(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107291424 A
(43)申请公布日 2017.10.24

(21)申请号 201610577356.0

(22)申请日 2016.07.20

(30)优先权数据
10-2016-0041120 2016.04.04 KR

(71)申请人 忆锐公司
地址 韩国首尔
申请人 延世大学校产学协力团

(72)发明人 郑溟随 张杰

(74)专利代理机构 北京安信方达知识产权代理有限公司 11262
代理人 陆建萍 郑霞

(51)Int.Cl.
G06F 9/38(2006.01)
G06F 12/084(2016.01)
G06F 12/0842(2016.01)
G06F 12/0862(2016.01)
G06F 12/0893(2016.01)
G06F 12/0882(2016.01)

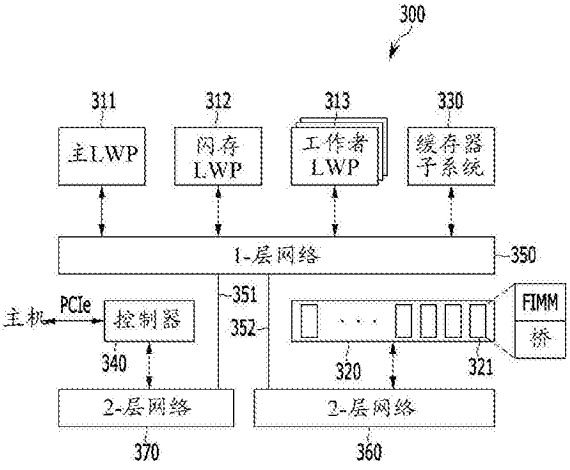
权利要求书2页 说明书11页 附图7页

(54)发明名称

基于闪存的加速器和包含其的计算设备

(57)摘要

本发明涉及基于闪存的加速器和包含其的计算设备。在基于闪存的加速器中，基于闪存的非易失性存储器按页存储数据，而缓冲器子系统按字或字节存储数据。加速器控制器管理基于闪存的非易失性存储器和缓存器子系统之间的数据移动。多个处理器处理缓存器子系统中存储的数据。网络集成基于闪存的非易失性存储器、缓存器子系统、加速器控制器和多个处理器。



CN 107291424 A

1.一种基于闪存的加速器，其通过补充主机的中央处理单元(CPU)的功能或独立于所述CPU来进行数据处理，所述基于闪存的加速器包括：

基于闪存的非易失性存储器，其按页存储数据；

缓存器子系统，其按字或字节存储数据；

加速器控制器，其管理所述基于闪存的非易失性存储器和所述缓存器子系统之间的数据移动；

多个处理器，其处理所述缓存器子系统中存储的数据；以及

网络，其集成所述基于闪存的非易失性存储器、所述缓存器子系统、所述加速器控制器和所述多个处理器。

2.根据权利要求1所述的基于闪存的加速器，其中，所述加速器控制器将连接所述基于闪存的加速器和所述主机的接口的基址寄存器映射到所述缓存器子系统或所述多个处理器中的处理器，并基于所述基址寄存器从所述主机接收请求。

3.根据权利要求2所述的基于闪存的加速器，其中，所述加速器控制器将所述缓存器子系统映射到所述基址寄存器的第一值，并基于所述第一值将数据从所述主机移到所述缓存器子系统。

4.根据权利要求3所述的基于闪存的加速器，其中，所述加速器控制器将所述多个处理器中的处理器映射到所述基址寄存器的第二值，并基于所述第二值通知来自所述主机的所述数据的类型。

5.根据权利要求1所述的基于闪存的加速器，其中，所述缓存器子系统包括：

第一存储器，其包含被映射到所述基于闪存的非易失性存储器的第一数据空间；以及

第二存储器，其存储指示在所述基于闪存的非易失性存储器和所述第一数据空间的页之间的映射的页表。

6.根据权利要求5所述的基于闪存的加速器，其中，所述第一存储器还包含用于从所述主机下载数据/向所述主机上传数据的第二数据空间。

7.根据权利要求6所述的基于闪存的加速器，其中，从所述主机下载的所述数据包含待由所述多个处理器中的处理器执行的应用。

8.根据权利要求5所述的基于闪存的加速器，其中，所述页表的页表条目包含被映射至所述基于闪存的非易失性存储器的物理闪存页号的所述第一数据空间的页号。

9.根据权利要求8所述的基于闪存的加速器，其中，所述页表条目还包含拥有所述页表条目的所有者的处理器标识符。

10.根据权利要求9所述的基于闪存的加速器，其中，当请求对所述基于闪存的非易失性存储器的存储器访问的请求者的处理器标识符不同于所述所有者的处理器标识符时，所述缓存器子系统拒绝所述请求者的访问请求。

11.根据权利要求8所述的基于闪存的加速器，其中，所述页表条目还包含指示所请求的数据是存在于所述第一数据空间还是存在于所述基于闪存的非易失性存储器中的当前位标记。

12.根据权利要求8所述的基于闪存的加速器，其中，所述第二存储器还存储映射表，所述映射表将从所述主机的虚拟地址获取的所述基于闪存的非易失性存储器的逻辑页号映射到所述基于闪存的非易失性存储器的物理闪存页号。

13.根据权利要求8所述的基于闪存的加速器，其中，所述第二存储器还存储包含应用的段的段头，以及

其中，所述段头包含指示由所述段使用的地址空间的范围的段信息。

14.根据权利要求13所述的基于闪存的加速器，其中，当关于所述主机的访问请求的地址在所述段头的所述地址空间的范围之内时，所述缓存器子系统拒绝所述主机的所述访问请求。

15.一种计算设备，包括：

根据权利要求1所述的基于闪存的加速器；

所述主机；以及

接口，其连接所述基于闪存的加速器和所述主机。

16.一种基于闪存的加速器，其通过补充主机的中央处理单元(CPU)的功能或独立于所述CPU来执行数据处理，所述基于闪存的加速器包括：

基于闪存的非易失性存储器；

缓存器子系统，包含

第一存储器，其包含被映射到所述基于闪存的非易失性存储器的第一数据空间，以及

控制器，其管理在所述基于闪存的非易失性存储器和所述第一数据空间之间的映射；

多个处理器，其处理所述缓存器子系统中存储的数据；以及

网络，其集成所述基于闪存的非易失性存储器、所述缓存器子系统和所述多个处理器。

17.根据权利要求16所述的基于闪存的加速器，其中，所述第一存储器还包含用于存储来自所述主机的应用的第二数据空间。

18.根据权利要求17所述的基于闪存的加速器，其中，所述应用由所述多个处理器中的处理器执行。

19.根据权利要求16所述的基于闪存的加速器，其中，所述缓存器子系统还包含指示在所述基于闪存的非易失性存储器和所述第一数据空间的页之间的映射的页表。

20.根据权利要求19所述的基于闪存的加速器，其中，所述页表的页表条目包含被映射至所述基于闪存的非易失性存储器的物理闪存页号的所述第一数据空间的页号。

基于闪存的加速器和包含其的计算设备

[0001] 相关申请的交叉引用

[0002] 本申请要求在2016年4月4日在韩国知识产权局提交的韩国专利申请第10-2016-0041120的优先权和权益，其全部内容通过引用并入本文。

[0003] 背景

[0004] (a)领域

[0005] 所描述的技术涉及到基于闪存的加速器和包含其的计算设备。

[0006] (b)相关技术的描述

[0007] 具有高计算并行性和相对低功率消耗的、诸如图形处理单元(GPU)和集成众核(MIC)设备的基于多核的加速器变得越来越受欢迎。在这种加速器中，多个处理核心共享执行控制并可经由线程级并行性和数据级并行性对大量的数据进行相同的操作。使用加速器和中央处理单元(CPU)的系统，与只有CPU的系统相比，可呈现明显的速度提升。

