



(12) 发明专利

(10) 授权公告号 CN 110471618 B

(45) 授权公告日 2023. 07. 18

(21) 申请号 201910385887.3

(22) 申请日 2019.05.09

(65) 同一申请的已公布的文献号
申请公布号 CN 110471618 A

(43) 申请公布日 2019.11.19

(30) 优先权数据
15/976,034 2018.05.10 US

(73) 专利权人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四
层847号邮箱

(72) 发明人 周平 杜宇 李舒

(74) 专利代理机构 北京清源汇知识产权代理事
务所(特殊普通合伙) 11644
专利代理师 冯德魁 窦晓慧

(51) Int.Cl.

G06F 3/06 (2006.01)

G06F 16/18 (2019.01)

(56) 对比文件

US 2012159040 A1,2012.06.21

US 2013117503 A1,2013.05.09

US 2017177541 A1,2017.06.22

Qingsong Wei et al..Extending SSD

Lifetime with Persistent In-memory

Metadata Mangement.《2016 IEEE

International Conference on Cluster

Computing》.2016,第308页-311页.

审查员 王田

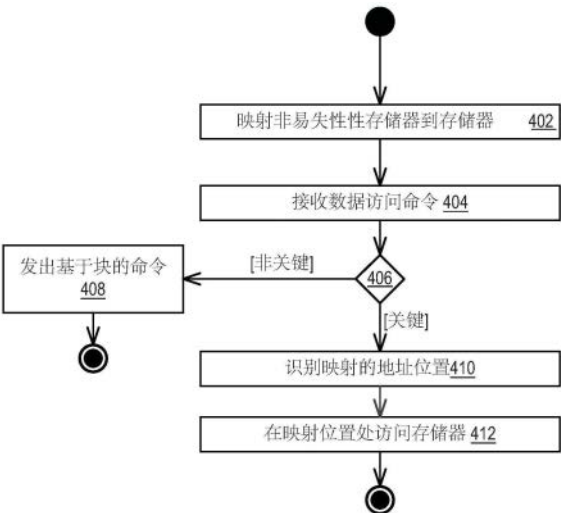
权利要求书2页 说明书13页 附图9页

(54) 发明名称

快速侧通道访问存储设备

(57) 摘要

公开了用于向存储设备提供高速数据路径的系统、方法和装置。在一个实施例中,公开了一种方法,所述方法包括由处理器接收数据访问命令,所述数据访问命令指定存储器中访问数据的位置;所述处理器通过第一数据路径向存储设备发出数据访问命令,所述第一数据路径包括非块数据路径;以及所述处理器通过所述第一数据路径和所述存储器访问所述非易失性存储组件,其中,所述存储设备的非易失性存储组件被映射到所述处理器可访问的存储器。



1. 一种访问存储数据的方法,所述方法包括:

由处理器接收数据访问命令,所述数据访问命令指定存储器中访问数据的位置;

由所述处理器通过第一数据路径向存储设备发出所述数据访问命令,所述第一数据路径包括非块数据路径;以及

由所述处理器通过所述第一数据路径和所述存储器访问非易失性存储组件,其中,所述存储设备的非易失性存储组件被映射到所述处理器能访问的存储器;通过所述第一数据路径和所述存储器访问所述非易失性存储组件包括将所述非易失性存储组件作为字节可寻址读高速缓存或字节可访问写高速缓存进行访问。

2. 如权利要求1所述的方法,将所述存储设备的存储器的非易失性存储组件映射到所述处理器能访问的存储器包括将动态随机存取存储器DRAM的一部分映射到存储器。

3. 如权利要求2所述的方法,将DRAM的一部分映射到存储器包括使用所述存储设备的外围组件互连PCI基址寄存器BAR将DRAM的一部分映射到存储器。

4. 如权利要求3所述的方法,将DRAM的一部分映射到存储器包括识别要映射的DRAM的所述部分的大小,所述识别包括基于提供给所述DRAM的断电电容计算最大部分大小。

5. 如权利要求1所述的方法,通过所述第一数据路径和所述存储器访问所述非易失性存储组件包括将基础数据访问命令的日志数据写入所述非易失性存储组件。

6. 如权利要求1所述的方法,将所述存储设备的存储器的非易失性存储组件映射到所述处理器能访问的存储器包括将存储级存储器SCM的一部分映射到所述存储器。

7. 根据权利要求6所述的方法,将SCM的一部分映射到存储器包括将相变存储器PCM、电阻随机存取存储器ReRAM或自旋转移扭矩RAM STT-RAM中的一个映射到所述存储器。

8. 如权利要求6所述的方法,将SCM的一部分映射到存储器包括经由控制器存储缓冲器CMB窗口将所述SCM映射到所述存储器。

9. 如权利要求1所述的方法,所述方法还包括:在通过所述第一数据路径接收到访问所述非易失性存储组件的结果之后,由所述处理器经由第二数据路径发出基于块的数据访问命令。

10. 一种访问存储数据的系统,所述系统包括:

存储设备,所述存储设备包括非易失性存储组件;

处理器;

存储器,所述存储器用于在其上有形地存储用于由所述处理器执行的程序逻辑,所存储的程序逻辑包括:

由所述处理器执行的、用于接收数据访问命令的逻辑,所述数据访问命令指定主机存储器中访问数据的位置;

由所述处理器执行的、用于经由第一数据路径向所述存储设备发出所述数据访问命令的逻辑,所述第一数据路径包含非块数据路径;以及

由所述处理器执行的、用于通过所述第一数据路径和所述存储器访问所述非易失性存储组件的逻辑,其中,所述存储设备的所述非易失性存储组件被映射到所述存储器;通过所述第一数据路径和所述主机存储器访问所述非易失性存储组件的逻辑包括将所述非易失性存储组件作为字节可寻址读高速缓存或字节可访问写高速缓存进行访问。

11. 如权利要求10所述的系统,用于将所述非易失性存储组件映射到所述存储器的逻

辑包括由所述处理器执行的、用于将动态随机存取存储器DRAM的一部分映射到所述主机存储器的逻辑。

12. 如权利要求11所述的系统,用于将DRAM的一部分映射到主机存储器的逻辑包括由所述处理器执行的、用于使用所述存储设备的外围组件互连PCI基地址寄存器BAR将DRAM的一部分映射到存储器的逻辑。

13. 如权利要求12所述的系统,用于将DRAM的一部分映射到主机存储器的逻辑包括识别要映射的DRAM的所述部分的大小,所述识别包括基于提供给所述DRAM的断电电容计算最大部分大小。

14. 如权利要求10所述的系统,用于通过所述第一数据路径和所述主机存储器访问所述非易失性存储组件的逻辑包括由所述处理器执行的、用于将基础数据访问命令的日志数据写入所述非易失性存储组件的逻辑。

15. 如权利要求10所述的系统,用于将非易失性存储组件映射到所述存储器的逻辑包括由所述处理器执行的、用于将存储级存储器SCM的一部分映射到所述主机存储器的逻辑。

16. 如权利要求15所述的系统,将SCM的一部分映射到主机存储器的逻辑包括由所述处理器执行的、用于将相变存储器PCM、电阻随机存取存储器ReRAM或自旋转移扭矩RAM STT-RAM中的一个映射到所述主机存储器的逻辑。

17. 如权利要求15所述的系统,用于将SCM的一部分映射到主机存储器的逻辑包括由所述处理器执行的、用于经由控制器存储缓冲器CMB窗口将所述SCM映射到所述主机存储器的逻辑。

18. 如权利要求10所述的系统,所述程序逻辑还包括由所述处理器执行的、用于在通过所述第一数据路径接收到访问所述非易失性存储组件的结果之后经由第二数据路径发出基于块的数据访问命令的逻辑。

快速侧通道访问存储设备

[0001] 本申请包括受版权保护的材料。版权所有者不反对任何人对专利和商标局文件或记录中出现的专利公开进行传真复制,但在其他方面保留所有版权。

背景技术

[0002] 所公开的实施例涉及存储设备,并且具体地,涉及用于改进到存储设备和从存储设备的关键数据传输的延迟的系统、设备和方法。

[0003] 传统上,存储设备采用基于块的数据路径和文件访问机制。写入存储设备和从存储设备读取的数据通过多层处理以将系统调用(例如,基于文件路径的请求)转换为可由存储设备处理的基于块的命令。本质上,这些层将应用层读取/写入转换为可由底层存储结构(例如,NAND闪存)理解的基于块的读取/写入。

[0004] 对存储设备的基于块的访问提供了规模优势。与位或字节级访问相比,可以以更大的块读取/写入数据。因此,可以用较少数量的块写入较大的数据项。然而,使用基于块的文件访问模式负面地影响替代的输入/输出(I/O)模式。

[0005] 在许多设备中,除了实际数据之外还读取/写入元数据。所述元数据可以包括例如要写入存储设备的日志或日志数据。此元数据通常小,远小于实际数据。通常,此元数据也位于文件访问的关键执行路径上。也就是说,必须在读取/写入实际数据之前处理元数据。

[0006] 利用基于块的处理的当前设备使这些小的元数据访问经历与所有数据相同的基于块的处理。因此,元数据由客户端设备文件处理软件的多个层处理。此附加处理会增加请求延迟的显著的时间。另外,附加处理通常是不必要的并且不适合于随机访问读取/写入。例如,块处理导致大约10微秒的延迟时间,而PCI总线的延迟大约是1微秒。当发生关键路径元数据访问的延迟时,整个文件访问受到显著影响,因为文件访问必须等待元数据访问的结果。

[0007] 因此,在当前系统中存在与改进关键路径I/O(例如元数据访问)相关联的延迟的需要。

发明内容

[0008] 所公开的实施例通过建立侧通道数据路径来解决这些和其他问题,以允许快速的读取/写入,同时保留传统的基于块的I/O通道,例如通过向例如基于PCI和NVMe的存储设备提供高速数据通道。

