



(12) 发明专利

(10) 授权公告号 CN 102282540 B

(45) 授权公告日 2015. 07. 01

(21) 申请号 200980154914. 0

(22) 申请日 2009. 12. 10

(30) 优先权数据

12/337, 159 2008. 12. 17 US

(85) PCT国际申请进入国家阶段日

2011. 07. 19

(86) PCT国际申请的申请数据

PCT/US2009/067409 2009. 12. 10

(87) PCT国际申请的公布数据

W02010/077751 EN 2010. 07. 08

(73) 专利权人 超威半导体公司

地址 美国加利福尼亚州

(72) 发明人 迈克尔·弗朗克

(74) 专利代理机构 上海胜康律师事务所 31263

代理人 李献忠

(51) Int. Cl.

G06F 9/38(2006. 01)

G06F 12/10(2006. 01)

(56) 对比文件

EP 1139222 A1, 2001. 10. 04, 说明书第 4 栏第 19 段 - 第 8 栏第 36 段和附图 3、4.

US 2008/0059771 A1, 2008. 03. 06, 说明书第 1 页第 6 段、第 4 页第 53 段 - 第 61 段和附图 1A.

US 2008/0059771 A1, 2008. 03. 06, 说明书第 1 页第 6 段、第 4 页第 53 段 - 第 61 段和附图 1A.

US 2008/0301408 A1, 2008. 12. 04, 说明书第 2 页第 19 段 - 第 12 页第 122 段和附图 2A.

CN 1782989 A, 2006. 06. 07, 全文.

US 2002/0010848 A1, 2002. 01. 24, 说明书第 2 页第 25 段 - 第 5 页第 44 段和附图 1、4.

审查员 陈敏

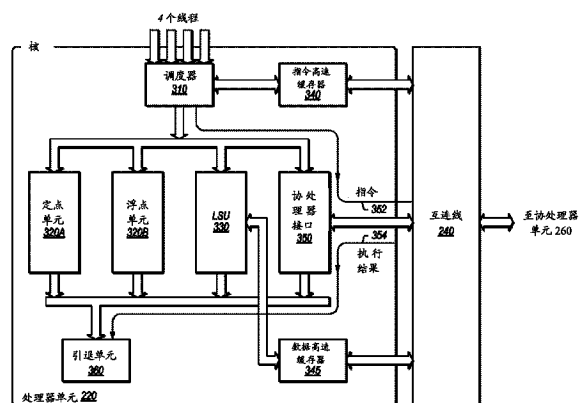
权利要求书2页 说明书14页 附图10页

(54) 发明名称

带有共享指令流的协处理器单元

(57) 摘要

本发明公开了一种处理器单元和协处理器单元。在一实施方式中,处理器单元包括在指令流中接收一组指令并且将所述一组指令提供给协处理器单元的功能单元。协处理器单元执行所述指令并且发起将响应于所述一组指令的一组执行结果传输给处理器单元的功能单元。处理器功能单元可以通过实现基于数据包的协议的共享总线电路耦合到协处理器单元上。在不同的实施方式中,功能单元可以更改位于协处理器单元内的转换后备缓冲器(TLB)中的条目、恢复和中止在协处理器单元上执行的线程等等。



1. 一种计算机系统,包括:

多个处理器功能单元,包括第一处理器功能单元,其配置成从调度器单元接收第一指令流中的第一组指令,将所述第一组指令提供给协处理器单元,并且从所述协处理器单元接收响应于所述第一组指令的第一组执行结果,其中,所述第一组执行结果的传输由所述协处理器单元发起;以及

引退单元,其配置成引退由所述多个处理器功能单元执行的指令;

其中,所述第一处理器功能单元包括引退接口单元,该引退接口单元配置成向所述引退单元提供所述第一组执行结果以使所述第一组指令引退。

2. 根据权利要求 1 所述的计算机系统,还包括:

处理器单元,其包含所述第一处理器功能单元和所述调度器单元;以及
所述协处理器单元。

3. 根据权利要求 1 所述的计算机系统,还包括第二处理器功能单元,所述第二处理器功能单元配置成从所述调度器单元接收所述第一指令流中的第二组指令,其中,所述第二处理器功能单元还配置成实施一个或多个执行操作,以产生响应于所述第二组指令的第二组执行结果。

4. 根据权利要求 2 所述的计算机系统,还包括:

共享的总线电路,其使所述第一处理器功能单元耦合至所述协处理器单元,其中,所述总线电路配置成实现基于数据包的协议;以及

耦合至所述共享的总线电路的存储器访问装置。

5. 根据权利要求 4 所述的计算机系统,其中,所述基于数据包的协议是如下协议之一:超传输协议、PCI-Express 协议。

6. 根据权利要求 1 所述的计算机系统,还包括:

多个协处理器单元,该多个协处理器单元包含所述协处理器单元;

其中,所述第一处理器功能单元配置成将指令组提供给所述多个协处理器单元中的不同协处理器单元,并且从所述多个协处理器单元中的不同协处理器单元接收相应的执行结果组;

其中,所述第一处理器功能单元还配置成保存路由信息,并且基于所述路由信息将所述指令组路由至所述多个协处理器单元中的所述不同协处理器单元。

7. 根据权利要求 1 所述的计算机系统,其中,所述第一处理器功能单元配置成更改位于所述协处理器单元内的转换后备缓冲器(TLB)中的条目。

8. 根据权利要求 1 所述的计算机系统,其中,所述第一处理器功能单元还配置成:

从所述协处理器单元接收页错误的指示;

将所述页错误通知所述计算机系统的操作系统;以及

向所述协处理器单元指明所述操作系统已对所述页错误进行了维护。

9. 根据权利要求 1 所述的计算机系统,还包括:

包含所述第一处理器功能单元和所述调度器单元的多线程的处理器单元;以及

其中,所述协处理器单元配置成执行如下线程,所述线程具有在所述处理器单元上执行的响应线程。

10. 根据权利要求 1 所述的计算机系统,其中,所述第一处理器功能单元配置成恢复和

中止在所述协处理器单元上执行的线程。

11. 根据权利要求 1 所述的计算机系统,包括:

包含所述第一处理器功能单元和所述引退单元的处理器单元。

12. 根据权利要求 1 所述的计算机系统,其中,所述第一处理器功能单元配置成在所述协处理器单元处存储在所述协处理器单元上执行的线程的执行状态。

13. 一种协处理器单元,包括:

相伴处理器接口单元,其配置成从相伴处理器单元中的协处理器接口单元中接收指令,其中,所述指令来自所述相伴处理器单元共享的指令流;

处理核,其配置成执行所接收的所述指令,以便产生第一组执行结果并且发起所述第一组执行结果通过所述相伴处理器接口单元至所述相伴处理器单元的传送;

其中所述协处理器接口单元包括引退接口单元,其配置成直接向所述相伴处理器单元的引退单元提供被传送的第一组执行结果以使所述接收的指令引退,并且其中所述引退单元配置成引退由所述相伴处理器单元中的多个处理器功能单元执行的指令。

14. 根据权利要求 13 所述的协处理器单元,其中,所述处理核包含转换后备缓冲器(TLB),并且其中,所述处理核配置成:

通过所述相伴处理器接口单元来将页错误的指示传输至在相伴处理器上执行的操作系统;以及

通过所述相伴处理器接口单元来接收所述操作系统已对所述页错误进行了维护的指示。

15. 根据权利要求 13 所述的协处理器单元,其中,所述协处理器单元包含高速缓存器,并且其中,相伴处理器还配置成更改所述协处理器单元的所述高速缓存器中的条目。

16. 根据权利要求 13 所述的协处理器单元,其中,所述处理核配置成实施页表搜索。

17. 根据权利要求 13 所述的协处理器单元,其中,所述协处理器单元配置成中止和恢复在所述相伴处理器单元上执行的线程。

18. 根据权利要求 13 所述的协处理器单元,其中,所述协处理器单元配置成产生微代码指令,以便实现所接收的所述指令。

带有共享指令流的协处理器单元

技术领域

[0001] 本发明大体上涉及计算机处理,具体地说是,涉及在共享指令流的处理器单元和协处理器单元上执行指令。

发明内容

[0002] 公开了用于在协处理器单元上执行指令的各种技术。

[0003] 在一种实施方式中,公开了包含第一处理器功能单元的一种设备,所述第一处理器功能单元耦合至协处理器单元。在一种实施方式中,所述第一处理器功能单元配置成从调度器单元(scheduler unit)接收第一指令流中的第一组指令,并将所述第一组指令提供给协处理器单元。然后,所述第一处理器功能单元配置成从所述协处理器单元接收响应于所述第一组指令的第一组执行结果,在那里,所述协处理器单元发起(initiate)对所述第一组执行结果的传输。在某些实施方式中,所述设备包含总线电路,所述总线电路使所述第一处理器功能单元耦合至所述协处理器单元,并且所述总线电路配置成实现基于数据包的协议。在一些实施方式中,所述基于数据包的协议是超传输协议或者PCI-Express协议。在其他实施方式中,所述第一处理器功能单元配置成更改(alter)位于所述协处理器单元内的转换后备缓冲器(TLB)中的条目(entry)。

[0004] 在又一实施方式中,公开了包含处理器接口单元和处理核(processing core)的协处理器单元。所述处理器接口单元配置成从相伴(companion)处理器单元中的协处理器接口单元中接收指令,在那里,所述指令来自所述相伴处理器单元共享的指令流。所述处理核配置成执行所接收的所述指令,以便产生第一组执行结果并且发起将所述第一组执行结果通过所述处理器接口单元传送至所述相伴处理器单元。在某些实施方式中,所述处理核包含转换后备缓冲器(TLB),并且配置成通过所述处理器接口将页错误的指示传输至在所述相伴处理器上执行的操作系统,并且通过所述处理器接口来接收所述操作系统已对所述页错误进行了维护的指示。在各种实施方式中,所述处理核配置成实施页表搜索。在一些实施方式中,所述协处理器单元与所述相伴处理器单元一起分享对存储器的一致查看(coherent view)。

[0005] 在又一实施方式中,公开了一种有形的计算机可读的存储介质,所述存储介质存储能够由包含协处理器单元和相伴处理器单元的计算机系统执行的程序指令。所存储的所述程序指令包含第一组程序指令,每个所述第一组程序指令具有一个第一组操作码,在那里,所述第一组程序指令能够由所述相伴处理器单元之内的一个或多个功能单元来执行。所存储的所述程序指令还包含第二组程序指令,每个所述第二组程序指令具有一个第二组操作码,在那里,所述第二组程序指令能够在所述协处理器单元之内执行。响应于接收所述第二组程序指令之内的程序指令,所述相伴处理器单元设置成将这些指令提供给耦合到所述协处理器单元上的所述相伴处理器单元的协处理器接口单元。所述协处理器接口单元配置成将所述第二组程序指令之内的程序指令传送给所述协处理器单元以便执行,并且随后接收由协处理器单元发起的、对所传送的所述指令的执行结果的传输。