[0008] 加速器可处理比它们先前处理的更多的数据，且这种数据容量是可预期的。然而，加速器采用的板载存储器，其与主存储器相比大小要相对小。加速器因此使用非易失性存储器，例如固态盘(SSD)，其被连接到主机以便处理大型数据集。

[0009] 然而，加速器和非易失性存储器相互完全不连接，且由不同的软件堆栈管理。因此，许多冗余存储器分配/释放用户空间和核心空间和数据副本存在于用户空间和核心空间之间，以便从非易失性存储器读取数据或向非易失性存储器写入数据。另外，由于核心模块不能直接访问用户空间存储器，因此在核心空间和用户空间之间的存储器管理和数据副本开销是不可避免的。而且，核心模式和用户模式转换的开销以及数据副本还促成数据移动的长延迟。这些开销使得与加速器性能相比，速度提升的改进变得不明显。

[0010] 存在以主动存储的形式将特定的应用集成到SSD中以解决这些问题的许多现有研究。然而，所有这些研究均聚焦于诸如SSD的存储设备并利用现有的SSD控制器或采用定制的现场可编程门阵列(FPGA)来处理SSD中的数据。因此，只有在制造SSD时已经被集成的特定的应用才能够被执行，而通用计算应用则不能在数据存在的SSD附近被执行。

[0011] 概述

[0012] 本发明的实施方式提供了基于闪存的加速器和用于执行各种应用的计算设备。

[0013] 根据本发明的实施方式，提供了通过补充主机CPU的功能或独立于CPU来执行数据处理的基于闪存的加速器。基于闪存的加速器包含基于闪存的非易失性存储器、缓存器子系统、加速器控制器、多个处理器和网络。基于闪存的非易失性存储器按页存储数据，而缓冲器子系统按字或字节存储数据。加速器控制器管理在基于闪存的非易失性存储器和缓存器子系统之间的数据移动，而多个处理器处理在缓存器子系统中存储的数据。网络集成基于闪存的非易失性存储器、缓存器子系统、加速器控制器和多个处理器。

[0014] 加速器控制器可将连接基于闪存的加速器和主机的接口的基址寄存器映射到缓存器子系统或多个处理器中的处理器，并基于基址寄存器从主机接收请求。

[0015] 加速器控制器可将缓存器子系统映射到基址寄存器的第一值，并基于该第一值将数据从主机移到缓存器子系统。

[0016] 加速器控制器可将多个处理器中的处理器映射到基址寄存器的第二值，并基于该第二值通知来自主机的数据的类型。

[0017] 缓存器子系统可包含第一存储器和第二存储器，第一存储器包含映射到基于闪存的非易失性存储器的第一数据空间，第二存储器存储指示在基于闪存的非易失性存储器和第一数据空间的页之间的映射的页表。

[0018] 第一存储器还包含用于从主机下载数据/将数据上传到主机的第二数据空间。

[0019] 从主机下载的数据可包含待由多个处理器中的处理器执行的应用。

[0020] 页表的页表条目可包含被映射至基于闪存的非易失性存储器的物理闪存页号的第一数据空间的页号。

[0021] 页表条目还可包含拥有页表条目的所有者的处理器标识符。

[0022] 当请求者的处理器标识符不同于所有者的处理器标识符时，缓存器子系统可拒绝请求对基于闪存的非易失性存储器的存储器访问的请求者的访问请求。

[0023] 页表条目还可包含指示所请求的数据是存在于第一数据空间还是存在于基于闪存的非易失性存储器中的当前位标记。

[0024] 第二存储器还可存储映射表，映射表将从主机的虚拟地址获取的基于闪存的非易失性存储器的逻辑页号映射到基于闪存的非易失性存储器的物理闪存页号。

[0025] 第二存储器还可存储包含应用的段的段头，且段头可包含指示由段使用的地址空间范围的段信息。

[0026] 当用于主机的访问请求的地址在段头的地址空间范围之内时，缓存器子系统可拒绝主机的访问请求。

[0027] 根据本发明的另一个实施方式，提供了包含上文所述的基于闪存的加速器、主机以及连接基于闪存的加速器和主机的接口的计算设备。

[0028] 根据本发明的又一个实施方式，提供了通过补充主机CPU的功能或独立于CPU来执行数据处理的基于闪存的加速器。基于闪存的加速器包含基于闪存的非易失性存储器、缓存器子系统、多个处理器和网络。缓存器子系统包含第一存储器和控制器，第一存储器包含映射到基于闪存的非易失性存储器的第一数据空间，控制器管理在基于闪存的非易失性存储器和第一数据空间之间的映射。多个处理器处理在缓存器子系统上存储的数据，且网络集成基于闪存的非易失性存储器、缓存器子系统和多个处理器。

[0029] 第一存储器还可包含用于存储来自主机的应用的第二数据空间。

[0030] 应用可被多个处理器中的处理器执行。

[0031] 缓存器子系统还可包含指示在基于闪存的非易失性存储器和第一数据空间的页之间的映射的页表。

[0032] 页表的页表条目可包含第一数据空间的页号，该第一数据空间的页号被映射至基于闪存的非易失性存储器的物理闪存页号。

[0033] 附图简述

[0034] 图1是根据本发明的实施方式的计算设备的原理框图。

[0035] 图2是根据本发明的实施方式的基于闪存的加速器的原理框图。

[0036] 图3是图2中所示的基于闪存的加速器中的缓存器子系统的原理框图。

[0037] 图4是图2中所示的基于闪存的加速器中的端桥的原理框图。

[0038] 图5示出根据本发明的实施方式的加速器的通信接口。

[0039] 图6示出图5中所示的通信接口中的NDP核心执行的示例。

[0040] 图7示出根据本发明的实施方式的加速器中的地址虚拟化的缓存器子系统。

[0041] 图8是用于解释根据本发明的实施方式的加速器中的地址虚拟化的图。

[0042] 图9和图10是用于解释根据本发明的实施方式的加速器中的数据一致性的图。

[0043] 图11示出两个NDP核心的示例。

[0044] 图12示出静态核间调度的示例。

[0045] 图13示出动态核间调度的示例。

[0046] 图14示出两个NDP核心的另一个示例。

[0047] 图15示出按次序的核内调度的示例。

[0048] 图16示出不按次序的核内调度的示例。

具体实施方式

[0049] 在以下详细描述中，仅仅以说明的方式，已经示出并描述了本发明的仅特定实施方式。如本领域的技术人员将认识到的是，所描述的实施方案可以以各种不同的方式被修改，而全都不偏离本发明的精神或范围。相应地，附图和描述被认为是本质上阐示性的而不是限制性的。在整个说明书中，相似的参考标记表示相似的元素。

[0050] 图1是根据本发明的实施方式的计算设备的原理框图。图1示出了计算设备的一个示例，且根据本发明的实施方式的计算设备可通过使用不同的结构来实现。

[0051] 参考图1，根据本发明的实施方式的计算设备包含CPU 100、CPU侧存储器200和基于闪存的加速器300。加速器300是不同于通用CPU的补充数据处理设备，且可以用于通过补充CPU的功能来执行数据处理或独立于CPU来执行数据处理的计算机硬件。图形处理单元(GPU)或集成众核(MIC)设备是加速器300的一个示例。

[0052] 计算设备还包含用于将存储器200和加速器300与CPU 100连接的北桥400。加速器300可被连接至位于CPU侧的北桥140。例如，加速器300可经由PCIe(外围部件互联高速peripheral component interconnect express)链路被连接至北桥140。北桥140还可被称为存储器控制器中心(MCH)。

[0053] 虽然传统的加速器只包含用于并行的多个处理器，根据本发明的实施方式的加速器300是基于闪存的加速器，其将对应于加速器核心的多个处理器310物理上与基于闪存的非易失性存储器320集成在一起。

