



(10) 申请公布号 CN 105103234 A

(21) 申请号 201380070964.7

(51) Int. Cl.

(22) 申请日 2013.11.19

G11C 16/00(2006.01)

H01L 27/00(2006.01)

(30) 优先权数据

61/728394 2012. 11. 20 US

61/775327 2013.03.08 US

(85) PCT国际申请进入国家阶段日

2015. 07. 20

(86) PCT国际申请的申请数据

PCT/US2013/070789 2013. 11. 19

(87) PCT国际申请的公布数据

W02014/081719 EN 2014. 05. 30

(71) 申请人 查尔斯·I·派德尔

地址 美国加利福尼亚州

(72) 发明人 查尔斯·I·派德尔

M. 斯内尔格罗夫 R. 麦肯兹

X. 斯内尔格罗夫

(74) 专利代理机构 中国专利代理(香港)有限公司

司 72001

代理人 申屠伟进 陈岚

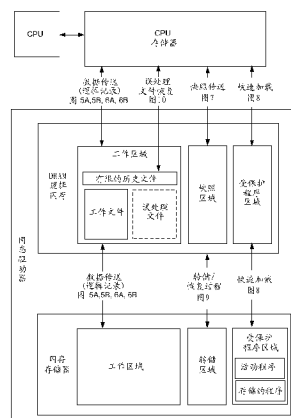
权利要求书5页 说明书19页 附图12页

(54) 发明名称

固态驱动器体系结构

(57) 摘要

一种固态驱动器包含 DRAM 逻辑闪存和闪存存储器,其中系统处理器仅对 DRAM 逻辑闪存读取和写入,这最小化对闪存存储器的写入。一种用于操作固态闪存装置的方法包含由 CPU 通过使用闪存命令和格式化将命令和数据发送到 DRAM 逻辑闪存而写入到固态驱动器。



1. 一种固态驱动器,包括:
DRAM 逻辑闪存;以及
闪存存储器;
其中系统总线读取和写入到 DRAM 逻辑闪存而不是闪存存储器。
2. 权利要求 1 的所述驱动器,其中系统处理器不能够直接命令固态驱动器写入到闪存存储器。
3. 权利要求 1 的所述驱动器,其中固态驱动器在预定点处从 DRAM 逻辑闪存将数据写入到闪存存储器,所述预定点独立于由系统处理器发出的命令。
4. 权利要求 3 的所述驱动器,其中预定点由含在固态驱动器内的逻辑来确定。
5. 权利要求 1 的所述驱动器,其中 DRAM 逻辑闪存通过以与 NAND 闪存相同的方式存储文件来模仿闪存存储器并且响应于闪存命令。
6. 权利要求 5 的所述驱动器,其中 DRAM 逻辑闪存通过使用 FAT 表、更新逻辑记录、组合文件、并且直接连接到 SATA 总线来模仿闪存存储器。
7. 权利要求 1 的所述驱动器,进一步包括能量存储装置,所述能量存储装置存储足够功率以在固态驱动器的外部功率被中断的情况下将存储在 DRAM 逻辑闪存中的数据内容移动到闪存存储器。
8. 权利要求 7 的所述驱动器,其中能量存储装置被含在驱动器的物理外壳内。
9. 权利要求 1 的所述驱动器,其中 DARM 逻辑闪存以全接口速度实施所有闪存命令。
10. 权利要求 1 的所述驱动器,其中 DARM 逻辑闪存存储程序追踪点和操作快照,并且最小化到闪存存储器的写入。
11. 权利要求 1 的所述驱动器,其中固态驱动器被配置成针对任何掉电提供附着的计算装置的最近状态的全恢复。
12. 权利要求 1 的所述驱动器,进一步包括用于管理 DRAM 逻辑闪存的独立微处理器。
13. 权利要求 1 的所述驱动器,进一步包括多个微处理器,所述多个微处理器用于并行并且以等同于与固态驱动器的外部接口的设计操作速度的速度来管理来往多个闪存存储器装置的数据传送。
14. 权利要求 1 的所述驱动器,其中 NAND 闪存的页面被个别地识别为具有错误并且在块中的其它页面仍然在使用时从块中消除。
15. 权利要求 14 的所述驱动器,其中闪存存储器包括多级单元 NAND 闪存,其中将页面个别地识别为具有错误并且在扇区中的其它页面仍然在使用时从块中消除它们使闪存存储器的寿命增加为当在扇区内的页面中检测到错误时消除整个扇区的闪存存储器的寿命的至少 10 倍。
16. 权利要求 1 的所述驱动器,其中所有数据被保留在 DRAM 逻辑闪存中直到数据被正确地写入到闪存存储器。
17. 权利要求 1 的所述驱动器,进一步包括用于向 DRAM 逻辑闪存的独立读取和写入操作的高速 DMA。
18. 权利要求 1 的所述驱动器,进一步包括高速内部总线,其中 DRAM 逻辑闪存和闪存存储器被分离地连接到高速内部总线,闪存存储器被划分成分离模块,每个模块由被配置成与高速内部总线独立地对接的不同微处理器控制。

19. 权利要求 18 的所述驱动器,其中微处理器基于运行 6502 指令的新实施方式的 6502 芯片组。

20. 权利要求 1 的所述驱动器,其中 DRAM 逻辑闪存通过将所有传送存储到闪存存储器并且选择地保持传送以最小化对闪存存储器的写入耗损来模仿闪存存储器存储。

21. 权利要求 1 的所述驱动器,其中从闪存存储器的读出速度是每秒至少 300 兆字节。

22. 权利要求 1 的所述驱动器,进一步包括独立的微处理器,所述独立的微处理器用于管理 SATA 接口读取和写入到 DRAM 逻辑闪存。

23. 权利要求 1 的所述驱动器,其中主微控制器从固态驱动器外部的中央处理器单元接收命令,其中主微控制器在主控制器不直接管理数据传送的情况下通过将子命令发送到其它微处理器来控制命令的执行,其中每个微处理器独立地操作。

24. 权利要求 1 的所述驱动器,进一步包括 DRAM 控制器,所述 DRAM 控制器被配置成测量针对文件的读取和写入活动,并且当针对文件的读取和写入活动停止预定时间段时,DRAM 控制器请求主控制器命令文件被移动到闪存存储器。

25. 权利要求 1 的所述驱动器,进一步包括:

内部总线,操作地连接 DRAM 逻辑闪存和闪存存储器;以及

主控制器,控制经过 DRAM 逻辑闪存与闪存存储器之间的内部总线发送的数据。

26. 权利要求 1 的所述驱动器,进一步包括多个独立闪存控制器,每个在分离的闪存存储器模块内,使得分离的闪存存储器模块能够被并行写入。

27. 权利要求 26 的所述驱动器,其中每个闪存存储器模块包括:

微处理器;

闪存存储器;

存储缓冲器,对微处理器和闪存存储器直接可访问;以及

接口,用于将数据直接写入到存储缓冲器。

28. 权利要求 27 的所述驱动器,其中每个闪存存储器模块被配置成将数据同时写入到第一存储缓冲器中并且从第二存储缓冲器中读出。

29. 权利要求 27 的所述驱动器,其中到存储缓冲器的传送速率处于全总线速度并且在总线时钟的控制下,并且从存储缓冲器中向外的传送速率处于闪存或处理器总线速度并且使用探测来执行。

30. 权利要求 27 的所述驱动器,其中存储器控制器被配置成将包括来自接口的数据、内部控制数据、和纠错码的数据传送到闪存管芯,其中来自接口的数据、内部控制数据、和纠错码通过单个 DMA 动作被传送到闪存管芯。

31. 权利要求 27 的所述驱动器,进一步包括缓冲器开关,所述缓冲器开关通过转变到空状态而在外部总线与内部总线之间切换存储缓冲器。

32. 权利要求 1 的所述驱动器,其中 DRAM 逻辑闪存包括:

逻辑闪存控制器;

DRAM 存储器;

DRAM 接口;

SATA DMA;以及

总线 DMA。

33. 一种用于操作固态闪存装置的方法,包括:由 CPU 通过使用闪存存储器命令以及格式化将命令和数据发送给 DRAM 逻辑闪存而写入到固态驱动器,其中 DRAM 逻辑闪存看来具有固态驱动器中的多个分离闪存存储器模块的组合工作区域的容量。

34. 权利要求 33 的所述方法,其中 CPU 不能够直接命令固态驱动器写入到闪存存储器模块。

35. 权利要求 34 的所述方法,其中主控制器命令在独立于由 CPU 发出的命令的预定地点从 DRAM 逻辑闪存将数据写入到闪存存储器模块。

36. 权利要求 33 的所述方法,进一步包括由 DRAM 逻辑闪存通过以下来模仿闪存存储器:

以与 DRAM 逻辑闪存中的 NAND 闪存相同的方式存储文件;并且
用 DRAM 逻辑闪存响应于闪存命令。

37. 权利要求 36 的所述方法,进一步包括由 DRAM 逻辑闪存通过以下来模仿闪存存储器:

使用 FAT 表;
更新逻辑记录;
组合文件;以及
直接连接到 SATA 总线。

38. 权利要求 33 的所述方法,进一步包括,如果固态闪存装置断电,则:

激活能量存储装置,所述能够存储装置存储足够的功率来将存储在 DRAM 逻辑闪存中的整个数据内容写入到闪存存储器;以及
将存储在 DRAM 逻辑闪存中的数据移动到闪存存储器。

39. 权利要求 38 的所述方法,进一步包括:
用感测电路感测固态闪存装置的断电;以及
由主控制器建立 DRAM 逻辑闪存与闪存存储器模块之间的数据路径。

40. 权利要求 39 的所述方法,进一步包括:
将表发送到闪存存储器的指定位置;
将选择的系统快照发送到转储闪存存储器;以及
将与未存储的逻辑记录相关联的数据写入到闪存存储器。

41. 权利要求 40 的所述方法,进一步包括为了在固态闪存装置断电之后恢复:
从闪存存储器的指定位置取回表;
用主控制器使用表来访问闪存存储器中的快照;
在固态闪存装置断电之前重构操作系统状态;以及
由重构的操作系统使用逻辑记录以从闪存存储器取回文件。

42. 权利要求 33 的所述方法,进一步包括:
将程序存储在闪存存储器模块的受保护程序区域中;
将存储的程序中的一个从受保护的程序区域传送到闪存存储器模块的活动程序区域;

将程序文件从活动程序区域移动到 DRAM 逻辑闪存中的受保护程序区域;以及
应操作系统的请求,将程序文件从 DRAM 逻辑闪存移动到 CPU 存储器。

43. 权利要求 33 的所述方法,进一步包括:
确定系统状态的快照应当被保存;以及
使用不使用逻辑记录数目的专用文件传送路线将快照保存在 DRAM 逻辑闪存的快照区域中;

当固态驱动器关断时,将至少一个选择的快照保存到闪存存储器模块。

44. 权利要求 33 的所述方法,进一步包括通过以下从固态闪存装置读取数据:
由固态闪存装置中的主控制器从 CPU 接收读取命令;
用主控制器确定由读取命令请求的文件是否被存储在 DRAM 逻辑闪存中;
如果文件被存储在 DRAM 逻辑闪存中,则将文件传送到 CPU;以及
如果文件不被存储在 DRAM 逻辑闪存中,则将来自主控制器的指令发送到闪存存储器模块中的闪存控制器以取回数据并且将它存储在 DRAM 逻辑闪存中,然后将数据从 DRAM 逻辑闪存传送到 CPU。

45. 权利要求 44 的所述方法,进一步包括:

在接收针对先前请求的数据的后续请求时,将数据从 DRAM 逻辑闪存传送到 CPU;
监控数据的变化;以及

如果数据变化,则在预定的点处将数据传送到闪存存储器模块。

46. 权利要求 33 的所述方法,进一步包括通过以下写入到固态闪存装置;

由 CPU 发送写入命令并且将要被写入的数据放置在系统总线上;

由固态闪存装置接收写入命令并且从系统总线接受数据;

将数据存储在 DRAM 逻辑闪存上;以及

向 CPU 报告向非易失性存储器的写入完成。

47. 权利要求 46 的所述方法,进一步包括用主控制器确定存储在 DRAM 逻辑闪存上的数据是否应当被存储在闪存存储器模块中。

48. 权利要求 47 的所述方法,其中如果写入命令和数据包括对 FAT 表的更新,则将数据存储在 DRAM 逻辑闪存中并且直到固态驱动器关断才将数据发送到闪存存储器模块。

49. 权利要求 46 的所述方法,进一步包括,如果主控制器确定 DRAM 逻辑闪存中的数据不正被使用,则通过以下将数据发送到闪存存储器模块:

顺序地启用闪存存储器模块;以及

在高速内部总线上顺序地将数据传送到启用的闪存存储器模块。

50. 权利要求 49 的所述方法,其中数据在所有的闪存存储器模块上面分带以允许并行写入到闪存存储器模块。

51. 权利要求 33 的所述方法,其中 DRAM 逻辑闪存包括易失性 CPU 存储器。

52. 权利要求 33 的所述方法,其中 CPU 写入到 DRAM 逻辑闪存最小化了固态驱动器中的非易失性闪存存储器上的耗损并且其中 DRAM 逻辑闪存具有比非易失性闪存存储器显著更快的读取和写入时间。

53. 权利要求 33 的所述方法,进一步包括快速加载操作,所述快速加载操作包括将软件可执行文件存储在对用户不可访问的 DRAM 逻辑闪存的受保护区域上。

54. 权利要求 53 的所述方法,其中软件可执行文件不使用逻辑记录数目来索引。

55. 权利要求 53 的所述方法,进一步包括使用与逻辑记录数据传送路径分离的协议路

径来加载软件可执行文件到 CPU 存储器中。

56. 权利要求 53 的所述方法,进一步包括作为后台操作更新 DRAM 逻辑闪存的受保护区域上的软件可执行文件。

57. 一种固态驱动器中的闪存存储器模块,所述闪存存储器模块包括:

多个闪存管芯;以及

闪存存储器控制器,包括;

微处理器;

多个存储缓冲器;

总线 DMA 接口,从外部总线将数据直接存放到多个存储缓冲器中;以及

闪存存储器 DMA 模块,在总线接口正在将来自外部总线的数据存放在第二存储缓冲器中时,直接访问多个存储缓冲器中的数据并且将数据从第一存储缓冲器传送到闪存管芯。

58. 权利要求 1 的所述驱动器,其中 DRAM 逻辑闪存被配置成以下面模式中的至少一个来操作:逻辑记录索引的数据传送和不通过逻辑记录索引的数据传送。

59. 权利要求 58 的所述驱动器,其中被写入到 DRAM 逻辑闪存的数据的部分不被写入到闪存存储器。

60. 权利要求 59 的所述驱动器,其中被写入到 DRAM 逻辑闪存的数据是快照,并且快照不被写入到闪存存储器。

61. 权利要求 59 的所述驱动器,其中被写入到 DRAM 逻辑闪存的数据是快速加载程序,并且快速加载程序不被写回到闪存存储器。

62. 权利要求 58 的所述驱动器,其中 DRAM 逻辑闪存被配置成用不通过逻辑记录索引的数据传送来操作,其中所述数据传送包括快速加载操作,所述快速加载操作将可执行文件放置在 DRAM 逻辑闪存中用于由 CPU 更迅速取回。

63. 权利要求 58 的所述驱动器,其中 DRAM 逻辑闪存被配置成用不通过逻辑记录索引的数据传送来操作,其中所述数据传送包括保存主机操作配置用于系统恢复的快照操作。

固态驱动器体系结构

[0001] 相关申请

本申请根据美国法典第 35 条 119 (e)款要求保护 2012 年 11 月 20 日提交的美国临时申请号 61/728,394 和 2013 年 3 月 8 日提交的美国临时申请号 61/775,327 的权益,所述临时申请通过引用整体结合在本文中。

背景技术

[0002] 计算装置将程序可执行文件和数据保存在非易失性存储器中。这使文件在被重启之后或在电源中断之后可用于计算装置。传统地,用于大文件的优选的非易失性存储已是硬盘驱动器。硬盘驱动器包含马达驱动轴上的旋转刚性盘片。数据通过浮置在盘片之上的气膜的磁头从盘片磁性读取和被写入到盘片。这些盘片典型地以每分钟 4,200 ~ 15,000 转数(rpm)的速度旋转。硬盘驱动器具有许多劣势,包含与旋转盘和移动磁头的机械性质相关的访问时间、高的功耗、机械故障、和低的抗冲击。