附图说明

[0006] 下面的详细描述参考了附图,现在对附图进行简要描述。

[0007] 图 1 示出计算机系统的实施方式的框图。

[0008] 图 2 示出包含处理器单元和一个或多个协处理器单元的处理器子系统的一个实施方式的框图。

[0009] 图 3 示出包含协处理器接口单元的处理器单元的一个实施方式的框图。

[0010] 图 4 示出协处理器单元的一个实施方式的框图。

[0011] 图 5 示出协处理器接口的一个实施方式的框图。

[0012] 图 6A 至图 6D 示出协处理器单元的各种实施方式的框图。

[0013] 图 7 示出通过协处理器接口在协处理器上执行指令的方法的一个实施方式的流程图。

[0014] 图 8 示出在协处理器单元上切换线程的方法的实施方式的流程图。

[0015] 图 9 示出由协处理器单元利用虚拟地址空间来实现的方法的一个实施方式的流程图。

[0016] 图 10 示出由第一处理器功能单元实现的方法的一个实施方式的流程图。

[0017] 图 11 示出由协处理器单元实现的方法的一个实施方式的流程图。

[0018] 图 12 示出计算机可读的介质的一个实施方式的框图。

具体实施方式

[0019] 本说明书包含参考“一个 (one) 实施方式”或者“一 (an) 实施方式”。短语“在一个实施方式中”或者“在一实施方式中”的出现不一定是指相同的实施方式。特定的特征、结构或者特性能够以与本公开文本一致的任何合适的方式组合。

[0020] 专业术语。以下段落针对本公开文本中的术语提供定义和 / 或语境 (包含附加的权利要求) :

[0021] “包括”。该术语是开放式的。像在附加的权利要求中所用到的那样,该术语不排除另外的结构或步骤。考虑权利要求这样描述:“一种包括第一处理器功能单元的设备。。。。。”。此项权利要求不排除所述设备包含另外的部件 (例如第二处理器功能单元、高速缓存器等)。

[0022] “计算机系统”。该术语具有其在现有技术中通常的且已被接受的含义,并且包含一个或多个一起操作的计算装置,每个计算装置包含一个或多个处理器单元和存储器。

[0023] “配置”。像在这里用到的那样,该术语的含义是:某一块硬件或者软件在操作时被布置用于实施某一特别的任务或者某些特别的任务。因此,“配置成”实施任务 A 的计算机系统的含义是:所述计算机系统包含电路、存储在存储器中的程序指令或者其他结构,它们在操作所述计算机系统期间实施或者能够用于实施任务 A。类似地,“配置成”实施任务 B 的计算机系统包含如下指令,即,如果由计算机系统来执行这些指令,就实施任务 B。

[0024] “软件”。该术语具有其在现有技术中通常的且已被接受的含义,其包含任何形式的计算机程序;程序包含对计算机系统的操作进行指引的指令组。

[0025] “软件应用”。从广义上讲,该术语是指软件实例。计算机备份程序是一种类型的

软件应用。该术语“软件应用”除了运行在操作系统上的程序之外还包含操作系统。

[0026] “处理器单元”。该术语具有其在现有技术中通常的且已被接受的含义,其至少包含配置成执行程序指令的任何电路系统(例如中央处理单元(CPU))。像在这里用到的那样,处理器单元可以是指“主”处理器单元或者协处理器单元。与协处理器单元配合进行操作的主处理器单元可以被认为是“主机(host)”或“相伴(companion)”处理器单元。处理器单元可以在单个电路小片(die)上具有一个或多个处理“核”。另外,处理器单元可以横跨多个电路小片地分布。

[0027] “处理核”。该术语具有其在现有技术中通常的且已被接受的含义,其具有在处理器单元内部的多数个处理电路(例如CPU)之一。例如,处理核可以是指多核(芯片级多处理器)装置中的一个核。

[0028] “协处理器单元”。从广义上讲,该术语是指用于对相伴处理器的功能进行增补的任何处理器单元。与相伴处理器共享指令流的协处理器单元不同于如下加速器单元,所述加速器单元接收与所述相伴处理器的指令流分开的指令流。

[0029] “指令”。该术语具有其在现有技术中通常的且已被接受的含义,其至少包含指引处理器单元或核实施某组操作的一组比特(bits)。在各种实施方式中,指令可以包含操作标识符(例如操作码)、直接数据(例如操作数)、地址等。

[0030] “指令流”。该术语可以描述提供给处理器单元或核的一系列连续及时指令(例如来自软件应用)。指令流可以包含在地址空间内连续的指令,以及相对于某原始排序来说无序地执行的指令。

[0031] “执行结果”。从广义上讲,该术语是指作为处理器单元或核执行指令的结果而出现的任何状态上的变化。例如,执行结果可以包含合成值(resultant value)(例如相加和)、状态/完成通知、异常指示(例如指示页错误的标志)、命令(例如告知处理器单元中止执行线程)等。

[0032] “调度器单元”。从广义上讲,该术语是指处理器单元或核内部的电路系统,其准备用于执行的指令(例如在处理器单元或核中读取、解码和/或调度用于执行的指令的电路系统,或者将指令指引至处理器单元或核内部的合适功能单元的电路系统)。

[0033] “功能单元”。该术语具有其在现有技术中通常的且已被接受的含义,其从广义上讲是指处理器单元或核内部执行指令或者使指令以其名义得以执行的任何电路系统。处理器单元或核可以包含每个都致力于实现特定功能的多个功能单元(例如定点运算指令、浮点运算指令、加载/保存指令等)。“功能单元”包含如下电路系统,所述电路系统例如接收由协处理器发起的、对执行结果的传输,并且使所述指令引退(retire)。

[0034] “共享的总线电路”。该术语是指由三个或更多个装置共享的任何总线电路。因此,共享的总线电路不同于两个装置之间的“点对点”总线。

[0035] “存储器访问装置”。从广义上讲,该术语是指能够使一个或多个处理单元访问存储器的电路系统(典型地是分立芯片)。

[0036] “线程”。像在这里用到的那样,该术语是指一部分程序,所述一部分程序能够独立于所述程序的其他方面运行。

[0037] “多线程”。该术语是指处理器单元或核支持同步执行多线程的能力。

[0038] “响应线程”。从广义上讲,该术语是指在不同处理器单元或核上执行的相关线程。

例如,如果在处理器上执行线程激活了在另一处理器上开始执行的线程,那么这些线程中的每个线程都可以被描述为“响应于”另一线程。在不同处理器单元或核上执行时与另一线程通信的线程也可以被描述为“响应线程”。

[0039] “恢复”。该术语是指在多线程处理的处理器单元或核上重启执行线程的动作。

[0040] “中止”。该术语是指在多线程处理的处理器单元或核上停止执行线程的动作。

[0041] “转换后备缓冲器 (TLB)”。该术语具有其在现有技术中通常的且已被接受的含义,其包含保存有从虚拟地址到物理地址的转换的高速缓存器。

[0042] “页表搜索 (Page table walk)”。该术语具有其在现有技术中通常的且已被接受的含义,并且包含:在 TLB 内部找寻从虚拟地址到物理地址的转换失败后,在存储器中查找所述转换的过程。

[0043] ***

[0044] 现在看图 1,描绘了计算机系统 100 的一个实施方式。本公开文本描述了包含在计算机系统内部的处理器单元和协处理器的各种实施方式。公开的计算机系统可以是任何合适类型的计算装置。如图所示,计算机系统 100 包含处理器子系统 180,该计算机子系统 180 通过互连线 160 (例如系统总线)耦合到存储器 120 和 I/O 接口 140 上。I/O 接口 140 耦合到一个或多个 I/O 装置 150 上。计算机系统 100 可以是任何不同类型的装置,包括(但不限于):个人电脑系统、台式电脑、便携式或笔记本电脑、大型电脑系统、掌上电脑、工作站、网络计算机、消费电子设备,如移动电话、寻呼机、个人数据助理 (PDA)。计算机系统 100 也可以是任何类型的网络化外围装置,例如像存储装置、交换机、调制解调器、路由器等。虽然单个的计算机系统 100 是为了方便起见而在图 1 中示出的,但是系统 100 也可以作为两个或两个以上的计算机系统一起运行来实现。

[0045] 处理器子系统 180 可以包括一个或多个处理器或处理单元。例如,处理器子系统 180 可以包括一个或多个处理器单元,所述一个或多个处理器单元耦合至一个或多个协处理器单元。在计算机系统 100 的各种实施方式中,处理器子系统 180 的多个实例可以耦合至互连线 160。在不同的实施方式中,处理器子系统 180 (或 180 内的每个处理器单元)都可以包含高速缓存器或者其他形式的板上存储器。在下面会结合图 2 对处理器子系统 180 加以详细描述。

[0046] 存储器 120 可以由处理器子系统 180 来使用。存储器 120 可以用不同的物理存储器介质来实现,如硬盘存储、软盘存储、可移动磁盘存储、闪速存储器、随机存取存储器 (RAM-SRAM、EDORAM、SDRAM、DDR SDRAM、**Rambus®** RAM 等)、ROM (PROM、EEPROM 等),诸如此类。计算机系统 100 中的存储器不限于存储器 120。更确切地说,计算机系统 100 可以说成是具有包含各种类型存储器/存储器单元的“存储器子系统”。例如,在一个实施方式中,计算机系统 100 的存储器子系统可以包括存储器 120、处理器子系统 180 中的高速缓存存储器以及 I/O 装置 150 上的存储器(例如硬盘驱动器、存储阵列)。因此,短语“存储器子系统”代表计算机系统 100 中各种类型的可能的存储器介质。在一些实施方式中,存储器子系统保存着能由处理器子系统 180 执行的程序指令。

[0047] 根据不同的实施方式,I/O 接口 140 可以是任何配置成耦合至其他装置并且与其他装置通信的各种类型的接口。在一个实施方式中,I/O 接口 140 是从前端总线至一个或多个后端总线的桥接芯片。I/O 接口 140 可以通过一个或多个响应的总线或其他接口耦合

到一个或多个 I/O 装置 150 上。I/O 装置的例子包括存储装置（硬盘驱动器、光盘驱动器、可移动闪存驱动器、存储阵列、SAN（存储区域网）或者它们的相关控制器）、网络接口装置（例如至本地网络或广域网）或者其他装置（例如图形、用户界面装置等）。在一个实施方式中，计算机系统 100 通过网络接口装置耦合到网络上。

[0048] 现在看图 2，图示说明了处理器子系统 180 的一个实施方式。如图所示，处理器子系统 180 包括处理器单元 220、协处理器单元 260A 和 260B，以及存储器控制器 280，它们全部通过互连线 240（例如点对点或共享总线电路）耦合在一起。在一个实施方式中，处理器单元 220 和协处理器单元 260 可以位于同一电路小片上。在另一实施方式中，处理器单元 220 和协处理器单元 260 可以位于分开的电路小片上。协处理器单元 260B 和存储器控制器 280 用虚线示出，以表明处理子系统 180 的所有实施方式都无需包括这些方框。（这并不是说，子系统 180 的其他部件是必需的。）例如，处理器单元 220 可以仅耦合至单个协处理器单元（如 260A）；可供选择地，单元 220 可以耦合至多个协处理器单元（例如 260A 和 260B）。在其他实施方式中，可以有附加的协处理器单元。在不同的实施方式中，处理器单元 220 和协处理器单元 260 可以共享一个共同的存储器控制器 280。存储器控制器 280 例如可以配置成访问主系统存储器（例如存储器 120）。在其他实施方式中，每个处理器单元 220 和协处理器单元 260 可以耦合至各自的存储器控制器。

[0049] 在一个实施方式中，处理器单元 220 是一种通用处理器单元（如中央处理单元（CPU）），其包括一个或多个执行单元。可供选择地，单元 220 可以是特殊用途的处理器，例如图形处理器。在一个实施方式中，处理器单元 220 配置成执行使用存储器控制器 280 从存储器 120 中读取的指令。单元 220 的体系结构可以具有不同的特征，例如它可以是流水线式的。在其他实施方式中，处理器单元 220 可以实现用于同时执行多个线程的多线程体系结构。处理器单元 220 可以执行（没有限制）特定应用的指令以及操作系统指令。这些指令可以允许实现任何数量的特征，包括（只是其中的一个例子）虚拟存储器。

