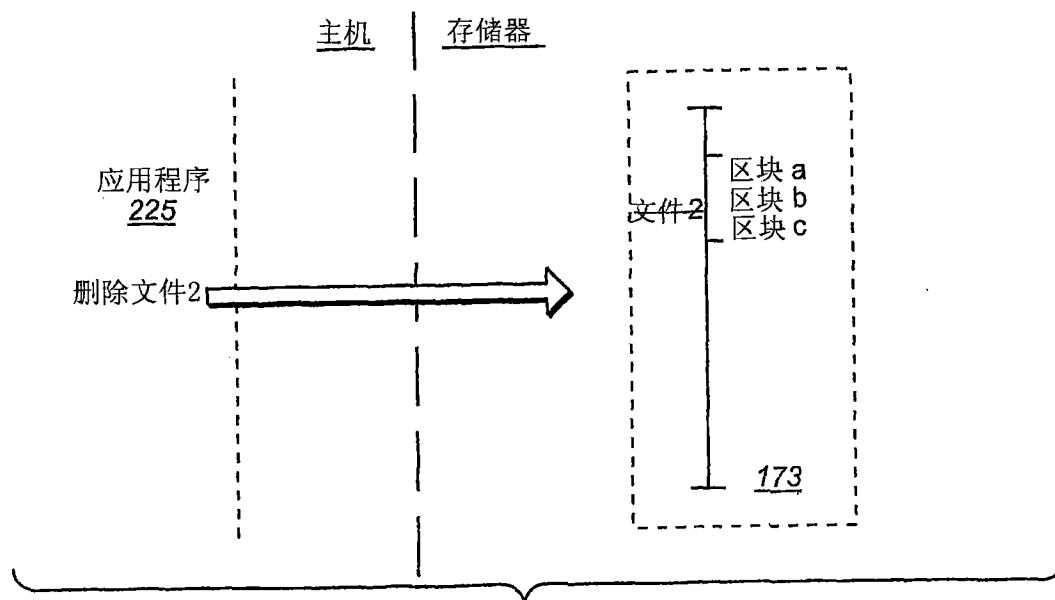
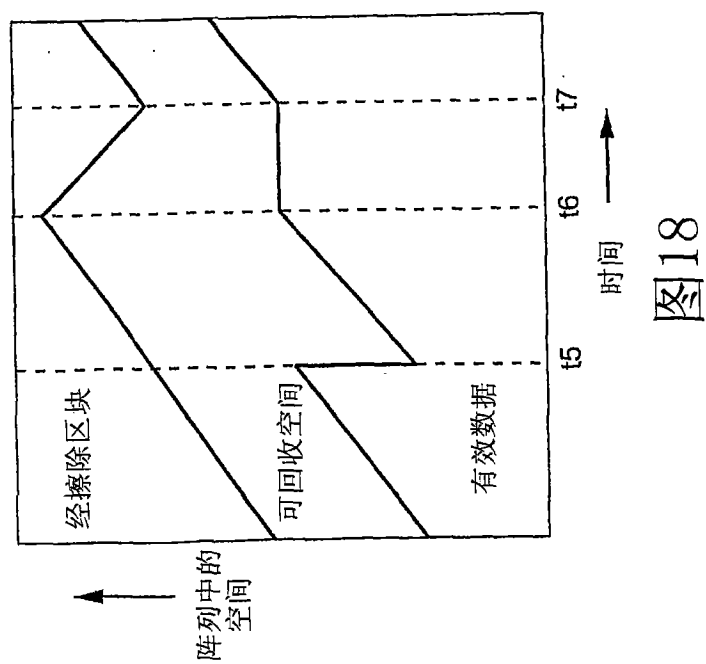
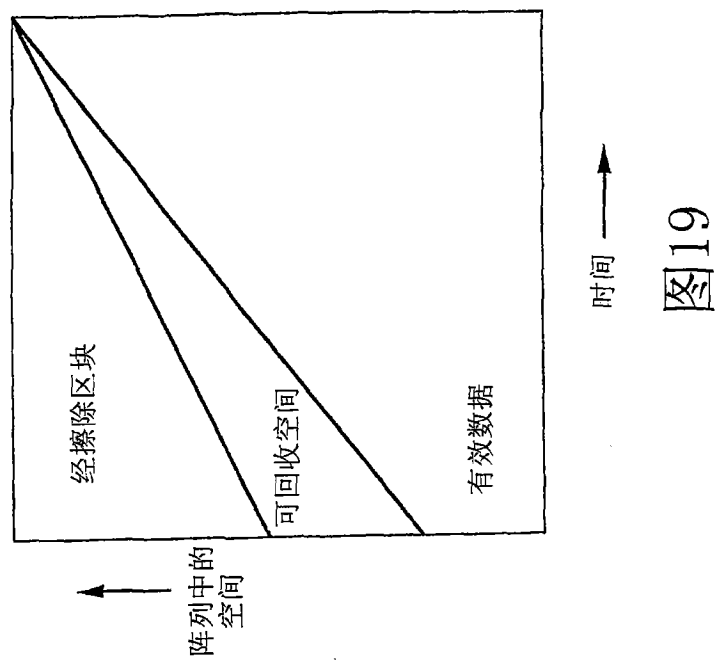


