(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 114286987 A
(43)申请公布日 2022.04.05

(21)申请号 202080060590.0

(22)申请日 2020.07.22

(30)优先权数据
16/519,311 2019.07.23 US

(85)PCT国际申请进入国家阶段日
2022.02.25

(86)PCT国际申请的申请数据
PCT/IB2020/056906 2020.07.22

(87)PCT国际申请的公布数据
WO2021/014382 EN 2021.01.28

(71)申请人 ATI科技无限责任公司
地址 加拿大安大略省

(72)发明人 邝东全 本杰明·坤·潘·陈 威廉·劳埃德·阿特金森

(74)专利代理机构 上海胜康律师事务所 31263
代理人 樊英如 张静

(51)Int.Cl.
G06F 12/09(2006.01)
G06F 12/06(2006.01)

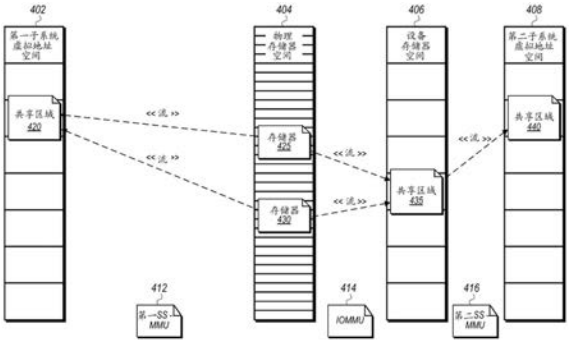
权利要求书3页 说明书5页 附图7页

(54)发明名称

用于异构计算的统一内核虚拟地址空间

(57)摘要

公开了用于实现用于异构计算的统一内核虚拟地址空间的系统、装置和方法。一种系统至少包括运行第一内核的第一子系统、输入/输出存储器管理单元(IOMMU)以及运行第二内核的第二子系统。为了在所述两个子系统之间共享存储器缓冲区，所述第一子系统在由所述第一子系统控制的系统存储器的一部分中分配存储器块。创建从所述第一子系统的所述内核地址空间的第一逻辑地址到所述存储器块的第一映射。然后，所述IOMMU创建第二映射，以从所述第二子系统的所述内核地址空间的第二逻辑地址映射所述存储器块的物理地址。这些映射允许所述第一和第二子系统共享引用所述存储器块的缓冲区指针。



CN 114286987 A

1.一种系统,其包括:

存储器子系统;

具有第一内核地址空间的第一子系统;

具有第二内核地址空间的第二子系统;以及

输入/输出存储器管理单元(IOMMU);

其中所述系统被配置为:

由所述第一子系统在与所述存储器子系统相对应的物理地址空间中的第一物理地址处分配存储器块;

由所述第一子系统创建所述第一内核地址空间中的第一逻辑地址到所述第一物理地址的映射,其中所述第一内核地址空间中的所述第一逻辑地址是与所述第一物理地址的第一偏移量;

由所述IOMMU选择设备地址,所述设备地址是与所述第二内核地址空间中的所述第一逻辑地址的第二偏移量;

创建所述设备地址空间中的所选择设备地址到所述第一物理地址的IOMMU映射;并且

通过所述IOMMU映射,启用所述第一内核地址空间和所述第二内核地址空间两者中的公共共享区域,以供所述第一子系统和所述第二子系统两者使用。

2.如权利要求1所述的系统,其中所述第一子系统被配置为向所述第二子系统传送指向所述存储器块的缓冲区指针,其中所述缓冲区指针在所述第一内核地址空间中指定。

3.如权利要求2所述的系统,其中所述系统还被配置为:

由所述第二子系统在不执行所述缓冲区指针转换的情况下执行引用所述第一内核地址空间中的所述缓冲区指针的存储器访问指令;并且

当由所述第二子系统访问所述存储器块时,通过所述IOMMU映射将所述缓冲区指针转换为所述物理地址空间。

4.如权利要求1所述的系统,其中:

所述第一内核地址空间中的逻辑地址是与所述物理地址空间中的对应地址的第一偏移量;

所述第二内核地址空间中的逻辑地址是与设备地址空间中的对应地址的第二偏移量;并且

所述IOMMU映射创建覆盖,使得所述第二内核地址空间中指向所述存储器块的逻辑地址与所述第一内核地址空间中的逻辑地址相同。

5.如权利要求1所述的系统,其中:

所述第一子系统执行第一操作系统;并且

所述第二子系统执行不同于所述第一操作系统的第二操作系统。

6.如权利要求5所述的系统,其中:

所述第一子系统包括第一存储器管理单元(MMU);并且

所述第二子系统包括不同于所述第一MMU的第二MMU。

7.如权利要求1所述的系统,其中所述公共共享区域映射到所述物理地址空间中的所述存储器块。

8.一种方法,其包括:

由具有第一内核地址空间的第一子系统在物理地址空间中的第一物理地址处分配存储器块;

由所述第一子系统创建所述第一内核地址空间中的第一逻辑地址到所述第一物理地址的映射,其中所述第一内核地址空间中的所述第一逻辑地址是与所述第一物理地址的第一偏移量;

由输入/输出存储器管理单元(IOMMU)选择设备地址,所述设备地址是与第二子系统的第二内核地址空间中的所述第一逻辑地址的第二偏移量;

由所述IOMMU创建所述设备地址空间中的所选择设备地址到所述第一物理地址的IOMMU映射;并且

通过所述IOMMU映射,启用所述第一内核地址空间和所述第二内核地址空间两者中的公共共享区域,以供所述第一子系统和所述第二子系统两者使用。

9.如权利要求8所述的方法,其还包括从所述第一子系统向所述第二子系统传送指向所述存储器块的缓冲区指针,其中所述缓冲区指针在所述第一内核地址空间中指定。

10.如权利要求9所述的方法,其还包括:

由所述第二子系统在不执行所述缓冲区指针转换的情况下执行引用所述第一内核地址空间中的所述缓冲区指针的存储器访问指令;并且

当由所述第二子系统访问所述存储器块时,通过所述IOMMU映射将所述缓冲区指针转换为所述物理地址空间。

11.如权利要求8所述的方法,其中:

所述第一内核地址空间中的逻辑地址是与所述物理地址空间中的对应地址的第一偏移量;

所述第二内核地址空间中的逻辑地址是与设备地址空间中的对应地址的第二偏移量;并且

所述方法还包括由所述IOMMU映射创建覆盖,使得所述第二内核地址空间中指向所述存储器块的逻辑地址与所述第一内核地址空间中的逻辑地址相同。

12.如权利要求8所述的方法,其中:

所述第一子系统执行第一操作系统;并且

所述第二子系统执行不同于所述第一操作系统的第二操作系统。

13.如权利要求12所述的方法,其中:

所述第一子系统包括第一存储器管理单元(MMU);并且

所述第二子系统包括不同于所述第一MMU的第二MMU。

14.如权利要求8所述的方法,其中所述公共共享区域映射到所述物理地址空间中的所述存储器块。

15.一种装置,其包括:

存储器,所述存储器存储程序指令;以及

至少一个处理器,所述至少一个处理器耦接到所述存储器,其中所述程序指令能够由所述至少一个处理器执行以:

由具有第一内核地址空间的第一子系统在物理地址空间中的第一物理地址处分配存储器块;

由所述第一子系统创建所述第一内核地址空间中的第一逻辑地址到所述第一物理地址的映射,其中所述第一内核地址空间中的所述第一逻辑地址是与所述第一物理地址的第一偏移量;

由输入/输出存储器管理单元(IOMMU)选择设备地址,所述设备地址是与第二子系统的第二内核地址空间中的所述第一逻辑地址的第二偏移量;

由所述IOMMU创建所述设备地址空间中的所选择设备地址到所述第一物理地址的IOMMU映射;并且

通过所述IOMMU映射,启用所述第一内核地址空间和所述第二内核地址空间两者中的公共共享区域,以供所述第一子系统和所述第二子系统两者使用。

16.如权利要求15所述的装置,其中所述程序指令能够由所述至少一个处理器执行以从所述第一子系统向所述第二子系统传送指向所述存储器块的缓冲区指针,其中所述缓冲区指针在所述第一内核地址空间中指定。

17.如权利要求16所述的装置,其中所述程序指令能够由所述至少一个处理器执行以:

由所述第二子系统在不执行所述缓冲区指针转换的情况下执行引用所述第一内核地址空间中的所述缓冲区指针的存储器访问指令;并且

当由所述第二子系统访问所述存储器块时,通过所述IOMMU映射将所述缓冲区指针转换为所述物理地址空间。

18.如权利要求15所述的装置,其中:

所述第一内核地址空间中的逻辑地址是与所述物理地址空间中的对应地址的第一偏移量;

所述第二内核地址空间中的逻辑地址是与设备地址空间中的对应地址的第二偏移量;并且

所述IOMMU映射创建覆盖,使得所述第二内核地址空间中指向所述存储器块的逻辑地址与所述第一内核地址空间中的逻辑地址相同。

19.如权利要求15所述的装置,其中:

所述第一子系统执行第一操作系统;并且

所述第二子系统执行不同于所述第一操作系统的第二操作系统。

20.如权利要求19所述的装置,其中:

所述第一子系统包括第一存储器管理单元(MMU);并且

所述第二子系统包括不同于所述第一MMU的第二MMU。

用于异构计算的统一内核虚拟地址空间

背景技术

[0001] 相关技术说明

[0002] 通常,在计算设备之间共享存储器缓冲区将需要设备在设备地址空间中或在物理地址空间中交换指针或句柄。

[0003] 大多数情况下,在使用该指针之前,计算设备会将指针转换为具有存储器管理单元(MMU)页表映射的本地虚拟地址空间。这使得指针本身或包含指针的结构的共享效率低下且具有挑战性。

附图说明

[0004] 通过结合附图参考以下描述,可更好地理解本文描述的方法和机制的优点,在附图中:

[0005] 图1是计算系统的一个实现方式的框图。

[0006] 图2是创建用于异构计算的统一内核虚拟地址空间的一个实现方式的图示。

[0007] 图3是在两个单独的子系统之间共享缓冲区的一个实现方式的图示。

[0008] 图4是将存储器区域映射到多个内核地址空间的一个实现方式的图示。

[0009] 图5是示出用于创建统一内核虚拟地址空间的方法的一个实现方式的通用流程图。

[0010] 图6是示出用于启用多个内核地址空间中的公共共享区域的方法的一个实现方式的通用流程图。

[0011] 图7是示出用于在两个不同子系统之间启用缓冲区指针共享的方法的一个实现方式的通用流程图。

具体实施方式

[0012] 在以下描述中,阐述了众多特定细节以提供对本文呈现的方法和机制的透彻理解。然而,本领域普通技术人员应认识到,可在没有这些具体细节的情况下实践各种实现方式。在一些情况下,并未详细示出众所周知的结构、部件、信号、计算机程序指令和技术,以避免混淆本文所述的方法。应了解,出于说明简明和清楚起见,在图中示出的元件不一定按比例绘制。例如,一些元件的尺寸可相对于其他元件被放大。

[0013] 本文公开了用于实现用于异构计算的统一内核虚拟地址空间的系统、装置和方法。在一个实现方式中,一种系统至少包括运行第一内核的第一子系统、输入/输出存储器管理单元(IOMMU)以及运行第二内核的第二子系统。在一个实现方式中,所述IOMMU创建统一内核地址空间,从而允许所述第一和第二子系统在内核级别共享存储器缓冲区。为了在所述两个子系统之间共享存储器缓冲区,所述第一子系统在由所述第一子系统控制的系统存储器的一部分中分配存储器块。创建了从所述第一子系统的所述第一内核地址空间的第一逻辑地址到所述存储器块的第一映射。然后,所述IOMMU创建第二映射,以从所述第二子系统的所述第二内核地址空间的第二逻辑地址映射该存储器块的物理地址。这些映射允许

所述第一和第二子系统共享所述内核地址空间中的引用所述存储器块的缓冲区指针。

[0014] 现在参考图1,示出了计算系统100的一个实现方式的框图。在一个实现方式中,计算系统100至少包括第一子系统110、第二子系统115、一个或多个输入/输出(I/O)接口120、输入/输出存储器管理单元(IOMMU)125、存储器子系统130以及一个或多个外围设备135。在其他实现方式中,计算系统100可包括其他部件,和/或计算系统100可以不同方式布置。

[0015] 在一个实现方式中,第一子系统110和第二子系统115具有不同的内核地址空间,但是由IOMMU 125为第一子系统110和第二子系统115创建统一内核地址空间。统一内核地址空间允许第一子系统110和第二子系统115在彼此之间传递指针并共享缓冲区。对于第一子系统110和第二子系统115,它们各自的内核地址空间包括内核逻辑地址和内核虚拟地址。在一些体系结构上,内核逻辑地址及其相关联的物理地址相差一个恒定的偏移量。内核虚拟地址不一定具有到表征内核逻辑地址的物理地址的线性、一对一映射。所有内核逻辑地址都是内核虚拟地址,但是内核虚拟地址不一定是内核逻辑地址。

[0016] 在一个实现方式中,第一子系统110和第二子系统115中的每一者都包括执行操作系统的一个或多个处理器。一个或多个处理器还在各种实现方式中执行一个或多个软件程序。第一子系统110和第二子系统115的一个或多个处理器包括任何数量和类型的处理单元(例如,中央处理单元(CPU)、图形处理单元(GPU)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、专用集成电路(AS IC))。此外,第一子系统110包括存储器管理单元(MMU)112,并且第二子系统115包括MMU 117,其中每个MMU处理其对应子系统的虚拟地址到物理地址的转换。虽然第一子系统110和第二子系统115具有不同的内核地址空间,但是由IOMMU 125为第一子系统110和第二子系统115创建了统一内核地址空间,从而允许第一子系统110和第二子系统115来回传递指针并共享存储器子系统130中的缓冲区。