[0054] 在一些实施方式中，加速器300的每个处理器可以是轻量级处理器(LWP)。在一个实施方式中，LWP可以是在高速网络上连接的低功率处理器。在这种情况下，LWP可通过高速网络与诸如加速器控制器和基于闪存的非易失性存储器的其他内部资源进行通信。此后，为了方便起见，加速器300的每个处理器被描述为LWP。

[0055] 在一些实施方式中，包含CPU 100和存储器200的系统可被称为主机。

[0056] 计算设备可以将各种应用卸载到加速器300，这允许加速器300直接执行应用。例如，这些应用可以是将计算从主机卸载到加速器300的近数据处理(NDP)应用。下文，为了方便应用，应用被描述为NDP应用，且NDP应用可被称为NDP核心。相应地，主机可访问加速器300以便卸载NDP核心或处理数据的读取/写入。在这种情况下，加速器300的LWP可通过执行

NDP核心来直接访问非易失性存储器。因此，可移除由传统加速器从非易失性存储器读取数据或向非易失性存储器写入数据所需的许多冗余存储器分配/释放和数据副本。

[0057] 接着，参考图2至图4描述了根据本发明的实施方式的基于闪存的加速器。

[0058] 图2是根据本发明的实施方式的基于闪存的加速器的原理框图，图3是图2中所示的基于闪存的加速器中的缓存器子系统的原理框图，而图4是图2中所示的基于闪存的加速器中的端桥的原理框图。

[0059] 参考图2，基于闪存的加速器300包含多个LWP 310、基于闪存的非易失性存储器320、缓存器子系统(BS)330和加速器控制器340。

[0060] 在一些实施方式中，LWP 310可根据超长指令字(VLIW)的架构来构建。LWP 310可以是全部连接的并共享单个存储器，即缓存器子系统330，相似于传统的对称多处理器架构。在一个实施方式中，基于每个LWP310执行的任务，LWP 310可被分为主LWP 311、闪存LWP 312和工作者LWP 313。主LWP 311可执行诸如NDP核心卸载和执行调度的管理工作，闪存LWP 312可执行闪存I/O(输入/输出)管理，而工作者LWP 313可执行近闪存的实际的数据处理。

[0061] 基于闪存的非易失性存储器320可包含多个闪存包321。基于闪存的非易失性存储器320通过将闪存包321集成到网络来构建内部存储池。下文，该基于闪存的非易失性存储器320被称为闪存主干。

[0062] 缓存器子系统330可操作为在用于读取并写入页中的数据的闪存主干320和用于按字或字节读取并写入数据的主机或LWP 310之间的缓冲存储器。例如，页可以是4KB至16KB。

[0063] 在一些实施方式中，缓存器子系统330可包含第一存储器331、第二存储器332和存储器控制器333，如在图3中所示。

[0064] 在一个实施方式中，第一存储器331可以是可按字或字节寻址的存储器。例如，诸如动态随机存取存储器(DRAM)的低功率存储器可被用作第一存储器331。第一存储器可被用于闪存管理和预取/缓存数据。第二存储器332可以是用于快速处理的存储器。例如，高速便签式存储器(SPM)可被用作第二存储器332。第二存储器可如同L2缓存一样快地服务对处理器网络的管理I/O请求。下文，为了方便起见，第一存储器331和第二存储器332被描述为低功率存储器和便签式存储器(SPM)。存储器控制器333可被提供作为低功率存储器331和SPM 332的管理。

[0065] 加速器控制器340管理在加速器300中的LWP 310和闪存主干320之间的数据移动或者在主机和加速器300的闪存主干320之间的数据移动，并管理页访问和字或字节访问之间的转换。在从主机或LWP 310接收到数据读取请求之后，加速器控制器340从缓存器子系统330读取对应的数据，并如果数据已经被存储到缓存器子系统330中，则将它们传输给主机或LWP 310。如果对应的数据未被存储在缓存器子系统330中，加速器控制器340将闪存主干320中的数据转换成按字或字节的数据，并将它们存储在缓存器子系统330中，且从缓存器子系统330读取数据并将它们传输至主机或LWP 310。在从主机或LWP 310接收到数据写入请求之后，加速器控制器340将对应的数据写入缓存器子系统330，并将写入缓存器子系统330的数据转换成按页的数据，并将它们传输给闪存主干320。

[0066] 由此，根据本发明的实施方式，由于加速器300将可按页寻址的闪存主干320的数据映射到缓冲器子系统330，可按字或字节寻址的主机或LWP 310可从闪存主干320读取数

据或将数据写入闪存主干320，而没有额外的操作。

[0067] 在一些实施方式中，当加速器300经由PCIe接口连接至北桥400时，加速器控制器340可以是PCIe控制器。

[0068] 再次参考图2，在一些实施方式中，LWP 310、闪存主干320、缓存器子系统330和加速器控制器340可通过网络350、360和370相互连接。LWP 310和缓存器子系统330可被连接至网络350，而网络350可经由网络开关351和352被连接至与闪存主干320相连的网络360和与加速器控制器340相连的网络370。

[0069] 在一个实施方式中，网络350、360和370可通过使用部分交叉开关(partial crossbar switch)将大型网络分为两组交叉开关(crossbar configuration)配置而形成。两组可包含多个简化的交叉开关(2-层)和流式交叉开关(streaming crossbar)(1-层)。集成LWP 310的网络350可以是1-层网络，而1-层网络可朝着高速网络进行设计。网络360和370可以是2-层网络。2-层网络的吞吐量可以足以接受闪存主干320和PCIe通信。

[0070] 在一些实施方式中，可通过使用在发明者的论文“Triple-A:A Non-SSD Based Autonomic All-Flash Array for High Performance Storage Systems”中定义的闪存直插式存储器模块(flash inline memory module)(FIMM)来形成闪存主干320的闪存包321。

[0071] 该论文通过引用并入本文。

[0072] 在FIMM中，多个闪存包通过单个数据通道进行集成。在一些实施方式中，单个数据通道可共享可容纳闪存地址和交易命令的16I/O引脚。因为实际上每个闪存包都具有其自己的I/O控制逻辑和一组数据寄存器，所以可经由就绪/繁忙(R/B)和芯片使能(CE)引脚来从外部处理闪存的所有的低等级业务。因此，FIMM不仅定义容易替换的架构，还提供了向主机暴露所有闪存内部的巨大潜力。

[0073] 尽管FIMM的标准定义了其机械接口和信号组，但是在上述论文中定义的FIMM的时钟频率和I/O引脚的数量可以不同于根据本发明的实施方式将FIMM应用到闪存包的情况下的FIMM的时钟频率和I/O引脚的数量。为了填补这一空缺，用于每个FIMM的端点桥可被添加至一些实施方式中，如图2和图4中所示。

[0074] 端点桥可将由闪存主干320接收的I/O请求转换到用于FIMM的时钟域中。如在图4中所示出的，端点桥包含闪存接口层(FIL)、发送/接收接口层、入站标记队列和出站标记队列。闪存接口层负责与闪存包(即FIMM)的接口，而发送/接收接口层负责通过网络370的发送/接收。入站和出站标记队列被用于缓存I/O请求。在时钟域转换期间，入站和出站标记队列可处理闪存业务，并接收或传输来自网络370的对应的数据。

[0075] 接着，参考图5和图6描述了根据本发明的实施方式的加速器的通信接口。

[0076] 图5示出了根据本发明的实施方式的加速器的通信接口，而图6示出了图5中所示的通信接口中的NDP核心执行的示例。

[0077] 参考图5，加速器控制器340从主机接收请求，而加速器300的LWP 310处理该请求。加速器控制器340包含物理层(PHY)341和核心342。

[0078] 连接加速器300和主机的接口的基址寄存器(BAR)(例如，PCIe基址寄存器)可被映射至核心342。在一些实施方式中，缓存器子系统330可被映射到具有BAR1值的基址寄存器，而主LWP 311的进程间通信中断寄存器(IPC-IR)可被映射到具有BAR2值的基址寄存器。加速器300的缓存器子系统330(尤其是缓存器子系统330的低功率存储器331)和主LWP311的