[0050] 如图 2 所示，处理器单元 220 可以作为相伴处理器耦合至一个或多个协处理器单元 260，允许单元 220 向协处理器单元 260 提供指令。由相伴处理器单元 220 提供给协处理器单元 260 的指令可以在一个共同的指令流（即单元 220 读取有待执行的指令，并提供那些读取的指令中的某些指令给单元 260 用于执行）之内。从处理器单元 220 提供给协处理器单元 260 的某些指令可以由处理器单元 220 内部的功能单元所产生的“控制”指令，用于控制对协处理器单元 260 的操作。在下面会结合图 3 对处理器单元 220 加以详细描述。

[0051] 协处理器单元 260A 和 260B 可以被用来帮助实施处理器单元 220 的工作。像处理器单元 220 那样，协处理器单元 260 并不局限于任何特殊的功能或体系结构。在各种实施方式中，协处理器单元 260A 和 260B 可以是通用的或者特殊用途的处理器（例如图形处理单元（GPU）、视频解码处理器、加密处理器、队列管理等）。在一个实施方式中，协处理器单元 260 可以作为现场可编程门阵列（FPGA）来实现。在一些实施方式中，协处理器单元 260 可以是流水线式的。在一些实施方式中，协处理器单元 260 可以采用多线程体系结构。在不同的实施方式中，协处理器单元 260 可以配置成执行微代码指令，以便执行从单元 220 收到的某些指令。在某些实施方式中，协处理器单元 260 可以支持虚拟存储器的使用。在下面会结合图 4 对协处理器单元 260 加以详细描述。

[0052] 在一个实施方式中，互连线 240 是一个共享的总线电路，其将处理器单元 220 耦合

至协处理器单元 260。在一个实施方式中,互连线 240 可以实现“虚拟通道”,其允许处理器单元 220 通过基于数据包的协议,例如超传输协议或者 PCI-Express 协议,与协处理器单元 260 通信。在一些实施方式中,互连线 240 可以是前端总线。在一个实施方式中,协处理器单元 260 可以通过北桥型装置耦合到处理器单元 220 上。

[0053] 在一个实施方式中,存储器控制器 280 配置成提供用于处理器单元 220 和 / 或协处理器单元 260 的接口,以便访问存储器(例如存储器 120)。存储器控制器 280 可以用于例如读取指令或者加载和保存数据。在一个实施方式中,处理器单元 220 可以用存储器控制器 280 来读取在处理器单元 220 或协处理器单元 260 中的用于执行的指令。在另一个实施方式中,协处理器单元 260 可以用存储器控制器 280 来读取其自身的指令或数据。

[0054] 在某些实施方式中,处理器单元 220 和协处理器单元 260 配置成用于实现共享存储器(例如存储器 120)的统一,在该共享存储器中,处理器单元 220 和 / 或协处理器单元 260 改变(例如无效或更新)保存在各自的处理器中的高速缓存的条目。在一个实施方式中,处理器单元 220 和协处理器单元 260 实现“探听(snooping)”协议,在该协议中,每个处理器都监视存储器总线(例如互连线 160),以确定是否使高速缓存的条目失效 / 更新高速缓存的条目。在可供选择的实施方式中,当高速缓存的条目将要失效或者更新时,处理器单元 220 和协处理器单元 260 相互指示。在一些实施方式中,处理器单元 220 和协处理器单元 260 配置成共享对虚拟存储器的统一查看(coherent view)。例如,处理器单元 220 和协处理器单元 260 可以针对一个由操作系统在处理器之一(例如处理器单元 220)上执行而产生的、共同的页表而维持分开的 TLB。

[0055] 现在看图 3,图示说明了处理器单元 220。处理器单元 220 是可以耦合到一个或多个协处理器单元 260 上的相伴处理器单元的实施方式。如图所示,处理器单元 220 包括调度器单元 310、定点单元 320A 和浮点单元 320B、加载保存单元(LSU)330、协处理器接口单元 350 和引退单元 360。在所示的实施方式中,调度器单元 310 耦合到处理器单元 220 内部的指令高速缓存器 340 上,虽然其他的实施方式可以不包括高速缓存器 340。同样地,LSU330 可以耦合到数据高速缓存器 345 上,如图 3 所示。单元 220 的其他实施方式可以不包括高速缓存器 345。在不同的实施方式中,处理器单元 220 包括一个或多个功能单元。需要注意的是,在不同的实施方式中,图 3 所示的功能单元 320、330 和 350 是示例性的,并且可以不被包括在处理器单元 220 中。在不同的实施方式中,处理器单元 220 可以包括单一的功能单元 320 或多数个功能单元 320(如图所示)。在一些实施方式中,处理器单元 220 可以包括多数个协处理器接口单元 350,其中,每个协处理器接口单元 350 都耦合到一个或多个协处理器单元 260 上。在一个实施方式中,协处理器接口单元 350 可以位于处理器单元 220 的外部。

[0056] 处理器单元 220 可以是配置成执行多个线程(例如 4 个线程)的多线程处理器单元。不同线程的指令通过调度器单元 310 被调度至调度器单元 320、330 或 350 用于执行。正如下文将描述的那样,协处理器接口单元 350 配置成将其接收到的指令提供给协处理器单元 260 用于执行,随后收到协处理器发起的执行结果的传输。当单元 320、330 和 350 可用时,将执行结果提供给引退单元 360。

[0057] 调度器单元 310 可以包含逻辑电路来读取和解码指令,从而准备在随后的框中将它们用于执行。在一个实施方式中,调度器单元 310 可以包括逻辑电路,以从指令高速缓存

器 340 读取指令。调度器单元 310 也可以实现逻辑电路,以应对指令高速缓存器缺失以及从虚拟指令读取地址至物理地址的转换(例如,调度器单元 310 可以包括指令转换后备缓冲器(ITLB)或硬件来实施页表搜索)。在一些实施方式中,调度器单元 310 可以基于指令中的信息将解码的指令路由至单元 320、330 和 350。例如,调度器单元 310 可以基于指令内部的操作代码(操作码)将指令路由至协处理器接口 350。在不同的实施方式中,调度器单元 310 可以实现多种调度技术。例如,在一个实施方式中,调度器单元 310 可以使指令排队并且使用无序执行(OoOE)算法来调度它们。此外,在一些实施方式中,调度器单元 310 可以包括逻辑电路,用于预测分支结果和/或读取目标地址。

[0058] 定点单元 320A 和浮点单元 320B 仅代表可以存在于处理器单元 220 的实施方式内部的功能单元。这些单元可以包含用于执行各种类型的指令并生成结果的逻辑电路。例如,各种指令可以实施算法、逻辑或移位操作。在不同的实施方式中,单元 320 可以包括用于执行这样的操作的一个或多个算术逻辑单元(ALU)。

[0059] 加载保存单元 330 包含用于执行加载和保存指令的逻辑电路。在一个实施方式中,加载保存单元 330 可以包括用于将数据加载和保存到数据高速缓存器 345 的逻辑电路。在某些实施方式中,加载保存单元 330 可以实现逻辑电路,以应对数据高速缓存器缺失以及从虚拟地址至物理地址的转换(例如,加载保存单元 330 可以包括数据转换后备缓冲器(DTLB)或硬件来实施页表搜索)。

[0060] 协处理器接口单元 350 是将指令 352 提供给一个或多个协处理器单元 260 并且接收响应的执行结果 354 的功能单元。在不同的实施方式中,指令 352 可以包括由单元 350(例如从单元 310)接收的指示,以及由单元 350 生成的命令/指示。例如,协处理器接口单元 350 可以接收与从调度器单元 310 的指令流相关的一组指令 352,并且将所述一组指令 352 提供给协处理器单元 260,在该协处理器单元 260 处生成一组执行结果 354。在该实施例中,协处理器单元 260 正在实施来自处理器单元 220 的指令流的指令。此外,协处理器接口单元 350 可以将指令或者命令 352 提供给协处理器单元 260,所述指令或者命令 352 不是处理器单元 220 的指令流的一部分。这些指令/命令可用于控制协处理器单元 260 的操作和/或协处理器单元 260 相对于处理器单元 220 的同步,以及任何其他合适的目的。在任何情况下,一旦协处理器单元 260 生成了一组执行结果 354,协处理器接口单元 350 就接收一组执行结果 354。在一个实施方式中,单元 350 将执行结果 354 提供给引退单元 360。在下面会结合图 5 对协处理器接口单元 350 加以详细描述。

[0061] 在一个实施方式中,引退单元 360 包含用于使从单元 320、330 和 350 接收的执行结果引退的逻辑电路。在一些实施方式中,引退单元 360 可以耦合至(或者物理上包含)一个或多个用于保存状态信息的寄存器组,在所述寄存器组中,当收到执行结果时,引退单元 360 可以更新寄存器中的状态信息。在一个实施方式中,该状态信息可以包括识别执行中的线程(例如线程 ID)或所述执行中的线程的响应进程(例如进程 ID)的数据。在另一个实施方式中,状态信息可以包括在协处理器单元 260 上执行的线程的一个或多个状态信息。在不同的实施方式中,引退单元 360 可以包含如下逻辑电路,所述逻辑电路用于:当在无序执行(OoOE)之后接收到一组指令的时候,使指令组排队,并且在引退期间使所述指令组重新排序。在另一个实施方式中,引退单元 360 可以包含用于处理在执行不同指令期间引发的(thrown)异常(例如页错误异常)的俘获逻辑电路(trap logic)。

[0062] 现在看图 4, 示出的是协处理器单元 260。协处理器单元 260 是可以通过互连线 240 耦合到处理器单元 220 的协处理器单元的实施方式。在一个实施方式中, 互连线 240 可以是控制外围设备或者实施特定功能的处理电路的、单独的总线。如图所示, 协处理器单元 260 包括相伴处理器接口 420 和处理核 440, 它们通过互连线 430 耦合在一起。在一个实施方式中, 处理核 440 可以包括功能单元 442 和寄存器组 444。在另一个实施方式中, 处理核 440 可以耦合至存储器控制器 280 并且包括高速缓存器 446。相伴处理器接口 420 配置成接收指令 352, 并且将执行结果 354 传输给处理器单元 220 的协处理器接口单元 350。寄存器组 444、高速缓存器 446 和至存储器控制器 280 的连接用虚线示出, 以便指明协处理器单元 260 能够以各种方式实现 (例如, 在一些实施方式中, 可以根据需要将不同的框包括在内或排除在外)。要注意的是, 示出的协处理器单元 260 包括单个处理核 440 和单个功能单元 442, 而在不同的实施方式中, 协处理器单元 260 可以包括多数个处理核 440/ 或多数个功能单元 442。在其他实施方式中, 协处理器单元 260 可以被附加在额外的协处理器单元上。

[0063] 相伴处理器接口 420 是协处理器单元 260 与处理器单元 220 的协处理器接口单元 350 之间的接口。在一个实施方式中, 相伴处理器接口 420 可以配置成在执行之前将所接收到的指令保存在指令队列中, 并且将它们转送给处理核 440, 在该处理核处, 随后执行所述指令。在另一个实施方式中, 相伴处理器接口 420 可以配置成在由处理核 440 生成之后将执行结果 354 保存在另一队列中, 并且可以将所述执行结果 354 传输给协处理器接口单元 350。在一个实施方式中, 响应于处理器单元 220 通过协处理器接口单元 350 传输针对执行结果 354 的请求, 相伴处理器接口 420 可以将执行结果 354 传输给协处理器接口 350。在可供选择的实施方式中, 相伴处理器接口 420 可以在执行结果 354 可用后立即发起对所述执行结果的传输 (也就是说, 无需从处理器单元 220 接收这样做的明确指令。)

