



(12) 发明专利

(10) 授权公告号 CN 101923499 B

(45) 授权公告日 2013. 11. 06

(21) 申请号 201010158039. 8

审查员 吴卿

(22) 申请日 2010. 03. 29

(30) 优先权数据

12/414, 385 2009. 03. 30 US

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 S · N · 特里卡

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 毛力 袁逸

(51) Int. Cl.

G06F 11/14(2006. 01)

G06F 12/16(2006. 01)

(56) 对比文件

US 2003/0120868 A1, 2003. 06. 26, 全文 .

US 2005/0177687 A1, 2005. 08. 11, 全文 .

CN 101099135 A, 2008. 01. 02, 全文 .

US 6219693 B1, 2001. 04. 17, 全文 .

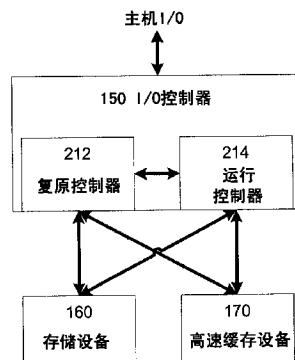
权利要求书4页 说明书11页 附图10页

(54) 发明名称

执行防电源故障高速缓存而无需原子元数据的技术

(57) 摘要

一种方法和系统允许将永久存储设备中的数据防电源故障地回写或直写高速缓存入高速缓存设备的一个或多个高速缓存行。当存储设备中的数据被高速缓存时，没有与任一高速缓存行关联的元数据被原子地写至高速缓存设备。如此，在高速缓存数据期间不需要专门的高速缓存硬件来允许元数据的原子写入。



1. 一种用于执行防电源故障高速缓存而无需原子元数据的方法,包括:

将第一设备的数据高速缓存入第二设备的多个高速缓存行中的一个或多个,包括:从关联于所述多个高速缓存行的状态信息判断所述多个高速缓存行中是否有任何高速缓存行被标记为未使用;

如果没有,则

逐出所述多个高速缓存行中的至少一个;以及

在所述状态信息中将所述至少一个被逐出的高速缓存行标记为未使用;

选择在所述状态信息中被标记为未使用的一个或多个高速缓存行;以及

将所述第一设备的所述数据高速缓存入所选择的一个或多个高速缓存行;

其中在所述数据的高速缓存期间,没有任何关联于任何所述高速缓存行的状态信息要随所述数据原子地存储在所述第二设备中。

2. 如权利要求1所述的方法,其特征在于,还包括当所述第一设备的所述数据被高速缓存入所述第二设备的一个或多个高速缓存行时,将关联于所述一个或多个高速缓存行的状态信息存储在存储器中。

3. 如权利要求2所述的方法,其特征在于,还包括:

当新数据要被写至所述一个或多个高速缓存行时,用所述新数据更新所述一个或多个高速缓存行中被高速缓存的数据;以及

基于所述新数据,在所述存储器中更新关联于所述一个或多个高速缓存行的所述状态信息。

4. 如权利要求3所述的方法,其特征在于,还包括:

当接收到重启、停机、休眠、或待机命令中的一个时,将所述存储器中的全部状态信息复制至所述第二设备。

5. 如权利要求3所述的方法,其特征在于,还包括当发生电源或系统故障时将所述第二设备复位。

6. 如权利要求3所述的方法,其特征在于,所述第二设备包括与所述存储器中的所述一个或多个高速缓存行关联的状态信息的副本,所述方法还包括响应于更新所述被高速缓存的数据,基于所述第二设备中所述状态信息的副本中的所述新数据来更新与所述一个或多个高速缓存行关联的状态信息。

7. 如权利要求3所述的方法,其特征在于,所述第二设备包括与所述存储器中所述一个或多个高速缓存行关联的状态信息的副本,所述方法还包括响应于更新所述被高速缓存的数据,基于所述第二设备中所述状态信息的副本中的所述新数据来将所述一个或多个高速缓存行标记为等待与所述一个或多个高速缓存行关联的状态信息的更新。

8. 如权利要求7所述的方法,其特征在于,还包括当接收到刷新、重启、停机、休眠或待机命令之一时,更新所述第二设备中的所述状态信息的副本中与所述一个或多个标记的高速缓存行关联的状态信息。

9. 如权利要求3所述的方法,其特征在于,还包括当接收到刷新命令时,将所述存储器中与所述一个或多个高速缓存行关联的状态信息复制到所述第二设备。

10. 如权利要求9所述的方法,其特征在于,将所述存储器中与所述一个或多个高速缓存行关联的状态信息复制到所述第二设备包括优化复制操作的次数。

11. 一种用于执行防电源故障高速缓存而无需原子元数据的装置，包括：

NAND 闪存，其具有多个高速缓存行的第一逻辑段以及多个元数据的第二逻辑段，每个元数据关联于所述多个高速缓存行中相应的一个，其中所述第一逻辑段不具有任何元数据；以及

控制器，用来将存储设备的数据高速缓存入一个或多个高速缓存行，并进一步用来：

确定是否所有的空闲高速缓存行被使用；

如果是，

逐出一个或多个选择的高速缓存行，所述选择基于逐出策略；

将逐出的一个或多个选择的高速缓存行添加至空闲高速缓存行；且其中用于将所述存储设备的数据高速缓存入所述一个或多个高速缓存行的控制器用于将所述存储设备的数据高速缓存入所述一个或多个空闲的高速缓存行。

12. 如权利要求 11 所述的装置，其特征在于，所述第二逻辑段的每个元数据在其关联的对应高速缓存行中包括被高速缓存的数据的序列号、逻辑块地址、状态信息、和牵制信息中的至少一个。

13. 如权利要求 11 所述的装置，其特征在于，还包括用于存储多个元数据的存储器，其中所述控制器进一步用来：

当新数据要被写至所述一个或多个高速缓存行时，用所述新数据更新所述一个或多个高速缓存行的被高速缓存的数据；以及

基于存储器中的新数据更新与所述一个或多个高速缓存行关联的元数据。

14. 如权利要求 13 所述的装置，其特征在于，所述控制器进一步用来：

用所述新数据更新所述存储设备；以及

当所述装置要被重引导、停机、或置于休眠状态时，用所述存储器中与所述一个或多个高速缓存行关联的元数据覆写所述 NAND 闪存中与所述一个或多个高速缓存行关联的元数据。

15. 如权利要求 13 所述的装置，其特征在于，所述存储器进一步存储高速缓存表，所述高速缓存表包含所述存储设备中的数据与所述多个高速缓存行之间的链接信息，并且当所述装置要被重引导、刷新、停机、或置于休眠状态时，所述控制器进一步用于将所述存储器中的所述高速缓存表复制至所述 NAND 闪存。

16. 如权利要求 13 所述的装置，其特征在于，当新数据要被写至所述一个或多个高速缓存行时，响应于更新被高速缓存的数据，所述控制器基于所述 NAND 闪存中的新数据更新与所述一个或多个高速缓存行关联的元数据。

17. 如权利要求 13 所述的装置，其特征在于，所述控制器进一步将所述一个或多个高速缓存行添加至清单，所述清单用来指示所述 NAND 闪存中与所述一个或多个高速缓存行关联的元数据正在等待用与所述存储器中的所述一个或多个高速缓存行关联的元数据更新。

18. 如权利要求 17 所述的装置，其特征在于，当所述装置被断电或重引导时或当所述 NAND 闪存要被刷新时，所述控制器进一步用所述存储器中与所述一个或多个高速缓存行关联的元数据来更新所述 NAND 闪存中与所述一个或多个高速缓存行关联的元数据。

19. 如权利要求 13 所述的装置，其特征在于，当接收到操作系统 OS 所发布的刷新命令

时,所述控制器进一步用所述存储器中的元数据覆写非易失介质中的元数据。

20. 如权利要求 18 所述的装置,其特征在于,当所述装置要被上电或重启时,所述控制器进一步用所述 NAND 闪存中的元数据恢复所述存储器中的元数据。

21. 如权利要求 15 所述的装置,其特征在于,当所述装置要被上电或重启时,所述控制器进一步将所述高速缓存表从所述 NAND 闪存复制至所述存储器。

22. 如权利要求 11 所述的装置,其特征在于,所述存储设备是硬盘、磁带驱动器、压缩盘、zip 盘、闪存和固态驱动器中的一个。

23. 如权利要求 11 所述的装置,其特征在于,所述控制器是块存储驱动器。

24. 一种用于执行防电源故障高速缓存而无需原子元数据的方法,包括:

将存储设备的数据回写或直写高速缓存入高速缓存设备的多个高速缓存行中的一个或多个,包括:

确定所述多个高速缓存行中是否有任何高速缓存行被标记为未使用;

如果没有,

逐出所述多个高速缓存行中的至少一个;以及

将所述至少一个逐出的高速缓存行标记为未使用;

选择一个或多个标记为未使用的高速缓存行;以及

将所述存储设备的数据高速缓存入所选择的一个或多个高速缓存行;

其中当所述数据被高速缓存时没有任何关联于任何所述高速缓存行的元数据要被原子地写入所述高速缓存设备。

25. 如权利要求 24 所述的方法,其特征在于,数据的直写高速缓存包括:

确定所述高速缓存设备的所述一个或多个高速缓存行要用新数据更新;

用所述新数据更新所述一个或多个确定的高速缓存行;

在存储器中至少基于所述新数据来更新与所述一个或多个确定的高速缓存行关联的元数据;以及

用所述新数据来更新所述存储设备。

26. 如权利要求 24 所述的方法,其特征在于,所述数据的回写高速缓存包括:

确定所述高速缓存设备的所述一个或多个高速缓存行要用新数据更新;

在存储器中至少基于所述新数据来更新与所述一个或多个确定的高速缓存行关联的元数据;

用所述新数据来更新所述一个或多个确定的高速缓存行;以及

标记所述一个或多个确定的高速缓存行,其中所述标记用于指示与所述高速缓存设备中的所述一个或多个确定的高速缓存行关联的元数据是要用所述存储器中与所述一个或多个确定的高速缓存行关联的元数据来更新的。

27. 如权利要求 24 所述的方法,其特征在于,所述数据的回写高速缓存还包括:

当接收到断电、重引导或刷新命令时,用所述存储器中与所述一个或多个确定的高速缓存行关联的元数据来更新所述高速缓存设备中与所述一个或多个确定的高速缓存行关联的元数据。