[0003] 固态驱动器(SSD)是非易失性存储装置,其使用集成电路来存储数据并且因此不含有移动的部分。SSD 具有优于硬盘驱动器的许多优势,包含更高的抗冲击、更低的访问时间、和更可变的因子。此外,SSD 典型地在操作期间比硬盘驱动器消耗少得多的功率。因此,SSD 允许更小、更薄的装置轮廓和电池充电上的更长操作。

附图说明

[0004] 附图图解在本文中描述的原理的各种示例,并且是说明书的部分。图解的示例仅仅是示例并且不限制权利要求的范围。

[0005] 图 1 是依据在本文中描述的原理的一个示例的说明性固态驱动器体系结构的框图。

[0006] 图 2 是依据在本文中描述的原理的一个示例的闪存存储器模块的框图。

[0007] 图 3 是依据在本文中描述的原理的一个示例的结合在图 2 中示出的闪存存储器模块的说明性固态驱动器体系结构的框图。

[0008] 图 4 是依据在本文中描述的原理的一个示例的出现在 CPU/CPU 存储器、DRAM 逻辑闪存、以及闪存存储器之间的各种交互和过程的图示。

[0009] 图 5A 示出依据在本文中描述的原理的一个示例的在将数据写入到固态驱动器期间的数据流。

[0010] 图 5B 是依据在本文中描述的原理的一个示例的描述用于将数据写入到固态驱动器的方法的流程图。

[0011] 图 6A 示出依据在本文中描述的原理的一个示例的在将数据从固态驱动器读取到 CPU 存储器期间的数据流。

[0012] 图 6B 是描述依据在本文中描述的原理的一个示例的用于将数据从固态驱动器读取到 CPU 存储器的方法的流程图。

[0013] 图 7 是示出依据在本文中描述的原理的一个示例的用于保存系统状态的快照的

方法的流程图。

[0014] 图 8 是依据在本文中描述的原理的一个示例的用于从 SSD 快速加载程序文件到 CPU 存储器的说明性方法的流程图。

[0015] 图 9 是依据在本文中描述的原理的一个示例的用于数据转储 / 恢复的方法的流程图。

[0016] 图 10 是依据在本文中描述的原理的一个示例的用于误处理的文件恢复的方法的流程图。

[0017] 贯穿附图,同样的参考标记指定类似但不必同样的元素。

具体实施方式

[0018] 固态驱动器(SSD)是非易失性存储装置,其使用集成电路诸如 NAND 闪存存储器来存储数据。SSD 具有许多优势,诸如高的抗冲击、低功率要求、更快的访问时间、和更可变的因子。然而,被用作固态驱动器中的存储器的集成电路具有有限的寿命。NAND 闪存的典型规范指定 NAND 闪存在故障之前仅能够被可靠地使用 1000-3000 个写入 / 擦除周期。该寿命限制特别麻烦,因为在当前的体系结构中 NAND 闪存的块每当该块所含有的数据的任何部分被改变时必须被擦除和重新写入。因此,使用 SSD 驱动器越频繁,它将出现故障越快。许多操作系统频繁地写入到非易失性存储器。例如,文件存取表(FAT 表)每当文件改变时被重新写入。每个 FAT 表更新包含多个擦除 / 写入周期。此外,许多操作系统周期地保存计算装置的当前状态的“快照”到非易失性存储器中。虽然这能够有益于恢复计算装置的操作,但是例行地保存大的快照到 NAND 闪存上能够显著地缩短 SSD 的寿命。因此,SSD 可能无法满足消费者预期并且可能要求频繁的取代。

[0019] 下面描述许多原理,其允许闪存存储器被有效地用作非易失性存储,尽管其有限数目的擦除 / 写入周期。下面描述的固态驱动器(SSD)体系结构通过创建 DRAM 逻辑闪存来充当闪存存储器之间的中间物并且然后当数据应当被写入到 NAND 闪存存储器时独立地存取来解决 NAND 闪存存储器的限制。这显著地改进 SSD 的操作速度和寿命,并且允许 SSD 被用作硬盘驱动器的即插即用备选。

[0020] 计算装置内的数据使用典型地分成两类:数据的创建 / 操纵期间的高的使用量以及然后当数据被存档或存储为运行程序时极较低的使用量。说明性 SSD 分离存储与计算装置的瞬时状态相关的数据的过程和闪存的永久存储能力。

[0021] 当计算装置掉电时,由计算装置的易失性存储器存储的数据丢失。下面描述的 SSD 通过允许在程序或数据文件的开发期间存储数据并且当计算装置掉电时进行保护免于数据丢失来促进数据文件的创建。

[0022] SSD 包含管理最优数目的闪存存储器装置的几个闪存接口控制器。在像 USB2 装置的简单系统中,一个中间控制器能够被用来直接管理闪存。然而,在高速系统中,几个控制器能够被并行操作来更迅速得多地管理数据。下面描述的原理也能够被应用到各种各样的总线和装置技术,包含 SATA 3 (每秒 500 兆字节)、USB 3.0 “超高速”装置,包含 USB 3.0 固态驱动器和存储装置。USB 3.0 规范指定每秒上至 4.8 吉位的传送速率、增加的最大总线功率和更高效的功率管理。

[0023] 在下面的描述中,出于解释的目的,许多特定细节被阐述以便提供对本系统和方

法的透彻理解。然而,本设备、系统和方法可以在没有这些特定细节的情况下实践对本领域技术人员来说将是显而易见的。在说明书中对“示例”或类似语言的引用意味着结合示例描述的特定特征、结构、或特性在至少那一个示例中但不必在其它示例中被包含。

[0024] 在下面的几个实例中,描述控制器,该控制器包含至少一个微处理器、只读存储器(ROM)和随机存取存储器(RAM)。微处理器、ROM 和 RAM 一起工作以实施控制器的功能。具有不同控制和 / 或硬件实施方式的不同微处理器的使用能够被用来实施在本文中描述的原理。

[0025] 图 1 示出包含逻辑闪存和非易失性闪存存储器的固态驱动器的一个实施方式。逻辑闪存包含易失性存储器的片段和独立的控制器。逻辑闪存被配置成实施闪存控制器的所有功能,使得中央处理单元(CPU) 当它正在实际读取和写入到逻辑闪存时将认为它正在读取和写入到闪存。逻辑闪存实施闪存存储器的行为,但没有闪存存储器的寿命、速度、或寻址限制。逻辑闪存以与闪存存储器相同的方式来存储文件并且响应于闪存的命令。进一步,逻辑闪存使用 FAT 表、更新逻辑记录、组合文件、并且被附着到 SATA 3 总线。因为逻辑闪存的易失性存储器具有实质上无限数目的读 / 写周期,所以系统处理器和操作系统能够存储与期望一样多的更新和快照。进一步,逻辑闪存存在读取和写入数据两者上是极其快速的。CPU 从逻辑闪存读取并且专有地写入到逻辑闪存,而写入和读取到闪存存储器由固态驱动器来专有地控制。逻辑闪存的使用允许所有闪存的命令以全接口速度来处理,并且最小化到闪存存储器的写入。这不同于高速缓存,因为高速缓存最终将所有东西写入到闪存存储器,并且仅被实施来增加速度和处理短的读取和写入。

[0026] SSD 内的主控制器独立地确定何时数据应当被传送到闪存存储器或从闪存存储器来传送。这显著地减少闪存存储器的写入 / 擦除周期的数目,因为 CPU 不直接访问闪存存储器。

[0027] 闪存存储器包含许多闪存存储器模块。每个闪存存储器模块包含独立的控制器和许多闪存管芯。通过使用独立控制器,SSD 能够并行地执行多个操作。这导致显著更快的读取和写入时间。

[0028] 下面段落描述用于开发 SSD 的各种原理,该 SSD 结合逻辑闪存和多个控制器。SSD 当前每吉字节存储比硬盘驱动器更昂贵。这主要由于被用来存储 SSD 中的数据非易失性存储器管芯的成本引起。存储器管芯典型地是闪存存储器,虽然已经提出其它类型的存储器,包含铁电随机存取存储器(FeRAM)、磁阻随机存取存储器(MRAM)、可编程金属化单元(PMC)、相变存储器(PCM)、以及其它技术。非易失性存储器类型中的这些类型中的每个具有优势和劣势。然而,闪存存储器是最成熟的技术并且具有每单位存储容量的最低成本。存在两种主要类型的闪存存储器:NOR 类型和 NAND 类型。NOR 和 NAND 闪存两者都将数据存储于由浮置栅晶体管制成的存储单元中。这些浮置栅晶体管在耗损开始恶化存储的完整性之前具有有限数目的程序擦除周期。例如,NOR 闪存存储器可以具有 100,000 周期的典型耐受等级,并且 NAND 闪存存储器可以具有 1,000 至 3000 周期的典型耐受等级。

[0029] NOR 类型闪存存储器允许独立地写入和 / 或读取单个字节。然而,该随机存取特征使 NOR 存储器每单位面积不太密集并且每单位存储更昂贵。NAND 类型闪存非常高密度并且具有每单位存储的对应更低的成本。然而,在当前芯片体系结构中,NAND 类型闪存必须被读取和编程在被称为块的更大片段中。该限制是重要的,因为变更块中的单个位要求块中

的整个写入空间的擦除和重新写入。出于解释的目的,NAND 类型闪存将被用在固态驱动器体系结构的说明性示例中。然而,在本文中描述的原理能够被应用到各种各样的非易失性存储器类型。

[0030] 如上面讨论的,NAND 类型闪存是不昂贵并且紧凑的,但具有以下劣势:在耗损开始恶化存储的完整性之前具有有限数目的程序擦除周期。该挑战被以下事实加重:虽然 NAND 类型闪存能够以位级来读取,但是 NAND 类型闪存必须以大的片段(“块”)而不是以位级来写入和擦除。因此,当块中的任何位改变时,块中的所有数据必须被拷贝到新的块。在拷贝过程期间,(一个或多个)新的位被结合到存储在新的块上的数据中。旧的块然后被擦除并且被再次使用。许多计算装置上的程序和操作系统频繁地读取和写入到硬盘驱动器,这可能导致 NAND 闪存的迅速退化。在这些工业标准的操作中,改变块中的甚至一个位要求整个块的拷贝和擦除。在下面的讨论中,描述原理,该原理提供保持块直到它是满的并且仅正更新已经被写入的页面。

[0031] 在一些 Apple® 操作系统中,用户的文件被连续地写入到硬驱动器以允许用户将机器恢复到先前状态。不仅系统恢复到最近状态,而且存在被称为“时间机”的程序,其允许系统被恢复到数月之前的任何先前状态。该程序压缩快照并且允许恢复到天但不是在那天期间的任何时段。然而,快照能够被维持使得恢复到先前几天的特定点是可能的。该时间机特征能够在恢复被误处理或丢失的文件上非常有用。恢复到犯错误之前的时间允许文件和系统状态的完全恢复。

[0032] 这些和其它频繁写入操作能够导致闪存存储器的早期故障,因为有限量的写入/擦除周期能够很快地被超过。每个新的写入要求旧的数据拷贝到新的块以添加新的数据。如上面讨论的,NAND 存储器中的每个存储器位置仅能够在基本上没有增加故障的可能性情况下被更新大约 1,000 至 3,000 次。存在尝试解决该问题的许多算法,诸如向存储器过供给备件和耗损均匀化算法,其试图将耗损均匀地扩展在整个闪存存储器上面而不是将其集中在相同块中。然而,这些技术可能增加成本并且降低固态驱动器的性能。

[0033] 下面示例描述各种固态驱动器(SSD)体系结构、方法和原理。这些 SSD 结合闪存存储器用于非易失性存储,并且被设计成具有比常规 SSD 长一个数量级的寿命以及以全总线速度操作,而不管闪存存储器的限制。

[0034] 说明性闪存存储器模块

图 2 是说明性闪存存储器模块的图示。如上面讨论的,闪存存储器是非易失性计算机存储,其能够被电擦除和重新编程。如上面讨论的,闪存存储器具有高的抗机械冲击、小的占用空间、可与动态随机存取存储器(RAM)相比的相对快速的读取时间,是能量高效的并且能够在没有功率的情况下存储数据一些年。闪存存储器被用在各种应用(包含个人计算机、移动装置、数字照相机、视频游戏、科学仪器、工业机器人、医学电子设备和其它装置)中。闪存存储器具有几个限制,包含慢的写入时间和有限的寿命。对于闪存存储器而言,写入次数典型地比读取次数大一个数量级。各种类型的闪存存储器的寿命典型地范围从 1000 到 3000 个擦除周期。闪存存储器的擦除引起递增的损害,其最终导致闪存存储器中的存储机构的故障。

[0035] 在图 2 中示出的说明性闪存存储器模块包含许多 NAND 闪存管芯。存储器控制器包含处理器、小量的随机存取存储器(RAM)、小量的只读存储器(ROM)、以及许多存储缓冲

器。该存储器控制器的示例由以下给出:Charles I. Peddle 的标题为“High Speed USB Controllers”的代理人案号 034901-303891 的美国专利申请号 61/774,175,其上面通过引用整体地结合在本文中。例如,存储器控制器可以基于具有 20 千字节的处理器可寻址 RAM、存储操作代码的 40 千字节的 ROM、以及八个存储缓冲器的 6502 处理器。存储器控制器从外部总线接受数据、在缓冲器中积累数据、并且将数据写入到 NAND 闪存管芯。外部总线可以是 USB 总线或高速内部总线。

[0036] 存储器控制器也包含高速直接存储存取(DMA)和闪存 DMA。大体上,DMA 协议包含地址计数器,其在数据传送期间自动和逐步地递增存储地址。DMA 协议也包含保持传送的字节的数目的追踪的计数器。为了开始 DMA 传送,两个命令被给出:起始的存储位置和告诉 DMA 要传送多少字节的计数。DMA 独立地传送起始于指定的存储位置的数据直到计数被用尽。DMA 协议的目的是允许全速传送到存储器和从存储器全速传送而不需要除了存储器时钟和使能以外的外部输入。这整体地消除要求微处理器直接涉入数据传送。这能够实现更高的传送速度,因为数据传送不被微处理器速度限制或者当 MPU 被重新指导向不同任务时被中断。

[0037] 在该申请中,存在具有不同功能性的两个独立的 DMA。高速 DMA (“总线 DMA”)控制数据从高速总线传送到一系列的存储缓冲器,并且闪存 DMA 将数据传送到闪存并且从闪存传送数据。在一个实施例中,从高速总线到存储缓冲器的数据传送是最高优先过程并且是中断驱动的。数据移动到闪存或从闪存的数据移动用探测来完成,因为过程能够以小的干扰被中断。进一步,探测生成对闪存存储器的定时信号的正控制。

[0038] 两个分离的 DMA 模块(高速 DMA 模块和闪存 DMA 模块)的使用提供几个优势。首先,通过包含两个分离的 DMA 模块,数据能够被同时地写入到存储缓冲器并且从存储缓冲器读取。此外,分离的 DMA 模块能够不同地操作并且被不同地控制以促进数据传送。例如,高速 DMA 可以正在高速时钟上操作并且将数据写入到一个存储缓冲器同时闪存 DMA 正在以更慢的速度从不同存储缓冲器中读出数据。相反,闪存 DMA 可以在闪存时钟上操作并且通过探测来操作。此外,闪存存储器模块生成、存储、和使用纠错码(ECC)来自动恢复具有由于写入和/或存储失败引起的有限数目的错误的错误的数据。除了的高速总线上接收的数据以外,闪存存储器模块也将附加的信息(包含损耗数目、逻辑记录数目、更新数目、和其它数据)写入到闪存存储器。该过程下面更详细地被描述。寄存器能够以各种时钟速率来运行并且在各种功能之间切换。

[0039] 上面给出的结构和体系结构只是闪存存储器装置的一个示例。能够使用各种其它结构。例如,更大的存储缓冲器、更大的扇区尺寸、更多存储缓冲器、不同数目的存储缓冲器和不同数目的闪存管芯能够被包含在体系结构中。

[0040] 说明性固态驱动器体系结构

图 3 示出 SSD 体系结构的说明性示例,该 SSD 体系结构结合许多闪存模块,诸如上面关于图 2 描述的那些。SSD 体系结构能够支撑 SATA 数据速率并且缓和 NAND 闪存存储器的有限寿命。在该示例中,SSD 外部的中央处理单元(CPU)被连接到 SATA 总线。SSD 驱动器经由 SATA 总线接受数据输入、命令、和输出数据。该输出数据被初始地存储在 DRAM 逻辑闪存中。DRAM 逻辑闪存包含 DRAM 控制器和大量的动态随机存取存储器(DRAM)。DRAM 逻辑闪存被连接到高速内部总线。除了连接到 DRAM 逻辑闪存以外,高速内部总线被连接到一系列的