[0009] 在一个实施例中,公开了一种方法,所述方法包括:由处理器将存储设备的非易失性存储组件映射到处理器可访问的存储器;由所述处理器接收数据访问命令,所述数据访问命令指定存储器中访问数据的位置;由所述处理器通过第一数据路径向存储设备发出所述数据访问命令,所述第一数据路径包括非块数据路径;以及由所述处理器通过所述第一数据路径和所述存储器访问所述非易失性存储组件。

[0010] 在另一个实施例中,公开了一种系统,所述系统包括:存储设备,所述存储设备包括非易失性存储组件;处理器;存储器,所述存储器用于在其上有形地存储用于由所述处理

器执行的程序逻辑,所存储的程序逻辑包括:由所述处理器执行的、用于映射非易失性存储组件到主机存储器的逻辑;由所述处理器执行的、用于接收数据访问命令的逻辑,所述数据访问命令指定主机存储器中访问数据的位置;由所述处理器执行的、用于通过第一数据路径向存储设备发出所述数据访问命令的逻辑,所述第一数据路径包含非块数据路径;以及由所述处理器执行的、用于通过所述第一数据路径和所述存储器访问所述非易失性存储组件的逻辑。

[0011] 如上所述,现有解决方案使用基于块的数据通道用于存储设备和主机设备之间的所有数据访问。将基于块的通道用于较小的数据访问(例如,元数据读取/写入、日志记录、高速缓存等)是低效的并且阻止了存储设备的整体I/O。本公开的实施例描述了通过引入高吞吐量、存储器映射的数据路径来解决该问题的技术方案,所述数据路径允许快速访问位于存储设备控制器(例如,现有DRAM或专用SCM)中的非易失性存储设备。所公开的实施例导致数据路径具有用于关键请求的明显更快的数据路径,从而将这些请求的吞吐量从10微秒提高到大约1微秒。

附图说明

[0012] 本公开的前述和其他目的、特征和优点将从如附图所示的实施例的以下描述中变得显而易见,其中附图标记在各个视图中指代相同的部分。附图不一定按比例绘制,而是将重点放在说明本公开的原理上。

[0013] 图1为示出根据本公开的一些实施例的计算系统的框图。

[0014] 图2为示出根据本公开的一些实施例的具有增强的元数据数据路径的计算系统的框图。

[0015] 图3为示出根据本公开的一些实施例的具有增强的元数据数据路径的计算系统的框图。

[0016] 图4为示出根据本公开的一些实施例的使用存储器映射地址来访问存储级存储设备中的数据的方法的流程图。

[0017] 图5为示出根据本公开的一些实施例的用于处理存储设备操作的方法的流程图。

[0018] 图6A为根据本公开的一些实施例的用于通过文件系统执行日志的方法的流程图。

[0019] 图6B为根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址写缓冲器的方法的流程图。

[0020] 图6C为根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址读高速缓存的方法的流程图。

[0021] 图6D为根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址读高速缓存的替代方法的流程图。

[0022] 图7为示出根据本公开的一些实施例的用于访问对象存储设备的设备的硬件图。

具体实施方式

[0023] 现在将在下文中参照附图更全面地描述本公开,附图形成本公开的一部分,并且通过图示的方式示出了某些示例实施例。然而,主题可以以各种不同的形式体现,因此,所涵盖或要求保护的主题旨在被解释为不限于本文所述的任何示例实施例;提供示例实施例

仅用于说明。同样地,旨在提出要求保护或涵盖的主题的相当广泛的范围。其中,例如,主题可以体现为方法、设备、组件或系统。因此,实施例可以例如采用硬件、软件、固件或其任何组合的形式(除软件本身之外)。因此,以下详细描述不应被视为具有限制意义。

[0024] 在整个说明书和权利要求书中,术语可以具有超出明确说明的含义的上下文中建议或暗示的细微差别的含义。同样地,本文中使用的短语“在一个实施例中”不一定是指同一个实施例,并且本文中使用的短语“在另一个实施例中”不一定是指不同的实施例。例如,旨在要求保护的主体包括整个或部分示例实施例的组合。

[0025] 通常,术语可以至少部分地从上下文中的使用来理解。例如,如本文所使用的诸如“和”、“或”或“和/或”的术语可以包括可以至少部分地取决于使用这些术语的上下文的各种含义。通常,“或”如果用于关联列表(诸如A、B或C),则意指A、B和C,此处使用的是包含性含义,以及A、B或C,此处使用的是排他性含义。另外,如本文所使用的,术语“一个或更多个”,至少部分地取决于上下文,可以用于以单数含义描述任何特征、结构或特征,或者可以用于以复数含义描述特征、结构或特征的组合。类似地,诸如“一”、“一个”或“所述”之类的术语可以被理解为传达单数用法或传达复数用法,这至少部分地取决于上下文。另外,术语“基于”可以被理解为不一定旨在传达一组排他性的因素,而是可以替代地允许存在不一定明确描述的其他因素,这至少部分地取决于上下文。

[0026] 下面参照方法和设备的框图和操作说明来描述本公开。应当理解,框图或操作说明的每个框以及框图或操作说明中的框的组合可以借助于模拟或数字硬件和计算机程序指令来实现。这些计算机程序指令可以被提供给改变其在本文中详述的功能的通用计算机的处理器、专用计算机、ASIC或其他可编程数据处理装置,使得通过计算机的处理器或者其他可编程数据处理装置执行的指令实现框图或操作块或块中指定的功能/动作。在某些替代方案中,在块中指出的功能/动作可以在操作说明中指出的顺序之外发生。例如,连续示出的两个框实际上可以基本上同时执行,或者这些框有时可以以相反的顺序执行,这取决于所涉及的功能/动作。

[0027] 可以将这些计算机程序指令提供给以下处理器:通用计算机,所述通用计算机用以将其功能改变为特殊目的;专用计算机;ASIC;或其他可编程数字数据处理装置,使得通过计算机的处理器或其他可编程数据处理装置执行的指令实现在框图或操作块或块中指定的功能/动作,从而根据本文的实施例转换它们的功能。

[0028] 出于本公开的目的,计算机可读介质(或计算机可读存储介质/介质)存储计算机数据,所述数据可包括机器可读形式的可由计算机执行的计算机程序代码(或计算机可执行指令)。作为示例而非限制,计算机可读介质可包括用于有形或固定存储数据的计算机可读存储介质或者用于对包含代码的信号进行瞬时解释的通信介质。本文中使用的计算机可读存储介质是指物理或有形存储(与信号相对),并且包括但不限于以任何方法或技术实现的易失性和非易失性、可移动和不可移动的介质,用于有形存储诸如计算机可读指令之类的信息、数据结构、程序模块或其他数据。计算机可读存储介质包括但不限于RAM、ROM、EPROM、EEPROM、闪存或其他固态存储器技术、CD-ROM、DVD或其他光学存储器、磁带盒、磁带、磁盘存储器或其他磁存储设备,或者可用于有形存储所需信息或数据或指令并且可由计算机或处理器访问的任何其他物理或材料介质。

[0029] 图1是示出根据本公开的一些实施例的计算系统的框图。

[0030] 在所示实施例中,应用(101)读取、写入和修改存储在NAND设备(123)上的数据。对NAND设备(123)的访问由控制器(111)调解,并且NAND设备(123)和应用(101)之间的请求由安装在设备(100)上的操作系统的各个层(103、105、107、109)处理。

[0031] 应用(101)包括在设备(100)上运行的任何应用。例如,应用(101)可以包括用户级应用(例如,web浏览器、数据库、文字处理器等)。替代地,或者结合前述内容,应用(101)可以包括系统级应用(例如,实用程序应用、壳等)。通常,应用(101)包括存储在NAND设备(123)中的读取、写入或修改数据的任何软件应用。

[0032] 应用(101)可以通过发出由设备(100)的操作系统提供的系统调用来访问NAND设备(123)中的数据。例如,在Linux环境中,应用(101)可以发出CLOSE、CREAT、OPEN、MKDIR、LINK、CHOWN和各种其他系统调用以对文件系统数据进行操作。所使用的特定系统调用不做限制并且任何操作系统上的任何系统调用均可以由应用(101)调用。

[0033] 文件系统(103)接收由应用(101)发出的系统调用。文件系统(103)可以包括标准文件系统(例如,ext3)或能够处理给定文件的系统调用的其他文件系统(例如,基于FUSE的文件系统)。如在本领域中已知的,可以基于正被访问的底层文件或正被访问的分区/驱动器来动态加载文件系统。系统调用包括特定于系统调用类型的有效载荷(例如,要写入的文件名和数据)。文件系统(103)执行各种操作以处理系统调用。

[0034] 在处理系统调用之后,文件系统(103)将数据转发到块层(105)。块层(105)是对系统支持的各种基于块的设备的抽象。虽然块层(105)可以执行各种功能(例如,调度),但是块层(105)的一个主要功能是将来自系统调用的比特流转换成适合于底层存储设备的块级命令。

[0035] 块层(105)将块数据转发到NVMe驱动器(107)。可以使用其他驱动器代替NVMe,并且NVMe驱动器(107)被主要用作总线协议的示例。NVMe驱动器(107)或类似驱动器将块级命令转换为符合NVMe协议(或合适的替代协议)的命令。例如,NVMe驱动器(107)可以将各种块放入一个或更多个深度队列中。

[0036] 最后,NVMe格式的数据被发送到PCIe(Peripheral Component Interconnect Express,外围组件高速互连)驱动器(109),所述PCIe驱动器(109)根据PCIe协议执行数据的最终转换。在其他实施例中,可以使用PCI(Peripheral Component Interconnect Express,外围组件互连)、PCI-X或其他总线协议。在PCIe格式化之后,设备(100)通过总线(119)将数据发送到存储控制器(111)。在一个实施例中,总线(119)包括NVMe总线,而其他总线可以根据系统使用的协议进行选择。