28. 一种用于执行防电源故障高速缓存而无需原子元数据的系统,包括:

NAND 闪存,其具有多个高速缓存行的第一逻辑段和多个元数据的第二逻辑段,每个元

数据关联于所述多个高速缓存行中对应的一个，其中所述第一逻辑段不具有任何元数据；以及

存储器控制器，用来将存储介质的数据回写或直写高速缓存入所述NAND闪存的所述第一逻辑段的一个或多个高速缓存行，并进一步用来确定是否所有的空闲高速缓存行被使用；

如果是，

逐出一个或多个选择的高速缓存行，所述选择基于逐出策略；

将逐出的一个或多个选择的高速缓存行添加至空闲高速缓存行；且其中用于将所述存储设备的数据高速缓存入所述一个或多个高速缓存行的控制器用于将所述存储设备的数据高速缓存入所述一个或多个空闲的高速缓存行，并且其中当所述数据被高速缓存时没有任何元数据要被原子地写入所述第一逻辑段。

29. 如权利要求28所述的系统，其中所述第二逻辑段的每个元数据在其关联的对应高速缓存行中包括被高速缓存的数据的序列号、逻辑块地址、状态信息和牵制信息中的至少一个。

## 执行防电源故障高速缓存而无需原子元数据的技术

### 发明领域

[0001] 本发明涉及高速缓存，更具体但非排他地涉及非易失介质中的防电源故障回写或直写高速缓存。

[0002] 背景描述

[0003] 存储子系统是计算机系统最慢的子系统之一，尤其是当存储子系统利用例如硬盘驱动器 (HDD) 的存储介质时。由于读 / 写头需要机械移动至 HDD 盘组上的特定位置以读 / 写数据，HDD 需要相对长的存取时间。

[0004] 为了提高 HDD 的性能，非易失高速缓存存储器可用来保留最近从 HDD 读取和写至 HDD 的结果。通过对 HDD 数据进行高速缓存，计算机系统的性能得以提高，且 HDD 可在更长的周期内保持停止运作 (spun down) 以减少计算机系统的功耗。

[0005] 然而，如果计算机系统的供电被始料未及地断开，则非易失高速缓存存储器中的数据必须再次与 HDD 关联以防止数据败坏。通过高速缓存数据写入支持原子元数据写入的专用高速缓存硬件可用来确保这种恢复准确地完成，但这增加了计算机系统的成本。

[0006] 附图简述

[0007] 本发明的实施例的特征和优点将从下面对主题事项的详细说明中得以明瞭，其中：

[0008] 图 1 示出根据本发明一个实施例的用于实现本文描述的方法的系统；

[0009] 图 2 示出根据本发明一个实施例的 I/O 控制器的方框图；

[0010] 图 3 示出根据本发明一个实施例的操作系统中的模块的方框图；

[0011] 图 4 示出根据本发明一个实施例的高速缓存设备的配置；

[0012] 图 5 示出根据本发明一个实施例的直写高速缓存方案的流程图；

[0013] 图 6A 示出根据本发明一个实施例的回写高速缓存方案的流程图；

[0014] 图 6B 示出根据本发明一个实施例的回写高速缓存方案的流程图；

[0015] 图 6C 示出根据本发明一个实施例的回写高速缓存方案的流程图；

[0016] 图 7 示出根据本发明一个实施例的将数据插入高速缓存行的方法的流程图；以及

[0017] 图 8A 和 8B 示出根据本发明一个实施例的实现回写高速缓存方案的伪代码。

[0018] 详细说明

[0019] 本文描述的本发明实施例是通过示例示出的而不是在附图中作出限定。为了解说的简单和清楚，附图中示出的要素不一定按比例绘出。例如，一些要素的尺寸可相对于其它要素放大以清楚表示。此外，在认为合适的情形下，在附图中重复出现的许多附图标记用来指示相应或相似的要素。说明书中对本发明“一个实施例”或“一实施例”的引述表示结合该实施例描述的具体特征、结构或特点包含在本发明的至少一个实施例中。因此，在说明书中多处出现的短语“在一个实施例中”不一定全部指向同一实施例。

[0020] 本发明的实施例提供一种方法和系统，允许在永久存储设备中将数据防电源故障回写或直写高速缓存入高速缓存设备的一条或多条高速缓存行，该高速缓存设备不要求原子元数据。当存储设备中的数据被高速缓存时，没有与任意高速缓存行关联的元数据原子

地写入高速缓存设备。如此,就不需要在高速缓存数据的过程中原子地写入元数据的专门的高速缓存硬件。

[0021] 在本发明的一个实施例中,与高速缓存行关联的元数据包括但不限于:数据在经过高速缓存的存储设备上的位置,例如数据的逻辑块地址 (LBA) ;序列号;高速缓存行状态,例如数据是干净的还是腐败的;存储设备的高速缓存 LBA 的牵制 (pin) 信息等。存储设备包括但不限于固态驱动器 (SSD)、HDD、独立盘卷的冗余 (RAID) 组盘、磁带驱动器、压缩盘 (CD)、软盘、通用串行总线 (USB) 闪存驱动器或任意其它形式的非易失或永久计算机数据存储介质。高速缓存设备包括但不限于非易失介质、SSD、NAND 闪存、相变存储器或任意其它形式的非易失或永久计算机数据存储介质。

[0022] 图 1 示出用于实现根据本发明一个实施例在本文中披露的方法的系统 100。系统 100 包括但不限于台式计算机、膝上计算机、笔记本计算机、上网本计算机、个人数字助理 (PDA)、服务器、工作站、蜂窝电话、移动计算设备、互联网设施或任意其它类型的计算设备。在另一实施例中,用来实现本文描述方法的系统 100 可以是芯片系统上 (SOC) 的系统。

[0023] 系统 100 包括存储器 / 图形控制器 120 和 I/O 控制器 150。存储器 / 图形控制器 120 一般提供存储器和 I/O 管理功能,还有可由处理器 110 访问或使用的许多通用和 / 或专用寄存器、定时器等。处理器 110 可使用一个或多个处理器实现或使用多核处理器实现。根据本发明的一个实施例, I/O 控制器 150 允许将存储设备 160 中的数据防电源故障回写或直写高速缓存入高速缓存设备 170 或非易失存储器 144 的一条或多条高速缓存行。

[0024] 存储器 / 图形控制器 120 执行多种功能,以使处理器 110 访问包含易失存储器 142 和 / 或非易失存储器 144 的主存储器 140 并与之通信。在本发明的另一实施例中,另一易失存储器 142(图 1 未示出)内嵌在存储设备 160 中以对存储设备 160 的数据进行高速缓存。根据本发明的另一实施例,存储器 / 图形控制器 120 可取代 I/O 控制器 150 而允许将存储设备 160 中的数据防电源故障回写或直写高速缓存入高速缓存设备 170 的一条或多条高速缓存行。

[0025] 易失存储器 142 包括但不限于同步动态随机存取存储器 (SDRAM)、动态随机存取存储器 (DRAM)、RAMBUS DRAM(RDRAM) 和 / 或任意其它类型的随机存取存储器设备。非易失存储器 144 包括但不限于 NAND 闪存、只读存储器 (ROM)、电可擦除可编程 ROM (EEPROM) 和 / 或任意其它要求类型的存储器设备。主存储器 140 存储由处理器 110 执行的信息和指令。在处理器 110 执行指令的同时,主存储器 140 也可存储临时变量或其它中间信息。在本发明的另一实施例中,存储器 / 图形控制器 120 是处理器 110 的一部分。

[0026] 存储器 / 图形控制器 120 连接于显示设备 130,该显示设备 130 包括但不限于液晶显示器 (LCD)、阴极射线管 (CRT) 显示器或任何其它形式的视频显示设备。I/O 控制器 150 耦合于但不限于存储设备 160、高速缓存设备 170、网络接口 180 和键盘 / 鼠标 190。具体地说,I/O 控制器 150 执行多种功能以使处理器 110 与存储设备 160、高速缓存设备 170、网络接口 180 和键盘 / 鼠标 190 通信。在一个实施例中,高速缓存设备 170 可以是存储设备 160 的一部分。

[0027] 网络接口 180 是使用任意类型的公知网络接口标准实现的,包括但不限于以太网接口、USB 接口、外围组件互连 (PCI) 直通接口、无线接口和 / 或任意其它合适类型的接口。无线接口根据但不限于电气和电子工程师协会 (IEEE) 无线标准族 802.11、家用插头

AV (HPAV)、超宽带 (UWB)、蓝牙、WiMax 或任意其它形式的无线通信协议而工作。

[0028] 在本发明的一个实施例中，图 1 所示的总线是由与之相连的全部组件共享的通信链路。在本发明的另一实施例中，图 1 所示总线是彼此连接的多对组件之间的点对点通信链路。尽管图 1 示出的组件作为系统 100 中的分立模块予以描述，然而由这些模块中的一些实现的功能可集成在单个半导体电路中或使用两个或更多分立集成电路来实现。例如，尽管存储器 / 图形控制器 120 和 I/O 控制器 150 表示为分立模块，然而本领域内技术人员很容易理解存储器 / 图形控制器 120 和 I/O 控制器 150 可集成在单个半导体电路中。

[0029] 图 2 示出根据本发明一个实施例的 I/O 控制器 150 的方框图 200。I/O 控制器 150 具有复原控制器 212 和运行控制器 214。在本发明的一个实施例中，运行控制器 214 具有基于探试的高速缓存策略以确定存储设备 160 的数据是被高速缓存还是从高速缓存设备 170 中逐出。探试包括但不限于最近访问的 LBA、LBA 的牵制 (pin) 信息等。在本发明的一个实施例中，运行控制器 214 也执行高速缓存方案，例如检测高速缓存命中或高速缓存未命中以及高速缓存或逐出命令的队列化。

[0030] 在本发明的一个实施例中，运行控制器 214 利用高速缓存设备 170 的完整数据容量以高速缓存存储设备 160 的数据。在本发明的另一实施例中，运行控制器 214 利用高速缓存设备 170 完整数据容量的一部分以高速缓存存储设备 160 的数据。例如，在本发明的一个实施例中，运行控制器 214 利用高速缓存设备 170 的一半完整数据容量来高速缓存存储设备 160 的数据并将高速缓存设备 170 的另一半完整数据容量用作存储介质。