[0064] 处理核 440 配置成执行指令 352, 以便使用一个或多个功能单元 442 生成执行结果 354。指令 352 可以具有不同程度的复杂性, 并且具有不同的执行要求。在一个实施方式中, 处理核 440 可以执行不要求访问存储器或寄存器的指令 352 (例如只包含直接数据的指令)。在其他实施方式中, 处理核 440 可以在执行指令 352 时访问寄存器组 444。在一些实施方式中, 处理核 440 可以执行利用存储器控制器 280 来访问板载高速缓存器或存储器 120 的指令 (例如包括地址信息的指令, 如加载 / 保存指令)。在一些实施方式中, 处理核 440 可以在访问存储器 120 时将虚拟地址转换成物理地址。

[0065] 处理核 440 可以执行来自各种来源的指令。在一个实施方式中, 处理核 440 可以只执行由处理器单元 220 提供的指令 352。在另一个实施方式中, 处理核 440 可以在从处理器单元 220 接收一组初始指令 352 之后, 读取额外的指令。在不同的实施方式中, 处理核 440 可以在执行之前从指令 352 生成微代码指令。

[0066] 如上面提及的那样, 在一个实施方式中, 指令 352 可以包括是处理器单元 220 的指令流的一部分的一组指令。提供给协处理器单元 260 的指令可以例如包括协处理器操作码, 用于标明它们在单元 260 内部 (例如在特定的功能单元 442 上) 是可执行的。在另一种实施方式中, 指令 352 可以是由控制对协处理器单元 260 的执行的单元 350 生成的指令 / 命令 (即与处理器单元 220 的指令流不相关的指令)。例如, 在一个实施方式中, 这些命令可以是可执行的, 用于控制处理器单元 220 和协处理器单元 260 的通信 (例如调整相伴处理器接口 420 的方位)。在另一个例子中, 这些命令可以指导处理器核 440, 以执行各种操

作,例如改变(即更新或无效)所保存的(例如寄存器组 444 或高速缓存器 446 中的)数据、管理存储器访问(例如更新或者无效 TLB 中的转换)、保存执行线程的状态信息(例如扩展的保存命令(EXSAVE)等。

[0067] 执行结果 354 可以包括各种作为执行指令 352 的结果所产生的信息中的任何信息。在一个实施方式中,执行结果 354 可以包括所生成的合成值(例如,一组加密指令可以产生一组加密的数据)。在另一种实施方式中,执行结果 354 可以包括状态/完成通知信息(例如,指示出在处理核 440 上执行的特殊线程的完成)。在其他的实施方式中,执行结果 354 可以包括异常或故障信息(例如页错误指示)。在一些实施方式中,执行结果 354 可以包括用于控制对协处理器接口单元 350 的操作的命令(例如控制处理器单元 220 和协处理器单元 260 的通信的命令)。

[0068] 在下面会结合图 6A 至 6D 对协处理器单元 260 的各种实施方式加以详细描述。

[0069] 现在看图 5,示出的是协处理器接口单元 350。协处理器接口单元 350 是用于将处理器单元 220 耦合至协处理器单元 260 的协处理器接口单元的实施方式。如图所示,协处理器接口单元 350 包括调度器接口单元 510、路由接口单元 520、引退接口单元 530、协处理器命令单元 540 和寄存器组 550。在所描绘的实施方式中,路由接口单元 520 包括路由表 524。在单元 350 的各种实施方式中,寄存器组 550 和/或路由表 524 可以(或者可以不)被包括在内,如由它们各自的虚线所指示的那样。

[0070] 如上面指出的那样,协处理器接口单元 350 配置成将来自处理器单元 220 的指令流的指令提供给协处理器单元 260。在一个实施方式中,协处理器接口单元 350 通过调度器接口单元 510 接收指令,并且通过路由接口单元 520 将指令 352 传输至协处理器单元 260。一旦协处理器单元 260 执行完所述指令,路由接口单元 520 就接收执行结果 354,然后协处理器接口单元 350 通过引退接口单元 530 将执行结果提供给引退单元 360。

[0071] 在一个实施方式中,路由接口单元 520 是通过互连线 240 生成和接收总线流量(bus traffic)(例如指令/命令 352 和执行结果 354)的接口。在一些实施方式中,路由接口单元 520 采用基于数据包的通信协议,如超传输、PCI Express、快速路径(Quick Path)等。在某些实施方式中,路由接口单元 520 可以将路由信息保存在路由表 524 中,这可以保存针对多个协处理器单元 260 的源和目标地地址信息。

[0072] 在一个实施方式中,协处理器命令单元 540 配置成产生不是处理器单元 220 指令流的一部分的指令/命令。例如,协处理器命令单元 540 可以生成对协处理器单元 260 的操作进行控制的命令,如上面在图 4 的内容中所描述的那样。在一些实施方式中,协处理器命令单元 540 可以生成对互连线 240 和/或相伴处理器接口 420 的操作进行控制的命令(例如调整超传输通信参数的命令等)。在不同的实施方式中,协处理器命令单元 540 可以生成响应于外部指示(例如调度器单元 310、引退单元 360 等)的命令。在其他实施方式中,协处理器命令单元 540 可以分析与调度器接口单元 510 和/或引退接口单元 530 相关的流量,以确定是否生成命令。

[0073] 在不同的实施方式中,协处理器接口单元 350 包括寄存器组 550,用于保存与单元 350 的各种操作相关的信息。例如,协处理器接口单元 350 可以保存与具有响应的协处理器线程的、处理器单元 220 上的线程相关的信息。在一些实施方式中,协处理器接口单元 350 保存与不同协处理器 260 相关的信息,例如与执行协处理器线程相关的信息、针对特殊指

令组 352 的完成状态信息等。

[0074] 在不同实施方式中,协处理器接口单元 350(例如在寄存器组 550 和 / 或存储器 120 中)储存如下信息,所述信息涉及某些协处理器单元 260(例如在 CPUID 树中)的不同性能 / 功能。在一些实施方式中,协处理器接口单元 350 也可以配置成引发 (throw) 如下事件中的异常:任何来自处理器单元 220 的指令流试图以无效的方式查询协处理器单元 260(例如查询不存在的协处理器单元 260、查询协处理器单元 260 的不可用的特征)。

[0075] 现在看图 6A 至图 6D,一系列框图展现的是协处理器单元 260 的各种实施方式。

[0076] 图 6A 图示说明了实现多线程处理核的协处理器单元 260 的实施方式。如图所示,协处理器单元 260 包括通过互连线 430 耦合的相伴处理器接口 420 和处理核 440。在一个实施方式中,处理核 440 包括功能单元 610A、功能单元 610B 和寄存器组 620。在一些实施方式中,每个功能单元 610 都配置成针对给定的线程执行指令 352,并且生成一组响应的执行结果 354。在某些实施方式中,寄存器组 620 可以保存表示特殊线程的信息(例如线程标识、进程标识 PID、内容标识 CID 等)。所示的协处理器单元 260 带有具有多个功能单元 610 的单个处理核 440,而在其他实施方式中,协处理器单元 260 包括含有多数个功能单元 610 的多数个处理核 440。

[0077] 如前面提及的那样,在处理器单元 220 上执行的线程可以调用 (invoke) 在协处理器单元 260 上执行的特殊线程和 / 或与在协处理器单元 260 上执行的特殊线程通信。例如,在一个实施方式中,只有一个单独的协处理器线程可以与响应的单独的处理器线程配对。在另一个实施方式中,多个协处理器线程可以与响应的单独的处理器线程配对。在一些实施方式中,单独的处理器线程可以配对成在多个功能单元 610 和 / 或多个协处理器单元 260 上执行的响应的协处理器线程。

[0078] 在不同的实施方式中,协处理器接口单元 350 提供控制响应的协处理器线程的执行的指令 / 命令(例如通过由协处理器命令单元 540 生成的指令 / 命令)。例如,这些指令 / 命令可以包括用于中止正在协处理器单元 260 上执行的线程或者恢复执行先前正在协处理器单元 260 上执行的线程的指令。在不同的实施方式中,在处理器单元 220 上执行的操作使用指令 352 将新的协处理器线程初始化。在一些实施方式中,指令 352 可以包括用于执行内容切换 (context switch) 的命令,其中协处理器单元 260 中止执行中的线程并且切换到另一个非执行中的线程。在这种实施例中,协处理器单元 260 可以执行指令,以保存执行中的线程的状态和复原非执行中的线程的状态(例如使用 EXSAVE 指令)。在某些实施方式中,协处理器单元 260 将执行中的线程的所保存的状态从寄存器组 620 传送至存储器 120。在一个实施方式中,协处理器单元 260 可以直接访问存储器 120,以存储和检索 (retrieve) 状态信息。在另一个实施方式中,协处理器单元 260 可以通过处理器单元 220 存储和取回状态信息。在下面会结合图 8 对线程切换的方法加以详细描述。

[0079] 在一些实施方式中,协处理器单元 260 可以通过执行结果 354 控制响应的处理器线程的执行。在一个实施方式中,执行结果 354 可以包括完成状态信息,所述完成状态信息用于恢复和 / 或中止响应的处理器线程(例如,处理器线程可以阻止未完成的响应的执行中的协处理器线程)。在其他实施方式中,执行结果 354 可以包括用于恢复或中止处理器线程的明确的命令(例如,当在处理器单元 220 上执行内容切换时)。

[0080] 图 6B 图示说明了执行微代码指令的协处理器单元 260 的实施方式。如图所示,协

处理器单元 260 包括通过互连线 430 耦合的相伴处理器接口 420 和处理核 440。在所示的实施方式中,处理核 440 包括微代码汇编编译器 630 和功能单元 610。在一般情况下,微代码汇编编译器 630 配置成接收一组指令(例如指令 352),并生成能在功能单元 610 上执行的一组响应的微代码指令。微代码汇编编译器 630 用虚线示出,用以表明在不同实施方式中它可以(或者可以不)包含在协处理器单元 260 内部。例如,微代码汇编编译器 630 可以位于处理器单元 220 内部(例如位于协处理器接口 350 内部),以便指令 352 包括从处理器单元 220 的指令流中产生的微代码指令。

[0081] 图 6C 图示说明了实现虚拟存储器的协处理器单元 260 的实施方式。如图所示,协处理器单元 260 包括通过互连线 430 耦合的相伴处理器接口 420 和处理核 440。在所示的实施方式中,处理核 440 包括读取单元 640、功能单元 610、加载保存单元(LSU)650 和存储器管理单元(MMU)660。此外,读取单元 640 和 LSU650 各自包括指令转换后备缓冲器(ITLB)642 和数据转换后备缓冲器(DTLB)。在不同的实施方式中,可以(或者可以不)实现框的其他组合,例如,可以省略读取单元 640 和 / 或 LSU650,可以采用不同于 ITLB642 和 DTLB652 的单个 TLB,等等。

[0082] 在不同的实施方式中,协处理器单元 260 可以配置成通过读取额外的指令或者执行加载 / 存储指令来访问存储器。在一个实施方式中,读取单元 640、LSU650 各自检索高速缓存在 ITLB642 和 DTLB652 中的虚拟存储器转换(translation)。如果在适当的 TLB 中没有找到转换,那么在一些实施方式中,存储器管理单元(MMU)660 配置为针对所述转换实施页表搜索。在某些实施方式中,MMU660 查询 CR3 寄存器 662,以便在实施页表搜索之前获取当前页表的存储器地址。如果在页表中没有找到所述转换,那么在一些实施方式中,MMU660 针对在处理器 220 上执行的操作系统引发页错误异常。在下面会结合图 9 对访问存储器和实施页表搜索的方法加以详细描述。