[0037] 控制器(111)经由主机接口(113)接收NVMe块。在一个实施例中,主机接口(113)被配置为处理PCIe分组并从PCIe分组中提取底层NVMe块。固件(115)用于根据NVMe协议(或类似协议)处理未包装的分组。在一个实施例中,固件(115)从NVMe块中提取逻辑块地址(LBA)并将其转换为物理块地址(PBA)。通过与控制(111)本地存储的DRAM(121)促进了这种转换。DRAM(121)存储LBA到PBA的映射,并因此被用于将接收的LBA转换为NAND设备之一的PBA(123)。固件然后将基于LBA的命令转换为基于PBA的命令,并经由NAND接口(117)和通道(125)向NAND设备(123)发出命令。NAND接口(117)处理基于PBA的命令并生成NAND设备(123)所需的NAND信令以访问底层PBA。

[0038] NAND设备(123)包括一个或更多个NAND封装,每个封装包括多个块,每个块包括多

个页面。在一些实施例中,可以使用其他非易失性存储器(NVM)代替NAND存储器。

[0039] 当检索数据时,上述过程被反过来执行,并且为了清楚起见,不再包括具体细节。NAND设备(123)将PBA处的数据返回到NAND接口(117)。NAND接口(117)处理数据并将数据转发到固件(115)。作为响应,固件(115)使用DRAM(121)将数据的PBA转换为LBA,并将响应发送到主机接口(113)。在一个实施例中,主机接口(113)根据系统使用的协议(例如,PCIe和NVMe)重新格式化数据,并通过总线(119)将数据发送到设备(100)。设备(100)使用PCIe驱动器(109)处理PCIe分组,然后使用NVMe驱动器(107)处理所得到的分组。结果数据被返回到块层(105),所述块层(105)对返回的数据执行块操作,并将数据发送到文件系统(103),所述文件系统(103)响应于应用(101)发出的系统调用而返回适当的数据。

[0040] 可以看出,所有上述操作都是基于块的。块是系统使用的数据的固定大小。当在基于块的模式下操作时,由于使用了标准块大小,在确定块的大小时不再考虑底层数据的大小和类型。因此,即使要读取或要写入的数据小,系统也会通过系统的每一层分配并处理整个块。作为一个示例,元数据读取和写入通常包括对底层存储设备的的小的随机访问请求。块处理这些类型的请求所涉及的开销固有地减慢了系统的响应时间。此外,元数据访问通常会在执行的“关键路径”上。也就是说,对元数据的访问经常导致阻止对与元数据相关联的底层数据的操作。此外,设备(100)的各个层所采用的处理量为每个存储操作增加了大量的处理时间(例如,在10微秒之内)。由于块操作(层103和层105)以及NVMe(层107)处理的开销,导致所述处理时间。相比之下,独立的PCIe处理开销约为1微秒。因此,使用基于块的接口操作元数据会导致整个系统的显著减速。

[0041] 已经提出了各种技术来弥补这种缺陷。例如,最近的NVMe修订版已经增加了控制器内存缓冲区(CMB)功能,所述CMB功能将存储设备的内部缓冲区暴露给设备(100)地址空间。然后,设备(100)可以访问作为存储器映射的位置的CMB。然而,CMB被主要设计用于加速NVMe命令处理,并且有一些限制。首先,CMB的大小非常小,因为大多数DRAM(121)用于将LBA映射到PBA。其次,CMB是易失性存储器设备(例如,DRAM或SRAM),因此在断电或循环期间会丢失所有数据。这两个功能虽然对命令处理很有用,但是却使CMB不适合元数据访问或存储。

[0042] 图2是示出根据本公开的一些实施例的具有增强的元数据数据路径的计算系统的框图。

[0043] 在所示实施例中,各种组件(例如,100、101、103、105、107、109、111、113、115、117、119、121和123)可以执行与图1中描述的操作类似的操作,并且这些功能的描述通过引用结合在此。具体而言,图2所示的系统可以执行向设备(100)和NAND设备(123)(例如,经由NVMe总线(119))的基于块级别的传输数据或者从设备(100)和NAND设备(123)(例如,经由NVMe总线(119))的基于块级别的传输数据。然而,所示实施例另外提供了流线型元数据数据路径及其操作模式。

[0044] 在所示实施例中,控制器(111)包括另外的存储级存储器(SCM)设备(201)。SCM(201)可以包括相变存储器(PCM)。PCM是一种非易失性存储器类型,它使用不同的物理相位来表示数据。替代地或结合前述内容,SCM(201)可以包括电阻随机存取存储器(ReRAM)。ReRAM是使用不同电阻来表示的非易失性存储器类型。替代地或结合前述内容,SCM(201)可包括自旋转移扭矩RAM(STT-RAM)。STT-RAM是一种非易失性存储器类型,它使用不同的磁取

向来表示数据。可以在SCM(201)中使用其他类型的存储器。在一个实施例中,存储器的类型仅必须支持字节寻址、低延迟、大容量和非易失性。

[0045] 在所示实施例中,SCM(201)嵌入在控制器(111)内。在一个实施例中,SCM(201)可以安装到控制器的板上并在通信上联接到固件(115)。在所示实施例中,除了DRAM(121)之外,SCM(201)也被包括在控制器上。

[0046] SCM(201)经由控制器(111)的持久CMB(203)暴露给客户端系统。在一个实施例中,所使用的CMB的数量可以基于实现而变化。通常,CMB是PCIe基地址寄存器(BAR)或者用于存储与NVMe块命令相关联的数据的BAR内的区域。在一个实施例中,CMB被直接映射到设备(100)的存储器位置。因此,客户端(100)可以通过直接存储器访问来读取和写入CMB。

[0047] 在传统设备中,CMB被映射到DRAM的区域并且主要用于命令处理。然而,在所示实施例中,CMB被映射到SCM(201)而非映射到DRAM(121)。因此,控制器经由客户端设备上的指配的存储器位置将SCM(201)直接暴露给设备(100)。由于SCM(201)是持久的,因此通过经由CMB(203)将SCM(201)映射到设备(100)的存储器,设备(100)可以访问作为外部永久存储器的SCM(201)。另外,由于SCM(201)独立于DRAM(121),因此可以根据控制器的需要确定SCM(201)的大小,并且可以调整SCM(201)的大小以存储所确定的尽可能多的或尽可能少的数据。

[0048] 如图所示,进出SCM(201)的数据使用直接存储器访问(DMA)数据路径(205、207)。在所示实施例中,数据由应用(101)写入存储器映射文件(例如,使用Linux系统上的mmap)。文件系统(103)经由PCIe驱动器(205)将文件映射到SCM(201)。因此,当写入存储器映射文件时,元数据可以被直接传送到SCM(201)而不访问块层(105)和NVMe驱动器(107),从而显著减少延迟。在一个实施例中,SCM(201)经由可被应用(101)访问的预定义的存储器映射文件暴露给应用(101)。值得注意的是,如图所示,应用(101)仍然可以发出用于非元数据操作的基于块的命令。

[0049] 图3是示出根据本公开的一些实施例的具有增强的元数据数据路径的计算系统的框图。

[0050] 在所示实施例中,各种组件(例如,100、101、103、105、107、109、111、113、115、117、119、121和123)可以执行与图1中描述的操作类似的操作,并且这些功能的描述通过引用结合在此。

[0051] 在所示实施例中,控制器(111)包括一个或更多个PCI BAR。PCI BAR用于寻址实现控制器的PCI设备。必须通过映射到设备(100)的I/O端口地址空间或存储器映射的地址空间来启用给定的BAR。设备(100)上的主机软件对控制器(111)的BAR进行编程,以通过将配置命令写入控制器(111)来通知PCI设备其地址映射。每个PCI设备可以具有六个BAR,每个BAR包括32位地址。替代地,或者结合前述内容,BAR可包括三个64位地址。如果设备具有存储器映射空间(例如内部DRAM),则可以将其映射到主机地址空间,使得主机软件可以通过存储器加载/存储操作访问设备。

[0052] 在所示实施例中,DRAM(121)的一部分被指配给PCI BAR(301)。因此,当系统将PCI BAR窗口映射到其地址空间时,主机可以使用BAR(301)作为字节可寻址的读取/写入窗口。使用所述BAR(301),应用(101)可以通过两个接口访问数据。在第一接口中,应用(101)通过标准的基于块的NVMe接口(例如,数据路径119)访问NAND设备(123)。在第二接口中,应用

(101)通过字节可寻址PCI BAR(301) (例如,经由数据库303、305)使用“存储器模式”来直接访问DRAM(121)。通过这种混合设计,应用(101)可以根据其I/O模式选择最佳接口。例如,可以通过BAR窗口(301)来提供高优先级或延迟敏感的请求,而可以通过基于块的NVMe总线(119)来提供低优先级或面向吞吐量的请求。

[0053] 如上所述,控制器(111)利用DRAM(121)来存储LBA到PBA映射。因此,DRAM(121)的一部分可以被隔离以执行LBA/PBA匹配。DRAM(121)的剩余部分可以用作高速缓存或缓冲器。固件(115)可以通过PCI BAR(301)强制执行高速缓存或缓冲区的大小调整。也就是说,PCI BAR(301)可以指向DRAM(121)中的第一空闲存储器位置。在一些实施例中,可以将多个PCI BAR映射到DRAM(121)。在该实施例中,第一PCI BAR可以被映射到DRAM(121)中的空闲空间的第一部分,而第二PCI BAR可以被映射到空闲空间内中间的DRAM(121)地址。