[0031] 在本发明的一个实施例中，复原控制器 212 和运行控制器 214 允许将存储设备 160 中的数据防电源故障回写或直写高速缓存入高速缓存设备 170。相关领域技术人员很容易理解，也可采用其它的高速缓存方案而不会影响本发明的工作。在本发明的一个实施例中，在系统 100 出故障的情形下，复原控制器 212 和运行控制器 214 保持存储设备 160 中的数据和高速缓存设备 170 中高速缓存的数据的完整性或一致性。系统 100 的故障事件包括但不限于，功率丢失故障、操作系统 (OS) 崩溃故障、系统 100 的不正当停止运作以及不在系统 100 正常操作状态下的其它事件。

[0032] 在本发明一个实施例中，复原控制器 212 在已发生故障事件后复原高速缓存设备 170 中的高速缓存行的高速缓存状态。在本发明的其它实施例中，复原控制器 212 处理其它事件，包括但不限于分离检测和处理、在运行控制器 214 初始化前处理全部 I/O 数据等。尽管复原控制器 212 和运行控制器 214 描述为图 2 中 I/O 控制器 150 的一部分，然而这并不构成限制。复原控制器 212 和运行控制器 214 可一同实现在同一硬件或软件模块中或在不同硬件或软件模块中单独实现。

[0033] 在本发明的一个实施例中，复原控制器 212 和运行控制器 214 是存储器 / 图形控制器 120 的一部分。在本发明的另一实施例中，复原控制器 212 和运行控制器 214 也可合并为单个控制器。相关领域内技术人员很容易理解可采用不同配置的复原控制器 212 和运行控制器 214 而不会影响本发明的工作。例如，在本发明的一个实施例中，复原控制器 212 实现为存储在系统 100 的供选 ROM 中的固件，而运行控制器 214 实现在系统 100 上执行的 OS 的块存储驱动器中。

[0034] 图 3 示出根据本发明一个实施例的 OS 中的模块的方框图 300。OS 具有应用层 310 和文件系统 320。应用层 310 能访问由文件系统 320 组织的文件。OS 还具有存储驱动器堆

栈 330 和块驱动器 340。根据本发明的一个实施例，块驱动器 340 具有运行 / 复原控制器 344。块驱动器 340 可包括运行控制器、复原控制器或运动控制器和复原控制器两者。

[0035] 运行 / 复原控制器 344 耦合于存储设备 160 和高速缓存设备 170 并且将存储设备 160 中的数据高速缓存入高速缓存设备 170。在高速缓存存储设备 160 中的数据的过程中，没有与高速缓存设备 170 的任何高速缓存行关联的状态信息或元数据原子地存储在高速缓存设备 170 中。在本发明的一个实施例中，OS 利用回写高速缓存方案，其中要写至存储设备 160 的任何数据首先被写至高速缓存设备 170。OS 系统在写将数据至高速缓存设备 170 后不是立即将该数据写至存储设备 160，而是等待适宜时间才将数据写至存储设备。如此，存储设备 160 的数据访问被最小化并且 OS 在执行其它指令前不需要等待将数据写至存储设备 160。由于高速缓存设备 170 的数据访问速率高于存储设备 160 的数据访问速率，回写高速缓存方案有利于加速系统 100 的存储子系统。

[0036] 当利用回写高速缓存方案时，存储设备 160 中的数据可与高速缓存设备 170 中的高速缓存数据不同步。在本发明的一个实施例中，当系统 100 的处理器 110、存储设备 160 或高速缓存设备 170 的利用率是正被使用时，运行 / 复原控制器 344 使高速缓存设备 170 中的高速缓存数据与存储设备 160 中的数据同步。例如，在本发明的一个实施例中，运行 / 复原控制器 344 确定系统 100 中处理器 110 的利用率低于某一阈值就使高速缓存设备 170 中尚未同步的高速缓存数据与存储设备 160 中的数据同步。相关领域内技术人员很容易理解可采用其它方案或策略以执行高速缓存设备 170 中的数据的背景同步而不会影响本发明的工作。

[0037] OS 可将定期刷新命令发布给 I/O 控制器 150 以确保所有之前写入的数据是非易失的。在本发明的一个实施例中，当刷新命令结束时，I/O 控制器 150 确保数据和元数据更新在存储设备 160 和高速缓存设备 170 中是非易失的，并且即使出现例如系统 100 电源故障的不当停止运作，也能恢复全部之前写入的数据。

[0038] 在本发明的另一实施例中，OS 利用直写高速缓存方案，这种情况下存储设备 160 中的数据和高速缓存设备 170 中高速缓存的数据永远是同步的。当 OS 执行写操作时，高速缓存设备 170 和存储设备 160 被写入同一数据。

[0039] 由于不需要专门的高速缓存硬件来启用防电源故障直写和回写高速缓存，因此本发明的实施例允许降低系统 100 的研发成本。例如，在本发明的一个实施例中，使用相对小尺寸的 SSD 来高速缓存一个或多个大尺寸硬盘驱动器而不需要专门的高速缓存硬件。

[0040] 图 4 示出根据本发明一个实施例的高速缓存设备 170 的配置 400。高速缓存设备 170 的配置 400 示出分组的元数据 401 的逻辑段以及高速缓存行 402 的另一逻辑段。高速缓存设备 170 的块宽度 405 表示高速缓存设备 170 的数据位宽度。在本发明的另一实施例中，高速缓存设备 170 的配置 400 也可包括针对例如数据存储或数据索引的其它目的的其它逻辑段（图 4 中未示出）。

[0041] 作为示例，高速缓存行 402 的逻辑段示出具有八个高速缓存行（高速缓存行 0-7），它们用来高速缓存存储设备 160 的数据。高速缓存行 402 的逻辑段不包含与任意高速缓存行 402 关联的任何元数据。本领域内技术人员很容易理解，高速缓存设备 170 可具有多于八个高速缓存行以高速缓存存储设备 160 的数据。在本发明的一个实施例中，高速缓存设备 170 的每个高速缓存行存储存储设备 160 的紧邻数据。在本发明的另一实施例中，高速

缓存设备 170 的每个高速缓存行不存储存储设备 160 的紧邻数据。块宽度 405 不局限于特定的位宽度。在本发明的一个实施例中,块宽度 405 是高速缓存设备和运行 / 复原控制器 344 之间的通信链路的总线宽度。例如,在本发明的一个实施例中,如果高速缓存设备和运行 / 复原控制器 344 之间的通信链路的总线宽度为 64 位,则块宽度 405 可设置成等于 64 位的倍数的位宽度。在本发明的另一实施例中,块宽度 405 可设置成存储存储设备 160 的 LBA 的倍数。例如,高速缓存设备的每个高速缓存行被设置成能存储存储设备 160 的四个 LBA 的块宽度 405。

[0042] 在本发明的一个实施例中,分组的元数据 401 的逻辑段具有以分组方式存储的元数据 0-7 因而多个元数据彼此紧邻地存储,每个元数据与一个不同的高速缓存行关联。例如,元数据 0 410 关联于高速缓存行 0 450,元数据 1 411 关联于高速缓存行 1 415,依此类推。在本发明的一个实施例中,分组的元数据 401 针对每个元数据块具有一个完整性标记。元数据 0-3 410、411、412 和 413 具有完整性标记 1 430 而元数据 4-7 414、415、416 和 417 具有完整性标记 2 440。完整性标记 430、440 防止系统 100 始料未及的停止运作或故障事件而破坏数据结构。在本发明的一个实施例中,分组元数据 401 的逻辑段紧邻地位于高速缓存设备 170 中以加快对分组元数据 401 的访问。在本发明的另一实施例中,分组元数据 401 的逻辑段不紧邻地位于高速缓存设备 170 中。在又一实施例中,完整性标记 430 和 440 未被存储在分组元数据 401 的逻辑段中。

[0043] 在本发明一个实施例中,为了促成高速缓存设备 170 中的回写或直写高速缓存,OS 维持易失存储器 142 中的高速缓存行的信息。该高速缓存行的信息包括但不限于未经使用或不留有存储设备 160 的任何数据的高速缓存行的清单、具有存储设备 160 中的数据或 LBA 和存储该数据或 LBA 的高速缓存设备 170 中的高速缓存行之间的链接信息的高速缓存表、高速缓存设备 170 中能以分组形式或不同形式存储的全部高速缓存行的元数据、易失存储器 142 中仍要写至高速缓存设备 170 的元数据的各个元数据的高速缓存行的列表等。在本发明的一个实施例中,OS 保留易失存储器 142 中的高速缓存设备 170 的分组元数据 401 的逻辑段的副本以利于高速缓存设备 170 中的回写或直写。在本发明的一个实施例中,高速缓存表可实现为散列表、树形表或任意其它搜索数据结构。

[0044] 图 5 示出根据本发明一个实施例的直写高速缓存方案的流程图 500。在步骤 510 中,运行控制器检查故障事件是否发生。在本发明的一个实施例中,运行控制器检查寄存器或标志是否指示故障事件已发生。在本发明的一个实施例中,步骤 510 检查系统 100 是否不当地断电。在本发明的另一实施例中,步骤 510 检查 OS 是否已崩溃或出故障。如果存在故障事件,则在步骤 512 运行控制器将高速缓存设备 170 复位。

[0045] 流程在步骤 512 将高速缓存设备 170 复位后回到步骤 510。在本发明的一个实施例中,运行控制器通过将高速缓存设备 170 的全部高速缓存行添加至空闲或未使用的高速缓存行的清单而将高速缓存设备 170 复位。高速缓存行的清单告知运行控制器清单中的高速缓存行可用于高速缓存存储设备 160 的数据。在本发明的另一实施例中,运行控制器通过将高速缓存设备 170 的全部高速缓存行著录或标记为未使用而将高速缓存设备 170 复位。

[0046] 如果不存在故障事件,运行控制器在步骤 520 检查是否存在正当地使系统 100 断电的请求。系统 100 的正当断电或停止运作指 OS 向系统 100 发布例如但不局限于重启命

令、停止运作命令、休眠命令、待机命令或使系统 100 断电的任意命令的事件。如果存在正当地使系统 100 断电的请求，则运行控制器在步骤 522 将与高速缓存设备 170 的全部高速缓存行关联的分组元数据从易失存储器 142 复制至高速缓存设备 170。在本发明的一个实施例中，运行控制器将与高速缓存设备 170 的全部高速缓存行关联的分组元数据从易失存储器 142 复制至分组元数据 401 的逻辑段。在可选步骤 524，运行控制器将高速缓存表从易失存储器 142 复制至高速缓存设备 170 且流程 500 回到步骤 510。