闪存存储器装置和主控制器。在一些实施例中,可以存在控制高速内部总线的操作的分离的总线控制器。可替代地,主控制器和总线控制器的功能性能被组合,使得主控制器执行总线控制器的功能。高速内部总线允许任何这些实体之间的双向通信。总线控制器(或充当总线控制器的主控制器)独立地选择将要接收或传送数据的装置。这允许针对每个装置个别地(并且在一些示例中顺序地)控制数据流。例如,DRAM 控制器能够在总线控制器正在从 DRAM 中向外并且到闪存存储器装置中传送数据时接受来自 SATA 总线的数据。这些同时操作允许更高效的操作和更高的总体吞吐量。闪存操作能够具有时间变化,所以由管理高速总线和与逻辑闪存控制器协调的主/总线控制器来完成数据的最后同步。该平衡方案允许 SATA 接口或其它接口在读取或写入到 SSD 时以全总线速度运行。

[0041] 控制器

SSD 体系结构使用许多控制器来管理内部数据流。主控制器从计算装置的中央处理单元接收指令,并且管理固态闪存驱动器的操作以执行指令。主控制器指导总线、每个闪存存储器装置中的闪存存储器控制器、和逻辑闪存控制器的操作。在一个实施方式中,每个这些控制器是如期望的那样简单的微处理器系统。依据一个说明性示例,控制器(主控制器和可选总线控制器、DRAM 控制器、八个闪存控制器)中的每个是具有其自己的微处理器、用于存储代码的 ROM、RAM、和一系列寄存器的完全独立的系统。例如,控制器可以是基于与 32 千字节的 RAM 和 24 千字节的 ROM 组合的 6502 处理器。逻辑闪存控制器通过控制 DMA 传送并且与逻辑闪存控制器对接来管理到 DRAM 中和从 DRAM 中向外的数据传送。逻辑闪存控制器在主控制器的指导下管理 DRAM 逻辑闪存。主控制器管理 DRAM 与闪存存储器之间的数据的传送。个别的闪存控制器处理闪存存储器的页面模式结构、错误纠正、和耗损均匀化。每个闪存存储器装置中的存储器控制器管理高速内部总线与 NAND 闪存管芯之间的数据的传送。

[0042] 多个内部控制器的使用提供许多益处。控制器能够执行专用的功能,该专用的功能特别适配于控制器在灵活地与其它控制器协调时正在控制的装置。例如,存储控制器可以以第一时钟速度与高速总线对接,并且然后管理以不同时钟速度正被写入到 NAND 闪存管芯的数据。此外,存储器控制器可以当它们已经完成任务时发送信号给主控制器。这允许主控制器智能地分配资源以最大化数据传送速率。

[0043] 直接存储存取接口

直接存储存取(DMA)接口管理被连接到总线的每个控制器的数据的传送。如上面讨论的,DMA 是硬件实施的协议,其允许计算机内的硬件子系统独立于控制器访问系统存储器。控制器能够发起传送,在传送在进行时做其它工作,并且一旦传送完成就接收来自 DMA 控制器的反馈。例如,SATA DMA 处理从 SATA 总线到 DRAM 逻辑闪存的数据传送。总线 DMA 处理 DRAM 逻辑闪存与高速内部总线之间的数据传送。类似地,高速内部总线与每个闪存存储器装置之间的 DMA 接口管理到闪存存储器装置中和从闪存存储器装置中向外的数据传送。

[0044] 使用 DMA 技术维持写入闪存和传送数据到接口/从接口传送数据两者的速度。如上面讨论的,DMA 协议包含地址计数器,该地址计数器在数据传送期间自动和逐步递增存储地址。DMA 协议的目的是在没有除了存储器时钟和使能以外的外部输入情况下允许跨接口的全速传送。这整体地消除要求微处理器直接涉入数据传送,并且能够实现更高的传送速度,因为数据传送不被控制处理器限制或当控制处理器被重新指导向不同任务时被中断。

[0045] 为了开始 DMA 传送,控制处理器可以加载具有地址、用于 DMA 操作的数目的计数和

其它使能功能的控制寄存器。数据传送然后根据控制寄存器中的参数发生。DMA 可以被配置成使得可以在传送期间添加其它数据,诸如纠错数据、逻辑记录、和管家功能。DMA 协议能够触发各种响应以给控制处理器发送信号表示数据传送完成或者提供状态更新。这允许 DMA 传送一完成就存取数据。此外,使用中断来信号通知数据传送的状态允许探询 SSD 内的多个存储器存储组件之间的数据的样式并行分布。

[0046] DRAM 逻辑闪存

DRAM 逻辑闪存中的 DRAM 使用电容器阵列来存储数据。电容器可以被充电或被放电。这两个状态表示位的两个值。因为电容器泄漏电荷,所以电容器的状态最终消失,除非电容器电荷被周期地刷新。该刷新发生在大约 10 至 100 毫秒的间隔之上。DRAM 是非常简单的,具有可忽略的读 / 写周期耗损、并且能够非常密集地被包装在管芯上。此外,DRAM 提供极其快速的写入和读取时间(大约 10 至 100 纳秒)。DRAM 的操作被 DRAM 控制器控制。在该示例中,DRAM 具有双倍数据速率类型三同步动态随机存取存储器(DDR3 SDRAM)的 8 吉字节的总容量。在其它实施方式中,DRAM 可以具有更大(例如,16GB 吉字节)或更小量的存储。对于功率管理而言,DRAM 能够以 800 兆赫兹的时钟速度操作。然而,任何适合的时钟速度和 DRAM 量能够被包含在设计中。DRAM 逻辑闪存以与闪存相同的方式来存储文件并且响应于闪存命令。进一步,DRAM 逻辑闪存使用文件分配表、更新逻辑记录、组合文件、并且被附着到 SATA 总线。

[0047] 出于许多原因,DRAM 逻辑闪存不是高速缓存。例如,高速缓存是供 CPU 寻找数据的备选位置。如果数据不在高速缓存中,则 CPU 访问下面的非易失性存储器。相反,DRAM 逻辑闪存是对于 CPU 直接可访问的 SSD 中的唯一存储器。实际的 NAND 闪存在主控制器的控制下并且对于 CPU 不可直接访问。DRAM 逻辑闪存充当 CPU 与 NAND 闪存之间的看门者(gatekeeper)。通过将 NAND 闪存与 CPU 指令分离,NAND 闪存不经受操作系统的许多特性,包含频繁写入。这允许操作系统在没有修改的情况下运行同时保护 NAND 闪存的寿命。

[0048] 数据和文件仅被存储到 DRAM 逻辑闪存直到被删除或观察不到活动。大体上,DRAM 逻辑闪存中的数据由逻辑记录来组织用于用户控制数据,并且通过 FAT 表来引用以控制各种数据记录的操作。然而,在一些实例中,DRAM 逻辑闪存可以在没有使用逻辑记录的情况下接收、存储、和传送数据。例如,下面描述的快照和快速加载(FastLoad)程序不使用逻辑记录。然而,所有数据传送模式(从 DRAM 逻辑闪存中向外到闪存存储器的数据移动)仅被主控制器支配。主控制器可以基于许多因素(包含没有使用文件)作出关于何时将数据或文件从 DRAM 逻辑闪存中移出的决定。由主控制器接收的 CPU 命令可以对将数据移动到闪存中 / 从闪存中移出的主控制器具有一些影响,但是正是主控制器实际做出关于从中取回数据或写入数据的决定。例如,如果 CPU 请求数据文件,则主控制器确定文件是否在 DRAM 逻辑闪存中。如果请求的数据不在 DRAM 逻辑闪存中,则主控制器从闪存取回它并且将它存储在 DRAM 逻辑闪存中。

[0049] 在一些实例中,文件和 / 或数据可以仅被存储在 DRAM 逻辑闪存上并且从不被传送到闪存存储器。例如,暂时数据文件可以被创建用于瞬时操作(诸如搜索)。在其它示例中,文件可以被创建用于将被发送到另一个系统或被远程系统存储的信件或电子邮件。当文件被发送到远程系统时,文件能够被删除。

[0050] 高速缓存在 CPU 看来正好具有实际存在于高速缓存中的物理存储的量。相反,

DRAM 逻辑闪存看来具有远大于组成 DRAM 逻辑闪存的存储的物理容量的容量。DRAM 逻辑闪存看来具有等同于 NAND 闪存的总工作非易失性存储的容量。

[0051] 高速缓存在 CPU 看来是易失性存储器。相反, DRAM 逻辑闪存看来是极其快速的非易失性存储器。当 CPU 将数据写入到高速缓存时, CPU 假定数据实际上不在非易失性存储中。CPU 继续管理数据流直到数据实际上被存储在跟随高速缓存的非易失性存储中。当高速缓存非预期地失去功率时, 高速缓存中的数据失去并且 CPU 必须在没有它的情况下恢复。所有高速缓存事务失败或被写入到非易失性闪存存储器, 从而增加耗损并且延迟系统。

[0052] 相反, CPU 和操作系统假定 DRAM 逻辑闪存是非易失性存储器存储。DRAM 逻辑闪存报告写入到它的数据被存储在非易失性闪存存储器上, 甚至经过它实际上被存储在 DRAM 逻辑闪存中。当 SSD 失去功率时, CPU 正确地假定存储在 DRAM 逻辑闪存中的数据被存储在非易失性存储器中。这是正确的, 因为 SSD 具有用于将 DRAM 逻辑闪存中的数据转储到 NAND 闪存的自包含并且自供电的系统。在一个实施方式中, NAND 闪存被配置成额外供应备件以适应能够被存储在 DRAM 逻辑闪存中的所有数据的数据转储。

[0053] 高速缓存被设计成最小化对存储在更慢的存储器中的数据的访问时间。在典型的高速缓存操作中, 高速缓存尽可能快地将数据写入到非易失性存储, 但继续保持写入的数据以最小化访问时间。相反, DRAM 逻辑闪存被设计成最小化对下面的存储器的写入。SSD 中的主控制器仅将不正被用于从 DRAM 逻辑闪存传送到闪存存储器的数据作为目标。

[0054] 高速内部总线

如上面讨论的, 高速内部总线允许与其连接的任何这些组件之间的双向通信。在一个示例中, 主控制器个别地经过高速内部总线将数据指导向存储器控制器。为了实施向闪存的写入传送, 逻辑闪存控制器 / 接口将 DRAM 逻辑闪存连接到高速内部总线并且使用 DRAM DMA 实现到指定的文件位置的传送。使用该技术, 数据能够经过 DRAM 逻辑闪存从 CPU 直接地传送到闪存存储器。例如, 高速内部总线可以是 8 个位宽的并且能够以每秒至少 400 兆字节 (MB) 的速度操作。经过 8 位总线以 400 兆赫兹 (或更高) 操作的数据传送速率将会是每秒近似 400 兆字节。

[0055] 闪存存储器装置

如上面关于图 2 讨论的, 每个闪存存储器装置包含存储器控制器和许多 NAND 闪存管芯, 其组成闪存存储器。闪存存储器被划分成扇区、页面、块和平面。在该示例中, 扇区是具有用于报头和纠错码 (ECC) 信息的附加空间的近似 512 字节。在其它实施方式中, 扇区可以更大。页面是一组扇区、块是一组页面、并且平面是页面的收集。在一个示例中, 页面包含用于数据的 8192 字节和用于报头信息的附加空间。块可以是一组 256 个页面并且平面是一组 2096 个块。装置可以包含任何数目的平面。例如, 32 吉字节装置可以包含 2 个平面或 8192 个块。256 吉字节装置可以包含 16 个平面或 65, 536 个块。典型地, 当非可恢复或可修复的闪存数据错误发生在扇区中时, 整个块被标记为坏的。然而, 使用页面模式控制器, 仅损坏的页面被标记为坏的并且被四处映射。这进一步被描述在 Charles I. Peddle 的美国专利号 8, 122, 319 中, 其整体通过引用被结合于此。

[0056] 如上面讨论的, 闪存存储器的整个块当块中的页面中的一个中的单个位不可操作时传统地被认为不可用。因此, 缺陷位可以以 128 KB 或更多来减少闪存存储器的存储容量。当多个缺陷位被分散在许多块当中时, 闪存存储器可能无法满足容量标准并且可能被

丢弃。然而,许多完整地功能页面仍然在每个故障的块内。如下面示出的,通过识别不可操作的页面而不是不可操作的块,闪存存储器的许多存储容量可以被回收。

[0057] 各种命令被用来访问闪存存储器。例如,对闪存存储器的读取和写入命令可以操作在单个页面上。然而,擦除命令影响整个块。除了块擦除操作以外,几乎所有操作可以被执行在单个页面上。一旦块中的页面被擦除,它们可以以避免不可操作页面的方式而被选择性写入。

[0058] 虽然闪存存储器本身可以不包含仅选择块内的可操作页面的逻辑,但是存储器控制器可以被配置成识别、选择、和操作在仅可操作的页面上。存储器控制器可以被实施为与闪存存储器分离和不同的半导体芯片。存储器控制器将数据到闪存存储器和从闪存存储器的传送进行协调。存储器控制器通过发送适当命令和存储地址到一个或多个闪存装置来处理来自外部装置的请求。依据一个实施例,存储器控制器可以生成芯片选择、块选择、行选择、和列选择信号以传输到一个或多个闪存存储器。存储器控制器也可以监控到闪存存储器装置和从闪存存储器装置的数据传送的控制信号、状态信号、定时约束、和其它方面。

[0059] 存储器控制器可以将来自外部系统的虚拟存储地址(诸如,逻辑记录)转化成一个或多个闪存存储器装置上的物理地址。存储器控制器可以从处理器接收请求某些数据的查询。作为响应,存储器控制器可以确定其中请求的数据被物理地存储在一个或多个闪存存储器装置中的对应的块、页面、和字节。存储器控制器然后将存储地址值和控制信号的正确序列发给闪存存储器装置以取回请求的数据。

[0060] 类似地,存储器控制器可以将写入请求转化成块擦除、地址选择、和写入命令的适当序列以将数据存储到闪存存储器装置上。实际上,存储器控制器可以允许各种系统和组件访问闪存存储器装置的存储器同时隐藏与闪存存储器装置的页面模式接口的复杂性。例如,当闪存存储器装置中的先前写入的数据被更新时,旧的数据以及新的数据被写入到新的块并且旧的块被擦除。存储器控制器可以生成并且执行操作的正确序列以执行存储操作。存储器控制器也可以识别哪些块含有足够数目的可操作页面来完成操作。在数据从源块被传送到目的块的情况下,目的块被选择以至少含有与源块相同量的存储容量,但是目的块可以仍然包含一个或多个不可操作的页面或扇区。

[0061] 为了追踪每个块内的可操作的页面的数目,存储器控制器可以建立“好的页面”表、“坏的块”表、具有针对存储器的每个页面的“好的”或“坏的”指定的表、或其它指示符。“坏的块”表可以识别不可操作的页面并且因此间接识别可操作的页面。存储器控制器或其它元件然后可以被配置成读取和写入到任何页面,除了被列为不可操作的那些以外。可操作页面的指示可以包含一个或多个参考、指针、地址、表、列表、集、标识符、标记、符号、记号、代码、或等式、或可以允许识别可操作页面的其它信息。

[0062] 在一个实施例中,可操作页面的表可以被存储在闪存存储器的指定的一个或多个块中。例如,通过存储器控制器对整个闪存存储器装置的彻底测试可以发生在指示不完整、不可读、缺失、或损害时。该类型的测试可以发生在存储器控制器和/或闪存存储器装置第一次通电时。例如通过纠错码(ECC)模块的附加测试可以在检测在使用期间故障的页面的闪存存储器装置的操作期间被执行。在闪存存储器操作期间使用的错误检测方法可以包含但不限于生成校验和、比较校验和、执行冗余校验、生成奇偶值、执行奇偶校验、以及执行其它错误检测算法。如果在页面中检测到故障,则 ECC 模块可以警告闪存控制器故障发生或

进行的操作不成功。闪存控制器然后可以在新的页面中重复操作或者另外纠正错误。如果页面具有可恢复可重复的错误,则那个页面被丢弃。主控制器然后采取适当动作通过它们在表中的指定来排除这些页面。从此以后,缺陷的页面不被使用。