[0054] 在一个实施例中,DRAM(121)的BAR映射部分可以用作字节可寻址读高速缓存。该高速缓存可以提供接近亚微秒的延迟并且经由数据路径(303、305)显著地改善读取请求的服务质量(QoS)。例如,从主动服务于编程或擦除命令的NAND通道读取数据会导致长延迟。诸如“程序暂停”之类的一些机制可以缓解这种延迟,但是这些机制给固件(115)带来了额外的复杂性并且依赖于NAND支持。相反,通过使用BAR(301),固件可以在发出编程/擦除命令之前从通道预取热数据,使得冲突的读取请求可以通过BAR(301)提供服务。在另一个实施例中,BAR(301)允许从“开放”块(即,尚未完全编程的块)读取数据。由于NAND特性,这些读取请求将遭受由较高误码率引起的较长延迟。通过从BAR(301)中的开放块缓存数据,读取请求将不受这种NAND限制的影响。

[0055] 在第二实施例中,DRAM(121)的BAR映射部分可以用作字节可寻址写缓冲器。在该实施例中,BAR(301)用于服务延迟敏感的请求,诸如预写日志记录和日志记录。在这些情况下,设备(100)首先用一定的阈值配置写缓冲器。然后它将数据附加到写缓冲区(如果数据不连续,则与LBA一起)。一旦写缓冲器中的数据达到阈值,固件(115)就自动将数据“刷新”回NAND设备(123)。根据DRAM(121)和断电电容器的大小,可以在BAR(301)中配置多个写缓冲器,允许多个写流。

[0056] 图4是示出根据本公开的一些实施例的使用存储器映射地址来访问存储级存储器设备中的数据的方法的流程图。

[0057] 在步骤402中,所述方法将非易失性存储设备映射到主机存储器位置。

[0058] 在一个实施例中,非易失性存储设备包括位于非易失性存储设备的控制器中的DRAM的一部分。在另一个实施例中,非易失性存储设备包括安装在存储设备的控制器上的SCM模块。在所示实施例中,映射非易失性存储设备包括使用存储设备的PCI BAR窗口来映射设备。在一个实施例中,PCI BAR窗口包括存储设备的CMB窗口。

[0059] 在步骤404中,所述方法接收数据访问命令。

[0060] 在一个实施例中,数据访问命令是指读取、写入或以其他方式操作存储设备上的数据的任何命令。数据访问命令的示例包括读取、写入、删除、更改属性和其他类型的命令。在一个实施例中,数据访问命令包括由主机应用发出的系统调用。在每个实施例中,数据访问命令可以包括文件路径或文件名。在一些实施例中,写入命令或更新命令包括要写入的数据。

[0061] 在步骤406中,所述方法确定数据访问命令是关键的还是非关键的。

[0062] 所述方法可以以各种方式确定命令是否关键。在一个实施例中，操作系统为关键数据访问提供单独的系统调用。在一个实施例中，这些单独的系统调用可以包括用限定的前缀（例如，“关键”打开的“C_OPEN”）作为前缀的标准POSIX系统调用。在另一个实施例中，POSIX系统调用可以被修改为包括指示调用是关键的关键标志。

[0063] 在其他实施例中，不需要对操作系统进行修改。相反，一个或更多个文件路径可以被存储器映射到存储设备的关键位置。例如，如上所述，固态设备(SSD)的CMB机制使得能够将存储器区域映射到主机设备存储器的存储器映射存储区域。当初始化存储设备时，设备的驱动器（例如，PCIe驱动器）可以将逻辑存储设备映射到CMB位置。在该实施例中，在步骤406中，所述方法检查数据访问命令的文件路径，并且如果所述命令正在访问在初始化期间映射的关键存储位置，则自动将所述命令归类为关键。

[0064] 在其他实施例中，数据访问命令的有效载荷可以另外用于将命令归类为关键或非关键。例如，所述方法可以分析写命令的有效载荷并确定有效载荷是否低于预定阈值。在此示例中，阈值可以适当地小，以确保只有小数据访问有效载荷触发关键的确定。因此，通过使用阈值，所述方法可以将所有较小的I/O访问路由到关键路径。

[0065] 在步骤408中，如果数据访问命令是非关键的，则所述方法发出基于块的命令。

[0066] 如上所述，所述方法可以继续将系统调用转换为基于块的访问命令。在一些实施例中，此过程需要在将命令发送到存储设备的控制器之前，通过块层、NVMe驱动器和PCIe驱动器传递系统调用。

[0067] 在步骤410中，所述方法在将数据访问命令归类为关键访问时识别映射的地址位置。

[0068] 在一个实施例中，所述方法识别映射到存储设备的CMB或BAR位置的主机设备存储器位置。在启动时，主机设备读取存储设备的BAR以识别应在主机设备和存储设备之间需要映射多少字节以及需要什么类型的映射（例如，内存访问）。主机设备分配此存储器并写入BAR，写入的数据包括PCI设备将响应的主机存储器中的地址。

[0069] 在一个实施例中，所述方法基于数据访问命令中包括的文件路径以及数据访问命令中包括的任何偏移来计算映射地址。因此，如果命令请求以6字节的偏移量向映射文件写入8个字节，则所述方法将从映射的初始地址的起始位置计算6个字节的存储器地址，并将此地址用作存储器写入的映射地址位置。

[0070] 在步骤412中，所述方法在映射的位置处访问本地存储器。

[0071] 在一个实施例中，访问本地存储器包括发出存储器读取、写入和其他命令。由于本地存储器地址被映射到存储设备内的内部位置，因此对这些本地存储器位置的访问被转发到相应的CMB或BAR地址，并且最终到底层存储设备（SCM或DRAM）。

[0072] 图5是示出根据本公开的一些实施例的用于处理存储设备操作的方法的流程图。

[0073] 在步骤502中，所述方法接收存储器映射的I/O分组。

[0074] 在一个实施例中，所述分组包括PCIe分组。在上图中，PCIe驱动器可以生成分组。例如，在步骤410中访问存储器位置可以包括将值写入特定存储器位置。物理（或虚拟）系统存储器位置和数据二者与由PCIe驱动器补充的附加元数据一起被封装在分组中。在一个实施例中，所述方法可以另外将分组标记为关键或非关键（例如，使用PCI TLP分组格式的标记字段）。

[0075] 在步骤504中,所述方法从数据包中提取主机地址(即,本地存储器地址),并且在步骤506中,将主机地址转换为本地存储设备地址。

[0076] 如上所述,修改后的存储设备支持基于块的I/O访问以及存储器I/O二者。在一个实施例中,步骤506中的转换处理包括将主机地址转换为SCM存储位置中的存储器位置。在另一个实施例中,转换包括将主机地址转换为存储设备DRAM中的存储器位置。可以基于存储设备的BAR配置来配置确定要应用哪个转换。

[0077] 在步骤506中,所述方法在设备的存储器设备上执行操作。

[0078] 在一个实施例中,所述方法对嵌入在设备内的SCM执行读取、写入和其他操作。替代地,所述操作可以包括要在本地DRAM上执行的读取、写入和其他操作。

[0079] 图6A是根据本公开的一些实施例的用于通过文件系统执行日志的方法的流程图。

[0080] 在步骤602中,所述方法接收文件访问命令。在一个实施例中,此命令可以包括由应用发出的任何系统调用。通常,系统调用表示应用要读取、写入或以其他方式访问的数据。在所示实施例中,数据被存储在诸如SSD的存储设备上。

[0081] 在步骤604中,所述方法将日志条目写入主机存储器中的位置。

[0082] 在一个实施例中,可以使用存储器存储和加载命令来执行步骤604。如上所述,底层SSD上的存储位置被映射(经由PCI BAR)到主机存储器,因此在步骤604,所述方法读取/写入这些存储器位置并将日志条目传送到底层存储设备(例如,SCM或SSD控制器的专用DRAM部分)。日志条目的具体形式并不限于此。实际上,日志条目的具体形式可取决于存储设备所使用的底层文件系统。

[0083] 在步骤606中,所述方法使用块I/O来发出文件命令。

[0084] 只有在接收到步骤604中的日志记录成功的消息之后,才可以执行块I/O命令的发出。由于日志条目是使用存储器加载/存储命令而编写的,因此与使用块I/O进行所有日志条目写入的现有系统相比,此消息的返回速度要快得多。如上所述,块I/O命令由主机设备的各个层(例如,文件系统驱动器、NVMe驱动器、PCIe驱动器等)处理,并随后在底层存储设备上执行。

[0085] 图6B是根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址写缓冲器的方法的流程图。

[0086] 在步骤608中,主机软件调整写入窗口的大小。

[0087] 在一个实施例中,主机软件基于如结合图3所描述的支持写入窗口的DRAM的大小来调整写入窗口的大小。替代地,或者结合前述内容,所述方法可以另外基于提供给SSD的DRAM和/或控制器的断电电容来调整写入窗口的大小。在一个实施例中,断电电容表示在断电时控制器/DRAM所存储的电荷量。在一个实施例中,该值被存储在存储设备的PCI配置空间中。断电电容用于计算写入窗口的最大大小,所述最大大小包括在对断电电容器所存储的电荷进行操作的同时可以从DRAM转移到持久存储器的最大量的数据。在一些实施例中,可以调整单个写入窗口的大小。替代地,所述方法可以调整多个写入窗口的大小。在该实施例中,每个写入窗口可以被指配给主机设备上的存储器位置(经由PCI BAR)。因此,所述方法可以支持多个写入通道。

[0088] 在步骤610中,所述方法识别关键数据。

[0089] 在一个实施例中,由主机应用软件执行关键数据的识别。识别关键数据的细节在

图4和5中进行描述,并且这些图的描述通过引入被包括在本文中。作为一个示例,关键数据可以包括应用必须尽可能快地保持的高价值数据。作为另一个例子,关键数据可以包括日志数据。

[0090] 在步骤612中,所述方法将关键数据写入存储器位置。

[0091] 在一个实施例中,将关键数据写入存储器位置包括主机应用发出“存储器存储”系统调用并将关键数据包括作为存储器存储的一部分。如上所述,主机存储器位置可以经由PCI BAR机制映射到存储设备的DRAM。因此,存储器存储导致主机设备经由PCIe驱动器(或类似的驱动器)将关键数据传送到SSD,因此绕过块级处理。