[0047] 如果不存在正当地使系统 100 断电的请求，则运行控制器在步骤 530 中检查是否存在更新数据或将数据插入高速缓存设备 170 的高速缓存行的请求。例如，在本发明的一个实施例中，当 OS 想要将数据写至存储设备 160 中的特定地址位置时，运行控制器检查高速缓存表以判断存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果存在高速缓存命中，即在该特定地址位置的数据被高速缓存在高速缓存设备 170 中，则运行控制器接收一请求以更新存储特定地址位置数据的匹配高速缓存行。如果存在高速缓存未命中，即特定地址位置的数据不被高速缓存在高速缓存设备 170 中，则运行控制器接收一请求以将特定地址位置的数据插入高速缓存设备 170 的高速缓存行。

[0048] 如果存在更新数据或将数据插入高速缓存设备 170 的高速缓存行的请求，则运行控制器在步骤 532 基于要写入的新数据，在易失存储器 142 中更新与高速缓存行关联的分组元数据或状态信息。在步骤 534，运行控制器用新数据更新高速缓存行和存储设备 160。当步骤 534 结束时，高速缓存设备 170 和存储设备 160 中的数据是同步的。

[0049] 如果没有请求更新数据或将数据插入高速缓存设备 170 的高速缓存行，则复原控制器在步骤 540 检查是否存在任何系统 100 的上电通知。如果是，则复原控制器在步骤 542 恢复高速缓存设备 170 中经分组的元数据或将其复制入易失存储器 142。在可选步骤 544，如果高速缓存表已在系统 100 停止运转之前被保存，则复原控制器恢复高速缓存设备 170 中的高速缓存表或将其复制至易失存储器 142，随后流程 500 返回到步骤 510。

[0050] 如果不是，运行控制器在步骤 550 检查是否存在从高速缓存设备 170 读取数据的请求。例如，在本发明的一个实施例中，当 OS 想要从存储设备 160 中的特定地址位置读取数据时，运行控制器接收请求以从高速缓存设备 170 读取数据。如果存在从高速缓存设备 170 读取数据的请求，则运行控制器在步骤 552 检查高速缓存表以判断在存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果不存在从高速缓存设备 170 读取数据的请求，则流程返回到步骤 510。

[0051] 在步骤 554，运行控制器检查是否存在高速缓存命中，即存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果是，运行控制器在步骤 556 从高速缓存设备 170 读取数据并将数据返回给 OS，然后流程 500 回到步骤 510。如果不是，运行控制器在步骤 558 将高速缓存未命中发送给 OS。在本发明的一个实施例中，当存在高速缓存未命中时，运行控制器在步骤 558 访问存储设备 160 中特定地址位置的数据并将数据返回给 OS，然后流程 500 返回到步骤 510。

[0052] 在本发明的一个实施例中，当利用直写高速缓存方案时，运行控制器不在运行过程中在高速缓存设备 170 中写入或更新分组数据。由于存储设备 160 和高速缓存设备 170 中的数据永远是同步的，当例如功率丢失事件的故障事件发生时，可将高速缓存设备 170 复位。由于即使在功率丢失事件中也能维持存储设备 160 中的数据完整，因此系统 100 是

防电源故障的。

[0053] 图 6A 示出根据本发明一个实施例的回写高速缓存方案的流程图 600。在步骤 610 中, 运行控制器检查是否存在更新高速缓存设备 170 的高速缓存行的请求。如果存在更新高速缓存行的请求, 则运行控制器在步骤 612 用新数据更新相关的高速缓存行。在步骤 614, 运行控制器基于要写入到易失存储器 142 中的新数据更新与高速缓存行关联的分组元数据或状态数据。在步骤 616, 运行控制器基于要写入高速缓存设备 170 的新数据更新与高速缓存行关联的分组元数据或状态信息。在本发明的另一实施例中, 运行控制器在步骤 616 将易失存储器 142 中与高速缓存行关联的分组元数据或状态信息复制入与高速缓存设备 170 的分组元数据 401 的逻辑段中的高速缓存行关联的相关分组元数据中。流程 600 在步骤 616 结束后返回到步骤 610。

[0054] 如果不存在更新高速缓存行的请求, 则运行控制器在步骤 620 检查是否存在请求以正当地使系统 100 的断电。如果存在正当地使系统 100 掉电的请求, 则运行控制器在可选步骤 624 将高速缓存表从易失存储器 142 复制至高速缓存设备 170, 然后流程 600 返回到步骤 610。如果不存在正当地使系统 100 的断电的请求, 则运行控制器在步骤 630 检查 OS 是否已发布刷新命令。如果 OS 已发布刷新命令, 则运行控制器在步骤 632 刷新存储设备 160 和高速缓存设备 170 两者中的任意易失数据。

[0055] 如果 OS 尚未发布刷新命令, 则复原控制器在步骤 640 检查是否存在任何系统 100 的上电通知。如果是, 则复原控制器在步骤 642 恢复高速缓存设备 170 中的分组元数据并将其复制入易失存储器 142。在可选步骤 644 中, 如果高速缓存表已在系统 100 停止运转之前被保存, 则复原控制器恢复高速缓存设备 170 中的高速缓存表或将其复制至易失存储器 142, 然后流程 600 返回到步骤 610。

[0056] 如果不是, 运行控制器在步骤 650 检查是否存在从高速缓存设备 170 读取数据的请求。如果存在从高速缓存设备 170 读取数据的请求, 则运行控制器在步骤 652 检查高速缓存表以判断存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果不存在从高速缓存设备 170 读取数据的请求, 则流程返回到步骤 610。

[0057] 在步骤 654, 运行控制器检查是否存在高速缓存命中, 即存储设备 160 中特定地址位置的数据被高速缓存入高速缓存设备 170。如果是, 运行控制器在步骤 656 从高速缓存设备 170 读取数据并将数据返回给 OS, 然后流程 600 回到步骤 610。如果不是, 运行控制器在步骤 658 将高速缓存未命中发送给 OS。在本发明的一个实施例中, 当存在高速缓存未命中时, 运行控制器在步骤 658 访问存储设备 160 中特定地址位置的数据并将数据返回给 OS, 随后流程 600 回到步骤 610。图 6A 的回写高速缓存方案要求对高速缓存设备 170 附加写入操作, 以对新数据的每次高速缓存行写操作更新与这些高速缓存行关联的分组元数据。

[0058] 图 6B 示出根据本发明一个实施例的回写高速缓存方案的流程图 660。在步骤 610 中, 运行控制器检查是否存在更新高速缓存设备 170 的高速缓存行的请求。如果存在更新高速缓存行的请求, 则运行控制器在步骤 612 用新数据更新相关高速缓存行。在步骤 614, 运行控制器基于要写入的新数据在易失存储器 142 更新与高速缓存行关联的分组元数据或状态信息。在步骤 615, 运行控制器基于高速缓存设备 170 中的新数据将高速缓存行标记为等待与高速缓存行关联的分组元数据的更新。在本发明的一个实施例中, 运行控制器通过将高速缓存行添加至等待元数据写入的易失存储器 142 中的清单而对高速缓存行作出

标记。等待元数据写入的清单包括具有在易失存储器 142 和高速缓存设备 170 之间没有同步的关联的分组元数据的高速缓存行。

[0059] 如果不存在更新高速缓存行的请求，则运行控制器在步骤 620 检查是否存在正当地使系统 100 断电的请求。如果存在正当地使系统 100 断电的请求，则运行控制器将易失存储器 142 中全部等待的分组元数据写入高速缓存设备 170 中的分组元数据。在本发明的一个实施例中，运行控制器从等待元数据写入的清单确定要更新或写入哪个元数据。在可选步骤 624 中，运行控制器将高速缓存表从易失存储器 142 复制至高速缓存设备 170，然后流程 660 回到步骤 610。

[0060] 如果不存在正当地使系统 100 断电的请求，则运行控制器在步骤 630 检查 OS 是否已发布刷新命令。如果已发布刷新命令，则运行控制器在步骤 631 将易失存储器 142 中的全部等待的分组元数据更新为高速缓存设备 170 中的分组元数据。在本发明的另一实施例中，运行控制器在步骤 631 的一次连续写操作中将全部分组元数据从易失存储器 142 更新或复制至高速缓存设备 170。在步骤 632，运行控制器刷新存储设备 160 和高速缓存设备 170 中的任何易失数据。

[0061] 如果没有发布刷新命令，复原控制器在步骤 640 检查是否存在任何系统 100 的上电通知。如果是，复原控制器在步骤 642 将高速缓存设备 170 中的分组元数据恢复或复制入易失存储器 142。在可选步骤 644 中，如果在系统 100 停止运转之前已保存高速缓存表，则复原控制器将高速缓存设备 170 中的高速缓存表恢复或复制至易失存储器 142，流程 660 回到步骤 610。

[0062] 如果不是，运行控制器在步骤 650 检查是否存在从高速缓存设备 170 读取数据的请求。如果存在从高速缓存设备 170 读取数据的请求，则运行控制器在步骤 652 检查高速缓存表以判断在存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果不存在从高速缓存设备 170 读取数据的请求，则流程 660 返回到步骤 610。

[0063] 在步骤 654，运行控制器检查是否存在高速缓存命中，即存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果是，运行控制器在步骤 656 从高速缓存设备 170 读取数据并将数据返回给 OS，然后流程 660 回到步骤 610。如果不是，运行控制器在步骤 658 将高速缓存未命中发送给 OS。在本发明的一个实施例中，当存在高速缓存未命中时，运行控制器在步骤 658 访问存储设备 160 中特定地址位置的数据并将数据返回给 OS，然后流程 660 返回到步骤 610。图 6B 的回写高速缓存方案需要对高速缓存设备 170 的可选附加写入，以对应每次刷新或断电事件更新与高速缓存行关联的分组元数据。

[0064] 图 6C 示出根据本发明一个实施例的回写高速缓存方案的流程图 680。图 6C 是结合图 6B 讨论的，因为流程 680 是流程 660 的变化形式。除了步骤 631，流程 660 中的所有步骤均适用于流程 680，并且不再对这些步骤予以重复说明。在流程 680 中，在步骤 630 从 OS 接收刷新命令后，运行控制器在步骤 662 检查等待的元数据写入清单中是否存在邻近的等待写入。作为示例，假设等待元数据写入清单具有针对七个高速缓存行（高速缓存行 5、6、7、9、12、13 和 45）的等待元数据写入。