[0083] 如上面提及的那样,在不同的实施方式中,处理器单元 220 和协处理器单元 260 配置成维持对存储器的统一查看。就这点而论,指令 352 和执行结果 354 可以包括管理存储器一致性的命令。在一个实施方式中,指令 352 可以包括用于改变(例如更新 / 无效)高速缓存的数据的命令,所述数据如高速缓存在 ITLB642 和 DTLB652 中的译文。在其他实施方式中,指令 352 可以包括用于更新 CR3 寄存器 662 的命令。同样地,在不同的实施方式中,执行结果 354 可以包括用于更新或者无效处理器 220 中(例如在指令高速缓存器 340、数据高速缓存器 345、任何 TLB 等的内部)的高速缓存的数据的命令。

[0084] 图 6D 图示说明了实现图形引擎(graphics engine)的协处理器单元 260 的实施方式。如图所示,协处理器单元 260 包括通过互连线 430 耦合的相伴处理器接口 420 和处理核 440。此外,处理核 440 包括图形编译器前端(graphical frontend)670 和存储器管理单元(MMU)660。在所示的实施方式中,图形编译器前端 670 耦合至图形编译器后端 692,相应地,该图形编译器后端耦合至显示单元 694。如虚线所示,MMU660 可以访问存储器 120(例如通过存储器控制器 280)和 / 或本地存储器 682。处理核 440 包括图形编译器前端 670,而在一个实施方式中,处理核 440 可以包括图形编译器后端 692。可供选择地,图形编译器后端 692 可以位于其他地方(例如图形卡或显示单元 694 中)。

[0085] 在不同的实施方式中,协处理器单元 260 执行来自处理器 220 的指令流的专门的图形指令。在一般情况下,图形编译器前端 670 实施在图像处理之前发生的各种图形操作,

例如多边形生成、光照效果,或者其他现有技术公知的图形操作。在一些实施方式中,图形编译器前端 760 可以包括指令汇编编译器 672,所述指令汇编编译器 672 生成用于响应的图形编译器后端 692 的图形指令。

[0086] 现在看图 7,描绘了方法 700 的流程图。方法 700 是由处理器单元 220 实现的、用于在协处理器单元 260 上执行指令的方法的一个实施方式。方法 700 以步骤 710 开始,所述步骤 710 包括处理器单元 220 读取、解码和调度一组指令。此时,在不同的实施方式中,调度器单元 310 解码指令,并将它们路由到各自的功能单元(例如单元 320、330 和 350)。在一个实施方式中,当指令包含与协处理器指令相关的特定操作码时,调度器单元 310 将所述指令路由至协处理器接口单元 350。在可供选择的实施方式中,可以基于其他标准,例如特定范围的存储器地址、PID 等,路由指令。在步骤 720 中,协处理器接口单元 350 将指令组 352 传输至协处理器单元 260(例如使用超传输或 PCI-Express 协议通过耦合到协处理器单元 260 上的北桥型装置)。在那里,协处理器单元 260 执行指令 352,以产生一组响应的执行结果 354。在步骤 730 中,协处理器接口单元 350 通过协处理器单元所发起的传输来接收执行结果 354,并且在步骤 750 中,例如通过引退单元 360 使相应的指令引退。在各种实施方式中,引退单元 360 可以存储执行结果 354 的合成值,并处理任何引发的异常。

[0087] 现在看图 8,描绘了方法 800 的流程图。方法 800 是用于在协处理器单元 260 上切换线程的方法的实施方式。例如,当在处理器单元 220 上执行的处理器线程对新的线程进行初始化或者恢复执行在协处理器单元 260 存在的线程时,可以实施步骤 810 至 840。如果所述处理器线程使未完成的新的/存在的线程中止,那么可以实施步骤 850 至 860,以恢复所述处理器线程。需要注意的是,在不同的实施方式中,能够以各种方式实施方法 800(例如,方法 800 可以不包括实施步骤 850 至 860,可以包括额外的未示出的步骤等)。

[0088] 方法 800 以步骤 810 开始,其中在处理器单元 220 上执行的线程需要使用相应的协处理器线程。在一个实施方式中,执行中的处理器线程指出这一要求,方法是:通过针对操作系统引发异常来调度新的或存在的线程。

[0089] 在步骤 820 中,处理器单元 220 确定是否存在可用的协处理器资源。在一个实施方式中,处理器单元 220 通过访问协处理器接口单元 350(例如寄存器组 550)中的信息来确定这种可用性。在另一实施方式中,处理器单元 220 通过协处理器命令来询问协处理器单元 260。如果协处理器资源可用,那么方法 800 继续进行步骤 840。否则,方法 800 继续进行步骤 830。

[0090] 在步骤 830 中,处理器单元 220 指令协处理器单元 260,以中止执行中的协处理器线程。在不同的实施方式中,协处理器单元 260 可以存储被中止的所述线程的状态信息(例如通过直接将所述状态存储在存储器 120 中,或者将所述状态传输给处理器单元 220 用于存储)。

[0091] 在步骤 840 中,处理器单元 220 的操作系统为了在协处理器单元 260 上执行而调度所请求的线程。如果所请求的线程先前已经在执行中,那么协处理器单元 260 可以加载与线程相关的任何先前的状态信息。

[0092] 一旦完成所请求的协处理器线程的执行,协处理器单元 260 就可以产生指明这种完成的执行结果 354。如果处理器线程在步骤 850 中受阻(即处理器线程阻碍了未完成的协处理器线程),那么处理器单元 220 在步骤 860 中恢复执行所述处理器线程。需要注意的

是,在一些实施方式中,协处理器单元 260 可以直接指令操作系统恢复所述处理器线程(例如通过在完成执行响应的协处理器线程时引发异常)。

[0093] 现在看图 9,描述了方法 900 的流程图。方法 900 是一种采用虚拟地址空间由协处理器单元 260 实现的方法的实施方式。

[0094] 方法 900 以步骤 910 开始,其中协处理器单元 260 生成将虚拟地址转换成物理地址 910 的请求。例如,当读取指令或访问存储器时,协处理器单元 260 可以实施虚拟存储器转换。

[0095] 在步骤 920 中,协处理器单元 260 确定:在协处理器 TLB(例如 ITLB642 或 DTLB 652)中是否发现(即存在)虚拟地址。如果在 TLB 中发现虚拟地址,那么所述转换被用来在步骤 930 中确定相应的物理地址。但是,如果未在 TLB 中发现虚拟地址,那么方法 900 继续进行步骤 940。

[0096] 在步骤 940 中,协处理器单元 260 实施页表搜索(例如使用 MMU660),以便寻找相应的转换。如果在步骤 950 中,在页表中发现虚拟地址,那么协处理器单元 260 可将所述转换加载到 TLB 中,并在步骤 960 中确定相应的物理地址。另一方面,如果在步骤 950 中,在页表中未发现虚拟地址,那么方法 900 继续进行步骤 970。

[0097] 在步骤 970 中,协处理器单元 260 用信号将页错误传达给处理器单元 220。在一个实施方式中,协处理器单元 260 生成如下执行结果 354,所述执行结果 354 包括可由在处理器单元 220 上执行的操作系统使用的页错误异常。在一些实施方式中,当操作系统已对页错误进行了维护时,协处理器接口单元 350 可以告知协处理器单元 260(例如通过由协处理器命令单元 540 生成的命令)。

[0098] 现在看图 10,描述了方法 1000 的流程图。方法 1000 是由第一处理器功能单元(例如协处理器接口单元 350)实现的方法的一个实施方式。在步骤 1010 中,第一处理器功能单元在来自调度器单元 310 的第一指令流中接收第一组指令(例如指令 352)。在步骤 1020 中,第一处理器功能单元将所述第一组指令提供给协处理器单元(例如单元 260)。然后,协处理器单元 260 可以执行指令 352,生成一组响应的执行结果 354。在步骤 1030 中,第一处理器功能单元接收响应于来自协处理器单元 260 的所述第一组指令的第一组执行结果(例如执行结果 354),其中所述协处理器单元 260 发起所述传输。

[0099] 现在看图 11,描绘了方法 1100 的流程图。方法 1100 是由协处理器单元 260 实现的方法的一个实施方式。在步骤 1110 中,协处理器单元 260 在相伴处理器接口单元(例如相伴处理器接口 420)处,接收来自相伴处理器单元(例如处理器单元 220)中的协处理器接口单元(例如协处理器接口单元 350)的指令 352。在步骤 1120 中,协处理器单元 260 在处理核 440 上执行所收到的指令 352,以生成第一组执行结果 354。在步骤 1130 中,协处理器单元 260 发起所述第一组执行结果 354 通过相伴处理器接口单元(例如相伴处理器接口 420)至所述相伴处理器单元(例如处理器单元 220)的传送。

[0100] 现在看图 12,描述了计算机可读介质 1210。计算机可读介质 1210 是存储如下指令的产品的实施方式,所述指令能由包括处理器单元 220 和协处理器单元 260 的计算机系统 100 执行。如图所示,计算机可读介质 1210 包括能由处理器单元 220 内部的一个或多个功能单元执行的一组处理器指令 1221 和由协处理器单元 260 执行的一组协处理器指令 1222。在所示的实施方式中,组 1221 中的每个处理器指令 1231 都包括处理器操作码位 1241,而组

1222 中的每个协处理器指令 1232 都包括协处理器操作码位 1242。在不同的实施方式中,指令 1231 和 1232 可以包括地址位 1244 和 / 或直接数据 1246。(指令 1231 和 1232 的不同于已述那些情况的其他布置当然也是有可能的。例如,所有指令都无需包括直接数据或地址位。)

[0101] 调度器单元 310 可以使用不同标准来调度 / 路由指令。在一个实施方式中,调度器 310 基于操作码位路由指令(例如,处理器指令 1231 包括处理器操作码位 1241,而协处理器指令 1232 包括协处理器操作码位 1242)。在另一种实施方式中,调度器 310 基于地址位 1244 路由指令(例如,某些地址与处理器指令相关,而另一些则与协处理器指令相关)。在其他实施方式中,调度器 310 基于其他标准,例如程序 1220 之内的指令的顺序,来调度用于执行的指令。

[0102] 计算机可读介质 1210 可以包括各种介质,所述介质存储在执行期间使用的程序指令和 / 或数据。在一个实施方式中,计算机可读介质 1210 是一种产品,例如有形的计算机可读的存储介质,其可以包括计算机系统 100 的存储器子系统的不同部分。在其他实施方式中,有形的计算机可读介质 1210 可以包括存储介质或存储器介质,如磁介质(例如磁盘)或光学介质(例如 CD、DVD 及相关技术等)。有形的计算机可读介质 1210 可以是易失性的或非易失性的存储器。例如,有形的计算机可读介质 1210 可以是(但不限于)SDRAM、DDR SDRAM、RDRAM、SRAM、闪速存储器,并且具有各种类型的 ROM,等等。图 1 例如包括有形的计算机可读介质(例如存储器 120),其相应地能够用于存储能由处理器单元 220 或协处理器单元 260 执行的程序指令。

[0103] 在其他实施方式中,计算机可读介质 1210 是非有形的介质,其可以对信号(例如电信号、电磁信号或光信号)上的指令和 / 或数据进行存储 / 编码,这些信号通过通信介质、链路和 / 或系统(例如电缆、网络等)无论有线地、无线地还是两者兼而有之地进行传送。这样的信号可以根据上述说明运载实现的指令和 / 或数据。例如,能在计算机系统 100 上执行的指令可以传输到遍及不同类型的计算机网络。