[0092] 在步骤614中,所述方法将关键数据附加到写入窗口。

[0093] 在一个实施例中,写入窗口是指SSD设备在控制器上使用的DRAM的连续部分。在所示实施例中,写入窗口充当写附加日志。因此,所有数据都被写入最后的可用DRAM存储器位置。因此,在所述实施例中,所述方法不需要如图5中所执行的转换存储器位置。

[0094] 在步骤616中,如果达到窗口的容量,则所述方法刷新写入窗口。

[0095] 如上所述,写入窗口可以具有基于例如DRAM的大小和/或设备的断电电容的固定容量。在一个实施例中,在每次附加到写入窗口之后,SSD设备的固件检查在DRAM中分配的写入窗口以确定写入窗口是否已满。替代地,固件可以比较当前附加地址与写入窗口的大小,以确定写入窗口是否已满。

[0096] 如果写入窗口已满,则固件启动刷新过程以清空写入窗口。在一个实施例中,刷新过程将DRAM的内容传送到持久存储位置(例如,在NAND闪存阵列上)。在一些实施例中,持久存储包括单独的元数据存储区域。在所述实施例中,固件将整个写入窗口附加到单独的元数据存储位置。

[0097] 图6C是根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址读高速缓存的方法的流程图。

[0098] 在步骤618中,所述方法接收编程或擦除(P/E)命令。

[0099] P/E命令指示SSD分别从底层NAND闪存阵列的页面或块写入和/或擦除数据。在一个实施例中,P/E命令对应于对闪存阵列的页面的更新。

[0100] 在步骤620中,所述方法预取包含受P/E命令影响的页面的块。在一个实施例中,所述方法发出读取命令以检索给定闪存块中的所有页面。

[0101] 在步骤624中,所述方法将页面高速缓存在块中。在一个实施例中,所述方法通过将每个页面写入专用于读高速缓存操作的DRAM的一部分来高速缓存每个页面。在一个实施例中,读高速缓存的大小仅受可用DRAM空间的限制。

[0102] 如图所示,步骤624和步骤626-630可以并行发生。

[0103] 在高速缓存受P/E命令影响的块之后,所述方法继续照常执行P/E命令。

[0104] 然而,在执行所述命令的同时,主机设备可以对存储在正在进行编程/擦除的块中的页面发出读取。在现有系统中,此读取需要缓冲,直到编程/擦除循环完成为止。在一些系统中,“程序暂停”功能可用于暂停编程操作。然而,此特征在SSD的固件中引入了显著的复杂性。相反,所述方法在步骤626中接收冲突的读取,并在步骤628中从高速缓存中检索相应的页面。最后,所述方法将高速缓存的数据630返回给请求的应用。

[0105] 如上所述,DRAM中的高速缓存包括来自在步骤624中进行编程和/或擦除的块的所

有页面。因此,当所述方法接收到冲突的读取命令时,所述方法从DRAM中的读高速缓存中检索所请求的页面。在一个实施例中,使用单独的存储器地址到LBA映射来将接收到的LBA从P/E命令映射到DRAM存储器地址。

[0106] 图6D是根据本公开的一些实施例的用于在存储设备的本地存储器中实现字节可寻址读高速缓存的替代方法的流程图。

[0107] 在步骤632中,所述方法接收程序命令。在一个实施例中,程序命令类似于结合图6C描述的程序命令,其公开内容通过引用整体并入本文中。

[0108] 在步骤634中,所述方法识别与页面相关联的块。在一个实施例中,使用包括在DRAM中的LBA到PBA映射来识别块标识符,如上所述。

[0109] 步骤636和步骤626可以并行执行。

[0110] 在步骤636中,在高速缓存受程序命令影响的块(“打开”块)之后,所述方法继续执行对底层块的编程操作。在此处理期间,在传统SSD中,块将被标记为“打开”,并且将阻止所有读取、写入或其他操作访问块。

[0111] 相反,如图6D中所示,所述方法允许在步骤626中接收冲突的读取请求。在一个实施例中,冲突的读取请求是指对在步骤636中被打开并被编程的块上的页面的读取。因此,在步骤638中,所述方法接收读取命令。在一些实施例中,所述方法可以确认要读取的页面被包括在打开的块中。然后,所述方法在步骤640中从高速缓存中检索打开的块。在一些实施例中,所述方法可以利用页面偏移来识别与页面对应的特定存储器位置。最后,在步骤642中,所述方法响应于读取而返回高速缓存的数据。

[0112] 图7是示出根据本公开的一些实施例的用于访问对象存储设备的设备的硬件图。

[0113] 客户端设备可以包括比图7中所示的组件更多或更少的组件。然而,所示的组件足以公开用于实现本公开的说明性实施例。

[0114] 如图7所示,客户端设备包括经由总线(714)与大容量存储器(704)通信的处理单元(CPU)(702)。客户端设备还包括一个或更多个网络接口(716)、音频接口(718)、显示器(720)、小键盘(722)、照明器(724)、输入/输出接口(726)和相机或其他光学、热或电磁传感器(728)。如本领域技术人员所理解的,客户端设备可以包括一个相机/传感器(728)或多个相机/传感器(728)。

[0115] 客户端设备可以可选地与基站(未示出)通信,或者直接与另一计算设备通信。网络接口(716)包括用于将客户端设备联接到一个或更多个网络的电路,并且被构造用于与一种或更多种通信协议和技术一起使用。网络接口(716)有时被称为收发器、收发设备或网络接口卡(NIC)。

[0116] 音频接口(718)被布置为产生和接收音频信号,诸如人声的声音。例如,音频接口(718)可以联接到扬声器和麦克风(未示出)以实现与其他人的通信并且为某些动作生成音频确认。显示器(720)可以是液晶显示器(LCD)、气体等离子体、发光二极管(LED)或与计算设备一起使用的任何其他类型的显示器。显示器(720)还可以包括触敏屏幕,所述触敏屏幕被布置成接收来自诸如手写笔的对象或来自人手的手指的输入。

[0117] 键盘(722)可以包括被布置为从用户接收输入的任何输入设备。例如,键盘(722)可以包括按钮数字拨盘或键盘。键盘(722)还可以包括与选择并发送图像相关联的命令按钮。照明器(724)可以提供状态指示并且提供光。照明器(724)可以在特定时间段内或响应

于事件而保持激活。例如,当照明器(724)激活时,它可以对键盘(722)上的按钮进行背光,并在客户端设备通电的同时保持开启。另外,当执行特定动作(诸如拨打另一个客户端设备)时,照明器(724)可以以各种图案对这些按钮进行背光。照明器(724)还可以使位于客户端设备的透明或半透明壳体内部的光源响应于动作而发光。

[0118] 客户端设备还包括用于与外部设备(例如UPS或交换机设备,或者图7中未示出的其他输入端或设备)通信的输入/输出接口(726)。输入/输出接口(726)可以利用一种或多种通信技术,诸如USB、红外、蓝牙等。

[0119] 大容量存储器(704)包括RAM(706)、ROM(710)和其他存储装置。大容量存储器(704)示出了用于存储诸如计算机可读指令、数据结构、程序模块或其他数据之类的信息的计算机存储介质的另一示例。大容量存储器(704)存储用于控制客户端设备的低级操作的基本输入/输出系统(“BIOS”)(712)。大容量存储器还可以存储用于控制客户端设备的操作的操作系统。应当理解,此组件可以包括诸如UNIX或LINUX的版本之类的通用操作系统或者诸如Windows Client™或Symbian®操作系统之类的专用客户端通信操作系统。操作系统可以包括Java虚拟机模块或与Java虚拟机模块接口,所述Java虚拟机模块能够通过Java应用程序控制硬件组件和操作系统操作。

[0120] 存储器(704)还包括PCIe设备驱动器(708)。如上所述,PCIe设备驱动器(708)向存储设备(未示出)发出读取、写入和其他命令。在一个实施例中,PCIe设备驱动器(708)向SSD(730)发出这些命令。在一个实施例中,SSD(730)可以包括固态驱动器或者实现一个或更多个NAND闪存芯片(诸如通过引用结合于此的在图2和图3中描述的芯片)的类似设备。通常,SSD(730)包括实现诸如图2至图5和图6A至图6D中所示的直接存储器访问之类的直接存储器访问的任何基于NAND闪存的设备。

[0121] 出于本公开的目的,模块可以是软件、硬件或固件(或其组合)系统、过程或功能、或其组件,以用于执行或促进本文所描述的过程、特征和/或函数(有或没有人为的互动或增强)。模块可以包括子模块。模块的软件组件可以被存储在计算机可读介质上,用于由处理器执行。模块可以是一个或更多个服务器的组成部分,或者由一个或更多个服务器加载和执行。可以将一个或更多个模块分组为引擎或应用。

[0122] 本领域技术人员将认识到,本公开的方法和系统可以以许多方式实现,并且因此不受前述示例性实施例和示例的限制。换句话说,由单个或多个组件以各种方式组合的硬件和软件或固件以及各个功能执行的功能元件可以在客户端级或服务器级或两者之间分布在软件应用中。在这方面,本文描述的不同实施例的任何数量的特征可以被组合成单个或多个实施例,并且具有少于或多于本文中描述的所有特征的替代实施例是可能的。

[0123] 功能也可以全部或部分地以现在已知的或将变为已知的方式分布在多个组件之间。因此,大量的软件/硬件/固件组合可以实现本文中描述的功能、特征、接口和偏好。此外,本公开的范围涵盖执行描述的特征和功能接口的常规已知方式以及可以对本文描述的硬件或软件或固件组件进行的那些改变和修改,正如本领域技术人员现在和将来可以理解的。

[0124] 此外,通过示例的方式提供了作为本公开中的流程图呈现和描述的方法的实施例,以便提供对所述技术的更完整的理解。所公开的方法不限于本文中给出的操作和逻辑流。考虑了这样的实施例,其中改变了各种操作的顺序并且其中被描述为作为较大操作的