[0017] 存储器子系统130包括任何数量和类型的存储器设备。例如,存储器子系统130中的存储器类型可包括高带宽存储器(HBM)、非易失性存储器(NVM)、动态随机存取存储器(DRAM)、静态随机存取存储器(SRAM)、NAND闪存存储器、NOR闪存存储器、铁电随机存取存储器(FeRAM)等。I/O接口120表示任何数量和类型的I/O接口(例如,外围部件互连(PCI)总线、PCI扩展(PC I-X)、PCI E(PCI高速)总线、吉比特以太网(GBE)总线、通用串行总线(USB))。各种类型的外围设备135可联接到I/O接口120。此类外围设备135包括(但不限于)显示器、键盘、鼠标、打印机、扫描仪、操纵杆或其他类型的游戏控制器、媒体记录设备、外部存储设备、网络接口卡等。

[0018] 在一个实现方式中,为了创建在第一子系统110与第二子系统115之间共享的存储器空间,在由第一子系统110管理的系统存储器的一部分中分配存储器块。在分配了初始存储器块之后,将适当的I/O虚拟地址(VA)分配给第二子系统115。在一个实现方式中,创建了从第二子系统115的内核地址空间到存储器块的物理地址的IOMMU映射。在该实现方式中,IOMMU 125执行第二子系统115到存储器块的虚拟地址映射。然后,当分配了附加存储器时,调用堆分配函数,并基于之前创建的相同I/O VA地址映射地址。然后,向第二子系统115发送消息,向第二子系统115通知统一地址。

[0019] 在各种实现方式中,计算系统100是计算机、膝上型电脑、移动设备、游戏控制台、服务器、流式传输设备、可穿戴设备或各种其他类型的计算系统或设备中的任一种。应注意,计算系统100的部件的数量因实现方式而不同。在其他实现方式中,存在与图1中示出的

数量相比更多或更少的每种部件。还应注意，在其他实现方式中，计算系统100包括图1中未示出的其他部件。另外，在其他实现方式中，计算系统100以与图1中所示不同的方式进行结构化。

[0020] 现在转向图2，示出了创建用于异构计算的统一内核虚拟地址空间的一个实现方式的图示。自图2顶部的垂直虚线从左到右表示第一子系统202、第一子系统202的MMU 204、在第一子系统202上执行的设备驱动程序206、存储器的共享区域208、IOMMU 211、第二子系统214的MMU 212以及第二子系统214。出于本论述的目的，假设第一子系统202具有第一操作系统，第二子系统214具有第二操作系统，并且第二操作系统不同于第一操作系统。出于本论述的目的，还假设第一子系统202和第二子系统214是异构计算系统的一部分。在一个实现方式中，对于该异构计算系统，第一子系统202和第二子系统214各自执行工作负载的某个部分，并且为了执行工作负载，第一子系统202和第二子系统214在彼此之间共享缓冲区指针和缓冲区。图2的图示示出了分配要在第一子系统202与第二子系统214之间共享的存储器的一个示例。

[0021] 在一个实现方式中，由设备驱动程序206执行第一步骤205以创建堆(即，共享存储器区域)。如本文所使用，术语“堆”被定义为映射到物理存储器池的虚拟存储器池。接下来，在步骤210中，在物理存储器子系统中分配期望大小的堆。在步骤215中，创建从内核到堆的映射。在一个实现方式中，当首次创建分拆堆时，新的标志指示堆是否应该来自内核逻辑地址空间。例如，kmalloc函数返回内核逻辑地址空间中的存储器。在一个实现方式中，对于Linux®操作系统，存储器管理器使用Linux genpool库管理为堆分配的缓冲区。在该实现方式中，当分配了缓冲区时，在内部池中标记所述缓冲区并返回物理地址。然后分拆堆用缓冲区和sg_table描述符包装该物理地址。在一个实现方式中，当缓冲区映射在内核地址空间中时，heap_map_kernel函数在步骤220中使用kmap而不是vmap来映射缓冲区。函数kmap基于逻辑映射将缓冲区映射到给定的虚拟地址。kmap或vmap的选择由在分拆堆创建期间给出的新标志控制。替代地，使用genpool库、分拆应用程序编程接口(API)和新标志与内核映射一起创建堆。在这种情况下，内核映射返回预先映射的地址。用户模式映射仍将即时应用。对于第二子系统214，由第一子系统202分配的任何共享存储器缓冲区的管理与分拆堆包装的genpool类似。新的API允许向分拆堆添加外部缓冲区。第二子系统214上的本地任务使用相同的API从该分拆堆分配。在步骤220之后，共享存储器区域的分配完成。

[0022] 接下来，在步骤225中，由IOMMU 211为共享存储器区域分配输入/输出(I/O)虚拟地址。然后，在步骤230中，为共享存储器区域保留连续的DMA地址空间块。接下来，在步骤235中，共享存储器区域由IOMMU 211映射到第二子系统214的内核地址空间。应注意，在设备驱动程序206关闭之前不应释放IOMMU映射。在步骤240中使从第一子系统202的内核地址空间到共享存储器区域的映射无效。然后，在步骤245中，通过执行存储器释放函数来释放内核地址空间映射。

[0023] 现在参考图3，示出了在两个单独的子系统之间共享缓冲区的一个实现方式的图示。自图3顶部向下延伸的垂直虚线从左到右表示第一子系统302、第一子系统302的MMU 304、在第一子系统302上执行的设备驱动程序306、存储器的共享区域308、IOMMU 310、第二子系统314的MMU 312以及第二子系统314。出于本论述的目的，假设第一子系统302具有第一操作系统，第二子系统314具有第二操作系统，并且第二操作系统不同于第一操作系统。

[0024] 对于循环交换305，分配了存储器块并且由设备驱动程序306创建了从第一子系统302的内核地址空间到存储器块的物理地址的映射。然后由MMU 304维护所述映射。存储器块可以专门分配给第二子系统314，在第一子系统302与第二子系统314之间共享，或者专门分配给第一子系统302。然后，从第一子系统302向第二子系统314发送具有存储器块的地址和大小的消息。在一个实现方式中，可以带外发送消息。在第二子系统314接收到消息之后，由MMU 312创建和维护从第二子系统314的内核地址空间到存储器块的物理地址的映射。对于循环交换315，使用共享区域308在第一子系统302与第二子系统314之间交换数据。通过统一内核虚拟地址，缓冲区指针1^{st}_SS_buf和2^{nd}_SS_buf是相同的并且可以在第一子系统302与第二子系统314之间自由交换并且使用genpoo l库进一步分区。

[0025] 现在转向图4，示出了将存储器区域映射到多个内核地址空间的一个实现方式的图示。在一个实现方式中，异构计算系统(例如，图1的系统100)包括具有其自己的独立操作系统的多个不同子系统。图4中所示的垂直取向的矩形块表示异构计算系统的不同部件的地址空间。从左到右，图4中所示的地址空间是：第一子系统虚拟地址空间402、物理存储器空间404、设备存储器空间406和第二子系统虚拟地址空间408。