IPC-IR可通过基址寄存器而被暴露给主机。低功率存储器331可处理加速器300的内容迁移，而主LWP 311的IPC-IR可处理加速器300的计算控制。

[0079] 在加速器300和主机之间的通信开始处，加速器控制器340的PHY 341将进入请求从主机转达到核心342。PHY 341可处理接口定时要求，例如，PCIe定时要求。核心342基于由主机指示的基址寄存器BAR1和BAR2来解析/安排来自主机的数据并将它们转发到缓存器子系统330或主LWP 311的IPC-IR。

[0080] 主机通过指示基址寄存器BAR1将数据迁移到缓存器子系统330。一旦主机完成将数据迁移到缓存器子系统330，主机可通过指示基址寄存器BAR2来通知主LWP 311迁移的完成。在一些实施方式中，主机可通过使用不同的事件标识符(ID)更新主LWP 311的IPC-IR来通知迁移的完成。事件ID可被用于通知事件的类型。

[0081] 如果主机卸载NDP内核，主LWP 311准备NDP内核执行并向工作者LWP 313指示NDP内核执行。在这个情况下，事件ID可指示所迁移的数据是NDP内核的可执行映像。

[0082] 如果主机请求I/O服务，那么主LWP 311经由用于I/O服务的闪存执行接口向闪存LWP 312发送信号，而闪存LWP 312执行闪存主干320上的数据读取/写入。在这个情况下，事件ID可指示所迁移的数据是I/O服务。如果存在在与其上根据I/O服务来执行数据读取/写入的闪存主干320的相同的地方上处理数据的工作者LWP 313，则I/O服务可引起一致性问题。为了解决这个问题，在一些实施方式中，存储器权限控制可由闪存LWP 312和主LWP 311一起实施。存储器权限控制将在下文中进行描述。

[0083] 在一些实施方式中，如在图5中所示的，可由NDP描述表(NDT)来将NDP内核呈现给主LWP 311。在一个实施方式中，NDP描述表可以具有与可执行的并可链接的格式(ELP)相似的形式。

[0084] NDP描述表可包含可执行文件，该可执行文件包含诸如NDP内核代码(例如，.text)的预定义的段，并还可包含段头。段头可包含诸如对应的段名称、开始地址和长度的段信息。在一些实施方式中，NDP描述表的段头可带来定义关于NDP内核使用的输入阵列变量的地址空间的输入信息和定义用于NDP内核使用的输出阵列变量的地址空间的输出信息，不同于ELF。

[0085] 如参考图5描述的，如果主机将NDP内核卸载到缓存器子系统330，NDP内核可通过NDP描述表呈现给主LWP 311。然后，如在图6中所示的，主LWP 311首先通过控制寄存器冻结目标工作者LWP 313(S610)。主LWP 311然后解析NDP描述表，从缓存器子系统330装载目标NDP内核(例如，.text)，并将装载的NDP内核分配给工作者LWP 313的L2缓存(S620)。另外，主LWP 311将段头的输入和输出信息移动到缓存器子系统330，即缓存器子系统330的SPM 332(S630)，并管理闪存主干320的合适的地址空间(S640)。而且，基于输入和输出信息，对应于地址空间的空间被分配给缓存器子系统330的低功率存储器331，并对应于地址空间的空间被映射到闪存主干320的地址空间(S640)。接着，主LWP 311经由工作者LWP 313的引导地址寄存器来更新NDP内核(例如，.text)的开始地址，再次重设控制寄存器，并通过触发目标LWP 313的IPC-IR来启动NDP内核(S650)。相应地，NDP内核可被卸载到加速器300并被工作者LWP 313执行。在NDP内核执行处，可通过使用被映射的闪存主干320的地址空间和缓存器子系统330的空间来实施数据读取/写入。

[0086] 在一些实施方式中，由于闪存LWP 312处理I/O请求，在SPM 332上实现消息缓存机

制的队列子系统(q-子系统)可被提供为闪存执行接口。q-子系统可提供诸如create()、open()、alloc_msg()、put()、delete_msg()和delete()的通用队列接口。create()创建队列,open()打开队列,alloc_msg()分配消息,put()发送消息,delete_msg()删除消息,以及delete()删除队列。在一个实施方式中,q-子系统可由具有用于主LWP 311和缓存器子系统330的仲裁器的双向I/O缓存器形成。使用这个通用队列应用编程接口(API),主LWP 311和缓存器子系统330可与闪存LWP 312进行通信,而无需其他接口协议实现。

[0087] 如上文所述的,根据本发明的实施方式,各种NDP内核可被卸载到加速器300并被执行。在NDP内核执行处,加速器300可从闪存主干320读取数据或将输入写入闪存主干320,而利用主机进行额外的数据复制/移动。

[0088] 接着,参考图7、图8、图9和图10描述了在根据本发明的实施方式的加速器300中,用于在闪存主干320和缓存器子系统330之间进行映射的地址虚拟化。

[0089] 图7示出用于根据本发明的实施方式的加速器中的地址虚拟化的缓存器子系统,且图8是用于解释根据本发明的实施方式的加速器中的地址虚拟化的图。图9和图10是用于解释根据本发明的实施方式的加速器中的数据一致性的图。

[0090] 对于闪存地址虚拟化,可引入制造与传统工作寄存器空间兼容的闪存地址空间的机器(例如,软件)。该机器可在存储器控制器333上实现。下文,被描述为存储器控制器333的操作的部分可由该机器执行。

[0091] 参考图7,缓存器子系统330的低功率存储器331可分区为NDP数据空间331a和闪存数据空间331b,且可将地址映射到闪存数据空间331b和闪存主干320之间。低功率存储器331的NDP数据空间331a可通过基址寄存器(例如,图5的BAR1)暴露给主机,且可向主机上传NDP内核的可执行文件或I/O请求/从主机下载NDP内核的可执行文件或I/O请求。为了映射两个不同的地址域,缓存器子系统330的SPM 332保存页表332a,页表332a可映射闪存数据空间331b和闪存主干320的页。

[0092] 由SPM 332管理的页表332a的页表条目(PTE)可包含被映射到闪存主干320的物理闪存页号(FPN)的闪存数据空间331b的页号(PNF)、当前位(P)、拥有页表条目的所有者的LWP ID(LID)和重写标志位(D),并且可通过进入存储器请求的虚拟地址来进行引用。虚拟地址可被用于获取存储器控制器333需要发送消息给闪存LWP 312的逻辑页号(LPN)。

[0093] SPM 332还包含段头332b和映射表332c。段头332b包含诸如对应的段名称、开始地址和长度的段信息。段对应于预定的可执行文件,例如,NDP内核代码(例如,.text)。映射表332c被提供在闪存转换层(FTL)上并将由主机使用的虚拟地址(即,逻辑地址)映射到由闪存暴露的物理地址。为此,映射表332c提供逻辑页号与物理闪存页号(FPN)之间的映射。

[0094] NDP数据空间331a可被主LWP 311使用,映射表332c可被闪存LWP 312使用,以及闪存数据空间331b和页表332a可被工作者LWP 313使用。

[0095] 参考图8,当从工作者LWP 313的L1缓存中漏掉由NDP内核执行请求的存储器访问时,存储器直接对存储器控制器333进行访问(S810)。存储器控制器333检查缓存器子系统330的SPM 332中的段头332b(S820)。如果用于存储器访问的存储器地址位于由段头332b定义的地址范围(例如,由段头332b的开始地址和长度定义的地址范围)之内时,存储器控制器333询问由存储器地址索引的页表条目332a(S830)。