[0065] 在假定的场景中，由于高速缓存行 5、6、7 是紧邻的且高速缓存行 12 和 13 也是紧邻的，流程 680 进至步骤 664。在步骤 664 中，运行控制器将针对高速缓存行 5、6 和 7 的元数据写入合并为一次元数据写入。另外将针对高速缓存行 12 和 13 的元数据写入合并为另

一次元数据写入。因此，运行控制器具有四次元数据写入（5、6 和 7、9 的合并写入，12、13 和 45 的合并写入）而不是原始的七次元数据写入。在步骤 670，运行控制器执行步骤 664 的四次元数据写入。

[0066] 在另一示例中，假设等待元数据写入的列表具有针对五个高速缓存行（高速缓存行 3、9、11、14 和 45）的等待元数据写入。在假定的场景中，流程 680 进至步骤 662 以检查在等待高速缓存行要写入的地址位置是否存在小间隔。在本发明的一个实施例中，如果一起写入高速缓存行所花费的时间短于将这些高速缓存行单独写入所花费的时间，则认为高速缓存行之间的间隔小。例如，如果更新与高速缓存行 9、10、11 关联的元数据所需的时间短于单独更新与高速缓存行 9、11 关联的元数据所需的时间，则认为高速缓存行 9、11 之间的间隔小。在本发明的一个实施例中，即便不需要更新与高速缓存行 10 关联的元数据，将高速缓存行的元数据更新结合起来仍然减少了更新高速缓存行所需的时间。

[0067] 在假定的场景中，高速缓存行 9、11 之间的间隔以及高速缓存行 11、14 之间的间隔假设为小并且流程进至步骤 668。在步骤 668，运行控制器将其间具有小间隔的高速缓存行合并为一个大的元数据高速缓存写。例如，在假定的场景中，运行控制器将针对高速缓存行 9、11 和 14 的元数据更新结合为针对高速缓存行 9-14 的一次元数据更新，即便高速缓存行 10、12 和 13 不需要修正。在步骤 670，运行控制器执行步骤 664 中的组合元数据写入并且流程进至流程 660 中的步骤 634。步骤 664 和 668 优化操作以更新高速缓存设备 170 中等待的分组元数据。在流程 680，在本发明的另一实施例中，只执行步骤 662、664 和步骤 666、668 中的一个步骤。相关领域内技术人员很容易理解可进行其它优化以减少更新高速缓存设备 170 中等待的元数据更新的时间而不会影响本发明的工作。

[0068] 图 6A、6B 和 6C 所示的回写高速缓存方案不解释为限定。相关领域内技术人员很容易理解可实现步骤的多种组合或修正而不影响本发明工作。系统 100 的用户可确定利用图 6A、6B 和 6C 中三个回写高速缓存方案中的一个并也可利用图 6A、6B 和 6C 的三个回写高速缓存方案的任意组合形式。

[0069] 图 7 示出根据本发明一个实施例的将数据插入高速缓存设备 170 的高速缓存行的方法的流程图 700。在步骤 710，运行控制器检查是否存在将数据插入高速缓存设备 170 的高速缓存行的请求。例如在本发明的一个实施例中，当 OS 想要将数据写至存储设备 160 的特定地址位置时，运行控制器检查高速缓存表以判断存储设备 160 中特定地址位置的数据是否被高速缓存在高速缓存设备 170 中。如果不存在高速缓存命中，即特定地址位置的数据未高速缓存在高速缓存设备 170 中，则运行控制器可接收请求以将数据插入高速缓存设备 170 的高速缓存行。

[0070] 如果不存在将数据插入高速缓存行的请求，流程结束。如果存在将数据插入高速缓存行的请求，则运行控制器在步骤 720 检查高速缓存设备 170 中是否存在任何空闲的高速缓存行。在本发明的一个实施例中，高速缓存设备 170 中全部未使用的高速缓存行被著录或标记为空闲的高速缓存行。在本发明的另一实施例中，高速缓存设备 170 中未使用的高速缓存行的固定部分被著录或标记为空闲高速缓存行。例如在一个实施例中，运行控制器可将高速缓存设备 170 的五个高速缓存行标记为空闲高速缓存行。如果不存在空闲高速缓存行，则运行控制器在步骤 722 基于逐出策略选择高速缓存设备 170 要被逐出的一个或多个高速缓存行。逐出策略包括但不限于逐出最近很少使用的高速缓存行、逐出高速缓

存设备 170 的第一高速缓存行等。

[0071] 在步骤 724, 所选择的高速缓存行被运行控制器逐出。在本发明的一个实施例中, 运行控制器通过将所选高速缓存行中已高速缓存的数据 (如果其尚未同步) 写至存储设备 160 而逐出所选择的高速缓存行。在步骤 726, 运行控制器将逐出的高速缓存行著录或标记为空闲高速缓存行并且流程进至步骤 730。如果存在空闲的高速缓存行, 则运行控制器在步骤 730 选择高速缓存设备 170 的一个或多个空闲的高速缓存行以高速缓存要写入的数据。空闲高速缓存行的选择策略包括但不限于, 首先可用的空闲高速缓存行、最近很少使用的空闲高速缓存行等。在步骤 740, 运行控制器将数据写至选定的空闲高速缓存行。

[0072] 在步骤 750, 运行控制器基于新数据更新易失存储器 142 中与所选高速缓存行关联的分组元数据或状态信息。在步骤 750 后, 如果利用流程 660 或 680 的回写高速缓存方案, 流程 700 执行可选步骤 760, 其中运行控制器基于新数据将高速缓存设备 170 中的高速缓存行标记为等待更新与高速缓存行关联的分组元数据, 或如果利用流程 600 的回写高速缓存方案, 则执行可选步骤 770, 其中运行控制器基于新数据更新与高速缓存设备 170 中的所选高速缓存行关联的分组元数据或状态信息。流程在可选步骤 760 或 770 完成后结束。

[0073] 由于例如系统 100 电源故障的故障事件可能导致存储设备 160 和高速缓存设备 170 中的数据完整性问题, 高速缓存行的逐出需要立即更新与高速缓存设备 170 中的高速缓存行关联的分组元数据。如此, 高速缓存行的逐出在每次逐出后都需要与高速缓存设备 170 中的高速缓存行关联的元数据写入。然而, 在高速缓存行每次逐出后执行附加的元数据写入引发开销。为了避免开销, 图 7 展示的将数据插入高速缓存设备 170 的高速缓存行的方法包括将新数据插入空闲的高速缓存行而不是具有高速缓存数据的高速缓存行。

[0074] 例如, 为了便于说明, 假设运行控制器接收对存储设备 160 的 LBA1 插入数据的请求。假设高速缓存行 4 从存储设备 160 的 LBA5 开始存储数据。如果在用来自存储设备 160 的 LBA1 的数据写入高速缓存行 4 之后但在更新与高速缓存行 4 关联的元数据之前发生故障事件, 系统 100 一旦重启或重引导事件发生就会看到高速缓存行 4 具有基于与高速缓存行 4 关联的元数据的来自 LBA5 的数据。然而这是错误的, 由于高速缓存行 4 已用来自存储设备 160 的 LBA1 的数据予以更新。

[0075] 通过如图 7 的流程 700 所述地将新数据插入空闲高速缓存行, 发生的故障事件不影响存储设备 160 和高速缓存设备 170 的数据完整性。例如, 为了便于说明, 当运行控制器接收将来自存储设备 160 的 LBA1 的数据插入高速缓存设备 170 的请求时, 运行控制器选择空闲的高速缓存行以高速缓存来自存储设备 160 的 LBA1 的数据。如果故障事件发生在用来自存储设备 160 的 LBA1 的数据更新空闲高速缓存行之后但发生在更新与空闲高速缓存行关联的元数据之前, 则故障事件不影响存储设备 160 和高速缓存设备 170 的数据完整性。由于发生的是故障事件而不是刷新事件, 因此可丢弃新数据而不会影响系统 100。

[0076] 直写高速缓存方案不局限于图 5 所示的算法。在本发明的另一实施例中, 直写高速缓存方案可利用图 6A、6B 和 6C 所示的回写高速缓存算法之一。回写高速缓存方案可利用图 6A、6B 和 6C 所示的回写高速缓存算法以及图 7 的算法之一。如果直写高速缓存方案利用图 6A、6B 和 6C 所示回写高速缓存算法和图 7 所示算法之一, 则直写高速缓存在不当地停止运作期间也能保持激活。

[0077] 图 8A 和 8B 示出用于实现本发明一个实施例的回写高速缓存方案的伪代码 800 和

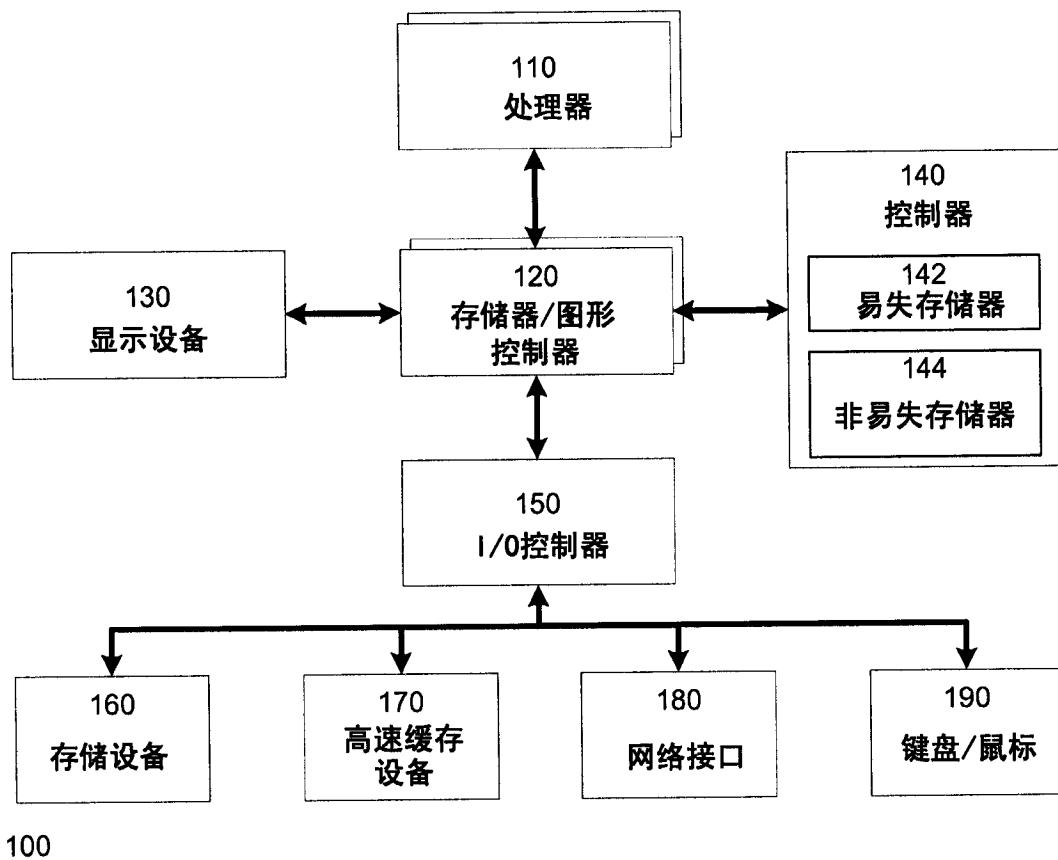
850。为了便于说明,将 HDD 作为存储设备 160 的例示并将 SSD 作为高速缓存设备 170 的例示。相关领域内技术人员很容易理解伪代码 800、850 的工作机理并不再对其进行详细说明。