[0026] 在一个实现方式中，第一子系统虚拟地址空间402包括共享区域420，所述共享区域被映射到物理存储器空间404的存储器块425和存储器块430。在一个实现方式中，共享区域420到存储器块425和存储器块430的映射由第一子系统MMU 412创建和维护。为了允许共享区域420与第二子系统共享，存储器块425和存储器块430由IOMMU 414映射到设备存储器空间406的共享区域435。然后，共享区域435被第二子系统MMU 416映射到第二子系统虚拟地址空间408的共享区域440。通过这种映射方案，第一子系统和第二子系统能够在其内核地址空间中与彼此共享缓冲区指针。

[0027] 现在转向图5，示出了用于创建统一内核虚拟地址空间的方法500的一个实现方式。出于论述的目的，此实现方式中的步骤和图6至图7的那些步骤按顺序次序示出。然而，应注意，在所描述方法的各种实现方式中，可同时执行、可按与所示不同的次序执行、或者可完全省略所描述要素中的一个或多个。还在需要时执行其他附加的要素。本文中所描述的各种系统或装置中的任一者被配置为实现方法500。

[0028] 第一子系统在对应于存储器子系统的物理地址空间中的第一物理地址处分配存储器块(框505)。在一个实现方式中，第一子系统执行具有第一内核地址空间的第一操作系统。接下来，第一子系统创建第一内核地址空间中的第一逻辑地址到第一物理地址的映射(框510)。应注意，第一内核地址空间的第一逻辑地址空间是与第一物理地址的第一线性偏移量。然后，I OMMU创建第二内核地址空间中的第二逻辑地址到第一物理地址的IOMMU映射(框515)。在一个实现方式中，第二内核地址空间与第二子系统相关联。应注意，第二内核地址空间的第二逻辑地址空间是与第一物理地址的第二线性偏移量。

[0029] 接下来，第一子系统将缓冲区指针传送到第二子系统，其中所述缓冲区指针指向第一逻辑地址(框520)。然后，第二子系统用缓冲区指针生成访问请求并将所述访问请求传送到IOMMU(框525)。接下来，IOMMU使用先前创建的IOMMU映射将缓冲区指针的虚拟地址转换为第一物理地址(框530)。然后，第二子系统访问第一物理地址处的存储器块(框535)。在框535之后，方法500结束。

[0030] 现在转向图6，示出了用于启用多个内核地址空间中的公共共享区域的方法600的

一个实现方式。第一子系统在对应于存储器子系统的物理地址空间中的第一物理地址处分配存储器块(框605)。出于本论述的目的,假设方法600所描述的系统包括第一子系统和第二子系统。还假设第一子系统执行具有第一内核地址空间的第一操作系统,并且第二子系统执行具有第二内核地址空间的第二操作系统。第一子系统创建第一内核地址空间中的第一逻辑地址到第一物理地址的映射,其中第一内核地址空间中的第一逻辑地址是与第一物理地址的第一偏移量(框610)。

[0031] 接下来,IOMMU选择设备地址,所述设备地址是与第二内核地址空间中的第一逻辑地址的第二偏移量(框615)。然后,I OMMU创建设备地址空间中的所选择设备地址到第一物理地址的IOMMU映射(框620)。IOMMU映射启用第一内核地址空间和第二内核地址空间两者中的公共共享区域,以供第一子系统和第二子系统两者使用(框625)。在框625之后,方法600结束。

[0032] 现在转向图7,示出了用于在两个不同子系统之间启用缓冲区指针共享的方法700的一个实现方式。第一子系统将第一内核地址空间的第一区域映射到物理地址空间中的第二区域(框705)。在一个实现方式中,物理地址空间对应于由第一子系统控制的系统存储器。接下来,IOMMU将第二内核地址空间的第三区域映射到设备地址空间中的第四区域(框710)。出于本论述的目的,假设第二内核地址空间对应于不同于第一子系统的第二子系统。然后,I OMMU将设备地址空间中的第四区域映射到物理地址空间中的第二区域,以使第一和第三区域中的地址指向第二区域中的匹配(即,相同)的地址(框715)。第一子系统和第二子系统能够共享第一或第二内核地址空间中的引用物理地址空间中的第二区域的缓冲区指针(框720)。在框720之后,方法700结束。

[0033] 在各种实现方式中,使用软件应用程序的程序指令来实现本文描述的方法和/或机制。例如,设想到可由通用处理器或专用处理器执行的程序指令。在各种实现方式中,此类程序指令可以由高级编程语言表示。在其他实现方式中,程序指令可从高级编程语言编译成二进制形式、中间形式或其他形式。替代地,可以写入描述硬件的行为或设计的程序指令。此类程序指令可以由诸如C的高级编程语言表示。替代地,可以使用诸如Ver i l og的硬件设计语言(HDL)。在各种实现方式中,程序指令存储在多种非暂时性计算机可读存储介质中的任一种上。在使用期间,计算系统可访问存储介质,以将程序指令提供到计算系统以用于程序执行。一般来讲,此类计算系统至少包括一个或多个存储器和被配置为执行程序指令的一个或多个处理器。

[0034] 应强调,上述实现方式仅是实现方式的非限制性示例。一旦充分了解上述公开内容,许多变型和修改将对本领域技术人员显而易见。所附权利要求意图被解释为涵盖所有此类变型和修改。