[0063] 当一个或多个指示被更新时,内部操作和数据传送可以被完成以向访问闪存存储器装置的系统并且最终向闪存存储器装置的人类用户隐藏故障和重新配置。因此,故障将不干扰用户的总体体验,并且将不要求通过外部系统补偿。依据一个实施例,这可以用可以在初始化、测试、或其它阶段期间保留的备用块、页面、和 / 或扇区来完成。当故障发生时,用于故障的块、页面、和 / 或扇区的数据和地址可以被备用块、页面、和 / 或扇区来取代。一个或多个指示然后可以被更新以反映针对数据的新的逻辑存储地址和物理存储地址。

[0064] 总的来说,闪存存储器控制器中的基于页面的故障管理允许存储器控制器访问闪存存储器块内的“好的页面”表或每个页面的功能性的其他指示符。存储器控制器然后能够利用每个块中的可操作页面来执行读取、写入和擦除命令,即使块含有一个或多个不可操作的页面。页面模式的使用允许闪存存储器的寿命的显著延长。进一步,页面模式的使用允许具有更低寿命级和 / 或更高数目的错误的闪存存储器的更高效使用。这些芯片能够被有效地使用并且在实施如上面描述的页面模式故障管理的装置中具有延长的寿命,而不是丢弃具有错误的这些闪存存储器芯片。

[0065] 存储器控制器使用 DMA 协议接受来自高速内部总线的数据、在其内部缓冲器中积累数据并且将数据写入到 NAND 闪存管芯。每个闪存存储器模块被配置成提供到闪存管芯和从闪存管芯的每秒近似 40 兆字节的数据传送速度。这些并行闪存存储器模块可以具有许多配置,包含被描述在 Charles Peddle 的标题为“High Speed USB Controllers”的代理人案号 034901-303891 的美国专利申请号 61/774, 175 中的那些,该申请通过引用整体被结合于此。例如,可以存在并行的八个闪存存储器模块。在一个实施方式中,每个闪存驱动器包含四个闪存管芯。每个闪存管芯包含 8 吉字节的存储,从而导致 256 吉字节的总的闪存存储。这些驱动器被配置成并行操作,从而提供近似每秒 320 兆字节的传送速率用于数据写入。从闪存存储器读取数据显著地快于写入数据到闪存存储器。因此,闪存存储器模块可以在读取操作期间展示对应更高的数据传送速率。

[0066] 在 CPU 存储器、DRAM 逻辑闪存、以及闪存存储器之间移动数据

图 4 是示出被用来在 CPU 存储器、DRAM 逻辑闪存、以及闪存存储器之间传送数据的各种方法的概观的图示。用于传送数据的每种方法被更详细地描述在下面描述和附图中。

[0067] 在图 4 中示出的系统包含可操作地连接到 CPU 存储器的 CPU。如在图 3 中示出的, CPU 和 CPU 存储器通过系统总线 (SATA 总线) 被连接到固态驱动器中的 DRAM 逻辑闪存。DRAM 逻辑闪存通过高速内部总线被连接到闪存存储器。

[0068] 如上面讨论的, DRAM 逻辑闪存与闪存存储器之间的数据移动由主控制器来独立地管理。主控制器可以对从 CPU 接收的命令做出反应,但 CPU 不直接控制到闪存存储器的传送并且不知道这些传送发生。

[0069] 系统能够实施存储器之间的各种数据传送以实现特定目标。大体上,计算装置发送关于被称为文件的数据收集的命令。命令是相当简单的:读取该文件、写入该文件、或更新现有的文件。命令来到 SSD 作为 SATA 命令,其被主控制器解释。来自外部总线的数据以全速度被流传送到逻辑闪存中,并且逻辑闪存控制器被指导以存储或取代关联的数据文件

的先前版本。外部总线可以是 SATA 总线、USB 总线、或任何其它适当协议或技术。当计算装置想要读回文件或文件的部分时,读取的命令被初始地给予逻辑控制器,该逻辑控制器被指导以从存储在其存储器中的数据检索期望的数据。如果数据不在 DRAM 逻辑闪存中,则它在主控制器的指导下从闪存装置被存储在那里并且然后以高速传送到计算装置。该数据被维持在 DRAM 逻辑闪存中,因为它很可能被更新和重新加载。

[0070] 图 4 示出五个不同的数据传送技术的说明性示例。第一技术是在图 5A、5B、6A、和 6B 中更详细地描述的使用逻辑记录的数据传送。处理器处理逻辑记录,而控制器处理物理记录。转化表被用来转换 / 索引逻辑记录与物理记录之间的关系。被用于 SSD 的转化表不同于被用在硬盘存储系统中的转化表。该数据传送技术使用逻辑记录按 CPU 指导从 CPU 存储器索引和存储数据到 DRAM 逻辑闪存的工作区域,并且随后在主控制器的指导下存储不正被使用的数据文件到闪存存储器的工作区域。主控制器也指导从闪存存储器中的工作区域中到 DRAM 逻辑闪存的工作区域的数据的取出用于由 CPU 存储器直接取回。

[0071] 第二数据传送技术是快照流,其允许系统状态的快照被周期地保存使得系统能够被恢复到先前状态。该技术被更详细地描述在图 7 中。在该实施方式中,数据在没有使用逻辑记录的情况下被传送到 DRAM 逻辑闪存内的指定快照区域。其它技术(诸如表)能够被用来索引和恢复快照数据。在一个示例中,快照按时间索引。最近的快照在恢复计算装置的状态中是最有价值的。更早的快照在使用系统期间被新的快照重写。这存储最近的快照用于在未来恢复操作中使用。快照被存储在 DRAM 逻辑闪存并且直到掉电才被传送到闪存存储器中。在掉电时,仅选择的数目的快照被存储到闪存存储器作为转储过程的部分。这能够通过减少对闪存存储器进行的写入的数目来显著增加闪存存储器的寿命。相反,Apple 笔记本将快照直接存储到闪存存储器。该快照操作在使用笔记本期间频繁地发生并且导致显著的闪存耗损。

[0072] 第三数据传送 / 存储技术是在图 8 中描述的快速加载过程。快速加载过程包含存储程序文件的闪存存储器中的受保护程序区域。在该示例中,闪存存储器中的受保护程序区域具有两个分区:一个用于存储的程序和一个用于活动的程序。快速加载过程由不是使用 DRAM 闪存的标准控制器的部分的附加电路和逻辑来实施。当用户获得权限来访问存储的程序时,快速加载过程能够被用来将程序文件移动到 DRAM 逻辑闪存中的受保护程序区域。从那里,CPU 能够请求程序文件被传送到 CPU 存储器用于执行。

[0073] 第四数据传送 / 存储技术是在图 9 中更详细地描述的转储 / 恢复过程。转储 / 恢复过程发生在掉电或断电时。转储 / 恢复过程将数据从 DRAM 逻辑闪存中向外移动到闪存存储器中的指定的转储区域中。为了恢复,功率被恢复到计算装置,并且转储区域中的数据被传送回到 DRAM 逻辑闪存中并且然后到 CPU 存储器。

[0074] 第五数据传送技术是在图 10 中描述的误处理的文件恢复技术。“误处理的”指代其中对文件进行非意图和非期望的改变的任何实例。例如,文件的所有或部分可以被删除,或者期望的文件可以被另一个文件重写。误处理的文件恢复技术提供 CPU 已命令要从 DRAM 逻辑闪存中删除的文件的恢复,以从 DRAM 逻辑闪存恢复。DRAM 逻辑闪存具有等待时段,在该等待时段期间误处理的文件能够仍然被存储在 DRAM 逻辑闪存中。有限的历史文件被用来索引和恢复这些误处理文件,其仍然驻留在 DRAM 逻辑闪存上。下面描述这些技术中的每个。

[0075] 虽然 DRAM 逻辑闪存被图解为 SSD 的主要物理部分,但是在一些实施方式中,DRAM 逻辑闪存可以被构建在 CPU 易失性存储器中,其中 CPU 提供 DRAM 逻辑闪存的读取、写入、和闪存操作的控制。然而,在常规的 CPU/CPU 存储器系统中,不存在维持功率同时在功率下降时将数据转储到易失性 CPU 存储器的机制。为了成功实施在本文中讨论的原理,计算装置能够具有独立的电源以允许存储在 CPU 存储器中的数据从 DRAM 逻辑闪存被转储到固态驱动器中的闪存。

[0076] 此外,DRAM 逻辑闪存可以被用来或被配置成仅执行上面描述的功能的部分。例如,DRAM 逻辑闪存可以被配置成在没有被用于逻辑记录类型数据传送的情况下提供快速加载操作或快照操作。这可以显著地简化 DRAM 逻辑闪存的操作。在其它实施方式中,DRAM 逻辑闪存可以被分割成操作上特定的部分。与微控制器相关联的易失性存储器可以被映射到快照 / 快速加载片段(区域)和逻辑记录片段(区域)中。快照 / 快速加载片段可以存储使用特殊协议被传送到闪存并且可以或可以不通过逻辑记录索引的数据。在一些情况下,DRAM 逻辑闪存的量可以被增加以容纳该映射到分离的片段 / 区域中。

[0077] 例如,非逻辑记录区域可以存储快照数据或快速加载数据。非逻辑记录区域中的数据可以或可以不被加载到闪存中。例如,如果存储器的非逻辑记录区域含有 4 个快照,其能够被用来恢复计算装置的状态,则仅一个或两个的最近快照可以在掉电时实际被保存到闪存。其它更早的快照被简单地丢弃。

[0078] 在另一个示例中,快速加载程序可以被加载到易失性存储器的非逻辑记录区域中。这些快速加载程序是可执行文件,该可执行文件典型地在正常操作期间被修改并且不被写回到闪存存储器。存在用于更新程序的备选逻辑路径。该备选逻辑路径允许软件程序师更新软件。约束和协议可以约束除了软件分发者以外的任何人访问授权的软件程序师访问该区域。这是因为存在针对不被公开给任何未授权的个人或团体的这些区域的特定命令。

[0079] 为了保护软件的拷贝,操作系统可以不允许加载到快速加载的代码拷贝到系统中的用户端口或任何其它端口。快速加载大体上是针对用户的只读功能。这防止用户对程序的修改和分配。如上面讨论的,可以存在协议中的分离命令,其仅可用于软件发行人(或其它授权的用户),使得仅他们能够更新软件。

[0080] 在其它示例中,可以存在多个分离的存储器控制器。在一个实施方式中,可以存在用于被包含在存储器装置中的标准易失性存储器的一个控制器和用于 DRAM 逻辑闪存的一个控制器。进一步,标准易失性存储器可以被保护,并且对于用户和或计算装置而言不直接可寻址或可访问。在一些实例中,标准易失性可以具有特殊约束或使用。在一些实施例中,存储器可以看来是系统易失性存储器的扩展。

[0081] 将文件写入到固态驱动器

在日常操作期间,CPU 使用相同的协议将文件写入到 SSD,它将会使用该协议将数据写入到典型的硬驱动器。例如,CPU 可以使用以下技术:使用逻辑记录写入和读取到 SSD。SSD 驱动器的内部操作独立于 CPU 操作,并且向 CPU 隐藏。如上面讨论的,SSD 驱动器接受来自 CPU 的数据,但是以克服 NAND 闪存存储器的速度和寿命限制的独特方式来内部地管理和存储数据。然而,SSD 驱动器控制 SSD 驱动器与 CPU 之间的接口,使得它完全在 CPU 看来它正在写入到硬驱动器或日常闪存驱动器。因此,SSD 是能够被用在任何各种计算装置中的即

插即用存储器存储装置,并且明显地提供较高的数据传送速率、长的寿命、以及低的功耗。

[0082] 图 5A 是示出经过 SSD 体系结构的允许极其快速的数据传送速率的说明性数据流的图示。数据经过系统总线通过 CPU 操作来初始地传送。例如,系统总线可以是 SATA 总线。数据使用 DMA 过程从系统总线传送到 DRAM 逻辑闪存。如上面讨论的,DRAM 逻辑闪存存储用于以后由 CPU 取回的数据。CPU 仅知道 DRAM 逻辑闪存,该 DRAM 逻辑闪存看来是具有闪存存储器的存储容量的极其快速的非易失性固态存储器。

[0083] 如果主控制器确定它是适当的,则主控制器决定从 DRAM 逻辑闪存中将数据写入到闪存存储器。可以在 SSD 内存在任何数目的闪存存储器模块。例如,SSD 体系结构可以包含八个闪存存储器模块。出于说明的目的,图 5A 仅示出那些装置中的四个。如上面关于图 3 讨论的,每个闪存存储器装置包含具有缓冲器的存储器控制器和许多闪存存储器管芯。例如,每个闪存存储器控制器可以控制到四个、八个、或十六个分离管芯的分布。在许多闪存存储器控制器当中的数据分布提供许多益处,包含同时写入到多个闪存管芯。这补偿在闪存存储器的当前实施方式中固有的相对慢的写入时间。主控制器和各种闪存存储器控制器的独立性允许具有最小等待时间的并行数据传送。为了能够实现并行写入和并行读取操作,文件跨各种闪存存储器管芯而被分带。在图 5A 中,这被图解为每个闪存存储器管芯中的编号的框。当文件被写入时,主控制器顺序地寻址个别闪存存储器控制器,其进而顺序地寻址它们控制的闪存管芯。这导致文件跨管芯分布并且文件的各种部分被同时写入到不同闪存管芯。当文件从闪存管芯被取回时,它能够从管芯并行地来读取。这在图 6A 中示出。

[0084] 用于将文件写入到 SSD 的说明性方法被示出在图 5B 中。在第一步骤中,CPU 发送写入命令并且放置要被写入到 SSD 的数据在 SATA 总线上(步骤 505)。写入命令被传送到主控制器,主控制器命令 DRAM 控制器接受数据并且将它传送到 DRAM 存储器(步骤 510)。DRAM 控制器可以被配置成区分命令和其它数据,并且将命令经过分离总线发送到主控制器。该示例,DRAM 控制器将写入命令发送到主控制器。当主控制器解释写入命令时,它警告 DRAM 控制器新的数据正在来临。DRAM 控制器寻找命令中的逻辑记录并且搜索其表中的逻辑记录以确定数据是否已经被含在 DRAM 逻辑闪存中。例如,当前逻辑记录可能是另一个更大文件的部分。DRAM 控制器能够通过搜索针对每个文件的开始和结束逻辑记录确定数据已经在 DRAM 逻辑闪存中。如果当前逻辑记录在针对每个文件的开始和结束逻辑记录之间,则数据当前被存储在 DRAM 逻辑闪存中并且将被重写。为了允许如上面描述的误处理文件恢复,逻辑记录数据用取代的文件的地址的记法被写入在 DRAM 逻辑闪存中的新的空间。然而,如果逻辑记录不在 DRAM 逻辑闪存中并且表示新的数据,则新的记录被创建用于新的逻辑记录并且数据被收集在 DRAM 逻辑闪存中。当写入完成时,文件分配表(FAT)被更新并且经过 SATA 接口被 CPU 发送。在一些操作系统中,每个 FAT 表项目的存储能够涉及多个、冗余写入周期。如果从 SSD 请求 FAT 表,则从 DRAM 逻辑闪存服务请求。FAT 表被存储在 DRAM 逻辑闪存中并且在掉电时仅被保存到闪存存储器。这能够将数千个擦除 / 写入周期保存在闪存存储器中。

[0085] 逻辑闪存控制器建立 SATA DMA 并且管理到 DRAM 逻辑闪存中的数据传送(步骤 515)。如上面讨论的,被用在 DRAM 逻辑闪存中的 DRAM 存储器是极其快速的随机存取存储器。DMA 传送、专用的 DRAM 控制器、和极其快速的 DRAM 存储器的组合意味着存储在 DRAM 逻

辑闪存中的数据以典型地被 SATA 总线限制的速度对于 CPU 而言容易地并且迅速地可访问。DRAM 逻辑闪存被用来存储频繁存取的数据。这使 SSD 中的闪存存储器装置与过多的写入周期隔离。逻辑闪存控制器按闪存文件来管理 DRAM 中的数据,包含使用闪存技术来合并和更新数据(步骤 520)。这允许 DRAM 逻辑闪存以与标准闪存存储器相同的方式但以高得多的速度来与 SATA 总线对接。