[0078] 尽管已描述了所披露的主题事项的实施例示例,然而本领域内技术人员很容易理解可代替地使用实现所披露主题事项的许多其它方法。在前面的描述中已记载了所披露主题事项的多个方面。为了便于说明,给出具体的数目、系统和配置以提供对主题事项的透彻理解。然而,本领域内技术人员可从实践本公开明确主题事项得益于而无需具体细节。在其它情形下,将公知的特征、组件或模块省去、简化、组合或分割以不混淆所披露的主题事项。

[0079] 术语“可作用”在本文中表示设备、系统、协议等当处于掉电状态时可作用或可适应地实现其想要的功能。所披露主题事项的各个实施例可实现为硬件、固件、软件或其组合,或参照或结合例如指令、功能、进程、数据结构、逻辑、应用程序、当由机器访问时致使机器执行任务和定义关键数据类型或低层硬件背景或产生结果的设计的模拟、模仿和制作的设计表示或格式的程序代码进行描述。

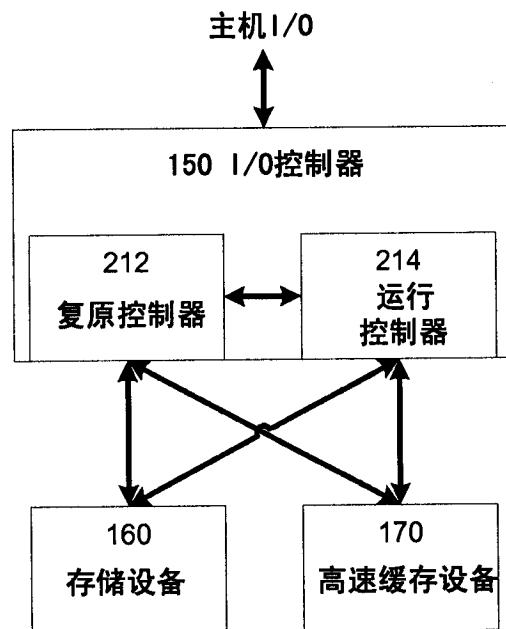
[0080] 附图中示出的技术可使用在例如通用计算机或计算设备的一个或多个计算设备上存储和执行的代码和数据来实现。这些计算设备存储并传输(在内部或与网络上的其它计算设备)代码和数据,例如运用机器可读存储介质(例如磁盘、光盘、随机存取存储器、只读存储器、闪存设备、相变存储器)和计算机可读通信介质(例如电、光、声或其它形式的传播信号——例如载波、红外信号、数字信号等)。

[0081] 尽管已结合示例性实施例对所披露的主题事项进行了说明,然而这种说明不应当解释成限定的意思。本领域内技术人员所熟知的主题事项的示例性实施例以及其它实施例的多种修正形式视为落在所披露主题事项的范围内。