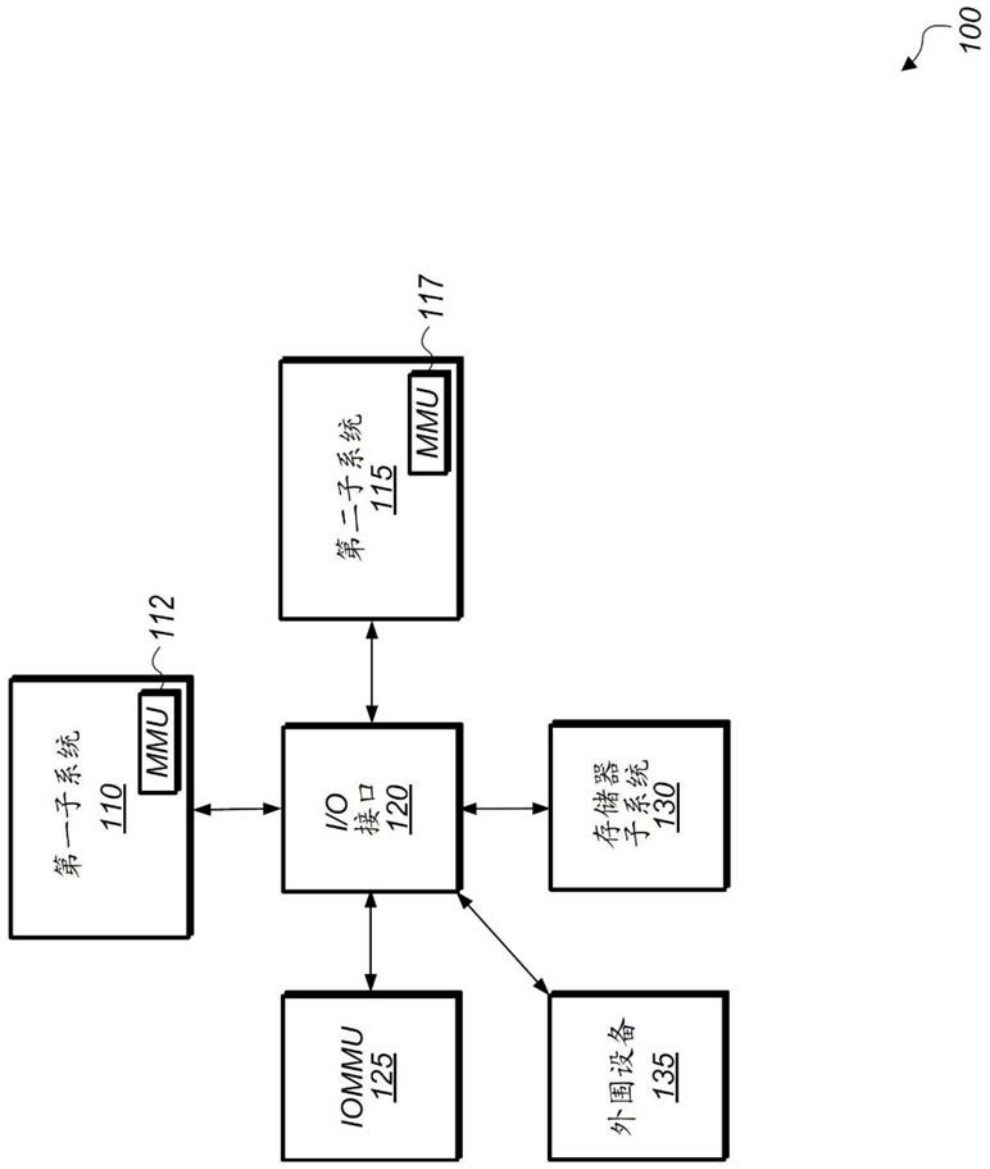

100
117
第二子系统
115
MMU
112
第一子系统
110
MMU
I/O
接口
120
存储器
子系统
130
IOMMU
125
外围设备
135

图1

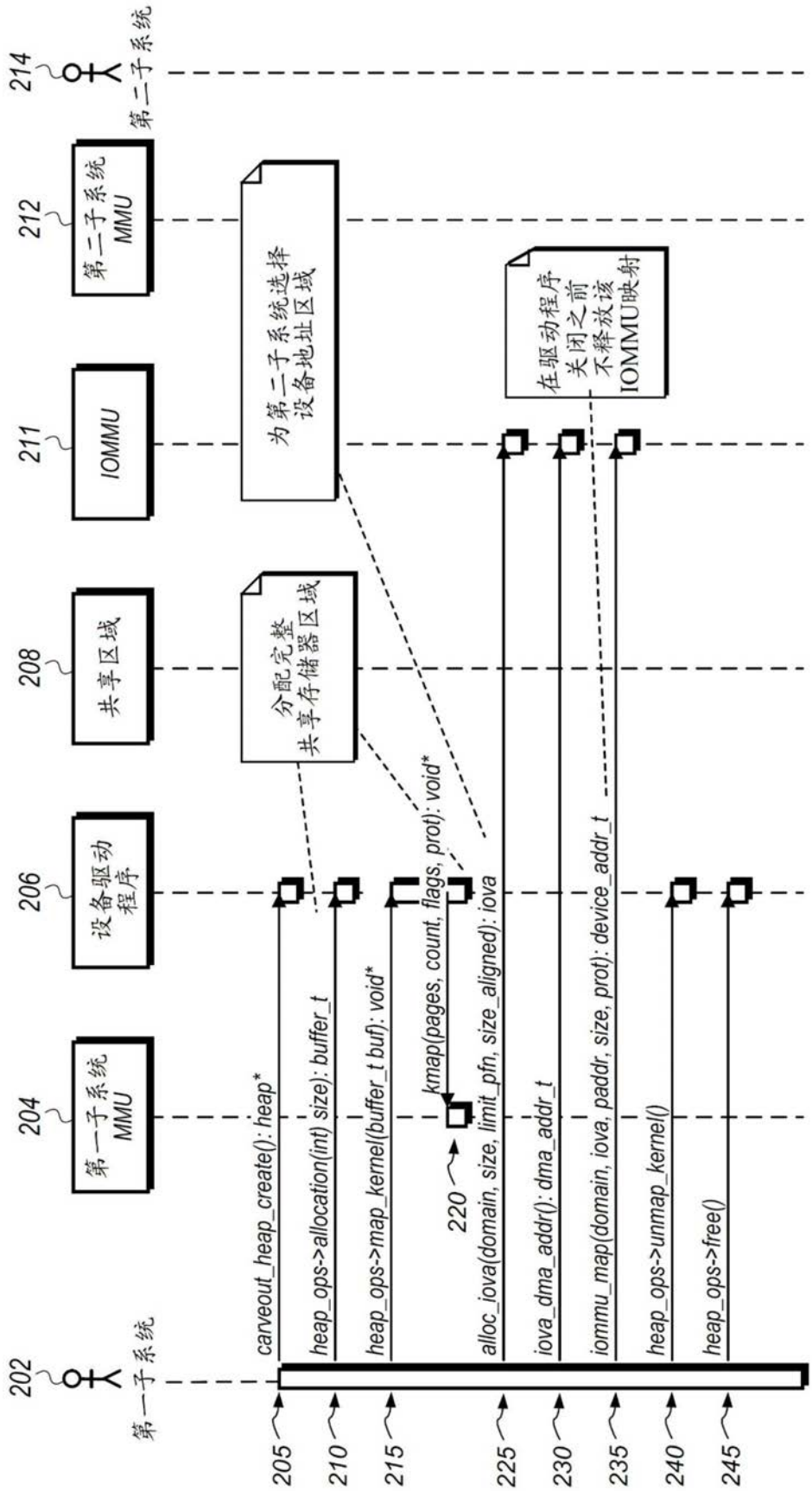

202
第一子系统
204
第一子系统 MMU
206
设备驱动程序
208
共享区域
211
IOMMU
212
第二子系统 MMU
214
第二子系统
为第二子系统选择设备地址区域
分配完整共享存储器区域
在驱动程序关闭之前不释放该IOMMU映射
carveout_heap_create(): heap*
heap_ops->allocation(int) size): buffer_t
heap_ops->map_kernel(buffer_t buf): void*
kmap(pages, count, flags, prot): void*
220
alloc_iova(domain, size, limit_pfn, size_aligned): iova
iova_dma_addr(): dma_addr_t
iommu_map(domain, iova, paddr, size, prot): device_addr_t
heap_ops->unmap_kernel()
heap_ops->free()
205
210
215
225
230
235
240
245