[0104] ***

[0105] 虽然上面已经描述了具体实施方式,但这些实施方式并非意在限制本公开文本的保护范围,即使其中关于特定特征仅描述了一个单一的实施方式。本公开文本中的特征的示例意在进行说明而不是限制,除非另作说明。上面的描述意在覆盖对于从本公开文本中得到启示的本领域技术人员来说显而易见的替换、修改和等同。

[0106] 本公开文本的保护范围包括在这里(明确地或暗示地)公开的特征中的任何特征或者其组合,或者对其的任何概括,不管其是否有利于解决在这里论述的问题中的任何之一或者全部。因此,可能在本申请的申请过程中针对特征的任何组合提出新的权利要求(或对其主张优先权的请求)。尤其是,参考附加的权利要求,从属权利要求的特征能够与独立权利要求的特征组合,而相应的独立权利要求的特征能够以任何合适的方式组合并且不仅仅以在附加的权利要求中列举的具体组合方式来组合。

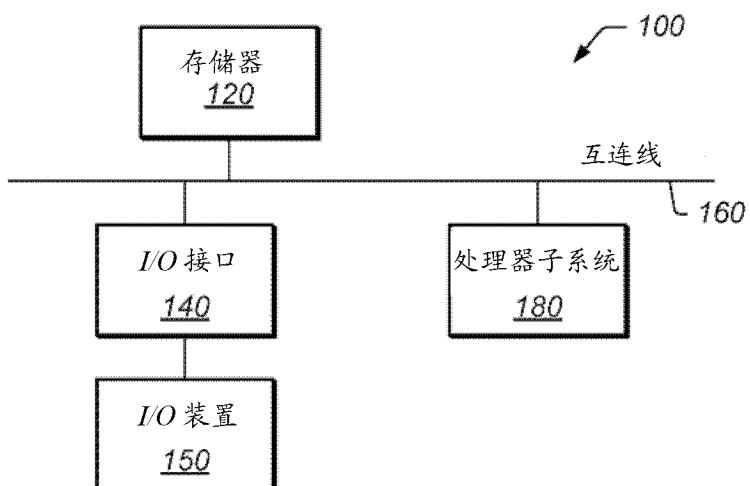


图 1

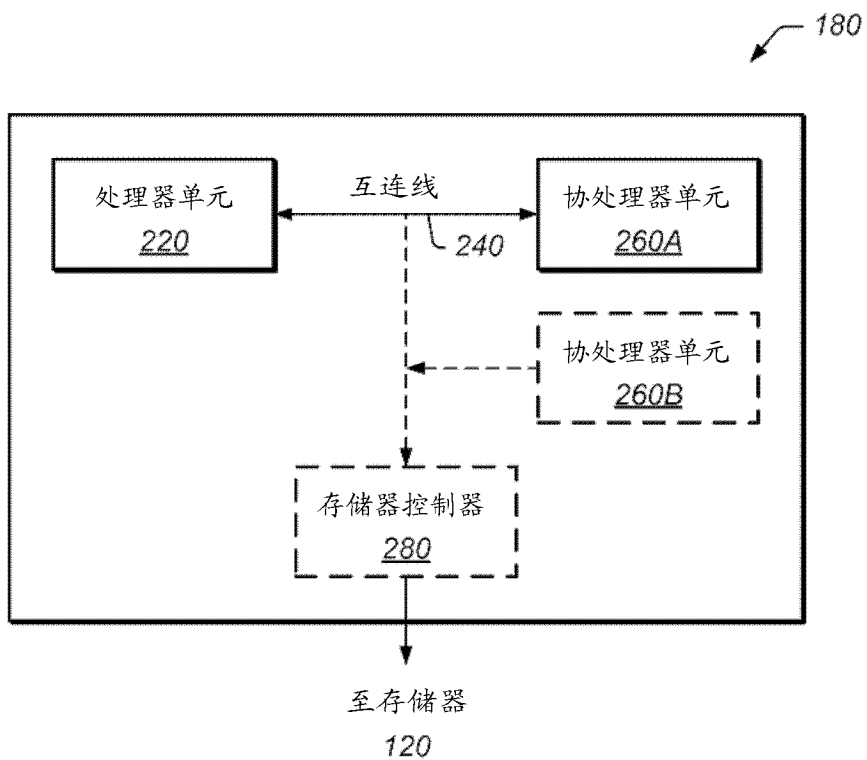


图 2

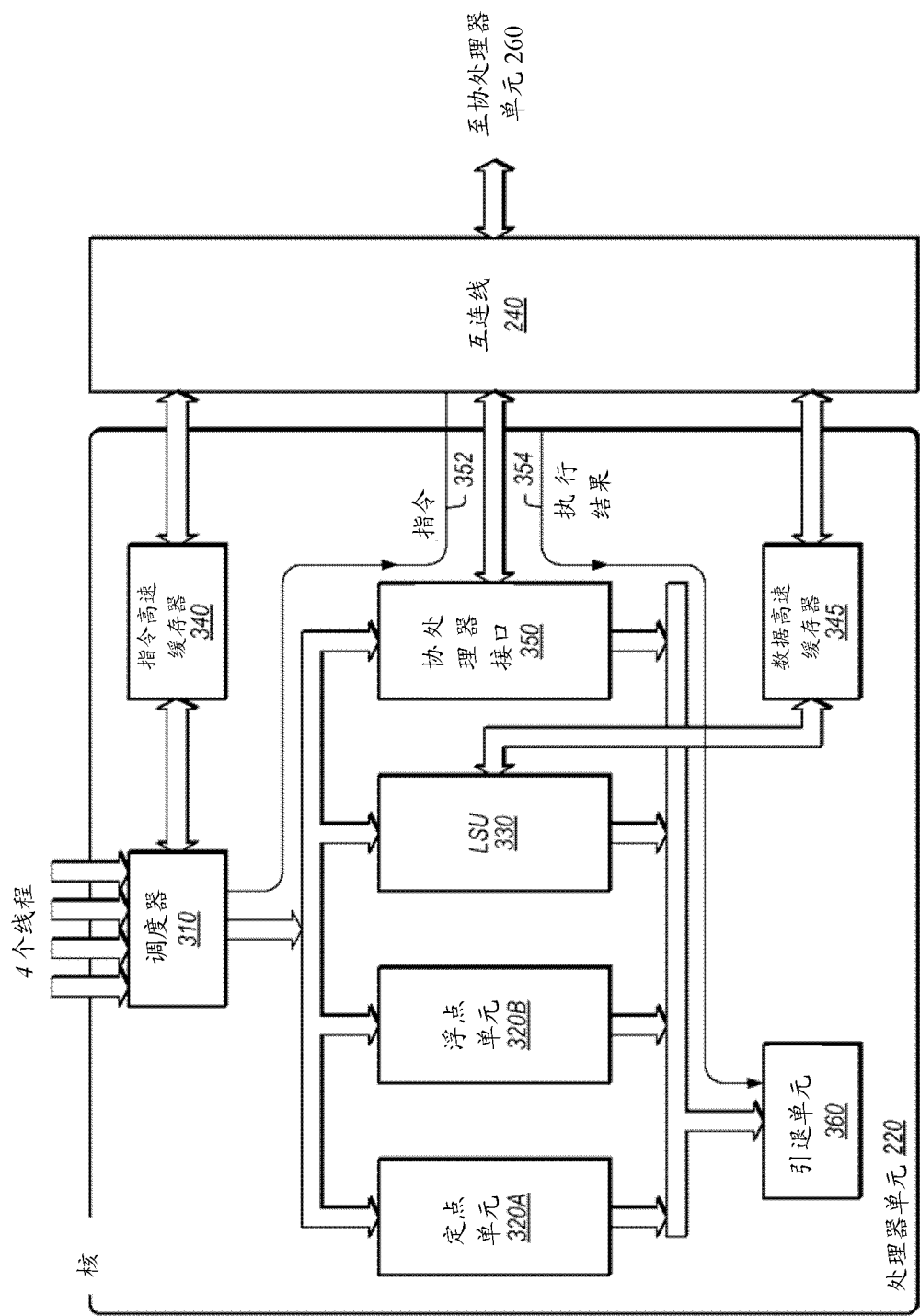


图 3

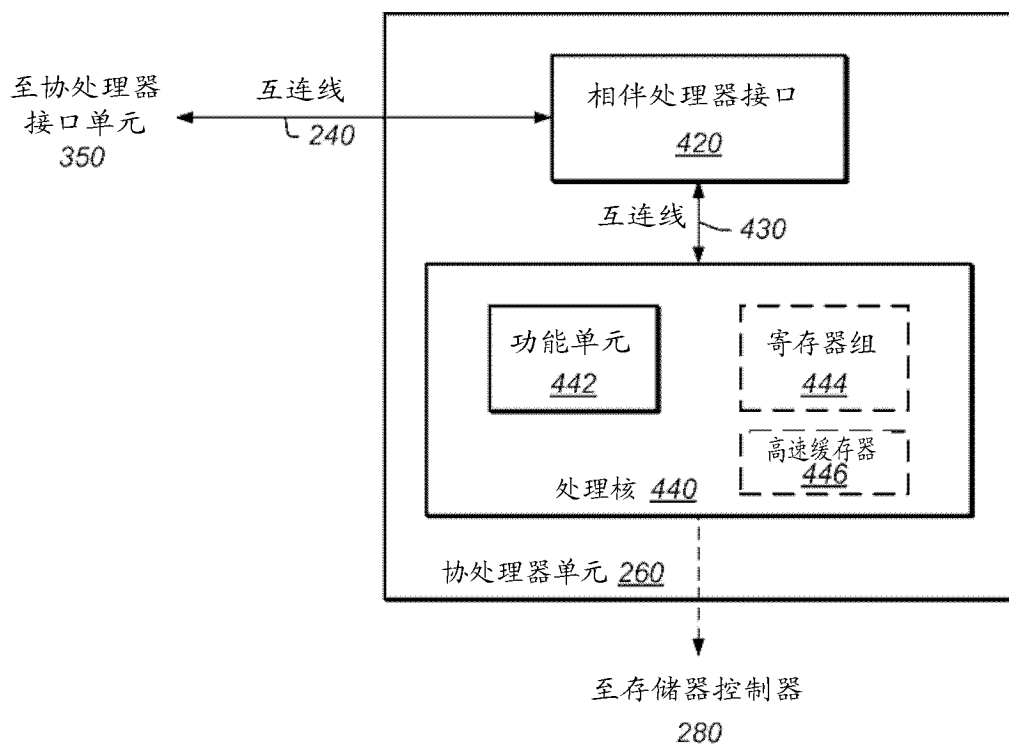


图 4

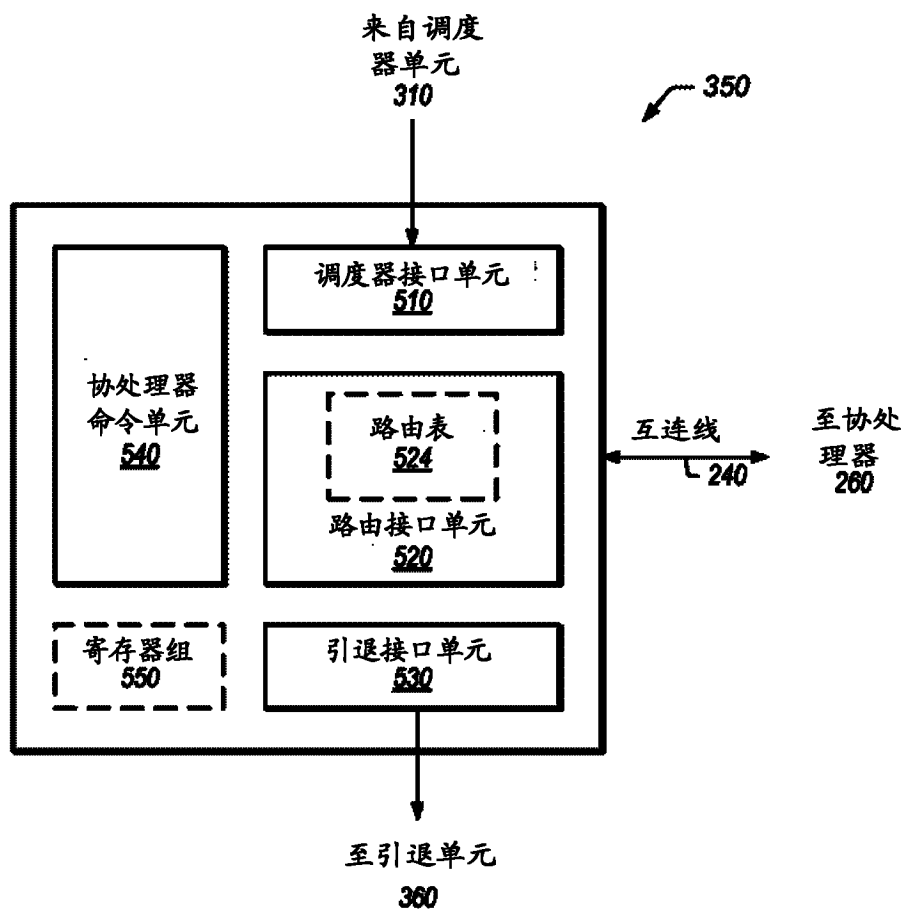


图 5

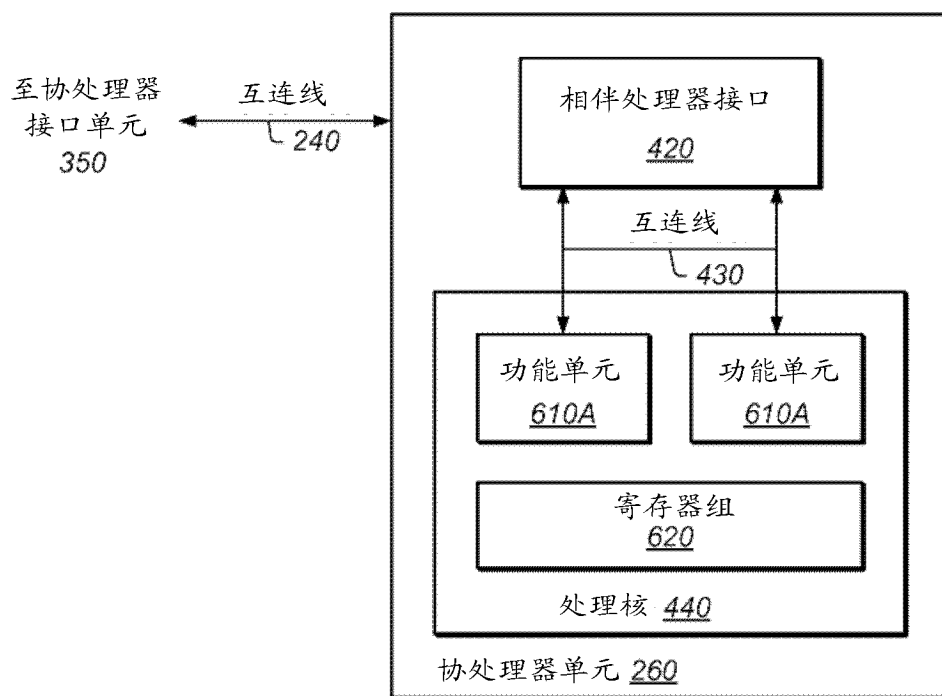


图 6A

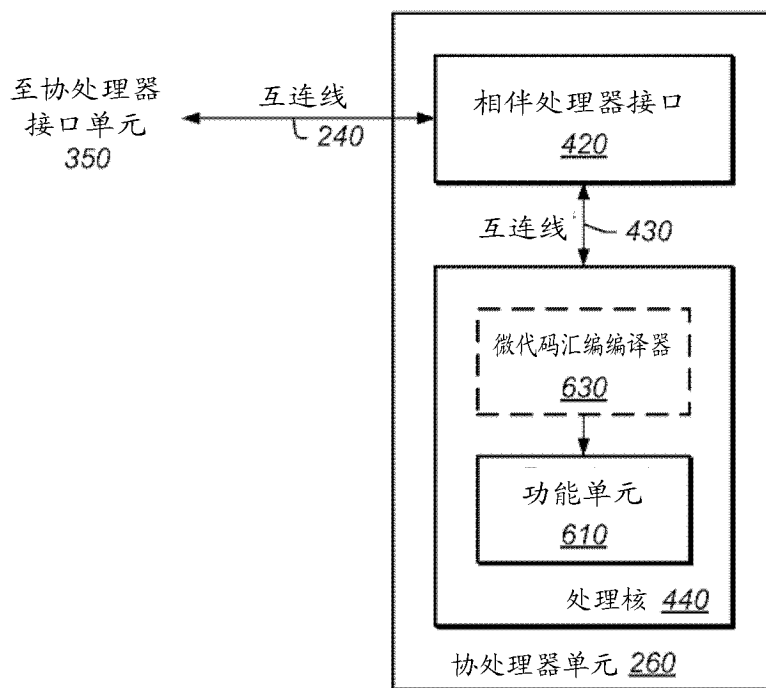


图 6B