[0096] 如果页表条目332a的LID不同于请求存储器访问的请求者的LID,则存储器控制器

333通过拒绝存储器访问请求来保护缓存器子系统330。由于与拥有SPM 332的页表条目332a的LWP无关的请求者的请求是未授权的访问请求，存储器控制器333可通过拒绝该存储器访问请求来保护缓存器子系统330。

[0097] 如果页表条目332a的LID等于请求者的LID(即，请求是授权的访问请求)，则存储器控制器333检查当前位(P)标记。如果当前位(P)标记是“0”，存储器控制器333将请求读取的消息传输到闪存LWP312(S840)。在一些实施方式中，可通过使用闪存页的大小划分来自主机的存储器访问请求的虚拟地址来获取存储器控制器333需要发送消息到闪存LWP 312的逻辑页号(LPN)。

[0098] 闪存LWP 312然后使用映射表332c将逻辑页号(LPN)转换成物理闪存页号(FPN)(S850)，并通过从闪存主干320的物理闪存页号(FPN)读取数据来将对应的数据带到闪存数据空间331b的页号(S860)。由于数据存在于闪存数据空间331b中，存储器控制器333更新当前位(P)标记(S870)并将数据供应给工作者LWP 313的L1缓存(S880)。

[0099] 如果当存储器控制器333查看当前位(P)标记时当前位(P)标记是“1”(S840)，则对应的数据存在于闪存数据空间331b中。因此，存储器控制器333可将闪存数据空间331b中的数据供应给工作者LWP 313的L1缓存而没有从闪存主干320带来数据。

[0100] 如果多个工作者LWP 313在如上所述的闪存的相同位置上访问，则可通过将页表条目332a的LID与请求者的LID进行比较来保持闪存数据空间331b的一致性。然而，由主机和工作者LWP 313并行请求的存储器访问可损害数据一致性。考虑图9和图10中所示的示例，基于在虚拟地址0x11处读取数据的主机请求，闪存LWP 312通过查找映射表332c将从虚拟地址0x11获取的逻辑页号(LPN)0x0转换成闪存页号(FPN)0x80，并提供来自闪存主干320的闪存页号(FPN)0x80的数据A[]。在这种情况下，工作者LWP 313可尝试将位于闪存数据空间331b的页号(PNF)0x22处的数据B[]写入到虚拟地址0x11。然后，可由工作者LWP 313改变数据A[]。即，主机和工作者LWP 313可同时访问数据A[]以损害数据一致性。为了防止该情况，在一些实施方式中，闪存LWP 312查看与主机请求有关的SPM 332中的段头332b(S1010)，如图10中所示。

[0101] 如果从主机请求的虚拟地址获取的目标逻辑页号(LPN)位于由特定NDP内核使用的地址空间范围之内(其由SPM 332的段头332b描述)，则加速器300拒绝主机请求的访问(S1020)并通知主机权限拒绝(S1030)。例如，如在图9中例示的，主机可请求读取在虚拟地址0x11处的数据。在这种情况下，由于虚拟地址0x11位于其中开始地址为0x10且长度是5的段的地址空间范围之内，参考段头332b，加速器300可拒绝主机请求的访问。因此，闪存主干320的数据一致性可被保持。

[0102] 由此，当多个LWP 310执行各种NDP内核时，闪存主干320可通过使用缓存器子系统330与传统的工作存储器空间兼容。而且，因为可由缓存器子系统330的存储器控制器333上实现的机器执行这些功能，可物理地合并多个LWP 310和闪存主干320，而不需要额外的操作系统。

[0103] 在一些实施方式中，不同于使用单指令多线程(SIMT)模型的传统加速器，根据本发明的实施方式的加速器的LWP可并行执行不同类型的NDP内核，其每个可包含各种操作功能。这使得用户能够卸载不同的应用并执行不同类型的NDP。接着，描述了用于实施不同类型的NDP的NDP内核执行调度。

[0104] 根据本发明的实施方式，可为NDP内核执行调度提供核间执行调度和核内执行调度。在核间执行调度中，每个工作者LWP可执行特定的NDP内核，该特定的NDP内核以单个指令程序实施从头到尾的数据处理。在核内执行调度中，可将NDP内核分成多个代码块，且可基于输入数据布局在多个工作者LWP之间同时执行多个代码块。

[0105] 首先，参考图11、图12和图13描述了根据本发明的实施方式的加速器中的核间执行调度。

[0106] 图11示出了两个NDP内核的示例，图12示出了静态核间调度的示例，而图13示出了动态核间调度的示例。

[0107] 可将核间执行调度分类为静态核间调度和动态核间调度。

[0108] 参考图11中所示的示例，提供了两个NDP内核NDP0和NDP2，两个NDP实例i0和i1与NDP内核NDP0相关联，而两个NDP实例i2和i3与NDP内核NDP2相关联。

[0109] 在根据一个实施方式的静态核间调度中，从主机接收的进入的NDP请求基于其执行类型被静态分配到特定NDP。例如，参考图12，在静态核间调度中，NDP内核NDP0的实例i0和i1可被静态分配到工作者LWP LWP0，而NDP内核NDP1的实例i2和i3可被静态分配到工作者LWP LWP2。

[0110] 静态核间调度容易实现和管理多样化的NDP内核。然而，在用于每个NDP内核的执行时间未能很好地平衡的情况下，静态核间调度可带来不良资源利用。

[0111] 在根据另一个实施方式的动态核间调度中，对于静态调度后面的不良资源利用问题，主LWP可基于其服务可用性向工作者LWP池中的任何工作者LWP动态分配NDP请求。例如，在调度开始处，主LWP可向所有的工作者LWP以轮循方式分配不同类型的NDP内核。在此之后，每当工作者LWP通过完成实例运行经过IPC-IR向主LWP发送消息时，主LWP可将下一个可用的NDP内核实例分配给紧接着的工作者LWP。例如，如在图12中所示的，主LWP可将两个NDP内核NDP0和NDP2的实例i0、i1、i2和i3按顺序分配给四个工作者LWP LWP0、LWP1、LWP2和LWP3。然后，由于可将实例i1和i3与实例i0和i2一起并行供应，资源利用可被提高并时间间隙可以被节省。

[0112] 下面参考图14、图15和图16描述了根据本发明的实施方式的加速器中的核内执行调度。

[0113] 图14示出了两个NDP内核的另一个示例，图15示出了按次序的核内调度的示例，而图16示出了不按次序的核内调度的示例。

[0114] 可将核内执行调度分类为按次序的核内调度和不按次序的核内调度。

[0115] 实际上NDP内核可由被称为微块的多组代码段组成。每组对其输入/输出数据具有执行依赖性。以特定顺序执行微块，但是具有被称为screen的操作，其可在微块中在I/O矩阵的不同部分上并行工作。参考图14中所示的示例，提供了两个NDP内核NDP0和NDP2，两个NDP实例i0和i1与NDP内核NDP0相关联，而两个NDP实例i2和i3与NDP内核NDP2相关联。每个实例包含两个微块m0和m1。在这种情况下，实例i0的微块m0包含两个screen S1和S2，实例i0的微块m1包含两个screen Sa和Sb。实例i1的微块m0包含一个screen S1，而实例i1的微块m1包含两个screen Sa和Sb。实例i2的微块m0包含两个screen S1和S2，而实例i2的微块m1包含一个screen Sa。实例i3的微块m0包含两个screen S1和S2，而实例i3的微块m1包含一个screen Sa。