图2

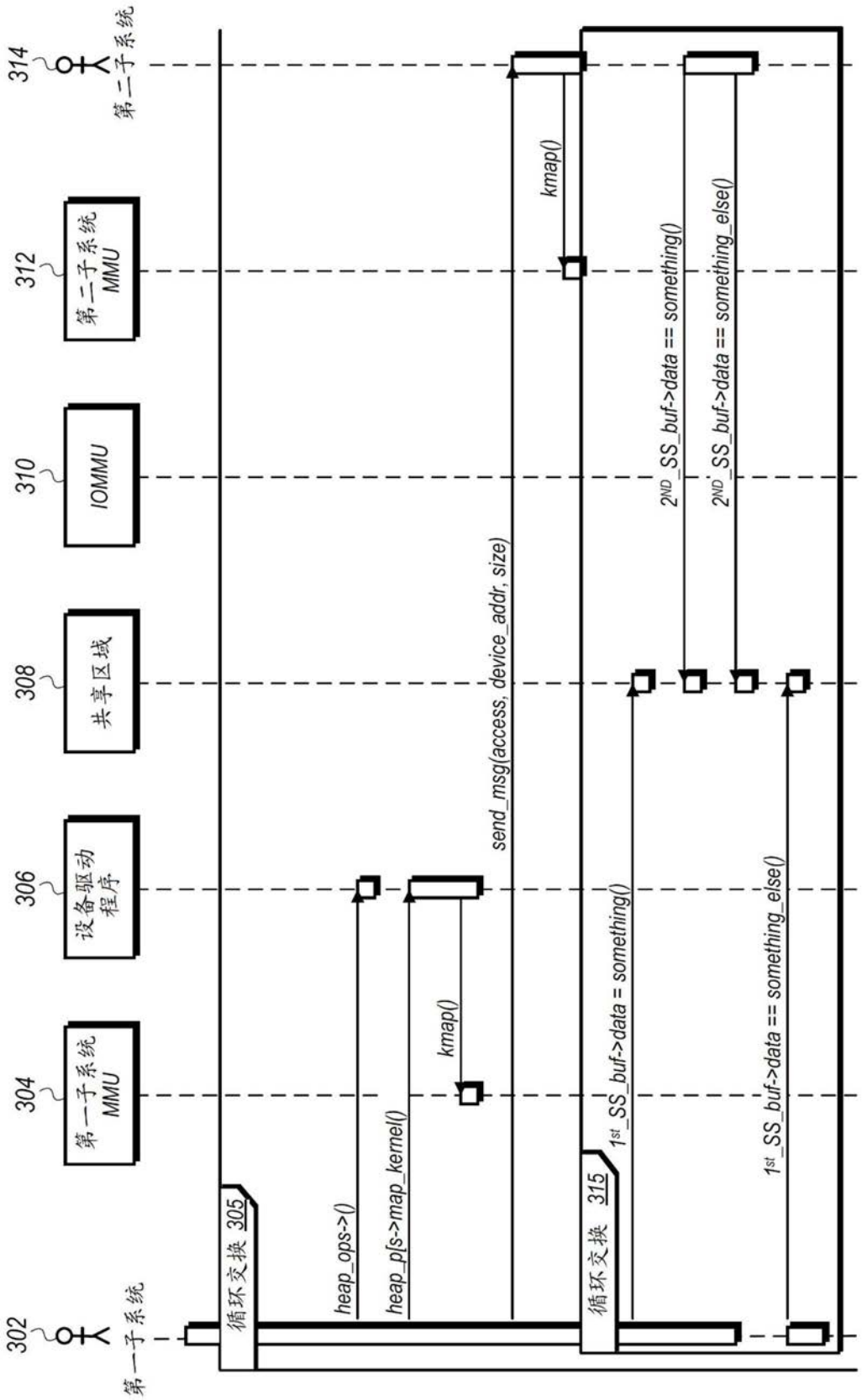

302
第一子系统
304
第一子系统 MMU
306
设备驱动程序
308
共享区域
310
IOMMU
312
第二子系统 MMU
314
第二子系统
循环交换 305
heap_ops->()
heap_p[s->map_kernel()
kmap()
send_msg(access, device_addr, size)
kmap()
循环交换 315
1st_SS_buf->data = something()
2ND_SS_buf->data == something()
2ND'_SS_buf->data == something_else()
1st_SS_buf->data == something_else()

图3

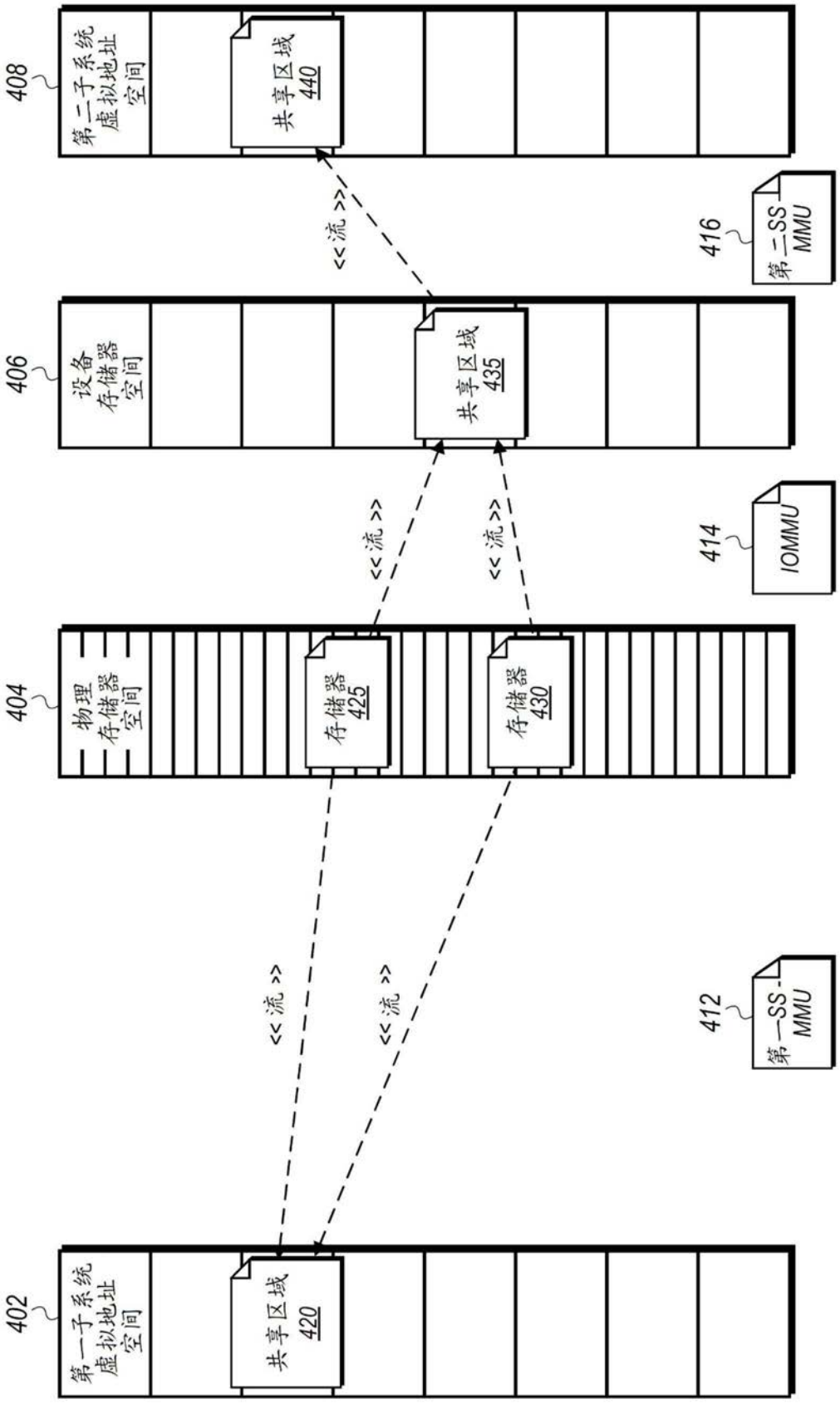

408
第二子系统虚拟地址空间
共享区域 440
<< 流 >>
416
第二SS MMU
406
设备存储器空间
共享区域 435
<< 流 >>
<< 流 >>
414
IOMMU
404
物理存储器空间
存储器 425
存储器 430
<< 流 >>
<< 流 >>
412
第一SS-1 MMU
402
第一子系统虚拟地址空间
共享区域 420