100

图 1



200

图 2

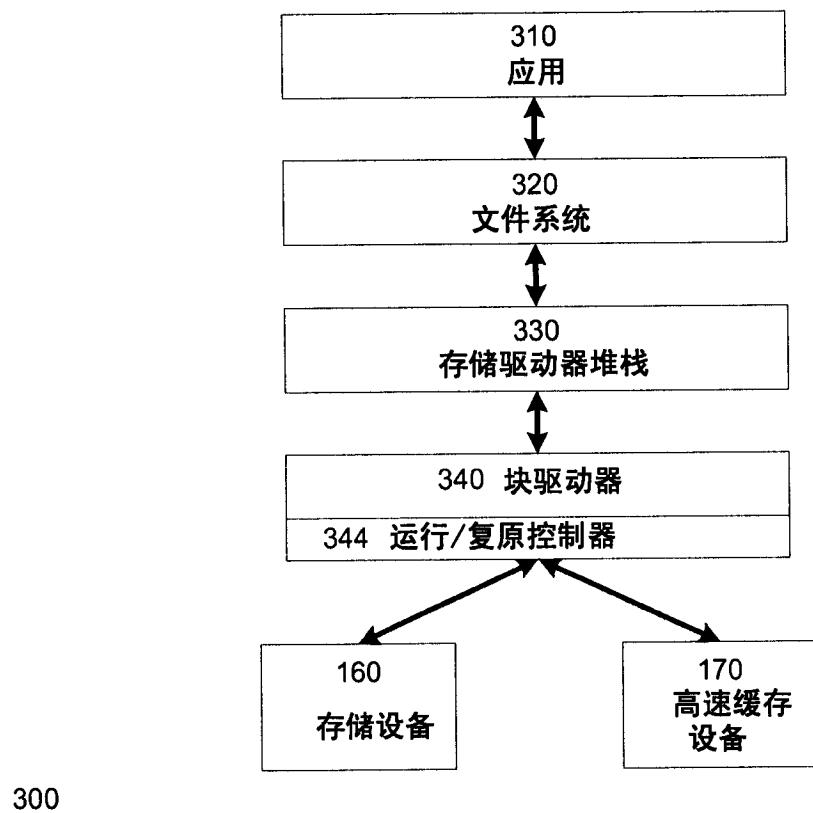


图 3

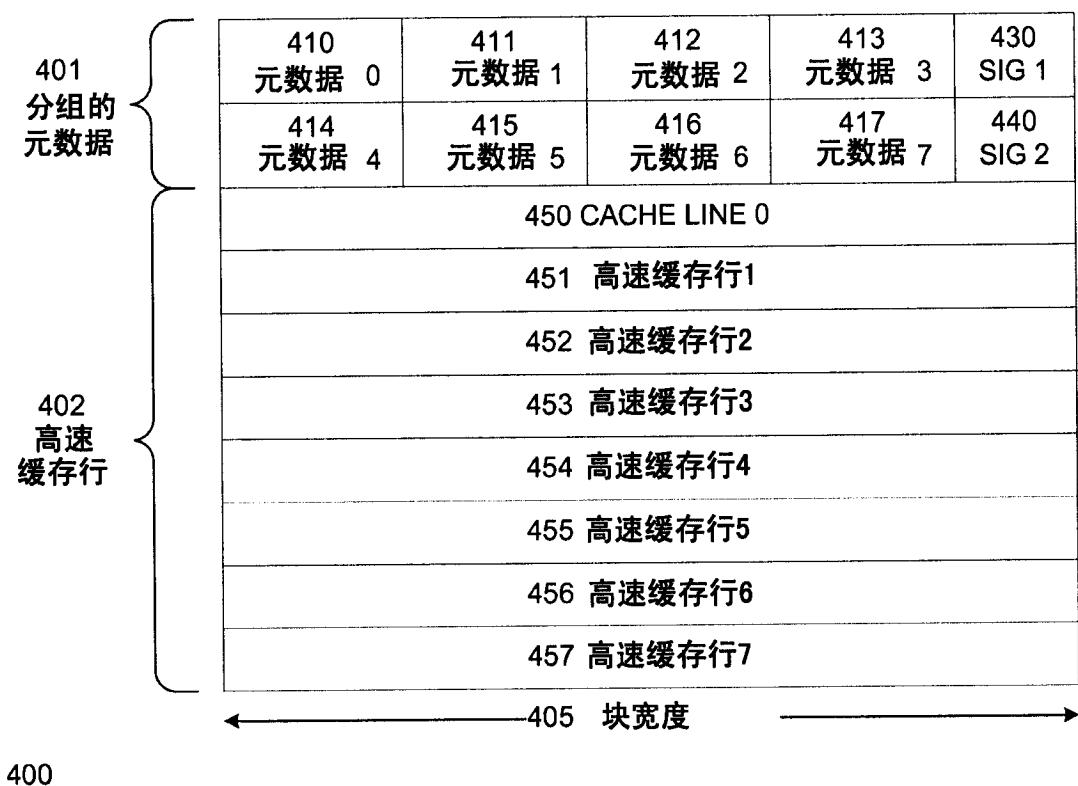
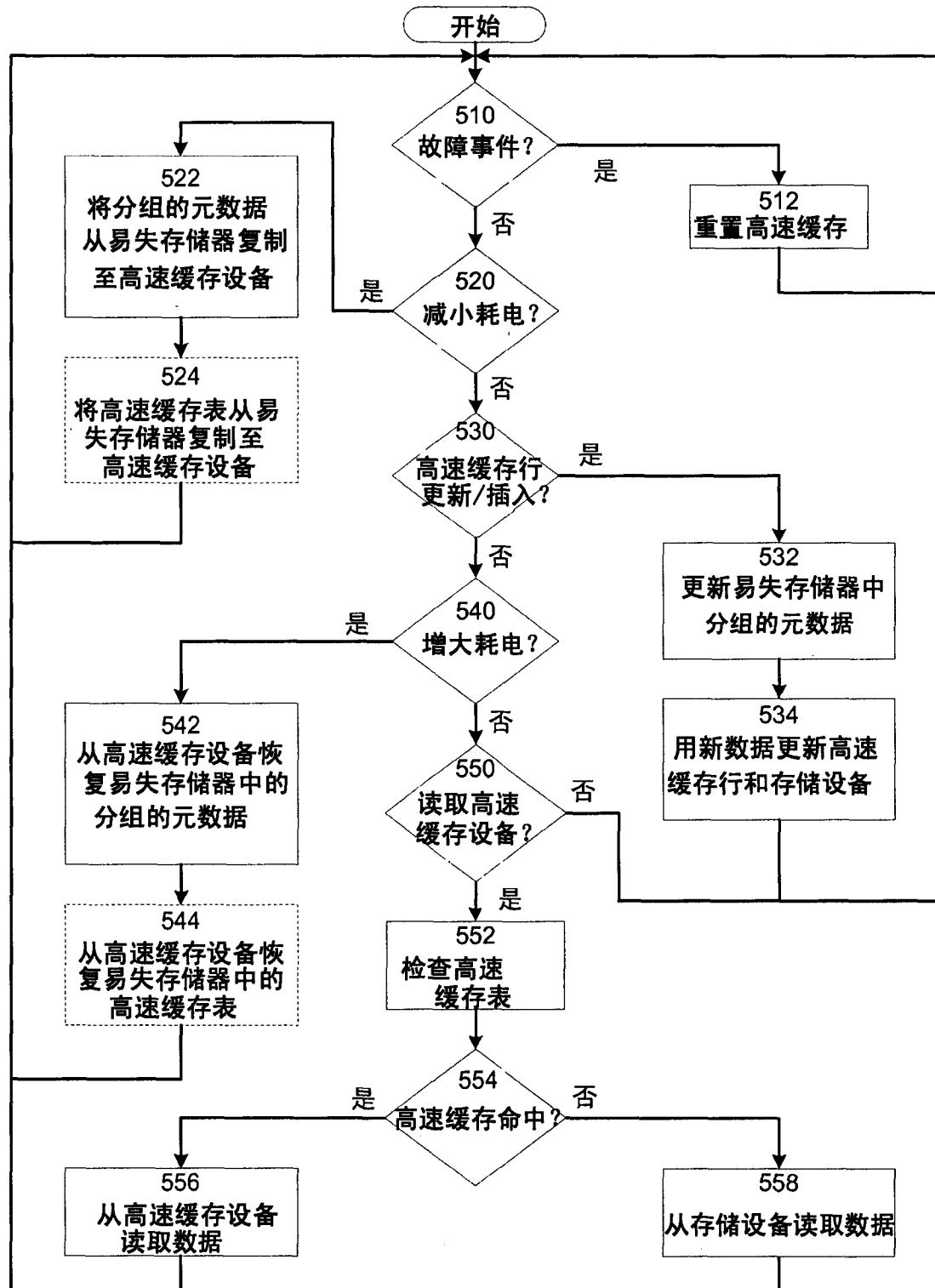
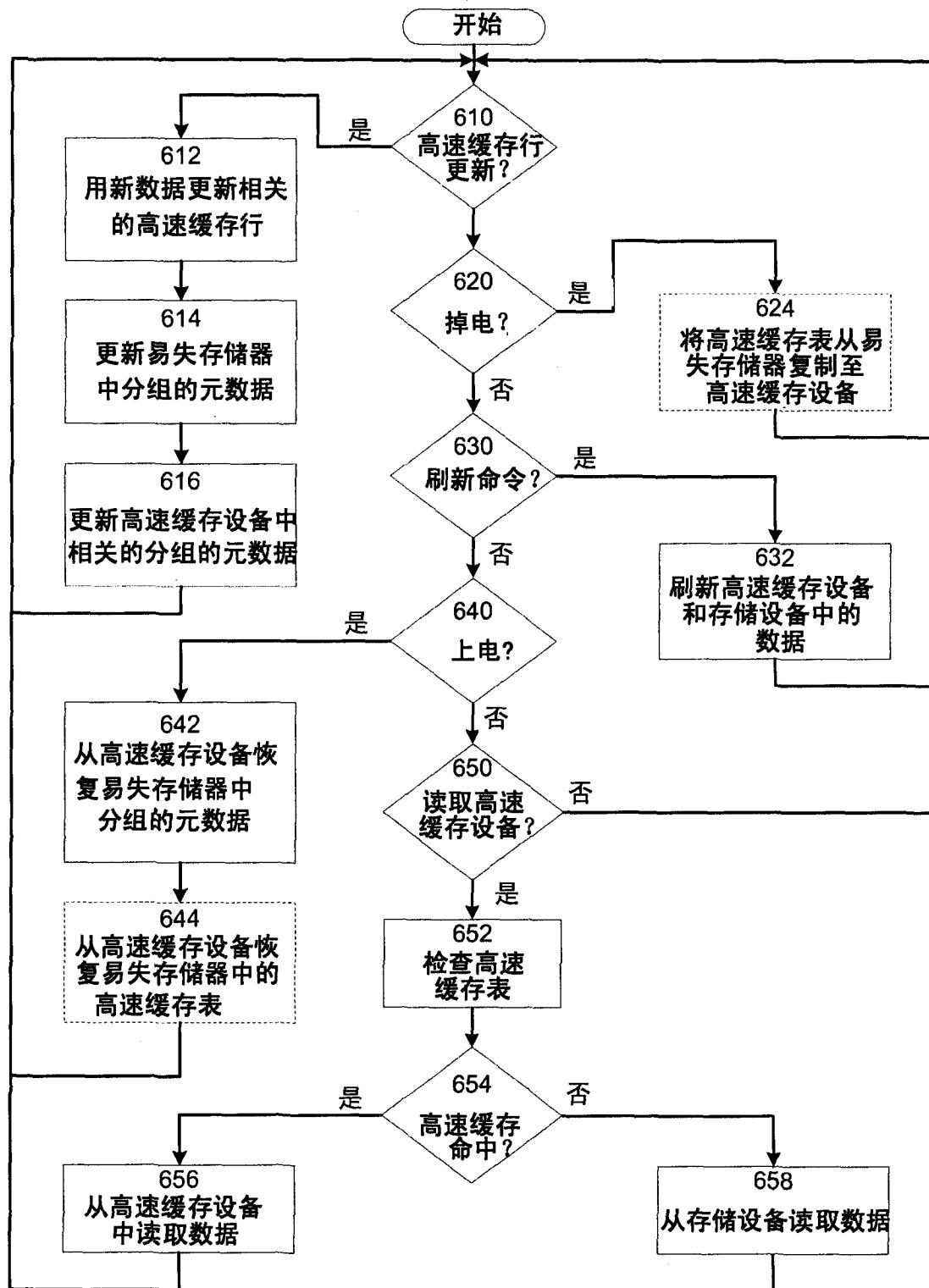