图 17



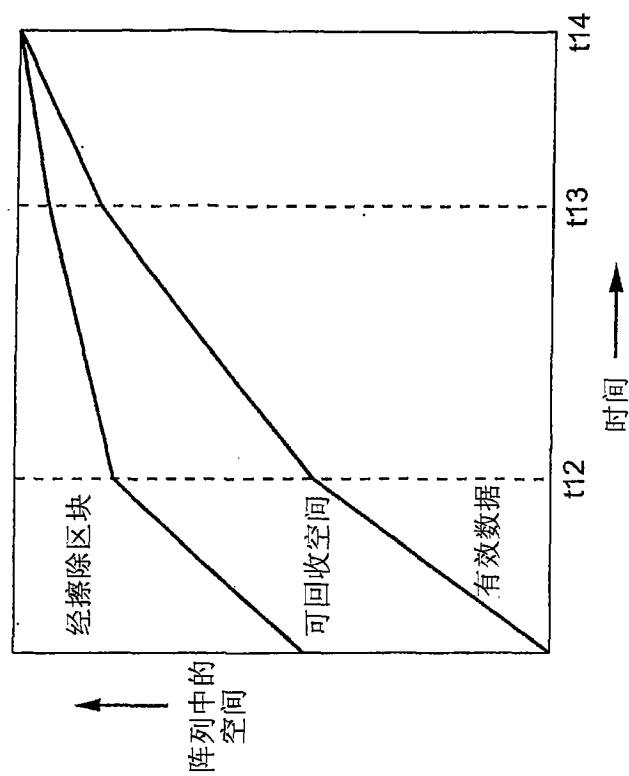


图20

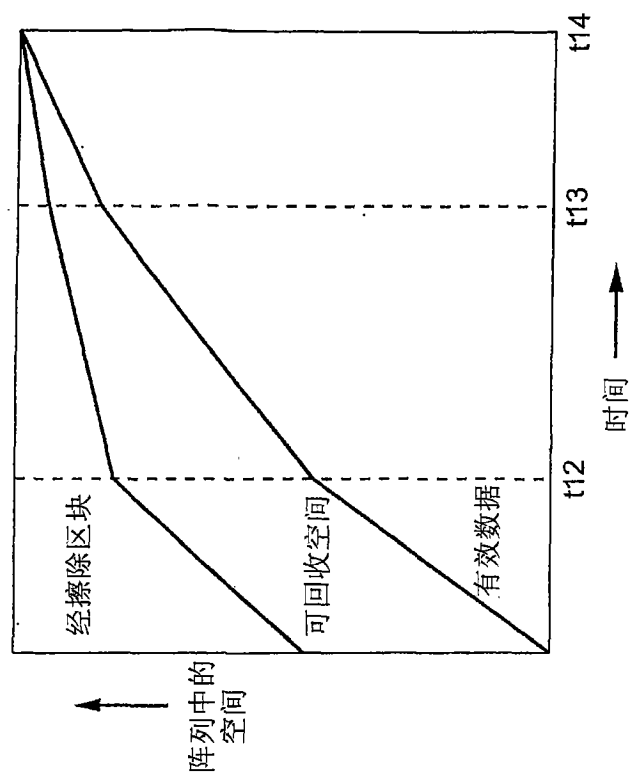


图21