图4

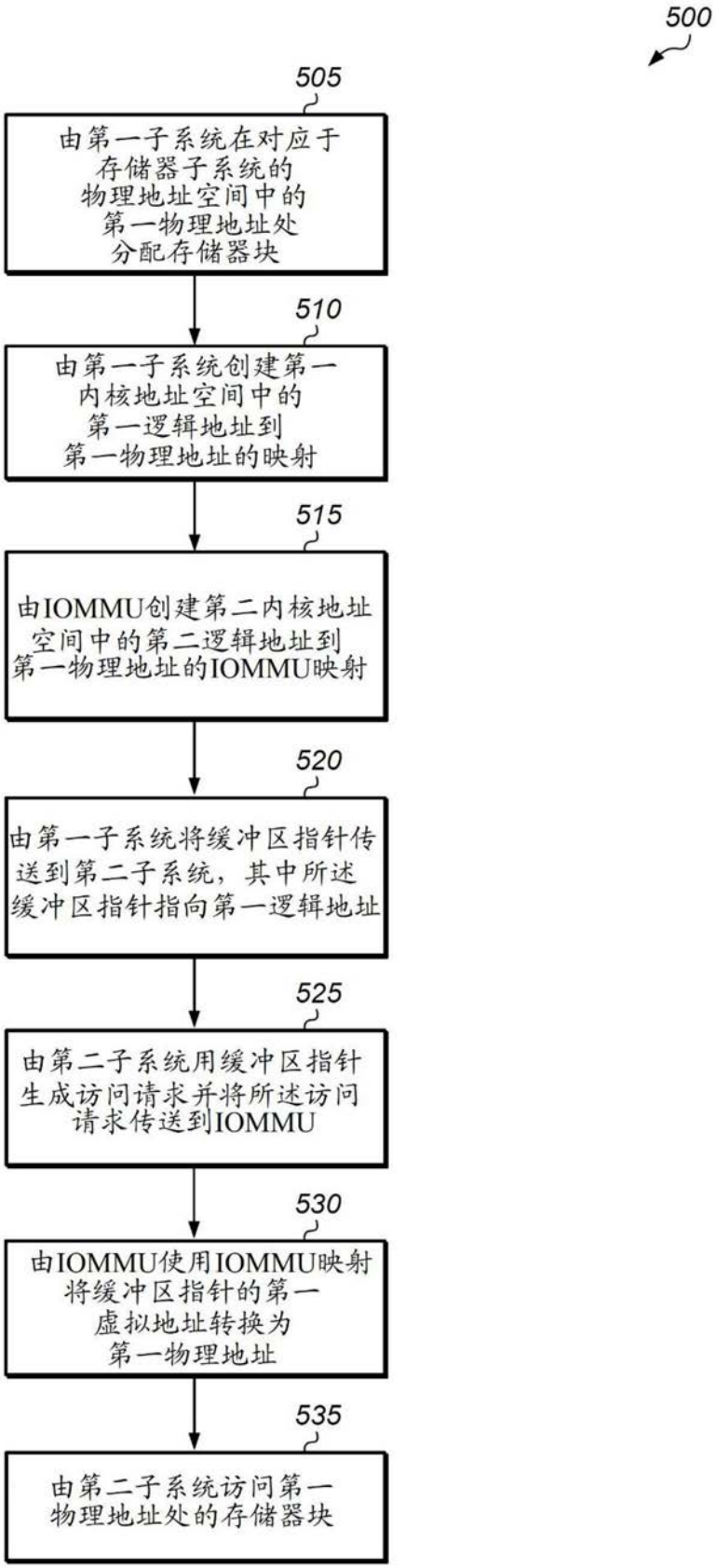

500
505
由第一子系统在对应于存储器子系统的物理地址空间中的第一物理地址处分配存储器块
510
由第一子系统创建第一内核地址空间中的第一逻辑地址到第一物理地址的映射
515
由IOMMU创建第二内核地址空间中的第二逻辑地址到第一物理地址的IOMMU映射
520
由第一子系统将缓冲区指针传送到第二子系统，其中所述缓冲区指针指向第一逻辑地址
525
由第二子系统用缓冲区指针生成访问请求并将所述访问请求传送到IOMMU
530
由IOMMU使用IOMMU映射将缓冲区指针的第一虚拟地址转换为第一物理地址
535
由第二子系统访问第一物理地址处的存储器块