[0086] 不存在 SATA 数据与闪存数据之间的时间相关性。闪存存储器和存储在闪存存储器上的数据对于 CPU 不直接可访问,而被主控制器控制。CPU 仅与 DRAM 逻辑闪存对接,其中命令数据从 DRAM 逻辑闪存被传送到主控制器。逻辑闪存控制器周期地评价数据的使用,并且确定数据是否应当从 DRAM 逻辑闪存被写入到 NAND 闪存存储器(步骤 525)。例如,由 CPU 在使用中的文件可以在用户正在用文件工作的时间期间被规则地保存到 SSD 驱动器。在用户用完文件之后,文件能够在它被再次访问之前暂停活动数天或数月。存储在 DRAM 逻辑闪存中的数据在指定的保存点处被写入到 NAND 闪存存储器。例如,存储在 DRAM 逻辑闪存中的数据可以当文件被关闭时或者当计算机掉电时被传送到 NAND 闪存存储器。其它保存点可以发生在 DRAM 逻辑闪存的容量几乎被消耗完时。在这种情况下,不太频繁保存的文件能够被传送到闪存存储器。

[0087] 在主控制器的控制下从 DRAM 逻辑闪存到 NAND 闪存存储器的数据传送现在将被描述。当主控制器作出要从 DRAM 逻辑闪存将数据写入到闪存存储器装置的决定时,它将命令发送到逻辑闪存控制器,其识别要被传送的数据并且警告总线控制器数据传送(步骤 530)。主控制器将命令数据放置到内部总线上,其警告 / 启用闪存控制器使得它们能够接收 / 取回期望的数据。逻辑闪存控制器设置适当的寄存器值以配置内部总线 DMA 用于传送,并且由主控制器识别的数据通过总线 DMA 被放置在高速内部总线上(步骤 535)。主控制器(或可替代地可选总线控制器)然后开始传送具有被寻址到个别闪存控制器的特定数据片段的数据(步骤 540)。各种技术能够被用来管理经过高速内部总线的数据的传送。在一个实施方式中,被加载到内部总线上的数据包含指示数据序列的开始标志、指示数据序列的结束标志、以及识别数据被寻址到的组件的结构。每个闪存控制器监视数据流中的其标识符并且将适当数据片段转向其内部存储。在其它实施方式中,可以存在被连接到每个存储器控制器的分离命令 / 使能线。当数据旨在用于特定闪存存储器模块时,连接到该存储器控制器的使能线被断言,而针对其它存储器控制器的使能线不被断言。该配置被示出在图 3 中。

[0088] 高速总线操作在时钟上,该时钟确保来往总线的数据传送以每秒 400 MB 来执行。总线控制器指导数据从 DRAM 逻辑到闪存存储器装置以每秒 300 + MB 的全数据速率的传送。在数据传送期间,主控制器在第一 DMA 周期期间将数据顺序地指导到第一闪存寄存器,并且然后在第二 DMA 周期期间到第二闪存寄存器等等。主控制器将数据跨八个不同闪存控制器顺序地分布(步骤 545)。数据以每秒 40 MB 并行地从闪存控制器中的寄存器中被顺序读出到闪存管芯(步骤 550)。被加载的寄存器(闪存存储器缓冲器)使它们的时钟从总线速度切换到闪存速度。八个闪存控制器并行操作(针对每个以每秒 40 MB)导致每秒 320 MB 的总体传送速率。然而,额外的每秒 20 MB 允许附加的开销数据诸如纠错码(ECC)被写入到闪存存储器中。此外,可以存在许多附加操作(在闪存存储器的维护期间执行的这样的额外写入或读取)。该附加开销使针对八个并行闪存驱动器的 40 到 50 MB 传送速率近似等于

内部总线上的每秒 400 MB 的传送速率。

[0089] SSD 也可以具有许多附加特征。例如,SSD 可以被分割成不同存取和安全级的各种部分。例如,SSD 的受保护部分可以被指定用于软件可执行文件。SSD 的该受保护部分可以由用户或由操作系统直接可访问。例如,SSD 的受保护部分可以不被逻辑记录数目索引。因此,不存在用户或操作系统访问受保护部分的机制。替代地,受保护部分可以仅可用于软件供应商用于加载新的软件和更新现有的软件。受保护部分能够通过具有特殊命令的不同技术来寻址,该特殊命令特定于该类型的数据。例如,等同于逻辑记录的地址能够被使用但在不同查找表上来索引。

[0090] 为了运行含在(一个或多个)受保护部分中的软件,软件能够被传送到第二“只读”部分并且被操作系统访问。该技术优势中的一个软件可执行文件能够独立于用户正在做的事而被更新。例如,用户可以正在使用 Window®操作系统和 Microsoft Office®应用程序来编辑文档。在后台中,软件供应商可以正在推出对存储在 SSD 的受保护部分中的 Window®操作系统可执行文件的更新。用户的工作不被中断。在多数的用户情形诸如文档预备或访问互联网中,很少或没有到 SSD/ 来自 SSD 的通信业务。

[0091] 因此,新的数据能够在没有不利地影响闪存驱动器的性能情况下被流传送到 SSD 的(一个或多个)受保护部分。下一次用户启动系统,新版本的操作系统将驱动器的受保护部分被加载到“只读”部分中,并且经过 DRAM 逻辑闪存被传送到 CPU。在电源关断或故障时,不存在对尝试保存这些可执行文件的计算系统的需要,因为它们没有被改变并且已经被存储在驱动器的受保护部分上。

[0092] 此外或可替代地,可以存在被指定用于存储快照的驱动器的特殊部分。如上面讨论的,快照是时间上给定点处的计算装置的完整状态的记录。快照允许计算装置恢复到那个状态。

[0093] 从固态驱动器取回文件

图 6A 是计算装置中的读取操作的图示。如上面讨论的,CPU 经过 SATA 总线和 SATA DMA 直接与 DRAM 逻辑闪存通信以取回数据。当接收读取命令时,主控制器确定数据是否被存储在 DRAM 逻辑闪存中。如果它是,则数据从 DRAM 逻辑闪存被发送到 CPU。如果它不是,主控制器确定它在闪存存储器模块上被存储在何处并且取回它。在许多实例中,数据可以跨多个闪存存储器模块和每个模块内的多个管芯来分带。因此,数据文件可以通过同时从使用多个独立控制器的多个管芯读取数据来迅速地取回。

[0094] 图 6B 是用于从闪存存储器读取数据的说明性方法(600)的流程图。当需要新的或更新的文件时,CPU 经由 SATA 接口将具有对应于所需数据的逻辑记录数目的读取命令发送到 SSD(步骤 605)。读取命令被接收在 DRAM 逻辑闪存中并且被传送到主控制器(步骤 610)。例如,DRAM 控制器可以被配置成标识命令并且经过特殊高速总线将它们发送到主控制器,从而警告主控制器新的命令已经被接收。主控制器或 DRAM 控制器确定请求的数据是否被存储在 DRAM 逻辑闪存中(确定 612)。在一些实施方式中,DRAM 控制器追踪如读取或写入的所有文件,使得响应于来自 CPU 或主控制器的请求,它能够将正确数据发送到 SATA 接口。DRAM 控制器寻找命令中的逻辑记录并且在它的表中搜索逻辑记录以确定数据是否已经被含在 DRAM 逻辑闪存中。如果请求的数据在 DRAM 逻辑闪存中(“是”),则主控制器命令 DRAM 控制器将请求的数据放置在 SATA 总线上(步骤 614)。DRAM 配置 SATA DMA 用于数据的

传送(步骤 616)。数据被放置在 SATA 总线上并且被 CPU 接收(步骤 618)。

[0095] 如果请求的数据不被存储在 DRAM 逻辑闪存中(“否”),则主控制器将指令发送到各种闪存控制器以将数据放置在内部总线上。闪存控制器配置它们的个别 DMA 以进行从 NAND 闪存管芯到内部总线的数据传送(步骤 620)。逻辑闪存控制器配置总线 DMA 以接收数据并且将它传送到 DRAM 逻辑闪存中。逻辑闪存控制器也配置 SATA DMA 以从 DRAM 逻辑闪存中向外并且到 SATA 总线上传送数据(步骤 625)。从闪存存储器的该传送以每秒 300 到 400 兆字节的速度来进行。针对相同数据的后续请求通过 DRAM 逻辑闪存而不是从闪存存储器以全 SATA 速率来实现(步骤 630)。在从闪存存储器的数据传送之后,DRAM 逻辑闪存允许以最大的 SATA 速度(每秒从 300 到 1000 MB)来执行所有后续事务。

[0096] CPU 在程序操作中使用数据并且可以周期地将数据重新写入到 SSD(步骤 635)。逻辑闪存控制器追踪数据的变化并且合并文件,使得它一直准备好以单个写入来写入到 NAND 闪存装置(步骤 640)。如果新的文件从 CPU 被接收并且它是对当前存在 DRAM 逻辑闪存中的文件的更新,则与新的文件相关联的所有逻辑记录被写入到 DRAM 逻辑闪存中的新的位置并且新的文件被写入。使旧的数据文件的位置可用于未来写入中含有的数据。这意味着所有当前文件在 DRAM 中的一个地方,使得它们能够在掉电时被高效地存储在闪存存储器。然而,如果 DRAM 逻辑闪存中的数据没有被改变(如对于许多可执行文件的情况那样),则不需要将它写回到 NAND 闪存存储器,因为它的同样副本已经被存储在闪存存储器中。对存储在 DRAM 逻辑闪存中的数据的改变能够使用“脏位(dirty bit)”来指定。如果存储在 DRAM 逻辑闪存中的文件不被改变,则脏位仍然不改变并且文件不在保存点处重新写入到闪存存储器。如果数据在它在 DRAM 逻辑闪存中时已经被改变,则由脏位和数据指示的这个在系统掉电之前被写入到非易失性闪存存储器(步骤 645)。使用脏位来追踪存储在 DRAM 逻辑闪存中的数据的改变允许系统在 NAND 闪存存储器上保存时间和耗损。贯穿上面描述的过程,以逻辑记录级来处理所有通信。这使数据处理过程对于所有控制器和对于 CPU 均匀和透明。

[0097] 快照

图 7 是用于存储系统状态的快照的说明性方法的流程图。如上面讨论的,将 DRAM 逻辑闪存结合到 SSD 中允许系统“快照”和其它频繁写入操作在没有负面地影响闪存存储器的寿命情况下由操作系统来执行。快照不作为逻辑记录被传送/存储。如上面关于图 4 讨论的,具有分离指令的特殊路径被用于存储和取回快照。快照被存储在 DRAM 逻辑闪存中的快照区域并且仅在掉电时被传送到闪存存储器。快照不使用逻辑记录来存储。替代地,使用被称为快照记录的表。具有特殊指令的特殊路径被用于存储和取回快照。因为分离路径和指令被用于管理快照,所以快照能够在没有干扰使用逻辑记录的正常数据流的情况下来保存。大多数快照不被永久存储在非易失性闪存存储器上,而是被暂时存储在 DRAM 逻辑闪存中。当计算装置被关断或断电时,选择的快照从 DRAM 逻辑闪存被传送到闪存存储器。该传送和恢复过程由操作系统来协调。

[0098] 现在参考图 7,用户/操作系统/CPU 决定快照应当被保存(705)。计算装置的状态被保存到 DRAM 逻辑闪存上的受保护区域(“快照区域”)。操作系统做出关于哪个数据被保存的决定。例如,没有理由包含先前已经被保存到文件的快照中的数据。快照传送不使用逻辑记录来执行并且数据不使用逻辑记录或块查找表来存储。快照区域中的存储有数据的表(“快照记录”)被用来索引/组织各种快照。典型地,通过拍摄快照的时间来组织快照

(710)。

[0099] 当 CPU 发送关断命令或者电源电路感测到功率损失并且发送中断时,快照保存序列作为优先将最近两个快照存储到闪存存储器中。该过程也不使用逻辑记录或块查找表。指定的块作好准备来接收快照数据。关于快照的组织 and 如何取回它们的相关数据被存储在具有快照的表中(715)。

[0100] 当系统恢复 / 重启时,快照恢复代码从闪存到 DRAM 逻辑闪存并且回到 CPU 存储器中取回快照(720)。计算装置然后能够从快照保存的状态重新恢复操作。

[0101] 快速加载操作

图 8 是用于快速加载操作的说明性方法(800)的流程图。程序在制造 / 配置期间被存储在 SSD 中的闪存存储器的受保护程序区域的存储部分中。用户不能够直接访问受保护程序区域(805)。当用户接收适当权限来使用特定程序时,程序从存储部分被传送到受保护程序区域的活动部分(810)。

[0102] 为了预备在活动区域中使用一个或多个程序,程序文件被移动到 DRAM 逻辑闪存的受保护程序区域中。该传送不使用逻辑记录,而是批量传送。受保护程序区域类似地对于用户不可直接访问(815)。

[0103] DRAM 逻辑闪存的受保护程序区域中的程序被传送到 CPU 存储器用于执行。程序文件典型地不被包含在快照或转储 / 恢复操作中,因为程序文件在执行期间不被改变。由程序文件生成的文件(包含配置和数据文件)能够被包含在快照和转储 / 恢复操作中(820)。

[0104] 对程序文件(包含用户还没有权限访问的程序文件)的更新能够使用后台操作从网络被流传送到 SSD 的受保护程序区域(825)。这些更新的程序然后能够在用户支付或授权请求时迅速地成为可用。

[0105] 在掉电期间保存数据

图 9 是用于当机器关断或断电时将存储在 DRAM 逻辑闪存中的数据保存到非易失性闪存存储器的说明性方法(700)的流程图。功率电容器电路或其它能量存储装置也可以被包含在 SSD 中。能量存储装置存储足够的能量来将存储在易失性 DRAM 存储器中的所有数据写入到非易失性 NAND 闪存存储器中(步骤 905)。当存在对 SSD 的功率中断时,电源中的传感器开始断电存储操作。此外,断电存储操作可以被用户或操作系统触发(步骤 910)。在断电存储操作期间,来自功率电容器的能量被用来刷新 DRAM 以保护数据、操作控制器、将数据传送到存储器控制器、以及将数据写入在 NAND 闪存存储器中。主控制器建立数据路径,并且 DRAM 控制器开始经由内部总线将数据发送到存储器控制器。数据以全速度被写入到每个存储器模块中。每个数据文件包含将文件识别为断电存储操作的部分的报头(步骤 915)。每个数据片段包含允许转储和恢复程序精确地按存储的那样恢复逻辑闪存数据的逻辑记录标志。八个闪存存储器控制器中的每个监视其逻辑记录标志。当闪存控制器识别其逻辑记录标志时,控制器将后续数据存储在其闪存存储器中(步骤 920)。因为转储顺序加载整个闪存存储器,所以没有擦除周期并且写入速度能够接近每秒 300 MB。理论上,完整的 8 GB DRAM 存储器能够在 24 秒内被转储到闪存中。在 DRAM 逻辑闪存不是全满或者含有没有被改变的文件的的情况下,传送时间能够显著更少。能量存储装置具有电容性功率电路,该电容性功率电路被设计成维持向 SSD 驱动器的功率用于 DRAM 逻辑闪存中所有数据到闪存存储器的最大速度数据传送。

[0106] 在一个实施例中,闪存管芯中的备用块被用来在断电存储操作期间存储数据。备用块已经是空白的,所以擦除延迟不发生。备用块贯穿每个管芯来分布。因此,快照物理上不位于一个接连的位置。然而,被包含在每个数据片段中的报头识别下一个数据片段。因此,通过将第一数据片段存储在已知的位置,主控制器能够以它们被写入的相同次序(先入先出)恢复所有数据文件。

[0107] 在电容性功率电路被耗尽之前,指针能够被存储以帮助重启。在一个实施方式中,主处理器积累被加载到闪存中的逻辑记录的目录。该目录被写入在闪存存储器上的受保护区域中。当计算装置被重启时,从受保护区域取回目录。主控制器然后使用表来控制逻辑闪存的操作。

[0108] 恢复过程是断电存储过程的反转。操作系统感测重启并且使快照被取回。任何必要的表或其它索引数据首先从闪存中的转储区域被取回(935)。这些表可以被存储在主控制器的存储器中或被存储在 DRAM 逻辑闪存中。主控制器然后使用这些表来访问快照并且在断电存储操作之前重构操作系统状态(940)。在一个实施方式中,保存的数据的第一片段被传送到逻辑闪存,后面是数据的第二片段等等直到所有数据被再次存储在 DRAM 逻辑闪存上。该操作将数据的干净版本恢复到 DRAM 逻辑闪存。恢复的操作系统然后使用逻辑记录表来命令主控制器从逻辑闪存取回所要求的文件。