[0116] 在根据一个实施方式的按次序的核内调度中，每个实例的每个微块可被按次序地执行，且在每个实例的每个微块中包含的所有screen可被不同的LWP同时执行。例如，如在图15中所示的，在时间T0处执行实例i0中的微块m0的screen S1和S2之后，可在时间T1处执行实例i0中的微块m1的screen Sa和Sb。在此之后，可在时间T2处执行实例i1中的微块m0的screen S1，然后在时间T3处执行实例i1中的微块m1的screen Sa。随后，在时间T4处执行实例i2中的微块m0的screen S1和S2之后，可在时间T5处执行实例i2中的微块m1的screen Sa。在此之后，可在时间T6处可执行实例i3中的微块m0的screen S1和S2，并然后在时间T7处可执行实例i3中的微块m1的screen Sa。可由不同的LWP LWP0和LWP1同时执行实例i0中的微块m0的两个screen S1和S2。相似地，可由不同的LWP同时执行同一个实例中的同一个微块的多个screen。

[0117] 由此，由于同时执行多个screen，与按顺序执行一个实例的screen的内核执行调度相比，可减少执行时间。

[0118] 在根据另一个实施方式的不按次序的核内调度中，与不同实例和不同微块有关的许多screen可以不按次序的方式来执行，而不像按次序的核内调度。如果在特定时间处存在任何的空闲LWP，不按次序的核内调度可从存在于不同实例边界或不同的NDP内核边界的不同微块借走几个screen。从而，可减少微块的执行时间，并可增强总的系统性能。

[0119] 因为，如图15所示的，两个LWP LWP2和LWP3在时间T0处空闲，不同微块的screen可在时间T0处被填充。例如，实例i0中的微块m0的screen S1和S2、实例i1中的微块m0的screen S1和实例i2中的微块m0的screen S1可在时间T0处同时被不同的LWP LWP0、LWP1、LWP2和LWP3执行。另外，实例i2中的微块m0的screen S2、实例i3中的微块m0的screen S1和S2和实例i0中的微块m1的screen Sa可在时间T1处同时被不同的LWP LWP0、LWP1、LWP2和LWP3执行。而且，实例i0中的微块m1的screen Sb、实例i1中的微块m1的screen Sa、实例i2中的微块m1的screen Sa和实例i3中的微块m1的screen Sa可在时间T2处同时被不同的LWP LWP0、LWP1、LWP2和LWP3执行。

[0120] 下面描述了在实际的硬件上实现根据本发明的实施方式的加速器300之后测量的结果。

[0121] 如在以下表1中所表示的，运行在1GHz时钟下的八个LWP(每个均具有其自己的64KB L1缓存和512KB L2缓存)被用于性能测量。在八个LWP中，六个LWP被用做工作者LWP，且两个LWP被用做主LWP和闪存LWP。在缓存器子系统中，基于4MB SRAM的SPM被用作为SPM，而2GB DDR3L DRAM用作低功率存储器。而且，32闪存晶片被用作基于闪存的非易失性存储器。

[0122] 表1

[0123]

部件	规格	工作频率	端口宽度	估计的带宽
LWP	8个处理器	1GHz	128b	16000MB/s
L1/L2缓存	64KB/512KB	500MHz	256b	16000MB/s
SPM	共享、4MB	500MHz	256b	16000MB/s
DRAM	DDR3L,2GB	800MHz	64b	6400MB/s
闪存	32个晶片	200MHz	64b	3200MB/s

PCIe	v2.0,2条线路	5GHz	2b	1024MB/s
1-层交叉开关	256条线路	500MHz	256b	16000MB/s
2-层交叉开关	128条线路	333MHz	128b	5328MB/s

[0124] 在这种情况下，与CPU驱动的数据处理和基于GPU的数据处理方式相比，根据本发明的实施方式的加速器可以提高性能7.8x和75％，同时降低能量损耗88％和57％。

[0125] 虽然已经结合目前被认为是实际的实施方式的实施方案描述了本发明，但应理解的是，本发明不限于公开的实施方式，而是相反地，旨在覆盖包括在所附权利要求的精神和范围内的各种修改和等同布置。

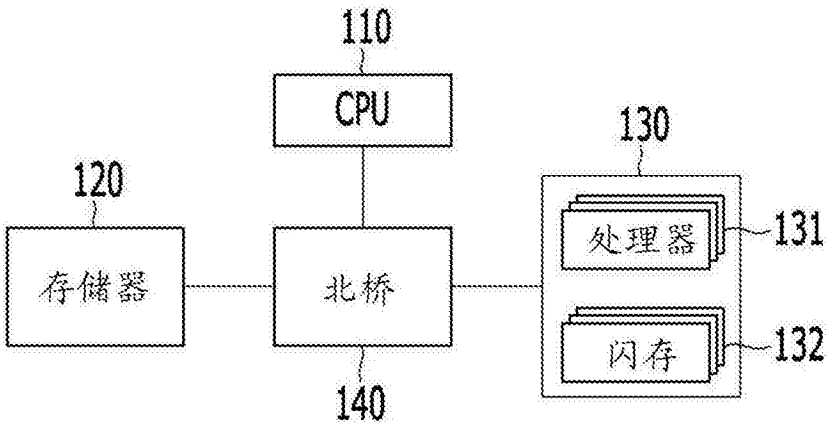

110
CPU
130
120
处理器
131
存储器
北桥
闪存
132
140

图1

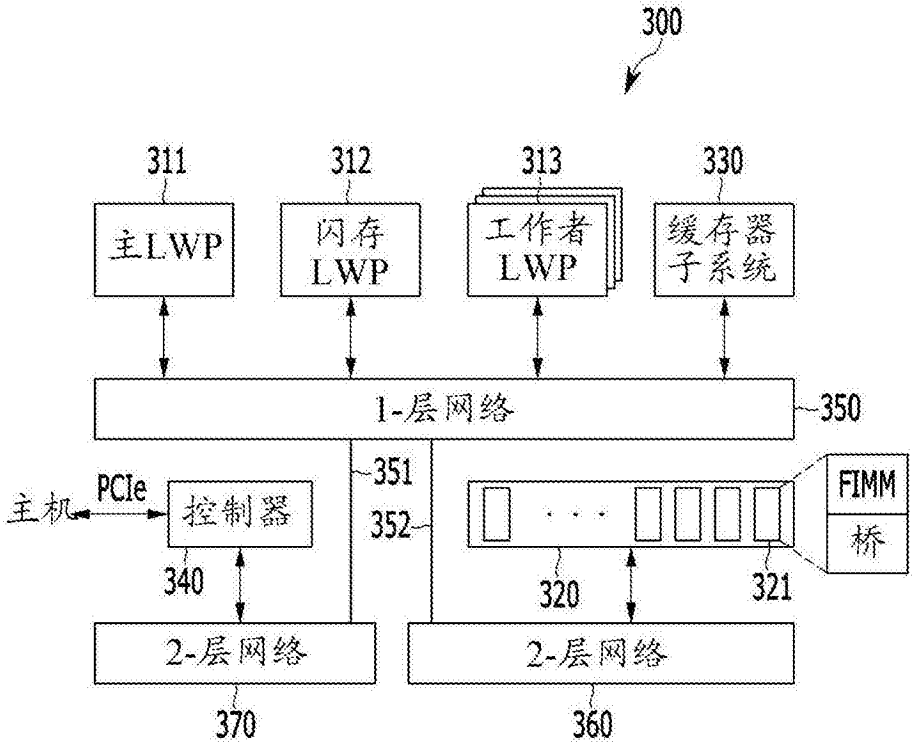

300
311
312
313
330
主LWP
闪存 LWP
工作者 LWP
缓存器 子系统
1-层网络
350
351
主机
PCIe
控制器
352
FIMM
桥
340
320
321
2-层网络
2-层网络
370
360