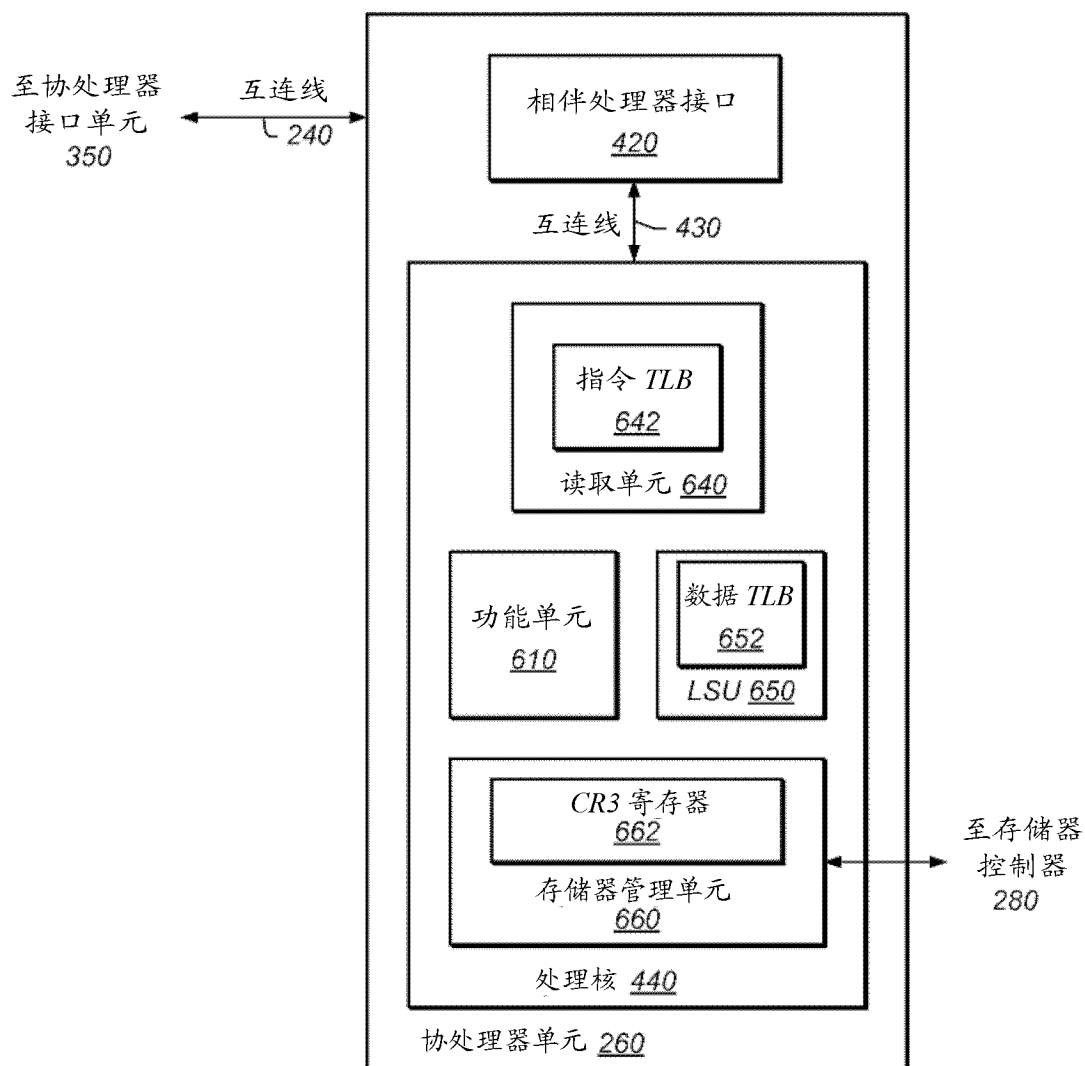


图 6C

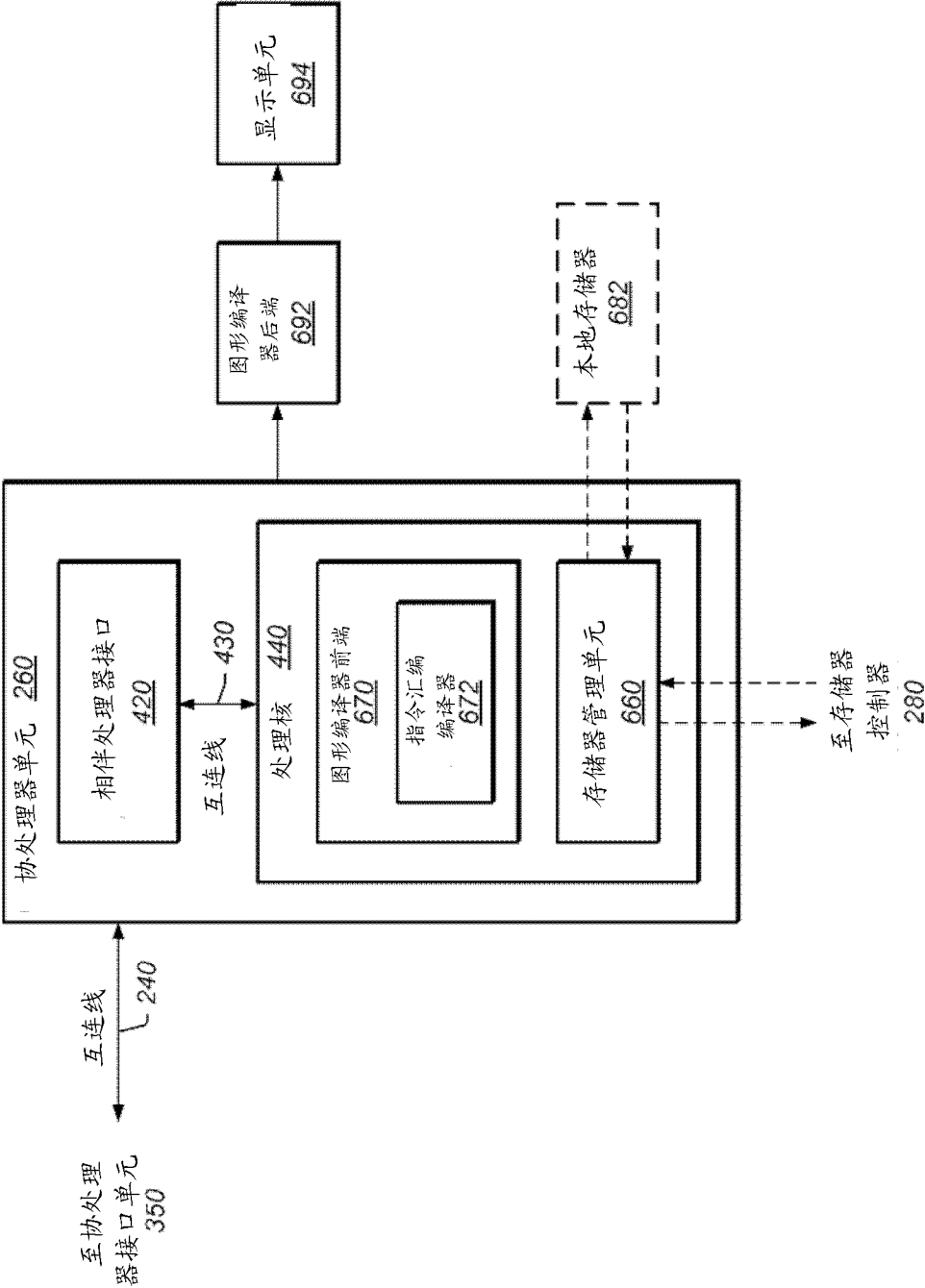


图 6D

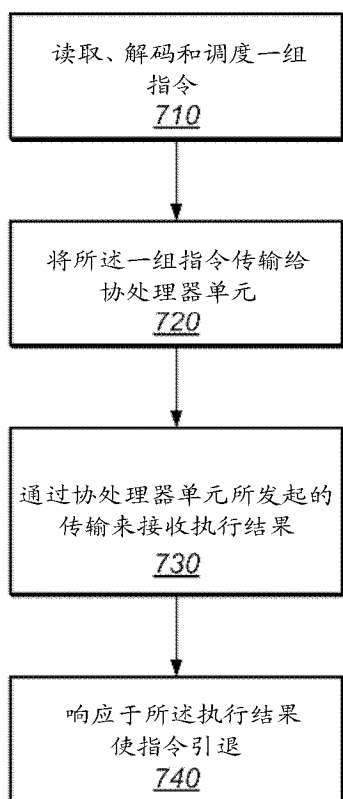


图 7

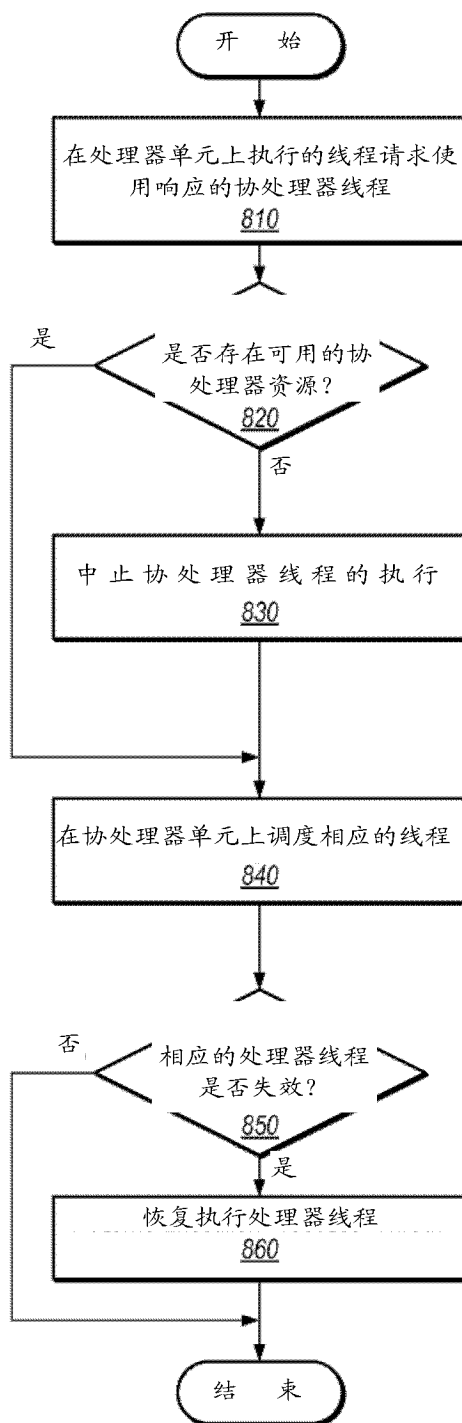


图 8

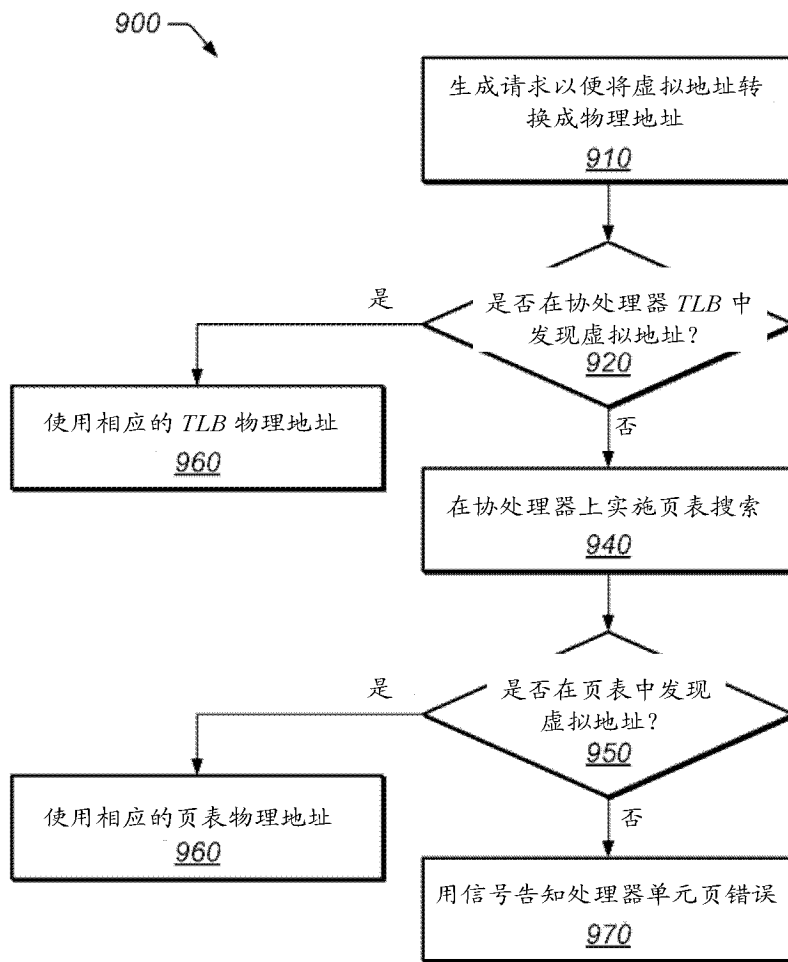


图 9

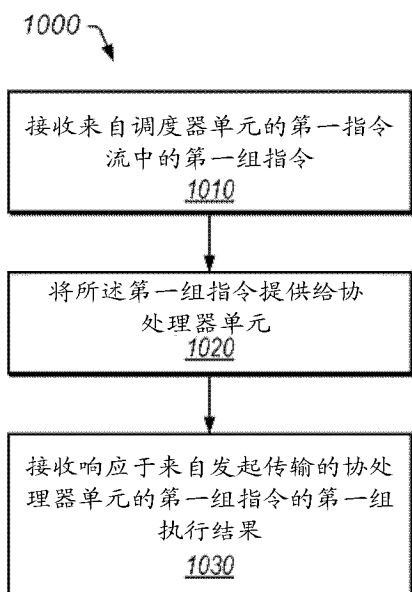


图 10

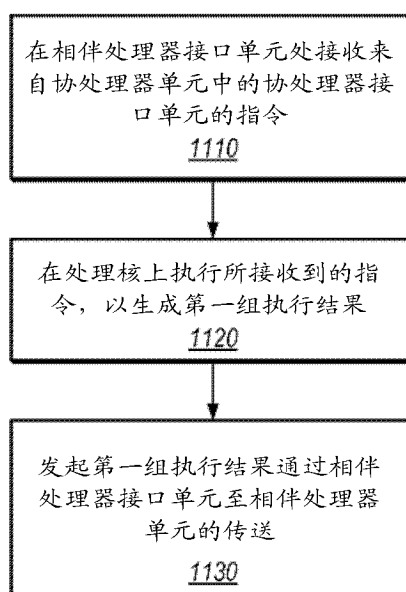


图 11

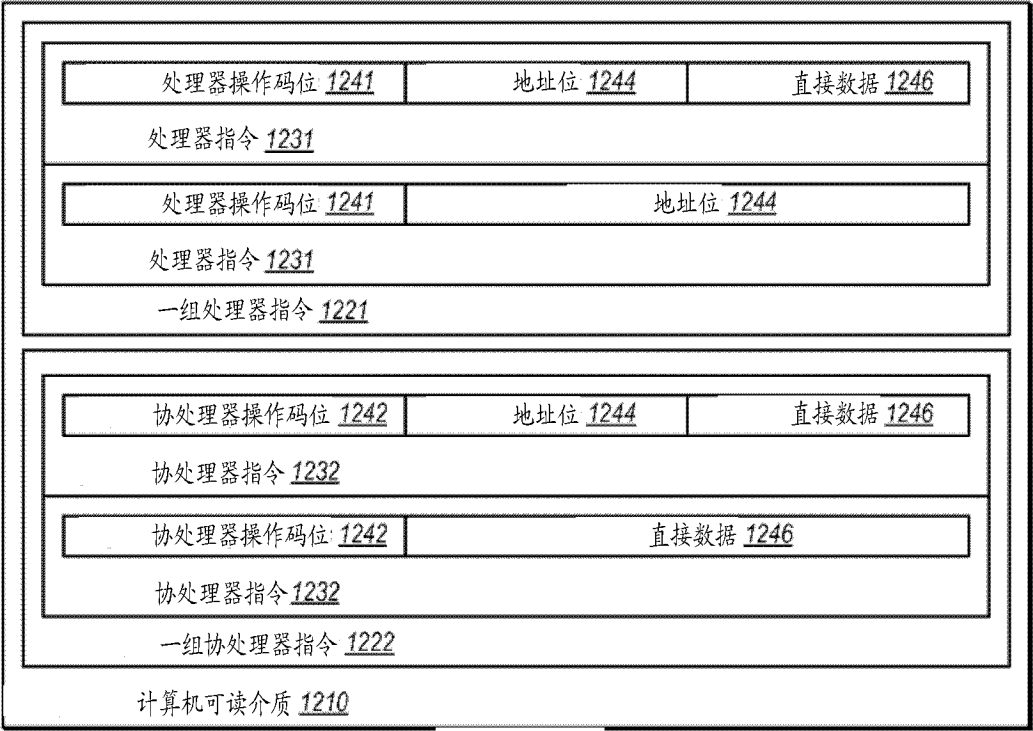


图 12