[0109] 大体上,恢复的序列将在 CPU 和操作系统的控制下。操作系统将命令所需的各种文件的加载。在一些实施方式中,可以存在对是开放的程序的转储引用。如果转储引用是恒定的,则这些不被重新写入。主控制器可以维持针对每个开放程序的面包屑(bread crumbs)的集合,使得恢复过程能够将代码重设到运行的最新位置。然而,并非所有程序将具有面包屑,而是将被加载为恢复序列的部分。

[0110] 误处理文件恢复

图 10 是用于恢复存储在 DRAM 逻辑闪存中的误处理数据文件的说明性方法(1000)的流程图。例如,用户可能偶然关闭文件而没有正确地保存它。CPU 将指令发送到 SSD 以删除文件(1005)。

[0111] 文件的删除被记录在 FAT 表中。然而,文件中的数据仍然被存储在 DRAM 逻辑闪存中一时间段,并且被有限的文件历史索引(1010)。用户发现文件被关闭而没有正确地保存它,并且请求操作系统恢复文件(1015)。操作系统访问有限的文件历史以确定组成“删除的”文件的数据是否仍然存在 DRAM 逻辑闪存上(1020)。如果“删除的”文件仍然存在,则它从 DRAM 逻辑闪存中被取回并且被发送到 CPU 存储器(1025)。

[0112] 结论

总而言之,上面描述的说明性 SSD 体系结构提供硬盘驱动器的即插即用备选。上面描述许多原理,其允许闪存存储器被有效地用作非易失性存储而不管它的有限数目的擦除/写入周期。DRAM 逻辑闪存的使用模仿闪存的行为,允许以全接口速度处理所有闪存命令并且最小化到 NAND 闪存存储器的写入。就系统处理器而言,它一直在 SSD 内写入到闪存存储器。然而,系统处理器正在写入到 DRAM,其充当逻辑闪存但没有 NAND 闪存存储器的寿命或寻址限制。DRAM 逻辑闪存以与闪存相同的方式来存储文件并且响应于闪存命令。进一步,DRAM 逻辑闪存使用 FAT 表、更新逻辑记录、组合文件、并且被附着到 SATA 总线。因为 DRAM 逻辑闪存具有实际上无限数目的读/写周期,所以系统处理器和操作系统能存储与期望的

一样多的更新和快照。进一步, DRAM 逻辑闪存存在读取和写入数据两者上极其快速。SSD 存储足够功率以在外部功率被中断的情况下将存储在 DRAM 逻辑闪存中的整个数据内容移动到闪存存储器。

[0113] SSD 中的闪存控制器处理逻辑记录转化、错误检测和恢复、和装置优化。在一些实施例中, 出于使用的速度和舒适性, 每个闪存接口装置可以控制 2 到 4 个管芯。主控制器(并且在一些实施例中总线控制器) 控制 DRAM 逻辑闪存与每个闪存控制器之间的数据传送。

[0114] 如上面讨论的, DRAM 存储器及其控制器组成 DRAM 逻辑闪存。DRAM 逻辑闪存中的数据由局域微处理器(DRAM 控制器、逻辑闪存控制器、和主控制器) 来管理以完全看来是闪存驱动器。针对与 SATA 系统的所有通信的所有事务仅经过该接口发生。DRAM 逻辑闪存一直以全 SATA 速度从 SATA 总线读取并且写入到 SATA 总线。因此, DRAM 逻辑闪存完全看来是闪存装置, 但具有显著更高的数据传送速率。这使 SSD 操作对于计算装置透明, 该计算装置能够正如它正在写入到标准闪存或硬驱动器装置那样运行。

[0115] DRAM 逻辑闪存不是高速缓存, 并且不用作高速缓存。DRAM 逻辑闪存中的文件正如它们将在闪存中那样通过逻辑记录到物理位置映射和文件管理来写入。DRAM 控制器接受闪存命令并且实施它们使得 CPU 一直相信它正在写入到闪存存储器驱动器。然而, CPU 一直读取并写入到 DRAM 逻辑闪存。CPU 不直接访问闪存存储器。闪存仅在由主控制器确定的特定预定的点处被写入。这些点独立于 CPU 命令并且不能够直接被 CPU 触发。

[0116] 上面给出的实施方式只是在本文中描述的原理的说明性示例。能够使用各种其它配置和体系结构。例如, DRAM 控制器和逻辑闪存控制器的功能能够被组合成单个控制器。在其它实施方式中, 主控制器和总线控制器的功能能够被组合。总线、易失性存储器、以及非易失性存储器装置的数目和类型能够变化以适应各种设计参数和新的技术。例如, 虽然上面的示例中描述 SATA 总线、DRAM 存储器、和 NAND 存储器, 但是能够使用各种其它总线和存储器技术。

[0117] 此外, 在图 1-4 中示出并且上面描述的体系结构只是示例。能够使用许多备选体系结构。虽然上面讨论的原理示出充当 CPU 与一系列闪存存储器之间的看门者的 DRAM 逻辑闪存, 但是 DRAM 逻辑闪存也能够被插入在 CPU 与常规旋转盘片硬驱动器或其它非易失性存储器之间。

[0118] 因此, SSD 是标准硬驱动器的简易取代并且不要求对计算装置的任何编程或物理改变。

[0119] 前述描述已经仅被呈现以图解和描述所描述的原理的示例。该描述不旨在为详尽的或者将这些原理限制到所公开的任何精确形式。许多修改和变化鉴于上面的教导是可能的。