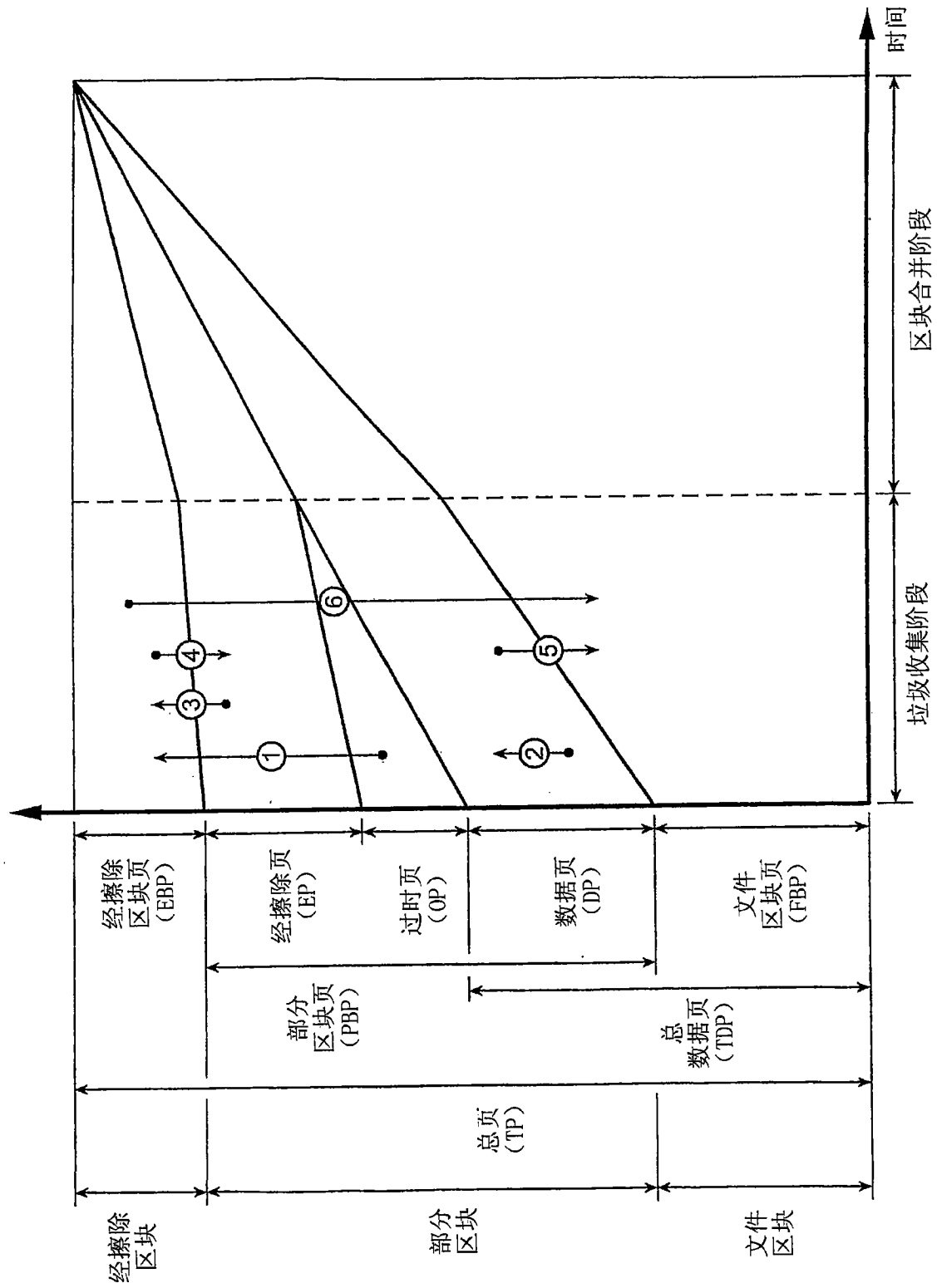


图22

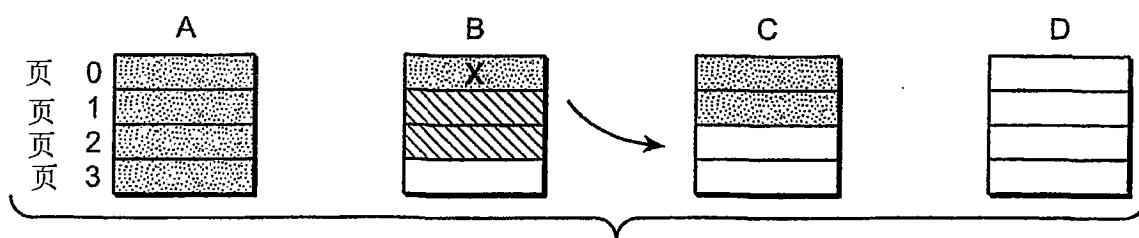


图 23A

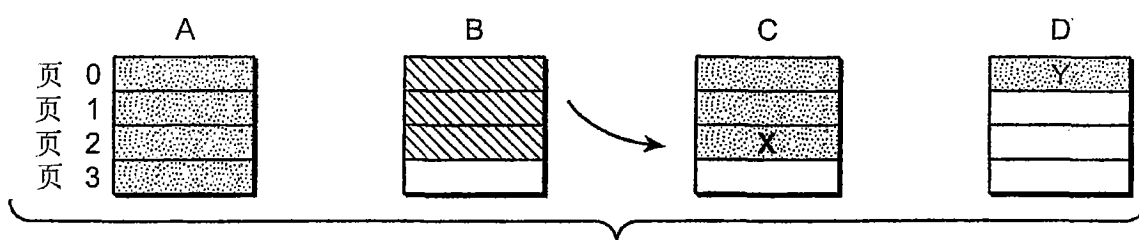