一部分的子操作是独立执行的。

[0125] 尽管出于本公开的目的描述了各种实施例,但是不应认为这些实施例将本公开的教导限制于那些实施例。可以对上述元件和操作进行各种改变和修改,以获得保持在本公开中描述的系统 and 过程的范围内的结果。

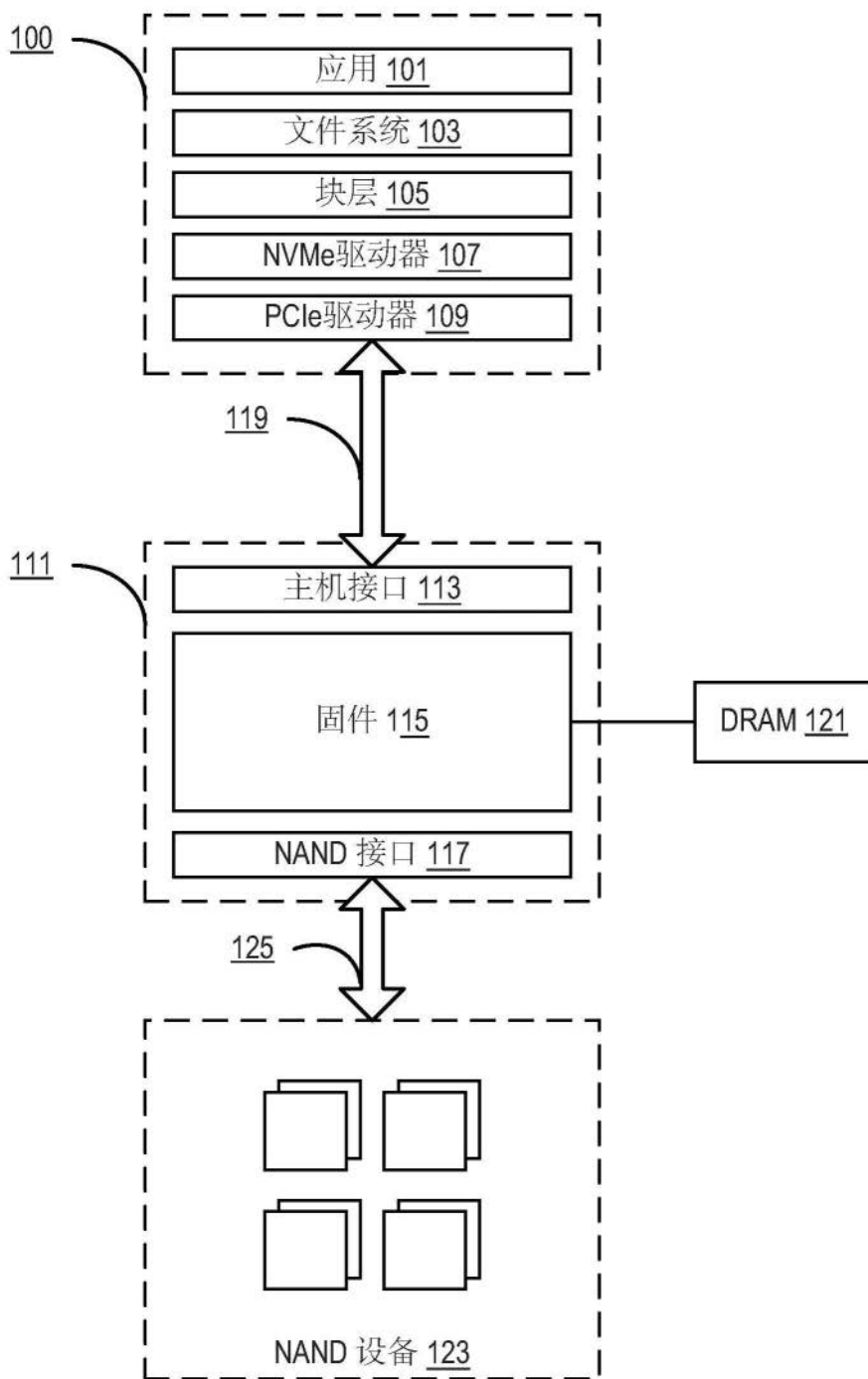


图1

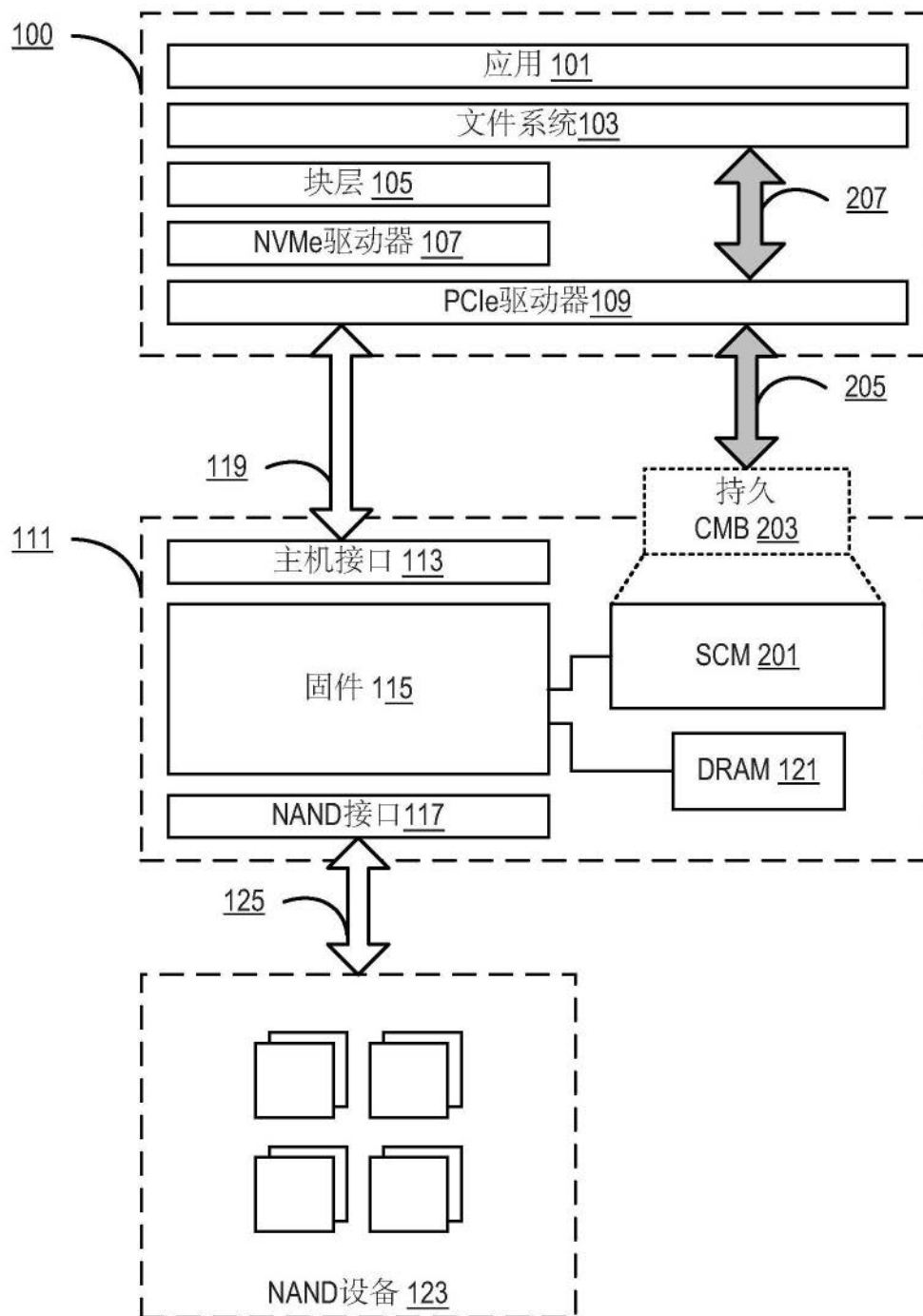