图2

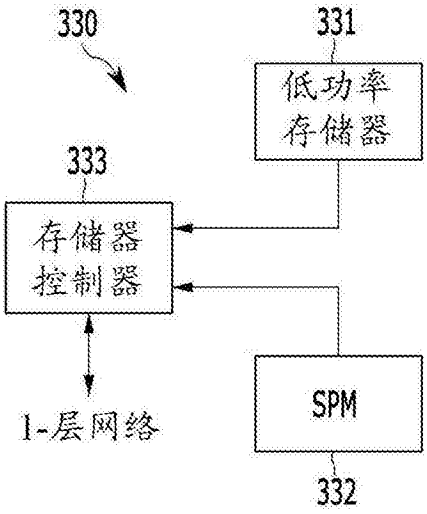

330
331
低功率
存储器
333
存储器
控制器
1-层网络
SPM
332

图3

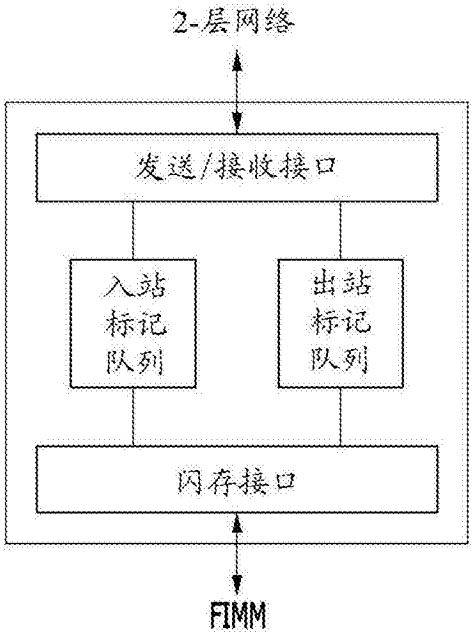

2-层网络
发送/接收接口
入站
标记
队列
出站
标记
队列
闪存接口
FIMM

图4

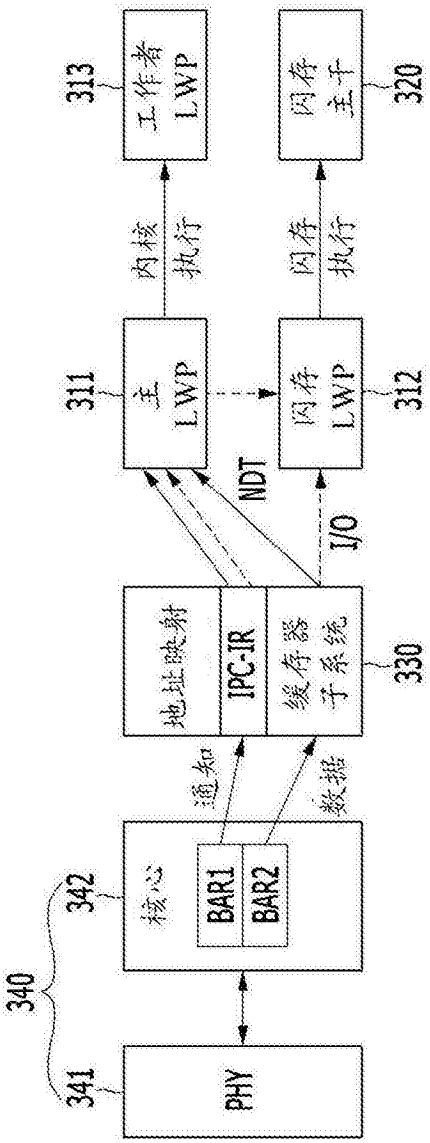

313
工作者 LWP
闪存 主干
320
内核 执行
闪存 执行
311
主 LWP
闪存 LWP
312
NDT
I/O
地址映射
IPC-IR
缓存器 子系统
330
通知
数据
342
核心
BAR1
BAR2
340
341
PHY

图5

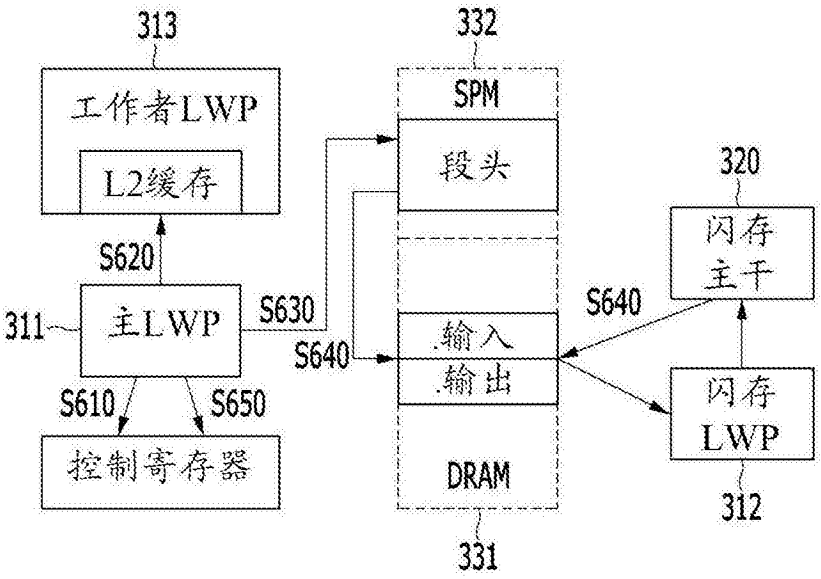

313
工作者LWP
L2缓存
S620
311
主LWP
S630
S610
S650
控制寄存器
332
SPM
段头
S640
输入
输出
DRAM
331
320
闪存
主干
S640
闪存
LWP
312

图6

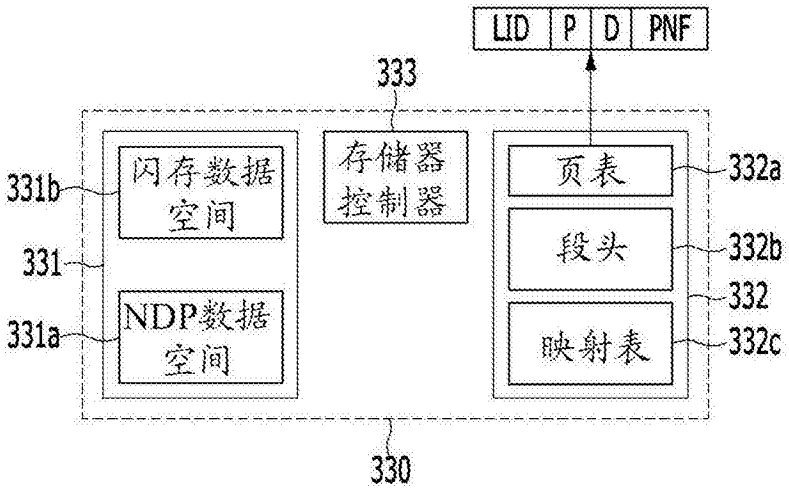

LID
P
D
PNF
333
331b
闪存数据
空间
331
331a
NDP数据
空间
存储器
控制器
页表
332a
段头
332b
332
映射表
332c
330