图 23B

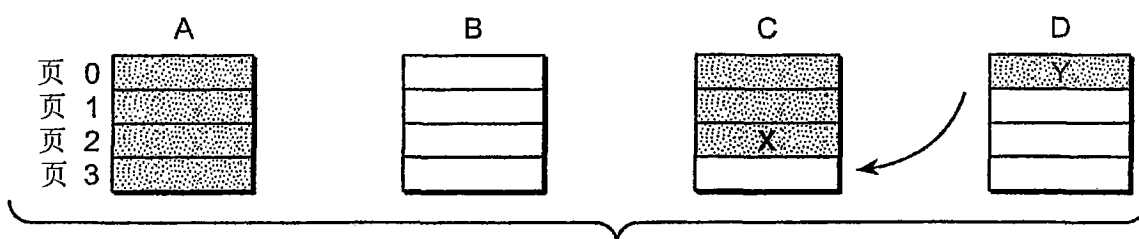


图 23C

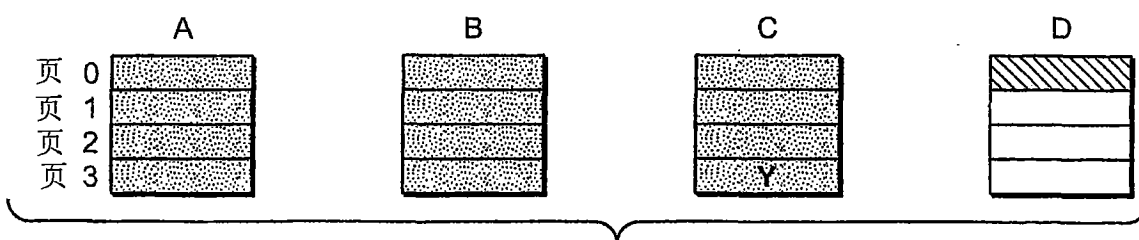


图 23D