图2

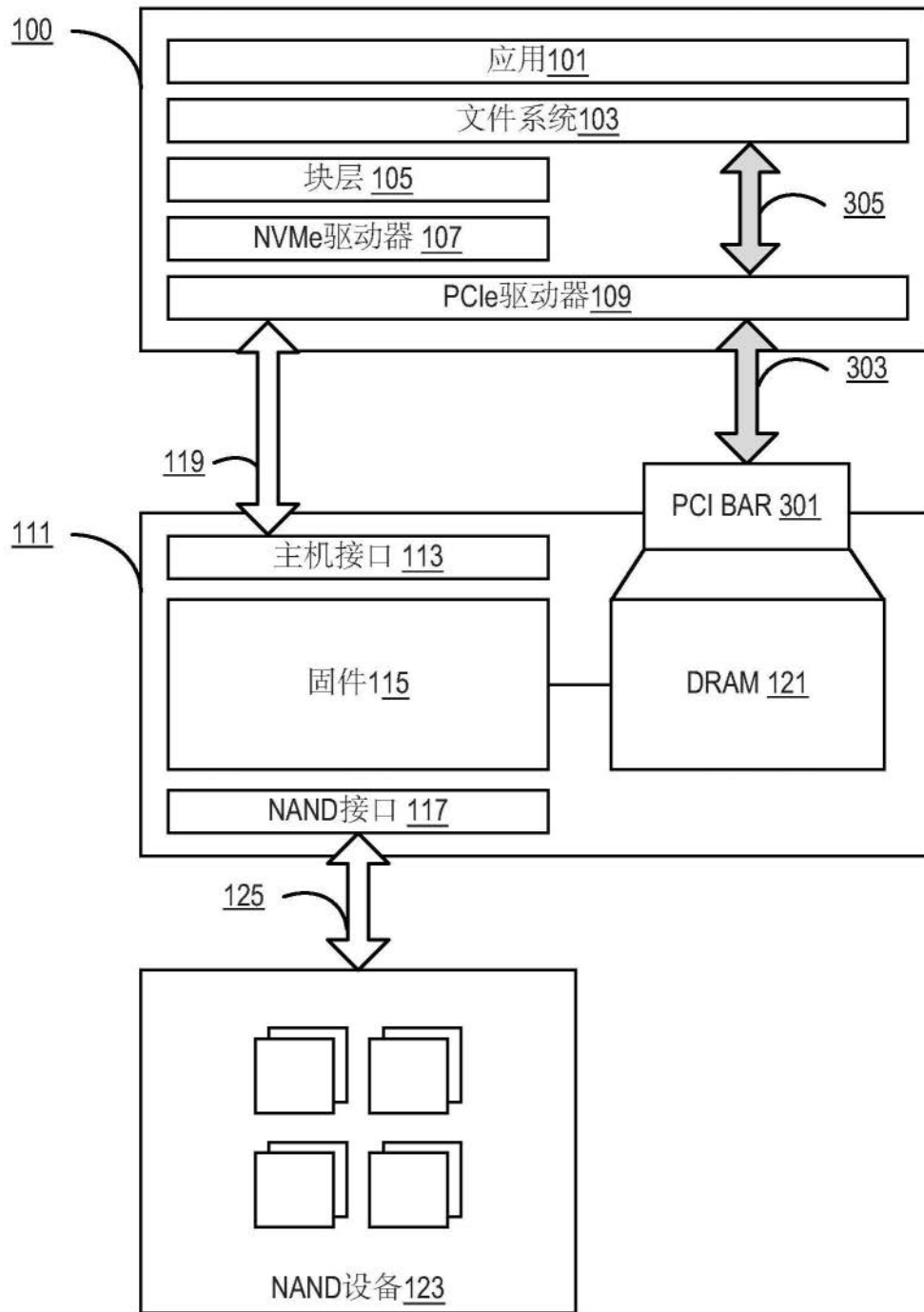


图3

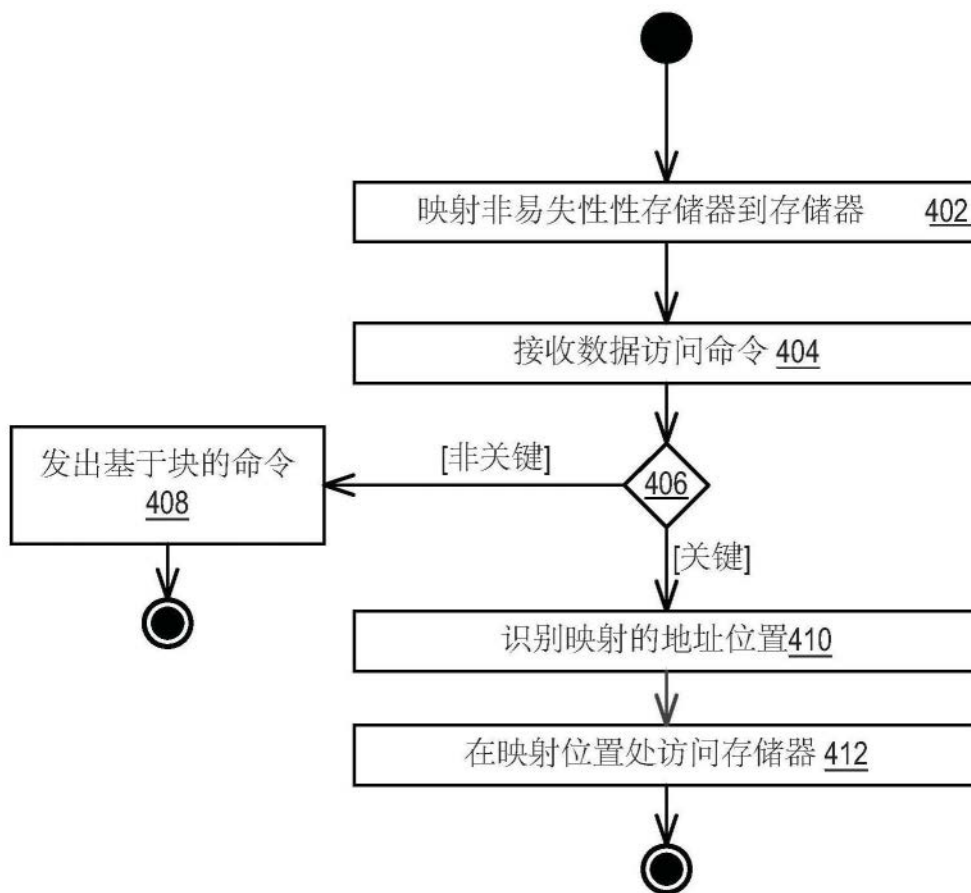


图4

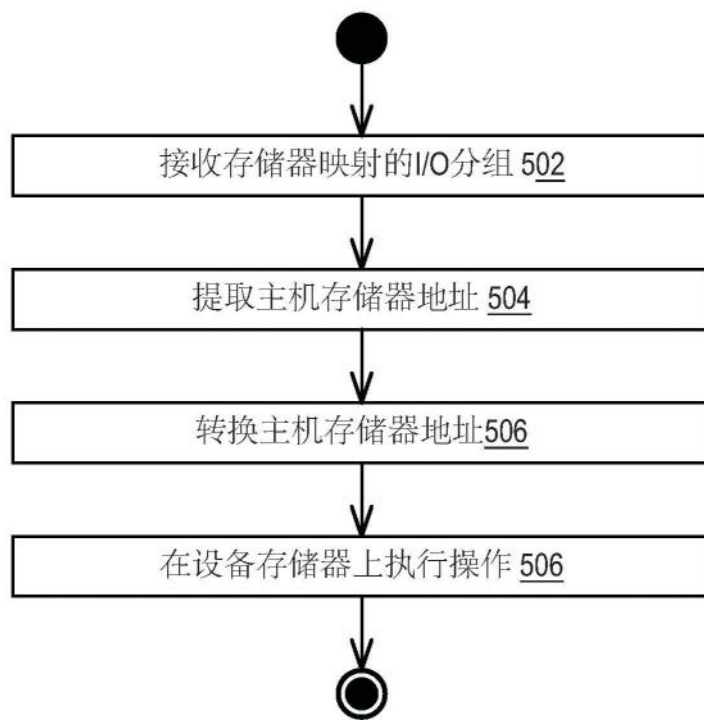


图5

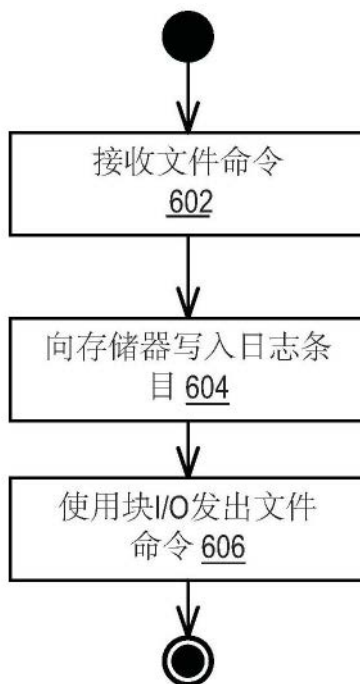