图5

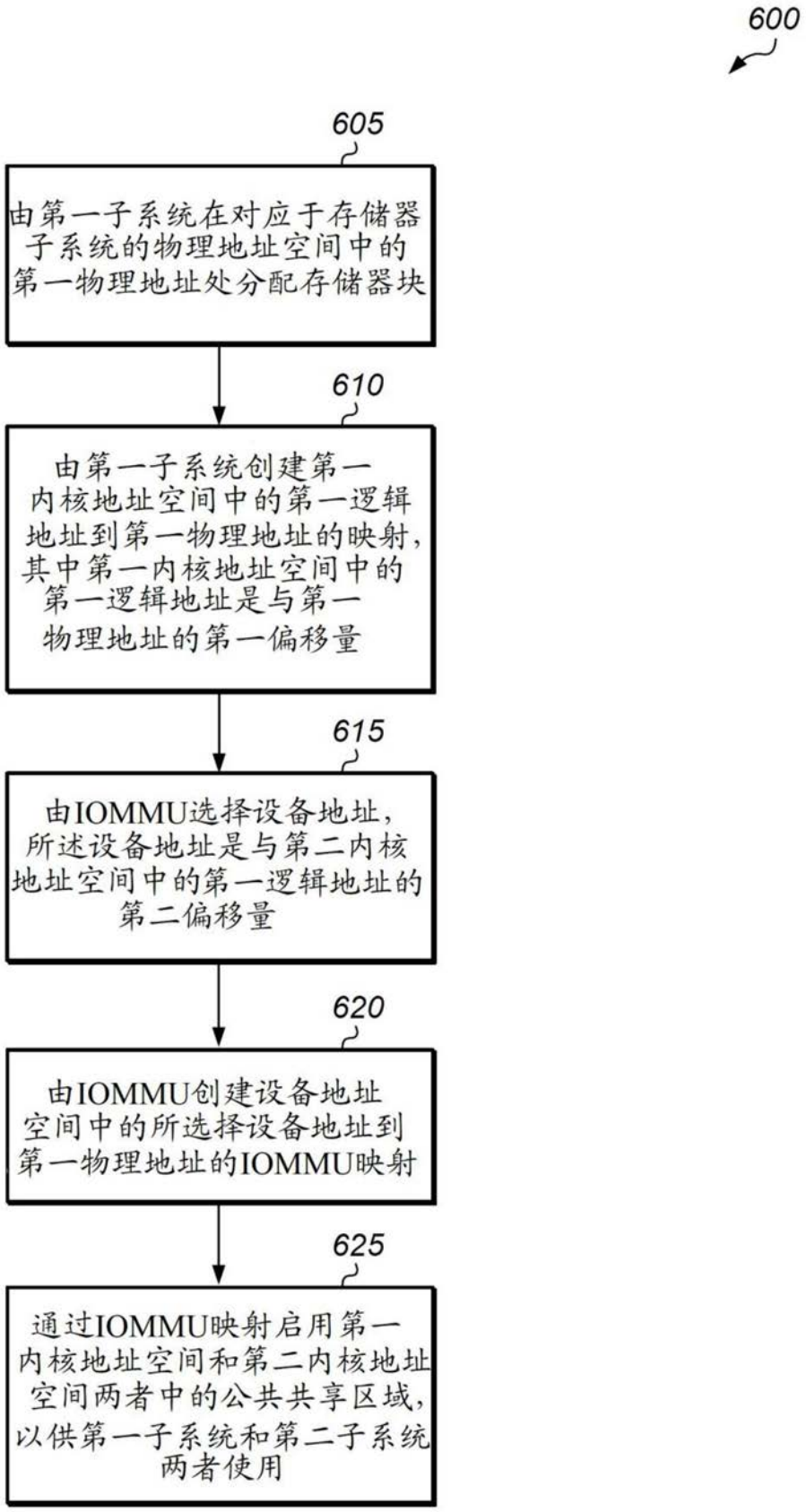

600
605
由第一子系统在对应于存储器子系统的物理地址空间中的第一物理地址处分配存储器块
610
由第一子系统创建第一内核地址空间中的第一逻辑地址到第一物理地址的映射，其中第一内核地址空间中的第一逻辑地址是与第一物理地址的第一偏移量
615
由IOMMU选择设备地址，所述设备地址是与第二内核地址空间中的第一逻辑地址的第二偏移量
620
由IOMMU创建设备地址空间中的所选择设备地址到第一物理地址的IOMMU映射
625
通过IOMMU映射启用第一内核地址空间和第二内核地址空间两者中的公共共享区域，以供第一子系统和第二子系统两者使用

图6

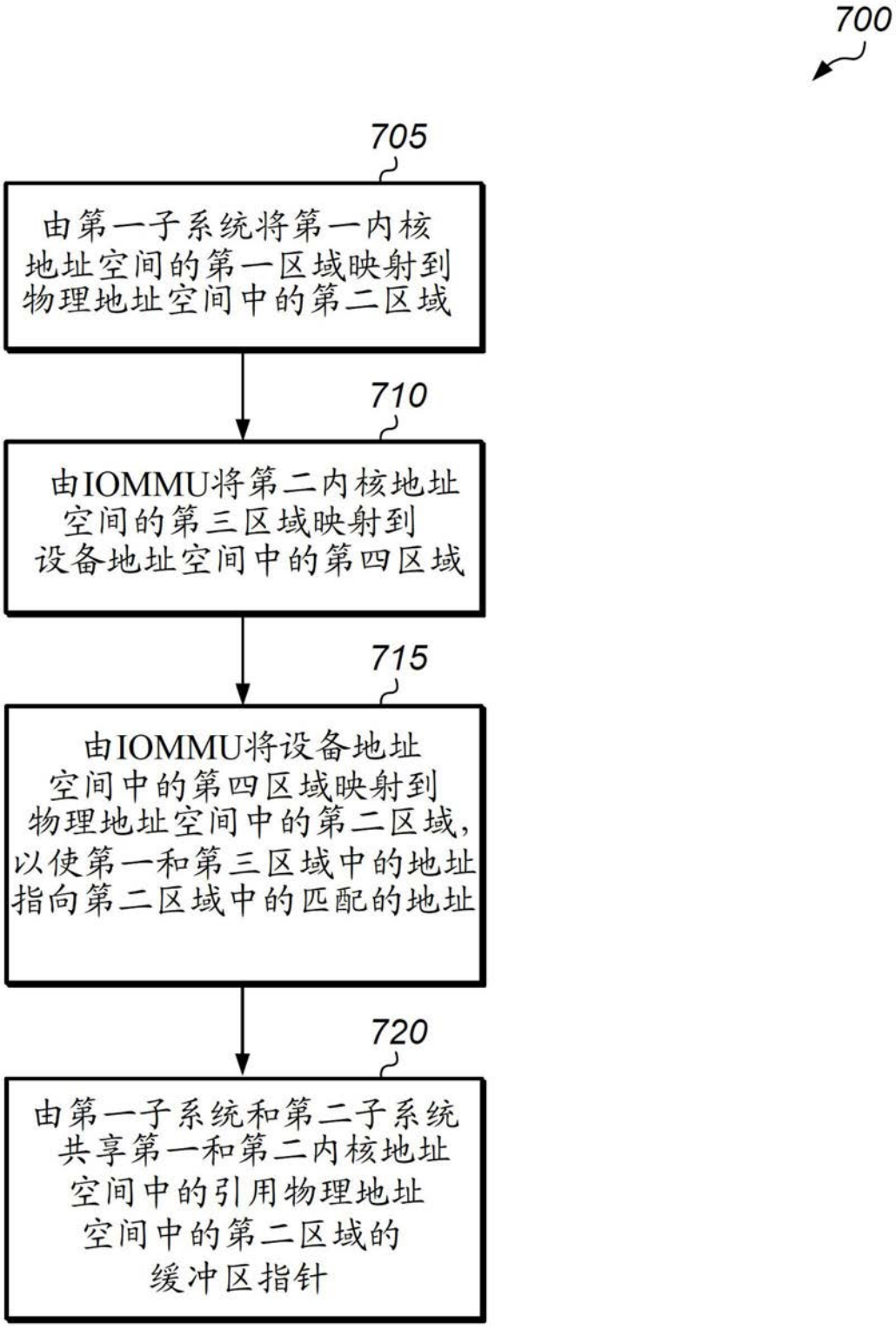

700
705
由第一子系统将第一内核地址空间的第一区域映射到物理地址空间中的第二区域
710
由IOMMU将第二内核地址空间的第三区域映射到设备地址空间中的第四区域
715
由IOMMU将设备地址空间中的第四区域映射到物理地址空间中的第二区域，以使第一和第三区域中的地址指向第二区域中的匹配的地址
720
由第一子系统和第二子系统共享第一和第二内核地址空间中的引用物理地址空间中的第二区域的缓冲区指针

图7