图7

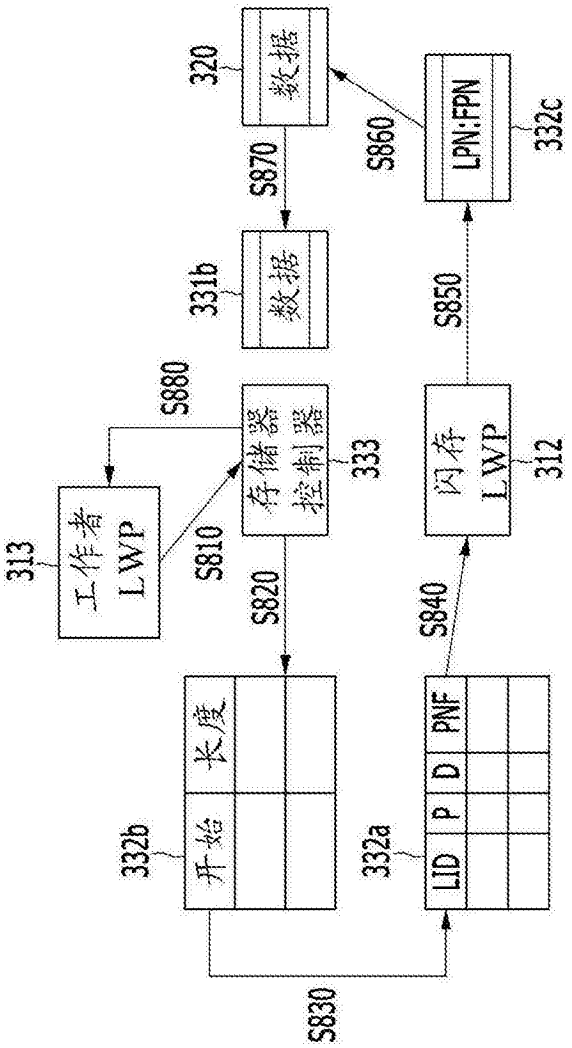

320
数据
S870
331b
数据
S880
存储器控制器
333
313
工作者LWP
S810
S820
长度
开始
332b
S830
S860
LPN:FPN
332c
S850
闪存LWP
312
S840
PNF
D
P
LID
332a

图8

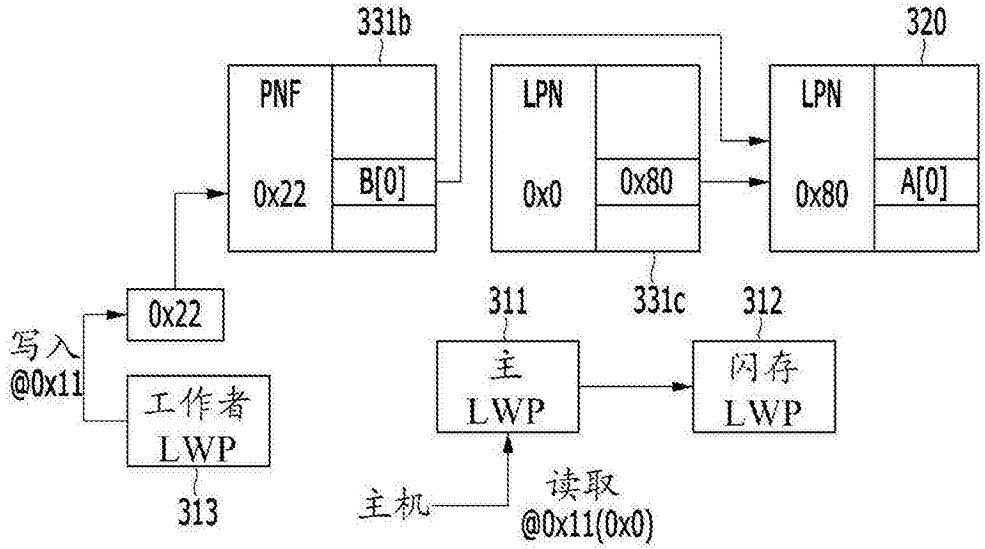

331b
320
PNF
0x22
B[0]
LPN
0x0
0x80
LPN
0x80
A[0]
0x22
写入
@0x11
工作者
LWP
313
311
主
LWP
331c
312
闪存
LWP
主机
读取
@0x11(0x0)

图9

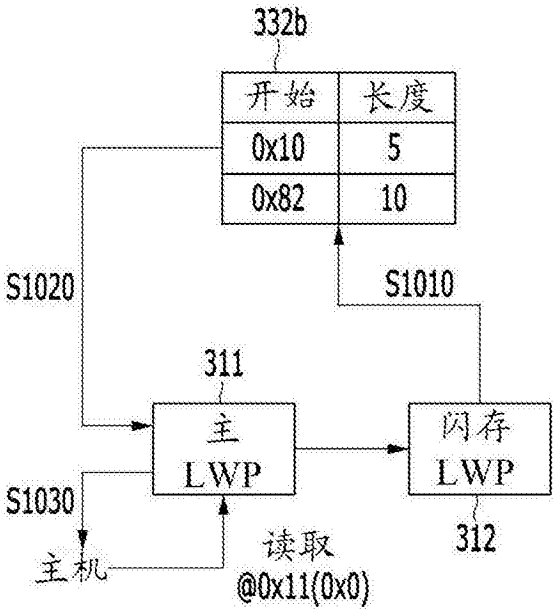


图10

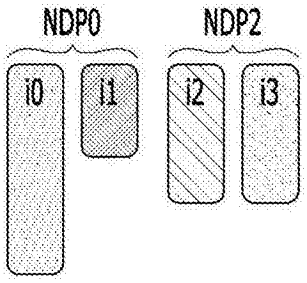


图11

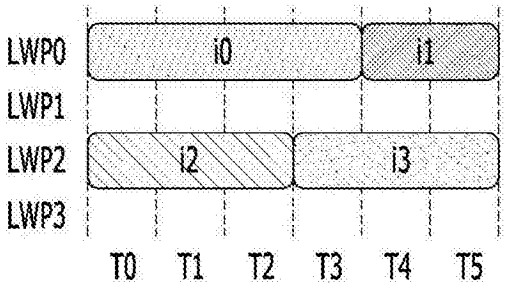


图12

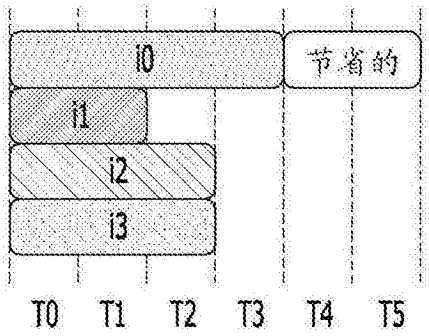


图13

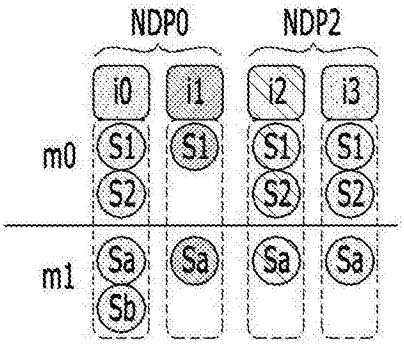


图14

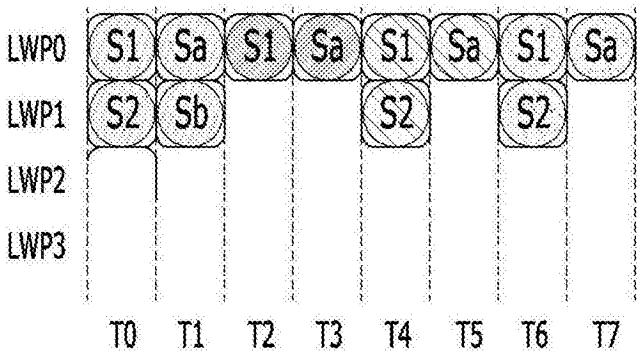


图15

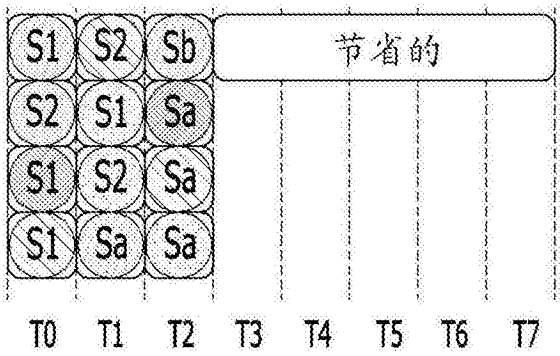


图16