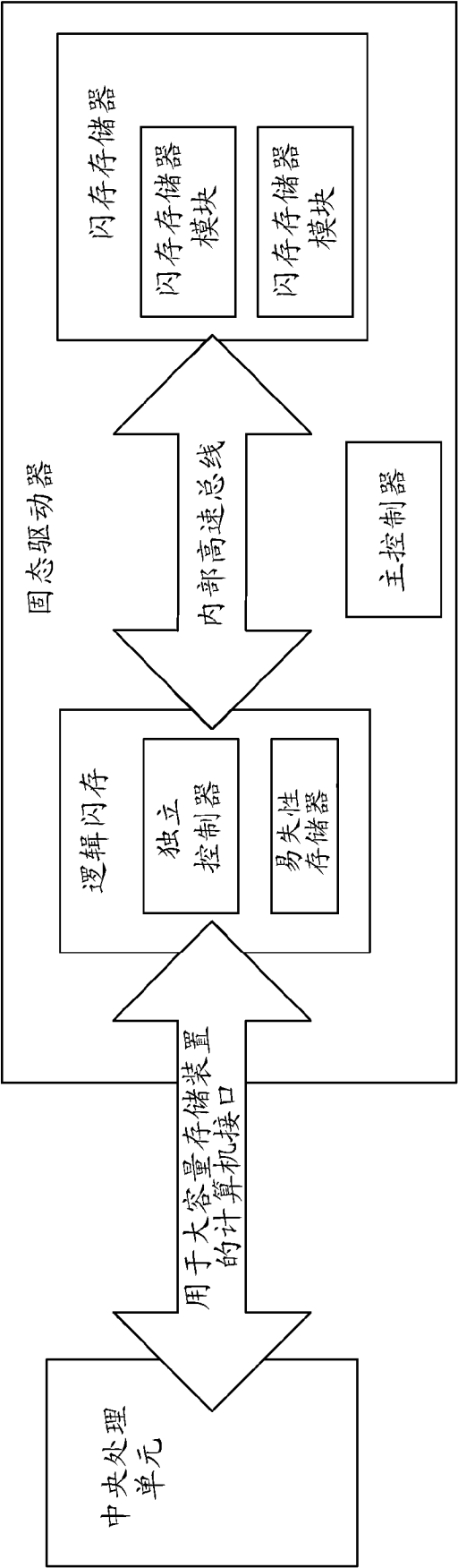


图 1

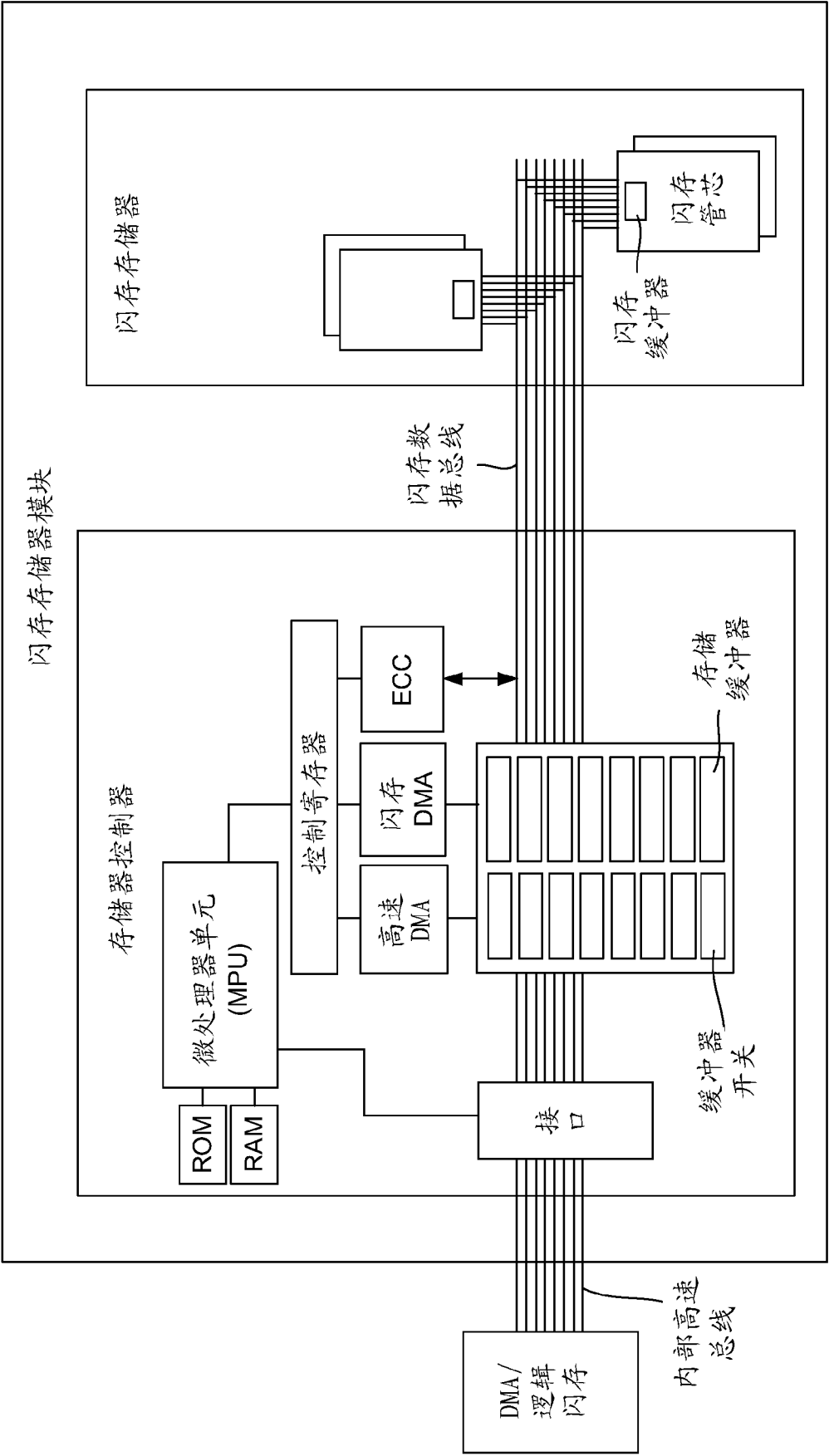


图 2

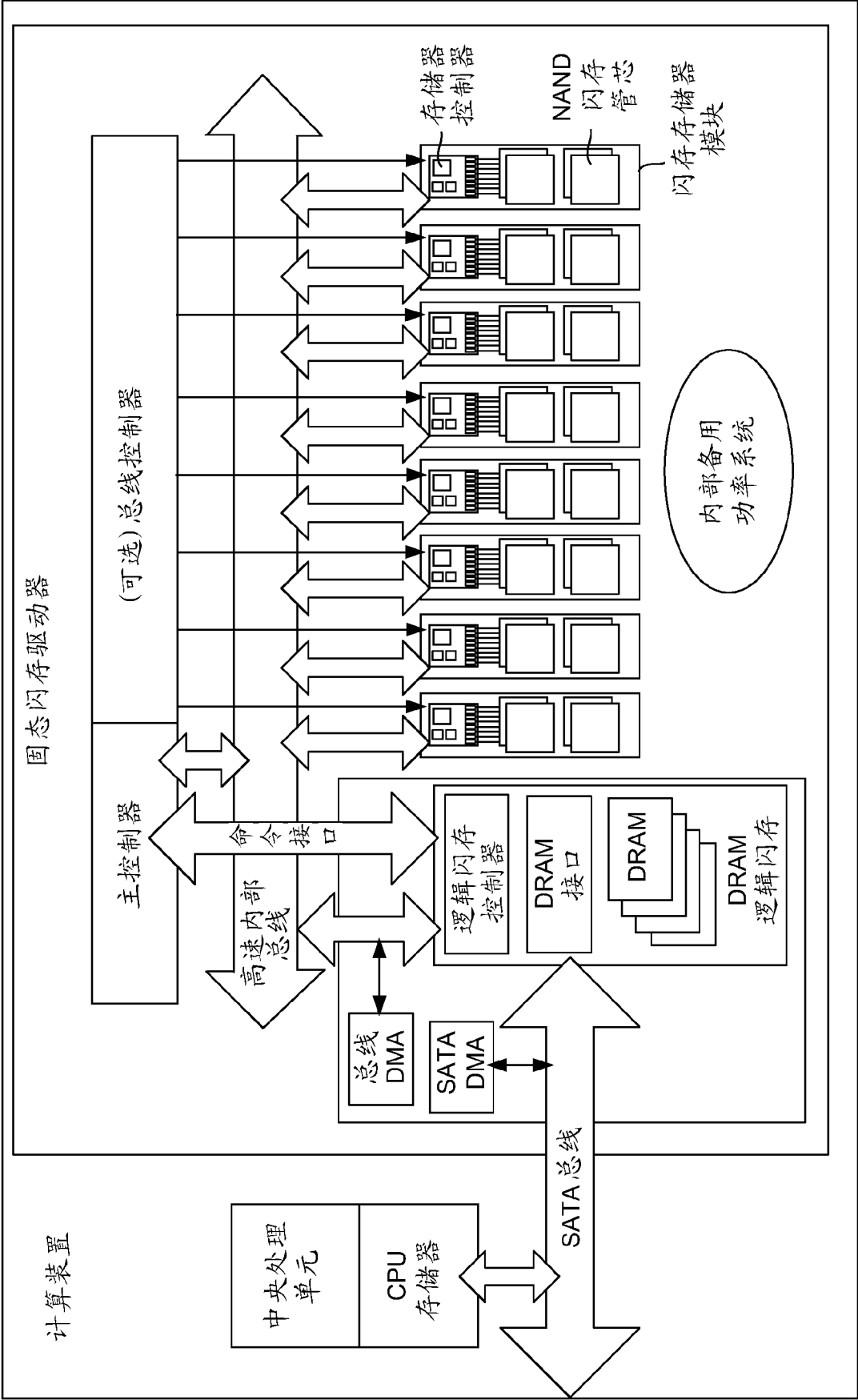


图 3

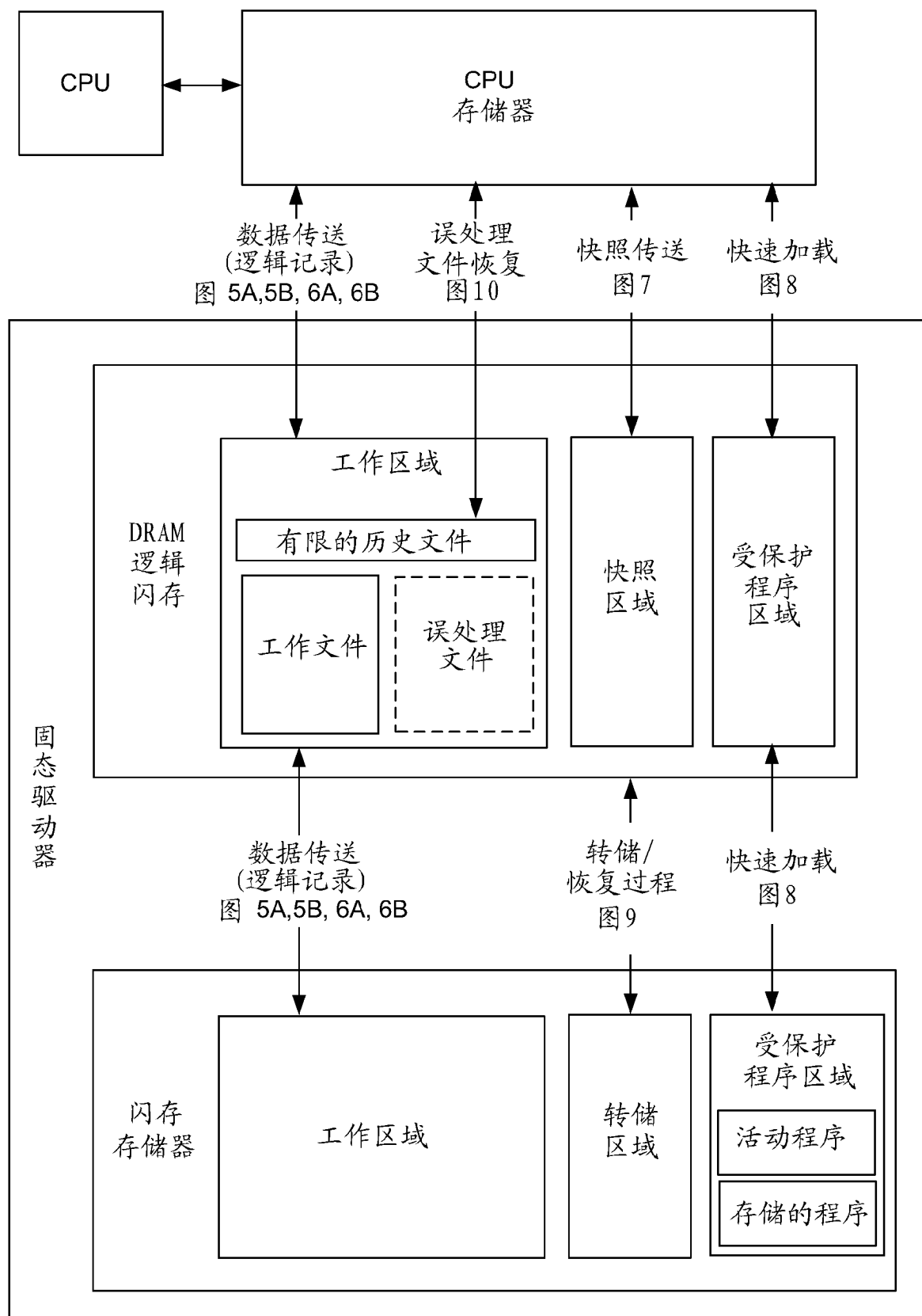


图 4

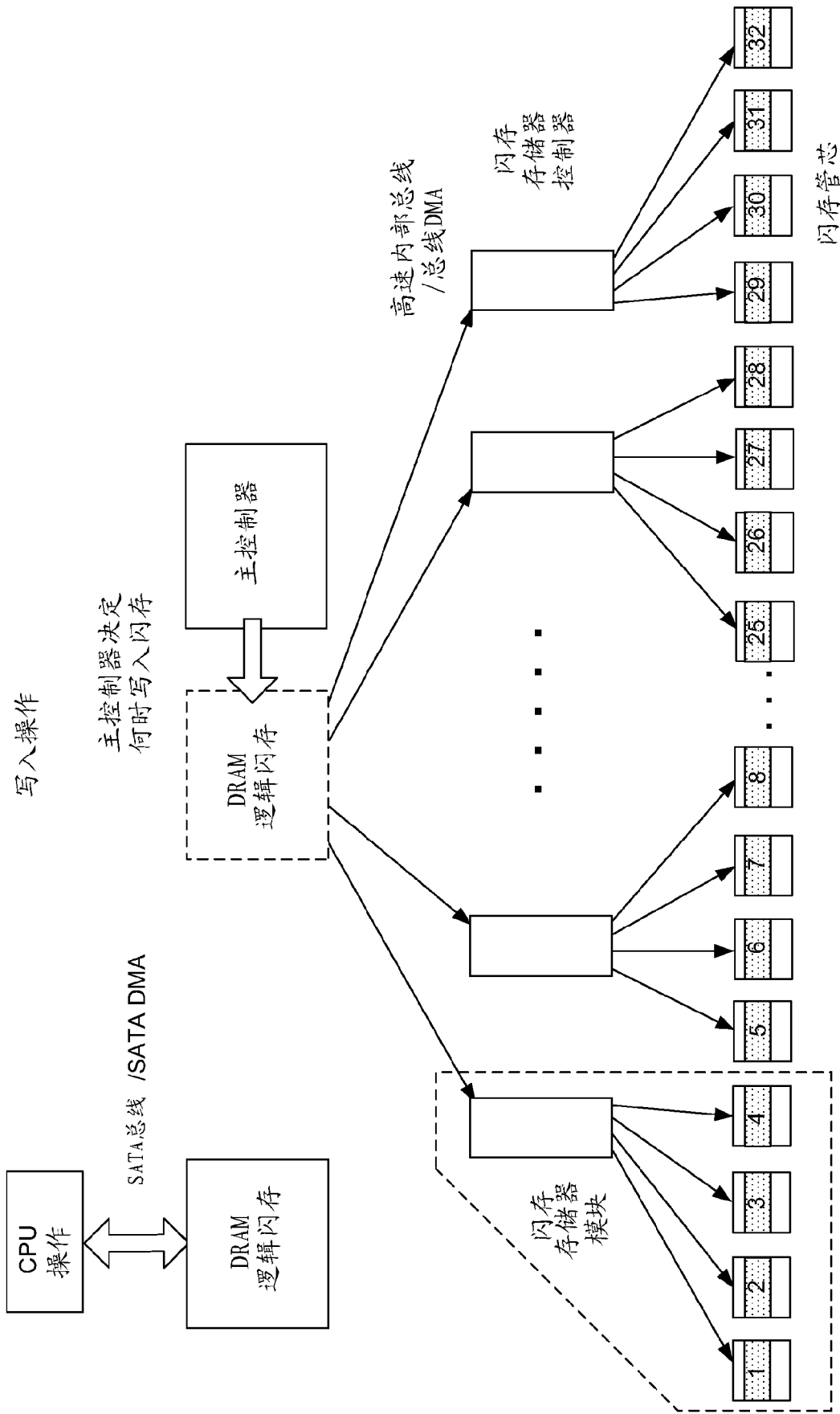


图 5A

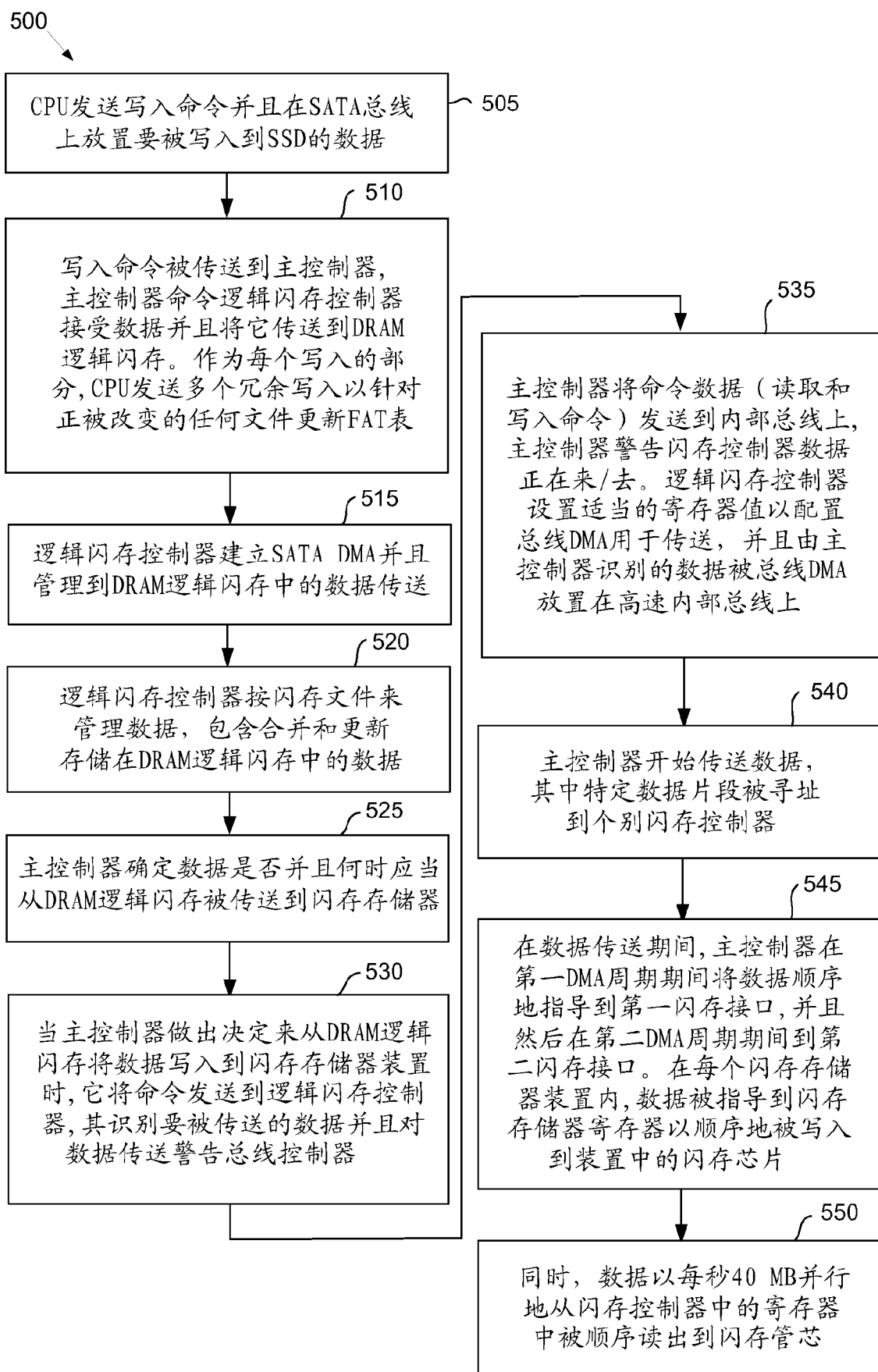


图 5B