图6A

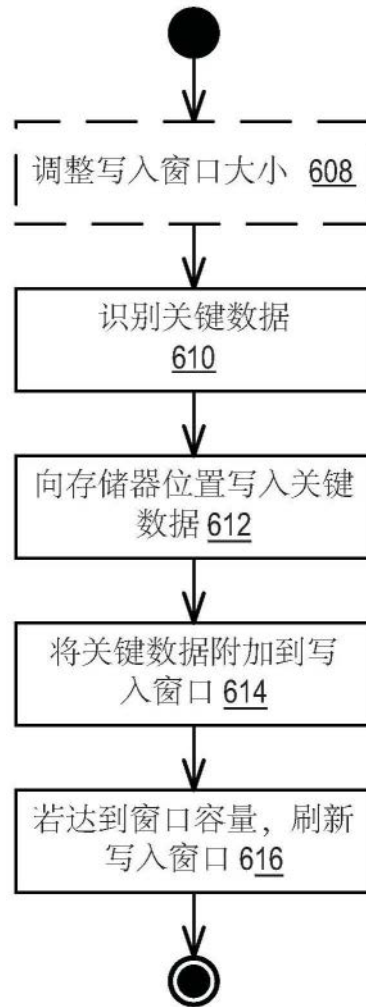


图6B

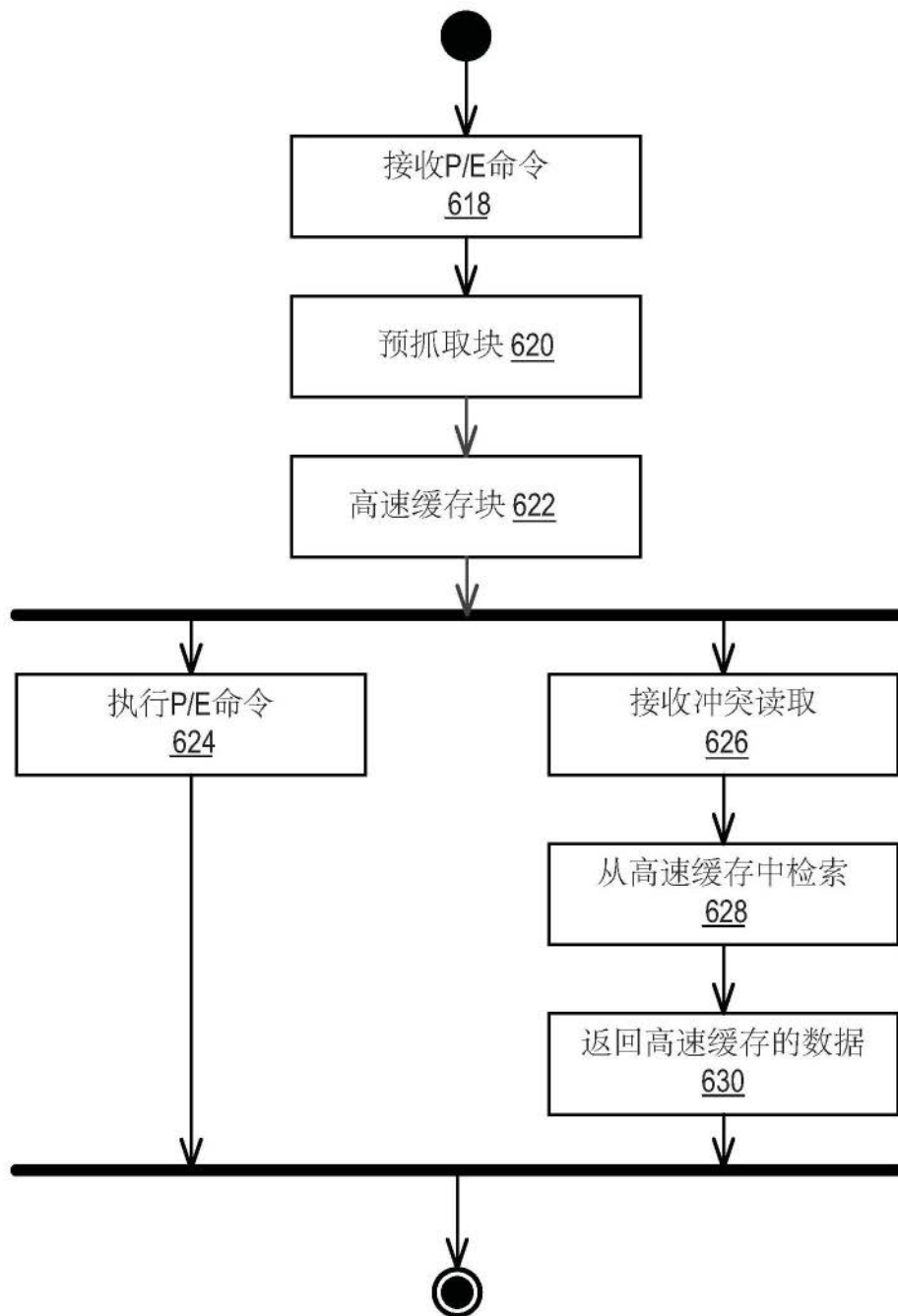


图6C

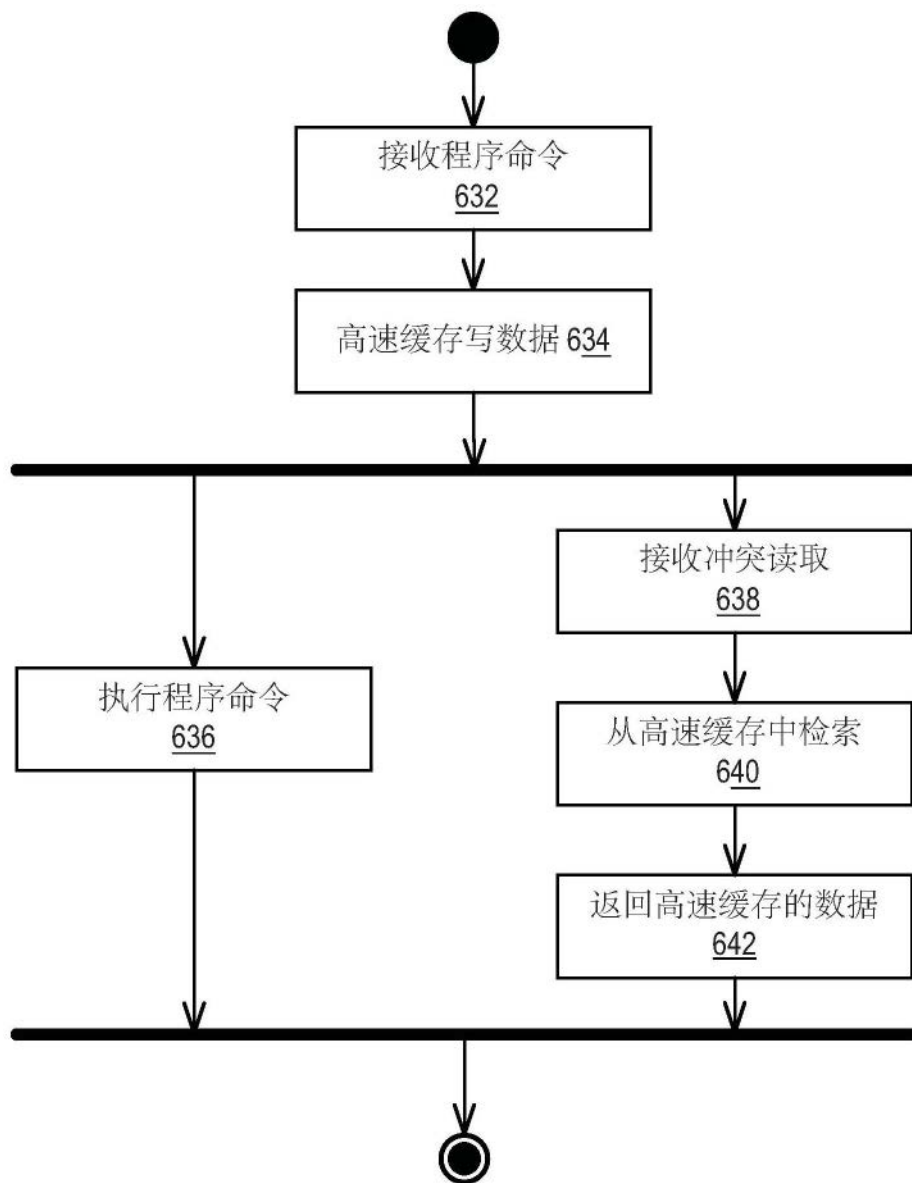


图6D

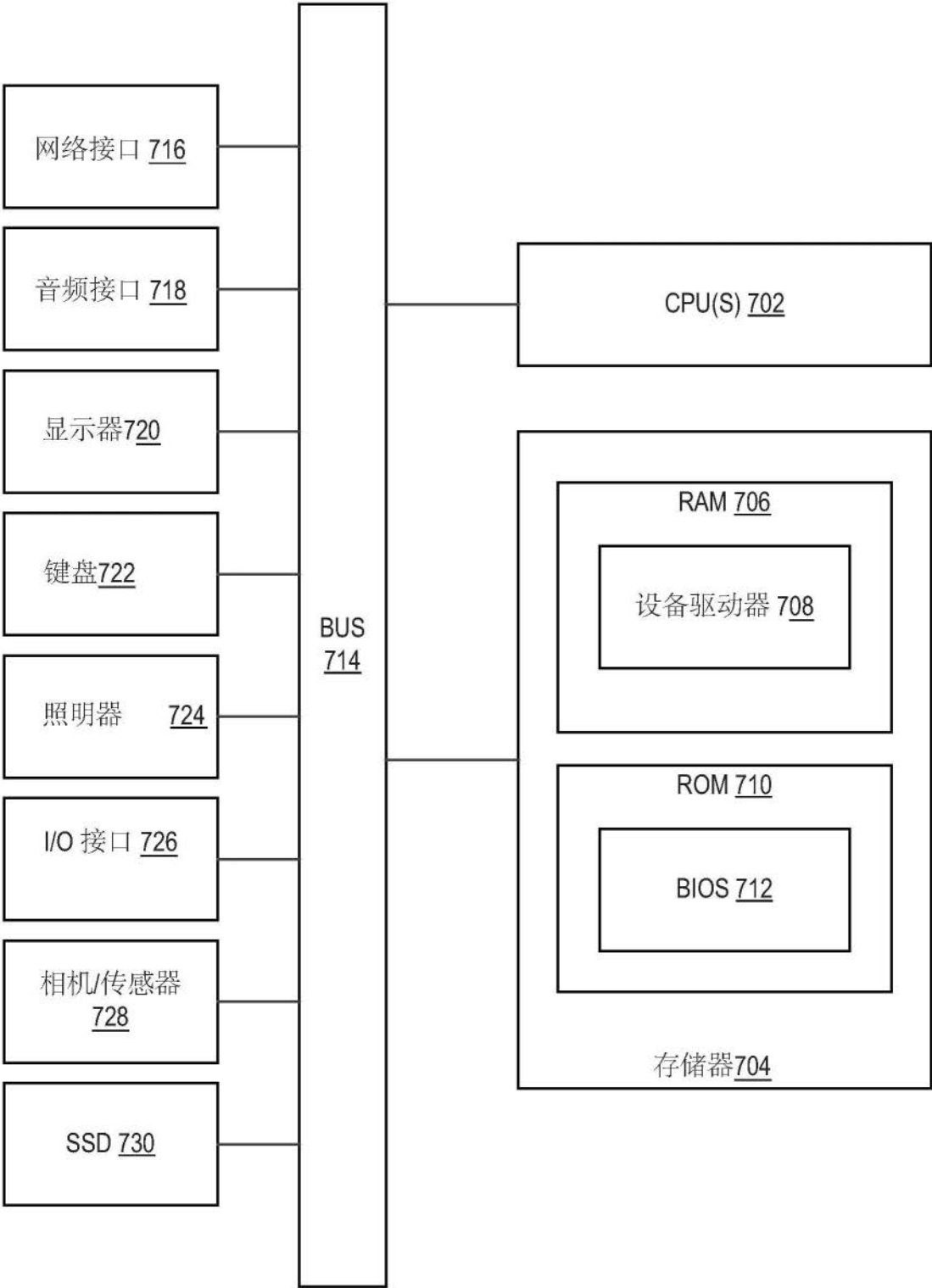


图7