图 4



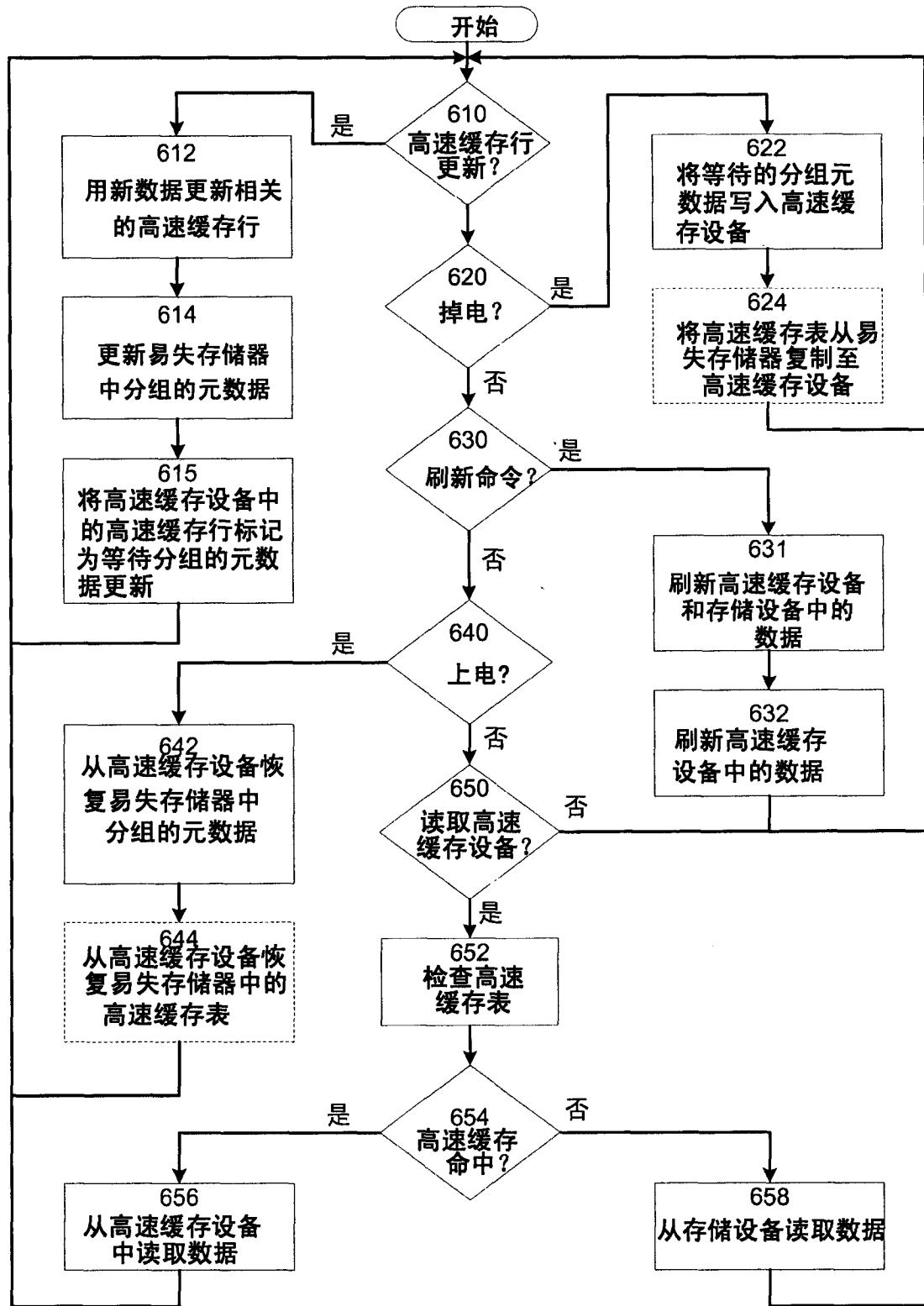
500

图 5



600

图 6A



660

图 6B

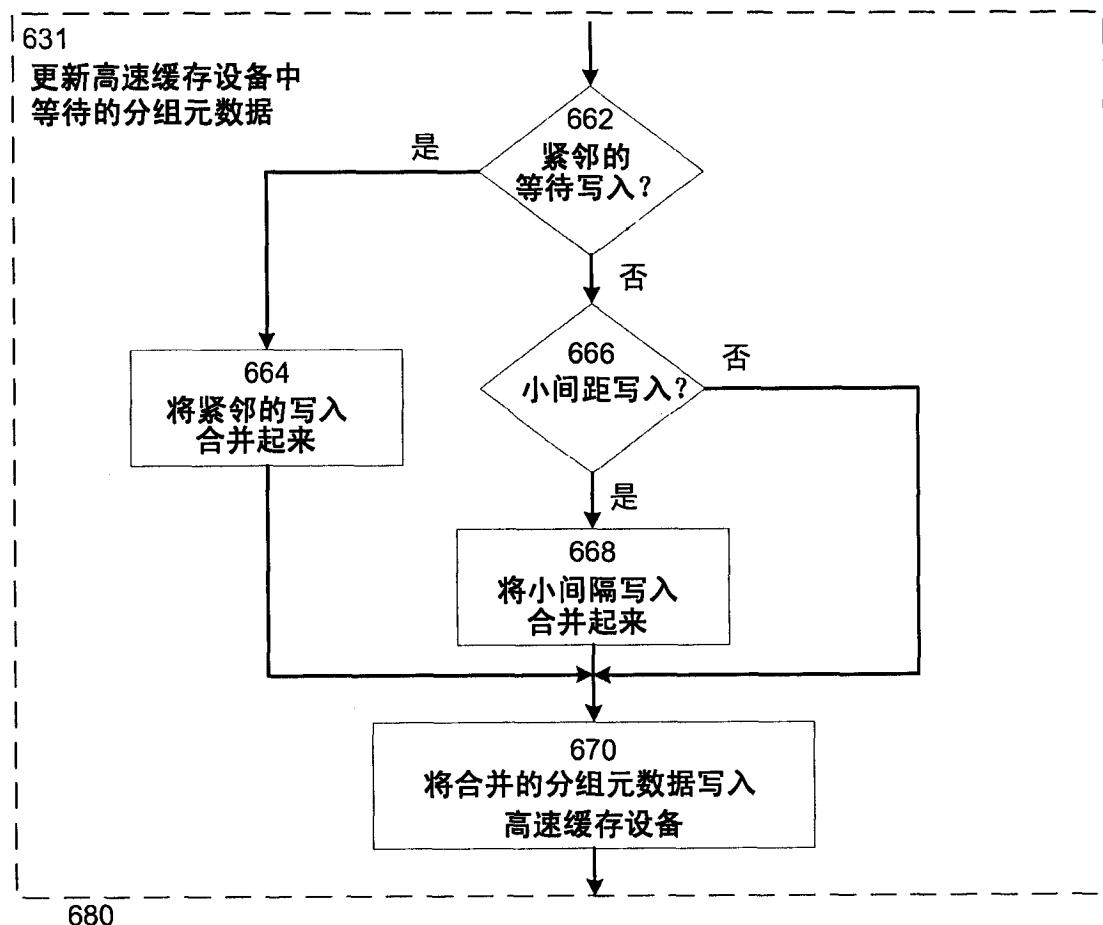
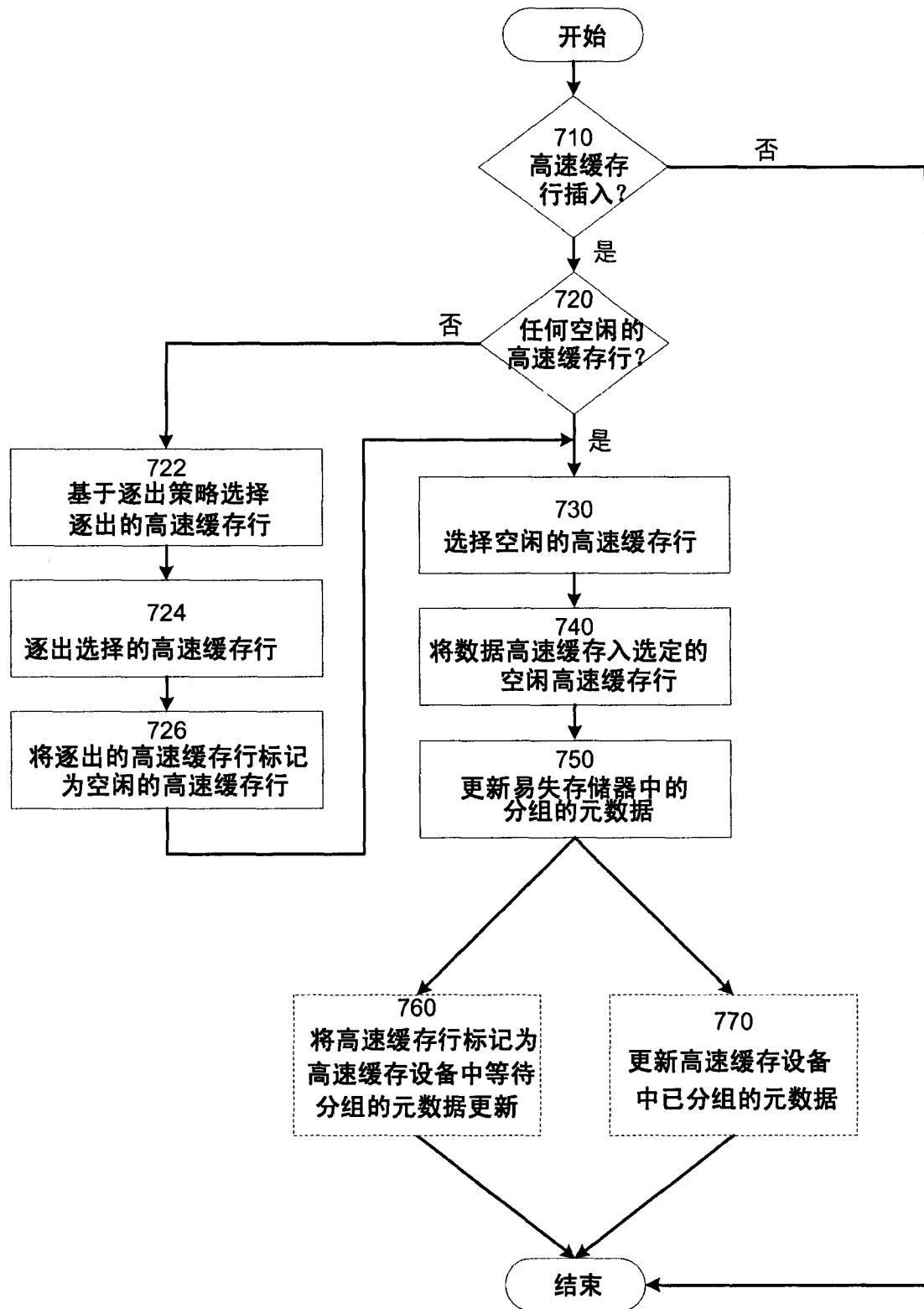


图 6C



700

图 7

```

// Volatile structures to maintain
SPARES = List of cache lines that do not hold data
LBA2CL = Cache Table. Maps LBA to a cache line. LBA2CL[x] is the cache line that
holds the data for LBA x, or is -1 otherwise.
META = contains metadata for all the cache lines. META[cl] is the metadata
corresponding to cache line cl. META[cl] contains the LBA number of the cached HDD
sector, and may contain other information.
PENDING_META_WRITES = list of cache lines numbers for which the metadata has
not been written non-volatilely.

// Layout of the data on the non-volatile media
NV_META = Non-volatile metadata. NV_META[cl] contains metadata for cacheline cl.
NV_CACHE = Cache data. NV_CACHE[cl] contains the user data stored in cacheline
cl.

// InitializeNvStructures (Done only when cache is initialized or reset)
For cl = 0 to (NumCachelines – 1)
    nvmeta[cl].LBA = -1
End for
Write nvmeta to NV_META

// InitializeVolatileStructures
// Initializes the volatile structures from the information stored non-volatilely.
// Done on every startup (proper or improper)
SPARES = {}
PENDING_META_WRITES = {}
Read nvmeta from NV_META
For cl = 0 to NumCachelines – 1
    lba = nvmeta[cl].lba
    META[cl] = nvmeta[cl]
    If (lba >= 0)
        LBA2CL[lba] = cl;
    Else
        SPARES = SPARES union { cl }
    End if
End for

// ProperShutdown
CacheFlush

```

```

// Insert (LBA L, Data D, Metadata M)
// Inserts the specified data and metadata into the cache.
// The metadata remains volatile until a subsequent flush completes.
If SPARES is empty then
    IbaList = PickLBAsToEvict ()           // runs the cache's eviction policy.
    Evict(IbaList)                      // may call this multiple times to evict multiple
                                         // cachelines. This adds an element to
                                         // SPARES.
Endif cl = SPARES.RemoveElement ()
Write D to NV_CACHE[cl]
PENDING_META_WRITES = PENDING_META_WRITES union { (cl,L) }
META[cl] = M
LBA2CL[L] = cl

// Evict (IbaList)
// Removes the specified cachelines from the cache.
For each L in IbaList
    cl = LBA2CL[L]
    Clean the cacheline cl if its dirty (i.e., if not clean, read data from cache and
    write to disk).
    META[cl] = zeroes
    LBA2CL[L] = -1
    SPARES = SPARES union { cl }
    PENDING_META_WRITES = PENDING_META_WRITES union { (cl,L) }
End for UpdateNvMetadata ()

// ReadFromCache (LBA L)
// Reads data from the cache, or returns an error if it's a miss
cl = LBA2CL[L]
if (cl < 0)
    return NOT_IN_CACHE
end if
Read SSD data from NV_CACHE[cl], and return it.

// CacheFlush
// Ensures that all data and metadata in the cache is non-volatile
UpdateNvMetadata ()
Issue Flush to the SSD

// UpdateNvMeta
// Writes any pending metadata writes to the cache
For each cl in PENDING_META_WRITES
    X = SSD sector that contains NV_META[cl]
    Update SSD sector X with information in META[cl]
End for

```