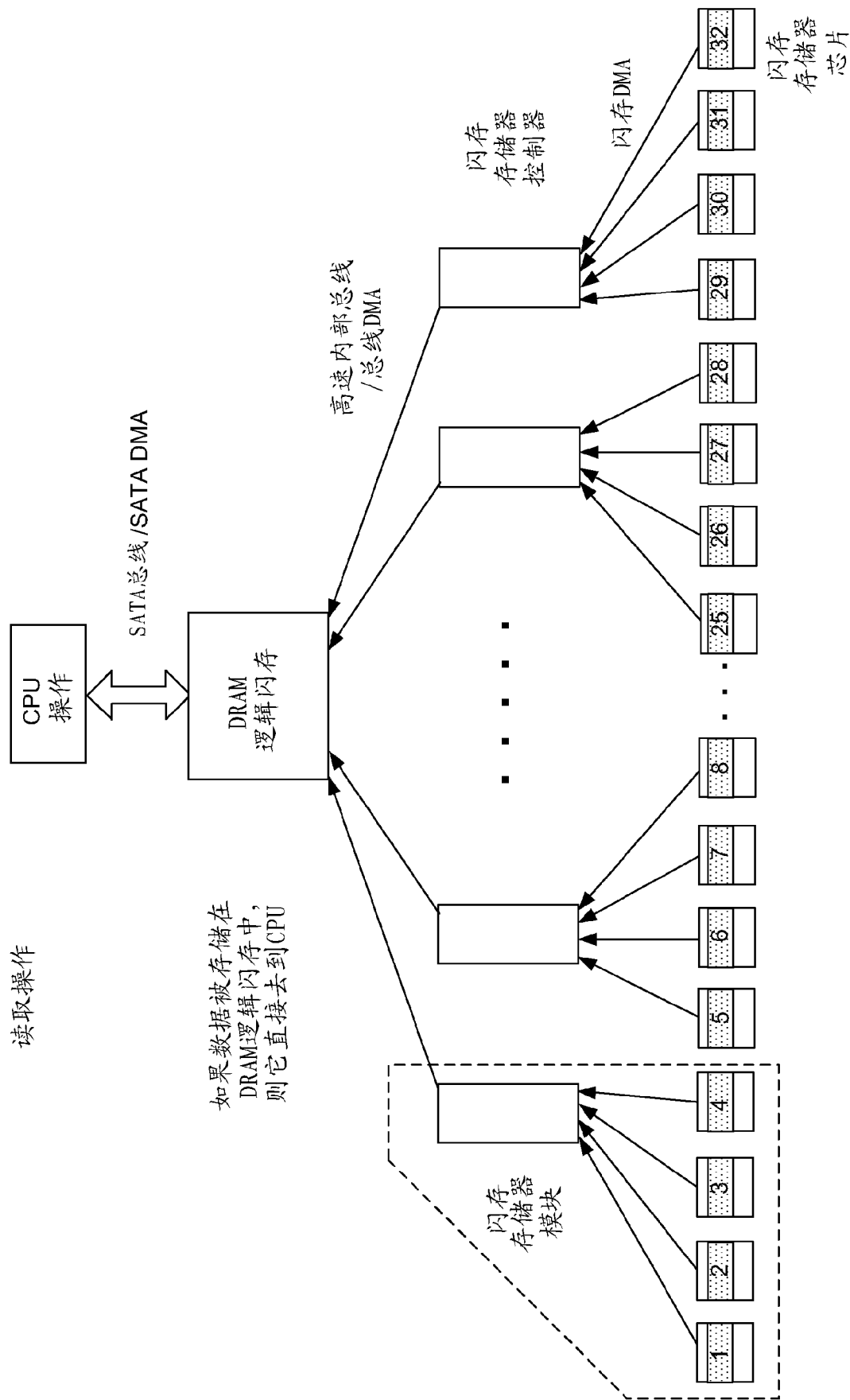


图 6A

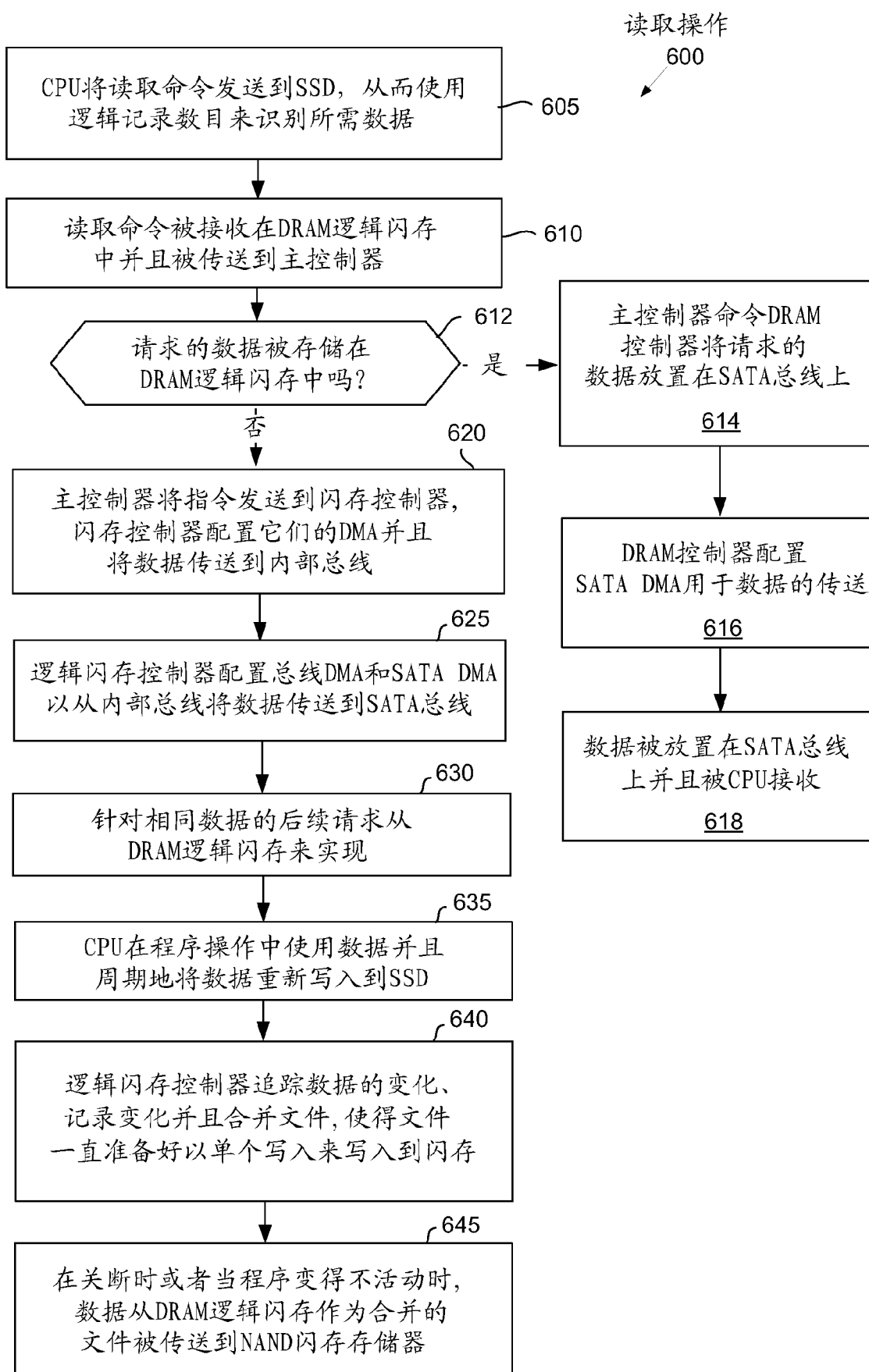


图 6B

快照操作

700

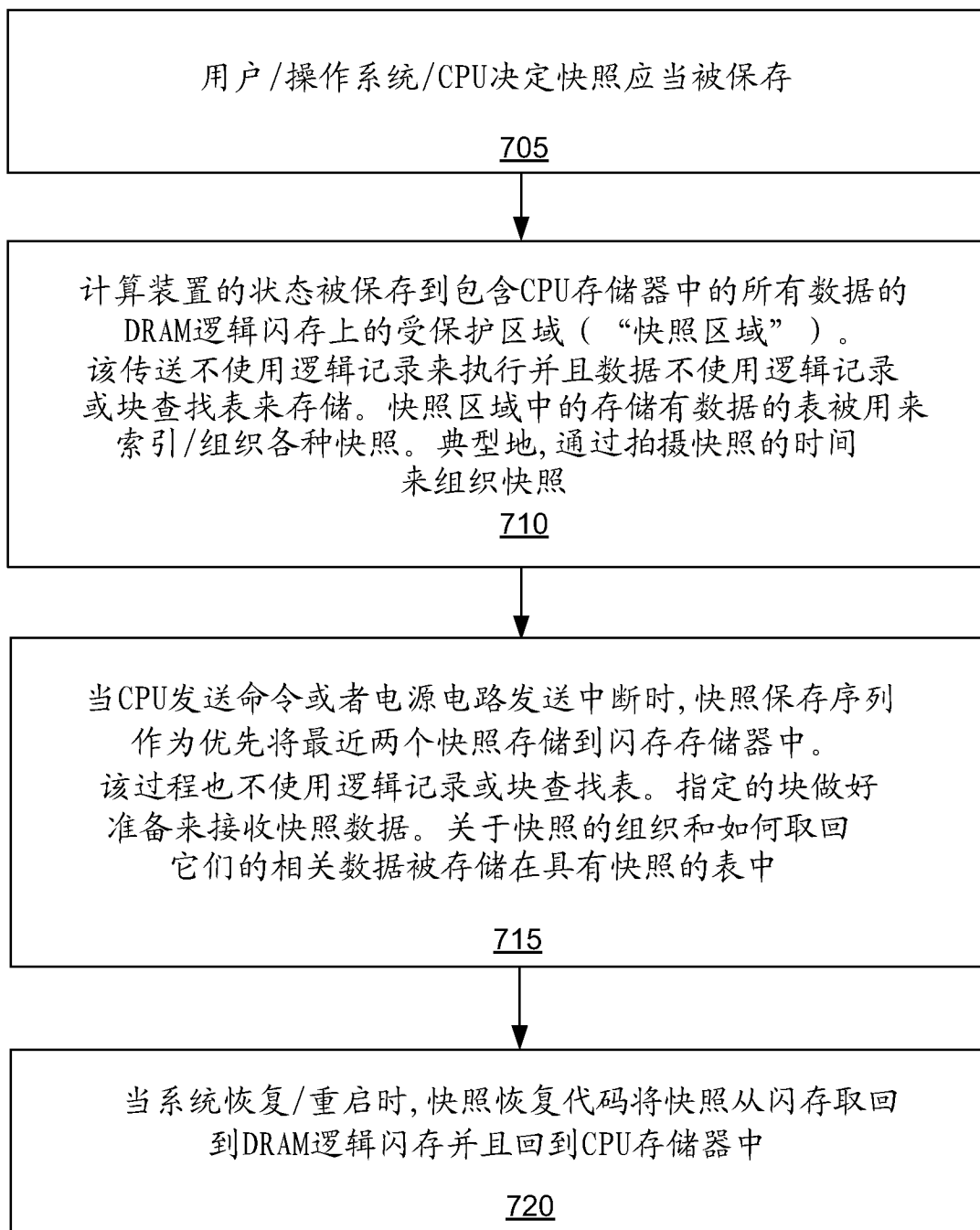


图 7

快速加载操作

800

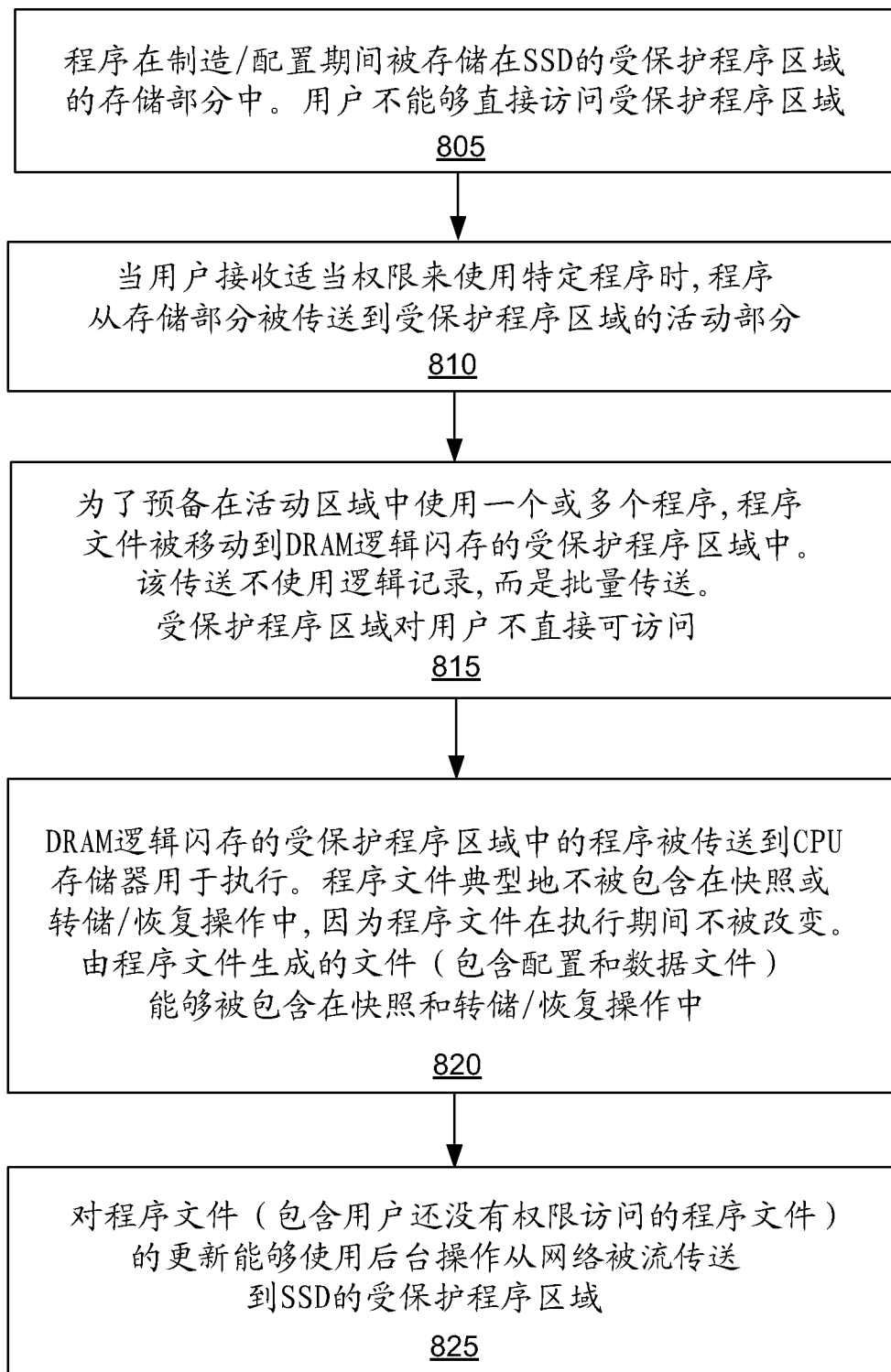


图 8

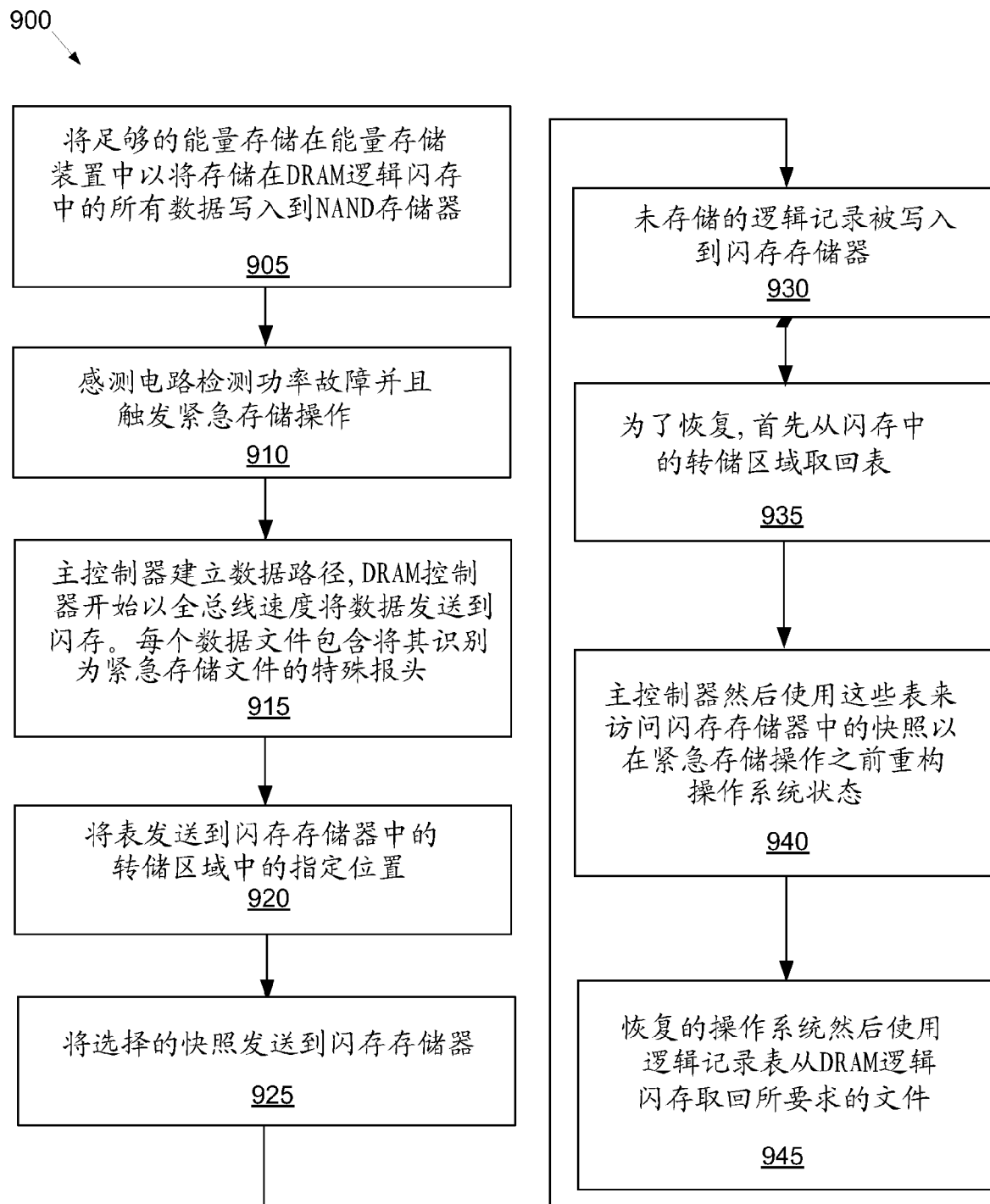


图 9

删除的文件恢复
1000

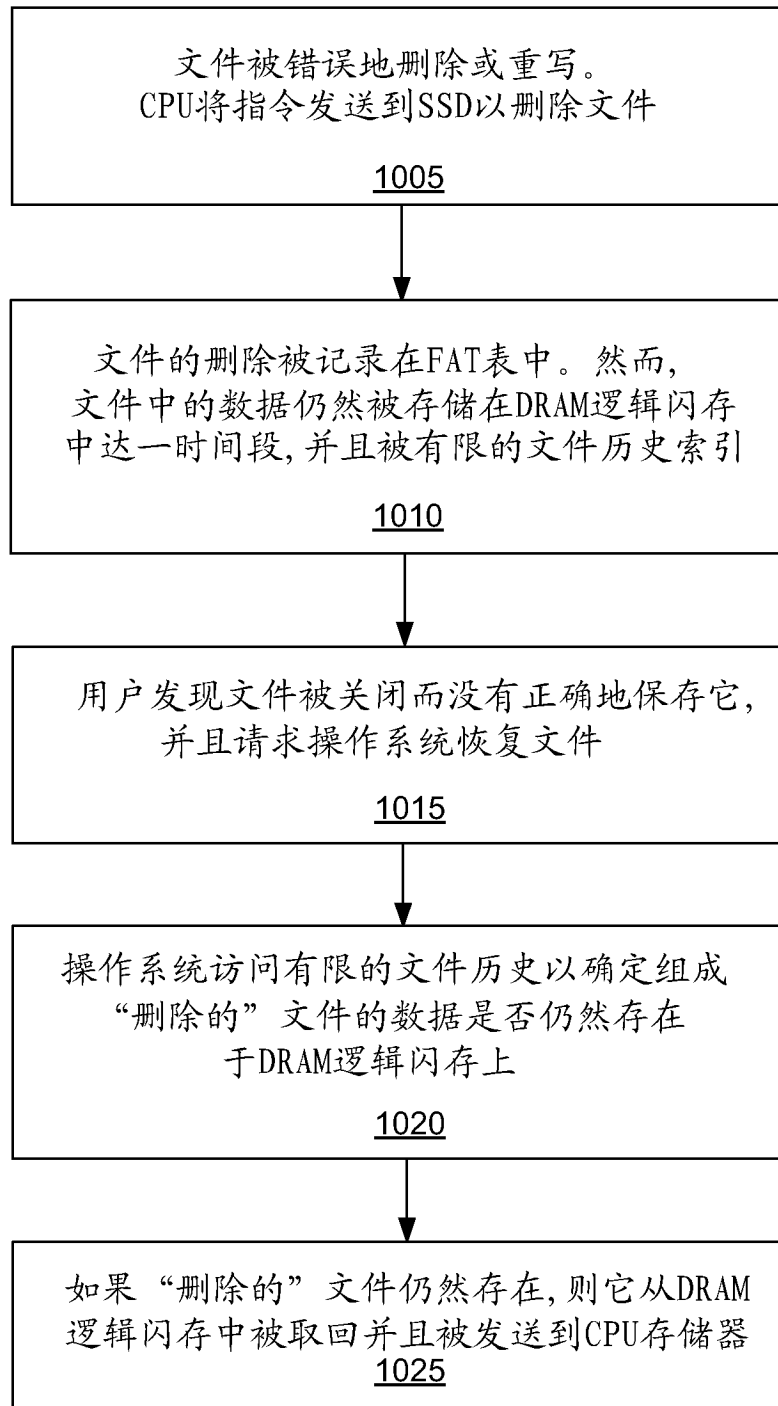


图 10